BAB II

TINJAUAN PUSTAKA

II.1. Sistem Pendukung Keputusan

Menurut Jurnal Hilyah M. (2012 : 50) Konsep Sistem Pendukung Keputusan pertama kali diperkenalkan pada awal tahun 1970-an oleh Michael S. Scott Morton dengan istilah *Management Decision System* (Sprague, 1982). Konsep pendukung keputusan ditandai dengan sistem interaktif berbasis komputer yang membantu pengambil keputusan memanfaatkan data dan model untuk menyelesaikan masalah-masalah yang tidak terstruktur. Pada dasarnya SPK diracang untuk mendukung seluruh tahap pengambilan keputusan mulai dari mengidentifikasi masalah, memilih data yang relevan, menentukan pendekatan yang digunakan dalam prosen pengambilan keputusan, sampai mengevaluasi pemilihan alternatif.

Menurut Jurnal Ahmad Khaidir (2014 : 149), Pengambilan keputusan merupakan proses pemilihan alternative tindakan untuk mencapai tujuan atau sasaran tertentu. Pengambilan keputusan dilakukan dengan pendekatans istematis terhadap permasalahan melalui proses pengumpulan data menjadi informasi serta ditambah dengan faktor - faktor yang perlu dipertimbangkan dalam pengambilan keputusan.

Menurut Keen dan Scoot Morton : “Sistem Pendukung Keputusan merupakan penggabungan sumber-sumber kecerdasan individu dengan kemampuan komponen untuk memperbaiki kualitas keputusan. Sistem Pendukung Keputusan juga merupakan sistem informasi berbasis komputer untuk manajemen pengambilan keputusan yang menangani masalah-masalah semi

struktur “.

Dengan pengertian diatas dapat dijelaskan bahwa sistem pendukung keputusan bukan merupakan alat pengambilan keputusan, melainkan merupakan sistem yang membantu pengambil keputusan dengan melengkapi mereka dengan informasi dari data yang telah diolah dengan relevan dan diperlukan untuk membuat keputusan tentang suatu masalah dengan lebih cepat dan akurat. Sehingga sistem ini tidak dimaksudkan untuk menggantikan pengambilan keputusan dalam proses pembuatan keputusan.

Alter (2002 : 38) mendefinisikan Sistem pendukung keputusan atau *Decision Support Systems* (DSS) adalah sistem informasi interkatif yang menyediakan informasi, pemodelan, dan pemanipulasian data yang digunakan untuk membantu pengambilan keputusan pada situasi yang semiterstruktur dan situasi yang tidak terstruktur dimana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat. Konsep DSS dikemukaakan pertama kali oleh Scoot Morton pada tahun 1971 (Turban, McLean, dan Wetherbe, 1999). Beliau mendefinisikan cikal bakal DSS tersebut sebagai : “*system* berbasis computer yang interaktif, yang membantu pengambil keputusan dengan menggunakan data dan model untuk memecahkan persoalan-persoalan tak terstruktur”

II.1.1. Pengertian Sistem

Menurut Jurnal Dina Andhayati (2010 : 146) Suatu sistem didefenisikan sebagai suatu kesatuan yang terdiri dari dua atau lebih komponen atau subsistem yang berintegrasi untuk mencapai suatu tujuan. Suatu system mempunyai

karateristik yaitu mempunyai komponen, batas system, lingkungan luar sistem, penghubung, masukan, keluaran, pengolah dan sasaran.

Informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna bagi penerimanya yang menggambarkan suatu kejadian yang nyata yang digunakan untuk pengambilan keputusan. Sumber dari informasi adalah data. Data merupakan bentuk yang masih mentah, belum dapat bercerita banyak, sehingga perlu diolah lebih lanjut. Data diolah melalui suatu model untuk dihasilkan informasi. Data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai input, diproses kembali lewat suatu model dan seterusnya membentuk siklus.

Sistem Informasi dapat didefinisikan sebagai suatu sistem di dalam suatu organisasi yang merupakan kombinasi dari orang, fasilitas, teknologi, media, prosedur dan pengendalian yang ditujukan untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin tertentu, memberi sinyal kepada manajemen dan yang lainnya terhadap kejadian internal dan eksternal yang penting dan menyediakan suatu dasar informasi untuk pengambilan keputusan yang cerdik.

Sistem Informasi Manajemen (SIM) dapat didefinisikan sebagai kumpulan dari interaksi sistem-sistem informasi yang bertanggung jawab mengumpulakn dan mengolah data untuk menyediakan informasi yang berguna untuk semua

tingkatan manajemen di dalam kegiatan perencaaan dan pengendalian. SIM yang komplek dapat melibatkan elemen komputer, sehingga SIM selalu berhubungan dengan pengolahan informasi yang didasarkan pada komputer. (Jogiyanto, 1992)

II.1.2. Karakterisitik Sistem

Model Umum sebuah sistem terdiri dari input, proses, output. Hal ini merupakan konsep sebuah sistem yang sangat sederhana mengingat sebuah sistem dapat mempunyai beberapa masukan dan keluaran sekaligus. Selain itu sebuah sistem juga memiliki karakteristik atau sifat-sifat tertentu, yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem. Adapun karakteristik yang dimaksud adalah sebagai berikut: (Tata Sutabri ; 2012 : 13)

1. Komponen Sistem (*Components*)

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

2. Batasan Sistem (*Boundary*)

   Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem lainnya atau sistem dengan lengkungan luar. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisah-pisahkan.

3. Lingkungan Luar Sistem (*Environtment)*

Bentuk apapun yang ada di luar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut dengan lingkungan luar sistem ini dapat menguntungkan dan dapat juga merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energy bagi setiap sistem tersebut, yang dengan demikian lingkungan luar tersebut harus selalu dijaga dan dipelihara. Sedangkan lingkungan luar yang merugikan harus dikendalikan. Kalau tidak maka akan mengganggu kelangsungan hidup sistem tersebut.

4. Penghubung Sistem (*Interface*)

   Media yang menghubungkan sistem dengan subsistem yang lain disebut dengan penghubung sistem atau interface. Penghubung ini memungkinkan sumber-sumber daya mengalir dari suatu subsistem ke subsistem yang lain. Keluaran suatu subsistem akan menjadi masukan untuk subsistem yang lain dengan melewati penguhubung. Dengan demikian terjadi suatu integrasi sistem yang membentuk suatu kesatuan. (Tata Sutabri ; 2012 : 13)

II.1.3. Pengertian Keputusan

Menurut Jurnal Ellya Sestri (2013 : 100) pengambilan keputusan merupakan hal vital dalam menentukan kebijakan yang harus diambil dalam menghadapi persaingan di dunia bisnis. Pengambilan keputusan dapat dipengaruhi oleh beberapa aspek, dan hal ini dapat mempengaruhi kecepatan dalam mengambil keputusan dimana pengambilan keputusan harus cepat dan akurat.

II.1.4. Pengertian Sistem Pendukung Keputusan

Dalam Jurnal Dita Monita (2013 : 30) menjelaskan Sistem pendukung keputusan adalah bagian dari sistem informasi berbasis computer (termasuk sistem berbasis pengetahuan (manajemen pengetahuan) yang dipakai untuk mendukung pengambilan keputusan dalam suatu organisasi atau perusahaan. Dapat juga dikatakan sebagai sistem komputer yang mengolah datamenjadi informasi untuk mengambil keputusan dari masalah semiterstruktur yang spesifik.

Dalam Jurnal Heny Pratiwi (2014 : 96), Menurut Kendal dan Kendall (2002), *Decision Support System* (DSS) atau sistem pendukung keputusan hampir sama dengan sistem informasi manajemen tradisioanal karena keduanya tergantung pada basisdata sebagai sumber data. SPK menekankan pada fungsi pendukung pembuatan keputusan diseluruh tahap-tahapnya, sebagai pendamping keputusan aktual yang masih dibuat oleh wewenang eksekutif sebagai pembuat keputusan. Pada dasarnya sistem pendukung keputusan adalah sistem yang tidak bisa dipisahkan dari teknologi komputer. Secara umum SPK berfungsi membantu pengambilan keputusan secara efektif sehingga permasalahan yang dihadapi dapat dengan cepat mendapatkan solusinya.

