

# MANAJEMEN PROYEK SISTEM INFORMASI



## DAFTAR ISI

|  |           |
|--|-----------|
| DAFTAR ISI .....   | i         |
| DAFTAR GAMBAR.....   | v         |
| DAFTAR TABEL .....   | vi        |
| <b>BAB I PENDAHULUAN .....</b>   | <b>1</b>  |
| <b>1.1 Definisi Sistem Informasi.....</b>  | <b>1</b>  |
| <b>1.2 Sistem Informasi Dan Sejarahnya.....</b>                                      | <b>1</b>  |
| <b>1.3 Sistem Informasi Dan Software .....</b>                                       | <b>2</b>  |
| <b>1.4 Sistem Informasi Dan Teknologi Informasi.....</b>                             | <b>3</b>  |
| <b>1.5 Sistem Dengan Perencanaan .....</b>   | <b>3</b>  |
| <b>1.6 Definisi Proyek.....</b>  | <b>4</b>  |
| <b>1.7 Manajemen Proyek .....</b>  | <b>5</b>  |
| <b>1.8 Siklus Pengembangan Software (SDLC-Software Development Life Cycle) .....</b> | <b>5</b>  |
| <b>1.9 Pemrograman .....</b>   | <b>6</b>  |
| <b>1.10 Database .....</b>   | <b>6</b>  |
| <b>BAB 2 MANAJEMEN PROYEK DALAM PRAKTEK .....</b>                                    | <b>8</b>  |
| <b>2.1 Stakeholder .....</b>   | <b>8</b>  |
| <b>2.1.1 Pemilik Proyek.....</b>   | <b>8</b>  |
| <b>2.1.2 Komite Pengarah.....</b>  | <b>8</b>  |
| <b>2.1.3 Pengguna Hasil Proyek.....</b>  | <b>9</b>  |
| <b>2.1.4 Pelaksana Proyek .....</b>  | <b>9</b>  |
| <b>2.2 Metodologi dalam Manajemen Proyek .....</b>                                   | <b>12</b> |
| <b>2.2.1 Rational Unified Process .....</b>  | <b>13</b> |
| <b>2.2.2 Extreme Project Management .....</b>  | <b>13</b> |
| <b>2.2.3 Agile Project Management.....</b>   | <b>15</b> |
| <b>2.2.4 Metodologi Tradisional.....</b>   | <b>16</b> |
| <b>2.3 Alokasi Sumberdaya .....</b>  | <b>19</b> |
| <b>2.4 Subproyek .....</b>   | <b>20</b> |
| <b>2.5 Penjadwalan.....</b>  | <b>21</b> |
| <b>2.6 Kegagalan Proyek Sistem Informasi .....</b>                                   | <b>22</b> |

|                |  |           |
|----------------|--|-----------|
| 2.7            | <i>Project Management Office (PMO)</i> .....                       | 27        |
| <b>BAB III</b> | <b>PERAN MANAJER PROYEK</b> .....                                  | <b>30</b> |
| 3.1            | <b>Apa yang Dilakukan oleh Seorang Manajer Proyek?</b> .....       | <b>30</b> |
| 3.2            | <b>Keahlian yang Dibutuhkan</b> .....                              | <b>31</b> |
| 3.2.1          | <b>Keahlian Personal</b> .....                                     | <b>31</b> |
| 3.2.2          | <b>Keahlian Teknis</b> .....                                       | <b>32</b> |
| 3.2.3          | <b>Keahlian Manajemen</b> .....                                    | <b>34</b> |
| 3.2.4          | <b>Keahlian Menangani Situasi</b> .....                            | <b>34</b> |
| 3.3            | <b>Kualifikasi Manajer Proyek</b> .....                            | <b>34</b> |
| 3.4            | <b>Tugas dan Tanggung Jawab</b> .....                              | <b>35</b> |
| 3.5            | <b>Kualitas Manajer Proyek</b> .....                               | <b>37</b> |
| 3.6            | <b>Manajer Proyek Sebagai Pemimpin Visioner</b> .....              | <b>39</b> |
| <b>BAB IV</b>  | <b>FASE INISIASI</b> .....   | <b>40</b> |
| 4.1            | <i>Feasibility Study</i> .....                                     | <b>40</b> |
| 4.2            | <i>Requirements Analysis</i> .....                                 | <b>42</b> |
| 4.2.1          | <b>Definisi</b> .....  | <b>42</b> |
| 4.2.2          | <b>Jenis-jenis</b> .....   | <b>43</b> |
| 4.2.3          | <b>Menyusun <i>System Specifications (SRS)</i></b> .....           | <b>43</b> |
| 4.2.4          | <b>Perubahan dan Manajemen</b> .....                               | <b>45</b> |
| 4.2.5          | <b><i>Requirements</i> Sebagai Penyebab Kegagalan Proyek</b> ..... | <b>47</b> |
| 4.3            | <i>Project Scope Document</i> .....                                | <b>49</b> |
| 4.4            | <b>Penyusunan Tim</b> .....  | <b>50</b> |
| 4.5            | <b>Manajemen Risiko</b> .....                                      | <b>51</b> |
| 4.6            | <b>Proposal</b> .....  | <b>53</b> |
| 4.6.1          | <b>Pengertian Proposal</b> .....                                   | <b>53</b> |
| 4.6.2          | <b>Struktur Proposal</b> .....                                     | <b>54</b> |
| 4.6.3          | <b>Presentasi dan Revisi</b> .....                                 | <b>55</b> |
| 4.7            | <b>Kontrak/Surat Perintah Kerja (SPK)</b> .....                    | <b>56</b> |
| 4.8            | <i>Project Charter</i> .....                                       | <b>56</b> |
| 4.9            | <i>Project Kick-Off</i> .....                                      | <b>57</b> |
| <b>BAB V</b>   | <b>FASE PERENCANAAN</b> .....                                      | <b>59</b> |
| 5.1            | <b>Rencana Proyek</b> .....  | <b>59</b> |

|                                       |  |    |
|---------------------------------------|--|----|
| 5.1.1                                 | <i>Work Breakdown Structure</i> .....          | 59 |
| 5.1.2                                 | Estimasi.....                                  | 61 |
| 5.2                                   | Rencana Sumber Daya Manusia.....               | 62 |
| 5.3                                   | Rencana Keuangan .....                         | 67 |
| 5.4                                   | Rencana Kualitas.....                          | 68 |
| 5.5                                   | Rencana Risiko .....                           | 68 |
| 5.6                                   | Rencana Penerimaan .....                       | 69 |
| 5.7                                   | Rencana Komunikasi.....                        | 69 |
| 5.8                                   | Rencana Pengadaan .....                        | 70 |
| 5.9                                   | Tinjauan Perencanaan Proyek.....               | 70 |
| <b>BAB VI FASE PENGEMBANGAN</b> ..... |  | 71 |
| 6.1                                   | Requirements.....                              | 71 |
| 6.2                                   | Desain Sistem Dan Software.....                | 71 |
| 6.2.1                                 | Spesifikasi Fungsional Dan Teknis.....         | 71 |
| 6.2.2                                 | Risiko Dan Mitigasi .....                      | 72 |
| 6.2.3                                 | Desain Sistem.....                             | 72 |
| 6.2.4                                 | Pemodelan (Modelling).....                     | 73 |
| 6.2.5                                 | Desain <i>Software</i> .....                   | 73 |
| 6.2.6                                 | Unified Model Language (UML).....              | 74 |
| 6.2.7                                 | Desain Antarmuka Pengguna .....                | 77 |
| 6.2.8                                 | Desain Database .....                          | 77 |
| 6.3                                   | Pembangunan Software.....                      | 78 |
| 6.3.1                                 | Pemrograman.....                               | 78 |
| 6.3.2                                 | Integrasi Sistem.....                          | 80 |
| 6.3.3                                 | Debugging.....                                 | 80 |
| 6.3.4                                 | Programmer .....                               | 81 |
| 6.4                                   | <i>Quality Assurance (QA)</i> .....            | 82 |
| 6.4.1                                 | Rencana Kualitas ( <i>Quality Plan</i> ) ..... | 82 |
| 6.4.2                                 | Testing.....                                   | 84 |
| 6.4.3                                 | Metodologi Testing .....                       | 84 |
| 6.4.4                                 | Strategi Testing .....                         | 84 |
| 6.4.5                                 | Rencana Test (Test Plan) .....                 | 85 |

|  |  |            |
|--|--|------------|
| 6.5  | Dokumentasi .....  | 88         |
| 6.5.1  | Dokumentasi Proses.....  | 89         |
| 6.5.2  | Dokumentasi Produk.....  | 89         |
| 6.5.3  | Kualitas Dokumen.....  | 91         |
| 6.5.4  | Standard Operating Procedure (SOP).....                        | 95         |
| 6.5.5  | Disaster Recovery Plan (DSP).....                              | 96         |
| <b>BAB VII FASE DELIVERY .....</b>                                 |  | <b>98</b>  |
| 7.1  | Deployment .....   | 98         |
| 7.1.1  | Instalasi.....   | 98         |
| 7.1.2  | Pengaturan Database.....                                       | 101        |
| 7.1.3  | Migrasi dan Konversi Data.....                                 | 102        |
| 7.1.4  | Eksekusi Secara Terbatas .....                                 | 103        |
| 7.1.5  | Eksekusi Secara Paralel .....                                  | 104        |
| 7.1.6  | Evaluasi.....  | 104        |
| 7.2  | Pelatihan.....   | 104        |
| 7.2.1  | Pelatihan Pengguna Akhir (End User Training) .....             | 104        |
| 7.2.2  | Pelatihan Administrator.....                                   | 105        |
| 7.2.3  | Pelatihan Eksekutif (Executive Training) .....                 | 105        |
| 7.3  | Manajemen Perubahan.....                                       | 105        |
| 7.4  | Produksi .....   | 106        |
| <b>BAB VIII FASE AKHIR DAN PEMELIHARAAN SISTEM INFORMASI .....</b> |  | <b>108</b> |
| 8.1  | Penutupan Proyek.....  | 108        |
| 8.1.1  | Verifikasi Lingkup Pekerjaan .....                             | 108        |
| 8.1.2  | Penutupan Kontrak .....  | 109        |
| 8.1.3  | Penutupan Administratif .....                                  | 110        |
| 8.1.4  | Hal yang Dipelajari.....                                       | 110        |
| 8.1.5  | Pencatatan Historis Proyek.....                                | 111        |
| 8.1.6  | Post Implementation Evaluation Report (PIER) .....             | 111        |
| 8.2  | Kontrak Pemeliharaan Sistem (System Maintenance Contract)..... | 111        |
| <b>DAFTAR PUSTAKA .....</b>  |  | <b>113</b> |

## DAFTAR GAMBAR

|   |                                     |
|---|-------------------------------------|
| Gambar 1.1 The Waterfall Model .....                | <b>Error! Bookmark not defined.</b> |
| Gambar 2.1 <i>Project Management Triangle</i> ..... | <b>Error! Bookmark not defined.</b> |
| Gambar 2.2 RUP <i>Workflow</i> .....                | <b>Error! Bookmark not defined.</b> |
| Gambar 2.3 Metodologi Tradisional vs. Agile .....   | <b>Error! Bookmark not defined.</b> |
| Gambar 6.1 Contoh <i>Use Case Diagram</i> .....     | 74                                  |
| Gambar 6.2 Contoh <i>Sequence Diagram</i> .....     | 75                                  |
| Gambar 6.3 Contoh <i>Activity Diagram</i> .....     | 76                                  |
| Gambar 6.4 Contoh <i>Class Diagram</i> .....        | 76                                  |
| Gambar 6.5 Bug yang pertama kali ditemukan .....    | 80                                  |
| Gambar 6.6 Model Proses Testing .....               | 84                                  |

## DAFTAR TABEL

|  |    |
|--|----|
| Tabel 5.1 Posisi dan Tanggung Jawab Tim Proyek ..... | 63 |
| Tabel 6.1 Struktur Dokumen .....                     | 91 |

# **BAB I**

## **PENDAHULUAN**

### **1.1 Definisi Sistem Informasi**

Sistem adalah entitas atau unit yang terdiri dari dua atau lebih komponen atau subsistem (sistem yang lebih kecil) yang terhubung dan bekerja sama untuk mencapai tujuan tertentu. Kehidupan sehari-hari memaparkan kita pada banyak sistem. Sepeda adalah alat yang dirancang untuk digunakan sebagai moda transportasi. Ini memiliki dua roda, pedal, rantai, roda gigi, setang, dan rem. Kendaraan itu sendiri adalah sebuah sistem. Mirip dengan instalasi listrik, ataupun institusi seperti bank dan sekolah (Patria, 2018).

Informasi dapat dipandang sebagai pemrosesan input yang telah diatur, diberi tujuan, dan bermanfaat bagi penerimanya. Informasi berbeda dengan data. Data dapat dicirikan sebagai fakta yang tidak diproses atau sewenang-wenang yang digunakan sebagai input dalam prosedur penghasil informasi. Agar bermanfaat, informasi harus memiliki arti penting. Informasi harus dapat dipercaya, relevan, tepat waktu, lengkap, dapat dipahami, dan dapat diverifikasi agar memiliki makna (Supriyanto et al., 2021).

Selain itu, informasi memiliki sejumlah kualitas, termasuk ketergantungan, relevansi, ketepatan waktu, kelengkapan, pemahaman, dan verifikasi. Informasi dianggap dapat diandalkan ketika bebas dari kesalahan dan secara akurat menggambarkan situasi atau operasi organisasi. Jika informasi dapat membantu pengambilan keputusan, itu penting karena memberi pembuat keputusan informasi atau nilai baru (Oktaviyani & Bachtiar, 2017).

Oleh karena itu, sistem informasi adalah metode yang terorganisir dengan baik untuk mengumpulkan, menyimpan, dan mengatur data serta menganalisis, menyajikan, dan mengarsipkannya untuk membantu setiap bisnis atau organisasi mencapai tujuannya. Sistem informasi bisa formal atau informal. Sistem informasi akuntansi, produksi dan penjualan merupakan contoh sistem informasi formal yang memang secara resmi memiliki tanggung jawab untuk menghasilkan informasi yang akurat. Sedangkan sistem informasi informal adalah kebalikannya, berasal dari bagian-bagian organisasi yang tidak secara resmi memberikan informasi, seperti misalnya bagian legal (Tantra, 2012).

### **1.2 Sistem Informasi Dan Sejarahnya**

Secara historis, sistem informasi modern telah berkembang pesat dengan munculnya komputer pribadi (PC). Awalnya dikenal sebagai komputer mikro, PC masih banyak digunakan oleh akademisi dan ilmuwan hingga saat ini. Untuk lingkungan bisnis, komputer lebih banyak digunakan di dunia perbankan pada saat itu, di mana yang dioperasikan adalah komputer-komputer berukuran besar dari kelas mini dan mainframe.

Sistem informasi yang digunakan dalam lingkungan bisnis sering berbentuk laporan pada mesin tik atau terkadang dengan tulisan tangan. Karena laporan bisnis umumnya berbentuk tabel, maka sangat membantu jika ada software lembar kerja yang

sangat membantu pekerjaan tabulasi, apalagi ada berbagai rumus dasar di dalam software lembar kerja tersebut, seperti penjumlahan, pengurangan, dan sebagainya. Dalam perkembangan selanjutnya, manfaat komputer semakin nyata ketika muncul bahasa pemrograman. Dengan program ini, tugas harian dapat diotomatisasi dan disederhanakan.

Pemrogram/programmer mengotomatiskan entri data, pemrosesan data, dan pembuatan laporan (input, pemrosesan data, output) menggunakan bahasa pemrograman, yang memungkinkan operator komputer menyelesaikan tugas rutin dengan lebih cepat dan efektif. Pemrogram membutuhkan penjelasan menyeluruh tentang sistem informasi yang diperlukan untuk membuat program. Hal ini memerlukan bantuan seorang analis sistem yang menganalisis sistem informasi manual, mengubahnya menjadi sebuah desain, dan kemudian, dengan menggunakan desain tersebut sebagai dasar, membuat sebuah program yang pada akhirnya menghasilkan perangkat lunak. Operator komputer kemudian menggunakan program untuk membuat informasi seperti yang dihasilkan oleh sistem manual.

Sistem informasi yang dibangun terintegrasi antar divisi, bahkan antar cabang dan anak perusahaan, dan tidak lagi hanya mewakili satu area bisnis. Lebih dari sekadar bahasa pemrograman, alat untuk pemrograman telah maju. Bahkan database memiliki fasilitas yang membantu penyediaan informasi dengan cara yang lebih bermanfaat, seperti untuk keperluan business intelligence, data warehousing, dan sebagainya. Database tidak lagi hanya berfungsi sebagai penyimpan data.

Sistem informasi yang siap pakai dan adaptable mudah diakses sudah tersedia luas dari berbagai vendor, baik di pasar dalam negeri maupun luar negeri. Aplikasi sistem informasi dengan jangkauan yang sangat luas ditawarkan oleh banyak vendor. ERP (Enterprise Resource Planning) adalah salah satu contohnya. Ada juga yang menjalankan fungsi tertentu untuk kebutuhan kompleks dalam proses bisnis perusahaan, seperti CRM (Customer Relationship Management) dan SCM (Supply Chain Management). Dengan perkembangan sebagaimana yang dapat dilihat sejak awal hingga kini, tentunya dibutuhkan tim dengan kemampuan komunikasi yang baik untuk menciptakan dan mengimplementasikan sistem informasi yang dapat melayani kebutuhan bisnis secara efektif dan efisien (Tantra, 2012).

### **1.3 Sistem Informasi Dan Software**

*Software* (perangkat lunak) adalah seperangkat instruksi yang dapat dipahami dan digunakan oleh komputer atau alat pengolah data untuk memproses data dengan cara yang dimaksudkan/dikehendaki. Representasi dari sistem informasi adalah software atau perangkat lunak. Perangkat lunak dibuat untuk memudahkan pengguna melacak aktivitas bisnis internal seperti transaksi jual beli. Tentu saja kendala prosedural dapat dikurangi, namun perangkat lunak (software) dirancang untuk mempersingkat proses pencatatan, pencocokan, dan pembuatan laporan. Kendala tersebut antara lain ketidakakuratan laporan, kesalahan pencatatan, keterlambatan pembuatan laporan, dan distribusi informasi yang tidak merata antar departemen (Arfan et al., 2019).

Namun, harus diingat bahwa perangkat lunak hanyalah alat untuk membuat prosedur kerja sehari-hari menjadi lebih mudah dan efisien. Jika tidak digunakan dengan benar sesuai prosedur yang telah ditentukan, hasil akhirnya pasti tidak sesuai dengan kebutuhan organisasi ataupun perusahaan (Tantra, 2012).

#### **1.4 Sistem Informasi Dan Teknologi Informasi**

Sebuah sistem informasi terdiri dari sejumlah bagian yang menangani data sehingga dapat diubah menjadi informasi yang berguna dan digunakan untuk memajukan tujuan organisasi atau perusahaan (Maydianto & Ridho, 2021). Teknologi informasi (perangkat keras dan perangkat lunak) adalah alat yang digunakan oleh perusahaan untuk mencapai tujuannya.

Sistem informasi dapat dibuat dengan menggunakan teknologi informasi berdasarkan kriteria yang diinginkan konsumennya. Penggunaan jaringan lokal biasa sudah cukup jika sistem informasi hanya akan digunakan oleh pengguna yang terbatas. Namun, jika pengguna aktif tersebar di beberapa lokasi yang berbeda secara geografis, teknologinya harus disesuaikan dengan situasi. Termasuk sistem keamanan serta jaringannya.

Baik teknologi maupun bisnis selalu berubah. Oleh karena itu, kemampuan teknologi informasi untuk beradaptasi dengan perkembangan bisnis harus diperhitungkan saat menggunakannya. Ini membuatnya penting untuk memikirkan dengan hati-hati tentang pendekatan terbaik saat melakukan investasi teknologi informasi. Pikirkan tentang menerapkan teknologi yang terus berkembang, memiliki dukungan teknis yang baik, dan mudah digunakan. Jangan membeli teknologi hanya demi popularitas. Gunakan jasa konsultan IT, jika perlu, untuk memilih teknologi terbaik (Tantra, 2012).

Dalam lingkungan bisnis yang kompetitif saat ini, perusahaan yang jauh lebih besar tidak serta merta mengalahkan perusahaan yang lebih kecil. Teknologi informasi dapat memberikan bisnis keunggulan kompetitif. Di sisi lain, manajemen teknologi informasi yang tidak tepat dapat mendorong perusahaan ke ambang kegagalan di pasar. Ini mungkin terwujud dalam berbagai cara, termasuk kehilangan klien, memproduksi produk yang tidak sesuai dengan permintaan konsumen, memecat anggota staf yang dapat diandalkan, dan lain-lain (Indrayani, 2012).

#### **1.5 Sistem Dengan Perencanaan**

Perencanaan adalah proses menciptakan tindakan yang akan diambil untuk mencapai tujuan yang dipilih. Perencanaan melibatkan pembuatan berbagai keputusan dan memberikan pembenaran bagi mereka, termasuk memutuskan tujuan, kebijakan, program, metode dan prosedur tertentu, dan kegiatan berdasarkan jadwal harian (Suryapermana, 2016).

Perencanaan sistem adalah fase awal pemilihan sistem, digunakan untuk menilai kelayakan proyek, menentukan anggaran awal, pengeluaran, dan keuntungan, serta faktor

lain yang meningkatkan kinerja manajemen. Untuk membantu organisasi mencapai tujuannya, rencana sistem mengevaluasi tujuan strategis jangka panjang dan prioritas sistem informasi. Perencanaan sistem juga memerlukan pengorganisasian, pengawasan, dan penggunaan sistem informasi (Devi, 2012).

Perencanaan sangat penting untuk pembangunan sistem informasi. Ini mirip dengan membangun rumah di mana Anda harus terlebih dahulu membuat desain untuk bagaimana ruangan akan didistribusikan, serta untuk ukuran, bahan, dan faktor lainnya. Serupa dengan sistem informasi, jika dipaksakan maka sistem akan menjadi tidak stabil dan sulit untuk dipertahankan, apalagi jika ingin dikembangkan lebih lanjut (Tantra, 2012).

## **1.6 Definisi Proyek**

Proyek adalah kumpulan tindakan yang terjadi selama waktu tertentu dengan distribusi sumber daya dan tujuan untuk menyelesaikan tugas yang diberikan. Proyek adalah upaya jangka pendek untuk menciptakan produk, layanan, atau hasil yang spesifik/unik. Ingatlah bahwa proyek dibedakan dari pekerjaan normal dengan kata "sementara". Karena proyek bersifat sementara, pekerjaan selesai dan tim yang dibentuk memiliki awal dan akhir. Namun, barang atau jasa yang diciptakan bersifat permanen. Unik berarti hasil dari proyek merupakan hal baru dengan ciri-ciri yang membedakannya dari hasil sebelumnya. (PMBOK, 2013).

Ukuran suatu proyek bisa besar dengan membutuhkan banyak sumber daya dan teknologi canggih, tetapi bisa juga kecil yang hanya dikerjakan oleh satu dua orang dengan tidak melibatkan sumberdaya yang terlalu banyak dan berlangsung dalam waktu singkat. Suatu proyek besar bisa saja dibagi-bagi dalam beberapa sub-proyek kecil. Misalnya, dalam proyek pembangunan rumah, ada sub proyek yang menangani pekerjaan pondasi, yang menangani konstruksi, serta sanitasi, listrik, dan lain-lain. Setiap subproyek diawasi oleh manajer proyek dan memiliki sumber daya, jangka waktu, dan sumber dayanya sendiri.

((Suatu proyek sistem informasi mungkin merupakan subproyek dari proyek yang lebih besar dalam rekayasa ulang perusahaan) di mana subproyek sistem informasi ini harus ditentukan sendiri dengan sumber dayanya sendiri tetapi akan diintegrasikan dengan proyek induk pada tahap selanjutnya. Ini berarti bahwa suatu subproyek pada akhirnya harus tetap berada dalam lingkup yang ditentukan oleh proyek besar yang menjadi induknya. Sehingga hasil sub proyek harus dapat mendukung tujuan utama proyek.

Dari penjelasan di atas dapat kita lihat bahwa suatu proyek dapat ditentukan sejak awal, apakah merupakan proyek tunggal atau bagian dari proyek lain yang lebih besar, atau bahkan proyek ini merupakan induk dari berbagai sub proyek di bawahnya (Tantra, 2012).

## 1.7 Manajemen Proyek

Proses merencanakan, mengatur, membimbing, dan mengelola tindakan anggota organisasi dan sumber daya lainnya untuk mencapai tujuan organisasi yang ditetapkan adalah manajemen proyek. Tujuan dari manajemen proyek adalah untuk menangani fungsi-fungsi manajemen sehingga hasil terbaik diperoleh sesuai dengan kebutuhan saat ini dan yang telah ditentukan dan mampu mengelola sumber daya seefektif dan seefisien mungkin.

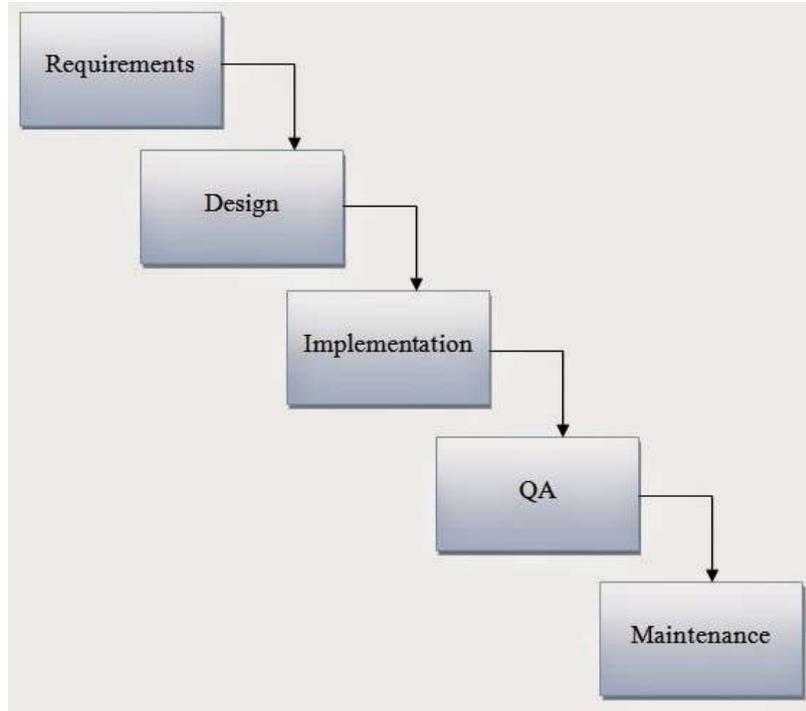
Adapun beberapa fungsi dari manajemen proyek, adalah:

1. Fungsi perencanaan (Planning)  
Membuat penilaian yang akan menangani data dan informasi spesifik untuk penggunaan di masa mendatang, seperti membuat rencana jangka panjang dan jangka pendek, adalah tujuan dari fungsi ini.
2. Fungsi Organisasi (Organizing)  
Untuk mencapai tujuan organisasi, seperti menyusun ruang lingkup kegiatan, fungsi organisasi berusaha untuk menyatukan kumpulan kegiatan manusia yang memiliki kegiatan tertentu, saling berhubungan, dan berinteraksi dengan lingkungannya.
3. Fungsi Pelaksanaan (Actuating)  
Fungsi implementasi berusaha untuk mengkoordinasikan semua pemain organisasi yang relevan dalam melaksanakan proyek dan kegiatan, termasuk motivasi dan arahan tugas.
4. Fungsi Pengendalian (Controlling)  
Fungsi kontrol adalah untuk menilai tingkat kinerja dan memeriksa, mengevaluasi, dan menyarankan cara untuk meningkatkan aktivitas (Arianie & Puspitasari, 2017).

## 1.8 Siklus Pengembangan Software (SDLC-Software Development Life Cycle)

SDLC memainkan peran penting dalam proyek sistem informasi pengembangan perangkat lunak dan diintegrasikan ke dalam organisasi kerja untuk manajemen proyek. The Waterfall Model ini pertama kali diperkenalkan oleh Winston W. Royce (1929-1995) meskipun belum dinamakan demikian. Dalam perkembangan selanjutnya, banyak model yang dimodifikasi dari bentuk aslinya, namun secara umum masih mengacu pada model yang diperkenalkan oleh Royce. Selain Waterfall, ada beberapa teori SDLC lainnya, antara lain Prototyping, Incremental, Spiral, dan RAD.

Instrumen kunci dalam metodologi pengembangan perangkat lunak (SDM) adalah SDLC. Sejak 1960-an, SDM telah berkembang, sebagian karena bisnis yang semakin signifikan telah beralih ke komputerisasi. SDLC bisa dibilang merupakan metodologi resmi yang paling mapan untuk membuat sistem informasi. Tujuannya agar pengembangan sistem informasi lebih metodis, sistematis, dan menyeluruh (Tantra, 2012).



**Gambar 1. 1** *The Waterfall Model*

### **1.9 Pemrograman**

emrograman merupakan bagian penting dari SDLC. Sebuah sistem informasi berbasis software tentu tidak bisa terlepas dari tugas ini. Pada umumnya, program terdiri dari rangkaian perintah atau instruksi yang dapat dimengerti oleh komputer. Instruksi tersebut berfungsi untuk mengatur tugas-tugas yang akan dilakukan oleh komputer agar menghasilkan keluaran yang diinginkan. Program juga bisa diartikan sebagai kumpulan instruksi tertulis yang dibuat oleh programmer atau sebagai bagian dari perangkat lunak yang dapat dijalankan. Pemrograman melibatkan urutan perintah yang diberikan ke komputer untuk menjalankan tugas-tugas tertentu. Perintah-perintah ini membutuhkan bahasa tertentu yang dapat dimengerti oleh komputer. Kegiatan pemrograman utamanya dilakukan oleh programmer yang menerima hasil desain dalam bentuk deliverables untuk diimplementasikan dalam bentuk software (Suryoprayogo & Sidiq, 2016).

### **1.10 Database**

Sebelum melaksanakan proyek pengembangan sistem informasi, ada satu hal penting yang harus dipertimbangkan, yaitu menentukan bagaimana data akan disimpan. Saat ini, terdapat banyak software pengelola database (database server) di pasaran, masing-masing dengan kelebihan dan harga yang berbeda. Penentuan ini harus dilakukan sejak awal karena akan mempengaruhi bagaimana sistem informasi dibangun.

Programmer juga harus memahami cara memberikan perintah manipulasi data pada database server dan perangkat keras yang dibutuhkan olehnya. Oleh karena itu, perencanaan awal harus mempertimbangkan dengan baik pilihan software pengelola database yang akan digunakan (Tantra, 2012).

Database adalah kumpulan data yang saling terhubung dan diorganisir berdasarkan struktur atau skema tertentu. Database tersimpan di dalam hardware komputer dan dapat dimanipulasi menggunakan software, seperti diperbarui, dicari, diolah dengan perhitungan tertentu, dan dihapus, untuk mencapai tujuan tertentu. DBMS (Database Management System) adalah sistem yang dirancang untuk mengorganisasi dan memproses database di dalam komputer. Sistem ini mampu melakukan operasi pada data dengan referensi data yang sama dan dapat diakses oleh berbagai aplikasi. Meskipun basis data dan DBMS sangat penting dalam konsep basis data, keduanya memiliki arti yang berbeda. Basis data merujuk pada data atau rekaman dalam sebuah database, sedangkan DBMS merujuk pada aplikasi yang digunakan untuk mengelola data atau rekaman tersebut, termasuk menambahkan, menghapus, mengubah, dan menampilkan database itu sendiri (Fikry, 2019).

Beberapa poin penting yang bisa dijadikan pertimbangan memilih DBMS antara lain:

1. Kapasitas penyimpanan sesuai kebutuhan jangka panjang.
2. Keamanan data, memiliki password dan hak akses yang dapat dikonfigurasi dengan aman.
3. Persyaratan perangkat keras yang dapat dipenuhi dengan sumber daya yang ada.
4. Kemampuan yang memadai untuk menangani transaksi sesuai kebutuhan pengguna.
5. Vendor terkenal dan produk yang terus diperbarui.
6. Kompatibel dengan bahasa pemrograman yang digunakan oleh programmer.
7. Dapat diakses dengan alat manajemen basis data, baik yang terintegrasi maupun yang tersedia di pasaran.
8. Dapat dipertahankan (maintenance) oleh staf teknologi informasi yang ada.
9. Fasilitas dan metode pencadangan dan pemulihan standar.
10. Harga.
11. Fasilitas tambahan seperti kemampuan sinkronisasi atau replikasi antar server, dapat dipublikasikan dengan mudah jika diperlukan, tidak tergantung pada perangkat keras tertentu, dan sebagainya.
12. Dapat dikembangkan lebih lanjut untuk kebutuhan seperti business intelligence atau data warehousing (Tantra, 2012).

## BAB 2 MANAJEMEN PROYEK DALAM PRAKTEK

### 2.1 Stakeholder

Pemilik proyek (project owner), komite pengarah (*steering committee*), pengguna hasil proyek (user), dan pelaksana proyek merupakan *stakeholder* yang dimaksud dalam proyek (Setiawan, 2019).

#### 2.1.1 Pemilik Proyek

Seseorang atau organisasi yang menyediakan tenaga kerja untuk membuat struktur dan memasok uang atau biaya lain untuk proyek tersebut dikenal sebagai pemilik proyek (*owner*) (PRATAMA & Harahap, 2022). Pemilik sering mengatur dana sponsor atau bertindak sebagai penggalangan dana. Kepentingan pribadi dan profesionalnya akan terlayani oleh inisiatif dan hasilnya (Tantra, 2012).

#### 2.1.2 Komite Pengarah

Dengan mempertimbangkan area yang biasanya disebut sebagai batas Segitiga Manajemen Proyek (*Project Management Triangle*), komite pengarah dibentuk untuk memastikan bahwa proyek dilaksanakan sebagaimana dimaksud dan tetap berada di jalur untuk menghasilkan hasil yang konsisten dengan kualitas yang dibutuhkan. Segitiga ini terdiri atas:

1. Waktu

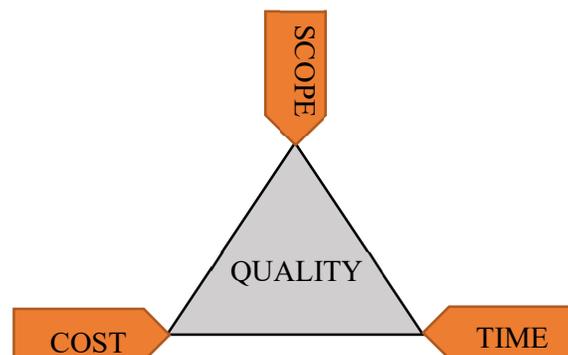
Batasan waktu untuk menyelesaikan proyek sesuai jadwal yang ditentukan.

2. Biaya

Batasan anggaran yang disediakan untuk pelaksanaan proyek.

3. Ruang lingkup

Batasan atas aktivitas yang harus dilakukan untuk menghasilkan hasil akhir proyek sesuai yang diharapkan (Tantra, 2012).



Gambar 2.1 *Project Management Triangle*

Ada hubungan antara ketiga batasan ini. Misalnya, memperluas ruang lingkup proyek akan menghasilkan peningkatan waktu yang dibutuhkan untuk menyelesaikannya, yang akan meningkatkan biaya proyek. Untuk menjaga agar semuanya tetap dalam batas wajar, kolaborasi, terutama pengarahan dan pemantauan proyek oleh pihak-pihak selain pelaksana proyek, sangatlah penting.

Pentingnya fungsi pengarahan ini disebabkan oleh banyak inisiatif sistem informasi yang secara statistik termasuk dalam kategori kegagalan. Jika proyek sistem informasi tidak mencapai ketiga hal dibawah ini, proyek tersebut dianggap gagal.

1. Selesai sesuai jadwal (*time constraint*).
2. Menggunakan anggaran di bawah atau sesuai yang ditentukan (*cost constraint*).
3. Sistem bekerja seperti yang diharapkan (*scope constraint*) (Tantra, 2012).

### **2.1.3 Pengguna Hasil Proyek**

Perangkat lunak yang akan digunakan tentunya akan menjadi hasil akhir dari suatu proyek sistem informasi. Program ini pada akhirnya akan beroperasi secara operasional setelah melewati beberapa tahapan. Seorang pengguna baru pasti membutuhkan bantuan di awal. Pengguna berpengalaman kadang-kadang bisa menjadi pendampingnya. Selain itu, pengguna dapat mengungkapkan ketidaksetujuan mereka terhadap perangkat lunak yang telah selesai. Biasanya, itu adalah kurangnya keterlibatannya sejak awal proyek. Atau bisa jadi dia memiliki pengalaman dengan perangkat lunak serupa yang menurutnya lebih ramah pengguna dan lebih sesuai dengan kebutuhannya. Komite pengarah harus berperan dalam situasi ini. Bersikaplah profesional dalam pendekatan dan jelaskan keuntungan dari sistem informasi yang baru (Tantra, 2012).

### **2.1.4 Pelaksana Proyek**

Pelaksana proyek biasanya bekerja dalam kelompok. Orang-orang dengan keahlian yang diperlukan untuk melaksanakan proyek membentuk anggota tim. Manajer proyek bertanggung jawab atas grup. Meskipun bersifat ad hoc sepanjang proyek, struktur organisasi tim dapat dibentuk jika perlu sehingga setiap orang dapat bekerja sama dengan baik. Usaha skala besar, khususnya, membutuhkan ini. Setiap anggota tim berfungsi sebagai sumber daya untuk tugas-tugas yang unik dalam hal waktu, anggaran, dan cakupannya (Tantra, 2012).

Dalam sebuah proyek sistem informasi, ada beberapa orang dengan tugas-tugas sebagai berikut:

#### **2.1.4.1 Manajer Proyek**

Manajer proyek membutuhkan berbagai keterampilan manajerial, termasuk manajemen waktu, manajemen biaya, manajemen risiko, ruang lingkup pekerjaan, pengadaan, sumber daya manusia, dan integrasi. Sebagai pemimpin tim dalam suatu

proyek konstruksi, manajer proyek diharapkan mampu menghasilkan proyek sesuai dengan kualitas, waktu, dan biaya yang ditentukan. Mengingat bahwa mengelola sumber daya manusia dalam proyek konstruksi membutuhkan banyak tenaga kerja dan kemampuan ini memerlukan koordinasi, pengelolaan, dan kepemimpinan, yang semuanya berdampak pada kinerja karyawan, itu adalah salah satu keterampilan manajerial yang harus dikuasai oleh seorang manajer proyek (Wiguna & Winoto, 2020). Sertifikasi PMP (*Project Management Professional*), yang diberikan oleh PMI (*Project Management Institute*), sering menjadi persyaratan bagi manajer proyek. Mengikuti tes yang diberikan oleh PMI diperlukan bagi manajer proyek untuk mendapatkan kredensial ini.

Namun, sertifikasi hanya berfungsi sebagai deklarasi formal dari kompetensi manajer proyek. Yang lebih dibutuhkan adalah seorang pemimpin yang berpengalaman dalam proyek yang telah berhasil diselesaikan. Seperti yang dinyatakan sebelumnya, memperoleh sertifikasi bukanlah kebutuhan untuk memenuhi syarat sebagai manajer proyek. Seseorang dipaksa untuk mendapatkan sertifikat dengan cepat meskipun mereka menginginkannya.

#### **2.1.4.2 Analis Sistem**

Profesi ini memiliki beberapa nama, antara lain *Business System Analyst*, *Business Process Analyst*, dan lain-lain. Fungsi utamanya adalah melakukan analisis terhadap sistem informasi yang akan dibangun. Tugas seorang analis sistem seringkali juga melibatkan melakukan analisis kebutuhan (*requirements analysis*) melalui simulasi, wawancara, penelitian, pengamatan, analisis kasus (*uses case*) dan metode lainnya. Perancang sistem kemudian akan menggunakannya untuk menghubungkan temuan analisis dengan pemrogram setelah dia mengumpulkan informasi yang memadai. Pekerjaan desain sistem juga dapat diselesaikan dengan hasil yang lebih baik jika analis sistem memiliki keterampilan yang diperlukan. Karena miskomunikasi dapat berkembang jika analis sistem harus melalui orang tambahan untuk menyelesaikan desain sistem dan kemudian mengirimkannya ke pemrogram. Kesalahan persepsi ini dapat berujung pada kegagalan untuk mendapatkan kualitas hasil akhir proyek yang diinginkan klien (Fatta, 2007).

Hasil dari analisis kebutuhan harus dirangkai dalam bentuk dokumen ruang lingkup proyek (*project scope document - PSD*) sebelum diubah menjadi desain yang berfungsi sebagai dasar bagi pemrogram untuk melakukan pekerjaannya. Antara pelaksana proyek dan pelanggan, dokumen PSD ini akan menjelaskan apa yang diharapkan terjadi sebagai akibat dari pelaksanaan proyek. Agar kedua pihak memiliki pemahaman yang sama tentang proyek dan kesimpulannya, hal ini diperlukan. PSD disatukan menggunakan bahasa yang jelas dan tepat yang tidak terlalu teknis. Langkah inisialisasi dari bab berikut memiliki informasi lebih lanjut tentang analisis kebutuhan dan dokumen PSD ini.

#### 2.1.4.3 Desainer Sistem

Perancang sistem akan membuat sistem yang akan dikembangkan menggunakan FSD. Untuk mempersingkat waktu antara hasil analisis dan pemrograman, analisis sistem dengan kemampuan desain teknis dapat melakukannya. Struktur sistem, yang terdiri dari komponen perangkat lunak, tampilan, dan interaksi antar komponen tersebut, akan dikembangkan oleh perancang. Untuk sistem yang semakin rumit, langkah yang dikenal sebagai *software architecture* akan digunakan untuk merancang program, yang bekerja sama seperti ketika seorang arsitek membangun rumah. Biasanya, output dari desain sistem dalam satu proyek berbentuk dokumen yang mencakup desain di berbagai tingkatan, dimulai dengan gambaran umum dan terus turun ke tingkat modul. Semakin detail hasil desain akan semakin mempermudah pemrogram untuk mewujudkannya, namun juga membutuhkan waktu yang lebih banyak dalam menyusunnya (Tantra, 2012).

#### 2.1.4.4 Pemrogram

Seorang *programmer* menggunakan berbagai sebutan, termasuk *programmer*, *developer*, *software engineer*, *analyst programmer*, dan sebagainya. Tanggung jawab utama pemrogram adalah mengubah spesifikasi sistem yang disediakan oleh analisis sistem menjadi instruksi yang dapat dieksekusi komputer (Prabowo & Agustina, 2017). Pemrograman, *programming*, atau *coding* adalah semua istilah untuk kegiatan ini. Menulis bahasa program hanyalah salah satu aspek dari pemrograman; aspek lain termasuk pengujian (*testing*), *debugging/troubleshooting* dan pemeliharaan (*maintenance*). Untuk membuat program sesuai dengan apa yang ditentukan dalam rancangan sistem, seorang *programmer* tentunya harus mahir dalam bahasa pemrograman dan memiliki kemampuan penalaran yang memadai. Karena setiap program yang dibuatnya memiliki potensi untuk mengakses *database*, baik untuk mengambil, mengubah, atau menyimpan data, dia juga perlu menjadi pengelola *database* yang ahli. Pemrogram juga harus terlibat dalam meneliti kemajuan teknis, terutama yang berkaitan dengan bahasa pemrograman yang mereka gunakan (Tantra, 2012).

#### 2.1.4.5 Tester

Seorang *tester* dapat membantu dalam mengidentifikasi masalah perangkat lunak seperti bug, kesalahan, dan kelemahan logika selain menguji perangkat lunak untuk melihat apakah itu diproduksi sesuai dengan desain. Yang pertama, *tester* harus memahami sistem informasi berdasarkan kebutuhan klien. *Tester* dapat menulis skrip pengujian yang disusun berdasarkan kasus penggunaan untuk membantu implementasi pengujian. Yang kedua tidak sepenuhnya memastikan bahwa program bebas dari kesalahan, meskipun dapat mengurangi kemungkinan kesalahan perangkat lunak sebelum digunakan.

