

BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Sistem merupakan kumpulan elemen yang saling berkaitan yang bertanggung jawab memproses masukan (*input*) sehingga menghasilkan keluaran (*output*). Suatu sistem di dalam 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. (Kusrini ; 2007 : 11)

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* atau 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)

(Kusrini ; 2007 : 10)

Mengingat bahwa EIS, DSS, dan MIS digunakan untuk mendukung manajemen, maka ketiga sistem tersebut sering disebut sistem pendukung manajemen (*management support system* atau MSS). (Kusrini ; 2007 : 11)

II.1.1. Sistem Pendukung Keputusan (DSS)

DSS merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan dan pemanipulasian data. Sistem itu digunakan untuk membantu pengambilan keputusan dalam situasi yang semiterstruktur dan situasi yang tidak terstruktur, di mana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat (Alter, 2002). (Kusrini ; 2007 : 15-16)

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

Aplikasi DSS menggunakan data, memberikan antarmuka pengguna yang mudah, dan dapat menggabungkan pemikiran pengambilan keputusan. Tujuan dari DSS adalah (Turban, 2005) :

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 produktivitas.
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.
8. Mengatasi keterbatasan *kognitif* dalam pemrosesan dan penyimpanan.
(Kusrini ; 2007 : 16-17)

Ditinjau dari tingkat teknologinya, DSS dibagi menjadi 3, yaitu :

1. SPK spesifik
SPK spesifik bertujuan membantu memecahkan suatu masalah dengan karakteristik tertentu. Misalnya, SPK penentuan harga satuan barang.
2. Pembangkit SPK
Suatu *software* yang khusus digunakan untuk membangun dan mengembangkan SPK. Pembangkit SPK akan memudahkan perancangan dalam membangun SPK spesifik.

3. Perlengkapan SPK

Berupa *software* dan *hardware* yang digunakan atau mendukung pembangun SPK spesifik maupun pembangkit SPK.

(Kusrini ; 2007 : 18)

Berdasarkan tingkat dukungannya, DSS dibagi menjadi 6, yaitu :

1. *Retrieve Information Elements*

Inilah dukungan terendah yang bisa diberikan oleh DSS, yakni berupa akses selektif terhadap informasi. Misalnya manajer bermaksud mencari tau informasi mengenai data penjualan atas suatu area pemasaran tertentu.

2. *Analyze Entire File*

Dalam tahap ini, para manajer diberi akses untuk melihat dan menganalisis *file* secara lengkap.

3. *Prepare Reports from Multiple Files*

Dukungan seperti ini cenderung dibutuhkan mengingat para manajer berhubungan dengan banyak aktivitas dalam suatu momen tertentu.

4. *Estimate Decision Consequences*

Dalam tahapan ini, manajer dimungkinkan untuk melihat dampak dari setiap keputusan yang mungkin diambil.

5. *Propose Decision*

Dukungan tahapan ini sedikit lebih maju. Suatu alternatif keputusan bisa disodorkan ke hadapan manajer untuk dipertimbangkan.

6. *Make Decision*

Ini adalah jenis dukungan yang sangat diharapkan dari DSS. Tahapan ini akan memberikan sebuah keputusan yang tinggal menunggu *legitimasi* dari manajer untuk dijalankan. (Kusrini ; 2007 : 18-19)

Keputusan yang diambil untuk menyelesaikan suatu masalah dilihat dari keterstrukturannya yang bisa dibagi menjadi :

1. Keputusan terstruktur (*structured decision*)

Keputusan terstruktur adalah keputusan yang dilakukan secara berulang-ulang dan bersifat rutin. Prosedur pengambilan keputusan sangatlah jelas. Keputusan tersebut terutama dilakukan pada manajemen tingkat bawah. Misalnya, keputusan pemesanan barang dan keputusan penagihan piutang.

2. Keputusan semiterstruktur (*semistructured decision*)

Keputusan semiterstruktur adalah keputusan yang memiliki dua sifat. Sebagian keputusan bisa ditangan oleh komputer dan yang lain tetap harus dilakukan oleh pengambil keputusan. Prosedur dalam pengambilan keputusan tersebut secara garis besar sudah ada, tetapi ada beberapa hal yang masih memerlukan kebijakan dari pengambil keputusan. Biasanya, keputusan semacam ini diambil oleh manajer level menengah dalam suatu organisasi. Contoh keputusan jenis ini adalah pengevaluasian kredit, penjadwalan produksi dan pengendalian persediaan.

3. Keputusan tak terstruktur (*unstructured decision*)

Keputusan tak terstruktur adalah keputusan yang penangannya rumit, karena tidak terjadi berulang-ulang atau tidak selalu terjadi. Keputusan tersebut menuntut pengalaman dan berbagai sumber yang bersifat eksternal. Keputusan tersebut umumnya terjadi pada manajer tingkat atas. Contohnya adalah keputusan untuk pengembangan teknologi baru, keputusan untuk bergabung dengan perusahaan lain dan perekrutan *eksekutif*. (Kusrini ; 2007 : 19-20)

II.1.2. Konsep Sistem Pendukung Keputusan (DSS)

Pada dasarnya sistem pendukung keputusan merupakan pengembangan lebih lanjut dari sistem informasi manajemen terkomputerisasi yang dirancang sedemikian rupa sehingga bersifat interaktif dengan pemakainya. Sifat interaktif dimaksudkan untuk memudahkan integrasi antara berbagai komponen dalam proses pengambilan keputusan seperti prosedur, kebijakan, teknik analisis, serta pengalaman dan wawasan manajerial guna membentuk suatu kerangka keputusan bersifat fleksibel. (Desi Leha Kurniasih ; 2013 : 7)

Konsep Sistem Pendukung Keputusan (SPK)/*Decision Support Sistem* (DSS) pertama kali diungkapkan pada awal tahun 1970-an oleh Michael S. Scott Morton dengan istilah *Management Decision Sistem*. Sistem tersebut adalah suatu sistem yang berbasis komputer yang ditujukan untuk membantu pengambil keputusan dengan memanfaatkan data dan model tertentu untuk memecahkan berbagai persoalan yang tidak terstruktur. . (Desi Leha Kurniasih ; 2013 : 7)

II.1.3. Ciri - Ciri Sistem Pendukung Keputusan (DSS)

Menurut Kosasi dan Kusrini (2007), adapun ciri-ciri sebuah SPK seperti yang dirumuskan oleh Alters Keen adalah sebagai berikut:

1. SPK ditujukan untuk membantu pengambilan keputusan-keputusan yang kurang terstruktur dan umumnya dihadapi oleh para manajer yang berada di tingkat puncak.
2. SPK merupakan gabungan antara kumpulan model kualitatif dan kumpulan data
3. SPK memiliki fasilitas interaktif yang dapat mempermudah hubungan antara manusia dengan komputer.
4. SPK bersifat luwes dan dapat menyesuaikan dengan perubahan-perubahan yang terjadi. (Desi Leha Kurniasih ; 2013 : 7)

II.1.4. Karakteristik, Kemampuan, dan Keterbatasan SPK

Sehubungan banyaknya definisi yang dikemukakan mengenai pengertian dan penerapan dari sebuah SPK, sehingga menyebabkan terdapat banyak sekali pandangan mengenai sistem tersebut. Selanjutnya Turban (1996), menjelaskan terdapat sejumlah karakteristik dan kemampuan dari SPK yaitu:

- a. Karakteristik SPK
 1. Mendukung seluruh kegiatan organisasi
 2. Mendukung beberapa keputusan yang saling berinteraksi
 3. Dapat digunakan berulang kali dan bersifat konstan
 4. Terdapat dua komponen utama, yaitu data dan model
 5. Menggunakan baik data eksternal dan internal

6. Memiliki kemampuan *what-if analysis* dan *goal seeking analysis*
7. Menggunakan beberapa model kuantitatif.

(Desi Leha Kurniasih ; 2013 : 7)

b. Kemampuan SPK

1. Menunjang pembuatan keputusan manajemen dalam menangani masalah semi terstruktur dan tidak terstruktur
2. Membantu manajer pada berbagai tingkatan manajemen, mulai dari manajemen tingkat atas sampai manajemen tingkat bawah
3. Menunjang pembuatan keputusan secara kelompok maupun perorangan
4. Menunjang pembuatan keputusan yang saling bergantung dan berurutan
5. Menunjang tahap-tahap pembuatan keputusan antara lain *intelligensi*, *desain*, *choice*, dan *implementation*
6. Kemampuan untuk melakukan adaptasi setiap saat dan bersifat fleksibel
7. Kemudahan melakukan interaksi sistem
8. Meningkatkan efektivitas dalam pembuatan keputusan daripada efisiensi
9. Mudah dikembangkan oleh pemakai akhir
10. Kemampuan pemodelan dan analisis pembuatan keputusan
11. Kemudahan melakukan pengaksesan berbagai sumber dan format data.

(Desi Leha Kurniasih ; 2013 : 7)

c. Keterbatasan SPK

1. Ada beberapa kemampuan manajemen dan bakat manusia yang tidak dapat dimodelkan, sehingga model yang ada dalam sistem tidak semuanya mencerminkan persoalan sebenarnya.
2. Kemampuan suatu SPK terbatas pada pembendaharaan pengetahuan yang dimilikinya (pengetahuan dasar serta model dasar).
3. Proses-proses yang dapat dilakukan oleh SPK biasanya tergantung juga pada kemampuan perangkat lunak yang digunakannya.

(Desi Leha Kurniasih ; 2013 : 7-8)

SPK tidak memiliki kemampuan intuisi seperti yang dimiliki oleh manusia. Karena walau bagaimana pun canggihnya suatu SPK, hanyalah sautu kumpulan perangkat keras, perangkat lunak dan sistem operasi yang tidak dilengkapi dengan kemampuan berpikir. (Desi Leha Kurniasih ; 2013 : 8)

II.1.5. Tahapan Sistem Pengambilan Keputusan

Menurut Herbert A. Simon ada 4 tahap yang harus dilalui dalam proses pengambilan keputusan yaitu :

1. Penelusuran (*intelligence*)

Tahap ini merupakan tahap pendefinisian masalah serta identifikasi informasi yang dibutuhkan yang berkaitan dengan persoalan yang dihadapi serta keputusan yang akan diambil.

2. Perancangan (*design*)

Tahap ini merupakan tahap analisa dalam kaitan mencari atau merumuskan alternatif-alternatif pemecahan masalah.

3. Pemilihan (*choice*)

Yaitu memilih alternatif solusi yang diperkirakan paling sesuai.

4. Implementasi (*implementation*)

Tahap ini merupakan tahap pelaksanaan dari keputusan yang telah diambil. (Desi Leha Kurniasih ; 2013 : 8)

II.2. Pengertian BLT (Bantuan Langsung Tunai)

Bantuan langsung tunai (BLT), merupakan suatu bentuk bantuan dari pemerintah sebagai bentuk kompensasi dari kenaikan harga Bahan Bakar Minyak(BBM), yang tentunya mengimbas kepada kehidupan masyarakat luas termasuk kalangan masyarakat miskin. Untuk mendapatkan bantuan langsung tunai ini, pemerintah menetapkan beberapa kriteria dalam menentukan siapa saja yang berhak menerima bantuan tersebut. Kriteria tersebut dituangkan kedalam 14 *point*, yang harus dipenuhi oleh setiap rumah tangga sasaran yang akan menerima bantuan langsung tunai tersebut dan diharapkan ke-14 *point* tersebut mampu benar-benar menyaring penerima bantuan langsung tunai tersebut tidak salah sasaran. (Dita Monita ; 2013 : 29)

II.2.1. Kriteria Penilaian

Seiring dipertanyakan siapa golongan (individu, rumah tangga, keluarga atau kelompok) yang pantas mendapatkan dana langsung tunai. Badan Pusat Statistik telah melakukan pendataan jumlah keluarga dengan menggunakan variabel atau indikator masing-masing sebagai kriteria penerima dana tunai

langsung. Syarat-syarat yang harus dipenuhi warga miskin yang menerima BLT antara lain:

1. Luas lantai bangunan tempat tinggal kurang dari 8 meter persegi untuk masing – masing anggota keluarga.
2. Jenis lantai bangunan tempat tinggal terbuat dari tanah, bambu, kayu berkualitas rendah.
3. Jenis dinding bangunan tempat tinggal terbuat dari bambu, rumbia, kayu berkualitas rendah.
4. Fasilitas jamban tidak ada, atau ada tetapi dimiliki secara bersama-sama dengan keluarga lain.
5. Sumber air untuk minum/memasak berasal dari sumur/mata air tak terlindung, air sungai, danau, atau air hujan.
6. Sumber penerangan di rumah bukan listrik.
7. Bahan bakar yang digunakan memasak berasal dari kayu bakar, arang, atau minyak tanah.
8. Dalam seminggu tidak pernah mengonsumsi daging, susu, atau hanya sekali dalam seminggu.
9. Dalam setahun paling tidak hanya mampu membeli pakaian baru satu stel.
10. Makan dalam sehari hanya satu kali atau dua kali.
11. Tidak mampu membayar anggota keluarga berobat ke puskesmas atau poliklinik.
12. Pekerjaan utama kepala rumah tangga adalah petani dengan luas lahan setengah hektare, buruh tani, kuli bangunan, tukang batu, tukang becak,

pemulung, atau pekerja informal lainnya dengan pendapatan maksimal Rp600 ribu per bulan.