Dalam Jurnal Heny Pratiwi (2014 : 96), Menurut Turban (2005), tujuan sistem pendukung keputusan yaitu :

1. Membantu manajer dalam mengambil keputusan atas masalah semi terstuktur.
2. Memberikan dukungan atas pertimbangan manajer dan bukan dimaksudkan untuk menggantikan manajer.

3. Meningkatkan efektivitas keputusan yang diambil manajer lebih dari pada perbaikan efesiensinya.
4. Kecepatan komputasi. Komputer memungkinkan para pengambil keputusan untuk melakukan banyak komputasi secara cepat dengan biaya yang rendah.
5. Peningkatan produktivitas. Membangun satu kelompok pengambilan keputusan, terutama oleh para pakar, akan meningkatkan biaya. Pendukung berupa perangkat terkomputerisasi dapat mengurangi kelompok dan memungkinkan anggotanya untuk berada diberbagai lokasi yang berbeda-beda.
6. Meningkatkan kualitas. Komputer dapat meningkatkan kualitas keputusan yang dibuat.
7. Berdaya asing. Manajemen dan pemberdayaan sumber daya perusahaan.
8. Mengatasi keterbatasan kognitif dalam pemrosesan dan penyimpanan.

Dalam Jurnal Iskandar Z. Nasibu (2009 : 182-183) Sistem Pendukung Keputusan (*Decision Support System/DSS*) adalah sebuah sistem yang memberikan dukungan kepada seorang manajer, atau kepada sekelompok manajer yang relatif yang bekerja sebagai tim pemecah masalah, dalam memecahkan masalah semi terstruktur dengan memberikan informasi atau saran mengenai keputusan tertentu. Informasi tersebut dapat diberikan dalam bentuk laporan berkala, laporan khusus, maupun output dalam model matematis. Model tersebut juga mempunyai kemampuan untuk memberikan saran dalam tingkat yang bervariasi.

Karakteristik umum dari sebuah sistem pendukung keputusan (DSS) yang ideal yaitu:

1. DSS adalah sebuah sistem berbasis komputer dengan antarmuka antara mesin/ komputer dan pengguna.
2. DSS ditujukan untuk membantu pembuat keputusan dalam menyelesaikan suatu masalah dalam berbagai level manajemen dan bukan untuk menggantikan posisi manusia sebagai pembuat keputusan.
3. DSS menggunakan data, basis data dan analisa model-model keputusan.
4. DSS bersifat adaptif, efektif, interaktif, *easy to use* dan fleksibel.
5. DSS menyediakan akses tertahap berbagai macam format dan tipe sumber data *(data source).*

Selanjutnya di dalam DSS terdapat tiga tujuan yang harus dicapai yaitu:

1. Membantu manajer dalam pembuatan keputusan untuk memecahkan masalah semi terstruktur.
2. Mendukung keputusan manajer, dan bukannya mengubah atau menggati keputusan tersebut.
3. Meningkatkan efektivitas menajer dalam pembuatan keputusan, dan bukannya peningkatan efisiensi.

Ketiga tujuan ini berkaitan dengan tiga prinsip dasar dari konsep DSS, yaitu struktur masalah, dukungan keputusan, dan efektivitas keputusan.

II.1.5. Tahap-Tahap Pengambilan Keputusan

Dalam Jurnal Hilyah M. (2012 : 50), Menurut Herbert A. Simon (Kadarsyah Suryadi dan Ali Ramdhani, 2002, 15-16) model yang menggambarkan

proses pengambilan keputusan. Proses Pengambilan Keputusan melibatkan 4 tahapan, yaitu:

a. Tahap *Intelligence*

   Dalam tahap merupakan proses penelusuran dan pendeteksian dari lingkup problematika serta proses pengenalan masalah. Data masukan diperoleh, diproses, dan diuji dalam rangka mengidentifikasi masalah.

b. Tahap *Design*

   Dalam tahap ini merupakan proses menemukan, mengembangkan dan menganalisis alternatif tindakan yang bisa dilakukan.

c. Tahap *Choice*

   Dalam tahap ini dilakukan proses pemilihan diantara berbagai alternatif tindakan yang mungkin dijalankan.

d. Tahap *Implementation*

   Dalam tahap ini pengambil keputusan menjalankan rangkaian aksi pemecahan yang dipilih di tahap *choice.* Implementasi yang sukses ditandai dengan terjawabnya masalah yang dihadapi, sementara kegagalan ditandai dengan tetap adanya masalah yang sedang dicoba untuk diatasi.

II.1.6. Komponen Sistem Pendukung Keputusan

Dalam Jurnal Nila Susanti dan Sri Winiarti (2013 : 329) Sistem Pendukung Keputusan terdiri atas empat komponen utama, yaitu:

1. Subsistem manajemen data berfungsi sebagai memasukkan suatu *database* yang berisi data yang relevan untuk situasi dan dikelola oleh perangkat lunak

yang disebut yang disebut sistem manajemen *database* (DBMS). *Knowledge Base* berisi semua fakta, ide, hubungan dan interaksi suatu domain tertentu.

2. Subsistem manajemen basis pengetahuan bertugas untuk mendukung semua subsistem lain atau bertindak sebagai suatu komponen independen. Ia memberikan intelegensi untuk memperbesar pengetahuan pengambil keputusan.
3. Subsistem manajemen model merupakan paket perangkat lunak yang memasukkan model keuangan statistik, ilmu manajemen atau model kuantitatif lainnya yang memberikan kapabilitas analitik dan manajemen perangkat lunak yang tepat.
4. Subsistem antarmuka pengguna (dialog) untuk mengimplementasikan sistem kedalam program aplikasi sehinggga pengguna atau pemakai dapat berkomunikasi dengan sistem yang dirancang.

Dalam Jurnal Heny Pratiwi (2014 : 96-97), Menurut Sudiyantoro (2005), komponen dalam sistem pendukung keputusan meliputi 8 bagian yaitu :

1. Perangkat Keras

   Perangkat keras ini akan terhubung dengan komputer lain menggunakan sistem jaringan, sehingga memudahkan dalam pengambilan data pada organisasi tersebut.

2. Perangkat Lunak

   Perangkat lunak sistem pendukung keputusan sering disebut juga dengan *DSS Generator,* berisi modul-modul untuk database, model dan *dialogue management*.

3. Sumber Data

   Database sistem pendukung keputusan berisi data dan informasi yang diambil dari dari data organisasi, ekternal dan data manajer secara individu.

4. Sumber Model

   Model ini berisi kumpulan model matematika dan teknik analisis yang disimpan dalam program dan berkas yang berbeda-beda. Komponen dari model ini dapat di kombinasikan dengan perangkat lunak tertentu untuk mendukung sebuah keputusan yang akan diambil.

5. Sumber Daya Manusia

   Sistem pendukung keputusan dapat digunakan oleh para manajer dan staf khusus untuk membuat keputusan ini juga dapat dikembangkan oleh penggunanya sesuai dengan keperluan para pengguna tersebut.

6. Model Sisem Pendukung Keputusan

   Model merupakan komponen yang penting dalam SPK. Model memilii pengertian yang berarti memisahkan dari dunia nyata dengan melukiskan komponen utama dan menghubungkannya dengan sistem dan kejadian lainnnya.

7. Lembar Kerja Elektronik

   Lembar kerja elektronik memudahkan pengguna membuat model dengan cara mengisi data dan menghubungkannya sesuai dengan format yang telah disediakan. Pengguna dapat melakukan beberapa perubahan dan mengevaluasi secara tampilan grafik.

8. Sistem Pendukung Keputusan Kelompok

Merupakan suatu sistem berbasis komputer yang mendukung kelompok-kelompok orang yang terlibat dalam suatu tugas atau tujuan bersama dan menyediakan tampilan antarmuka pada sat lingkungan yang digunakan bersama.

II.I.7. Kriteria Sistem Pendukung Keputusan

Dalam Jurnal Dita Monita (2013 : 30) Sistem pendukung keputusan dirancang secara khusus untuk mendukung seseorang yang harus mengambil keputusan-keputusan tertentu [1]. Berikut ini beberapa criteria sistem pendukung keputusan, yaitu:

1. Interaktif

   Sistem pendukung keputusan memiliki *user interface* yang komunikatif sehingga pemakai dapat melakukan akses secara cepat ke data dan memperoleh informasiyang dibutuhkan.

