

BAB II TINJAUAN PUSTAKA

2.1 Konsep Dasar Sistem

2.1.1 Pengertian Sistem

Sebelum mempelajari mengenai sistem lebih jauh, sebaiknya kita terlebih dahulu mengetahui definisi sistem itu sendiri. Berikut beberapa definisi sistem dari pendapat pakar antara lain:

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu (Jerry Fith Gerald dalam buku Indrajit, 2000).

Sistem merupakan seperangkat unsur yang saling terikat dalam suatu antar relasi diantara unsur-unsur tersebut dengan lingkungan (Ludwig Von Bartalanfy dalam buku Indrajit, 2000).

Sistem adalah suatu kumpulan kesatuan dan perangkat hubungan satu sama lain (Anatol Raporot dalam buku Indrajit, 2000).

Sistem adalah setiap kesatuan secara konseptual atau fisik yang terdiri dari bagian-bagian dalam keadaan saling tergantung satu sama lainnya (L. Ackoff dalam buku Indrajit, 2000).

Kata “sistem” mengandung arti dari komponen-komponen yang memiliki unsur keterkaitan antara satu dan lainnya (Indrajit, 2000).

2.1.2 Karakteristik Sistem

Sistem memiliki beberapa karakteristik tertentu, diantaranya adalah Komponen (*Component*), Batas Sistem (*Boundary*), Lingkungan Luar Sistem (*Environment*), Penghubung Sistem (*Interface*), Masukan sistem (*Input*), Keluaran, Sistem (*Output*), Pengolah Sistem (*Process*), dan Tujuan Sistem (*Goal*).

Komponen (*Component*) adalah suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, bekerja sama membentuk satu kesatuan. Komponen-komponen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap sistem tidak peduli betapapun kecilnya, selalu mengandung komponen-komponen atau subsistem-subsistem.

Batas Sistem (*Boundary*) merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan, karena dengan batas sistem ini fungsi dan tugas dari subsistem yang satu dengan lainnya berbeda tetapi tetap saling berinteraksi. Batas suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut..

Lingkungan Luar Sistem (*Environment*) segala sesuatu diluar dari batas sistem yang mempengaruhi operasi dari suatu sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan atau merugikan. Lingkungan luar yang menguntungkan harus dipelihara dan dijaga agar tidak hilang pengaruhnya, sedangkan lingkungan luar yang merugikan harus dimusnahkan dikendalikan agar tidak mengganggu operasi sistem.

Penghubung Sistem (*Interface*) merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Untuk membentuk satu kesatuan, sehingga sumber-sumber daya mengalir dari subsistem yang satu ke subsistem yang lainnya. Dengan kata lain output dari suatu subsistem akan menjadi input dari subsistem yang lainnya.

Masukan sistem (*Input*) merupakan energi yang dimasukkan ke dalam sistem. Masukan dapat berupa Masukan Perawatan (*Maintenance Input*) adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Masukan Sinyal (*Signal Input*) adalah energi yang diproses untuk didapatkan keluaran.

Keluaran Sistem (*Output*) merupakan hasil dari energi yang diolah oleh sistem. Pengolah Sistem (*Process*) merupakan bagian yang memproses masukan untuk menjadi keluaran yang diinginkan. Tujuan Sistem (*Goal*) ialah setiap sistem pasti mempunyai tujuan ataupun sasaran yang mempengaruhi input yang dibutuhkan dan output yang dihasilkan. Dengan kata lain suatu sistem akan dikatakan berhasil kalau pengoperasian sistem itu mengenai sasaran atau tujuannya. Sistem yang tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya (Jogianto, 1995).

2.2.3 Klasifikasi Sistem

Sistem dapat diklasifikasi dari beberapa sudut pandang, diantaranya sebagai berikut:

Sistem abstrak dan sistem fisik. Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Misalnya sistem teologia, yaitu sistem yang berupa pemikiran-pemikiran hubungan antara manusia dengan Tuhan. Sistem fisik merupakan sistem yang ada secara fisik misalnya sistem komputer, sistem akuntansi, dan sistem produksi.

Sistem alamiah (*natural system*) dan sistem buatan manusia (*human made system*). Sistem alamiah adalah sistem yang terjadi melalui proses alam, misalnya sistem perputaran bumi. Sistem buatan manusia adalah sistem yang dirancang oleh manusia. Sistem buatan manusia yang melibatkan interaksi antara manusia dengan mesin disebut dengan *human machine system* atau ada yang menyebut dengan *man machine system*, karena menyangkut penggunaan komputer yang berinteraksi dengan manusia.

Sistem tertentu (*deterministic system*) dan sistem tak tentu (*probabilistic system*). Sistem tertentu beroperasi tertentu dengan tingkah laku yang sudah dapat diprediksi. Interaksi diantara bagian-bagiannya dapat dideteksi dengan pasti, sehingga keluaran dari sistem dapat diramalkan. Sistem komputer adalah contoh dari sistem tertentu yang tingkah lakunya dapat dipastikan berdasarkan program-program yang dijalankan. Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

Sistem tertutup (*close system*) dan sistem terbuka (*open system*). Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh oleh lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak diluarnya. Secara teoritis sistem tertutup ada, tetapi kenyataannya tidak ada sistem yang benar-benar tertutup yang ada hanyalah *relatively closed system* (secara relatif tertutup, tidak benar-benar tertutup). Sedangkan sistem terbuka adalah sistem yang berhubungan dan terpengaruh oleh lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk lingkungan luar atau subsistem yang lainnya. Karena sistem sifatnya

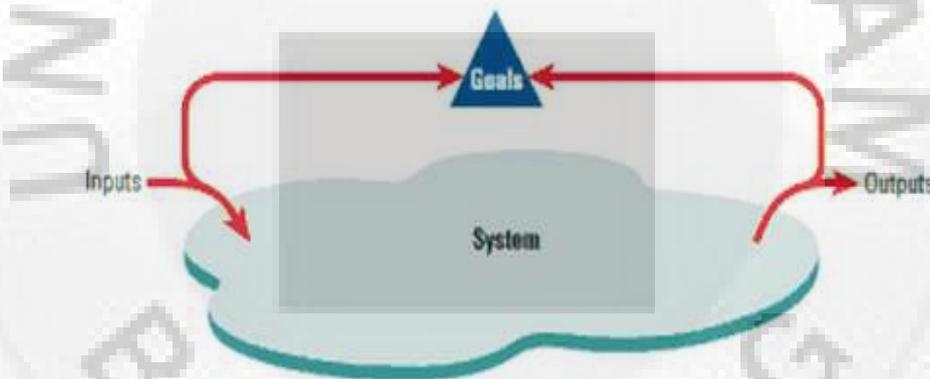
terbuka dan terpengaruh oleh lingkungan luarnya, maka suatu sistem harus mempunyai sistem pengendalian yang baik. Sistem-sistem yang baik harus dirancang sedemikian rupa, sehingga secara relatif tertutup karena sistem tertutup akan bekerja secara otomatis dan terbuka untuk pengaruh yang baik saja (Jogianto, 1995).