13. Pendidikan tertinggi yang ditamatkan kepala rumah tangga bersangkutan tidak lebih dari SD.

14. Tidak memiliki harta senilai Rp500 ribu seperti tabungan, perhiasan emas, TV berwarna, ternak, sepeda motor [kredit/non-kredit], kapal motor, tanah, atau barang modal lainnya. (www.Blogberita.Net).

II.3. FMADM (*Fuzzy Multiple Attribute Decission Making*)

Fuzzy Multiple Attribute Decision Making (FMADM) adalah suatu metode yang digunakan untuk mencari alternatif optimal dari sejumlah alternatif dengan kriteria tertentu. Inti dari FMADM adalah menentukan nilai bobot untuk setiap atribut, kemudian dilanjutkan dengan proses perankingan yang akan menyeleksi alternatif yang sudah diberikan. Pada dasarnya, ada tiga pendekatan untuk mencari nilai bobot atribut, yaitu pendekatan subyektif, pendekatan obyektif dan pendekatan integrasi antara subyektif dan obyektif. Masing-masing pendekatan memiliki kelebihan dan kelemahan. Pada pendekatan subyektif, nilai bobot ditentukan berdasarkan subyektifitas dari para pengambil keputusan, sehingga beberapa faktor dalam proses perankingan alternatif bisa ditentukan secara bebas, sedangkan pada pendekatan obyektif, nilai bobot dihitung secara matematis sehingga mengabaikan subyektivitas dari pengambil keputusan. (Yulison,dkk ; 2012 : 46)

Ada beberapa metode yang dapat digunakan untuk menyelesaikan masalah FMADM, antara lain:

- a. *Simple Additive Weighting Method* (SAW)
- b. *Weighted Product* (WP)
- c. *ELimination Et Choix TRaduisant larealitE* (ELECTRE)
- d. *Technique for Order Preference by Similarity to Ideal Solution* (TOPSIS)
- e. *Analytic Hierarchy Process* (AHP) (Yulison,dkk ; 2012 : 46-47)

II.3.1. SAW (*Simple Additive Weighting*)

Metode SAW sering juga dikenal istilah metode penjumlahan terbobot. Konsep dasar metode SAW adalah mencari penjumlahan terbobot dari *rating* kinerja pada setiap alternatif pada semua atribut. Metode SAW membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat diperbandingkan dengan semua *rating* alternatif yang ada.

(Yulison,dkk ; 2012 : 47)

$$r_{ij} = \begin{cases} \frac{X_{ij}}{\text{Max } X_{ij}} & \text{Jika } j \text{ adalah atribut keuntungan (benefit)} \\ \frac{\text{Min } X_{ij}}{X_{ij}} & \text{Jika } j \text{ adalah atribut biaya (cost)} \end{cases}$$

r_{ij} = nilai *rating* kinerja ternormalisasi

x_{ij} = nilai atribut yang dimiliki dari setiap kriteria

Max x_{ij} = nilai terbesar dari setiap kriteria i

Min x_{ij} = nilai terkecil dari setiap kriteria i

benefit = jika nilai terbesar adalah terbaik

cost = jika nilai terkecil adalah terbaik

dimana r_{ij} adalah *rating* kinerja ternormalisasi dari alternatif A_i pada atribut C_j ;

$i=1,2,\dots,m$ dan $j=1,2,\dots,n$. Nilai preferensi untuk setiap alternatif (V_i) diberikan sebagai:

$$V_i = \sum_{j=1}^n w_j r_{ij}$$

Keterangan :

V_i = ranking untuk setiap alternatif

w_j = nilai bobot dari setiap kriteria

r_{ij} = nilai *rating* kinerja ternormalisasi

Nilai V_i yang lebih besar mengindikasikan bahwa alternatif A_i lebih terpilih.

(Yulison,dkk ; 2012 : 47)

II.3.2. Langkah Penyelesaian SAW (*Simple Additive Weighting*)

Dalam penelitian ini menggunakan metode FMADM dengan metode SAW. Adapun langkah-langkahnya adalah: Memberikan nilai setiap alternatif (A_i) pada setiap kriteria (C_j) yang sudah ditentukan, dimana nilai $i=1,2,\dots,m$ dan $j=1,2,\dots,n$.

1. Memberikan nilai bobot (W) yang juga didapatkan berdasarkan nilai *crisp*.
2. Melakukan normalisasi matriks dengan cara menghitung nilai *rating* kinerja ternormalisasi (r_{ij}) dari alternatif A_i pada atribut C_j berdasarkan persamaan yang disesuaikan dengan jenis atribut –atribut keuntungan/*benefit*=maksimum atau atribut biaya/*cost*=minimum). Apabila berupa artibut keuntungan maka nilai *crisp* (X_{ij}) dari setiap kolom

atribut dibagi dengan nilai *crisp* MAX ($\text{MAX } X_{ij}$) dari tiap kolom, sedangkan untuk atribut biaya, nilai *crisp* MIN ($\text{MIN } X_{ij}$) dari tiap kolom atribut dibagi dengan nilai *crisp* (X_{ij}) setiap kolom.

3. Melakukan proses perankingan untuk setiap alternatif (V_i) dengan cara mengalikan nilai bobot (w_i) dengan nilai *rating* kinerja ternormalisasi (r_{ij}).
(Yulison,dkk ; 2012 : 47)

II.3.3 AHP (*Analytical Hierarchy Process*)

Analytical Hierarchy Process merupakan salah satu metode untuk membantu menyusun suatu prioritas dari berbagai pilihan dengan menggunakan berbagai kriteria. Karena sifatnya yang multikriteria, *Analytical Hierarchy Process* cukup banyak digunakan dalam penyusunan prioritas. Sebagai contoh untuk menyusun prioritas penelitian, pihak manajemen lembaga penelitian sering menggunakan beberapa kriteria seperti dampak penelitian, biaya, kemampuan SDM, dan waktu pelaksanaan. (Dita Monita ; 2013 : 30)

Disamping bersifat multikriteria, *Analytical Hierarchy Process* juga didasarkan pada suatu proses yang terstruktur dan logis. Pemilihan atau penyusunan prioritas dilakukan dengan suatu prosedur yang logis dan terstruktur. Kegiatan tersebut dilakukan oleh ahliahli yang representatif berkaitan dengan alternatif-alternatif yang disusun prioritasnya. (Dita Monita ; 2013 : 30)

