

## **BAB II**

### **LANDASAN TEORI**

#### **II.1. Sistem Pendukung Keputusan**

##### **II.1.1. Pengertian Sistem Pendukung Keputusan**

Sistem Pendukung Keputusan (SPK) adalah sistem yang berbasis komputer yang interaktif, yang membantu pengambilan keputusan memanfaatkan data dan model untuk menyelesaikan masalah-masalah tak terstruktur dan semi struktur (I Beerawa, dkk : 2013 ; 4).

Sistem Pendukung Keputusan (SPK) adalah sistem yang berbasis model yang terdiri dari prosedur-prosedur dalam pemrosesan data dan pertimbangannya untuk membantu manajer dalam mengambil keputusan (I Beerawa, dkk : 2013 ; 4).

Sistem Pendukung Keputusan (*Decision Support Systems*) adalah sebuah sistem yang dimaksudkan untuk mendukung para pengambil keputusan manajerial dalam situasi keputusan semiterstruktur (Muqtadir, Purdianto : 2013 ; 2).

*Decision Support System* atau Sistem Pendukung Keputusan (SPK) didefinisikan secara luas sebagai sebuah sistem berbasis komputer yang membantu orang-orang untuk menggunakan komunikasi komputer, data, dokumen, pengetahuan dan model untuk mengatasi masalah dan membuat keputusan. SPK adalah sistem tambahan atau sistem pembantu. SPK tidak dimaksudkan untuk menggantikan ahli pengambil keputusan (Pembayun, dkk : 2007 ; 3).

Komponen utama dari sebuah Sistem Pendukung Keputusan adalah DBMS, MBMS, dan antarmuka pengguna. *Database* mewakili subsistem manajemen data yang dibangun secara khusus untuk menyimpan data pegawai, dan master kriteria, data hasil penghitungan nilai tes tertulis, microteaching, wawancara, dan psikotest. DBMS yang digunakan SQL Server (Pambayun, dkk : 2007 ; 5).

*Decision Support System* didefinisikan sebagai sistem komputer yang mampu memberikan kemampuan yang baik kemampuan pemecahana masalah maupun kemampuan pengkomunikasian untuk masalah semi terstruktur. Secara khusus, DSS didefinisikan sebagai suatu sistem yang mendukung kerja seorang manager mampu sekelompok manager dalam memecahkan masalah semi terstruktur dengan cara memberikan informasi ataupun usulan menuju pada keputusan tertentu.

DSS ini bisa berbentuk sistem manual maupun sistem terkomputeriasi. Namun dalam buku ini ditekankan pada sistem penunjang keputusan yang pelaksanaannya berbasis pada komputer.

Dari definisi diatas bisa disimpulkan bahwa tujuan DSS dalam proses pengambilan keputusan adalah :

1. Membantu menjawab masalah semi-terstruktur.
2. Membantu manager dalam pengambilan keputusan, bukan menggantikannya
3. Meningkatkan efektifitas pengambilan keputusan seorang majer dari pada efisiennya.

a. Karakteristik Sistem Pendukung Keputusan

Karakteristik dari sistem pendukung keputusan (*Decision Support System*) adalah sebagai berikut :

1. Tujuan utama dari Sistem Pendukung Keputusan adalah untuk memperbaiki mutu keputusan serta performance. Sistem Pendukung Keputusan tidak hanya sekedar menyajikan informasi yang lebih banyak, lebih baik dan lebih akurat pada waktu yang tepat saja.
2. Sistem Pendukung Keputusan ditujukan untuk environment yang kompleks, kurang terstruktur dan bahkan politis sifatnya.
3. Sistem Pendukung Keputusan bertumpu pada laporan perkecualian dan macamnya untuk menunjang proses identifikasi masalah.
4. Sistem Pendukung Keputusan berkombinasi “*modelling*” dan teknik-teknik analisa yang lain dengan fungsi penyajian kembali data.
5. Sistem Pendukung Keputusan berfokus pada prinsip “mudah dipakai” dan “fleksibel” dalam berhadapan dengan pemakai tertentu atau sekelompok pemakai.
6. Proses pengambilan keputusan.

b. Konfigurasi Sistem Pendukung Keputusan (*Decision Support System*)

Terdapat 3 komponen utama, yaitu data management, model management, dan *user interface*.

Aliran kerja dari DSS ini adalah sebagai berikut:

1. Data management melakukan pengambilan data yang diperlukan baik dari database yang berisi data internal maupun database yang berisi data eksternal. Jadi, fungsi komponen data disini jelas sebagai penyedia data yang diperlukan oleh sistem DSS.
2. Model management melalui modelbase management melakukan interaksi baik dengan *use interface* untuk mendapatkan perintah maupun data management untuk mendapatkan data yang diolah. Model base management akan menggunakan model base yang berisi model-model yang digunakan dalam DSS yang dengan bantuan perangkat lunak atau engine yang ada didalam modeling tools melakukan pengolahan data yang kemudian hasilnya dikembalikan lewat model base management untuk dikirim ke *user interface*.
3. *Use interface* digunakan untuk berinteraksi antar user dengan DSS, baik untuk memasukkan informasi ke sistem maupun menampilkan informasi ke user. Karena begitu pentingnya komponen *user interface* bagi suatu sistem DSS, maka kita harus merancang suatu *user interface* yang bisa mudah dipelajari dan digunakan user dan

laporan yang bisa secara mudah dimengerti oleh pengguna  
(V.M.Christian S : 2014 ; 4-5).

