**BAB II**

**LANDASAN TEORI**

* 1. **Sistem Pendukung Keputusan**
     1. **Pengertian sistem**

Gerald et al dalam Jogiyanto HM, (1999 : 1) mendefinisikan: “Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu”

Demikian pula didefinisikan oleh penulis lain “Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai tujuan tertentu”(Davis et al dalam JogiyantoHM,1999 :2).

Kedua definisi di atas sama benarnya dan tidak saling bertentangan. Yang berbeda hanyalah cara pendekatan yang dilakukan pada sistem. Karena pada hakekatnya setiap komponen sistem, untuk dapat saling berinteraksi dan untuk dapat mencapai tujuan tertentu harus melakukan sejumlah prosedur, metode, dan cara kerja yang juga saling berinteraksi.

JogiyantoHM dalam buku ”Analisis dan Disain”menguraikan beberapa karakteristik sistem sebagai berikut :

1. Komponen Sistem

Komponen-komponen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

1. Batas Sistem

Batas sistem (*boundary*) merupakan daerah yang membatasi antara suatu sistem dengan sistem lainnya atau dengan lingkungan luarnya.

1. Lingkungan Luar Sistem

Lingkungan luar sistem (*environment*)adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem.

1. Penghubung Sistem

Penghubung (*interface*) merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lainnya.

1. Masukan Sistem

Masukan (*input*)adalah energi yang dimasukkan kedalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*)dan masukan sinyal (*signal input*).

1. Keluaran Sistem

Keluaran sistem (*output*) adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

1. Pengolahan Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran.

1. Sasaran Sistem

Suatu sistem memiliki (*goal*) atau sasaran (*objective*). Kalau suatu sitem tidak mempunyai sasaran, maka operasi tidak akan ada gunanya.

* + 1. **Pengertian Keputusan**

Jenis keputusan menurut Simon dalam buku “*Management Information Systems*” edisi 10 terbagi menjadi 2 yaitu :

1. Keputusan terprogram (*programmed decision*) bersifat repetitif dan rutin, dalam hal prosedur tertentu digunakan untuk menanganinya sehingga keputusan tersebut tidak perlakukan *de novo* (baru) setiap kali terjadi.
2. Keputusan yang tidak terprogram (*nonprogrammed decision*) bersifat baru, tidak terstruktur dan jarang konsekuen. Tidak ada metode yang pasti untuk menangani masalah ini.

Simon menjelaskan konsep keputusan terprogram dan tak terprogram penting karena masing-masing memerlukan teknik yang berbeda.

Tahap-tahap pengambilan keputusan menurut Simon :

1. Kegiatan intelegen, yaitu mengamati lingkungan mencari kondisi-kondisi yang perlu diperbaiki.
2. Kegiatan merancang, yaitu menemukan, mengembangkan, dan menganalisis berbagai alternatif tindakan yang mungkin.
3. Kegiatan memilih, yaitu meilih suatu rangkaian tindakan tertentu dari beberapa yang tersedia.
4. Kegiatan menelaah, yaitu menilai pilihan-pilihan yang lain.
   * 1. **Sistem pendukung keputusan (SPK)**

Jogiyanto HM (2005 : 327) mendefinisikan: “Suatu sistem pendukung keputusan (SPK) atau *decision support sytems* (DSS) didefinisikan sebagai suatu sistem informasi untuk membantu manajer level menengah untuk proses pengambilan keputusan setengah terstruktur (*semi structured*) supaya lebih efektif dengan menggunakan model-model analitis dan data yang tersedia”.

Demikian pula didefinisikan oleh penulis lain “Sistem pendukung keputusan merupakan pasangan dari intelektual sumber daya manusia dengan kemampuan dari komputer untuk memperbaiki kualitas dari keputusan, yaitu sistem pendukung keputusan yang terkomputerisasi bagi pembuat keputusan manajemen yang menghadapi masalah semi struktur” (Efraim dkk, dalam buku “Berbagai makalah Sistem Informasi dalam KNSI 2009).

Dari definisi tersebut dapat disimpulkan bahwa sistem pendukung keputusan adalah suatu sistem yang utamanya berbasis komputer untuk membantu para pengambil keputusan untuk memecahkan masalah baik yang bersifat semi terstruktur maupun yang tidak terstruktur melalui suatu model.

Sistem Pendukung Keputusan merupakan suatu penerapan sistem informasi yang ditujukan untuk membantu para pimpinan dalam mengambil keputusan. Hal yang terpenting dari pengertian ini adalah sistem pendukung keputusan merupakan alat pelengkap bagi mereka yang terlibat dalam pengambilan keputusan.

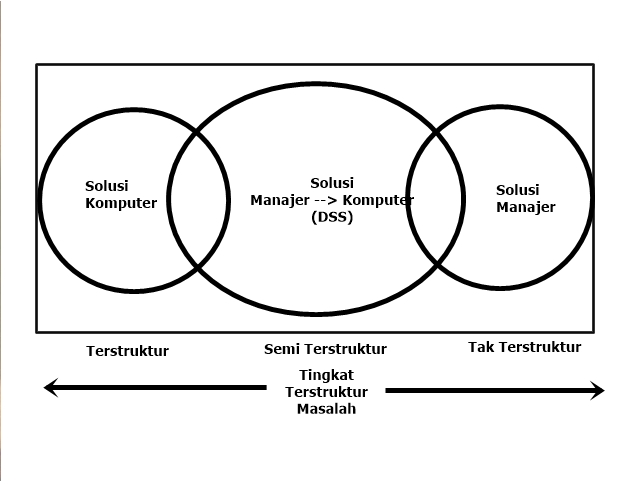
Konsep Sistem Pendukung Keputusan (DSS) dimulai pada akhir tahun 1960-an dengan *timesharing* komputer. Untuk pertama kalinya seseorang dapat berinteraksi langsung dengan komputer tanpa harus melalui spesialis informasi. Baru pada tahun 1971, istilah DSS diciptakan oleh G. Anthony Gorry dan Michael S. Scott Morton, keduanya professor MIT. Mereka merasa perlunya suatu kerangka kerja untuk mengarahkan aplikasi komputer kepada pengambilan keputusan manajemen dan mengembangkan apa yang telah dikenal sebagai Garry & Scott Morton Grid. Matrik (*Grid*) ini didasarkan pada konsep Simon mengenai keputusan terprogram dan tak terprogram serta tingkat-tingkat manajemen Robert N. Anthony.

* + 1. **Tujuan sistem pendukung keputusan**

Dari definisi di atas, maka dapat diketahui tujuan dari SPK adalah sebagai berikutini :

1. Membantu manajer mengambil keputusan setengah terstruktur yang dihadapi oleh manajer level menengah.
2. Membantu atau mendukung manajemen mengambil keputusan bukan menggantikannya.
3. Meningkatkan efektifitas pengambilan keputusan manajemen.

Tujuan dari SPK ini konsisten dengan yang diutarakan oleh Keen dan Morton (1978). Peter G. Kreen dan Scott Morton adalah pioneer DSS dari MITmenggambarkan sebagai berikut ini. (Jogiyanto. HM, 2005 : 328)



**Gambar 2.1.** SPK fokus pada solusi permasalahan *semistructured*

Gambar 2.1 menggambarkan hubungan antara struktur masalah dengan tingkat dukungan yang dapat disediakan oleh komputer. Komputer dapat diterapkan pada bagian masalah yang terstruktur, tetapi manager bertanggung jawab atas bagian yang tak terstruktur menerapkan penilaian atau intuisi dan melakukan analisis. Manajer dan komputer bekerja sama sebagai tim pemecahan masalah yang berada di area semi-terstruktur yang luas.

**2.1.5. *Fuzzy Multiple Attribute Decision Making* (FMADM)**

*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 subyektifitas dari pengambil keputusan. (Kusumadewi, 2006).

