

BAB II TINJAUAN PUSTAKA

2.1 Kecerdasan Buatan

Kecerdasan buatan (AI) mengacu pada simulasi kecerdasan manusia yang diprogram untuk berpikir seperti manusia dan mengikuti tindakannya. Istilah ini dapat diterapkan pada mesin yang menunjukkan sifat yang berhubungan dengan pikiran manusia, seperti pembelajaran dan pemecahan masalah kemudian diproses oleh komputer. Proses ini melibatkan perolehan informasi melalui penggunaan dan pendekatan penalaran. Topik penelitian AI meliputi pemecahan masalah, penalaran, perencanaan, bahasa alami, pemrograman, dan pembelajaran mesin. Otomasi, robotika, perangkat lunak canggih, dan program komputer menjadi ciri perkembangan di bidang kecerdasan buatan. (Hapsari, 2021)

Kecerdasan buatan merupakan bidang ilmu komputer yang khusus mengembangkan perangkat lunak dan perangkat keras yang secara sempurna dapat meniru beberapa fungsi otak manusia atau cabang ilmu komputer yang mempelajari otomatisasi *intelligent*. Kecerdasan buatan didasarkan pada prinsip teori dan terapan yang relevan. (Pratama & Munawaroh, 2019):

1. Struktur data yang digunakan dalam representasi pengetahuan.
2. Algoritma dalam penerapan pengetahuan.
3. Mengimplementasi teknik bahasa dan pemrograman.

Aplikasi kecerdasan buatan adalah pengetahuan atau lebih besarnya adalah basis pengetahuan. Pengetahuan merupakan informasi untuk lebih mudah memecahkan masalah dan mengambil keputusan. Pengetahuan terdiri dari fakta, pemikiran, teori, prosedur, dan hubungan satu sama lain.

Bidang-bidang teknik kecerdasan buatan diantaranya adalah sistem pakar, logika *fuzzy*, jaringan syaraf tiruan, pengolahan bahasa alami, pengolahan citra digital (*digital image processing*) dan robotik.

Kecerdasan buatan memiliki sifat penting dalam bidang penerapannya, antara lain (Pratama & Munawaroh, 2019) :

1. Komputer digunakan untuk melakukan pertimbangan menggunakan simbol.

2. Fokus terhadap solusi algoritma tidak memberikan respon. Berdasar pada pencarian heuristik sebagai teknik pemecah masalah Kecerdasan Buatan.
3. Usaha yang dilakukan ditujukan untuk menangkap dan memanipulasi sifat AI daripada metode numerik.
4. Melakukan usaha untuk menangani arti semantik dan bentuk sintaks.
5. Jawaban yang diberikan bersifat cukup (*sufficient*). Metode pemecahan problema heuristik hasil eksak atau optimal akan berharga atau tidak mungkin dilakukan.
6. Penggunaan pengetahuan khusus dalam memecahkan masalah dalam dasar sistem pakar.
7. Penggunaan pengetahuan tingkat meta mempengaruhi pengendalian dari strategi pemecahan masalah. Hal ini merupakan persoalan yang sangat sulit dan hanya ditujukan pada sejumlah kecil sistem, dan muncul sebagai objek riset.

2.2 Sistem Pakar

Based System adalah aplikasi komputer yang dirancang untuk mendukung pengambilan keputusan dan pemecahan masalah di bidang tertentu. Sistem ini didasarkan pada pengetahuan dan metode analisis yang ditentukan oleh para ahli sesuai dengan bidang keahliannya. Sistem ini disebut sistem pakar karena mempunyai fungsi dan peran yang sama dengan seorang pakar, pengetahuan dan pengalamannya diperlukan untuk menyelesaikan permasalahan. Sistem berfungsi sebagai sistem pendukung keputusan utama atau sistem pendukung manajemen. (Rukun & Hayadi, 2018)

Sistem pakar termasuk dalam kecerdasan buatan. Sistem Pakar menggunakan pengetahuan, fakta, dan teknik berfikir dalam menyelesaikan masalah yang diselesaikan oleh seorang pakar. Sistem Pakar digunakan orang awam untuk meningkatkan kemampuan dalam memecahkan masalah dan memperbanyak sumber pengetahuan yang semakin langka.

Program sistem pakar dapat menggantikan keberadaan seorang pakar. Alasan mengapa sistem pakar dikembangkan adalah sebagai berikut :

1. Menyediakan kepakaran diberbagai waktu dan kondisi.
2. Otomatis mengerjakan tugas rutin yang membutuhkan seorang pakar.
3. Menggunakan jasa seorang pakar memerlukan biaya yang mahal.
4. Kepakaran juga dibutuhkan pada lingkungan yang tidak bersahabat.

4.2.1 Ciri – Ciri Sistem pakar

Ciri sistem pakar menurut (Rukun & Hayadi, 2018) adalah sebagai berikut :

1. Terbatas pada domain tertentu.
2. Memberikan penalaran pada data yang tidak pasti.
3. Mengemukakan rangkaian alasan dengan cara yang mudah dipahami.
4. Berdasar pada kaidah tertentu.
5. Dirancang dan dikembangkan secara bertahap.

2.2.2 Manfaat Sistem Pakar

Sistem pakar sangat populer karena sangat banyak kemampuan dan manfaatnya:

1. Meningkatkan produktivitas, karena sistem pakar lebih cepat daripada manusia.
2. Orang awam bekerja seperti seorang pakar.
3. Meningkatkan kualitas, karena mengurangi kesalahan.
4. Akses pengetahuan mudah seperti seorang pakar.
5. Digunakan sebagai media pelengkap dalam pelatihan.
6. Pengguna pemula sistem pakar menjadi lebih berpengalaman.
7. Meningkatkan kemampuan menyelesaikan masalah karena sistem pakar mengambil sumber pengetahuan dari banyak pakar.

2.2.3 Komponen Sistem Pakar

Program sistem pakar terdiri dari komponen yang mutlak.

Komponen itu adalah sebagai berikut :

1. Basis Pengetahuan (*Knowledge Base*)

Basis pengetahuan inti program sistem pakar karena basis ini merupakan representasi pengetahuan (*Knowledge Representation*) dari seorang pakar.

2. Basis Data

Basis data mengandung bagian semua fakta, baik fakta dari mulai beroperasi maupun fakta saat pengambilan kesimpulan.

3. Mesin Inferensi

Mesin inferensi mengandung mekanisme fungsi berfikir dan pola penalaran sistem yang digunakan oleh seorang pakar. Mekanisme menganalisa masalah tertentu dan mencari jawaban atau kesimpulan yang terbaik. Mesin inferensi mengacak dengan mencocokkan basis pengetahuan dengan fakta yang ada dalam basis data. Teknik inferensi yaitu *Backward Chaining* yang memulai penalaran dari kesimpulan hipotesa menuju fakta yang mengandung hipotesa dan *Forward Chaining* merupakan kebalikan dari *Backward Chaining* yaitu memulai dari sekumpulan data menuju kesimpulan. (Rukun & Hayadi, 2018)

Kedua metode inferensi dipengaruhi oleh tiga macam teknik penelusuran yaitu: *Depth-first search* melakukan penelusuran kaidah secara mendalam dari simpul akar bergerak menurun ke tingkat dalam yang berurutan. *Breadth-first search* bergerak dari simpul akar, simpul yang ada pada setiap tingkat diuji sebelum pindah ke tingkat selanjutnya. *Best first search* bekerja berdasarkan kombinasi kedua metode sebelumnya.