### **II.1.2. Konsep Dasar Sistem Pendukung Keputusan**

Defenisi sistem adalah sekumpulan hal atau kegiatan atau elemen subsistem yang saling bekerja sama atau yang dihubungkan dengan cara-cara tertentu sehingga membentuk satu kesatuan untuk melaksanakan suatu fungsi guna mencapai suatu tujuan (I Beerawa, dkk : 2013 ; 4).

Proses-proses yang terjadi pada kerangka kerja *Decision Support* dibedakan atas :

1. Terstruktur, mengacu pada permasalahan rutin dan berulang untuk solusi standar yang ada.
2. Tak terstruktur, adalah “*Fuzzy*” permasalahan kompleks dimana tak ada solusi serta merta. Masalah yang tak terstruktur adalah tak adanya 3 fase proses yang terstruktur.
3. Semi terstruktur, terdapat beberapa keputusan terstruktur, tetapi tak semuanya dari fase-fase yang ada.(I Beerawa : 2013 ; 3).

Aplikasi sistem pendukung keputusan bisa terdiri dari beberapa subsistem :

#### **1. Subsistem Manajemen Data**

Subsistem manajemen data memasukkan satu *database* yang berisi data yang relevan untuk suatu situasi dan dikelola oleh perangkat lunak yang disebut sistem manajemen *database* (DBMS).

## **2. Subsistem Manajemen Model**

Merupakan paket perangkat lunak yang memasukkan model keuangan, statistik ilmu manajemen, atau model kuantitatif lain yang memberikan kapasitas analistik dan manajemen perangkat lunak yang tepat. Bahasa-bahasa pemodelan untuk membangun model-model kustom juga dimasukkan. Perangkat lunak itu juga sering disebut sistem manajemen basis model (MBMS). Komponen tersebut bisa dikondisikan ke penyimpanan korporat atau eksternal yang ada pada model.

## **3. Subsistem Antarmuka Pengguna**

Pengguna berkomunikasi dengan memerintahkan DBS melalui subsistem ini.

## **4. Subsistem Manajemen Berbasis Pengetahuan**

Banyak masalah terstruktur dan bahkan semi terstruktur yang sangat kompleks sehingga solusinya memerlukan keahlian. Keahlian tersebut dapat diberikan oleh sistem cerdas lainnya. Oleh karena itu makin banyak DSS canggih yang dilengkapi dengan satu komponen yang disebut *subsistem manajemen berbasis pengetahuan*. Komponen ini dapat menyediakan keahlian yang diperlukan untuk memecahkan beberapa aspek masalah dan memberikan pengetahuan yang dapat meningkatkan operasi komponen DSS yang lain (Pambayun : 2007 ; 3).

## **II.2. Keputusan**

### **II.2.1. Pengertian Keputusan**

Keputusan adalah suatu proses yang dilaksanakan seseorang berdasarkan pengetahuan dan informasi yang ada dengan harapan atau tujuan. Keputusan dapat diambil dari beberapa alternatif-alternatif yang ada (Nurul : 2010 ; 16).

### **II.2.2. Jenis – Jenis Keputusan**

Keputusan – keputusan yang dibuat pada dasarnya dikelompokkan dalam dua jenis, antara lain(Herbert A.Simon) :

#### **1. Keputusan Terprogram**

Keputusan terprogram adalah keputusan yang dapat dispesifikasikan sebelumnya sebagai seperangkat aturan atau prosedur keputusan.

#### **2. Keputusan Tidak Terprogram**

Keputusan tidak terprogram adalah keputusan yang terjadi hanya satu kali atau berubah-ubah setiap saat ketika diperlukan.

Keputusan dalam suatu sistem keputusan terbuka (berada dalam suatu lingkungan yang rumit dan sebagian tak diketahui) adalah merupakan keputusan tidak terprogram karena tidak mungkin menspesifikasikan semua faktor – faktornya sebelum melakukan pengambilan keputusan (Dahria Muhammad, et al : Mei 2014).

Sistem Pendukung keputusan (*SPK*) merupakan sebuah sistem yang memiliki kriteria sebagai berikut :

1. Penggunaan model, komunikasi antara pengambil keputusan dan sistem terjalin melalui model – model matematis, jadi pengambil keputusan bertanggung jawab membangun model matematis berdasarkan permasalahan yang dihadapinya.
2. Berbasis komputer, sistem ini mempertemukan penilaian manusia (pengambilan keputusan) dengan informasi komputer. Informasi komputer ini berasal dari perangkat lunak komputer yang merupakan implementasi dari metode numeris untuk permasalahan matematis yang bersangkutan.
3. Fleksibel, sistem harus dapat beradaptasi terhadap timbulnya perubahan pada model yang telah diberikannya kepada sistem, ataupun memberikan model baru.
4. Interaktif dan mudah digunakan, pengambil keputusan bertanggung jawab untuk menentukan apakah jawaban yang diberikan oleh sistem memuaskan atau tidak. Bagaimana pun juga sistem bertugas mendukung, bukan menggantikan pengambil keputusan. Jadi sistem harus memiliki kemampuan interaktif : pengambil keputusan harus diijinkan untuk menjelajahi alternatif pengambil keputusan harus diijinkan untuk menjelajahi alternatif jawaban dengan cara memvariasi parameter – parameter yang ada pada sistem.