2.2.4 Pemodelan Sistem

2.2.4.1 Organisasi dalam sistem

Organisasi adalah suatu sistem besar yang terdiri dari sub-sub sistem yang terhubung satu sama lainnya. Sub-sub sistem tersebut saling mempengaruhi oleh tiga level keputusan manajemen (Operasional, *middle* manajemen, dan strategi manajemen). Budaya perusahaan mempengaruhi bagaimana orang bekerja didalam sub-sistem yang saling berhubungan.

Berikut Gambar 2.1 interaksi sistem dengan lingkungan:



Gambar 2. 1 Interaksi sistem dengan lingkungan

Sumber: K.Kendall dan J. Kendall (2013)

Suatu sistem berhubungan dengan lingkungannya melalui Input dan Output. Output sistem digunakan sebagai *feedback* untuk membandingkan kinerja dengan tujuan.

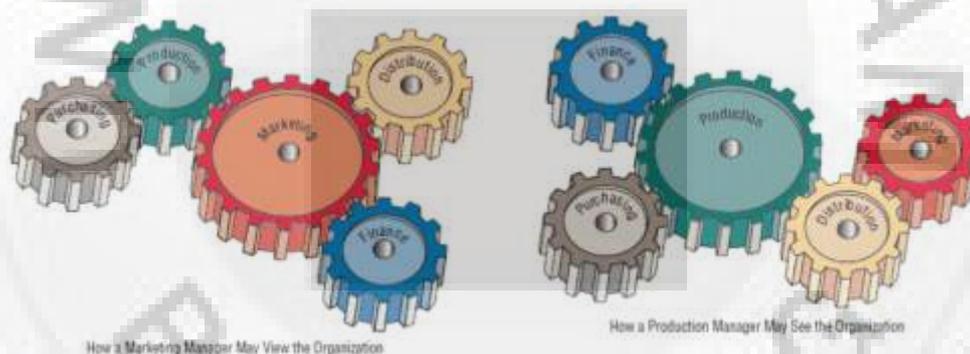
Pada perspektif sistem setiap anggota sub-sistem harus menyadari bahwa pekerjaan mereka terkait dengan sub-sub sistem lainnya. Berikut karakteristik sub-sistem pada Gambar 2.2.



Gambar 2. 2 Keterkaitan sub-sistem

Sumber: K.Kendall dan J. Kendall (2013)

Permasalahan timbul ketika masing-masing manajer melihat bahwa sub-sistem (departemen) mereka adalah paling penting dan menjadi penggerak bagi sub sistem lainnya. Berikut Gambar 2.3 pergerakan sub-sistem.



Gambar 2. 3 Pergerakan sub-sistem

Sumber: K.Kendall dan J. Kendall (2013)

2.2.4.2 Enterprise Resource Planning (ERP)

Enterprise System yang sering disebut sebagai *Enterprise Resource Planning (ERP)* sistem adalah istilah yang menjelaskan Sistem Informasi terintegrasi. Software ERP membantu aliran informasi yang melintasi are fungsional didalam organisasi. Implementasi solusi ERP asalah sulit karena untuk menganalisa sistem yang berjalan saat ini dan disesuaikan dengan model ERP ke sistem tersebut (K.Kendall dan J. Kendall, 2013).

2.3 Konsep Dasar Informasi

Menurut Robert N. Anthony dan John Dearden. Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya.

2.3.1 Siklus Informasi

Data merupakan bentuk yang masih mentah yang belum dapat bercerita banyak, sehingga perlu diolah lebih lanjut. Data diolah melalui suatu model untuk dihasilkan informasi. Data yang diolah untuk menghasilkan informasi menggunakan suatu model proses tertentu.

Menurut Jogianto (1999), data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lainnya yang akan membuat sejumlah data kembali. Data tersebut akan ditanggap sebagai input, diproses kembali lewat suatu model dan membentuk suatu siklus. Menurut John Burch disebut dengan siklus informasi (*information cycle*) atau siklus ini disebut juga dengan siklus pengolahan data (*data processing cycles*). Pengolahan data menjadi informasi itu merupakan suatu siklus, yang terdiri dari beberapa tahap seperti pada Gambar 2.4, adapun siklus tersebut dapat dijelaskan sebagai berikut:

1. Pengumpulan data.

Pada tahap ini dilakukan suatu proses pengumpulan data yang asli dengan cara tertentu, seperti sampling, data transaksi, data warehouse, dan lain sebagainya yang biasanya merupakan proses pencatatan data ke dalam suatu file.

2. *Input.*

Tahap ini merupakan suatu proses pemasukan data dan prosedur pengolahan data ke dalam komputer melalui alat input seperti keyboard. Prosedur pengolahan data itu merupakan urutan langkah untuk mengolah data yang ditulis dalam suatu bahasa pemrograman yang disebut program.

3. Pengolahan data.

Tahap ini merupakan tahap di mana data diolah sesuai dengan prosedur yang telah dimasukkan. Kegiatan pengolahan data ini meliputi pengumpulan

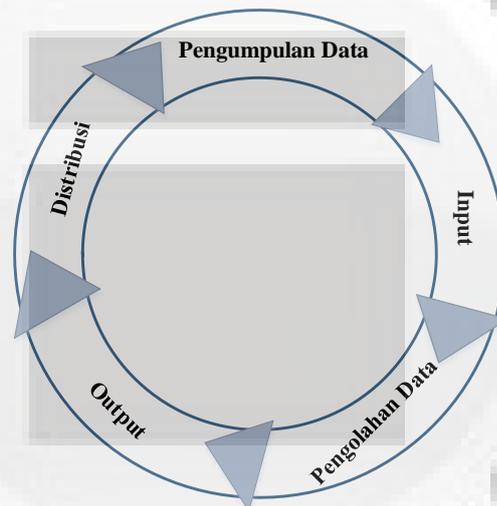
data, klasifikasi (pengelompokkan), kalkulasi, pengurutan, penggabungan, peringkasan baik dalam bentuk tabel maupun grafik, penyimpanan dan pembacaan data dari tempat penyimpanan data.

