

BAB II

LANDASAN TEORI

II.1. Konsep Sistem Informasi

II.1.1. Pengertian Sistem

Sistem adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu. Struktur sistem merupakan unsur-unsur yang membentuk sistem tersebut. Sedangkan proses sistem menjelaskan cara kerja setiap unsur sistem tersebut dalam mencapai tujuan sistem. Dari definisi ini dapat dirinci lebih lanjut pengertian sistem secara umum, yaitu sebagai berikut :

1. Setiap sistem terdiri dari berbagai unsur.
2. Unsur-unsur tersebut merupakan bagian yang tak terpisahkan dari sistem yang bersangkutan.
3. Unsur-unsur didalam sistem tersebut bekerja sama untuk mencapai tujuan sistem.
4. Suatu sistem merupakan bagian dari sistem lain yang lebih besar

Gordon B.Davis dalam bukunya menyatakan bahwa sistem bisa berupa abstrak atau fisik. Sistem yang abstrak adalah susunan gagasan-gagasan atau konsepsi yang teratur yang saling bergantung. Sedangkan sistem yang bersifat fisik adalah serangkaian unsur yang bekerja sama untuk mencapai suatu tujuan. (Tata Sutabri ; 2012 : 5-6).

Suatu sistem dapat terdiri dari bagian-bagian sistem atau subsistem. Subistem-subsistem tersebut berinteraksi sedemikian rupa sehingga tercapai satu kesatuan yang terpadu dan terintegrasi (*integrated*).

Norman L. Enger mengatakan dalam bukunya bahwa subsistem adalah serangkaian kegiatan yang dapat ditentukan identitasnya yang berhubungan dalam suatu sistem.

Sedangkan Gordon B. Davis dalam bukunya mengatakan bahwa sistem terbagi atas beberapa faktor atau unsur ke dalam beberapa subsistem. Batasan dan penghubung atau *interace* di dalam suatu sistem ditelaah secara cermat untuk menjamin bahwa hubungan antar subsistem didefinisikan secara jelas dan bahwa jumlah semua subsistem merupakan keseluruhan sistem (Tata Sutabri ; 2012 : 10).

II.1.2. Karakteristik Sistem

Model umum sebuah sistem terdiri dari input, proses, dan output. Hal ini merupakan konsep sebuah sistem yang sangat sederhana mengingat sebuah sistem dapat mempunyai beberapa masukan dan keluaran sekaligus. Adapun karakteristik yang dimaksud adalah sebagai berikut : (Tata Sutabri ; 2012 : 13-14).

1. Komponen Sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang bekerja sama membentuk satu kesatuan. Komponen-komponen sistem tersebut dapat berupa satu bentuk sistem. Setiap subsistem memiliki sifat-sifat sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

2. Batasan Sistem (*Boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem lainnya atau sistem dengan lingkungan luarnya.

3. Lingkungan Luar Sistem (*Environment*)

Bentuk apapun yang ada di luar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut dengan lingkungan luar sistem.

4. Penghubung Sistem (*Interface*)

Media yang menghubungkan sistem dengan subsistem yang lain dengan penghubung sistem atau *interface*.

5. Masukan Sistem (*Input*)

Energi yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*). Sebagai contoh didalam suatu unit sistem komputer, “*program*” adalah *maintenance input* yang digunakan untuk mengoperasikan komputer. Sementara data adalah signal input yang akan diolah menjadi informasi.

6. Keluaran Sistem (*Output*)

Hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain.

7. Pengolah Sistem (*Process*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran.

8. Sasaran Sistem (*Objective*)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat deterministik.

II.1.3. Defenisi Sistem Informasi

Suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial, dan merupakan kegiatan strategi dari suatu organisasi, serta menyediakan laporan-laporan yang diperlukan oleh pihak luar.

Berdasarkan dukungan kepada pemakainya, sistem informasi dibagi menjadi :

1. Sistem Pemrosesan Transaksi (*Transaction Processing System* atau TPS)
2. Sistem Informasi Manajemen (*Management Information System* atau MIS)
3. Sistem Otomasi Perkantoran (*Office Automation System / OAS*)
4. Sistem Pendukung Keputusan (*Decision Support System* atau DSS)
5. Sistem Informasi Eksekutif (*Executive Information System* atau EIS)
6. Sistem Pendukung Kelompok (*Group Support System* atau GSS)
7. Sistem Pendukung Cerdas (*Intelligent Support System* atau ISS).

Mengingat bahwa EIS, DSS, dan MIS digunakan untuk mendukung keputusan manajemen, maka ketiga sistem tersebut sering disebut Sistem Pendukung Manajemen (*Management Support System* atau MSS). (Kusrini ; 2007 : 11-12).

II.2. Konsep Sistem Pendukung Keputusan

II.2.1. Sistem Pendukung Keputusan / *Decision Support System* (DSS)

Decision Support System merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan, dan pemanipulasian data. Sistem ini digunakan untuk membantu pengambilan keputusan dalam situasi yang semiterstruktur dan situasi yang tidak terstruktur, dimana tak seorangpun tahu secara pasti bagaimana keputusan seharusnya dibuat (Alter, 2002).

Decision Support System biasanya dibangun untuk mendukung solusi atas suatu masalah atau untuk mengevaluasi suatu peluang. *Decision Support System* yang seperti itu disebut aplikasi *Decision Support System*. Aplikasi *Decision Support System* digunakan dalam pengambilan keputusan. Aplikasi *Decision Support System* menggunakan CBIS (*Computer Based Information Systems*) yang fleksibel, interaktif, dan dapat diadaptasi, yang dikembangkan untuk mendukung solusi atas masalah manajemen spesifik yang tidak tersruktur. (Kusrini ; 2007 : 15-16).

II.2.2. Tujuan *Decision Support System* (DSS)

Tujuan dari *Decision Support System* (DSS) adalah (Turban, 2005) :
(Kusrini ; 2007 : 16-17)

1. Membantu manajer dalam pengambilan keputusan atas masalah semi terstruktur.
2. Memberikan dukungan atas pertimbangan manajer dan bukannya dimaksudkan untuk menggantikan fungsi manajer.
3. Meningkatkan efektivitas keputusan yang diambil manajer lebih dari pada perbaikan efisiensinya.
4. Kecepatan komputasi. Komputer memungkinkan para pengambil keputusan untuk melakukan banyak komputasi secara cepat dengan biaya yang rendah.
5. Peningkatan produktifitas. Membangun satu kelompok pengambilan keputusan, terutama para pakar, bisa sangat mahal. Pendukung terkomputerisasi bisa mengurangi ukuran kelompok dan memungkinkan para anggotanya untuk berada diberbagai lokasi yang berbeda-beda (menghemat biaya perjalanan).
6. Dukungan kualitas. Komputer bisa meningkatkan kualitas keputusan yang dibuat. Sebagai contoh, semakin banyak data yang diakses, makin banyak juga alternatif yang bisa dievaluasi.
7. Berdaya saing. Manajemen dan pemberdayaan sumber daya perusahaan. Tekanan persaingan menyebabkan tugas pengambilan keputusan menjadi sulit. Persaingan didasarkan tidak hanya pada harga, tetapi juga pada kualitas, kecepatan, kustomasi produk, dan dukungan pelanggan. Teknologi

pengambilan keputusan bisa menciptakan pemberdayaan yang signifikan dengan cara memperbolehkan seseorang untuk membuat keputusan yang baik secara cepat, bahkan jika mereka memiliki pengetahuan yang kurang.

8. Mengatasi keterbatasan kognitif dalam pemrosesan dan penyimpanan. Menurut Simon (1977), otak manusia memiliki kemampuan yang terbatas untuk memproses dan menyimpan informasi. Orang-orang kadang sulit mengingat dan menggunakan sebuah informasi dengan cara bebas dari kesalahan.

Ditinjau dari tingkat teknologinya, *Decision Support System* (DSS) dibagi menjadi 3, yaitu : (Kusrini ; 2007: 18).