### II.3. Profile Matching

Pencocokan profil (*profile matching*) adalah sebuah mekanisme pengambilan *keputusan* dengan mengamsumsikan bahwa terdapat tingkat variabel prediktor yang ideal yang harus dimiliki oleh pelamar, bukannya tingkat minimal yang harus dipenuhi atau dilewati.

Dalam proses *profile matching*, akan dilakukan proses perbandingan antara kompetensi individu ke dalam kompetensi standar. Semakin kecil gap yang dihasilkan maka bobot nilainya semakin besar. Calon yang memiliki bobot nilai yang besar berarti memiliki peluang lebih besar untuk dapat menempati posisi sebagai manager (Pambayun, dkk : 2007 ; 3). Langkah-langkah perhitungan dalam *profile matching*

#### II.3.1. Pemetaan Gap Kompetensi

Gap yang dimaksud disini adalah perbedaan/selisih value masing-masing aspek/atribut dengan value target.  $Gap = Value Atribut - Value Target$  (1)

#### II.3.2. Pembobotan

Setelah diperoleh Gap pada masing-masing calon karyawan, setiap profil calon karyawan diberi bobot nilai sesuai ketentuan pada Tabel Bobot Nilai Gap

**Tabel II.1 Tabel Contoh Bobot Nilai Gap**

Selisih	Bobot Nilai	Keterangan
0	5	Tidak ada selisih (Kompetensi sesuai yang dibutuhkan).
1	4.5	Kompetensi individu kelebihan 1 tingkat/level.
-1	4	Kompetensi individu kekurangan 1 tingkat/level
2	3.5	Kompetensi individu kelebihan 2 tingkat/level.
-2	3	Kompetensi individu kekurangan 2 tingkat/level.
3	2.5	Kompetensi individu kelebihan 3 tingkat/level.
-3	2	Kompetensi individu kekurangan 3 tingkat/level.

4	1.5	Kompetensi individu kelebihan 4 tingkat/level.
-4	1	Kompetensi individu kekurangan 4 tingkat/level.

(Sumber: Kusumaning Hati Pambayun : 2007)

### II.3.3. Perhitungan dan Pengelompokkan

Setelah menentukan bobot nilai gap untuk semua aspek dengan cara sama, setiap aspek dibagi lagi menjadi dua kelompok *Core Factor* (faktor utama) dan *Secondary Factor* (faktor pendukung). Perhitungan *core factor* dapat ditunjukkan dalam persamaan (2)

$$NCF = \frac{\sum NC}{\sum IC} \dots\dots\dots(1)$$

Keterangan :

NCF : Nilai rata-rata *core factor*

NC : Jumlah total nilai *core factor*

IC : Jumlah *item core factor*

Sedangkan untuk perhitungan *secondary factor* dapat ditunjukkan pada

Persamaan (3) dibawah ini :

$$NSF = \frac{\sum NS}{\sum IS} \dots\dots\dots(2)$$

Keterangan :

NSF : Nilai rata-rata *secondary factor*

NF : Jumlah total nilai *secondary factor*

IS : Jumlah *item secondary factor*

### II.3.4. Perhitungan Nilai Total

Dari hasil perhitungan dari tiap aspek di atas kemudian dihitung nilai total berdasarkan presentase dari *core* dan *secondary factor* yang diperkirakan berpengaruh terhadap kinerja tiap-tiap profil. Contoh perhitungan dapat dilihat pada rumus dibawah ini :

$$N = (x) \% NCF + (y) \% NSF \dots\dots\dots(3)$$

Keterangan:

- NCF : Nilai rata-rata *core factor*  
 NSF : Nilai rata-rata *secondary factor*  
 N : Nilai total dari tiap aspek  
 (x)% : Nilai yang diinputkan (60%)  
 (y)% : Nilai persen yang diinputkan (40%)

### II.3.5. Perhitungan Penilaian Rangking

Hasil akhir dari *profile matching* adalah *rangking* dari kandidat yang diajukan untuk mengisi suatu jabatan tertentu. Penentuan rangking mengacu pada hasil perhitungan tertentu. Perhitugan tersebut dapat ditunjukkan pada persamaan (5) dibawah ini :

$$Rangking = \%Nk + \%Ns \dots\dots\dots(4)$$

Keterangan :

- Nk : Nilai Kinerja Karyawan  
 Ns : Nilai Status Karyawan  
 (Nk, Ns)% : Nilai persen yang diinputkan (70%,30%)

Keterangan :

Nilai persen yang diinputkan (%60,%40)(PirandaWidaraAnanta : 2013 ; 577)

#### **II.4. Kebijakan Mengenai Gaji**

Setiap perusahaan harus membuat kebijakan-kebijakan mengenai kegiatan perusahaan yang sedang dikelola. Kebijakan-kebijakan tersebut antara lain mengenai pengangkatan pegawai baru, pemberhentian pegawai, pemberian kenaikan jabatan bagi pegawai yang berprestasi dan menentukan berapa besar tarif gaji dan upah lembur (Ramdani, dkk : 2012 ; 2).

