BAB II
LANDASAN TEORI

II.1 Sistem

Berikut ini akan dijelaskan beberapa definisi tentang sistem :

1. Alter mendefinisikan sistem sebagai berikut :

   A system is a set of interacting components that operate together to accomplish a purpose [1].

2. Sedangkan Lucas mendefinisikannya sebagai berikut :

   A system is an organized, interacting, interdependent, and integrated set of components or variables [5].

Dari kedua definisi tersebut dapat disimpulkan bahwa sistem adalah sekumpulan komponen atau variabel yang saling berinteraksi dan bekerja bersama untuk mencapai sebuah tujuan.

Elemen-elemen sebuah sistem :

1. Tujuan (*purpose*)

   Tujuan sebuah sistem adalah alasan mengapa sistem tersebut ada dan tujuan itulah yang menjadi referensi untuk mengukur keberhasilannya.

2. Subsistem (*subsystem*)

   Subsistem adalah salah satu komponen sistem itu sendiri (sistem yang lebih kecil). Sebuah subsistem menerima masukan dan melakukan pekerjaan untuk menghasilkan keluaran.

3. Lingkungan (*environment*)

   Lingkungan adalah segala sesuatu yang berada di luar batas sebuah sistem yang berhubungan dengan sistem itu sendiri.

4. Batas (*boundary*)

   Batas sebuah sistem memisahkan sistem dari lingkungannya. Batas tersebut menjelaskan apa saja yang berada di dalam dan di luar sistem.

5. Masukan (*input*) dan keluaran (*output*)

Sebuah sistem menerima masukan dari lingkungannya dan mengembalikan hasilnya kepada lingkungannya pula.

6. Transformasi (*transformation*)

   Transformasi pada sebuah sistem menggambarkan suatu proses pekerjaan untuk mengubah *input* menjadi *output*.

7. Hubungan (*connection*)

   Hubungan antara subsistem-subsistem adalah aliran informasi dan material antar subsistem.

8. Mekanisme kontrol (*control mechanism*)

   Mekanisme kontrol adalah peraturan-peraturan dan prosedur-prosedur dimana subsistem-subsistem beroperasi dan saling berinteraksi.

UNIT PELAYANAN TEKNIS PERPUSTAKAAN UNIKOM

II.2 Informasi

Informasi berbeda dengan data. Data adalah semua fakta, gambar, suara yang diperlukan atau tidak diperlukan untuk suatu pekerjaan tertentu. Dibawah ini akan diberikan beberapa definisi mengenai informasi :

1. Alter mendefinisikan informasi sebagai berikut :

   > *Information is data whose form and content are appropriate for a particular use* [1].

2. Sedangkan Laudon mendefinisikan informasi sebagai berikut :

   > *Information is data that have been shaped into a form that is meaningful and useful to human being* [4].

3. Dan Lucas memberikan pengertian tentang definisi sebagai berikut :

   > *Information is some tangible or intangible entity that reduces uncertainty about a state or event* [5].

Dari beberapa definisi diatas dapat disimpulkan bahwa informasi adalah data yang telah diproses atau diolah sedemikian rupa sehingga berguna bagi pemakainya untuk memenuhi tujuan tertentu dan dapat mengurangi ketidakpastian atas suatu kejadian atau status.

Sebuah informasi dapat dikatakan baik bila memenuhi kriteria sebagai berikut : akurat, berhubungan, lengkap, waktu yang tepat, efektivitas biaya, dapat diuji, dan andal. Ada beberapa sumber yang dapat dijadikan sebagai sumber informasi diantaranya adalah organisasi itu sendiri (sebagai sumber utama), hasil pengamatan, pengalaman, lingkungan eksternal, dan peraturan pemerintah. Sebuah informasi mempunyai nilai apabila dapat memberikan nilai tambah (manfaat) terhadap sistem dan nilai tersebut ditentukan oleh pengguna berdasarkan kasus yang dihadapi.

II.3 Sistem Informasi

1. Alter mendefinisikan sistem informasi sebagai berikut :

   An information system is a combination of work practices, information, people, and information technologies organized to accomplish goals in an organization [1].

2. Definisi sistem informasi yang diberikan oleh Laudon adalah :

   An information system can be defined technically as a set of interrelated components that collect (or retrieve), process, store, and distribute information to support decision making and control in an organization [4].

3. Dan Lucas mendefinisikannya sebagai berikut :

   An information system is a set of organized procedures that, when executed, provide information for decision making and/or control of the organization [5].

Dari beberapa definisi diatas dapat disimpulkan bahwa sistem informasi adalah kombinasi dari komponen-komponen yang saling berhubungan dan terorganisir dimana ketika dijalankan akan memberikan informasi yang dibutuhkan bagi pengguna dalam pengambilan keputusan untuk mencapai sebuah tujuan dari perusahaan.

Tujuan sistem informasi bagi :

- Perusahaan
  - Pengolahan informasi secara efisien dan efektif

- Membentuk sistem kerja yang lebih efisien dan efektif
- Pengembangan sistem lebih mudah
- Perawatan sistem lebih mudah dan murah
- Pemanfaatan teknologi secara tepat guna

- Pekerja
  - Pengembangan diri menjadi manusia karya
- Lingkungan
  - Pertukaran informasi lebih mudah

Komponen sistem informasi :

- Tata cara kerja

  Tata cara kerja adalah metoda yang digunakan oleh manusia dengan menggunakan bantuan teknologi untuk melaksanakan suatu pekerjaan juga termasuk tata cara umum manusia berkoordinasi, berkomunikasi, membuat keputusan, dan lain-lain.
- Informasi

  Informasi merupakan data yang telah diolah, sehingga dapat digunakan oleh pengambil keputusan untuk mendukung pengambilan keputusan.
- Manusia

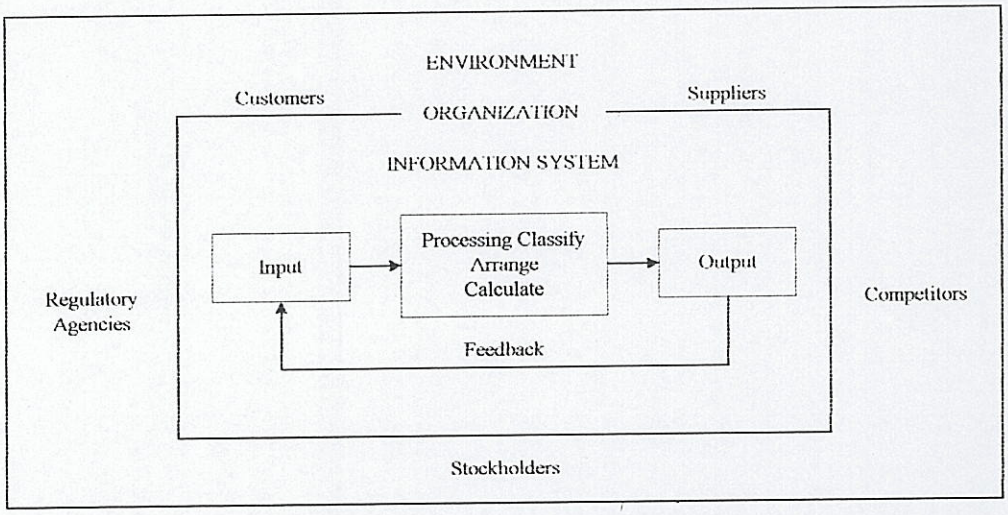
  Manusia merupakan komponen yang terpenting. Walaupun sebuah sistem sudah sepenuhnya terotomasi, namun masih tetap membutuhkan manusia sebagai pengambil keputusan akhir pada sebuah perusahaan.
- Teknologi informasi

  Merupakan perangkat keras dan perangkat lunak sebagai alat bantu pemrosesan data menjadi informasi.

Peran sistem informasi dalam perusahaan :

1. Menunjang pelaksanaan suatu tugas.
2. Mengaitkan perencanaan, pelaksanaan, dan pengendalian di dalam subsistem.
3. Mengkoordinasikan sub-subsistem.
4. Mengintegrasikan sub-subsistem.