1. Sistem Pendukung Keputusan (SPK) Spesifik

Sistem Pendukung Keputusan (SPK) spesifik bertujuan membantu memecahkan suatu masalah dengan karakteristik tertentu. Misalnya, Sistem Pendukung Keputusan penentuan harga satuan barang.

2. Pembangkit Sistem Pendukung Keputusan (SPK)

Suatu *software* yang khusus digunakan untuk membangun dan mengembangkan Sistem Pendukung Keputusan (SPK).

3. Perlengkapan Sistem Pendukung Keputusan (SPK)

Berupa *software* dan *hardware* yang digunakan atau mendukung pembangunan SPK spesifik maupun pembangkit Sistem Pendukung Keputusan (SPK).

Keputusan yang diambil untuk menyelesaikan suatu masalah dilihat dari keterstrukturannya yang bisa dibagi menjadi : (Kusrini ; 2007 : 19-20).

1. Keputusan Terstruktur (*Structured Decision*)

Keputusan terstruktur adalah keputusan yang dilakukan secara berulang-ulang dan bersifat rutin. Prosedur pengambilan keputusan sangatlah jelas.

2. Keputusan Semiterstruktur (*Semistructured Decision*)

Keputusan semiterstruktur adalah keputusan yang memiliki dua sifat. Sebagian keputusan bisa ditangani oleh komputer dan yang lain tetap harus dilakukan oleh pengambilan keputusan. Contoh keputusan jenis ini adalah pengevaluasian kredit, penjadwalan produksi, dan pengendalian sediaan.

3. Keputusan Tak Terstruktur (*Unstructured Decision*)

Keputusan tak terstruktur adalah keputusan yang penanganannya rumit karena tidak terjadi berulang-ulang atau tidak selalu terjadi. Keputusan terstruktur menuntut pengalaman dan berbagai sumber yang bersifat eksternal. Keputusan tersebut umumnya terjadi pada manajemen tingkat atas. Contohnya adalah keputusan untuk pengembangan teknologi baru, keputusan untuk bergabung dengan perusahaan lain, dan perekrutan eksekutif.

II.2.3. Karakteristik *Decision Support System* (DSS)

Decision Support System memiliki karakteristik dasar sebagai berikut (PowerD. J., 2002) : (Jurnal SEMNASTEKNOMEDIA 2015, STMIK AMIKOM Yogyakarta, 2015).

1. *Decision Support System* dirancang secara khusus untuk memfasilitasi proses pembuatan keputusan.

2. *Decision Support System* seharusnya lebih bersifat membantu, bukan menghasilkan keputusan.
3. *Decision Support System* harus mampu untuk menangani perubahan kebutuhan pembuat keputusan secara cepat. *Decision Support System* seperti tipe sistem informasi lainnya, pada dasarnya terdiri atas tiga bagian utama yaitu masukan, proses, serta keluaran. Yang membedakan *Decision Support System* dengan tipe sistem informasi lainnya adalah jenis masukan dan keluaran serta proses yang dilakukannya.

II.2.4. Komponen-komponen *Decision Support System* (DSS)

Komponen *Decision Support System* dapat berupa : (Sumber : Jurnal SEMNASTEKNOMEDIA 2015 STMIK AMIKOM Yogyakarta ; Adil Setiawan dan Surya Darma ; Februari 2015 : 2).

a. *Data Management,*

Termasuk *database*, yang mengandung data yang relevan untuk pelbagai situasi dan diatur oleh *software* yang disebut *Database Management Systems* (DBMS).

b. *Model Management,*

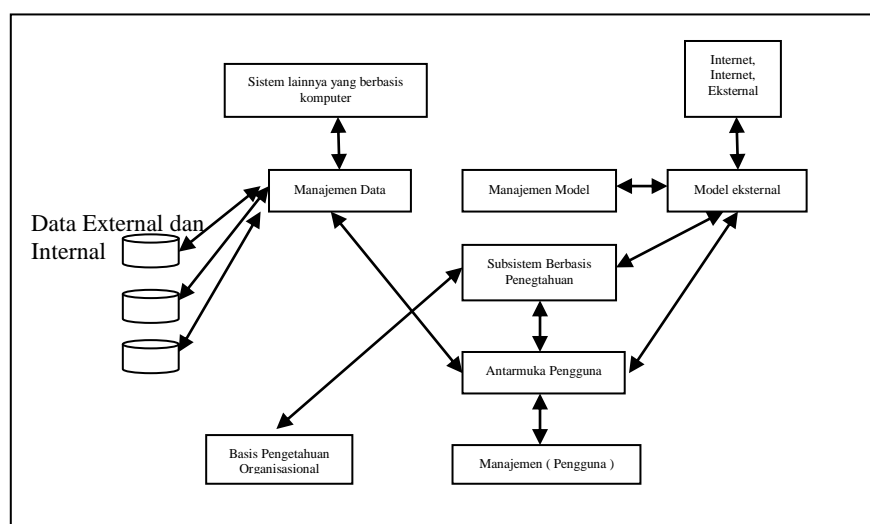
Melibatkan model *finansial, statistikal, management science*, atau pelbagai model kuantitatif lainnya, sehingga dapat memberikan kesistem suatu kemampuan analitis, dan manajemen *software* yang diperlukan.

c. *Communication (Dialog Subsystem),*

User dapat berkomunikasi dan memberikan perintah pada *Decision Support System* melalui subsistem ini. Ini berarti menyediakan antarmuka.

d. *Knowledge Management*,

Subsistem optional ini dapat mendukung subsistem lain atau bertindak sebagai komponen yang berdiri sendiri. Gambar II.1 menunjukkan model konseptual *Decision Support System*.



Gambar II.1. Model Konseptual *Decision Support System* (DSS)

(Sumber : Jurnal SEMNASTEKNOMEDIA 2015 STMIK AMIKOM Yogyakarta ;

Adil Setiawan dan Surya Darma ; Februari 2015 : 2)

II.2.5. Langkah-langkah Pemodelan dalam *Decision Support System* (DSS)

Saat melakukan pemodelan dalam pembangunan *Decision Support System* dilakukan langkah-langkah sebagai berikut : (Kusrini ; 2007 : 30-31).

1. Studi Kelayakan (*Intelligence*)

Pada tahap ini, sasaran ditentukan dan dilakukan pencarian prosedur, pengumpulan data, identifikasi masalah, identifikasi kepemilikan masalah, klasifikasi masalah, hingga akhirnya terbentuk sebuah pernyataan masalah.

Kepemilikan masalah berkaitan dengan bagian apa yang akan dibangun oleh *Decision Support System* dan apa tugas dari bagian tersebut sehingga model tersebut bisa relevan dengan kebutuhan si pemilik masalah.

2. Perancangan (*Design*)

Pada tahapan ini akan diformulasikan model yang akan digunakan dan kriteria-kriteria yang ditentukan. Setelah itu, dicari alternatif model yang bisa menyelesaikan permasalahan tersebut. Langkah selanjutnya adalah memprediksi keluaran yang mungkin. Kemudian, ditentukan variabel-variabel model.

3. Pemilihan (*Choice*)

Setelah pada tahap *design* ditentukan berbagai alternatif model beserta variabel-variabelnya, pada tahapan ini akan dilakukan pemilihan modelnya, termasuk solusi dari model tersebut. Selanjutnya dilakukan analisis sensitivitas, yakni dengan mengganti beberapa variabel.

4. Membuat *Decision Support System*

Setelah menentukan modelnya, berikutnya adalah mengimplementasikan dalam aplikasi *Decision Support System*.

II.3. Metode *Preference Ranking Organization For Enrichment Evaluation (PROMETHEE)*.

Metode *Promethee* termasuk kedalam kelompok pemecahan masalah *Multi Criteria Decision Making (MCDM)* atau pengambilan keputusan kriteria majemuk yang merupakan disiplin ilmu yang sangat penting dalam pengambilan keputusan atau suatu masalah yang memiliki lebih dari satu kriteria (*multikriteria*).