*Tester* dapat menjadi anggota tim *Software Quality Assurance* (SQA), yang menilai kualitas perangkat lunak sebelum dirilis, dalam arti yang lebih luas. Anda dapat

mempekerjakan individu di luar tim yang pada akhirnya akan memanfaatkan program selain pengujian internal perusahaan. Ini sering dilakukan oleh penyedia perangkat lunak besar seperti Microsoft, yang terus-menerus membuat versi beta dari produk terbarunya tersedia untuk pengguna, baik secara umum (melalui unduhan situs) atau untuk memilih kelompok pengguna, biasanya calon konsumen untuk produk tertentu. Diantisipasi bahwa masalah perangkat lunak dapat ditemukan dalam situasi yang tidak diperhitungkan oleh pemrogram berkat ketersediaan fungsi pengujian yang terpisah (Tantra, 2012).

#### 2.1.4.6 Anggota Lain dalam Tim

Instalator (*deployer*), sistem integrator, *trainer* dan pendukung teknis (*technical support*) adalah beberapa anggota tim lainnya. Pada perangkat keras klien, *deployer* akan menginstal program. Saat berintegrasi dengan sistem tambahan saat ini atau yang akan datang, integrator sistem mungkin diperlukan. Tugas *trainer* melakukan pelatihan penggunaan *software* pada pengguna operasional. Sementara itu, *technical support* akan akan memelihara sistem yang terpasang. Mungkin ada lebih banyak anggota dalam sistem yang kompleks (Tantra, 2012).

## 2.2 Metodologi dalam Manajemen Proyek

Pelaksana proyek memerlukan aturan atau metodologi untuk melaksanakan proyek yang dipilih. Karena itu, berbagai teknik dan pendekatan, seperti *agile*, *extreme*, *interactive*, *incremental* dan *phases* dapat digunakan dalam manajemen proyek (Tantra, 2012). Metode yang digunakan dalam buku ini didasarkan pada tahapan (fase) karena merupakan metode yang paling umum dan mudah digunakan untuk proyek skala kecil, menengah, dan besar. Berdasarkan metode tradisional ini, ada urutan yang harus diikuti oleh manajemen proyek dari awal hingga akhir. Langkah-langkahnya adalah sebagai berikut:

1. Fase inisialisasi (*initiation stage*).
2. Fase perencanaan atau perancangan (*planning or design stage*).
3. Fase pelaksanaan atau produksi (*execution or production stage*).
4. Sistem pengawasan dan pengendalian (*monitoring and controlling systems*).
5. Fase penyelesaian (*completion stage*) (Institute, 2013).

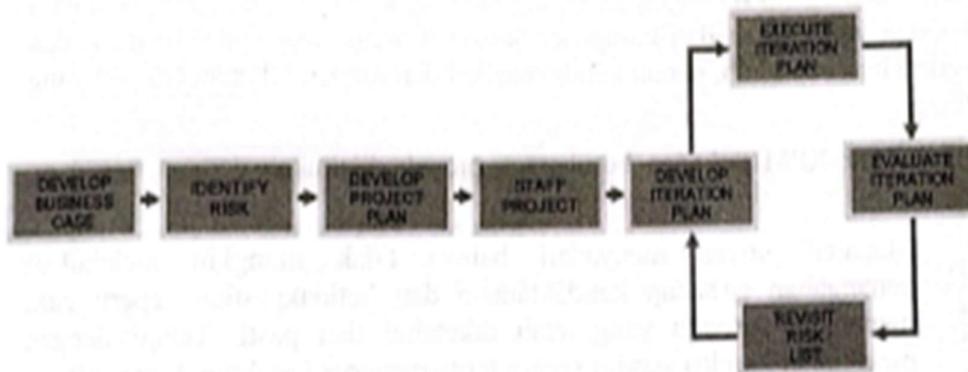
Meskipun tahapan-tahapan ini mengikuti satu sama lain secara berurutan, tidak setiap proyek harus melewati semuanya; bahkan ada yang harus mengulang tahap 2, 3, dan 4 beberapa kali. Hasil dari setiap langkah akan disampaikan sebagai kiriman dan digunakan sebagai input untuk fase berikutnya. Strategi ini sesuai dengan model *waterfall* dari siklus hidup pengembangan perangkat lunak (SDLC), yang juga mengikuti perkembangan linier dari satu tahap ke tahap berikutnya. Selain itu, banyak bisnis menggunakan *Rational Unified Process* (RUP), yang dibuat oleh *Rational Software*, saat menerapkan metodologi ini.

### 2.2.1 Rational Unified Process

RUP adalah metode untuk mengalokasikan tugas dan tanggung jawab dalam organisasi pengembangan perangkat lunak yang merupakan bagian dari proses rekayasa perangkat lunak. Tujuannya adalah untuk menjamin produksi perangkat lunak berkualitas tinggi yang memenuhi kebutuhan pelanggan dalam jangka waktu dan anggaran yang telah ditentukan. Dengan memberi setiap anggota tim akses ke basis pengetahuan yang berisi instruksi, template, dan alat untuk membantu aktivitas penting dalam pengembangan perangkat lunak, RUP bertujuan untuk meningkatkan produktivitas tim (Rosa et al., 2019). *Knowledge base* ini berisi pengalaman-pengalaman terbaik yang terbukti berhasil (*best practices*), yakni:

1. Pengembangan software secara iteratif
2. Mengelola requirements
3. Menggunakan component-based architectures.
4. Model software secara visual
5. Verifikasi kualitas software
6. Pengendalian perubahan-perubahan pada software

Sementara prosesnya masih mengikuti tahapan yang sama seperti dalam siklus hidup manajemen proyek, setiap fase selalu mencakup peninjauan setiap penyampaian yang dihasilkan untuk mencapai kualitas yang diinginkan dan memperhitungkan perubahan dinamis. Basis pengetahuan yang dapat diakses oleh semua anggota tim dan mereka dapat berkontribusi untuk berkembang diperlukan untuk ini. Akibatnya, hasil akhir proyek berkualitas tinggi dan menawarkan keuntungan yang bermanfaat bagi semua pemangku kepentingan (Tantra, 2012).



Gambar 2.2 RUP Workflow

### 2.2.2 Extreme Project Management

*Extreme Project Management* (XPM) adalah teknik yang dirancang untuk proyek ketika kebutuhan tidak diketahui dan pendekatan yang fleksibel dan gesit diperlukan untuk menghadapi keadaan yang ambigu dan terus berubah. Untuk menciptakan nilai dalam lingkungan yang rumit dan kacau, XPM adalah keterampilan memfasilitasi dan

mengendalikan ide, emosi, dan interaksi (Purwanto & Sari, 2015). Jadi, proyek yang bersifat ekstrem adalah:

1. *Requirements* dan ekspektasi yang terus berubah-ubah.
2. Waktu pelaksanaan yang sangat ketat.
3. Suasana pendukung yang tidak tetap dan terus berubah.
4. *Self-correcting*, banyak variabel dan ketidakjelasan internal maupun eksternal pada proyek maupun organisasi sponsor. Tak ada yang mencatat dan mengarahkan.
5. Hanya dapat dipimpin dari atas dan dikelola dari bawah sebagai individu, pasangan, atau kelompok *stakeholder* yang melakukan modifikasi koreksi diri (*self-correcting*) selama proyek sambil tetap mengingat hasil akhir proyek yang diinginkan. Tidak dapat dikelola dalam kerangka top-down

Jelas bahwa teknik tradisional yang lebih terfokus pada proyek-proyek yang tetap, dapat diukur, dan dikendalikan, tidak dapat digunakan untuk melaksanakan proyek-proyek seperti ini sesuai dengan tahapan yang direncanakan. Konsekuensinya, XPM adalah seni dan ilmu (science) dalam memfasilitasi dan mengelola aliran pikiran, emosi, dan interaksi sedemikian rupa sehingga menghasilkan hasil yang bermanfaat dalam kondisi kacau dan kompleks, termasuk situasi yang cepat dan sering berubah, tidak pasti, dan memiliki tingkat stres yang tinggi.

Kontribusi XPM terhadap kesuksesan proyek dilakukan dengan tiga cara, yaitu:

1. Kesadaran manajer proyek bahwa sulit untuk menangani ketidaktahuan dan ketidakpastian dengan cara yang sama seperti Anda menangani sesuatu yang diketahui dan pasti, semuanya berkontribusi pada keberhasilan proyek. Namun, ini dapat dikelola jika koreksi diri dilakukan secara *real-time*.
2. Berikan perhatian Anda pada dedikasi untuk mencapai dan mempertahankan tujuan proyek dengan menyalakan antusiasme dan jaminan di antara para *stakeholder* kunci.
3. XPM bukan hanya sebuah metodologi, alat manajemen proyek, atau template; itu juga menggunakan pendekatan komprehensif yang berfokus pada bisnis, berbasis realitas, dan berpusat pada sumber daya manusia.

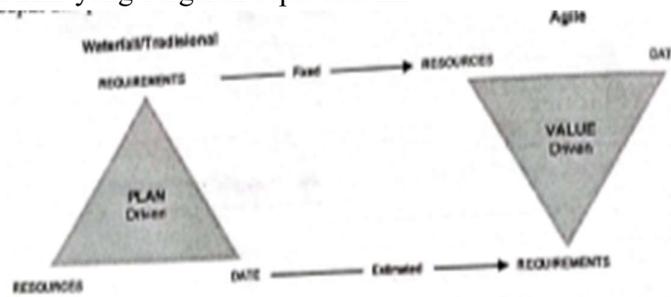
XPM dapat dikatakan berhasil jika mencapai hal-hal berikut:

1. Klien merasa puas dengan hasil sementara yang dicapai dan proyek berada pada jalur yang tepat.
2. Klien senang dengan hasilnya dan dengan bagaimana proyek memenuhi kriteria keberhasilan yang telah disepakati.

- Keuntungan dari hasil proyek dapat dikompromikan. Manfaat komersial proyek yang diproyeksikan dapat diukur dan jelas.
- Anggota tim senang dengan hasil proyek dan bersemangat untuk mengerjakan prakarsa serupa di masa mendatang (Tantra, 2012).

### 2.2.3 Agile Project Management

Metodologi ini mengantisipasi inisiatif e-bisnis yang bergerak cepat dan sering berubah dengan berfokus pada orang, hasil, proses minimal, dan kerja sama maksimal (Purwanto & Sari, 2015). Kita harus cepat beradaptasi dengan perubahan kebutuhan selama fase pengembangan dan menawarkan peningkatan produk yang mengikuti evolusi cepat pengembangan perangkat lunak dan perubahan kebutuhan pengguna agar dapat bersaing di lingkungan bisnis yang bergerak cepat saat ini.



Gambar 2.3 Metodologi Tradisional vs. Agile

*Agile development* memiliki keuntungan jika tim dapat bereaksi lebih cepat terhadap perubahan untuk (Syamsibar, 2021):

- Menurunkan harga transfer informasi dari satu orang ke orang lain.
- Mempersingkat interval antara pengambilan keputusan dan realisasi dampaknya.
- Menempatkan orang lebih dekat secara fisik.
- Menggantikan dokumen dengan pembicaraan langsung dan di papan tulis.
- Menumbuhkan rasa kebersamaan yang lebih kuat di antara anggota tim untuk mendorong pembagian informasi penting dengan cepat.
- Adanya kehadiran dan akan lebih baik keterlibatan dari klien yang ahli dalam tim.
- Bekerja dengan peningkatan yang berkelanjutan.

Menerapkan metodologi ini tepat jika (Tantra, 2012):

- Pengguna secara aktif berkontribusi pada *requirements analysis* dan *modeling*.
- Perubahan dalam *requirements* selalu disambut secara terbuka dan segera diantisipasi.
- Bekerja atas prioritas tertinggi yang diprioritaskan oleh stakeholder dan selanjutnya berfokus pada hal yang berisiko tinggi saat pelaksanaan proyek berlangsung.

4. *Modeling* menggunakan pendekatan berulang (*iterative*) dan meningkat (*incremental*).
5. Pengembangan perangkat lunak adalah fokus utama, bukan model atau dokumentasi.
6. Penyusunan model dalam tim secara bersama-sama dan semua anggota dapat memberi masukan.
7. Selalu aktif berusaha agar segala sesuatunya tetap simpel.
8. Meninggalkan sebagian besar atau semua model saat pengembangan berlangsung.
9. Pemilik proyek memberikan keputusan bisnis, dan pengembang memberikan keputusan teknis.
10. Isi dari model lebih penting secara signifikan daripada format atau bentuk model tersebut.
11. Sangat penting untuk melakukan pengujian terhadap apa yang digambarkan dalam model.

Keuntungan menggunakan agile project management adalah (Syamsibar, 2021):

1. Mempersingkat waktu siklus pengembangan sampai 75%.
2. Beban kerja yang lebih stabil.
3. Pemanfaatan beban kerja yang lebih tinggi sebagaimana dalam pengembangan software skala besar, sistem software dengan jumlah pemrogram yang tetap.
4. Fleksibilitas yang lebih tinggi untuk perubahan rencana manajemen dan pengembangan.
5. Kualitas yang lebih baik karena umpan balik dini dari klien.

Namun, metodologi ini memiliki kekurangan, seperti kesulitan dalam menentukan posisi proyek karena pengulangan fase untuk memperhitungkan perubahan persyaratan, keterbatasan proyek kecil dan menengah karena kurangnya penekanan pada rencana formal keseluruhan proyek, proyek, dan kebutuhan tenaga kerja staf yang bermotivasi tinggi, kompeten, namun kadang-kadang tidak tersedia.

#### **2.2.4 Metodologi Tradisional**

Fase adalah penekanan dari metodologi tradisional. Fase inisiasi, fase perencanaan, fase implementasi, fase pengawasan dan pengendalian, dan fase penyelesaian semuanya diikuti secara berurutan oleh tahapan-tahapan ini. Hasil dari setiap langkah akan disampaikan dan digunakan sebagai masukan untuk tahap berikutnya (Purwanto & Sari, 2015).

#### 2.2.4.1 Fase Inisialisasi

Sangat penting untuk melakukan analisis kelayakan sebelum menentukan proyek. Sponsor dan pemilik proyek sering melakukan tugas ini. Suatu proyek bisa dikatakan layak untuk dilaksanakan jika memenuhi syarat kelayakan, seperti:

1. Menawarkan keuntungan kepada klien.
2. Menawarkan solusi atas masalah yang dialami pemilik proyek.
3. Dapat dilaksanakan dalam kerangka waktu dan kendala keuangan yang diantisipasi dan aset yang dapat diukur, seperti yang ditunjukkan oleh segitiga manajemen proyek.

Ada berbagai tahapan yang harus dilalui dalam studi kelayakan. Analisis kebutuhan adalah salah satu yang paling penting karena menentukan apakah proyek untuk mengembangkan sistem informasi layak dilakukan. Temuan studi kelayakan tersebut kemudian dimasukkan ke dalam proposal proyek yang dipresentasikan kepada para pemangku kepentingan. Tahap proyek berikutnya dapat dilakukan jika proposal diterima. Surat Perintah Kerja (SPK) dapat dibuat dan ditandatangani oleh manajer proyek dan pemilik sebagai sarana menugaskan proyek kepada pelaksana. Tetapi jika proyek dilaksanakan oleh pihak di luar organisasi/perusahaan pemilik proyek, misalnya oleh kontraktor TI maka harus ada perjanjian dalam bentuk kontrak yang mengikat kedua belah pihak. Dengan adanya kontrak dan proposal yang telah disetujui serta hasil analisis maka tahap selanjutnya dapat dilaksanakan (Tantra, 2012).

#### 2.2.4.2 Fase Perencanaan

Meskipun pihak lain, seperti *steering committee*, tetap menjalankan fungsi kontrol dari luar, namun tim pelaksana proyek lebih banyak terlibat dalam pelaksanaan fase ini. Kalaupun ada *requirements analysis* dari tahap sebelumnya, studi yang lebih menyeluruh diperlukan untuk menetapkan strategi dan desain untuk pengembangan sistem informasi.

Temuan analisis sering direvisi pada fase ini. Ini khas karena, karena alur kerja dan *Standard Operating Procedure* (SOP) organisasi/perusahaan yang tidak memadai, kemungkinan besar informasi yang dikumpulkan dari satu departemen ke departemen lain tidak konsisten atau bahkan tidak terhubung satu sama lain. Oleh karena itu, diperlukan re-engineering proses bisnis organisasi atau perusahaan dalam sejumlah proyek berskala besar, seperti implementasi ERP. Banyak kemajuan umum akan dibuat selama reengineering ini dalam hal struktur organisasi, sumber daya manusia, alur kerja, dan SOP. Akan ada birokrasi yang lebih sedikit, koordinasi tugas yang lebih baik yang saat ini tumpang tindih, format dokumen standar, dan pelaporan yang konsisten.

Tim pelaksana proyek akan membuat sejumlah dokumen setelah mengumpulkan data-data penting, yang akan digunakan sebagai landasan untuk melangkah ke tahap berikutnya. Produk akhir dari dokumen ini harus memiliki izin terlebih dahulu untuk mencegah keterlambatan proses ke tahap perencanaan dan desain lagi karena informasi yang baru ditemukan atau revisi dari *stakeholder* lainnya setelah mencapai tahap produksi (Tantra, 2012).

#### **2.2.4.3 Fase Pelaksanaan atau Pengembangan**

Tugas-tugas yang diuraikan pada fase sebelumnya diselesaikan pada fase ini untuk menghasilkan perangkat lunak yang sesuai dengan spesifikasi. Kegiatan berikut berada di bawah lingkup manajemen proyek sistem informasi:

1. Pemrograman (*development*)
2. *Testing*
3. *Quality assurance* (QA)
4. Dokumentasi

Fase ini seringkali membutuhkan lebih banyak waktu untuk diselesaikan daripada fase sebelumnya. Fase ini, tidak seperti yang lain, juga menciptakan produk berupa perangkat lunak yang nantinya akan dimanfaatkan oleh pelanggan, yakni oleh pihak selain tim pelaksana proyek. Oleh karena itu, aktivitas pengujian dan QA diperlukan dalam proyek sistem informasi yang besar dan kompleks.

Selain itu, dokumentasi merupakan tugas akhir yang juga tak kalah pentingnya. Hal ini sering diabaikan dengan asumsi bahwa pelatihan pengguna sudah memadai dan buku panduan (*manual book*) tidak diperlukan. Padahal buku panduan ini cukup penting, terutama menjadi referensi pertama saat pengguna mendapat hambatan dalam pengoperasian software, karena dalam pelatihan pengguna masih belum mengoperasikan software secara penuh dan masih didampingi oleh trainer sehingga belum menemukan hambatan yang berarti. Tetapi pada saat operasional, biasanya pengguna baru menemukan hambatan, karena belum terbiasa, tidak menemukan fungsi yang diinginkan ataupun ada prosedur yang tidak dimengerti. Di sini, fungsi buku panduan adalah menawarkan perbaikan sementara kepada pengguna sebelum menghubungi *technical support*.

Dokumentasi sistem adalah jenis dokumentasi lainnya. Dokumen ini sangat penting, terutama ketika sistem telah memasuki fase pemeliharaan atau akan menjalani pengembangan tambahan. Dokumen ini akan menjadi alat bantu bagi tim pemrogram yang melakukan pemeliharaan sistem, dalam mencari dan memperbaiki bugs dalam program, juga bagi tim pelaksana proyek baru jika sistem informasi dikembangkan lebih lanjut (Tantra, 2012).

#### **2.2.4.4 Sistem Pengawasan dan Kontrol**

Proses dilakukan selama fase ini untuk memantau pelaksanaan proyek untuk mencegah masalah apa pun yang mungkin ditemukan secara instan dan, jika diperlukan, tindakan perbaikan dapat segera dilakukan. Keuntungannya adalah kinerja proyek dapat sering dipantau dan dinilai, memungkinkan deteksi cepat setiap penyimpangan dari implementasi rencana dan desain proyek. Pengawasan dan pengendalian ini terdiri atas:

1. Mengukur aktivitas proyek yang tengah dilaksanakan (menentukan posisi pelaksanaan proyek saat ini).

2. Mengawasi variabel (biaya, waktu, sumberdaya dan sebagainya) proyek terhadap rencana dan desain yang telah disepakati (posisi yang seharusnya dicapai).
3. Identifikasi tindakan korektif jika terjadi penyimpangan (mengembali kan ke posisi yang seharusnya).
4. Mengarahkan pengendalian terpusat agar hanya setiap perubahan terhadap rencana proyek yang telah disetujui saja yang bisa diimplementasikan.

Harap diingat bahwa pengawasan dan pengendalian yang terlalu ketat dapat menyebabkan inefisiensi waktu karena setiap aktivitas proyek terus- menerus diukur. Tetapi jika terlalu longgar juga akan terlalu berisiko. Cara terbaik adalah melakukan pengawasan dan pengendalian hanya pada setiap akhir fase dan jika perlu melakukan review terhadap hasil dari setiap fase (Tantra, 2012).

#### **2.2.4.5 Fase Akhir**

Dalam fase ini proyek telah memasuki tahap akhir di mana produk software telah diinstalasikan, dioperasikan, dan dimanfaatkan oleh klien. Diperlukan serah terima formal sebagai bukti bahwa SPK atau kontrak telah ditutup sebagai tanda bahwa proyek telah selesai. *Stakeholder* melakukan tinjauan terakhir sebelum serah terima, yang dapat berlangsung singkat jika sistem pemantauan dan pengendalian diterapkan dengan benar selama pengembangan proyek. Ada dua aktivitas yang dilakukan dalam fase ini:

1. Penutupan proyek. *Stakeholder* melakukan tinjauan, serah terima formal, dan pemutusan SPK atau kontrak.
2. Memasuki masa maintenance yang dapat dilanjutkan dengan SPK atau kontrak baru. Maintenance penting mengingat produk software tidak bisa 100% bebas dari kemungkinan error dan bugs.

### **2.3 Alokasi Sumberdaya**

Anggota tim adalah sumber daya utama dalam manajemen proyek selain daripada mesin/alat bantu (misalnya PC, printer), keuangan, dan lain sebagainya yang bersifat terbatas. Anggota tim memiliki jumlah terbatas sumber daya manusia yang tersedia dalam hal kuantitas, keterampilan, pengetahuan, dan waktu. Sebelum memutuskan siapa yang perlu direkrut sebagai anggota tim, perlu untuk mengidentifikasi sumber daya manusia ini demi kesederhanaan. Strategi alokasi yang sesuai dengan tuntutan proyek harus diperhitungkan saat melakukan identifikasi ini.

Misalnya, dalam proyek besar dan rumit, seperti pembuatan rantai pasokan, ERP, dan sistem informasi manufaktur. Penting untuk membuat rencana distribusi sumber daya manusia, dimulai dengan analisis sistem, pemrogram, pelatih, dan pengerahan yang, selain memiliki kompetensi yang diperlukan, juga memiliki pengalaman yang diperlukan. Hampir setiap langkah membutuhkan sumber daya manusia yang substansial.

Ketersediaan waktu harus diperhitungkan saat memilih anggota tim. Setiap anggota tim idealnya tidak ditugaskan ke banyak proyek sekaligus. Namun, pembagian sumber daya (atau penjadwalan tugas di mana tugas dapat diselesaikan tanpa mengganggu satu sama lain) dapat dilakukan antar proyek. Misalnya, sambil menunggu waktunya untuk mengerjakan proyek baru yang masih dalam tahap perencanaan dan desain, seorang tester dapat menyelesaikan tugasnya untuk proyek lain.

Mengikuti alokasi waktu dan sumber daya, setiap fase akan tunduk pada batasan waktu dan sumber daya berdasarkan jumlah yang dialokasikan. Bahaya yang dapat timbul akibat batasan tersebut dapat lebih mudah diidentifikasi jika batasan tersebut dikaji lebih detail berdasarkan pengetahuan dan pengalaman. Jadi dimungkinkan untuk mengurangi risiko dengan membuat strategi untuk menghadapinya terlebih dahulu. Akibatnya, analisis risiko (risk analysis) dimasukkan ke dalam proses manajemen proyek.

Manajer proyek harus terampil dalam mengalokasikan secara efektif untuk menghindari kekurangan atau surplus waktu dan sumber daya, yang pada akhirnya dapat menghambat penyelesaian proyek. Akibatnya, seorang manajer proyek harus mampu mengarahkan, mengkoordinasikan, menangani masalah yang muncul, dan meramalkan bahaya (Tantra, 2012).

## **2.4 Subproyek**

Seseorang dapat memulai dengan mengerjakan proyek kecil atau subproyek dari proyek yang lebih besar untuk mendapatkan keahlian mengelola proyek. Sebuah proyek dapat dibagi menjadi beberapa proyek yang lebih kecil selain dapat dibagi menjadi unit kerja yang terorganisir. Misalnya, penerapan modul *production planning*, *material management*, dan *cost management* dapat dibagi menjadi proyek-proyek kecil yang nantinya dapat digabungkan. Sebuah tim dengan manajer proyeknya sendiri bertanggung jawab atas pelaksanaan setiap sub proyek. Setelah digabungkan, proyek dapat dikelola oleh seorang manajer proyek yang menggabungkan semua proyek yang lebih kecil menjadi satu kesatuan yang kohesif. Seseorang dapat belajar banyak tentang manajemen proyek dengan memimpin subproyek yang lebih kecil yang membutuhkan lebih sedikit waktu dan sumber daya secara keseluruhan.

Subproyek juga bisa menjadi tahapan di dalam proyek yang lebih besar yang ditangani oleh tim yang berbeda dari tim yang mengerjakan proyek utama. Mengalokasikan sumber daya akan lebih sederhana jika proyek dibagi menjadi komponen yang lebih kecil. Tanggung jawab penting lainnya dari seorang manajer proyek adalah manajemen sumber daya. Biasanya, ketika proyek besar dibagi menjadi pekerjaan atau subproyek yang lebih kecil dan lebih mudah, alokasi sumber daya juga menjadi lebih efektif. Namun selain faktor-faktor tersebut, deskripsi proyek juga harus mempertimbangkan kompleksitas proyek, kendala waktu, dan ketersediaan sumber daya saat ini. Sebuah proyek dasar akan menjadi lebih sulit jika dibagi lagi menjadi proyek-proyek yang lebih kecil, yang akan mencegah tercapainya efisiensi dan bahkan dapat

menyebabkan penundaan. Akibatnya, manajer proyek harus berhati-hati saat memilih kebijakan ini (Tantra, 2012).

## 2.5 Penjadwalan

Penjadwalan adalah faktor penting untuk dipertimbangkan setelah alokasi sumber daya. *Work Breakdown Structure* (WBS) harus diselesaikan sebelum penjadwalan. Kita akan memecah ruang lingkup proyek menjadi bagian-bagian yang dapat disampaikan yang dapat dikelola yang disebut *work packages*. Prosedur ini, menurut pengalaman, terjadi sebelum jadwal sebenarnya dibuat. WBS akan membantu manajer proyek menetapkan kriteria untuk jenis pekerjaan yang harus dilakukan, serta jumlah waktu yang dibutuhkan untuk menyelesaikan pekerjaan dan mencapai tonggak pencapaian di setiap fase.

Pelaksanaan hasil alokasi waktu untuk setiap langkah kemudian dijadwalkan berlangsung secara berurutan. Tentu saja, analisis kebutuhan harus diselesaikan sebelum penjadwalan dapat mulai dibuat. Manajer proyek dapat membuat perkiraan waktu untuk setiap langkah berdasarkan temuan studi ini, kemudian memasukkannya ke dalam proposal. Jadwal akan disepakati bersama dan ditentukan dalam SPK atau kontrak setelah pembicaraan dengan pemangku kepentingan dan beberapa iterasi modifikasi.

Jadwal yang telah disepakati dengan suara bulat oleh para pemangku kepentingan kemudian digunakan untuk menentukan jadwal rinci untuk anggota tim manajemen proyek. Untuk membuat jadwal menyeluruh, berbagai metode dapat diterapkan. Microsoft Project dapat digunakan untuk merencanakan penjadwalan dan administrasi sumber daya manajemen proyek.

Jika Microsoft Project tidak dapat diakses, kami dapat menggunakan program spreadsheet seperti Calc dari OpenOffice atau Microsoft Excel. Membuat kolom dengan lekukan untuk menampilkan deskripsi pekerjaan untuk setiap fase membutuhkan sedikit keterampilan.

Sederhananya, perangkat lunak spreadsheet hanya berguna untuk penjadwalan dan tidak memiliki kemampuan untuk memasukkan data sumber daya atau detail pada setiap tugas, seperti apakah selesai lebih awal atau bahkan melakukannya. Manajer proyek dapat dengan cepat menentukan apakah tugas berada di jalurnya dan mengambil tindakan cepat pada tugas yang telah melewati batas waktunya berkat kemampuan ini, yang sangat penting untuk pemantauan dan manajemen proyek. Tentunya hal ini hanya bisa dilakukan jika informasi dalam penjadwalan selalu di-update.

GanttProject adalah alat manajemen proyek alternatif yang bersifat freeware dan sangat kaya fitur. *Software* ini dibangun dengan bahasa pemrograman Java. Pertama-tama kita harus menginstal Java Runtime Environment sebelum kita dapat menjalankannya.

Software ini cukup lengkap dan bisa menjadi alternatif Microsoft Project, bahkan mampu membaca file yang disimpan dalam format .mpp. Properti untuk masing-masing task cukup lengkap, termasuk untuk alokasi sumberdaya manusia (Tantra, 2012).

## 2.6 Kegagalan Proyek Sistem Informasi

Mayoritas sumber daya manajemen proyek lebih menekankan teknik dengan memberikan contoh proyek yang berhasil. Pada kenyataannya, banyak inisiatif sistem informasi gagal di dunia nyata. Keahlian seorang manajer proyek mencakup tidak hanya mengetahui bagaimana menyelesaikan proyek dengan sukses tetapi juga mengetahui bagaimana menghindari konsekuensi proyek yang tidak diinginkan. Berikut ini merupakan jenis-jenis kegagalan proyek (Handayani et al., 2018):

1. Kegagalan perencanaan terjadi ketika sebuah proyek kekurangan nilai tambah atau kapasitas untuk menangani penghalang jalan yang muncul di sepanjang rute. Ini menunjukkan bahwa ada kekurangan dalam proyek sejak awal.
2. Kegagalan sponsor terjadi ketika manajer proyek tidak berpartisipasi secara aktif dan/atau tidak memiliki kekuatan untuk membuat pilihan yang penting bagi keberhasilan proyek.
3. Jika ruang lingkup tidak ditentukan dengan jelas, tim implementasi proyek akan mengalami kegagalan desain dan definisi/lingkup karena mereka tidak akan tahu persis apa yang disampaikan.
4. Kurangnya komunikasi atau menghindari diskusi terbuka tentang tantangan dan kekhawatiran terkait proyek mengakibatkan kegagalan komunikasi.
5. Ketika metodologi atau proses proyek mengabaikan kesalahan kecil sehingga variabel mitigasi yang sudah ada tidak pernah diterapkan, hasil kegagalan disiplin proyek.
6. Kegagalan pemasok/vendor terjadi ketika struktur hubungan menghalangi dialog atau koreksi. Karena birokrasi organisasi, hanya departemen pengadaan perusahaan—yang terpisah dari tim manajemen proyek—yang dapat berkomunikasi.

Saat proyek selesai, menurut Standish Report, hasilnya dapat dikategorikan sebagai berikut (Tantra, 2012):

1. Sukses, jika proyek selesai tepat waktu, dalam anggaran, dan sesuai dengan lingkup yang telah disetujui bersama.
2. Dalam tantangan, pada akhirnya proyek selesai saat deadline, dengan sedikit melampaui anggaran dan ada beberapa ketidak- sesuaian dengan lingkup yang telah disetujui.
3. Gagal, proyek melampaui apa yang disebut sebagai batasan proyek (Project Management Triangle) sebagaimana yang telah dijelaskan sebelumnya.

Berdasarkan penelitian, proyek yang berhasil adalah karena menerapkan beberapa hal berikut:

1. Keterlibatan klien dalam proyek, tetapi bukan berarti ikut campur dalam prosesnya.
2. Manajer proyek mendapat dukungan dari pihak eksekutif pemilik proyek.
3. Spesifikasi yang sangat jelas, yang sekaligus menunjukkan hubungan yang erat antara klien dan tim proyek.
4. Ekspektasi yang realistis dari klien terhadap hasil proyek.
5. Pengawasan dan pengendalian yang efektif tetapi tidak mengganggu proses yang sedang berlangsung dalam proyek.

Sedangkan untuk proyek yang hasilnya masih berada dalam tantangan, penyebabnya antara lain adalah:

1. Kurangnya masukan dari klien, dan hanya diberi komentar setelah hasil proyek diserahkan kepada klien.
1. Spesifikasi proyek kurang lengkap, yang mungkin disebabkan dari requirements analysis langsung dilakukan desain, tanpa melalui tahapan yang seharusnya.
2. Banyak perubahan terjadi sepanjang perjalanan proyek, dan manajer proyek berusaha memasukkan semua perubahan itu dalam sistem untuk memuaskan stakeholder di luar tim proyek.
3. Kurangnya dukungan dari pihak eksekutif pemilik proyek yang tampak dari ketidaktertarikannya terhadap jalannya proyek.
4. Anggota tim yang kurang kompeten dalam menjalankan tugas-tugasnya (Tantra, 2012).

Namun dalam hal proyek gagal, ada banyak kemungkinan penyebabnya. Faktor-faktor berikut perlu diperhitungkan untuk mencegah proyek diberi label sebagai kegagalan; menurut berbagai penelitian, tingkat kegagalan untuk proyek sistem informasi lebih besar dari tingkat keberhasilan. Sebuah proyek lebih cenderung gagal jika semakin besar dan rumit. Berikut adalah daftar faktor-faktor penting untuk dipertimbangkan yang dikumpulkan dari penelitian pada proyek sistem informasi yang berhasil dan gagal untuk mengurangi kemungkinan kegagalan (Tantra, 2012).

1. Tidak ada partisipasi klien.

Penting untuk mengikutsertakan klien atau pemangku kepentingan lainnya mulai dari perencanaan hingga implementasi. Ini mengacu pada pemberian umpan balik pada proyek yang sedang dilakukan, bukan kepada orang di luar tim manajemen proyek yang mengganggu proses. Pembuatan prototipe dan simulasi dapat digunakan untuk menarik minat dan partisipasi klien atau pemangku kepentingan dalam hal ini. Kesuksesan proyek dan hubungan yang sehat bergantung pada keterlibatan dan

partisipasi klien. Akibatnya, sepanjang siklus manajemen proyek, manajer proyek harus menekankan nilai pendekatan kemitraan dengan pemangku kepentingan.

2. Kurangnya penghargaan pada anggota tim.

Anggota tim tidak dipuji atau diberi hadiah untuk hasil yang sukses. Anggota tim akan termotivasi untuk tampil di level tertinggi mereka dengan menerima hadiah yang sesuai untuk usaha mereka. Penghargaan atau insentif yang memotivasi berdasarkan kinerja juga berguna.

3. Kurangnya keinginan untuk mendapatkan pengetahuan dari kegagalan proyek sebelumnya.

Tidak menggunakan metode yang sudah terbukti benar dan menghindari potensi jebakan dari proyek sebelumnya yang dilakukan oleh organisasi atau bisnis yang sama, atau pengalaman manajer proyek sendiri.

4. Tidak mengidentifikasi, menguraikan, dan mencatat *user requirements* dengan benar.

Salah satu dari tiga elemen kunci yang dapat mempengaruhi awal proyek yang sukses adalah manajemen *requirements*. Akibatnya, pengabaian untuk mengelolanya sejak awal dan perubahan yang terjadi seiring kemajuan proyek dapat menyebabkan kegagalan proyek.

5. Estimasi yang tidak memadai.

Banyak proyek tidak memiliki perkiraan akurat untuk waktu, uang, dan sumber daya yang dibutuhkan. Ketika proyek selesai, ini akan berdampak signifikan dan memberikan tekanan pada tim. Besarnya proyek harus ditentukan berdasarkan perkiraan oleh personel yang terlatih atau berpengalaman. Estimasi juga harus dibandingkan dengan proyek sejenis yang pernah dilaksanakan sebelumnya, apakah lebih atau kurang.

6. Tekanan komersial.

Akan ada tekanan pada komputasi dan daya saing bisnis jika hasil akhir proyek adalah perangkat lunak yang akan didistribusikan secara ekstensif. Jangka waktu terbatas karena pemasaran biasanya memiliki jadwal sendiri untuk peluncuran baru, oleh karena itu anggaran diperas setipis mungkin untuk memberikan margin keuntungan yang tinggi dari penjualan. Masalah ini akan menghasilkan batas yang sewenang-wenang.

7. Penyusunan anggota tim yang tidak tepat.

Penentuan dan pengorganisasian tim dipengaruhi oleh pilihan teknologi yang sesuai dengan basis teknologi yang digunakan dalam proyek sistem informasi. Anggota tim harus cukup berpengetahuan dan, jika perlu, memiliki pengalaman dengan proyek yang sebanding. Selain itu, alokasi harus menyertakan beban kerja proyek selama pelaksanaan karena melibatkan sumber daya manusia.

8. Kurangnya manajemen risiko.

Keberhasilan proyek bergantung pada pengenalan risiko dan pengambilan langkah-langkah untuk menguranginya. Mengamankan implementasi proyek akan

memerlukan identifikasi risiko yang akurat dan pembuatan prosedur mitigasi yang efektif. Berdasarkan pembelajaran dari proyek-proyek yang telah selesai, mitigasi juga dapat direncanakan. Sangat penting bagi manajer proyek untuk secara teratur menilai potensi rencana pengurangan risiko.

9. Gagal dalam menetapkan dan menangani ekspektasi.

Kegagalan proyek dapat diakibatkan oleh harapan *stakeholder* yang terlalu optimis atau tidak realistis. Untuk menetapkan harapan yang realistis di antara para pihak, manajer proyek harus berkonsultasi dengan pemangku kepentingan untuk mencapai pemahaman.

10. Manajemen kualitas yang lemah.

Untuk memastikan bahwa barang yang dihasilkan dari satu tahap ke tahap berikutnya, termasuk produk perangkat lunak akhir, berkualitas tinggi, manajer proyek harus melakukan evaluasi kualitas secara berkala. Ini bukan ide yang baik untuk menyerahkan kontrol kualitas kepada penguji saja; seluruh tim harus berfokus pada kualitas setiap saat.

11. Politik stakeholder.

Sebuah permainan politik internal dalam sebuah proyek dapat dimainkan oleh satu atau lebih pemangku kepentingan yang menginginkan proyek tersebut gagal. Tujuannya mungkin berbeda, seperti keengganan untuk menggunakan perangkat lunak baru sebagai hasil dari proyek atau keinginan untuk menghindari pengawasan yang disebabkan oleh penggunaan sistem informasi baru. Karena hal ini lebih berhubungan dengan masalah sosial, maka jika manajer proyek mendapat dukungan dari eksekutif/pemilik proyek serta *steering committee* yang memberikan pengawasan terhadap semua pihak, maka risiko masalah ini dapat diminimalisir.

12. Manajemen perubahan yang lemah.

Cara melakukan sesuatu dan bahkan struktur organisasi akan berubah sebagai akibat dari proyek sistem informasi. Bahkan mungkin penting dalam proyek-proyek besar, seperti yang telah disebutkan, untuk mengatur ulang dan merekayasa ulang semua komponen organisasi yang akan dipengaruhi oleh penerapan proyek sistem informasi baru.

13. Pelaksanaan pengembangan (*development*) yang tidak memadai.

Masalah ini terkait dengan komposisi tim yang tidak tepat, yaitu anggota tim pemrograman yang kurang pengetahuan atau pengalaman. Bisa juga terjadi akibat melakukan tugas rangkap pada tugas atau proyek lain, sehingga waktu kerja menjadi tidak berguna. Masalah lainnya adalah kurangnya komunikasi tim internal, yang menyebabkan beberapa masalah saat menerjemahkan desain ke dalam kode. Dengan manajemen testing yang efisien dan efektif dan dilaksanakan secara periodikal, maka masalah ini bisa ditangani.

14. Kurangnya komunikasi.

Karena beberapa masalah yang teridentifikasi biasanya disebabkan oleh komunikasi yang buruk, seperti yang dapat dilihat di atas, harus selalu ada komunikasi antara banyak pemangku kepentingan selama proyek berlangsung, terutama selama periode awal. Mungkin ada masalah jika semua orang percaya bahwa tim pelaksana proyek telah diberitahu tentang semuanya. Agar semua pihak yang berkepentingan dapat hadir dan menyuarakan idenya, manajer proyek harus berinisiatif untuk mengatur dan mengatur rapat koordinasi secara berkala.

#### 15. Percaya pada keajaiban.

Jangan tinggalkan proyek dan berharap keajaiban ketika mencapai tahap kritis dan mengalami kesulitan. Tahap penting ini sering terjadi karena mitigasi tidak direncanakan karena tidak ditemukan masalah dalam manajemen risiko. Bahkan jika kadang-kadang memerlukan pengorbanan waktu dan uang, seorang manajer proyek yang sukses harus berkolaborasi dengan cepat untuk menyelamatkan seluruh proyek. Namun, dalam situasi seperti ini, sangat penting agar produk jadi tetap dapat dikirimkan dalam kondisi yang dapat diterima.

Beberapa hal di bawah ini juga perlu diperhatikan agar proyek dapat sukses (Tantra, 2012).

##### 1. Perencanaan yang baik

Rencana yang baik adalah yang berorientasi ke depan dan dilakukan secara detail. Setiap aktivitas mendapat perhatian penuh agar tidak ada yang terlewatkan. Tidak hanya didokumentasikan saja, tetapi jika saat proyek berlangsung dan tidak sesuai dengan rencana, maka manajer proyek dapat melakukan perubahan terhadap rencana awal, disesuaikan dengan situasi, sehingga tahap berikutnya tetap terkendali.

##### 2. Tugas dan tanggung jawab yang jelas

Semua anggota tim harus memiliki pengertian yang jelas atas tugas dan tanggung jawab masing-masing dan selalu berorientasi pada kualitas yang diharapkan dari mereka. Koordinasi dari manajer proyek akan membantu mengingatkan bahwa setiap hasil (*output*) dari satu fase akan menjadi masukan (*input*) bagi fase berikutnya, sehingga agar produk akhir berkualitas baik maka keluaran dan masukan antarfasen selalu memiliki kualitas yang tinggi.

##### 3. Kontrol terhadap jadwal

Manajer proyek secara rutin harus selalu memperhatikan jadwal, waktu yang telah digunakan, milestone yang telah dicapai, perubahan dalam alokasi sumberdaya atau tugas, dan sebagainya. Melakukan evaluasi dan update secara teratur lebih baik daripada hanya sewaktu-waktu, karena jika terjadi perubahan yang tidak sesuai dengan rencana, atau proyek telah keluar jalur, manajer proyek dapat segera mengambil tindakan untuk mengembalikan proyek kepada posisi yang benar. Jika hanya dilakukan sewaktu-waktu, maka bisa jadi tindakan yang harus dilakukan terlalu besar dan berisiko karena penyimpangan sudah terlalu jauh saat disadari.

## 2.7 *Project Management Office (PMO)*

PMO berfungsi sebagai pusat untuk semua koordinasi dan komunikasi dalam program dan proyek perusahaan. Ini membantu manajer proyek dalam menempatkan peran pendukung ini sehingga proyek dapat diselesaikan dengan sukses (Nizar, 2018).

Tidak semua perusahaan atau organisasi membutuhkan PMO. Beberapa hal yang menjadi pertimbangan untuk membentuk PMO adalah (Tantra, 2012):

1. Lingkup proyek yang terus-menerus berubah menyebabkan perlunya tenaga ahli tambahan untuk menangani perubahan-perubahan ini.
2. Perusahaan atau organisasi sedang mengelola beberapa proyek sekaligus yang memanfaatkan sumberdaya yang sama.
3. Melibatkan beberapa kontraktor dan vendor yang berbeda karena besar dan kompleksnya proyek yang sedang ditangani, sehingga jika ada kontraktor atau vendor yang tidak memiliki satu atau beberapa sumberdaya yang diperlukan maka dapat menggunakan sumberdaya dari mitra lain yang diatur dalam PMO.
4. Perusahaan atau organisasi memerlukan laporan dan metrics secara konsolidasi dari berbagai proyek.
5. Perlunya satu sumber informasi untuk berkomunikasi dengan klien.
6. *Time to market* menjadi faktor penting dalam penyelesaian proyek.
7. Lokasi yang terpisah secara geografis, jika sistem akan diaplikasikan untuk lokasi-lokasi yang berjauhan.
8. Sumberdaya yang terbatas, tetapi harus menyelesaikan beberapa tugas sekaligus.