Ada beberapa metode yang dapat digunakan untuk mnyelesaikan masalah FMADM. antara lain (Kusumadewi, 2006) :

a. *Simple Additive Weighting Method* (SAW)

b. *Weighted Product* (WP)

c. *ELECTRE*

d. *Technique for Order Preference by Similarity to Ideal Solution* (TOPSIS)

e. *Analytic Hierarchy Process* (AHP)

* + 1. **Algoritma FMADM**

Algoritma FMADM adalah :

1. Memberikan nilai setiap alternatif (Ai) pada setiap kriteria (Cj) yang sudah ditentukan, dimana nilai tersebut diperoleh berdasarkan nilai crisp; i=1,2,…m dan j=1,2,…n.
2. Memberikan nilai bobot (W) yang juga didapatkan berdasarkan nilai crisp.
3. Melakukan normalisasi matriks dengan cara menghitung nilai rating kinerja ternormalisasi (rij) dari alternatif Ai pada atribut Cj berdasarkan persamaan yang disesuaikan dengan jenis atribut (atribut keuntungan/benefit=MAKSIMUM atau atribut biaya/cost=MINIMUM). Apabila berupa artibut keuntungan maka nilai crisp (Xij) dari setiap kolom atribut dibagi dengan nilai crisp MAX (MAX Xij) dari tiap kolom, sedangkan untuk atribut biaya, nilai crisp MIN (MIN Xij) dari tiap kolom atribut dibagi dengan nilai crisp (Xij) setiap kolom.
4. Melakukan proses perankingan dengan cara mengalikan matriks ternormalisasi (R) dengan nilai bobot (W).
5. Menentukan nilai preferensi untuk setiap alternatif (Vi) dengan cara menjumlahkan hasil kali antara matriks ternormalisasi (R) dengan nilai bobot (W). Nilai Vi yang lebih besar mengindikasikan bahwa alternatif Ai lebih terpilih. ( Kusumadewi , 2006).

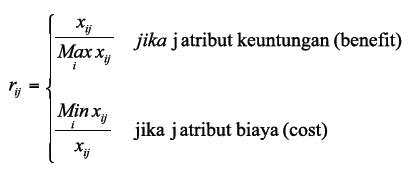
**2.1.7. Langkah Penyelesaian**

Dalam penelitian ini menggunakan FMADM metode SAW. Adapun langkah-langkahnya adalah:

1. Menentukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan, yaitu Ci.
2. Menentukan rating kecocokan setiap alternatif pada setiap kriteria.
3. Membuat matriks keputusan berdasarkan kriteria (Ci), kemudian melakukan normalisasi matriks berdasarkan persamaan yang disesuaikan dengan jenis atribut sehingga diperoleh matriks ternormalisasiR.
4. Hasil akhir diperoleh dari proses perankingan yaitu penjumlahan dari perkalian matriks ternormalisasi Rdengan vektor bobot sehingga diperoleh nilai terbesar yang dipilih sebagai alternatif terbaik (Ai) sebagai solusi.

**2.1.8. Metode *Simple Additive Weighting* (SAW)**

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



**Gambar 2.2** Formula Normalisasi metode SAW

Dimana : rij : Rating kinerja ternormalisasi

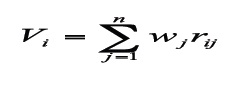
Maxi : Nilai maksimum dari setiap baris dan kolom

Mini : Nilai minimum dari setiap baris dan kolom.

Xij : Baris dan kolom dari matriks

rij adalah rating kinerja ternormalisasi dari alternatif Ai pada atribut Cj; i=1,2,...,m dan j=1,2,...,n.

Nilai preferensi untuk setiap alternative (Vi) diberikan rumus sebagai berikut:



**Gambar 2.3** Rumus Nilai Preferensi

Dimana : Vi : Nilai akhir dari alternative

Wi : Bobot yang telah ditentukan

rij : Normalisasi matriks

Nilai Vi yang lebih besar mengindikasikan bahwa alternatif Ai lebih terpilih.

Sedangkan untuk kriterianya terbagi dalam dua kategori yaitu untuk bernilai positif termasuk dalam  kriteria keuntungan dan yang bernilai negatif termasuk dalam kriteria biaya.

Keterangan :

A : Alternatif

C : Kriteria

W : Bobot Preferensi

V : Nilai preferensi untuk setiap alternatif

X : Nilai Alternatif dari setiap kriteria

* + 1. **Tahapan Metode SAW**

1. Menentukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan, yaitu Ci.
2. Menentukan  rating  kecocokan  setiap  alternatif   pada  setiap  kriteria.
3. Membuat  matriks  keputusan  berdasarkan  kriteria  (Ci),  kemudian melakukan  normalisasi  matriks  berdasarkan persamaan  yang disesuaikan  dengan  jenis  atribut sehingga  diperoleh  matriks ternormalisasi  R.
4. Hasil  akhir  diperoleh  dari  proses  perankingan  yaitu  penjumlahan  dari perkalian  matriks  ternormalisasi  R  dengan vector  bobot  sehingga diperoleh  nilai  terbesar  yang  dipilih sebagai  alternatif  terbaik  (Ai) sebagai  solusi.
   1. **Raskin**

Raskin merupakan subsidi pangan yang diperuntukkan bagi keluarga miskin sebagai upaya dari pemerintah untuk meningkatkan ketahanan pangan dan memberikan perlindungan pada keluarga miskin. Pendistribusian beras ini diharapkan mampu menjangkau keluarga miskin dengan mencukupi kebutuhan pangan beras melalui pendistribusian beras bersubsidi sebanyak 180Kg/RTS/Tahun dimana masing-masing keluarga akan menerima beras minimal 15 Kg/KK tiap bulan dengan harga bersih Rp 1.600/kg di titik-titik distribusi. Keberhasilan Program Raskin diukur berdasarkan tingkat pencapaian indikator 6T, yaitu: tepat sasaran, tepat jumlah, tepat harga, tepat waktu, tepat kualitas, dan tepat administrasi. bertujuan untuk mengurangi beban pengeluaran Rumah Tangga Sasaran (RTS) melalui pemenuhan sebagian kebutuhan pangan pokok dalam bentuk beras dan mencegah penurunan konsumsi energi dan protein. Selain itu raskin bertujuan untuk meningkatkan/membuka akses pangan keluarga melalui penjualan beras kepada keluarga penerima manfaat dengan jumlah yang telah ditentukan. Selain itu juga untuk meningkatkan akses masyarakat miskin dalam pemenuhan kebutuhan pangan pokok, sebagai salah satu hak dasarnya.

Peraturan perundangan yang menjadi landasan pelaksanaan program RASKIN adalah:

1. Undang-Undang No. 8 Tahun 1985 tentang Organisasi Masyarakat.

2. Undang-Undang No. 7 Tahun 1996, tentang Pangan.

3. Undang-Undang No. 19 Tahun 2003, tentang Badan Usaha Milik Negara (BUMN).

4. Undang-Undang No. 32 Tahun 2004, tentang Pemerintahan Daerah.

5. Undang-Undang No. 22 Tahun 2011, tentang Anggaran Pendapatan

dan Belanja Negara (APBN)TahunAnggaran 2012.