Sistem informasi pada suatu perusahaan dapat digunakan oleh setiap anggota perusahaan (terutama para manajer-manajer) yang membutuhkannya. Sistem informasi memuat informasi tentang manusia, tempat, dan hal-hal penting lainnya dalam perusahaan atau dalam lingkungan yang mengelilinginya. Hal tersebut dapat dilihat pada gambar dibawah ini



Gambar 2.1 Fungsi Sebuah Sistem Informasi

Ada tiga aktivitas dalam sebuah sistem informasi yang menghasilkan informasi yang dibutuhkan oleh perusahaan untuk membuat keputusan, pengawasan pekerjaan, menganalisa permasalahan, dan membuat produk atau jasa baru. Ketiga aktivitas tersebut adalah masukan/*input* (mengumpulkan data mentah dalam organisasi atau dari lingkungan eksternal), proses/*processing* (mengubah *input* menjadi bentuk yang lebih berarti), dan keluaran/*output* (memindahkan proses informasi kepada orang atau aktivitas dimana informasi tersebut akan digunakan). Sistem informasi juga membutuhkan *feedback*, yaitu *output* yang dikembalikan pada orang yang tepat di perusahaan untuk membantu mereka mengevaluasi atau memperbaiki *input*.

Macam-macam sistem informasi :

1. Sistem pemrosesan transaksi
   - Mengumpulkan dan menyimpan informasi mengenai transaksi

- Pelakunya adalah orang yang memroses transaksi
- Contoh dalam lingkup manufaktur : menelusuri pekerjaan yang masih dalam proses.
- Contoh dalam lingkup keuangan : pemrosesan pembayaran kartu kredit.

2. Sistem informasi manajemen
   - Mengubah data dari sistem pemrosesan transaksi menjadi informasi bagi pengaturan sebuah organisasi dan sekaligus memantaunya.
   - Pelakunya adalah manajer dan orang yang menerima umpan balik mengenai pekerjaannya.
   - Contoh dalam lingkup manufaktur : pelaporan produksi mingguan berdasarkan produk dan operasi.
   - Contoh dalam lingkup keuangan : pelaporan penerimaan.
3. Sistem pendukung keputusan
   - Mendukung pengambilan keputusan dengan menyediakan informasi, model, dan perangkat pendukung untuk menganalisis informasi.
   - Pelakunya adalah analis, manajer, dan profesional lainnya.
   - Contoh dalam lingkup manufaktur : sistem yang menampilkan spesifikasi bagi operator mesin.
   - Contoh dalam lingkup keuangan : sistem yang menampilkan persediaan.
   - Bisa dipakai untuk *repetitive* ataupun *non repetitive decision making*.
   - Memiliki *model based* di dalamnya.
4. Sistem informasi eksekutif
   - Menyediakan informasi yang siap diakses dan dalam bentuk yang interaktif.
   - Pelakunya adalah eksekutif dan manajer tingkat tinggi.
   - Contoh dalam lingkup manufaktur : akses fleksibel terhadap data produksi.
   - Contoh dalam lingkup keuangan : akses fleksibel terhadap rencana keuangan perusahaan.
5. Sistem pakar
   - Memungkinkan pengetahuan pakar untuk dapat tersedia bagi orang banyak.

- Mendukung penyelesaian masalah dalam lingkup tertentu.
- Pelakunya adalah orang yang menyelesaikan permasalahan dalam lingkungan dimana pakar dibutuhkan.
- Contoh dalam lingkup manufaktur : sistem untuk mendiagnosa kegagalan mesin.
- Contoh dalam lingkup keuangan : sistem untuk mengidentifikasi perubahan *trend*.
- Mengetahui *knowledge base*.

6. Sistem otomasi kantor
   - Membantu orang untuk mengakses dokumen dan pesan-pesan.
   - Menyediakan perangkat pendukung yang dapat mengefisiensikan serta mengefektifkan pekerjaan kantor secara umum.
   - Pelakunya adalah pekerja kantor.
   - Contoh dalam lingkup manufaktur dan keuangan : *spreadsheet*, *e-mail*.

II.4 Siklus Perancangan Sistem Informasi

Dalam perancangan sistem informasi ada beberapa tahap yang harus dilakukan, yaitu :

1. Tahap awal (*start up*)
2. Tahap perencanaan (*planning*)
3. Tahap analisis (*analysis*)
4. Tahap implementasi (*implementation*)
5. Tahap penggunaan dan perawatan (*use and maintenance*)

Masing-masing tahap tersebut akan dibahas pada sub bab berikut dibawah ini.

II.4.1 Tahap Awal (*Start Up*)

Pada tahap awal ini akan didefinisikan tentang sistem dan lingkungan. Definisi tentang sistem sudah dibahas pada sub bab sebelumnya. Lingkungan adalah segala sesuatu yang terdapat di luar batas sebuah sistem yang berhubungan

dengan sistem termasuk semua sumber yang dijadikan *input* bagi sistem dan penerima *output* dari sistem. Langkah awal yang perlu dilakukan adalah kita harus terlebih dahulu mengetahui tentang :

- Tujuan organisasi

  Tujuan organisasi menggambarkan suatu hasil yang ingin dicapai oleh sebuah organisasi baik dalam jangka pendek maupun jangka panjang. Meskipun tujuan organisasi bukan merupakan komponen dari sistem informasi, tetapi tujuan organisasi memainkan peranan yang penting dalam menentukan tata cara kerja. Tujuan organisasi menetapkan kriteria untuk memutuskan apakah tata cara kerja sebuah organisasi perlu diubah atau tidak.

- Struktur organisasi

  Struktur organisasi menggambarkan hubungan antara bagian-bagian yang terdapat pada suatu organisasi baik secara horisontal dan vertikal. Struktur organisasi juga menggambarkan garis wewenang dan tanggung jawab masing-masing bagian.

- *Standard operating procedure* (SOP)

  Standard operating procedure (SOP) adalah prosedur atau aturan standar yang digunakan untuk mengerjakan tugas/pekerjaan yang telah dikembangkan untuk menguasai situasi yang diharapkan.

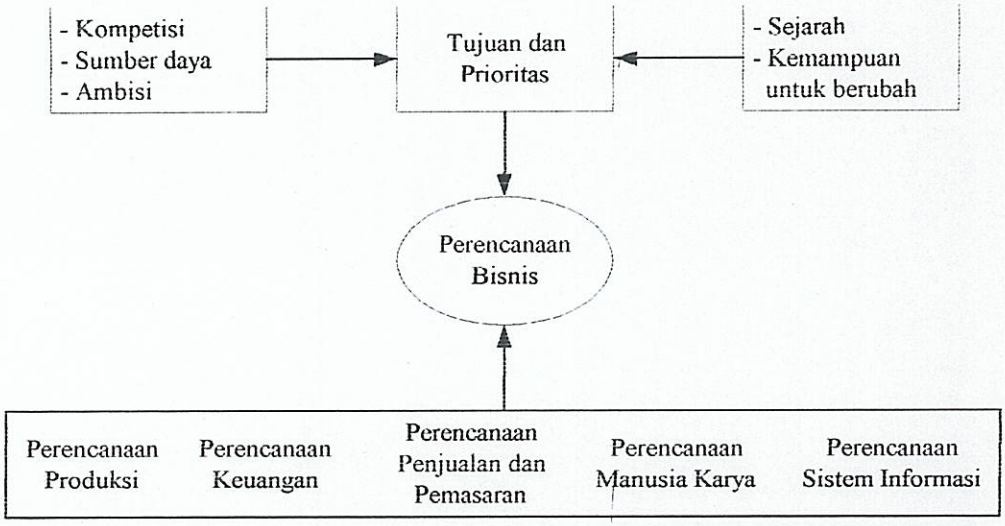
II.4.2 Tahap Perencanaan (*Planning*)

Perencanaan sistem informasi merupakan bagian dari perencanaan bisnis yang terintegrasi. Perencanaan bisnis merupakan proses mengidentifikasi tujuan dan prioritas organisasi serta tindakan yang akan dilakukan untuk mencapai tujuan.