PMO dapat menjawab kebutuhan untuk (Tantra, 2012):

1. Layanan-layanan Manajemen Proyek seperti *trainer*, konsultan, praktisi manajemen proyek dan teknis, antara lain dengan:
  - a. Klarifikasi peran proyek dan manajemen proyek dalam perusahaan.
  - b. Menentukan metodologi standar manajemen proyek termasuk alat bantu, suasana kerjasama, dan standar komunikasi.
  - c. Menyusun dokumen dan template standar untuk memfasilita estimasi proyek, perencanaan proyek, jadwal proyek, manajemen risiko, manajemen topik, manajemen perubahan, akseptasi proyek, dan laporan-laporan proyek.
  - d. Menyiapkan atau menyediakan pelatihan untuk manajemen proyek beserta alat bantu.

- e. Menyiapkan atau menyediakan pelatihan, pengarahan, bimbingan dan pendampingan, menyiapkan tenaga manajer proyek yang kompeten.
  - f. Menyediakan pusat informasi mengenai segala hal tentang proyek yang berlangsung.
  - g. Menyediakan konsultasi untuk menyesuaikan dengan tujuan bisnis
  - h. perusahaan dengan menggunakan solusi teknologi yang tepat.
  - i. Menyediakan sumberdaya yang semakin meningkat pemanfaatannya dalam seluruh bagian organisasi dan menyesuaikan dengan kebutuhan proyek, dengan keahlian khusus, tersedia, dan membantu keseimbangan beban kerja manajer proyek dan anggota tim proyek secara lintas proyek.
  - j. Mengatur dan mementingkan prioritas proyek.
  - k. Memanfaatkan ruang kerja secara kolaborasi, staf PMO dapat membantu manajer proyek dalam urusan administratif seperti menyiapkan jadwal proyek dan pengisiannya, pembuatan laporan dan distribusinya, dan mengurus dokumen-dokumen proyek lainnya.
2. Pengawal bagi metodologi, proses, dan metriks yang digunakan dalam perusahaan atau organisasi.
- a. *Review* dan audit terhadap implementasi manajemen proyek dalam perusahaan atau organisasi, untuk memastikan pengaplikasian cara-cara manajemen proyek yang baik dan memberikan asistensi untuk menyesuaikan dengan cara-cara standar manajemen proyek (melalui pelatihan, pengarahan dan pendampingan).
  - b. Menyediakan kantor yang berfokus pada klien secara terpusat, untuk mengurus hal-hal yang berhubungan dengan sponsor, klien atau stakeholder secara umum. Biasanya hal ini dilakukan untuk proyek yang ditujukan bagi pihak ketiga.
  - c. Melakukan, melaporkan, dan mengumpulkan informasi dari review manajemen proyek.
  - d. Menerima, melakukan konsolidasi, dan mendistribusikan informasi untuk semua proyek.
  - e. Memberikan asistensi dalam menyeleksi dan menganalisis proyek.
  - f. Menyediakan metriks proyek dan dashboard atau scoreboard proyek dari berbagai sudut pandang untuk perusahaan atau organisasi secara keseluruhan.

- g. Memberi kepastian untuk proyek yang penting dapat diselesaikan tepat waktu dan dalam anggaran dengan memberikan akuntabilitas yang obyektif pada setiap fase, dari inisialisasi sampai selesai.
  - h. Sebagai titik pusat kontrol dan komunikasi untuk segala masalah dan risiko dari keseluruhan lintas proyek.
  - i. Menyediakan kontrol terpusat untuk semua proyek yang ada dalam koordinasi PMO.
  - j. Meningkatkan koordinasi dan komunikasi lintas proyek.
  - k. Mengurangi beberapa biaya proyek karena beberapa pekerjaan dapat ditangani pada level PMO.
3. Mengumpulkan dan mendokumentasi kesuksesan dan kegagalan proyek.
- a. Mengumpulkan pengalaman-pengalaman dan data yang didapatkan dari proyek untuk digunakan dalam proyek berikutnya dan untuk meningkatkan metodologi proyek (dalam *knowledge base*).
  - b. Mengumpulkan dokumentasi dan teknik-teknik terbaik yang digunakan dalam proyek yang sukses untuk ditambahkan dalam *knowledge base* seperti *template*, estimasi, dan sebagainya.
  - c. Melihat dan mencari cara-cara terbaik dari perusahaan atau organisasi lain yang mungkin dapat diterapkan secara internal.
  - d. Karena PMO bersifat kolaboratif lintas proyek maka tentu diperlukan alat bantu untuk mempermudah tugas-tugas dan koordinasi di dalamnya. PMO dapat menggunakan software Lotus Notes dengan server Lotus Domino. Dalam *software* ini sudah ada aplikasi email, kalender, PIM, instant messaging, browser web, dan aplikasi yang bisa disesuaikan dengan fit yang lengkap. Lotus Notes juga mendukung replikasi, sehingga untuk proyek-proyek yang berada di lokasi yang jauh secara geografis tetap dapat bekerjasama melalui PMO. Selain itu juga ada *software* yang khusus didesain untuk PMO, seperti *projectmanager.com* (<http://www.projectmanager.com>) dan *Daptiv* (<http://www.daptiv.com>) yang berbasis web.

### **BAB III**

#### **PERAN MANAJER PROYEK**

Dalam sebuah proyek sistem informasi yang efektif, manajer proyek selalu berfokus pada 4P, yaitu *people*, *product*, *process*, dan *project*. Penulis ingin menambahkan satu P lagi, yaitu *planning*. Kelima hal ini dapat ditemukan pada pembahasan baik mengenai apa yang dilakukan oleh manajer proyek, keahlian yang diperlukan maupun tugas dan tanggung jawab manajer proyek.

#### **3.1 Apa yang Dilakukan oleh Seorang Manajer Proyek?**

Secara singkat, yang dilakukan oleh seorang manajer proyek adalah (Tantra, 2012):

1. Menerima otoritas dari pihak yang berkompeten. Pastikan manajer proyek mendapatkan otoritas yang cukup untuk mengelola sumber daya termasuk anggaran tanpa melalui birokrasi yang panjang. Meskipun demikian, hal ini tidak berarti bahwa manajer proyek harus memiliki otoritas penuh seperti melakukan pembelian langsung untuk kebutuhan proyek. Sebagai gantinya, ia dapat mengajukan permintaan melalui bagian pengadaan dengan persetujuan langsung dari direksi. Selain itu, beberapa otoritas juga harus didelegasikan ke anggota tim untuk memastikan mereka dapat melaksanakan tugas mereka secara efektif.
2. Untuk memastikan bahwa klien menerima sistem yang sesuai dengan kebutuhan dan keinginan mereka, diperlukan definisi kasus bisnis yang jelas dan dilakukan review secara teratur dengan evaluasi dan kontrol secara reguler.
3. Memulai dan merencanakan proyek dengan mengembangkan format, arahan, dan landasan yang memberikan peluang untuk mengukur penyimpangan/variasi dan mengendalikan perubahan.
4. Berkolaborasi dengan pengguna akhir, bekerja dengan sponsor proyek dan manajemen lainnya untuk mendapatkan kemajuan dan arah proyek dengan mencapai tujuan, meraih target, memecahkan masalah dan mengurangi risiko.
5. Kelola teknologi, orang, dan perubahan untuk mencapai tujuan, meraih target, menyelesaikan proyek tepat waktu dan sesuai anggaran.
6. Mengelola tim pelaksana proyek dengan menciptakan lingkungan yang mendukung dalam menyelesaikan aplikasi baru dengan biaya yang efisien.
7. Mampu menghadapi situasi yang tidak pasti, perubahan yang cepat, ketidakjelasan, kejutan, dan lingkungan yang belum terdefinisi dengan baik.
8. Mempertahankan hubungan yang baik dengan klien melalui laporan yang lengkap, jelas, dan formal sebagai wujud apresiasi terhadap hubungan yang produktif dan saling menghormati.
9. Memimpin proyek dengan memberi contoh, memberikan momentum dan motivasi untuk semua perhatian hingga proyek mencapai tujuannya..

## 3.2 Keahlian yang Dibutuhkan

Berikut adalah empat kategori yang dapat memandu pemahaman yang dibutuhkan untuk menjadi manajer proyek yang baik:

### 3.2.1 Keahlian Personal

Manajer proyek harus memiliki keterampilan pribadi, yaitu keterampilan dalam berurusan dengan banyak orang, seperti (Setyawati et al., 2021):

#### 1. Kepemimpinan

Karena sebagai manajer yang memimpin tim, tentunya harus memiliki kemampuan memimpin tim. Ada berbagai referensi yang mengajarkan teori kepemimpinan, namun di dunia nyata, kepemimpinan pada hakekatnya adalah bagaimana membangun hubungan yang baik antara Anda dan anggota tim Anda. Memotivasi orang lain untuk memiliki visi dan melakukan yang terbaik untuk mencapai tujuan mereka.

#### 2. Komunikasi yang baik

Berikan informasi yang jelas, jujur, jujur, dan tanpa pamrih yang memungkinkan semua anggota tim memahami lokasi dan status proyek serta hubungannya dengan klien, dan juga harus berkomunikasi secara jelas dengan pihak di luar tim.

#### 3. Negosiasi

Bernegosiasi, jika perlu, dengan semua pihak yang terlibat, secara langsung atau tak langsung.

#### 4. Mendengarkan

Ini adalah bagian dari komunikasi dua arah, tetapi sangat penting untuk mengumpulkan informasi dari anggota tim melalui mendengarkan dengan cermat, sehingga mereka benar-benar memahami apa yang sebenarnya diberikan.

#### 5. Membangun tim

Melakukan rekrutmen dimulai dengan pengorganisasian tim proyek. Berikan setiap anggota tim definisi yang jelas tentang tanggung jawab mereka, dan jika perlu, gantikan anggota tim yang berkinerja buruk sambil mempertimbangkan dampaknya terhadap proyek yang sedang berlangsung. Subjek ini akan dibahas secara lebih rinci di bawah ini untuk informasi lebih lanjut.

Manajer proyek dapat membantu anggota tim dalam berbagai cara, baik secara langsung maupun tidak sengaja. Beberapa contoh di antaranya adalah (Tantra, 2012):

#### 1. *Manage by example* (MBE).

Karena setiap anggota tim terus-menerus mengawasi setiap gerakan Anda, seorang manajer proyek harus dapat dipercaya, terbuka, mampu menangani orang dan proyek yang dia awasi, dan mampu berbicara dengan lugas dan langsung. Seseorang yang dapat secara efektif menjelaskan kapan harus bekerja keras dan kapan harus bersantai adalah manajer proyek.

#### 2. Sikap yang positif.

Manajer proyek harus selalu menjaga pandangan positif, terutama selama masa-masa sulit atau saat menghadapi tantangan. Tindakan negatif berpotensi mengikis kepercayaan dan melemahkan kebijakan.

3. Mendefinisikan ekspektasi.

Jika diperlukan, konfirmasikan secara tertulis dengan anggota tim apa yang diharapkan dari mereka untuk memastikan tidak ada tugas yang tidak dapat diselesaikan.

4. Bersikap bijak.

Manajemen proyek membutuhkan banyak individu dengan berbagai kemampuan dan dari berbagai tingkatan dan posisi. Yang terpenting, sangat penting untuk bersikap diplomatis dan sopan terhadap setiap orang dan anggota tim, serta memberi mereka waktu dan perhatian Anda. Dimungkinkan untuk berpikir tentang mengungkapkan rasa terima kasih atau kekaguman secara tertulis atau melalui email.

5. Bersikap terbuka.

Jika manajer proyek mudah didekati dan bersedia mengatasi kesulitan apa pun, semua orang akan menghormatinya. Jangan pernah menyembunyikan kesulitan atau menghindari penjelasan. Tingkat manajemen yang lebih tinggi harus dikonsultasikan jika manajer proyek dan timnya tidak dapat menyelesaikan masalah. Jangan pernah menyetujui apa pun yang tidak bisa Anda pertahankan.

6. Berikan yang terbaik.

Tidak peduli seberapa berbakat Anda dan tim Anda, jangan pernah goyah dari komitmen Anda untuk secara konsisten menghasilkan hasil terbaik. Jangan pernah memberikan hasil yang cukup baik atau di bawah standar. Minta seluruh tim untuk menjunjung tinggi janji ini.

Manajer proyek adalah sumber informasi utama bagi klien mengenai status dan situasi proyek, termasuk apakah proyek berjalan sesuai rencana, tetap sesuai anggaran, dan sesuai dengan jadwal yang telah ditentukan sebelumnya. Jika proyek menyimpang dari jalur, manajer proyek harus tetap dapat dengan tenang menjelaskan masalahnya kepada klien tanpa menyembunyikan kebenaran. Untuk memenuhi tujuan yang telah ditetapkan, penting untuk memberikan jaminan bahwa proyek akan tetap dilaksanakan.

### 3.2.2 Keahlian Teknis

Ada dua perspektif tentang kecakapan teknis. Beberapa orang percaya bahwa pengetahuan saja sudah cukup untuk menjadi seorang manajer proyek daripada keahlian teknis. Manajer level di bawahnya, seperti manajer pemrograman, manajer infrastruktur, dan sebagainya, bertanggung jawab atas spesifikasi teknis. Menurut sudut pandang lain, manajer proyek harus mahir secara teknis di bidang tempat mereka bekerja.

Sejumlah pengetahuan atau keahlian teknis yang relevan dengan proyek yang sedang ditangani, misalnya (Setyawati et al., 2021):

1. Manajemen konfigurasi; pengetahuan mengenai prinsip dan metode untuk merencanakan dan mengelola implementasi, update, atau integrasi komponen sistem informasi.

2. Manajemen data; pengetahuan mengenai prinsip, prosedur dan alat bantu manajemen data seperti misalnya teknik pemodelan, *backup* dan *recovery data*, *data dictionary*, data warehousing, data mining, data disposal, dan proses standardisasi data.
3. Manajemen informasi; yang meliputi penentuan kebutuhan informasi, mengetahui bagaimana atau di mana mengumpulkan informasi, dan mengatur dan memelihara data atau sistem informasi manajemen.
4. Strategi dan perencanaan sumberdaya informasi; pengetahuan mengenai prinsip, metode dan teknik assesment, perencanaan, manajemen, pemantauan dan evaluasi teknologi informasi, seperti IT baseline assessment, interagency functional analysis, contingency planning dan disaster recovery.
5. Keamanan jaringan dan sistem informasi; pengetahuan tentang metode, alat dan proses, termasuk pembuatan rencana keamanan informasi untuk menghindari kelemahan sistem informasi dan rencana pemulihan keamanan sistem informasi dan layanan jaringan.
6. Arsitektur teknologi informasi; pemahaman tentang desain arsitektur dan proses pengembangan yang digunakan untuk sistem informasi, termasuk struktur operasional internal dari sistem yang berinteraksi dengan sistem lain..
7. Penilaian (assessment) kinerja teknologi informasi; pengetahuan mengenai prinsip, metode dan alat-bantu untuk menilai efektivitas dan kepraktisan sistem teknologi informasi.
8. Desain infrastruktur, pengetahuan mengenai arsitektur dan topologi software, hardware dan jaringan, termasuk LAN, WAN, dan sistem telekomunikasi, komponen-komponennya, standar dan protokol yang terkait, dan bagaimana semuanya itu beroperasi dan ber-interaksi satu sama lain dan dengan software yang menyertainya.
9. Integrasi sistem; pengetahuan mengenai prinsip, metode dan prosedur untuk instalasi, integrasi dan optimasi komponen-komponen sistem informasi.
10. Siklus hidup sistem; pengetahuan mengenai konsep manajemen siklus hidup sistem yang dipergunakan untuk merencanakan, membangun, implementasi, mengoperasikan dan memelihara sistem informasi.
11. Pemahaman teknologi; pengetahuan mengenai perkembangan dan aplikasi baru teknologi informasi (hardware, software, telekomunikasi), teknologi yang menjanjikan dan aplikasinya ke proses bisnis, serta aplikasi dan implementasi sistem informasi untuk memenuhi kebutuhan perusahaan atau organisasi.

Selain itu, ada syarat pengetahuan yang biasanya tidak bersifat teknis tetapi harus dipelajari, yaitu:

1. Pengetahuan di bidang outsourced operations dan offshored operations.
2. Kapasitas untuk mengelola orang yang baik, baik di dalam maupun di luar tim.
3. Kapasitas untuk menginspirasi anggota tim.
4. Manajemen pemasok, karena kemungkinan akan berhubungan dengan pemasok, apalagi jika proyek yang ditangani banyak memerlukan berbagai instalasi infrastruktur maupun hardware (Tantra, 2012).

### **3.2.3 Keahlian Manajemen**

Seorang manajer proyek yang baik perlu mengetahui bidang bisnis dari proyek yang sedang ditangani. Dalam parameter proyek, manajer proyek juga harus melakukan peran manajer sumber daya manusia, manajer keuangan, dan manajer komunikasi. Pertimbangan untuk keterampilan manajerial meliputi hal-hal berikut:

1. Perencanaan, inisiasi, dan organisasi proyek.
2. Perekrutan staf dan mempertahankan mereka yang terbaik.
3. Negosiasi yang efektif.
4. Estimasi dan kontrol biaya yang akurat.
5. Pelaksanaan proyek dan pengendaliannya.
6. Presentasi dan pelaporan yang komunikatif.
7. Kepemimpinan dalam proyek dan personalia.
8. Manajemen risiko dan pengambilan keputusan.
9. Manajemen penanganan masalah yang efektif.
10. Meningkatkan dan mempertahankan kinerja tim yang baik.
11. Manajemen perubahan dalam organisasi (Tantra, 2012).

### **3.2.4 Keahlian Menangani Situasi**

Manajer proyek juga harus memiliki pengetahuan dan keterampilan yang diperlukan untuk menghadapi dan menyelesaikan masalah potensial yang dapat muncul selama pelaksanaan proyek dalam berbagai skenario, termasuk ketidaksepakatan dan situasi yang tidak pasti atau tidak jelas. Karena ini masalahnya, manajer proyek harus:

1. Fleksibel.
2. Keras dan tegas bila diperlukan.
3. Kreatif.
4. Menyerap banyak data dari berbagai sumber.
5. Sabar, tetapi bisa membedakan antara kesabaran dan melakukan tindakan.
6. Mampu menangani banyak tekanan terus-menerus yang sering tanpa henti.

Karena proyek seringkali tidak berjalan sesuai rencana, seorang manajer proyek yang terampil juga harus mampu menghadapi kejutan yang tidak terduga dan situasi yang tidak dapat diprediksi (Tantra, 2012).

### **3.3 Kualifikasi Manajer Proyek**

Kualifikasi yang membuat manajer proyek menjadi seseorang yang dapat dipercaya dan dianggap mampu memimpin pelaksanaan proyek dari awal sampai akhir adalah selanjutnya setelah mengetahui skill dan kemampuan apa saja yang dibutuhkan manajer proyek. Tentu saja, tidak perlu dikatakan lagi bahwa dia juga harus mampu memberikan hasil yang memenuhi harapan klien.

Manajer proyek dapat menunjukkan kualifikasinya sebagai seorang profesional dengan reputasi yang dimilikinya, yang dapat dibuktikan kepada klien. Selain dari CV,

profesionalisme seorang manajer proyek juga dapat disertifikasi melalui lembaga resmi, seperti misalnya dari *Project Management Institute* (PMI). Ada dua sertifikasi yang dikenal dari PMI, yaitu *Project Management Professional* (PMP) dan *Certified Associate in Project Management* (CAPM) (Mulyoto & Kurniali, 2013).

Untuk memperoleh sertifikat PMP, ada beberapa prasyarat yang harus dimiliki oleh seorang profesional, yaitu:

1. Kandidat adalah lulusan strata satu (S1) dari universitas yang diakui.
2. Minimal berpengalaman selama 4500 jam dalam manajemen proyek dengan pengalaman dalam lima kelompok proses/fase.
3. Masa proyek yang ditangani adalah selama tiga tahun dalam kurun waktu enam tahun sebelum pengajuan sertifikasi dilakukan.
4. Menunjukkan 36 bulan tersendiri (tidak *overlapping*) pengalaman dalam manajemen proyek.
5. Menunjukkan waktu hadir 35 jam dalam pendidikan di bidang manajemen proyek yang berhubungan dengan lingkup PMBOK.
6. Mengisi formulir verifikasi.
7. Mengikuti ujian yang berhubungan dengan pengetahuan, aplikasi dan analisis dalam manajemen proyek

Sedangkan syarat CAPM hampir sama kecuali pengalaman dalam manajemen proyek selama 1500 jam dan pendidikan selama 23 jam. Untuk mengikuti ujian seorang kandidat harus membayar sejumlah uang.

Selain sertifikasi PMP dan CAPM, juga ada sertifikasi yang dikeluarkan oleh *Computer Technology Industry Association* (CompTIA) yaitu Project+ yang memerlukan prasyarat selain mengikuti ujian.

Ingatlah bahwa sertifikasi berfungsi sebagai tanda bahwa keterampilan dan kredensial Anda diterima oleh organisasi yang sah. Tidak setiap klien menentukan bahwa manajer proyek yang akan mereka tunjuk harus bersertifikat (Tantra, 2012).

### **3.4 Tugas dan Tanggung Jawab**

Tugas yang diselesaikan oleh manajer proyek di seluruh proyek sering kali mencakup hal-hal berikut:

1. Mendefinisikan dan membatasi proyek dengan benar.
2. Identifikasi dan pilih material proyek dan sumber daya manusia.
3. Bertindak sebagai pemimpin tim proyek di setiap tahap.
4. Menghitung biaya proyek dan mengembangkan anggaran.
5. Kenali dan kendalikan setiap risiko dan risiko dalam suatu proyek.
6. Tetapkan dan pertahankan rencana proyek.
7. Mengontrol setiap modifikasi yang mempengaruhi proyek.
8. Memastikan semua penugasan proyek tetap sesuai jadwal dan tidak melebihi anggaran.
9. Mengenali politik organisasi dan menguasainya.

10. Mengontrol file proyek dan dokumen terkait.
11. Mengkomunikasikan dan memantau kemajuan proyek melalui rapat (*meeting*) dan pelaporan (*reporting*) (Lesmana & Antika, 2019).

Menjadi manajer proyek sistem informasi berarti ada beberapa tambahan tugas yang harus dilaksanakan, yakni (Tantra, 2012):

1. Identifikasi kebutuhan sistem informasi klien.
2. Analisis kebutuhan atau lingkungan sistem informasi.
3. Desain atau memimpin studi analitis, analisis biaya-keuntungan, atau riset lainnya.
4. Melakukan evaluasi, memantau, atau memastikan kelayakan proyek dibidang hukum, peraturan, kebijakan, standar atau prosedur.
5. Melakukan pembelian atau kontrak untuk layanan, peralatan, produk atau kebutuhan teknologi informasi lainnya.
6. Integrasi subsistem sistem informasi.
7. Menyusun strategi, rencana atau skenario pengujian (*testing*) sistem informasi.
8. Identifikasi standar atau kebutuhan konfigurasi infrastruktur atau manajemen perubahan (*change management*).
9. Berpartisipasi dalam kontrol perubahan (misalnya melakukan *review* permintaan perubahan konfigurasi).
10. Menyusun atau implementasi rencana dan prosedur keamanan sistem informasi.
11. Memastikan pelatihan dan dokumentasi yang memadai, yang berkaitan dengan produk tersedia dan dapat dimanfaatkan oleh pengguna.

Ada beberapa literatur mengenai manajemen proyek sistem informasi yang menggolongkan tugas-tugas manajer proyek berdasarkan *project management knowledge areas*. Ada delapan bidang manajemen yang termasuk dalam knowledge area ini, yaitu (Ardi et al., 2020):

1. Manajemen Integrasi (*Integration Management*), menyusun dan mengarahkan proyek secara keseluruhan.
2. Manajemen Ruang lingkup (*Scope Management*), perencanaan untuk mendefinisikan dan mengatur ruang lingkup proyek.
3. Manajemen Waktu dan Biaya (*Time and Cost Management*), merencanakan jadwal, mengalokasikan sumber daya, dan mengelola serta mengawasi keuangan proyek.
4. Manajemen Kualitas (*Quality Management*), berfokus pada apakah suatu proses memiliki kualitas yang cukup tinggi untuk memenuhi atau bahkan melebihi standar yang ditetapkan.
5. Manajemen Sumberdaya Manusia (*Human Resource Management*), menjamin bahwa tim mereka bergerak ke arah yang diinginkan, para manajer harus memperhatikan tim mereka, menginspirasi dan memotivasi mereka, dan memberikan kegembiraan..
6. Manajemen Komunikasi (*Communication Management*), manajer dituntut untuk membuat rencana komunikasi dan memastikannya terjadi baik secara vertikal maupun horizontal secara terbuka.

7. Manajemen Risiko (*Risk Management*), untuk mencegah gangguan proyek di tengah pelaksanaan, berbagai potensi risiko dalam proyek harus dikenali dan strategi mitigasi dan antisipasi ditetapkan.
8. Manajemen Pengadaan (*Procurement Management*), mengelola pengadaan berbagai kebutuhan proyek melalui vendor atau pemasok agar proyek dapat berjalan lancar dan berakhir sukses.

Tentunya ada kewajiban yang harus dipikirkan saat menjalankan tugas sebagai manajer proyek sistem informasi. Tugas ini meluas tidak hanya ke klien tapi juga ke setiap anggota grup manajemen proyek, atau pemangku kepentingan. Tugas manajer proyek termasuk dalam salah satu dari tiga kategori, yaitu (Tantra, 2012):

1. Fungsional
  1. Memahami *requirements* proyek.
  2. Membuat rencana dan mengontrol proyek.
  3. Melaksanakan atau implementasi proyek.
  4. Menyelesaikan proyek tepat waktu, dalam anggaran, dan mengikuti spesifikasi yang ditentukan.
2. Organisasi
  - a. Memahami konteks organisasi.
  - b. Melaksanakan wewenangnya.
  - c. Memahami cara memberikan pengaruh terhadap orang lain.
  - d. Menyelesaikan konflik.
  - e. Memahami situasi politik internal organisasi.
3. Tim
  - a. Memilih dan mengembangkan staf atau anggota tim.
  - b. Mendelegasikan tugas pada anggota tim
  - c. Menjadi penengah antara tim dengan pihak manajemen organisasi atau perusahaan klien
  - d. Komunikasi secara efektif dengan pihak lain.
  - e. Menunjukkan kepemimpinan.
  - f. Memberikan motivasi kepada anggota tim.
  - g. Membangun tim yang kuat, kompak dan handal.

### **3.5 Kualitas Manajer Proyek**

Berikut ini akan dijelaskan kriteria manajer proyek yang berkualitas, yang merupakan hasil penelitian ESI International, sebuah lembaga pelatihan manajer proyek internasional. Dengan mengikuti poin-poin di bawah ini Anda akan memiliki pedoman dalam meniti karir manajer proyek sistem informasi.

1. Berbagi inspirasi Visi

Manajer proyek yang kompeten dapat melihat gambaran besarnya proyek dan melaksanakannya. Seorang pemimpin visioner dapat menerima dan menerapkan

perubahan. Keterampilan kepemimpinan manajer proyek juga harus mencakup kemampuan untuk menginspirasi anggota timnya untuk memiliki visi bidang mereka dan melaksanakannya.

## 2. Komunikator yang baik

Keahlian kedua yang terpenting adalah kemampuan berkomunikasi baik secara horizontal (dalam tim) maupun vertikal (dengan klien). Manajer proyek harus dapat mengungkapkan tujuan, tugas, kinerja, harapan, dan komentar dengan cara yang jelas dan ringkas. Manajer proyek dapat mempromosikan pencapaian tim dan individu dengan menguraikan dengan jelas bagaimana mendapatkan tujuan yang dibutuhkan dan memajukan karir mereka melalui komunikasi yang efektif.

## 3. Integritas

Satu hal yang harus diperhatikan oleh manajer proyek adalah bahwa yang menjadi panduan bagi tim adalah tindakannya, bukan kata-kata. Kepemimpinan yang baik juga harus menunjukkan komitmen dan tindakan secara etis. Jadi tidak sekadar diucapkan saja. Kepemimpinan yang berintegritas adalah berdasarkan seperangkat nilai yang dipegang dan dibagikan kepada tim, serta secara konsisten bertindak dengan selalu mengacu kepada nilai-nilai tersebut.

## 4. Antusias

Manajer proyek yang antusias akan menginspirasi dan memotivasi seluruh tim, membuat mereka lebih bersemangat sebagai hasilnya. Kompak dan selalu optimis. Untuk berkomitmen mencapai tujuan Anda, Anda harus bertindak dengan semangat dan optimisme sebagai pemimpin yang antusias. Dengan melakukan ini, Anda dapat menunjukkan keyakinan Anda bahwa proyek dapat diselesaikan dan tujuan tercapai. Tim proyek akan mendapat manfaat jika Anda juga secara konsisten optimis.

## 5. Empati

Simpati dan empati tidak sama. Empati adalah kemampuan untuk memperhatikan kondisi seseorang tanpa terlibat secara emosional di dalamnya, berlawanan dengan belas kasihan, yang melibatkan keterlibatan secara emosional. Tunjukkan kepada anggota tim bahwa Anda peduli dengan kinerja mereka di tempat kerja dan ingat waktu setelah bekerja. Anda tidak bisa tegas dengan hanya berpartisipasi di tempat kerja atau dengan terus-menerus mendiskusikan pekerjaan baik selama maupun di luar jam kerja. Memiliki empati terhadap orang lain akan meningkatkan dinamika tim Anda, dan sebaliknya, mereka akan selalu ada untuk Anda.

## 6. Kompeten

Menjadi ahli dalam hal-hal teknis bukanlah persyaratan untuk manajer proyek yang kompeten. Seiring berkembangnya disiplin manajemen proyek, bakat dan keahlian manajer proyek dalam menyelesaikan proyek dengan sukses saat ini diberi bobot yang lebih besar dalam penugasan daripada keterampilan teknisnya. Atas dasar pengalaman praktis yang sukses, Anda akan lebih dihormati. Juga, Anda harus dapat menantang, memotivasi, memberi contoh, dan menginspirasi staf Anda untuk menjadi manajer yang efektif.

#### 7. Mampu mendelegasikan tugas

Kepercayaan adalah hal yang paling esensial dalam hubungan antara manajer proyek dengan timnya. Anda harus memberi kepercayaan kepada anggota tim untuk menjalankan tugas. Tetapi Anda juga harus selalu memeriksa dan mengontrol pekerjaan mereka, seberapa besar Anda mendelegasikan tugas, dan seberapa jauh Anda mengizinkan mereka untuk berpartisipasi. Orang yang tidak dapat memberikan kepercayaan kepada anggota timnya, biasanya akan gagal dalam memimpin dan hanya mampu bertindak sebagai micro-manager atau pada akhirnya mengerjakan semua hal sendiri.

#### 8. Mampu bekerja di bawah tekanan

Ingatlah bahwa, bertentangan dengan apa yang digambarkan dalam buku teks, tidak setiap proyek diselesaikan di dunia nyata sesuai jadwal, sesuai anggaran, dan tanpa masalah. Setiap proyek bisa jadi akan menghadapi masalah, tidak peduli seberapa besar atau kecilnya. Anda harus bisa menggairahkan tim dengan menyelesaikan berbagai masalah implementasi proyek sebagai tantangan. Jangan pernah menunjukkan kepada tim bahwa Anda tidak siap menghadapi tekanan..

#### 9. Mampu membangun tim

Kemampuan mendasar dari seorang manajer proyek adalah membangun tim. Ketika sebuah tim pertama kali dibentuk, kemungkinan para anggotanya tidak mengenal satu sama lain dengan baik dan memiliki latar belakang, keterampilan, dan bidang keahlian yang berbeda-beda. Manajer proyek perlu mengetahui prosedur dan dinamika kelompok yang diperlukan untuk mengubah tim yang terjalin longgar menjadi tim yang dapat bekerja sama dan saling mendukung. Selain itu, Anda harus dapat mengenali kepribadian unik setiap anggota tim dan tahu kapan harus memanfaatkan bakat masing-masing.

#### 10. Mampu menyelesaikan masalah.

Meskipun penyelesaian masalah dapat dilakukan bersama-sama tim, tetapi manajer proyek juga diharapkan memiliki kemampuan pemecahan masalah sendiri secara kreatif jika diperlukan, sehingga tidak perlu menggelar rapat yang berlarut-larut jika masalah itu dapat diselesaikan dengan cepat.

### **3.6 Manajer Proyek Sebagai Pemimpin Visioner**

Seperti yang digambarkan di atas, Anda harus menyadari bahwa seorang manajer yang kompeten perlu memiliki pemahaman teknis, visi bisnis, keterampilan komunikasi, soft management skills, dan kemampuan untuk merencanakan, mengkoordinasikan, dan melaksanakan. Pada dasarnya, Anda adalah seorang manajer sekaligus pemimpin.

Oleh karena itu, meskipun tim Anda telah bekerja keras dan berkompeten dalam setiap aspek tugas masing-masing, Anda tetap membutuhkan seorang pemimpin yang dapat melihat keseluruhan proyek dari atas, tetap fokus pada visi yang telah ditetapkan sejak awal dan menjalankan fungsi kontrol selama proyek berlangsung agar tetap pada jalur yang benar.

## BAB IV FASE INISIASI

### 4.1 *Feasibility Study*

Studi kelayakan (*feasibility study*) adalah tindakan yang dilakukan untuk menentukan apakah suatu proyek layak untuk direalisasikan atau tidak (Dimiyati & Nurjaman, 2016). Proyek tidak layak jika hanya menghasilkan produk atau jasa yang hanya berfungsi satu kali saja; atau jika proyek yang akan dilaksanakan itu tidak memberi manfaat apapun selain hanya menambah panjang birokrasi.

Studi kelayakan terbagi dalam beberapa aktivitas atau satuan kerja (*job*):

#### 1. Wawancara

Wawancara harus ditujukan pada pemilik proyek dan sponsor, apa tujuan yang diharapkan dengan pembangunan sistem ini dan berapa lama waktu yang mereka berikan hingga sistem ini dapat diselesaikan dan digunakan sebagaimana harapan mereka. Anda perlu mengarahkan ekspektasi pemilik proyek agar tetap realistis dan memungkinkan untuk dicapai dengan tenggat waktu yang ditentukan. Selain itu, pastikan juga Anda mendapatkan akses terhadap bagian organisasi dan stafnya untuk melakukan wawancara lebih detail, bila perlu secara tertulis. Pastikan juga pemilik proyek telah menginformasikan kepada pihak-pihak terkait agar Anda tidak terkendala dalam mengakses informasi yang diperlukan.

Selain kepada para staf yang bekerja langsung, sebaiknya wawancara juga dilakukan pada staf yang tidak berhubungan langsung, yaitu bagian memasok pekerjaan pada mereka, ataupun bagian yang menerima hasil kerja mereka.

Wawancara tidak selalu mudah dilakukan oleh karena kendala waktu. Jika para staf yang diwawancarai juga tetap melakukan tugas mereka, tentu wawancara tidak efektif karena terinterupsi oleh pekerjaan. Sebagai manajer proyek, Anda harus menentukan waktu untuk wawancara di hari-hari di mana para staf tersebut lebih senggang dan meminta kepada atasan atau pihak yang bertanggung jawab terhadap bagian tersebut untuk memberikan waktu kepada staf itu tanpa terganggu oleh rutinitas pekerjaan.

Delegasikan pekerjaan ini kepada mereka yang mampu menyusun pertanyaan secara sistematis dan mencatat hasil yang didapatkan pada dokumen yang tersedia. Pencatatan dapat dilakukan dengan format yang terstruktur dan sistematis, sehingga jika proyek ini nantinya layak untuk diteruskan, maka hasil wawancara ini bisa menjadi dasar untuk menyusun analisis kebutuhan (*requirement analysis*) lebih lanjut. Tugas tim Anda hanya mengumpulkan informasi kelayakan dari proyek yang akan dilaksanakan, dan bukan untuk menyusun *requirement analysis*, sehingga tidak perlu membebani tim untuk mendapatkan informasi sedetail mungkin.

#### 2. Kunjungan ke lokasi (*Site visit*)

Misalnya, proyek yang akan datang adalah sistem ritel. Akibatnya, tim Anda juga harus mengunjungi lokasi toko retail untuk mempelajari bagaimana bisnis dijalankan. Grup dapat melacak dan mengawasi semuanya mulai dari menerima barang hingga mengelola inventaris, penjualan, dan keuangan. Jika sebuah perusahaan memiliki

banyak toko, penting juga untuk memperhatikan bagaimana manajemen rantai pasokan dan proses distribusi barang dilakukan. Dari sana, Anda dapat mengamati bagaimana sistem kerja manual dan digital diterapkan, cara kerja kontrol, cara mengantisipasi masalah keamanan, dan cara menanganinya jika muncul.

Saat mengunjungi lokasi sebaiknya tim dapat ditemani oleh bagian dari stakeholder, atau oleh salah seorang anggota tim manajemen pemilik proyek, sehingga dapat lebih mudah mengakses lokasi-lokasi yang penting untuk diamati. Dengan pergi ke lokasi, proyek dapat dirancang untuk memenuhi kondisi lapangan dan memberikan hasil yang efektif.

### 3. Pengumpulan Dokumen

Setelah melakukan wawancara dan mengunjungi lokasi, sebagai bahan untuk masuk ke tahap penyusunan hasil kerja, maka dokumen-dokumen yang digunakan dalam sistem yang sudah berjalan, baik manual maupun terkomputerisasi, harus dikumpulkan. Pengumpulan dokumen bisa dilakukan sambil melakukan wawancara. Umumnya dokumen-dokumen dalam suatu sistem terdiri atas:

#### a. Formulir Input (*Input Forms*)

Adalah formulir-formulir yang digunakan untuk mencatat data-data yang diperlukan dalam melakukan suatu unit proses. Misalnya saat melakukan proses pembelian barang, maka staf yang bersangkutan akan mencatat data pemasok (jika belum ada), lalu membuat form purchase order berdasarkan penawaran dari pemasok tersebut. Formulir input yang di sini adalah formulir data pemasok dan suna penawaran dari pemasok.

#### b. Formulir Output (*Output Forms*)

Adalah formulir-formulir yang merupakan hasil suatu proses yang kemudian akan diteruskan untuk digunakan bagian lain dan/atau pihak di luar organisasi perusahaan. Misalnya, seperti contoh di atas adalah *purchase order*, *invoice*, *delivery order*, dan sebagainya. Kadang-kadang formulir ini bisa terdiri atas beberapa rangkap yang didistribusikan ke berbagai bagian. Bisa juga berupa nota sederhana yang ditulis secara manual.

#### c. Laporan-laporan

Semua proses seringkali berujung pada laporan yang diberikan kepada manajemen dan dewan direksi yang tepat. Ada laporan yang menunjukkan data khusus untuk pemeriksaan, ada yang berupa rekapitulasi laporan manajemen, ada pula yang bersifat analitis untuk digunakan dalam pengambilan keputusan. Terlepas dari transaksi objektif, data dapat dimasukkan dalam beberapa laporan secara subyektif, oleh karena itu tim Anda harus mengetahui sumber data untuk laporan tersebut. Ini harus diperhitungkan untuk menilai kelayakan proyek karena hal itu kadang-kadang terjadi (Tantra, 2012).

Secara formal, hasil studi kelayakan dapat disusun dalam bentuk laporan yang ditujukan kepada *stakeholder*. Laporan ini berisi antara lain:

### 1. Tujuan Studi Kelayakan

Jelaskan alasan di balik studi kelayakan yang sedang dilakukan. Tidak harus rinci, tapi setidaknya bisa menyampaikan gambaran tentang apa yang akan dicapai jika studi kelayakan selesai.

### 2. Latar Belakang

Jelaskan mengenai faktor-faktor yang mendorong diadakannya studi kelayakan tersebut. Bisa berupa masalah-masalah yang hendak diatasi ataupun juga pengembangan sistem kepada tingkatan yang lebih baik.

### 3. Solusi yang Diajukan

Hasil studi kelayakan berbentuk solusi yang diajukan klien untuk mengatasi masalah dari bagian latar belakang. Jika demikian, opsi alternatif juga dapat disiapkan dan diajukan.

### 4. Analisis Biaya-Manfaat

Analisis biaya-manfaat dapat dimasukkan untuk melengkapi temuan studi kelayakan untuk mengevaluasi efektivitas solusi yang disarankan dan menunjukkan apakah biaya akan masuk akal mengingat hasil yang diprediksi.

Contoh struktur laporan studi kelayakan dapat dilihat pada appendix 1(Tantra, 2012).

## 4.2 *Requirements Analysis*

*Requirements Analysis* sangat penting karena merupakan dasar dari perencanaan proyek untuk menentukan apa saja yang perlu dipersiapkan agar proyek dapat terlaksana dengan baik. juga merupakan fondasi dari aktivitas berikutnya dalam pengembangan sistem dan manajemen proyek (Nugroho, 2016). Oleh karena itu, tidak hanya manajer proyek, tetapi semua *stakeholder* harus berkomitmen dalam proses yang efektif. Yang dimaksud dengan proses adalah:

1. Penyusunan (mengumpulkan, menganalisis, spesifikasi dan validasi ).
2. Manajemen (melaksanakan sesudah disepakati) (Tantra, 2012).

### 4.2.1 Definisi

Menurut standar IEEE (*Guide for Developing System Specifications*), adalah pernyataan mengenai:

1. Fungsionalitas sistem (kapabilitas).
2. Dapat divalidasi.
3. Harus sesuai dengan sistem yang berjalan.
4. Solusi untuk masalah klien.
5. Memenuhi kriteria dengan kondisi yang terukur dan dibatasi oleh constraints.

Intinya, adalah kebutuhan proyek yang didokumentasikan, dan dikumpulkan untuk mengidentifikasi *constraints* yang spesifik (*scope*) dari setiap komponen proyek dan berfungsi sebagai dasar untuk setiap aktivitas yang berlangsung dalam proyek (Tantra, 2012).

#### 4.2.2 Jenis-jenis

Ada beberapa jenis , tergantung sumber datanya, yaitu (Nugraha, 2011):

1. *Business*

Terdiri atas bisnis proses dari sistem yang akan dibangun, batasan-batasan (*constraints*) seperti biaya, sumberdaya, waktu dan sebagainya. Hal ini bisa berasal dari wawancara yang dilakukan terhadap staf yang melakukan pekerjaan dari bisnis proses tersebut, ataupun atasannya, yang bisa dituangkan dalam bentuk *workflow processes*.

2. *Stakeholder*

Terdiri dari sekumpulan spesifikasi untuk sistem yang akan dikembangkan yang selaras dengan tujuan internal dan eksternal bisnis atau organisasi.

3. *End-User*

Adalah kebutuhan dari staf yang berinteraksi secara langsung maupun tidak langsung terhadap sistem yang akan dibangun. Selain proses, biasanya juga menyangkut dokumentasi (petunjuk penggunaan dan administrasi), bahkan sampai kepada antarmuka pengguna. Yang perlu diwaspadai adalah tidak perlu semua kebutuhan ini dipenuhi, karena sebagian kebutuhan ini lebih bersifat nice to have daripada *musthave*.

4. *System*

Disusun berdasarkan *business objectives* dan *stakeholder* , yang disusun dengan pendekatan teknis secara formal dan terstruktur. *System* merupakan tingkatan tertinggi dari keseluruhan sistem, yang dapat terdiri dari beberapa subsistem seperti subsistem *hardware*, subsistem *software*, subsistem *network*, dan sebagainya.

5. *Software*

Adalah *requirements* yang secara tepat menguraikan persyaratan pengguna untuk fungsionalitas perangkat lunak. Dokumen perangkat lunak ini akan berfungsi sebagai titik awal yang penting untuk fase desain dan pengembangan, membantu memahami semuanya dan menggabungkannya ke dalam sistem yang akan dibuat..

#### 4.2.3 Menyusun *System Specifications* (SRS)

SRS pada dasarnya adalah deskripsi tertulis dari sistem yang diharapkan klien dan hubungannya dengan jumlah waktu yang tersisa sebelum desain atau pengembangan dimulai. Ketika datang untuk memahami persyaratan seperti yang disepakati dan asalkan tidak ada perubahan sama sekali, dapat dikatakan bahwa SRS berfungsi sebagai semacam jaminan dua arah antara klien dan tim proyek. (Hartono, 2018).