4. *Output.*

Hasil pengolahan data akan ditampilkan pada alat output seperti monitor dan printer sebagai informasi.

5. *Distribusi.*

Setelah proses pengolahan data dilakukan, maka informasi yang dihasilkan harus segera didistribusikan. Proses pendistribusian ini tidak boleh terlambat dan harus diberikan kepada yang berkepentingan, sebab hasil pengolahan tersebut akan menjadi bahan pertimbangan dalam pengambilan keputusan atau menjadi data dalam pengolahan data selanjutnya.



Gambar 2. 4 Siklus Informasi

2.3.2 Kualitas Informasi

Kualitas dari suatu informasi tergantung dari tiga hal, yaitu informasi harus akurat, tepat pada waktunya dan relevan, diantaranya:

1. Akurat, berarti informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi sampai ke penerima informasi kemungkinan banyak terjadi gangguan yang dapat merubah atau merusak informasi tersebut.

2. Tepat pada waktunya, berarti informasi yang datang pada penerima tidak boleh terlambat.
3. Relevan, berarti informasi tersebut mempunyai manfaat untuk pemakainya

2.3.3 Nilai Informasi

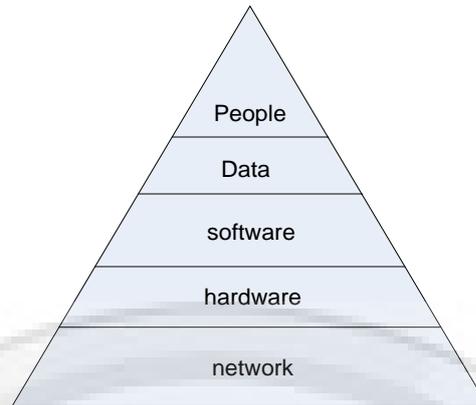
Nilai informasi ditentukan dari dua hal, yaitu manfaat dan biaya mendapatkannya. Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya.

2.4 Konsep Dasar Sistem Informasi

Sistem informasi merupakan suatu kumpulan dari komponen-komponen dalam perusahaan atau organisasi yang berhubungan dengan proses penciptaan dan pengaliran informasi (Indrajit, 2000).

Teknologi informasi hanya merupakan hanya salah satu komponen kecil saja dalam format perusahaan. Komponen-komponen lainnya adalah: proses dan prosedur, struktur organisasi, sumber daya manusia, produk, pelanggan, *supplier*, rekanan dan sebagainya. Keandalan suatu sistem informasi dalam perusahaan atau organisasi terletak pada keterkaitan antar komponen yang ada, sehingga dapat dihasilkan dan dialirkan suatu informasi yang berguna (akurat, terpercaya, detail, cepat, relevan, dan sebagainya) untuk lembaga yang bersangkutan.

Suatu sistem informasi merupakan kombinasi terorganisir dari orang-orang (*people*), perangkat keras (*hardware*), perangkat lunak (*software*), jaringan komunikasi (*communication network*), dan sumber data (*data resource*) yang mengumpulkan, mentransformasikan, dan menyebarkan informasi dalam suatu organisasi (O'Brien, 1998). Orang-orang mengandalkan sistem informasi untuk berkomunikasi satu sama lain menggunakan serangkaian peralatan fisik (*hardware*), intruksi-intruksi dan prosedur-prosedur pemrosesan informasi (*software*), saluran komunikasi (*network*), dan data yang tersimpan (*data resource*) sejak munculnya peradaban. Berikut komponen sistem informasi pada Gambar 2.5.



Gambar 2. 5 Komponen sistem informasi
 Sumber: O'Brien (1998)

2.5 Pengertian Pengembangan Sistem

Pengembangan sistem (*system development*) dapat berarti menyusun sistem yang baru untuk menggantikan sistem yang lama secara keseluruhan atau memperbaiki sistem yang telah ada. Sistem yang lama perlu diperhatikan atau diganti disebabkan karena beberapa hal, yaitu sebagai berikut:

1. Adanya permasalahan-permasalahan (*problems*) yang timbul di sistem yang lama.
2. Untuk meraih kesempatan-kesempatan (*opportunities*)
3. Adanya intruksi-intruksi (*directivies*)

Karena adanya permasalahan, kesempatan atau intruksi, maka sistem yang baru perlu dikembangkan untuk memecahkan permasalahan-permasalahan yang timbul, meraih kesempatan-kesempatan yang ada atau memenuhi intruksi yang diberikan. Berikut Gambar 2.6 Aliran pengembangan sistem.



Gambar 2. 6 Aliran pengembangan sistem

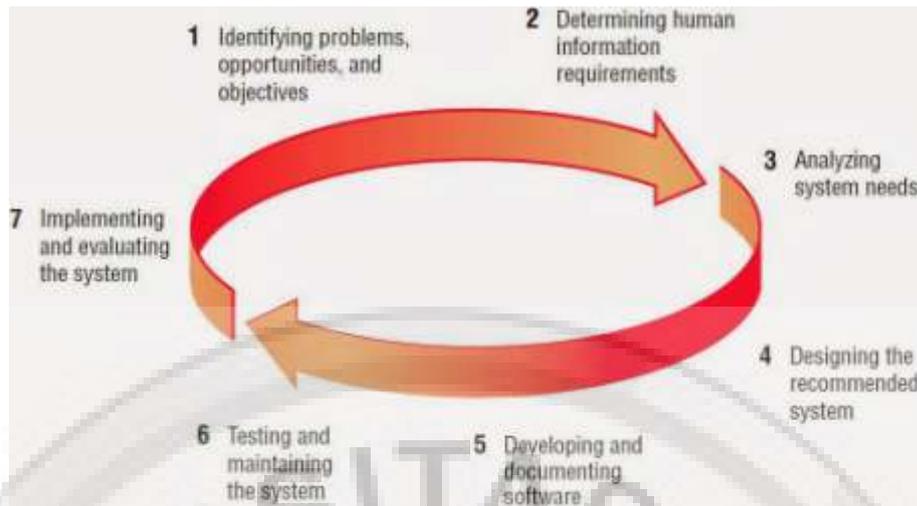
Dengan telah dikembangkannya sistem yang baru, maka diharapkan akan terjadi peningkatan-peningkatan di sistem yang baru (Jogianto, 1999).