#### **II.5. Metode *Black Box Testing***

Metode ujicoba blackbox memfokuskan pada keperluan fungsional dari software. Karena itu ujicoba blackbox memungkinkan pengembang software untuk himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program. Ujicoba coba bukan merupakan alternatif dari ujicoba whitebox, tetapi merupakan pendekatan yang melengkapi untuk menemukan kesalahan yang lainnya, selain menggunakan metode whitebox (ayuliana : 2009).

#### **II.6. Pengertian Normalisasi Basis Data**

Normalisasi adalah proses pembentukan struktur basis data sehingga sebagian besar ambiguity bisa dihilangkan.

Normalisasi data merupakan suatu proses untuk mendapatkan struktur tabel atau relasi yang efisien dan bebas dari anomali, dan mengacu pada cara data item dikelompokkan ke dalam struktur record.

Normalisasi adalah suatu teknik untuk mengorganisasi data ke dalam tabel – tabel untuk memenuhi kebutuhan pemakai didalam suatu organisasi (Saputro : 2011).

### **II.6.1. Tujuan**

Untuk menghilangkan kerangkapan data, untuk mengurangi kompleksitas, untuk mempermudah pemodifikasian data.

### **II.6.2. Proses Normalisasi**

- a) Data diuraikan dalam bentuk tabel, selanjutnya dianalisis berdasarkan persyaratan tertentu ke beberapa tingkat.
- b) Apabila tabel yang diuji belum memenuhi persyaratan tertentu, maka tabel tersebut perlu dipecah menjadi beberapa tabel yang lebih sederhana sampai memenuhi bentuk yang optimal.

### **II.6.3. Tahapan Normalisasi**

- i. Bentuk tidak Normal : Menghilangkan perulangan group
- ii. Bentuk Normal Pertama (1NF) : Menghilangkan ketergantungan sebagian.
- iii. Bentuk Normal Kedua(2NF) : Menghilangkan ketergantungan transitif.
- iv. Bentuk Normal Ketiga(3NF) : Menghilangkan anomali – anomali hasil dari ketergantungan fungsional.

- v. Bentuk Normal Boyce-Codd (BCNF) : Menghilangkan ketergantungan Multivalued.
- vi. Bentuk Normal Keempat (4NF) : Menghilangkan anomali – anomali yang tersisa.
- vii. Bentuk normal Kelima

### Tabel II.2. Contoh Tabel 1NF

- Contoh tabel yang tidak memenuhi 1stNF

Tabel mahasiswa = (nim, nama\_mhs, alamat\_mhs, hobi)

Nim	Nama_mhs	Alamat_mhs	Hobi
040001	Sintawati	Tlogosari	Menari, Membaca
040002	Andika	Peterongan	Berenang
040003	Setiawan	Ngaliyan	Basket, Sepak Bola
040004	Kartika Sari	Banyumanik	Menyanyi
040005	Asti Ananta	Kaliwiru	Membaca, Musik, Basket

Atribut hobi memiliki nilai banyak dalam satu baris.

- Contoh tabel yang memenuhi 1stNF

Nim	Nama_mhs	Alamat_mhs	Hobi
040001	Sintawati	Tlogosari	Menari
040001	Sintawati	Tlogosari	Membaca
040002	Andika	Peterongan	Berenang
040003	Setiawan	Ngaliyan	Basket
040003	Setiawan	Ngaliyan	Sepak Bola
040004	Kartika Sari	Banyumanik	Menyanyi
040005	Asti Ananta	Kaliwiru	Membaca
040005	Asti Ananta	Kaliwiru	Musik
040005	Asti Ananta	Kaliwiru	Basket

(Sumber : Wahyu Eko Saputro : 2012)

- Contoh tabel yang belum normalisasi

**Tabel II.3 Relasi Mahasiswa dalam bentuk UnNormalized/UNF**

Nim	Nama_mhs	Kode_mk
61521	Andika	SDT231, KDT220,SBD211
61300	Sari	SBD211
61425	Dewi	RPL202,SDT231
61230	Rudi	SD231,RPL202

(Sumber : Wahyu Eko Saputro : 2012)

Kemudian diubah kedalam bentuk pertama, tabel sebagai berikut

#### II.6.4. Bentuk Normal Tahap Pertama (1 Normal Form)

Suatu tabel memenuhi 1NF jika dan hanya jika tabel tersebut tidak memiliki atribut bernilai banyak atau lebih dari satu atribut dengan domain nilai yang sama.

**Tabel II.4. Relasi Mahasiswa dalam bentuk 1stNF**

Nim	Nama_mhs	Kode_mk
61521	Andika	SDT231
61521	Andika	KDT220
61521	Andika	SBD211
61300	Sari	SBD211
61425	Dewi	RPL202
61425	Dewi	SDT231
61230	Rudi	SD231
61230	Rudi	RPL202

(Sumber : Wahyu Eko Saputro : 2012)

#### II.6.5. Bentuk Normal Tahap Kedua (2nd Normal Form)

Suatu tabel yang memenuhi 2ndNF jika dan hanya jika : tabel tersebut memenuhi 1stNF dan setiap atribut yang bukan kunci utama (primary key)

tergantung secara fungsional terhadap semua atribut kunci dan bukan tergantung secara fungsional dan hanya pada sebagian atribut kunci utama.

