

Penelitian

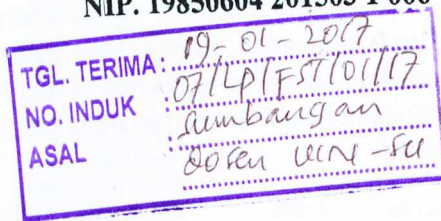
**PERANCANGAN APLIKASI PENATAAN LETAK BARANG
PADA MINIMARKET MENGGUNAKAN METODE
APRIORI**



P
729.24
ZUF
8
E.1

Oleh :

ILKA ZUFRIA, M.Kom
NIP. 19850604 201503 1 006



**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
MEDAN
2017**

KATA PENGANTAR

Bismillahirrahmanirrahim...

Dengan segala kerendahan hati, penulis sampaikan puji syukur kepada Allah SWT berkat Rahmat dan Hidayah-Nya memberi kesehatan, pengetahuan dan kesempatan kepada penulis sehingga dapat menyelesaikan penelitian ini yang berjudul **“Perancangan Aplikasi Penataan Letak Barang Pada Minimarket Dengan Metode Apriori”**.

Dalam menyelesaikan penelitian ini banyak bantuan bimbingan dari berbagai pihak, baik berupa materil, spiritual, maupun informasi. Sehingga penelitian ini dapat diselesaikan. Maka selayaknya penulis mengucapkan terima kasih kepada :

1. Bapak Dr. H. M. Jamil, MA selaku Dekan Fakultas Sains dan Teknologi UIN Sumatera Utara Medan
2. Ibu Dr. Rina Filia Sari, M.Si selaku Wakil Dekan I Fakultas Sains dan Teknologi UIN Sumatera Utara Medan
3. Bapak M.Irwan Padli Nst, MM, M,Kom selaku Kaprodi Sistem Informasi Fakultas Sains dan Teknologi UIN Sumatera Utara Medan sekaligus Konsultan pada penelitian ini
4. Bapak/ibu rekan-rekan dosen tetap Fakultas Sains dan Teknologi UIN Sumatera Utara Medan

Menyadari kekurangan dan keterbatasan pada penelitian ini, maka penulis tetap mengharapkan kritik dan saran dari berbagai pihak agar penelitian ini bisa dikembangkan dikemudian hari. Akhir kata semoga penelitian ini bermanfaat bagi kita semua dan Semoga Allah SWT berkenan memberikan berkahnya sehingga semua harapan dan cita-cita penulis dapat terkabulkan. Amin

Medan, Januari 2017

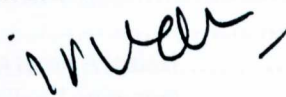
Ilka Zufria, M.Kom

REKOMENDASI

Setelah membaca dan menelaah hasil penelitian yang berjudul **“Perancangan Aplikasi Penataan Letak Barang Pada Minimarket Dengan Metode Apriori”**. Yang dilakukan oleh Ilka Zufria, M.Kom maka saya berkesimpulan bahwa hasil penelitian ini dapat diterima sebagai karya tulis berupa hasil penelitian. Demikianlah rekomendasi diberikan kepada yang bersangkutan untuk dapat dipergunakan sebagaimana mestinya.

Medan, Januari 2017

Konsultan



M. Irwan Padli Nst, MM, M.Kom
NIP. 19750213 200604 1 003

DAFTAR ISI

KATA PENGANTAR	ii
REKOMENDASI	iii
DAFTAR ISI	iv
DAFTAR TABEL	vi
DAFTAR GAMBAR	vii
ABSTRAK	viii
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan dan Manfaat Penelitian.....	3
1.4.1 Tujuan Penelitian.....	3
1.4.2 Manfaat Penelitian.....	3
1.5 Sistematika Tulisan.....	4
BAB II TINJAUAN PUSTAKA	6
2.1 Aplikasi	6
2.2 Data Mining.....	7
2.3 Algoritma Apriori.....	14
2.3.1 Pembentukan Aturan Asosiasi.....	16
2.4 UML (<i>Unified Modeling Language</i>).....	18
2.4.1 <i>Usecase Diagram</i>	20
2.4.2 <i>Class Diagram</i>	25
2.4.3 <i>Activity Diagram</i>	29
2.5 Aplikasi Web.....	31
2.6 PHP.....	32
2.7 MySQL Database.....	36
BAB III METODOLOGI PENELITIAN	38
3.1 Kerangka Kerja Penelitian.....	38
3.2 Objek Penelitian.....	40
3.3 Definisi Operasional Variabel.....	40
3.4 Tehnik Pengumpulan Data.....	41
3.5 Metode Analisis Data.....	42

BAB IV	ANALISIS DAN PEMBAHASAN.....	45
4.1	Analisis Kebutuhan Sistem.....	45
4.1.1	Analisa Kebutuhan Perangkat Keras (<i>Hardware</i>).....	45
4.1.2	Analisa Kebutuhan Perangkat Lunak (<i>Software</i>).....	45
4.2	Perancangan Sistem Dengan UML.....	46
4.2.1	<i>UseCase</i>	46
4.2.2	<i>Activity Diagram</i>	51
4.2.3	<i>Class Diagram</i>	53
4.3	Analisis Sistem.....	54
4.4	Perancangan Sistem.....	60
4.5	Implementasi Sistem.....	70
BAB V	KESIMPULAN DAN SARAN.....	77
5.1	Kesimpulan.....	77
5.2	Keterbatasan.....	77
5.3	Rekomendasi.....	77

DAFTAR PUSTAKA

DAFTAR TABEL

Table 2.1	Tabel Data Contoh Estimasi.....	12
Table 2.2	Simbol Usecase Diagram.....	23
Table 2.3	Simbol – simbol <i>Class</i> Diagram	27
Table 2.4	Simbol – simbol <i>Activity</i> Diagram.....	30
Table 2.5	Daftar Operator Penugasan.....	35
Table 2.6	Operator Pembanding.....	36
Tabel 3.1	Transaksi.....	42
Tabel 3.2	Calon 2-itemset.....	43
Tabel 3.3	Aturan Asosiasi Final.....	44
Tabel 4.1	Deskripsi Aktor yang Diusulkan.....	48
Tabel 4.2	Defenisi <i>Usecase</i>	49
Tabel 4.3	Data Transaksi Pembelian Barang.....	54
Tabel 4.4	Perhitungan <i>Support</i> Untuk Candidat 1-itemset... ..	56
Tabel 4.5	Perhitungan <i>Support</i> Untuk 1-Itemset.....	57
Tabel 4.6	Perhitungan <i>Frequent 1-itemset</i>	58
Tabel 4.7	Perhitungan <i>Frequent 2-itemset</i>	58
Tabel 4.8	Hasil Kategori.....	59