Faktor-faktor penting dalam melakukan perencanaan sistem informasi adalah :

1. Kebutuhan akan perubahan
2. Pemrosesan dan pengendalian sentralisasi dan desentralisasi
3. Lokasi perangkat keras dan data

4. Standar, kepemilikan, dan pengendalian



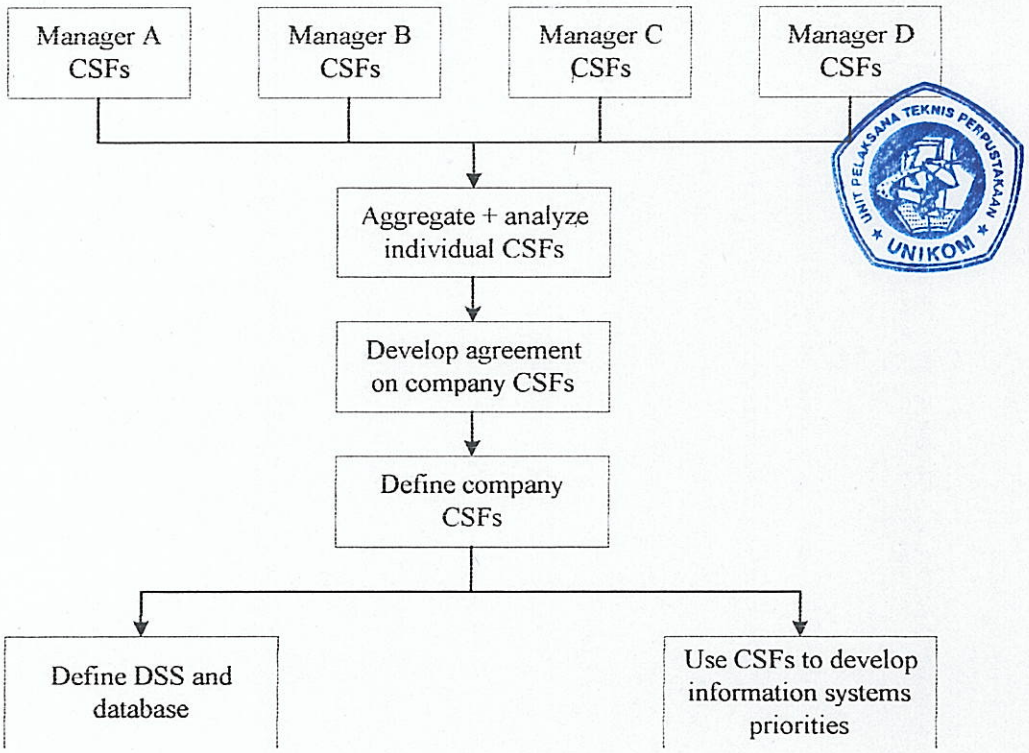
Gambar 2.2 Hubungan Sistem Informasi dengan Perencanaan Bisnis

Ada beberapa metode untuk melakukan perencanaan bisnis, yaitu : *Business System Planning* (BSP), *Critical Success Factor* (CSF), *Structured Analysis and Design Technique* (SADT), dan *Hierarchy Input Process Output* (HIPO). Berikut ini akan dijelaskan satu per satu mengenai keempat metode diatas.

- *Critical Success Factor* (CSF)

  Critical Success Factor merupakan pendekatan untuk mengidentifikasi faktor-faktor kritis yang berpengaruh secara signifikan terhadap keberhasilan pelaksanaan bisnis. Sistem informasi yang menunjang faktor-faktor tersebut memiliki prioritas tinggi. Pendekatan CSF merupakan metodologi penting dalam perencanaan sistem informasi. CSF dapat digunakan secara mandiri atau diikutsertakan dalam metodologi perencanaan lainnya sebagai suatu komponen penting dalam proses perencanaan. Pendekatan ini dikembangkan oleh John Rockart dan kawan-kawan yang merasa tidak puas akan kekurangan emphasis pada kebutuhan informasi penting yang diberikan oleh para manajer, yang terdapat pada metode-metode perencanaan sistem informasi lainnya.

Maksud utamanya adalah bahwa kebutuhan-kebutuhan informasi organisasi harus ditentukan oleh CSF-nya, yang merupakan sejumlah faktor-faktor penting terhadap suatu kesuksesan suatu perusahaan atau proyek. Hal tersebut merupakan area-area penting dimana performansi yang berhasil menjamin kesuksesan organisasi dan tercapainya tujuan-tujuan organisasi. Sistem informasi dirancang untuk mengukur performansi tiap-tiap CSF secara berkesinambungan, dan melaporkan informasi ini kepada pihak manajemen.



UNIT PELAKSANA TEKNIS PERPUSTAKAAN UNIKOM

Gambar 2.3 Pengembangan CSF

Dalam perencanaan sistem informasi dengan menggunakan pendekatan ini, tim manajer bersama-sama menentukan masing-masing CSF dan pandangan mereka masing-masing terhadap CSF bagi organisasi. Tim manajemen diminta untuk mengidentifikasi tujuan-tujuan mereka dan CSF yang dapat menjamin tercapainya tujuan-tujuan tersebut. Hal ini meliputi pertimbangan strategi kompetitif perusahaan, posisi perusahaan dalam dunia

industri, lingkungan ekonomi dan politik. Manajer-manajer diminta untuk mengidentifikasikan sejumlah area penting yang terbatas (sekitar 5 hingga 10) dimana "*things have to go right*", "*value would hurt most*", atau informasi-informasi yang paling diinginkan mengenai sesuatu jika mereka sedang tidak berada di tempat. Sebagai contoh, manajer-manajer pada sebuah supermarket mungkin berpikir mengenai *pricing*, *product mix*, *inventory turnover*, dan promosi penjualan sebagai CSF. Performansi pada area-area ini dapat diukur dari *gross margin* dan persentase *net margin*, perbandingan ruang lingkup produk, rasio *inventory turnover*, dan hasil-hasil penjualan dengan waktu dan jumlah promosi penjualan.

Tahapan menentukan CSF :

- Identifikasi misi utama dari organisasi
- Identifikasi faktor-faktor kritis bagi keberhasilan
- Untuk setiap faktor-faktor kritis, identifikasi tolok ukur unjuk kerjanya
- Pengembangan sistem untuk mengumpulkan dan menggunakan informasi

- *Business System Planning* (BSP)

Business System Planning adalah suatu pendekatan terstruktur yang mendukung suatu organisasi dalam mengembangkan rencana-rencana sistem informasi dengan tujuan untuk memenuhi kebutuhan akan informasi-informasi jangka pendek maupun jangka panjang. Satu hal penting dalam metodologi BSP adalah bahwa suatu sistem informasi perusahaan harus direncanakan dari atas ke bawah dan diimplementasikan satu demi satu dari bawah ke atas.

Top down planning membutuhkan suatu kelompok perencana dari para eksekutif puncak meletakkan misi strategis dan tujuan-tujuan dari organisasi. Kemudian kelompok perencana dari para manajer memformulasikan bagaimana tujuan-tujuan tersebut diimplementasikan dalam bentuk fungsi-fungsi dasar (*marketing*, *manufacturing*, dan lain-lain) dan proses (pemesanan, *shipping*, *receiving*, dan sebagainya). Selanjutnya tim manajer dan para ahli sistem informasi meramalkan atau memperkirakan kemampuan-kemampuan dari *information technology* (IT) yang mungkin dibutuhkan untuk mendukung

proses-proses dasar tersebut. Akhirnya arsitektur IT dibentuk berupa rancangan-rancangan bagi *technology platform* (sistem komputer, *software*, *networks*) dan *database* organisasi yang dapat menyediakan kemampuan-kemampuan yang dibutuhkan oleh bisnis tersebut.

Bottom up implementation meliputi kegiatan-kegiatan pengembangan aplikasi yang dihasilkan oleh kelompok-kelompok *end users* dan informasi sistem profesional. Tujuan mereka adalah untuk mengembangkan aplikasi-aplikasi bisnis spesifik (misal: *electronic data interchange*) yang bertumpu pada teknologi platform yang desainnya ditentukan oleh *hardware*, *software*, *database*, dan *network* dari arsitektur IT. Tiap aplikasi harus mampu melayani sebuah fungsi bisnis yang mendukung misi dan tujuan dari organisasi.

Langkah-langkah membuat BSP :

- Definisikan proses bisnis
- Definisikan data bisnis
- Definisikan arsitektur informasi sekarang
- Definisikan tujuan bisnis dan lingkungan
- Membuat sebuah arsitektur informasi untuk masa yang akan datang

Pada dasarnya, BSP memberikan gambaran mengenai siapa yang membuat data (informasi), siapa yang menggunakan data, dan bagaimana keterlibatan pengguna terhadap data tersebut.