4. Antar Muka Pemakai (*User Interface*)

User Interface bagian penghubung antara program sistem pakar dengan pemakainya. Program akan mengajukan pertanyaan berbentuk "ya / tidak" atau berbentuk menu pilihan. Melalui jawaban yang diberikan oleh pemakai, sistem pakar mengambil kesimpulan berupa informasi sesuai dengan sifat dari sistem pakar.

2.3 Penalaran Berbasis Aturan (*Rule Based Reasoning*)

Penalaran berbasis aturan, pengetahuan direpresentasikan menggunakan aturan *IF-THEN*. Aturan ini digunakan ketika pakar dapat menyelesaikan masalah secara berurutan dengan sejumlah pengetahuan pakar. Aturan ini juga digunakan untuk penjelasan tentang langkah pencapaian solusi.

2.4 Pelacakan ke Depan (*Forward Chaining*)

Forward Chaining adalah teknik pencarian dengan fakta kemudian mencocokkan dengan bagian *IF* dari *rules IF-THEN*. *Rule* dapat dieksekusi dengan sebuah fakta baru (bagian *THEN*) dan ditambahkan ke dalam *database*. Setiap pencocokan dimulai dari *rule* teratas. Setiap *rule* hanya boleh dieksekusi satu kali. Proses pencocokan berhenti bila tidak ada lagi *rule* yang dapat dieksekusi. Metode pencarian yang digunakan adalah *Depth-First Search*, *Breadth-First Search*, atau *Best-First Search*. (Azis et al., 2019)

Contoh forward chaining :

R1: *IF (y AND d) THEN Z;*

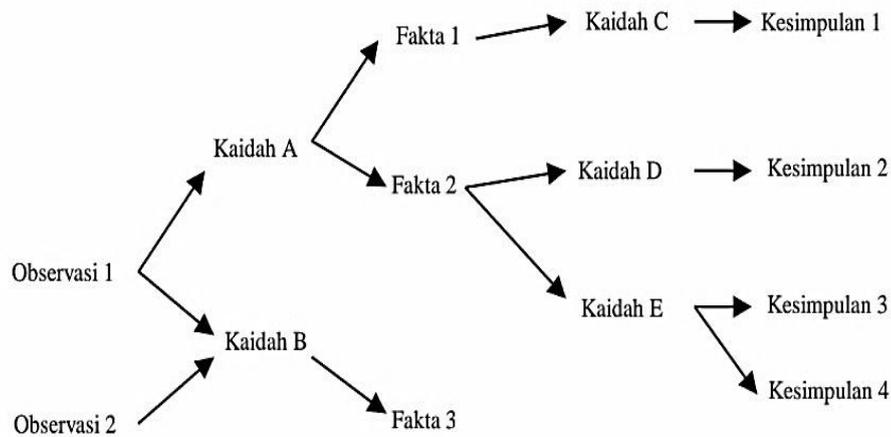
R2: *IF (X AND 6 AND E) THEN Y* R3: *IF a THEN X*

R4: *IF THEN I*

R5 : *IF (1 AND m) THEN n*

Fakta: a, b,c,d, dan e bernilai benar.

Goal: Menentukan apakah z bernilai benar atau salah.



Gambar 2. 1 Diagram Pelacakan ke Depan (*Forward Chaining*)

Sumber : (Rukun & Hayadi, 2018)

Pendekatan dilakukan mencari solusi optimal berdasarkan pengetahuan yang ada, dan pencarian dimulai dengan menentukan dimana dan bagaimana menyelesaikan masalah melalui proses optimal. Seperti yang telah dijelaskan sebelumnya, *forward chaining* dikenal sebagai penalaran *bottom-up* karena kesimpulan yang mengarah dari tingkat rendah ke tingkat tinggi berdasarkan fakta. Bentuk penalaran ini berbentuk pohon keputusan. Algoritma ini rangkaian maju yang sederhana dan mengasumsikan fakta yang diketahui dan mendorong kesimpulan berdasarkan fakta tersebut. Proses ini diulangi sampai ditemukan konklusi yang benar.

Penalaran bersifat *bottom-up*, dan fakta tidak dapat dipecah menjadi satuan terkecil yang bermakna, sehingga merupakan satuan dasar paradigma berbasis pengetahuan. Kata "bebek" bisa memiliki arti yang berbeda, yang berarti kita akan mendapatkan huruf b, e, b, e, k yang tidak berarti. Fakta harus disajikan dalam bentuk sederhana sebagai data. (Darnila et al., 2019)

Langkah-langkah yang harus dilakukan dalam membuat sistem forward chaining berbasis aturan (Samsudin, 2016):

a. Pendefinisian Masalah

Tahap ini meliputi pemilihan domain masalah dan akuisipengetahuan.

b. Pendefinisian Data Input

Sistem forward chaining memerlukan data awal untuk memulai inferensi.

c. Pendefinisian struktur Pengendalian Data

Aplikasi yang kompleks memerlukan premis tambahan untuk membantu mengendalikan pengaktifan suatu aturan.

d. Penulisan Kode Awal

Tahap ini berguna untuk menentukan apakah sistem telah menangkap domain pengetahuan secara efektif dalam struktur aturan yang baik.

e. Pengujian sistem

Pengujian sistem dilakukan dengan beberapa aturan untuk menguji sejauh mana sistem berjalan dengan benar.

f. Perancangan antar Muka

Antarmuka adalah salah satu komponen penting dari suatu sistem. Perancangan antarmuka dibuat bersama-sama dengan pembuatan basis pengetahuan.

g. Pengembangan Sistem

Pengembangan sistem meliputi penambahan antarmuka dan pengetahuan sesuai dengan prototype sistem.

h. Evaluasi Sistem

Pada tahap ini dilakukan pengujian sistem dengan masalah yang sebenarnya. Jika sistem belum berjalan dengan baik maka akan dilakukan pengembangan kembali.