DAFTAR GAMBAR

Gambar 2.1	Logo PHP.....	33
Gambar 2.2	Logo MySQL.....	37
Gambar 3.1	Kerangka Kerja Penelitian.....	38
Gambar 4.1	<i>Usecase Diagram</i>	47
Gambar 4.2	<i>Activity Diagram</i>	52
Gambar 4.3	<i>Class Diagram</i>	53
Gambar 4.4	Perancangan Login.....	60
Gambar 4.5	Perancangan Menu Utama.....	61
Gambar 4.6	Tampil Barang.....	62
Gambar 4.7	Input Barang.....	63
Gambar 4.8	Tampil Transaksi.....	64
Gambar 4.9	Tambah Transaksi.....	65
Gambar 4.10	Data Minimum.....	66
Gambar 4.11	Input <i>Support</i> dan <i>Minimum Confidence</i> ...	67
Gambar 4.12	Hasil Analisis Apriori.....	68
Gambar 4.13	Hasil Kategori.....	69
Gambar 4.14	Halaman Menu Utama.....	70
Gambar 4.15	Login.....	71
Gambar 4.16	Data Barang.....	71
Gambar 4.17	Halaman Tampil Barang.....	72
Gambar 4.18	Halaman Tambah Barang.....	73
Gambar 4.19	Halaman Tambah Transaksi.....	74
Gambar 4.20	Tambah Minimum Support.....	74
Gambar 4.21	Hasil Analisa Apriori.....	75
Gambar 4.22	Hasil Kategori.....	76

ABSTRAK

Meningkatnya penjualan pada market-market menjadikan barang yang dijual sangat banyak dan berbagai jenis barang juga disediakan. Pemilik Mini Market mengalami kesulitan untuk melakukan proses tata letak barang yang sering dibeli oleh konsumen secara bersamaan.

Pada data mining, algoritma apriori merupakan salah satu solusi pemecahan masalah yaitu mengetahui barang yang dibeli secara bersamaan oleh konsumen. Dalam algoritma apriori penentuan nilai support sangat menentukan barang yang sering dibeli bersamaan. Penulis harus menetapkan terlebih dahulu nilai support minimal yang dijadikan acuan.

Kata kunci : Penjualan, Barang, Konsumen, Algoritma, Apriori, Support

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Teknologi saat ini berkembang pesat, hampir seluruh kehidupan manusia menggunakan teknologi komputer. Salah satu aspek yang tidak terlepas dari teknologi komputer adalah aspek perekonomian. Salah satu contoh mini market merupakan usaha perekonomian menengah masyarakat. Yang paling sering teknologi.

Mini market menjual berbagai jenis barang mulai barang sandang hingga makanan cemilan. Mini market meletakkan barangnya terkadang tidak pada tempatnya misalnya makanan diletakkan dekat sabun, coklat diletakkan dekat es krim dan lain sebagainya. Pengaturan letak barang yang sesuai sesuai dengan kebutuhan masyarakat sangatlah penting, agar masyarakat dapat dengan mudah mengambil barang yang sangat diperlukan. Perlunya barang diletakkan sesuai pada tempatnya agar mempermudah konsumen untuk mencari barang yang dibutuhkan.

Algoritma apriori adalah algoritma yang digunakan untuk menghasilkan *association rule* dengan pola *if then*, dimana *k-itemset* digunakan untuk mengeksplorasi *itemset* dengan melakukan pendekatan *iterative* yang dikenal dengan

pencarian *level-wise* (¹Robiyanto, 2015). Dalam penelitian ini dibuat analisis asosiasi dengan algoritma *asosiasi* untuk menata barang sesuai dengan tempatnya pada mini market. Berdasarkan uraian latar belakang tersebut, penulis melakukan penelitian dengan judul "***Perancangan Aplikasi Penataan Letak Barang Pada Minimarket Dengan Metode Apriori .***" Dengan pendekatan algoritma apriori ini penataan barang di minimarket dapat tertata dengan baik sehingga konsumen dengan mudah mencari barang kebutuhan yang mereka cari.

1.2 Rumusan Masalah

Adapun masalah yang dapat dirumuskan dalam penelitian ini antara lain :

- a. Bagaimana mengelola tata letak barang pada minimarket dengan memanfaatkan sebuah aplikasi?
- b. Bagaimana menerapkan metode *apriori* dalam aplikasi tata letak barang minimarket.
- c. Bagaimana aplikasi ini dapat membantu minimarket dalam mengefisienkan proses tata letak barangnya

1.3 Batasan Masalah

¹ Robiyanto, (2015). "*Implementasi Data Mining dengan Metode Algoritma Apriori dalam Menentukan Pola Pembelian Obat*" hal 105.

Agar penelitian ini lebih terarah dan tujuan yang diharapkan dapat tercapai penulis menentukan batasan-batasan masalahnya sebagai berikut :

- a. Aplikasi ini digunakan untuk penataan tata letak barang di di minimarket.
- b. Aplikasi ini menggunakan bahasa pemrograman PHP dan database MySQL.
- c. Aplikasi ini menerapkan algoritma *apriori*.

1.4 Tujuan dan Manfaat Penelitian

1.4.1 Tujuan Penelitian

- a. Menerapkan algoritma *apriori* dalam aplikasi tata letak barang pada minimarket.
- b. Mengembangkan sistem baru dengan mengganti sistem manual dalam penataan barang di minimarket saat ini.

1.4.2 Manfaat Penelitian

- a. Mempermudah mini market dalam penataan barang dan tata letak barangnya
- b. Mempermudah pemilik mini market mengetahui jenis barang yang sering terjual.
- c. Mengefisiensikan proses tata kelola dan tata letak barang dari berdasarkan daya minat konsumen pada minimarket

1.5 Sistematika Penulisan

Sistematika penulisan laporan penelitian ini terdiri dari 5 bab yaitu :

BAB I PENDAHULUAN

Bab ini berisi latar belakang masalah, rumusan masalah, batasan masalah, tujuan dan manfaat penelitian serta sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini berisi teori-teori untuk menyusun penelitian ini. Teori yang digunakan harus sesuai dengan penelitian.

