

BAB II

LANDASAN TEORI

II.1. Teori Sistem

Menurut Kusriani (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 system yang lebih menekankan pada elemen-elemen atau kelompoknya, yang didalam hal ini system 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 system sebagai jaringan kerja dari prosedur, yang lebih menekankan urutan operasi didalam system. 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 system yang lebih menekankan pada elemen-elemen atau komponennya mendefinisikan system sebagai sekumpulan elemen-elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan. Dengan demikian didalam suatu system, komponen-komponen ini tidak dapat

berdiri sendiri, tetapi sebaliknya, saling berhubungan hingga berbentuk suatu kesatuan hingga tujuan system 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 Pendukung Keputusan

Sistem pendukung keputusan (Inggris: *decision support systems* disingkat DSS) adalah bagian dari sistem informasi berbasis komputer (termasuk sistem berbasis pengetahuan (manajemen pengetahuan) yang dipakai untuk mendukung pengambilan keputusan dalam suatu organisasi atau perusahaan.

II.2.1. Pengertian Sistem Pendukung Keputusan

Menurut Dicky Nofriansyah (2014:1), Sistem pendukung keputusan (SPK) dibangun untuk mendukung solusi atas suatu masalah atau untuk suatu peluang. Aplikasi Sistem pendukung keputusan (SPK) digunakan untuk pengambilan keputusan. Aplikasi Sistem pendukung keputusan (SPK) menggunakan *CBIS* (*Computer Based Information System*) yang fleksibel, interaktif dan dapat diadaptasi yang dikembangkan untuk mendukung solusi masalah manajemen spesifik yang tidak terstruktur.

Menurut Bonczek dkk, (1980) dalam buku "*Decision Support System and intelligent system* (Turban 2005:137) mendefinisikan Sistem pendukung keputusan (SPK) sebagai system berbasis computer yang terdiri dari tiga komponen yang saling berinteraksi, system bahasa (mekanisme untuk memberikan komunikasi antara pengguna dan komponen system pendukung keputusan lain), system pengetahuan (repository pengetahuan domain masalah yang ada pada Sistem pendukung keputusan (SPK) atau sebagai data atau sebagai prosedur), dan system pemrosesan masalah (hubungan antara dua komponen lainnya, terdiri dari satu atau lebih kapabilitas manipulasi masalah umum yang diperlukan untuk pengambilan keputusan).

II.2.2. Karakteristik Sistem Pendukung Keputusan

Menurut Dicky Nofriansyah (2014:2), Karakteristik Sistem pendukung keputusan (SPK) yaitu :

1. Mendukung proses pengambilan keputusan suatu organisasi atau perusahaan.
2. Adanya interface manusia/mesin dimana manusia (*user*) tetap memegang kontrol proses pengambilan keputusan.
3. Mendukung pengambilan keputusan untuk membahas masalah terstruktur, semi terstruktur serta mendukung beberapa keputusan yang saling berinteraksi.
4. Memiliki kapasitas dialog untuk memperoleh informasi sesuai dengan kebutuhan.
5. Memiliki sub sistem yang terintegrasi sedemikian rupa sehingga dapat berfungsi sebagai kesatuan sistem.

II.2.3. Ciri-Ciri Sistem Pendukung Keputusan

Menurut Dicky Nofriansyah (2014:2), Kriteria atau ciri-ciri sistem pendukung keputusan adalah sebagai berikut :

1. Banyak pilihan/alternatif
2. Ada kendala atau surat
3. Mengikuti suatu pola atau model tingkah laku, baik yang terstruktur maupun tidak terstruktur.
4. Banyak input/Variabel
5. Ada faktor resiko, dibutuhkan kecepatan, ketepatan dan keakuratan .

II.2.4. Fase Dalam Sistem Pendukung Keputusan

Menurut Dicky Nofriansyah (2014:2), Tiga fase dalam pengambilan keputusan yaitu :

a. *Intelligence*

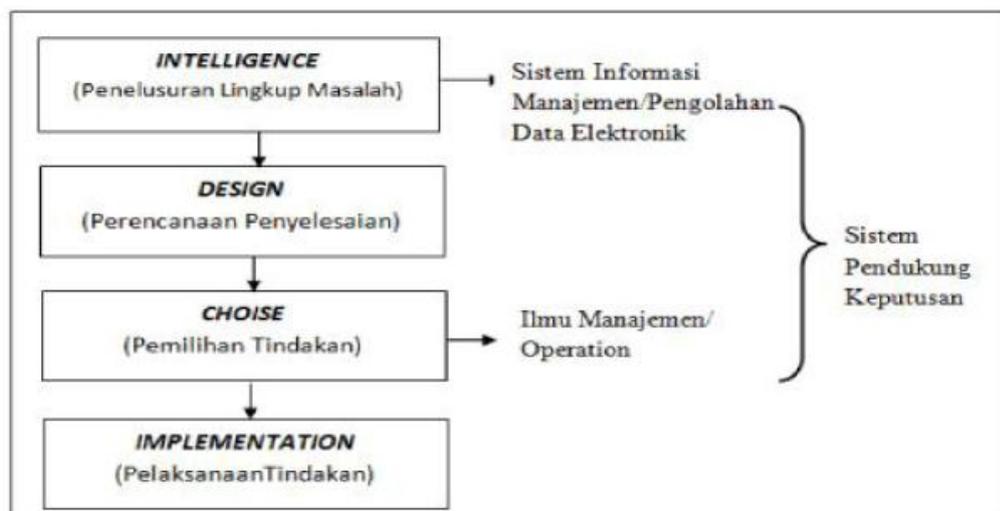
Tahap ini merupakan proses penelusuran dan pendeteksian dari ruang lingkup problematika secara proses pengenalan masalah. Data masukan diperoleh dan diuji dalam rangka mengidentifikasi masalah.

b. *Design*

Tahap ini merupakan proses menemukan, mengembangkan dan menganalisis alternatif tindakan yang bisa dilakukan. Tahap melakukan pengujian kelayakan solusi.

c. *Choice*

Pada tahap ini dilakukan proses pemilihan diantara berbagai alternatif tindakan yang mungkin dijalankan. Hasil pemilihan tersebut kemudian diimplementasikan kedalam proses pengambilan keputusan.