- *Structured Analysis and Design Technique* (SADT)

Structured Analysis and Design Technique (SADT) adalah suatu produk dari *Soft Technique Incorporated* dan dikembangkan sebagai sebuah metodologi lengkap. SADT digunakan untuk mengontrol perkakas-perkakas yang digunakan dalam menganalisa dan merancang suatu sistem. SADT menyediakan representasi grafis dari fungsi-fungsi hirarki dan data dalam suatu sistem. SADT menganut konsep dekomposisi, yaitu menggambarkan terlebih dahulu sistem secara utuh sebagai tingkat tertinggi dan memecah-mecahnya menjadi lebih terinci.

Untuk metode ini, dilakukan analisis secara terstruktur mengenai tahapan-tahapan dari setiap aktivitas (apa nama aktivitasnya, mekanisme yang digunakan untuk menjalankan aktivitas tersebut, apa yang dikontrol) dan juga mengenai aliran data dan informasi yang terjadi pada semua bagian (apa saja *databasenya*, apa kontrolnya, dan bagaimana *storage* datanya).

- *Hierarchy Input Process Output* (HIPO)

Sistem *Hierarchy Input Process Output* (HIPO) digunakan untuk *layout* kebutuhan-kebutuhan akan sistem informasi, dengan gambaran *file-file input* dan *output* atau melaporkan laporan-laporan yang berhubungan dengan proses-proses yang terjadi. Tingkat kedetilan cukup umum sehingga IPO tidak berguna dalam menyampaikan logika tingkat awal kepada seorang *programmer*. HIPO memberikan suatu gambaran dari sistem (ada *input*, proses, dan *output*). HIPO juga menunjukkan bagaimana proses-proses individual berhubungan satu sama lain dan berhubungan dengan *file-file* data.

Langkah-langkah pembuatan HIPO *chart* :

- Buat *hierarchy chart*, jangan terlalu detil. *Hierarchy chart* sebaiknya tidak lebih dari tiga tingkat, termasuk *the root (top) node.*
- Dilakukan mulai dari level abstrak tertinggi (satu level di bawah proses *the root* – paling atas).
- Identifikasi masukan-masukan dasar dan *output-output* yang berhubungan dengan sistem.
- Identifikasi proses-proses utama yang mentransformasikan *input* menjadi *output*. Hubungkan *input* dan *output* dengan proses.
- Buat IPO chart. Buat daftar setiap proses secara berurutan. Hubungkan *file input* dan *file output* dengan proses-proses individual dengan anak panah.

Input yang dibutuhkan dalam HIPO berupa data yang nantinya dapat diproses atau diolah untuk dijadikan informasi (*output*). Sebagai contoh : data harian *scrap* dihitung untuk mendapatkan laporan *scrap* per minggu.

II.4.3 Tahap Analisis (*Analysis*)

Tujuan analisis sistem informasi adalah untuk mengidentifikasi dan mencari solusi permasalahan sistem informasi dalam suatu organisasi. Tahapan kerja pendekatan sistem menurut Raymond McLeod adalah :

- Persiapan
- Analisis persyaratan dengan menggunakan BSP, CSF, HIPO, SADT
- Definisi masalah
- Solusi masalah

Teknik dan peralatan kerja perancangan sistem informasi antara lain :

- Metodologi terstruktur, yang mencakup :
  - Analisis terstruktur, contoh alat kerja adalah DFD
  - Perancangan terstruktur, contoh alat kerja adalah diagram struktur
  - Pemrograman terstruktur, contoh alat kerja adalah *system flowchart*
- *Prototype*
- Alat kerja otomatis, seperti *application generator*

Dasar-dasar pemilihan alat kerja yang terbaik tergantung pada beberapa faktor berikut :

- DFD dirancang untuk kemudahan bagi pemakai akhir, oleh karena itu dalam proyek sistem informasi tahap awal digunakan DFD
- DFD merupakan dasar bagi sebagian besar komponen grafis pada produk-produk CASE. Jika suatu perusahaan menggunakan CASE, maka penggunaan HIPO dan *system flowchart* adalah tidak tepat.
- Alat kerja yang dipilih tergantung dari standar dokumentasi yang ditetapkan perusahaan.

II.4.4 Tahap Implementasi (*Implementation*)

Tahap ini mencakup penggunaan perangkat keras dan lunak, pemrograman, percobaan dan pelatihan. Konversi sistem lama dan baru dibagi menjadi :

- *Parallel strategy*, operasi antara sistem lama dan sistem baru dilakukan secara bersamaan untuk jangka waktu tertentu, baru jika aman maka sistem lama dihentikan.
- *Direct cutover*, sistem lama diputus dan digantikan oleh sistem yang baru.
- *Pilot study*, memilih beberapa daerah yang "tidak terlalu berbahaya" dan mempraktekkan sistem baru disana, jika sukses baru diberlakukan untuk semuanya.
- *Phased approach*, pelaksanaan dilakukan secara bertahap.

II.4.5 Tahap Penggunaan dan Perawatan (*Use and Maintenance*)

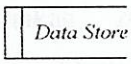
Pada tahap ini dilakukan peninjauan kembali implementasi yang telah dilakukan. Perawatan sistem yang dilakukan antara lain :

- Memperbaiki kesalahan
- Menjaga agar sistem selalu baru
- Mengembangkan sistem
- Mengatasi virus, perangkat keras, dan masalah sistem operasi
- Merawat *database*

II.5 *Data Flow Diagram* (DFD)

Data flow diagram (DFD) adalah sebuah diagram grafis yang menggambarkan aliran data dari suatu proses (fungsi) ke proses (fungsi) lain dalam sebuah sistem (informasi) yang diamati. *Data flow diagram* diperlukan untuk menganalisis pergerakan data/informasi dari suatu bagian ke bagian yang lain dalam satu organisasi, proses/transformasi yang dialami oleh 'sebuah' data/informasi dan bagaimana hasil transformasi tersebut.

Pembuatan *data flow diagram* menekankan pada logika dan struktur yang mendasari terbentuknya suatu organisasi/sistem. Terdapat empat symbol pokok dalam sebuah DFD, yaitu :

	Simbol	Arti
1.		Menunjukkan suatu proses/transformasi/kerja/aktivitas. Selalu diisi dengan sebuah kata kerja. Selalu ada aliran data/informasi yang masuk dan keluar (selalu ada *input* dan *output*)
2.		*Entity*, merupakan bagian subyek ataupun obyek dari sebuah proses. Selalu berupa kata benda (orang, bagian, departemen, dll)
3.		*Data flow* (aliran data) menunjukkan dari mana data/informasi berasal dan tujuannya.
4.		*Data store* : tempat penyimpanan data/informasi, biasanya merupakan suatu *file database* dengan format tertentu.

Gambar 2.4 Simbol *Data Flow Diagram* (DFD)

UNIT PELAKSANA TEKNIS PERPUSTAKAAN UNIKOM

Langkah-langkah pembuatan DFD :

1. Buat sebuah daftar (*list*) dari aktifitas bisnis yang sedang diamati. Biasanya dapat dilihat dari *job description* masing-masing bagian dalam sebuah organisasi. Amati juga aliran pekerjaan (*work flow*) yang dilakukan dalam organisasi tersebut. Gunakan daftar aktifitas dan aliran pekerjaan tersebut dan pilah-pilah menjadi :
   - *Entity*
   - Data/informasi (beserta alirannya, dari mana ke mana)
   - Proses yang dilakukan (apa *input*-nya, apa *output*-nya)
   - *Data store* : mana yang bisa dianggap sebagai tempat penyimpanan data (lemari arsip, *file*, tabel, dan sebagainya)
2. Buat sebuah *context diagram* yang menggambarkan hubungan antara *entity* dan data/informasi, dari dan ke sistem yang diamati. Jangan menggambarkan proses dan *data store* terlalu rinci lebih dulu.
3. Buat diagram 0 (level 0) yang memperlihatkan proses/transformasi. Tetap dijaga agar tidak terlalu rinci. Mulai perlihatkan *data store* pada level ini.

4. Buat diagram anak (*child diagram*) untuk setiap proses yang ada pada diagram 0.
5. Periksa kalau ada kesalahan dan pastikan tindakan Anda memberi label pada tiap proses dan *data flow* memang benar-benar berguna.
6. Kembangkan DFD fisik (*physical* DFD) dari *logical* DFD yang sudah dikembangkan pada langkah 1 sampai 5 di atas. Bedakan proses mana yang dilakukan manual dan mana yang dapat diotomatisasi, tentukan juga *file-file* dan laporan-laporan apa saja yang diperlukan. Tambahkan proses-proses kendali untuk menjadi indikator bila sebuah proses telah selesai dilakukan ataupun bila timbul kesalahan.
7. Pisahkan *physical* DFD tersebut dengan mengelompokkan atau memisahkannya sesuai dengan kesamaan yang mungkin muncul. Kegunaannya adalah untuk mempermudah proses pembuatan program dan implementasinya.