BAB III METODOLOGI PENELITIAN

Bab ini berisi kerangka kerja dan mendefinisikan ruang lingkup masalah.

BAB IV ANALISIS DAN PEMBAHASAN

Bab ini berisi tentang analisis masalah, perancangan aplikasi dan pengujian aplikasi yang telah dibangun.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan secara singkat terhadap hasil penelitian, kelemahan-kelemahan terhadap

aplikasi yang penulis buat sehingga dapat diperbaiki di kemudian hari. Di bab ini juga berisi masukan bagi mahasiswa yang lain untuk mengembangkan penelitian dengan topik yang sama.

LAMPIRAN

BAB II

TINJAUAN PUSTAKA

2.1 Aplikasi

Perangkat lunak aplikasi (*software application*) adalah suatu program yang menentukan aktivitas pemrosesan informasi yang dibutuhkan untuk penyelesaian tugas-tugas khusus pemakai komputer (Ruslan, 2013²). Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer, tapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media.

Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau suite aplikasi (*application suite*) (Ruslan, 2013²). Contohnya adalah *Microsoft Office* dan *OpenOffice.org*, yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya. Aplikasi-aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan tiap aplikasi. Sering kali, mereka memiliki kemampuan untuk saling berinteraksi satu sama lain sehingga

² Ruslan, (2013). "*Aplikasi Pengolahan Data Karyawan Dengan Pendekatan Microsoft Visual*", Palembang.

menguntungkan pengguna. Contohnya, suatu lembar kerja dapat dibenamkan dalam suatu dokumen pengolah kata walaupun dibuat pada aplikasi lembar kerja yang terpisah.

Aplikasi dapat digolongkan menjadi beberapa kelas, antara lain (www.wikipedia.com/aplikasi):

1. Perangkat lunak perusahaan (*enterprise*)
2. Perangkat lunak infrastruktur perusahaan
3. Perangkat lunak informasi kerja
4. Perangkat lunak media dan hiburan
5. Perangkat lunak pendidikan
6. Perangkat lunak pengembangan media
7. Perangkat lunak rekayasa produk

Pada pengertian umumnya, aplikasi adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya aplikasi merupakan suatu perangkat komputer yang siap pakai bagi user.

2.2 Data Mining

Analisis asosiasi atau *association rule mining* adalah teknik *data mining* untuk menemukan aturan asosiasi antara suatu kombinasi *item* (Robiyanto, 2015:105³). *Interestingness measure* yang dapat digunakan dalam *data mining* adalah:

³ Robiyanto, (2015). "Implementasi Data Mining dengan Metode Algoritma Apriori dalam Menentukan Pola Pembelian Obat" hal.105

a. *Support*, adalah suatu ukuran yang menunjukkan seberapa besar tingkat dominasi suatu *item* atau *itemset* dari keseluruhan transaksi.

b. *Confidence*, adalah suatu ukuran yang menunjukkan hubungan antar dua *item* secara *conditional* (berdasarkan suatu kondisi tertentu).

Istilah data mining memiliki beberapa padanan seperti *knowledge discovery* ataupun *pattern recognition*. Istilah *knowledge discovery* atau penemuan pengetahuan tepat digunakan karena tujuan utama dari data mining memang untuk mendapatkan pengetahuan yang masih tersembunyi dalam bongkahan data. Istilah *pattern recognition* atau pengenalan pola pun tepat untuk digunakan karena pengetahuan yang hendak digali memang berbentuk pola-pola yang mungkin juga masih perlu digali dari dalam bongkahan data yang tengah dihadapi.

Jadi, data mining dapat dikatakan disiplin ilmu yang tujuan utamanya adalah untuk menemukan, menggali, atau menambang pengetahuan dari data atau informasi yang kita miliki.

Terdapat 6 fungsi dalam data mining yaitu :

1. Fungsi deskripsi (*description*)
2. Fungsi estimasi (*estimation*)

3. Fungsi prediksi (*prediction*)
4. Fungsi klasifikasi (*classification*)
5. Fungsi pengelompokkan (*classification*)
6. Fungsi asosiasi (*Association*)

Keenam fungsi data mining tersebut dapat dipilah menjadi Fungsi minor atau fungsi tambahan yang meliputi ketiga fungsi yang pertama, yaitu deskripsi, estimasi dan prediksi.

1. Fungsi mayor atau fungsi utama yang meliputi ketiga fungsi yaitu klasifikasi, pengelompokkan dan asosiasi.
2. Data mining dapat diartikan sebagai proses pengekstrakan informasi yang implisit dan pengetahuan potensial yang bermanfaat bagi orang-orang yang tidak mengetahui sebelumnya (Hemlata Sahu, volume 1, 114⁴).

Data mining juga populer dengan istilah *Knowledge Discovery In Database* (KDD). Adapun proses KDD adalah sebagai berikut :

1. *Data Selection* : pemilihan data dari sekumpulan data operasional perlu dilakukan sebelum tahap penggalian informasi dalam KDD dimulai.

⁴Hemlata Sahu (2015). " *A Brief Overview on Data Mining Survey*" hal.114

2. *Preprocessing* : sebelum proses data mining dapat dilaksanakan, perlu dilakukan proses *cleaning* dengan tujuan untuk membuang duplikasi data, memeriksa data yang inkonsisten, dan memperbaiki kesalahan pada data, seperti kesalahan cetak. Juga dilakukan proses *enrichment*, yaitu proses memperkaya data yang sudah ada dengan data atau informasi lain yang relevan dan diperlukan untuk KDD, seperti data atau informasi eksternal.
3. *Transformation* : yaitu proses *coding* pada data yang telah dipilih, sehingga data tersebut sesuai untuk proses *data mining*. Proses *coding* dalam KDD merupakan proses kreatif dan sangat tergantung pada jenis atau pola informasi yang akan dicari dalam database.
4. *Data mining* : proses mencari pola atau informasi menarik dalam data terpilih dengan menggunakan teknik atau metode tertentu.
5. *Interpretation / evaluation* : pola informasi yang dihasilkan dari proses data mining perlu ditampilkan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan. Tahap ini merupakan bagian dari proses KDD yang disebut dengan *interpretation*. Tahap ini mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesa yang ada sebelumnya atau tidak.