Metode *Analytical Hierarchy Process* merupakan salah satu model untuk pengambilan keputusan yang dapat membantu kerangka berfikir manusia. Metode ini mula-mula dikembangkan oleh Thomas L. Saaty pada tahun 70-an. Dasar berpikirnya metode *Analytical Hierarchy Process* adalah proses membentuk skor

secara numeric untuk menyusun ranking setiap alternatif keputusan berbasis pada bagaimana sebaiknya alternatif itu dicocokkan dengan kriteria pembuat keputusan. (Dita Monita ; 2013 : 30)

Proses pengambilan keputusan pada dasarnya adalah memilih suatu alternatif. Peralatan utama *Analytical Hierarchy Process* adalah sebuah hierarki

fungsional dengan input utamanya persepsi manusia. Dengan hierarki, suatu masalah kompleks dan tidak terstruktur dipecahkan ke dalam kelompok-kelompoknya. Kemudian kelompok-kelompok tersebut diatur menjadi suatu bentuk hierarki. (Dita Monita ; 2013 : 30)

Suatu tujuan yang bersifat umum dapat dijabarkan dalam beberapa subtujuan yang lebih terperinci dan dapat menjelaskan maksud tujuan umum. Penjabaran ini dapat dilakukan terus sehingga diperoleh tujuan yang bersifat operasional. Pada hierarki terendah dilakukan proses evaluasi atas alternatif-alternatif yang merupakan ukuran dari pencapaian tujuan utama dan pada hierarki terendah ini dapat ditetapkan dalam satuan apa suatu kriteria diukur. (Dita Monita ; 2013 : 30)

Dalam penjabaran hierarki tujuan, tidak ada suatu pedoman yang pasti mengenai seberapa jauh pembuat keputusan menjabarkan menjadi tujuan yang lebih rendah. Pengambil keputusanlah yang menentukan saat penjabaran tujuan ini berhenti, dengan memperhatikan keuntungan atau kekurangan yang diperoleh bila tujuan tersebut terperinci lebih lanjut. Beberapa hal yang perlu diperhatikan dalam melakukan proses penjabaran hierarki tujuan yaitu:

1. Pada saat penjabaran tujuan ke dalam sub tujuan yang lebih rinci harus selalu memperhatikan apakah setiap tujuan yang lebih tinggi tercakup dalam sub tujuan tersebut.
2. Meskipun hal tersebut dapat dipenuhi, juga perlu menghindari terjadinya pembagian yang terlampau banyak baik dalam arah horizontal maupun vertikal.
3. Untuk itu sebelum menetapkan tujuan harus dapat menjabarkan hierarki tersebut sampai dengan tujuan yang paling lebih rendah dengan cara melakukan tes kepentingan. (Dita Monita ; 2013 : 31)

Analytical Hierarchy Process sering digunakan sebagai metode pemecahan masalah dibanding dengan metode yang lain karena alasan-alasan sebagai berikut :

- a. Struktur yang berhirarki, sebagai konsekuensi dari kriteria yang dipilih, sampai pada subkriteria yang paling dalam.
- b. Memperhitungkan validitas sampai dengan batas toleransi inkonsistensi berbagai kriteria dan alternatif yang dipilih oleh pengambil keputusan.
- c. Memperhitungkan daya tahan output analisis sensitivitas pengambilan keputusan. (Dita Monita ; 2013 : 31)

Analytical hierarchy process dapat memfasilitasi evaluasi pro dan kontra tersebut secara rasional. Dengan demikian, *Analytical hierarchy process* dapat memberikan solusi yang optimal dengan cara yang transparan melalui:

1. Analisis keputusan secara kuantitatif dan kualitatif.
2. Evaluasi dan representasi solusi secara sederhana melalui model hirarki

3. Argumen yang logis.
4. Pengujian kualitas keputusan.
5. Waktu yang dibutuhkan relatif singkat. (Dita Monita ; 2013 : 31)

II.3.4 Langkah Dan Prosedur AHP (*Analytical hierarchy Process*)

Untuk memecahkan suatu masalah dengan menggunakan metode *Analytical Hierarchy Process* diperlukan langkah-langkah sebagai berikut:

1. Mendefinisikan permasalahan dan menentukan tujuan.
2. Menyusun masalah kedalam suatu struktur hierarki sehingga permasalahan yang kompleks dapat ditinjau dari sisi yang detail dan terstruktur.
3. Menyusun prioritas untuk tiap elemen masalah.
4. Melakukan pengujian konsistensi terhadap perbandingan antar elemen yang didapatkan pada tiap tingkat hierarki. (Dita Monita ; 2013 : 31)

Dalam suatu kelompok yang besar, proses penetapan prioritas lebih mudah ditangani dengan membagi para anggota menjadi subkelompok yang lebih kecil dan terspesialisasi, yang masing-masing menangani suatu masalah dengan bidang tertentu dimana anggotanya mempunyai keahlian khusus. Apabila subkelompok ini digabungkan, maka nilai setiap matrik harus diperdebatkan dan diperbaiki. Akan tetapi perdebatan dapat dihindari dan pendapat perseorangan diambil melalui kuisioner dengan membuat nilai akhir dengan menggunakan rata-rata geometric seperti dibawah ini:

$$a_w = \sqrt[n]{a_1 x_2 x \dots x a_n}$$

Keterangan :

a_i = penilaian responden ke - i

a_w = penilaian gabungan

n = banyaknya responden

Perhitungan *Analytical Hierachy Process* Saaty(1993) menjelaskan bahwa elemen-elemen pada setiap baris dari matrik persegi merupakan hasil perbandingan berpasangan. Setiap matrik *pairwise comparison* dicari eigenvektornya untuk mendapat *local priority*. (Dita Monita ; 2013 : 32)

Skala perbandingan berpasangan didasarkan pada nilai-nilai fundamental *Analytical Hierarchy Process* dengan pembobotan dari nilai i untuk sama penting, sampai dengan 9 untuk sangat penting sekali. Berdasarkan susunan matrik perbandingan berpasangan dihasilkan sejumlah elemen pada elemen didalam tingkat yang ada atasnya. (Dita Monita ; 2013 : 32)

Penyimpangan dari konsistensi dinyatakan dalam indeks konsistensi yang didapat dari rumus:

$$CI = \frac{\lambda_{maks} - n}{n - 1}$$

Keterangan :

λ_{maks} = eigenvalue maksimum

n = ukuran matrik
(saaty, 1993)

