

BAB II

LANDASAN TEORI

II.1. Teori Sistem

Menurut Kusrini (2010:5), Kata Sistem mempunyai beberapa pengertian, tergantung dari sudut mana kata tersebut didefinisikan. Secara garis besar ada dua pendekatan yang dilakukan yaitu :

1. Pendekatan sistem yang lebih menekankan pada elemen-elemen atau kelompoknya, yang didalam hal ini sistem ini didefinisikan sebagai suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu aturan tertentu.
2. Pendekatan sistem sebagai jaringan kerja dari prosedur, yang lebih menekankan urutan operasi didalam sistem. Prosedur didefinisikan sebagai urutan operasi kerja (tuliskan-menulis), yang biasanya melibatkan beberapa orang didalam satu atau lebih departemen yang diterapkan untuk menjamin penanganan yang seragam dari transaksi bisnis yang terjadi.

Pendekatan sistem yang lebih menekankan pada elemen-elemen atau komponennya mendefinisikan sistem sebagai sekumpulan elemen-elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan. Dengan demikian didalam suatu sistem, komponen-komponen ini tidak dapat berdiri sendiri, tetapi sebaliknya, saling berhubungan hingga berbentuk suatu kesatuan hingga tujuan sistem dapat tercapai.

II.1.1. Karakteristik Sistem

Menurut Kusurini (2010:6), Sistem mempunyai beberapa karakteristik atau sifat-sifat tertentu antara lain :

1. Komponen sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerja sama untuk membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian dari sistem. Setiap sistem tidak peduli berapapun kecilnya, selalu mengandung komponen-komponen atau subsistem.

2. Batasan sistem (*Boundary*)

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya.

3. Lingkungan luar sistem (*Environment*)

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut

4. Penghubung sistem (*Interface*)

Merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui perhubungan ini memungkinkan sumber-sumber daya mengalir dari subsistem yang lainnya.

5. Masukan sistem (*Input*)

Merupakan energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*).

6. Keluaran sistem (*Output*)

Keluaran adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

7. Pengolah sistem (*Process*)

Suatu sistem yang dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran.

8. Sasaran sistem (*Objective*)

Suatu sistem pasti mempunyai tujuan atau sasaran. Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.

II.1.2. Klasifikasi Sistem

Menurut Kusrini (2010:7), Sistem dapat diklasifikasikan dari beberapa sudut pandangan, diantaranya sebagai berikut :

1. Sistem abstrak dan sistem fisik.

Sistem abstrak adalah sistem yang berisi gagasan atau konsep. Misalnya, sistem teologi yang berisi gagasan tentang hubungan manusia dan Tuhan. Sistem fisik merupakan sistem yang secara fisik dapat dilihat. Misalnya sistem komputer, sistem sekolah, sistem akuntansi, dan sistem transportasi.

2. Sistem alamiah dan sistem buatan manusia.

Sistem alamiah adalah sistem yang terjadi karena alam (tidak dibuat manusia). Misalnya, sistem tata surya. Sistem buatan manusia adalah sistem yang dibuat oleh manusia. Misalnya, sistem komputer dan sistem mobil.

3. Sistem tertentu dan sistem tak tentu.

Sistem tertentu adalah sistem yang operasinya dapat diprediksi secara tepat. Misalnya, sistem komputer. Sistem tak tentu adalah sistem yang tak dapat diramal dengan pasti karena mengandung unsur probabilitas. Misalnya, sistem arisan dan sistem sediaan.

4. Sistem tertutup dan sistem terbuka

Sistem tertutup adalah sistem yang tidak bertukar materi, informasi, atau energi dengan lingkungan. Misalnya, reaksi kimia dalam tabung terisolasi. Sistem terbuka adalah sistem yang berhubungan dengan lingkungan dan dipengaruhi oleh lingkungan.

II.2. Sistem Pakar

Menurut Kusrini(2008:3), Sistem pakar aplikasi berbasis computer yang dipergunakan untuk menyelesaikan masalah sebagaimana yang dipikirkan oleh pakar. Pakar yang dimaksud disini adalah orang yang mempunyai keahlian khusus yang dapat menyelesaikan masalah yang tidak dapat diselesaikan oleh orang awam. Sebagai contoh dokter adalah seorang pakar yang mampu mendiagnosis penyakit yang diderita oleh pasien serta dapat memberikan penatalaksanaan terhadap penyakit tersebut. Tidak semua orang dapat mengambil keputusan mengenai diagnosis dan penatalaksanaan suatu penyakit. Contoh yang lain montir adalah seorang yang punya keahlian dan pengalaman dalam menyelesaikan kerusakan mesin motor/mobil, psikolog adalah orang yang ahli dalam memahami kepribadian seseorang dan lain-lain.

Sistem pakar, yang mencoba memecahkan masalah yang biasanya yang bias dipecahkan oleh seorang pakar, dipandang berhasil ketika mampu mengambil keputusan seperti yang dilakukan oleh pakar aslinya baik dari sisi pengambilan keputusannya maupun hasil keputusan yang diperoleh.

Sebuah sistem pakar memiliki 2 komponen utama yaitu basis pengetahuan dan mesin inferensi. Basis pengetahuan merupakan tempat penyimpanan pengetahuan dalam memori komputer, dimana pengetahuan ini diambil dari pengetahuan pakar.

Ada banyak cara merepresentasikan pengetahuan diantaranya adalah logika (*logic*), jaringan semantic (*semantic nets*), objek atribut value (OAV), bingkai (*frame*) dan kaidah produksi (*production rule*). Mesin inferensi merupakan otak dari aplikasi sistem pakar, bagian inilah yang menuntun user untuk memasukkan fakta sehingga diperoleh kesimpulan. Apa yang dilakukan oleh mesin inferensi ini didasarkan pada pengetahuan didalam basis pengetahuan.