**Tabel II.5. Relasi Mahasiswa dalam bentuk 2ndNF**

Tabel Mahasiswa\_1

Nim	Nama_mhs
61521	Andika
61521	Andika
61521	Andika
61300	Sari
61425	Dewi
61425	Dewi
61230	Rudi
61230	Rudi

Tabel Mahasiswa\_2

Nama_mhs	Kode_mk
Andika	SDT231
Andika	KDT220
Andika	SBD211
Sari	SBD211
Dewi	RPL202
Dewi	SDT231
Rudi	SDT231
Rudi	RPL202

(Sumber : Wahyu Eko Saputro : 2012)

### II.6.6. Bentuk Normal Tahap Ketiga (3rd Normal Form)

Suatu tabel memenuhi bentuk normal 3rdNF jika dan hanya jika : Tabel tersebut memenuhi 2ndNF dan tidak ada atribut bukan kunci tergantung secara transitive pada kunci utama.

**Tabel II.6. Relasi Mahasiswa\_C3A bentuk 3rdNF**

<b>Nim</b>	<b>Kode_Mk</b>	<b>Nilai</b>
61521	SDT231	B
61521	KDT220	A
61521	SBD211	B
61300	SBD211	A
61425	RPL202	C
61425	SDT231	B
61230	SDT231	B
61230	RPL202	C

(Sumber : Wahyu Eko Saputro : 2012)

#### **II.6.7. Bentuk Normal Tahap Keempat (4th Normal Form)**

Suatu tabel memenuhi bentuk normal 4rdNF jika dan hanya jika : Memenuhi kriteria BCNF dan setiap atribut didalamnya tidak mengalami ketergantungan pada banyak nilai atau dengan kalimat lain, bahwa semua atribut yang mengalami ketergantungan pada banyak nilai adalah bergantung secara fungsional (functionally dependency).

#### **II.6.8. Bentuk Normal Tahap Kelima (5th Normal Form)**

Suatu tabel memenuhi bentuk normal 5thNF jika dan hanya jika : Kerelasian antar data dalam relasi tersebut tidak dapat direkonstruksi dari struktur relasi yang memuat atribut yang lebih sedikit (Saputro : 2011).

#### **II.7. Unified Modelling Language (UML)**

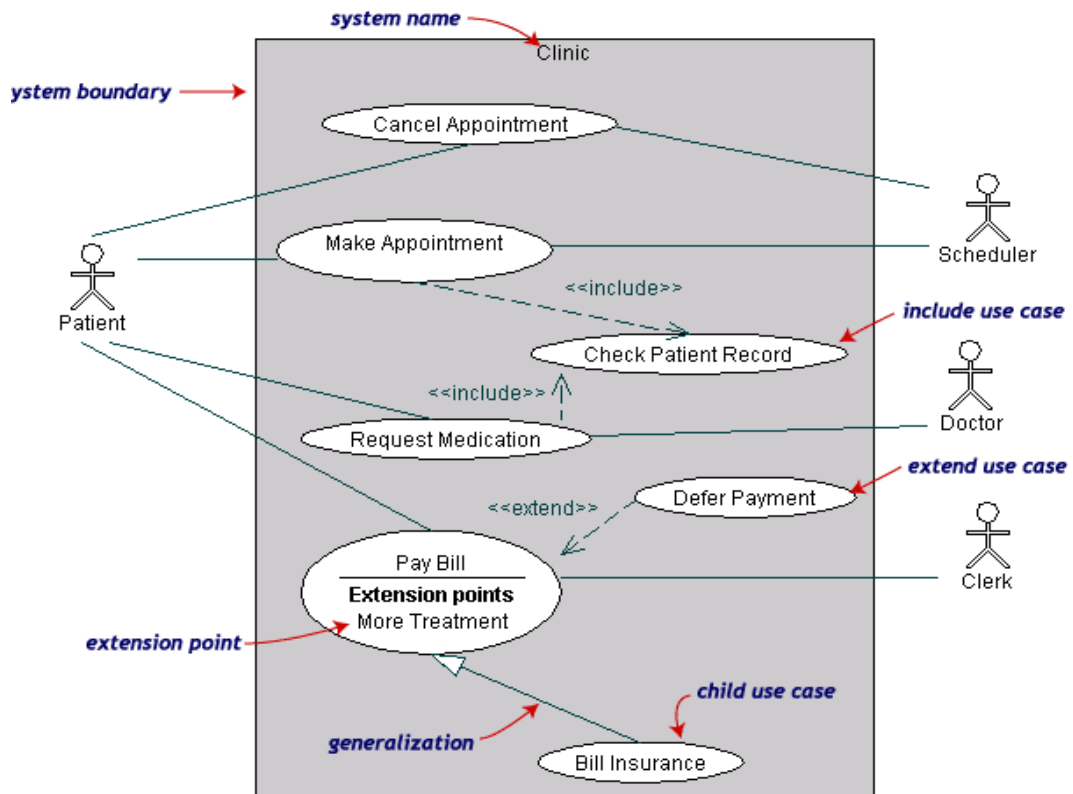
*Unified Modelling Language (UML)* adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML kita dapat

membuat model untuk semua jenis aplikasi piranti lunak. Dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasiobjek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap digunakan untuk modelling aplikasi prosedural dalam VB atau C (Dharwiyanti : 2003).