2. Fleksibel

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

3. Data Kualitas

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

4. Prosedur Pakar

   Sistem pendukung keputusan mengandung suatu prosedur yang dirancang berdasarkan rumusan formal atau juga berupa prosedur kepakaran seseorang atau kelompok dalam menyelesaikan suatu bidang masalah dengan fenomena tertentu.

II.I.8. Struktur Keputusan dalam Sistem Pendukung Keputusan

Dalam Jurnal Heny Pratiwi (2014 : 97), Menurut Kusrini (2007), keputusan yang diambil untuk menyelesaikan suatu masalah dilihat dari stuktur masalahnya terbagi menjadi tiga yaitu :

1. Keputusan terstuktur (*Structured Decision*)

   Keputusan yang dilakukan secara berulang-ulang dan bersifat rutin. Prosedur pengambilan keputusan sangat jelas. Keputusan tersebut dilakukan pada manajemen tingkat bawah (operasional). Misalnya, keputusan pemesanan barang dan keputusan penagihan piutang.

2. Keputusan semi terstuktur (*Semi Structured Decision*)

   Keputusan yang memiliki dua sifat. Sebagian keputusan bisa ditangani oleh komputer sedangkan yang lain tetap harus dilakukan oleh pengambil keputusan. Produser dalam pengambilan keputusan tersebut secara garis besar sudah ada, tetapi beberapa hal yang masih memerlukan kebijakan dari pengambil keputusan. Prosedur dalam pengambilan keputusan tersebut secara garis besar sudah ada, tetapi beberapa hal yang masih memerlukan kebijakan dari pengambilan keputusan. Biasanya keputusan semacam ini diambil oleh

manajer tingkat menengah (taktikal). Contoh keputusan jenis ini adalah evaluasi kredit, penjadwalan produksi dan pengendalian persediaan.

3. Keputusan Tak Terstruktur (*Unstructured Decision*)

   Keputusan yang penanganannya rumit karena tidak selalu terjadi. Keputusan tersebut menurut pengalaman dan berbagai sumber yang bersifat eksternal. Keputusan tersebut umumnya terjadi pada manajemen tingkat atas (strategis). Contonya adalah keputusan untuk pengembangan teknologi baru, keputusan bergabung dengan perusahaan lain dan perekrutan eksekutif.

II.2. Metode *Multifactor Evaluation Process* (MFEP)

Menurut Jurnal Heny Pratiwi (2014 : 97), Menurut Render B and Stair (2002), MFEP adalah metode kuantitatif yang menggunakan "*weighting system*". Dalam pengambilan keputusan multi faktor, pengambil keputusan secara subjektif dan intuitif menimbang berbagai faktor yang mempunyai pengaruh penting terhadap alternatif pilihan mereka. Untuk keputusan yang berpengaruh secara strategis, lebih dianjurkan menggunakan sebuah pendekatan kuantitatif seperti MFEP. Dalam MFEP seluruh kriteria yang menjadi faktor penting dalam melakukan pertimbangan diberikan bobot (*weightting*) yang sesuai. Langkah yang sama juga dilakukan terhadap alternatif yang akan dipilih, kemudian dilakan evaluasi berkaitan dengan faktor pertimbangan tersebut. Metode MFEP menentukan bahwa alternatif dengan nilai tertinggi adalah solusi terbaik berdasarkan kriteria yang telah dipilih.

Penggunaan model MFEB dapat direalisasikan dengan contoh berikut : Steve Marcel, seorang lulusan sarjana bedang bisnis mencari beberapa lowongan pekerjaan. Setalah mendiskusikan gambaran pekerjaan yang akan dikerjakannya dengan penasehat didiknya dan departemen direktur pusat penempatan pegawai, Steve menyatakan ada tiga faktor yang terpenting baginya yaitu gaji, peluang karir yang lebih baik, dan lokasi tempat kerja. Steve sudah memutuskan bahwa peluang jenjang karir merupakan faktor yang terpenting baginya. Faktor tersebut diberinya nilai skala 0.6. Steve menempatkan gaji diurutan berikutnya dengan nilai skala 0.3. Terakhir, Steve memberikan nilai skala 0.1 untuk tempat kerja. Seperti masalah pada model MFEP yang lain, nilai skala jika dijumlahkan harus sama dengan satu (1). Nilai bobot untuk faktor dapat dilihat pada tabel II.1.

Pada saat itu, Steve merasa yakin bahwa ia diterima di perusahaan AA, perusahaan EDS. Ltd. Dan perusahaan PW. Inc. Untuk setiap perusahaan, steve menghitung rata-rata variasi faktor dari nilai skala 0 sampai 1. Untuk perusahaan AA, stave memberikan faktor gaji dengan nilai skala 0.7. Peluang jenjang karir dengan nilai skala 0.9 dan lokasi tempat kerja dengan nilai 0.6. Untuk perusahaan EDS,Ltd. Steve memberikan faktor gaji dengan nilai skala 0.8, peluang jenjang karir dengn skala 0.7 dan lokasi tempat kerja dengan skala 0.8. Untuk perusahaan PW.Inc, Steve memberikan nilai faktor gaji dengan nilai skala 0.9, peluang jenjang karir dengan nilai skala 0.6 dan lokasi tempat kerja dengan nilai skala 0.9. Hasilnya dapat dilihat pada tabel II.2.

Tabel II.1. Tabel Nilai Bobot Untuk Faktor

Faktor	Bobot Faktor
Gaji	0.3
Kenaikan Karir	0.6
Lokasi	0.1

Sumber : Heny Pratiwi Volume 5 No 2 (2014 : 98)

Tabel II.2. Tabel Nilai Bobot Untuk Faktor

Faktor	AA	EDS	PW
Gaji	0.7	0.8	0.9
Kenaikan Karir	0.9	0.7	0.5
Lokasi	0.6	0.8	0.9

Sumber : Heny Pratiwi Volume 5 No 2 (2014 : 98)

Tabel II.3. Tabel Nilai Evaluasi Perusahaan AA.Co

Faktor	Bobot	Evaluasi	Evaluasi Bobot
Gaji	0.3	0.7	0.21
Kenaikan Karir	0.6	0.9	0.54
Lokasi	0.1	0.6	0.06
Total	1		0.81

Sumber : Heny Pratiwi Volume 5 No 2 (2014 : 98)

Tabel II.4. Tabel Nilai Evaluasi Perusahaan EDS.Ltd

Faktor	Bobot	Evaluasi	Evaluasi Bobot
Gaji	0.3	0.8	0.24
Kenaikan Karir	0.6	0.7	0.42
Lokasi	0.1	0.8	0.08
Total	1		0.74

Sumber : Heny Pratiwi Volume 5 No 2 (2014 : 98)

Tabel II.5. Tabel Nilai Evaluasi Perusahaan PW.Inc

Faktor	Bobot	Evaluasi	Evaluasi Bobot
Gaji	0.3	0.9	0.27
Kenaikan Karir	0.6	0.6	0.36
Lokasi	0.1	0.9	0.09
Total	1		0.72

Sumber : Heny Pratiwi Volume 5 No 2 (2014 : 98)

Dari setiap perusahaan (seperti yang dilihat pada tabel II.3, II.4 dan II.5). Perusahaan AA mendapat total bobot faktor yang paling tinggi yaitu 0.81. Dengan menggunakan metode MFEP, Steve mengambil keputusan untuk bekerja di perusahaan AA.

Dari informasi yang diperoleh, Steve dapat menghitung total bobot evaluasi dari setiap kriteria pekerjaan. Setiap perusahaan menghasilkan nilai evaluasi dari tiga faktor dan bobot faktor dikalikan dengan nilai evaluasi dan dijumlahkan untuk memperoleh total hasil evaluasi. Nilai evaluasi dapat dilihat pada tabel II.3, II.4, dan II.5.

II.2.1. Konsep Dasar Penggunaan Metode MFEP

Menurut Jurnal Ahmad Khaidir (2014 : 150), Dibawah ini merupakan langkah-langkah proses perhitungan menggunakan metode MFEP, yaitu:

1. Menentukan faktor dan bobot faktor dimana total pembobotan harus sama dengan 1 (∑ pembobotan = 1), yaitu *factor weight*.
2. Mengisikan nilai untuk setiap faktor yang mempengaruhi dalam pengambilan keputusan dari data-data yang akan diproses, nilai yang dimasukkan dalam proses pengambilan keputusan merupakan

nilai objektif, yaitu sudah pasti yaitu factor evaluation yang nilaianya antara 0 -1.