II.2.1. Representasi Pengetahuan

Menurut Kusri(2008:6), Representasi pengetahuan merupakan metode yang digunakan untuk mengkodekan pengetahuan dalam sebuah sistem pakar. Representasi dimaksudkan untuk menangkap sifat-sifat penting masalah dan membuat informasi itu dapat diakses oleh prosedur pemecahan masalah.

Adapun karakteristik dari metode representasi pengetahuan adalah :

1. Harus bisa deprogram dalam bahasa pemrograman atau dengan *shells* dan hasilnya disimpan dalam memori.
2. Dirancang sedemikian sehingga dapat dijadikan sebagai proses penalaran

3. Model representasi pengetahuan merupakan sebuah struktur data yang dapat dimanipulasi oleh mesin inferensi dan pencarian aktivitas pencocokan pola.

Seperti telah disampaikan, bahwa system pakar ada beberapa metode representasi pengetahuan. Jika pengetahuan berupa pengetahuan yang bersifat deklaratif, maka metode representasi pengetahuan yang cocok adalah jaringan semantik, frame dan logika predikat. Tetapi jika pengetahuannya berupa pengetahuan prosedural yang merepresentasikan aksi dan prosedur, maka metode representasi pengetahuan yang cocok adalah kaidah produksi.

II.2.2. Kaidah Produksi

Menurut Kusrini(2008:6), Pengetahuan dalam kaidah produksi direpresentasikan dalam bentuk :

Jika [Attecedent] Maka [Konsekuen]

Jika [Kondisi] Maka [Aksi]

Jika [Premis] Maka [Konklusi]

Contoh :

Aturan 1

JIKA Terjadi Luka

MAKA diberikan betadine

Aturan 2

JIKA Tidak punya uang cash

MAKA ambil tabungan

Aturan 3

JIKA Bersin-bersin

MAKA Tersebang influenza

Aturan 4

JIKA Bensin Habis

MAKA Motor dapat akan hidup

Aturan terkadang menggunakan operator logika AND atau OR. Misalnya

Aturan 5

JIKA Dana mencukupi

JIKA pengiriman dilakukan kurang dari sebulan

MAKA Beli laser printer

Aturan 6

JIKA Kontraktor tidak bisa menyelesaikan pekerjaan tepat waktu

ATAU biaya melebihi anggaran

MAKA Kontrak batal

Aturan dalam kaidah produksi diklasifikasikan menjadi kaidah derajat pertama dan kaidah meta. Kaidah derajat pertama aturan yang bagian konklusinya tidak menjadi premis bagi kaidah lain. Sebaliknya kaidah meta merupakan kaidah yang konklusinya merupakan premis bagi kaidah yang lain. Kaidah meta merupakan kaidah yang memberi penjelasan bagi kaidah lain. Aturan 8 merupakan contoh kaidah meta dan aturan 9 merupakan contoh kaidah derajat pertama.

Aturan 7

JIKA Kontraktor Pus

DAN Cepat lelah

DAN sering kesemutan

MAKA Anemia

Aturan 8

JIKA Anemia

DAN Batu kronis

MAKA TBC

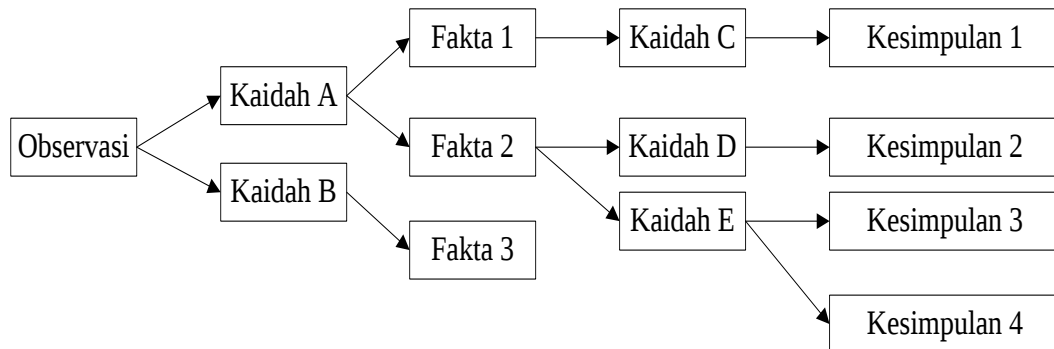
Untuk hasil yang lebih akurat dalam memecahkan masalah dalam suatu domain, biasanya dibutuhkan aturan yang cukup banyak karena masing-masing aturan berisi detail pengetahuan. Jumlah aturan akan menggambarkan kompleksitas sistem pakar.

