

BAB II

LANDASAN TEORI

II.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 “*Decission 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.1.1 Karakteristik Sistem Pendukung Keputusan

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

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

II.1.2 Ciri-Ciri Sistem Pendukung Keputusan

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

- a. Banyak pilihan/alternatif
- b. Ada kendala atau surat
- c. Mengikuti suatu pola atau model tingkah laku, baik yang terstruktur maupun tidak terstruktur.
- d. Banyak input/Variabel
- e. Ada faktor resiko, dibutuhkan kecepatan, ketepatan dan keakuratan .

II.1.3 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 peng enalan 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 tahai 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
Sumber : Dicky Nofriansyah, (2014:2)

II.1.4 Komponen Sistem Pendukung Keputusan

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

a. 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*)

b. 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 dijaga fleksibilitasnya. Hal lain yang harus diperhatikan adalah pada setiap model yang disimpan hendaknya ditambahkan rincian, keterangan dan penjelasan yang komprehensif mengenai model yang dibuat.

c. 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.1.5 Tujuan Sistem Pendukung Keputusan

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

- a. Membantu dalam pengambilan keputusan atas masalah yang terstruktur
- b. Memberikan dukungan atas pertimbangan manager dan bukannya dimaksudkan untuk menggantikan fungsi manager.
- c. Meningkatkan efektifitas keputusan yang diambil lebih dari perbaikan efesiensinya.
- d. Kecepatan komputasi komputer memungkinkan para pengambil keputusan untuk banyak melakukan komputasi secara cepat dengan biaya yang sangat rendah.
- e. 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.1.6 Konsep Dasar Sistem Pendukung Keputusan

Sistem Pendukung Keputusan (SPK) mulai dikembangkan pada tahun 1960-an, tetapi istilah sistem pendukung keputusan itu sendiri baru muncul pada tahun 1971, yang diciptakan oleh G. Anthony Gorry dan Micheal S.Scott Morton, keduanya adalah profesor di MIT. Hal itu mereka lakukan dengan tujuan untuk

menciptakan kerangka kerja guna mengarahkan aplikasi komputer kepada pengambilan keputusan manajemen.

Sementara itu, perintis sistem pendukung keputusan yang lain dari MIT, yaitu Peter G.W. Keen yang bekerja sama dengan Scott Morton telah mendefinisikan tiga tujuan yang harus dicapai oleh sistem pendukung keputusan, yaitu:

1. Sistem harus dapat membantu manajer dalam membuat keputusan guna memecahkan masalah semi terstruktur.
2. Sistem harus dapat mendukung manajer, bukan mencoba menggantikannya.
3. Sistem harus dapat meningkatkan efektivitas pengambilan keputusan manajer.

Tujuan-tujuan tersebut mengacu pada tiga prinsip dasar sistem pendukung keputusan, yaitu:

a. Struktur masalah

Untuk masalah yang terstruktur, penyelesaian dapat dilakukan dengan menggunakan rumus-rumus yang sesuai, sedangkan untuk masalah terstruktur tidak dapat dikomputerisasi. Sementara itu, sistem pendukung keputusan dikembangkan khususnya untuk menyelesaikan masalah yang semi-terstruktur.

b. Dukungan keputusan

Sistem pendukung keputusan tidak dimaksudkan untuk menggantikan manajer, karena komputer berada di bagian terstruktur, sementara manajer berada di bagian tak terstruktur untuk memberikan penilaian dan melakukan

analisis. Manajer dan komputer bekerja sama sebagai sebuah tim pemecah masalah semi terstruktur.

c. Efektivitas keputusan

Tujuan utama dari sistem pendukung keputusan bukanlah mempersingkat waktu pengambilan keputusan, tetapi agar keputusan yang dihasilkan dapat lebih baik.