2.5 Metode Certainty Factor

Nilai faktor kepastian pertama kali digunakan pada MYCIN, yaitu sistem pakar untuk mendiagnosa jenis penyakit infeksi darah, kemudian menentukan jenis obat dan dosisnya. Faktor kepastian merupakan penurunan dan pengembangan dari teori peluang berkondisi (*Bayes theorem*). Faktor kepastian didapatkan dari operasi pengurangan nilai kepercayaan (*measure of belief*) oleh nilai ketidakpercayaan (*measure of disbelief*). Tujuan utama penggunaan faktor kepastian adalah untuk memproses ketidakpastian dari fakta dan gejala dengan menghindari keperluan

data dan perhitungan yang besar. Hal yang menjadi masalah adalah faktor kepastian tersebut sering tidak konsisten dan sulit dibuktikan kebenarannya secara matematik. (Marimin, 2017)

Nilai probabilitas (peluang) merupakan nilai pendukung dan nilai penghambat. Nilai ini pertama kali digunakan dan dikembangkan pada PROSPEKTOR : sistem pakar untuk menginterpretasikan kandungan bahan tambang di suatu lokasi. Kelemahan dari teknik ini adalah dalam segi pengumpulan data dan akuisisi pengetahuan. Teknik ini baru dapat dipercaya jika digunakan data yang cukup besar jumlahnya. Kelemahannya lainnya adalah perhitungan probabilitas lebih condong pada persoalan-persoalan yang bersifat frekuentif. Padahal dalam kenyataan banyak persoalan-persoalan yang bersifat kemungkinan (*possibilistic*). Skala ukuran *measure of belief* (MB) dan *measure of disbelief* (MD) digunakan untuk mengukur kekuatan fakta. Notasi yang digunakan adalah :

MB[he] = X, berarti "ukuran kenaikan tingkat kepercayaan untuk hipotesis h, didasarkan pada fakta e sama dengan X"
 MD[he] = Y, berarti "ukuran kenaikan tingkat ketidakpercayaan untuk hipotesis h, didasarkan pada fakta e sama dengan Y"

Fakta e bukan hanya hasil observasi, dapat dari hipotesa lain sebagai subjek konfirmasi. Sebagai ilustrasi MB[h₁, h₂] menunjukkan kepercayaan untuk hipotesis h₁ yang diberikan setelah hipotesa h₂ adalah benar (Marimin, 2017).

Sesuai dengan teori probabilitas maka probabilitas seorang pakar P(h) menunjukkan tingkat kepercayaan untuk h. Jadi 1 - P(h) digambarkan sebagai perkiraan ketidakpercayaan pakar mengenai kebenaran hipotesis h. Jika P(h/e) lebih besar dibanding P(h) maka penelitian dari e menaikkan kepercayaan pakar untuk h dan menurunkan ketidakpercayaan pakar tentang h. Hal ini menurunkan rumus ketidakpercayaan dalam perbandingan:

$$\frac{P(h/e) - P(h)}{1 - P(h)}$$

Sebaliknya bila P(h/e) lebih kecil dibanding P(h) maka penelitian e akan menurunkan rumus kepercayaan pakar untuk h dan menaikkan ketidakpercayaan pakar mengenai kebenaran h, untuk kasus ini diberikan dalam perbandingan :

$$\frac{P(h) - P(h/e)}{P(h)}$$

Kenaikan kepercayaan h yang dihasilkan dari penelitian e diukur melalui erbandingan ini. Persamaan kondisional dan peluang awal dari hasil perbandingan tersebut adalah:

$$MB\{h,e\} = \begin{cases} 1 & \text{if } P(h) = 1 \\ \frac{\text{Max}[P(h/e), P(h)] - P(h)}{\text{Max}[1,0] - P(h)} & \text{, untuk nilai lain} \end{cases}$$

$$MD\{h,e\} = \begin{cases} 1 & \text{if } P(h) = 0 \\ \frac{\text{Min}[P(h/e), P(h)] - P(h)}{\text{Min}[1,0] - P(h)} & \text{, untuk nilai lain} \end{cases}$$

Ukuran ketiga didefinisikan sebagai *Certainty Factor* (CF) menggabungkan MB dan MD dengan rumus:

$$CF[h,e] = MB[h,e] - MD[h,e]$$

Certainty factor merupakan gabungan kepercayaan dan ketidakpercayaan, umumnya dinyatakan dalam satu harga. Diperlukan angka sebagai pembanding dari kekuatan fakta pada hipotesis. Menurut karakteristik dari ukuran kepercayaan sebagai berikut :

1. Kisaran nilai:
 - a. $0 < MB[he] < 1$
 - b. $0 < MD[he] < 1$
 - c. $-1 < CF[h,e] < +1$
2. Kekuatan fakta dan hipotesis *mutually* eksklusif jika h ditunjukkan dengan nilai tertentu $[P(h/e) = 1]$:
 - a. $MB[he] = \{1 - P(h)\} / (1 - P(h)) = 1$

- b. $MD[h,e] = 0$
 - c. $CF[h,e] = 1$
3. Jika negasi dari h , ditunjukkan dengan nilai tertentu ($P(not\ h/e) - 1$):
- a. $MB[h,e] = 0$
 - b. $MD[h,e] = \{0 - P(h)\}/[0 - P(h)] = 1$
 - c. $CF[h,e] = -1$

Catatan : $MB [not\ h,e] = 1$ jika dan hanya jika $MD[h,e] = 1$.

2.6 Website

Website merupakan suatu media dengan sejumlah halaman yang saling berhubungan (*hyperlink*) dan kemampuan menyajikan informasi dalam bentuk teks, gambar, video, audio, animasi, atau kombinasi keduanya. Saat ini *website* pada umumnya bersifat dinamis, namun dahulu ada *website* statis yang jarang dan hampir tidak ada. Ciri utama suatu *website* adalah halaman yang saling berhubungan dengan domain sebagai alamat (URL) atau *World Wide Web* (www) dan hosting sebagai media penyimpanan data dalam jumlah besar. Website dapat diakses melalui jaringan Internet dengan menggunakan platform yang disebut browser, seperti: Contoh: *Chrome, Mozilla Firefox, Internet Explorer (IE), Opera*, dll.

Website dibuat dalam mode *localhost*, artinya dapat mendesain, membuat, dan memodifikasi *website* tanpa menggunakan Internet. Jika ingin membangun *website* sampai mode publik di Internet, memerlukan beberapa aplikasi, antara lain *database* (MySQL, Oracle), *server web Apache*, editor PHP (Macromedia, Notepad++, dll.), dan *browser*. Aplikasi berbasis web biasanya menggunakan bahasa pemrograman seperti *Hypertext Preprocessor* (PHP) atau *Active Server Pages* (ASP) dan *Hypertext Markup Language* (HTML), *Cascading Style Sheets* (CSS), dan *JavaScript*. Pada level profesional, programmer membutuhkan aplikasi tambahan lainnya seperti Photoshop, Coreldraw, dll. Setelah aplikasi terinstal di komputer, maka akan berlangsung proses perancangan dan pengkodean *website*. (Elgamar, 2020)

2.7 Personal Home Page (PHP)

PHP singkatan dari *Hypertext Preprocessor* merupakan bahasa pemrograman yang biasa digunakan untuk memproses, membuat, dan mengembangkan situs web, bersamaan dengan HTML. PHP menjadi bahasa yang wajib dimiliki oleh para programmer web dan masih menjadi standar bahasa pemrograman *server side* untuk membangun website. PHP disebut sebagai bahasa pemrograman *server side* karena diproses di komputer server. PHP digunakan untuk membuat website pribadi. Setelah beberapa tahun pengembangan, PHP telah menjadi bahasa pemrograman web yang kuat yang digunakan tidak hanya untuk membuat halaman web sederhana, tetapi juga untuk situs web populer yang digunakan oleh jutaan orang, seperti Wikipedia, WordPress, dan Joomla Masu.