Gambar II.1. Fase Proses Pengambilan Keputusan

II.2.5 Komponen Sistem Pendukung Keputusan

Menurut Dicky Nofriansyah (2014:3), Secara garis besar sistem pendukung keputusan dibangun oleh tiga komponen utama yaitu :

1. Sub Sistem Data (*Database*)

Sub sistem data merupakan komponen sistem pendukung keputusan yang berguna sebagai penyedia data bagi sistem. Data tersebut disimpan untuk diorganisasikan dalam sebuah basis data yang diorganisasikan oleh suatu sistem yang disebut dengan Sistem Manajemen Sistem Basis Data (*Database Management System*)

2. Sub sistem Model

Model adalah suatu tiruan dari alam nyata. Kendala yang sering dihadapi dalam merancang model adalah bahwa model yang dirancang tidak mampu mencerminkan seluruh variabel alam nyata, sehingga keputusan yang diambil tidak sesuai dengan kebutuhan. Oleh karena itu, dalam menyimpan berbagai model harus diperhatikan dan harus diajaga fleksibilitasnya. Hal lain yang harus diperhatikan adalah pada setiap model yang disimpan hendaknya ditambahkan rincian, keterangan dan penjelasan yang komprehensif mengenai model yang dibuat.

3. Sub sistem dialog (*User System Interface*)

Sub sistem dialog adalah fasilitas yang mampu mengintegrasikan sistem yang terpasang dengan pengguna secara interaktif, yang dikenal dengan sub sistem dialog. Melalui sub sistem dialog sistem

diimplementasikan sehingga pengguna dapat berkomunikasi dengan sistem yang dibuat.

II.2.6 Tujuan Sistem Pendukung Keputusan

Menurut Dicky Nofriansyah (2014:4), Tujuan dari system pendukung keputusan adalah sebaagi berikut :

1. Membantu dalam pengambilan keputusan atas masalah yang terstruktur
2. Memberikan dukungan atas pertimbangan manager dan bukannya dimaksudkan untuk menggantikan fungsi manager.
3. Meningkatkan efektifitas keputusan yang diambil lebih dari perbaikan efesiensinya.
4. Kecepatan komputasi komputer memungkinkan para pengambil keputusan untuk banyak melakukan komputasi secara cepat dengan biaya yang sangat rendah.
5. Peningkatan produktifitas membangun suatu kelompok pengambil keputusan, terutama para pakar bisa sangat mahal. Sistem pendukung keputusan komputerisasi mengurangi ukuran kelompok dan memungkinkan para anggotanya untuk berada diberbagai lokasi yang berbeda-beda (menghemat biaya perjalanan). Selain itu produktifitas staf pendukung (misalnya analisis keuangan dan hokum) bisa ditingkatkan. Produktivitas juga ditingkatkan menggunakan peralatan optimalisasi yang menjalankan sebuah bisnis.

II.3 Reward

Menurut Rumiris Siahaan (2013:20), *Reward* merupakan sebagai bentuk apresiasi usaha untuk mendapatkan tenaga kerja yang profesional sesuai dengan tuntutan jabatan diperlukan suatu pembinaan yang berkeimbangan, yaitu suatu usaha kegiatan perencanaan, pengorganisasian, penggunaan, dan pemeliharaan tenaga kerja agar mampu melaksanakan tugas dengan efektif dan efisien. Sebagai langkah nyata dalam hasil peembinaan maka diadakan pemberian *reward* pegawai yang telah menunjukkan prestasi kerja yang baik.

Reward dalam perusahaan kerap dalam bentuk pemberian berupa piagam dan sejumlah uang dari perusahaan untuk pegawai yang mempunyai prestasi. Ada juga perusahaan yang memberikan *reward* kepada pegawai karena masa kerja dan pengabdianya dapat dijadikan teladan bagi pegawai lainnya. Pemberian *reward* karena masa kerja pegawai bertujuan untuk memotivasi gairah dan loyalitas perusahaan. Pemberian *reward* tersebut merupakan upaya perusahaan dalam memberikan balas jasa atas hasil kerja pegawai, sehingga dapat mendorong pegawai bekerja lebih giat dan berpotensi. Pegawai memerlukan suatu *reward* pada saat hasil kerjanya telah memenuhi atau bahkan melebihi standar yang telah ditentukan oleh perusahaan. *Reward* ini dapat berupa pujian. Tidak hanya pegawai yang melakukan kesalahan memperoleh makian dari pimpinan. Pegawai bekerja mempunyai tujuan, antara lain untuk memperoleh penghasilan agar kebutuhan dan keinginannya dapat direalisasikan.

Reward ialah sesuatu yang diberikan kepada perorangan atau kelompok jika mereka melakukan suatu keunggulan di bidang tertentu. *Reward* biasanya

diberikan dalam bentuk medali, piala, gelar, sertifikat, plakat atau pita. Suatu *reward* kadang – kadang disertai dengan pemberian hadiah berupa uang seperti hadiah nobel untuk kontribusi terhadap masyarakat, dan hadiah pulitzer untuk *reward* disbanding literatur. *Reward* bisa juga diberikan oleh masyarakat karena pencapaian seseorang tanpa hadiah apa – apa.

