

BAB II

LANDASAN TEORI

II.1. Teori Sistem

Menurut Kusri (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 (tulis-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 Kusri (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 Kusri (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 Kusri(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 dua 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 Kusrini (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 sistem 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 Kusriani(2008:6), Pengetahuan dalam kaidah produksi direpresentasikan dalam bentuk :

Jika [Antecedent] Maka [Konsekuensi]

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 Terserang 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

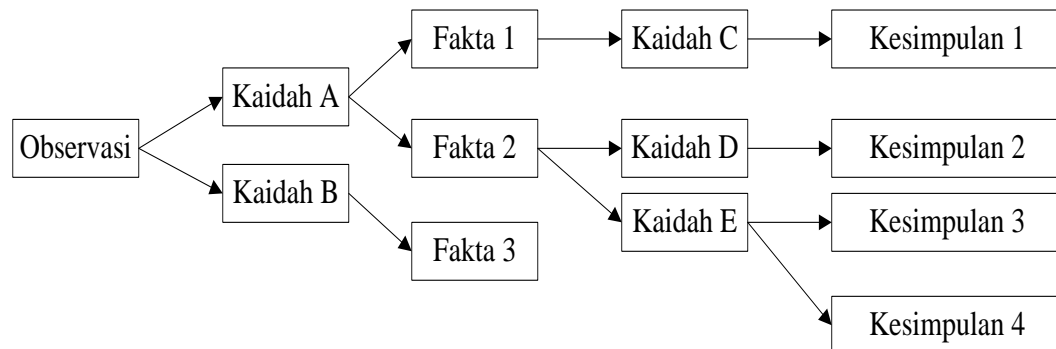
Untuk hasil yang lebih akurat dalam memecahkan masalah dalam suatu domain, biasanya dibutuhkan aturan yang cukup banyak karena masing-masing aturan berisi detail pengetahuan. Jumlah aturan akan menggambarkan kompleksitas sistem pakar.

II.2.3. Metode Inferensi

Menurut Kusri (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

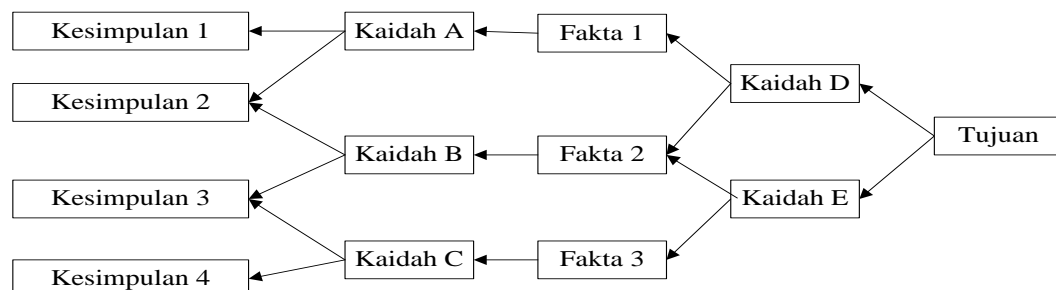
Strategi dari sistem ini adalah dimulai dari inputan beberapa fakta, kemudian menurunkan beberapa fakta dari aturan-aturan yang cocok pada knowledge base dan melanjutkan prosesnya sampai jawaban sesuai. Forward chaining dapat dikatakan sebagai penelusuran deduktif.



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

II.3. Metode *Bayesian Network*

Menurut Sri Rahayu (2013:131), Metode Bayes merupakan metode yang baik didalam mesin pembelajaran berdasarkan data training, dengan menggunakan probabilitas bersyarat sebagai dasarnya. Metode Bayes juga merupakan suatu metode untuk menghasilkan estimasi parameter dengan menggabungkan informasi dari sampel dan informasi lain yang telah tersedia sebelumnya. Keunggulan utama dalam penggunaan Metode Bayes adalah penyederhanaan dari cara klasik yang penuh dengan integral untuk memperoleh model marginal.

Probabilitas *Bayes* merupakan salah satu cara yang baik untuk mengatasi ketidakpastian data dengan menggunakan formula bayes yang dinyatakan dengan rumus :

$$P(H|E) = \frac{P(E \cap H_i)}{P(E)}$$

Keterangan :

$P(H | E)$: probabilitas hipotesis H jika diberikan *evidence* E

$P(E | H)$: probabilitas munculnya *evidence* apapun

$P(E)$: probabilitas *evidence* E

Dalam bidang kedokteran teorema *Bayes* sudah dikenal tapi teorema ini lebih banyak diterapkan dalam logika kedokteran modern (Cutler: 1991).Teorema

ini lebih banyak diterapkan pada hal-hal yang berkenaan dengan probabilitas serta kemungkinan dari penyakit dan gejala-gejala yang berkaitan.