3. Proses perhitungan weight evaluation yang merupakan proses perhitungan bobot antara factor weight dan factor evaluation dengan serta penjumlahan seluruh hasil weight evaluations untuk memperoleh total hasil evaluasi.

Penggunaan model MFEP dapat direalisasikan dengan contoh berikut :

$$WE = FW \times E$$

$$\sum WE = \sum (FW \times E)$$

Sumber : Ahmad Khaidir Volume 4 No 3 (2014 : 150)

Keterangan :

WE = Weighted Evaluation

FW = *Factor Weight*

E = Evaluation

$\sum$WE = Total Weighted Evaluation

II.3. Pengertian PHP

PHP merupakan bahasa skrip yang digunakan untuk membuat halaman web yang dinamis. PHP bersifat *open product*. Penggunaan dapat mengubah *source code* dan mendistribusikannya secara bebas serta diedarkan secara gratis . PHP bersifat *server scripting* yang dapat ditambhakan kedalam HTML, sehingga suatu halaman web tidak lagi bersifat statis, namun bersifat dinamis. Sifat *server-side* berarti pengerjaan skrip PHP akan dilakukan di sebuah *web server*, kemudian hasilnya akan dikirimkan ke *browser*. Salah satu *web server* yang paling umum

digunakan untuk PHP adalah Apache. PHP dapat dijalankan pada sistem operasi *Unix, Windows*, dan Mac OS X.

PHP Hypertext Preprocessor atau sering disebut PHP merupakan bahasa pemrograman berbasis server-side yang dapat melakukan parsing script php menjadi script web sehingga dari sisi client menghasilkan suatu tampilan yang menarik. PHP merupakan pengembangan dari FI atau Form Interface yang dibuat oleh Rasmus Lerdoff pada tahun 1995. (YM Kusuma Ardhana ; 2012 ; 88)

PHP (*PHP Hypertext Preprocessor)* adalah kode/skrip yang akan dieksekusi pada server side. Skrip PHP akan membuat suatu aplikasi dapat di-integrasikan ke dalam HTML, sehingga suatu halaman web tidak lagi bersifat statis, namun menjadi bersifat dinamis. Sifat server-side berarti pengerjaan skrip dilakukan di server, baru kemudian hasilnya dikirimkan ke browser. (Deni Sutaji ; 2012 ; 2)

II.3.1. Aturan PHP

Adapun aturan penulisan skrip PHP ada dua cara, yaitu:

1. *Embedded Script*

   Dengan cara meletakkan tag PHP diantara tag-tag HTML, contohnya:

   ```
   <html>
   <body>
   <?php echo ”Belajar”;?>
   </body>
   </html>
   ```

2. *Non Embedded Script*

Dengan cara ini, semua script HTML diletakkan didalam skrip PHP. Contohnya:

```
<?php
Echo ”<html>”;
Echo ”<body>”;
Echo ”Belajar PHP”;
Echo ”</body>”;
Echo ”</html>”;
?>
```
(Deni Sutaji ; 2012 ; 2)

II.4. Pengertian Database

Database merupakan sekumpulan data yang tersusun dengan aturan tertentu dalam bentuk tabel. Adapaun secara fungsi, database merupakan suatu tempat yang dipergunakan untuk menyimpan sekumpulan data dalam format tertentu. Saat ini terdapat puluhan jenis database yang secara aktif di kembangkan dan dipergunakan dalam aplikasi.

Dilihat dari lokasi penyimpanan data, database dibagi menjadi dua, yaitu database lokal dan database server. Contoh database yang termasuk dalam database lokal di antaranya SQLite, Dbase, Firebird, dan Paradox. Adapun contoh database server di antaranya adalah MySQL, Oracel, SQL Server, Interbase, dan PostgreSQL. (Wahana Komputer ; 2011 ; 3-4)

II.4.1. Konsep *Database*

Menurut Jurnal Edhy Susanta (2011 : 545-546) James Martin menyatakan bahwa suatu *database* mempunyai enam kriteria penting yang harus di penuhi, yaitu :

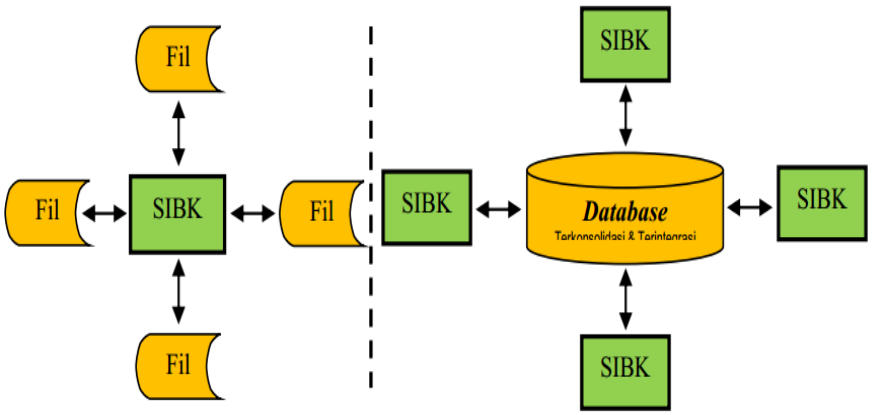
1. Berorientasi pada data (*data oriented*) dan bukan berorientasi pada program (*program oriented*) yang akan menggunakannya.
2. Dapat digunakan oleh banyak pengguna atau program aplikasi tanpa perlu mengubah *database.*
3. Dapat berkembang dengan mudah, baik volume maupun strukturnya.
4. Dapat memenuhi kebutuhan sistem baru secara mudah.
5. Dapat digunakan dengan cara yang berbeda-beda.
6. Kerangkapan data (*data redudancy*) dalam *database* minimal.

Sementara itu, Abraham Silberschatz, Henry F.Korth, dan S. Sudarshan menyatakan bahwa pengguna sistem file untuk penyimpanan data memiliki kelemahan yang terkait dengan enam permasalahan berikut :

1. Kerangkapan data dan inkonsistensi, yaitu format file yang berbeda-beda dan duplikasi data dalam file-file yang berbeda akan memerlukan penulisan program baru untuk memenuhi setiap tugas baru.
2. Isolasi data, dapat terjadi akibat penggunaan banyak file dengan format yang berbeda-beda.
3. Permasalahan integritas, dimana batasan-batasan integritas yang menjadi bagian kode program akan menimbulkan kesulitan saat penambahan atau perubahan batasan integritas.

4. Atomisitas pada proses update, bahwa kesalahan data dalam database dapat terjadi akibat status update yang tidak konsisten atau tidak lengkap.
5. Permasalahan akibat akses bersama oleh banyak pengguna, dimana akses bersama diperlukan untuk peningkatan kinerja namun akses bersamaan yang tidak terkontrol dapat menimbulkan inkonsistensi data.
6. Permasalahan keamanan data.

Menurut J. L. Whitten & L. D. Bentley, perancangan database untuk SIBK berbeda dengan perancangan database konvensional. Gambaran perbedaan antara file konvensional dan database ditunjukkan oleh Gambar 1.