II.1.7. Pengertian Keputusan

Beberapa definisi keputusan yang dikemukakan para ahli dijelaskan sebagai berikut:

1. Menurut Ralph C. Davis

Keputusan adalah hasil pemecahan masalah yang dihadapinya dengan tegas. Suatu keputusan merupakan jawaban yang pasti terhadap suatu pertanyaan. Keputusan harus dapat menjawab pertanyaan tentang apa yang dibicarakan dalam hubungannya dengan perencanaan. Keputusan dapat pula berupa tindakan terhadap pelaksanaan yang sangat menyimpang dari rencana semula.

2. Menurut Mary Follet

Keputusan adalah suatu atau sebagai hukum situasi. Apabila semua fakta dari situasi itu dapat diperolehnya dan semua yang terlibat, baik pengawas maupun pelaksana mau mentaati hukumnya atau ketentuannya, maka tidak sama dengan mentaati perintah. Wewenang tinggal dijalankan, tetapi itu merupakan wewenang dari hukum situasi.

3. Menurut James A.F.Stoner

Keputusan adalah pemilihan di antara alternatif-alternatif. Definisi ini mengandung tiga pengertian, yaitu:

1. Ada pilihan atas dasar logika atau pertimbangan.
2. Ada beberapa alternatif yang harus dan dipilih salah satu yang terbaik.
3. Ada tujuan yang ingin dicapai, dan keputusan itu makin mendekatkan pada tujuan tertentu.

4. Menurut Prof. Dr. Prajudi Atmosudirjo, SH

Keputusan adalah suatu pengakhiran daripada proses pemikiran tentang suatu masalah atau problem untuk menjawab pertanyaan apa yang harus diperbuat guna mengatasi masalah tersebut, dengan menjatuhkan pilihan pada suatu alternatif

Dan pengertian keputusan diatas, dapat ditarik suatu kesimpulan bahwa keputusan merupakan suatu pemecahan masalah sebagai suatu hukum situasi yang dilakukan melalui pemilihan satu alternatif dari beberapa alternatif (Hasan, 2004).

II.1.8. Kriteria Sistem Pendukung Keputusan

Sistem pendukung keputusan dirancang secara khusus untuk mendukung seseorang yang harus mengambil keputusan-keputusan tertentu. Berikut ini beberapa karakteristik sistem pendukung keputusan :

1. Interaktif

Sistem pendukung keputusan (SPK) memiliki user interface yang komunikatif sehingga pemakai dapat melakukan akses secara cepat ke data dan memperoleh informasi yang dibutuhkan.

2. Fleksibel

Sistem pendukung keputusan (SPK) memiliki sebanyak mungkin variabel masukan, kemampuan untuk mengolah dan memberikan keluaran yang menyajikan alternatif-alternatif keputusan kepada pemakai.

3. Data kualitas

Sistem pendukung keputusan (SPK) memiliki kemampuan menerima data kualitas yang dikuantitaskan yang sifatnya subyektif dari pemakainya, sebagai data masukan untuk pengolahan data. Misalnya: penilaian terhadap kecantikan yang bersifat kualitas, dapat dikuantitaskan dengan pemberian bobot nilai seperti 75 atau 90.

4. Prosedur Pakar

Sistem pendukung keputusan (SPK) mengandung suatu prosedur yang dirancang berdasarkan rumusan formal atau juga beberapa prosedur kepakaran seseorang atau kelompok dalam menyelesaikan suatu bidang masalah dengan fenomena tertentu (Riki Cukil Nuryadin 2011).

II.1.9 Pengertian Pengambilan Keputusan

Beberapa definisi pengambilan keputusan yang dikemukakan para ahli dijelaskan sebagai berikut :

1. Menurut George R. Terry (2001,55)

Pengambilan keputusan adalah pemilihan alternatif perilaku (kelakuan) tertentu dari dua atau lebih alternatif yang ada.

2. Menurut S.P. Siagian (2002,40)

Pengambilan keputusan adalah suatu pendekatan yang sistematis terhadap hakikat alternatif yang dihadapi dan mengambil tindakan yang menurut perhitungan merupakan tindakan yang paling tepat.