Menurut Brans dan Marcshal (1999;15), *Promethee* yang merupakan singkatan dari *Preference Ranking Organization For Enrichment Evaluation* adalah metode *outranking* yang menawarkan cara yang fleksibel dan sederhana kepada user (pembuat keputusan) untuk menganalisis masalah-masalah *multikriteria*. Prinsip yang digunakan adalah penetapan prioritas alternatif yang telah ditetapkan berdasarkan pertimbangan kaidah dasar :

$$\text{Max } \{f_1(x), f_2(x), f_3(x), \dots, f_i(x), \dots, f_k(x)\} I_x$$

Dimana k adalah sejumlah kumpulan alternatif dan f_i ($i=1,2,\dots, k$) merupakan nilai/ukuran relatif kriteria untuk masing-masing alternatif.

Termasuk dalam keluarga dari metode *outranking* yang dikembangkan oleh B.Roy (Brans et al, 1999) dan meliputi dua fase :

1. Membangun hubungan *outranking* dari K , dimana K adalah sejumlah kumpulan alternatif
2. Eksploitasi dari hubungan ini memberikan jawaban optimasi kriteria dalam paradigma permasalahan multikriteria.

Dalam fase pertama, nilai hubungan *outranking* berdasarkan pertimbangan dominasi masing-masing kriteria. Indeks preferensi ditentukan dan nilai *outranking* secara grafis disajikan berdasarkan preferensi dari pembuat keputusan. (Makalah Ilmiah Informasi dan Teknologi Ilmiah (INTI), Vol. 1, No. 1 ; Dewi Safitri Hutabarat ; 2013 : 14).

Tujuan utama dari pendekatan *PROMETHEE* ini adalah untuk mempermudah proses pengambilan keputusan dengan cara mengelompokkan tipe keputusan menjadi 6 fungsi kriteria yang cukup dapat mewakili semua jenis keputusan untuk menyelesaikan kasus-kasus sehari-hari dan melakukan kuantifikasi derajat preferensi dengan menggunakan maksimum 2 parameter yang memiliki karakteristik ekonomi yang signifikan. (Jurnal Itenas Rekayasa, Vol. 13, No. 4 ; Ambar Harsono, et al ; Oktober-Desember 2009 : 187)

II.3.1 Nilai Hubungan Outranking dalam *Promethee*

1. Dominasi Kriteria

Nilai f merupakan nilai nyata dari suatu kriteria : (Jurnal Pelita Informatika Budi Darma, Vol. 4, No. 2 ; Reizha Arsita ; Agustus 2013 : 107)

$$F:K \rightarrow R$$

Untuk setiap alternatif $a \in K$, $f(a)$ merupakan evaluasi dari alternatif tersebut untuk suatu kriteria. Pada saat dua alternatif dibandingkan, $a, b \in K$, harus dapat ditentukan perbandingan preferensinya. Penyampaian intensitas (P) dari preferensi alternatif a terhadap b sedemikian rupa sehingga :

- a. $P(a,b) = 0$, berarti tidak ada (*indefferent*) antara a dan b, atau tidak ada preferensi dari a lebih baik dari b.
- b. $P(a,b) \sim 0$, berarti lemah preferensi dari a lebih baik dari b.
- c. $P(a,b) \sim 1$, berarti kuat preferensi dari a lebih baik dari b.
- d. $P(a,b) = 1$, berarti mutlak preferensi dari a lebih baik dari b.

Dalam metode ini, fungsi preferensi sering kali menghasilkan nilai fungsi yang berbeda antara dua evaluasi, sehingga :

$$P(a, b) = P(f(a) - f(b))$$

2. Rekomendasi Fungsi Preferensi Untuk Keperluan Aplikasi

Dalam *Promethee* disajikan enam bentuk fungsi preferensi kriteria. Enam preferensi tersebut adalah sebagai berikut : (Jurnal Pelita Informatika Budi Darma, Vol. 4, No. 2 ; Reizha Arsita ; Agustus 2013 : 107)

a. Kriteria Biasa (*Usual Criterion*)

$$H(d) = \begin{cases} 0 & \text{jika } d \leq 0 \\ 1 & \text{jika } d > 0 \end{cases}$$

Dimana :

$H(d)$ = fungsi selisih kriteria antar alternatif

d = selisih nilai kriteria $\{d = f(a) - f(b)\}$

Pada kasus ini, tidak ada beda (sama penting) antara a dan b jika dan hanya jika kriteria $f(a) = f(b)$; apabila kriteria pada masing-masing alternatif memiliki nilai berbeda, pembuat keputusan membuat preferensi mutlak untuk alternatif memiliki nilai yang baik.

b. Kriteria Quasi (*Quasi Criterion*)

$$H(d) = \begin{cases} 0 & \text{jika } d \leq q \\ 1 & \text{jika } d > q \end{cases}$$

Dimana :

H (d) = fungsi selisih kriteria antar alternatif

d = selisih nilai kriteria $\{d = f(a) - f(b)\}$

q = harus merupakan nilai tetap

Dua alternatif memiliki preferensi yang sama penting selama selisih atau nilai H(d) dari masing-masing alternatif untuk kriteria tertentu tidak melebihi nilai q, dan apabila selisih hasil evaluasi untuk masing-masing alternatif melebihi nilai q maka terjadi bentuk preferensi mutlak. Jika pembuat keputusan menggunakan kriteria kuasi, maka harus menentukan nilai q, dimana nilai ini dapat menjelaskan pengaruh yang signifikan dari suatu kriteria.

c. Kriteria Dengan Preferensi Linier

$$H(d) = \begin{cases} 0 & \text{jika } d \leq q \\ \frac{d}{p} & \text{jika } 0 \leq d \leq p \\ 1 & \text{jika } d > p \end{cases}$$

Dimana :

H (d) = fungsi selisih kriteria antar alternatif

d = selisih nilai kriteria $\{d = f(a) - f(b)\}$

q = nilai kecenderungan atas

Kriteria preferensi linier dapat menjelaskan bahwa selama nilai selisih memiliki nilai yang lebih rendah dari p, preferensi dari pembuat keputusan

meningkat secara linier dengan nilai d . Jika nilai d lebih besar dibandingkan dengan nilai p , maka terjadi preferensi mutlak. Pada saat pembuat keputusan mengidentifikasi beberapa kriteria untuk tipe ini, harus ditentukan nilai dari kecenderungan atas (nilai p).

d. Kriteria Level (*Level Criterion*)

$$H(d) = \begin{cases} 0 & \text{jika } d \leq q \\ 0,5 & \text{jika } q < d \leq p \\ 1 & \text{jika } d > p \end{cases}$$

Dimana :

$H(d)$ = fungsi selisih kriteria antar alternatif

d = selisih nilai kriteria $\{d = f(a) - f(b)\}$

p = nilai kecenderungan atas

q = harus merupakan nilai yang tetap

Dalam kasus ini, kecenderungan tidak berbeda q dan kecenderungan preferensi adalah ditentukan secara simultan. Jika d berada diantara nilai q dan p , hal ini berarti situasi preferensi yang lemah ($H(d) = 0,5$).

e. Kriteria dengan Preferensi Linier dan Area yang Tidak Berbeda

$$H(d) = \begin{cases} 0 & \text{jika } d \leq q \\ \frac{d - q}{p - q} & \text{jika } q \leq d \leq p \\ 1 & \text{jika } d > p \end{cases}$$

Dimana :

$H(d)$ = fungsi selisih kriteria antar alternatif

d = selisih nilai kriteria $\{d = f(a) - f(b)\}$

p = nilai kecenderungan atas

q = harus merupakan nilai yang tetap

Pada kasus ini, pengambil keputusan mempertimbangkan peningkatan preferensi secara linier dari tidak berbeda hingga preferensi mutlak dalam area antara dua kecenderungan q dan p .

f. Kriteria Gaussian (*Gaussian Criterion*)

$$H(d) = \begin{cases} 0 & \text{jika } d \leq 0 \\ 1 - e^{-\frac{d^2}{2a^2}} & \text{jika } d > 0 \end{cases}$$

Dimana :

$H(d)$ = fungsi selisih kriteria antar alternatif

d = selisih nilai kriteria $\{d = f(a) - f(b)\}$

II.3.2. Indeks Preferensi Multikriteria

Indeks preferensi multikriteria ditentukan berdasarkan rata-rata bobot dari fungsi preferensi P_i .

$$\varphi(a, b) = \sum_{i=1}^n \pi_i P_i(a, b): \forall a, b \in A$$

$\varphi(a, b)$ merupakan intensitas preferensi pembuat keputusan yang menyatakan bahwa alternatif a lebih baik dari alternatif b dengan pertimbangan secara simultan dari keseluruhan kriteria. Hal ini dapat disajikan dengan nilai antara nilai 0 dan 1, dengan ketentuan sebagai berikut :

- a. $\varphi(a, b) = 0$ menunjukkan preferensi yang lemah untuk alternatif $a >$ alternatif b berdasarkan semua kriteria.
- b. $\varphi(a, b) = 1$ menunjukkan preferensi yang kuat untuk alternatif $a >$ alternatif b berdasarkan semua kriteria.