6. Undang-Undang No. 18 Tahun 1986 tentang Pelaksanaan Undang-Undang No. 8 Tahun 1985.

7. Peraturan Pemerintah No. 68 Tahun 2002, tentang Ketahanan Pangan.

8. Peraturan Pemerintah No. 7 Tahun 2003, tentang Pendirian Perusahaan Umum (Perum) BULOG.

9. Peraturan Presiden RI No. 15 Tahun 2010, tentang Percepatan Penanggulangan Kemiskinan.

10. Peraturan Presiden RI No 29 Tahun 2011, tentang Rencana Kerja Pemerintah Tahun 2012.

11. Inpres No. 7 tahun 2009, tentang Kebijakan Pemberasan

12. Kepmenko Kesra No. 35 Tahun 2008 tentang Tim Koordinasi Raskin Pusat.

**Tabel 2.1 Kriteria Penerima Beras Miskin**

|  |  |  |
| --- | --- | --- |
| *Kriteria* | *Sub Kriteria* | *Penilaian* |
| Tingkat Pendapatan/Penghasilan |  | 0,25 – 1 |
| Kondisi Pemukiman/Perumahan Tidak Layak | * **Atap**   Seng  Rumbia  Genteng   * **Dinding**   Batu  Kayu  Bambu   * **Lantai**   Semen  Tanah  Keramik | 0,25  0,17  0,16 |
| Faktor Kesehatan Tidak Menunjang | * **Lumpuh**   Tidak bisa berjalan   * **Cacat**   Buta | 0,25  0.10 |
| Kondisi lingkungan/daerah tempat tinggal | * Tempat Tinggal yang Kurang Memadai * Kurangnya Tempat Umum/WC | 0,25  0,5 |

Sumber : Data Desa Moutong

* 1. **Sisklus Hidup Pengembangan Sistem**

Pengembangan sistem informasi yang berbasis komputer dapat merupakan tugas kompleks yang membutuhkan banyak sumber daya dan dapat memakan waktu berbulan-bulan bahkan bertahun tahun untuk menyelesaikannya. Proses pengembangan sistem melewati beberapa tahapan dari mulai sistem itu direncanakan sampai dengan sistem tersebut diterapkan, dioperasikan dan dipelihara. Bila operasi sistem yang sudah dikembangkan masih timbul kembali permasalahan-permasalahan yang kritis serta tidak dapat diatasi dalam tahap pemeliharaan sistem, maka perlu dikembangkan kembali suatu sistem untuk mengatasinya dan proses ini kembali ke tahap yang pertama, yaitu tahap perencanaan sistem. Siklus ini disebut dengan siklus hidup suatu sistem (*systems life cycle*). Daur atau siklus hidup dari pengembangan sistem merupakan suatu bentuk yang digunakan untuk menggambarkan tahapan utama dan langkah-langkah didalam tahapan tersebut dalam proses pengembangannya.

Siklus hidup pengembangan sistem dengan langkah-langkah utamanya yang akan digunakan adalah sebagai berikut : (Jogiyanto HM, 1999 : 52)

**Kebijakan dan perencanaan sistem**

**Analisis sistem**

**Desain (perancangan) sistem secara umum**

**Seleksi sistem**

**Implementasi ( penerapan) sistem**

**Perawatan sistem**

Awal proyek sistem

Pengembangan sistem

Manajemen sistem

**Gambar 2.4.**Siklus Hidup Pengembangan Sistem

* + 1. **Analisa sistem**

Jogiyanto HM (1999 : 129) mendefinisikan analisa sistem sebagai berikut:”Analisa sistem (*systems analysis*) sebagai penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya”

Tahap analisis merupakan tahap yang kritis dan sangat penting, karena kesalahan di dalam tahap ini akan menyebabkan juga kesalahan ditahap selanjutnya. Tahap analisa sistem mencakup studi kelayakan dan analisis kebutuhan.

Di dalam tahap analisis sistem terdapat langkah-langkah dasar yang harus dilakukan oleh analisis sistem, yaitu sebagai berikut :

1. *Identify,* yaitu mengidentifikasi masalah.

Mengidentifikasi (mengenal) masalah merupakan langkah pertama yang dilakukan dalam tahap analisis sistem. Masalah (*Problems*) dapat didefinisikan sebagai suatu pertanyaan yang diinginkan untuk dipecahkan.

1. *Understand,* yaitu memahami kerja dari sistem yang ada.

Langkah kedua dari tahap analisis sistem adalah memahami kerja dari sistem yang ada. Langkah ini dapat dilakukan dengan mempelajari secara terinci bagaimana sistem yang ada beroperasi. Untuk mempelajari operasi dari sistem ini diperlukan data yang dapat diperoleh dengan cara melakukan penelitian.

1. *Analyze,* yaitu menganalisis sistem tanpa *report*

Langkah ini dilakukan berdasarkan data yang telah diperoleh dari hasil penelitian yang telah dilakukan

1. *Report,* yaitu membuat laporan hasil analisis

Tujuan utama dari pembuatan laporan hasil analisis :

1. Pelaporan bahwa analisis telah selesai dilakukan
2. Meluruskan kesalahan-pengertian mengenai apa yang telah ditemukkan dan dianalisis oleh analisis sistem tetapi tidak sesuai menurut manajemen.
   * 1. **Desain sistem**

Setelah tahap analisis sistem selesai dilakukan, maka analis sistem telah mendapatkan gambaran dengan jelas apa yang harus dikerjakan. Tiba waktunya sekarang bagi analis sistem untuk memikirkan bagaimana membentuk sistem tersebut. Tahap ini disebut dengan desain sistem (*system design*).

Menurut Robert J.Verzello dan John Reuter, dalam Jogiyanto HM (1999 : 196) desain sistem adalah tahap setelah analisis dari siklus pengembangan sistem; pendefinisian dari kebutuhan-kebutuhan fungsional dan persiapan untuk rancang bangun implementasi menggambarkan bagaimana suatu sistem dibentuk.

Demikian pula Menurut John Burch dan Gary Grudnitski, dalam Jogiyanto HM (1999 : 196) desain sistem dapat didefinisikan sebagai penggambaran, perencanaan dan

pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam satu kesatuan yang utuh dan berfungsi.

Tahap desain sistem mempunyai dua tujuan utama yaitu :

1. Untuk memenuhi kebutuhan kepada pemakai sistem.
2. Untuk memberikan gambaran yang jelas dan rancang bangun yang lengkap kepada pemrogram komputer dan ahli-ahli teknis lainnya.

Desain sistem dapat dibagi dalam dua bagian yaitu desain sistem secara umum (*general systems design*) dan desain sistem secara terinci (*detailed system design*).

1. Desain sistem secara umum(*General System Design*).

Tujuan dari desain sistem secara umum adalah untuk memberikan gambaran secara umum kepada *user* tentang sistem yang baru, yang mana merupakan persiapan dari desain sistem secara rinci. Desain secara umum dilakukan oleh analis sistem untuk mengidentifikasikan komponen-komponen sistem informasi yang akan didesain secara rinci oleh pemrogram komputer dan ahli teknik lainnya.

Pada tahap ini, komponen-komponen sistem informasi dirancang dengan tujuan untuk dikomunikasikan kepada *user*. Komponen sistem informasi yang didesain adalah model, *input*, *database*, *output*, teknologi dan kontrol.

Desain sistem secara rinci (*detailed system design*).

1. Desain *input* terinci

Masukan merupakan awal dimulainya proses informasi. Bahan mentah dari informasi adalah data yang terjadi dari transaksi-transaksi yang dilakukanoleh organisasi. Data hasil transaksi merupakan masukan untuk sistem informasi. Hasil dari sistem informasi tidak lepas dari data yang dimasukkan. Desain*input* terinci dimulai dari desain dokumen dasar sebagai penangkap *input* yang pertama kali. Jika dokumen dasar tidak didesain dengan baik, kemungkinan *input* yang tercatat dapat salah bahkan kurang.

Fungsi dokumen dasar dalam penanganan arus data :

* Dapat menunjukkan macam dari data yang harus dikumpulkan.
* Data dapat dicatat dengan jelas, konsisten dan akurat.
* Dapat mendorong lengkapnya data disebabkan data yang dibutuhkandisebutkan satu persatu di dalam dokumen dasarnya.

1. Desain *output*  terinci.

Desain *output* terinci dimaksudkan untuk mengetahui bagaimana dan seperti apa bentuk *output*-*output* dari sistem yang baru. Desain*Output* Terinci terbagi atas dua yaitu desain*output* berbentuk laporan dimedia kertas dan desain*output* dalam bentuk dialog pada layar terminal.

1. Desain *Output*  Dalam Bentuk laporan

Desain ini dimaksudkan untuk menghasilkan *output* dalam bentuk laporan dimedia kertas. Bentuk laporan yang paling banyak digunakan adalah dalam bentuk tabel dan berbentuk grafik atau bagan.