Untuk membuat website, PHP bukan bahasa pemrograman yang harus digunakan, dapat juga membuat situs web menggunakan HTML. Situs web yang dibuat menggunakan HTML disebut situs web statis, dan memiliki halaman web tetap. Namun dengan menggunakan PHP, isi halaman web yang dibuat dapat ditampilkan tergantung situasi, yang disebut dengan website dinamis. Situs web dinamis memungkinkan menyimpan data dalam database atau membuat halaman berdasarkan masukan pengguna. (Widing Oetomo & Bambang Mahargiono, 2020)

Kode PHP biasanya dimasukkan ke dalam dokumen HTML untuk pembuatan web. PHP disebut juga sebagai bahasa *scripting* atau bahasa pemrograman skrip. Secara teknis, PHP modul yang berjalan di server web. Oleh karena itu, diperlukan aplikasi web server untuk menjalankan PHP. Ketika menginstal XAMPP, juga menginstal aplikasi server web Apache. PHP berjalan di server web, jadi setiap kali menjalankan file PHP, perlu menjalankan server web melalui aplikasi XAMPP.

Beberapa keunggulan menggunakan Bahasa pemrograman PHP (Habibi et al., 2020).

1. Keamanan

Keamanan sebuah program selain sistem operasi menjadi sangat penting. PHP menyediakan 3 jenis autentikasi user, yaitu http autentikasi, penggunaan *cookies* dan penggunaan *session*.

2. Integritas dengan *Database*

PHP mendukung integritas, kecepatan dan efisiensi akses ke *database* yang kebanyakan menggunakan *database* berjenis relational seperti *MySQL*, *PostgreSQL*, *Oracle*, *SQLite* dan lain-lain.

3. *Cross-platform*

PHP mendukung berbagai jenis sistem operasi seperti semua varian Linux, Microsoft Windows, Mac OS dan lain-lain.

4. Reliabilitas

PHP merupakan salah satu bahasa pemrograman yang berbasis web. Alasan utama adalah dukungan dokumentasi yang lengkap, aman dan banyak komunitas *helpdesk* untuk membantu para pengembang web sistem yang menggunakan PHP (Habibi et al., 2020)

5. Harga

PHP berada dalam lisensi GPL (*GNU Public License*). Hal ini berarti bahwa PHP bebas digunakan dan didistribusikan serta gratis. Saat ini jugabanyak *hosting* gratis dan *unlimited* *men-support* PHP.

6. Kemudahan Bermigrasi

Tujuannya adalah memperbaiki kinerja dan menambah fitur-fitur baru. Kelebihan ini karena banyaknya dukungan terhadap PHP sehingga berdampak PHP terus-menerus dikembangkan.

2.8 MySQL

Berkembangnya zaman, teknologi termasuk *software* berkembang pesat. Contoh perangkat lunak adalah MySQL. MySQL terus diperbarui oleh produser masing-masing. MySQL merupakan kelanjutan dari proyek UNIREG oleh Michael Monty Widenius dan TeX (perusahaan software asal Swedia). Singkatnya, MySQL adalah server database gratis yang dilisensikan di bawah GNU General Public License (GPL), sehingga dapat digunakan untuk keperluan pribadi dan komersial tanpa membayar lisensi. (Fitri, 2020)

MySQL adalah *system* manajemen basis data menggunakan perintah dasar SQL (*Structured Query Language*) yang terkenal. *Database management system* (DBMS) MySQL multi pengguna dan multi alur digunakan banyak orang. Menurut

(Fathoroni et al., 2020), *MySQL* adalah DBMS *open source* dengan dua bentuk lisensi, yaitu *Free Software* (perangkat lunak bebas) dan *Shareware* (perangkat lunak berpemilik yang penggunaannya terbatas).

MySQL masuk ke dalam jenis RDBMS (*Relational database Management Sistem*). Istilah semacam baris, kolom, tabel, dipakai pada *MySQL*. Contohnya dalam *MySQL* sebuah *database* terdapat satu atau beberapa tabel. Sebagai pengembang perangkat lunak terdapat beberapa alasan menggunakan *MySQL* untuk membuat basis data atau database (Fitri, 2020), yaitu sebagai berikut :

1. *Speed*

MySQL menyediakan sistem basis data berkecepatan tinggi yang sempurna untuk proyek kecil hingga menengah. Berfungsi untuk perusahaan pemula, dan tidak memiliki banyak fitur seperti *Oracle*.

2. *Open source*

MySQL digunakan secara gratis. Untuk versi komersial sudah diberikan tambahan fitur berupa kemampuan spesifik dan layanan *technical support* dari *MySQL*.

3. *Scalability*

Menangani database dengan skala besar yaitu dengan jumlah *record* lebih dari 50 juta.

4. *Connectivity and Security*

Database *MySQL* dapat diakses dari semua platform internet dengan hak akses tertentu. Database *MySQL* menggunakan enkripsi *password*, jadi database aman karena memiliki password akses.

5. *Flexibility/Portability*

MySQL digunakan untuk mengembangkan aplikasi berbasis desktop dan aplikasi berbasis web menggunakan teknologi yang beragam. *MySQL* memiliki fleksibilitas terhadap teknologi untuk membangun aplikasi, menggunakan PHP, Java, C++. Aplikasi dibangun dengan cara menyediakan *plugin* dan *driver* yang spesifik pada teknologi tersebut.

6. *Cross Platform Operating System/Lintas Platform Sistem Operasi*

MySQL dapat berjalan stabil di berbagai sistem operasi seperti windows,

Linux, Unix. Proses migrasi data antar sistem operasi dapat dilakukan dengan mudah. Database MySQL memiliki dukungan terhadap *stored procedure*, fungsi, *trigger*, *view*, SQL standar ANSI, yang tentu mempermudah dan mempercepat proses pengembangan aplikasi.

2.9 JQuery

JQuery adalah perpustakaan bawaan JavaScript untuk mengotomatiskan dan menyederhanakan perintah umum. Banyak perpustakaan lain yang serupa, tetapi JQuery sejauh ini paling populer karena dapat menjalankan perintah di browser lama. JQuery berjalan di dalam browser bersama JavaScript biasa, terutama digunakan untuk animasi dan AJAX. Ini sangat sulit untuk diprogram dalam JavaScript biasa, tetapi JQuery memungkinkan mengetik dalam beberapa baris pendek. (Aziz et al., 2019)

JQuery masuk dalam sebuah laman web dengan tag `<script>` tag; Contohnya: `<script src="/path/to/jquery.js"></script>`. JQuery memiliki banyak sekali plugin dan memperluas fungsinya melalui berbagai metode.

