

BAB II

TINJAUAN PUSTAKA

2.1 Sistem

Sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan. [1] Suatu sistem dapat dijelaskan dengan sederhana sebagai seperangkat elemen yang digabungkan satu dengan lainnya untuk suatu tujuan bersama. Suatu subsistem adalah bagian dari sistem yang lebih besar dengan mana kita berkepentingan. Semua sistem adalah bagian dari sistem yang lebih besar. Untuk maksud kita, organisasi adalah sistem dan bagiannya (devisi, departemen, fungsi, suatu dan sebagainya) adalah subsistem. [2]

Adapun karakteristik sistem adalah sebagai berikut :

a. Komponen sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerja sama membentuk suatu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem.

b. Batasan sistem (*Boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lain atau sistem dengan lingkungan luarnya. Batasan sistem ini kemungkinan suatu sistem dipandang sebagai suatu kesatuan yang tidak dapat dipisah-pisahkan.

c. Lingkungan luar sistem (*Environment*)

Bentuk apapun yang ada diluar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut lingkungan luar sistem. Lingkungan luar sistem ini dapat juga bersifat merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi bagi sistem tersebut.

d. Penghubung sistem (*Interface*)

Media yang menghubungkan sistem dengan subsistem lain disebut penghubung sistem atau *interface*. Penghubung ini memungkinkan sumber-sumber daya mengalir dari subsistem lain. Bentuk keluaran dari suatu subsistem akan menjadi masukan untuk subsistem lain melalui penghubung tersebut.

e. Masukan sistem (*Input*)

Energi yang dimasukkan kedalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*maintance input*) dan sinyal (*signal input*). Contoh, didalam suatu unit sistem komputer.

f. Keluaran sistem (*Output*)

Keluaran ini merupakan masukan bagi subsistem yang lain. Keluaran yang dihasilkan adalah informasi. Informasi ini dapat digunakan sebagai masukan untuk pengambilan keputusan atau hal-hal lain yang menjadi input bagi subsistem lain.

g. Pengolah sistem (*Proses*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran.

h. Sasaran Sistem (*Objective*)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat *deterministic*. Kalau suatu sistem tidak memiliki sasaran, maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan. [3]

2.2 Informasi

Informasi merupakan salah satu sumber daya penting dalam suatu organisasi; digunakan sebagai bahan pengambilan keputusan. [4] Data harus dibedakan dari informasi, dan perbedaan ini jelas serta penting untuk maksud kita. Data adalah fakta dan angka yang tidak sedang digunakan pada proses keputusan, dan biasanya berbentuk catatan historis yang dicatat dan diarsipkan tanpa maksud untuk segera diambil kembali untuk pengambilan keputusan. Sebagai contoh dapat berupa sebuah dokumen penunjang. Buku besar, dan sebagainya yang terdiri dari material sumber untuk perhitungan rugi laba. Materi serupa itu hanya merupakan perhatian historis bagi auditor luar.

Informasi terdiri dari data yang telah diambil kembali, diolah, atau sebaliknya digunakan untuk tujuan informasif atau kesimpulan, argumentasi, atau sebagai dasar untuk peramalan atau pengambilan keputusan. Suatu contoh disini dapat juga berupa sebuah dokumen penunjang yang telah disebutkan, tetapi dalam hal ini data dapat

digunakan oleh auditor intern, departemen pelayanan manajemen dari auditor luar, atau manajemen intern untuk perencanaan keuntungan dan pengendalian atau untuk tujuan pengambilan keputusan lainnya. [2]

2.3 Sistem Informasi

Sesungguhnya, yang dimaksud dengan sistem informasi tidak harus melibatkan komputer. Sistem informasi yang menggunakan komputer biasa disebut sistem informasi berbasis komputer (*Computer Based Information System atau CBIS*). Dalam praktik, istilah sistem informasi lebih sering dipakai tanpa embel-embel berbasis komputer walaupun dalam kenyataannya komputer merupakan bagian yang penting. Di buku ini, yang dimaksudkan dengan sistem informasi adalah sistem informasi yang berbasis komputer. Ada beragam definisi sistem informasi, antara lain sebagai berikut:

- a. Alter (1992) : Sistem informasi adalah kombinasi atar prosedur kerja, informasi, orang, dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi.
- b. Bodnar dan Hopwood (1993) : Sistem Informasi adalah kumpulan perangkat keras dan perangkat lunak yang dirancang untuk mentransformasikan data kedalam bentuk informasi yang berguna.
- c. Galinas, Oram, Dan Wiggins : Sistem Informasi adalah suatu sistem buatan manusia secara umum, terdiri atas sekumpulan komponen berbasis komputer dan manual yang dibuat untuk menghimpun, menyimpan, dan mengelola, data serta menyediakan informasi keluaran kepada para pemakai. [5]

2.4 Sistem Informasi Penjualan

Penjualan merupakan pembelian sesuatu (barang atau jasa) dari suatu pihak kepada pihak lainnya dengan mendapatkan ganti uang dari pihak tersebut. Penjualan juga merupakan suatu sumber pendapatan perusahaan, semakin besar penjualan maka semakin besar pula pendapatan yang diperoleh perusahaan.

Pada saat perusahaan menjual barang dagangannya, maka diperoleh pendapatan. Jumlah yang dibebankan kepada pembeli untuk barang dagang yang diserahkan merupakan pendapatan perusahaan yang bersangkutan. Penjualan dapat dilakukan secara kredit maupun tunai dan pada umumnya kepada beberapa langganan.

Seperti halnya waktu membeli, ketika menjual perusahaan juga terikat dengan syarat jual beli tertentu.

Pada waktu menjual, kadang-kadang perusahaan harus menerima pengembalian barang atau memberi potongan harga. Hal ini terjadi kalau barang yang dijual tidak sesuai dengan permintaan pembeli. Penerimaan kembali barang yang telah dijual disebut penjualan retur, sedang pemberian potongan harga disebut pengurangan harga. Pada umumnya penjualan retur dan pengurangan harga dicatat dalam satu perkiraan, yang disebut : Penjualan retur dan pengurangan harga. [6]

2.5 Sistem Informasi Pembelian

Dalam sebuah perusahaan dagang kegiatan pembelian meliputi pembelian aktiva produksi, pembelian barang dagang serta pembelian barang dagang dan jasa lain dalam kegiatan usaha. Pembelian dapat dilakukan secara kredit maupun tunai dan pada umumnya dilakukan kepada beberapa supplier dan konsumen. Pada waktu membeli barang dagang, perusahaan terikat pada suatu syarat jual beli tertentu. Pada syarat jual beli tertentu, termasuk dalam harga pokok barang, adalah ongkos angkut, asuransi dan lain-lain.

Apabila perusahaan tidak puas dengan kualitas barang yang dibeli, maka dengan persetujuan penjual dapat dimintakan pengurangan harga atau mengembalikan barang tersebut. Perkiraan untuk mencatat pengurangan harga dan pengembalian disebut “pembelian retur dan pengurangan harga”. [6]

2.6 Sistem Informasi Persediaan

Persediaan adalah bahan atau barang yang disimpan untuk ditemukan memenuhi tujuan tertentu, misalnya untuk digunakan dalam proses produksi atau perakitan, untuk dijual kembali, atau untuk suku cadang dari suatu peralatan atau mesin. Persediaan dapat berupa bahan mentah, bahan pembantu, bahan dalam proses, barang jadi, atau pun suku cadang. [7]