1. Desain *Output*  Dalam Bentuk Dialog Layar Terminal

Desain ini merupakan rancang bangun dari percakapan antara pemakai sistem (*user*) dengan komputer. Percakapan ini dapat terdiri dari proses memasukkan data ke sistem, menampilkan *output* informasi kepada *user* atau keduannya.

Beberapa strategi dalam membuat layar dialog terminal :

1. Dialog pertanyaan / jawaban.
2. Menu.

Menu banyak digunakan karena merupakan jalur pemakai yang mudah dipahami dan mudah digunakan. Menu berisi beberapa alternatif atau option atau pilihan yang disajikan kepada *user*. Pilihan menu akan lebih baik bila dikelompokkan sesuai fungsinya.

1. Desain *database* terinci.

Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan disimpanan luar komputer dan digunakan perangkat lunak tertentu untuk memanipulasinya. *Database* merupakan salah satu komponen yang penting disistem informasi karena berfungsi sebagai basis penyedia informasi bagi para pemakainya. Penerapan *database* dalam sistem informasi disebut *database system*.

Sistem basis data (*database system*) adalah suatu sistem informasi yang mengintegrasikan kumpulan dari data yang saling berhubungan satu dengan yang lainnya dan membuatnya tersedia untuk beberapa aplikasi yang bermacam-macam di dalam suatu organisasi. Dalam sistem basis data, tiap-tiap orang atau bagian dapat memandang *database* dari beberapa sudut pandang yang bebeda.

Pada tahap ini, desain*database* dimaksudkan untuk mendefinisikan isi atau struktur dari tiap-tiap file yang telah diidentifikasikan didesain secara umum.

1. Desain teknologi.

Tahap desain teknologi terbagi atas dua yaitu desain teknologi secara umum dan terinci. Pada tahap ini kita menentukan teknologi yang akan dipergunakan dalam menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan. Teknolgi yang dimaksud meliputi :

1. Perangkat Keras (*hardware*), yang terdiri dari alat masukan, alat pemroses, alat *output* dan simpanan luar.
2. Perangkat Lunak (*software*), yang terdiri dari perangkat lunak sistem operasi (*operating system*), perangkat lunak bahasa (*language software*) dan perangkat lunak (*application software*).
3. Sumber Daya Manusia (*brainware*), misalnya operator komputer, pemrogram, spesialis telekomunikasi, sistem analis dan sebagainya.

Desain teknologi sangat diperlukan pada tahap implementasi dan pengujian untuk membuktikan bahwa sistem dapat berjalan secara semestinya.

1. Desain model.

Tahap desain model terbagi menjadi dua yaitu desain model secara umum dan terinci. Tahap desain model secara umum berupa desain sistem secara fisik dan logika. Desain fisik dapat digambarkan dengan bagan alir dokumen.

Desain secara logika digambarkan dengan diagram arus data (DAD). Pada tahap desain model terinci, model akan mendefinisikan secara rinci urutan-urutan langkah dari masing-masing proses yang digambarkan di DAD. Urutan langkah proses ini diwakili oleh suatu program komputer.

**2.3.2.1 Perancangan konseptual**

Perancangan konseptual sering kali disebut dengan perancangan logis. Pada perancangan ini kebutuhan pemakai dan pemecahan masalah yang teridentifikasi selama tahap analisis sistem mulai dibuat untuk di implementasikan. Ada tiga langkah penting yang dilakukan dalam perancangan konseptual, yaitu evaluasi alternatif rancangan, penyiapan spesifikasi rancangan dan penyiapan laporan rancangan sistem secara konseptual.

Menurut Romney, Seinbart dan Cushing, 1997 dalam Abdul Kadir (2003 : 407) evaluasi yang dilakukan mengandung hal-hal berikut :

* Bagaimana alternatif-alternatif tersebut memenuhi sasaran sistem dan organisasi dengan baik?.
* Bagaimana alternatif-alternatif tersebut memenuhi kebutuhan pemakai dengan baik?.
* Apakah alternatif-alternatif tersebut layak secara ekonomi?.
* Apa saja keuntungan dan kerugian masing-masing?

Setelah alternatif rancangan dipilih, tahap selanjutnya adalah penyiapan spesifikasi rancangan yang elemen-elemen sebagai berikut :

* 1. Keluaran

Rancangan laporan mencakup frekuensi laporan (harian, mingguan, dsb), isi laporan, bentuk laporan dan laporan cukup ditampilkan pada layar atau perlu dicetak.

* 1. Penyimpan Data

Dalam hal ini, semua data yang diperlukan untuk membentuk laporan ditentukan lebih detail, termasuk ukuran data dan letaknya dalam berkas.

* 1. Masukan

Rancangan masukan meliputi data yang perlu dimasukkan kedalam sistem.

* 1. Prosedur Pemrosesan dan Operasi

Rancangan ini menjelaskan bagaimana data masukan diproses dan disimpan dalam rangka untuk menghasilkan laporan.

Langkah berikutnya adalah menyiapkan laporan rancangan sistem konseptual. Berdasarkan laporan inilah, perancangan sistem secara fisik dibuat.

**2.3.2.2** **Perancangan Fisik**

Pada perancangan ini, rancangan yang masih bersifat konsep diterjemahkan dalam bentuk fisik sehingga terbentuk spesifikasi lengkap tentang modul sistem dan antarmuka antar modul serta rancangan basis data secara fisik.

Beberapa hasil akhir setelah tahap perancangan fisik berakhir :

* Rancangan keluaran

Rancangan keluaran berupa bentuk laporan dan rancangan dokumen.

* Rancangan masukan

Rancangan masukan berupa rancangan layar untuk pemasukan data.

* Rancangan antarmuka pemakai dan sistem.

Rancangan ini berupa rancangan interaksi antar pemakai dan sistem, misalnya berupa menu, icon dan lain-lain.

* Rancangan *platform.*

Rancangan ini berupa rancangan yang menentukan *hardware*dan *software*yang akan digunakan.

* Rancangan basis data.

Rancangan ini berupa rancangan-rancangan berkas dalam basis data termasuk penentuan kapasitas masing-masing.

* Rancangan modul.

Rancangan ini berupa rancangan program yang dilengkapi dengan algoritma (cara modul / program kerja).

* Rancangan kontrol.

Rancangan ini berupa rancangan kontrol-kontrol yang digunakan dalam sistem seperti validasi, otorisasi dan audit data.

* Dokumentasi.

Berupa hasil dokumentasi hingga tahap perancangan fisik.

* Rencana pengujian.

Berupa rencana yang dipakai untuk menguji sistem.

* Rencana konversi.

Berupa rencana untuk menerapkan sistem baru terhadap sistem lama.