Di dalam jurnal nasional data mining diartikan proses yang menggunakan teknik statistik, matematika, kecerdasan

buatan, dan *machine learning* untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari *database* yang besar (Hendro, 2011:51⁵). Dalam jurnal nasional yang lain, data mining diartikan sebagai pengekstrakan informasi baru yang diambil dalam bongkahan data besar yang membantu dalam pengambilan keputusan (Nurhayati, 2014:95⁶).

Di dalam jurnal internasional data mining merupakan proses analisa yang dirancang untuk mengeksplorasi data dalam jumlah yang besar dan mencari pola yang konsisten dan hubungan yang sistematis antara variabel untuk membentuk *subset* data yang baru .

Menurut (Kohail, 2011:96⁷) "Data mining dimanfaatkan untuk membangun berbagai aplikasi seperti aplikasi meteorologi yang dapat digunakan untuk memprediksi angin topan, banjir, dan lain-lain".

Ketika diberi sekumpulan data, terkadang agak sulit menangkap arti kumpulan data tersebut. Sekumpulan angka-

⁵ Tacbir Hendro, (2011). "*Penerapan Data Mining Untuk Menganalisa Kemungkinan Pengunduran Diri Calon Mahasiswa Baru*", hal 51

⁶ Nurhayati, (2014). "*Metode Rough Set Untuk Melihat Perilaku Suami Yang Menjadi Akseptor Kb Vasektomi*." hal.95-96

⁷ Kohail, (2011). "*Implementation of Data Mining Techniques for Meteorological Data Analysis*" hal.96

angka tersebut perlu dirangkum sedemikian rupa agar dapat “berbicara” sehingga kita memiliki gambaran mengenai kumpulan data tersebut. Bayangkan seandainya data tersebut sebanyak 10.000. Untuk melakukan penelitian tersebut, tentu memerlukan banyak orang yang dipercaya. Menelusuri deretan angka sebanyak itu dengan mata manusia tentu tidak akan membawa manfaat apa pun. Oleh karena itu, kita memerlukan cara untuk menggambarkan sekumpulan data secara ringkas. Cara tersebut dinamakan deskripsi.

Tabel 2.1 Tabel Data Contoh Estimasi

Nomor Botol	Volume Air (ml)
1	2016
2	2025
3	1968
4	2007
5	2031
6	2055
7	2039
8	1981

(Buku Data Mining Karangan Susanto)

Volume yang sebenarnya diharapkan untuk setiap botol adalah 2 liter (2.000 ml). Data di atas menunjukkan bahwa mesin pengisi otomatis tersebut tidak mengisi tiap botol dengan volume air yang tepat sama seperti yang diharapkan. Jadi, untuk menjawab berapa volume air yang ada di dalam botol adalah dengan estimasi. Sebelum mengolah data tersebut, harus memperkirakan rata-rata dan standar deviasi volume air yang diisikan oleh mesin tersebut. Kegiatan memperkirakan ini disebut estimasi. Berbeda dengan prediksi, prediksi merupakan memprediksikan hasil dari sesuatu yang baru yang akan muncul selanjutnya. Jadi, prediksi memperkirakan hasil yang belum terjadi.

Di dalam jurnal internasional lainnya dikatakan bahwa data mining juga disebut *knowledge discovery in database* (KDD) adalah bidang menemukan informasi dan berpotensi berguna dari data dalam jumlah yang besar. Data mining telah diterapkan dalam sejumlah besar bidang seperti penjualan tiket, bioinformatika, dan kontra terorisme.

Pada jurnal nasional lainnya diterangkan bahwa data mining atau sering disebut *knowledge discovery in database* (KDD) adalah kegiatan yang meliputi pengumpulan, pemakaian data historis untuk menemukan keteraturan, pola atau hubungan dalam data berukuran besar. Keluaran data mining ini bisa dipakai untuk membantu pengambilan keputusan di masa depan.

Dari defenisi yang telah di jalaskan di atas, dapat disimpulkan bahwa :

1. Data mining adalah melakukan ekstraksi informasi tersembunyi dari database besar dan dapat dimanfaatkan untuk kepentingan di masa yang akan datang.
2. Data mining merupakan pengumpulan, pemakaian data historis untuk menemukan keteraturan, pola atau hubungan dalam data berukuran besar.

2.3 Algoritma Apriori

Algoritma Apriori adalah algoritma pengambilan data dengan aturan asosiatif (*Association rule*) untuk menentukan hubungan asosiatif suatu kombinasi item (Robiyanto, 2015:105⁸).

Mencari kombinasi *item* yang memenuhi syarat minimum dari nilai *support* dalam basis data. Nilai *support* sebuah *item* diperoleh dengan menggunakan rumus berikut:

$$\text{Support (A)} = \frac{\text{Jumlah transaksi mengandung A}}{\text{Total Transaksi}} \quad (1)$$

Nilai *support* dari 2 *item* diperoleh dengan menggunakan rumus:

$$\text{Support (A,B)} = P(A \cap B)$$

⁸ Robiyanto, (2015). "Implementasi Data Mining dengan Metode Algoritma Apriori dalam Menentukan Pola Pembelian Obat" hal.105

$Support(A,B) = \Sigma \text{transaksi mengandung } A \text{ dan } B \Sigma \text{transaksi}$ (2)

Algoritma Apriori Apriori adalah suatu algoritma yang sudah sangat dikenal dalam melakukan pencarian frequent itemset dengan menggunakan teknik association rule (Erwin, 2009:26). Algoritma *Apriori* menggunakan *knowledge* mengenai *frequent itemset* yang telah diketahui sebelumnya, untuk memproses informasi selanjutnya. Pada algoritma *Apriori* untuk menentukan kandidat-kandidat yang mungkin muncul dengan cara memperhatikan minimum support. Adapun dua proses utama yang dilakukan dalam algoritma *Apriori*, yaitu :

1. *Join* (penggabungan).

Pada proses ini setiap item dikombinasikan dengan item yang lainnya sampai tidak terbentuk kombinasi lagi.

2. *Prune* (pemangkasan).