II.2.3. Metode Inferensi

Menurut Kusrini (2008:8), Metode Inferensi merupakan proses untuk menghasilkan informasi dari fakta yang diketahui atau diasumsikan. Inferensi adalah konklusi logis (*logical conclusion*) atau implikasi berdasarkan informasi yang tersedia dalam hal ini akan digunakan metode inferensi dalam pengambilan kesimpulan. Ada dua metode inferensi yang penting dalam sistem pakar untuk menarik kesimpulan, yaitu :

1. Forward chaining

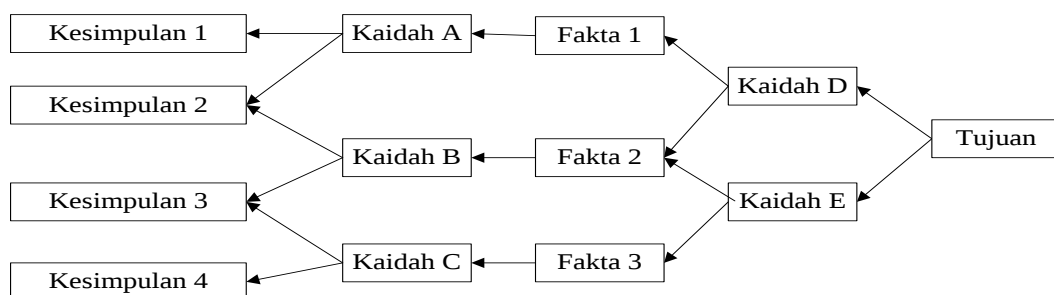
Strategi dari sistem ini adalah dimulai dari inputan beberapa fakta, kemudian menurunkan beberapa fakta dari aturan-aturan yang cocok pada knowledge base dan melanjutkan prosesnya sampai jawaban sesuai. *Forward chaining* dapat dikatakan sebagai penelusuran deduktif.



Gambar II.1. Diagram Pelacakan ke Depan
(Sumber : Kusri, Aplikasi Sistem Pakar; 2008))

2. Backward chaining

Strategi penarikan keputusan yang didasarkan dari hipotesa atau dugaan yang didapat dari informasi yang ada. Ciri dari strategi ini adalah pertanyaan user. Memperoleh fakta biasanya diajukan dalam bentuk “YA” atau “TIDAK”, proses ini berdampak dengan diterima atau tidaknya hipotesis.



Gambar II.2. Diagram Pelacakan ke Belakang
(Sumber : Kusri, Aplikasi Sistem Pakar; 2008),

II.3. Metode Dempster Shafer

Menurut Muhammad Dahria dkk (2013:5), *Dempster-Shafer* adalah suatu teori matematika untuk pembuktian berdasarkan belief functions and plausible reasoning (fungsi kepercayaan dan pemikiran yang masuk akal), yang digunakan untuk mengkombinasikan potongan informasi yang terpisah (bukti) untuk mengkalkulasi kemungkinan dari suatu peristiwa. Teori ini dikembangkan oleh *Arthur P. Dempster dan Glenn Shafer*

Secara umum teori Dempster-Shafer ditulis dalam suatu interval : $[Belief, Plausibility]$

1. *Belief (Bel)* adalah ukuran kekuatan evidence dalam mendukung suatu himpunan proposisi. Jika bernilai 0 maka mengindikasikan bahwa tidak ada evidence, dan jika bernilai 1 menunjukkan adanya kepastian. Dimana nilai bel yaitu (0-0.9).
2. *Plausibility (Pl)* dinotasikan sebagai : $Pl(s) = 1 - Bel(-s)$

Plausibility juga bernilai 0 sampai 1. Jika yakin akan -s, maka dapat dikatakan bahwa $Bel(-s)=1$, dan $Pl(-s)=0$.

Pada teori Dempster-Shafer dikenal adanya frame of discrement yang dinotasikan dengan θ . Frame ini merupakan semesta pembicaraan dari sekumpulan hipotesis. Tujuannya adalah mengaitkan ukuran kepercayaan elemen-elemen θ . Tidak semua evidence secara langsung mendukung tiaptiap elemen. Untuk itu perlu adanya probabilitas fungsi densitas (m). Nilai m tidak hanya mendefinisikan elemen-elemen θ saja, namun juga semua subsetnya. Sehingga jika θ berisi n elemen, maka subset θ adalah 2^n . Jumlah semua m dalam subset θ sama dengan 1.

Apabila tidak ada informasi apapun untuk memilih hipotesis, maka nilai : $m\{\theta\} = 1,0$. Apabila diketahui X adalah subset dari θ , dengan m_1 sebagai fungsi densitasnya, dan Y juga merupakan subset dari θ dengan m_2 sebagai fungsi densitasnya, maka dapat dibentuk fungsi kombinasi m_1 dan m_2 sebagai m_3 , yaitu :

$$m_3(Z) = \frac{\sum_{X \cap Y = Z} m_1(X).m_2(Y)}{1 - \sum_{X \cap Y = \Phi} m_1(X).m_2(Y)} \dots\dots\dots(1)$$

II.4. Mesin Foto Copy

Mesin foto copy adalah peralatan kantor yang membuat salinan ke atas kertas dari dokumen, buku, maupun sumber lain. Mesin foto copy zaman sekarang menggunakan *xerografi*, proses kering yang bekerja dengan bantuan listrik maupun panas. Mesin foto copy lainnya dapat menggunakan tinta.

Sejarah awal penemuan mesin foto copy (*photo copy* dalam bahasa Inggris) diawali oleh penelitian dan percobaan yang sangat panjang. Awalnya penemu sistem *Xerography*, *Chester Carlson*, mengawali pekerjaannya sebagai penyalin dokumen paten di sebuah perusahaan analisis paten, *Carlson* berpikir untuk mempercepat pekerjaannya yaitu dengan membuat sebuah alat yang bisa mencetak dokumen secara berulang-ulang. Ia pun membaca berbagai referensi mengenai mesin cetak. Akhirnya, ia menemukan konsep *elektrofotografi*, yang sekarang kita kenal sebagai mesin foto copy.

Pada 1938, ia membuat eksperimen kecil yang memanfaatkan bubuk jelaga (karbon) dan penyinaran cahaya dan memindahkan suatu tulisan dari sebuah medium ke medium yang lain. Ia juga menggunakan konsep yang disebut *photo-conductivity*, sebuah proses perubahan elektron jika terkena cahaya.