Bagan Alir sistem merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem.Bagan alir sistem digambarkan dengan simbol-simbol sebagai berikut :

|  |  |  |  |
| --- | --- | --- | --- |
| Tabel 2.2 Bagan Alir Sistem | | | |
| NO | **NAMA SIMBOL** | **SIMBOL** | KETERANGAN |
| 1. | Simbol Dokumen |  | Menunjukkan dokumen *input* dan *output* baik itu proses manual, mekanik, atau komputer |
| 2. | Simbol KegiatanManual |  | Menunjukan pekerjaan manual |
| 3. | Simbol Simpanan Offline |  | Menunjukkan file non-komputer yang diarsip urut angka (*numerical*), huruf (*alphabetical*), atau tanggal (*chronological*) |
| 4. | Simbol Kartu Plong |  | Menunjukkan *input* dan *output* yang menggunakan kartu plong (*punched card*). |
| 5. | Simbol Proses |  | Menunjukkankegiatan proses dari operasi program komputer |
| 6 | Simbol Operasi Luar |  | Menunjukkan operasi yang dilakukan di luar proses operasi komputer |
| 7. | Simbol Pengurutan Offline |  | Menunjukkan proses urut data di luar proses komputer. operasi luar, menunjukkan operasi yang dilakukan di luar proses operasi komputer |
| 8. | Simbol Pita Magnetik |  | Menunjukkan *input* dan *output* menggunakan pita *magnetic*. |
| 9. | Simbol Hard Disk |  | Menunjukkan *input* dan *output*  menggunakan *harddisk* |
| 10. | Simbol Diskette |  | Menunjukkan *input* dan *output* menggunakan *diskette* |
| NO | **NAMA SIMBOL** | **SIMBOL** | KETERANGAN |
| 11. | Simbol Drum Magnetik |  | Menunjukkan *input* dan *output* menggunakan drum magnetic |
| 12. | Simbol Pita Kertas Berlubang |  | Menunjukkan *input* dan *output* menggunakan pita kertas berlubang. |
| 13. | Simbol Keyboard |  | Menunjukkan *input* yang menggunakan *on-line keyboard* |
| 14. | Simbol Display |  | Menunjukkan *output* yang ditampilkan di monitor. |
| 15. | Simbol Pita Kontrol |  | Menunjukkan penggunaan pita kontrol (*control tape*) dalam *batch control* total untuk pencocokan di proses *batch processing*. |
| 16 | Simbol Hubungan Komunikasi |  | Menunjukkan proses transmisi data melalui *channel* komunikasi. |
| 17. | Simbol Garis Alir |  | Menunjukkan arus dari proses |
| 18. | Simbol Penjelasan |  | Menunjukkan penjelasan dari suatu proses |
| 19. | Simbol Penghubung |  | Menunjukkan penghubung ke halaman yang masih sama atau ke halaman yang lain |

Sumber : (Jogiyanto HM, 1999 : 796-799)

Untuk mempermudah penggambaran suatu sistem yang ada atau sistem yang baru yang akan dikembangkan secara logika tanpa memperhatikan lingkungan fisik di mana data tersebut mengalir atau lingkungan fisik di mana data tersebut akan disimpan, maka digunakan Diagram Arus Data (DAD) atau *Data Flow Diagram* (DFD). Dalam menggambarkan sistem perlu dilakukan pembentukan simbol, berikut ini simbol-simbol yang sering digunakan dalam DAD :

1. *External entity* (kesatuan luar) atau *boundary* (batas sistem).

Setiap sistem pasti mempunyai batas sistem *(boundary)* yang memisahkan suatu sistem dengan lingkungan luarnya. Sistem akan menerima *input* dan menghasilkan *output* kepada lingkungan luarnya. Kesatuan luar *(external entity)* merupakan kesatuan di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lain yang berada di lingkungan luarnya yang akan memberikan *input* serta menerima *output* dari sistem. (Jogiyanto HM, 1999 ;701)

**Gambar 2.5** Notasi kesatuan luar

1. *Data flow* (arus data).

Arus data ini menunjukkan arus atau aliran data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem. (Jogiyanto HM, 1999 ;702)

Nama Arus Data

**Gambar 2.6** Notasi arus data

1. *Process* (proses).

Suatu proses adalah kegiatan atau kerja yang dilakukan orang, mesin atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses. (Jogiyanto HM, 1999 ;705)

Identifikasi

Nama Proses

**Gambar 2.7** Notasi proses

1. *Data store* (simpanan data).

Simpanan data pada DFD dapat disimbolkan dengan sepasang garis horisontal paralel yang tertutup disalah satu ujungnya. (Jogiyanto HM, 1999 ;707)

Media Nama Data store

**Gambar 2.8** Notasi simpanan data

* + 1. **Implementasi Sistem**

Sistem telah dianalisa dan didesain secara rinci dan teknologi telah diseleksi dan dipilih. Tiba saatnya sekarang sistem untuk diimplementasikan (diterapkan). Tahap implementasi sistem merupakan tahap meletakkan sistem supaya siap untuk dioperasikan. Tahap implementasi sistem dapat terdiri dari langkah-langkah sebagai berikut:

1. Menerapkan rencana implementasi.

Rencana implementasi merupakan kegiatan awal dari tahap implementasi sistem. Rencana implementasi dimaksudkan terutama untuk mengatur biaya dan waktu yang dibutuhkan selama tahap implementasi.

1. Melakukan Kegiatan Implementasi

Kegiatan implementasi dilakukan dengan dasar kegiatan yang telah direncanakan dalam rencana implementasi. Kegiatan-kegiatan yang dapat dilakukan dalam tahap implementasi ini adalah sebagai berikut :

* 1. Persiapan tempat dan instalasi perangkat keras dan perangkat lunak.

Jika peralatan baru akan dimiliki, maka tempat atau ruangan untuk peralatan ini perlu dipersiapkan terlebih dahulu. Keamanan fisik dari tempat ini perlu juga dipertimbangkan. Sistem komputer yang besar membutuhkan tempat dengan lingkungan yang lebih harus diperhitungkan. Langkah selanjutnya setelah persiapan fisik tempat adalah menginstalasi perangkat keras yang sudah dikirim dan menginstalasi perangkat lunak yang sudah ada.

* 1. Pemrograman dan pengetesan sistem.

Pemrograman merupakan kegiatan menulis kode program yang akan dieksekusi oleh komputer. Kode program yang ditulis oleh pemrogram harus berdasarkan dokumentasi yang disediakan oleh analis sistem hasil dari desain sistem secara rinci. Sebelum program diterapkan, maka program harus terlebih dahulu bebas dari kesalahan-kesalahan. Oleh sebab itu, program harus diuji untuk menemukan kesalahan-kesalahan yang mungkin dapat terjadi. Program dites untuk tiap-tiap modul dan dilanjutkan dengan pengetesan untuk semua modul yang telah dirangkai.

* 1. Pengetesan sistem.

Pengetesan sistem biasanya dilakukan setelah pengetesan program. Pengetesan sistem dilakukan untuk memeriksa kekompakan antar komponen sistem yang diimplementasikan. Tujuan utama dari pengetesan sistem ini adalah untuk memastikan bahwa elemen-elemen atau komponen-komponen dari sistem telah berfungsi sesuai dengan yang diharapkan.

* + 1. **Operasi dan pemeliharaan**

Setelah masa sistem berjalan sepenuhnya menggantikan sistem lama, sistem memasuki pada tahapan operasi dan pemeliharaan. Zwass (1999) membagi pemeliharaan perangkat lunak menjadi tiga macam,yaitu :

1. Pemeliharaan perfektif.

Pemeliharaan Perfektifditujukan untuk memperbarui sistem lama sebagai tanggapan atas perubahan kebutuhan pemakai dan kebutuhan organisasi, meningkatkan efesiensi sistem, dan memperbaiki dokumentasi.

1. Pemeliharaan adaptif.

Pemeliharaan Adaptif berupa perubahan aplikasi untuk menyesuaikan diri terhadap lingkungan perangkat keras dan perangkat lunakbaru. Sebagai contoh pemeliharaan ini dapat berupa perubahan aplikasi dari *mainframe* ke lingkungan *client/server* atau mengkonversi dari sistem berbasis berkas ke lingkungan basis data.

1. Pemeliharaan korektif.

Pemeliharaan korektif berpa pembetulan atas kesalahan-kesalahan yang ditemukan pada saat sistem berjalan.

* 1. **Teknik Pengujian Sistem**
     1. ***White Box***

Pengujian perangkat lunak adalah elemen kritis dari jaminan kualitas perangkat lunak dan merepresentasikan kajian pokok dari spesifikasi, desain dan pengkodean.

Pengujian sistem / perangkat lunak memiliki sejumlah aturan yang berfungsi sebagai sasaran pengujian, diantaranya adalah sebagai berikut :

1. Pengujian adalah proses eksekusi suatu program dengan maksud menemukan kesalahan.
2. *Test case* yang baik adalah *test case* yang memiliki probabilitas tinggi untuk menemukan kesalahan yang belum pernah ditemukan sebelumnya.
3. Pengujian yang sukses adalah pengujian yang mengungkap semua kesalahan yang belum pernah ditemukan sebelumnya.