II.3.1 Fungsi dan Tujuan Reward

Menurut Rumiris Siahaan (2013:21), mengemukakan beberapa fungsi *reward* sebagai berikut :

1. Memperkuat motivasi untuk memacu diri agar mencapai prestasi.
2. Memberikan tanda bagi seseorang yang memiliki kemampuan lebih.
3. Bersifat Universal.

Adapun tujuan *reward* seperti yang dikemukakan oleh Taylor (dalam Manullang, 1994) menyatakan tujuan *reward* adalah sebagai berikut :

1. Menarik (merangsang) seseorang agar mau bergabung dengan perusahaan.
2. Mempertahankan karyawan yang ada agar tetap mau bekerja di perusahaan.
3. Memberi lebih banyak dorongan agar para karyawan tetap berprestasi.

Pencapaian tujuan perusahaan agar sesuai dengan yang diharapkan maka fungsi *reward* harus dilakukan sebelum terjadinya penyimpangan-penyimpangan sehingga lebih bersifat mencegah dibandingkan dengan tindakan – tindakan *reward* yang sudah terjadinya penyimpangan. Oleh karena itu, tujuan *reward* adalah

menjaga hasil pelaksanaan kegiatan sesuai dengan rencana, ketentuan-ketentuan dan instruksi yang telah ditetapkan benar-benar diimplementasikan, sebab *reward* yang baik akan tercipta tujuan *reward*.

II.3.2 Bentuk – bentuk Reward

Menurut Rumiris Siahaan (2013:21), *Reward* bermacam-macam, ada dalam bentuk bonus, promosi, penambahan tanggung jawab yang bagi beberapa pegawai bisa menjadi beban namun bagi beberapa pegawai lainnya dapat menjadi poin *reward* bagi dirinya. Tapi yang pasti perusahaan harus memberikan nilai lebih. Kadang kala tidak seluruhnya mengenai uang.

Bentuk – bentuk *reward* atau insentif dapat diklasifikasikan sebagai berikut :

1. Material berupa gaji/upah.

Kenaikan gaji/upah, rencana – rencana bonus, rencana – rencana perangsang.

2. Imbalan diluar gaji

Berupa istirahat kerja, dan bonus

3. Penghargaan sosial

Berupa *reward* informal, pujian, senyum, umpan balik evaluatif, isyarat – isyarat nonverbal, tepukan dibahu, meminta saran, undangan minum kopi bersama atau makan bersama, penghargaan formal, dan plakat dinding.

4. Tugas itu sendiri

Seperti perasaan berprestasi, pekerjaan dengan tanggung jawab lebih besar rotasi kerja, dan sebagainya.

5. Diterapkan sendiri

Berupa *reward* terhadap diri sendiri, pujian untuk diri sendiri, ucapan selamat untuk diri sendiri. Bentuk *reward* yang paling baik adalah membuat pegawai mengetahui kalau dirinya dihargai oleh perusahaan, bukan hanya oleh sekelompok kecil. Beberapa perusahaan kadang kala menempatkan foto pegawai yang paling berprestasi di area pabrik untuk menegaskan bahwa pegawai tersebut dapat menjadi panutan bagi yang lain.

II.4 Metode Perbandingan Eksponensial

Eriyatno (2007) “Analisis Metode Perbandingan” : Metode perbandingan eksponensial (MPE) merupakan salah satu metode untuk menentukan urutan prioritas alternative keputusan dengan kriteria jamak. Teknik ini digunakan sebagai pmbantu individu pengambilan keputusan untuk menggunakan rancang bangun yang telah terdefenisi dengan baik pada tahap proses.

Dalam menggunakan Metode Perbandingan Eksponensial ada beberapa tahap yang harus dilakukan, yaitu:

1. Menyusun alternatif-alternatif keputusan yang akan dipilih.
2. Menentukan kriteria atau perbandingan keputusan yang penting untuk dievaluasi.
3. Menentukan tingkat kepentingan dari setiap kriteria keputusan.

4. Melakukan penilaian terhadap semua alternative pada setiap kriteria.
5. Menghitung skor atau nilai total setiap alternative.
6. Menentukan urutan prioritas keputusan didasarkan pada skor atau nilai total masing-masing alternative.

Formulasi perhitungan skor untuk setiap alternative dalam Metode Perbandingan Eksponensial adalah sebagai berikut:

$$\text{Total Nilai (TN}_i) = \sum_{j=1}^m (RK_{ij})^{TKK_j}$$

TN_i = Total nilai alternatif ke-i

RK_{ij} = Derajat kepentingan relatif criteria ke-j pada pilihan keputusan i

TKK_j = Derajat kepentingan criteria keputusan ke-j; TKK_j > 0; bulat

m = Jumlah kriteria keputusan

n = Jumlah pilihan keputusan

j = 1,2,3,...; m = Jumlah kriteria

i = 1,2,3,...,n ; n = Jumlah pilihan alternatif

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.