3. Menurut James A.F. Stoner

Pengambilan keputusan adalah proses yang digunakan untuk memilih suatu tindakan sebagai cara pemecahan masalah. Dari pengertian-pengertian pengambilan keputusan diatas, dapat disimpulkan bahwa pengambilan keputusan merupakan suatu proses pemilihan alternatif terbaik dari beberapa alternatif secara sistematis untuk ditindaklanjuti (digunakan) sebagai suatu cara pemecahan masalah (Hasan, 2004).

II.2 Metode *Weighted Product Model*

Menurut Putra Jaya (2012), Metode *Weighted Product* memerlukan proses normalisasi karena metode ini menguluhkan hasil penilaian setiap atribut. Hasil perkalian tersebut belum bermakna jika belum dibandingkan (dibagi) dengan nilai standart. Bobot untuk atribut manfaat berfungsi sebagai pangkat positif dalam proses perkalian, sementara bobot biaya berfungsi sebagai pangkat negatif.

Metode WP Merupakan metode pengambilan keputusan dengan cara perkalian untuk menghubungkan rating atribut, dimana rating setiap atribut harus dipangkatkan dulu dengan bobot atribut yang bersangkutan. WP adalah salah satu analisis multi-kriteria keputusan (multi-criteria decision analysis / MCDA) yang sangat terkenal. Metode multi-kriteria pengambilan keputusan multi-criteria

decision making (MCDM). Metode MCDA, yang diberikan adalah satu set terbatas dari alternatif keputusan yang dijelaskan dalam hal sejumlah kriteria keputusan. Setiap alternatif keputusan dibandingkan dengan yang lain dengan mengalikan sejumlah rasio, satu untuk setiap kriteria keputusan. Setiap rasio diangkat ke kekuasaan setara dengan berat relatif dari kriteria yang sesuai.

Metode *Weighted Product* menggunakan perkalian untuk menghubungkan rating atribut, dimana rating setiap atribut harus dipangkatkan dulu dengan bobot atribut yang bersangkutan. Langkah-langkah dalam menggunakan metode ini adalah :

1. Menentukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan.
2. Menentukan rating kecocokan setiap alternatif pada setiap kriteria.
3. Menentukan bobot preferensi tiap kriteria.
4. Mengalikan seluruh atribut bagi sebuah alternatif dengan bobot sebagai pangkat positif untuk atribut keuntungan dan bobot berpangkat negatif untuk atribut biaya.

Metode *Weighted Product* menggunakan perkalian sebagai untung menghubungkan rating atribut, dimana rating setiap atribut harus dipangkatkan dulu dengan bobot yang bersangkutan. Proses ini sama halnya dengan proses normalisasi. Preferensi untuk *alternative* Si diberikan sebagai berikut :

1. Penentuan nilai bobot W

$$w_j = \frac{w_j}{\sum w_j} \dots\dots\dots (1)$$

2. Penentuan nilai Vektor S

$$S = (W_{ij}^{AWJ} \cdot w) \cdot (W_{in}^{Awn} \cdot w) \dots\dots\dots (2)$$

3. Penentuan nilai Vektor V

$$V_{jn} = \frac{S_i}{\sum S_i} \dots\dots\dots (3)$$

Dimana :

V = *Preferensi* alternatif dianalogikan sebagai vektor V

W = Bobot kriteria / subkriteria

j = Kriteria

i = Alternatif

n = Banyaknya kriteria

S = *Preferensi* alternatif dianalogikan sebagai vektor S

Langkah – langkah dalam perhitungan metode *Weighted Product* adalah sebagai berikut :

1. Mengalihkan seluruh atribut bagi seluruh alternative dengan bobot sebagai pangkat positif bagi atribut biaya.
2. Hasil perkalian dijumlahkan untuk menghasilkan nilai pada setiap alternative
3. Membagi nilai V bagi setiap alternatif dengan nilai pada setiap alternative
4. Ditemukan urutan alternatif terbaik yang akan menjadi keputusan.

II.3. 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.