Persediaan menurut pernyataan sekunder akuntansi adalah aktiva yang tersedia untuk dijual dalam kegiatan usaha normal perusahaan, aktiva dalam proses produksi dan atau dalam perjalanan atau dalam bentuk untuk perlengkapan (*supplies*) untuk digunakan dalam proses produksi atau pemberian jasa. Dan pengertian ini sangat

umum dan berlaku bagi perusahaan jasa, dagang maupun manufaktur. Persediaan juga meliputi barang yang dibeli dan disimpan untuk dijual kembali. Misalnya barang dagang yang dibeli oleh pengecer untuk dijual kembali, atau pengadaan tanah dan properti lainnya untuk dijual kembali. Persediaan juga mencakup barang jadi yang telah diproduksi, atau barang dalam penyelesaiannya yang sedang diproduksi perusahaan, dan termasuk bahan serta perlengkapan yang akan digunakan dalam proses produksi. Bagi perusahaan jasa persediaan meliputi biaya jasa seperti upah dan biaya personalia lainnya yang secara langsung menangani pemberian jasa, termasuk tenaga penyedia, dan overhead yang didistribusikan. [8]

Fungsi persediaan:

- a. Menghilangkan resiko keterlambatan pengiriman barang baku atau barang yang dibutuhkan perusahaan.
- b. Menghilangkan resiko jika material yang dipesan tidak baik sehingga harus dikembalikan.
- c. Menghilangkan resiko terhadap kenaikan harga barang atau inflasi.
- d. Untuk menyimpan barang baku yang dilakukan secara musiman sehingga perusahaan tidak akan kesulitan jika bahan itu tidak tersedia dipasaran.
- e. Mendapatkan keuntungan dari pembelian berdasarkan diskon kuantitas.

Persediaan dapat dikelompokkan kedalam empat jenis. Empat jenis kelompok persediaan tersebut adalah sebagai berikut:

- a. *Fluctuation stock* merupakan persediaan yang dimaksudkan untuk menjaga terjadinya fluktuasi permintaan yang tidak diperkirakan sebelumnya, dan untuk mengatasi bila terjadi kesalahan dalam perkiraan penjualan, waktu produksi, atau pengiriman barang.
- b. *Anticipation stock* merupakan persediaan untuk menghadapi permintaan yang dapat diramalkan, misalnya pada musim permintaan tinggi, tetapi kapasitas produksi pada saat itu tidak mampu memenuhi permintaan.
- c. *Lot-size inventory* merupakan persediaan yang diadakan dalam jumlah yang lebih besar dari pada kebutuhan saat itu. Persediaan dilakukan untuk mendapatkan keuntungan dari harga barang (berupa diskon) karena membeli dalam jumlah yang besar, atau untuk mendapatkan penghematan dari biaya pengangkutan perunit yang lebih rendah.

- d. *Pipeline inventory* merupakan persediaan yang dalam proses pengiriman dari tempat asal ketempat dimana barang itu akan digunakan. Misalnya , barang yang dikirim dari pabrik menuju ke tempat penjualan, yang dapat memakan waktu beberapa hari atau minggu. [7]

2.7 Data Flow Diagram (DFD)

Diagram arus data (*Data Flow Diagram*) menggunakan simbol-simbol untuk menyajikan entitas, proses, arus data, dan penyimpanan data yang berkaitan dengan suatu sistem.

Entitas dalam DFD adalah objek-objek eksternal dalam sistem yang dimodelkan. Entitas ini mewakili sumber dan tujuan dari data. Entitas dapat berupa sistem lain atau fungsi yang saling berinteraksi, atau berada di luar perusahaan, seperti pelanggan dan pemasok.

DFD digunakan untuk menyajikan sistem dalam beberapa tingkat perincian dari yang sangat umum ke yang sangat terperinci. DFD banyak digunakan oleh analis sistem untuk mewakili elemen logis dari sistem. Akan tetapi, teknik ini tidak mewakili sistem fisik. Dengan kata lain, DFD menunjukkan tugas logis yang sedang dilakukan, namun tidak menunjukkan cara melakukannya atau siapa (atau apa) yang melakukannya. Misalnya, DFD tidak menunjukkan apakah proses persetujuan penjualan terpisah secara fisik dari proses penagihan agar sesuai dengan tujuan pengendalian internal. [9]

Untuk membaca suatu DFD kita harus memahami dulu, elemen-elemen yang menyusun suatu DFD. Ada empat elemen yang menyusun suatu DFD, yaitu :

a. Proses

Aktivitas atau fungsi yang dilakukan untuk elemen bisnis yang spesifik, biasa berupa manual maupun terkomputerisasi.

b. Data Flow

Satu data tunggal atau kumpulan logis suatu data, selalu diawali atau berakhir pada suatu proses.

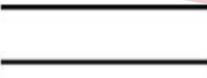
c. Data Store

Kumpulan data yang disimpan dengan cara tertentu. Data yang mengalir, disimpan dalam data store. Aliran data di-update atau ditambahkan ke data store.

d. External entity

Orang, organisasi, atau elemen yang berada diluar sistem dapat berinteraksi dengan sistem. Masing-masing elemen akan diberi lambing tertentu untuk membedakan satu dengan yang lain. Ada beberapa metode untuk menggambarkan elemen-elemen tersebut. Untuk lebih jelasnya dapat dilihat pada tabel berikut:

Tabel 2. 1 Simbol DFD

Notasi Yourdon DeMarco	Notasi Gane & Sarson	Deskripsi
		Simbol Entitas Eksternal / Terminator menggambarkan asal atau tujuan data di luar system
		Simbol lingkaran menggambarkan entitas atau proses dimana aliran data masuk ditransformasikan ke aliran data keluar
		Simbol aliran data menggambarkan aliran data
		Simbol file menggambarkan tempat data disimpan

Dalam melakukan pemodelan proses, dibutuhkan beberapa level DFD agar proses yang dimodelkan tersebut sesuai dengan sistem yang diinginkan. Semakin tinggi level DFD menunjukkan pemodelan yang semakin rinci.

Context Diagram: DFD pertama dalam proses bisnis. Menunjukkan konteks di mana proses bisnis berada. Menunjukkan semua proses bisnis dalam 1 proses tunggal (proses 0). *Context diagram* juga menunjukkan semua entitas luar yang menerima informasi dari atau memberikan informasi ke sistem.

Level 0 Diagram: Menunjukkan semua proses utama yang menyusun keseluruhan sistem. Level ini juga menunjukkan komponen internal dari proses 0 dan menunjukkan bagaimana proses-proses utama direlasikan menggunakan data flow. Pada level ini juga ditunjukkan bagaimana proses-proses utama terhubung dengan entitas eksternal. Pada level ini juga dilakukan penambahan data store.