Menurut Haviluddin (2011;3), Pada UML versi 2 terdiri atas tiga kategori dan memiliki 13 jenis diagram, dan diantaranya adalah sebagai berikut :

1. Struktur Diagram

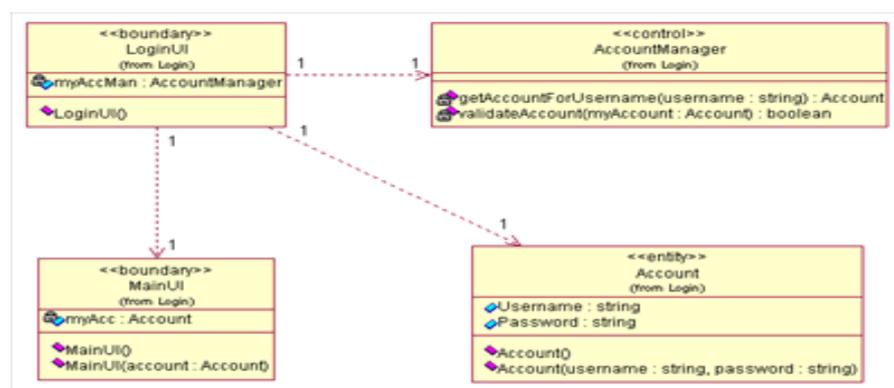
Menggambarkan elemen dari spesifikasi dimulai dengan kelas, obyek, dan hubungan mereka, dan beralih ke dokumen arsitektur logis dari suatu sistem. Struktur diagram dalam UML terdiri atas :

a. *Class* diagram

Class diagram menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan

hubungan antara kelas. *Class diagram* membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Selama tahap desain, *class diagram* berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat. *Class* memiliki tiga area pokok :

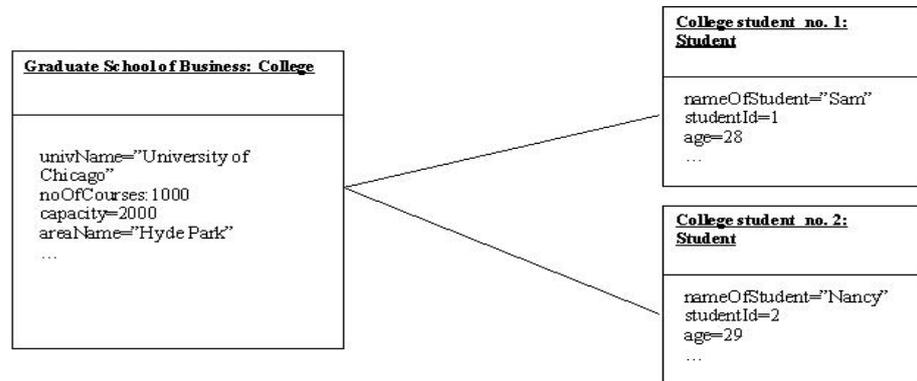
- 1). Nama (dan *stereotype*)
- 2). Atribut
- 3). Metoda



Gambar II.2. Notasi Class Diagram
(Sumber : Havaluddin ; 2011)

b. *Object Diagram*

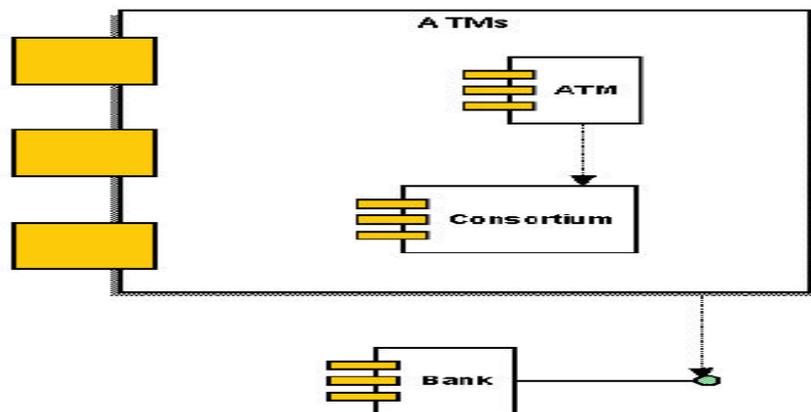
Object diagram menggambarkan kejelasan kelas dan warisan dan kadangkadang diambil ketika merencanakan kelas, atau untuk membantu pemangku kepentingan non-program yang mungkin menemukan diagram kelas terlalu abstrak. Berikut notasi *object diagram* dilihat pada gambar II.3.



Gambar II.3. Notasi Object Diagram
(Sumber : Haviluddin ; 2011)

c. *Component diagram*

Component diagram menggambarkan struktur fisik dari kode, pemetaan pandangan logis dari kelas proyek untuk kode aktual di mana logika ini dilaksanakan dan gambaran bentuk *Component Diagram* dilihat pada gambar II.4.

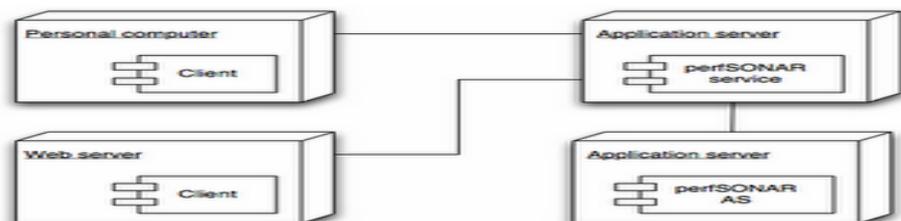


Gambar II.4. Notasi Component Diagram
(Sumber : Haviluddin ; 2011)

d. *Deployment diagram (Collaboration diagram in version 1.x)*

Deployment diagram memberikan gambaran dari arsitektur fisik perangkat lunak, perangkat keras, dan artefak dari sistem.