### **II.7.1. Use Case Diagram**

*Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “ apa “ yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* mempresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu. Misalnya login ke sistem. Meng-*create* sebuah daftar belanja, sebagainya (Dharwiyanti : 2003).

Contoh use case diagram :



**Gambar II.1. Use Case Diagram**

(Sumber : Sri Dharwiyanti : 2003)

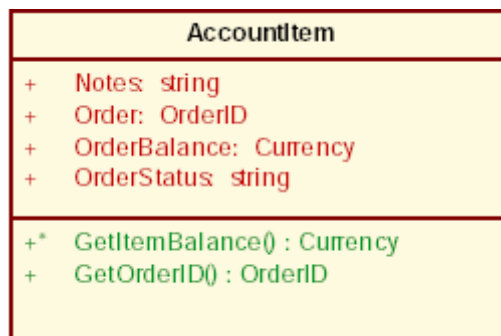
## II.7.2. Class Diagram

*Class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

*Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain (Dharwiyanti : 2003).

*Class* memiliki tiga area pokok :

1. Nama(dan streotype)
2. Atribut
3. Metoda



**Gambar II.2 : Class Diagram**

*(Sumber : Sri Dharwiyanti : 2003)*

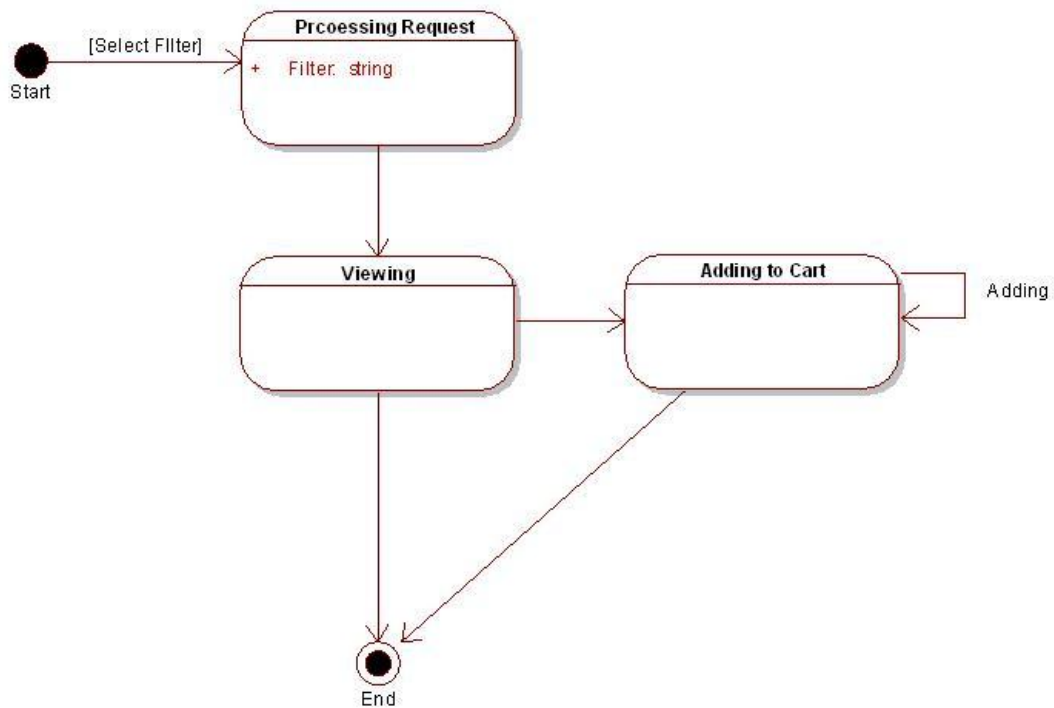
### II.7.3. StateChart Diagram

*Statechart diagram* menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari stimuli yang diterima. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*).

Dalam UML, *state* digambarkan berbentuk segiempat dengan sudut membulat dan memiliki nama sesuai kondisinya saat itu. Transisi antar *state* umumnya memiliki kondisi *guard* yang merupakan syarat terjadinya transisi yang bersangkutan, dituliskan dalam kurung siku. *Action* yang dilakukan sebagai akibat dari *event* tertentu dituliskan dengan diawali garis miring.

Titik awal dan akhir digambarkan berbentuk lingkaran berwarna penuh dan berwarna setengah (Dharwiyanti : 2003).