Dokumen SRS berisi pernyataan secara eksplisit dan dalam bahasa yang jelas mengenai fungsionalitas dan kemampuan yang harus ada dari sistem yang akan dibangun, baik dalam bentuk *software* aplikasi, ERP, E-Commerce dan sebagainya, termasuk batasan-batasan (*constraints*) yang dapat ditoleransi oleh sistem tersebut. SRS juga berfungsi sebagai *blueprint* untuk pelaksanaan proyek dengan waktu, biaya dan lingkup yang terkendali dengan baik. Dokumen SRS bisa disebut sebagai induk dari dokumen-

dokumen lain dalam proyek karena dokumen-dokumen lain seperti spesifikasi desain (*design specification*), *statement of work*, spesifikasi arsitektur sistem (*system architecture specification*), rencana testing dan validasi, dan perencanaan sistem, harus selalu mengacu pada SRS.

Perlu dicatat bahwa SRS hanya mencakup informasi fungsional dan nonfungsional, itu tidak menawarkan rekomendasi desain, perbaikan potensial untuk masalah teknologi atau bisnis klien, atau detail lain yang tidak terkait dengan pemahaman tim proyek tentang sistem yang diharapkan klien.

Penyusunan SRS dilakukan berdasarkan informasi yang telah didapatkan dari studi kelayakan yang dilakukan sebelumnya, dengan tambahan yang bisa didapatkan dari analisis lanjutan dari materi yang telah didapat. Karyawan/staff yang memiliki pengetahuan teknis harus digunakan untuk membuat SRS karena hal ini akan membantu tim pengembangan proyek menerjemahkan masukan klien ke dalam bentuk teknis yang dapat mereka pahami. Ia juga dapat bertindak sebagai "filter" untuk data yang diterima, memastikan bahwa hanya data yang benar-benar relevan dengan standar sistem yang dicatat. Dia harus berhati-hati untuk tidak menggunakan bahasa yang terlalu teknis yang tidak dapat dipahami oleh pemangku kepentingan awam atau pembaca non-teknis.

Menurut standar IEEE, SRS harus menjelaskan sembilan hal, yakni:

1. Interface
2. Kapabilitas fungsional
3. Tingkat kinerja
4. Struktur data
5. Keamanan
6. Reliability
7. Proteksi privasi
8. Kualitas
9. Batasan-batasan (Tantra, 2012).

Untuk mengorganisasikan hal-hal tersebut dalam bentuk dokumen terstruktur, maka diperlukan template yang sesuai. Template standar berdasarkan IEEE Standard 830-1998 untuk SRS dapat dilihat pada appendix 2. Template tidaklah baku, tetapi dapat dimodifikasi sesuai keperluan proyek.

Beberapa tip dalam menyusun SRS yang terstruktur:

1. Gunakan kalimat aktif yang mengandung kata kerja. Contohnya, "PO menggunakan batas *Term of Payment* dengan mengambilnya dari data *Supplier* yang dipilih...". Bukan "Batas *Term of Payment* diambil dari data *Supplier* untuk dipakai dalam PO..."
2. Gunakan kalimat definitif, yang tidak menimbulkan pengertian ganda. Misalnya kalimat, "Sistem memungkinkan *user* yang berbeda untuk membuka dokumen PO yang dibuat oleh *user* lain..." akan membingungkan karena kata "memungkinkan". Jika perlu, berikan penjelasan yang tegas meskipun definisinya menjadi panjang. "Sistem hanya memberikan akses untuk membuka PO pada *user* yang membuatnya. Jika *user* lain akan membuatnya, *user* tersebut memerlukan otorisasi yang lebih tinggi levelnya."

3. Gunakan kalimat yang terstruktur dengan konteks yang jelas dan dipahami oleh pengguna SRS. Contoh, “Setiap cabang bisa membuat PO untuk memesan barang ke Supplier, kemudian dikirimkan ke pusat lalu menggunakan PO tersebut untuk verifikasi *Invoice* dari *Supplier*.” Akan lebih terstruktur jika dituliskan sebagai berikut, “Setiap cabang memiliki otorisasi untuk membuat PO ke *Supplier*. Cabang mengirimkan data PO ke Pusat. Pusat menggunakan data PO dari cabang untuk verifikasi dengan *Invoice* dari *Supplier*.”
4. Hindari penggunaan kalimat yang terlalu teknis. Sebisa mungkin hindari istilah yang sulit dimengerti orang awam. Jika bisa menyampaikan “Transfer data bisa menggunakan fasilitas yang disediakan sistem, secara otomatis maupun manual...” maka hindari menyampaikannya dengan kalimat “Gunakan ftp untuk transfer data otomatis dengan *Ip Address* yang sudah diset secara otomatis, atau kirimkan dengan *Attachment* email...”
5. Penjelasan detail secara spesifik. Jangan sampai ada bagian penting yang dijelaskan terlalu umum. Misalnya, “Setiap *Invoice* harus melalui tahapan verifikasi.” Padahal verifikasi itu harus dijelaskan lebih detail lagi dengan kalimat “Verifikasi meliputi pemeriksaan piutang yang telah jatuh tempo dan batas piutang.” Jika ada otorisasi khusus untuk melewati proses “verifikasi”, maka juga harus dijelaskan secara detail.
6. Hindari penjelasan implementasi detail. Kadang kala dalam menyusun *requirements*, godaan ini sangat besar. Misalnya, “*Invoice* akan membentuk jurnal penjualan harian. Untuk itu ada satu prosedur yang akan melakukan update ke tabel jurnal.” Kalimat mengenai adanya prosedur tidak perlu ada. Jika memang perlu dijelaskan, gunakanlah kalimat yang terstruktur seperti yang dijelaskan di atas.
7. Jika ada, jelaskan pula skenario alternatif, terutama untuk situasi tertentu dalam sistem yang merupakan pengecualian.
8. Hindari penjelasan mengenai desain yang masih prematur. Yang perlu disusun adalah segala sesuatu yang akan menjadi bahan dalam desain, yang menjadi bagian dan tugas dari tahap selanjutnya (Tantra, 2012).

#### 4.2.4 Perubahan dan Manajemen

Konsekuensi utama dari masalah pada adalah pengerjaan ulang (*rework*), mengulangi sesuatu yang seharusnya sudah selesai. Menurut penelitian Barry W. Boehm dan Philip N. Papaccio, *rework* dapat mengkonsumsi 30 sampai 50 persen dari total *cost* (baik biaya maupun waktu) proses pengembangan, dan kesalahan dari *requirements* dapat menyebabkan pemborosan *cost* untuk *rework* 70 sampai 85 persen. Bahkan, menurut Grady, *cost* yang timbul akan lebih tinggi jika koreksi tidak dilakukan pada tahap awal proyek. Artinya, semakin dini kita dapat mendeteksi dan melakukan koreksi terhadap kesalahan, maka akan mengurangi pemborosan sumberdaya. Pasti sangat menyulitkan jika harus melakukan *rework* saat sudah memasuki tahap finalisasi proyek.

Secara teoritis, jika lebih banyak waktu dihabiskan untuk penyusunan pada tahap awal proyek, lebih banyak waktu akan disimpan dalam tahap implementasi berikutnya.

Namun demikian, masalah yang sering kami temui adalah klien menuntut untuk mengetahui hasil dari setiap tahap proyek sesegera mungkin. Klien mungkin menjadi tidak sabar dan percaya bahwa tidak ada kemajuan pada proyek meskipun tahap ini juga merupakan bagian darinya jika kita menghabiskan lebih banyak waktu untuk terus memperbarui dokumen dengan mengadakan banyak pertemuan, menganalisis temuan wawancara, dan mengumpulkan dokumen untuk mendapatkan SRS yang dipahami oleh kedua belah pihak.

Kadang kala bisa terjadi yang berubah karena proses bisnis klien memang selalu menyesuaikan dengan keadaan. Hal ini bisa terjadi pada perusahaan baru, atau perusahaan yang sedang melakukan rekayasa ulang (*re-engineering*) untuk meningkatkan efisiensi dan kinerja operasionalnya. Idealnya, pengembangan sistem informasi dilakukan sesudah proses bisnis ditetapkan. Namun, klien menganggap pengembangan sistem justru harus dilakukan pada saat bersamaan dengan proses *re-engineering* ataupun perubahan proses bisnis yang berlangsung, agar semuanya dapat siap bersamaan.

Salah satu cara untuk mengatasi situasi di mana terjadi pembaharuan terus-menerus pada yang kemudian akan berpengaruh pada desain dan pengembangan sistem adalah dengan menggunakan metode *Agile Development Project*, yang menggunakan pendekatan inkremental, yaitu membangun sebagian kecil dari sistem dalam waktu yang relatif cepat, dan bersama klien melakukan *review*. Demikian terus menerus sehingga tim pengembang dalam proyek dapat mengikuti keadaan lingkungan dan proses bisnis yang berubah-ubah. Tetapi, meskipun demikian, sebagai suatu proyek, tetap harus dibatasi oleh batasan waktu yang telah didefinisikan pada awal proyek. Dengan demikian maka tidak akan terjadi proyek yang tidak berujung oleh karena adanya pengembangan yang terus-menerus.

Jika perubahan terjadi sekali-sekali di sepanjang proyek, maka diperlukan penanganan manajemen dengan menggunakan alat bantu. Contohnya adalah *Micro Focus Optimal Trace* dan *Serena RTM*. Setiap perubahan harus di-*review* dan diukur seberapa besar pengaruhnya terhadap proyek yang sudah berjalan, sebelum disetujui untuk diimplementasikan. Dalam proyek besar, biasanya dibentuk suatu tim yang berwenang untuk mengontrol dan menyetujui perubahan-perubahan yang terjadi pada *requirements* sehingga dampaknya juga sudah siap untuk diantisipasi. Perencanaan proyek dan tahapan berikutnya harus disesuaikan lagi agar tetap dapat dilanjutkan dengan segala konsekuensi akibat dari perubahan *requirements* yang telah disepakati tersebut.

Dengan adanya manajemen *requirements* diharapkan dapat memberikan jalan keluar bagi perubahan yang terjadi pada *requirements* s proyek sudah memasuki tahap selanjutnya, seperti tahap desain atau malah sudah dalam tahap pengembangan (*development*). Manajemen bisa membantu dalam hal:

1. Memantau status sehingga dapat membantu dalam mengukur progres yang telah dicapai dalam tahap pelaksanaan proyek berikutnya karena pada dasarnya tahap setelah inisialisasi adalah tahap pemenuhan terhadap *requirements* yang telah disepakati, sesuai dengan jadwal dan sumberdaya yang telah dialokasikan.

2. Komunikasi dengan *stakeholder* mengenai perubahan *requirements* yang terjadi sehingga semua pihak siap mengantisipasi setiap dampak dan konsekuensinya.
3. Pemahaman yang lebih baik untuk setiap *requirements*, termasuk mengenai prioritas, status, metode verifikasi, asal-usul, penanggungjawab, rencana implementasi, dan sebagainya. Dengan adanya informasi ini maka manajer proyek dapat memonitor secara lebih detail untuk setiap *requirements* klien.
4. Pencatatan secara historikal perubahan *requirements*, termasuk informasi atas permintaan siapa, dampak dan implikasi, penerapan dan konsekuensi yang harus dihadapi. Dengan demikian, jika terjadi penggantian personal dalam proyek, terutama dari pihak klien yang terlibat dalam perubahan ini, maka jejak perubahan dapat ditelusuri dan dipertanggungjawabkan.
5. Dengan menerapkan tools seperti *Micro Focus Optimal Trace* atau *Serena RTM*, manajemen *requirements* dapat dikembangkan lebih jauh untuk memberikan informasi dalam hal analisis dampak perubahan, akses kontrol terhadap *requirements* serta penggunaan kembali database *requirements*, sehingga menjadi semacam *knowledge base* (Tantra, 2012).

#### **4.2.5 Requirements Sebagai Penyebab Kegagalan Proyek**

Dari berbagai studi ditemukan bahwa 70% dari proyek yang gagal adalah karena *requirements* yang tidak lengkap dan akurat. Bahkan menurut Carey Schwaber, “cacat” dalam *requirements* adalah sumber dari mayoritas masalah yang ditemukan saat testing. Jadi seperti yang dikatakan sejak awal, mencegah kegagalan proyek haruslah dimulai sejak awal, yaitu saat penyusunan *requirements*. Berikut ini merupakan hal-hal yang menyebabkan kesalahan dalam menyusun *requirements* (Tantra, 2012).

1. Klien tidak benar-benar mengerti apa yang sebenarnya mereka inginkan.  
Masalah yang paling umum dalam penyusunan *requirements* adalah klien memiliki gambaran yang belum jelas mengenai sistem yang mereka kehendaki. Maka, adalah tugas bagi tim penyusun *requirements* untuk memberikan pertanyaan yang tepat dan melakukan analisis yang diperlukan agar dapat menyusun SRS yang benar. Untuk mengatasi hal ini, beberapa hal yang harus diperhatikan adalah:
  - a. Pastikan tim Anda meluangkan waktu yang diperlukan di awal proyek untuk memahami tujuan, keluaran, dan ruang lingkupnya.
  - b. Pahami secara akurat semua praduga klien dan evaluasi secara kritis risiko dan manfaat yang akan diwujudkan.
  - c. Berusahalah untuk mengartikulasikan visi proyek secara tertulis, menyoroti tujuan uniknya atau keunggulan klien serta penyelesaian yang diantisipasi dari masalah bisnis saat ini.
  - d. Untuk memastikan bahwa semua pihak setuju dengan apa yang diharapkan dari proyek, minta klien untuk membaca, memahami, dan menandatangani SRS yang telah selesai.

2. *Requirements* berubah saat proyek berlangsung.

Klien sering menyadari kekurangan dalam rencana awal setelah proyek dimulai dan desain atau prototipe terlihat, yang mendorongnya untuk merevisi kriteria untuk mencerminkan pengetahuan baru yang telah ditemukannya. Anda harus bersikap akomodatif sekaligus berhati-hati sebagai manajer proyek. Jika perubahannya substansial dan mendasar, niscaya akan mengganggu kemampuan proyek untuk dilanjutkan. Tapi, seperti yang telah dibahas sebelumnya, hal seperti itu jarang terjadi. Biasanya, klien ingin memodifikasi alur kerja ini agar sesuai dengan keadaan tertentu. Selain itu, klien terkadang membuat penyesuaian yang diinginkan. Anda harus berikap tegas untuk kasus seperti ini. Untuk mengantisipasi kemungkinan timbulnya permasalahan oleh karena perubahan *requirements*, Anda perlu:

- a. Memiliki prosedur yang jelas untuk menerima, menganalisis dan menggabungkan permintaan perubahan *requirements*, dan pastikan bahwa klien benar-benar memahami perubahan yang mereka usulkan. Dokumen untuk mengajukan perubahan ini bisa dilihat pada appendix 1.
- b. Tentukan milestone pada fase pengembangan, di mana jika sudah melaluinya, maka perubahan tidak diperbolehkan lagi. Misalnya, perubahan besar/mayor tidak diterima jika pengembangan sudah mencapai 75%.
- c. Pastikan bahwa perubahan yang diajukan (dan persetujuannya) dikomunikasikan dengan seluruh *stakeholder*, bersama dengan alasannya, dan rencana proyek harus diperbaharui segera.
- d. Klien menetapkan batasan waktu yang sewenang-wenang. Sering terjadi klien mengklaim bahwa sistem informasi yang dia butuhkan sangat mendesak sehingga dia menetapkan tenggat waktu sendiri. Jika kita menganggap itu tanpa melakukan studi menyeluruh dan pemahaman tentang ruang lingkup proyek dan sumber daya yang tersedia, proyek pasti akan mengalami kesulitan. Kendala waktu yang terlalu ketat bermasalah meskipun klien menawarkan sumber daya yang substansial karena kemungkinan beberapa fase-fase pengembangan harus diselesaikan secara berurutan dan tidak dapat diselesaikan secara paralel (yang bisa lebih cepat ketika sumber daya bekerja sama). Yang dapat Anda lakukan untuk menyiasatinya adalah:
  - 1) Konversikan SRS menjadi perencanaan proyek, uraikan tasks dan sumberdaya yang dibutuhkan pada setiap tahap, dan lakukan pemodelan skenario best-case, middle-case dan worst-case.
  - 2) Pastikan bahwa perencanaan proyek memperhitungkan ketersediaan sumberdaya dan menyiapkan waktu untuk testing dan pengujian kualitas.
  - 3) Bicarakanlah deadlines dengan klien dengan menunjukkan gambaran dari rencana proyek sebagai bukti pendukung atas pernyataan Anda. Jika perencanaan proyek Anda cukup beralasan, maka kemungkinan negosiasi antara Anda dan klien akan memecahkan kebuntuan.
- e. Kurangnya komunikasi antara klien dengan tim dan manajer proyek. Jika Anda menjadi manajer proyek, masalah komunikasi ini juga bisa terjadi karena klien

tidak memahami informasi yang disampaikan oleh Tim Proyek oleh karena istilah-istilah yang tidak dimengerti oleh klien. Atasilah hal ini dengan cara:

- 1) Buat risalah untuk setiap rapat dan distribusikan kepada semua pihak yang terlibat dalam rapat atau pihak yang harus melakukan tindakan terhadap keputusan rapat.
  - 2) Gunakan frasa yang dapat disetujui oleh klien dan tim proyek dan konsisten di seluruh dokumen proyek.
- f. Tim proyek tidak memiliki pengetahuan politik tentang organisasi klien. Ada kepentingan, perselisihan, intrik, diskusi, dan kompromi yang tidak dapat dihindari dalam suatu kelompok atau perusahaan. Selain memiliki kompetensi operasional dan fungsional, seorang manajer perlu mengetahui adanya niat tersembunyi dari pihak lain. Anda harus dapat membentuk aliansi dengan mereka yang akan mendukung Anda dan tawar-menawar dengan atau mencapai kesepakatan dengan orang-orang yang mungkin menghalangi jalan Anda. Kemampuan ini sangat penting terutama untuk proyek besar, di mana banyak terdapat birokrasi, organisasi yang besar, sehingga informasi juga tersebar di berbagai tempat, dan penyusunan *requirements* terhambat oleh masalah *trust*, konflik internal, serta informasi yang tidak efisien (melalui banyak jalur). Untuk mengatasi masalah ini, beberapa hal yang dapat Anda lakukan adalah:
- 1) Lakukan review network Anda dan identifikasikan informasi apa yang Anda perlukan serta kepada siapa Anda bisa mendapatkannya.
  - 2) Bangun hubungan baik dan pahami secara sistematis mengenai modal sosial Anda dalam organisasi
  - 3) Dekati pihak yang bertentangan dengan Anda dalam organisasi klien, dengan meminta kesediaannya berbagi pengalaman untuk hal-hal yang berkaitan dengan bidang dan pengalamannya.

### **4.3 Project Scope Document**

*Project Scope Document* menyatakan ruang lingkup proyek, yaitu mengenai seberapa luas jangkauan pelaksanaan proyek dan sampai di mana batas-batasnya. *Scope* atau ruang lingkup tidak hanya mengenai batasan biaya ataupun sumber daya, tetapi juga mengenai apa yang merupakan bagian dari hasil (*deliverables*) pelaksanaan proyek dan apa yang tidak. Jadi dapat dikatakan bahwa PSD adalah pedoman utama Anda untuk mengawali proyek, sebelum proyek itu benar-benar dimulai.

PSD membantu Anda dan stakeholder lain, terutama klien, untuk memahami secara jelas apa yang diharapkan dari proyek, sehingga tidak terjadi ekspektasi yang berlebihan. Hal ini harus dilakukan sedari awal, sehingga saat mengajukan proposal, klien sudah memahami secara jelas apa yang akan menjadi deliverable dengan ruang lingkup yang telah disepakati.

Format PSD harus cukup detail agar Anda dan klien memiliki pemahaman yang sama. Intinya, yang disampaikan dalam PSD adalah apa yang Anda dengar dari klien, apa

rencana yang akan Anda lakukan, dan apa yang dibutuhkan (sumberdaya, dukungan, kebutuhan dan biaya) dalam pelaksanaannya (Tantra, 2012).

Secara ringkas, PSD terdiri atas susunan sebagai berikut (Dikrozian et al., 2018):

1. Maksud dan Tujuan Proyek

Jelaskan tujuan dan sasaran proyek, seperti memecahkan masalah saat ini atau merekomendasikan sistem informasi baru untuk meningkatkan kinerja organisasi.

2. Rencana Kerja

Uraian tentang rencana tindakan untuk mencapai tujuan dan sasaran yang telah ditetapkan. Rencana kerja dapat dijelaskan secara singkat karena proposal yang diajukan akan dirinci lebih lanjut nanti.

3. *Deliverables*

Hasil dari proyek berupa *deliverables* yang akan diberikan kepada klien pada akhir proyek. Definisikan *deliverables* dengan jelas dan rinci sehingga ekspektasi klien terhadap hasil proyek menjadi jelas. Untuk itu harus ada kejelasan terlebih dahulu mengenai kriteria sukses yang harus disepakati antara pihak pemilik proyek/klien dengan manajer proyek. Deliverables harus memenuhi kriteria sukses ini agar nantinya proyek dapat memasuki fase akhir dengan status yang menunjukkan keberhasilan.

4. Batasan-batasan

Penjelasan detail tentang apa saja yang dibutuhkan untuk masing-masing batasan tersebut, baik batasan waktu, biaya, maupun ruang lingkup.

5. Kesimpulan

Sampaikan kesimpulan akhir mengenai proyek ini yang kemudian menjadi dasar bagi penyusunan proposal untuk melaksanakan proyek. Contoh PSD dapat dilihat pada appendix 3.

#### 4.4 Penyusunan Tim

Membangun tim harus dilakukan sebelum pelaksanaan proyek dan merupakan salah satu tugas utama manajer proyek. Beberapa perusahaan bahkan menuntut agar nama, jabatan, dan kualifikasi anggota tim dicantumkan saat mempresentasikan proposal guna mendongkrak tingkat kepercayaan klien.

Tim yang mengerjakan proyek berbeda dengan tim yang bekerja di departemen perusahaan, yang bersifat permanen dan melakukan tugas normal, karena mereka:

1. Tim proyek dibentuk untuk waktu yang terbatas, yaitu sepanjang proyek berlangsung. Anggota tim memiliki keahlian dan pengalaman yang berbeda-beda, tidak seperti tim departemen yang sama secara fungsional.
2. Proses kerja belum ditentukan secara definitif sebelum proyek berlangsung.
3. Tim akan menemukan banyak hal yang belum diprediksi pada saat proyek berlangsung, sehingga banyak hal masih belum pasti.
4. Tekanan kerja atau tingkat stress lebih tinggi, yang disebabkan banyak hal yang belum pasti (Tantra, 2012).

Jika mereka berasal dari pihak klien dan dipilih dari tim departemen, anggota tim tertentu juga bekerja dalam kombinasi dengan tugas sehari-hari. Ini selalu meningkatkan beban kerja dan sering mengarah pada metode kerja yang tidak efisien. Hal ini masih sering dilakukan, meskipun proyek tersebut membutuhkan keahlian orang tersebut.

Mempertimbangkan hal tersebut di atas, masalah berikut perlu dipertimbangkan saat membuat tim:

1. Pastikan setiap anggota tim memiliki gambaran menyeluruh tentang proyek yang sedang mereka kerjakan.
2. Nyatakan dengan jelas kewajiban dan tugas masing-masing anggota tim. Penting juga untuk menggambarkan otoritas yang mereka miliki dalam aspek-aspek tertentu, seperti mengakses dokumen yang diperlukan atau melakukan wawancara.
3. Sebelum proyek dimulai, setiap anggota harus mengenal satu sama lain dan memahami peran mereka di dalamnya. Ini akan membantu mereka bekerja sama dengan baik dan mencegah mereka saling menyalahkan jika muncul masalah.
4. Dalam menentukan peran dan tanggung jawab anggota tim sebagaimana posisi struktur organisasi proyek, maka Anda harus juga memperhatikan keahlian individual yang dimiliki oleh setiap anggota apakah sesuai dengan posisi yang ditempatinya dan jangan sampai menjadi kontraproduktif.

Faktor yang paling penting adalah kontribusi yang diharapkan dari setiap anggota tim untuk mencapai tujuan proyek, oleh karena itu manajer proyek harus objektif ketika memilih setiap anggota tim dan tidak melakukannya karena belas kasihan atau emosi. Saat memilih anggota, kandidat setidaknya harus memiliki pengetahuan yang diperlukan untuk proyek tersebut, terbuka untuk komunikasi, bersedia berpartisipasi aktif, dan dapat bekerja dalam tim. Richard Sheridan juga mengingatkan bahwa anggota tim tidak hanya dari developer software saja, meskipun hasil akhir proyek adalah software, tetapi juga melibatkan stakeholder yang mewakili unsur pemilik dan pengguna hasil proyek.

Sangat penting bagi setiap anggota penjaga untuk berkomunikasi satu sama lain. Adakan rapat secara rutin, misalnya mingguan, di samping rapat lain sesuai kebutuhan. Rapat rutin akan memberi kesempatan bagi tim untuk:

1. Membuat anggota tim tetap fokus pada tujuan proyek.
2. Menunjukkan komitmen yang tetap pada tujuan proyek
3. Identifikasi jika terjadi masalah dalam proyek atau konflik antar anggota tim proyek.
3. Evaluasi hasil kerja masing-masing anggota atau kelompok dari tim proyek dan pengaruhnya pada anggota atau kelompok lainnya.
5. Memberikan kesempatan untuk menentukan tujuan jangka pendek (mingguan) dan follow up untuk yang sudah ditentukan di minggu sebelumnya (Tantra, 2012).

#### **4.5 Manajemen Risiko**

Bart Jutte memberikan 10 *golden rules* dalam manajemen risiko untuk proyek berdasarkan pengalamannya sehingga bisa diterapkan juga pada proyek sistem informasi pada umumnya (S. H. Wibowo et al., 2022).

1. Jadikan manajemen risiko bagian dari proyek.  
Seringkali manajemen risiko tidak diterapkan sebagai bagian dari proyek, sehingga penanganannya hanya insidental saja. Hal ini mungkin terjadi pada organisasi atau perusahaan yang baru pertama kali melaksanakan proyek sehingga tidak menyadari adanya risiko-risiko yang bisa menghambat proyek. Bisa juga terjadi karena terlalu besar kepercayaan yang diberikan pada manajer proyek sehingga terlalu yakin bahwa semua risiko pasti bisa diatasi. Organisasi atau perusahaan yang profesional biasanya menjadikan manajemen risiko sebagai bagian dari operasional sehari-hari dan memasukkannya dalam rapat proyek dan sebagai bagian dari *training staff*.
2. Identifikasikan risiko sejak awal proyek.  
Untuk mendeteksi risiko proyek sejak awal, bisa didapatkan dari dua sumber utama, yaitu *people* dan *paper*, orang dan dokumen. Orang yang berkepentingan terhadap proyek (*stakeholder*) dapat membantu untuk mengidentifikasi risiko, bisa dari pendapat mereka maupun pengalaman yang mereka miliki. Anda juga dapat menerima umpan balik tentang potensi risiko dari orang-orang di luar proyek, terutama dari orang-orang yang pernah mengerjakan proyek serupa di masa lalu. Jika diikuti dengan hati-hati, makalah proyek juga bisa menjadi alat untuk mengidentifikasi risiko proyek. Rencana proyek, kasus bisnis, dan rencana sumber daya adalah contoh dokumen proyek yang dapat menjadi sumber.
3. Komunikasikan risiko-risiko yang ada  
Risiko yang diidentifikasi harus dibagikan di antara anggota tim, khususnya kepada manajer proyek. Selain itu, manajer proyek harus selalu dapat mengomunikasikan risiko kepada klien dan anggota tim tertentu. Pada rapat tim rutin, manajer proyek dapat mendiskusikan dan mengulangi potensi risiko, memastikan bahwa tim selalu menyadarinya dan siap mengantisipasinya.
4. Pertimbangkan baik ancaman (*threats*) maupun kesempatan (*opportunities*)  
Risiko bisa menjadi ancaman bagi tim, terutama jika menghambat bahkan memboroskan sumberdaya, baik finansial, waktu maupun tenaga. Tetapi, dengan adanya risiko, kadang kala tim dapat menemukan cara baru dalam menangani tugasnya sehingga bisa menjadi lebih efisien.
5. Klarifikasi penanggung jawab untuk setiap risiko  
Memberi anggota tim tugas untuk mengantisipasi risiko akan memberi mereka kesempatan untuk lebih pandai dalam menjalankan tanggung jawab mereka, dan berdasarkan pengalaman, mereka biasanya dapat menyelesaikan setiap pekerjaan dengan risiko yang sudah ada. Tentu saja, Anda juga harus menetapkan tanggung jawab risiko sesuai dengan tanggung jawab yang dilakukan oleh masing-masing anggota tim proyek.
6. Buat prioritas risiko  
Sementara beberapa risiko hanya menimbulkan kesulitan kecil, yang lain memiliki potensi untuk mempengaruhi proyek secara signifikan. Untuk melakukan itu, Anda harus memutuskan mana dari risiko yang teridentifikasi yang harus ditangani terlebih dahulu. Tetapkan kriteria untuk setiap risiko, seperti yang dapat mengakibatkan

- kelebihan biaya atau penundaan jadwal. Terlepas dari kriteria yang digunakan, Anda harus selalu menjaga konsistensi dan fokus pada risiko yang paling berpengaruh.
7. Melakukan analisa risiko  
Manajer proyek akan lebih mampu mengenali risiko berkat analisis risiko dan cenderung tidak bertindak tergesa-gesa atau sebelum waktunya. Setiap risiko dapat diperiksa secara individual untuk memastikan potensi penyebab dan akibatnya. Menganalisis setiap risiko dan menugaskan akuntabilitas kepada tim adalah manfaat lainnya.
  8. Buat rencana dan implemetasi tanggapan terhadap risiko  
Buat rencana di depan ancaman yang teridentifikasi dengan baik. Tiga jenis antisipasi yang mungkin: menghindari risiko, mengurangi risiko, dan menerima risiko. Proyek harus dipandu oleh opsi pertama, penghindaran risiko, untuk menghindari risiko dengan cara tertentu. Dengan menginvestasikan lebih banyak sumber daya dan mencermati risiko yang dapat berdampak signifikan, Anda juga dapat mengurangi potensi risiko yang dapat muncul. Jika analisis risiko dilakukan dengan benar, Anda dapat merencanakan hasil potensial dan membatasi pengaruhnya. Merangkul risiko merencanakan potensi risiko dan bagaimana mereka dapat memengaruhi proyek tanpa berusaha menghindari atau menguranginya. Jika mengambil tindakan proaktif tidak lagi menjadi pilihan, biasanya inilah yang dilakukan.
  9. Dokumentasikan risiko proyek  
Berikan rincian tentang siapa yang bertanggung jawab untuk setiap risiko, tetapkan prioritas, dan berikan temuan analisis yang dilakukan pada setiap risiko. Dokumentasi ini bermanfaat sebagai titik awal untuk langkah-langkah yang diperlukan dan sebagai alat manajemen risiko untuk proyek yang akan datang..
  10. Tentukan risiko dan tindakan yang diambil  
Saat menentukan tindakan yang akan dilakukan untuk antisipasi terhadap risiko, Anda juga harus bersiap untuk menentukan tindakan bagi risiko berikutnya.

## **4.6 Proposal**

Proposal harus merinci segala sesuatunya sebagai hasil analisis terhadap *requirements*, sehingga klien mengerti dasar dari pengenaan biaya terhadap proyek yang dilaksanakan. Selain itu, yang harus dicantumkan juga adalah biaya-biaya yang menyangkut tambahan peralatan yang harus dibeli. Harga yang dicantumkan adalah sesuai dengan yang ada di pasaran saat itu. Untuk klien internal, biaya peralatan ini juga sebaiknya disertakan, meskipun untuk biaya pelaksanaan bisa tidak dimasukkan.

Proposal juga menjadi media untuk mempromosikan perusahaan atau organisasi, termasuk pengalaman dan kesuksesan dalam melaksanakan proyek sejenis.

### **4.6.1 Pengertian Proposal**

Proposal dapat diartikan sebagai bentuk usulan baik berupa gagasan maupun pemikiran yang dituangkan dalam bentuk rencana kerja tertulis kepada pihak lain dengan

tujuan untuk menjabarkan atau menjelaskan kepada mereka tentang rencana kerja tersebut (Sugiarto, 2015).

Untuk proposal yang berdasarkan permintaan resmi, klien biasanya membuka peluang kepada beberapa penyedia jasa/vendor, dengan mengeluarkan *Invitation For Bid* (IFB). Umumnya, hal ini dilakukan oleh perusahaan besar atau instansi pemerintah. IFB biasanya diumumkan melalui media atau melalui situs perusahaan/instansi yang ditujukan pada rekanan resmi. Sebelum mengumumkan IFB, klien biasanya juga telah memberikan gambaran mengenai *requirements*, yang masih bersifat global, sehingga vendor bisa memberikan proposal sesuai kebutuhan klien. Untuk mencegah terjadinya manipulasi, biasanya rekanan harus terdaftar dengan syarat-syarat yang telah ditentukan dan harus dipenuhi. Sebaliknya klien juga memiliki peraturan operasionalnya, yaitu proposal harus lebih dari satu dan harus tertutup, hanya bisa dibuka oleh pihak yang berwenang. Proposal yang mendapatkan proyek harus memenuhi kriteria yang telah ditentukan sebelumnya oleh klien.

Proposal yang berdasarkan permintaan tidak resmi umumnya dilakukan oleh perusahaan-perusahaan menengah, atau juga jika nilai proyek kecil sehingga tidak memerlukan aktivitas formal. Biasanya didahului oleh pihak penyedia jasa/vendor yang menawarkan jasa atau produk kepada klien prospektif. Klien biasanya tidak lagi mencari vendor alternatif, atau kalaupun ada hanya sebagai referensi saja. Klien juga memberikan gambaran mengenai *requirements*, tetapi tidak secara resmi, biasanya berdasarkan informasi yang didapatkan dari rapat-rapat pendahuluan dengan pihak-pihak yang berkepentingan, seperti *feasibility study*.

Proposal lainnya adalah yang tanpa permintaan dari klien. Biasanya ini dilakukan berdasarkan informasi yang didapatkan tentang peluang untuk mengajukan proposal suatu proyek yang mungkin diperlukan klien. Proposal ini tidak didahului oleh permintaan berdasarkan kebutuhan klien ataupun *requirements* yang jelas (Tantra, 2012).

#### **4.6.2 Struktur Proposal**

Struktur proposal pada umumnya adalah sebagai berikut:

1. Ringkasan eksekutif  
Berikan penjelasan ringkas mengenai apa yang ditawarkan dalam proposal dan solusi serta tujuan yang hendak dicapai.
2. Tinjauan dan rincian *requirements*  
Deskripsikan secara ringkas mengenai *requirements* yang menjadi dasar proyek yang hendak dilaksanakan. Setelah itu, berikan penjelasan rinci untuk setiap *requirements* yang telah disepakati sebelumnya.

3. Solusi yang diajukan  
Proyek yang dilaksanakan tentu akan memberikan solusi sebagaimana yang diharapkan. Jelaskan solusi yang ditawarkan untuk *requirements* yang telah dijelaskan sebelumnya.
4. Uraian pekerjaan  
Setelah menentukan solusi, deskripsikan pekerjaan-pekerjaan yang menjadi bagian dalam pelaksanaan proyek dalam setiap fase. Jelaskan juga batasan-batasan dan lingkup pekerjaan yang termasuk dalam proyek. Pemahaman terhadap batasan dan lingkup ini akan membuat pekerjaan tidak meluas ke hal-hal di luar proyek.
5. Rencana implementasi  
Deskripsikan bagaimana pekerjaan yang telah diuraikan sebelumnya akan diimplementasikan. Berikan penjelasan juga tentang tugas dan tanggung jawab masing-masing bagian dalam struktur organisasi tim proyek terkait dengan implementasi yang direncanakan. Bila personal dalam tim proyek telah ditentukan, dapat juga disebutkan dalam bagian proposal ini.
6. Investasi/biaya  
Pengajuan nilai investasi dan biaya yang diperlukan untuk melaksanakan proyek hingga selesai.

Contoh struktur yang lebih lengkap dapat dilihat pada appendix 4. Selanjutnya, proposal harus diajukan dan dipresentasikan kepada klien ataupun bagian dari stakeholder yang berkepentingan dengan proyek (Tantra, 2012).

#### 4.6.3 Presentasi dan Revisi

Agenda presentasi adalah menyampaikan *requirements* dan solusi yang diajukan sebagaimana yang dijelaskan dalam proposal. Dari presentasi kemungkinan akan didapatkan masukan-masukan baru yang sebelumnya tidak teridentifikasi. Perubahan-perubahan dari revisi harus dicantumkan pada bagian Versi Dokumen, yang memberikan informasi mengenai riwayat perubahan yang dilakukan pada proposal (Tantra, 2012).

Untuk meyakinkan pihak lain, seorang presenter harus menguasai materi pelajaran dengan baik. Untuk mengetahui bagaimana dan apa yang akan disajikan, presenter juga harus menempatkan diri pada posisi penonton. Agar presentasi dapat menarik perhatian audiens, presenter juga harus dapat menawarkan pengantar yang menarik. Selain itu, presentasi harus meyakinkan audiens bahwa informasi yang disajikan akan memberikan nilai bagi proyek. Untuk melakukan ini, presenter harus memiliki data yang solid dan persuasif baik dari segi teknis maupun praktis. Seorang presenter harus mampu menampilkan diri di depan audiens untuk memastikan presentasi yang sukses. Agar audiens memahami presentasi dengan baik dan sesuai dengan tujuannya, yaitu ide proyek dan teknologi yang dapat diterima oleh semua pihak, maka presentasi harus tersusun rapi, pidato tidak boleh tergesa-gesa, dan pengucapan harus jelas. Hasilnya, ide utama presentasi sebenarnya adalah: Menjual produk agar proposal dapat diterima (Sudarma, 2012).

#### 4.7 Kontrak/Surat Perintah Kerja (SPK)

Jika pihak di luar organisasi perusahaan bertanggung jawab untuk melaksanakan proyek, kontrak harus dibuat. Manajer proyek akan mendapatkan Surat Perintah Kerja yang menyatakan pelaksanaan proyek jika pelaksana proyek adalah tim yang terdiri dari karyawan perusahaan. Kontrak antara pemilik proyek dan pelaksana proyek yang berada di luar organisasi perusahaan akan dibahas lebih rinci di bagian ini. Kontrak harus memiliki banyak ketentuan utama, termasuk:

1. Deskripsi pihak-pihak yang berkepentingan terhadap proyek, yaitu pemilik proyek yang akan menerima hasil proyek dan pelaksana proyek yang akan mengerjakan pekerjaan yang telah ditentukan sesuai dengan ruang lingkup serta memenuhi *requirements* yang telah disepakati dengan hasil akhir yang memenuhi kriteria kualitas yang juga telah ditentukan.
2. Deskripsi penyampaian, seperti penjelasan tentang sistem yang akan dikembangkan atau produk akhir yang akan dihasilkan proyek, serta layanan tambahan seperti persiapan, konfigurasi, atau pelatihan pengguna.
3. Hak dan kewajiban masing-masing pihak yang dapat ditegakkan selama proyek berlangsung. Dalam kebanyakan kasus, konsekuensi pelanggaran hak dan kewajiban juga disebutkan.
4. Kesepakatan investasi atau biaya yang harus dibayarkan kepada pihak pelaksana proyek. Umumnya, pembayaran dibagi atas beberapa tahap sesuai dengan ketentuan yang disepakati kedua belah pihak. Pelaksana proyek menyelesaikan proyek sampai milestone tertentu dan membuat berita acara yang disetujui kedua belah pihak dan menjadi dasar untuk pembayaran tahap tersebut.
5. Jadwal pelaksanaan yang mengacu pada kewajiban pihak pelaksana proyek. Jadwal ini tidak perlu sampai detail tetapi cukup menampilkan titik-titik untuk masing-masing milestone saja.
6. Bagian penutup yang berisi nama-nama person in charge penanggung jawab untuk masing-masing pihak, misalnya manajer proyek, ketua steering committee, penanggung jawab implementasi dari pihak klien dan sekretaris tim. Jika ada garansi untuk deliverable, bisa juga dimasukkan di bagian ini (Tantra, 2012).

#### 4.8 *Project Charter*

Definisi proyek biasa disebut *Project Charter* atau *Terms of Reference* (TOR). Definisi ini bertujuan untuk mendokumentasikan visi, tujuan, lingkup, *deliverables*, organisasi proyek dan rencana implementasi. *Project charter* adalah dokumen yang secara formal menjelaskan adanya proyek dan memberi arahan tujuan dan manajemen proyek. Menjelaskan kerjasama antara perusahaan dengan kustomer atau Penyedia Jasa dengan Pengguna Jasa (Parlan et al., 2020). Untuk menyusun project charter, langkah-langkah yang perlu diikuti adalah sebagai berikut:

1. Visi, tujuan, lingkup dan *deliverables* proyek.  
Tetapkan visi proyek. Anda dapat memutuskan apa tujuan proyek, atau hasil yang ingin Anda capai saat mengimplementasikan proyek, berdasarkan visi ini. Setelah itu, tentukan lingkup proyek dan apa yang menjadi *deliverables* proyek.
2. Organisasi proyek.  
Tetapkan struktur organisasi proyek saat ini. Jelaskan seluruh tim klien, termasuk pemimpin, pemberi persetujuan, dan pembuat keputusan untuk setiap masalah. Berikan daftar kontak klien yang dapat dihubungi jika Anda memiliki pertanyaan tentang proyek tersebut. Berikutnya, tentukan stakeholder yang terlibat dalam proyek, deskripsikan personal untuk setiap bidang yang ditangani dalam proyek. Kemudian, sebutkan personal utama yang menjadi pemilik proyek, sponsor proyek, dewan proyek (jika ada) dan manajer proyek, beserta peran dan tanggung jawab masing-masing. Setelah nama masing-masing personal disebutkan, buatlah bagan organisasi dan garis yang menunjukkan wewenang dan pelaporan untuk setiap bagian dalam organisasi proyek.
3. Implementasi proyek.  
Pada tahap ini, jelaskan mengenai implementasi proyek yang terdiri dari rencana implementasi, *milestones*, *key dependencies* dan perencanaan sumberdaya. Rencana implementasi terdiri dari seluruh fase proyek, tahap-tahap dan aktivitas proyek. Dengan adanya penjelasan ini maka *stakeholder* akan lebih meyakini bahwa proyek telah dipersiapkan sepenuhnya. Selanjutnya deskripsikan perencanaan sumberdaya proyek, termasuk personal, keuangan, alat-alat dan bahan kerja.
4. Risiko dan masalah.  
Terakhir, seperti yang dibahas di bagian manajemen risiko, jelaskan risiko dan masalah apa saja yang mungkin muncul dan cara mengendalikannya.  
Untuk contoh format project charter, Anda dapat melihatnya pada appendix 5 (Tantra, 2012).

#### **4.9 Project Kick-Off**

Yaitu pertemuan untuk memberikan informasi tentang pelaksanaan proyek. Hal yang dilakukan saat *kick-off* adalah pengenalan tim dimana setiap tim memiliki perannya masing-masing, penjelasan ruang lingkup yang dikerjakan kepada tim, kemudian waktu yang tersedia dan tools dan teknologi yang akan digunakan dalam pembuatan *software* (Prasimsha & Putro, 2020). Berikut ini contoh agenda yang bisa menjadi pedoman untuk melakukan *kick-off meeting*.

Saat rapat, ciptakan suasana yang menyenangkan agar semua anggota dapat mengikutinya dengan santai tetapi tetap serius dan berpartisipasi penuh. Mulailah rapat dengan memperkenalkan diri Anda sebagai manajer proyek lalu jelaskan mengenai agenda *kick-off meeting* ini. Jelaskan mengenai proyek yang akan dilaksanakan, maksud dan tujuannya, serta hasil yang diharapkan. Perkenalkan juga anggota tim dan peran masing-masing secara singkat.

Gunakan dokumen *project charter* sebagai pedoman untuk menyampaikan rencana proyek kepada tim. Jelaskan rencana proyek dan batasan-batasan yang ada, terutama yang menyangkut tugas dan batasan waktu yang mengikat setiap tugas tersebut sesuai dengan yang telah disusun dalam *project charter*. Sampaikan juga kepada tim bahwa penjelasan detail satu per satu mengenai masing-masing tugas dan delegasinya kepada anggota tim dilakukan sesuai rencana proyek dan pembagian tugas secara rinci.

Sampaikan juga rencana komunikasi tim dengan menentukan:

1. Jadwal rapat status proyek mingguan.
2. Penggunaan alat bantu untuk manajemen proyek, komunikasi tim dan kolaborasi.
3. *Update* kepada *stakeholder* di luar tim proyek.
4. Sarana penyampaian informasi yang bersifat insidental, terutama yang berkaitan dengan risiko proyek ataupun penyimpangan yang teridentifikasi (Tantra, 2012).