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 sudut 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.3.1. Tujuan Basis Data

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 :

- a. Tidak adanya redudansi dan inkonsistensi data

Redudansi 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.

b. 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.

c. *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.3.2. Manfaat/Kelebihan Basis Data

Banyak manfaat yang diperoleh dengan menggunakan basis data, Manfaat/Kelebihan Basis Data dan kelebihan basis data diantaranya adalah :

a. 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 metematika. Dengan perancangan yang benar maka penyajian informasi dapat dilakukan dengan cepat dan mudah.

b. 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.

c. Pemusatan kontrol data

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

d. 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.

e. 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.

f. 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.

g. 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.

h. 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.

i. Pemakaian secara langsung

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

j. Kebebasan data

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

k. *User View*

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

II.3.3. Operasi Dasar Database

Beberapa operasi dasar basis data yaitu :

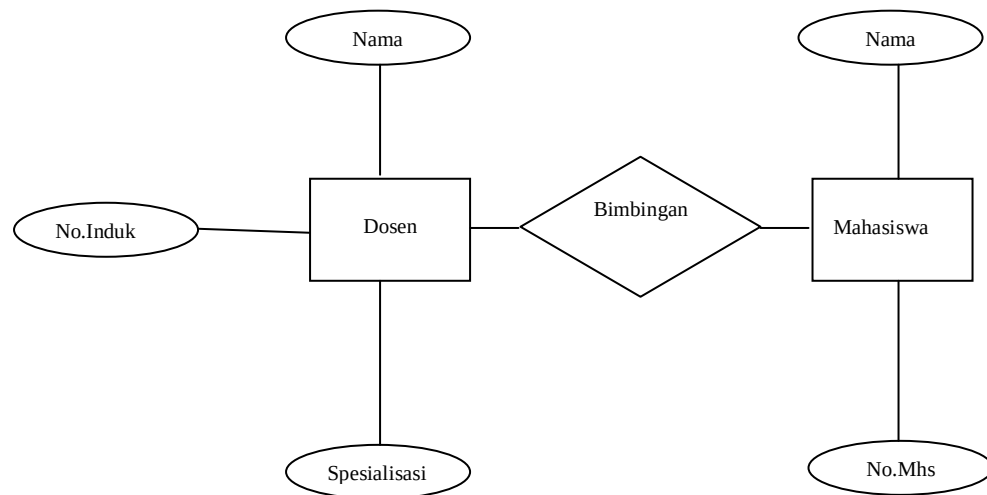
- a. Pembuatan basis data
- b. Penghapusan basis data
- c. Pembuatan file/tabel
- d. Penghapusan file/tabel
- e. Pengubahan tabel
- f. Penambahan/pengisian
- g. Pengambilan data

h. Penghapusan data

II.3.4. Pemodelan Basis Data

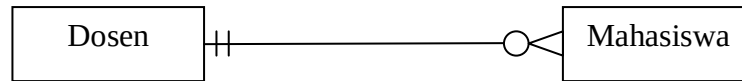
Menurut Samiaji Sarosa (2010:4), Model diperlukan untuk mendapatkan penyederhanaan dari kenyataan dan memungkinkan desainer program aplikasi bereksperimen dengan berbagai macam variable sebelum diaplikasikan ke system yang berjalan. Untuk merancang suatu aplikasi basis data alat yang biasa digunakan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

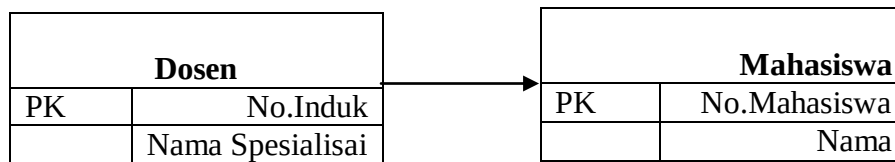


Gambar II.2. Diagram Dengan Notasi Chen ERD
Sumber : Samiaji Sarosa (2010:5)

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



Gambar II.3. Diagram Dengan Notasi Crows Foot
Sumber : Samiaji Sarosa (2010:5)



Gambar II.4. Diagram Dengan Notasi Relational
Sumber : Samiaji Sarosa (2010:5)

II.3.5. Normalisasi

Menurut Samiaji Sarosa (2010:6), 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 :

a. *Insertion Anomaly*

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

b. *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.4. Visual Basic 2010

Menurut Edi Winarno, (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.