II.6 Basis Data (*Database*)

Basis data terdiri dari dua kata, yaitu basis dan data. Kata "basis" dapat diartikan sebagai markas atau gudang, tempat bersarang/berkumpul. Sedangkan kata "data" sendiri dapat diartikan sebagai representasi fakta dunia nyata yang mewakili suatu obyek seperti manusia (pegawai, pembeli, mahasiswa, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya.

Dari definisi kedua kata tersebut maka kita dapat membuat suatu definisi bagi basis data, yaitu :

- Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
- Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redundansi) yang tidak perlu untuk memenuhi berbagai kebutuhan.

- Kumpulan *file*/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronis.

Basis data dapat dibayangkan sebagai sebuah lemari arsip. Jika kita memiliki sebuah lemari arsip maka kita dapat melakukan beberapa hal seperti memberi sampul/map pada kumpulan arsip yang akan disimpan, menentukan jenis/kelompok arsip, memberi penomoran dengan suatu pola tertentu pada setiap sampul/map lalu menyimpannya dengan urutan tertentu di dalam lemari. Pada dasarnya basis data memiliki prinsip kerja dan tujuan yang sama dengan lemari arsip. Prinsip utamanya adalah pengaturan data/arsip dan tujuan utamanya adalah kemudahan dan kecepatan dalam pengambilan kembali data/arsip. Perbedaannya hanya terletak pada media penyimpanan yang digunakan. Jika lemari arsip menggunakan lemari dari besi atau kayu sebagai media penyimpanan, maka basis data menggunakan media penyimpanan elektronis seperti disket atau *harddisk*.

Tidak semua bentuk penyimpanan data secara elektronis bisa disebut basis data. Misalnya penyimpanan dokumen berisi data dalam *file* teks (dengan program pengolah kata) atau *file spred sheet* tidak dapat disebut sebagai basis data karena didalamnya tidak terdapat pemilahan dan pengelompokkan data sesuai jenis/fungsi data sehingga akan menyulitkan pencarian data kelak. Yang sangat ditonjolkan dalam basis data adalah pengaturan/pemilahan/pengelompokkan data yang akan kita simpan sesuai fungsi/jenisnya.

Operasi-operasi dasar yang dapat dilakukan berkenaan dengan basis data dapat meliputi :

- Pembuatan basis data baru (*create database*) yang identik dengan pembuatan lemari arsip yang baru.
- Penghapusan basis data (*drop database*) yang identik dengan perusakan lemari arsip (sekaligus beserta isinya jika ada).
- Pembuatan *file*/tabel baru ke suatu basis data (*create table*) yang identik dengan penambahan map arsip baru ke sebuah lemari arsip yang telah ada.
- Penghapusan *file*/tabel dari suatu basis data (*drop table*) yang identik dengan perusakan map arsip lama yang ada di sebuah lemari arsip.

- Penambahan/pengisian data baru ke sebuah *file*/tabel di sebuah basis data (*insert*) yang identik dengan penambahan lembaran arsip ke sebuah map arsip.
- Pengambilan data dari sebuah *file*/tabel (*retrieve/search*) yang identik dengan pencarian lembaran arsip dari sebuah map arsip.
- Pengubahan data dari sebuah *file*/tabel (*update*) yang identik dengan perbaikan isi lembaran arsip yang ada di sebuah map arsip.
- Penghapusan data dari sebuah *file*/tabel (*delete*) yang identik dengan penghapusan sebuah lembaran arsip yang ada di sebuah map arsip.

Operasi yang berkenaan dengan pembuatan obyek (basis data dan tabel) merupakan operasi awal yang hanya dilakukan sekali dan berlaku seterusnya. Sedangkan operasi-operasi yang berkaitan dengan isi tabel (data) merupakan operasi rutin yang akan berlangsung berulang-ulang dan karena itu operasi-operasi inilah yang lebih tepat mewakili aktivitas pengelolaan (*management*) dan pengolahan (*processing*) data dalam basis data.

Pemanfaatan basis data dilakukan untuk memenuhi beberapa tujuan (obyektif) seperti berikut ini :

- Kecepatan dan kemudahan (*speed*)

  Pemanfaatan basis data memungkinkan kita untuk dapat menyimpan data atau melakukan perubahan/manipulasi terhadap data atau menampilkan kembali data tersebut dengan lebih cepat dan mudah daripada jika kita menyimpan data secara manual (non elektronis) atau secara elektronis (tetapi tidak dalam bentuk penerapan basis data, misalnya dalam bentuk *spread sheet* atau dokumen teks biasa).
- Efisiensi ruang penyimpanan (*space*)

  Karena keterkaitan yang erat antar kelompok data dalam sebuah basis data, maka redundansi (pengulangan) data pasti akan selalu ada. Banyaknya redundansi ini tentu akan memperbesar ruang penyimpanan (baik di memori utama maupun memori sekunder) yang harus disediakan. Dengan basis data, efisiensi/optimalisasi penggunaan ruang penyimpanan dapat dilakukan karena kita dapat melakukan penekanan jumlah redundansi data baik dengan

menerapkan sejumlah pengkodean atan dengan membuat relasi-relasi (dalam bentuk *file*) antar kelompok data yang saling berhubungan.

- Keakuratan (*accuracy*)

  Pemanfaatan pengkodean atau pembentukan relasi antar data bersama dengan penerapan aturan/batasan (*constraint*) tipe data, domain data, keunikan data, dan sebagainya, yang secara ketat dapat diterapkan dalam sebuah basis data, sangat berguna untuk menekan ketidakakuratan pemasukan/penyimpanan data.

- Ketersediaan (*availability*)

  Pertumbuhan data (baik dari sisi jumlah maupun jenisnya) sejalan dengan waktu akan semakin membutuhkan ruang penyimpanan yang besar, padahal tidak semua data itu selalu digunakan/dibutuhkan. Karena itu kita dapat memilah adanya data utama/master/referensi, data transaksi, data histori hingga data kadaluarsa. Data yang sudah jarang atau bahkan tidak pernah lagi digunakan dapat diatur untuk dilepaskan dari sistem basis data yang sedang aktif (menjadi *off-line*) baik dengan cara penghapusan atau dengan memindahkannya ke media penyimpanan *off-line* (seperti *removeable disk* atau *tape*). Di sisi lain, karena kepentingan pemakaian data, sebuah basis data dapat memiliki data yang disebar di banyak lokasi geografis.

- Kelengkapan (*completeness*)

  Lengkap/tidaknya data yang kita kelola dalam sebuah basis data bersifat relatif (baik terhadap kebutuhan pemakai maupun terhadap waktu). Bila seorang pemakai sudah menganggap bahwa data yang dipelihara sudah lengkap, maka pemakai yang lain belum tentu berpendapat sama. Atau jika sekarang sudah dianggap lengkap belum tentu di masa yang akan datang juga demikian. Dalam sebuah basis data, disamping data kita juga harus menyimpan struktur, baik yang mendefinisikan obyek-obyek dalam basis data maupun definisi detail dari tiap obyek seperti struktur *file*/tabel atau indeks. Untuk mengakomodasi kebutuhan kelengkapan data yang semakin berkembang, maka kita tidak hanya dapat menambah *record-record* data, tetapi juga dapat melakukan perubahan struktur dalam basis data, baik dalam bentuk

penambahan obyek baru (tabel) atau dengan penambahan *field-field* baru pada suatu tabel.

- Keamanan (*security*)

  Ada sejumlah sistem (aplikasi) pengelola basis data yang tidak menerapkan aspek keamanan dalam penggunaan basis data. Tetapi untuk sistem yang besar dan serius, aspek keamanan juga dapat diterapkan dengan ketat. Dengan begitu kita dapat menentukan siapa saja pemakai yang boleh menggunakan basis data beserta obyek-obyek didalamnya dan menentukan jenis-jenis operasi apa saja yang boleh dilakukannya.