## **BAB V**

### **FASE PERENCANAAN**

Tujuan dari perencanaan proyek adalah untuk memberikan manajer proyek kerangka kerja untuk membuat prediksi yang akurat tentang kebutuhan sumber daya, biaya, dan jadwal. Estimasi ini dibuat dalam waktu singkat pada awal pelaksanaan proyek dan sering direvisi seiring berjalannya waktu. Estimasi harus memperhitungkan semua potensi hasil positif dan negatif.

#### **5.1 Rencana Proyek**

Persiapan pertama adalah membuat rencana. Dalam rencana proyek ini kita akan menggali lebih dalam untuk aktivitas dalam fase pengembangan, pembagian task dan pengelompokannya dalam *Work Breakdown Structure* (WBS), estimasi jadwal dan durasi pelaksanaan serta pelaksanaan dan/ata penanggung jawab task tersebut.

##### **5.1.1 Work Breakdown Structure**

*Work Breakdown Structure* (WBS) adalah dekomposisi hirarki yang berorientasi pada hasil kerja yang akan dilaksanakan tim pengembang untuk mewujudkan tujuan proyek. WBS adalah landasan perencanaan, pelaksanaan, pengendalian, pemantauan, dan pelaporan proyek yang efektif. WBS dapat digunakan pada proyek yang kompleks agar lebih mudah untuk dikelola, diperkirakan, dan diawasi secara efektif (Fatia et al., 2022).

WBS berbentuk struktur pohon yang menunjukkan bagian dari pekerjaan yang diperlukan untuk mencapai tujuan. Dalam proyek, WBS biasanya dimulai dengan tujuan akhir yang hendak dicapai, lalu membaginya menjadi komponen-komponen yang dapat dikelola dalam hal ukuran, durasi dan penanggung jawab (contohnya, sistem, subsistem, komponen, task, subtask dan paket pekerjaan *works-package*) termasuk juga semua langkah yang diperlukan untuk mencapai tujuan.

Berikut adalah pedoman dasar untuk mengelola WBS:

#### 1. Aturan 100%

Aturan ini menegaskan bahwa dalam menyusun WBS harus memasukkan seluruh lingkup dan *deliverables* proyek yang telah didefinisikan sebelumnya. Total pekerjaan yang ada pada suatu level akan sama dengan pekerjaan yang ada di level atasnya. Dengan demikian jika seluruh pekerjaan dalam hierarki dijumlahkan maka hasilnya adalah 100% dari keseluruhan pekerjaan yang harus dilaksanakan dalam proyek, tidak boleh kurang ataupun lebih.

#### 2. Hubungan antar bagian yang eksklusif

Setiap bagian atau komponen dari WBS tidak boleh tumpang tindih dalam hal lingkup kerja yang merupakan bagiannya. Hal ini harus dihindari karena dapat menyebabkan duplikasi pekerjaan dan kerancuan dalam perhitungan durasi maupun sumberdaya. Untuk menghindarinya, setiap komponen dari hasil WBS perlu memiliki informasi *milestone*, *deliverables*, aktivitas, lingkup dan kadang-kadang ditambahkan juga

informasi tanggal, sumberdaya dan biaya, sehingga lebih jelas dan dapat dibedakan dengan komponen lainnya.

3. Rencanakan hasil yang ingin dicapai, bukan pelaksanaannya

Dengan berorientasi pada hasil atau *deliverables* maka WBS akan lebih terstruktur dalam memenuhi semua kriteria yang telah ditentukan dalam lingkup proyek maupun lingkup sampai level komponen terminal, sehingga akan memenuhi aturan 100%. Tetapi jika dalam menyusun WBS lebih menekankan pada perencanaan terhadap aktivitas yang dilakukan untuk setiap komponen, maka hal ini biasanya akan membawa pada definisi kerja yang tidak sesuai untuk komponen tersebut, bisa lebih ataupun kurang. Oleh karena itu, hindarilah perencanaan berdasarkan pelaksanaan *task* pada setiap komponen WBS, tetapi lebih ditekankan pada *deliverables*. *Deliverables* tidak selalu berarti hasil fisik yang tampak (*tangible*), yang dapat disampaikan kepada klien atau stakeholder lainnya. Dalam hal pelaksanaan proyek, *deliverables* ini bisa berupa pelimpahan hasil suatu *task* kepada *task* berikutnya. Maka agar *task* berikutnya dapat mulai dikerjakan tanpa tertunda dan sesuai kriteria, durasi dari *task* sebelumnya harus mendapatkan porsi waktu yang cukup pada saat dilakukan, dengan berorientasi pada *deliverable* ke *task* berikutnya.

Dalam kelompok *System Development*, yang pertama kali dilakukan adalah *System Analysis and Design (SAD)*. *Deliverable* dari SAD adalah dokumen teknis desain *software*. Desain ini tidak menjadi input bagi *task* yang langsung berurutan, *Internal Test Plan and Test Script*, tetapi untuk *System Construction* di mana sistem mulai dibangun. Setelah *System Construction* selesai, maka *deliverable* yang dihasilkan beserta *deliverable* dari *Internal Test Plan and Test Script* menjadi *input* bagi *task* berikutnya, yaitu *Internal Test*.

4. Tingkatan detail

Pertanyaan yang umum saat melaksanakan WBS adalah sampai berapa detail proses penguraian terhadap *task* harus dilakukan. Hal ini penting untuk diperhatikan agar hasil WBS tidak terlalu luas dan membuat pengalokasian sumber daya dan pembuatan jadwal kerja menjadi terlalu memberatkan. Pedoman berikut dapat digunakan untuk menentukan tingkat detail dalam WBS:

- a. Setiap komponen atau *task* dalam WBS tidak melebihi 80 jam kerja
- b. Setiap komponen atau *task* harus tidak melebihi satu periode pelaporan. Misalnya, jika periode pelaporan proyek adalah 1 bulan, maka tidak ada aktivitas yang boleh melebihi 1 bulan dalam pelaksanaannya.
- c. Batas kewajaran, yakni gunakan pertimbangan yang masuk akal dalam melakukan penguraian, agar tidak terlalu detail maupun terlalu luas.

Suatu komponen *task* harus memiliki kriteria sebagai berikut.

- a. Dapat diperkirakan secara realistis.
- b. Tidak dapat dijelaskan secara wajar lebih detail.
- c. Patuhi peraturan tersebut diatas.
- d. Menghasilkan *deliverable* yang terukur.
- e. Dapat disubkontrakkan atau oleh pihak ketiga jika perlu.

## 5. Komponen terminal

Setiap WBS dapat diuraikan sampai tingkatan terbawah yang tidak dapat dibagi lebih detail lagi. Komponen ini disebut komponen terminal. Komponen terminal adalah suatu *task* yang dapat diestimasi baik untuk kebutuhan sumberdaya, anggaran maupun durasi, terhubung satu sama lain sebagai saling ketergantungan dan dapat dijadwalkan. Komponen terminal ini harus memenuhi aturan tingkatan detail. Untuk proyek-proyek sederhana tertentu, komponen terminal dapat dicapai pada saat tingkat detail baru mencapai tingkat dua atau tiga, sehingga Anda tidak perlu terlalu kaku dalam mengikuti pedoman yang ditetapkan. WBS yang digunakan pada contoh di atas dapat mencapai level kedalaman hingga level lima untuk proyek skala menengah (Tantra, 2012).

### 5.1.2 Estimasi

Ada dua faktor yang mendorong perlunya estimasi yang akurat, yaitu:

1. Estimasi menentukan perencanaan dalam mencapai waktu penyelesaian proyek sesuai dengan kesepakatan sehingga pada akhirnya menentukan penilaian orang lain terhadap kehandalan dan kompetensi manajer proyek.
2. Dalam proyek yang dikerjakan bagi pihak ketiga, estimasi juga menentukan perhitungan terhadap biaya pelaksanaan proyek yang pada akhirnya menentukan nilai jual proyek, sehingga hal ini menentukan profitabilitas proyek.

Estimasi untuk setiap *task* didasarkan pada pengalaman masing-masing personel. Ini adalah metode *bottom up*, yang pendekatannya berbeda dengan metode sebelumnya, *top down*, di mana sebagai manajer proyek, Anda yang melakukan estimasi (tentunya setelah melakukan rapat dengan tim) dan membagi-bagikannya dalam WBS.

Estimasi juga bukan merupakan harga mati. Estimasi juga bukan biaya yang ditetapkan. Anda masih dapat membuat perubahan pada jadwal dan durasi yang telah Anda buat saat proyek berkembang. Hal ini masih bisa dimaklumi selama Anda tidak melakukan penyesuaian yang signifikan. Untuk setiap aktivitas yang Anda perkirakan, Anda juga harus menyertakan waktu toleransi.

Anda juga harus menyediakan waktu untuk masing-masing hal berikut:

1. *Task* yang bersifat mendesak dan memiliki prioritas yang lebih tinggi dari yang ada.
2. Keadaan darurat.
3. Rapat internal dan eksternal.
4. Hari libur, cuti maupun anggota tim yang sakit atau berhalangan.
5. Hubungan dengan klien, supplier maupun pihak lain.
6. Kerusakan peralatan kerja.
7. Alat kerja yang dibutuhkan terlambat dihantarkan oleh supplier.
8. Interupsi dari berbagai pihak.
9. Prioritas dan jadwal lain, misalnya jika berhubungan dengan regulasi atau birokrasi.
10. Penolakan dari *quality control*.
11. Kejadian tak terduga lainnya (Tantra, 2012).

## 5.2 Rencana Sumber Daya Manusia

Perlu diperhatikan bahwa dalam fase perencanaan ini sebaiknya tim sudah solid dan tidak ada penyusunan ulang lagi, terutama dalam hal organisasi proyek dan pelaksana proyek. Apalagi jika perencanaan sudah memasuki estimasi waktu, karena jika personal pelaksana suatu *task* diganti oleh orang lain yang tidak memiliki keahlian yang setingkat dengan yang digantikannya, maka durasi *task* yang sudah dialokasikan bisa saja tidak sesuai lagi.

Anda harus memilih anggota tim yang dapat diandalkan untuk proyek Anda yang memiliki kemampuan yang diperlukan dan, jika memungkinkan, pengalaman yang cukup. Namun, kadang-kadang Anda tidak bisa mendapatkan *the dream team*. Suatu tim proyek bisa terdiri atas berbagai personal yang memiliki keahlian dan pengalaman yang beragam.

Mengembangkan rencana sumber daya manusia memerlukan pertimbangan rencana organisasi proyek dan fungsi organisasi. Anda dapat memulai perekrutan setelah membuat struktur organisasi proyek sesuai dengan deskripsi pekerjaan yang akan ditangani oleh masing-masing anggota tim proyek. Sama seperti proyek, yang bersifat sementara, organisasi proyek pun hanya ada selama proyek berlangsung. Semua personal yang ada akan kembali ke posisi seperti sediakala pada saat proyek berakhir. Untuk itu perlu ada kesepahaman antara manajer proyek dengan anggota tim proyek yang juga bekerja di luar proyek dalam tugas sehari-hari, dan jika perlu juga dengan atasan dari personal yang bersangkutan, sehingga selama tugas-tugasnya di proyek, maka pekerjaan sehari-harinya dapat didelegasikan dahulu. Idealnya juga selama proyek berlangsung, maka setiap anggota tim menganggap manajer proyek adalah atasannya sehingga tidak ada keragu-raguan dalam menjalankan tugas-tugas yang telah ditetapkan. Dengan organisasi proyek juga membantu agar alokasi *task* dapat dilakukan dengan efisien, tidak tumpang tindih dan menghindari adanya konflik.

Hal penting yang perlu diperhatikan dalam menyusun struktur organisasi proyek adalah jangan sampai terjadi birokrasi yang terlalu panjang. Birokrasi seperti ini biasanya terjadi dalam proyek skala besar, di mana tim yang terlibat dapat mencapai ratusan orang, sehingga hierarki proyek menjadi terlalu dalam. Untuk mengatasi masalah seperti ini, biasanya proyek dipecahkan menjadi beberapa sub-proyek yang masing-masing memiliki struktur organisasi sendiri sesuai tujuan sub-proyek dan *deliverables* yang telah ditetapkan untuknya. Para manajer sub-proyek ini kemudian akan berinteraksi dengan manajer proyek utama yang mengoordinasikan *deliverable* dari setiap sub-proyek untuk mencapai tujuan akhir proyek utama. Dengan demikian, birokrasi yang terlalu rumit dapat dihindari.

Dua faktor penting yang harus diperhatikan dalam menyusun struktur organisasi ini adalah:

1. Spesialisasi

Dalam melakukan WBS, proyek yang besar akan terurai menjadi *task*, dan setiap *task* ini akan dilaksanakan oleh satu orang ataupun sebuah tim yang terspesialisasi. Spesialisasi ini selain dari sisi fungsionalitas *task* juga dari sisi pelaksananya.

Contohnya, *task System Analysis and Design* harus dilaksanakan oleh Analis Sistem, sedangkan *System Construction* harus dilaksanakan oleh para *programmer*. Lebih jauh lagi, spesialisasi ini juga perlu mem-pertimbangkan teknologi yang digunakan, seperti bahasa pemrograman atau infrastruktur jaringan.

## 2. Koordinasi

Meskipun melaksanakan masing-masing task yang terspesialisasi, semua anggota tim juga harus terkoordinasi dengan baik, sehingga keseluruhan proyek dapat berjalan sesuai rencana yang telah disusun. Tanpa koordinasi yang baik dari manajer proyek, sebagus apapun tim yang disusun tidak akan dapat mencapai tujuan akhir proyek, karena bisa terjadi setiap personal akan mengerjakan task yang menjadi tanggung jawabnya tanpa memperhatikan kerjasama dengan ataupun interaksinya dengan anggota tim lain dan *tasks* yang terkait dengannya.

Gagasan pengumpulan sumber daya tradisional berfungsi sebagai dasar kerangka kerja ini. Tim yang diatur berdasarkan sumber daya bersama cenderung memiliki anggota yang merasa lebih nyaman, tetapi mereka tidak selalu bersatu untuk menghasilkan ide yang lebih baik. Berbagai tim akan berbicara satu sama lain dan bekerja sama untuk memecahkan suatu masalah. Misalnya, tim pengguna dan tim aplikasi masing-masing akan mengerjakan prototipe.

Dalam beberapa lingkungan teknologi informasi, ada anggapan dari sebagian staf bahwa adanya pemisahan dengan bisnis dan pengguna sistem lebih bermanfaat bagi mereka. Hal ini karena kehadiran pengguna dapat menghambat kemajuan pekerjaan. Meskipun keyakinan ini ada benarnya, bekerja sama dengan pengguna dan lingkungan bisnis benar-benar meningkatkan kualitas solusi dan menurunkan kemungkinan ditolak oleh audiens target. Selain itu, akan lebih mudah bagi Anda untuk memberikan deskripsi untuk setiap entitas dalam struktur organisasi setelah struktur organisasi proyek dibuat. Tujuan dari deskripsi ini adalah untuk memperjelas peran dan tanggung jawab setiap anggota tim (*roles and responsibilities*).

**Tabel 5.1** Posisi dan Tanggung Jawab Tim Proyek

Sumber: [www.ePMbook.com](http://www.ePMbook.com)

| Posisi            | Penjelasan  |
|-------------------|---|
| Sponsor Proyek    | Personal yang melihat perlunya perubahan dan memiliki kewenangan untuk membuat sesuatu terjadi. Mungkin ada beberapa sponsor yang secara kolektif memiliki peran ini. Beberapa otoritas dan dukungan yang lebih tinggi dalam organisasi atau perusahaan juga harus diajak untuk bergabung ke dalam peran ini. |
| Sponsor Pendukung | Untuk berhasil dalam semua aspek dari proyek di semua bagian organisasi, mungkin perlu juga untuk menetapkan beberapa sponsor pendukung pada tingkatan yang berbeda dan dalam unit organisasi yang berbeda.   |
| Direktur Proyek   | Personal dengan wewenang eksekutif yang sebenarnya atas proyek tersebut. Direktur Proyek memiliki akuntabilitas dan tanggung jawab  |

|  |   |
|--|---|
|  | <p>penuh untuk keberhasilan proyek, dan memiliki kekuasaan untuk membuat semua keputusan, namun tetap berada pada pengawasan badan eksekutif.</p>   |
| Komite Eksekutif                                 | <p>Sebuah badan yang mewakili otoritas eksekutif keseluruhan organisasi. Komite ini bisa saja merupakan Dewan Direktur, atau bisa juga sub-komite yang didelegasikan Dewan Direktur.</p>  |
| Komite Pengarah atau Dewan Proyek                | <p>Komite yang bertugas untuk melakukan pengawasan rutin proyek. Secara kolektif mereka harus mewakili semua bidang partisipasi dalam proyek dan mereka harus memiliki wewenang untuk mengambil keputusan atas nama bidang tersebut. Anggota komite biasanya adalah Kepala Departemen, Wakil Presiden, atau Direktur, bersama dengan perwakilan eksternal. Direktur Proyek dan Manajer Proyek biasanya akan melapor kepada Komite Pengarah.</p> |
| Manajer Proyek                                   | <p>Personal yang bertanggungjawab sehari-hari atas pelaksanaan dan keberhasilan proyek. Manajer Proyek biasanya memiliki kontrol atas semua sumber daya proyek</p>  |
| Manajer dan Staf PMO (project management office) | <p>“Kantor Proyek” menyediakan layanan dukungan bersama untuk manajer proyek dan tim proyek secara keseluruhan. Seringkali fungsi ini memiliki seorang manajer ditambah staf pendukung. Tanggung jawab umum termasuk pengendalian dan penelusuran rencana detail, mengelola dokumentasi, pembuatan laporan, dan lain-lain, juga dapat menjadi lokasi untuk staff spesialis pendukung tim, misalnya, desainer pelatihan.</p>                     |
| Akuntan Proyek                                   | <p>Sebuah proyek besar mungkin memerlukan akuntan sendiri untuk berurusan dengan pengadaan, sub- kontraktor pengeluaran, akuntansi joint venture, pencatatan kemajuan dan pelaporan keuangan.</p>   |
| Pemimpin Tim                                     | <p>Biasanya proyek ini akan dibagi menjadi berbagai sub-tim, masing-masing dengan pemimpin tim sendiri. Pemimpin tim akan bertanggung jawab untuk pengelolaan dan pembinaan dari sub-tim. Mereka juga memiliki tanggung jawab untuk mengelola dan mencatat secara detail sub-rencana untuk tim mereka.</p>  |
| Manajer Fasilitator Perubahan Organisasi         | <p>Seorang spesialis dalam mengidentifikasi isu-isu, persyaratan dan solusi tentang perubahan organisasi yaitu faktor-faktor rasional, politis dan emosional perusahaan atau individu dalam membawa perubahan bisnis yang diinginkan.</p>   |
| Spesialis Komunikasi                             | <p>Seorang spesialis untuk mengomunikasikan pesan dalam organisasi. Biasanya akan ada berbagai media komunikasi yang harus dieksploitasi proyek dalam mencapai tujuannya.</p>   |
| Spesialis Rekayasa-ulang proses bisnis           | <p>Seorang spesialis dalam proses dan teknik rekayasa ulang proses bisnis untuk mendapatkan kinerja yang optimal.</p>   |

|                                       |  |
|---------------------------------------|--|
| Pemilik Proses                        | Seseorang dalam organisasi yang memiliki kontrol secara keseluruhan, otoritas, dan akuntabilitas untuk setiap proses bisnis yang diberikan.  |
| Spesialis Proses                      | Seorang ahli dalam solusi best practice untuk proses bisnis tertentu.  |
| Manajer Proses                        | Seorang manajer dalam organisasi dengan pemahaman dan pengalaman detail tentang bagaimana suatu proses beroperasi.   |
| Pemodel Proses                        | Seorang spesialis dalam pemodelan proses bisnis sehingga potensi perbaikan dapat didefinisikan dan diukur  |
| Arsitek Solusi                        | Seorang spesialis dalam mendefinisikan solusi bisnis secara keseluruhan dengan tanggung jawab untuk “gambaran besar” sistem informasi yang akan dibangun.  |
| Arsitek Teknis                        | Seorang spesialis dalam mendefinisikan komponen- komponen teknis dari solusi bisnis dengan tanggung jawab untuk arsitektur teknis solusi.  |
| Spesialis Desain Organisasi           | Seorang spesialis dalam penilaian kebutuhan dan tingkat kemampuan sumber daya, ditambah dengan desain dan pencapaian sumber daya yang telah direvisi dan struktur organisasi.  |
| Desainer Solusi                       | Seorang spesialis dalam melakukan desain detail komponen solusi. Pada kenyataannya, ada berbagai jenis spesialis dalam kategori ini, sebagai kebutuhan masing- masing personal untuk menjadi spesialis dalam aspek dari solusi yang mereka desain, misalnya database desainer, desainer website, desainer program, desainer paket konfigurasi, desainer jaringan, desainer prosedur, dan sebagainya. |
| Developer/Programmer                  | Seorang spesialis dalam pembuatan komponen solusi. Sekali lagi, ada berbagai jenis programmer tergantung pada apa yang sedang diprogram.   |
| Spesialis Jaringan dan Telekomunikasi | Seorang spesialis dalam desain dan konstruksi jaringan dan telekomunikasi. Mereka akan berurusan dengan masalah jaringan internal dan eksternal, seperti arsitektur, hardware, kapasitas bandwidth,  |
| Spesialis Pemasaran                   | Seorang spesialis dalam pemasaran. Meskipun solusi memiliki unsur eksternal, tetapi penting untuk mempertimbangkan bagaimana membuatnya menarik bagi orang- orang eksternal dan badan-badan yang bersangkutan. Khususnya dengan eSolutions, pertimbangan tersebut akan membentuk bagian penting dari desain solusi, bukan hanya sekadar sebuah aktivitas setelah solusi selesai.                     |
| Spesialis Pelatihan                   | Seorang spesialis dalam mengidentifikasi kebutuhan pelatihan, kemudian mendesain pendekatan pelatihan dan konten untuk memenuhi kebutuhan tersebut.  |
| Developer Pelatihan                   | Seorang spesialis dalam pengembangan materi pelatihan. Seringkali pendekatan yang paling efisien untuk penyampaian pelatihan memerlukan beberapa atau semua konten dibuat dan dikirimkan secara elektronik.  |

|                                |  |
|--------------------------------|--|
| Trainer                        | Seseorang dengan keterampilan dan pengetahuan yang diperlukan untuk memberikan pelatihan.  |
| Penulis Teknis/Dokumenter      | Seorang spesialis dalam pembuatan dokumentasi yang akurat-baik untuk penggunaan sehari-hari dari solusi dan sebagai dokumentasi desain, untuk referensi di masa mendatang. Dokumentasi dalam solusi yang modern biasanya akan didukung secara elektronik, misalnya menggunakan perangkat lunak alur kerja dan informasi bantuan secara konteks-sensitif. |
| Pengguna Akhir/End User        | Pengguna akhir merupakan sumber daya berharga dalam tim-mereka dapat didayagunakan untuk berbagai keperluan yang terkait dengan konstruksi, desain dan pengiriman dari solusi bisnis. Sebagai mantan anggota tim, mereka akan kembali ke departemen mereka sebagai ahli dan pelatih untuk solusi baru.   |
| Analisis/Staf Operasi Komputer | Seorang spesialis untuk cara penggunaan solusi yang sudah berjalan. Prosedur operasi akan mencakup operasi rutin, kontrol, keamanan, backup/recovery, rencana bencana, dll.  |
| Manajer/Staf Fasilitas         | Seorang yang bertanggung jawab untuk penyediaan akomodasi yang sesuai bagi organisasi yang telah diperbaharui untuk melakukan proses bar. Ini bisa saja hanya beberapa penyesuaian fasilitas yang ada atau mungkin sejumlah akuisisi dan pembangunan fasilitas baru secara keseluruhan.  |
| Pengacara/Penasehat Hukum      | Seorang spesialis hukum yang akan menangani masalah-masalah kontrak seperti berbagai detail kontrak untuk penyediaan peralatan atau jasa.  |
| Auditor Eksternal              | Akuntan eksternal yang bertanggung jawab untuk audit organisasi. Mereka mungkin perlu untuk meninjau rencana, desain dan solusi lengkap untuk memastikan memenuhi standar yang memadai dari perspektif audit.  |
| Auditor Internal               | Seorang karyawan dari organisasi atau perusahaan yang diberikan tanggung jawab untuk mempertahankan standar dan prosedur dalam organisasi atau perusahaan.   |
| Regulator Eksternal            | Banyak industri dan organisasi yang harus mematuhi berbagai bentuk peraturan oleh regulator eksternal. Bisa juga ada kebutuhan untuk bekerja sama dengan regulator ini, atau untuk mengurus catatan tertentu atau informasi untuk memenuhi apa yang dibutuhkan mereka.   |
| Manajer Kualitas               | Seseorang yang bertanggung jawab untuk proses dan prosedur yang menjamin tercapainya tingkat kualitas yang diharapkan.   |
| Auditor Kualitas               | Seseorang bertanggung jawab atas Audit Kualitas.   |

### 5.3 Rencana Keuangan

Perhitungan anggaran untuk proyek sistem informasi biasanya dilakukan dengan estimasi yang didasarkan pada jangka waktu pelaksanaan. Jika diperlukan investasi tambahan dalam bentuk *hardware* dan infrastruktur, maka anggaran juga harus mencakupnya. Setelah proyek ditetapkan dalam bentuk kontrak, maka anggaran yang juga sudah disetujui ini harus direncanakan penggunaannya secara cermat. Jika proyek ini bersifat komersial maka harus dialokasikan pula sejumlah nilai sebagai margin keuntungan perusahaan pelaksana proyek.

Pengembangan alokasi biaya untuk proyek membutuhkan ketelitian yang tinggi. Memastikan tidak ada pemborosan di awal proyek tetapi ada penghematan di akhir adalah penting saat membagi alokasi biaya untuk setiap aktivitas yang harus diselesaikan. Salah satu pendekatan paling sederhana didasarkan pada WBS yang sudah Anda buat. Setiap pekerjaan yang dikompilasi dalam WBS memiliki perkiraan waktu eksekusi. Lamanya waktu bervariasi sesuai dengan unsur-unsur tugas dan mungkin juga dipengaruhi oleh pelaksana. Durasi pelaksanaan mungkin lebih pendek jika pelaksana memiliki pengetahuan dan pengalaman yang diperlukan, tetapi jika pelaksana tidak memiliki keterampilan yang diperlukan untuk menyelesaikan tugas, waktu pelaksanaan dapat lebih lama dari waktu yang diberikan dalam perkiraan. Dengan membuat rencana sumber daya manusia yang tepat, Anda dapat mengatasi hal ini.

Setelah Anda memiliki estimasi waktu pelaksanaan, Anda dapat memberi bobot nilai per satuan waktu untuk setiap task. Bobot nilai ini bisa bervariasi, tergantung dari karakteristik *task* maupun keahlian yang dibutuhkan. Task untuk pemrograman mungkin memiliki nilai lebih tinggi daripada task dokumentasi. Demikian juga mungkin nilai programmer lebih tinggi daripada database administrator. Anda dapat menentukan bobot untuk semua task yang ada dalam WBS, lalu dikalikan dengan durasi pelaksanaannya. Maka, nilai total keseluruhan akan menjadi total biaya proyek. Selanjutnya, Anda tinggal menyusun rencana keuangan proyek.

Anda harus terlebih dahulu melakukan perbandingan antara nilai proyek dan biaya keseluruhannya. Jangan sampai melebihi anggaran atau tidak seimbang. Anda kemudian harus mengubah rencana WBS dan sumber daya manusia jika hal itu terjadi. Faktor pengganda biaya, atau satuan waktu, dapat diturunkan dengan melakukan penyesuaian dengan penggunaan alokasi waktu yang lebih efektif. Selain itu, Anda mungkin dapat mengubah personel yang Anda pekerjakan.

Yang kedua, harus dipastikan kapan waktu menerima dan mengeluarkan dana proyek. Jika personal mendapat honor khusus dalam pelaksanaan proyek di luar dari gaji tetap yang telah mereka peroleh dalam pekerjaan sehari-hari, maka tentunya harus dipastikan periode pembayaran honor ini. Selain itu, mungkin juga ada biaya-biaya lain seperti transportasi dari dan ke lokasi proyek, sewa tempat, pembelian hardware dan peralatan penunjang. Anda juga mungkin perlu mengalokasikan dana untuk konsumsi saat rapat proyek, atau jika tim harus lembur untuk mengejar jadwal. Selain menyusun anggaran proyek sebagai bagian dari rencana keuangan, Anda juga harus mempersiapkan

staf administrasi proyek untuk melakukan update laporan keuangan proyek nantinya saat proyek sudah berlangsung.

Tahap yang terakhir adalah meminta pengesahan dari pihak steering committee, agar nantinya dapat dipertanggungjawabkan setelah proyek selesai, bahwa semua pengeluaran sudah mengikuti anggaran yang telah disahkan. Namun, Anda juga perlu menyebutkan bahwa jika keadaan pada saat pelaksanaan berubah, maka rencana keuangan dapat berubah. Karena hampir semua rencana berpotensi untuk diubah, hal ini penting untuk disampaikan agar tidak terjadi kesalahpahaman dengan pihak lain (Tantra, 2012).

#### **5.4 Rencana Kualitas**

Untuk menyusun rencana kualitas, lakukanlah langkah-langkah berikut ini:

1. Definisikan kualitas yang diharapkan dari deliverables proyek  
Berikan batasan-batasan kualitas deliverables proyek sesuai *requirements* yang telah ditetapkan dalam lingkup proyek. Batasan ini harus berupa kalimat positif dan jika perlu lebih ditegaskan agar menjadi acuan dalam menghasilkan kualitas deliverables terbaik dari setiap tahapan proyek.
2. Tentukan kriteria pemenuhan kualitas dan standar pencapaiannya  
Definisi kualitas yang telah disebutkan harus memenuhi kriteria sesuai *requirements*. Untuk memenuhi kriteria ini tentu diperlukan standar yang menjadi tolok ukur dalam setiap proses yang dilakukan agar kriteria kualitas terpenuhi.
3. Definisikan proses yang dibutuhkan untuk mencapai tujuan kualitas  
Definisikan bagaimana proses pelaksanaan yang sesuai standar untuk memenuhi kriteria tersebut. Dapat dibuat dalam bentuk garis besar sebagai pedoman pelaksanaan proyek, dan tidak perlu sampai secara detail untuk setiap task.
4. Deskripsikan quality assurance dan cara melakukan kontrol kualitas  
Saat proses dilaksanakan, kontrol perlu dilakukan agar tetap sesuai standar kualitas, dan lakukan quality assurance terhadap deliverables apakah memenuhi kriteria kualitas. Jika belum memenuhi kriteria maka harus ada tindakan yang diambil, misalnya rework ataupun evaluasi terhadap kriteria dan standar sehingga bisa diketahui apakah kriteria dan standar sesuai dengan kinerja tim proyek (Tantra, 2012).

#### **5.5 Rencana Risiko**

Tujuan menciptakan strategi risiko adalah untuk memutuskan tindakan proaktif untuk risiko yang diidentifikasi sebelumnya. Rencana risiko ini terdiri dari:

1. Daftar semua risiko yang telah teridentifikasi pada fase inisialisasi.
2. Urutan risiko berdasarkan kemungkinan terjadinya.
3. Urutan risiko ditentukan oleh konsekuensi potensial dari masing-masing risiko.
4. Risiko diurutkan berdasarkan kepentingan penanganannya.

5. Tindakan-tindakan yang harus dipersiapkan untuk mengurangi potensi terjadinya risiko.
6. Tindakan-tindakan yang harus dipersiapkan apabila terjadi risiko untuk mengurangi dampak yang merugikan.
7. Proses untuk menangani risiko selama proyek berlangsung (Tantra, 2012).

### **5.6 Rencana Penerimaan**

Penerimaan dapat didefinisikan sebagai persetujuan dari klien bahwa deliverables proyek sudah sesuai dengan kriteria kualitas yang disepakati dan berdasarkan batasan-batasan proyek, yaitu waktu penyelesaian, lingkup proyek dan biaya yang digunakan.

Strategi penerimaan sangat penting untuk keberhasilan penutupan proyek dan kesimpulan yang direncanakan. Dengan demikian, rencana penerimaan merupakan komponen dari fase terakhir proyek. Berikut ini termasuk dalam rencana proyek:

1. Daftar dari milestones yang harus dicapai dan deliverables yang harus dihasilkan dari keseluruhan proyek.
2. Semua kriteria dan standar penerimaan terhadap deliverables yang ditentukan oleh klien.
3. Penjelasan mengenai bagaimana deliverables akan ditinjau (review) untuk kemudian diputuskan apakah sudah sesuai dengan kriteria dan memenuhi standar yang telah ditentukan sebelumnya.
4. Proses untuk mendapatkan penerimaan dari klien terhadap deliverables yang telah dihasilkan (Tantra, 2012).

### **5.7 Rencana Komunikasi**

Rencana komunikasi berisi berbagai informasi yang harus disampaikan kepada seluruh stakeholder proyek agar mereka tetap mendapatkan status dari proyek yang sedang dilaksanakan (Khadafi et al., 2020). Informasi yang mudah diakses dapat membantu mendeteksi potensi risiko dan mencegah kesalahpahaman di antara para pihak karena setiap orang akan merasa terlibat dalam proyek tersebut.

Rencana komunikasi terdiri atas:

1. Jenis informasi yang dibutuhkan oleh setiap bagian stakeholder proyek.
2. Jadwal distribusi informasi, metode komunikasi yang digunakan dan bagaimana atau dalam bentuk apa informasi disampaikan.
3. Daftar personel yang dilibatkan dalam penyediaan informasi dan menyampaikan komunikasi tersebut.
4. Proses yang mengatur komunikasi, yaitu bagaimana informasi dikumpulkan, dikelola dan didistribusikan (Tantra, 2012).

## **5.8 Rencana Pengadaan**

Rencana pengadaan memberikan rincian deskripsi tentang produk (barang dan jasa) yang diperoleh dari penyalur, pembenaran atas mendapatkan setiap produk secara eksternal dan bukan dari dalam bisnis, dan jadwal untuk pengiriman produk.. Ini juga menguraikan proses pemilihan penyalur yang diinginkan (proses tender) dan pemesanan dan pengiriman produk (proses pengadaan) (Sitanggang et al., 2019).

Rencana pengadaan berisi antara lain:

1. Informasi produk yang akan dibeli dari vendor, termasuk spesifikasi teknis secara menyeluruh. Berikan detail kebutuhan barang, jika relevan.
2. Jelaskan proses pengadaan, seperti apakah produk dibeli dengan layanan instalasi atau apakah layanan instalasi merupakan kontrak terpisah yang harus dilakukan oleh vendor lain. Cantumkan informasi apakah produk yang akan dibeli masih tersedia atau sudah discontinue.
3. Jadwal pengadaan produk disesuaikan dengan jadwal proyek sehingga tidak terjadi penundaan karena ketidaktersediaan produk tersebut pada waktunya.
4. Perlu tidaknya proses tender untuk pengadaan ini dan apakah proses tersebut mengikuti prosedur yang ada di organisasi atau perusahaan klien, atau mengikuti prosedur yang berlaku untuk proyek.
5. Adanya vendor tertentu yang direkomendasikan, serta informasi mengenai latar belakang yang mendukung rekomendasi tersebut. seperti ketepatan waktu, layanan, dan dukungan, atau bahkan reputasi..

Bentuk dokumen perencanaan proyek dapat dilihat pada Appendix 6.

## **5.9 Tinjauan Perencanaan Proyek**

Karena situasi dan kondisi sepanjang pelaksanaan proyek selalu berubah, perencanaan masih perlu direvisi. Beberapa potensi bahaya yang dapat mengakibatkan kegagalan proyek dapat terjadi jika Anda terus mendasarkan proyek Anda pada perencanaan saat ini ketika lingkungan tidak lagi menguntungkan. Untuk melakukannya, Anda harus fleksibel dalam perencanaan proyek agar dapat menyesuaikan dengan lingkungan implementasi.

Dalam proses pelaksanaan, setiap task yang telah didefinisikan dengan WBS akan dikerjakan satu per satu. Hasil kerja ini akan melalui proses kontrol agar setiap deliverable memenuhi rencana yang telah disusun, terutama yang menyangkut rencana kualitas dan risiko, serta juga memperhatikan kontrol pada anggaran dan waktu. Bila terjadi hal-hal yang memerlukan penyesuaian, maka Anda dan tim pelaksana harus kembali ke meja perencanaan untuk melakukan perubahan-perubahan yang diperlukan. Seluruh aktivitas ini akan terus berulang selama proyek berlangsung, sehingga digambarkan sebagai lingkaran yang saling terhubung. Pengulangan ini akan berhenti apabila telah dicapai titik akhir pekerjaan yang memenuhi kriteria dan standar (Tantra, 2012).

## **BAB VI**

### **FASE PENGEMBANGAN**

Pada fase pengembangan, semua persiapan untuk fase penyelesaian dan desain diterjemahkan ke dalam hasil yang dapat dilihat dan digunakan klien dengan segera. Pelaksanaan pekerjaan pengembangan ini pada hakekatnya adalah pembangunan sistem informasi dengan deliverables berupa software dan komponen pendukungnya, seperti dokumentasi sistem, SOP (System Operating Procedure), dokumentasi pengembangan sampai dengan user manual dan materi pelatihan penggunaan software. Adapun tahapan pengembangan software *Waterfall Model* adalah sebagai berikut.

#### **6.1 Requirements**

Langkah ini melibatkan konsultasi pengguna sistem untuk menentukan fitur, batasan, dan tujuan sistem. Ini semua akan dijelaskan secara khusus dan digunakan sebagai spesifikasi sistem (Basriyanto, 2022). Dokumen *System Requirements Specification* (SRS) dan *Project Scope Document* (PSD) merupakan hal penting sebagai penunjang langkah berikut, yaitu desain sistem dan software.

#### **6.2 Desain Sistem Dan Software**

Desain pada sistem dan software melibatkan pengaturan struktur komponen *software* yang akan digunakan dalam pengembangan software itu sendiri. Jika sistem adalah kumpulan dari *deliverables* manajemen proyek sistem informasi, maka *software* sendiri adalah bagian dari sistem tersebut. Proses desain dan pengembangan keduanya saling terkait karena software merupakan bagian utama dari sistem informasi. Proses desain sistem dan software melibatkan fase-fase berikut (Tantra, 2012):

##### **6.2.1 Spesifikasi Fungsional Dan Teknis**

Spesifikasi fungsional software adalah dokumen yang akan menjelaskan apa yang harus dilakukan oleh software dan bagaimana software harus melakukannya. Ini menentukan karakteristik, fungsi dan komponen sistem, batasan dan keterbatasan software. Spesifikasi fungsional software akan menentukan bagaimana software merespons input dan mengekstraksi output yang akan dihasilkannya (Rachmad et al., 2023).

Secara teknis, spesifikasi sistem dapat dilihat dari dokumen SRS yang berisi spesifikasi fungsional dan nonfungsional, tetapi tidak menjelaskan solusi yang diusulkan untuk memenuhi spesifikasi tersebut. Solusi yang diusulkan akan dirancang dalam hal desain sistem dan desain software. Selain itu, spesifikasi teknis juga dapat dideskripsikan melalui *Functional Specification Document* (FSD). Dokumen ini menggambarkan bagaimana *software* akan berfungsi dan fungsi-fungsi apa yang harus dimilikinya agar dapat memenuhi *requirements*.

Selain fungsional juga terdapat spesifikasi teknis yang dituangkan dalam dokumen *Software Technical Specification* (STS). Berbeda dengan FSD, STS lebih kepada deskripsi tentang aspek teknis dari software yang hendak didesain, seperti platform yang digunakan, bahasa pemrograman, database dan struktur data, koneksi *software* dengan database, arsitektur *software*, topologi jaringan dan sebagainya. Dalam dokumen STS ini dapat digunakan bahasa teknis karena memang dokumen ini lebih ditujukan bagi desainer sistem dan developer (pemrogram) sehingga penyampaiannya akan lebih mudah dipahami. STS dapat dilengkapi dengan skema dan diagram agar lebih memperjelas deskripsi yang diberikan (Tantra, 2012).

### 6.2.2 Risiko Dan Mitigasi

Manajemen risiko harus selalu dijadikan bagian dari proyek, dan potensi risiko ada pada setiap fase. Risiko kegagalan proyek akan semakin tinggi jika proses desain baik sistem maupun *software* tidak dilakukan secara tepat. Desain yang tidak dilakukan dengan baik akan berpotensi terhadap kegagalan *deliverables* yang merupakan hasil akhir dari proyek itu sendiri.

Selain menemukan potensi risiko pada tahap ini, penting untuk menyiapkan rencana untuk mengatasi risiko tersebut. Lebih baik jika rencana mitigasi disertakan pada tahap pengembangan dan menjadi bagian dari desain. Hal ini bertujuan agar jika terjadi masalah pada tahap pengembangan, rencana mitigasi yang sesuai dengan desain sudah tersedia, sehingga proyek dapat terus berjalan atau setidaknya tidak terhambat terlalu lama. Seorang pengembang yang sudah menulis ribuan baris kode program akan sangat terganggu atau merasa tertekan jika harus mengulang pekerjaannya hanya karena kesalahan pada tahap desain. Tidak semua risiko dapat diidentifikasi pada saat mempersiapkan desain (Tantra, 2012).

### 6.2.3 Desain Sistem

Dalam konteks manajemen proyek, tahap desain sistem adalah bagian dari fase pengembangan yang menjadi dasar bagi konstruksi sistem yang akan dibangun. Meskipun pembahasan tentang desain sistem memiliki aspek yang lebih mendalam, dalam bab ini fokus diberikan pada keterkaitannya dengan manajemen proyek.

Desain sistem bertujuan untuk membuat sebuah sistem yang memenuhi spesifikasi kebutuhan fungsional, mencapai target, serta memenuhi kebutuhan secara implisit atau eksplisit dalam hal performansi dan penggunaan sumber daya. Selain itu, desain sistem juga memperhatikan batasan-batasan pada proses desain dari segi biaya, waktu, dan perangkat yang tersedia. Hal ini bertujuan untuk memberikan kepuasan meskipun bersifat informal, terhadap hasil akhir sistem yang dibangun. (Yulianto et al., 2009).

Ada beberapa metode untuk melakukan desain sistem. Metode yang cukup populer tahun 1970-an sampai dengan 1990-an adalah desain sistem terstruktur yang dirintis oleh Edward Yourdon, yang lebih dikenal sebagai *Yourdon System Method* (YSM) yang diklaim sebagai metode generasi ketiga. Metode ini menggunakan *context diagram* yang

menggambarkan hasil analisis dalam bentuk sumber data, aliran dan batasan-batasan sistem. Kemudian setelah melalui serangkaian aktivitas analisis, maka desain sistem lebih diperinci dengan teknik-teknik tradisional, seperti *Entity Relationship Diagram* (ERD), normalisasi, *Data Flow Diagram* (DFD), *State Transition Diagram* (STD), *Structured English*, dan sebagainya. Berikutnya adalah metode analisis dan desain sistem yang berorientasi obyek (*Object Oriented System Analysis and Design Method*). Alat bantu yang banyak digunakan untuk metode ini adalah *Unified Modelling Language* (UML) (Tantra, 2012).

#### **6.2.4 Pemodelan (Modelling)**

Model merupakan representasi abstrak dari sistem yang akan dibangun, yang ditampilkan melalui deskripsi proses dari beberapa perspektif tertentu. Proses pemodelan software adalah serangkaian aktivitas yang terkait dan koheren, yang bertujuan untuk menentukan, merancang, mengimplementasikan, dan menguji sistem software (Zufria, 2013).

Untuk mengembangkan sistem yang semakin kompleks maka pemodelan mutlak diperlukan untuk mempermudah para *developer* memahami apa yang mereka bangun. Dengan melakukan pemodelan maka *developer* dapat:

1. Untuk memahami sistem yang akan dibangun, perlu dilakukan definisi proses dalam bentuk model.
2. Pemahaman akan cara kerja suatu proses dan asumsi yang mendasarinya sangat penting dalam pengembangan sistem.
3. Pemodelan memudahkan dalam menentukan sumber input sistem, baik dari luar (melalui input data) maupun dari dalam sistem (dari modul-modul yang saling terhubung).
4. Pemodelan membantu menentukan keterhubungan antara komponen-komponen sistem (modul-modul) sehingga memudahkan pemahaman terhadap sistem secara keseluruhan.
5. Pemodelan membantu dalam menelusuri kembali desain hingga pada tahap persyaratan (requirements) untuk memastikan bahwa sistem yang dibangun sesuai dengan kebutuhan (Tantra, 2012).