2.6 Penjelasan Siklus Hidup Pengembangan Sistem

Pengembangan sistem informasi yang berbasis computer dapat merupakan tugas kompleks yang membutuhkan banyak sumber 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 pertama yaitu, tahap perencanaan sistem. Siklus ini disebut dengan siklus hidup dari pengembangan sistem merupakan suatu langkah didalam tahapan tersebut dalam proses pengembangannya (Jogianto, 1999).

System Development Life Cycle (SDLC) adalah pendekatan bertahap untuk analisis dan desain yang menyatakan bahwa sistem yang terbaik yang dikembangkan melalui penggunaan siklus kegiatan khusus analisis dan pengguna.

Analisis setuju pada persis berapa banyak fase ada dalam SDLC, tapi mereka umumnya memuji pendekatan yang terorganisasinya. Di sini kita telah membagi siklus menjadi tujuh tahap, seperti yang ditunjukkan pada Gambar 2.7. Meskipun setiap fase disajikan discretely, tidak pernah dilakukan sebagai langkah terpisah. Sebaliknya, beberapa kegiatan dapat terjadi secara bersamaan, dan kegiatan dapat diulang (K.Kendall dan J. Kendall, 2013).



Gambar 2. 7 *System Development Life Cycle (SDLC)*

Sumber: K.Kendall dan J. Kendall (2013)

1. Mengidentifikasi Masalah, Peluang, dan Tujuan

Pada fase pertama dari siklus hidup pengembangan sistem, analis yang bersangkutan dengan masalah benar mengidentifikasi, peluang, dan tujuan. Tahap ini sangat penting untuk keberhasilan seluruh proyek, karena tidak ada yang mau buang waktu berikutnya mengatasi masalah yang salah. Tahap pertama mengharuskan analis melihat secara jujur apa yang terjadi dalam bisnis.

Kemudian, bersama-sama dengan anggota organisasi lainnya, analis titik-titik masalah. Seringkali orang lain akan memunculkan masalah ini, dan mereka adalah alasan analis awalnya dipanggil masuk peluang situasi bahwa analis percaya dapat ditingkatkan melalui penggunaan sistem informasi terkomputerisasi. Merebut peluang memungkinkan bisnis untuk mendapatkan keunggulan kompetitif atau menetapkan standar industri.

Mengidentifikasi tujuan juga merupakan komponen penting dari tahap pertama. Analis harus terlebih dahulu mengetahui apa bisnis coba lakukan. Kemudian analis akan dapat melihat apakah beberapa aspek dari aplikasi sistem informasi dapat membantu bisnis mencapai tujuannya dengan mengatasi masalah atau peluang khusus.

Orang-orang yang terlibat dalam tahap pertama adalah pengguna, analis, dan manajer sistem koordinasi proyek. Kegiatan dalam tahap ini terdiri dari wawancara manajemen pengguna, meringkas pengetahuan yang diperoleh,

memperkirakan lingkup proyek, dan mendokumentasikan hasil. Output dari tahap ini adalah laporan kelayakan berisi definisi masalah dan meringkas tujuan.

Manajemen kemudian harus membuat keputusan tentang apakah akan melanjutkan dengan proyek yang diusulkan. Jika kelompok pengguna tidak memiliki dana yang cukup dalam anggaran atau ingin mengatasi masalah yang tidak terkait, atau jika masalah tidak memerlukan sistem komputer, solusi yang berbeda dapat direkomendasikan, dan proyek sistem tidak melanjutkan lebih jauh.

2. Menentukan Kebutuhan Informasi Manusia

Tahap berikutnya analisis kebutuhan informasi manusia adalah bahwa penentuan kebutuhan manusia pengguna yang terlibat, dengan menggunakan berbagai alat untuk memahami bagaimana pengguna berinteraksi dalam konteks kerja dengan sistem informasi mereka saat ini. Analisis akan menggunakan metode interaktif seperti wawancara, sampling dan menyelidiki data keras, dan kuesioner, bersama dengan metode mencolok, seperti perilaku mengamati pengambil keputusan dan lingkungan kantor mereka, dan metode, seperti prototyping mencakup semua.

Analisis sistem perlu mengetahui detail dari fungsi sistem saat ini: siapa (orang-orang yang terlibat), apa (kegiatan usaha), di mana (lingkungan di mana pekerjaan berlangsung), ketika (waktu), dan bagaimana (bagaimana prosedur saat dilakukan) bisnis yang diteliti. Analisis kemudian harus bertanya mengapa bisnis menggunakan sistem saat ini. Mungkin ada alasan yang baik untuk melakukan bisnis dengan menggunakan metode saat ini, dan ini harus dipertimbangkan ketika merancang sistem baru.

Pada akhir tahap ini, analisis harus memahami bagaimana pengguna menyelesaikan pekerjaan mereka saat berinteraksi dengan komputer dan mulai tahu bagaimana membuat sistem baru yang lebih berguna dan bermanfaat. Analisis juga harus tahu bagaimana fungsi bisnis dan memiliki informasi lengkap tentang orang-orang, tujuan, data, dan prosedur yang terlibat.

3. Menganalisis Kebutuhan Sistem

Tahap berikutnya bahwa analisis sistem melakukan melibatkan kebutuhan sistem analisis. Sekali lagi, alat-alat khusus dan teknik membantu analisis membuat penentuan persyaratan. Alat seperti *Data Flow Diagram* (DFD) untuk memetakan

input, proses, dan output fungsi bisnis, atau diagram aktivitas atau diagram sequence untuk menunjukkan urutan kejadian, menggambarkan sistem dalam bentuk terstruktur, grafis. Dari aliran data, urutan, atau diagram lain, kamus data dikembangkan yang berisi daftar semua item data yang digunakan dalam sistem, serta spesifikasi mereka.