Indeks konsistensi (C1), matriks random dengan skala penelitian 1 samapi dengan 9, beserta kebalikannya sebagai indeks random (R1). Berdasarkan perhitungan Saaty dengan 500 sampel, jika *judgement* numeric diambil secara acak dari skala 1/9, 1/8, ..., 1,2, ..., 9 akan diperoleh rata-rata konsistensi untuk matrik dengan ukuran berbeda. (Dita Monita ; 2013 : 32)

Perbandingan antara C1 dan R1 untuk suatu matrik didefinisikan sebagai rasio konsistensi (CR). Untuk model *Analytical Hierarchy Process* matrik perbandingan dapat diterima jika nilai konsistensinya tidak lebih dari 0,1 atau sama dengan 0,1. (Dita Monita ; 2013 : 32)

II.4. UML (*Unified Modelling Language*)

UML (*Unified Modelling Language*) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi obyek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain. (Munawar ; 2005 : 17)

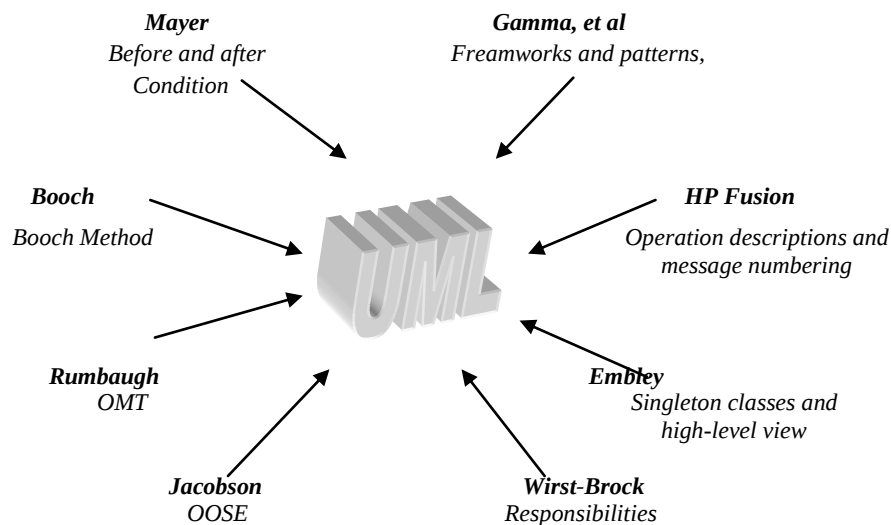
UML merupakan kesatuan dari bahasa pemodelan yang dikembangkan oleh Booch, *Object Modelling Technique* (OMT) dan *Object Oriented Software Engineering* (OOSE). Metode Booch dari Grady Booch sangat terkenal dengan nama metode *Design Object Oriented*. Metode ini menjadikan proses analisis dan design ke dalam empat tahapan interaktif, yaitu dengan mengidentifikasi kelas-kelas dan obyek-obyek, identifikasi semantik dari hubungan obyek dan kelas tersebut, perincian interface dan implementasi. (Munawar ; 2005 : 17)

Keunggulan metode Booch adalah pada detil dan dengan notasi dan elemen. Pemodelan OMT yang dikembangkan oleh rumbaugh didasarkan pada analisis terstruktur dan pemodelan *entity-relationship*. Keunggulan metode ini

adalah dalam penotasian yang mendukung semua konsep OO. Keunggulan yang lain adalah mudah dipelajari karena memiliki notasi yang sederhana namun mencakup seluruh tahapan dalam rekayasa perangkat lunak.

(Munawar ; 2005 : 18)

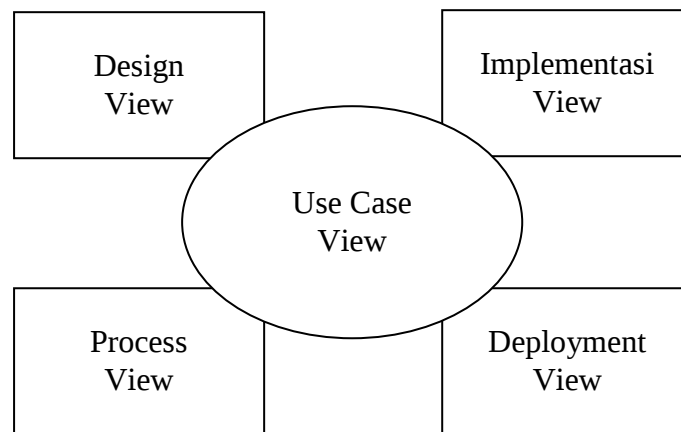
Dengan UML, metode Booch, OMT dan OOSE digabungkan dengan membuang elemen-elemen yang tidak praktis ditambah dengan elemen-elemen dari metode lain yang lebih efektif dan elemen-elemen baru yang belum ada pada metode terdahulu sehingga UML lebih ekspresif dan seragam daripada metode lainnya. Adapun unsur-unsur pembentuk UML ditunjukkan pada gambar II.1 dibawah ini. (Munawar ; 2005 : 18)



Gambar II.1. Unsur-unsur pembentuk UML
Sumber : Munawar ; 2005 : 18

UML dibangun atas model 4+1 view. Model ini didasarkan pada fakta bahwa struktur sebuah sistem dideskripsikan dalam 5 view dimana salah satu diantaranya *use case view*. *Use case view* ini memegang peran khusus untuk

mengintegrasikan content ke *view* yang lain. UML dibangun atas model 4+1 *view* dapat dilihat pada gambar II.2 dibawah ini. (Munawar ; 2005 : 20)



Gambar II.2 : Model 4+1 view

Sumber : Munawar ; 2005 : 20

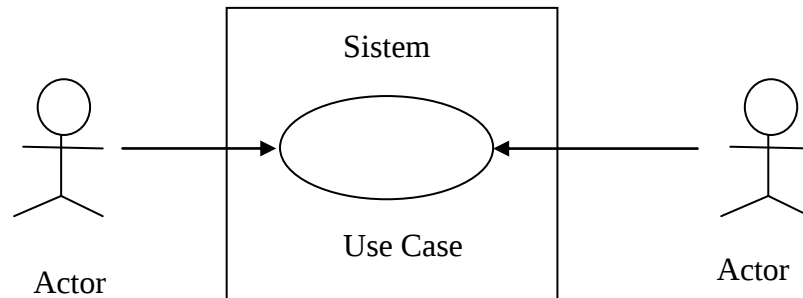
Adapun jenis-jenis dari tipe diagram UML adalah sebagai berikut :