Indeks preferensi ditentukan berdasarkan nilai hubungan outranking pada sejumlah kriteria dari masing-masing alternatif. Hubungan ini dapat disajikan sebagai grafik nilai outranking, node-nodenya merupakan alternatif berdasarkan penilaian kriteria tertentu. (Jurnal Pelita Informatika Budi Darma, Vol. 4, No. 2 ; Reizha Arsita ; Agustus 2013 : 108)

II.3.3. *Promethee Ranking*

Perhitungan arah preferensi dipertimbangkan berdasarkan nilai indeks : (Jurnal Pelita Informatika Budi Darma, Vol. 4, No. 2 ; Reizha Arsita ; Agustus 2013 : 108)

1. *Leaving Flow*

$$\varphi^+(a) = \frac{1}{n-1} \sum_{x \in A} \varphi(a, x)$$

2. *Entering Flow*

$$\varphi^-(a) = \frac{1}{n-1} \sum_{x \in A} \varphi(a, x)$$

3. *Net Flow*

$$\varphi(a) = \varphi^+(a) - \varphi^-(a)$$

Keterangan :

$\varphi(a, x)$ = menunjukkan preferensi bahwa alternatif lebih baik dari alternatif x .

$\varphi(a, x)$ = menunjukkan preferensi bahwa alternatif x lebih baik dari alternatif.

- $\varphi^+(a)$ = *Leaving Flow*, digunakan untuk menentukan urutan prioritas pada proses *Promethee I* yang menggunakan urutan persial
- $\varphi^-(a)$ = *Entering Flow*, digunakan untuk menentukan urutan prioritas pada proses *Promethee I* yang menggunakan urutan parsial.
- $\varphi(a)$ = *Net Flow*, digunakan untuk menghasilkan keputusan akhir penentuan urutan dalam menyelesaikan masalah sehingga menghasilkan urutan lengkap.

II.4. Kartu Perlindungan Sosial (KPS)

Kartu Perlindungan Sosial (KPS) adalah kartu yang diterbitkan oleh pemerintah dalam rangka pelaksanaan Program Percepatan dan Perluasan Perlindungan Sosial (P4S) dan Bantuan Langsung Sementara Masyarakat (BLSM). Kartu Perlindungan Sosial (KPS) memuat informasi Nama Kepala Rumah Tangga, Nama Pasangan Kepala Rumah Tangga, Nama Anggota Rumah Tangga Lain, Alamat Rumah Tangga, Nomor Kartu Keluarga, dilengkapi dengan kode batang (*barcode*) beserta nomor identitas Kartu Perlindungan Sosial (KPS) yang unik. Bagian depan bertuliskan Kartu Perlindungan Sosial dengan logo Garuda, dan masa berlaku kartu. (Tim Sosialisasi Penyesuaian Subsidi Bahan Bakar Minyak ; 2013 : 44)



Gambar II.2. Kartu Perlindungan Sosial

(Sumber : Tim Sosialisasi Penyesuaian Subsidi

Bahan Bakar Minyak ; 2013 : 44)

II.4.1. Manfaat bagi Rumah Tangga Pemegang Kartu

Adapun manfaat kartu perlindungan sosial bagi rumah tangga penerima, antara lain : (Tim Sosialisasi Penyesuaian Subsidi Bahan Bakar Minyak ; 2013 : 44-45)

1. Kartu Perlindungan Sosial digunakan sebagai penanda Rumah Tangga miskin dan rentan.
2. Kartu Perlindungan Sosial digunakan untuk mendapatkan manfaat Program Percepatan dan Perluasan Perlindungan Sosial (P4S) dan Bantuan Langsung Sementara Masyarakat (BLSM)

II.4.2. Basis Data Terpadu Sebagai Sumber Data Kartu Perlindungan Sosial (KPS)

Data Rumah Tangga Sasaran (RTS) bersumber dari Basis Data Terpadu (BDT) yang dikelola oleh Tim Nasional Percepatan Penanggulangan Kemiskinan

(TNP2K). Pendataan Data Rumah Tangga Sasaran (RTS) telah dilakukan sebanyak tiga kali oleh Badan Pusat Statistik (BPS), yaitu : Pendataan Sosial Ekonomi (PSE) pada tahun 2005, Pendataan Program Perlindungan Sosial (PPLS) pada tahun 2008, dan yang terakhir Pendataan Program Perlindungan Sosial (PPLS) pada tahun 2011.

Dalam rangka meningkatkan keakuratan data Rumah Tangga Sasaran (RTS), metodologi pendataan Rumah Tangga Sasaran (RTS) disempurnakan, yang mana penyempurnaan metodologi tersebut dikoordinasikan oleh Tim Nasional Percepatan Penanggulangan Kemiskinan. Pendataan dilapangan untuk mencacah seluruh karakteristik Rumah Tangga sasaran dilakukan oleh Badan Pusat Statistik (BPS). Hasil pencacahan tersebut disampaikan kepada Tim Nasional Percepatan Penanggulangan Kemiskinan untuk diolah sehingga menghasilkan 40% data

Rumah Tangga dengan status sosial ekonomi terendah. Data tersebut kemudian dikelola sebagai Basis Data Terpadu (BDT). Berdasarkan Basis Data Terpadu (BDT), diputuskan bahwa Kartu Perlindungan Sosial diberikan kepada 25% Rumah Tangga dengan status sosial ekonomi terendah. Sebagaimana diketahui, bahwa jumlah penduduk yang hidup di bawah garis kemiskinan pada bulan September 2012 adalah 11,66%. Maka, pemberian Kartu Perlindungan Sosial tidak hanya mencakup mereka yang miskin namun juga mencakup mereka yang rentan. (Tim Sosialisasi Penyesuaian Subsidi Bahan Bakar Minyak ; 2013 :

II.5. Pengertian Basis Data

Sebuah basis data adalah sebuah kumpulan data yang saling berhubungan secara logis dan merupakan sebuah penjelasan dari data tersebut, yang didesain untuk menemukan data yang dibutuhkan oleh sebuah organisasi. Basis data juga merupakan sekumpulan data elemen data yang saling terintegrasi (Indrajani ; 2014 : 70).

Berdasarkan tingkat kompleksitas nilai data, tingkatan data dapat disusun dalam sebuah hierarki, mulai dari yang paling sederhana hingga paling sederhana hingga paling kompleks (Edy Sutanta ; 2011 : 35-36).

1. Sistem basis data, merupakan sekumpulan subsistem yang terdiri atas basis data dengan para pemakai yang menggunakan basis data secara bersama-sama, personal-personal yang merancang dan mengelola basis data, teknik-teknik untuk merancang dan mengelola basis data, serta sistem komputer untuk mendukungnya.
2. Basis data, merupakan sekumpulan dari bermacam-macam tipe *record* yang memiliki hubungan antar-*record* dan rincian data terhadap obyek tertentu.
3. File, merupakan sekumpulan *record* sejenis secara relasi yang tersimpan dalam media penyimpanan sekunder.
4. *Record*, merupakan *field*/atribut/data item yang saling berhubungan terhadap obyek tertentu.
5. *Data item/field*/atribut, merupakan unit terkecil yang disebut data, sekumpulan *byte* yang mempunyai makna.