#### **6.2.5 Desain Software**

Secara umum, desain software merupakan tahap yang lebih detail dari desain sistem. Tahap ini melibatkan perincian sistem yang telah direpresentasikan dalam bentuk model agar lebih mudah dipahami oleh para *developer*. Dalam tahap desain software, dilakukan penyusunan representasi software yang akan digunakan, baik dalam bentuk arsitektur maupun detail. Pemodelan software dapat dilakukan dengan cara yang sederhana seperti menggunakan flowchart dan pseudocode, tetapi saat ini telah tersedia banyak alat bantu yang lebih kompleks seperti UML yang menjadi salah satu yang paling populer (Tantra, 2012).

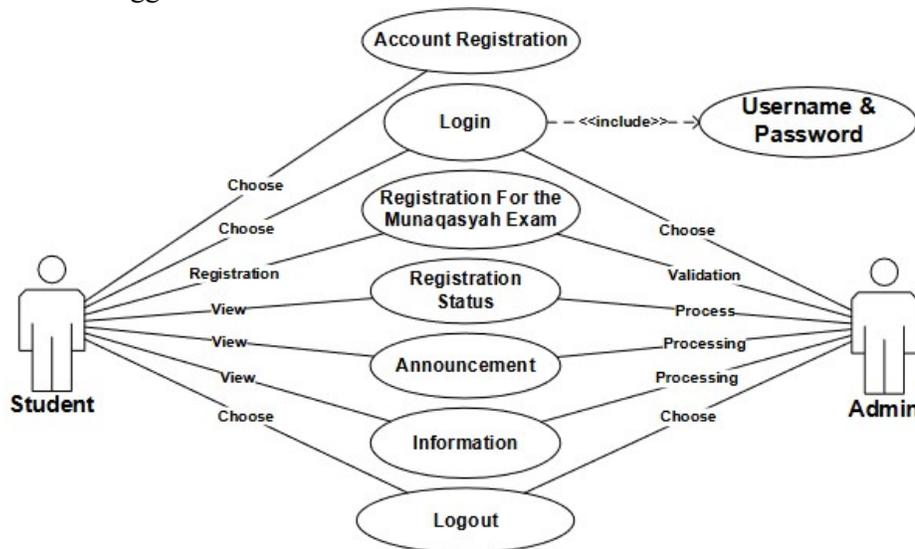
### 6.2.6 Unified Model Language (UML)

*UML (Unified Modeling Language)* merupakan bahasa berbasis grafik atau gambar untuk memvisualisasikan, menentukan, membangun, dan mendokumentasikan sistem pengembangan software berdasarkan *OO (Object-Oriented)*. *UML* sendiri juga menyediakan standar untuk penulisan *blue print system*, yang meliputi konsep proses bisnis, penulisan kelas-kelas dalam bahasa pemrograman tertentu, skema *database*, dan juga komponen yang dibutuhkan dalam sistem software (Suendri, 2018).

*UML* adalah teknologi yang membantu dalam pengembangan software berbasis objek. Alat-alat yang digunakan dalam desain berorientasi objek berbasis *UML* adalah *use case diagram*, *activity diagram*, *sequence diagram*, *class diagram* dan lain sebagainya (Samsudin et al., 2019).

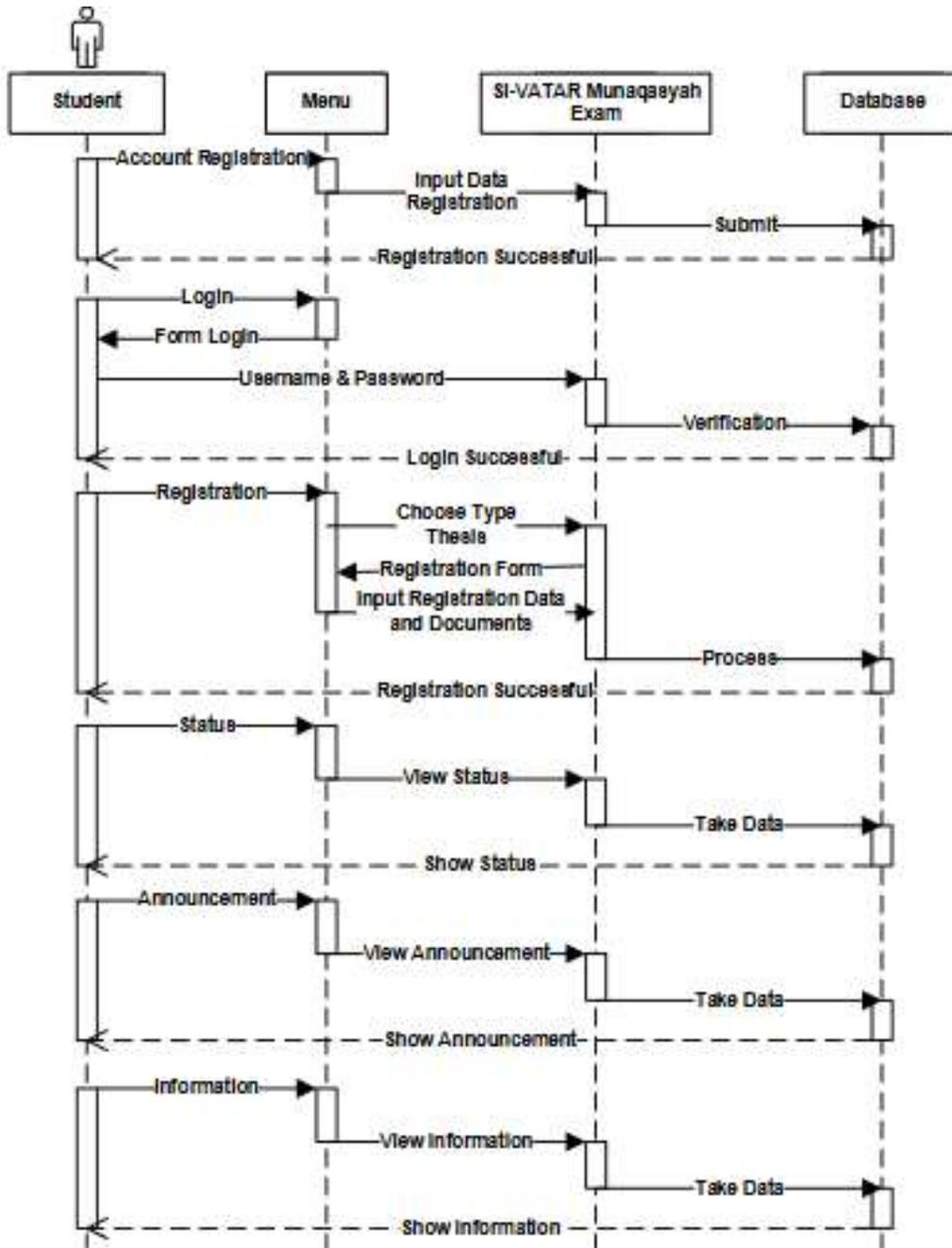
Adapun diagram yang biasa digunakan pada pemodelan UML didalam proyek sistem informasi adalah sebagai berikut:

1. *Use Case Diagram* adalah diagram untuk memodelkan perilaku suatu sistem yang akan dirancang dengan menggambarkan interaksi antara satu atau lebih aktor yang akan menggunakan sistem tersebut.



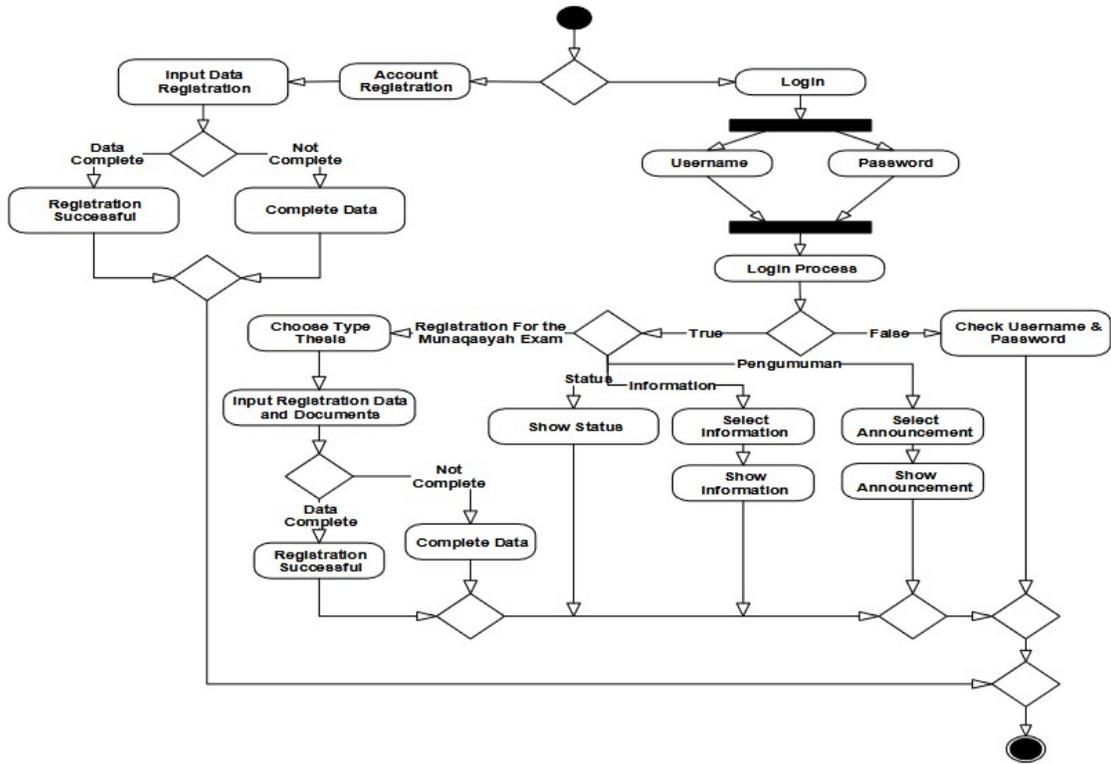
**Gambar 6. 1** Contoh *Use Case Diagram*

2. *Sequence Diagram* adalah diagram UML yang memberikan gambaran koneksi objek yang diatur dalam permintaan waktu. *Sequence Diagram* adalah penggambaran sedikit demi sedikit, termasuk urutan perubahan konsisten yang terjadi harus dilakukan untuk memberikan *use case diagram* yang sesuai.



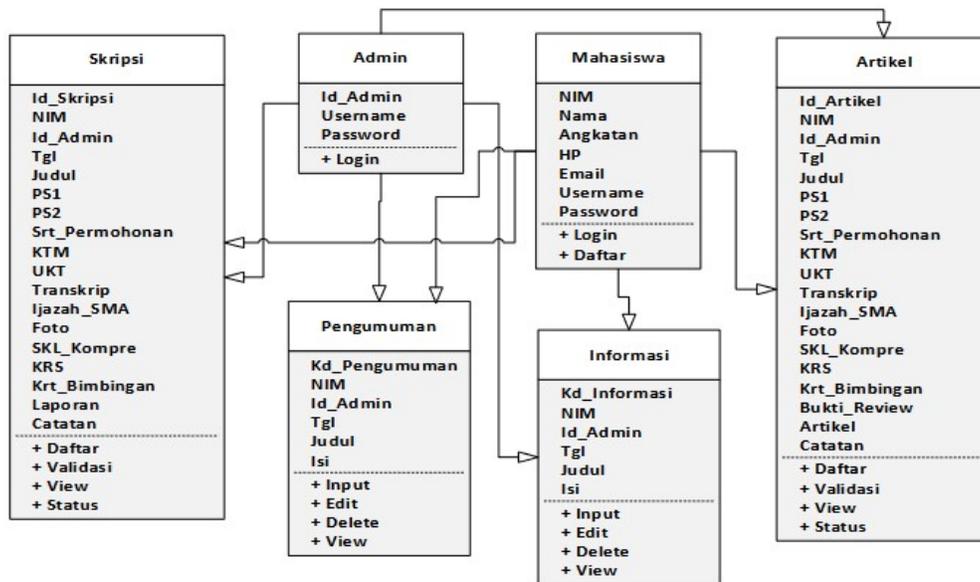
Gambar 6.2 Contoh *Sequence Diagram*

3. *Activity Diagram* menggambarkan alur kerja atau aktivitas dari suatu sistem yang ada di dalam software.



Gambar 6. 3 Contoh Activity Diagram

4. *Class Diagram* menunjukkan aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Selama tahap perancangan, *class diagram* berperan dalam menangkap struktur semua kelas yang membentuk arsitektur yang dibuat (Alda et al., 2022).



Gambar 6. 4 Contoh Class Diagram

### **6.2.7 Desain Antarmuka Pengguna**

Antarmuka pengguna, yang mencakup semua hardware, software, tampilan, menu, fungsi, keluaran, dan fitur yang mempengaruhi komunikasi dua arah antara pengguna dan komputer, menjelaskan bagaimana pengguna berinteraksi dengan sistem komputer. Karena antarmuka pengguna adalah tempat mereka berkomunikasi dengan sistem, hal itu dapat memengaruhi produktivitas secara signifikan. Pengguna lebih produktif jika antarmuka memungkinkan mereka menangani aktivitas yang lebih kompleks. Sebaliknya, produktivitas menurun jika antarmuka sulit digunakan. (Harahap et al., 2022).

Membuat desain antarmuka yang baik selama fase pengembangan memberikan keuntungan sebagai berikut:

- Menghasilkan produk software yang lebih baik.

Seperti yang telah disebutkan di atas, pengguna sering beranggapan bahwa antarmuka yang baik mencerminkan software yang baik pula.

- Meningkatkan kepuasan dan retensi klien.

Antarmuka yang user friendly akan memberikan kepuasan bagi pengguna sehingga mereka akan tetap terus mempertahankan penggunaan software Anda.

- Lebih fleksibel dalam memberikan respon terhadap umpan balik klien.

Dengan desain yang baik maka jika ada perubahan yang diinginkan klien, maka antarmuka dapat diubah secara efisien tanpa mengorbankan waktu dan kemudahan penggunaan.

- Mempersingkat waktu siklus pengembangan software.

Proses pengembangan akan berjalan lebih cepat jika desain antarmuka selesai sebelum tahap pemrograman dimulai karena programmer tidak perlu membuat antarmuka. Selain itu, programmer akan mendapatkan pengetahuan yang lebih baik tentang proses pengguna yang optimal. (Tantra, 2012).

### **6.2.8 Desain Database**

Database adalah kumpulan data yang disimpan untuk waktu yang lama, mungkin bertahun-tahun, dan sangat penting untuk organisasi dari semua ukuran. Mayoritas bisnis menggunakan database sebagai ruang penyimpanan untuk data penting mereka. Penggunaan dan pengelolaan basis data dimungkinkan oleh software khusus yang dikenal sebagai Database Management System (DBMS). Munculnya pengetahuan dan teknologi canggih di ranah basis data menyebabkan terciptanya DBMS (Putri, 2022).

Untuk memproses data pengguna dan menyajikan informasi dalam bentuk laporan, grafik, dan visual lainnya, sistem informasi bisnis membutuhkan database untuk menyimpan data. Persyaratan yang jelas harus dipenuhi saat merancang database, serta mempertimbangkan fitur-fiturnya.

Basis data yang digunakan harus mampu mereplikasi data, menghubungkan ke sistem lain, membuat cadangan dan memulihkan data, serta mengelola data (skalabilitas). Database Management System (DBMS) yang dipilih, yang merupakan software khusus

untuk menangani pengembangan, pemeliharaan, dan penggunaan basis data, mengatur semua faktor ini. Database harus mampu didesain untuk memenuhi requirements yang telah ditetapkan.

Secara definisi, desain database adalah proses untuk menghasilkan model data dari database. Desain database yang baik akan memberikan keuntungan sebagai berikut:

- Membagi informasi menjadi tabel-tabel yang berdasarkan subyek untuk mengurangi redundansi data.
- Tabel-tabel kemudian dapat dihubungkan sesuai keperluan.
- Memberikan dukungan dan memastikan akurasi dan integritas informasi.
- Memiliki kemampuan mengolah data dan menghasilkan laporan (Tantra, 2012).

### **6.3 Pembangunan Software**

Pada tahap ini, bentuk software yang akan digunakan oleh pengguna mulai dibangun, seperti sebuah gedung dimana konstruksinya mulai disusun dan dalam tahapan yang signifikan akan mulai tampak bentuk bangunannya. Aktivitas pembangunan ini, dalam proyek sistem informasi, dapat dibagi dua bagian utama yaitu:

#### **6.3.1 Pemrograman**

Pemrograman adalah serangkaian instruksi untuk menginstruksikan komputer agar melakukan suatu tindakan. Untuk memahami instruksi-instruksi ini, diperlukan bahasa pemrograman yang dapat dimengerti oleh komputer. Tujuan dari proses pemrograman tidak hanya untuk memungkinkan komputer menulis instruksi, tetapi juga untuk memperbaiki masalah dan memudahkan pengguna komputer untuk menggunakan perangkat tersebut. (Suryoprayogo & Sidiq, 2016). Aktivitas pemrograman utamanya melibatkan programmer yang menerima deliverables dari proses desain untuk diwujudkan dalam bentuk software.

Manajer proyek bertugas menentukan tim programmer dan menunjuk pemimpin tim. Pemimpin tim inilah yang kemudian akan melakukan pengaturan pembagian pekerjaan pemrograman kepada setiap programmer anggotanya. Pembagian tugas juga harus sesuai dengan sumber daya dan persyaratan kerja yang sudah diketahui saat melakukan WBS. Dalam kolaborasi, selain masing-masing programmer melakukan tugas yang telah diberikan kepadanya, ia juga harus menjaga agar bagian program yang dikerjakannya dapat berinteraksi dengan bagian lainnya yang dikerjakan oleh programmer lain. Agar proses integrasi program dapat berlangsung dengan baik, maka saat pemrograman dilakukan diperlukan alat bantu untuk mendukung kolaborasi ini. Salah satu software kolaborasi yang cukup baik dan bisa didapatkan secara gratis adalah CodeBeamer dari Intland Software (<http://www.intland.com>). CodeBeamer berbasis web, sehingga pengelolaannya bisa dilakukan baik secara lokal maupun dari lokasi yang berbeda. Software ini mendukung pemrograman kolaboratif dan juga menyertakan alat manajemen

proyek, menjadikannya alat yang hebat untuk proyek sistem informasi. Setiap anggota tim akan memiliki kredensial masuk dan kata sandi mereka sendiri, dan jika ada pembaruan proyek, semua anggota tim akan menerima informasi terbaru.

Teknik membagi tugas pemrograman tidak hanya membagi satu proyek secara merata untuk setiap pengembang. Biasanya, satu atau lebih programmer inti berdedikasi untuk memprogram modul kompleks utama, sementara programmer lain membuat program yang lebih kecil atau lebih sederhana yang mendukung modul kompleks tersebut. Contoh program dukungan tersebut adalah:

- Library

Adalah kumpulan program pembantu yang digunakan untuk membangun modul program lain, biasanya berupa prosedur atau fungsi yang dapat dipanggil dari modul lain. Contoh paling umum adalah file dengan ekstensi .DLL.

- Utilitas

Ini adalah bagian dari program yang tidak berhubungan langsung dengan bagian fungsional dari program. Contoh alat adalah fungsi ekspor data, konfigurasi data atau printer, manajemen pengguna atau pengaturan parameter.

- Laporan

Semua data yang dimasukkan ke dalam software pada akhirnya akan diproses dan menjadi informasi yang berguna bagi penggunanya. Informasi ini biasanya dalam bentuk laporan atau mungkin dalam bentuk grafik.

Program pembantu ini kemudian akan ditautkan ke program utama dan kemudian diuji untuk integrasi dan debugging sehingga benar-benar dapat menangani semua software. Program yang telah dibuat dan sedang dibuat harus disimpan dengan pengaturan yang harus ditentukan oleh pemimpin tim pengembangan dan manajer proyek. Adapun tujuannya agar:

- Setiap pengembangan dan perubahan dapat terdokumentasi dengan baik sehingga mudah untuk ditindaklanjuti jika terjadi pergantian personel di tim programmer.
- Manajer proyek bisa memantau pengembangan software dan mengambil tindakan yang diperlukan jika terjadi penyimpangan dari jadwal atau memenuhi persyaratan.
- Backup setiap pengembangan versi dapat dilakukan secara berkala, sehingga mudah untuk melacak setiap pencapaian dalam pengembangan software. Jika terjadi kesalahan dan Anda harus memulihkan, manajer proyek dapat mengidentifikasi titik milestone yang harus dikembalikan program setelah pemulihan selesai.
- Memastikan source code tidak disalahgunakan oleh pihak yang tidak berhak, karena source code merupakan “bahan mentah” yang sangat penting untuk pengembangan software, dan jika dirusak atau diubah secara tidak perlu, akan menyebabkan kerusakan fatal pada software yang dibangun.
- Manajer proyek juga harus memahami bahwa source code dan backup terbaru harus disimpan di lokasi terpisah sehingga jika terjadi masalah pada media penyimpanan fisik (seperti harddisk yang rusak) pada program yang sedang dikerjakan tidak akan mempengaruhi backup yang ada.

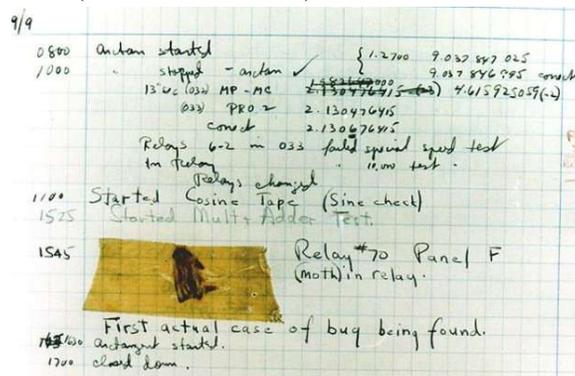
- Strategi backup harus ditentukan dengan jelas, termasuk pencadangan terjadwal, lokasi penyimpanan, dan prosedur pemulihan (jika diperlukan) (Tantra, 2012).

### 6.3.2 Integrasi Sistem

Pada titik ini, manajer proyek harus memastikan bahwa integrasi sistem berarti software dibentuk secara fisik sehingga semua persyaratan dalam desain diimplementasikan sepenuhnya. Integrasi sistem adalah konsep sistem dimana satu sistem saling berhubungan dengan sistem lainnya sesuai kebutuhan. Hal ini berguna untuk keberlangsungan sistem informasi yang juga dibutuhkan oleh sistem lainnya, atau dimana output dari satu sistem menjadi input dari sistem lainnya. Sistem terintegrasi adalah kumpulan prosedur yang digunakan untuk menghubungkan secara fisik dan fungsional berbagai sistem informasi dan aplikasi dalam konteks sistem informasi. Suatu komponen organisasi yang memasukkan data dapat diakses oleh bagian lain dari organisasi yang sama atau terpisah, yang berimplikasi pada proses bisnis. Misalnya, departemen pembelian dan departemen keuangan dapat mengakses data dari modul pembelian. Atau individu pada tingkat yang berbeda dalam departemen penjualan, seperti penjual, manajer inventaris, penyelia, dan manajer penjualan, dapat melihat data penjualan (Warkim & Sensuse, 2017).

### 6.3.3 Debugging

Istilah debugging digunakan untuk menggambarkan aktivitas yang dilakukan programmer untuk menemukan masalah atau potensi masalah dalam program yang mereka tulis. Debugging berbeda dengan pengujian sistem karena debugging masih dilakukan di lingkungan pemrograman, hanya menggunakan data percobaan, tidak menggunakan data kasus (Utama et al., 2022).



Gambar 6.5 Bug yang pertama kali ditemukan

Bug adalah salah satu masalah software yang paling umum. Bug adalah kesalahan, cacat, malfungsi, atau kerusakan pada software yang mencegah sistem bekerja sebagaimana mestinya. Kesalahan software dapat terjadi karena kesalahan pengembang yang tidak disengaja. Pengembang dapat membuat kesalahan dalam kode sumber atau dalam desain

software itu sendiri (Azhar & Rochimah, 2016). Bug adalah kesalahan logis dan teknis dalam suatu program. Kesalahan logika dapat berasal dari penyelesaian desain atau pemrograman. Dalam perancangan, kesalahan dapat muncul karena kesalahan (kesalahan) murni dari perancang software, sehingga ketika diwujudkan dalam bentuk program, kesalahan tersebut terbawa sehingga menyebabkan software memproses atau memberikan informasi yang tidak benar. Bug juga bisa terjadi, meskipun desainnya sudah benar, tetapi mengandung kemungkinan bug karena tidak memperhitungkan interaksi dengan modul lain dalam sistem. Ini bisa terjadi, misalnya saat menerima data, atau dalam bentuk variabel yang tipe datanya berbeda antara parameter dan parameter sebenarnya dari modul lain. Untuk mengatasi kesalahan tersebut, debugging harus dilakukan setelah integrasi sistem. Lakukan ini dengan data contoh untuk beberapa kondisi, dan periksa keluaran yang dihasilkan oleh perhitungan. Kesalahan teknis dalam pemrograman juga bisa mengakibatkan hal yang sama. Kesalahan ini biasanya terjadi saat programmer teknis menulis kode dalam bahasa pemrograman. Jadi kesalahan teknis lebih berkaitan dengan kemampuan programmer untuk melakukan pekerjaan itu.

Integrated Development Environment (IDE) saat ini menyediakan alat debugging terintegrasi sehingga pemrogram dapat melangkah melalui simulasi software. Pemrogram hanya perlu menjalankan aplikasi, dan program akan dieksekusi baris demi baris sehingga hasilnya dapat diamati. Jika Anda menemukan nilai yang tidak biasa atau salah atau menyebabkan kesalahan, Anda dapat menghentikan eksekusi program dan memperbaikinya. Dalam beberapa IDE yang bagus, bahkan pemrogram dapat menelusuri modul, fungsi, atau prosedur lain yang dipanggil dari program yang sedang dijalankan, membuatnya lebih mudah untuk men-debug sistem terintegrasi. Seringkali, bug tidak ditemukan hingga software dirilis dan digunakan oleh klien. Perbaikan harus memberi tahu pengguna dan memberikan cara termudah untuk menginstal perbaikan. Biasanya fix ini dipasang sebagai patch atau disebut juga dengan fix sementara. Jika masalah ditemukan cukup serius, perbaikan biasanya menghasilkan perubahan pada nomor versi rilis software, baik minor maupun mayor. Perubahan versi minor ditandai dengan bertambahnya nomor versi di sebelah kanan titik, misalnya 1.0 menjadi 1.1, atau bahkan 1.0.0 menjadi 1.0.1. Ketika versi utama berubah, termasuk perubahan yang sangat penting, nomor versi perubahan adalah nomor versi di sebelah kiri titik, misalnya 1.1 ke 2.0. Perubahan pada rilis ini harus selalu dicatat sebagai riwayat versi, termasuk perbaikan dan perubahan yang dilakukan pada setiap rilis (besar dan kecil), agar pembaruan dapat lebih mudah dicari jika diperlukan (Tantra, 2012).

#### **6.3.4 Programmer**

Programmer harus menerjemahkan requirements yang ditentukan dalam rancangan menjadi kode program, jika dikompilasi menjadi software yang merupakan jantung dari sistem informasi, software harus memiliki sifat yang sesuai dengan fungsi yang dimaksudkan, dapat digunakan secara efektif, dan tidak lebih. Contoh yang menarik adalah mesin pencari Google, yang hanya memiliki satu fungsi di luar, yaitu untuk

mencari informasi, meskipun ada berbagai teknologi dan algoritma yang sangat kompleks di baliknya. Untuk menjadi seorang programmer syarat yang paling utama tentunya menguasai bahasa pemrograman, karena bahasa pemrograman adalah satu-satunya alat penulisan kode program untuk membangun software.

Selain itu, programmer juga perlu memiliki kemampuan logika untuk menerjemahkan desain menjadi kode program. Untuk menyelesaikan tugas-tugas yang telah ditentukan sesuai jadwal, programmer juga harus memiliki kemampuan untuk bekerja di bawah tekanan. Agar program dapat dilacak dengan mudah jika terjadi masalah, akan lebih baik lagi jika programmer juga dapat menangani tugas dokumentasi. Pemrogram harus dapat bekerja secara mandiri maupun kolaboratif. Di sebagian besar proyek, kerja sama lebih dari sekadar bekerja dengan pemrogram lain untuk menyertakan anggota tim proyek yang beragam (Ardiana et al., 2021).

#### **6.4 *Quality Assurance (QA)***

Untuk menciptakan sistem yang terjamin kualitasnya, Quality Assurance (QA) melibatkan pengujian sistem atau produk aplikasi untuk memastikan bahwa produk tersebut diproduksi sesuai dengan persyaratan yang ditetapkan (Jauhari et al., 2019). Sekalipun proyek selesai sesuai jadwal dan memenuhi semua persyaratan, klien mungkin tidak puas dengan produk jadinya. Keluhan seperti tidak praktis, tidak terintegrasi dengan baik, berjalan lambat, dan hal serupa lainnya dapat digunakan untuk mengidentifikasi produk berkualitas rendah. Berdasarkan pernyataan di atas, ada dua aspek kualitas yang perlu diperhatikan, yaitu:

- Kualitas deliverables, yaitu kualitas dari sistem informasi yang dihasilkan oleh pelaksanaan proyek.
- Kualitas proyek, yaitu kualitas yang berhubungan dengan pelaksanaan pekerjaan dalam proyek.

Selain keluaran dengan kualitas yang diharapkan, proyek yang baik juga dapat menghasilkan produk di bawah standar. Namun, tidak menutup kemungkinan proyek-proyek dikerjakan dengan asal-asalan namun tetap menghasilkan keluaran yang berkualitas tinggi. Proyek harus memiliki rencana kualitas untuk menghindari situasi di mana hasil yang dihasilkan kurang jelas tentang kualitasnya (Tantra, 2012).

##### **6.4.1 *Rencana Kualitas (Quality Plan)***

Berikut adalah langkah-langkah untuk membuat rencana kualitas yang baik:

1. Menentukan sasaran, yang bersifat kuantitatif, sehingga pencapaiannya dapat diukur. Sasaran ini dapat berupa misalnya: "mengurangi sampai maksimal hanya 1 bugs dalam satu modul transaksi". Jangan membuat sasaran berupa "zero defect" atau software yang dihasilkan tidak mengandung kesalahan sama sekali, karena hal ini tidak dapat dicapai dalam waktu yang singkat.
2. Tentukan ukuran apa yang sedang dibangun. Jika diukur dalam satuan, tentukan ukuran unit software, seperti berapa baris kode program yang ada, atau berapa banyak

prosedur dan fungsi yang ada di dalamnya. Dalam proyek, langkah-langkah yang menjadi dasar perencanaan mutu tentu saja software terintegrasi sebagai sistem informasi. Hal ini dapat diukur dengan jumlah modul, jumlah tabel, jumlah prosedur dan fungsi, dan komponen lain dalam software.

3. Pelaksana aktivitas untuk mengurangi kesalahan dalam software. Dalam kegiatan ini, Anda harus mulai menyusun hal-hal sebagai berikut:
  - a. Strategi testing.
  - b. Perencanaan testing.
  - c. Spesifikasi testing.
  - d. Prosedur testing.

Rencana mutu juga harus mencakup unsur-unsur berikut:

- Hal-hal apa saja yang perlu melalui pemeriksaan kualitas. Khususnya adalah deliverables proyek, termasuk desain, dokumen, dan tentunya software,
- Cara apa yang terbaik untuk melakukan pemeriksaan terhadap kualitas. Pilihan terbaik adalah mengikuti standar pemeriksaan untuk setiap deliverables. Rencana kualitas harus memberikan instruksi untuk melakukan inspeksi software yang komprehensif dan mudah diikuti.
- Kapan harus melakukan pemeriksaan kualitas. Dalam kebanyakan kasus, pemeriksaan kualitas dilakukan beberapa saat sebelum software rilis. Nyatanya, semakin cepat inspeksi dilakukan, semakin kecil kemungkinan akan ada masalah atau pengerjaan ulang yang lebih besar.
- Siapa saja yang harus terlibat. Intinya, metode pemeriksaan kualitas ini dapat mencakup hampir semua anggota tim proyek. Namun demikian, terutama anggota tim yang terlibat aktif dalam menciptakan hasil akhir proyek, seperti desainer, pemrogram, dan penguji, serta pengguna software.
- Material kualitas apa saja yang akan digunakan. Ini bisa berupa dokumen-dokumen yang dihasilkan dalam proses desain, software itu sendiri, database dan semua bagian dari deliverables.

Manajer proyek dapat berfokus pada hal-hal berikut untuk lebih memahami nilai kualitas:

- Lebih akrab dengan rekayasa software dan software quality assurance serta menerapkan Total Quality Management (TQM).
- Berbagi tanggung jawab dalam hal kontrol kualitas dengan anggota tim proyek.
- Memberikan pelatihan yang relevan dengan peningkatan kualitas software.
- Menyediakan alat bantu yang memadai bagi programmer dalam pelaksanaan tugasnya.
- Laksanakan perencanaan yang efektif seperti diuraikan di atas.
- Gunakan alat bantu untuk melakukan pengukuran kualitas seperti Software Engineering Institute's Capability Maturity Model.
- Mempromosikan kualitas dengan berbagi informasi pada forum-forum dalam proyek.

- Harus ada dukungan yang jelas untuk peningkatan kualitas dan menentang kualitas yang berkompromi untuk mematuhi batasan tertentu.
- Memberikan reward kepada anggota tim yang melaksanakan tugasnya sesuai dengan standar kualitas, sebagaimana ditentukan oleh evaluasi bersama (Tantra, 2012).

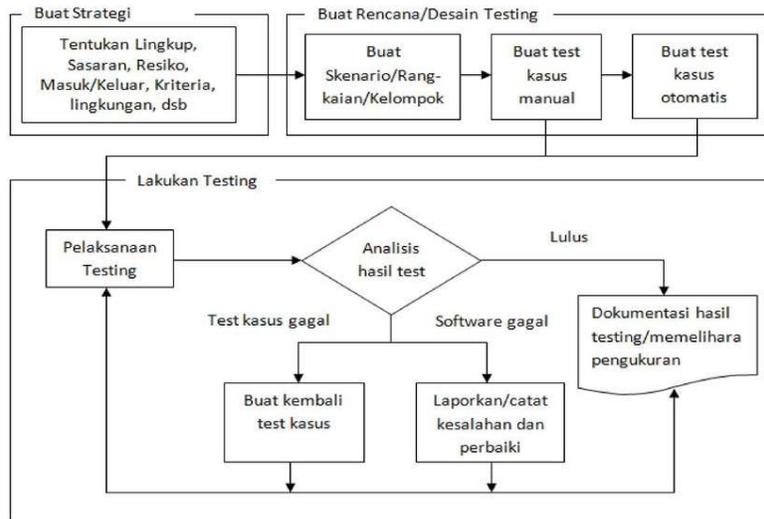
### 6.4.2 Testing

Testing merupakan metode yang digunakan untuk mengevaluasi kualitas, akurasi, dan kelengkapan software komputer yang sedang dibangun. Pengujian harus secara sistematis mengidentifikasi berbagai kelas kesalahan dalam waktu paling sedikit dan dengan sedikit usaha. Informasi yang dikumpulkan selama pengujian juga dapat memberikan petunjuk tentang keandalan dan kualitas program (Istiyana, 2022).

Untuk mengetahui apakah komponen sistem beroperasi dengan benar, maka dilakukanlah suatu pengujian. Menemukan kekurangan atau kesalahan yang perlu diperbaiki adalah tujuan lain dari pendekatan ini. (Kusrini & Koniyo, 2007).

### 6.4.3 Metodologi Testing

Model proses testing dapat digambarkan sebagai berikut (Tantra, 2012):



**Gambar 6. 6** Model Proses Testing

### 6.4.4 Strategi Testing

Pendekatan pengujian software memastikan bahwa pengembangan software berlangsung tanpa gangguan, bahwa software saat ini sesuai dengan spesifikasi, dan program bebas cacat (Huda et al., 2022). Setiap tingkat tes memiliki strategi pengujian yang dikembangkan untuk itu. Tim pengujian akan menganalisis persyaratan tim proyek, membuat strategi pengujian, dan membicarakan rencana.

Deskripsi hardware, software, alat uji, dan rincian peran dan tanggung jawab setiap orang yang berpartisipasi dalam proses pengujian diperlukan untuk menetapkan strategi pengujian. Selain itu, software teknis dan fungsional memerlukan dokumentasi persyaratan, penyesuaian, dan desain teknis dan fungsional (Tantra, 2012).

#### **6.4.5 Rencana Test (Test Plan)**

Rencana test (pengujian) menjelaskan unit software yang akan diuji, level di mana setiap unit akan diuji, urutan pengujian akan dijalankan, bagaimana strategi pengujian akan diterapkan, dan lingkungan tempat pengujian akan dijalankan.

Berdasarkan temuan studi tentang persyaratan fungsional dan pengelompokan logis dari fungsi bisnis, skenario testing dan kasus dibuat. Ini kemudian dapat dijelaskan dalam bentuk skrip pengujian, kemudian tentukan parameter, data pengujian, dan hasil yang diantisipasi (seperti pembaruan basis data, file keluaran, laporan, dan sebagainya). Skenario test didesain untuk mewakili keadaan biasa maupun yang tidak biasa/ pengecualian yang mungkin terjadi dalam sistem.

Setelah itu, skrip pengujian akan digunakan untuk mengimplementasikan skenario pengujian. Langkah-langkah yang diperlukan dalam satu atau beberapa skenario ditentukan oleh skrip. Transaksi atau prosedur yang terjadi dalam keadaan operasional biasa diwakili oleh skrip pengujian. Skrip ini juga menyediakan sejumlah skenario pengujian dan data spesifik yang akan digunakan dalam pengujian. Persyaratan harus dirujuk dalam skrip pengujian, dan harus diverifikasi bahwa pengujian dilakukan dalam ruang lingkup yang ditentukan.

Sebelum proses pengujian dimulai, data yang diperlukan untuk itu harus disiapkan. Ini akan digunakan untuk memeriksa fungsionalitas sistem dalam pengaturan yang terkontrol. Adakan pertemuan untuk membahas persiapan pengujian sebelum memulai pengujian, dan pastikan semuanya telah disiapkan sesuai dengan strategi pengujian yang dibuat (Tantra, 2012).

Berikut ini adalah beberapa tingkatan dalam melakukan testing. Dalam setiap tingkatan testing, dokumentasi harus selalu dilakukan.

##### **1. Testing Unit (Unit Testing)**

Dalam testing ini, terdapat dua cara yang dapat diterapkan yaitu analisis statis dan analisis dinamis. Analisis statis melakukan penyelidikan terhadap kode sumber (source code) software tanpa menjalankannya. Sedangkan analisis dinamis adalah mengamati jalannya software pada saat dijalankan. Tes unit biasanya ditulis dan dijalankan oleh pengembang software untuk memastikan bahwa kode sesuai dengan desainnya dan berperilaku sebagaimana mestinya. Tujuan dari pengujian unit adalah untuk mengisolasi setiap bagian dari program dan menunjukkan bahwa masing-masing bagian sudah benar (Istiyana, 2022).

##### **2. Testing Integrasi Sistem dan Walkthrough**

Adalah aktivitas pengujian software di mana modul software individual dirakit dan diuji sebagai sebuah kelompok. Pengujian integrasi mengambil modul input yang telah diuji unit, mengelompokkannya ke dalam agregat yang lebih besar, menerapkan pengujian

yang ditentukan dalam rencana pengujian integrasi ke agregat tersebut, dan memberikan output sistem terintegrasi yang siap untuk pengujian sistem (Istiyana, 2022). Tahapan ini juga dapat dimulai dengan berbagai persiapan teknis, seperti:

- Membuat dan mengatur koneksi untuk database yang akan digunakan.
- Kelola pengaturan, hak, dan otoritas setiap pengguna.
- Tetapkan parameter keamanan sistem.
- Melakukan perubahan parameter sistem, seperti:
  - Mengaktifkan fitur dan fasilitas sistem.
  - Penomoran transaksi.
  - Koneksi dengan sistem ataupun modul lainnya.
- Membuat prosedur pencadangan dan pemulihan (Tantra, 2012).

### 3. Testing Fungsional (Functional Test)

Dalam tahap ini, Anda akan menguji fungsionalitas setiap modul pada titik ini untuk melihat apakah kinerjanya sesuai rencana. Proses input dan output untuk setiap modul perlu diperiksa secara menyeluruh. Sepanjang tes ini, ada dua faktor yang perlu diingat:

- Verifikasi: menentukan apakah sistem dibangun sesuai dengan rencana.
- Validasi: menentukan apakah sistem yang dibuat sesuai dengan keinginan klien.

Verifikasi berkaitan dengan persyaratan di mana tim proyek membuat spesifikasi sistem informasi. Metode blackbox, yang tidak memerlukan pengetahuan tentang cara kerja internal software, biasanya digunakan dalam pengujian ini. Untuk memverifikasi bahwa modul yang diuji beroperasi sebagaimana mestinya, perlu untuk menguji apakah data input dan output sesuai dengan standar. Akibatnya, pengujian menyeluruh juga harus dilakukan, termasuk menggunakan berbagai pengaturan dan kondisi dan menghasilkan hasil yang konsisten dengannya.

Fungsionalitas modul harus memenuhi kebutuhan penggunanya, sedangkan validasi lebih mementingkan kualitas. Software harus mudah digunakan, efektif, dan tidak boleh menghalangi pekerjaan pengguna, sebaliknya itu harus digunakan sebagai alat untuk membantu mereka menyelesaikannya (Tantra, 2012).

### 4. Testing Kasus (Test Case)

Testing yang dilakukan sebelumnya seperti yang disebutkan di atas biasanya masih menggunakan data sampel yang tidak sesuai dengan kondisi kerja sebenarnya atau rutinitas kerja sehari-hari. Misalnya, input data dalam satu DO untuk menguji transaksi seringkali dibatasi hingga 5 item, meskipun dalam transaksi sehari-hari yang sebenarnya, satu DO biasanya memiliki hingga 20 item produk.

Pengguna dapat disertakan dalam pengujian kasus ini sehingga pemahaman tim pengujian tentang skenario aktual dapat ditingkatkan. Sistem yang lengkap perlu digunakan langkah demi langkah, seperti saat melakukan penelusuran. Fokus pada proses bisnis yang telah ditetapkan. Lakukan proses pelatihan software terlebih dahulu, meskipun tidak harus mendalam, agar lebih mudah melakukan pengujian dengan pengguna. Jelaskan secara singkat fitur dan kemampuan software, termasuk cara menggunakan berbagai menu, tombol, dan aksesoris lainnya. Untuk menguji skenario ini, tim pengujian dapat bekerja dengan atau bersama pengguna (Tantra, 2012).

#### 5. Testing Paralel/Audit

Testing ini dilakukan bersamaan dengan software lama yang sudah digunakan, dan dipastikan hasilnya akurat sehingga dapat dibandingkan dengan program baru yang akan datang. Jika sebelumnya tidak ada software yang digunakan, perbandingan juga dapat dilakukan secara manual (Istiyana, 2022).

#### 6. Testing Usability

Testing usability mengobservasi pengguna sebuah desain kemudian diambil data dan menganalisisnya. Data kemudian dikumpulkan dan dianalisis. Untuk menghasilkan rekomendasi perbaikan sistem di masa mendatang, teknik ini dapat digunakan untuk mengidentifikasi pengguna, memilih fungsi, membuat tugas pengujian kegunaan, menentukan waktu pengujian kegunaan, melakukan pengujian kegunaan, merekam hasil pengujian, mengukur kegunaan menggunakan kuesioner, dan melakukan analisis (Sriwulandari et al., 2014).

#### 7. Testing End-to-End

Pengujian ini dilakukan untuk semua software dalam skenario yang mendekati kondisi operasional sebenarnya, seperti koneksi dengan basis data, jaringan, telekomunikasi, atau hubungan dengan software atau hardware lain, sama seperti pengujian integrasi sistem (Tantra, 2012).