Intinya, dengan proses ini, gambar bisa digandakan dengan proses perubahan elektron tersebut.

Sebagian besar literatur menyebutkan, temuan *Carlson* menciptakan proses mengkopi dengan menggunakan energi elektrostatik, yaitu *xenography*. Nama *xenography* berasal dari bahasa Yunani, *radical xeros* (kering) dan *graphos* (menulis). Karena, dalam prosesnya tidak melibatkan cairan kimia, tak seperti teknologi sebelumnya. Melalui teknik ini, *Chester Carlson* telah menemukan cara yang merombak paradigma penulis ulangan sebuah dokumen, yang nantinya akan menjadi proses yang disebut foto copy. Teknik ini kemudian dipatenkan pada 6 Oktober 1942.

Selama beberapa tahun, ia mencoba menyempurnakan temuannya ini. Meski sangat berguna, mesin elektrofotografi ini tidak diminati banyak orang, karena mesin tersebut dianggap tidak memiliki masa depan yang menjanjikan. *Chester* yang berhasil membuat alat itu harus berjualan konsep bertahun-tahun lamanya agar mesin foto copy itu bisa dijual di pasaran. Berbagai perusahaan besar seperti Kodak yang menjual peralatan dan proses pemotretan, IBM dan *General Electric*, menolak temuan itu.

Setelah hampir putus asa, *Chester* mendapat mitra pertama *Batelle Memorial Institute* yang bersedia memodali dengan dana dan usaha dan kemudian bersama sama berhasil meyakinkan Haloid, sebuah perusahaan menengah *Haloid Corporation, New York* yang menjual kertas foto mau menjadi mitranya untuk mengembangkan temuannya.

II.5. *Unified Modeling Language (UML)*

Adi Nugroho (2010:6), *Unified Modeling Language (UML)* adalah bahasa pemodelan untuk system atau perangkat lunak yang berparadigma “berorientasi objek” . Pemodelan (*Modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami. Dalam hal ini sasaran model sesungguhnya adalah abstraksi segala sesuatu yang ada diplanet bumi menjadi gambaran-gambaran umum yang lebih mudah dipahami dan dipelajari. Adapun tujuan pemodelan (dalam rangka pengembangan system/perangkat lunak aplikasi) sebagai sarana analisis, pemasahaman visualisasi dan komunikasi antar anggota tim pengembang.

II.5.1. **Pengenalan UML**

Menurut Julius Hermawan (2010:7), UML (*Unified Modeling Language*) adalah bahasa standard yang digunakan untuk menjelaskan dan memvisualisasikan artifak dan proses analisis dan desain berorientasi objek. UML Menyediakan standar pada notasi dan diagram yang bias digunakan untuk memodelkan suatu system. UML dikembangkan oleh tiga pendekar “berorientasi objek” yaitu *Gradi Booch*, Jim Rumbaugh dan *Ivar Jacobson*. UML menjadi bahasa yang bias digunakan untuk berkomunikasi dalam prespektif objek antara user dengan developer, antara developer analisis dengan developer desain dan antara developer desain dengan developer pemrograman.

UML memungkinkan developer melakukan pemodelan secara visual, yaitu penekanan pada penggambaran, bukan didominasi oleh narasi. Pemodelan visual

membantu untuk menangkap struktur dan kelakuan dari si objek, mempermudah penggambaran interaksi antara elemen dalam sistem dan mempertahankan konsistensi antara desain dan implementasi dalam bahasa pemrograman.

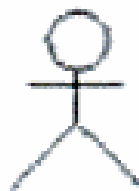
Namun karena UML hanya merupakan bahasa pemodelan maka UML bukanlah rujukan bagaimana melakukan analisis dan desain berorientasi objek. Untuk mengetahui bagaimana melakukan analisis dan desain berorientasi objek secara baik, sudah terdapat beberapa metodologi yang bias diikuti.

II.5.2. Notasi dan Artifak dalam UML

Menurut Julius Hermawan (2010:13), UML menyediakan beberapa notasi dan artifak standard yang bias digunakan sebagai alat komunikasi bagi para proses analisis dan desain. Artifak didalam UML didefenisikan sebagai informasi dalam berbagai bentuk yang digunakan atau dihasilkan dalam proses pengembangan perangkat lunak.

1. Aktor

Aktor adalah segala sesuatu yang berinteraksi dengan system aplikasi computer. Jadi actor ini bisa berupa orang, perangkat keras atau juga objek lain dalam sistem yang sama. Biasanya yang dilakukan oleh Aktor adalah memberikan informasi pada system dan\atau memerintahkan sistem untuk melakukan sesuatu.

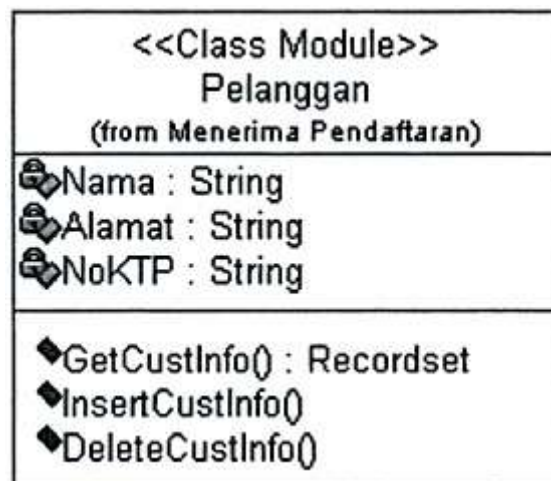


Gambar II.3. Notasi Aktor

2. Class

Class merupakan pembentuk utama dari system berorientasi objek karena *class* menunjukan kumpulan objek yang memiliki atribut dan operasi yang sama.