Pada proses ini, hasil dari item yang telah dikombinasikan tadi lalu dipangkas dengan menggunakan minimum support yang telah ditentukan oleh user. Dua proses utama tersebut merupakan langkah yang akan dilakukan untuk mendapat *frequent itemset*,

Walaupun algoritma *Apriori* mudah untuk dipahami dan diimplementasikan dibandingkan dengan algoritma yang lainnya yang memang diterapkan untuk proses *association rule*, akan tetapi algoritma *Apriori* juga memiliki kekurangan yaitu, untuk melakukan pencarian *frequent itemset*, algoritma *Apriori* harus melakukan *scanning database* berulang kali untuk setiap

kombinasi item. Hal tersebut menyebabkan banyaknya waktu yang dibutuhkan untuk melakukan *scanning database*.

2.3.1. Pembentukan Aturan Asosiasi

Setelah semua pola frekuensi tinggi ditemukan, barulah dicari aturan asosiasi yang memenuhi syarat minimum untuk *confidence* dengan menghitung *confidence* aturan asosiatif A U B. Nilai *confidence* dari aturan A U B diperoleh dengan rumus berikut.

$$\text{Confidence} = P(B|A) = \frac{\Sigma \text{transaksi mengandung A dan B}}{\Sigma \text{Transaks mengandung A}} \quad (3)$$

Untuk menentukan aturan asosiasi yang akan dipilih maka harus diurutkan berdasarkan *Support* × *Confidence*. Aturan diambil sebanyak *n* aturan yang memiliki hasil terbesar.

Langkah-langkah Proses Penghitungan Association Rule
Proses perhitungan association rule terdiri dari beberapa tahap.

1. Sistem men-scan database untuk mendapatkan kandidat 1-itemset (himpunan item yang terdiri dari 1 item) dan menghitung nilai support nya. Misalnya T

adalah himpunan semua jenis ikan dalam database
Support dari suatu itemset (S) merupakan presentase dari transaksi dalam T yang mengandung S Kemudian nilai support tersebut dibandingkan dengan minimum support yang telah ditentukan, jika nilainya lebih besar atau sama dengan

minimum support maka itemset tersebut termasuk dalam *large itemset*.

2. Itemset yang tidak termasuk dalam *large itemset* tidak diikuti dalam iterasi selanjutnya (diprune).

3. Pada iterasi kedua sistem akan menggunakan hasil *large itemset* pada iterasi pertama (L_1) untuk membentuk kandidat *itemset* kedua (L_2). Pada iterasi

selanjutnya sistem akan menggunakan hasil *large itemset* pada iterasi sebelumnya (L_{k-1}) untuk membentuk kandidat *itemset* berikutnya (L_k). Sistem

akan menggabungkan (join) L_{k-1} dengan L_{k-1} untuk mendapatkan L_k , seperti pada iterasi sebelumnya sistem akan menghapus (prune) kombinasi *itemset* yang tidak termasuk dalam *large itemset*

4. Setelah dilakukan operasi join, maka pasangan *itemset* baru dari hasil proses join tersebut juga dihitung supportnya. e-Indonesia Initiative 2008 (eII2008)

Konferensi dan Temu Nasional Teknologi Informasi dan Komunikasi untuk Indonesia 21-23 Mei 2008, Jakarta 3

5. Proses pembentukan kandidat yang terdiri dari proses join dan prune akan terus dilakukan hingga himpunan kandidat *itemset*nya null, atau sudah tidak ada lagi

kandidat yang akan dibentuk.

6. Setelah itu, dari hasil frequent itemset tersebut dibentuk association rule yang memenuhi nilai support dan confidence yang telah ditentukan.
7. Pada pembentukan association rule, nilai yang sama seperti "kerapu, kakap" = "kakap, kerapu", dianggap sebagai satu nilai.
8. Association rule yang terbentuk harus memenuhi nilai minimum yang telah ditentukan.
9. Untuk setiap large itemset L , kita cari himpunan bagian L yang tidak kosong. Untuk setiap himpunan bagian tersebut, dihasilkan rule dengan bentuk $a \Rightarrow (L - a)$ jika $\text{support}(L)$ dan $\text{support}(a)$ lebih besar dari minimum support.

2.4 UML (*Unified Modelling Language*)

UML (*Unified Modelling Language*) adalah sebuah bahasa yang telah menjadi standar dalam industri visualisasi, merancang dan mendokumentasikan sistem peranti lunak (Yuni Sugiarti, 2013:34⁹). Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti *class*, sistem operasi dan jaringan apapun serta ditulis dalam pemrograman apapun.

⁹ Yuni Sugiarti, (2013). "*Analisis Dan Perancangan UML (Unified Modeling Language) Generated VB. 6.*" Bandung : Graha Ilmu

UML (*Unified Modelling Language*) bisa digunakan untuk :

- a. Menggambarkan batasan sistem dan fungsi-fungsi sistem secara umum.
- b. Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum.
- c. Menggambarkan representasi struktur statik.
- d. Membuat model *behavior* yang menggambarkan kebiasaan atau sifat sebuah sistem
- e. Menyatakan arsitektur implementasi fisik.
- f. Menyampaikan atau memperluas *functionality* dengan *stereotypes*.

UML (*Unified Modelling Language*) merupakan salah satu alat bantu yang handal dalam bidang pengembangan sistem berorientasi objek karena UML menyediakan bahasa pemodelan visual yang memungkinkan pengembang sistem membuat *blue print*. UML berfungsi sebagai jembatan dalam mengkomunikasikan beberapa aspek dalam sistem melalui sejumlah elemen grafis yang bisa dikombinasikan menjadi diagram. UML mempunyai banyak diagram yang dapat mengakomodasi berbagai sudut pandang dari suatu perangkat lunak yang akan dibangun. Diagram-diagram tersebut digunakan untuk:

1. Mengkomunikasi ide.
2. Melahirkan ide-ide baru dan peluang-peluang baru.



3. Menguji ide dan membuat prediksi memahami struktur dan relasi-relasinya.

2.4.1 Usecase Diagram

Dalam membuat sebuah sistem, langkah awal yang perlu dilakukan adalah menentukan kebutuhan. Kebutuhan yang dimaksud adalah kebutuhan fungsional dan kebutuhan nonfungsional. Kebutuhan fungsional adalah kebutuhan pengguna dan stakeholder yang akan dimiliki oleh sistem, dimana kebutuhan ini akan digunakan oleh pengguna dan stakeholder (Yuni Sigiarti, 2013 :41). Kebutuhan nonfungsional adalah kebutuhan yang memperhatikan hal-hal berikut yaitu performansi, kebutuhan dalam menggunakan sistem, kehandalan sistem, keamanan sistem, keuangan, legalitas, dan operasional (Yuni Sigiarti, 2013 :41).