#### 8. Testing Regresi (Regression Testing)

Program diubah ketika modul perusahaan tambahan disertakan sebagai bagian dari pengujian integrasi. Input atau output baru dapat terjadi, saluran aliran data baru dapat dibuat, dan logika kontrol baru dapat digunakan. Fungsi yang selalu beroperasi tanpa hambatan sekarang mungkin mengalami masalah. Pengujian regresi memastikan bahwa modifikasi atau perbaikan tidak menyebabkan kesalahan tambahan. Data yang sama yang digunakan dalam uji awal juga harus digunakan dalam uji regresi (Istiyana, 2022).

#### 9. Testing Kinerja (Performance Testing)

Komputer, jaringan, aplikasi software, atau kecepatan atau efisiensi perangkat dapat dievaluasi menggunakan pengujian kinerja software. Menggunakan tolak ukur yang ditetapkan, metode pengujian ini menilai seberapa baik kinerja sistem baru dibandingkan dengan sistem sejenis lainnya (Istiyana, 2022).

#### 10. Testing Beban (Load Testing)

Stabilitas dan keandalan sistem diperiksa melalui pengujian beban. Evaluasi ini dapat menunjukkan seberapa baik sistem menangani kesalahan dan kondisi beban yang sangat tinggi. Jumlah pengguna yang mengakses situs web sekaligus dapat bervariasi, jadi kami akan menggunakan hasil uji beban untuk mengevaluasi seberapa cepat dan seberapa baik situs web merespons ketika sejumlah pengguna tertentu mengaksesnya sekaligus (Suprpto & Sasongko, 2021).

#### 11. Testing Instalasi (Installation Testing)

Metode pengujian ini bertujuan untuk menentukan apakah ada masalah dengan pengguna selama penginstalan dan pengoperasian program di setiap perangkat mobile pengguna (Purbaningtyas, 2019).

#### 12. Testing Keamanan/Penetrasi (Security/Penetration Testing)

Adalah prosedur untuk menetapkan apakah sistem informasi mengamankan data dan mempertahankan fungsinya sebagaimana dimaksud atau tidak, dan itu terdiri dari enam komponen: integritas, otentikasi, otorisasi, ketersediaan, dan non-penolakan. Strategi pengujian ini juga memastikan bahwa sistem baru memiliki kontrol akses yang sesuai dan menggunakannya (Istiyana, 2022).

#### 13. Testing Recovery/Error

Bertujuan untuk menilai ketahanan aplikasi terhadap kegagalan fungsi hardware, crash, dan masalah terkait lainnya. Pengujian pemulihan melibatkan pemecahan software secara sengaja untuk memastikan bahwa proses pemulihan dilakukan dengan benar (Istiyana, 2022).

#### 14. Testing Kompatibilitas (Compatibility Testing)

Kompatibilitas dengan hardware dan software lain diperlukan untuk beberapa software. Misalnya, software retail harus dapat berinteraksi secara cepat dengan sistem POS, dan software absensi harus dapat dibaca dari mesin absensi. Lakukanlah uji kompatibilitas ini. Pastikan data yang dikirimkan atau diterima dari program lain benar dan akurat jika software tersebut juga harus terhubung dengan software lain (Tantra, 2012).

#### 15. Testing Perbandingan (Comparison Testing)

Pengujian ini membandingkan program dengan software lama yang telah ada atau dengan item software serupa lainnya (seperti dari pesaing) (Tantra, 2012).

#### 16. Testing Penerimaan (Acceptance Test)

Sebelum software diterapkan dan digunakan secara maksimal, pengujian ini memberi pengguna kesempatan untuk mengonfirmasi fungsinya. Meskipun pengguna klien terlibat aktif dalam pengujian penerimaan ini, tim proyek yang relevan terus memberikan bantuan penuh. Setelah itu, kriteria persetujuan akan dibuat oleh tim pengujian dan pengguna (Tantra, 2012).

#### 17. Testing Alpha

Sekelompok pengguna akhir menguji proyek dalam bentuk aplikasi dalam pengujian alpha, memberi tahu pengembang proyek jika ada bug yang tersisa setelah mereka menemukan dan melaporkan kesalahan apa pun dalam aplikasi (Aldisa, 2022).

#### 18. Testing Beta

Pengujian beta adalah pengujian yang dilakukan oleh pengguna di lingkungannya sendiri ketika pengembang tidak dapat lagi mengontrol lingkungan software (Wibisono & Baskoro, 2002).

### **6.5 Dokumentasi**

Setiap proyek sistem informasi menghasilkan sejumlah besar dokumentasi yang terkait dengan semua tahapan proyek. Dalam proyek kecil, mungkin hanya ada beberapa folder file, sedangkan dalam proyek besar, mungkin ada beberapa lemari arsip. Dalam proyek sistem informasi berskala besar, bahkan dapat menempati beberapa ruangan, sebanding dengan perpustakaan. Penyusunan dokumentasi pasti akan memakan banyak waktu dan juga memakan sebagian dari anggaran proyek, sehingga manajer proyek harus memperhatikan hal ini, karena dokumentasi sangat penting untuk fungsi-fungsi berikut:

- Media komunikasi antara anggota tim proyek dan pemangku kepentingan (stakeholder) lainnya.
- Tempat menyimpan informasi yang dibutuhkan oleh tim yang akan melakukan pemeliharaan sistem (system maintenance).
- Memberikan informasi kepada manajemen untuk membantu menyiapkan rencana, anggaran, dan jadwal untuk pengembangan software.
- Penyediaan informasi tentang cara menggunakan dan mengelola sistem (Tantra, 2012).

Secara umum dokumentasi dapat dibagi dalam dua kelompok yaitu:

### **6.5.1 Dokumentasi Proses**

Aktivitas ini mendokumentasikan proses pengembangan software dan proses pemeliharannya. Dokumentasi proses dapat dibagi lagi menjadi beberapa kategori yaitu:

- Rencana, perkiraan, dan jadwal, yang merupakan dokumen yang dihasilkan oleh manajer proyek untuk memperkirakan dan mengontrol proses pengembangan software.
- Laporan, yaitu dokumen yang menjelaskan penggunaan sumber daya proyek selama proyek berlangsung.
- Standar, yaitu dokumen yang menjelaskan bagaimana proses akan dilaksanakan, dapat didasarkan pada standar internal, nasional atau internasional.
- Dokumen kerja, biasanya dokumen komunikasi teknis dalam proyek, mencatat pemikiran dan ide anggota tim proyek, dan merupakan versi tentatif dari dokumen produk. Dokumen ini juga mencakup strategi implementasi yang teridentifikasi dan pemecahan masalah, seringkali mencantumkan alasan untuk keputusan desain.
- Catatan dan pesan email yang menangkap bagian dari komunikasi harian anggota tim proyek (Tantra, 2012).

### **6.5.2 Dokumentasi Produk**

Dokumentasi ini berlaku untuk produk software yang dikirimkan ke klien dan penggunanya. Berbeda dengan dokumentasi proses pada umumnya, dokumentasi produk memiliki umur simpan yang lebih lama dan berkembang seiring dengan perkembangan software yang akan dibuat. Dokumentasi produk ini dibagi menjadi:

- Dokumentasi Pengguna

Dokumentasi pengguna adalah dokumentasi yang disertakan dengan software yang berisi petunjuk terperinci untuk software. Dokumentasi pengguna menjelaskan setiap fitur software dan menjelaskan cara menggunakan setiap fitur. Selain itu, dokumentasi pengguna dapat memberikan penjelasan tentang setiap masalah atau kesalahan yang muncul dan bagaimana cara mengatasinya (Najwaini & SN, 2012). Penting untuk membedakan pengguna menjadi:

- Pengguna akhir, yang menggunakan software untuk membantu mereka melakukan pekerjaannya dan tidak peduli dengan cara kerja komputer atau cara sistem dikelola.
- Administrator sistem bertanggung jawab untuk mengelola sistem yang digunakan oleh pengguna akhir. Jika sistem melibatkan jaringan, orang ini dapat bertindak sebagai administrator jaringan, mungkin ahli dalam memperbaiki sistem secara teknis, atau bertindak sebagai perantara antara pengguna dan pembuat sistem.

Untuk dapat menjawab kebutuhan dari berbagai tingkatan pengguna, setidaknya perlu dibuat 5 jenis dokumentasi pengguna, atau setidaknya dibagi menjadi 5 bab terpisah. Jenis dokumen pengguna ini yaitu:

- Deskripsi fungsional sistem memberikan penjelasan tentang persyaratan sistem dan menjelaskan secara singkat apa yang disediakan sistem. Dokumen ini memberikan tinjauan singkat tentang sistem di mana pengguna dapat mempelajarinya dan memutuskan apakah itu yang dia butuhkan.
- Dokumentasi instalasi sistem untuk administrator sistem. Dokumentasi ini harus merinci cara menginstal sistem di lingkungan tertentu, termasuk persyaratan hardware minimum. Meskipun biasanya proses instalasi dilakukan dengan bantuan "wizard", dokumentasi ini tetap diperlukan jika muncul masalah selama proses instalasi, seperti konflik dengan sistem operasi atau sistem lain.
- Manual pengantar memberikan informasi pengantar informal yang menjelaskan penggunaan "normal" dari sistem. Informasi ini berupa bagaimana sistem melakukan booting dan apa saja yang dapat dimanfaatkan oleh pengguna. Ilustrasi juga disediakan untuk memperjelas informasi. Informasi lengkap tentang cara melakukan pemulihan jika terjadi kesalahan pengguna.
- Manual referensi sistem menjelaskan fungsi sistem dan penggunaannya untuk pengguna berpengalaman. Dokumen ini juga menyediakan daftar lengkap pesan error dan penjelasannya, serta cara mengatasinya. Dokumen tersebut juga harus memberikan deskripsi teknis formal dari sistem.
- Manual administrasi sistem harus memberikan penjelasan tentang perintah dan kontrol sistem. Jelaskan pesan yang akan ditampilkan sistem ketika berinteraksi dengan sistem lain dan bagaimana menanggapi (Tantra, 2012).

- Dokumentasi Sistem

Dokumentasi sistem terdiri dari semua dokumen yang menggambarkan sistem, mulai dari spesifikasi kebutuhan sampai 1 dengan rencana pengujian akhir penerimaan (final acceptance test). Dokumen sistem juga harus disusun dengan tinjauan yang dapat mengarahkan pembaca ke arah penjelasan yang lebih formal dan terperinci dari setiap aspek sistem.

Dokumentasi sistem yang lengkap dapat terdiri dari dokumen-dokumen berikut:

- Persyaratan dokumen dan alasan yang mendasarinya.
- Dokumen yang menjelaskan arsitektur sistem secara keseluruhan, termasuk dokumentasi basis data.

- Dokumen yang menjelaskan modul atau program yang merupakan bagian dari sistem.
- Dokumen yang menjelaskan fungsionalitas dan antarmuka dari setiap komponen dalam sistem.
- Daftar kode sumber program (program source code listing) dimana pada bagian yang kompleks perlu diberikan komentar yang menjelaskan proses pada bagian tersebut. Jika program menggunakan nama yang bermakna (misalnya penamaan variabel, prosedur, fungsi dan sebagainya) dan dilakukan dengan cara yang baik dan terstruktur, maka akan lebih mudah untuk dipahami dan dipelihara. Saat ini, kode sumber sebagian besar disimpan secara digital daripada dicetak, karena jumlah baris program bisa sangat panjang.
- Dokumen validasi yang menjelaskan bagaimana setiap program divalidasi dan bagaimana informasi validasi terkait dengan persyaratan.
- Panduan pemeliharaan sistem yang menjelaskan masalah umum yang diketahui, juga menjelaskan bagian-bagian sistem yang bergantung pada hardware dan software (Tantra, 2012).

### 6.5.3 Kualitas Dokumen

Jika dokumen tidak lengkap, kualitas sistem juga akan menurun, karena dokumen yang jelas sangat membantu untuk pemeliharaan sistem, misalnya, dan membuat pekerjaan menjadi lebih efisien.

#### 1. Struktur Dokumen

Dokumen terstruktur pasti akan memudahkan orang untuk memahami dan menggunakannya. Berhati-hatilah untuk membagi materi menjadi beberapa bagian terpisah sehingga jika ada perubahan tidak akan mempengaruhi bagian lain dari dokumen tersebut. Struktur dokumen dapat diatur dalam bentuk bab, yang selanjutnya dibagi menjadi beberapa bagian dan subbagian. Pengguna akan menemukan informasi yang mereka butuhkan lebih cepat dengan bantuan dokumen terstruktur..

Struktur dokumen berikut disarankan untuk dokumentasi pengguna sesuai dengan Standar IEEE 2001 (Najwaini & SN, 2012):

Tabel 6.1 Struktur Dokumen

| Komponen          | Deskripsi   |
|-------------------|---|
| Data Identifikasi | Informasi yang memungkinkan Anda mengidentifikasi dokumen secara khusus, seperti judul atau nomor |
| Daftar Isi        | Nama bab/seksi dan nomor halaman  |
| Daftar Gambar     | Nomor dan judul gambar  |

|  |   |
|--|---|
| Informasi Mengenai Penggunaan Dokumen    | Penjelasan tentang cara menggunakan dokumen untuk berbagai pembaca  |
| Konsep Pengoperasian                     | Penjelasan tentang landasan teoritis untuk menggunakan software   |
| Prosedur                                 | Pentunjuk tentang cara menggunakan software yang dimaksudkan untuk membantu Anda menyelesaikan tugas Anda |
| Informasi Perintah-perintah Software     | Penjelasan dari setiap perintah yang dapat didukung oleh software   |
| Pesan Kesalahan dan Penyelesaian Masalah | Penjelasan pesan kesalahan software, beserta langkah-langkah untuk memperbaikinya.                        |
| Sumber Informasi Lain yang Terkait       | Referensi atau tautan ke dokumen tambahan yang menawarkan detail lebih lanjut                             |
| Fitur-fitur Navigasi                     | Fitur yang memudahkan pembaca untuk menavigasi ke bagian lain dan memahami di mana teks saat ini berada.  |
| Indeks                                   | Daftar dari istilah-istilah kunci dan halaman referensinya  |
| Kemampuan Pencarian                      | Fitur ini merupakan penyempurnaan untuk mempermudah pencarian topik atau istilah dalam dokumen digital    |

Tidak diharuskan untuk memasukkan semua elemen yang tercantum di atas karena standar dokumentasi ini luas yang dapat digunakan sebagai panduan saat membuat dokumentasi. Pedoman berikut juga dapat digunakan sebagai referensi saat menyusun dokumen (Tantra, 2012):

- Semua dokumen, seberapapun singkatnya, harus memiliki halaman judul yang memberikan identifikasi dokumen berupa:
  - Proyek
  - Nama Dokumen
  - Penulis
  - Tanggal Pembuatan
  - Jenis Dokumen
  - Manajemen Konfigurasi dan Informasi QA

- Pembaca yang Dituju
- Tingkat Kerahasiaan Dokumen
- Informasi untuk Pencarian Dokumen
- Informasi Hak Cipta
- Jika dokumen cukup panjang, bab, bagian, dan subbagian harus dibagi, dan daftar isi harus disertakan untuk menunjukkan letak setiap bab, bagian, dan subbagian. Format bab-halaman dapat digunakan untuk penomoran halaman, di mana penomoran selalu diatur ulang di awal setiap bab. Ini akan membuat segalanya lebih sederhana sehingga, jika terjadi perubahan, yang diperlukan hanyalah mencetak bab yang relevan, membiarkan halaman yang tersisa tidak terpengaruh.
- Sebuah dokumen harus memiliki indeks jika mengandung banyak informasi sehingga pembaca dapat lebih cepat dan efektif menemukan detail yang mereka butuhkan tentang suatu istilah.
- Sebuah glosarium diperlukan untuk menjelaskan kata-kata teknis yang digunakan dalam dokumen jika dimaksudkan untuk pembaca yang lebih besar dengan kemampuan untuk memahami berbagai subjek teknis.

## 2. Standar Dokumen

Standar berfungsi sebagai dasar untuk menentukan tingkat kualitas dokumen dengan memastikan konsistensi dalam tampilan, struktur, dan kualitas dokumen. Ada berbagai standar lain yang dapat diperhitungkan dalam pengembangan dokumen selain standar IEEE tersebut di atas:

### a. Standar Proses

Standar ini menguraikan proses yang harus diikuti saat membuat dokumen. Ini umumnya memerlukan pengembangan alat dan proses kompilasi dokumen QA untuk menghasilkan dokumen berkualitas tinggi. Teknik QA ini perlu diadaptasi agar dapat digunakan dengan berbagai jenis dokumen, terutama kertas formal. Jenis dokumen dan audiens target memengaruhi tingkat kualitas dokumen.

### b. Standar Produk

Seperti yang dikatakan sebelumnya, dokumen terkait harus memiliki struktur yang konsisten serta tampilan yang konsisten. Terlepas dari kenyataan bahwa standar produk dokumen bervariasi tergantung pada setiap proyek, beberapa aspek cenderung mematuhi pedoman tertentu, seperti:

- Standar Identifikasi Dokumen

Kita mengetahui banyak dokumen yang menggambarkan setiap aktivitas dalam fase ini berkat fase sebelumnya. Setiap dokumen dalam sebuah proyek harus memiliki identitas yang berbeda untuk membuat organisasi lebih mudah mengingat banyaknya kertas yang terlibat. Tentukan standar untuk penomoran dokumen-dokumen tersebut.

- Standar Struktur Dokumen

Standar untuk setiap jenis dokumen harus ditetapkan, seperti yang telah dijelaskan sebelumnya, agar lebih disederhanakan. Ketentuan untuk penomoran halaman, gaya penomoran bab, bagian, dan subbagian ditentukan oleh standar ini.

- Standar Presentasi Dokumen

Standar ini sebagian besar digunakan untuk menentukan tampilan kertas, termasuk pedoman penggunaan surat, penggunaan logo dan nama perusahaan, ukuran kertas, dan penjiilidan.

- Standar Pembaruan Dokumen

Jika ada perubahan pada sistem, tentu dokumen terkait akan ikut berubah dan harus diperbarui. Setiap modifikasi ini harus dapat diakses dengan jelas oleh pengguna dokumen. Gunakan nomor versi dokumen, tanggal diubah, dan nama yang melakukan pembaruan.

### c. Standar Interchange

Saat ini, komunikasi secara digital sudah menjadi hal yang lumrah, dan ini termasuk pengiriman kertas sehingga setiap pemangku kepentingan dapat menerimanya secara digital daripada dalam bentuk cetak. Untuk membuatnya lebih ramping, perlu ada kesepakatan standar untuk itu. Teknologi yang digunakan untuk pembuatan dan distribusi dokumen ditentukan oleh standar pertukaran atau pertukaran dokumen. Format dokumen Adobe Portable Document Format (PDF) yang banyak digunakan saat ini dapat digunakan jika dokumen bersifat satu arah, artinya berisi informasi dari satu pihak dan tidak mengantisipasi modifikasi dari pihak lain. Jenis dokumen ini termasuk, misalnya, manual yang diunduh dengan software. Gunakan format dokumen umum untuk pengeditan, seperti Microsoft Word atau Open Office Document, jika dokumen yang dirilis dimaksudkan untuk meminta umpan balik dari pihak lain yang menerimanya.

## 3. Gaya Penulisan

Gaya penulisan yang tepat diperlukan untuk teks yang layak dan mudah dipahami. Menulis dokumen membutuhkan beberapa langkah dan tidak mudah. Baca kembali dokumen setelah ditulis, lalu buat perubahan yang diperlukan. Mintalah bantuan kepada pembaca yang tidak memiliki latar belakang teknis dan minta umpan balik mereka tentang seberapa baik penjelasan dokumen yang ditulis dengan baik dan mudah dipahami. Memeriksa ulang dokumen yang sudah selesai adalah ide yang bagus untuk memastikan tidak ada kesalahan atau kelalaian. Manfaatkan alat dalam software pengolah kata untuk memverifikasi tata bahasa dan bahasa yang digunakan, meskipun Anda masih perlu melakukan koreksi yang diperlukan secara manual..

## 4. Dokumentasi On-line

Dokumen online biasanya digunakan saat software pengiriman berisi konten digital. File readme adalah bentuk paling dasar, dan biasanya menawarkan detail singkat tentang produk, seperti prasyarat yang harus dipenuhi sebelum penginstalan atau bahkan informasi terbaru yang tidak diberi cukup waktu untuk disertakan dalam buku pegangan pengguna. Dokumentasi online juga tersedia dalam bentuk file html yang dapat diakses di web browser dan berisi hyperlink ke referensi terkait. Pendistribusian dokumen akan lebih efektif dan praktis jika sudah berbentuk. Dokumentasi online memang memiliki

beberapa keterbatasan, sehingga tidak dapat sepenuhnya menggantikan dokumentasi tradisional berbasis kertas. Beberapa kekurangan ini antara lain:

- Tidak seperti buku, di mana orang dapat dengan cepat membolak-balik dan terkadang menemukan materi segar yang ingin mereka ketahui, tidak mudah untuk hanya melihat sekilas. Dokumen online harus dilihat dengan bantuan teknologi karena sangat terstruktur. Meskipun memiliki kemampuan pencarian, hasilnya seringkali tidak sesuai dengan kebutuhan pengguna.
- Karena pengaruh resolusi layar, kontras, dan pencahayaan, membaca di monitor komputer tidak selalu mudah dan kadang-kadang dapat membebani mata.
- Karena banyaknya tautan yang menghubungkan satu referensi dengan referensi lainnya, orang sering "tersesat" dalam dokumen online.

Oleh karena itu, dokumentasi on-line lebih baik hanya menjadi pendamping dari dokumentasi konvensional (Tantra, 2012).

#### **6.5.4 Standard Operating Procedure (SOP)**

SOP mengacu pada seperangkat instruksi kerja tertulis yang terstandarisasi (terdokumentasi) yang menjelaskan bagaimana menjalankan bisnis, di mana harus dilakukan, kapan harus dilakukan, dan siapa yang harus melakukannya. Kuncinya adalah SOP mengendalikan proses kerja, termasuk siapa yang harus melakukannya, siapa yang bertanggung jawab, siapa yang memberikan persetujuan, kapan harus diselesaikan, catatan apa yang harus dibuat, dan data pendukung lainnya. SOP pada hakekatnya adalah teknik manajemen untuk menetapkan konsistensi dalam praktik perusahaan, konsistensi dalam praktik kerja, dan konsistensi dalam kualitas proses atau produk yang akan datang (P., 2016).

SOP menjelaskan bagaimana tugas diselesaikan untuk mendukung kualitas data dan memastikan konsistensi teknis dan kualitatif di seluruh sistem. SOP dibuat khusus untuk organisasi, bisnis, atau komponen penyusunnya dengan tujuan membantu mereka dalam mengatur dan menegakkan kepastian standar mutu selama pelaksanaan tugasnya. Jika SOP tidak ditulis dengan benar, mereka akan menghambat pelaksanaan tugas, tetapi SOP yang ditulis dengan baik juga tidak efektif jika tidak diikuti secara konsisten. Dokumen SOP harus mudah diakses sehingga dapat dikonsultasikan dengan cepat bila diperlukan. Manajer Sistem Informasi biasanya adalah orang yang bertanggung jawab dalam perusahaan yang secara konsisten memantau penggunaan SOP. Karena bidang akuntansi biasanya memiliki hubungan langsung dengan konsistensi sistem organisasi atau perusahaan, seorang manajer akuntansi dapat dipekerjakan jika belum ada. SOP adalah salah satu area yang ditangani oleh Chief Information Officer (CIO) di organisasi atau bisnis besar.

SOP juga dapat dicakup dalam pelatihan yang ditawarkan kepada anggota staf oleh perusahaan. Masih akan ada pekerjaan rutin yang dilakukan secara manual, seperti mengontrol pengeluaran barang dari gudang, membuat inventaris, atau mencatat temuan survei lapangan, meskipun sistem informasi perusahaan telah didukung software yang

terintegrasi. SOP harus memberikan instruksi yang jelas kepada pelaksana sebagai aturan utama.

SOP harus ditulis secara ringkas, metodelis, dan dengan gaya yang dapat diakses. Gunakan standar yang sama yang telah disebutkan sebelumnya untuk persiapan dokumentasi. Konten dalam SOP tidak boleh kabur atau sulit dipahami, juga tidak boleh terlalu rumit. Hindari menggunakan "Anda" dan berbicara dengan suara aktif. SOP sederhana harus dibuat, dan jika perlu, bagan alur dan visual harus ditambahkan. Meskipun SOP tersebut sederhana, namun tetap harus dijelaskan secara mendalam agar seseorang yang hanya memiliki pengetahuan dasar dan pengalaman minimal dapat mengikuti langkah-langkah yang digariskannya secara akurat.

Orang yang menulis dan menyusun SOP harus memiliki pemahaman menyeluruh tentang prosedur yang perlu didokumentasikan, termasuk semua kemungkinan pengecualian dan penyimpangan. Oleh karena itu, hasil tulisan harus dievaluasi dan diperiksa sekali lagi oleh pihak yang berbeda untuk memastikan tidak ada langkah yang terlewatkan dalam proses. Setiap langkah proses harus jelas, mudah dipahami dan diterapkan, menutup segala potensi penyimpangan, dan memenuhi standar kualitas yang dipersyaratkan. Manajemen organisasi atau bisnis yang menerapkan SOP juga harus memberikan persetujuannya setelah audit dan penyesuaian yang diperlukan. SOP hanya dapat dilaksanakan secara penuh dengan persetujuan manajemen. SOP harus selalu direview setelah diimplementasikan agar dapat menyesuaikan dengan perubahan yang terjadi di dalam organisasi atau bisnis, karena sistem selalu dinamis dan berubah seiring dengan pertumbuhan organisasi atau bisnis. Pemeriksaan ini dapat dilakukan setiap satu hingga dua tahun sekali. Meski begitu, perbarui SOP tanpa harus menunggu pemeriksaan rutin jika ada modifikasi kritis yang harus segera dilakukan.

SOP dapat menjadi pedoman dalam melakukan audit, khususnya audit sistem, selain sebagai pedoman dalam melaksanakan tugas sehari-hari dalam organisasi. Selain itu, audit mungkin mengungkapkan apakah SOP perlu diubah atau tidak (Tantra, 2012).

#### **6.5.5 Disaster Recovery Plan (DSP)**

Merupakan bagian perencanaan dari sebuah institusi untuk melakukan tahapan tertentu yang nantinya akan menjamin kelangsungan pelayanan (khususnya dari segi sistem informasi) yang diberikan tanpa mengurangi kapabilitas serta kinerja dari sebuah sistem jika terjadi sebuah bencana didalamnya (Haryadi et al., 2019). DSP merupakan bagian dari Business Continuity Plan (BCP), yaitu suatu rencana dalam skala organisasi atau perusahaan untuk memulihkan kelangsungan operasional atau bisnis setelah terjadi peristiwa yang menyebabkan organisasi atau perusahaan tidak dapat beroperasi secara normal, misalnya bencana. Sebenarnya BCP secara lengkap terdiri atas:

- Business Resumption Plan
- Occupant Emergency Plan
- Incident Management Plan
- Continuity of Operation Plan
- Disaster Recovery Plan

Dari seluruh rencana di atas, hanya Incident Management Plan (IMP) yang berkaitan dengan teknologi informasi khususnya infrastruktur bila terjadi serangan cyber, tetapi tidak sampai mengaktifkan DSP.

Kemampuan organisasi atau korporasi untuk menjalankan proses yang mendukung operasi perusahaan dapat terganggu sebagian atau seluruhnya oleh bencana yang terjadi secara tiba-tiba. DSP menetapkan pedoman, prosedur, dan disiplin untuk menjamin bahwa proses bisnis yang penting akan terus berjalan bahkan jika timbul masalah dengan infrastruktur atau sistem informasi yang mendasarinya. Berikut adalah beberapa hal penting yang perlu diingat saat membuat DSP:

- Siapkan tim penyusun rencana.
- Lakukan penilaian dan audit risiko. Tentukan prioritas aplikasi dan jaringan.
- Susun strategi pemulihan.
- Siapkan inventaris dan dokumentasi rencana.
- Susun kriteria dan prosedur.
- Implementasikan rencana.

Anggota tim inti dari unit bisnis yang berkepentingan harus dimasukkan dalam tim penyusun DRP. Untuk membuat DRP, tim ini perlu memahami prosedur bisnis, teknologi, jaringan, dan sistem informasi. Sebuah tim harus melakukan analisis risiko dan penilaian efek bisnis untuk setidaknya 10 kemungkinan bencana. Tetapkan peringkat prioritas untuk setiap proses dan sistem bisnis setelah mempertimbangkan potensi risiko. Pertahankan inventaris untuk rencana tersebut dan buat daftar lengkap alat, lokasi, pemasok, dan kontak yang diperlukan. Tujuannya adalah untuk menawarkan pemulihan yang realistis, efisien, dan terjangkau untuk semua bidang teknologi terkait.

Tujuan utama DRP adalah membuat sistem yang mendukung proses bisnis kritis dan vital kembali online. Tujuannya adalah agar perusahaan atau organisasi dapat melanjutkan aktivitas bisnis reguler secepat mungkin. DRP harus berkonsentrasi pada perencanaan berbasis teknologi karena sebagian besar operasi bisnis yang signifikan dan sangat signifikan bergantung pada ketersediaan teknologi dan sistem informasi, jaringan, dan data. Ini menyiratkan bahwa DRP harus dibuat untuk setiap area organisasi atau bisnis dengan sistem berbasis komputer. Namun, DRP tidak selalu mengharuskan untuk melampaui. Rencanakan saja strategi implementasi untuk sistem yang masih bisa diimplementasikan secara manual sambil menunggu pemulihan, jangan menyiapkan apa pun untuk sistem yang vital atau sangat penting. Dengan demikian, biaya untuk inventaris DRP dapat lebih efisien (Tantra, 2012).

## **BAB VII**

### **FASE DELIVERY**

Tahap-tahap pelaksanaan dalam fase delivery terbagi atas tiga bagian utama, yaitu deployment, pelatihan, dan produksi. Kegiatan yang dilakukan pada fase delivery ini bisa dibagi atas 4 tahapan secara berurut. Setiap tahap dijelaskan di bawah ini.

#### **7.1 Deployment**

Pada tahapan deployment ini adalah tugas yang mencoba merilis aplikasi yang telah dibuat oleh pengembang. Bergantung pada jenis aplikasinya, penerapan dapat dilakukan dalam berbagai metode (Haryana, 2019).

##### **7.1.1 Instalasi**

Jika pengguna melakukan instalasi, prosedurnya harus dirancang agar lebih sederhana bagi pengguna. Basis data, software tambahan, plug-in, software aplikasi server, driver, dan pengaturan lainnya juga termasuk dalam instalasi sistem informasi selain software itu sendiri. Biasanya, penginstalan ini harus diselesaikan dalam urutan tertentu, seperti menginstal server database terlebih dahulu, diikuti dengan driver untuk sumber data (seperti ODBC), dan terakhir menghubungkan aplikasi server ke database. Setelah itu, aplikasi client dengan server. Untuk mempermudah semua urutan ini, Anda dapat membuat paket aplikasi instalasi yang menjalankan semua langkah secara otomatis, sehingga pengguna cukup melakukan klik atau memasukkan informasi lokasi instalasi. Untuk deliverables yang berupa aplikasi berbasis web, selain menginstal di server aplikasi, mungkin perlu menambahkan add-on, ekstensi, atau plug-in ke browser, seperti .NET framework, platform Java, FTP, atau Flash, agar program dapat berjalan dengan baik. Jika instalasi tambahan tidak dapat dikemas dengan cara ini, pengguna dapat diberi petunjuk tentang cara mendapatkannya dari Anda. Biasanya, pemberitahuan yang menyatakan bahwa instalasi tambahan diperlukan akan muncul saat aplikasi diluncurkan. Yang perlu Anda lakukan untuk mempermudah pengguna adalah menyediakan URL tempat mereka dapat mengunduh add-on, ekstensi, atau plug-in. Cukup mengklik link URL akan membawa pengguna ke petunjuk instalasi.

Lingkungan di mana software akan diinstal juga harus diperhitungkan selain mempersiapkan prosedur instalasi. Lingkungan ini dapat mengacu pada hardware, jaringan, koneksi internet, dan elemen terkait infrastruktur lainnya. Anda harus membuat daftar hal-hal yang harus disiapkan sesuai dengan kriteria masing-masing sehingga Anda dapat melacak semua persiapan untuk lingkungan ini. Kru instalasi, termasuk klien, kemudian diberi akses ke daftar periksa ini, khususnya jika instalasi tersebar di beberapa lokasi dan klien bertanggung jawab untuk melakukan persiapan yang diperlukan. Item daftar periksa ini meliputi:

1. Spesifikasi dan konfigurasi hardware: Jenis prosesor, RAM minimum yang dibutuhkan, ruang harddisk yang harus ada, CD ROM dan sebagainya.
2. Sistem operasi: Untuk komputer klien dan server, versi yang diperlukan, paket layanan yang diperlukan, persyaratan untuk kerangka kerja .NET, atau persyaratan lainnya. Pastikan kebutuhan aplikasi server sudah terinstal terlebih dahulu dan berfungsi jika software yang akan diinstal berbasis web.
3. Konfigurasi jaringan: Jika sistem perlu menjadi bagian dari jaringan, pastikan sudah diatur dengan benar dan sistem secara keseluruhan dapat berkomunikasi. Pastikan sistem dapat berkomunikasi dan bertukar data jika jaringan harus terhubung ke jaringan lain, seperti melalui Wide Area Network (WAN), Virtual Private Network (VPN), atau koneksi internet biasa.
4. Driver: Periksa untuk melihat apakah driver yang diperlukan telah diinstal untuk periferal atau perangkat tambahan apa pun agar dapat berfungsi dengan baik.
5. Keamanan sistem: Verifikasi keamanan jaringan terhadap akses luar yang tidak sah jika sistem terhubung ke internet atau jaringan lain melalui WAN. Selain itu, konfirmasikan bahwa antivirus telah diatur dan diperbarui.
6. Database: setiap sistem informasi bisnis pasti memerlukan database, baik database yang bersifat local maupun server. Periksa versi database yang dibutuhkan. Untuk mengaktifkan software agar terhubung ke database, instal driver yang diperlukan untuk membuat sumber data. Periksa untuk melihat apakah protokol tertentu yang digunakan oleh software untuk terhubung ke database kompatibel dengan sistem operasi yang digunakan.
7. Shared resource/library seperti file-file dynamic link libraries (DLL): beberapa sudah disertakan dengan sistem operasi, tetapi yang lain mungkin diperlukan untuk kompiler atau runtime yang diperlukan agar software berfungsi dengan sempurna. Jika Anda kesulitan menemukan file DLL atau OCX (yang menyimpan komponen ActiveX), coba cari secara online.
8. Dokumentasi: semua dokumen yang diperlukan oleh pengguna dan telah disiapkan dalam fase pengembangan harus sudah disiapkan untuk bisa digunakan setiap saat selama fase delivery ini (Tantra, 2012).

Selain persiapan di atas yang diasumsikan sebagai instalasi baru, Anda mungkin juga berhadapan dengan instalasi yang bersifat upgrade atau peningkatan software dari versi lama ke versi yang terbaru. Untuk keadaan ini, Anda harus menentukan apakah software sebelumnya perlu di-uninstall terlebih dahulu atau penginstalan dapat diselesaikan segera. Demikian pula, penting untuk mengonfirmasi bahwa versi software saat ini dan pengaturan yang diperlukan oleh versi sebelumnya kompatibel. Terkadang, agar software baru berfungsi dengan baik di lingkungan operasi, konfigurasi atau persyaratan software yang ada masih harus dipenuhi. Semua ini harus benar-benar diperiksa dengan teliti agar tidak terjadi konflik dengan software baru atau upgrade yang akan diinstalasikan.

Semua penginstalan software harus didokumentasikan setelah proses penginstalan selesai, terutama jika software yang diinstal memerlukan lisensi. Sekalipun software

dibuat oleh tim internal organisasi atau korporasi, kebijakan kontrol lisensi sangat penting untuk mencegah penyalahgunaan software. Anda akan dapat melacak penginstalan software dengan kontrol lisensi, dan tidak akan ada kebocoran penginstalan software di luar perusahaan Anda atau klien Anda. Selain itu, kontrol lisensi mempermudah Anda untuk melakukan tugas pemeliharaan seperti memberikan pembaruan atau tambalan penginstalan software. Selain menjadi pengontrol inventaris untuk software yang diinstal, pencatatan lisensi juga dapat mencakup sistem operasi yang diinstal dan semua software lainnya. Software adalah aset bisnis, terutama jika dibuat berdasarkan kontrak dengan nilai tinggi.

Berbeda dengan deployment secara tradisional, dalam cloud computing, software delivery lebih berupa layanan (service). Opsi baru untuk memberi orang akses ke software disediakan oleh cloud computing. Segala sesuatu dalam penerapan penyebaran ini, termasuk penginstalan dan pengaturan lainnya, terjadi di server. Proses penginstalan dan penyiapan disembunyikan dari pengguna. Sebuah paradigma baru dalam komputasi terdistribusi, cloud computing menghadirkan berbagai ide, konsep, teknologi, dan jenis arsitektur dengan cara berorientasi layanan. Pengguna tidak perlu terhubung langsung ke hardware yang menampung sumber daya ini untuk mengaksesnya melalui koneksi jaringan Internet yang cepat. Karena persyaratan hardware dan software yang lebih sedikit karena proses komputasi berada di server yang jauh, biaya dan pemeliharaan proses dapat dikurangi (Kurniawan, 2015). Lokasi ini mungkin berada di host di pusat data atau host di lokasi penyedia layanan (application service provider). Selain program, data dapat disimpan di cloud. Selama ada browser dan akses internet, individu dapat bekerja di mana pun mereka berada di perangkat apa pun. Sebagai ilustrasi, pertimbangkan Google Docs, yang menawarkan alat spreadsheet, dokumen, dan presentasi bersama dengan penyimpanan data (<httpdocs.google.com>). Dokumen yang biasanya Anda kerjakan menggunakan paket software perkantoran dapat dibuat, diedit, dan disimpan langsung melalui browser di perangkat Anda, seperti PC atau tablet, dan Anda dapat melakukannya di mana pun ada sambungan internet (Muttaqin et al., 2022). Biasanya, software akan menerima pembaruan setelah penginstalan awal. Pembaruan biasanya digunakan untuk menambal masalah, memperbaiki kesalahan, dan menyelaraskan software dengan spesifikasi. Perubahan masih sering dilakukan bahkan setelah software digunakan secara operasional karena kerusakan baru ditemukan sebagai hasil dari laporan pengguna atau dari tim pengembangan software itu sendiri, atau perubahan untuk memenuhi kebutuhan tambahan dari pengguna. Ada berbagai pendekatan untuk melakukan pembaruan ini::

1. Menginstal ulang software diperlukan bila tersedia perbaikan atau pembaruan yang cukup sederhana karena harus dihapus instalannya terlebih dahulu sebelum pemutakhiran dapat diinstal. Proses ini biasanya berjalan cukup rumit sehingga memerlukan tindakan dari tim deployment.
2. Patching, yaitu memberikan sebuah file executable yang apabila dieksekusi akan menguraikan file-file yang terkandung di dalamnya yang akan menimpa ataupun menambah software lama dengan koreksi- koreksi yang telah dibuat.

3. Online update, yaitu update yang dilakukan dengan melakukan koneksi ke host tertentu kemudian software akan melakukan download update yang diperlukannya dan melakukan instalasi secara otomatis sendiri. Contoh update seperti ini dapat ditemukan pada software antivirus atau browser. Bahkan sistem update yang baik akan memberikan notification bahwa ada update baru yang siap untuk didownload.
4. Pembaruan otomatis adalah sistem pembaruan yang menggunakan software yang terhubung ke internet untuk memeriksa pembaruan terbaru. Jika ada, pembaruan akan secara otomatis diunduh dan diinstal oleh software juga.
5. Pembaruan software terpusat yang dikenal sebagai pembaruan tanpa pengawasan. Update bisa dilakukan secara push, artinya secara terpusat update akan dikirimkan ke setiap PC yang terinstalasi dengan software. Komputer di pusat yang mendapatkan update melakukan koneksi dengan komputer pengguna, mengirimkan update, dan melakukan instalasi.
6. Pembaruan server, atau perubahan software yang hanya dilakukan ke server, langsung terlihat oleh semua PC klien yang terhubung ke server. Aplikasi untuk cloud computing atau software berbasis web biasanya menerima pembaruan seperti ini..

Siapkan pemutakhiran software sesuai dengan persyaratan software dan karakteristik organisasi atau bisnis pengguna. Tahap awal dalam berkomunikasi dengan pengguna melalui kiriman adalah prosedur instalasi. Selain itu, setelah software diinstal, tahap selanjutnya adalah menyiapkannya untuk digunakan dalam operasi (Tantra, 2012).

### **7.1.2 Pengaturan Database**

Setiap software yang dibuat untuk tujuan perusahaan harus memasukkan database. Semua data transaksi yang terjadi selama operasi reguler disimpan oleh database sebagai fungsi utamanya. Setelah data terkumpul, software dapat menghasilkan laporan dan analisis berdasarkan kebutuhan bisnis atau organisasi. Basis data harus disusun sesuai dengan software yang dikembangkan agar mampu menyimpan data transaksi dan menyampaikan informasi dalam bentuk laporan (Mandar, 2017).

Proses pembuatan struktur database, yang meliputi pembuatan tabel, pemilihan indeks, pembuatan relasi antar tabel, dan penyimpanan stored procedure, dilakukan setelah proses instalasi database selesai. Agar beberapa solusi server basis data dapat beroperasi dengan sebaik-baiknya, diperlukan penyesuaian atau penyiapan. Jumlah memori yang dialokasikan, bandwidth yang diperlukan untuk komunikasi antara software dan database, menentukan jumlah koneksi konkuren, keamanan akses, pengaturan hak pengguna, dan pilihan lain di antara pengaturan ini. Profesional basis data, terkadang dikenal sebagai administrator basis data, terkadang diminta untuk melakukan penginstalan dan konfigurasi software berskala besar. Database administrator (DBA) adalah spesialis yang diperlukan untuk mengelola basis data dalam operasi sehari-hari. Tugas DBA meliputi pengelolaan dan pemeliharaan database, termasuk database multidimensi, akun pengguna, ketersediaan database, pemulihan, dan pelaporan. Seiring dengan merancang

dan mengimplementasikan keamanan dan otomatisasi server, dia juga melacak dan menyelesaikan masalah SQL Server (Susilawati & Gunawan, 2018).

Pembentukan struktur database ini bisa dibuat secara otomatis dari dalam software. Dalam pengembangannya, fungsi create database dan tabel- tabel ini dapat disertakan sebagai bagian dari setup software. Hal ini akan mempermudah proses instalasi database. Setelah instalasi dan konfigurasi dilakukan, pada beberapa sistem, diperlukan pengaturan secara manual untuk menghubungkan antara software dengan database. Misalnya, untuk membuat koneksi ini, Anda harus terlebih dahulu menginstal driver database atau menggunakan ODBC. Lakukan tes untuk memastikan software dan koneksi database dapat berfungsi dengan baik, terutama jika koneksi ini berada di jaringan yang sibuk dengan banyak pengguna dan terutama jika database berada di lokasi yang jauh dari software. Dalam situasi seperti ini, masalah infrastruktur harus dipastikan berfungsi dengan baik terlebih dahulu, agar tidak menjadi rintangan, sebelum menginstal dan mengkonfigurasi database (Tantra, 2012).

### **7.1.3 Migrasi dan Konversi Data**

Agar migrasi data dapat berlangsung dengan baik maka diperlukan langkah-langkah sebagai berikut (Tantra, 2012):

1. Eksplorasi database asal.

Dalam langkah pertama ini, lakukanlah eksplorasi terhadap database asal, dan identifikasikan data sesuai desain database baru. Database lama bisa saja terdiri dari ribuan kolom, tetapi mungkin saja ada beberapa kolom di antaranya yang terduplikasi atau juga tidak diperlu- kan dalam proses migrasi. Mengidentifikasi data yang diperlukan dan lokasinya pada saat ini, serta menentukan data mana yang redundan dan tidak dapat digunakan oleh sistem yang baru sangatlah penting.