II.4.1. Use Case Diagram

Use Case adalah deskripsi fungsi dari sebuah sistem dari *perspektif* pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Model *use case* adalah bagian dari *requirement* (jacobson et all, 1992). Termasuk disini adalah *problem domain object* model dan penjelasan tentang *user interface*. *Use case* memberikan spesifikasi fungsi-fungsi yang ditawarkan oleh sistem dari *perspektif user*. *Use case* adalah alat bantu terbaik guna menstimulasi pengguna potensial untuk mengatakan tentang suatu sistem dari sudut pandangnya. Ide dasarnya bagaimana melibatkan penggunaan sistem di fase-fase awal analisis dan perancangan sistem. Dengan demikian diharapkan akan bisa dibangun suatu sistem yang bisa membantu pengguna, perlu diingat bahwa *use case* mewakili pandangan diluar

sistem. Diagram *Use Case* menunjukkan 3 aspek dari sistem yaitu : *actor*, *use case* dan *system/sub system boundary*. *Actor* mewakili peran orang, *system* yang lain atau alat ketika berkomunikasi dengan *use case*. Adapun gambar notasi *use case*

dilihat
gambar



dapat
pada
II.3

dibawah ini: (Munawar ; 2005 : 64)

Gambar II.3 Notasi Use Case Diagram
Sumber : Munawar ; 2005 : 64

II.4.2. Class Diagram

Class Diagram merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas didalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class Diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan.

(Windu Gata dan Grace Gata ; 2010 : 8)

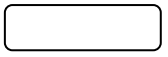
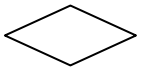
Class Diagram secara khas meliputi : Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operation/Method*), dan *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. (Windu Gata dan Grace Gata ; 2010 : 8)

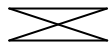
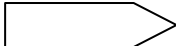
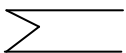
II.4.3. Activity Diagram

Activity Diagram adalah teknik untuk mendiskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. *Activity Diagram* mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa. Adapun simbol *activity diagram* dapat dilihat pada tabel II.1:

(Munawar ; 2005 : 109)

Tabel II.1. Simbol Activity Diagram

SIMBOL	KETERANGAN
	Titik Awal
	Titik Akhir
	Activity
	Pilihan untuk pengambilan keputusan
	Fork : digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu

	Rake ; menunjukkan adanya dekomposisi
	Tanda waktu
	Tanda pengiriman
	Tanda penerimaan
	Aliran Akhir (Flow Final)

Sumber : Munawar ; 2005 : 110

II.4.4. Sequence Diagram

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. (Windu Gata dan Grace Gata ; 2010 : 7)

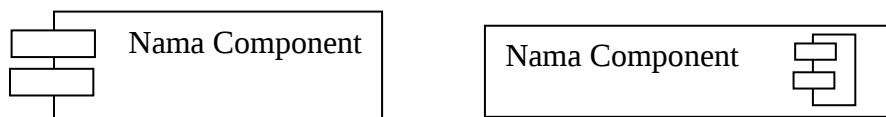
II.4.5. Collaboration Diagram

Collaboration diagram adalah perluasan dari *obyek diagram* (*obyek diagram* menunjukkan obyek-obyek dan hubungannya satu dengan yang lain). *Collaboration diagram* menunjukkan *message-message* obyek yang dikirimkan satu sama lain. Untuk menunjukkan sebuah pesan, buatlah tanda panah di dekat garis asosiasi di antara 2 obyek. Arah panah menunjukkan obyek yang menerima pesan. Label di dekat panah menunjukkan pesannya apa. *Tipikal message* meminta kepada *obyek* yang menerima untuk menjalankan salah satu *operationnya*. Sepasang tanda kurung digunakan untuk mengakhiri *message*. Jika ada parameter bisa diletakkan di antara tanda kurung. (Munawar ; 2005 : 101)

II.4.6. Component Diagram

Component Diagram mengandung *component*, *interface* dan *relationship*.

Notasi *component Diagram* dapat dilihat seperti gambar II.4 di bawah ini :



Gambar II.4 Notasi Component Diagram

Sumber : Munawar ; 2005 : 122

Menurut Fowler (2004) hal penting pada *component* adalah *component* mewakili potongan-potongan yang *independent* yang bisa dipesan dan diperbaharui sewaktu-waktu. Dengan demikian pembagian sistem ke dalam *component-component* lebih banyak didorong oleh kepentingan marketing daripada kepentingan teknis. (Munawar ; 2005 : 122)

II.4.7. Deployment Diagram

Deployment Diagram menunjukkan tata letak sebuah sistem secara fisik, menampilkan bagian-bagian *software* yang berjalan pada bagian-bagian *hardware*. Sistem terdiri dari node-node dimana setiap node diwakili untuk sebuah kubus. Garis yang menghubungkan antara 2 kubus menunjukkan hubungan di antara kedua node tersebut. Tipe *node* bisa berupa *device* yang berwujud *hardware* dan bisa juga *processor*. (Munawar ; 2005 : 127)

II.5. Basis Data (Database)

Database menurut Stephens dan Plew (2000), adalah mekanisme yang digunakan untuk menyimpan informasi atau data. Informasi adalah sesuatu yang kita gunakan sehari-hari untuk berbagai alasan. Dengan *database*, pengguna

dapat menyimpan data secara terorganisasi. Setelah data disimpan, informasi harus mudah diambil. Kriteria dapat digunakan untuk mengambil informasi. Cara data disimpan dalam *database* menentukan seberapa mudah mencari informasi berdasarkan banyak kriteria. Data pun harus mudah ditambahkan ke dalam *database*, dimodifikasi, dan dihapus. (Janner Simarmata dan Iman Paryudi; 2006:1)

II.5.1. SQL Server 2008

SQL Server 2008 adalah sebuah terobosan baru dari *Microsoft* 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* 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. (Wahana ; 2010 : 2)

Hardware yang dibutuhkan *SQL Server 2008* minimal yaitu *Processor 1 Ghz*, *memory 512 Mb*, *OS Windows Xp 32 bit - Windows 7*. Walaupun dapat diinstal pada sistem komputer dengan *memory 512 Mb*, tetapi disarankan menggunakan *memory 1 Gb*. Dan untuk jaringannya diperlukan *shared memory*, *TCP/IP*, *named pipes* dan *VIA (Virtual Interface Adapter)*. (Wahana ; 2010 : 2)

II.6. Normalisasi

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basisdata relasional. Pada dasarnya, normalisasi

adalah proses dua langkah yang meletakkan data dalam bentuk tabulasi dengan menghilangkan kelompok berulang lalu menghilangkan data yang terduplikasi dari table relasional. (Janner Simarmata dan Iman Paryudi ; 2006 : 77)