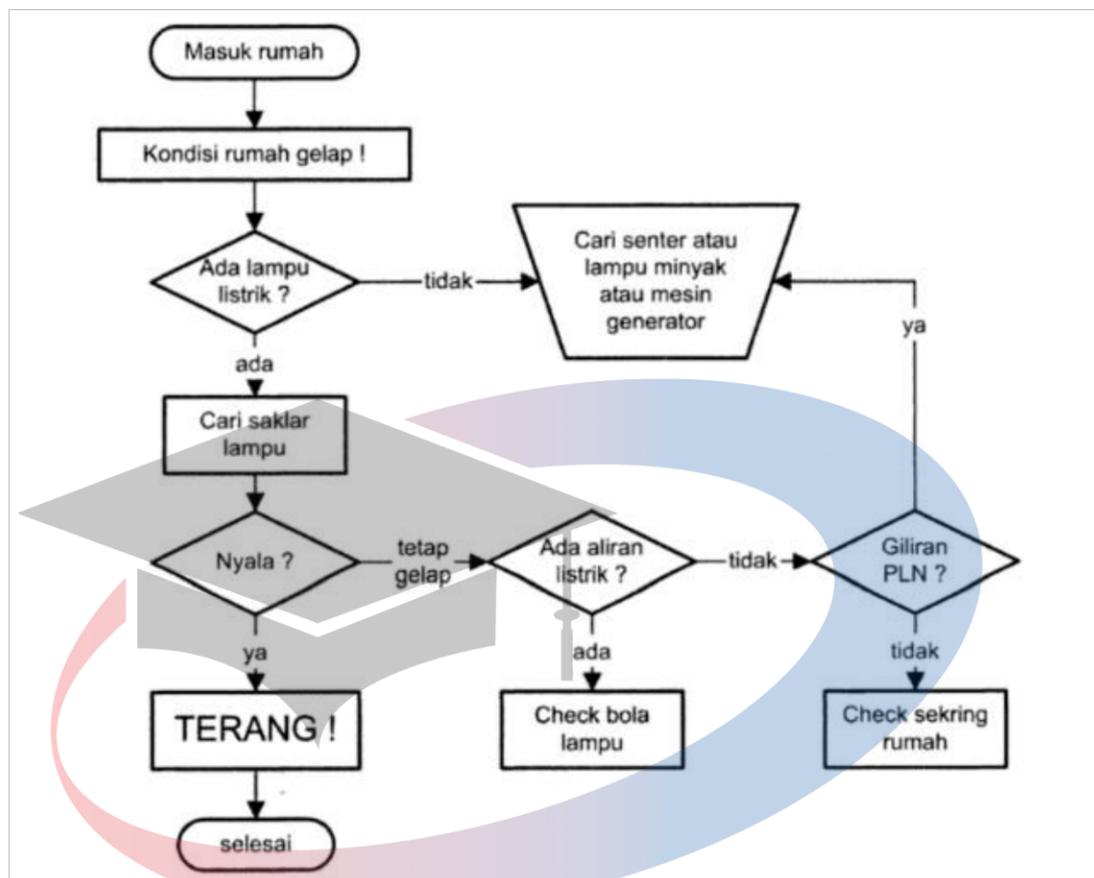
Level 1 Diagram: Umumnya diagram level 1 diciptakan dari setiap proses utama dari level 0. Level ini menunjukkan proses-proses internal yang menyusun setiap proses-proses utama dalam level 0, sekaligus menunjukkan bagaimana informasi berpindah dari satu proses ke proses yang lainnya. Jika misalnya proses induk dipecah, katakanlah menjadi 3 proses anak, maka 3 proses anak ini secara utuh menyusun proses induk.

Level 2 Diagram: Menunjukkan semua proses yang menyusun sebuah proses pada level 1. Bisa saja penyusunan DFD tidak mencapai level 2 ini, atau mungkin harus dilanjutkan ke level berikutnya (level 3, level 4, dan seterusnya). [10]

2.8 Flowchart Of Document (FOD)

2.8.1 Umum

Flowchart atau bagian alir, awal mulanya memang berkembang dari industri computer yaitu untuk menggambarkan urutan proses penyelesaian masalah. Namun seperti kata pepatah lama bahwa ‘satu gambar adalah sejuta kata’ maka suatu flowchart dapat dengan mudah menjelaskan suatu urutan-urutan proses yang relative rumit bila diuraikan dalam kata-kata. Dengan visualisasi maka adanya *bottleneck* (penumpukan) atau ketidak-efisienan dari suatu proses dapat terdeteksi untuk dilakukan perbaikan. Oleh karena itu flowchart juga diterima di kalangan lain (manufaktur, sains, militer, manajemen, dsb). Flowchart sangat berguna khususnya untuk menjelaskan urutan-urutan proses yang pelaksanaannya mempunyai banyak option pilihan atau percabangan. Contoh lain flowchart untuk penyelesaian masalah sehari-hari tentang ‘kondisi di dalam rumah yang gelap dan langkah-langkah mengatasinya’. Misalnya:



Gambar 2. 1 Flowchart dalam kehidupan sehari-hari

Dalam menggambar flowchart diperlukan symbol-simbol yang berbentuk persegi, belah ketupat, maupun bentuk lain yang kemudian dihubungkan dengan garis-garis berarah (garis dengan simbol anak panah), setiap symbol pada diagram tersebut menunjukkan langkah penyelesaian, sedangkan garis berpanah menunjukkan urutan kejadian. Tentu saja symbol yang dipilih harus yang biasa dipakai oleh komunitas yang dituju (yang akan membaca dan memakai flowchart tersebut) sehingga ada kesamaan pengertian, walaupun ada sedikit modifikasi di sana-sini tidaklah menjadi masalah serius. Kadang-kala memang diperlukan kreativitas dan keberanian agar flowchart tersebut menjadi menarik dan berfungsi baik.

Flowchart dalam pemrograman computer kadangkalah diibaratkan sebagai cetak-biru suatu rencana bangunan yang akan dikerjakan para insinyur. Seperti diketahui bahwa para insinyur dan arsitek memerlukan gambar rencana, baik itu gambar arsitek, gambar struktur maupun gambar utilitas sebelum memulai suatu pekerjaan konstruksi suatu bangunan gedung. Hal yang sama juga berlaku, pemrogram yang baik sebaiknya

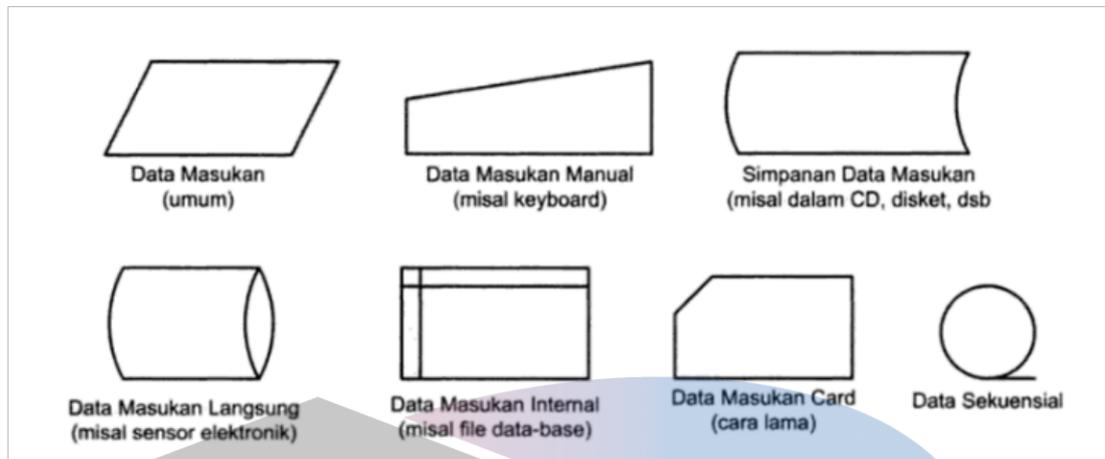
menggambarkan flowchart terlebih dahulu sebelum menuliskannya pada computer program.

Meskipun demikian dalam praktek, khususnya para pemrogram yang sudah 'karatan' yang umumnya sudah mempunyai kemampuan untuk menuangkan gambaran alur program (flowchart) langsung di otaknya. Sehingga secara fisik seakan tidak diperlukan lagi, flowchart dalam penulisan program. Namun yang jelas, jika pembuatan program tersebut dilakukan suatu team pemrogram maka adanya gambar flowchart dapat secara mudah digunakan untuk mengomunikasikan ide satu dengan yang lainnya sehingga ide-ide anggota team dapat seling bersinergi bersama. Flowchart juga dapat berfungsi sebagai perantara antara pemrogram dan pelaku bisnis untuk memberikan gambaran mengenai cara kerja program sehingga pembeli mempunyai keyakinan bahwa program yang akan dibeli sesuai dengan kebutuhannya.

Flowchart membantu memahami urutan-urutan logika yang rumit dan panjang. Jelasnya dengan flowchart maka mengkomunikasikan jalannya program ke orang lain (bukan pemrogram) akan lebih mudah. Jadi pembuatan flowchart adalah mutlak khususnya untuk melengkapi cara dokumentasi suatu listing program computer yang relatif rumit sehingga dokumentasi tersebut suatu saat apabila diperlukan kembali dapat dengan mudah dipahami.