Class digunakan untuk mengimplementasikan *interface*.



Gambar II.4. Notasi Class

Class digunakan untuk mengabstraksikan elemen-elemen dari system yang dibangun. *Class* bisa untuk direpresentasikan baik perangkat lunak maupun perangkat keras, baik konsep maupun benda nyata. Atribut digunakan untuk menyimpan informasi. Nama atribut menggunakan kata benda yang bisa dengan jelas direpresentasikan informasi yang disimpan didalamnya. Operasi menunjukkan sesuatu yang bisa dilakukan oleh objek, dan menggunakan kata kerja.

3. Interface

Interface merupakan kumpulan informasi tanpa implementasi dari suatu *class*. Implementasi operasi dari suatu *interface* dijabarkan oleh operasi didalam *class*. Oleh karena itu keberadaan *interface* selalu disertai oleh *class* yang

mengimplementasikan operasinya. *Interface* ini merupakan salah satu cara mewujudkan prinsip enkapsulasi dalam objek.



Gambar II.5. Notasi Interface

4. *Use Case*

Use case menjelaskan urutan kegiatan yang dilakukan Aktor dan system untuk mencapai tujuan tertentu. Walaupun menjelaskan kegiatan namun *use case* hanya menjelaskan apa yang dilakukan oleh actor dan system, bukan bagaimana system melakukan kegiatan tersebut.



Gambar II.6. Notasi Use Case

Didalam *use case* terdapat teks untuk menjelaskan urutan kegiatan yang disebut *use case specification*. *use case specification* terdiri dari :

5. Nama *Use Case*

Mencantumkan nama dari use case yang bersangkutan. Sebaiknya diawali dengan kata kerja untuk menunjukkan suatu aktivitas

6. Deskripsi singkat

Menjelaskan secara singkat dalam 1 atau 2 kalimat tentang tujuan dari use case ini.

7. Aliran Normal (*Basic Flow*)

Ini adalah jantung dari *use case*. Menjelaskan interaksi antara actor dan system dalam kondisi normal, yaitu segala sesuatu berjalan dengan lancar tiada halangan atau hambatan dalam mencapai tujuan dalam *use case*.

8. Aliran Alternatif (*Alternative Flow*)

Merupakan pelengkap dari *basic flow* tidak ada yang sempurna dalam setiap kali *use case* berlangsung. Didalam *Alternative Flow* ini dijelaskan dalam apa yang terjadi bila suatu halangan atau hambatan terjadi sewaktu *use case* berlangsung. Ini terutama berhubungan dengan *error* yang mungkin terjadi terutama karena system kekurangan data untuk diolah.

9. *Special Requirement*

Berisi kebutuhan lain yang belum tercakup dalam kebutuhan normal dan alternatif. Biasanya secara tegas dibedakan bahwa *basic flow* dan *alternate flow* menangani kebutuhan fungsional dari *use case* sementara *Special Requirement* yang tidak berhubungan dengan kebutuhan fungsional, misalnya kecepatan transaksi maksimum artinya berapa cepat dan berapa lama, kapasitas akses yaitu jumlah user yang akan mengakses dalam waktu bersamaan.

10. *Pre-Condition*

Menjelaskan persyaratan yang harus dipenuhi sebelum *use case* bisa dimulai.

11. *Post-Condition*

Menjelaskan kondisi yang berubah atau terjadi saat *use case* selesai dieksekusi.

12. *Interaction*

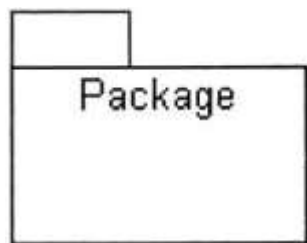
Digunakan untuk menunjukan baik aliran pesan maupun informasi antara objek maupun antara hubungan objek. Biasanya *Interaction* dilengkapi juga dengan teks bernama *operation signature* yang tersusun dari mana operasi, parameter yang dikirim dan type parameter yang dikembalikan.



Gambar II.7. Notasi *Interaction*

13. *Package*

Package adalah kontainer atau wadah konseptual yang digunakan untuk mengelompokkan elemen-elemen dari system yang sedang dibangun, sehingga bisa dibuat model menjadi lebih sederhana. Tujuannya adalah untuk mempermudah pengelihatian dari model yang sedang dibangun.



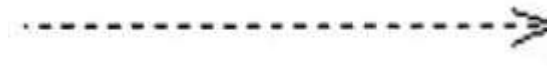
Gambar II.8. Notasi *Package*

14. *Note*

Note digunakan untuk memberikan keterangan dan komentar tambahan dari suatu elemen sehingga bisa langsung terlampir dalam model. *Note* ini bisa ditempelkan ke semua elemen notasi yang lain.

15. *Dependency*

Dependency merupakan relasi yang menunjukkan bahwa perubahan pada salah satu elemen member pengaruh pada elemen lain.



Gambar II.9. Notasi *Dependency*

II.6. Pengertian Basis Data (*Database*)

Basis data merupakan kumpulan dari data-data yang saling terkait dan saling berhubungan satu dengan yang lainnya. Basis data adalah kumpulankumpulan *file* yang saling berkaitan.

Menurut Kusri (2010, p2), pengertian Basis Data adalah kumpulan data yang saling berelasi. Data sendiri merupakan fakta mengenai obyek, orang, dan lain-lain. Data dinyatakan dengan nilai (angka, deretan karakter, atau symbol).