Kebutuhan fungsional akan digambarkan melalui sebuah diagram yang dinamakan diagram usecase. (Yuni Sigiarti, 2013 :41). UML dengan *usecase* berkaitan dengan interaksi antara sistem dan aktor eksternal. (Jakimi dan M. El Koutbi, 2009¹⁰). *Usecase* merupakan pemodelan untuk menggambarkan perilaku (*behavior*) sistem yang akan dibuat.

Diagram *usecase* menggambarkan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Pendekatan *usecase* berbeda dengan metode tradisional.

¹⁰ Jakimi dan M. L Khoutbi, (2009). "An Object-Oriented Approach to UML Scenarios Engineering and Code Generation

Pendekatan tradisional berfokus pada bagaimana memecahkan persoalan besar menjadi persoalan-persoalan yang lebih kecil, sementara pendekatan *usecase* berfokus pada apa yang pemakai harapkan dari sistem. (Sholiq, 2006 : 61¹¹).

Didalam diagram *usecase* terdapat aktor. Langkah-langkah menemukan aktor dalam *usecase* adalah sebagai berikut :

1. Gunakan kata tanya siapa yang akan menggunakan sistem ?
2. Apakah sistem tersebut akan memberikan nilai bagi aktor ?

Jika sudah berhasil menemukan aktor, maka untuk menemukan *usecase* akan lebih mudah dilakukan. Sebuah *usecase* harus mendefinisikan sebuah pekerjaan, dimana pekerjaan tersebut akan memberikan nilai yang bermanfaat bagi aktor.

Untuk menentukan *usecase* mulailah dari sudut pandang aktor, misalnya dengan bertanya :

1. Informasi apa sajakah yang akan didapatkan aktor dari sistem ?
2. Apakah ada kejadian dari sistem yang perlu diberitahukan ke aktor ?

Sedangkan dari sudut pandang sistem, misalnya dengan pertanyaan sebagai berikut :

¹¹ Sholiq, (2006). "Pemodelan Sistem Informasi Berorientasi Objek Dengan UML." Bandung : Graha Ilmu.hal 61

1. Apakah ada informasi yang perlu disimpan atau diambil dari sistem ?
2. Apakah ada informasi yang harus dimasukkan oleh aktor ?

Setiap *usecase* harus dijelaskan alur prosesnya melalui sebuah deskripsi *usecase* atau skenario *usecase*. Deskripsi *usecase* berisi :

1. Nama *usecase* yaitu penamaan *usecase* yang menggunakan kata kerja.
2. Deskripsi yaitu penjelasan mengenai tujuan *usecase* dan nilai yang akan didapatkan oleh aktor.

Tidak semua aktor adalah manusia, bisa saja sistem lain yang berinteraksi dengan sistem yang kita buat. Sistem dibangun untuk menyediakan kebutuhan bagi aktor.

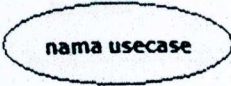

1. Kondisi sebelum (*post-condition*) yaitu kondisi-kondisi yang sudah dipenuhi ketika *usecase* sudah dilaksanakan alur dasar (*basic flow*) yaitu alur yang menceritakan jika semua aksi yang dilakukan adalah benar atau proses yang harusnya terjadi.
2. Alur alternatif (*alternatif flow*) yaitu alur yang menceritakan aksi alternatif, yang berbeda dari alur dasar.

Diagram *usecase* adalah sebuah diagram yang menjelaskan apa yang harus dilakukan oleh sistem pada level konseptual sehingga kita akan memahami apakah keputusan

yang diambil oleh sistem adalah benar atau tidak. (Yuni Sigiarti, 2013 :45¹²).

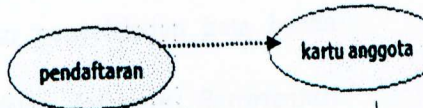
Dari beberapa penjelasan *usecase* dapat ditarik kesimpulan bahwa *usecase* menggambarkan perilaku sistem yang akan dibuat dengan interaksi satu atau lebih aktor dalam sistem yang dibuat. *Usecase* memudahkan dalam hal pemodelan aliran sistem yang akan dibuat.

Tabel 2.2 Simbol Usecase Diagram

Simbol	Deskripsi
 nama usecase	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja.
 nama aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan di buat sendiri. Simbol dari aktor

¹² Yuni Sugiarti, (2013). "Analisis Dan Perancangan UML (Unified Modeling Language) Generated VB. 6." Bandung : Graha Ilmu hal.45

	adalah gambar orang, tetapi aktor bukan berarti harus orang. Biasanya diawali dengan menggunakan kata benda.
<u>Asosiasi/ association</u>	Komunikasi antara aktor dan usecase yang berpartisipasi pada usecase atau usecase memiliki interaksi dengan aktor.
Extend ← «extend»	Relasi usecase tambahan ke sebuah usecase dimana usecase yang ditambahkan dapat berdiri sendiri.
Include «include» →	Relasi usecase tambahan ke sebuah usecase dimana usecase yang ditambahkan memerlukan usecase ini untuk menjalankan fungsinya atau syarat dijalankan usecase ini.



Ada 2 sudut pandang

	mengenai include di usecase, include berarti usecase yang ditambahkan akan selalu dipanggil saat usecase tambahan dijalankan.
--	---

2.4.2 Class Diagram

Class Diagram adalah menunjukkan interaksi antara *class* dalam sistem (Sholiq, 2006:13¹³). *Class Diagram* membantu pengembang mendapatkan struktur sistem sebelum menuliskan kode program, membantu untuk memastikan bahwa sistem adalah rancangan terbaik. (Sholiq, 2006 : 102). *Class* adalah sebuah kategori yang membungkus informasi dan perilaku. Secara tradisional, sistem dibangun dengan ide dasar bahwa akan menyimpan informasi pada sisi basis data dan perilaku pengolahannya pada sisi aplikasi.

Hal terpenting sebelum membuat *Class Diagram* merupakan cara terbaik dalam menemukan *class-class* adalah dimulai dari memperhatikan aliran kejadian (*flow of event*) dari suatu *usecase*. Dengan melakukan penyeleksian kata benda

¹³ Sholiq, (2006). "Pemodelan Sistem Informasi Berorientasi Objek Dengan UML." Bandung : Graha Ilmu, hal 13 & 102

dalam aliran kejadian, dapat ditemukan *class-class* dalam sistem.