Secara umum teorema *Bayes* dengan E kejadian dan Hipotesis H dapat dituliskan dalam bentuk :

$$\begin{aligned}
 P(H|E) &= \frac{P(E \cap H_i)}{\sum_j P(E \cap H_j)} \\
 &= \frac{P(H|H_i)P(H_i)}{\sum_j P(E|H_i)P(H_j)} \\
 &= \frac{P(H|H_i)P(H_i)}{P(E)}
 \end{aligned}$$

Teorema *Bayes* dapat dikembangkan jika setelah dilakukan pengujian terhadap hipotesis kemudian muncul lebih dari satu *evidence*. Dalam hal ini maka persamaannya akan menjadi:

$$P(H|E, e) = P(H|E) \frac{P(e|E, H)}{P(e|E)}$$

Keterangan :

e : *evidence* lama

E : *evidence* baru

$P(H|E, e)$: probabilitas hipotesis H benar jika muncul *evidence* baru E dari *evidence* baru E dari *evidence* lama

$P(H|E)$: probabilitas hipotesis H benar jika diberikan *evidence* E.

$P(e|E, H)$: kaitan antar e dan E jika hipotesis H benar.

$P(e|E)$: kaitan antara e dan E tanpa memandang hipotesis apapun.

Contoh kasus

Berikut ini adalah contoh penyelesaian *Bayesian Network* jika diketahui 2 gejala sebagai berikut :

Tabel II.1. Contoh Gejala Pilihan Penyakit Sistem Pencernaan Pada Manusia

Kode Gejala	Nama Gejala	Nilai A	Nilai B
GK01	Gejala A	0.7	0.5
GK02	Gejala B	0.6	0.5

Keterangan

Nilai A : tanpa memperhatikan penyakit

Nilai B : Memperhatikan penyakit

$P(\text{Rabun Ayam} \mid \text{Gejala A})$

$$= \frac{(0.5) * (0.7)}{(0.5) + (0.5)}$$

$$= \frac{0.35}{1}$$

$$= 0.35$$

$P(\text{Rabun Ayam} \mid \text{Gejala B})$

$$= \frac{(0.5) * (0.6)}{(0.5) + (0.5)}$$

$$= \frac{0.3}{1}$$

$$= 0.3$$

Perhitungan total bayes

Total Bayes = Bayes1 + Bayes 2

= 0.35 + 0.3

= 0.65

II.4. Sistem Pencernaan Manusia

Menurut Ade Eviyanti,S.Kom (2014:3), Gangguan sistem pencernaan dapat disebabkan oleh pola makan, kebiasaan hidup, infeksi, maupun gangguan alat-alat dalam. Beberapa gangguan yang terjadi dalam pencernaan makanan adalah sebagai berikut :

1. Diare

Penyakit ini diakibatkan oleh infeksi bakteri pada kolon sehingga mengakibatkan gangguan pada penyerapan air, akibatnya feses menjadi encer atau mencret.

2. Gastritis

Gastitris yaitu radang kronis yang terjadi pada lapisan mukosa dinding lambung, penyebabnya karena makanan yang terkena kuman atau kelebihan HCl.

3. Kolitis Ulseratif

Merupakan penyakit usus infia matorik, bersama dengan penyakit cronh, kolitis Ulseratif merupakan penyakit inflamasi kronik pada kolon yang sering kambuh. Penyebab penyakit ini merupakan faktor predisposisi yang berkaitan adalah keturunan, imunologi, infeksi virus atau bakteri (masih spekulatif).

4. Penyakit Crohn

Merupakan salah satu penyakit usus inflamatorik, yang dapat menyerang seluruh bagian saluran gastrointestinal, mulai dari mulut(berupa stomatitis) sampai lesi pada anus. Kebiasaan merokok meningkatkan resiko mendapat penyakit *crohn*.

II.5. Unified Modeling Language (UML)

Adi Nugroho (2010:6), *Unified Modeling Language (UML) adalah bahasa pemodelan untuk sistem 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 sistem/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 sistem. 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.