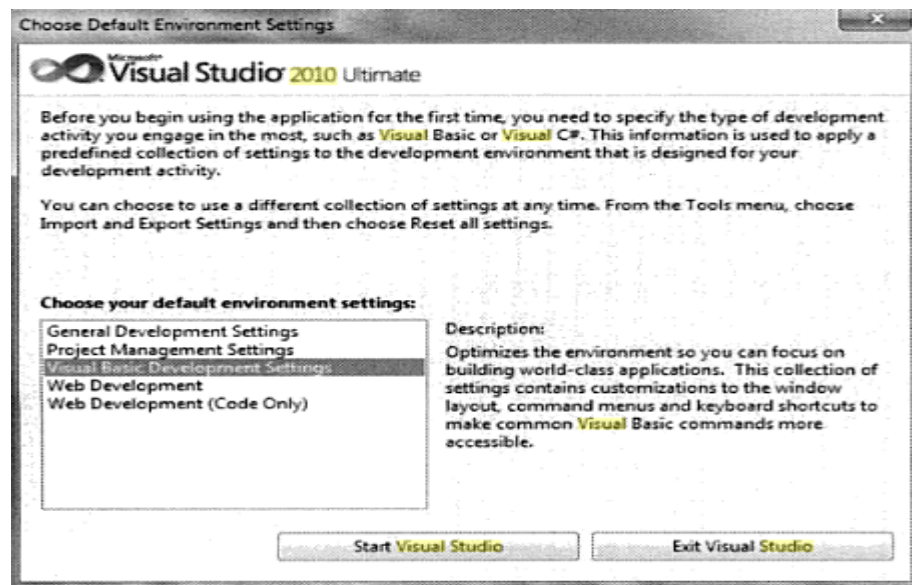
Visual Studio 2010 pada dasarnya adalah sebuah bahasa pemrograman komputer. Dimana pengertian dari bahasa pemrograman itu adalah perintah-perintah atau instruksi yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu. Visual Studio 2010 selain disebut dengan bahasa pemrograman, juga sering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

Beberapa kemampuan atau manfaat dari Visual Studio 2010 diantaranya seperti :

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.