- Kebersamaan pemakaian (*sharability*)

  Pemakai basis data seringkali tidak terbatas pada satu pemakai saja atau di satu lokasi saja atau oleh satu sistem/aplikasi saja. Misalnya data pegawai dalam basis data kepegawaian dapat digunakan oleh banyak pemakai antara lain oleh sejumlah departemen dalam perusahaan atau oleh banyak sistem (sistem penggajian, sistem akuntansi, sistem *inventory*, dan sebagainya). Basis data yang dikelola oleh sistem (aplikasi) yang mendukung lingkungan *multiuser* akan dapat memenuhi kebutuhan ini tetapi tetap dengan menjaga atau menghindari terhadap munculnya persoalan baru seperti inkonsistensi data (karena data yang sama diubah oleh banyak pemakai pada saat bersamaan) atau kondisi *deadlock* (karena ada banyak pemakai yang saling menunggu untuk menggunakan data).

Hampir di semua aspek pemanfaatan perangkat komputer dalam sebuah organisasi/perusahaan senantiasa berhubungan dengan basis data. Perangkat komputer dalam suatu organisasi/perusahaan biasanya digunakan untuk menjalankan fungsi pengelolaan sistem informasi, yang dewasa ini sudah menjadi suatu keharusan, demi untuk meningkatkan efisiensi, daya saing, keakuratan, kecepatan operasional organisasi/perusahaan. Basis data merupakan salah satu komponen utama dalam setiap sistem informasi. Tidak ada sistem informasi yang bisa dibuat/dijalankan tanpa adanya basis data.

Bidang-bidang fungsional yang telah umum memanfaatkan basis data untuk efisiensi, akurasi, dan kecepatan operasi antara lain adalah :

- Kepegawaian : untuk berbagai perusahaan yang memiliki banyak pegawai.
- Pergudangan (*inventory*) : untuk perusahaan manufaktur (pabrikan), grosir (*reseller*), apotik, dan lain-lain.
- Akuntansi : untuk berbagai perusahaan.
- Reservasi : untuk hotel, pesawat, kereta api, dan lain-lain.
- Layanan pelanggan (*customer care*) : untuk perusahaan yang berhubungan dengan banyak pelanggan (bank, konsultan, dan lain-lain)

II.7 Normalisasi Data

Perancangan basis data diperlukan agar kita dapat memiliki basis data yang kompak dan efisien dalam penggunaan ruang penyimpanan, cepat dalam pengaksesan, dan mudah dalam menambah, mengubah, dan menghapus data. Dalam merancang basis data, kita dapat melakukannya dengan menerapkan normalisasi terhadap struktur tabel yang telah diketahui atau dengan cara langsung membuat model *Entity Relationship* (ER). Normalisasi merupakan salah satu cara untuk membangun desain lojik basis data relasional yang tidak secara langsung berkaitan dengan model data, tetapi dengan menerapkan sejumlah aturan dan kriteria standar untuk menghasilkan struktur tabel yang normal.

Dalam pendekatan normalisasi, seorang perancang basis data bertitik tolak dari situasi yang nyata. Perancang tersebut telah memiliki *item-item* data yang siap ditempatkan dalam baris dan kolom pada tabel-tabel relasional, demikian juga dengan sejumlah aturan tentang keterhubungan antara *item-item* data tersebut. Sementara itu untuk pendekatan kedua (ER), pembuatan model data lebih tepat dilakukan jika yang telah diketahui adalah prinsip-prinsip sistem secara keseluruhan. Tetapi kadang-kadang kedua pendekatan ini (Normalisasi dan ER) dilakukan secara bersama-sama, berganti-ganti. Dari fakta yang telah kita miliki, kita lakukan normalisasi dan untuk kepentingan evaluasi dan dokumentasi, hasil normalisasi itu kita wujudkan dalam sebuah model data. Model data yang sudah jadi tersebut dapat dimodifikasi dengan pertimbangan tertentu dan hasil modifikasinya kemudian kita implementasikan dalam bentuk sejumlah struktur

tabel dalam sebuah basis data. Struktur ini kemudian dapat kita uji kembali dengan menerapkan aturan-aturan normalisasi hingga akhirnya kita peroleh sebuah struktur basis data yang benar-benar efektif dan efisien.

Sebelum membahas normalisasi lebih jauh, maka kita akan membahas terlebih dahulu beberapa pengertian mengenai atribut, *key*, *domain*, dan ketergantungan fungsional.

II.7.1 Atribut Tabel

Dalam perancangan basis data dikenal suatu istilah yaitu atribut dimana istilah ini sebenarnya identik dengan pemakaian istilah kolom data. Istilah atribut umum digunakan dalam perancangan basis data karena istilah tersebut lebih impresif dalam menunjukkan fungsinya sebagai pembentuk karakteristik (sifat-sifat) yang melekat pada sebuah tabel, terlebih lagi penerapan aturan-aturan normalisasi terhadap atribut-atribut pada sebuah tabel dapat mengakibatkan penghilangan kolom tertentu, penambahan kolom baru, atau bahkan penambahan tabel baru.

Disamping penamaan yang unik berdasarkan fungsinya di setiap tabel, atribut-atribut tersebut dapat dibedakan berdasarkan sejumlah pengelompokkan, yaitu :

1. *Key* dan atribut deskriptif

   Key adalah satu atau gabungan dari beberapa atribut yang dapat membedakan semua baris data (*row*) dalam tabel secara unik. Artinya, jika suatu atribut dijadikan *key* maka tidak boleh ada dua atau lebih baris data dengan nilai yang sama untuk atribut tersebut. Ada tiga macam *key* yang dapat diterapkan pada suatu tabel, yaitu :

   - *Superkey* : merupakan satu atau lebih atribut (kumpulan atribut) yang dapat membedakan setiap baris data dalam sebuah tabel secara unik. Pada sebuah tabel mungkin saja terdapat lebih dari satu kumpulan atribut yang bersifat seperti itu.

- *Candidate key* : merupakan kumpulan atribut minimal yang dapat membedakan setiap baris data dalam sebuah tabel secara unik. Sebuah *candidate key* tidak boleh berisi atribut atau kumpulan atribut yang telah menjadi *superkey* yang lain (atau dengan kata lain, sebuah *candidate key* pastilah *superkey*, tetapi belum tentu sebaliknya).
- *Primary key* : pada sebuah tabel mungkin saja terdapat lebih dari satu *candidate key*, dan salah satunya dapat dijadikan sebagai *primary key*. Pemilihan *primary key* dari sejumlah *candidate key* umumnya didasari oleh :
  - *key* tersebut lebih sering (lebih natural) untuk dijadikan sebagai acuan.
  - *key* tersebut lebih ringkas.
  - jaminan keunikan *key* tersebut lebih baik.

Atribut deskriptif adalah atribut-atribut yang tidak menjadi atau merupakan anggota dari *primary key*.

UNIT PELAYANAN TEKNIS PERPUSTAKAAN UNIKOM

2. Atribut sederhana (*simple attribute*) dan atribut komposit (*composite attribute*)

   Atribut sederhana adalah atribut atomik yang tidak dapat dipilah lagi, sedangkan atribut komposit adalah atribut yang masih dapat diuraikan lagi menjadi sub-sub atribut yang masing-masing memiliki makna. Jika dalam penerapannya tidak ada aktivitas pengolahan data yang hanya melibatkan sub-sub atribut tersebut secara khusus, maka pemakaian atribut komposit ini dapat terus dipertahankan untuk kesederhanaan struktur tabel. Tetapi jika kemudian ada kebutuhan untuk melakukan pengolahan data terhadap sub-sub atribut secara khusus, maka atribut komposit dapat di dekomposisi menjadi atribut sederhana.

3. Atribut bernilai tunggal (*single-valued attribute*) dan atribut bernilai banyak (*multivalued attribute*)

   Atribut bernilai tunggal ditujukan pada atribut-atribut yang memiliki paling banyak satu nilai untuk setiap baris data, sedangkan atribut bernilai banyak ditujukan pada atribut-atribut yang dapat kita isi dengan lebih dari satu nilai tetapi jenisnya sama.