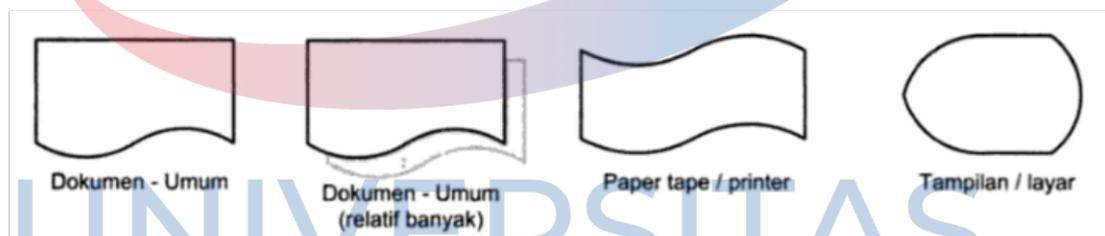
2.8.2 Petunjuk Membuat Flowchart

Flowchart umumnya dibuat dengan symbol-simbol standart (yang telah disepakati bersama), tetapi bila perlu dapat juga dibuat symbol khusus asalkan pemakaiannya konsisten tidak berubah-ubah. Beberapa bentuk-bentuk symbol flowchart yang biasa digunakan untuk pemrograman komputer.



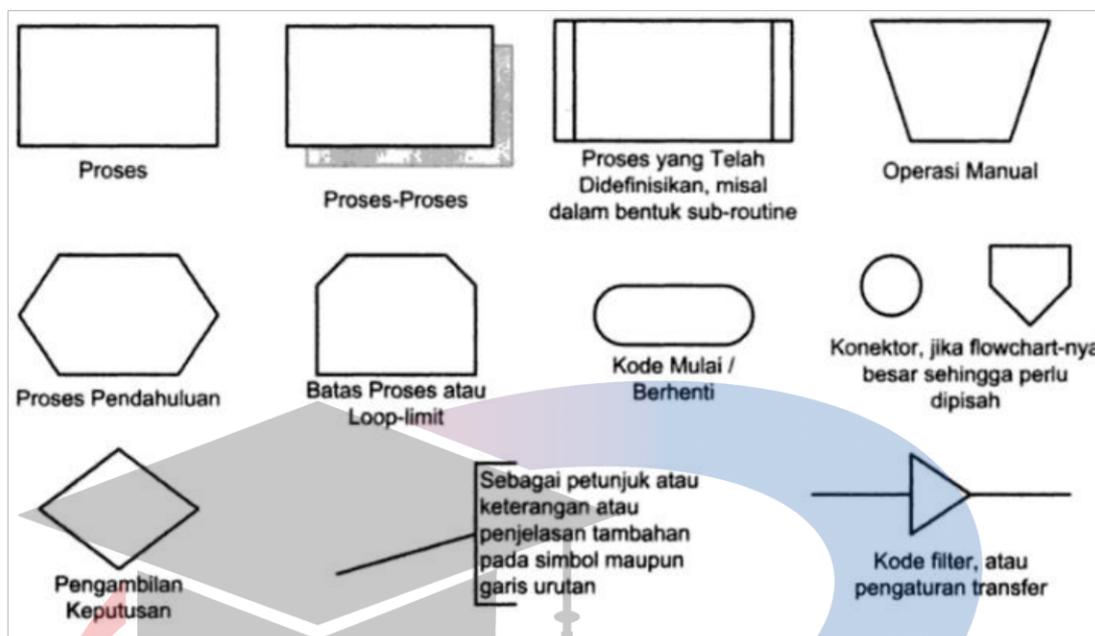
Gambar 2. 2 Simbol Masukan (simbol data)

Sangat penting membedakan data masukan dan data keluaran. Pada program sederhana maka 'data masukan umum' sudah mencukupi, tetapi dengan semakin kompleksnya program, kadang-kadang diperlukan simbol lain agar mudah menelaah proses-proses apa yang selanjutnya akan mengikuti.



Gambar 2. 3 Simbol Keluaran (Output Data)

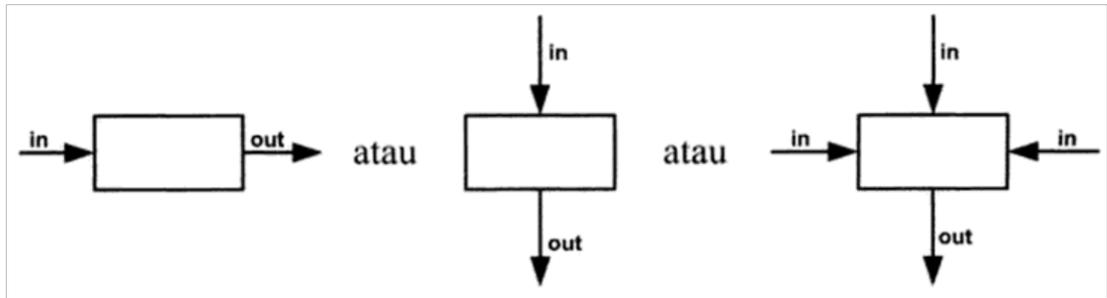
Untuk data keluaran, symbol yang relative lebih sedikit variasinya, kadang kala untuk menunjukkan bahwa outputnya banyak atau untuk menunjukkan bahwa item tersebut perlu perhatian maka simbolnya diberi bayangan.



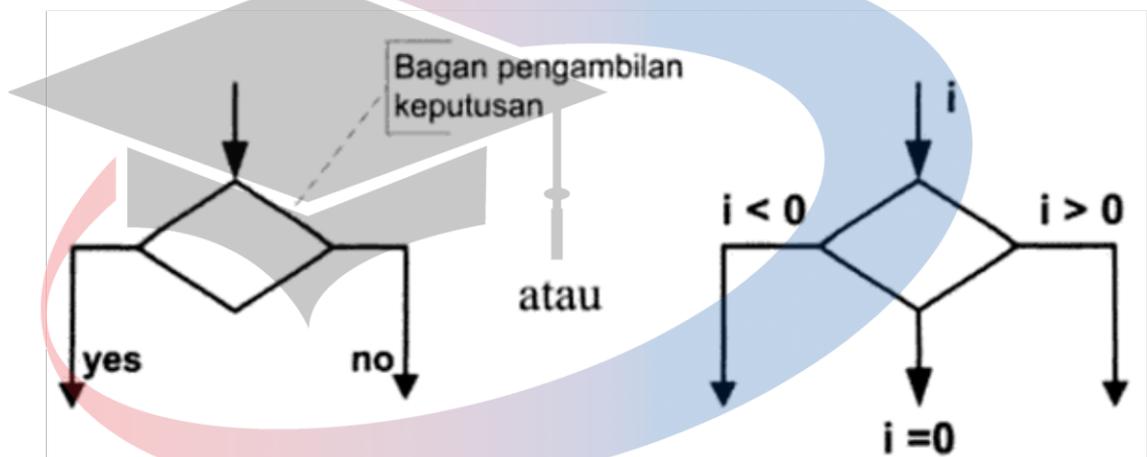
Gambar 2. 4 Simbol Proses atau Lainnya

Beberapa petunjuk dalam pembuatan bagan alir.

- Dalam menggambarkan flowchart yang benar, semua proses-proses harus diurutkan secara logika. Perlu diidentifikasi dari mana suatu proses akan dimulai (start), data-data input apa yang diperlukan, hal apa yang menyebabkan suatu proses harus diulang (looping) dan proses menghasilkan apa, akhirnya kapan suatu proses selesai.
- Flowchart harus cukup jelas, rapid an mudah diikuti. Boleh saja menggunakan symbol-simbol yang sedikit dimodifikasi dari yang biasa dipakai, tetapi yang jelas jangan gunakan gambar atau symbol sama atau mirip untuk fungsi yang berbeda sehingga dapat membingungkan, jadi pemakaian symbol harus konsisten.
- Umumnya arah aliran dari setiap bagan kerja adalah dari kiri ke kanan atau dari atas ke bawah, tetapi itu tidak mutlak tergantung kasus-kasus yang diselesaikan. Selain itu sebaiknya hanya ada satu keluaran dari suatu bagan pemrosesan, untuk masukannya bias lebih dari satu tapi sebaiknya tidak lebih tiga aliran masuk. Garis aliran diagonal relative jarang karena dapat menimbulkan persalingan.