6. *Data aggregate*, merupakan sekumpulan data item/*field*/atribut dengan ciri tertentu dan diberi nama.
7. *Byte*, adalah bagian terkecil yang dialamatkan dalam memori. *Byte* merupakan sekumpulan *bit* yang secara konvensional terdiri atas kombinasi 8 *bit* biner yang menyatakan sebuah karakter dalam memori (1 *byte* = 1 karakter).
8. *Bit*, adalah sistem biner yang terdiri atas dua macam nilai, yaitu 0 dan 1. Sistem biner merupakan dasar yang dapat digunakan untuk komunikasi antara manusia dan mesin (komputer).

II.6. Normalisasi

Normalisasi diartikan sebagai suatu teknik yang menstrukturkan/mendekomposisi data dalam cara-cara tertentu untuk mencegah timbulnya permasalahan pengolahan data dalam basis data. Permasalahan yang dimaksud adalah berkaitan dengan penyimpangan-penyimpangan (*anomalies*) yang terjadi akibat adanya kerangkapan data dalam relasi dan in-efisiensi pengolahan (Martin, 1975). (Edy Sutanta ; 2011 : 174)

Proses normalisasi menghasilkan relasi yang optimal, yaitu (Martin, 1975)
: (Edy Sutanta ; 2011 : 175)

1. Memiliki struktur *record* yang konsisten secara logik;
2. Memiliki struktur *record* yang mudah untuk dimengerti;
3. Memiliki struktur *record* yang sederhana dalam pemeliharaan;

4. Memiliki struktur *record* yang mudah ditampilkan kembali untuk memenuhi kebutuhan pengguna;
5. Minimalisasi kerangkapan data guna meningkatkan kinerja sistem.

Secara berturut-turut masing-masing level normal tersebut dibahas berikut ini, dimulai dari bentuk tidak normal. (Edy Sutanta ; 2011 : 176-179)

1. Relasi bentuk tidak normal (*Un Normalized Form* / UNF)

Relasi-relasi yang dirancang tanpa mengindahkan batasan dalam defisi basis data dan karakteristik *Relational Database Management System* (RDBM) menghasilkan relasi *Un Normalized Form* (UNF). Bentuk ini harus di hindari dalam perancangan relasi dalam basis data. Relasi *Un Normalized Form* (UNF) mempunyai kriteria sebagai berikut :

- a. Jika relasi mempunyai bentuk *non flat file* (dapat terjadi akibat data disimpan sesuai dengan kedatangannya, tidak memiliki struktur tertentu, terjadi duplikasi atau tidak lengkap)
 - b. Jika relasi membuat *set atribut* berulang (*non single values*)
 - c. Jika relasi membuat *atribut non atomic value*
2. Relasi bentuk normal pertama (*First Norm Form* / 1NF)

Relasi disebut juga *First Norm Form* (1NF) jika memenuhi kriteria sebagai berikut :

- a. Jika seluruh atribut dalam relasi bernilai *atomic* (*atomic value*)
- b. Jika seluruh atribut dalam relasi bernilai tunggal (*single value*)
- c. Jika relasi tidak memuat set atribut berulang
- d. Jika semua record mempunyai sejumlah atribut yang sama.

Permasalahan dalam *First Normal Form* (1NF) adalah sebagai berikut.

- a. Tidak dapat menyisipkan informasi parsial
 - b. Terhapusnya informasi ketika menghapus sebuah *record*
3. Bentuk normal kedua (*Second Normal Form* / 2NF)

Relasi disebut sebagai *Second Normal Form* (2NF) jika memenuhi kriteria sebagai berikut :

- a. Jika memenuhi kriteria *First Normal Form* (1NF)
- b. Jika semua atribut nonkunci *Functional Dependence* (FD) pada *Primary Key* (PK)

Permasalahan dalam *Second Normal Form* / 2NF adalah sebagai berikut

- a. Kerangkapan data (*data redundancy*)
- b. Pembaharuan yang tidak benar dapat menimbulkan inkonsistensi data (*data inconsistency*)
- c. Proses pembaharuan data tidak efisien

Kriteria tersebut mengidentifikasi bahwa antara atribut dalam *Second Normal Form* masih mungkin mengalami *Third Normal Form*. Selain itu, relasi *Second Normal Form* (2NF) menuntut telah didefinisikan atribut *Primary Key* (PK) dalam relasi. Mengubah relasi *First Normal Form* (1NF) menjadi bentuk *Second Normal Form* (2NF) dapat dilakukan dengan mengubah struktur relasi dengan cara :

- a. Identifikasikan *Functional Dependence* (FD) relasi *First Normal Form* (1NF)

- b. Berdasarkan informasi tersebut, dekomposisi relasi *First Normal Form* (1NF) menjadi relasi-relasi baru sesuai *Functional Dependence* nya. Jika menggunakan diagram maka simpul-simpul yang berada pada puncak diagram ketergantungan data bertindak *Primary Key* (PK) pada relasi baru

4. Bentuk normal ketiga (*Third Normal Form* / 3NF)

Suatu relasi disebut sebagai *Third Normal Form* jika memenuhi kriteria sebagai berikut :

- a. Jika memenuhi kriteria *Second Normal Form* (2NF)
- b. Jika setiap atribut nonkunci tidak (TDF) (*Non Transitive Dependency*) terhadap *Primary Key* (PK)

Permasalahan dalam *Third Normal Form* (3NF) adalah keberadaan penentu yang tidak merupakan bagian dari *Primary Key* (PK) menghasilkan duplikasi rinci data pada atribut yang berfungsi sebagai *Foreign Key* (FK) (duplikasi berbeda dengan keterangan data)

Mengubah relasi *Second Normal Form* (2NF) menjadi bentuk *Third Normal Form* (3NF) dapat dilakukan dengan mengubah struktur relasi dengan cara :

- a. Identifikasi TDF relasi *Second Normal Form* (2NF)
- b. Berdasarkan informasi tersebut, dekomposisi relasi *Second Normal Form* (2NF) menjadi relasi-relasi baru sesuai TDF-nya.

5. Bentuk normal *Boyce-Codd* (*Boyce-Codd Norm Form* / BCNF)

Bentuk normal *Boyce-Codd Norm Form* (BCNF) dikemukakan oleh R.F. Boyce dan E.F. Codd. Suatu relasi disebut sebagai *Boyce-Codd Norm Form* (BCNF) jika memenuhi kriteria sebagai berikut :

- a. Jika memenuhi kriteria *Third Norm Form* (3NF)
- b. Jika semua atribut penentu (determinan) merupakan CK

6. Bentuk normal keempat (*Forth Norm Form* / 4NF)

Relasi disebut sebagai *Forth Norm Form* (4NF) jika memenuhi kriteria sebagai berikut :

- a. Jika memenuhi kriteria *Boyce-Codd Norm Form*.
- b. Jika setiap atribut didalamnya tidak mengalami ketergantungan pada banyak nilai.

7. Bentuk normal kelima (*Fifth Norm Form* / 5NF)

Suatu relasi memenuhi kriteria *Fifth Norm Form* (5NF) jika kerelasian antar data dalam relasi tersebut tidak dapat direkonstruksi dari struktur relasi yang sederhana.