Berikut ini kelebihan JQuery, antara lain :

1. Kompatibel dengan hampir seluruh browser.
2. JQuery telah digunakan oleh *website-website* raksasa.
3. Kompatibel dengan seluruh versi CSS (dari CSS 1 sampai CSS3).
4. Dokumentasi dan tutorialnya lengkap, bisa langsung dilihat di <http://jquery.com>.
5. Didukung oleh banyak komunitas.
6. Di-support oleh *plugin* yang lengkap.
7. *File*-nya hanya satu dan ukurannya relatif kecil, sekitar 20kb.
8. *Open source* atau *Free*.
9. JQuery lebih diminati oleh para *developer web* saat ini.
10. Mempermudah akses dan manipulasi elemen tertentu pada dokumen.
11. Mempermudah modifikasi/perubahan tampilan halaman web.
12. Mempersingkat Ajax (*Asynchronous JavaScript and XML*).
13. Memiliki API (*Application Programming Interface*).

14. Mampu merespons interaksi antara *user* dengan halaman web dengan lebih cepat.
15. Menyediakan fasilitas untuk membuat animasi sekelas *Flash* dengan mudah.

2.10 JavaScript

JavaScript adalah bahasa untuk menulis program yang membuat dokumen HTML yang ditampilkan di browser tidak hanya lebih indah tetapi juga lebih interaktif. JavaScript menyediakan beberapa fungsi pada halaman web dan memungkinkan program dilihat melalui antarmuka di web. (Aziz et al., 2019).

JavaScript adalah bahasa *scripting* yang tidak memerlukan kompiler untuk menjalankannya, cukup dengan interpreter. Tidak diperlukan proses kompilasi sebelumnya untuk menjalankan program. Proses kompilasi dilakukan terlebih dahulu agar program dapat dijalankan di browser.

Web Netscape Navigator dan *Internet Explorer* adalah contoh penerjemah karena kedua browser dilengkapi dengan penerjemah JavaScript. Namun, tidak semua browser web merupakan penerjemah JavaScript, karena browser belum tentu dilengkapi dengan penerjemah tersebut. JavaScript adalah bahasa skrip yang ringan dan mudah digunakan. Dengan JavaScript, website tidak hanya menjadi halaman data dan informasi saja, namun juga menjadi program aplikasi dengan antarmuka web.

JavaScript merupakan bahasa pemrograman yang tidak membutuhkan lisensi untuk menggunakannya. Jika *browser* web yang digunakan mendukung JavaScript maka dapat langsung membuat aplikasi berbasis web dengan menggunakan JavaScript. Umumnya program JavaScript adalah program yang disisipkan ke dalam web, sehingga halaman web menjadi aplikasi yang berjalan pada browser web.

Tidak semua sistem operasi menggunakan JavaScript, seperti sistem operasi MS Windows, menggunakan istilah *Windows Scripting Host* (WSH) sebagai interpreter JavaScript dan VBScript, program yang dibuat dengan JavaScript dan VBScript langsung dijalankan tanpa harus menggunakan *browser* web terlebih dahulu. *Browser* web dasarnya suatu *interpreter* menampilkan dokumen dalam

format khusus, dan menerjemahkan script berupa *tag* HTML. Teks yang diberi *tag* ditampilkan sesuai dengan format yang telah didefinisikan.

JavaScript merupakan program script yang dijalankan oleh interpreter yang dibangun pada web browser sehingga dapat menjalankan program JavaScript. Program JavaScript dimasukkan ke dalam dokumen HTML yang ditandai dengan tag yang diawali dengan `<script>` dan diakhiri dengan `</script>`. Program JavaScript ini akan berjalan jika browser web Anda memiliki penerjemah JavaScript. Hasil eksekusi ini juga biasanya berupa elemen dokumen HTML, sehingga hasil program ditampilkan sebagai bagian dari dokumen HTML lainnya, sehingga menghasilkan keseluruhan dokumen HTML yang dapat dilihat di browser Web. (Aziz et al., 2019)

Dokumen HTML mengandung program JavaScript di dalamnya yang ditampilkan dalam browser, jika dilihat dengan menggunakan menu *View Source* (dalam browser Internet Explorer) atau dengan menggunakan *View Page Source* (dalam browser Mozilla).

2.11 Unified Modelling Language (UML)

Pada tahun 1997, UML diluncurkan oleh Object Management Group (OMG). Rumbaugh, Ivar Jacobson, dan Grady Booch adalah pengembang utama UML. UML awalnya dimaksudkan sebagai bahasa desain untuk pengembangan aplikasi perangkat lunak komputer yang tersedia. Kini UML tidak hanya digunakan untuk perancangan perangkat lunak, tetapi juga untuk perancangan berbagai model. UML telah menjadi bahasa pemodelan standar karena tidak bergantung pada bahasa pemrograman komputer dan perangkat lunak pemodelan tertentu.

UML merupakan produk dari pendekatan berorientasi objek, yang biasanya dikontraskan dengan pendekatan matematis. Pendekatan matematis merepresentasikan menggunakan variabel dan fungsi keadaan untuk merepresentasikan model dan menyampaikan “budaya” sains. Pendekatan berorientasi objek, di sisi lain, mewakili model yang mencakup objek atau entitas yang ada baik secara fisik (manusia, lautan, pohon, dll) atau secara virtual (klub pemancing, kelompok penonton, dll).

UML merupakan bahasa yang sederhana dan grafis. UML memanfaatkan diagram untuk merepresentasikan model. Diagram-diagram yang dipakai dalam UML adalah diagram use case, kelas, sekuens, transisi state, aktivitas, dan komponen. Ragam perangkat lunak komputer menyediakan bisa dipakai untuk mengimplementasikan diagram-diagram UML. Berikut akan dibahas diagram use case, kelas, transisi state, dan sekuens yang lebih sering dipakai untuk merepresentasikan sebuah model (Purnomo, 2019)

Tujuan UML diantaranya adalah:

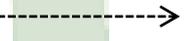
- 1) Memberikan model yang siap pakai, bahasa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.
- 2) Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa.
- 3) Menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan. (Zufria, 2013)

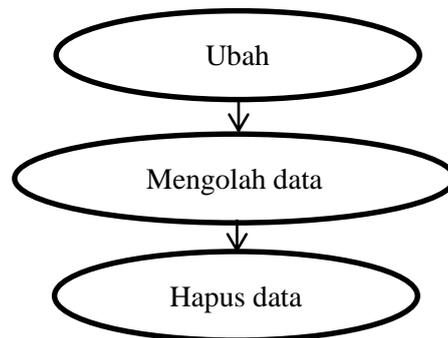
2.11.1 Use Case Diagram

Diagram *use case* merupakan model perilaku sistem informasi yang sedang dibangun. Sebuah *use case* menggambarkan interaksi antara satu atau lebih aktor dan sistem informasi yang sedang disiapkan. *Use case* digunakan untuk menentukan kapabilitas apa saja yang ada pada suatu sistem informasi dan siapa saja yang berhak menggunakan kapabilitas tersebut. Simbol untuk diagram *use case* ditunjukkan pada Tabel 1 di bawah ini:

Tabel 2.1 Simbol Use Case

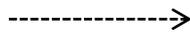
Simbol	Deskripsi
--------	-----------

<p><i>Use Case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal di awal frase nama use case.</p>
<p>Aktor / <i>actor</i></p>  <p>Nama aktor</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>
<p>Asosiasi / <i>association</i></p> 	<p>Komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan actor.</p>
<p>Ekstensi/extend</p> <p><<extends></p> 	<p>Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu mirip dengan prinsip inheritance pada pemrograman berorientasi objek; biasanya use case tambahan memiliki nama depan yang sama dengan use case yang ditambahkan misal Arah panah mengarah pada use case yang ditambahkan; biasanya use case yang menjadi extend-nya merupakan jenis yang sama dengan use case yang menjadi induknya.</p>
<p>Generalisasi/ <i>generalization</i></p>	 <p>Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya: arah panah mengarah pada use case yang menjadi generalisasinya (umum)</p>

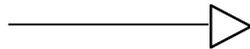


Menggunakan/ *include* / *uses* Relasi *use case* tambahan ke sebuah *use case* di mana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya atau sebagai syarat dijalankan *use case* ini ada dua sudut pandang yang cukup besar mengenai *include* di *use case*:

<<include>>



<<uses>>



1. Include berarti *use case* yang ditambahkan akan selalu dipanggil saat *use case* tambahan dijalankan, misal pada kasus berikut:

2. Include berarti *use case* yang tambahan akan selalu melakukan pengecekan apakah *use case* yang ditambahkan telah dijalankan sebelum *use case* tambahan dijalankan, misal pada kasus berikut:

Kedua interpretasi diatas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.

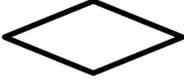
SUMATERA UTARA MEDAN

Sumber : S, Rosa A. dan Shalahuddin, M, (2018 : 162)

2.11.1 Activity Diagram

Diagram aktivitas menggambarkan alur kerja atau aktivitas suatu sistem, proses bisnis, atau menu perangkat lunak. Diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan oleh aktor. Di bawah ini adalah simbol diagram aktivitas yang ditunjukkan pada Tabel 2:

Tabel 2.2 Simbol Activity Diagram

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas terjadi.

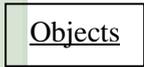
Sumber : S, Rosa A. dan Shalahuddin, M, (2018 : 163)

2.11.2 Sequence Diagram

Sequence diagram menggambarkan perilaku objek dalam use case dengan menggambarkan masa hidup objek dan pesan yang dikirim dan diterima antar objek. Oleh karena itu, untuk menggambar sequence diagram, perlu menggambar objek yang terlibat dalam *use case* dan metode kelas yang dipakai sebagai objek itu, juga dibuat diagram sequence untuk melihat skenario *use case*. (Rosa dan Shalahuddin : 2018)

Berikut adalah simbol-simbol activity diagram, seperti terlihat pada tabel 3 berikut ini:

Tabel 2.3 Simbol Sequence Diagram

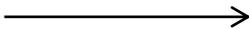
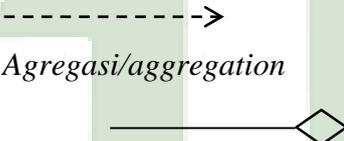
Nama Komponen	Keterangan	Simbol
<i>Lifeline</i>	<i>Lifeline</i> mengindikasikan keberadaan sebuah <i>object</i> dalam batas waktu. Notasi untuk <i>Lifeline</i> adalah garis putus-putus vertikal yang ditarik dari sebuah object	
<i>Activation</i>	<i>Activation</i> dinotasikan sebagai sebuah kotak segi empat yang digambar pada sebuah <i>lifeline</i> . Mengindikasikan sebuah obyek yang akan melakukan sebuah aksi	
<i>Message</i>	<i>Message</i> digambarkan dengan anak panah horizontal secara <i>Activation</i> Message mengindikasikan komunikasi antara <i>object-object</i>	
<i>Object</i>	Object merupakan <i>instance</i> dari sebuah <i>class</i> dan dituliskan tersusun secara <i>horizontal</i> . Digambarkan sebagai sebuah <i>class</i> (kotak) dengan nama object didalamnya yang diawali dengan sebuah titik koma	
<i>Actor</i>	<i>Actor</i> juga dapat berkomunikasi	

Sumber : Jurnal (Nurdam, 2014)

2.11.3 Class Diagram

Class diagram menggambarkan struktur sistem dalam kaitannya dengan definisi kelas yang dibuat untuk membangun sistem. Kelas memiliki atribut dan metode atau operasi. Simbol diagram kelas yang ditunjukkan pada Tabel 4 di bawah ini adalah:

Tabel 2.4 Simbol Class Diagram

Nama <i>interface</i>		Deskripsi
Kelas Nama kelas +atribut Operasi (O)		Kelas pada struktur sistem
Asosiasi / <i>association</i>		Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi berarah <i>directed association</i> 	/	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Generalisasi 		Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan <i>dependency</i>	/	Relasi antar kelas dengan makna kebergantungan antar kelas
 <i>Agregasi/aggregation</i>		Relasi antar kelas dengan makna semua- bagian(whole - part).

Sumber : Rosa dan Shalahuddin, M, 2018.

2.12 Bagan Alir (*Flowchart*)

Flowchart adalah suatu bentuk gambar/diagram dengan aliran yang berurutan dalam satu atau dua arah. Diagram alur digunakan untuk mewakili dan merancang program. Oleh karena itu, *flowchart* harus mampu mengekspresikan komponennya dalam bahasa pemrograman. *Flowchart* dan algoritma yang dibuat sebelum suatu program dibuat digunakan oleh pengembang program untuk menentukan alur logis dari program tersebut, dan *flowchart* serta algoritma yang dibuat setelah program dibuat digunakan oleh orang lain untuk menentukan alur logis dari program tersebut alur suatu program.

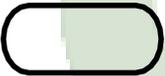
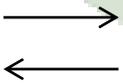
Tujuan dari diagram alur adalah dengan menggunakan simbol untuk menjelaskan tahapan penyelesaian masalah secara sederhana, rinci, jelas, dan ringkas (Lestari Pratiwi, 2020). Petunjuk yang harus diperhatikan dalam membuat

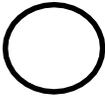
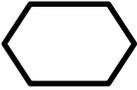
flowchart, seperti:

1. *Flowchart* digambarkan dari halaman atas ke bawah dan dari kiri kekanan.
2. Aktivitas yang digambarkan harus didefinisikan secara hati-hati dan definisi harus dapat dimengerti oleh pembacanya.
3. Kapan aktivitas dimulai dan berakhir harus ditentukan secara jelas dan terinci.
4. Setiap langkah dari aktivitas harus diuraikan dengan menggunakan deskripsi kata kerja.
5. Setiap langkah dari aktivitas harus pada urutan yang benar.
6. Lingkup aktivitas yang sedang digambarkan harus ditelusuri dengan hati-hati.
7. Gunakan simbol yang standar.