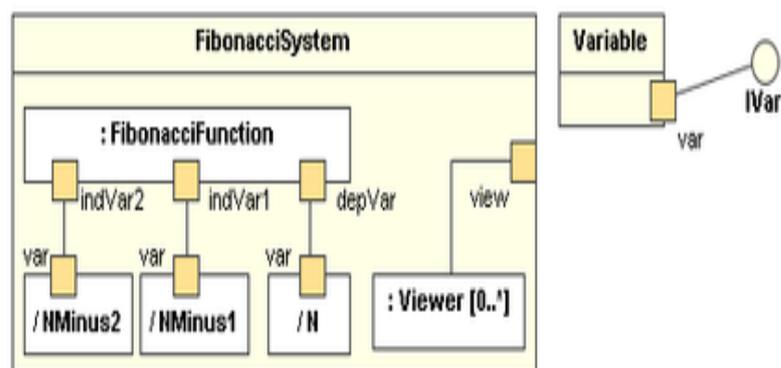
Deployment diagram dapat dianggap sebagai ujung spektrum dari kasus penggunaan, menggambarkan bentuk fisik dari sistem yang bertentangan dengan gambar konseptual dari pengguna dan perangkat berinteraksi dengan sistem dan diagram ini dapat dilihat pada Gambar II.5.



Gambar II.5. Notasi *Deployment Diagram*
(Sumber : (Sumber : Haviluddin ; 2011)

e. *Composite structure diagram*

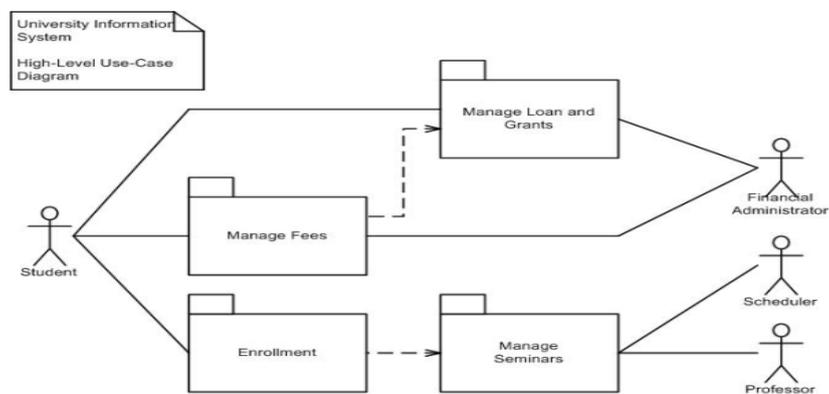
Sebuah diagram struktur komposit mirip dengan diagram kelas, tetapi menggambarkan bagian individu, bukan seluruh kelas. Kita dapat menambahkan konektor untuk menghubungkan dua atau lebih bagian dalam atau ketergantungan hubungan asosiasi dan bentuk dari diagram ini dapat dilihat pada Gambar II.6.



Gambar II.6. Notasi *Composite structure Diagram*
(Sumber : (Sumber : Haviluddin ; 2011)

f. *Package diagram*

Paket diagram biasanya digunakan untuk menggambarkan tingkat organisasi yang tinggi dari suatu proyek *software*. Atau dengan kata lain untuk menghasilkan diagram ketergantungan paket untuk setiap paket dalam Pohon Model dan bentuk dari diagram ini dapat dilihat pada Gambar II.7.



Gambar II.7. Notasi *Package Diagram*
(Sumber : (Sumber : Havaluddin ; 2011))

2. *Behavior Diagram*

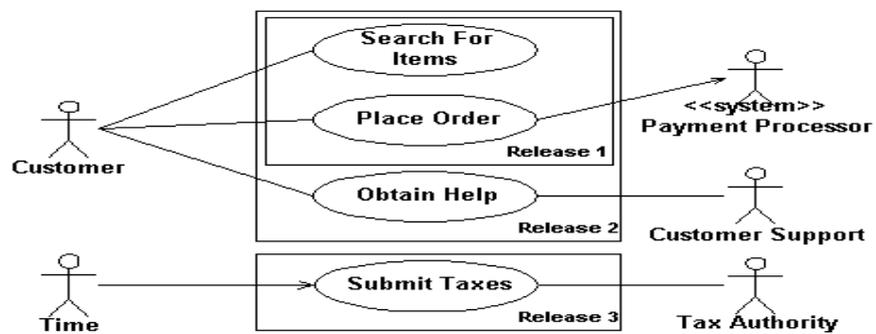
Menggambarkan ciri-ciri *behavior/metode/ fungsi* dari sebuah sistem atau *business process*. *Behavior diagram* dalam UML terdiri atas :

g. *Use case diagram*

Diagram yang menggambarkan *actor, use case* dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur

untuk aktor. Sebuah *use case* digambarkan sebagai elips horizontal dalam suatu diagram UML *use case*. *Use Case* memiliki dua istilah yaitu :

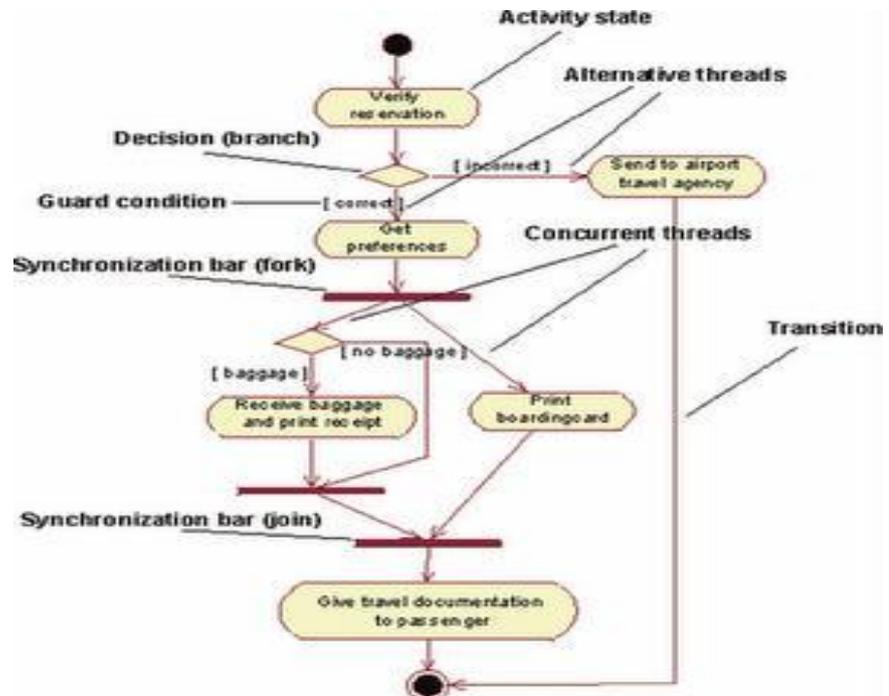
- 1). *System use case*; interaksi dengan sistem.
- 2). *Business use case*; interaksi bisnis dengan konsumen atau kejadian nyata.