Basis data dapat didefinisikan dalam berbagai susut pandang seperti berikut:

1. Himpunan kelompok data yang saling berhubungan yang diorganisasi sedemikian rupa sehingga kelak dapat dimanfaatkan dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa tanpa pengulangan (*redundancy*) yang tidak perlu, untuk memenuhi kebutuhan.
3. Kumpulan *file*/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpan elektronik.

II.6.2. Tujuan Basis Data

Menurut Kusrini (2010, p2), Basis data bertujuan untuk mengatur data sehingga diperoleh kemudahan, ketepatan dan kecepatan dalam pengambilan kembali. Untuk mencapai tujuan, syarat basisdata yang baik adalah sebagai berikut :

1. Tidak adanya redudansi dan inkonsistensi data

Redudansis terjadi jika suatu informasi disimpan dibeberapa tempat. Misalnya ada data mahasiswa yang memuat nim, nama, alamat dan atribut lainnya, sementara kita punya data lain tentang data KHS mahasiswa yang isinya terdapat NIM, nama, mata kuliah dan nilai. Pada kedua data tersebut kita temukan atribut nama.

2. Kesulitan pengaksesan data

Basis data memiliki fasilitas untuk melakukan pencarian informasi dengan menggunakan query ataupun dari tool yang melibatkan tabelnya. Dengan fasilitas ini, bisa segera langsung melihat data dari software DBMNnya.

3. Multiple user

Basis data memungkinkan penggunaan data secara bersama-sama oleh banyak pengguna pada saat yang bersamaan atau pada saat yang berbeda. Dengan meletakkan basis data pada bagian server yang bisa diakses dari banyak client, sudah menyediakan akses kesemua pengguna dari komputer client ke sumber informasi yaitu basis data.

II.6.2. Manfaat/Kelebihan Basis Data

Menurut Kusrini (2010:5), Banyak manfaat yang diperoleh dengan menggunakan basis data, Manfaat/Kelebihan Basis Data dan kelebihan basis data diantaranya adalah :

1. Kecepatan dan kemudahan

Dengan menggunakan basis data pengambilan informasi dapat dilakukan dengan cepat dan mudah. Basis data memiliki kemampuan dalam mengelompokkan, mengurutkan bahkan perhitungan dengan matematika. Dengan perancangan yang benar maka penyajian informasi dapat dilakukan dengan cepat dan mudah.

2. Kebersamaan pemakai (*sharability*)

Sebuah basis data dapat digunakan oleh banyak user dan banyak aplikasi. Untuk data yang diperlukan oleh banyak bagian/orang, tidak perlu dilakukan pencacatan dimasing-masing bagian/orang, tetapi cukup dengan satu basis data untuk dipakai bersama.

3. Pemusatan kontrol data

Karena cukup satu basis data untuk banyak keperluan, pengontrolan terhadap data juga cukup dilakukan disatu tempat saja.

4. Efisiensi ruang penyimpanan

Dengan pemakaian bersama, tidak perlu menyediakan tempat penyimpanan diberbagai tempat tetapi cukup satu saja, sehingga ini dapat menghemat ruang penyimpanan yang dimiliki oleh sebuah organisasi.

5. Keakuratan (*Accuracy*)

Penerapan secara tepat acuan tipe data, domain data, keunikan data, hubungan antar data, dan lain-lain, dapat menekan ketidakakuratan dalam pemasukan/penyimpanan data.

6. Ketersediaan (*Availability*)

Dengan basis data, semua data dapat dibackup, memilah-milah data mana yang masih diperlukan yang perlu disimpan ke tempat lain. Hal ini mengingat pertumbuhan transaksi sebuah organisasi dari lain waktu ke waktu membutuhkan penyimpanan yang semakin besar.

7. Keamanan (*Security*)

Kebanyakan DBMS dilengkapi dengan fasilitas manajemen pengguna. Pengguna diberi hak akses yang berbeda-beda sesuai dengan kepentingan dan posisinya. Basis data bisa diberikan password untuk membatasi orang yang diaksesnya.

8. Kemudahan dalam pembuatan program aplikasi baru

Penggunaan basis data merupakan bagian dari perkembangan teknologi. Dengan adanya basis data pembuatan aplikasi bisa memanfaatkan kemampuan dari DBMS. Sehingga membuat aplikasi tidak perlu mengurus penyimpanan data, tetapi cukup mengatur interface untuk pengguna.

9. Pemakaian secara langsung

Basis data memiliki fasilitas yang lengkap untuk melihat datanya secara langsung dengan tools yang disediakan oleh DBMS.

10. Kebebasan data

Perubahan dapat dilakukan pada level DBMS tanpa harus membongkar kembali program aplikasinya.

11. User View

Basis data menyediakan pandangan yang berbeda-beda untuk tiap-tiap pengguna.

II.6.3. Operasi Dasar Database

Menurut Kusri (2010: 9), Beberapa operasi dasar basis data yaitu :

1. Pembuatan basis data
2. Penghapusan basis data
3. Pembuatan file/tabel
4. Penghapusan file/tabel
5. Pengubahan tabel
6. Penambahan/pengisian
7. Pengambilan data
8. Penghapusan data
- 9.

II.6.4. Normalisasi

Menurut Samiaji Sarosa (2010:5), Normalisasi adalah teknik yang dirancang untuk merancang tabel basis data relasional untuk meminimalkan duplikasi data dan menghindarkan basis data tersebut anomali. Suatu basis data dikatakan tidak normal jika terjadi 3 (tiga) anomali berikut :

1. *Insertion Anomaly*

Anomali yang terjadi jika ada data yang tidak bisa disisipkan kedalam table.