8. Bentuk normal kunci domain (*Domain Key Norm Form* / DKNF)

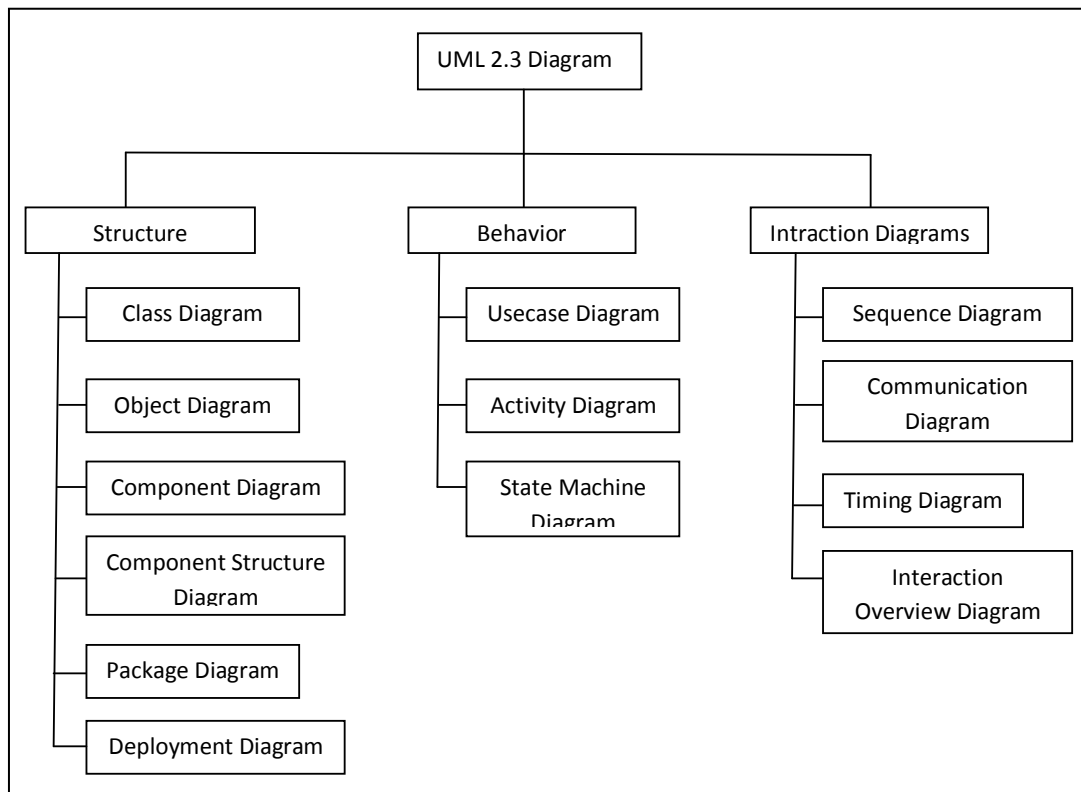
Relasi disebut sebagai *Domain Key Norm Form* (DKNF) jika setiap batasan dapat disimpulkan secara sederhana dengan mengetahui sekumpulan nama atribut dan domainnya selama menggunakan sekumpulan atribut pada kuncinya.

II.7. Mengenal *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 dapat dibuat 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 lebih cocok untuk penulisan piranti lunak dalam bahasa berorientasi objek seperti C++, Java, atau VB. NET. (Jurnal Teknologi Informatika DINAMIK Vol. 14, No. 1; Prastuti Sulistyorini ; Juni 2009 : 23-24)

Saat ini sebagian besar para perancang sistem informasi dalam menggambarkan informasi dengan memanfaatkan UML diagram dengan tujuan utama untuk membantu tim proyek berkomunikasi, mengeksplorasi potensi desain, dan memvalidasi desain arsitektur perangkat lunak atau pembuat program.

Secara filosofi UML diilhami oleh konsep yang telah ada yaitu konsep permodelan *Object Oriented* karena konsep ini menganalogikan sistem seperti kehidupan nyata yang didominasi oleh Program Studi Ilmu Komputer Universitas Mulawarman obyek dan digambarkan atau dinotasikan dalam simbol-simbol yang cukup spesifik. (Jurnal Informatika Mulawarman Vol. 6 No. 1 ; Haviluddin ; Februari 2011 : 1-2)



Gambar II.3. Diagram *Unified Modelling Language* (UML)

(Sumber : Jurnal Informatika Mulawarman Vol. 6 No. 1 ; Havaluddin ; Februari 2011 : 1-2)

II.7.1. Tujuan Pemanfaatan *Unified Modelling Language* (UML)

Tujuan dari penggunaan diagram seperti diungkapkan oleh Schuller J. (2004), *“The purpose of the diagrams is to present multiple views of a system; this set of multiple views is called a model”*.

Berikut tujuan utama dalam desain UML adalah (Sugrue J. 2009) : (Jurnal Informatika Mulawarman Vol. 6 No. 1 ; Havaluddin ; Februari 2011 : 2)

1. Menyediakan bagi pengguna (analisis dan desain sistem) suatu bahasa pemodelan visual yang ekspresif sehingga mereka dapat mengembangkan dan melakukan pertukaran model data yang bermakna.
2. Menyediakan mekanisme yang spesialisasi untuk memperluas konsep inti.
3. Karena merupakan bahasa pemodelan visual dalam proses pembangunannya maka UML bersifat independen terhadap bahasa pemrograman tertentu.
4. Memberikan dasar formal untuk pemahaman bahasa pemodelan.
5. Mendorong pertumbuhan pasar terhadap penggunaan alat desain sistem yang berorientasi objek (OO).
6. Mendukung konsep pembangunan tingkat yang lebih tinggi seperti kolaborasi, kerangka, pola dan komponen terhadap suatu sistem.
7. Memiliki integrasi praktik terbaik.

Ada 4 (empat) prinsip dasar dari pemrograman berorientasi obyek yang menjadi dasar kemunculan UML, yaitu *abstraksi*, *enkapsulasi*, *modularitas* dan *hirarki*. Berikut dijelaskan satu persatu secara singkat. (Jurnal Informatika Mulawarman Vol. 6 No. 1 ; Havaluddin ; Febuari 2011 : 2)

1. *Abstraksi* memfokuskan perhatian pada karakteristik obyek yang paling penting dan paling dominan yang bisa digunakan untuk membedakan obyek tersebut dari obyek lainnya.
2. *Enkapsulasi* menyembunyikan banyak hal yang terdapat dalam obyek yang tidak perlu diketahui oleh obyek lain. Dalam praktek pemrograman, enkapsulasi diwujudkan dengan membuat suatu kelas interface yang akan dipanggil oleh obyek lain, sementara didalam obyek yang dipanggil terdapat

kelas lain yang mengimplementasikan apa yang terdapat dalam kelas *interface*.

3. *Modularitas* membagi sistem yang rumit menjadi bagian-bagian yang lebih kecil yang bisa mempermudah *developer* memahami dan mengelola obyek tersebut.
4. *Hirarki* berhubungan dengan abstraksi dan modularitas, yaitu pembagian berdasarkan urutan dan pengelompokan tertentu. Misalnya untuk menentukan obyek mana yang berada pada kelompok yang sama, obyek mana yang merupakan komponen dari obyek yang memiliki hirarki lebih tinggi. Semakin rendah hirarki obyek berarti semakin jauh abstraksi dilakukan terhadap suatu obyek.

II.7.2. Komponen-Komponen *Unified Modelling Language* (UML)

Sejauh ini para pakar merasa lebih mudah dalam menganalisa dan mendesain atau memodelkan suatu sistem karena UML memiliki seperangkat aturan dan notasi dalam bentuk grafis yang cukup spesifik (Sugrue J. 2009).

Komponen atau notasi UML diturunkan dari 3 (tiga) notasi yang telah ada sebelumnya yaitu *Grady Booch, OOD (Object-Oriented Design)*, *Jim Rumbaugh, OMT (Object Modelling Technique)*, dan *Ivar Jacobson OOSE (Object-Oriented Software Engineering)*.

Pada UML versi 2 terdiri atas tiga kategori, diantaranya : (Jurnal Informatika Mulawarman Vol. 6 No. 1 ; Havaluddin ; Febuari 2011 : 3)

1. Struktur Diagram

Menggambarkan elemen dari spesifikasi dimulai dengan kelas, obyek, dan hubungan mereka, dan beralih ke dokumen arsitektur logis dari suatu sistem.