Gambar II.1. File Konvensional Versus Database

(Sumber : Edhy Susanta Volume 5 No 2 ; 2011 : 545)

Penyusunan database dimaksudkan untuk untuk mengatasi permasalahan-permasalahan pada saat pengolahan data. Menurut CJ. Date, pengelolaan data dengan pendekatan database akan memberikan keuntungan berikut :

1. Kerangkapan data dapat diminimalkan. Jika file-file database dalam program aplikasi diciptakan oleh perancang yang berbeda dengan selang waktu lama, beberapa bagian data seringkali mengalami kerangkapan.
2. Inkonsistensi data dapat dihindari. Database yang bebas dari kerangkapan data akan terhindar dari munculnya data-data yang tidak konsisten.
3. Data dalam database dapat digunakan bersama (multiuser). Dalam rangka meningkatkan kinerja sistem dan untuk memperoleh waktu respon yang cepat, beberapa sistem mengijinkan banyak pengguna untuk dapat meng-update data secara simultan. Salah satu alasan penyusunan database adalah karena nantinya data tersebut akan digunakan oleh banyak pengguna atau program aplikasi yang berbeda secara bersama.
4. Standarisasi data dapat dilakukan. Definisi file database di dalam kamus data memungkinkan untuk menerapkan standarisasi data dalam database.
5. Pembatasan untuk keamanan data dapat diterapkan. Batasan akses data dalam database dapat diatur sehingga hanya pengguna tertentu yang mempunyai wewenang saja yang dapat mengaksesnya.
6. Integritas data dapat terpelihara. Integritas berhubungan dengan kinerja sistem agar dapat melakukan kendali/kontrol pada semua bagian sistem sehingga sistem selalu beroperasi dalam pengendalian penuh. Integritas data berhubungan dengan pengendalian sistem yang dirancang agar sistem tersebut dapat beroperasi sesuai batasan dan aturan yang ditetapkan.
7. Perbedaan kebutuhan data dapat diseimbangkan. Setiap pengguna dalam sistem memiliki kebutuhan yang berbeda-beda. Penyusunan database yang

benar akan menyeimbangkan perbedaan-perbedaan kebutuhan tersebut, karena pengguna akan menggunakan database yang sama.

II.5. Pengertian MySQL

MySQL *database server* adalah RDBMS (*Relasional Database Management Sistem*) yang dapat menangani data bervolume besar. Meskipun begitu, tidak menuntut *resource* yang besar. MySQL adalah *database* yang paling populer diantara *database* yang lain.

MySQL adalah program database yang mampu mengirim dan menerima data dengan sangat cepat dan multi user. MySQL memiliki dua bentuk lisensi, yaitu free software dan shareware. Penulis sendiri dalam menjelaskan buku ini menggunakan MySQL yang free software karena bebas menggunakan databse ini untuk keperluan probadi atau usaha tanpa harus membeli atau membayar lisensi yang berada di bawah lisensi GNU/GPL (General Public License), yang dapat di download pada alamat resminya *http://www.mysql.com*. MySQL sudah cukup lama dikembangkan, beberapa *fase* penting dalam pengembangan MySQL adalah sebagai berikut:

- MySQL dirilis pertama kali secara secara internal pada 23 Mei 1995
- Versi *windows* dirilis pada 8 Januari 1998 untuk *windows* 95 dan *windows* NT
- Versi 3.23 : beta dari Juni 2000, dan dirilis pada Januari 2001.
- Versi 4.0 : beta dari Agustus 2002, dan dirilis pada Maret 2003 (unions)

(Wahana Komputer ; 2010 ; 5)

II.6. *Unified Modelling Languag* (UML)

UML singkatan dari Unified Modeling Language yang berarti bahasa pe,odelan standar. Chonoles mengatakan sebagai bahasa, berarti UML memiliki sintaks dan semantic. Ketika kita membuat model menggunakan konsep UML ada aturan-aturan yang harus diikuti. Bagaimana elemen pada model-model yang kita buat berhubungan satu dengan lainnya harus mengikuti standar yang ada. UML bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya. (Prabowo et al; 2011 : 6)

UML diaplikasikan untuk maksud tertentu, biasanya antara lain:

1. Merancang perangkat lunak
2. Sarana komunikasi antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasi sistem yang ada, proses-proses dan organisasinya.

Blok pembangunan utama UML adalah diagram. Beberapa diagram ada yang rinci (jenis *timing diagram*) dan lainnya ada yang bersifat umum (misalnya diagram kelas). Para pengembangan sistem berorientasi objek menggunakan bahasa model untuk menggambar, membangun dan mendokumentasikan sistem yang mereka rancang. UML memungkinkan para anggota team untuk bekerja sama dengan bahasa model yang sama dengan mengaplikasikan beragam sistem. Intinya, UML merupakan alat komunikasi yang konsisten dalam mensupport para pengembangan sistem saat ini. (Prabowo Pudjo Widodo, Herlawati; 2011 : 6-7)

II.6.1. Diagram-Diagram UML

Beberapa leteratur menyebutkan bahwa UML menyediakan Sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa yang digabung, misalnya diagram komunikasi, diagram urutan, dan diagram pewaktuan digabung menjadi diagram interaksi. Namun demikian model-model itu dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis. Jenis diagram itu antara lain:

1. Diagram Kelas, bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodela sistem berorientasi objek . Meskipun bersifat statis, sering pula diagram kelas mmemuat kelas-kelas aktif.
2. Diagram paket (Package Diagram), bersifat statis. Diagram ini memperlihatkan kumpulan kelas-kelas, merupakan bagian dari diagram komponen.
3. Diagram *use case*, bersifat statis. Diagram ini memperlihatkan himpunan *use case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna.
4. Diagram interaksi dan *sequence* (urutan), bersifat dinamis. Diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.

5. Diagram komunikasi (*Communication Diagram*), bersifat dinamis. Diagram sebagai pengganti diagram kolaborasi UML 14 yang menekankan organisasi structural dari objek-objek yang menerima seta mengirim pesan.
6. Diagram *Statechart* (*Statechart Diagram*), bersifat dinamis. Diagram status memperlihatkan keadaan-keadaan pada sistem, memuat status (*state*), transisi, kejadian serta aktifitas. Diagram ini teruama penting untuk memperlihatkan sifat dinamis dari antarmuka (*interface*), kelas, kolaborasi, dan terutama penting pada pemodelan sistem-sisten yang reaktif.
7. Diagram aktivitas (*Aktivity Diagram*), bersifat dinamis. Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi sistem dan member tekanan pada aliran kendali antar objek.
8. Diagram komponen (*Component Diagram*), bersifat statis. Diagram komponen ini memeperlihatkan organisasi serta keberuntungan sistem / perangkat lunak pada komponen-komponen yang telah ada sebelumnya. Diagram ini berhubungan dengan diagram kelas dimana komponen secara tipikal dipetakan ke dalam satu atau lebih kelas-kelas, antaarmuka-antramuka serta kolaborasi-kolaborasi.
9. Diagram *Deployment (Deployment Diagram)*, bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (*run-time*). Memuat simpul-simpul beserta komponen-komponen yang ada di dalamnya. Diagram *deployment* berhubungan erat dengan diagram komponen dimana diagram ini

memuat satu atau lebih komponen-komponen. Diagram ini sangat berguna saat aplikasi kita berlaku sebagai aplikasi yang dijalankan pada banyak mesin (*Distributed Computing*)

Kesembilan diagram ini tidak mutlak harus digunakan dalam pengembangan perangkat lunak, semuanya dibuat sesuai dengan kebutuhan. Pada UML dimungkinkan kita menggunakan diagram-diagram lainnya (misalnya *Data Flow Diagram, Entity Relationship Diagram* dan sebagainya). (Prabowo Pudjo Widodo, Herlawati; 2011 : 10-12)

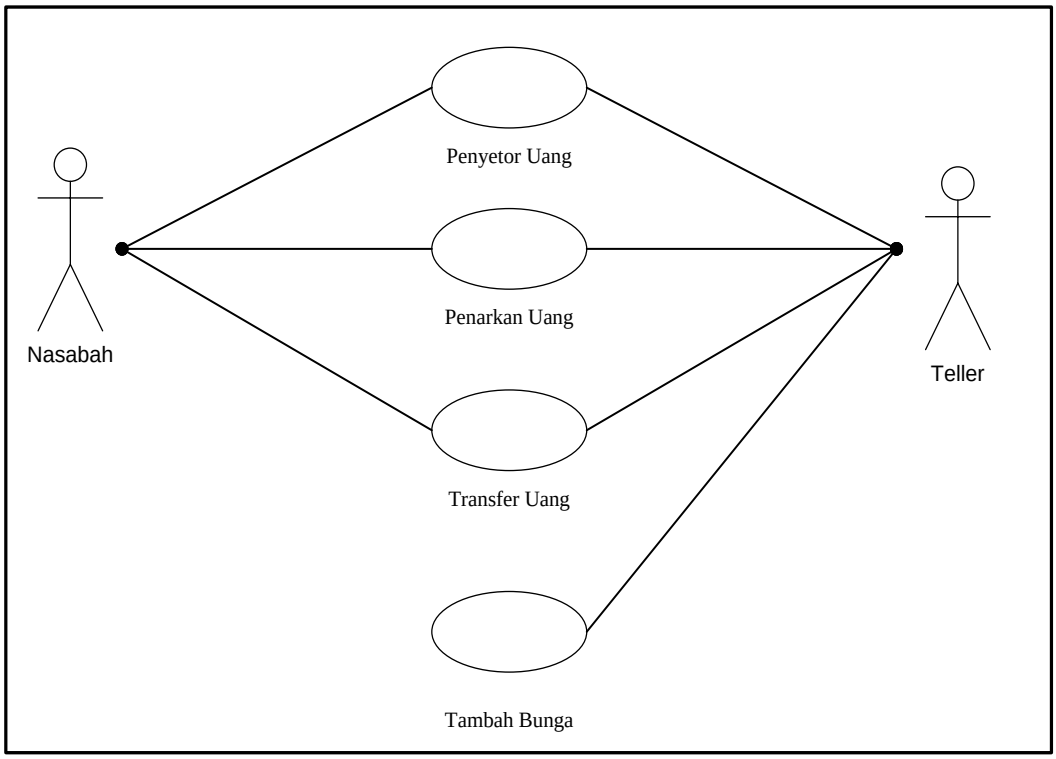
a. Diagram *Use Case* (*Use Case Diagram*)