- d. Hanya satu aliran masuk pada bagan pengambilan keputusan, keluaran minimal dua atau tiga aliran.



- e. Hanya ada satu aliran ke luar/masuk dari bagan awal/akhir. [11]



2.9 Data Dictionary atau Kamus Data

Kamus data adalah suatu aplikasi khusus dari jenis kamus-kamus yang digunakan sebagai referensi kehidupan setiap hari. Kamus data merupakan hasil referensi data mengenai data, suatu data yang disusun oleh penganalisis sistem untuk membimbing mereka selama melakukan analisis dan design. Sebagai suatu dokumen, kamus data mengumpulkan dan mengkoordinasikan istilah- istilah data tertentu, dan menjelaskan apa arti setiap istilah yang ada.

Berikut adalah simbol- simbol yang digunakandalam kamus data:

Tabel 2. 2 Simbol- simbol kamus data

Notasi	Keterangan
=	Terdiri dari
+	Dan
{ }	Elemen- elemen repetitif (kelompok berulang)
[]	Salah satu dari dua situasi tertentu
()	Pilihan (boleh dikosongkan)

Kamus data otomatis sangat berguna karena memiliki kapasitas dalam hal referensi silang *item- item* data, dengan demikian memungkinkan dilakukannya perubahan- perubahan program terhadap semua program yang berbagi suatu elemen biasa. Fitur ini menggantikan pengubahan program serampangan atau mencegah penundaan sampai program tidak bisa berjalan karena perubahan tersebut tidak diimplementasikan terhadap program yang telah diperbaharui. Jelasnya kamus data otomatis menjadi sangat penting untuk sistem- sistem besar karena mampu menghasilkan ribuan elemen data yang dikatalogkan dan dibuat referensi silang.

Sebagai Tambahan untuk dokumentasi serta mengurangi redundansi, kamus data bisa digunakan untuk :

1. Melakukan validasi diagram aliran data dalam hal kelengkapan dan keakuratan.
2. Menyediakan suatu titik awal untuk mengembangkan layar dan laporan laporan.
3. Menentukan muatan data yang disimpan di *file*.

Dalam kamus data dikenal adanya struktur data. Struktur data adalah bagan struktur yang biasanya terdiri dari beberapa *item* tertentu yang memuat informasi- informasi mengenai elemen- elemen pada kamus data. Struktur data biasanya digambarkan dengan menggunakan rotasi aljabar. Berikut ini adalah simbol notasi struktur data dalam kamus data:

1. Tanda sama dengan (=), artinya ”terdiri dari”

2. Tanda plus (+), artinya "dan"
3. Tanda kurung kurawal { }, artinya menunjukkan elemen- elemen repetitive, juga disebut kelompok berulang atau tabel- tabel.
4. Tanda kurung siku [], artinya menunjukkan salah satu dari dua situasi tertentu. Satu elemen bisa ada sedangkan elemen lainnya juga bisa ada tetapi tidak bisa kedua-duanya ada secara bersamaan. Elemen- elemen yang ada didalam tanda kurung ini saling terpisah satu sama lain.
5. Tanda kurung (), merupakan suatu elemen yang bersifat pilihan. Elemen- elemen yang bersifat pilihan ini bisa dikosongkan pada layar masukan atau bisa juga dengan memuat spasi atau nol untuk *field- field* numeric pada struktur *file*. [12]

2.10 Basis Data

Pemrosesan basis data sebagai perangkat andalan sangat diperlukan oleh berbagai institusi dan perusahaan. Dalam pengembangan sistem informasi diperlukan basis data sebagai media penyimpanan data. Kehadiran basis data dapat meningkatkan kinerja perusahaan dan dapat meningkatkan daya saing perusahaan tersebut.

Basis data, dapat mempercepat upaya pelayanan kepada pelanggan, menghasilkan informasi dengan cepat dan tepat sehingga membantu pengambil keputusan untuk segera memutuskan suatu masalah berdasarkan informasi yang ada. Banyak aplikasi yang dapat dibuat dengan berlandaskan pada basis data antara lain semua transaksi perbankan, aplikasi pemesanan dan penjadwalan penerbangan, proses registrasi dan pencatatan data mahasiswa pada perguruan tinggi, aplikasi pemrosesan penjualan, pembelian dan pencatatan data barang pada perusahaan dagang, pencatatan data pegawai beserta aktifitasnya termasuk operasi penggajian pada suatu perusahaan dan sebagainya.

Dengan memanfaatkan teknologi jaringan, kemampuan basis data dapat di optimalka, misalnya transaksi antar cabang pada sebuah perbankan secara online. Begitu banyak keuntungan yang dapat diperoleh dengan pemanfaatan basis data. Basis data dapat meningkatkan daya guna perangkat komputer yang mungkin tadinya hanya untuk keperluan *game* atau pengetikan dengan aplikasi *Microsoft Office*.

2.9.1 Defenisi Basis Data

Basis data adalah kumpulan data yang saling berelasi. Data sendiri merupakan fakta mengenai obyek, orang, dan lain-lain. Data dinyatakan dengan nilai (angka, deretan karakter, atau simbol). Basis data dapat didefenisikan dalam berbagai sudut pandang seperti berikut:

1. Himpunan kelompok data yang saling berhubungan yang diorganisasi sedemikian rupa sehingga kelak dapat dimanfaatkan dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa tanpa pengulangan (*redudancy*) yang tidak perlu, untuk memenuhi kebutuhan.
3. Kumpulan arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

2.9.2 Tujuan Basis Data

Basis data bertujuan untuk mengatur data sehingga diperoleh kemudahan, ketepatan, dan kecepatan dalam pengambilan kembali. Untuk mencapai tujuannya, syarat sebuah basis data yang baik adalah sebagai berikut:

- a. Tidak adanya redundansi dan inkonsisten data

Redundansi terjadi jika suatu informasi disimpan di beberapa tempat, Misalnya, ada data mahasiswa yang memuat NIM, nama, alamat, dan atribut lainnya, sementara kita punya data lain tentang data KHS mahasiswa yang isinya terdapat NIM, nama, mata kuliah, dan nilai. Pada kedua data tersebut kita temukan ada atribut nama.

- b. Mengatasi masalah kesulitan akses data

Sebuah perusahaan biasa mengirimkan surat penawaran barang kepada para pelanggannya, akan tetapi data mengenai pelanggan tersebut tidak pernah dijadikan basis data. Suatu saat hendak dicetak laporan mengenai jumlah pelanggan perusahaan, sekalipun data-data pelanggan diketahui, tetap akan sulit mendapatkan laporan yang dimaksud karena data pelanggan tidak pernah dikumpulkan untuk menjadi basis data.

c. Mengatasi masalah isolasi data

Jika kasus pada bagian keuangan dan bagian personalia sebelumnya tidak terpecahkan, tentunya akan sulit untuk mendapatkan data dalam format yang sama, oleh karena itu diperlukan basis data yang struktur rancangannya mempunyai aturan yang baku.

d. Multiple user

Data base yang baik tentunya bertumpu pada suatu pusat data. Data yang terpusat dan terintegrasi tersebut harus bisa digunakan secara bersamaan (berbagai pakai) antara pengguna yang satu dengan yang lainnya, untuk itu diperlukan adanya jaringan LAN (Local Area Network).

e. Masalah keamanan data (*security data*)