4. Atribut harus bernilai (*mandatory attribute*) dan nilai Null

Mandatory attribute adalah satu atau beberapa atribut pada sebuah tabel yang harus berisi data, jadi nilainya tidak boleh kosong. Selain *mandatory attribute* ada pula atribut-atribut lain pada suatu tabel yang nilainya boleh dikosongkan (*non mandatory attribute*). Nilai (konstanta) null digunakan untuk menyatakan/mengisi atribut-atribut demikian yang nilainya memang belum siap atau tidak ada. Nilai null tidak sama dengan spasi walaupun pada waktu nilainya ditampilkan sama-sama tidak memperlihatkan apa-apa. Perbedaan tersebut dapat dilihat dari segi makna maupun dari segi representasi fisik. Dari segi makna, pengisian spasi ke suatu atribut berarti atribut tersebut memiliki nilai yaitu spasi, sedangkan pengisian nilai null berarti atribut tersebut belum/tidak memiliki nilai. Dari segi representasi fisik, nilai spasi ekivalen dengan karakter ke-32 dalam tabel ASCII, sedangkan nilai null ekivalen dengan karakter ke-0.

5. Atribut turunan (*derived attribute*)

   Atribut turunan adalah atribut yang nilai-nilainya diperoleh dari pengolahan atau dapat diturunkan dari atribut atau tabel lain yang berhubungan. Atribut turunan dapat dihilangkan dari sebuah tabel karena nilai-nilainya bergantung pada nilai yang ada di atribut lainnya.

II.7.2 *Domain* dan Tipe Data

Untuk keperluan penentuan struktur tiap tabel, maka kita perlu untuk menetapkan tipe data pada setiap atribut (kolom). Penetapan tipe data ini akan berimplikasi pada adanya batas-batas nilai yang mungkin disimpan/diisikan ke setiap atribut (kolom) tersebut. Misalnya, jika kita menetapkan tipe data untuk sebuah atribut (kolom) adalah *integer*, maka kita hanya mungkin untuk menyimpan data angka yang bulat di antara –32.768 hingga 32.767. Kita tidak mungkin untuk memasukkan data di luar batas nilai tersebut seperti data pecahan atau data yang berupa *string* (kumpulan huruf).

Istilah *domain* memiliki banyak kesamaan pengertian dengan fungsi tipe data, tetapi tipe data lebih merujuk pada kemampuan penyimpanan data yang

mungkin bagi suatu atribut secara fisik tanpa melihat layak atau tidaknya data tersebut bila dilihat dari kenyataan pemakaiannya. Sementara itu, *domain* nilai lebih ditekankan pada batas-batas nilai yang diperbolehkan bagi suatu atribut dilihat dari kenyataan yang ada.

Pada saat kita sedang melakukan perancangan basis data, yang perlu kita lihat dan pertimbangkan hanyalah *domain* nilai dari setiap atribut. Dan penetapan tipe data bagi suatu atribut baru akan relevan dan penting untuk diperhitungkan pada saat implementasi basis data.

II.7.3 Ketergantungan Fungsional (*Functional Dependency*)

Misalkan kita memiliki sebuah tabel X yang memiliki paling sedikit dua buah atribut, yaitu A dan B, maka kita dapat membuat suatu notasi seperti berikut ini : A $\rightarrow$ B, yang berarti A secara fungsional menentukan B atau B secara fungsional tergantung pada A jika dan hanya jika untuk setiap kumpulan baris data (*row*) yang ada di tabel X, pasti ada dua baris data (*row*) di tabel X dengan nilai untuk A yang sama, maka nilai untuk B pasti juga sama.

II.7.4 Normalisasi dengan Ketergantungan Fungsional

Sebuah basis data dapat dikatakan baik jika setiap tabel yang menjadi unsur pembentuk basis data tersebut juga telah berada dalam keadaan baik atau normal. Sebuah tabel dapat dikatagorikan baik (efisien) atau normal jika telah memenuhi tiga kriteria, yaitu :

1. Jika ada dekomposisi (penguraian) tabel, maka dekomposisinya harus dijamin aman (*Lossless-Join Decomposition*).

   Dekomposisi adalah salah satu cara untuk mendapatkan sebuah tabel yang baik, tetapi jika tidak hati-hati, maka cara ini dapat menyebabkan kesalahan. Suatu dekomposisi dapat dikatakan aman (*lossless-join decomposition*) jika tabel-tabel hasil dekomposisi kita gabungkan kembali maka tabel-tabel tersebut dapat menghasilkan tabel awal sebelum didekomposisi.

2. Terpeliharanya ketergantungan fungsional pada saat perubahan data (*Dependency Preservation*).

   Ketika kita melakukan perubahan data, maka kita harus bisa menjamin agar perubahan tersebut tidak menghasilkan inkonsistensi data yang mengakibatkan ketergantungan fungsional yang sudah benar menjadi tidak terpenuhi. Tetapi dalam upaya untuk memelihara ketergantungan fungsional yang ada untuk tetap terpenuhi maka prosesnya harus dapat dilakukan dengan efisien.

3. Tidak melanggar *Boyce-Code Normal Form* (BCNF).

   Sebuah tabel dikatakan berada dalam *Boyce-Code Normal Form* (BCNF) jika untuk semua ketergantungan fungsional dengan notasi $X \rightarrow Y$, maka X harus merupakan *superkey* pada tabel tersebut. Jika tidak demikian, maka tabel tersebut harus didekomposisi berdasarkan ketergantungan fungsional yang ada sehingga X menjadi *superkey* dari tabel-tabel hasil dekomposisi.

Jika kriteria ketiga (BCNF) tidak dapat terpenuhi, maka paling tidak, tabel tersebut tidak melanggar bentuk normal tahap ketiga (*3nd Normal Form/3NF*). Sebuah tabel dikatakan berada dalam bentuk normal tahap ketiga (3NF) jika untuk setiap ketergantungan fungsional dengan notasi $X \rightarrow A$, dimana A mewakili semua atribut tunggal di dalam tabel yang tidak ada di dalam X, maka:

- X haruslah *superkey* pada tabel tersebut
- Atau A merupakan bagian dari *primary key* pada tabel tersebut

Ketiga kriteria diatas merupakan kriteria minimal bagi sebuah tabel untuk dapat disebut sebagai tabel yang normal/efisien. Selain kriteria-kriteria tersebut, kita juga dapat menerapkan kriteria-kriteria lain yang juga tercakup dalam kerangka normalisasi, walaupun bukan merupakan kriteria utama. Kriteria-kriteria lain tersebut adalah :

- Bentuk Normal tahap Pertama (*1st Normal Form/1NF*)

  Bentuk normal tahap pertama (1NF) terpenuhi jika sebuah tabel tidak memiliki atribut bernilai banyak (*multivalued attribute*) atau lebih dari satu atribut dengan *domain* nilai yang sama.

- Bentuk Normal tahap Kedua (*2nd Normal Form/2NF*)

Bentuk normal tahap kedua (2NF) terpenuhi jika pada sebuah tabel, semua atribut yang tidak termasuk dalam *primary key* memiliki ketergantungan fungsional pada *primary key* secara utuh. Sebuah tabel dikatakan tidak memenuhi 2NF jika ketergantungannya hanya bersifat parsial (hanya bergantung pada sebagian dari *primary key*).

- Bentuk Normal tahap Keempat (*4th Normal Form/4NF*)

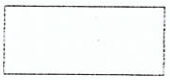
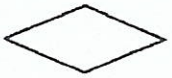
  Bentuk normal tahap keempat (4NF) berkaitan dengan sifat ketergantungan banyak nilai (*multivalued dependency*) pada suatu tabel yang merupakan pengembangan dari ketergantungan fungsional.

- Bentuk Normal tahap Kelima (*5th Normal Form/5NF*)

  Bentuk normal tahap kelima (5NF) atau sering juga disebut *Project-Join Normal Form*/PJNF berhubungan dengan ketergantungan relasi antar tabel (*join dependency*).

II.8 *Entity Relationship Diagram* (ERD)

Entity Relationship model merupakan model data konseptual tingkat tinggi yang dikembangkan oleh Chen (1976) untuk memfasilitasi desain basis data. Model data konseptual adalah sekumpulan konsep yang mendeskripsikan struktur dari basis data dan transaksi yang dilakukan terhadap data tersebut. Simbolo-simbol yang digunakan dalam ERD :

Simbol	Arti
	*Attribute*
	*Entity*
	*Relationship*

Gambar 2.5 Simbol *Entity Relationship Diagram* (ERD)

Model ER terdiri dari tiga komponen :

1. *Entity types*

   Entity type adalah objek atau konsep yang diidentifikasi oleh suatu enterprise yang berdiri sendiri. *Entity types* merepresentasikan sekumpulan "objek" di dalam "dunia nyata" yang memiliki sifat yang sama. Semua *entity type* dapat berdiri sendiri dan dapat berupa objek yang nyata berwujud atau objek yang nyata secara konsep. *Entity* adalah contoh (*instance*) dari *entity type* yang diidentifikasikan secara unik.