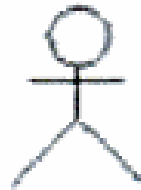
II.5.2. Notasi Dan Artifak Dalam UML

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 sistem aplikasi computer. Jadi actor ini bisa berupa orang, perangkat keras atau juga objek lain

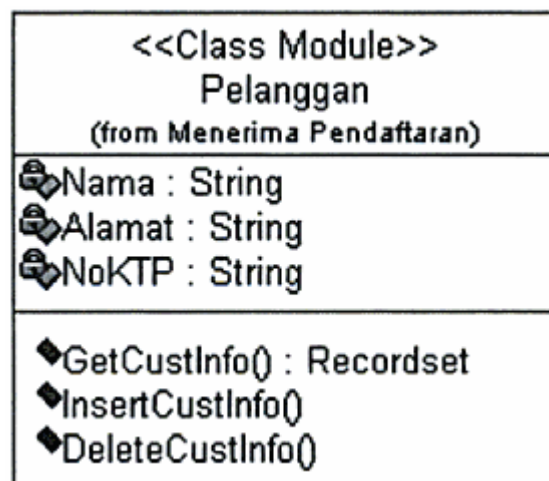
dalam sistem yang sama. Biasanya yang dilakukan oleh Aktor adalah memberikan informasi pada sistem dan\atau memerintahkan sistem untuk melakukan sesuatu.



Gambar II.3. Notasi Aktor

2. *Class*

Class merupakan pembentuk utama dari sistem berorientasi objek karena *class* menunjukkan 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 sistem 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 sistem untuk mencapai tujuan tertentu. Walaupun menjelaskan kegiatan namun *use case* hanya menjelaskan apa yang dilakukan oleh *actor* dan sistem, bukan bagaimana sistem 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 :

1) Nama *Use Case*

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

2) Deskripsi singkat

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

3) Aliran Normal (*Basic Flow*)

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

4) 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 *sistem* kekurangan data untuk diolah.

5) *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.

6) *Pre-Condition*

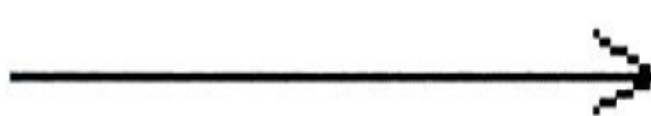
Menjelaskan persyaratan yang harus dipenuhi sebelum *use case* bisa dimulai.

7) *Post-Condition*

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

5. *Interaction*

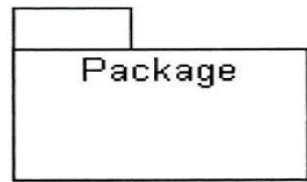
Digunakan untuk menunjukkan baik aliran pesan maupun informasi antara objek maupun antara hubungan objek. Biasanya *Interaction* dilengkapi juga dengan teks bernama *operation signature* yang tersusun dari mana operasi, parameter yang dikirim dan type parameter yang dikembalikan.



Gambar II.7. Notasi *Interaction*

6. *Package*

Package adalah kontainer atau wadah konseptual yang digunakan untuk mengelompokkan elemen-elemen dari *sistem* 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*

7. *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.

8. *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 Kusrini (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.6.1. 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 Kusri (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 (*shareability*)

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, p9). Beberapa operasi dasar basis data yaitu :

1. Pembuatan basis data
2. Penghapusan basis data
3. Pembuatan file/tabel

4. Penghapusan file/tabel
5. Perubahan tabel
6. Penambahan/pengisian
7. Pengambilan data
8. Penghapusan data

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 :

- a. *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.

- b. *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.

c. *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. Visual Basic 2010

Menurut Edi Winarno ST, M.Eng dkk (2010:1), Visual Basic adalah bahasa pemrograman klasik, legendaris yang paling banyak dipakai oleh programmer di dunia. Pemrograman ini dipakai oleh jutaan programmer dan tercatat sebagai program yang paling disukai oleh mayoritas orang.

Visual Studio 2010 pada dasarnya adalah sebuah bahasa pemrograman komputer. Dimana pengertian dari bahasa pemrograman itu adalah perintah-perintah atau instruksi yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu. Visual Studio 2010 selain disebut dengan bahasa pemrograman, juga sering disebut sebagai sarana (Tool) untuk menghasilkan program-program aplikasi berbasis windows.