Use Case menggambarkan *external view* dari sistem yang akan kita buat modelnya. MenurutPooley (2005:15) mengatakan bahwa model *use case* dapat dijabarkan dalam diagram, tetapi yang perlu diingat, diagram tidak identik dengan model karena model lebih luas dari diagram.

Komponen pembentuk diagram *use case* adalah:

1) Aktor (*actor*), menggambarkan pihak-pihak yang berperan dalam sistem.
2) *Use Case*, aktifitas/sarana yang disiapkan oleh bisnis / sistem.
3) Hubungan (*Link*), actor mana saja yang terlibat dalam *use case* ini.

Gambar dibawah ini merupakan salah satu contoh bentuk diagram *use case* yaitu :

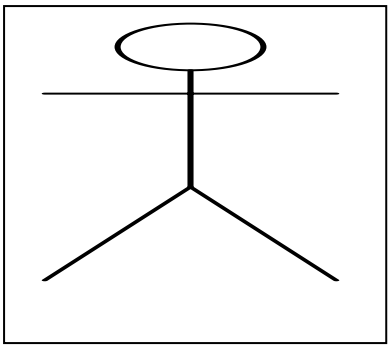


Gambar II.2. Diagram *Use Case*

(Sumber : Prabowo Pudjo Widodo dan Herlawati ;2011 : 17)

a) Aktor

Menurut Chonoles (2003 : 17) menyarankan sebelum membuat *use case* dan menentukan aktornya, agar mengidentifikasikan siapa saja pihak yang terlibat dalam sistem kita. Pihak yang terlibat biasanya dinamakan *stakeholder*.

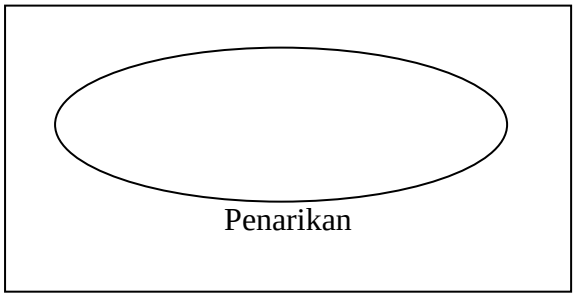


Gambar II.3. Aktor

(Sumber : Prabowo Pudjo Widodo dan Herlawati ; 2011 : 17)

b) Use Case

Menurut Pilone (2005 : 21) use case menggambarkan fungsi tertentu dalam suatu sistem berupa komponen, kejadian atau kelas. Sedangkan menurut Whitten (2004 : 258) mengartikan use case sebagai urutan langkah-langkah yang secara tindakan saling terkait (scenario), baik terotomatisasi maupun secara manual, untuk tujuan melengkapi suatu tugas bisnis tunggal. Use case digambarkan dalam bentuk ellips/oval pada gambar II.3 sebagai berikut :



Gambar II.4. Simbol Use Case

(Sumber : Prabowo Pudjo Widodo dan Herlawati ; 2011 : 22)

Use case sangat menentukan karakteristik sistem yang kita buat, oleh karena itu (Chonoles, 2003: 22) menawarkan cara untuk mengahasilkan use case yang baik, yakni:

1) Pilihlah nama yang baik

   Use case adalah sebuah *behavior* (perilaku), jadi seharusnya dalam frase kata kerja. Untuk membuat namanya lebih detil, tambahkan kata benda yang mengidentifikasikan dampak aksinya terhadap suatu kelas objek. Oleh karena itu diagram *use case* seharusnya berhubungan dengan diagram kelas.

2) Ilustrasikan perilaku dengan lengkap

   Use case dimulai dari inisiasi oleh aktor primer dan berakhir pada actor dan menghasilkan tujuan. Jangan membuat *use case* kecuali Anda mengetahui tujuannya. Sebagai contoh, memilih jenis tempat tidur (*king size*, *queen size* atau dobel) saat tamu memesan tidak dapat dijadikan use case karena merupakan bagian dari use case pemesanan kamar dan tidak dapat berdiri sendiri (tidak mungkin tamu memesan kamar tidur jenis king tapi tidak memesan kamar hotel).

3) Identifikasi perilaku dengan lengkap

   Untuk mencapai tujuan dan menghasilkan nilai tertentu dari aktor, use case harus lengkap. Ketika member nama pada *use case*, plihlah frasa kata kerja yang implikasinya hingga selesai. Misalnya gunakan frasa *reserve a room* (pemesanan kamar) dan jangan *reserving a room* (memesan kamar) karena memesan kamar menggambarkan perilaku yang belum selesai.

4) Menyediakan *Use case* lawan (inverse)

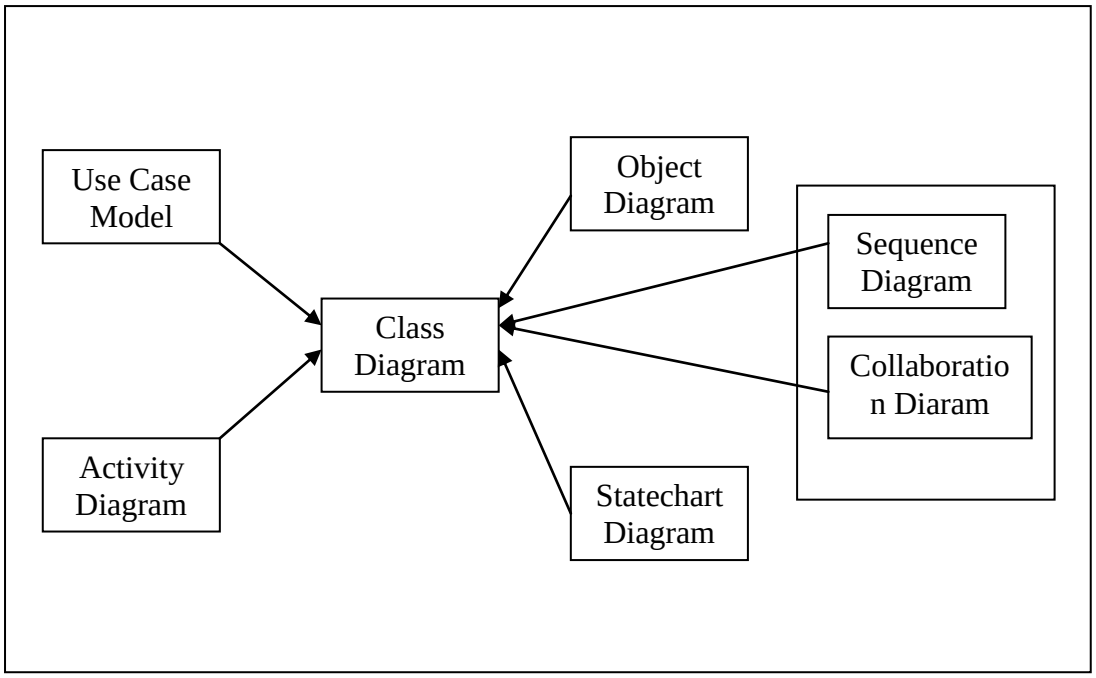
Kita biasanya membutuhkan *use case* yang membatalkan tujuan misalnya pada *use case* pemesanan kamar, dibutuhkan pada *use case* pembatalan pesanan kamar.