Beberapa struktur diagram dalam UML terdiri atas :

a. *Class diagram*

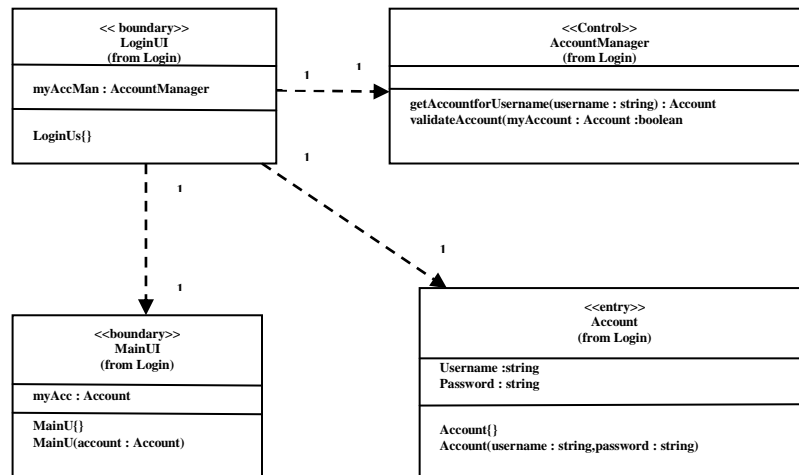
Class diagram menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas.

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.4. Notasi Class Diagram

(Sumber : Jurnal Informatika Mulawarman Vol. 6 No. 1 ; Haviluddin ; Febuari
2011 : 3)

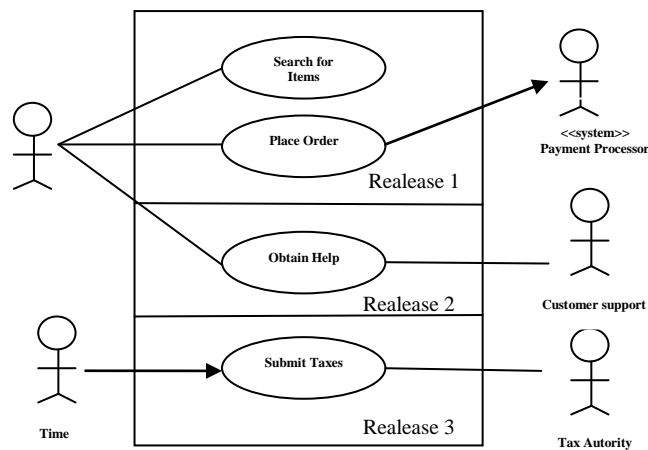
2. Behavior Diagram

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

a. 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



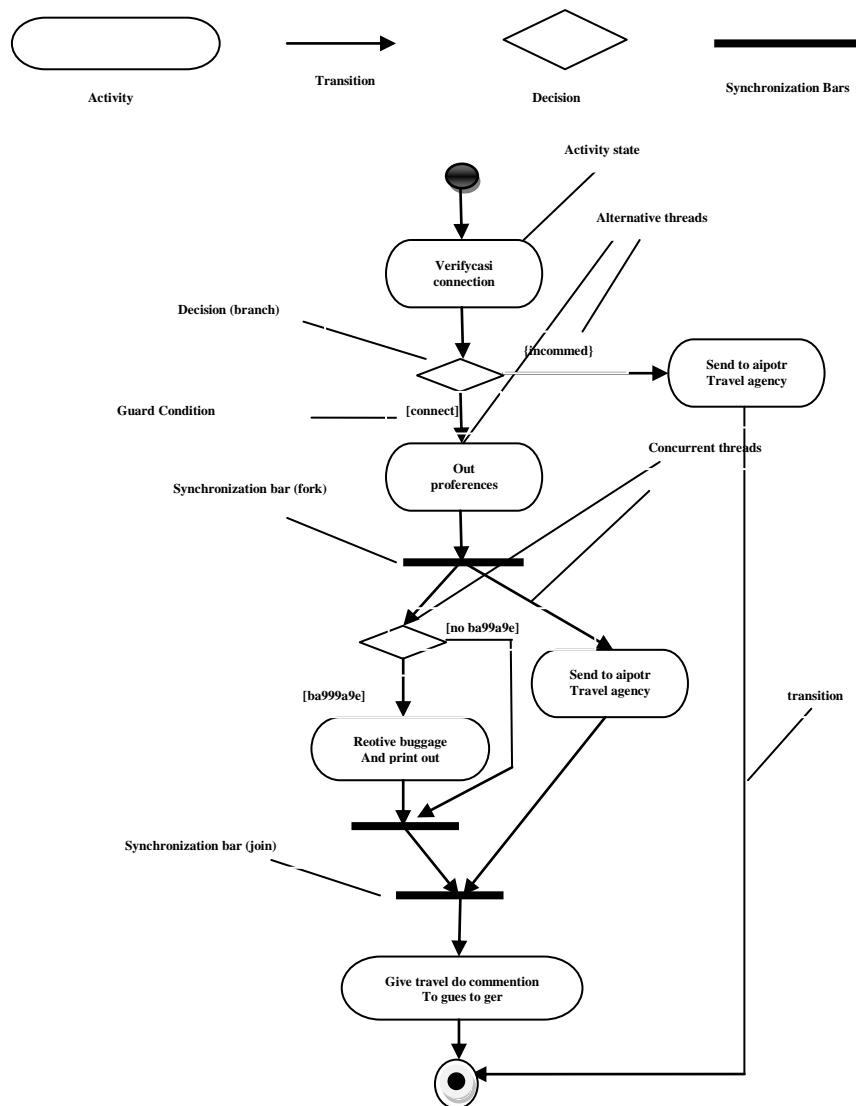
Gambar II.5. Notasi Use Case Diagram

(Sumber : Jurnal Informatika Mulawarman Vol. 6 No. 1 ; Havaluddin ; Febuari
2011 : 4)

b. Activity diagram

Menggambarkan aktifitas-aktifitas, objek, *state*, *transisi state* dan *event*.

Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas. Berikut notasi *object diagram* dapat dilihat pada Gambar II.6. di bawah ini.



Gambar II.6. Notasi Activity Diagram

(Sumber : Jurnal Informatika Mulawarman Vol. 6 No. 1 ; Havaluddin ; Febuari
2011 : 4)

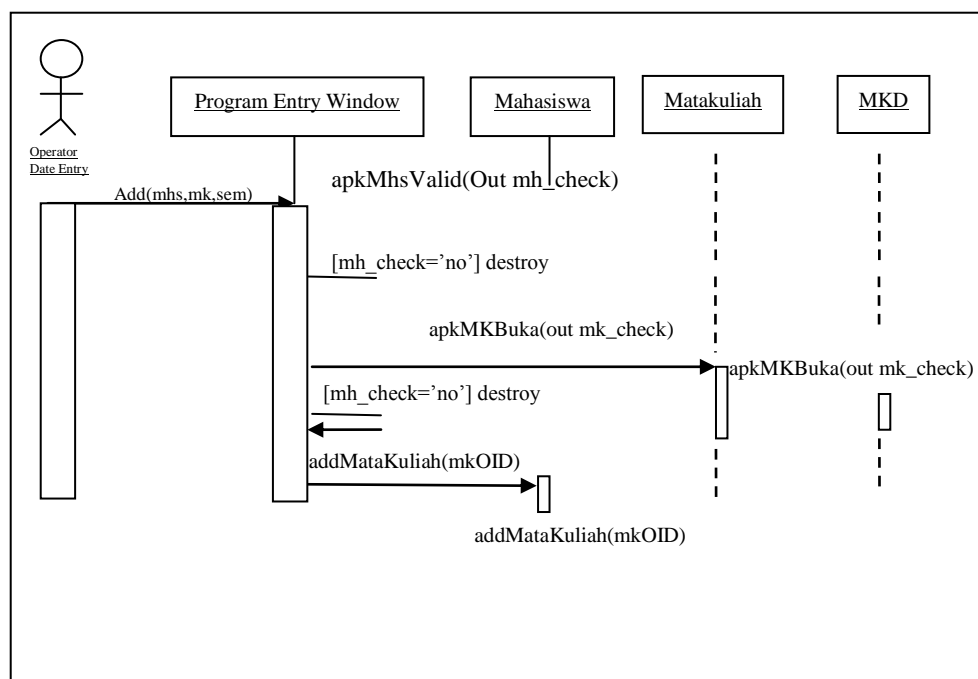
3. Interaction Diagram

Bagian dari *behavior diagram* yang menggambarkan interaksi objek.