2. Lakukan penilaian terhadap data.

Dalam tahap ini Anda harus melakukan penilaian terhadap data yang didapatkan dari sumber-sumber yang telah diidentifikasi pada tahap pertama tadi. Untuk melakukan penilaian dibutuhkan profiling data, yaitu proses sistematis dalam melakukan pengamatan dan analisis terhadap seluruh isi kolom dari tabel yang menjadi sumber data asal. Profiling mengidentifikasi data yang rusak atau cacat sampai ke level tabel dan kolom. Profiling juga merupakan proses yang menjadi bagian yang utuh dari proses evaluasi kesesuaian data dan memastikan requirements sistem baru terpenuhi. Profiling meliputi pemeriksaan terhadap nilai record yang sebenarnya dan informasi metadata dari data sumber. Banyak migrasi data yang dilakukan tanpa sebelumnya memeriksa tingkat kualitas dari sumber data yang menyebabkan tingginya risiko kegagalan migrasi database. Dengan melakukan profiling sejak awal akan menurunkan tingkat risiko pekerjaan yang berlebihan, penundaan, atau bahkan kegagalan proyek.

3. Desain proses migrasi.

Anda menentukan arsitektur teknis dan desain prosedur migrasi database pada langkah ini. Anda juga harus menentukan prosedur pengujian dan peralihan ke produksi. Tentukan

juga apakah transisi ini berlangsung secara paralel, zero downtime ataukah database lama dapat langsung di-non-aktifkan saat migrasi selesai dilakukan.

#### 4. Bangun proses migrasi.

Prosedur migrasi dibangun setelah tahap desain selesai agar dapat diimplementasikan sebagaimana dimaksud. Proses migrasi dapat dibuat dengan menggunakan skrip SQL, software migrasi yang dibuat secara khusus untuk melakukan migrasi sesuai dengan desain, atau dapat digunakan secara sederhana dengan memanfaatkan software migrasi yang ada dan penetapan aturannya.

#### 5. Pelaksanaan migrasi.

Setelah proses migrasi dibangun dan melalui tahap testing, maka dalam tahap ini adalah saatnya Anda melakukan migrasi yang sebenarnya dari data sumber menjadi database baru yang akan digunakan dalam sistem baru.

#### 6. Transisi.

Sistem dengan database baru tidak secara otomatis memulai saat proses migrasi selesai. Dalam jadwal proyek, biasanya mencakup waktu untuk berbagai persiapan serta transisi ke sistem baru berdasarkan WBS yang direncanakan. Oleh karena itu, fase transisi adalah waktu setelah migrasi selesai dan sebelum produksi. Sistem dapat berfungsi secara paralel dengan sistem sebelumnya untuk sementara selama masa transisi ini sesuai dengan desain sebelumnya. Mungkin juga fase transisi hanyalah tahap akhir dari perencanaan sebelum produksi dimulai.

#### 7. Produksi.

Sistem dengan database baru disiapkan untuk memasuki tahap produksi setelah waktu transisi selesai. Bagian berikut akan membahas lebih detail tentang tahap produksi ini.

### **7.1.4 Eksekusi Secara Terbatas**

Anda dapat langsung melakukan pelatihan dan memulai produksi pada proyek sistem informasi skala kecil dan menengah. Namun, ada kalanya sistem perlu digunakan di lingkungan yang terbatas, dengan sejumlah kecil pengguna yang telah ditentukan sebelumnya, atau hanya di beberapa fungsi tertentu dari keseluruhan sistem yang ada..

Eksekusi terbatas harus disusun dahulu sebelumnya pada fase perencanaan proyek. Eksekusi terbatas harus direncanakan terlebih dahulu pada tahap perencanaan proyek untuk menghindari keterlambatan tanggal penyelesaian proyek. Proses perencanaan ini harus termasuk memutuskan pengguna mana yang akan ditugaskan untuk menjalankan fungsi mana sebelum sistem secara keseluruhan. Eksekusi terbatas, yang seringkali berlangsung dalam waktu singkat, berupaya memperkenalkan pengguna dengan menggunakan sistem dan mengamati bagaimana responsnya terhadap tugas sehari-hari. Sebagai bagian dari evaluasi seluruh tahap penerapan, temuan pengamatan dalam pelaksanaan terbatas kemudian didokumentasikan dan dinilai. Berdasarkan pengalaman mereka menggunakan sistem terbatas, pengguna yang mengikuti pelaksanaan terbatas ini juga dapat dipekerjakan sebagai instruktur untuk pelatihan penggunaan sistem informasi (Tantra, 2012).

### **7.1.5 Eksekusi Secara Paralel**

Anda dapat menambahkan eksekusi secara paralel dengan sistem lama selain eksekusi terbatas. Bersama dengan sistem lama yang masih digunakan, sistem baru dijalankan secara terbatas. Tujuannya adalah untuk membandingkan penggunaan sistem baru dan data yang dihasilkannya. Hasil dari eksekusi paralel tidak akan mempengaruhi fase implementasi proyek berikutnya, tetapi masih didokumentasikan.

Seiring dengan eksekusi yang terbatas, waktu eksekusi juga perlu dibatasi untuk mencegah terjebak dalam fase produksi aktual sistem informasi (Tantra, 2012).

### **7.1.6 Evaluasi**

Periksa keseluruhan rangkaian dan pastikan semua aktivitas dicatat pada akhir fase penerapan. Pertimbangkan untuk menggabungkan tindakan tersebut di akhir dan fase pemeliharaan atau di proyek Anda berikutnya jika ada tugas fase penerapan yang tidak dapat Anda lakukan secara efektif. (Tantra, 2012).

## **7.2 Pelatihan**

Selain tidak terlalu panjang atau mengharuskan peserta menyerap terlalu banyak informasi, pengaturan pelatihan harus mempertimbangkan efektivitas hasil pelatihan dan efisiensi penjadwalan untuk memastikan bahwa ketika dilaksanakan akan memberikan informasi yang cukup kepada peserta pelatihan. Pelatihan tidak hanya harus mencakup teori tetapi juga melibatkan penggunaan aktual dari sistem informasi. Teori tersebut dapat dijelaskan dengan memberikan presentasi yang menawan. Sertakan data terkait pelatihan yang merupakan representasi dari situasi aktual.

Harap diingat bahwa pelatihan yang harus dibahas saat ini, bukan sistem informasi itu sendiri. Oleh karena itu, Anda harus tegas dan melarang peserta mengungkapkan bagaimana sistem seharusnya bekerja menurut pendapat mereka. Karena sudah ada sesi-sesi untuk membahas dan memperdebatkan sistem informasi, termasuk saat pengujian dilakukan, pada fase-fase sebelumnya, Anda telah menerima berbagai masukan. Pengguna tidak memberikan umpan balik atau terlibat dalam diskusi selama waktu pelatihan; sebaliknya, mereka belajar bagaimana menggunakan sistem informasi.

Peserta pelatihan dipisahkan menjadi beberapa kelompok sesuai tanggung jawabnya dalam menjalankan sistem informasi. Pelatihan akan dibagi ke dalam kategori berikut berdasarkan fungsi umum peserta (Tantra, 2012).

### **7.2.1 Pelatihan Pengguna Akhir (End User Training)**

Pengguna akhir adalah orang-orang yang akan menggunakan sistem yang dibangun (Hall, 2007). Pelatihan harus difokuskan dalam penggunaan praktis. Pada akhir pelatihan

diharapkan pengguna akhir sudah menguasai pengoperasian untuk pekerjaan sehari-hari, termasuk penggunaan berbagai tombol, pemeriksaan data, dan pelaporannya.

### **7.2.2 Pelatihan Administrator**

Administrator bertanggung jawab untuk mengatur sistem, memelihara database, dan seringkali menjadi orang yang paling mengetahui sistem sehingga mereka dapat membantu pengguna lain. Karena lebih ekstensif dan terkadang lebih teknis daripada pengguna lain, pelatihan untuk administrator harus diberikan dalam pengaturan terpisah. Oleh karena itu, administrator harus sudah mengetahui dan berpengalaman dengan sistem.

Administrator harus dapat mengonfigurasi sistem, termasuk pengaturan pengguna, keamanan, dan konfigurasi, serta mengelola database, termasuk mencadangkan, memelihara, memulihkan, dan memberikan akses. Selain itu, keterampilan administrator seperti penginstalan dan pemecahan masalah dapat ditambahkan..

### **7.2.3 Pelatihan Eksekutif (Executive Training)**

Eksekutif tidak memerlukan pelatihan sistem mendalam seperti yang dilakukan pengguna akhir. Mereka membutuhkan metode untuk mendapatkan data yang dihasilkan sistem. Mereka juga harus mendapatkan pelatihan dalam menggunakan alat kecerdasan bisnis jika sistem memilikinya.

Eksekutif biasanya terbiasa melihat berbagai laporan untuk membantu mereka membuat keputusan. Akibatnya, mereka juga dapat mendeteksi perbedaan apa pun dalam informasi yang mereka dapatkan dengan segera. Apakah anomali tersebut disebabkan oleh masalah sistem atau kesalahan data, kadang-kadang dapat digunakan sebagai bahan evaluasi..

Pelatihan eksekutif tidak dapat dijadwalkan dengan cara yang sama seperti pelatihan untuk pengguna biasa karena selain kendala waktu, materi pelatihan juga berbeda. Bahkan setelah proyek selesai, pelatihan eksekutif masih dapat diatur karena mungkin tidak menjadi prioritas utama selama fase pelatihan. Meski begitu, Anda harus menjadwalkannya agar tidak menghalangi penyelesaian pekerjaan.

## **7.3 Manajemen Perubahan**

Seluruh bagian organisasi atau perusahaan baik yang terlibat langsung maupun tidak langsung harus sudah siap sebelum sistem tersebut memasuki tahap produksi yaitu pada saat sistem tersebut benar-benar digunakan untuk operasional sehari-hari dan telah menggantikan sistem yang lama secara keseluruhan. Ini termasuk administrasi, prosedur kerja, akses informasi, dan bahkan struktur organisasi. Dampak sistem informasi mungkin bersifat lokal, hanya memengaruhi sebagian kecil organisasi atau bisnis, atau global. Keduanya pasti membutuhkan perencanaan yang diatur oleh manajemen perubahan organisasi.

Manajemen perubahan adalah alat, proses dan juga teknik untuk mengelola seluruh akibat yang dihasilkan karena adanya perubahan dalam sebuah organisasi. Tingkat keberhasilan suatu upaya manajemen perubahan juga dapat dipengaruhi oleh perilaku individu atau organisasi. Mempertahankan kelangsungan hidup jangka pendek dan jangka panjang perusahaan adalah tujuan dari manajemen perubahan di dalam organisasi. Bisnis dapat menyesuaikan diri dengan perubahan lingkungan internal (seperti budaya tenaga kerja WFH (Work From Home), perubahan taktik pemasaran, perubahan teknologi dan peralatan, dll) serta perubahan lingkungan eksternal (seperti perubahan di pasar, perubahan perilaku masyarakat (“new normal”), perubahan peraturan, undang-undang, kebijakan pemerintah, jaringan internet, dan lain-lain). Meningkatkan kemandirian perusahaan adalah tujuan utama manajemen perubahan agar lebih produktif dan kompetitif di pasar ekonomi digital. Inisiatif ini melibatkan peningkatan sistem dan struktur organisasi, peningkatan efisiensi tenaga kerja, dan penerapan strategi perusahaan (A. Wibowo, 2021).

#### **7.4 Produksi**

Tahap akhir dari fase pengiriman, yang dikenal sebagai produksi, melihat sistem digunakan sepenuhnya untuk tugas-tugas rutin sesuai dengan tujuan awal desainnya. Meskipun demikian, sistem tersebut masih membutuhkan bantuan, observasi, review, dan evaluasi. Anda harus merencanakan berapa lama fase mentoring ini akan berlangsung dalam keseluruhan jadwal proyek sehingga semua pemangku kepentingan klien siap untuk mengambil alih seluruh pengelolaan sistem secara mandiri.

Berikut adalah beberapa hal yang dapat menjadi saran untuk membantu Anda memastikan bahwa tahap produksi ini berjalan dengan lancar sehingga Anda siap menyerahkan sistem informasi kepada klien pada tahap berikut:

1. Siapkan checklist untuk hal-hal yang harus Anda perhatikan dalam memasuki tahap produksi. Contoh checklist ini dapat Anda lihat pada Appendix 11.
2. Pastikan bahwa seluruh pengguna telah memakai versi terakhir dari software dan dokumentasinya, dan mereka telah memahami versi tersebut.
3. Pastikan bahwa semua sistem tambahan, termasuk tindakan keamanan untuk jaringan dan database serta sistem cadangan, beroperasi penuh.
4. Untuk memastikan semuanya berjalan lancar, awasi setidaknya dua siklus transaksi hingga selesai.
5. Selalu siapkan strategi mitigasi masalah. Meskipun Anda menggunakan manajemen perubahan selama masa transisi, mungkin masih ada detail tertentu yang hilang dalam keadaan tertentu. Persiapkan mitigasi yang sesuai agar jika terjadi masalah, sistem informasi tetap dapat beroperasi, meskipun dengan batasan tertentu, dan dapat dipulihkan dengan cukup cepat.
6. Pastikan selalu ada akses dukungan teknis dalam bentuk mentoring atau help desk. Mungkin akan ada permintaan untuk membantu operasi sistem debug selama tahap awal produksi.

7. Komunikasikan masa produksi ini kepada semua pemangku kepentingan sehingga semua orang memahami bahwa sistem telah beroperasi penuh dan proyek sedang dalam tahap akhir persiapan untuk serah terima (Tantra, 2012).

## **BAB VIII**

### **FASE AKHIR DAN PEMELIHARAAN SISTEM INFORMASI**

#### **8.1 Penutupan Proyek**

Fase penutupan proyek ini terdiri dari 7 langkah. Masing-masing akan dibahas secara berurutan berikut ini.

##### **8.1.1 Verifikasi Lingkup Pekerjaan**

Audit harus dilakukan untuk memverifikasi bahwa semua ruang lingkup pekerjaan telah diselesaikan sesuai dengan requirements sebelum proyek secara resmi berakhir. Tindakan berikut harus diambil untuk melakukan audit ini:

1. Periksa apakah persyaratan sistem informasi telah terpenuhi. Setelah pengujian tim internal dan pengujian pengguna, umpan balik dari pengguna selama tahap produksi juga dapat digunakan sebagai informasi untuk verifikasi.
2. Verifikasi setiap dokumen tambahan. Banyak dokumen dihasilkan selama proyek berlangsung, tetapi instruksi penggunaan sistem yang dapat membantu pengguna menggunakan sistem informasi dalam pekerjaan sehari-hari mereka adalah dokumen pendukung yang paling penting untuk divalidasi. Seperti yang dikatakan di bab sebelumnya, dokumen ini harus sesuai dengan produk akhir pengguna.
3. Verifikasi apakah seluruh deliverables telah tersedia dan klien telah menerimanya. Audit harus dilakukan dengan pemeriksaan seluruh deliverables secara fisik. Hal ini biasanya disebut audit konfigurasi fisik. Sebagai tanda bahwa deliverables telah diterima oleh klien maka klien harus menandatangani dokumen penerimaan (delivery acceptance form).
4. Mengevaluasi tingkat kepuasan klien. Ketika sebuah proyek memenuhi kriteria kepuasan klien, itu dapat dikatakan selesai. Landasan penilaian ini adalah persyaratan agar lebih berimbang. Sedangkan penilaian dapat dilakukan melalui wawancara atau kuesioner. Jika unsur penilaian ini ditemukan yang tidak memuaskan klien, mereka dapat digunakan sebagai masukan dalam proyek pengembangan sistem di masa mendatang.

Setelah menyelesaikan langkah-langkah tersebut di atas, Anda harus mengoordinasikan tim proyek untuk melakukan persiapan berikut untuk penutupan proyek:

1. Rapat pembahasan akhir proyek.

Mengadakan rapat diskusi akhir dengan seluruh tim, sponsor, dan beberapa pemangku kepentingan yang berkepentingan langsung untuk mengambil keputusan tentang proyek yang telah dilaksanakan berdasarkan hasil audit. Diskusikan semua temuan verifikasi dan penilaian secara objektif sehingga dapat menjadi dasar untuk penentuan keadaan proyek.

2. Tentukan proyek sukses atau gagal.

Banyak faktor yang mempengaruhi apakah suatu proyek berhasil atau tidak. Bahkan ketika proyek telah selesai dan klien telah menerima semua deliverables, masih dapat

dianggap gagal jika memakan waktu lebih lama dari yang diharapkan atau biaya lebih dari perkiraan semula. Sponsor proyek, eksekutif, dan manajer fungsional organisasi atau perusahaan klien bersama-sama menyepakati kriteria keberhasilan, yang menjadi dasar keputusan untuk menyatakan proyek berhasil. Selain berfungsi sebagai standar tujuan manajer proyek, kriteria ini juga harus ditetapkan di awal proyek untuk mencegah kriteria disusun secara subyektif berdasarkan hasil akhir proyek yang diketahui. Tim proyek selanjutnya harus melakukan evaluasi sesuai dengan kriteria ini sebelum menyerahkan hasil evaluasi kepada sponsor proyek untuk persetujuan formal keberhasilan atau kegagalan proyek.

### 3. Lakukan kajian terhadap kinerja.

Setelah proyek dilaksanakan, penting untuk menilai kinerja tim proyek secara keseluruhan serta manajer proyek, baik secara individu maupun kolektif. Supervisor Anda serta sponsor proyek akan mengevaluasi manajer proyek. Anda sendiri juga memberikan evaluasi kepada anggota tim proyek Anda. Anda mungkin mengevaluasi tim desain terlebih dahulu, diikuti oleh tim pengembangan dan tim teknis lainnya. Kemudian, nilai setiap anggota tim Anda secara terpisah. Evaluasi kinerja ini harus diselesaikan, terutama jika proyek dianggap gagal. Temuan evaluasi kemudian dapat digunakan sebagai panduan dan instruksi untuk lebih meningkatkan kinerja proyek berikutnya (Tantra, 2012).

## 8.1.2 Penutupan Kontrak

Siklus hidup proyek harus diakhiri dengan penutupan proyek, yang harus diformalkan untuk memastikan bahwa klien menerima semua keuntungan bisnis proyek. Laporan penutupan proyek ditulis dan diserahkan untuk disetujui oleh klien dan/atau sponsor proyek. Manajer proyek bertanggung jawab untuk melaksanakan tugas-tugas yang dinyatakan dalam laporan penutupan proyek, dan hanya setelah semua tugas tersebut selesai barulah proyek dinyatakan selesai (Sitanggang et al., 2019).

Kontrak penutupan yang menyatakan bahwa semua pekerjaan telah selesai sesuai dengan kontrak diperlukan untuk menutup proyek secara resmi. Baik pemilik proyek maupun pelaksana proyek yang merupakan pihak pertama yang menandatangani kontrak terlibat dalam penutupan ini. Dokumen akhir proyek harus ditandatangani oleh pihak yang bertanggung jawab atas nama manajer proyek dan pengguna sistem informasi jika kedua belah pihak berasal dari perusahaan atau organisasi yang sama. Dokumen tersebut harus ditandatangani oleh pemilik proyek, yang biasanya diwakili oleh pimpinan organisasi atau perusahaan, dan pelaksana proyek, yang biasanya diwakili oleh pimpinan organisasi yang bertanggung jawab untuk melaksanakan proyek, atau di paling tidak supervisor manajer proyek.

Sebelum menandatangani kontrak penutupan, pelaksana proyek harus memastikan bahwa semua masalah dan masalah yang mungkin masih ada pada audit sebelumnya telah diselesaikan atau setidaknya memiliki solusi yang dapat diterima bersama. Karena penutupan proyek berfungsi sebagai pernyataan formal bahwa klien telah menerima

semua hasil proyek, inilah alasannya. Selain itu, konfirmasikan bahwa semua kewajiban klien, seperti pembayaran, telah dipenuhi dengan benar.

Dokumen penutupan kontrak seringkali berbentuk laporan resmi yang mencantumkan pihak-pihak yang menandatangani kontrak, nomor dan subjeknya, nomor dokumen penerimaan pengiriman, serta tanggal penyelesaian dan penyelesaian proyek. Transisi ke fase pemeliharaan sistem, yang akan diakui oleh kontrak baru secara terpisah, juga dapat diumumkan setelah transfer hasil proyek, jika diperlukan (Tantra, 2012).

### **8.1.3 Penutupan Administratif**

Proses penutupan administrasi mencakup semua dokumentasi, interaksi, dan aktivitas tata kelola dan akuntabilitas terkait yang diperlukan untuk melakukan proses penutupan administrasi proyek (Yanti, 2009).

Selain itu, masalah kekayaan intelektual harus diselesaikan dengan klien. Misalnya, lisensi server basis data per server atau per pengguna yang terhubung, atau lisensi untuk menggunakan software pihak ketiga seperti alat intelijen bisnis, pembuat laporan, dll. Jika sistem informasi berbasis aplikasi web, pastikan juga pengelolaan dengan penyedia hosting dan jaringan internet sudah berjalan dan semuanya aktif (Tantra, 2012).

### **8.1.4 Hal yang Dipelajari**

Setiap proyek, baik yang berhasil diselesaikan maupun dinyatakan gagal, masih terdapat berbagai hal yang dapat dipelajari selama proyek berlangsung.

Untuk mengumpulkan informasi ini, Anda dapat melakukan survei terhadap anggota tim atau pemangku kepentingan yang terkait langsung dengan proyek, yang dapat diberikan selama penyelesaian proyek. Beberapa aspek proyek yang dapat diselidiki meliputi:

1. Hal-hal umum mengenai proyek dan komunikasi.
2. Masalah estimasi jadwal, apakah cukup akurat atau perlu banyak penyesuaian.
3. Masalah anggaran, apakah anggaran yang disediakan cukup, atau perlu ditambah, atau bahkan diperluas.
4. Apakah proses desain, implementasi dan pengujian dapat dilakukan secara efektif sesuai kebutuhan.
5. Masalah persepsi proses, apakah sesuai dengan rencana proyek, apakah dapat dilaksanakan dengan baik, dan apakah berjalan efektif.
6. Apakah produk yang dihasilkan proyek memenuhi persyaratan dan apakah tujuan proyek dapat tercapai.
7. Kepuasan klien, apakah sistem informasi yang dihasilkan dapat digunakan dan mendukung operasional sehari-hari klien untuk meningkatkan efisiensi.
8. Risiko dan Mitigasi, apakah proyek dapat mengidentifikasi risiko yang mungkin terjadi, dan jika terjadi dapat diprediksi dengan baik sehingga tidak menghambat pelaksanaan proyek.
9. Metodologi manajemen proyek, apakah dapat mendukung pelaksanaan proyek

10. Penutupan proyek, bagaimana proses penutupannya, dan apakah masih ada pekerjaan yang harus diselesaikan. Tujuan dari survei ini adalah untuk mengetahui apa yang dapat dipelajari dari proyek mengingat pengalaman proyek masih segar dalam ingatan anggota tim (Tantra, 2012).

### **8.1.5 Pencatatan Historis Proyek**

Manajer proyek memang perlu mengumpulkan data dari proyek sebelumnya, terutama saat merencanakan dan mengatur proyek berikutnya. Walaupun proyek yang akan diimplementasikan berbeda dengan proyek sebelumnya, seperti proyek instalasi server yang dilanjutkan dengan proyek pengembangan software, ada beberapa hal yang dapat dijadikan analogi dalam estimasi proyek. Data historis ini harus ditata dalam bentuk yang mudah diakses dan dicari sehingga dapat segera ditemukan saat dibutuhkan.

1. Merencanakan dan mengimplementasikan durasi waktu implementasi.
2. Perencanaan dan pelaksanaan jam kerja.
3. Salinan dari setiap perubahan yang disetujui pada rencana manajemen proyek.
4. Semua notulen rapat.
5. Laporan kinerja subkontraktor.
6. Laporan kepuasan klien.
7. Tinjauan akhir proyek/lesson learned (Tantra, 2012).

### **8.1.6 Post Implementation Evaluation Report (PIER)**

Tujuan berfungsi untuk merekam informasi sejarah proyek yang membantu pembaca memahami kekuatan dan kelemahan dari pelaksanaan proyek. Berdasarkan pengetahuan ini, rekomendasi untuk proyek berikut dapat dibuat. Data penting berikut ini dapat disertakan dalam laporan akhir ini:

1. Keberhasilan proyek secara keseluruhan.
2. Organisasi proyek.
3. Teknik yang digunakan untuk mencapai hasil.
4. Jadwal dan anggaran.
5. Risiko yang muncul dan solusinya.
6. Kekuatan dan kelemahan proyek.
7. Rekomendasi tim proyek (Tantra, 2012).

## **8.2 Kontrak Pemeliharaan Sistem (System Maintenance Contract)**

Sistem informasi yang sudah sampai pada tahap produksi dan digunakan sehari-hari untuk mendukung operasional suatu perusahaan atau organisasi masih memerlukan bantuan teknis. Secara alami, dukungan ini harus ditawarkan oleh tim yang memiliki pemahaman menyeluruh tentang sistem dan mampu menangani dan mengatasi masalah apa pun yang mungkin timbul.

Anda juga bisa menyiapkan tim helpdesk yang akan ditempatkan secara terpusat atau di lokasi tempat sistem informasi bekerja namun harus tetap dapat dijangkau saat serah terima sistem informasi terjadi. Semua pertanyaan dan permintaan pengguna untuk dukungan teknis akan ditangani oleh staf helpdesk ini. Karena itu, Anda harus memberikan pelatihan khusus kepada staf ini agar mereka dapat mendukung pengguna yang membutuhkan bantuan, memelihara sistem, dan memberikan instruksi dan arahan kepada pelanggan tersebut..

Pilihan lainnya adalah menugaskan sejumlah anggota tim teknis dari proyek untuk melaksanakan tanggung jawab dukungan teknis seperti menawarkan bantuan kepada pengguna yang memerlukannya dan melakukan pemeliharaan sistem selama jangka waktu yang telah ditentukan. Untuk itu diperlukan satu kontrak terpisah yang biasanya disebut sebagai kontrak pemeliharaan sistem (System Maintenance Contract - SMC).

Dalam SMC, spesifikasi pekerjaan harus disebutkan secara jelas, dan meskipun memiliki batas waktu, SMC bukanlah proyek, karena karakteristiknya hanya memberikan dukungan teknis (technical support) untuk sistem informasi terpasang, dan melakukan pemeliharaan sistem. Hal-hal yang dapat dimasukkan dalam pemeliharaan sistem adalah:

1. Corrective maintenance. Jika bug minor atau ketidakkonsistenan sistem ditemukan, pemeliharaan sistem dilakukan untuk mengatasi masalah dan memberikan peningkatan pada sistem. Selain itu, koreksi ini dibuat sebagai tambalan atau "tambalan" ke sistem, bukan sebagai perubahan sistem. Sistem juga dapat diperbarui secara berkala sebagai alternatif.
2. Adaptive maintenance. Karena perubahan lingkungan operasi dimana sistem informasi digunakan, seperti perubahan platform, peralatan yang lebih banyak, penambahan pengguna, dan sebagainya, pemeliharaan sistem ini melibatkan pemberian pembaruan secara berkala..
3. Perfective maintenance. Untuk meningkatkan kinerja sistem informasi dilakukan pemeliharaan sistem. Ini diperlukan karena peningkatan transaksi, volume data yang perlu diproses, jumlah orang yang mengakses sistem, dll.
4. Preventive maintenance. Sebagai persiapan untuk potensi masalah sistem, pemeliharaan sedang dilakukan pada sistem ini. Misalnya, pertumbuhan besar-besaran dalam transaksi, sistem yang dapat digunakan banyak orang dari jarak jauh, dan keamanan terhadap masalah keamanan sistem dan data. Pencadangan data, kebijakan manajemen risiko, dan strategi pemulihan data adalah bagian dari pemeliharaan sistem pencegahan.

Adanya kesepakatan berarti tim ini harus memenuhi service level agreement (SLA) yang disepakati. Dukungan teknis dapat diberikan tidak hanya berdasarkan kontrak, tetapi juga dalam bentuk dukungan dan pemeliharaan di tempat, yaitu bantuan melalui kunjungan rutin ke pengguna untuk pemeriksaan dan pemeliharaan sistem. Selain itu, layanan on-call dan pemeliharaan juga tersedia. Semua layanan ini harus disediakan dalam kerangka SLA untuk menjaga kualitas layanan yang diberikan (Tantra, 2012).

## DAFTAR PUSTAKA

- Alda, M., Sibarani, F. H., & Nasution, A. (2022). Validation Information System Munaqasyah Exam Registration Based Android. *Sinkron*, 7(4), 2280–2291.
- Aldisa, R. T. (2022). Penggunaan Metode Certainty Factor Pada Sistem Pakar Deteksi Kerusakan Perangkat Keras (Hardware) Komputer di Laboratorium Berbasis Android. *Journal of Information System Research (JOSH)*, 3(3).
- Ardi, P. H., Elmuna, E. A. F., Zamroni, M. A., & Yaqin, M. A. (2020). Implementasi Project Management Body of Knowledge (PMBOK) pada Organisasi Pondok Pesantren. *ILKOMNIKA: Journal of Computer Science and Applied Informatics*, 2(3).
- Ardiana, D. P. Y., Romindo, Gustiana, Z., Wijaya, A. H., Pakpahan, A. F., Yuswardi, Y., Jamaludin, Rahmah, S. A., Ahmad, N., Prasetio, A., Munsarif, M., & Kaunan, F. J. (2021). *Sistem Basis Data Lanjutan*. Yayasan Kita Menulis.
- Arfan, S., Yaumi, M., & Yusuf T, M. (2019). Penerapan Sistem Informasi Manajemen Dalam Pengolahan Data Peserta Didik Di Mi Nasrul Haq. *Idarah: Jurnal Manajemen Pendidikan*, 3(2), 289.
- Arianie, G. P., & Puspitasari, N. B. (2017). Perencanaan Manajemen Proyek Dalam Meningkatkan Efisiensi Dan Efektifitas Sumber Daya Perusahaan (Studi Kasus : Qiscus Pte Ltd)Perencanaan Manajemen Proyek Dalam Meningkatkan Efisiensi Dan Efektifitas Sumber Daya Perusahaan (Studi Kasus : Qiscus Pte Ltd). *J@ti Undip : Jurnal Teknik Industri*, 12(3), 189.
- Azhar, N. F., & Rochimah, S. (2016). Memprediksi Waktu Memperbaiki Bug dari Laporan Bug Menggunakan Klasifikasi Random Forest. *Jurnal Sistem dan Informatika*, 11(1), 156–164.
- Basriyanto. (2022). *Pengembangan Sistem Informasi Akademik Siswa (SIAS) Berbasis Web*. Guepedia.
- Devi, C. (2012). Peranan Perencanaan Sistem Informasi Terhadap Kualitas Sistem. *Berkala Ilmiah Mahasiswa Akuntansi*, 1(1), 42–46.
- Dikrozian, R., Nisa, N. M. B., Putra, R., Pamungkas, Suryapratama, A., & Ulum, P. (2018). Manajemen Proyek Pembuatan Sistem Informasi Promosi Melalui Website. *Seminar Nasional Informatika, Sistem Informasi Dan Keamanan Siber (SEINASI-KESI)*.
- Dimiyati, H., & Nurjaman, K. (2016). *Manajemen Proyek*. Pustaka Setia.
- Fatia, S., Safitri, A. H., Galuh, Y., Yaqin, M. A., & Fauzan, A. C. (2022). Pemodelan Proses Bisnis Organisasi Sekolah Berbasis Work Breakdown Structure Berdasarkan Standar Nasional Pendidikan. *ILKOMNIKA: Journal of Computer Science and Applied Informatics*, 4(2).

- Fatta, H. Al. (2007). *Analisis dan Perancangan Sistem Informasi untuk Keunggulan Bersaing Perusahaan dan Organisasi Modern*. C.V Andi Offset.
- Fikry, M. (2019). *Basis Data*. Unimal Press.
- Hall, J. A. (2007). *Sisem Informasi Akuntansi*. Salemba Empat.
- Handayani, P. W., Hidayanto, A. N., Pinem, A. A., Hapsari, I. C., Azzahro, F., Sandhyaduhita, P. I., & Ayuningtiyas, D. (2018). *Pengantar Sistem Informasi Manajemen Rumah Sakit (SIMRS)*. Rajawali Pers.
- Harahap, E. F., Adisuwiryono, S., & Fitriana, R. (2022). *Analisis dan Perancangan Sistem Informasi*. Wawasan Ilmu.
- Hartono, S. (2018). *Penerapan Software Requirement Specifications pada Sistem Website Sinode GKJ di Salatiga: Studi Kasus pada Bagian PDILM Sinode Gkj Salatiga*. Universitas Kristen Satya Wacana.
- Haryadi, E., Abdussomad, & Robi. (2019). Implementasi Sistem Backup Data Perusahaan Sebagai Bagian dari Disaster Recovery Plan. *SAINSTECH: Jurnal Penelitian dan Pengkajian Sains dan Teknologi*, 6(11).
- Haryana, K. S. (2019). Penerapan agile development methods dengan framework scrum pada perancangan perangkat lunak kehadiran rapat umum berbasis Qr-Code. *Jurnal Computech & Bisnis (e-Journal)*, 13(2).
- Huda, M., Yaqin, M. A., Kurniawan, R. F., & Choiri, M. W. F. (2022). Survei Strategi Pengujian Software Menggunakan Metode Systematic Literature Review. *ILKOMNIKA: Journal of Computer Science and Applied Informatics*, 4(1).
- Indrayani, H. (2012). Penerapan Teknologi Informasi Dalam Peningkatan Efektivitas, Efisiensi Dan Produktivitas Perusahaan. *Jurnal EL-RIYASAH*, 3(1), 48–56.
- Institute, P. M. (2013). *A Guide to the Project Management Body of Knowledge, Fifth Edition*. Newtown Square.
- Istiyana, A. N. (2022). *Buku Ajar Audit Sistem Informasi*. Nas Media Pustaka.
- Jauhari, A., Anamisa, D. R., & Mufarroha, F. A. (2019). *Rekayasa Perangkat Lunak*. Media Nusa Creative.
- Khadafi, S., Puspita, I. A., & Tripiawan, W. (2020). Developing Project Stakeholder Management Dan Rencana Komunikasi Pada Proyek Pengadaan Dan Pemasangan Outside Plant Fiber Optic (osp-fo) Sttf Pt-3 Tahap 1 2020 Di Pt Xyz Bandung Barat. *eProceedings of Engineering*, 7(1).
- Kurniawan, E. (2015). PENERAPAN TEKNOLOGI CLOUD COMPUTING DI UNIVERSITAS Studi Kasus : Fakultas Teknologi Informasi UKDW. *Eksis*, 08(01), 29–36.
- Kusrini, & Koniyo, A. (2007). *Tuntunan Praktis Membangun Sistem Informasi Akuntansi dengan Visual Basic dan Microsoft Sql Server+cd*. ANDI.

- Lesmana, I. P. D., & Antika, E. (2019). *Manajemen Proyek dengan Scrum*. CV. Absolute Media.
- Mandar, R. (2017). *Kitab Kumpulan Tips, Latihan, dan Soal Database*. PT. Elex Media Komputindo.
- Maydianto, & Ridho, M. R. (2021). Rancang Bangun Sistem Informasi Point of Sale Dengan Framework Codeigniter Pada Cv Powershop. *Jurnal Comasie*, 04(02), 50–59.
- Mulyoto, D. P., & Kurniali, S. (2013). *Super Project Manager - Tip & Trik Memahami PMBOK Guide 5th*. PT. Elex Media Komputindo.
- Muttaqin, Sinambela, M., A., A., Parewe, A. M. A. K., Sari, A. N., Fadhillah, Y., Sari, N. R., Saputra, H., Darsin, D., Pakpahan, A. F., Siregar, M. N. H., & Sasongko, D. (2022). *Teknologi Cloud Computing*. Yayasan Kita Menulis.
- Najwaini, E., & SN, A. (2012). DOKUMENTASI SEBAGAI BAGIAN DARI PERANGKAT LUNAK. *Seminar Nasional Teknologi Informasi & Komunikasi Terapan 2012*, 2(1).
- Nizar, N. I. (2018). ANALISIS TINGKAT KEMATANGAN PMO DAN TATA KELOLA PROYEK BERBASIS TEKNOLOGI INFORMASI PADA BANK X. *Jurnal Mandiri*, 2(2).
- Nugraha, D. W. (2011). Software requirement dalam membangun sistem informasi pelayanan publik. *MEKTEK*, 13(3).
- Nugroho, O. I. A. (2016). Sistem Informasi Kepegawaian Berbasis Web Untuk Peningkatan Kinerja Karyawan CV. Gunung Mas Purwokerto. *JURNAL MEDIA APLIKOM*, 7(2).
- Oktaviyani, N., & Bachtiar, L. (2017). Sistem Informasi Administrasi Bimbingan Belajar Pada English Solution Sampit Berbasis Web. *Jurnal UNDA*, 6(3), 4.
- P., A. (2016). *Langkah-Langkah Efektif Menyusun SOP*. Huta Publisher.
- Parlan, M., Agustine, D., Basid, A., & Hidayanto, M. (2020). Peranan Konsultan Manajemen Konstruksi Terhadap Proses Pembangunan Gedung Bertingkat (Studi Kasus : Mari Gold BSD City). *JIMTEK : Jurnal Ilmiah Fakultas Teknik*, 1(2).
- Patria, G. (2018). Sistem Informasi Pengelolaan Penggunaan Ruang Rapat Dan Jadwal Acara Ditjen Sdppi Berbasis Web. *JUPITER : Jurnal Penerapan Ilmu-ilmu Komputer*, 4(1), 1–6.
- PMBOK. (2013). *A Guide to the Project Management Body of Knowledge*. Project Management Institute, Inc.
- Prabowo, W. S., & Agustina, C. (2017). Perancangan Sistem Informasi Pengolahan Nilai Rapor Berbasis Web Pada SMK Negeri 1 Purworejo. *Jurnal Khatulistiwa Informatika*, 5(1).

- Prasimsha, O., & Putro, H. P. (2020). No Title. *AUTOMATA*, 1(2).
- PRATAMA, R., & Harahap, G. Y. (2022). PROYEK PEMBANGUNAN LIVING PLAZA MEDAN. *Laporan Kerja Praktek Mahasiswa Teknik*, 1(3).
- Purbaningtyas, R. (2019). Penerapan fuctional testing pada uji kelayakan aplikasi mobile smart malnutrition detection. *Jurnal Teknologi Informasi*, 18(3).
- Purwanto, R. A., & Sari, P. K. (2015). Analisa Tingkat Kapabilitas Manajemen Proyek Pt. Agate International Dengan Menggunakan Model Cmm. *eProceedings of Management*, 2(3).
- Putri, R. A. (2022). *Buku Ajar Basis Data*. Media Sains Indonesia.
- Rachmad, Y. E., Tampubolon, L. P. D., Siska, Purbaratri, W., Sudipa, I. G. I., Desyanti, Sulistianto, Migunani, Ariana, A. A. G. B., Faried, M. I., Atmojo, D., & Kurniawan, H. (2023). *Rekayasa Perangkat Lunak*. PT. Sonpedia Publishing Indonesia.
- Rosa, A. C., Sunardi, H., & Setiawan, H. (2019). Rekayasa Augmented Reality Planet dalam Tata Surya sebagai Media Pembelajaran Bagi Siswa SMP Negeri 57 Palembang. *Jurnal Ilmiah Informatika Global*, 10(1). <https://doi.org/10.36982/jig.v10i1.728>
- Samsudin, Irawan, M. D., & Harahap, A. H. (2019). Mobile App Education Gangguan Pencernaan Manusia Berbasis Multimedia Menggunakan Adobe Animate Cc. *Jurnal Teknologi Informasi*, 3(2), 141–148.
- Setiawan, E. (2019). Manajemen proyek Sistem Informasi Penggajian Berbasis Web. *Jurnal Teknik*, 17(2).
- Setyawati, E., Santamoko, R., Handoko, A. L., & Setiawan, P. (2021). *Manajemen Proyek Sistem Informasi*. Insan Cendekia Mandiri.
- Sitanggang, N., Simarmata, J., & Luthan, P. L. A. (2019). *Pengantar Konsep Manajemen Proyek untuk Teknik*. Yayasan Kita Menulis.
- Sriwulandari, A., Hidayati, H., & Pudjoatmojo, B. (2014). Analisis dan Evaluasi Aspek Usability Pada Web HRMIS Telkom University Menggunakan Usability Testing. *eProceedings of Engineering*.
- Sudarma, M. (2012). *Manajemen Proyek Teknologi Informasi*. Udayana University Press.
- Suendri. (2018). Implementasi Diagram UML (Unified Modelling Language) Pada Perancangan Sistem Informasi Remunerasi Dosen Dengan Database Oracle (Studi Kasus: UIN Sumatera Utara Medan). *Jurnal Ilmu Komputer dan Informatika*, 3(1), 1–9.
- Sugiarto, E. (2015). *Menyusun Proposal Penelitian Kualitatif: Skripsi dan Tesis: Suaka Media*. Suaka Media.
- Suprpto, A., & Sasongko, D. (2021). Evaluasi Performa Website Berdasarkan Pengujian Beban Dan Stress Menggunakan Loadimpact (Studi Kasus Website Iain Salatiga).

- Network Engineering Research Operation*, 6(1).
- Supriyanto, B. S. Y., Pellokila, N. S. M., & ... (2021). Pendampingan Pengelolaan Risiko dan Pengendalian Internal Siklus Pengeluaran Kas Desa Sumberejo, Dadapayam, Kabupaten Semarang. *Magistrorum Et ...*, 02(02), 343–356.
- Suryapermana, N. (2016). Perencanaan dan Sistem Manajemen Pembelajaran. *Tsarwah: Jurnal Ekonomi dan Bisnis Islam*, 1(2), 29–44.
- Suryoprayogo, A., & Sidiq. (2016). Implementasi Algoritma Brute Force Pada Perancangan Aplikasi Kamus Bahasa Sunda Berbasis Android. *Informatika Dan Bisnis*, 5(1), 1–12.
- Susilawati, & Gunawan, A. (2018). Perangkat Lunak Identifikasi Kerusakan Televisi Tabung. *Prosiding Seminar Nasional Rekayasa Teknologi Informasi*, November.
- Syamsibar. (2021). Konsep Manajemen Agile;(Methodologi dalam Manajemen Proyek yang berfokus pada manusia, hasil, metode). *JURNAL SIPATOKKONG BPSDM SULSEL*, 2(2).
- Tantra, R. (2012). *Manajemen Proyek Sistem Informasi*. ANDI.
- Utama, J. A., Yuana, H., & Febrinita, F. (2022). ANALISIS DEBUGGING SISTEM I MOBIL PERIZINAN V2 PADA FITUR APLIKASI DENGAN TEKNIK ELICITATION DI DINAS PENANAMAN MODAL DAN PELAYANAN. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 6(1).
- Warkim, & Sensuse, D. I. (2017). Model Integrasi Sistem dengan Pendekatan Metode Service Oriented Architecture dan Model View Controller pada Pusat Penelitian Perkembangan Iptek Lembaga Ilmu Pengetahuan Indonesia. *Jurnal Teknik Informatika dan Sistem Informasi*, 3(1).
- Wibisono, W., & Baskoro, F. (2002). Pengujian perangkat lunak dengan menggunakan model behaviour UML. *Jurnal Ilmiah Teknologi Informasi*, 1(1).
- Wibowo, A. (2021). *MANAJEMEN PERUBAHAN, Change Management*. Yayasan Prima Agus Teknik.
- Wibowo, S. H., Lestantri, I. D., Adriantantri, E., Muammar, Irawan, J. D., Jamaludin, Suryani, A. I., Defiariany, Simanjuntak, P., Turaina, R., Yanto, G., Putri, N. E., & Wahidin, A. J. (2022). *Manajemen Proyek Teknologi Informatika (I)*. Global Eksekutif Teknologi.
- Wiguna, A., & Winoto, S. (2020). PENGARUH GAYA KEPEMIMPINAN MANAJER PROYEK TERHADAP KINERJA KARYAWAN PADA PERUSAHAAN JASA KONSTRUKSI. *Prosiding Seminar Intelektual Muda*.
- Yanti, Y. (2009). Model Rancangan Sistem Informasi Project Monitoring. *CommIT (Communication and Information Technology) Journal*, 3(2).
- Yulianto, A. A., Gartina, I., Astuti, R., Dewi, S., Sari, S. K., & Witanti, W. (2009). *Analisis dan Perancangan Sistem Informasi*. Politeknik Telkom Bandung.

Zufria, I. (2013). Pemodelan Berbasis UML ( Unified Modeling Language ) dengan Strategi Teknik Orientasi Objek User Centered Design ( UCD ) dalam Sistem Administrasi Pendidikan Pemodelan Berbasis UML ( Unified Modeling Language ) dengan. *Journal Sains & Teknologi*, 1(1), 1–16.