Pengujian *White Box*, adalah metode pengujian yang menggunakan struktur kontrol desain prosedur untuk memperoleh *test case*. Dengan menggunakan metode *white box*, perekayasa sistem dapat melakukan *test case* yang memberikan jaminan bahwa semua jalur independen pada suatu modul telah digunakan paling tidak satu kali, menggunakan semua keputusan logis pada sisi *true* dan *false*, mengeksekusi semua *loop* pada batasan mereka dan pada batas operasional mereka, dan menggunakan stuktur data internal untuk menjamin validitasnya. Pengujian *Basis Path* adalah teknik pengujian *white box* yang diusulkan pertama kali oleh Tom McCabe. Metode *basis path* ini memungkinkan desainer *test case* mengukur kompleksitas logis dari desain prosedural dan menggunakannya sebagai pedoman untuk menetapkan basis set dari jalur eksekusi. (Roger S. Pressman, 2002 : 536).

**9**

**10**

**6**

**7**

**8**

**1**

**2**

**3**

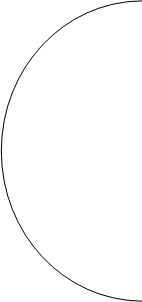
**5**

**4**

**11**

**Gambar 2.9** Bagan Alir

Bagan alir digunakan untuk menggambarkan struktur kontrol program. Dan untuk menggambarkan grafik alir, harus memperhatikan representasi desain prosedural pada bagan alir. Pada gambar dibawah ini, grafik alir memetakan bagan alir tersebut ke dalam grafik alir yang sesuai (dengan mengasumsikan bahwa tidak ada kondisi senyawa yang diisikan di dalam diamond keputusan dari bagan alir tersebut). Masing-masing lingkaran, yang disebut *simpul* grafik alir, merepresentasikan satu atau lebih statemen prosedural. Urutan kotak proses dan permata keputusan dapat memetakan simpul tunggal. Anak panah tersebut yang disebut *edges* atau *links*, merepresentasikan aliran kontrol dan analog dengan anak panah bagan alir. *Edge* harus berhenti pada suatu simpul, meskipun bila simpul tersebut tidak merepresentasikan statemen prosedural. (Roger S. Pressman, 2002 : 536).

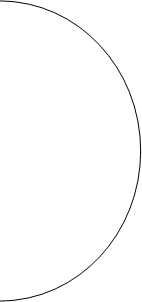


*Edge*

Simpul

Predikat

*Node*



Region

R1

R4

R2

R3

**Gambar 2.10** Grafik Alir

*Node* adalah lingkaran yang merepresentasikan satu atau lebih statemen prosedural.

*Edge* adalah anak panah pada grafik alir.

*Region* adalah area yang membatasi *edge* dan *node*

Simpul Predikat adalah simpul atau *node* yang berisi kondisi yang ditandai dengan dua atau lebih *edge* yang berasal darinya.

Darigambar *flowgraph* di atas didapat :

*Path* 1 =1– 11

*Path* 2 =1– 2 – 3 – 4 – 5 – 10– 1–11

*Path* 3 =1– 2 – 3 – 6 – 8 – 9 – 10– 1 – 11

*Path* 4 =1– 2 – 3 – 6 – 7 – 9–10–1–11

*Path* 1,2,3,4 yang telah didefinisikan diatas merupakan *basis set* untuk diagram alir.

*Cyclomatic complexity* digunakan untuk mencari jumlah *path* dalam satu *flowgraph*.dapat dipergunakan rumusan sebagai berikut :

1.Jumlah region grafikalir sesuaidengan*cyclomaticcomplexity*.

2.*Cyclomatixcomplexity*V(G) untuk grafik alir dihitung dengan rumus:

***V(G) =E– N +2*** …………………. (1)

Dimana:

E= jumlah *edge* pada grafikalir

N= jumlah *node* pada grafikalir

Jalur 3 : 1 – 2 – 3 – 6 – 8 – 9 – 10 – 1 – 11

Jalur 4 : 1 – 2 – 3 – 6 – 7 – 9 – 10 – 1 – 11

Jalur 1, 2, 3, dan 4 yang ditentukan di atas terdiri dari sebuah basis set untuk grafik alir pada gambar 2.10 Bagaimana kita tahu banyaknya jalur yang dicari? Komputasi kompleksitas siklomatis memberikan jawaban. Fondasi kompleksitas siklomatis adalah teori grafik, dan memberi kita metriks perangkat lunak yang sangat berguna. Kompleksitas dihitung dalam salah satu dari tiga cara berikut :

1. Jumlah region grafik alir sesuai dengan kompleksitas siklomatis.
2. Kompleksitas siklomatis, *V(G),* untuk grafik alir *G* ditentukan sebagai

*V(G)* = *E – N* + 2 di mana *E* adalah jumlah *edge* grafik alir dan *N* adalah jumlah simpul grafik alir.

1. Kompleksitas siklomatis, *V(G),* untuk grafik alir G juga ditentukan sebagai *V(G)* = P + 1, dimana P adalah jumlah simpul predikat yang diisikan dalam grafik alir *G*.

Pada gambar 2.10 grafik alir, kompleksitas siklomatis dapat dihitung dengan menggunakan masing-masing dari algoritma yang ditulis di atas :

1. Grafik alir mempunyai 4 region
2. *V(G)* = 11 *edge* – 9 simpul + 2 = 4
3. *V(G)* = 3 simpul yang diperkirakan + 1 =4

Dengan demikian, kompleksitas siklomatis dari grafik alir pada gambar 2.10 adalah 4. Yang lebih penting, nilai untuk *V(G)* memberi kita batas atas untuk jumlah jalur independen yang membentuk basis set, dan implikasinya, batas atas jumlah pengujian yang harus didesain dan dieksekusi untuk menjamin semua statemen program.

* + 1. ***Black Box***

Pengujian *Black-Box* berusaha menemukan kesalahan dalam kategori :

* Fungsi tidak benar atau hilang
* Kesalahan antar muka
* Kesalahan pada struktur data (pengaksesan basis data)
* Kesalahan inisialisasi dan akhir program
* Kesalahan performasi.

Pengujian ini berfokus pada persyaratan fungsional perangkat lunak dan merupakan komplemen dari pengujian *White-Box*. Hal tersebut dapat dicapai melalui :

* Pengujian *Graph-based* : dimulai dengan membuat grafik sekumpulan *node* yang mempresentasikan objek (misal *New File*, Layar baru dengan atributnya), link (hubungan antar objek), *node-weight* (misal nilai data tertentu seperti atribut layar, perilaku), dan link-weight (karakteristik suatu link, misal menu select).
* *Equivalence Partitioning* : membagi domain *input* untuk pengujian agar diperoleh kelas-kelas kesalahan (misal kelompok data karakter, atau atribut yang lain).
* Analisis Nilai Batas : pengujian berdasarkan nilai batas domain *input*.
* Pengujian Perbandingan : disebut juga pengujian *back-to-back* yang diterapkan pada pada suatu versi perangkat lunak atau perangkat lunak redundan untuk memastikan konsistensinya.
  1. ***Database Management System***

DBMS (*Database Management System*) adalah suatu perangkat lunak yang ditujukan untuk menangani penciptaan, pemeliharaan, dan pengendalian akses data. Dengan menggunakan perangkat lunak ini pengelolaan data menjadi mudah dilakukan. Selain itu perangkat lunak ini juga menyediakan berbagai piranti yang berguna. Misalnya piranti yang memudahkan dalam membuat berbagai bentuk laporan.

* + 1. **Pengertian *database***

*Database* (basis data) merupakan kumpulan data yang saling berhubungan. Hubungan antar data dapat ditunjukan dengan adanya *field*/ kolom kunci dari tiap file/tabel yang ada. Dalam satu file atau table terdapat *record-record* yang sejenis, sama besar, sama bentuk, yang merupakan satu kumpulan entitas yang seragam. Satu *record* (umumnya digambarkan sebagai baris data) terdiri dari field yangsaling berhubungan menunjukan bahwa field tersebut dalam satu pengertian yang lengkap dan disimpan dalam satu *record*.