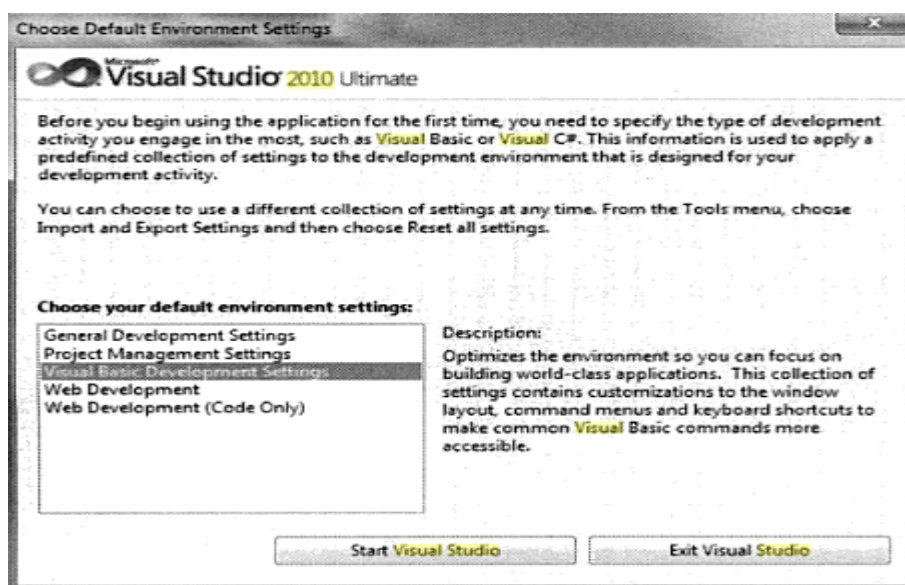
Beberapa kemampuan atau manfaat dari Visual Studio 2010 diantaranya seperti berikut :

1. Untuk membuat program aplikasi berbasis windows.

2. Untuk membuat objek-objek pembantu program seperti, misalnya : kontrol ActiveX, file Help, aplikasi Internet dan sebagainya.
3. Menguji program (debugging) dan menghasilkan program berakhiran EXE yang bersifat executable atau dapat langsung dijalankan.

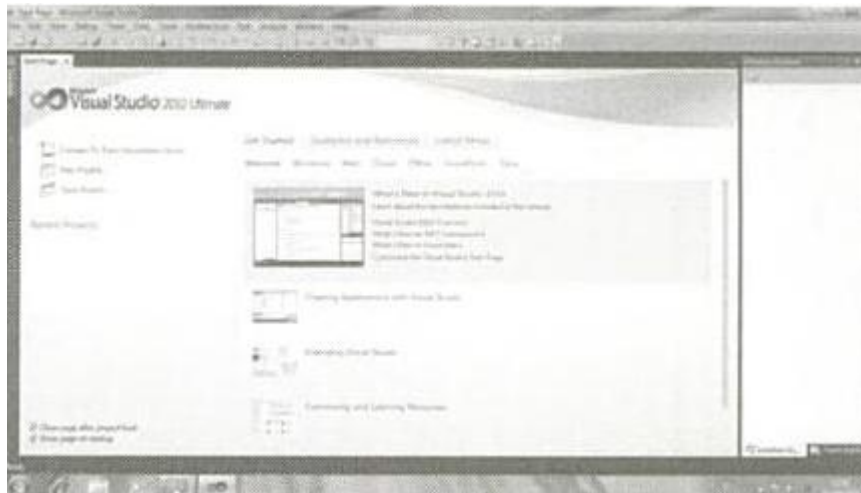
II.7.1. Antar Muka Visual Basic 2010

Saat menjalankan Visual Basic 2010 pertama kali muncul jendela *choose default environment settings*. Disini bisa memilih apakah ingin memilih antar muka di Visual Studio. Untuk *programmer* Visual Basic lebih baik memilih *Visual Basic Development Centre*.