**Gambar II.8. Notasi Use Case Diagram
(Sumber : (Sumber : Havaluddin ; 2011))**

h. Activity Diagram

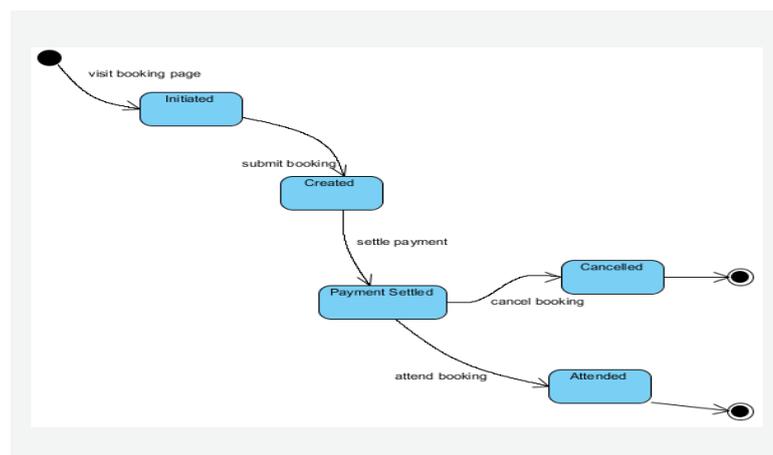
Menggambarkan aktifitas-aktifitas, objek, *state*, transisi *state* dan *event*. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk segala aktivitasnya, dan gambarannya dapat dilihat pada Gambar II.9.



Gambar II.9. Notasi Activity Diagram
(Sumber : (Sumber : Haviluddin ; 2011)

i. *State Machine* diagram (*State chart* diagram in version 1.x)

Menggambaran *state*, transisi *state* dan *event* dan untuk gambaran dari diagram ini dapat dilihat pada Gambar II.10.



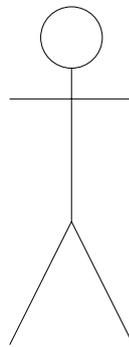
Gambar II.10. Notasi State Machine Diagram
(Sumber : (Sumber : Haviluddin ; 2011)

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



Gambar II.11. Notasi Aktor

2. Class

Class merupakan pembentuk utama dari system berorientasi objek karena *class* menunjukkan kumpulan objek yang memiliki atribut dan operasi yang sama. *Class* digunakan untuk mengimplementasikan *interface*

Tabel II.1. Notasi Class

<<Class Module>> Pelanggan (Form Menerima Pendaftaran)
Nama : String Alamat : String NoKTP : String
➤ GetCustInfo():Recordset ➤ InsertCustInfo ➤ DeleteCustInfo()

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.

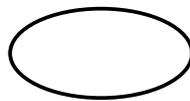
3. *Interface*

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.12. 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.13. Notasi *Use Case*

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

a. Nama *Use Case*

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

b. Deskripsi singkat

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

c. 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*.

d. 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.

e. *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.

f. *Pre-Condition*

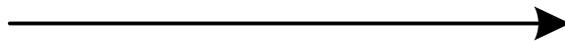
Menjelaskan persyaratan yang harus dipenuhi sebelum *use case* bisa dimulai.

g. *Post-Condition*

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

5. *Interaction*

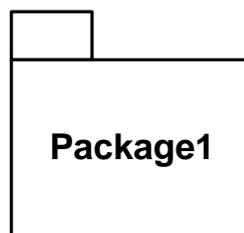
Digunakan untuk menunjukkan baik aliran pesan maupun informasi antara objek maupun antara hubungan objek. Biasanya *Interaction* dilengkapi juga dengan teks bernama *operation signature* yang tersusun dari mana operasi, parameter yang dikirim dan type parameter yang dikembalikan.



Gambar II.14. Notasi *Interaction*

6. *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 pengelihatn dari model yang sedang dibangun.



Gambar II.15. 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.16. 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, 2), 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, 2), 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 redundansi dan inkonsistensi data

Redudansis terjadi jika suatu informasi disimpan di beberapa 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 :

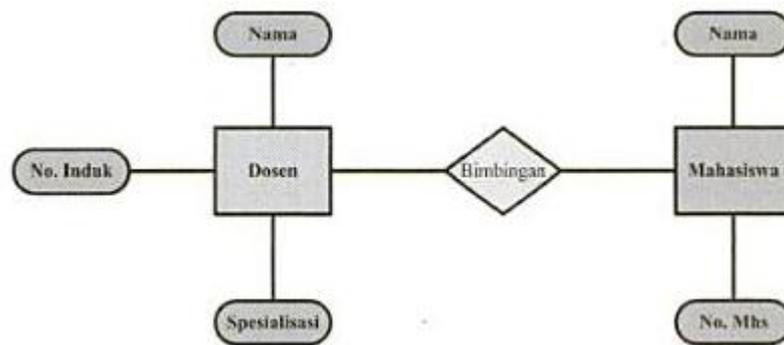
1. Pembuatan basis data
2. Penghapusan basis data
3. Pembuatan file/tabel
4. Penghapusan file/tabel
5. Pengubahan tabel
6. Penambahan/pengisian
7. Pengambilan data
8. Penghapusan data