2. *Update/Modification anomaly*

Anomali yang terjadi jika ada perubahan pada suatu item data maka harus mengubah lebih dari satu baris data.

Langkah-langkah normalisasi sampai pada bentuk 3NF adalah sebagai berikut :

1. *First Normal Form (1NF)*

Untuk menjadi 1NF suatu table harus memenuhi dua syarat. Syarat pertama tidak ada kelompok data atau *field* yang berulang. Syarat kedua harus ada *primary key (PK)* atau kunci unik, atau kunci yang membedakan satu baris dengan baris yang lain dalam satu table. Pada dasarnya sebuah table selamat tidak ada kolom yang sama merupakan bentuk table dengan 1NF.

2. *Second Normal Form (2NF)*

Untuk menjadi 2NF suatu table harus berada dalam kondisi 1NF dan tidak memiliki *partial dependencies*. *Partial dependencies* adalah suatu kondisi jika atribut non kunci (Non PK) tergantung sebagian tetapi bukan seluruhnya pada PK.

3. *Third Normal Form (3NF)*

Untuk menjadi 3NF suatu table harus berada dalam kondisi 2NF dan tidak memiliki *transitive dependencies*. *Transitive dependencies* adalah suatu kondisi dengan adanya ketergantungan fungsional antara 2 atau lebih atribut non kunci (Non PK).

II.7. Java

Java telah hadir dalam dunia pemrograman selama satu decade lebih. Sudah ada beberapa bahasa pemrograman lainnya yang berusaha menyamai bahkan menggantikan kedudukan Java sebagai bahasa pemrograman yang pertama kali memperkenalkan pemrograman lintas platform secara independen dan tidak tergantung dalam sebuah mesin.

II.7.1. Sekilas Sejarah Java

Menurut Wahana Komputer (2010:1), Bahasa pemrograman Java dari sebuah tim pengembangan *software* dari Sun *Microsystem* yang dipimpin oleh *James Gosling* dan *Patrick Naughton*. Pada tahun 1991, Sun *Microsystem* mengembangkan sebuah bahasa pemrograman yang berukuran kecil untuk diimplementasikan pada alat elektronik rumah tangga.

Oleh karena itu keharusan sebuah bahasa pemrograman yang kecil, menghasilkan kode yang kecil pula dan harus platform independen membuat tim pada proyek terinspirasi oleh ide pemrograman yang sama yang telah ditemukan oleh Niklaus Wirth, penemu Pascal. Jadi penemu Bahasa Pascal juga memiliki pemikiran tentang sebuah *software* bahasa pemrograman yang portabel dan tidak tergantung pada sebuah platform atau mesin.

Booming Bahasa Java dimulai pada tahun 1995 ketika Netscape memutuskan untuk menggunakan Java pada web browser-nya, yaitu *Netscape Navigator* pada Januari 1996. Hal ini kemudian diikuti oleh raksasa-raksasa *software* seperti IBM, Symantec, Inprise dan masih banyak lagi termasuk Microsoft dengan *internet explorer*-nya.

II.7.2. Karakteristik Java

Menurut Wahana Komputer (2010:6), Karakteristik Java adalah sebagai berikut :

1. *Java is simple*

Sebenarnya tidak ada satupun bahasa pemrograman yang disebut sederhana, akan tetapi bila dibandingkan dengan pendahulunya seperti Bahasa C++ yang merupakan bahasa pemrograman yang mendominasi dunia pemrograman Java menjadi lebih sederhana.

2. *Java is object oriented*

Pemrograman berorientasi objek adalah pendekatan atau metodologi perancangan system berdasarkan objek. Metode ini menggantikan metode procedural yang telah lama dilakukan. Dalam pemrograman berorientasi objek semua hal dapat dianggap objek.

3. *Java is distributed*

Distributed Computing adalah metode komputerisasi dengan menggunakan dan beberapa komputer yang dihubungkan dengan jaringan untuk mengelola tugas-tugas tertentu. Java telah memiliki kemampuan *Networking* yang bagus, yang menjadikan menulis program *networking* seperti mengirim dan menerima data dari sebuah file.

4. *Java is interpreted*

Karakteristik yang satu ini penting untuk dimengerti oleh pengguna Java yang baru pertama kali mengenal Java. Java adalah bahasa pemrograman yang menggunakan interpreter atau penerjemah supaya dapat menjalankan program.

5. *Java is robust*

Robust artinya dapat diandalkan Java merupakan bahasa pemrograman yang dapat diandalkan.

6. *Java is Secure*

Sebuah bahasa pemrograman internet, Java digunakan pada lingkungan *networking* dan terdistribusi.

7. *Java is Architecture Neutral*

Program yang dihasilkan oleh Java tidak tergantung oleh arsitektur komputer tertentu.

8. *Java is Portable*

Program Java dapat dibawa kemana-mana dan dijalankan dimana-mana.

9. *Java is Performance*

Kinerja dan performa pemrograman Bahasa Java ini sering mendapatkan kritikan atau dianggap lambat beberapa developer.

II.8. MySQL

Menurut Wahana Komputer (2010:5), Mysql adalah program database yang mampu mengirim dan menerima data dengan sangat cepat dan *multi user*. Mysql memiliki dua bentuk lisensi yaitu *free software* dan *shareware*. Mysql sudah cukup lama dikembangkan, beberapa fase penting dalam pengembangan mysql adalah sebagai berikut :

1. Mysql dirilis pertama kali secara internal pada 3 Mei 1995
2. Versi Windows dirilis pada 8 Januari 1998 untuk windows 95 dan windows 1995.
3. Versi 2.32, beta dari Juni 2000, dan dirilis pada Januari 2001.