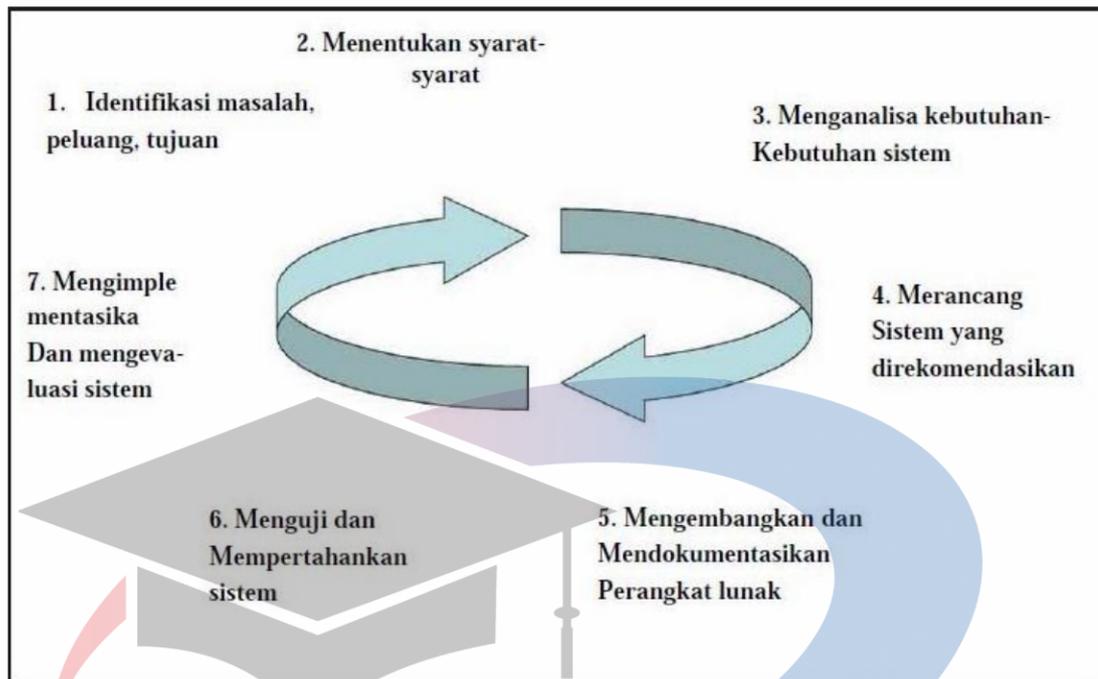
Tidak semua pemakai database boleh mengakses dengan bebas basis data. Ada aturan dan kewenangan yang dibuat oleh DBA dan harus dipatuhi oleh semua pengguna basis data, karena itulah dalam jaringan komputer ada hak dan wewenang user.

2.11 *System Development Life Cycle (SDLC)*

System Development Life Cycle (SDLC) adalah pendekatan melalui beberapa tahap untuk menganalisis dan merancang sistem.

Siklus hidup pengembangan sistem adalah pendekatan melalui beberapa tahap untuk menganalisis dan merancang sistem yang dimana sistem tersebut telah dikembangkan dengan sangat baik melalui penggunaan siklus kegiatan penganalisis dan pemakai secara spesifik.

Siklus hidup pengembangan sistem dibagi dalam tujuh tahap. Meskipun masing-masing tahap ditampilkan secara terpisah, namun tidak pernah tercapai sebagai satu langkah terpisah. Melainkan beberapa aktivitas muncul secara simultan, dan aktivitas tersebut dilakukan secara berulang-ulang. Lebih berguna lagi memikirkan bahwa SHPS bisa dicapai dalam tahap-tahap (dengan aktivitas berulang yang saling tumpang tindih satu sama lainnya dan menuju ke tujuan terakhir) dan tidak dalam langkah-langkah terpisah. [13]



Gambar 2. 5 *System Development Life Cycle*

1. Mengidentifikasi masalah, peluang, dan tujuan.

Di tahap pertama dari siklus hidup pengembangan sistem ini, penganalisis mengidentifikasi masalah, peluang, dan tujuan-tujuan yang hendak dicapai. Tahap ini sangat penting bagi keberhasilan proyek, karena tidak seorangpun yang ingin membuang-buang waktu kalau tujuan masalah yang keliru. Tahap pertama ini berarti bahwa penganalisis melihat dengan jujur pada apa yang terjadi di dalam bisnis. Kemudian, bersama-sama dengan anggota organisasional lain, penganalisis menentukan dengan tepat masalah-masalah tersebut. Peluang adalah situasi dimana penganalisis yakin bahwa peningkatan bisa dilakukan melalui penggunaan sistem informasi terkomputerisasi. Peluang dan tujuan didapatkan dengan mengidentifikasi masalah-masalah yang terjadi di dalam perusahaan tersebut.

2. Menentukan syarat-syarat informasi.

Tahap ini, penganalisis memasukkan apa saja yang menentukan syarat-syarat informasi untuk para pemakai yang terlibat. Di antara perangkat-perangkat yang digunakan untuk menetapkan syarat-syarat informasi di dalam bisnis diantaranya

ialah menentukan sampel dan memeriksa data mentah, wawancara, mengamati perilaku pembuat keputusan dan lingkungan kantor dan *prototyping*.

3. Menganalisis kebutuhan-kebutuhan sistem.

Tahap berikut ini membantu penganalisis dalam menganalisis kebutuhan dari sistem dengan perangkat dan teknik-teknik tertentu. Perangkat yang dimaksud adalah penggunaan diagram alir data untuk menyusun daftar *input*, proses, dan output fungsi bisnis dalam bentuk grafik terstruktur. Dari diagram alir data, dikembangkan suatu kamus data berisikan daftar seluruh item data yang digunakan dalam sistem.

4. Merancang sistem yang direkomendasikan.

Dalam tahap desain dari siklus hidup pengembangan sistem, penganalisa sistem menggunakan informasi-informasi yang terkumpul sebelumnya untuk mencapai desain sistem informasi yang *logic*. Penganalisis merancang prosedur *data-entry* sedemikian rupa sehingga data yang dimasukkan ke dalam sistem informasi benar-benar akurat. Selain itu, penganalisis menggunakan teknik-teknik bentuk dan perancangan layar tertentu untuk menjamin keefektifan *input* sistem informasi.

5. Mengembangkan dan mendokumentasikan perangkat lunak.

Dalam tahap kelima dari siklus hidup pengembangan sistem, penganalisis bekerja bersama-sama dengan pemrogram untuk mengembangkan suatu perangkat lunak awal yang diperlukan. Selain itu penganalisis juga bekerja sama dengan pemakai untuk mengembangkan dokumentasi perangkat lunak yang efektif.

6. Menguji dan mempertahankan sistem.

Sebelum sistem informasi dapat dipergunakan, maka harus dilakukan pengujian terlebih dahulu. Akan bisa menghemat biaya bila dapat menangkap adanya masalah sebelum sistem tersebut ditetapkan. Sebagian pengujian dilakukan oleh pemrogram sendiri, dan yang lainnya dilakukan oleh penganalisis sistem.

7. Mengimplementasikan dan mengevaluasi sistem.

Di tahap terakhir dari pengembangan sistem, penganalisis membantu untuk mengimplementasikan sistem informasi. Tahap ini melibatkan pelatihan bagi pemakai untuk mengendalikan sistem. Sebagian pelatihan tersebut dilakukan oleh

vendor, namun kesalahan pelatihan merupakan tanggung jawab penganalisis sistem. [13]

2.12 Normalisasi

Normalisasi adalah suatu proses untuk membuat data yang tidak normal menjadi data yang normal. Bentuk data yang tidak normal/data mentah biasanya disebut juga *unnormalized form*. Tujuan dari normalisasi adalah :

- a. Menghindari inkonsistensi data.
- b. Menghindari terjadinya redundancy data.

1. Inkonsistensi data

Ketika kita mengisi data maka bisa saja terjadi kesalahan pengisian data. Misalnya jika kita mengisi data mahasiswa kedalam sebuah tabel maka bisa saja kita menginputkan data yang salah tetapi tidak ada pengecekan terhadap *database* sehingga data tetap bisa diisikan.