II.6.4. Pemodelan Basis Data

Menurut Samiaji Sarosa (2010:4), Model diperlukan untuk mendapatkan penyederhanaan dari kenyataan dan memungkinkan desainer program program aplikasi bereksperimen dengan berbagai macam variable sebelum diaplikasikan ke system yang berjalan. Untuk merancang suatu aplikasi basis data alat yang biasa

digunakan adalah *Entity Relationship Diagram (ERD)*. ERD didasarkan dari artikel yang dipublikasikan oleh Peter Phin Shan Chen.

Ada beberapa case tool menamakan notasi ERD yang digunakan sebagai chen ERD. *Entity Relationship Model* adalah abstraksi konseptual yang mewakili struktur dari suatu basis data.

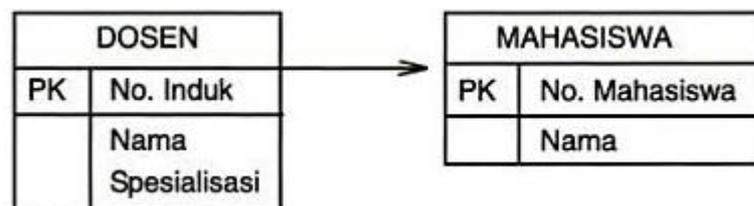


Gambar II.17. Diagram Dengan Notasi Chen ERD

Dalam perkembangannya banyak diciptakan notasi ERD yang berbeda-beda seperti terlihat gambar dibawah ini.



Gambar II.18. Diagram Dengan Notasi Crows Foot



Gambar II.19. Diagram Dengan Notasi Relational

II.6.5. 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. Visual Basic 2010

Menurut Edi Winarno (2010:1), Visual Basic adalah bahasa pemrograman klasik, legendaris yang paling banyak dipakai oleh programmer didunia. 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.

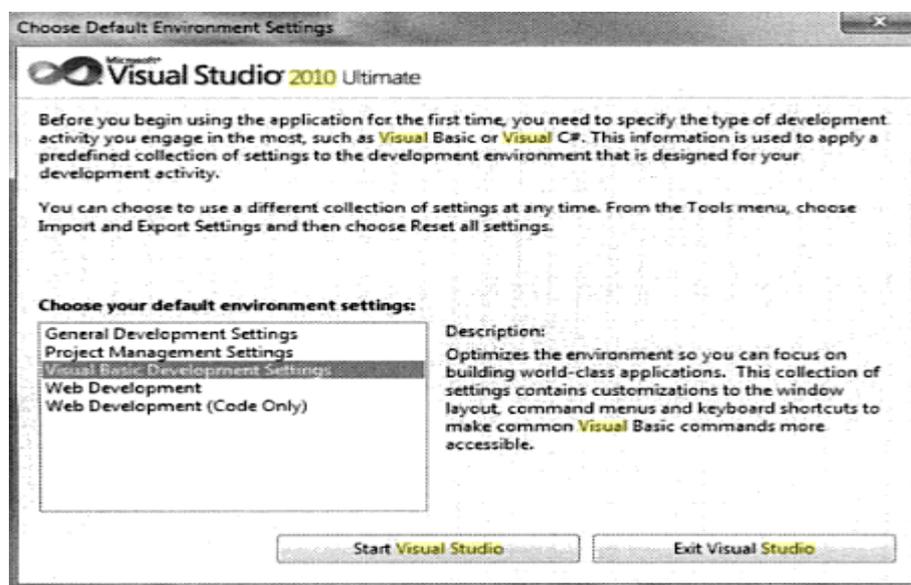
Beberapa kemampuan atau manfaat dari Visual Studio 2010 diantaranya seperti :

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

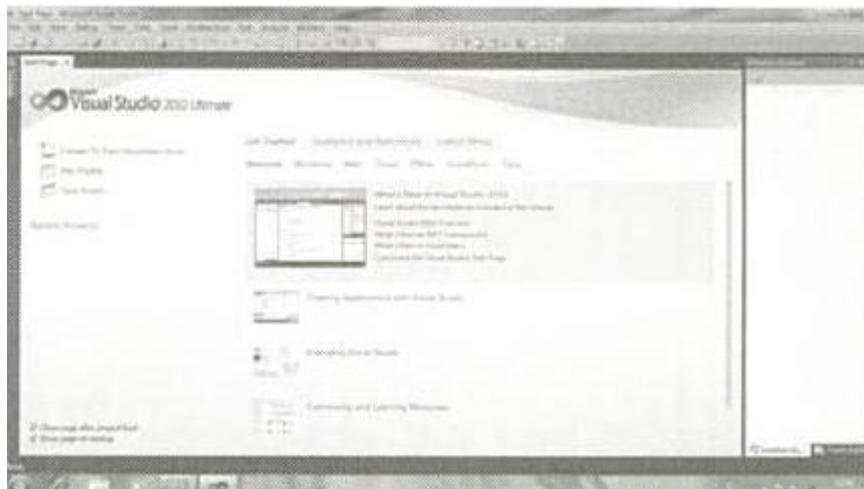
Saat menjalankan Visual Basic 2010 pertama kali muncul jendela *chosedefault environment settings*. Disini bisa memilih apakah ingin memilih antar muka di Visual Studio. Untuk *programmer* Visual Basic lebih baik memilih *Visual Basic Development Centre*.