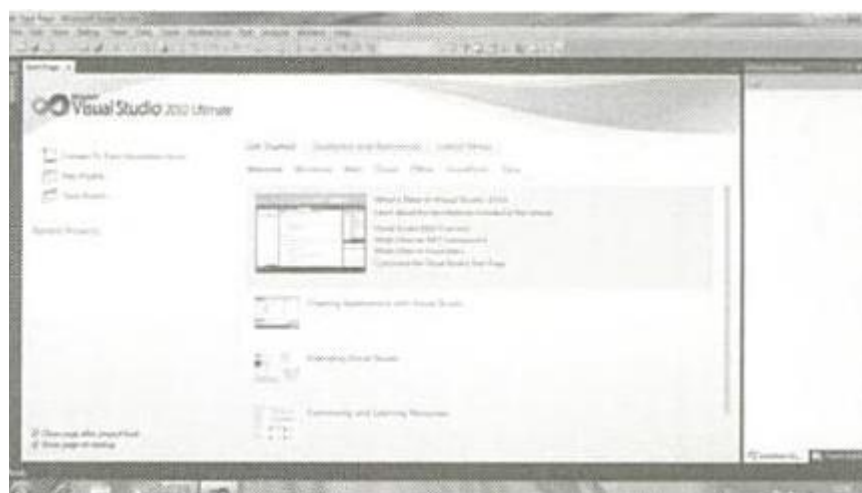
II.4.1. Antar Muka Visual Basic 2010

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



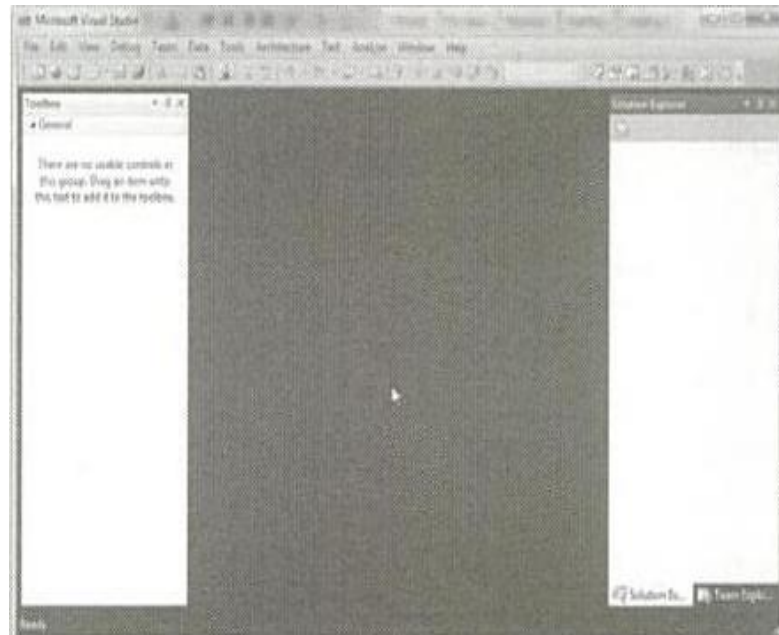
Gambar II.5 Form Chose Default Environtment Settings
Sumber : Edi Winarno(2010:30)

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.6. Start Page Visual Basic 2010
Sumber : Edi Winarno(2010:31)

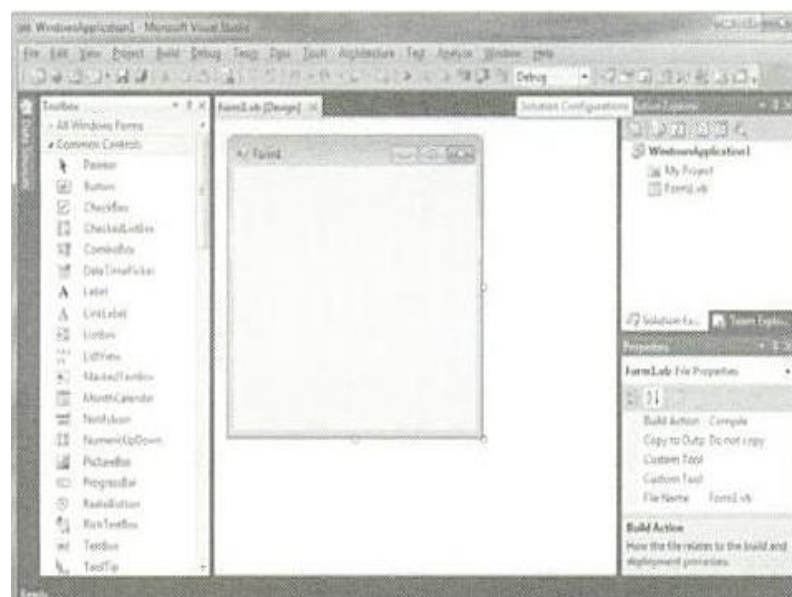
Jika *start page* ditutup terlihat tampilan sebagai berikut :



Gambar II.7. Tampilan IDE (Integrated Development Environment) setelah *Start Page* ditutup

Sumber : Edi Winarno(2010:32)

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



Gambar II.8. Tampilan lengkap IDE

Sumber : Edi Winarno(2010:33)

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 , 2010:1)

II.5. 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.5.1 Kebutuhan *Hardware*

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

- a. Prosescor minimal 1 GHz
- b. Memori minimal 512 MB
- c. Sistem Operasi Windows

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

- a. *Sharer Memory*
- b. TCP/IP
- c. *Named Pipes*
- d. *Virtual Interface Adapter (VIA)*(Wahana Komputer, 2010:2).