Perhatikan contoh kasus berikut ini:

Tabel 2. 3 Tabel yang Inkonsisten

NIM	Nama	Jurusan	Kepala Jurusan
001	A	TI	X
002	B	TI	X
003	C	MI	Y
004	D	MI	X

Perhatikan tabel tersebut. Kepala jurusan TI bernama X sedangkan kepala jurusan MI bernama Y. pada baris data ke-4, D berasal dari jurusannya adalah X padahal x adalah kepala jurusan TI. Data seperti ini disebut data inkonsisten.

2. Redudancy data

Redudancy data adalah data dalam tabel yang lebih dan tidak berguna. Jika dihapus maka tidak akan berpengaruh pada *database*.

Contoh berikut ini adalah redundancy data :

Tabel 2. 4 Tabel yang redundan

NIM	Nama	Jurusan	Kepala Jurusan
001	A	TI	X
002	B	TI	X
003	C	MI	Y
004	D	MI	X
002	B	TI	X

Pada baris ke-2 dan ke-5, datanya sama. Seharusnya kita hanya memiliki satu data saja. Jadi dapat dihapus salah satu. Contoh lain dari redundancy data adalah penulisan nama kepala jurusan yang berulang-ulang. Setiap kali kita mengisi data kita harus mengisi nama kepala jurusannya. Hal ini jelas memperlambat kerja sehingga untuk jurusan sebaiknya kita pisahkan lagi menjadi sebuah tabel baru yang berisi data jurusan dan nama kepala jurusannya. Pada contoh diatas bisa dibuat tabel baru seperti berikut:

Tabel 2. 5 Memisahkan Tabel

Nim	Nama	Kode Jur
001	A	1
002	B	1
003	C	2
004	D	2

Kode	Kajur	Jur
1	X	TI
2	Y	MI

Sebuah tabel dikatakan sudah normal apabila:

- Nilai tiap kolomnya hanya Satu
- Semua nilai yang diberikan ke setiap kolom harus sesuai dengan tipe data kolom tersebut.
- Nama kolom harus unik di mana setiap kolom memiliki nama yang berbeda.
- Tidak ada 2 record yang sama

- e. Urutan kolom tidak signifikan/tidak penting.
- f. Urutan record juga tidak signifikan

Pada dasarnya terdapat 7 level normalisasi, yaitu sebagai berikut:

- a. First Normal Form (1NF)
- b. Second Normal Form (2NF)
- c. Third Normal Form (3NF)
- d. Boyce-Codd Normal Form (4NF)
- e. Fourth Normal Form (4NF)
- f. Fifth Normal Form (5NF)
- g. Domain-Key Normal Form (DKNF)

Masing-masing level normalisasi mempunyai aturan tersendiri.

3. First Normal Form

Suatu tabel dikatakan dalam keadaan First Normal Form (1NF) jika:

- a. Tidak ada perulangan record data dalam tabel.
- b. Setiap sel memiliki satu nilai saja. Artinya, tidak ada perulangan group dan array.
- c. Data yang diinput memiliki tipe data yang sama dengan tipe data kolom dalam tabel.

4. Second Normal Form

Suatu tabel dikatakan dalam keadaan Second Normal Form (2NF) jika tabel tersebut sudah dalam keadaan First Normal Form (1NF) dan jika semua atribut yang bukan kunci tabel, baik primary key maupun foreign key tergantung pada semua kunci dalam tabel.

5. Third Normal Form

Suatu tabel dikatakan dalam keadaan Third Normal Form (3NF) jika tabel tersebut sudah dalam keadaan Second Normal Form (2NF) dan jika tidak terdapat ketergantungan yang transitif. Artinya, data-data yang mungkin diisi berulang-ulang dapat dibuat sebuah tabel baru.

6. Boyce-Codd Normal Form (BCNF)

Tabel dikatakan dalam keadaan Boyce-Codd Normal Form (BCNF) jika tabel tersebut dalam keadaan Third Normal Form (3NF) dan setiap determinan adalah kunci kandidat.

7. Fourth Normal Form (4NF)

Suatu tabel dikatakan dalam keadaan Fourth Normal Form (4NF) jika tabel tersebut dalam keadaan Boyce-Codd Normal Form (BCNF) dan jika tidak terdapat ketergantungan nilai ganda.

8. Fifth Normal Form (5NF)

Tabel dikatakan dalam keadaan Fourth Normal Form (5NF) jika tabel tersebut dalam keadaan Fourth Normal Form (4NF) dan jika setiap ketergantungan dalam join pada tabel sudah konsekuensi dan kunci kandidat pada tabel tersebut.

9. Domain-Key Normal Form (DKNF)

Tabel dalam keadaan Domain-Key Normal Form (DKNF) jika tabel tersebut dalam keadaan Fifth Normal Form (5NF) dan jika setiap constraint pada tabel adalah konsekuensi logis dari definisi kunci domain.

Pada kenyataannya kita biasanya hanya melakukan proses normalisasi pada level Third Normal Form (3NF). Sebenarnya proses normalisasi sampai pada tahap Third Normal Form (3NF) adalah:

- a. Menghilangkan group yang berulang.
- b. Menghilangkan data redundan.
- c. Menghilangkan kolom-kolom yang tidak tergantung pada key.

Untuk lebih memahami proses normalisasi, perhatikan contoh berikut ini. Jika kita mempunyai tabel data mahasiswa seperti berikut ini:

Tabel 2. 6 Tabel yang Tidak Normal

Nim
Nama
Kode_Jur
Nama_Jur
Kepala_Jur
Kode_MtKul1...n
Nama_MtKul1...n
Nilai_MtKul1...n

Jika kita memasukkan data-data ke tabel tersebut, kita harus memasukkan data mahasiswa berulang-ulang untuk setiap mata kuliah yang berbeda.

Untuk itu kita perlu memisahkan tabel data mahasiswa dengan tabel mata kuliah menjadi 2 tabel yang saling berhubungan. Proses seperti inilah yang dinamakan dengan First Normal Form (1NF).

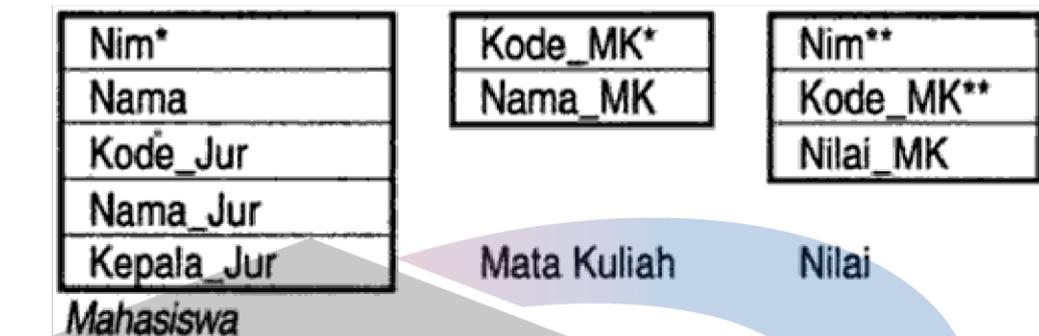
Tabel 2. 7 Hasil First Normal Form (1NF)

Nim*
Nama
Kode_Jur
Nama_Jur
Kepala_Jur

Nim**
Kode_MK
Nama_MK
Nilai_MK
Nilai

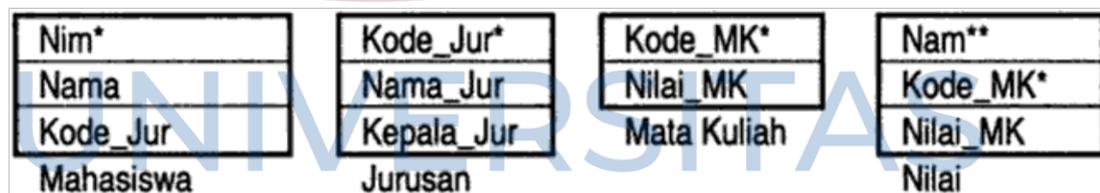
Dalam Pemrograman database, primary key suatu tabel di lambangkan dengan tanda asterik tunggal (*) sedangkan foreign Key dilambangkan dengan tanda asterik ganda (**). Proses normalisasi selanjutnya adalah Second Normal Form (2NF) yang pada perinsipnya adalah memisahkan tabel yang memiliki relasi many-to-many menjadi one-to-many.