* + 1. **E.R diagram.**

Model E-R adalah suatu model yang digunakan untuk menggambarkan data dalam bentuk entitas, atribut dan hubungan antar entitas.

**Tabel 2.3** Simbol‐Simbol ER‐Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| **NOTASI** | **KET** | **NOTASI** | **KET** |
|  | *Entity*  *Weak Entity*  *Relationship*  *Identifying*  *Relationship*  Atribut |  | Atribut*PrimaryKey*  Atribut  *Multivalue*  Atribut  *Composite*  Atribut  *Derivatif* |

***Entity*** adalah obyek yang dapat dibedakan dalam dunia nyata. *Entity* set adalah kumpulan dari *entity* yang sejenis, *entity* set dapat berupa :

1. Obyek secara fisik : Rumah, Kendaraan, Peralatan
2. Obyek secara konsep : Pekerjaan, Perusahaan, Rencana.

**Atribut** adalah karakteristik dari *entity* atau *relationship*, yang menyediakan penjelasan detail tentang *entity* atau *relationship* tersebut. Nilai Atribut merupakan suatu data actual atau informasi yang disimpan pada suatu atribut didalam suatu *entity* atau *relationship*. Jenis-jenis atribut :

a. *Key*: Atribut yang digunakan untuk menentukan suatu *entity* secara unik.

b. Atribut *Simple* : Atribut yang bernilai tunggal.

c. Atribut *Multivalue*: Atribut yang memiliki sekelompok nilai untuk setiap instan *entity*.

d. Atribut *Composite* : Suatu atribut yang terdiri dari beberapa atribut yang lebih kecil yang mempunyai arti tertentu.

e. Atribut *Derivatif* : Suatu atribut yang dihasilkan dari atribut yang lain.

***Relationship*** adalah hubungan yang terjadi antara satu atau lebih *entity*. *Relationship* set adalah kumpulan *relationship* yang sejenis, derajat dari *relationship* menjelaskan jumlah *entity* yang berpartisipasi dalam suatu *relationship* :

a. *Unary Degree* (Derajat Satu)

b. *Binary Degree* (DerajatDua)

c. *Ternary Degree* (Derajat Tiga)

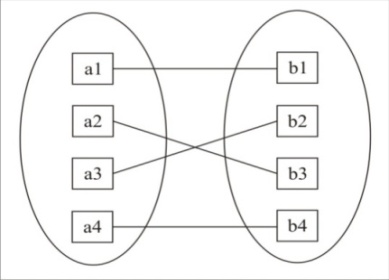
**Weak *Entity*** adalah suatu *Entity* dimana keberadaan dari *entity* tersebut tergantung dari keberadaan *entity* lain. *Entity* yang merupakan induknya disebut *Identifying Owner* dan *relationship*nya disebut *Identifying Relationship*. Weak *Entity* selalu mempunyai *Total Participation constraint*  dengan *Identifying Owner*.

* + 1. **Hubungan antar tabel**

Dalam perancangan Basis Data terdapat hubungan-hubungan yang terjadi antar tabel, hubungan-hubungan antar tabel tersebut adalah:

1. Hubungan *One to One*

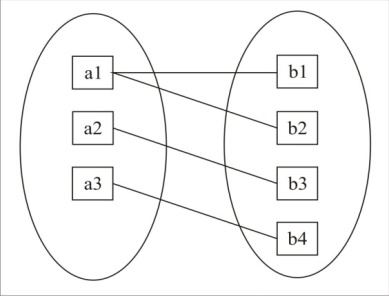
Hubungan *One to One* merupakan hubungan antara satu tabel induk yang dihubungkan dengan satu tabel anak yang lainnya, yang dihubungkan berdasarkan atribut kunci yang terdapat pada masing-masing tabel.



**Gambar 2.11** Contoh Hubungan *One to One*

1. Hubungan *One to Many*

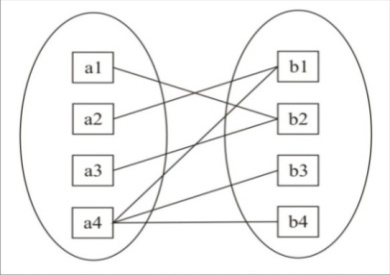
Hubungan *One to Many* merupakan hubungan dari satu tabel induk yang dihubungkan dengan banyak tabel anak lainnya, dimana hubungan yang terjadi berdasarkan atribut kunci yang ada pada tabel induk.



**Gambar 2.12** Contoh Hubungan *One to Many*

1. Hubungan *Many to Many*

Hubungan *Many to Many* merupakan hubungan keseluruhan yang berasal dari banyak tabel yang mempunyai hubungan dengan banyak tabel yang lainnya.



**Gambar 2.13** Contoh Hubungan *Many to Many*

* + 1. **Jenis *Key* (Kunci)**
    2. *Super Key*

*Super Key* adalah satu atribut atau kumpulan atribut yang secara unik mengindentifikasi sebuah tuple atau *record* di dalam relasi atau himpunan dari satu atau lebih entitas yang dapat digunakan untuk mengidentifikasikan secara unik sebuah entitas dalam entitas set.

* + 1. *Candidate Key*

*Candidate Key adalah* satu atribut atau satu set atribut yang mengidentifikasikan secara unik suatu kejadian spesifik dari *entity.* Jika satu*candidate key* berisi lebih dari satu atribut maka disebut sebagai *composite* key (kunci campuran atau kunci gabungan).

* + 1. *Primary Key*

*Primary key* adalah suatu atribut atau satu set atribut yang tidak hanya mengidentifikasikan secara unik suatu kejadian spesifik, tetapi juga dapat mewakili setiap kejadian dari suatu *entity.*

* + 1. *Alternate Key*

*Alternate Key* adalah *candidate key* yang tidak dipakai sebagai *primary key*. *Alternate key* sering dipakai sebagai kunci pengurutan dalam laporan.

* + 1. *Foreign Key*

*Foreign Key* adalah satu atributyang melengkapi satu *relationship* yang menunjukkan ke induknya.

* 1. **Perangkat Lunak Pendukung**

Perangkat lunak pendukung yang digunakan penulis dalam membangun sistem ini ada beberapa diantaranya *Microsoft Visual Basic* 6.0 digunakan untuk membangun program, dan MySQL Server yang digunakan sebagai basis data.

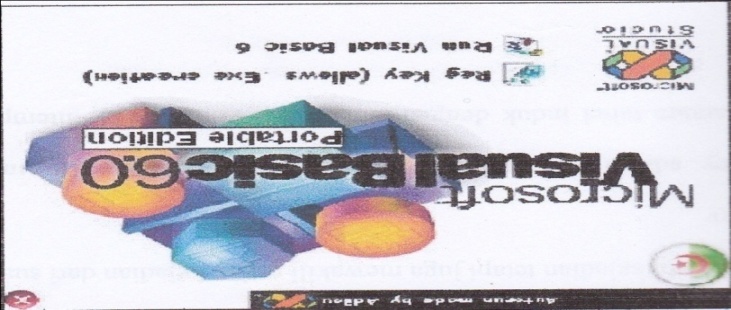
* + 1. **Microsoft Visual Basic 6.0**

Microsoft Visual Basic 6.0 adalah merupakan sebuah bahasa pemrograman komputer yang menjadi sasaran (*tools*) untuk menghasilkan program-program aplikasi yang berbasis windows.

Sarana akses data yang lebih cepat dan handal untuk membuat aplikasi basis data yang berkemampuan tinggi. Tambahan control-control baru lebih canggih serta peningkatan stuktur bahasa visual basic. Visual Basic memiliki beberapa versi atau edisi yang disesuaikan dengan kebutuhan pemakainya.