II.5.2 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 adalag sebagai berikut :

- a. Menurut cara pemrosesan data pada prosesor makan Microsoft mengelompokkan produk ini berdasarkan dua jenis yaitu :
 - 1. Versi 32 Bit (x86), yang biasanya digunakan untuk komputer *single processor* (Pentium 4) atau lebih tepatnya processor 32 bit atau Windows XP
 - 2. Versi 64 Bit (x64), yang biasanya digunakan oleh computer yang lebih sari satu processor (Misalnya *Core 2 duo*) dan system operasi 64 bit, Vista dan Windows 7.

- b. Sedangkan secara keseluruhan terdapat versi-versi seperti berikut :
1. Versi *Compact* ini adalah versi “tipis” dari semua versi yang ada
 2. Versi *Express* ini adalah versi “ringan”

II.5.3 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)

II.6. *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.6.1. 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 system 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.6.2. 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

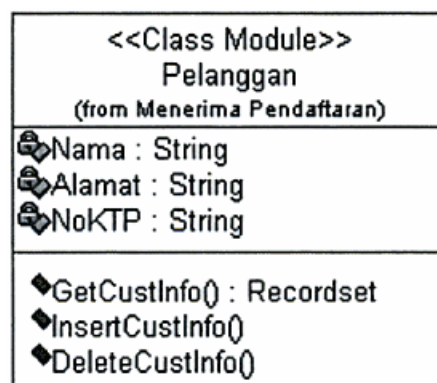
Aktor adalah segala sesuatu yang berinteraksi dengan system aplikasi computer. Jadi actor ini bisa berupa orang, perangkat keras atau juga objek lain dalam system yang sama. Biasanya yang dilakukan oleh Aktor adalah memberikan informasi pada system dan\atau memerintahkan system untuk melakukan sesuatu.



Gambar II.9. Notasi Aktor
Sumber : Julius Hermawan (2010:14)

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*



Gambar II.10. Notasi Class
Sumber : Julius Hermawan (2010:14)

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.11. Notasi Interface
Sumber : Julius Hermawan (2010:15)

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.12. Notasi Use Case
Sumber : Julius Hermawan (2010:16)

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*

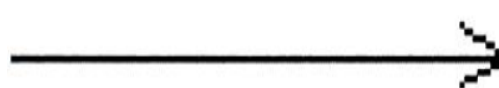
Menjelaskan persyaratan yang harus dipenuhi sebelum *use case* bisa dimulai.

g. *Post-Condition*

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

5. *Interaction*

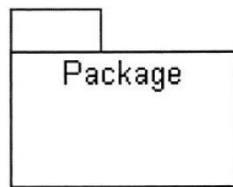
Digunakan untuk menunjukan baik aliran pesan maupun informasi antara objek maupun antara hubungan objek. Biasanya *Interaction* dilengkapi juga dengan teks bernama *operation signature* yang tersusun dari mana operasi, parameter yang dikirim dan type parameter yang dikembalikan.



Gambar II.13. Notasi *Interaction*
Sumber : Julius Hermawan (2010:18)

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 pengelihatian dari model yang sedang dibangun.



Gambar II.14. Notasi *Package*
Sumber : Julius Hermawan (2010:19)

7. *Note*

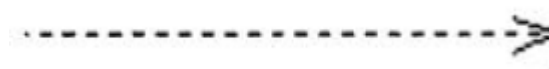
Note digunakan untuk memberikan keterangan dan komentar tambahan dari suatu elemen sehingga bisa langsung terlampir dalam model. *Note* ini bisa ditempelkan ke semua elemen notasi yang lain.



Gambar II.15. Notas *Note*
Sumber : Julius Hermawan (2010:19)

8. *Dependency*

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



Gambar II.16. Notasi *Dependency*
Sumber : Julius Hermawan (2010:20)