Teori normalisasi didasarkan pada konsep bentuk normal. Sebuah table relasional dikatakan berada pada bentuk normal tertentu jika table memenuhi himpunan batasan tertentu. Ada lima bentuk normal yang telah ditemukan. (Janner Simarmata dan Iman Paryudi ; 2006 : 77)

II.6.1. Bentuk – Bentuk Normalisasi

1. Bentuk Normal Pertama (1NF)

Sebagai contoh yaitu sebuah perusahaan yang mendapatkan barang dari sejumlah pemasok. Masing-masing pemasok berada pada satu kota. Sebuah kota dapat mempunyai lebih dari satu pemasok dan masing-masing kota mempunyai kode status tersendiri. Masing-masing pemasok bisa menyediakan banyak barang. Tabel relasionalnya dapat dilihat pada gambar II.5 dibawah ini : (Janner Simarmata dan Iman Paryudi ; 2006 : 79-80)

PEMASOK (p#,status,kota,b#,qty) di mana

p# : kode pemasok (kunci utama)

status : kode status kota

kota : nama kota

b# : barang yang dipasok

qty : jumlah barang yang dipasoks

p#	status	Kota	b#	qty
p1	20	Yogyakarta	b1	300
p1	20	Yogyakarta	b2	200
p1	20	Yogyakarta	b3	400
p1	20	Yogyakarta	b4	200
p1	20	Yogyakarta	b5	100
p1	20	Yogyakarta	b6	100
p2	10	Medan	b1	300
p2	10	Medan	b2	400
p3	10	Medan	b2	200
p4	20	Yogyakarta	b2	200
p4	20	Yogyakarta	b4	300
p4	20	Yogyakarta	b5	400

Gambar II.5 Tabel Bentuk Normal Pertama (1NF)
Sumber : Janner Simarmata dan Iman Paryudi ; 2006 : 80

2. Bentuk Normal Kedua (2NF)

Bentuk normal kedua menyatakan bahwa table dengan kunci utama gabungan hanya dapat berada pada 1NF, tetapi tidak pada 2NF. Sebuah table relasional berada pada bentuk normal kedua jika dia berada pada 1NF dan setiap kolom bukan kunci yang sepenuhnya tergantung pada kunci utama. Ini dapat digambarkan dengan membuat daftar ketergantungan fungsional : (Janner Simarmata dan Iman Paryudi ; 2006 : 81)

$p\# \rightarrow kota, status$

$kota \rightarrow status$

$(p\#, b\#) \rightarrow qty$

Proses mengubah tabel 1NF ke 2NF adalah :

1. Tentukan sembarang kolom penentu selain kunci gabungan dan kolom-kolom yang ditentukannya.
2. Buat dan beri nama tabel baru untuk masing-masing penentu dan kolom-kolom yang ditentukannya.
3. Pindahkan kolom-kolom yang ditentukan dari tabel asal ke tabel baru. Penentu akan menjadi kunci utama pada tabel baru.
4. Hapus kolom yang baru dipindahkan dari tabel asal, kecuali penentu yang akan berfungsi sebagai kunci tamu.
5. Tabel asal bisa diberi nama baru.

(Janner Simarmata dan Iman Paryudi ; 2006 : 81)

Pada contoh, memindahkan kolom, p#,status,dan kota ke tabel baru yang disebut PEMASOK2. Kolom p# menjadi kunci utama tabel ini. Berikut tabel bentuk normal kedua dapat dilihat pada gambar II.6 dibawah ini :

(Janner Simarmata dan Iman Paryudi ; 2006 : 82)

PEMASOK2

p#	status	Kota
p1	20	Yogyakarta
p2	10	Medan
p3	10	Medan
p4	20	Yogyakarta
p5	30	Bandung

BARANG

p#	b#	qty
p1	b1	300
p1	b2	200
p1	b3	400
p1	b4	200
p1	b5	100
p1	b6	100
p2	b1	300
p2	b2	400
p3	b2	200
p4	b2	200
p4	b4	300
p4	b5	400

Gambar II.6 Tabel Bentuk Normal Kedua (2NF)
Sumber : Janner Simarmata dan Iman Paryudi ; 2006 : 82

3. Bentuk Normal Ketiga (3NF)

Bentuk normal ketiga mengharuskan semua kolom pada tabel relasional tergantung hanya pada kunci utama. Sebuah tabel berada pada bentuk normal ketiga (3NF) jika tabel sudah berada pada 2NF dan setiap kolom yang bukan kunci tidak tergantung secara transitif pada kunci utamanya. (Janner Simarmata dan Iman Paryudi ; 2006 : 82)

Konsep ketergantungan transitif dapat digambarkan dengan menunjukkan ketergantungan fungsional pada PEMASOK2, yaitu :

PEMASOK2.p# $\rightarrow$ PEMASOK.status

PEMASOK2.p# $\rightarrow$ PEMASOK.kota

PEMASOK2.kota $\rightarrow$ PEMASOK.status

Proses mengubah tabel menjadi 3NF adalah :

1. Tentukan semua penentu selain kunci utama dan kolom yang ditentukannya.
2. Buat dan beri nama tabel baru untuk masing-masing penentu dan kolom yang ditentukannya.

3. Pindahkan kolom yang ditentukan dari tabel asal ke tabel baru. Penentu menjadi kunci utama tabel baru.
4. Hapus kolom yang baru saja dipindahkan dari tabel asal, kecuali penentu yang akan berfungsi sebagai kunci tamu.
5. Tabel asal bisa diberi nama baru.

(Janner Simarmata dan Iman Paryudi ; 2006 : 83)

Untuk mengubah PEMASOK2 menjadi 3NF, membuat tabel baru yang disebut KOTA_STATUS dan memindahkan kolom kota dan status ke tabel baru. Status dihapus dari tabel asal, kota tetap dibiarkan karena akan berfungsi sebagai kunci asing bagi KOTA_STATUS dan tabel asal diberi nama baru PEMASOK_KOTA. Berikut tabel bentuk normal ketiga dapat dilihat pada gambar II.7 dibawah ini : (Janner Simarmata dan Iman Paryudi ; 2006 : 83)

PEMASOK_KOTA

p#	kota
p1	Yogyakarta
p2	Medan
p3	Medan
p4	Yogyakarta
p5	Bandung

KOTA_STATUS

Kota	status
Yogyakarta	20
Medan	10
Bandung	30
Semarang	40

Gambar II.7 Tabel Bentuk Normal Ketiga (3NF)