Simbol yang digunakan untuk menggambarkan algoritma dalam bentuk diagram alir yaitu :

Tabel 2.5 Simbol *Flowchart* dan Fungsinya

Simbol	Nama	Fungsi
	Terminator	Menyatakan awal/akhir suatu program.
	Garis Alir	Menyatakan jalannya arus/proses.
	<i>Input/Output</i> data	Menyatakan <i>input/output</i> suatu program.
	Proses	Menyatakan proses pengolahan data.
	<i>Decision</i>	Menunjukkan pilihan kondisi tertentu (ya/tidak)

	<i>On page connector</i>	Penghubung bagian <i>flowchart</i> pada satu halaman
	<i>Off page connector</i>	Penghubung bagian <i>flowchart</i> pada halaman berbeda
	Subprogram	Proses menjalankan subprogram
	<i>Preparation</i>	Pemberian nilai awal

Sumber : (Lestari Pratiwi, 2020)

2.13 Kulit Wajah

Menurut (Aristasari, 2021), kulit adalah salah satu bagian tubuh yang mendapatkan efek dari penuaan. Selain itu, kesehatan kulit saat ini menjadi perhatian sebagian besar manusia. Sama halnya dengan bagian tubuh lain, kulit juga bisa mengalami gangguan kesehatan. Gangguan kesehatan dapat disebabkan oleh faktor internal (hormon dan penyakit) dan eksternal (diet, lingkungan, gaya hidup). Kulit memiliki fungsi yang unik yang ada hubungannya dengan gizi, oleh karena itu merupakan *barier* yang kuat terhadap keluar masuknya cairan. Setiap orang dilahirkan dengan jenis kulit tertentu yang mencerminkan genetika dan etnisnya. Jenis kulit ditentukan berdasarkan berapa banyak minyak yang dihasilkan di folikel dari kelenjar sebaceous dan jumlah lemak yang ada di antara sel.

Jenis kulit diklasifikasikan menjadi empat: kulit normal, kulit kering, kulit berminyak, kulit kombinasi, dan (ditambah dengan) kulit sensitif. Khususnya untuk jenis kulit sensitif, semakin banyak hal yang harus dibahas untuk dijadikan pertimbangan apakah kulit sensitif ini bisa dikategorikan menjadi jenis atau kondisi kulit.

Perbedaan dari setiap jenis kulit wajah bergantung pada :

- 1) Seberapa banyak kadar cairan di dalam kulit yang memengaruhelastisitas kulit wajah.
- 2) Seberapa berminyak wajah.
- 3) Seberapa sensitif kulit wajah

Berikut ini beberapa jenis kulit pada wajah yang dapat diklasifikasikan antara lain :

1. Kulit Normal

Kulit normal yaitu kulit yang berada dalam kondisi seimbang, tidak terlalu berminyak dan tidak terlalu kering. Jenis kulit ini tidak banyak menunjukkan masalah ketika pemiliknya masih muda sehingga jenis kulit ini sering diabaikan keadaannya. Hal ini mengakibatkan timbulnya masalah, antara lain kerusakan kulit atau kulit menjadi kering, terutama pada daerah mata, pipi, dan leher sehingga kulit di daerah-daerah ini menjadi sangat tipis. Orang dewasa sangat jarang sekali memiliki jenis kulit normal (Windiyati & Tjahjono, 2019). Ciri-ciri atau karakteristik kulit normal yaitu:

- a. Ukuran pori-pori kecil atau medium.
- b. Kelembapan kulit bagus.
- c. Tekstur kulit halus; tidak terlalu tebal, juga tidak terlalu tipis.
- d. Kelenjar kulit bekerja secara normal, peredaran darah baik.
- e. Elastisitas kulit bagus (ketika kulit masih muda).
- f. Pigmentasi kulit warnanya terlihat halus.
- g. Jarang sekali mendapat gangguan jerawat.
- h. Warna kulit sehat (karena sirkulasi darah yang bagus).
- i. Kulit terasa kencang ketika disentuh. Kulit biasanya bebas dari kerusakan.

2. Kulit Kering

Kulit kering adalah kulit yang memiliki jumlah sebum yang terbatas sehingga sering kali kelembapan kulit berkurang dengan cepat. Keadaan ini disebut dehidrasi kulit. Dan, pada kulit yang demikian ini sering terdapat milia disekitar mata dan area lain yang benar-benar kering. Ciri-ciri atau karakteristikkulit kering yaitu :

- a. Pori-pori kecil dan ketat.
- b. Kelembapan kulit tidak baik.
- c. Tekstur kulit kasar dan tipis; kulit tampak seperti berpetak-petak kecil dan berserpih.
- d. Pembuluh darah kapiler banyak yang pecah.
- e. Mengalami penuaan dini, menyebabkan kerutan terlihat jelas pada daerah mata, mulut, dan leher.
- f. Pigmentasi kulit yang tidak rata seperti ephelides (bintik-bintik) biasanya menyertai kulit.

Faktor- faktor yang memperburuk keadaan kulit kering :

- a. Lingkungan (polusi, pendinginan udara, dan paparan suhu yang ekstrem, yaitu suhu yang terlalu panas atau terlalu dingin).
- b. Paparan yang berlebihan dari UV.
- c. Perawatan kulit yang tidak benar.

3. Kulit Berminyak

Kulit berminyak terjadi karena kelenjar *sebaceous* sangat aktif pada saat pubertas karena distimulasi oleh hormon pria (*androgen*). Sementara, meningkatnya produksi sebum seringkali mengakibatkan penampilan kulit menjadi tidak baik. Namun begitu, aktivitas kelenjar *sebaceous* biasanya mulai menurun ketika menginjak usia 20-an tahun (Windiyati & Tjahjono, 2019). Ciri- ciri atau karakteristik kulit berminyak yaitu :

- a. Pori-pori besar.
- b. Kelembapan kulit sangat tinggi
- c. Kulit kasar dan tebal.
- d. Sel kulit mati menutupi sebum sehingga peredaran darah dan peredaran getah bening pada kulit tidak lancar.
- e. Kekenyalan kulit bagus selama kulit dijaga dengan baik dari kelebihan sebum.
- f. Kemungkinan terjadi pigmentasi yang tidak merata.

- g. Warna kulit pucat dan mengilat akibat adanya sebum yang tersisa karena produksi yang berlebihan.
- h. Memiliki beberapa kelainan atau beberapa penyakit kulit yang sering terjadi, misalnya timbulnya komedo, *pustule*, *papul*, *milla*, atau kista *sebacea*.