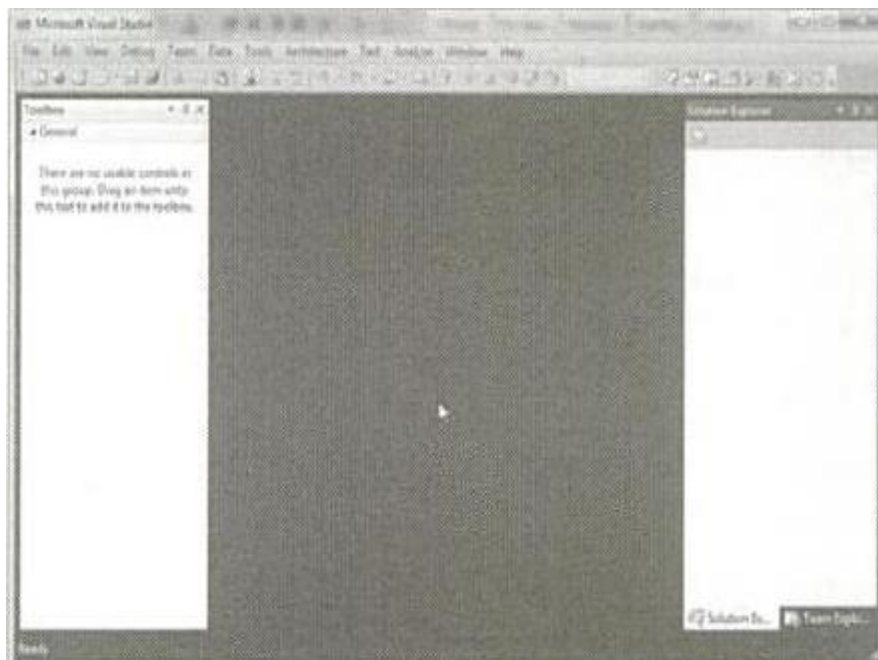
Gambar II.10 Form Chose Default Environment Settings

Dibagian awal visual basic, bisa memilih *Start Page*. *Start Page* adalah halaman yang mencantumkan informasi-informasi seputar program dan juga informasi RSS dari sumber tertentu. Jika tidak ingin menampilkan hal ini hilangkan tanda centang pada *Show Page On Startup*.



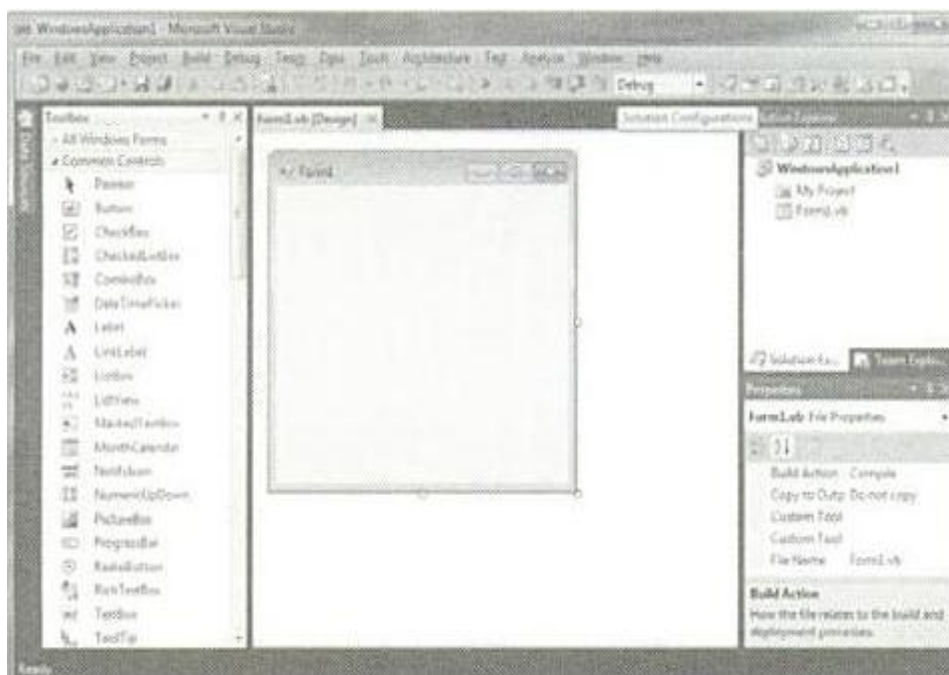
Gambar II.11. Start Page Visual Basic 2010

Jika *start page* ditutup terlihat tampilan sebagai berikut :



Gambar II.12. Tampilan IDE (*Integrated Development Environment*) Setelah *Start Page* Ditutup

Jika ada sebuah form yang terlihat, tampilan lengkap IDE seperti gambar berikut ini.



Gambar II.13. Tampilan Lengkap IDE

Komponen-komponen dari IDE adalah :

1. Dibagian kiri terdapat toolbox yang menampilkan semua objek tool yang bisa dimasukkan kedalam form untuk membuat program.
2. Dibagian tengah terdapat tempat meletakkan form dan kode, baik disaat desain ataupun pada saat program dijalankan.
3. Dibagian kanan terdapat solution explorer yang merupakan explorer untuk melihat file-file disebuah objek.
4. Dikanan bawah terdapat propertis untuk melihat properti dari nilai-nilai pada objek yang dipilih dibagian tengah. (Edi Winarno ST, M.Eng dkk, 2010:1).