4. Versi 4.0 : Beta dari bulan Juni 2004, dirilis pada bulan Oktober 2005
5. Versi 5.0 : Beta dari Bulan Maret 2005, dirilis pada Bulan Oktober 2005
6. Sun Microsystem membeli Mysql AB pada tanggal 26 Pebruari 2008.
7. Versi 5.1 : Dirilis 27 November 2008

Mysql database server adalah *RDBMS (Relational Database Management System)* yang dapat menangani data yang bervolume besar. Meskipun begitu, tidak menuntut *resource* yang besar. Mysql adalah database yang paling populer diantara database-database lainnya.

II.8.1. Kelebihan dan Keuntungan Memakai MySQL

Menurut Wahana Komputer (2010:6), Dalam dunia programming ada beberapa database yang sering digunakan antara lain Mysql, Oracle, PostgreSQL, MSql, Microsoft Sql Server dan lain-lain. Ketika dibandingkan antara Mysql dengan system manajemen database yang lain, Mysql memiliki beberapa kelebihan dan keuntungan dibandingkan database lain diantaranya adalah :

1. Banyak ahli berpendapat bawah Mysql merupakan server tercepat
2. Mysql merupakan system manajemen database yang *open source* yaitu *software* ini bersifat *free* atau bebas digunakan oleh perseorangan atau instansi tanpa harus membeli atau membayar kepada pembuatnya.
3. Mysql mempunyai ferporma yang tinggi namun simpel
4. Database Mysql mengerti bahasa SQL (*Structured Query Languange*)
5. Mysql dapat diakses melalaui protocol ODBC (*Open Database Connectivity*) buatan Microsoft. Ini menyebabkan Mysql dapat diakses oleh banyak software.

6. Semua *client* dapat mengakses *Server* dalam satu waktu tanpa harus menunggu yang lain mengakses database.
7. Mysql dapat diakses dari semua tempat di internet dengan hak akses tertentu
8. Mysql merupakan database yang mampu menyimpan data berkapasitas besar, sampai berukuran gigabyte
9. Mysql dapat berjalan diberbagai *operating system* seperti Linux, Windows, Solaris dan lain-lain.

II.8.2. Kemampuan MySQL Server

Abdul Kadir (Yeni Kustiyahningsih, Devie Rosa Anamisa, 2010), Ketika dibandingkan antara *MySQL* dengan *system* database yang lain, maka perlu dipikirkan apa yang paling penting untuk anda. Apakah performa, support, *fitur-fitur SQL*, kondisi keamanan dalam lisensi, atau masalah harga. Dengan pertimbangan tersebut, *MySQL* memiliki banyak keunggulan, antara lain :

1. Kecepatan

Banyak ahli berpendapat *MySQL* merupakan server tercepat.

2. Kemudahan penggunaan

MySQL punya performa tinggi namun merupakan database yang sederhana sehingga mudah disetup dan dikonfigurasi

3. Harga

MySQL gratis untuk penggunaan tertentu.

4. Mendukung query language

MySQL mengerti bahasa *SQL (Structured Query Language)* yang merupakan pilihan system database modern. Anda juga dapat mengakses *MySQL* lewat *protocol ODBC (Open Database Connectivity)* buatan Microsoft.

5. Kapabilitas

Banyak klien dapat mengakses server dalam satu waktu. Mereka dapat menggunakan banyak database secara simultan.

6. Konetifitas dan sekuritas

Database *MySQL* dapat diakses dari semua tempat di Internet dengan hak akses tertentu.

7. Pertabilitas

MySQL dapat berjalan dalam banyak varian *UNIX* dengan baik, sebaik seperti saat berjalan di *system non-UNIX*.

8. Distribusi yang terbuka

MySQL mudah didapatkan dan memiliki *source code* yang boleh disebarluaskan sehingga bisa dikembangkan lebih lanjut.

Bagaimanapun, mungkin yang paling menarik dari semua karakteristik adalah kenyataan bahwa *MySQL* adalah gratis. Hal ini benar karena *T.c.X* menawarkan *MySQL* sebagai produk gratis untuk umum.

Ketersediaan *MySQL 5.6*, *open-source* database paling populer di dunia, Versi terbaru Open Source Database yang paling populer di dunia dan Keunggulan *MySQL* adalah sebagai berikut :

1. Dirancang untuk mendukung generasi web berikutnya, cloud dan aplikasi mobile. Dengan peningkatan kualitas kinerja, skalabilitas, kehandalan dan pengelolaan.
2. Fitur-fitur baru dan perangkat tambahan dalam *MySQL 5.6* tersedia untuk pengguna *MySQL* di Edisi *MySQL Community*. *MySQL 5.6* menyediakan peningkatan skalabilitas linier, membantu pengguna untuk memperluas

kemampuan hardware yang modern. Dengan versi ini, pengguna bisa mendapatkan pengalaman untuk mengembangkan permintaan yang disederhanakan dan eksekusi yang lebih cepat, transaksional yang lebih baik dan ketersediaan aplikasi, fleksibilitas akses NoSQL, replikasi dan instrumentasi yang ditingkatkan.

3. Waktu Eksekusi dan Diagnosa yang Lebih Baik disediakan melalui diagnostik *MySQL Optimizer*.
4. Hasil Kinerja yang Lebih Hebat dan Ketersediaan Aplikasi dengan mesin penyimpanan *Inno DB*.
5. Meningkatkan *scale-Out* dan Ketersediaan Tinggi pada replikasi fitur baru di *MySQL*.
6. Peningkatan kinerja: adanya instrumentasi baru memungkinkan pengguna untuk memantau objek, aplikasi, user dan sumber secara lebih baik.