Selama fase ini analisis sistem juga menganalisis keputusan terstruktur dibuat. Keputusan terstruktur adalah mereka yang kondisi, kondisi alternatif, tindakan, dan aturan tindakan dapat ditentukan. Ada tiga metode utama untuk analisis keputusan terstruktur: English terstruktur, tabel keputusan, dan pohon keputusan.

Pada titik ini dalam *System Development Life Cycle (SDLC)*, analisis sistem mempersiapkan proposal sistem yang merangkum apa yang telah mengetahui tentang pengguna, kegunaan, dan kegunaan dari sistem saat ini; menyediakan analisis biaya-manfaat alternatif; dan membuat rekomendasi tentang apa (jika ada) harus dilakukan. Jika salah satu dari rekomendasi yang diterima oleh manajemen, analisis hasil disepanjang arah itu. Setiap masalah sistem yang unik, dan tidak pernah ada hanya satu solusi yang tepat. Cara di mana rekomendasi atau solusi dirumuskan tergantung pada kualitas individu dan pelatihan profesional dari masing-masing analisis dan interaksi analisis dengan pengguna dalam konteks lingkungan kerja mereka.

4. Merancang Sistem Direkomendasikan

Pada tahap desain *System Development Life Cycle (SDLC)*, analisis sistem menggunakan informasi yang dikumpulkan sebelumnya untuk mencapai desain logis dari sistem informasi. Analisis desain prosedur bagi pengguna untuk membantu mereka secara akurat memasukkan data sehingga data akan masuk ke sistem informasi sudah benar. Selain itu, analisis menyediakan bagi pengguna untuk menyelesaikan masukan yang efektif untuk sistem informasi dengan menggunakan teknik bentuk yang baik dan halaman Web atau desain layar.

Bagian dari desain logis dari sistem informasi merancang *human-computer interaction (HCI)*. Antarmuka menghubungkan pengguna dengan sistem dan dengan demikian sangat penting. *User interface* dirancang dengan bantuan pengguna untuk memastikan bahwa sistem ini terdengar, terbaca, dan aman, serta menarik dan menyenangkan untuk digunakan. Contoh antarmuka pengguna fisik

termasuk *keyboard* (untuk mengetik dalam pertanyaan dan jawaban), menu pada layar (untuk memperoleh perintah pengguna), dan berbagai antarmuka pengguna grafis yang menggunakan mouse atau layar sentuh.

Tahap desain juga termasuk *database* merancang yang akan menyimpan banyak data yang dibutuhkan oleh pengambil keputusan dalam organisasi. Pengguna mendapatkan keuntungan dari database terorganisir dengan baik yang logis untuk mereka dan sesuai dengan cara mereka melihat pekerjaan mereka. Dalam fase ini analis juga bekerja dengan pengguna untuk merancang output (baik di layar atau dicetak) yang memenuhi kebutuhan informasi mereka.

Akhirnya, analis harus merancang pengendalian dan prosedur cadangan untuk melindungi sistem dan data, dan untuk menghasilkan paket spesifikasi program programmer. Setiap paket harus mengandung *input* dan *output layout*, spesifikasi berkas, dan rincian pengolahan; mungkin juga termasuk pohon keputusan atau tabel, UML atau diagram aliran data, dan nama-nama dan fungsi dari kode *prewritten* yang baik tertulis di rumah atau menggunakan kode atau perpustakaan kelas yang lain.

5. Mengembangkan dan Mendokumentasikan *Software*

Pada tahap kelima dari *System Development Life Cycle (SDLC)*, analis bekerja dengan programmer untuk mengembangkan perangkat lunak asli yang diperlukan. Selama fase ini analis bekerja dengan pengguna untuk mengembangkan dokumentasi yang efektif untuk perangkat lunak, termasuk manual prosedur, bantuan online, dan situs-situs Web yang menampilkan *Frequently Asked Questions (FAQ)*, pada *Read Me files* dikirimkan dengan perangkat lunak baru. Karena pengguna yang terlibat dari awal, dokumentasi fase harus menjawab pertanyaan-pertanyaan mereka telah mengangkat dan dipecahkan bersama-sama dengan analis. Dokumentasi memberitahu pengguna bagaimana menggunakan perangkat lunak dan apa yang harus dilakukan jika terjadi masalah *software*.

Programmer memiliki peran penting dalam fase ini karena mereka merancang, kode, dan menghapus kesalahan sintaksis dari program komputer. Untuk memastikan kualitas, programmer dapat melakukan baik desain atau

walkthrough kode, menjelaskan bagian-bagian rumit dari program untuk tim programmer lain.

6. Pengujian dan Mempertahankan Sistem

Sebelum sistem informasi dapat digunakan, itu harus diuji. Hal ini jauh lebih murah untuk menangkap masalah sebelum sistem tersebut ditandatangani ke pengguna. Beberapa pengujian selesai oleh programmer sendiri, sebagian oleh analis sistem dalam hubungannya dengan programmer. Serangkaian tes untuk menunjukkan masalah dijalankan pertama dengan data sampel dan akhirnya dengan data aktual dari sistem saat ini.

Seringkali rencana uji dibuat di awal *System Development Life Cycle (SDLC)* dan diperhalus sebagai proyek berlangsung. Pemeliharaan sistem dan dokumentasinya dimulai pada fase ini dan dilakukan secara rutin sepanjang hidup sistem informasi. Sebagian besar pekerjaan rutin programmer terdiri dari pemeliharaan, dan bisnis menghabiskan banyak uang untuk perawatan. Beberapa perawatan, seperti update program, dapat dilakukan secara otomatis melalui situs penjual di Web. Banyak prosedur yang sistematis analis mempekerjakan seluruh *System Development Life Cycle (SDLC)* dapat membantu memastikan perawatan yang disimpan ke minimum.

7. Mengimplementasikan dan Mengevaluasi Sistem

Pada fase terakhir ini pengembangan sistem, analis membantu menerapkan sistem informasi. Fase ini melibatkan pengguna pelatihan untuk menangani sistem. Vendor melakukan beberapa pelatihan, tetapi pengawasan pelatihan adalah tanggung jawab analis sistem. Selain itu, analis perlu merencanakan untuk kelancaran konversi dari sistem lama ke yang baru. Proses ini meliputi mengkonversi file dari format lama ke yang baru, atau membangun database, memasang peralatan, dan membawa sistem baru ke dalam produksi. Evaluasi disertakan sebagai bagian dari tahap akhir ini *System Development Life Cycle (SDLC)* sebagian besar untuk kepentingan diskusi. Sebenarnya, evaluasi berlangsung selama setiap fase. Kriteria Akey yang harus dipenuhi adalah apakah pengguna yang dituju memang menggunakan sistem. Perlu dicatat bahwa sistem kerja sering siklus. Ketika seorang analis selesai satu tahap pengembangan sistem

dan dilanjutkan dengan berikutnya, penemuan masalah dapat memaksa analisis untuk kembali ke fase sebelumnya dan memodifikasi pekerjaan dilakukan di sana.