II.8. SQL Server 2008 *Express Edition*

Menurut Wahana Komputer (2010:2), SQL Server 2008 *Express Edition* sebuah terobosan baru dalam bidang database, SQL Server adalah sebuah DBMS

(*Database Management System*) yang dibuat oleh Microsoft untuk ikut berkecimpung dalam persaingan dunia pengolahan data menyusul pendahulunya seperti IBM dan Oracle. SQL Server 2008 *Express Edition* dibuat pada saat kemajuan dalam bidang hardware semakin pesat. Oleh karena itu sudah dapat dipastikan bahwa SQL Server 2008 *Express Edition* membawa terobosan dalam bidang pengolahan dan penyimpanan data.

II.8.1 Kebutuhan *Hardware*

Adapun *hardware* yang diperlukan untuk instalasi SQL Server 2008 *Express Edition* minimal adalah sebagai berikut :

1. Prosescor minimal 1 GHz
2. Memori minimal 512 MB
3. Sistem Operasi Windows

Biar dapat diinstal pada system computer dengan memoti 512 MB, tetapi disarankan menggunakan memori 1 GB. Sedangkan untuk jaringannya diperlukan adalah :

1. *Sharer Memory*
2. TCP/IP
3. *Named Pipes*
4. *Virtual Interface Adapter (VIA)*(Wahana Komputer, 2010:2)

II.8.2 Versi SQL Server 2008 *Express Edition*

Microsoft merilis SQL Server 2008 *Express Edition* dalam beberapa versi yang disesuaikan dengan segmen-segmen pasar yang dituju. Versi-versi tersebut adalah sebagai berikut :

1. Menurut cara pemrosesan data pada prosesor maka Microsoft mengelompokkan produk ini berdasarkan dua jenis yaitu :
 - a. Versi 32 Bit (x86), yang biasanya digunakan untuk komputer *single processor* (Pentium 4) atau lebih tepatnya processor 32 bit atau Windows XP.
 - b. Versi 64 Bit (x64), yang biasanya digunakan oleh computer yang lebih dari satu processor (Misalnya *Core 2 duo*) dan system operasi 64 bit, Vista dan Windows 7.
2. Sedangkan secara keseluruhan terdapat versi-versi seperti berikut :
 - a. Versi *Compact* ini adalah versi “tipis” dari semua versi yang ada.
 - b. Versi *Express* ini adalah versi “ringan”.

II.8.3 Instalasi SQL Server 2008 *Express Edition*

Proses instalasi SQL Server 2008 *Express Edition* tidak sama dengan instalasi versi-versi sebelumnya. Proses SQL Server 2008 *Express Edition* agak panjang melalui beberapa tahapan. Tahapan yang dilakukan akan membawa beberapa pilihan yang akan diisi dalam *setting* sebuah *server database*. Berikut ini adalah pilihan-pilihan yang akan dijumpai dalam proses instalasi SQL Server 2008 *Express Edition*.

1. Tempat direktori utama dan penyimpanan file database

Direktori utama adalah adalah direktori dimana semua file program akan ditempatkan dan file-file tersebut tidak akan berubah selama anda menjalankan SQL server. Direktori utama secara standard akan berada dalam direktori “C:\Program Files\Microsoft SQL Server”.

2. Penggunaan *Multiple instance*

Instance adalah sebuah turunan dari server database SQL Server. Karena sebuah tiruan maka sebuah *Instance* memiliki fungsi yang sama dengan database server aslinya. Arti sebenarnya *Instance SQL Server* adalah sebuah server database yang tidak men-*sharing* sistemnya dan database user dengan database server lainnya yang ada dalam komputer yang sama.

3. Jasa *Autentification User* (Menggunakan Windows atau mixed)

Autentification User diperlukan supaya server tidak dapat dipergunakan oleh orang yang tidak bertanggungjawab dan tidak berhak. Dalam SQL server ada dua *Autentification User* yang dapat digunakan yaitu :

1. Mode Windows, Pada mode ini SQL Server akan melakukan autentifikasi dengan menggunakan level login pada system operasi.
2. Mode Mixel atau campuran, mode ini menginjinkan *user* untuk masuk kedalam system SQL server dengan menggunakan *Account* yang dibuat di system operasi windows atau juga menggunakan *account* yang di *set up* ada SQL Server (Wahana Komputer, 2010:2).