Cara kedua adalah dengan menguji objek-objek dalam diagram sekuensial atau diagram kolaborasi. Ada dua cara yang bisa dilakukan berkaitan dengan urutan pendefinisian antar *class-class* dalam diagram *class* dan diagram sekuensial. Kedua cara tersebut adalah

1. Dengan membuat diagram sekuensial lebih dulu, kemudian melanjutkannya dengan membuat diagram *class*.
2. Dengan menemukan *class-class* dan membuat diagram *class* terlebih dahulu, relasinya untuk membuat diagram sekuensial.

Kedua pendekatan di atas, mempunyai keuntungan dan kerugian. Biasanya, kegiatan dan keputusan desain dilakukan dengan membuat diagram sekuensial terlebih dahulu, kemudian dilanjutkan membuat diagram *class*.

Salah satu keuntungan dengan membuat diagram sekuensial terlebih dulu adalah pengembang perangkat lunak lebih berhati-hati dengan menguji langkah demi langkah objek apa saja yang diperlukan untuk memenuhi fungsionalitas yang disajikan dalam aliran kejadian (*flow of events*), dan memastikan bahwa masing-masing *class* digunakan dalam fungsionalitas. Atau dengan kata lain, membuat diagram sekuensial lebih dulu untuk mendapatkan rancangan yang terbaik, dan tidak dibatasi hanya pada *class-class* yang sudah di defenisikan sebelumnya.

Class memiliki tiga area pokok yaitu :

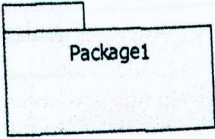
1. Nama
2. Atribut
3. Operasi

Contoh *class* manusia :



1. Atribut : nama, usia, tanggal lahir.
2. Method / Operasi : berjalan, makan, minum.




Atribut adalah karakteristik data yang dimiliki suatu objek dalam *class* (Yuni Sugiarti, 2013:58¹⁴). Operasi adalah fungsi atau transformasi yang dapat diaplikasikan ke suatu objek dalam *class* (Yuni Sugiarti, 2013:58). Misalnya suatu objek dalam *class* manusia mungkin memiliki fungsi-fungsi tersenyum, marah, makan, minum, menerima perlakuan tertentu, dan sebagainya.

Tabel 2.3 Simbol – Simbol Class Diagram

Simbol	Deskripsi
	Package merupakan sebuah bungkusan ari satua atau lebih kelas.

¹⁴ Yuni Sugiarti, (2013). "Analisis Dan Perancangan UML (Unified Modeling Language) Generated VB. 6." Bandung : Graha Ilmu.hal 58

<p>Operasi</p> <table border="1" data-bbox="336 674 497 828"> <tr> <td>nama kelas</td> </tr> <tr> <td>+Attribute1</td> </tr> <tr> <td>+Attribute2</td> </tr> <tr> <td>+Operation1()</td> </tr> </table>	nama kelas	+Attribute1	+Attribute2	+Operation1()	<p>Kelas pada struktur sistem.</p>
nama kelas					
+Attribute1					
+Attribute2					
+Operation1()					
<p>Antarmuka / interface</p> 	<p>Sama dengan konsep interface dalam pemrograman berorientasi objek.</p>				
<p>Asosiasi</p> 	<p>Relasi antar kelas dengan makna umum, asosiasi biasanya disertai dengan multiplicity.</p>				
<p>Asosiasi berarah / directed asosiasi →</p>	<p>Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang</p>				

:	lain, asosiasi biasanya disertai dengan multiplicity.
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus).
Kebergantungan / <i>defedency</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas.
Agregasi 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>).



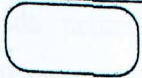

2.4.3 Activity Diagram


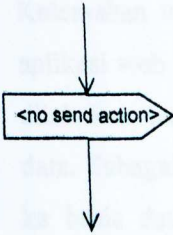
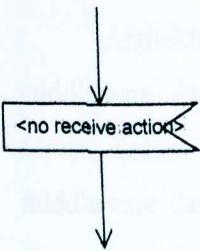

Activity Diagram adalah sebuah cara untuk memodelkan aliran kerja (*workflow*) dari *usecase* dalam bentuk grafik (Sholiq, 2006:42). Diagram ini menunjukkan langkah-langkah di dalam aliran kerja, titik keputusan dalam aliran kerja, siapa yang bertanggung jawab menyelesaikan masing-masing aktivitas, dan objek-objek yang digunakan dalam aliran kerja.

Elemen-elemen yang digunakan dalam *activity diagram* untuk memodelkan aliran kerja melalui *usecase* adalah sebagai berikut :

- a. *Swimlanes*, menunjukkan siapa yang bertanggung jawab melakukan aktivitas dalam suatu diagram.
- b. *Activities*, adalah kegiatan dalam aliran kerja.
- c. *Action*, langka-langkah dalam sebuah aktivitas. Aksi bisa terjadi pada saat memasuki aktivitas, meninggalkan aktivitas, saat di dalam aktivitas atau pada kejadian yang spesifik.
- d. *Object bisnis*, adalah entitas-entitas yang digunakan dalam aliran kerja.
- e. *Transition*, menunjukkan bagaimana aliran kerja itu berjalan dari satu aktivitas ke aktivitas lainnya.
- f. *Decision Point*, menunjukkan di mana sebuah keputusan perlu dibuat dalam aliran kerja.
- g. *Synchronization*, menunjukkan dua atau lebih langkah dalam aliran kerja berjalan secara serentak.
- h. *Start State*, menunjukkan di mana aliran kerja itu dimulai.
- i. *End State*, menunjukkan di mana aliran kerja itu berakhir.

Tabel 2.4 Simbol-Simbol Activity Diagram

Simbol	Keterangan
	Titik Awal
	Titik Akhir
	Aktivitiy
	Pilihan untuk pengambilan keputusan

Simbol	Keterangan
	Fork digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan parallel menjadi satu.
	Tanda pengiriman
	Tanda penerimaan
	Aliran akhir (<i>flow final</i>).