Gambar II.20. Form Chose Default Environment Settings

(Sumber : Edi Winarno,et al ; 2010)

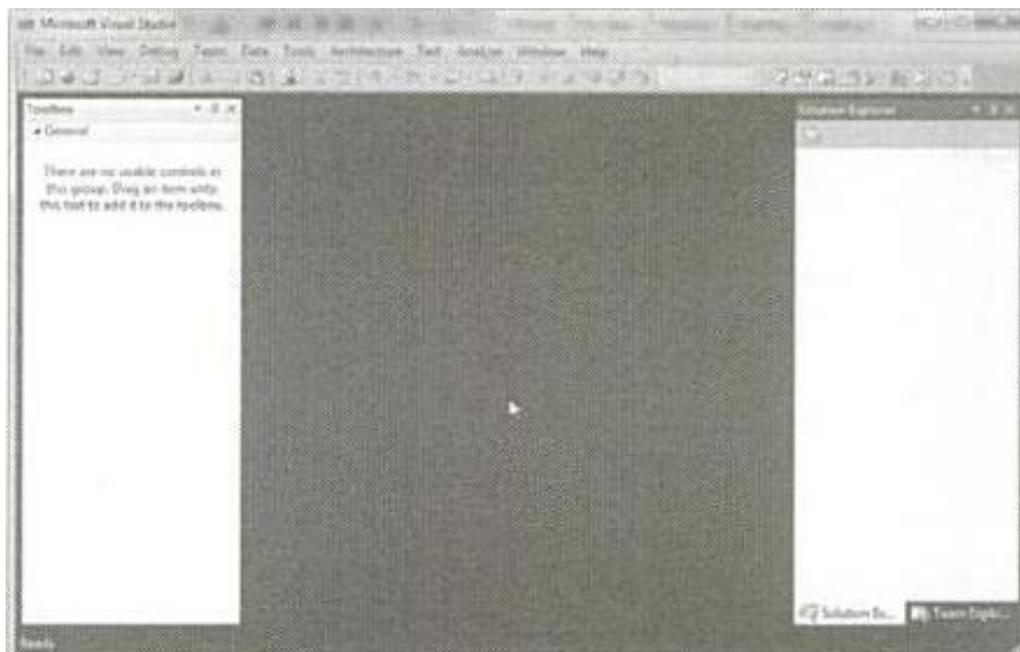
Dibagian awal visual basic, bisa memilih *Start Page*. *Start Page* adalah halaman yang mencantumkan informasi-informasi seputar program dan juga informasi RSS dari sumber tertentu. Jika tidak ingin menampilkan hal ini hilangkan tanda centang pada *Show Page On Startup*.



Gambar II.21. Start Page Visual Basic 2010

(Sumber : Edi Winarno, et al ; 2010)

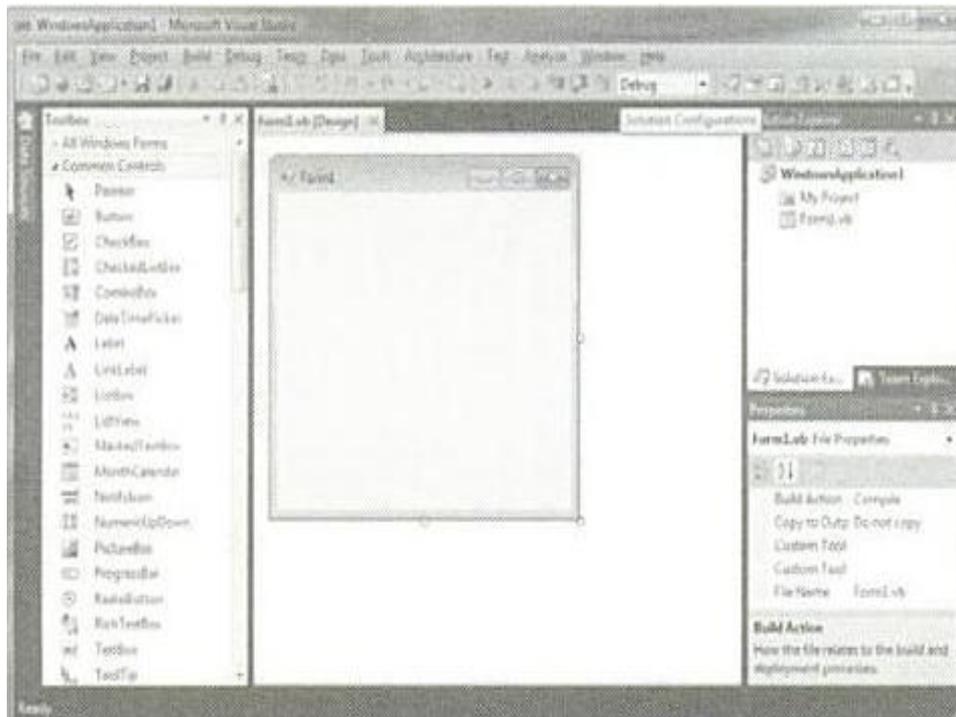
Jika *start page* ditutup terlihat tampilan sebagai berikut :



Gambar II.22. Tampilan IDE (*Integrated Development Environment*) setelah *Start Page* ditutup

(Sumber : Edi Winarno, et al ; 2010)

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



Gambar II.23. Tampilan lengkap IDE
(Sumber : Edi Winarno, et al ; 2010)

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 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 :

- a. Mode Windows, Pada mode ini SQL Server akan melakukan autentifikasi dengan menggunakan level login pada system operasi.
- b. 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* pada SQL Server (Wahana Komputer, 2010:2).