2.7 Pengertian Perancangan Sistem

Dalam mendefinisikan Desain Sistem menurut Robert J. Verjello / John Reuter III, John Burch & Gary Grudnitsk dan George M. Scott, memiliki pengertian-pengertian menurut masing-masing diantaranya adalah:

Tahap setelah analisis dari siklus pengembangan sistem: pendefinisian dari kebutuhan-kebutuhan fungsional dan persiapan untuk rancang bangun implementasi; menggambarkan bagaimana suatu sistem dibentuk (Robert J. Verjello / John Reuter III dalam buku Jogianto, 1999).

Desain sistem dapat didefinisikan sebagai penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam suatu kesatuan yang utuh dan berfungsi (John Burch & Gary Grudnitsk dalam buku Jogianto, 1999).

Desain sistem menentukan bagaimana suatu sistem akan menyelesaikan apa yang mesti diselesaikan; tahap ini menyangkut konfigurasi dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem sehingga setelah instalasi dari sistem akan benar – benar memuaskan rancang bangun yang telah ditetapkan pada akhir tahap analisis sistem (George M. Scott dalam buku Jogianto, 1999).

Perancangan Sistem adalah menentukan bagaimana mencapai sasaran yang ditetapkan yang melibatkan pembentukan (*configuring*) perangkat lunak dan komponen perangkat keras sistem dimana setelah pemasangan sistem akan memenuhi spesifikasi yang dibuat pada akhir fase analisis sistem (George M.Scot dalam buku Jogianto, 1999).

2.8 Tujuan Perancangan Sistem

Menurut Jogianto (1999), Tahap desain Sistem memiliki dua maksud dan tujuan utama, diantaranya adalah:

1. Untuk memenuhi kebutuhan kepada pemakai sistem

2. Untuk memberikan gambaran yang jelas dan rancangan yang lengkap kepada pemrogram komputer dan ahli-ahli teknik lainnya yang terlibat.

Untuk mencapai tujuan ini, analisis sistem harus dapat mencapai sasaran-sasaran sebagai berikut:

1. Desain sistem harus berguna, mudah dipahami dan nantinya mudah digunakan.
2. Desain sistem harus dapat mendukung tujuan utama perusahaan sesuai dengan yang telah didefinisikan pada tahap perencanaan sistem yang dilanjutkan pada tahap analisis sistem.
3. Desain sistem harus efisien dan efektif untuk dapat mendukung pengolahan transaksi.
4. Desain sistem harus dapat mempersiapkan rancang bangun yang terinci untuk masing-masing komponen dari sistem informasi.

2.9 Alat Dalam Perancangan Sistem

Alat-alat (*tool*) yang digunakan dalam suatu metodologi umumnya berupa suatu gambar atau diagram atau grafik, misalnya *Data Flow Diagram* (DFD) dan *Integration DEFinition language 0* (IDEF0). Penggunaan diagram atau gambar ini dipandang lebih mengenal dan lebih dimengerti. Alat-alat yang digunakan juga ada yang tidak berupa gambar atau grafik seperti misalnya data *dictionary*.

2.9.1 Sistem Design *Data Flow Diagram* (DFD)

Menurut Jogiyanto (1999) DFD adalah suatu cara untuk menggambarkan model aliran data. Dari mana data berasal, proses apa yang dilalui, apa yang dihasilkan dan kemudian di mana data akan disimpan.

2.9.1.1 Definisi *Data Flow Diagram* (DFD)

- ❖ Alat pembuatan model yang memungkinkan profesional sistem, untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi.

- ❖ Alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program.
- ❖ Suatu diagram yang menggunakan notasi-notasi untuk menggambarkan arus dari data sistem, yang penggunaannya sangat membantu untuk memahami sistem secara logika, tersruktur dan jelas
- ❖ DFD ini sering disebut juga dengan nama *Bubble chart*, *Bubble diagram*, model proses, diagram alur kerja, atau model fungsi.

Adapun kegunaan-kegunaan menggunakan DFD diantaranya:

1. Dapat menggambarkan sistem secara terstruktur dengan memecah-mecah menjadi level yang lebih rendah.
2. Dapat menunjukkan aliran data di sistem.
3. Dapat menunjukkan simpanan data (basis data).
4. Dapat menunjukkan Kesatuan Luar (terminator).
5. DFD memberikan konsep yang jelas dalam memahami suatu masalah.
6. Simbol yang dipakai sederhana dan mudah untuk dipelajari.

2.9.1.2 Kelemahan-kelemahan *Data Flow Diagram* (DFD)

Selain memiliki kegunaan, DFD juga memiliki beberapa kelemahan, yakni:

1. DFD tidak menunjukkan proses pengulangan (*loop*).
2. DFD tidak menunjukkan proses perhitungan.
3. DFD tidak memperlihatkan aliran kontrol.
4. Lemah dalam konsep model untuk pendeskripsian data dan basis data.

2.9.1.3 Levelisasi *Data Flow Diagram* (DFD)

Didalam *Data Flow Diagram* (DFD) terdapat proses perubahan input menjadi output. Untuk mempermudah pembacaan *Data Flow Diagram* (DFD) dibuat bertingkat / level dari mulai level 0 sampai dengan level yang diperlukan diantaranya yaitu:

1. Diagram Konteks

Diagram Konteks merupakan level tertinggi dari *Data Flow Diagram* (DFD), yang memperlihatkan sistem sebagai sebuah proses. Tujuannya adalah memberikan pandangan umum sistem. Diagram Konteks memperlihatkan sebuah proses yang berinteraksi dengan lingkungannya. Ada *External Entity* yang memberikan masukan (*input*) dan ada pihak yang menerima keluaran (*output*) dari sistem.