   Entity type dapat dikategorikan menjadi dua jenis yaitu *weak entity type* dan *strong entity type*. *Weak entity type* (disebut juga sebagai *child*, *dependent*, atau *subordinate entities*) adalah tipe *entity* yang keberadaannya diakibatkan atau tergantung dari keberadaan *entity type* lainnya. Jika *entity type* lainnya tidak ada maka *weak entity type* ini tidak akan muncul. *Weak entity type* digambarkan dengan sebuah kotak yang garis kotaknya adalah dua. *Strong entity type* (disebut juga sebagai *parent*, *owner*, dan *dominant entities*) adalah tipe *entity* yang keberadaannya tidak tergantung dari keberadaan *entity type* lainnya.

UNIT PELAKSANA TEKNIS PERPUSTAKAAN UNIKOM

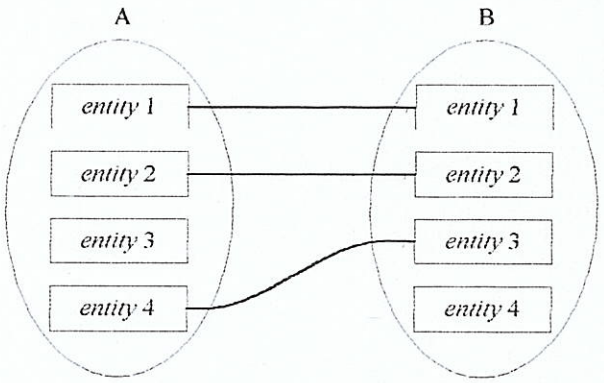
2. *Relationship types*

   Relationship type adalah hubungan antara *entity type*. *Relationship type* adalah sekelompok hubungan antara dua atau lebih *entity types*. Masing-masing *relationship* memiliki nama yang menggambarkan fungsi hubungan tersebut. *Relationship* adalah hubungan dari *entities* dimana hubungan termasuk satu *entity* dari masing-masing *entity type*. Setiap hubungan yang unik yang muncul dari suatu *relationship type* dapat disebut sebagai *relationship*. Secara diagram, *relationship* digambarkan sebagai *diamond* yang diberi label nama hubungannya. *Diamond* dengan dua garis pembentuk menggambarkan hubungan antara *entity* lemah dengan *entity* kuat.

   Kardinalitas/derajat relasi menunjukkan jumlah maksimum *entity* yang dapat berelasi dengan *entity* pada himpunan *entity* yang lain. Kardinalitas relasi yang terjadi di antara dua himpunan *entity* dapat berupa :

   - Satu ke Satu (*One to One*)

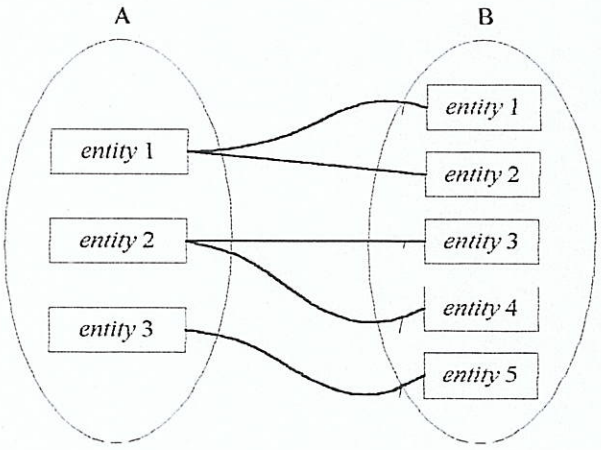
Hubungan satu ke satu berarti setiap *entity* pada himpunan *entity* A berhubungan dengan paling banyak dengan satu *entity* pada himpunan *entity* B, dan begitu juga sebaliknya.



Gambar 2.6 Kardinalitas Relasi Satu ke Satu

- Satu ke Banyak (*One to Many*)

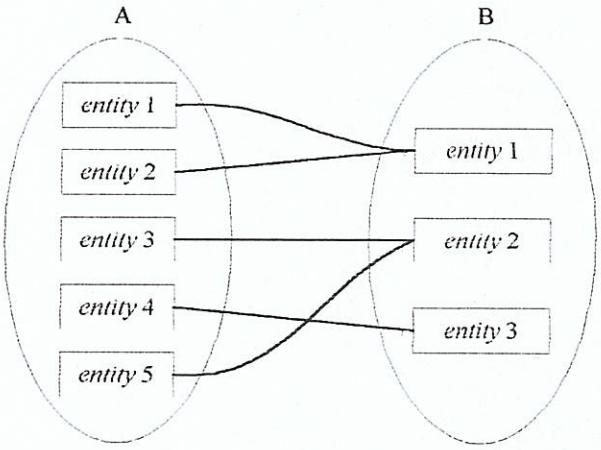
  Hubungan satu ke banyak berarti setiap *entity* pada himpunan *entity* A dapat berhubungan dengan banyak *entity* pada himpunan *entity* B, tetapi tidak sebaliknya, dimana setiap *entity* pada himpunan *entity* B berhubungan paling banyak dengan satu *entity* pada himpunan *entity* A.



Gambar 2.7 Kardinalitas Relasi Satu ke Banyak

- Banyak ke Satu (*Many to One*)

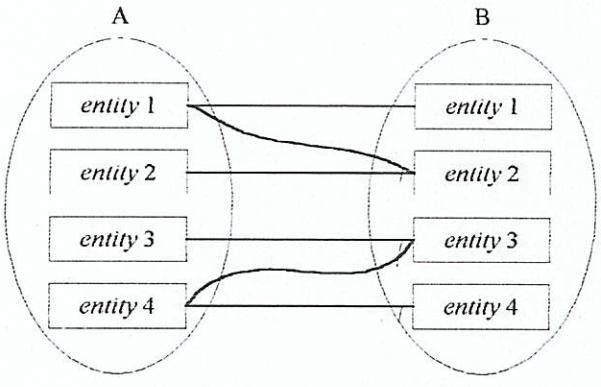
  Hubungan banyak ke satu berarti setiap *entity* pada himpunan *entity* A berhubungan dengan paling banyak dengan satu *entity* pada himpunan *entity* B, tetapi tidak sebaliknya, dimana setiap *entity* pada himpunan *entity* A berhubungan paling banyak dengan satu *entity* pada himpunan *entity* B.



Gambar 2.8 Kardinalitas Relasi Banyak ke Satu

- Banyak ke Banyak (Many to Many)

  Hubungan banyak ke banyak berarti setiap *entity* pada himpunan *entity* A dapat berhubungan dengan banyak *entity* pada himpunan *entity* B, dan demikian juga sebaliknya.



Gambar 2.9 Kardinalitas Relasi Banyak ke Banyak

3. *Attributes*

Attribute adalah sesuatu yang menjadi milik (properti) dari *entity* ataupun *relationship type*. Atribut dari *entity type* menyimpan informasi atau nilai yang mendeskripsikan *entity* tersebut. *Relationship* yang menghubungkan antar *entity* juga dapat memiliki atribut. Atribut digambarkan dengan lambang elips dan diberi label nama atribut di dalamnya. Garis pembentuk elips adalah garis putus-putus jika atribut adalah atribut *derivatif*, sedangkan garis pembentuk adalah dua garis jika atribut adalah atribut *multi-valued*.

Tahap-tahap pembuatan ERD :

1. Mengidentifikasi dan menetapkan seluruh himpunan *entity* yang akan terlibat.
2. Menentukan atribut-atribut *key* dari masing-masing himpunan *entity*.
3. Mengidentifikasi dan menetapkan seluruh himpunan relasi di antara himpunan *entity*-himpunan *entity* yang ada beserta *foreign key* nya.
4. Menentukan derajat/kardinalitas relasi untuk setiap himpunan relasi.
5. Melengkapi himpunan *entity* dan himpunan relasi dengan atribut-atribut deskriptif (*non key*).


UNIVERSITAS KOMPUTER INDONESIA
UNIKOM