5) Batasi Use case hingga satu perilaku saja

Kadang kita cenderung membuat *use case* yang menghasilkan lebih dari satu tujuan aktivitas. Guna menghindari kerancuan, jagalah *use case* kita hanya fokus pada satu hal. Misalnya, pengguna *use case* menghasilkan ketidakfokusan, karena memiliki dua perilaku yang berbeda. Untuk menyediakan penjelasan detail terhadap segala kemungkinan yang terjadi pada *use case*, apa yang terjadi dan apa respon sistem.

b. Diagram kelas (Class Diagram)

Diagram kelas adalah inti dasri proses pemodelan objek, baik forward engineering maupun reverse engineering memanfaatkan diagram ini. Forward engineering adalah proses perubahan model menjadi kode program dsedangkan reserve engineering sebaliknya merubah kode program menjadi model.



Gambar II.5. Hubungan diagram kelas dengan diagram UML lainnya

(Sumber : Prabowo Pudjo Widodo dan Herlawati ; 2011 : 38)

c. Diagram Aktivitas (Activity Diagram)

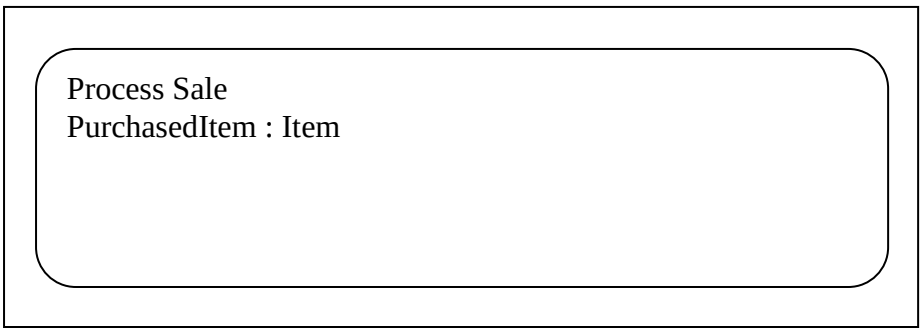
Diagram aktivitas lebih memfokuskan dari pada eksekusi dan alur sistem dari pada bagaimana sistem itu dirakit. Diagram ini tidak hanya memodelkan model bisnis juaga. Diagram aktivitas menunjukkan aktivitas sistem dalam bentuk kumpulan aksi-aksi. Ketika digunakan dalam pemodelan software, diagram aktivitas mempresentasikan pemanggilan suatu fungsi tertentu misalnya *call*. Sedangkan bila digunakan dalam pemodelan bisnis, diagram ini menggambarkan aktivitas yang dipicu oleh kejadian-kejadian di luar seperti pemesanan atau kejadian-kejadian internal misalnya proses penggajian tiap jumat sore.

Aktivitas merupakan kumpulan aksi-aksi. Aksi-aksi melakukan langkah sekali saja tidak boleh dipecah menjadi beberapa langkah lagi. Contoh akasi yaitu:

1) Fungsi matematika
2) Pemanggilan perilaku
3) Pemrosesan data

Ketika kita menggunakan diagram aktivitas untuk memodelkan perilaku suatu classifier, classifier dikatakan konteks dari aktivitas. Aktivitas dapat mengakses atribut dan operasi classifier, tiap objek yang terhubung dan parameter-parameter jika aktivitas memiliki hubungan dengan perilaku. Ketika digunakan untuk model proses bisnis, informasi itu biasanya disebut process-relevant data. Aktivitas diharapkan dapat digunakan ulang dalam suatu aplikasi, sedangkan aksi biasanya specific dan digunakan hanya untuk aktivitas tertentu.

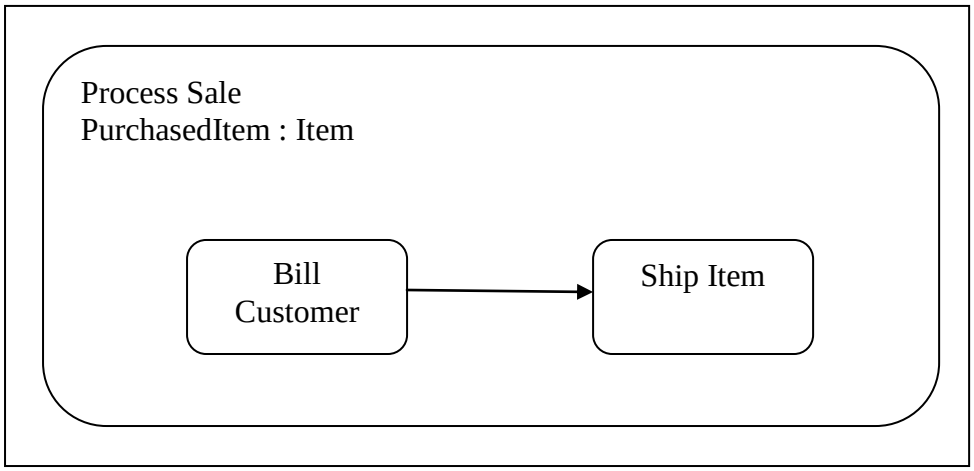
Aktivitas digambarkan dengan persegi panjang tumpu, namanya ditulis di kiri atas. Parameter yang terlibat dalam aktivitas ditulis dibawahnya.



Gambar II.6. Aktivitas sederhana tanpa rincian

(Sumber : Prabowo Pudjo Widodo dan Herlawati ; 2011 : 145)

Detail aktivitas dapat dimasukkan di dalam kotak. Akan diperlihatkan dengan symbol yang sama dengan aktivitas dan namanya diletakkan didalam persegi panjang.

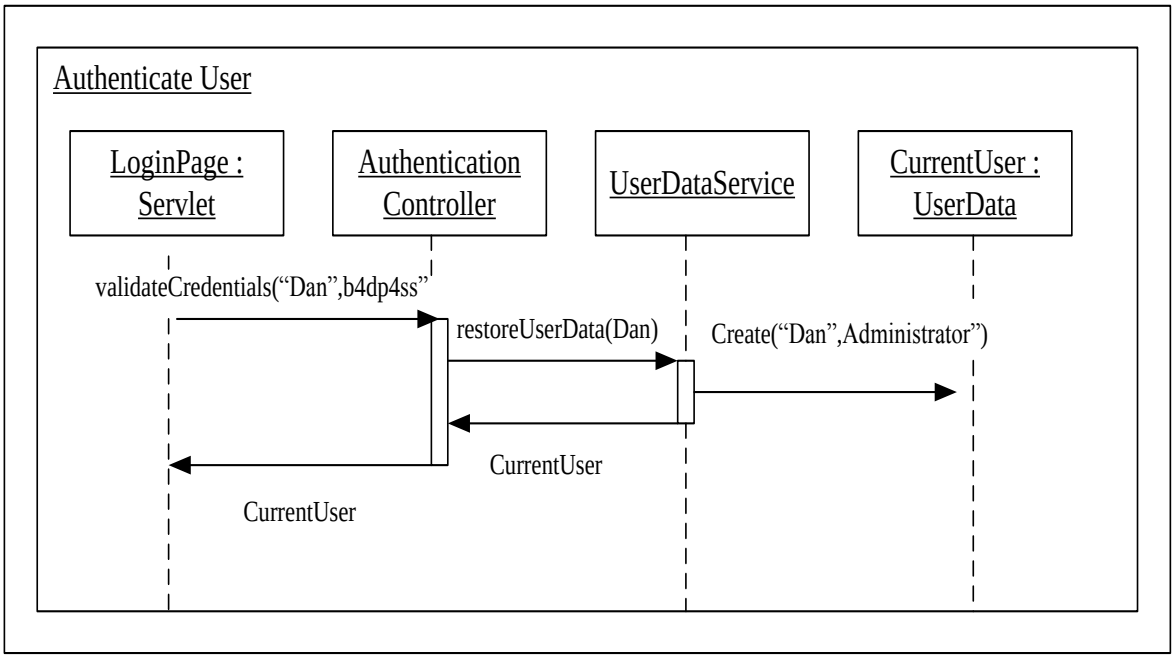


Gambar II.7. Aktivitas dengan detail sederhana

(Sumber :Prabowo Pudjo Widodo dan Herlawati ; 2011 : 145)

d. Sequence Diagram

Menurut (Douglas, 2004 : 174) menyebutkan ada tiga diagram primer UML dalam memodelkan scenario interaksi, yaitu: diagram urutan (*sequence diagram),* diagram waktu (*timing diagram*) dan diagram komunikasi (*communication diagram*). Namun demikian (Pilone, 2005 : 174) menyatakan bahwa diagram yang paling banyak dipakai adalah diagram urutan. Gambar II.6. memperlihatkan contoh diagram urutan dengan notasi-notasinya yang akan dijelaskan nanti.