Sumber : Janner Simarmata dan Iman Paryudi ; 2006 : 83

4. Bentuk Normal *Boyce-Code* (BCNF)

Bentuk normal *Boyce-Code* (BCNF) adalah versi 3NF yang lebih teliti dan berhubungan dengan tabel relasional yang mempunyai banyak kunci kandidat, kunci kandidat gabungan dan kunci kandidat yang saling tumpang tindih. (Janner Simarmata dan Iman Paryudi ; 2006 : 84)

BCNF didasarkan pada konsep tertentu. Sebuah kolom penentu adalah kolom dimana kolom-kolom lain sepenuhnya tergantung secara fungsional. Sebuah tabel relasional berada pada BCNF jika dan hanya jika setiap penentu adalah kunci kandidat. (Janner Simarmata dan Iman Paryudi ; 2006 : 85)

5. Bentuk Normal Keempat (4NF)

Bentuk normal keempat (4NF) didasarkan pada konsep ketergantungan *multivalued* (MVD). Sebuah ketergantungan *multivalued* terjadi ketika dalam sebuah tabel relasional yang mengandung setidaknya tiga kolom, satu kolom mempunyai banyak baris bernilai sama, tetapi kolom lain bernilai berbeda. (Janner Simarmata dan Iman Paryudi ; 2006 : 85)

Misalnya, pegawai ditugaskan ke banyak proyek dan ia mempunyai banyak keahlian. Jika kita mencatat informasi ini pada satu tabel, ketiga atribut harus digunakan sebagai kunci karena tidak ada satu atribut pun yang dapat secara unik mengidentifikasi sebuah record. Untuk mengubah sebuah tabel dengan ketergantungan *multivalued* kedalam 4NF, pindahkan masing-masing pasangan MVD ke tabel baru. Berikut tabel bentuk normal keempat dapat dilihat pada gambar II.8 dibawah ini : (Janner Simarmata dan Iman Paryudi ; 2006 : 85-86)

PEGAWAI_PROYEK

peg#	Pry#
1211	p1
1211	p3

PEGAWAI_AHLI

peg#	ahli
1211	Analisis
1211	Perancangan
1211	Pemrograman

Gambar II.8 Tabel Bentuk Normal Keempat (4NF)
Sumber : Janner Simarmata dan Iman Paryudi ; 2006 : 86

II.7. *Microsoft Visual Studio 2010*

Visual Basic 2010 merupakan salah satu bagian dari produk pemrograman terbaru yang dikeluarkan oleh *Microsoft*, yaitu *Microsoft Visual Studio 2010*. Sebagai produk lingkungan pengembangan terintegrasi atau IDE andalan yang dikeluarkan oleh *Microsoft Visual Studio 2010* menambahkan perbaikan-perbaikan fitur dan fitur baru yang lebih lengkap dibandingkan versi *Visual Studio* pendahulunya, yaitu *Microsoft Visual Studio 2008*. (Wahana ; 2010 : 2)

Visual Studio merupakan produk pemrograman andalan dari *Microsoft Corporation*, yang 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*. (Wahana ; 2010 : 2)

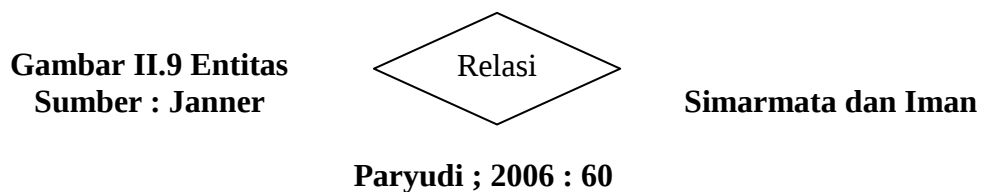
II.8. ERD (*Entity Relationship Diagram*)

Entity Relationship Diagram adalah alat pemodelan data utama dan akan membantu mengorganisasi data dalam suatu proyek ke dalam entitas-entitas dan menentukan hubungan antar entitas. Proses memungkinkan analis menghasilkan struktur basisdata yang baik sehingga data dapat disimpan dan diambil secara efisien. Adapun Struktur logis (skema *database*) dapat ditunjukkan secara grafis dengan diagram ER yaitu sebagai berikut :

(Janner Simarmata dan Iman Paryudi ; 2006 : 67)

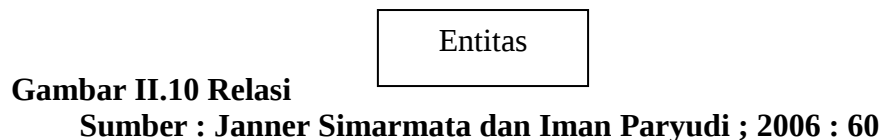
II.8.1. Entitas (*Entity*)

Entitas adalah sesuatu yang nyata atau abstrak dimana kita akan menyimpan data. Ada 4 kelas entitas, yaitu misalnya pegawai, pembayaran, kampus, dan buku. Contoh suatu entitas disebut instansi, misalnya pegawai Adi, pembayaran Joko, dan lain sebagainya. Entitas digambarkan dalam bentuk persegi panjang. Berikut lambang entitas ditunjukkan pada gambar II.9 dibawah ini :
(Janner Simarmata dan Iman Paryudi ; 2006 : 67)



II.8.2. Relasi (*Relationship*)

Relasi adalah hubungan alamiah yang terjadi antara satu atau lebih entitas, misalnya proses pembayaran pegawai. Kardinalitas menentukan kejadian suatu entitas untuk satu kejadian pada entitas yang berhubungan. Misalnya, mahasiswa bisa mengambil banyak mata kuliah. Relasi digambarkan dalam bentuk belah ketupat. Berikut lambang relasi ditunjukkan pada gambar II.10 dibawah ini :
(Janner Simarmata dan Iman Paryudi ; 2006 : 67)



II.8.3. Atribut (*Attribute*)

Atribut adalah ciri umum semua atau sebagian besar instansi pada entitas tertentu. Sebutan lain atribut adalah properti, elemen data, dan *field*. Misalnya nama, alamat, nomor pegawai, dan gaji adalah atribut entitas pegawai. Sebuah atribut atau kombinasi atribut yang mengidentifikasikan satu dan hanya satu instansi suatu entitas disebut kunci utama atau pengenal. Misalnya, nomor pegawai adalah kunci utama untuk pegawai. Atribut digambarkan dalam bentuk elips. Berikut lambang atribut ditunjukkan pada gambar II.11 dibawah ini :

(Janner Simarmata dan Iman Paryudi ; 2006 : 67)

Gambar II.11 Atribut
Sumber : Janner Simarmata dan Iman Paryudi ; 2006 : 60