2. Diagram Zero

Diagram yang berada satu level dibawah Diagram Konteks yang menggambarkan proses-proses utama dari sistem. hal yang digambarkan dalam diagram Zero adalah proses utama dari sistem serta hubungan terminator atau entitas proses, data flow dan data store.

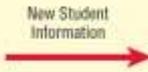
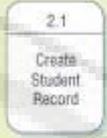
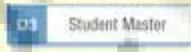
3. Level Diagram (Diagram Level n)

Diagram level n merupakan hasil dekomposisi dari diagram zero, yang menjelaskan proses secara lebih terperinci.

Turunan langsung dari diagram zero dinamakan diagram level 1. Dan apabila diagram level 1 dapat diuraikan lagi maka akan terbentuk diagram level 2, dan seterusnya.

2.9.1.4 Simbol *Data Flow Diagram* (DFD)

Empat simbol dasar yang digunakan untuk memetakan pergerakan data pada diagram arus data: persegi ganda, panah, persegi panjang dengan sudut membulat, dan persegi panjang terbuka (ditutup pada sisi kiri dan buka berakhir di sebelah kanan), seperti yang ditunjukkan pada Gambar 2.8. Seluruh sistem dan banyak subsistem dapat digambarkan secara grafis dengan empat simbol ini dalam kombinasi (K.Kendall dan J. Kendall, 2013).

Symbol	Meaning	Example
	Entity	
	Data Flow	
	Process	
	Data Store	

Gambar 2. 8 Simbol DFD

Sumber: K.Kendall dan J. Kendall (2013)

External entity adalah departemen, bisnis, orang, ataupun mesin yang mengirim ataupun menerima data dari sistem.

Arrow menunjukkan gerakan data dari satu titik ke titik lainnya (dinyatakan dengan *noun*)

Proses menunjukkan proses transformasi, proses selalu terkait dengan perubahan dari data, data yang keluar dari proses selalu dinamai berbeda dengan data yang masuk. Suatu proses dinomori menunjukkan level dari diagram.

Data store dinamai secara nouns yang menunjukkan orang, tempat ataupun sesuatu. Data sementaras seperti file sementara tidak digambarkan pada *Data Flow Diagram* (DFD). Masing-masing data store dinamai secara unik D1, D2, dan seterusnya.

2.9.2 Sistem Design *Integration DEFinition language 0* (IDEF0)

Integration DEFinition language 0 (IDEF0) merupakan bahasa pemodelan yang menggunakan gambar dengan disertai penjelasan yang komprehensif untuk menjelaskan tahapan/metodologi pengembangan dari suatu sistem. Sistem dimodelkan sebagai kumpulan fungsi-fungsi yang saling berkaitan satu dengan yang lain untuk membentuk suatu fungsi utama. Fungsi-fungsi tersebut

menjelaskan apa yang dikerjakan oleh sistem, sehingga apa saja yang mengontrol, memproses, diproses, dan dihasilkan oleh sistem tersebut dapat diketahui.

Building blocks adalah komponen penyusun sistem yang digambarkan dalam model. *Building blocks* ada dua macam, yaitu:

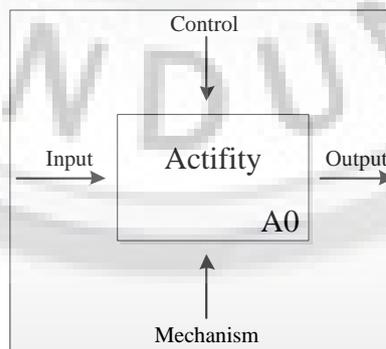
- a. Aktivitas, yaitu komponen suatu sistem yang menjalankan atau melakukan suatu tindakan .
- b. *ICOM*, yaitu komponen suatu sistem yang dipergunakan oleh suatu aktivitas.

ICOM terdiri dari :

1. *Input* : sesuatu yang ditransformasikan oleh suatu aktivitas.
2. *Control* : sesuatu yang menentukan bagaimana suatu aktivitas terjadi tetapi tidak ditransformasikan olehnya.
3. *Output* : sesuatu yang dihasilkan oleh aktivitas
4. *Mechanism* : orang, fasilitas, mesin, atau lainnya yang menjalankan aktivitas.

Integration DEFinition language 0 (IDEF0) memandang suatu sistem sebagai sesuatu yang terdiri dari kumpulan aktivitas yang menggunakan *ICOM-ICOM* untuk mewujudkan tugas-tugasnya. Aktivitas dan *ICOM* merupakan komponen penyusun sistem yang harus diidentifikasi dalam pembentukan model. Dengan kata lain, model dari suatu sistem dengan menggunakan metode IDEF0 adalah merupakan penggambaran aktivitas dan *ICOM* suatu sistem.

Secara keseluruhan aktivitas dan *ICOM* digambarkan dalam Gambar 2.9.



Gambar 2. 9 *Building blocks* dalam IDEF0

Penggambaran model dalam IDEF0 dilakukan secara bertingkat (hirarkis) mulai dari aktivitas umum sampai rinciannya. Pada tingkat tertinggi disebut

Context Page yang berisi satu aktivitas yang menunjukkan seluruh sistem sebagai satu aktivitas dan memperlihatkan pula *interface* sistem dengan lingkungannya. *Context Diagram* juga biasa disebut Diagram A0 atau *Parent Diagram*.

Pada tingkatan berikutnya dibuat *Decomposition Page* atau *Child Diagram* yang merupakan rincian lebih jauh dari sistem. Setiap penjabaran dalam *Decomposition Page* dinamakan Diagram A1, Diagram A2, Diagram A3, Diagram A11, Diagram A22, dan seterusnya. Setiap *ICOM* yang muncul pada suatu *Parent Diagram* akan dirinci pada *child diagram*. *Child Diagram* akan terus dibentuk sampai tingkatan di mana proses berbentuk algoritma pengerjaan aktivitas tersebut.