Contoh *statechart diagram* :



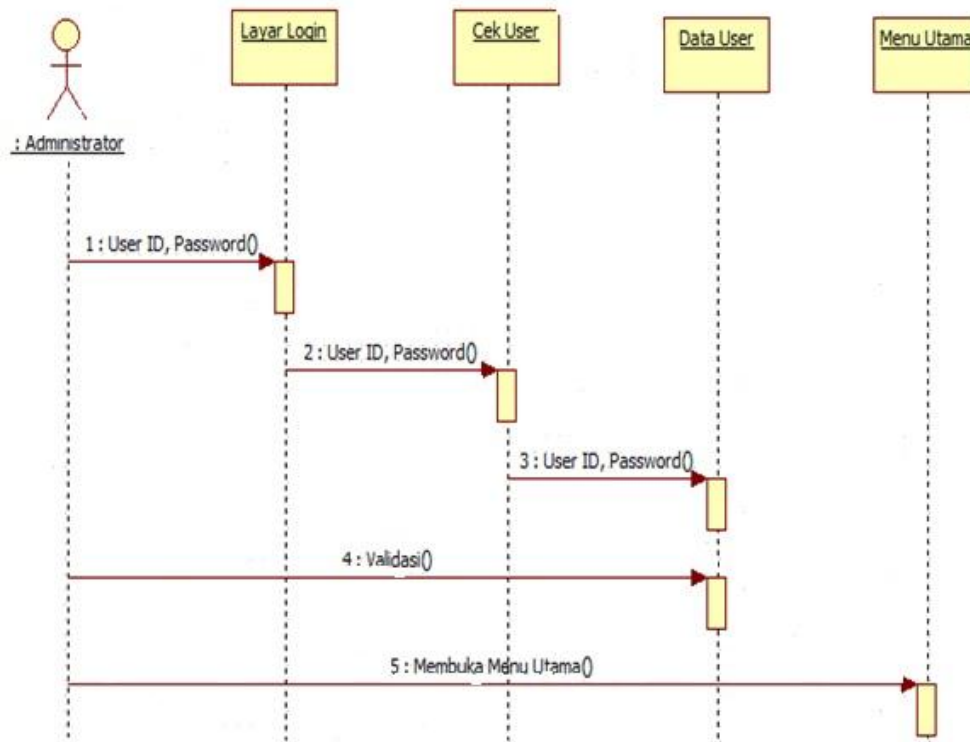
**Gambar II.3. Statechart Diagram**

(Sumber : Sri Dharwiyanti : 2003)

### II.7.3. Sequence Diagram

*Sequence Diagram* menggambarkan interaksi antar objek didalam dan disekitar sistem (termasuk) pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri antar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait) (Sri Dharwiyanti : 2003).

Contoh *Sequence Diagram*



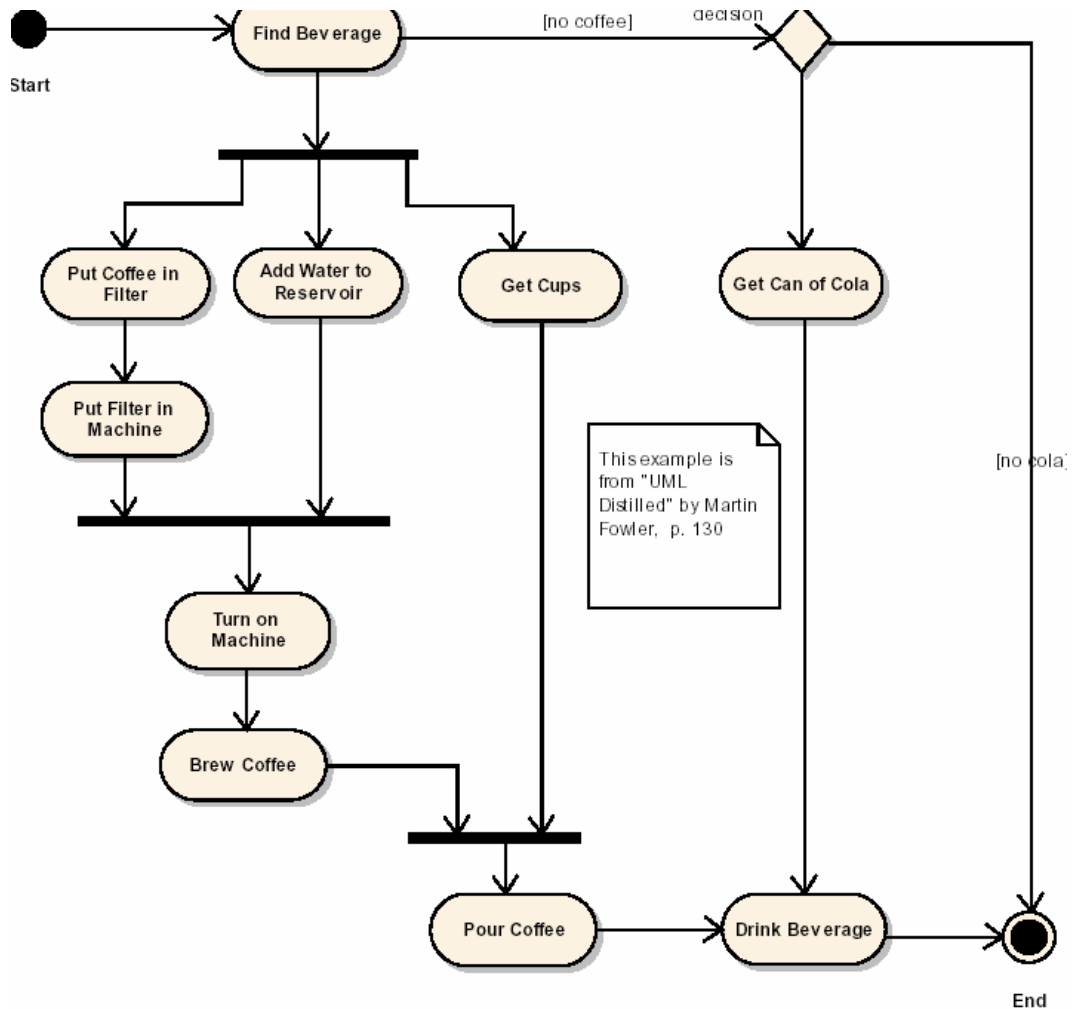
**Gambar II.4. Sequence Digram**

*(Sumber : Sri Dharwiyanti : 2003)*

#### **II.7.4. Activity Diagram**

*Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses model paralel yang mungkin terjadi pada beberapa eksekusi (Dharwiyanti : 2003).