Gambar II.8. Contoh diagram urutan

(Sumber : Prabowo Pudjo Widodo dan Herlawati ; 2011 : 175)

II.7. ERD (*Entity Relationship Diagram*)

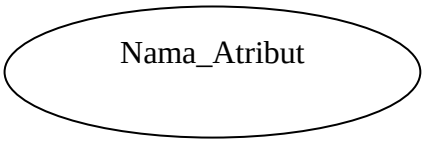
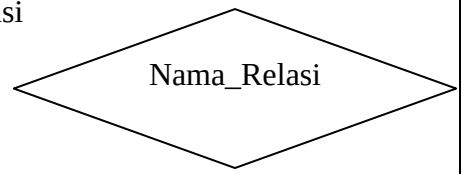
ERD (*Entity Relationship Diagram*) data model didasarkan pada persepsi terhadap dunia nyata yang tersusun atas kumpulan objek-objek dasar yang disebut entitas dan hubungan antar objek. Entitas adalah sesuatu atau objek dalam dunia nyata yang dapat dibedakan dari objek lain.

Sebagai tambahan, model ER menyajikan pula batasan dimana isi basis data harus menyesuaikan dengan batasan. Salah satu batasan yang penting adalah pemetaan kardinalitas (*mapping cardinalities*), yang menggambarkan jumlah entitas yang berhubungan dengan entitas lain melalui suatu relasi. (Janner Si ; 2010 : 60-61)

II.7.1. Simbol-simbol ERD (*Entity Relationship Diagram*)

Adapun symbol-simbol ERD (*Entity Relationship Diagram*) ditunjukkan pada table II.6.

Tabel II.6. Simbol-simbol ERD (*Entity Relationship Diagram*)

Simbol	Deskripsi
Entitas / *entity* 	Entitas merupakan data inti yang akan disimpan; bakal table pada basis data
Atribut 	Fiel atau koilom data yang perlu disimpan dalam suatu entitas
Relasi 	Relasi yang menghubungkan antara entitas; relasi biasanya diawali dengan kata kerja
Asosiasi / *Association* **1** ——— **0..***	Penghubung antar relasi dan entitas dimana di kedua ujungnya memiliki *multiplicity* kemungkinan jumlah pemakaian

(Sumber : Janner Si ; 2010 : 59-60)

II.8. Normalisai

Normalisasi adalah bagian perancangan basisdata. Tanpa Normalisasi, sistem basisdata menjadi tidak akurat, lambat, tidak efisien, serta tidak memberikan data yang diharapkan.

Pada waktu menormalisasi basisdata, ada empat tujuan yang harus dicapai, yaitu:

1. Mengatur data dalam kelompok-kelompok sehingga masing-masing kelompok hanya menangani bagian kecil sistem.
2. Meminimal jumlah data berulang dalam basisdata.
3. Membuat basisdata yang datanya diakses dan manipulasi secara cepat dan efisien tanpa melupakan integritas data.
4. Mengatur data sedemikian rupa sehingga ketika memodifikasi data, hanya mengubah satu tempat.

Tujuan Normalisai adalah membuat kumpulan table relasional yang bebas dari sata berulang dan dapat dimodifikasi secara benar dan konsisten. Ini berarti bahwa semua table pada basisdata relasional harus berada pada bentuk normal ketiga (3NF). Sebuah table relasional berada pada 3NF jika dan hanya jika semua kolom bukan kunci adalah saling independen berarti bahwa tidak ada kolom bukan kunci yang tergantung pada sembarang kombinasi kolom lainnya. Dua bentuk normal pertama aadalah langkah antara untuk mencapai tujuan, yaitu mempunyai semua table dalam 3NF. (Janner Si; 2010 : 77-78)

Tahapan normalisai terdiri dari beberapa bentuk yaitu sebagai berikut:

1. Bentuk normal pertama (1NF / *First Normal Form*)

Bentuk normal pertama memiliki ciri yaitu data berbentuk *flat file* (file datar), *record* disusun sesuai kedatangan, masih mungkin terjadi penyimpangan data (*anomaly* data). *Anomali* data dapat berupa *insert anomaly*, *delete anomaly*, *update anomaly* dan *redundancy* data (data duplikat). (Janner Si; 2010 : 79)

2. Bentuk Normal Kedua (2NF/ *Second Normal Form*)

Bentuk normal kedua memiliki ciri yaitu tidak terjadi *anomali* data, setiap *field / attribute* bukan kunci harus tergantung fungsi (*functional dependency*) terhadap *field / attribute* kunci, masih mungkin terjadi *transitive dependency* (*field* bukan kunci tergantung pada *field* bukan kunci dalam satu table). (Janner Si; 2010 : 81)

3. Bentuk Normal Ketiga (3NF / *Third Normal Form*)

Bentuk normal ketiga memiliki syarat yaitu table harus tidak terdapat transitive dependency (*field* bukan kunci tergantung pada *field* bukan kunci dalam satu tabel). (Janner Si; 2010 : 82)

4. Bentuk Normal *Boyce Codd* (BCNF / *Boyce Codd Normal Form*)

Pada tahap ini menghilangkan ketergantungan *field* bukan kunci secara persial (bagian) kunci dalam satu tabel. Apabila pada normal ketiga tidak lagi ditemukan *field* bukan kunci tergantung secara persial dalam satu tabel, maka normal ketiga juga merupakan bentuk BCNF. (Janner Si; 2010 : 84)

5. Bentuk Normal Kelima (5NF / *Five Normal Form*)

Empat bentuk normal pertama berdasarkan pada konsep ketergantungan fungsional, sedangkan bentuk normal kelima berdasarkan pada konsep

ketergantungan gabungan *(join dependence*). Ketergantungan gabungan berarti bahwa sebuah tabel., setelah didekomposisi menjadi tiga atau lebih tabel yang lebih kecil, harus dapat digabungkan kembali untuk membentuk tabel asal. Dengan kata lain, 5NF menunjukkan ketika sebuah tabel tidak dapat didekomposisi lagi. (Janner Si ; 2010 : 85)

II.9. *Dreamweaver*

Dreamweaver MX (MX 6, MX 7, MX 2004 dan MX 8) adalah suatu bentuk program editor web yang dibuat oleh Macromedia dengan alamat Web site www.macromedia.com. Dengan menggunakan program ini, seorang programmer web dapat dengan mudah membuat dan mendesain webnya, karena bersifat WYSIWYG (*What You Seee Is What You Get*). (Bunafit Nugroho ; 2009 : 1)

Dreamweaver MX dan 8 selain sebagai editor yang komplet juga dapat digunakan untuk membuat animasi sederhana yang berbentuk layer dengan bantuan JavaScript yang didukungnya. Dengan adanya program ini kita tidak akan susah-susah untuk mengetik skrip-skrip format HTML, PHP, JSP, ASP, JavaScript, CSS maupun bentuk program yang lainnya.

Sebagai editor, *Dreamweaver* mempunyai sifat yang WYSIWYG Dibaca (wai-si-wig) yang artinya apa yang kita lihat pada halaman desain, maka semuanya itu akan Kita peroleh pada *browser*. Dengan kelebihan ini sehingga seorang *programmer* (pembuat program) atau *desainer* (pembuat desain web) dapat langsung melihat hasil buatannya tanpa harus membukanya pada *browser*

(aplikasi pengakses web seperti Internet Explorer, Mozila, dll). (Bunafit Nugroho ; 2009 : 2)