Interaction diagram dalam UML salah satunya adalah :

a. *Sequence diagram*

Sequence diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya *sequence diagram* adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case diagram*.



Gambar II.7. Notasi Sequence Diagram

(Sumber : Jurnal Informatika Mulawarman Vol. 6 No. 1 ; Haviluddin ; Febuari 2011 : 5)

Untuk menggambarkan analisa dan desain diagram, UML memiliki seperangkat notasi yang akan digunakan ke dalam tiga kategori diatas yaitu *struktur diagram*, *behaviour diagram*, dan *interaction diagram*. Berikut beberapa notasi dalam *Unified Modelling Language (UML)* diantaranya :

- 1) *Actor*, menentukan peran yang dimainkan oleh user atau sistem lain yang berinteraksi dengan subjek. *Actor* adalah segala sesuatu yang berinteraksi langsung dengan sistem aplikasi komputer, seperti orang, benda atau lainnya. Tugas *actor* adalah memberikan informasi kepada sistem dan dapat memerintahkan sistem untuk melakukan sesuatu tugas.
- 2) *Class diagram*, Notasi utama dan yang paling mendasar pada diagram UML adalah notasi untuk mempresentasikan suatu *class* beserta dengan atribut dan operasinya. *Class* adalah pembentuk utama dari sistem berorientasi objek.
- 3) *Use Case* dan *use case specification*, *Use case* adalah deskripsi fungsi dari sebuah sistem perspektif pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem disebut skenario. *Use case* merupakan awal yang sangat baik untuk setiap fase pengembangan berbasis objek, design, testing, dan dokumentasi yang menggambarkan kebutuhan sistem dari sudut pandang di luar sistem. Perlu diingat bahwa *use case* hanya menetapkan apa yang seharusnya dikerjakan oleh sistem, yaitu kebutuhan fungsional sistem dan tidak untuk menentukan kebutuhan nonfungsional, misalnya: sasaran kinerja, bahasa pemrograman dan lain sebagainya.

- 4) *Interaction*, *Interaction* digunakan untuk menunjukkan baik aliran pesan atau informasi antar obyek maupun hubungan antar obyek.
- 5) *Association*, *Association* menggambarkan navigasi antar *class* (*navigation*), berapa banyak obyek lain yang bisa berhubungan dengan satu obyek (*multiplicity antar class*) dan apakah suatu class menjadi bagian dari class lainnya (*aggregation*).

II. 8. Keuntungan Dan Kelemahan *Unified Modeling Language* (UML)

Unified Modeling Language (UML) merupakan alat bantu, bahasa pemodelan yang dapat digunakan untuk rancang bangun berorientasi-objek. UML dapat digunakan untuk spesifikasi, visualisasi dan dokumentasi sistem pada fase pengembangan (Eriksson dan Penker,1998). (Sumber : Jurnal Informatika Mulawarman Vol. 6 No. 1 ; Havaluddin ; Febuari 2011 : 8-9)

II.8.1. Keuntungan

Karena merupakan bahasa pemodelan visual dalam proses pembangunannya maka UML bersifat independen terhadap bahasa pemrograman tertentu. Taylor (1992) menyatakan bahwa membangun software menggunakan pendekatan teknologi objek memberikan beberapa keuntungan, antara lain: memungkinkan penggunaan kembali objek yang ada (*reusable*), memungkinkan *software* yang baru dengan konstruksi yang lebih besar, *software* berorientasi objek secara umum lebih mudah dimodifikasi dan dirawat karena sebuah objek

dapat dimodifikasi tanpa banyak berpengaruh pada objek yang lain (Sumber : Jurnal Informatika Mulawarman Vol. 6 No. 1 ; Havaluddin ; Febuari 2011 : 9)

II.8.2. Kelemahan

UML dipandang masih mempunyai kekurangan terutama dalam meng-generate kode program secara komplit. Hal ini karena kurangnya cara memodelkan aspek kelakuan internal perangkat lunak untuk dipetakan ke dalam kode program. Seperti yang kita ketahui, diagram UML yang dapat menghasilkan kode hanyalah diagram class, namun itupun hanya sebatas kerangka kodenya saja dan tidak bisa meng-generate badan program-nya (Sumber : Jurnal Informatika Mulawarman Vol. 6 No. 1 ; Havaluddin ; Febuari 2011 : 9)

II.9. *Microsoft Visual Basic 2010*

Visual Basic 2010 merupakan salah satu pemrograman terbaru yang dikeluarkan oleh *Microsoft*, yaitu *Microsoft Visual Studio 2010*. *Visual Studio* merupakan produk pemrograman andalan dari *Microsoft Corporation*, dimana didalamnya berisi beberapa jenis *IDE* pemrograman seperti *Visual Basic*, *Visual C++*, *Visual Web Developer*, *Visual C#* dan *Visual F#*.

Semua *IDE* pemrograman tersebut sudah mendukung penuh implementasi *.Net Framework* terbaru, yaitu *.Net Framework 4.0* yang merupakan pengembangan dari *.Net Framework 3.5*. Adapun database standart yang disertaka adalah *SQL Server 2008 Express*.

Visual Basic 2010 merupakan versi perbaikan dan pengembangan dari versi pendahulunya, yaitu *Visual Basic 2008*. Beberapa pengembangan yang terdapat didalamnya antara lain dukungan terhadap *library* terbaru dari *Microsoft*, yaitu *.Net Framework 4.0*, dukungan terhadap aplikasi berbasis *Cloud Computing*, serta perluasan dukungan terhadap database-database, baik *standalone* maupun *database server*. (Wahana Komputer ; 2011 : 2).

II.10. Microsoft SQL Server 2008

SQL Server 2008 adalah sebuah terobosan baru dari *Microsoft* dalam bidang database. *SQL Server* adalah 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* dibuat pada saat kemajuan dalam bidang hardware sedemikian pesat. Oleh karena itu sudah dapat dipastikan bahwa *SQL Server 2008* membawa beberapa terobosan dalam bidang pengolahan dan penyimpanan data. *Microsoft* merilis *SQL Server 2008* dalam beberapa versi yang disesuaikan dengan segment-segment pasar yang dituju. Versi-versi tersebut adalah sebagai berikut.

Menurut cara pemrosesan data pada prosesor maka *Microsoft* mengelompokkan produk ini berdasarkan 2 jenis yaitu :

1. Versi 32-bit (x86), yang biasanya digunakan untuk komputer dengan *single prosesor* (Pentium 4) atau lebih tepatnya prosesor 32 bit dan sistem operasi Windows XP.

2. Versi 64-bit (x64), yang biasanya digunakan untuk komputer dengan lebih dari satu prosesor (Misalnya Core 2 Duo) dan sistem operasi 64 bit seperti Windows XP 64, Vista, dan Windows 7.

Sedangkan secara keseluruhan terdapat versi-versi seperti berikut ini:

1. Versi Compact, ini adalah versi “Tipis” dari semua versi yang ada. Versi ini seperti versi desktop pada *SQL Server 2000*. Versi ini juga digunakan pada handheld device seperti Pocket PC, PDA, SmartPhone, Tablet PC.
2. Versi Express, ini adalah versi “Ringan” dari semua versi yang ada (tetapi versi ini berbeda dengan versi compact) dan paling cocok untuk latihan para pengembang aplikasi. Versi ini memuat *Express Manager standar*, integrasi dengan CLR dan XML. (Jurnal Sistem Informasi ; Wenny Widya, et al ; 2009 : 3-4)