Contoh *activity diagram* :



**Gambar II.5. Activity Diagram**

(Sumber : Sri Dharwiyanti : 2003)

## II.8. Microsoft Visual Studio

Visual basic telah melalui sejumlah versi dan jauh berbeda dengan produk aslinya. Pada setiap versi baru Visual Basic ditingkatkan untuk memasukkan teknologi pengembangan perangkat-lunak. Perangkat-lunak Microsoft

diuntungkan dari penekanan perusahaan pada integrasi produk mereka yang juga telah ditingkatkan dengan setiap rilis terbarunya.

Visual basic 6 dirilis pada tahun 1998. Karena kebutuhan para pengembangan, versi Visual Basic ini membuat metode baru yang telah ditingkatkan untuk berinteraksi dengan SQL Server. Ini termasuk akses data yang telah ditingkatkan, tools dan kontrol baru untuk dipakai bersama database (seperti ADO Data Control), fitur Internet yang telah ditingkatkan (seperti kelas Web), dan sejumlah wizard baru( Siebold : 2001).

## **II.9. Database (Basis Data)**

Basis data (bahasa inggris:*database*), atau sering pula dieja basisdata, adalah kumpulan informasi yang disimpan didalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat lunak yang digunakan untuk mengelola dan memanggil kueri (query) basis data disebut sistem manajemen basis data (*database management system*, DBMS). Sistem basis data dipelajari dalam ilmu informasi (I Beerawa, dkk : 2013 ; 8).

## **II.10. SQL Server**

*SQL Server* telah lama dikenal dan fitur barunya terus ditambah dan ditingkatkan. *SQL Server* adalah bagian dari produk Back Office Microsoft, yang juga menyertakan *BackOffice Server*, *Exchange Server*, *Proxy Server*, *Site Server*, *Small Bussiness Server*, *SNA Server*, dan *System Management Server*.

*SQL Server* diawali dari kerja sama Microsoft dengan Sybase. Pada saat itu, Microsoft bermitra dengan IBM pada sebuah sistem operasi baru yang disebut OS/2. Mereka menginginkan sebuah database yang juga bisa dijalankan pada OS/2 sehingga mereka bermitra dengan Sybase untuk menyertakan produk DataServer mereka yang berbasis UNIX ke OS/2. Untuk mendapatkan pengakuan dan pangsa pasar, Microsoft juga bermitra dengan Ashton-Tate, memiliki mayoritas pasar database dengan produk dBase mereka.

Versi terbaru dan terbesar, *SQL Server 2000*, direncanakan akan dirilis pertengahan 2000, pada saat penulisan buku ini. Versi ini mengenakan skalabilitas perangkat-keras yang ditingkatkan, dan mendukung peranti-peranti berkisar dari handheld Windows CE hingga server-server cluster multiprosesor delapan-jalur. Versi ini juga memasukkan dukungan built-in bagi XML, dukungan failover 4-node, update dan penghapusan cascading, pencarian berbasis teks penuh yang telah ditingkatkan, dan kemampuan untuk menjalankan multi bentuk pada sebuah server tunggal (Diane Siebold : 2001).

### **II.11. *Data Definition Language (DDL)***

Satu paket bahasa DBMS yang berguna untuk melakukan spesifikasi terhadap skema basis data. Hasil komplikasi dari DDL adalah satu set tabel yang disimpan dalam file khusus yang disebut Data Directory/Dictionary. Secara umum perintah – perintah dalam DDL berhubungan dengan operasi – operasi dasar seperti membuat basis data baru, menghapus basis data, membuat table baru, menghapus

table, membuat index, mengubah struktur table. Contoh perintah DDL misalnya, Create Table, Create Index, Alter, dan Drop Database (andika : 2012).

### **II.12. *Data Manipulation Language (DML)***

Satu paket DBMS yang memperbolehkan pemakai untuk mengakses atau memanipulasi data sebagaimana yang telah diorganisasikan sebelumnya dalam model data yang tepat.

Dengan DML dapat dilakukan :

- Mengambil informasi yang tersimpan dalam basis data
- Menyisipkan informasi baru dalam basis data
- Menghapus informasi dari tabel

Terdapat dua tipe DML yaitu prosedural dan non prosedural. Prosedural DML membutuhkan pemakai untuk menspesifikasikan data yang dibutuhkan dan bagaimana cara mendapatkannya, sedangkan non prosedural DML membutuhkan pemakai untuk menspesifikasikan data apa yang dibutuhkan tanpa tahu bagaimana cara mendapatkannya (andika : 2012).