2.10 Entity Relationship Diagram

Entity Relationship Diagram atau bisa dikenal dengan E-R secara grafis menggambarkan isi sebuah *database*. Elemen-elemen *Entity Relationship Diagram* adalah sebagai berikut:

1. *Entity* (Entitas)

Pada E-R diagram, *entity* digambarkan dengan sebuah bentuk persegi panjang. *Entity* adalah sesuatu apa saja yang ada didalam sistem, nyata maupun abstrak dimana data tersimpan. Entitas diberi nama dengan kata bendahan dapat dikelompokan dalam empat jenis nama, yaitu: orang, benda, lokasi kejadian (terdapat unsur waktu didalamnya). Berdasarkan atribut kuncinya, entitas terbagi menjadi dua, yaitu:

a. Entitas kuat

Entitas yang mempunyai atribut kunci. Entitas ini bersifat mandiri, keberadaannya tidak bergantung pada entitas lainnya. Kebanyakan entitas dalam suatu organisasi dapat digolongkan sebagai entitas kuat (*strong entity*). Entitas kuat memiliki karakteristik yang unik (dinamakan *identifier*), yaitu sebuah atribut tunggal atau gabungan atribut yang secara unik dapat digunakan untuk membedakannya dari entitas kuat yang lain.

b. Entitas lemah

Entitas yang tidak mempunyai atribut kunci. Entitas lemah diidentifikasi dengan menghubungkan entitas tertentu dari tipe entitas yang lain ditambah atribut dari entitas lemah. Tipe entitas lain yang dipakai untuk mengidentifikasi suatu entitas lemah disebut *identifying owner* dan relasi yang menghubungkan entitas lemah dengan owner disebut *identifying relationship*.

2. *Relationship* (Relasi)

Pada E-R diagram, *relationship* dapat digambarkan dengan sebuah bentuk belah ketupat. *Relationship* adalah hubungan alamiah yang terjadi antara entitas. Pada umumnya *relationship* diberi nama dengan kata kerja dasar, sehingga memudahkan untuk melakukan pembacaan relasinya.

3. Atribut

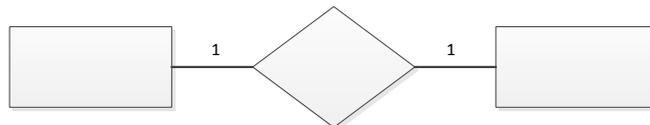
Secara umum atribut adalah sifat atau karakteristik dari tiap entitas maupun tiap *relationship*. Maksudnya adalah sesuatu yang menjelaskan apa sebenarnya yang dimaksud entitas maupun *relationship*, sehingga sering dikatakan bahwa atribut adalah elemen dari setiap entitas dan *relationship*.

4. Kardinalitas

Kardinalitas relasi menunjukkan jumlah maksimum tupel yang dapat berelasi dengan entitas yang lainnya. Dari sejumlah kemungkinan banyaknya hubungan yang terjadi dari entitas, kardinalitas relasi merujuk kepada hubungan maksimum yang terjadi dari entitas yang satu ke entitas yang lainnya dan begitu juga sebaliknya. Terdapat beberapa macam kardinalitas relasi, yaitu:

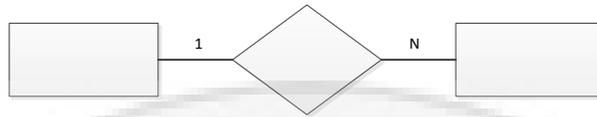
a. *One to one relationship*

Tingkat hubungan satu ke satu, dinyatakan dengan satu kejadian pada entitas pertama, hanya mempunyai satu hubungan dengan satu kejadian pada entitas kedua dan sebaliknya.



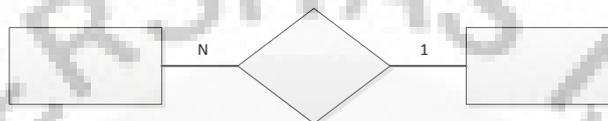
b. *One to many relationship*

Tingkat hubungan satu ke banyak adalah untuk satu kejadian pada entitas yang pertama dapat mempunyai banyak hubungan dengan kejadian pada entitas yang kedua.



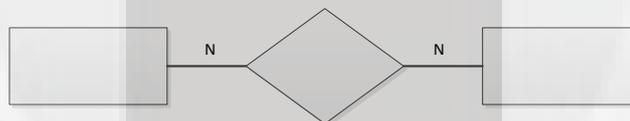
c. *Many to one relationship*

Untuk banyak kejadian pada entitas yang pertama hanya dapat mempunyai satu hubungan dengan kejadian pada entitas yang kedua.



d. *Many to many relationship*

Tingkat hubungan banyak ke banyak terjadi ketika jika tiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya, baik dilihat dari sisi entitas yang pertama maupun dilihat dari sisi yang kedua.



e. *Key (Kunci)*

Sebuah atribut atau set atribut yang nilainya mengidentifikasi entitas secara unik dalam suatu entitas. *Key* memiliki beberapa jenis sesuai dengan kegunaannya masing-masing, yaitu *primary key* (kunci utama), *foreign key* (kunci tamu)

2.11 Struktur Sistem Informasi Manajemen

Struktur konseptual suatu Sistem Informasi Manajemen adalah untuk subsistem fungsional yang terpisah ditambah suatu sistem *database*, yakni beberapa aplikasi umum dan model dasar analisis umum serta model keputusan. Yang lebih sederhananya penstrukturan sistem informasi manajemen. Dalam kerangka dasar sistem informasi terdiri dari:

1. *Hardware*
2. *Software*
3. *Brainware*

Struktur sistem informasi pada dasarnya dibedakan menjadi dua:

- 1) Sistem yang terstruktur (formal)

Sistem formal adalah sistem yang berjalan menurut norma-norma organisasi yang berlaku pada semua orang, sesuai dengan kedudukannya dalam organisasi.

- 2) Sistem nonformal

Sistem nonformal adalah sistem yang berlaku dilingkungan organisasi melalui saluran-saluran tidak resmi tetapi mempunyai pengaruh cukup kuat dalam kehidupan organisasi yang bersangkutan. (Gordon, 1999).

Struktur sistem informasi berdasarkan kegiatan Manajemen, struktur sistem informasi manajemen dapat dianggap sebagai suatu federasi subsistem yang didasarkan atas fungsi yang dilaksanakan dalam suatu organisasi. Dalam kegiatan perencanaan terdapat pengendalian operasional yaitu proses penempatan agar kegiatan operasional dilaksanakan secara efektif dan efisien. Pengendalian operasional menggunakan prosedur dan aturan keputusan yang telah ditentukan lebih dahulu dalam jangka waktu yang relative pendek. Dukungan pengolahan untuk pengendalian operasional terdiri atas; pengolahan transaksi, pengolahan laporan, dan pengolahan pertanyaan. (Gordon, 1999).