Visual basic mempunyai banyak kelebihan dibandingkan software atau bahasa pemograman lainnya. Dibawah ini adalah beberapa keuntungan tersebut:

1. Menghilangkan kompleksitas pemanggilan fungsi *windows* API (*Application Programming Interface*), karena banyak fungsi-fungsi tersebut sudah di “embedded” (Sistem Embedded adalah sistem yang dirancang khusus untuk tujuan tertentu demi meningkatkan fungsi suatu mesin) ke dalam syntax visual basic.
2. Cocok digunakan untuk mengembangkan aplikasi/program yang bersifat Rapid Aplication Development (adalah sebuah model proses perkembangan software sekuensial linear yang menekankan siklus perkembangan yang sangat pendek)
3. Juga cocok digunakan untuk membuat program/aplikasi bisnis, dan sebagainya.



**Gambar 2.14** *Interface* (antarmuka) *Microsoft Visual Basic 6.0*

**2. 6. 2MySQL *Server***

MySQL adalah sebuah [perangkat lunak](http://id.wikipedia.org/wiki/Perangkat_lunak) sistem manajemen [basis data](http://id.wikipedia.org/wiki/Basis_data) [SQL](http://id.wikipedia.org/wiki/SQL) (*Structure Query Language*). MySQL adalah sebuah implementasi dari sistem manajemen basisdata relasional ([RDBMS](http://id.wikipedia.org/wiki/RDBMS)) yang didistribusikan secara gratis dibawah [lisensi GPL](http://id.wikipedia.org/wiki/GPL) (*General Public License*). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya; [SQL](http://id.wikipedia.org/wiki/SQL) (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Kehandalan suatu sistem basis data ([DBMS](http://id.wikipedia.org/wiki/DBMS)) dapat diketahui dari cara kerja pengoptimasi-nya dalam melakukan proses perintah-perintah SQL yang dibuat oleh pengguna maupun program-program aplikasi yang memanfaatkannya.

Sebagai teladan basis data, MySQL mendukung operasi basis data transaksional maupun operasi basis data non-transaksional.

(<http://id.wikipedia.org/wiki/MySQL>)

MySQL memiliki beberapa keistimewaan, antara lain :

1. Portabilitas. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti *Windows*, *Linux, FreeBSD, Mac Os X Server, Solaris, Amiga*, dan masih banyak lagi.
2. Perangkat lunak sumber terbuka. MySQL didistribusikan sebagai [perangkat lunak sumber terbuka](http://id.wikipedia.org/wiki/Perangkat_lunak_sumber_terbuka), dibawah lisensi [GPL](http://id.wikipedia.org/wiki/GPL) sehingga dapat digunakan secara gratis
3. Multi-user. MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. *Performance tuning*, MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
5. Ragam tipe data. MySQL memiliki ragam tipe data yang sangat kaya, seperti *signed/ unsigned integer, float, double, char, text, date, timestamp*, dan lain-lain.
6. Perintah dan Fungsi. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (*query*).
7. Keamanan. MySQL memiliki beberapa lapisan keamanan seperti level [*subnetmask*](http://id.wikipedia.org/w/index.php?title=Subnetmask&action=edit&redlink=1), nama [*host*](http://id.wikipedia.org/w/index.php?title=Host&action=edit&redlink=1), dan izin akses user dengan sistem perizinan yang mendetail serta sandi terenkripsi.
8. Skalabilitas dan Pembatasan. MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman ([*records*](http://id.wikipedia.org/w/index.php?title=Records&action=edit&redlink=1)) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
9. Konektivitas. MySQL dapat melakukan koneksi dengan klien menggunakan protokol [TCP/IP](http://id.wikipedia.org/wiki/TCP/IP), [*Unix soket*](http://id.wikipedia.org/w/index.php?title=Unix_soket&action=edit&redlink=1) *(*[*UNIX*](http://id.wikipedia.org/wiki/UNIX)*),* atau [*Named Pipes*](http://id.wikipedia.org/w/index.php?title=Named_Pipes&action=edit&redlink=1) ([NT](http://id.wikipedia.org/wiki/NT)).
10. Lokalisasi. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.
11. Antar Muka. MySQL memiliki antar muka (interface) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi [API](http://id.wikipedia.org/wiki/API) (*Application Programming Interface).*
12. Klien dan Peralatan. MySQL dilengkapi dengan berbagai peralatan ([*tool*](http://id.wikipedia.org/wiki/Tool)) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk *online*.
13. Struktur tabel. MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani *ALTER TABLE*, dibandingkan basis data lainnya semacam [*PostgreSQL*](http://id.wikipedia.org/wiki/PostgreSQL) ataupun [*Oracle*](http://id.wikipedia.org/wiki/Oracle).

*MySQL* termasuk jenis RDBMS (*Relational Database Management* *System*). Itulah sebabnya istilah seperti tabel, baris, dan kolom digunakan pada MySQL.

Beberapa karakteristik yang dimiliki MySQL seperti yang disajikan pada tabel berikut :

**Tabel 2.4** Tabel Karakteristik MySQL

|  |  |  |
| --- | --- | --- |
| **NO** | **KARAKTERISTIK** | **DESKRIPSI** |
| 1 | Standar | MySQL mendukung *entry-level* ANSI SQL92 ODBC level 0-2 |
| 2 | *Character Set* | MySQL secara *default* menggunakan ISO-8859-I (Latin I) *character set* untuk data dan pengurutan |
| 3 | Bahasa Pemrograman | MySQL mendukung pemrograman aplikasi dalam bahasa Java, C, Perl, PHP, dan sebagainya |
| 4 | Tabel Besar | MySQL menyimpan tiap relasi (*table*) pada file terpisah di direktori basis data. |
| 5 | Kecepatan Dan Kemudahan Pemakaian | MySQL kira – kira tiga sampai empat kali lebih cepat dibanding basis data komersial, juga mudah dikelola |
| 6 | MySQL | MySQL adalah *open-source relational DBMS* |

Sumber : (Hariyanto, 2004:17)



**Gambar 2.15**  Logo *MySQL*

* 1. **Kerangka Pemikiran**

Dibutuhkan Sistem Pendukung Keputusan Seleksi Penerima Beras Miskin (Raskin) Pada Kantor Desa Moutong Kabupaten Bone Bolango

***Masalah***

***Peluang***

1. Bagaimana cara merancang sebuah Sistem Pendukung Keputusan Seleksi Penerima Beras Miskin (Raskin) Dengan Metode SAW Pada Desa Moutong Kabupaten Bone Bolango ?
2. Apakah Sistem Pendukung Keputusan Seleksi Penerima Beras Miskin (Raskin) Dengan Metode SAW Pada Desa Moutong Kabupaten Bone Bolango?

Sistem Pendukung Keputusan Seleksi Penerima Beras Miskin (Raskin) Dengan Metode SAW ( *Simple Additive Weighting* )

***Solusi***



***Analisis Sistem***



* Sistem Berjalan
* Sistem Yang Diusulkan

***Pembangunan Sistem***



**Desain *Sistem***



* Microsoft Visual Basic 6.0
* My SQL Server
* Desain Model
* Desain*UserInterface*
  + Desain*Output*
  + Desain*Input*
  + Desain Menu Utama
* Desain*Database*
* Desain Teknologi

***Pengujian Sistem***



* White Box
* Black Box

Kantor Desa Moutong Kecamatan TilongKabila Kabupaten Bone Bolango

***Implementasi Sistem***



***Tujuan***



1. Untuk mengetahui cara merekayasa Sistem Pendukung Keputusan Seleksi Penerima Beras Miskin (Raskin) Dengan Metode SAW Pada Desa Moutong Kabupaten Bone Bolango.
2. Sistem Pendukung Keputusan Seleksi Penerima Beras Miskin (Raskin) Dengan Metode SAW Dapat Di implementasikan pada Desa Moutong Kabupaten Bone Bolango.

**Gambar 2.16** Kerangka Pemikiran