2.5 Aplikasi Web

Aplikasi web dibangun dengan hanya menggunakan bahasa yang disebut *HTML (HyperText Markup Language)*. Pada perkembangan berikutnya, sejumlah skrip dan objek dikembangkan untuk memperluas kemampuan HTML seperti PHP dan ASP pada skrip dan Applet pada objek

(www.wikipedia.com). Aplikasi Web dapat dibagi menjadi dua jenis yaitu aplikasi web statis dan dinamis. Web statis dibentuk dengan menggunakan HTML. Kekurangan aplikasi seperti ini terletak pada keharusan untuk memelihara program secara terus menerus untuk mengikuti setiap perkembangan yang terjadi. Kelemahan ini diatasi oleh model aplikasi web dinamis. Pada aplikasi web dinamis, perubahan informasi dalam halaman web dilakukan tanpa perubahan program tetapi melalui perubahan data. Sebagai implementasi, aplikasi web dapat dikoneksikan ke basis data sehingga perubahan informasi dapat dilakukan oleh operator dan tidak menjadi tanggung jawab dari webmaster.

Arsitektur aplikasi web meliputi klien, web server, middleware dan basis data. Klien berinteraksi dengan web server. Secara internal, web server berkomunikasi dengan middleware dan middleware yang berkomunikasi dengan basis data. Contoh middleware adalah PHP dan ASP. Pada mekanisme aplikasi web dinamis, terjadi tambahan proses yaitu server menerjemahkan kode PHP menjadi kode HTML. Kode PHP yang diterjemahkan oleh mesin PHP yang akan diterima oleh klien.(Abdul Kadir, 2009¹⁵).

2.6 PHP

PHP (*PHP Hypertext Preprocessor*) adalah skrip yang dijalankan di server (Abdul Kadir, 2009:246).

¹⁵ Abdul Kadir, (2009). "*Trik Master PHP dan JQuery*", Andipublisher, Yogyakarta

b. Tipe Data

Tipe data dasar pada php ada tiga macam yaitu :

1. *Integer*
2. *Double*
3. *String*

Integer menyatakan tipe data bilangan bulat dengan jangkauan kira-kira dari -2 milyar hingga +2 milyar. *Double* menyatakan tipe data bilangan real atau tidak mengambang, yaitu bilangan yang mempunyai bagian pecahan. *String* menyatakan tipe data teks (sederetan karakter yang tidak menyatakan bilangan), misalnya nama barang atau nama orang.

c. Operator

Operator adalah simbol yang digunakan dalam program untuk melakukan suatu operasi, misalnya penjumlahan atau perkalian, perbandingan kesamaan dua buah nilai, atau bahkan memberikan nilai ke variabel (Abdul Kadir, 2009:259). Pada php terdapat sejumlah kelompok operator yaitu operator matematika dan logika.

1. Operator Aritmatika

Tabel 2.4 Daftar Operator Aritmatika

Operator	Kegunaan	Prioritas
+	Penjumlahan	Ketiga
-	Pengurangan	Ketiga
*	Perkalian	Kedua

/	Pembagian	Kedua
%	Sisa pembagian	Kedua
++	Penaikkan	Pertama
--	Penurunan	Pertama

2. Operator Penugasan

Operator penugasan pada PHP yang telah diperkenalkan adalah operator `=`. Selain operator ini terdapat sejumlah operator yang tergolong sebagai operator penugasan.

Tabel 2.5 Operator Penugasan

Operator	Kegunaan
<code>+=</code>	Menambahkan variabel di sisi kiri dengan nilai di sisi kanan
<code>-=</code>	Mengurangi isi variabel di sisi kiri dengan nilai di sisi kanan
<code>/=</code>	Membagi variabel di sisi kiri dengan nilai di sisi kanan
<code>%=</code>	Memperoleh sisa pembagian antara variabel di sisi kiri dengan nilai di sisi kanan

3. Operator Pembandingan

Operator pembandingan atau dikenal juga sebagai operator relasional adalah operator yang digunakan untuk melakukan pembandingan dua buah *operand* dan menghasilkan nilai benar

atau salah. Yang termasuk dalam operator pembandingan sebagai berikut :

Tabel 2.6 Operator Pembandingan

Operator	Kegunaan
==	Sama dengan
<	Kurang dari
>	Lebih dari
<=	Kurang dari atau sama dengan
>=	Lebih dari atau sama dengan
!=	Tidak sama dengan
◇	Tidak sama dengan

4. Operator Logika

Operator logika biasa digunakan untuk menggabungkan kondisi berganda dan menghasilkan sebuah ekspresi yang bernilai benar (nilai 1) atau salah (nilai 0). Yang tergolong sebagai operator kelompok ini adalah :

1. And atau &&
2. Or atau ||
3. Xor
4. |

2.7 MySQL Database

Kusrini, M. Kom (2006 :2) basis data adalah kumpulan data yang saling berelasi. Data sendiri merupakan fa

3.1 Kera

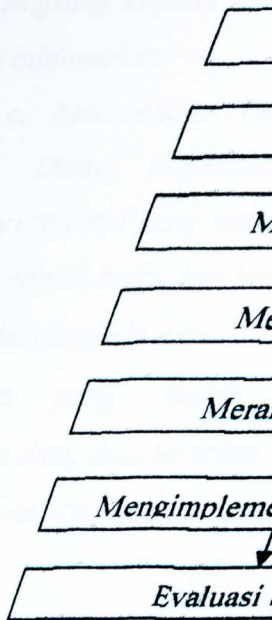
Per

sebagai data

dibeli oleh ko

yang sudah te

penulis dalam n



Gambar 3.1 Kerangka Ke

uraian Kerangka Kerja Penelitian

Berikut ini akan diuraikan dan dijelaskan kerangka kerja penelitian berdasarkan gambar 3.1 :

a. Mengidentifikasi masalah

Penulis mengidentifikasi masalah yang ada dalam penelitian dan membuat rumusan masalah tersebut untuk menemukan solusi terhadap masalah dalam penelitian ini.

b. Mengumpulkan data

Penulis mengumpulkan data dengan cara menanyakan langsung kepada penjaga gudang tentang data barang yang ada di minimarket.

c. Mempelajari literatur

Dalam penelitian yang dilakukan penulis, penulis mencari referensi yang sesuai dengan judul penelitian melalui media internet, buku dan lain-lain.

d. Menganalisis data

Data yang sudah terkumpul melalui tahapan pengumpulan data, data tersebut akan dianalisa agar data tersebut berguna dan teratur sesuai dengan kebutuhan dalam penelitian ini.

e. Merancang sistem

Tahapan selanjutnya, penulis melakukan perancangan terhadap sistem yang diusulkan penulis dalam penelitian ini. Perancangan sistem yang penulis buat terdiri dari