Faktor-faktor yang memperburuk kulit berminyak :

- a. Pemakaian produk perawatan kulit yang kasar.
- b. Diet/pola makan yang buruk.
- c. Pembersihan yang tidak cukup.
- d. Perawatan kulit yang tidak sesuai.

4. Kulit Kombinasi

Jenis kulit ini adalah perpaduan antara jenis kulit kering dan kulit berminyak. Bagian yang berminyak umumnya terdapat pada daerah dagu, hidung, dan dahi, yang dikenal sebagai *T-Zone* atau "Daerah-T". Bagian atas pipi ditunjukkan dengan tanda-tanda berminyak, tetapi pada sebagian wajah dan leher adalah kering. Ciri-ciri atau karakteristik kulit kombinasi yaitu:

- a. Pori-pori pada daerah *T-Zone* besar, namun pada daerah pipi pori-porinya kecil atau sedang .
- b. Pada daerah yang berminyak kelembapannya tinggi, tetapi pada daerah yang kering kelembapan kulitnya tidak bagus.
- c. Kulit kasar dan tebal pada daerah yang berminyak, sedangkan pada daerah yang kering kulit tampak tipis.
- d. Kulit tampak pucat pada daerah yang berminyak tetapi pada daerah yang kering kulit tampak sensitif.
- e. Kekenyalan kulit pada daerah yang berminyak baik, tetapi sangat buruk pada kulit yang kering
- f. Pigmentasi yang terjadi tidak rata, biasanya terlihat *ephelides* dan *lentigo*.
- g. Pada *T-Zone* atau pada daerah yang berminyak tampak terjadi kelainan kulit seperti *pustul* dan komedo.
- h. *Milia* dan pembuluh darah kapiler yang rusak akan tampak pada daerah yang kering, biasanya pada pipi dan dekat daerah mata (Windiyati & Tjahjono,

2019).

2.14 Penelitian Terdahulu

Penelitian terdahulu ini menjadi salah satu acuan bagi penulis dalam melakukan penelitiannya dan memperkaya teori yang digunakan penulis dalam mengulas penelitian yang telah dilakukan. Pada penelitian-penelitian sebelumnya, tidak ditemukan penelitian yang sesuai dengan nama penelitian penulis. Namun demikian, penulis menggunakan beberapa penelitian sebagai referensi untuk memperkaya bahan penelitian penulis. Tabel 3.1 menampilkan artikel penelitian terdahulu berupa beberapa jurnal yang berkaitan dengan penelitian yang dilakukan penulis.

Tabel 2.6 Penelitian Terdahulu

No.	Judul	Sumber	Hasil Penelitian	Kelebihan & Kekurangan
1.	Kombinasi Metode Certainty Factor dan Forward Chaining untuk Identifikasi Jenis Kulit Wajah Berbasis Android	Syavira Cahyaningsih, Agung Triayudi, dan Ira Diana Sholihati (2021)	Mampu mempermudah dalam mengetahui jenis kulit wajah, hasil diagnosa dari perhitungan menggunakan aplikasi sistem pakar dan hasil perhitungan manual dari salah satu data reponden dari 100 data responden yaitu menghasilkan tingkat keyakinan yang sama, masing-masing menghasilkan persentase sebesar 99.45% dan hasil diagnosa menyatakan bahwa pengguna memiliki jenis kulit normal.	Kelebihannya adalah dalam aplikasi ini kombinasi metode certainty factor dengan teknik forward chaining sangat mungkin untuk diterapkan pada sistem pakar dalam upaya membantu pengguna untuk mengetahui jenis kulit wajah. Kekurangan adalah jurnal tidak memaparkan tujuan dari jurnal tersebut dan pada jurnal ini hanya menampilkan jenis kulit serta solusi bahan dalam skincare tapi tidak menampilkan saran produk yang sesuai.

2.	Sistem Pakar Untuk Mengidentifikasi Jenis Kulit Wajah dengan Metode <i>Certainty Factor</i>	Indyah Hartami Santi dan Bina Andari (2019)	Mengidentifikasi Jenis Kulit Wajah dengan memasukkan solusi perawatan. Metode yang digunakan Adalah Metode <i>Certainty Factor</i> yang dapat memilih berdasarkan pertimbangan dalam proses perhitungan, dan metode ini mencari kombinasi nilai kepercayaan tertinggi.	Kelebihannya solusi diberikan dengan membangun sistem pakar yang dapat mengidentifikasi jenis kulit wajah yang berbeda dengan menginput solusi perawatannya. Metode koefisien determinasi dipilih berdasarkan pertimbangan dalam proses perhitungan dan mencari kombinasi yang nilai keyakinannya paling tinggi. Pada awal penelitian, pengumpulan data dilakukan terhadap 40 responden perempuan, termasuk 100% responden. Kekurangannya adalah dalam aplikasi ini hanya menampilkan jenis kulit beserta solusinya saja, tidak ada memaparkan permasalahan dan rekomendasi produknya.
3.	Perancangan Sistem Pakar Penentuan Jenis Kulit Wajah Menggunakan Metode <i>Certainty Factor</i>	Rangga Pebrianto, Siti Nurhasanah Nugraha, dan Windu Gata (2019)	Aplikasi sistem pakar berbasis android, konsultan dapat melakukan pemeriksaan dengan mudah karena hanya membuka aplikasi android, dan pendiagnosaan dengan metode <i>certainty factor</i> ini	Kelebihannya adalah aplikasi ini konsultan dapat melakukan pemeriksaan dengan mudah karena hanya membuka aplikasi android. Pendiagnosaan dengan metode <i>certainty factor</i> ini dapat memberikan hasil diagnosa serta dapat

			dapat memberikan solusi produk yang tepat berdasarkan jenis kulit yang diagnosa oleh sistem.	memberikan solusi produk yang tepat berdasarkan jenis kulit yang di diagnosa oleh sistem. Kekurangannya adalah dalam aplikasi ini hanya menampilkan jenis kulit beserta solusinya saja, tidak ada memaparkan permasalahan dan rekomendasi produknya
4.	Sistem Pakar Diagnosa Penyakit Kulit Menggunakan Metode <i>Forward Chaining</i>	Anindita Dhiaksa (2016)	Mengenal dan mengatasi suatu gejala penyakit tanpa harus datang ke puskesmas untuk mengantri dan dapat mencegah gejala penyakit tersebut secara cepat. Hasil output sistem pakar sudah sesuai dengan kebutuhan pasien saatberkonsultasi.	Kelebihannya adalah hasil output sistem pakar sudah sesuai dengan kebutuhan pasien saat berkonsultasi dan secara manual hasil analisa sudah sesuai dengan aturan rule yang dibuat. Kekurangannya adalah belumada forum dimana user dapat chat dengan dokter apabila ada beberapa hal yang ingin ditanyakan perihal konsultasi.