Tabel 2. 8 Hasil Second Normal Form (2NF)



Proses normalisasi selanjutnya adalah Third Normal Form (3NF). Pada prinsipnya adalah menghilangkan data redundan. Pengisian data jurusan pada tabel mahasiswa akan berulang-ulang dengan data yang kemungkinan sama. Misalnya ada 10 anak TI maka kita harus mengisi data jurusan TI sebanyak 10 kali dengan data yang sama. Jelas hal ini akan memperlambat kerja kita.

Tabel 2. 9 Hasil Third Normal Form (3NF)



Tabel seperti diatas sudah cukup normal sehingga kita tidak perlu lagi melakukan proses normalisasi selanjutnya. Sebenarnya untuk proses normalisasi selanjutnya akan kita pakai jika database yang akan kita buat termasuk sistem database yang sangat besar. Untuk database-database dalam ukuran sedang cukup sampai dengan Third Normal Form (3NF).

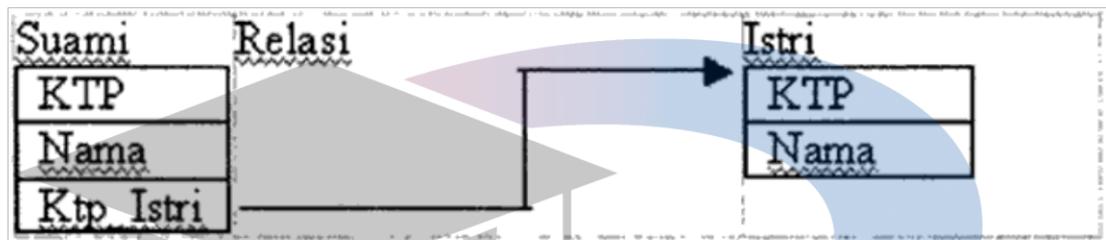
2.13 Relasi Antartabel

Berdasarkan jumlah data yang saling berelasi diantara tabel-tabel yang saling berhubungan, relasi antar tabel dapat dibagi menjadi 3, yaitu:

- a. One-to-one
- b. One-to-many
- c. Many-to-many

Jika hanya satu data dari sebuah tabel tepat berelasi dengan satu data dari tabel lain maka relasi tersebut dinamakan relasi one-to-one.

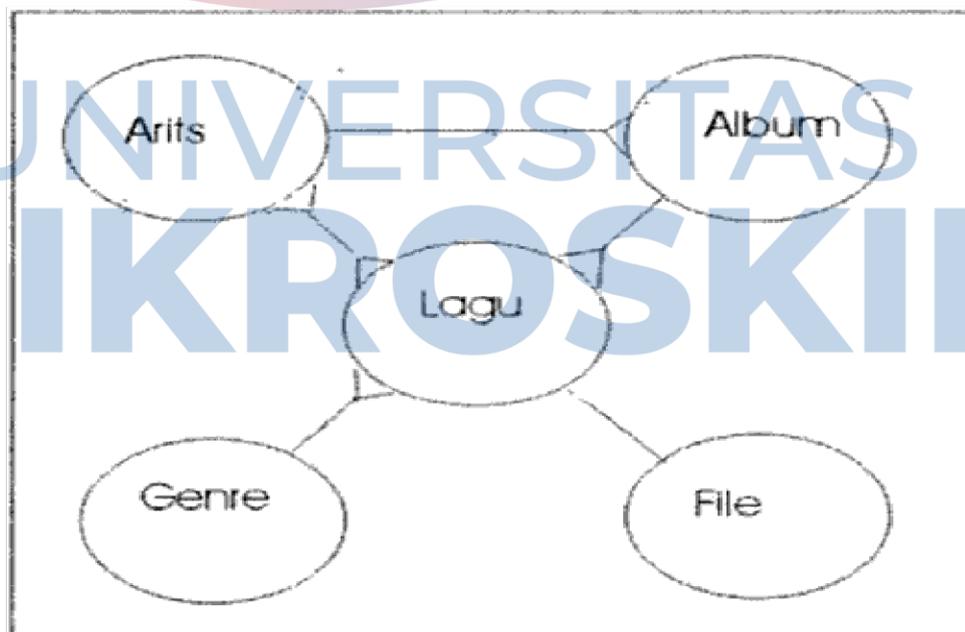
Perhatikan contoh berikut ini:



Gambar 2. 6 Relasi One-To-One

- a. Desain

Pada tahap ini kita merancang hubungan antartabel kedalam bentuk gambar seperti berikut ini:



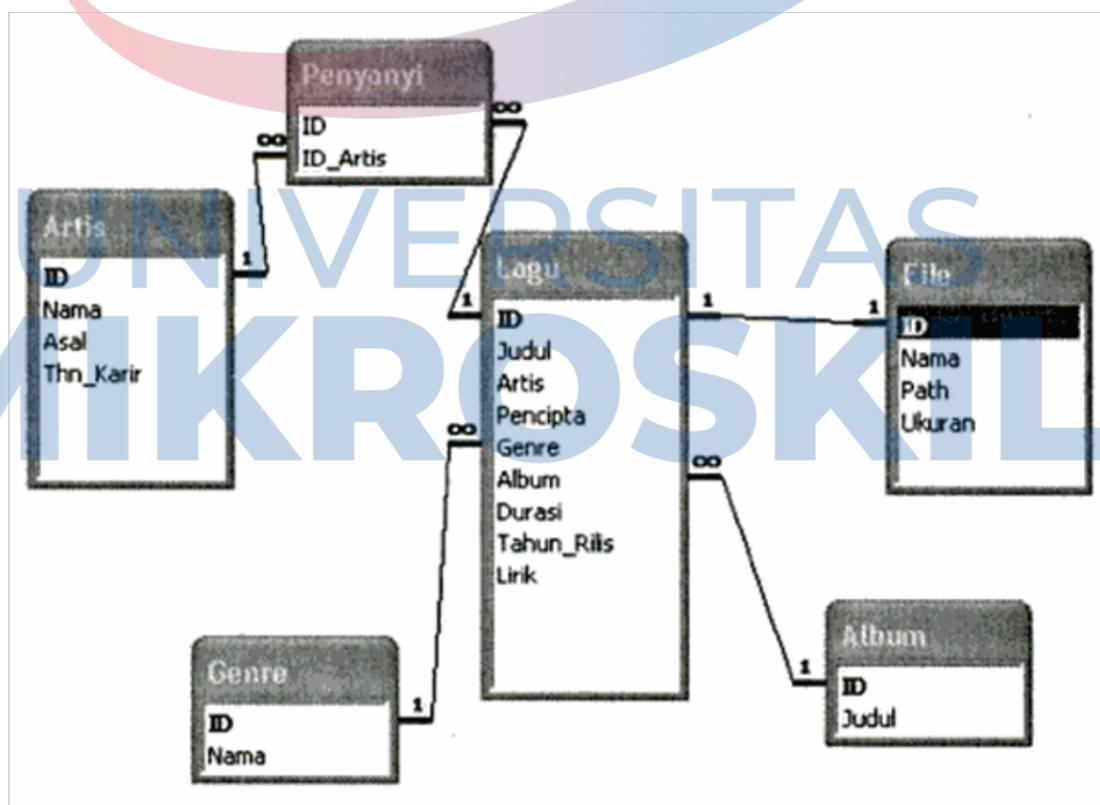
Gambar 2. 7 Relasi Antartabel

Dalam relasi tersebut redundan, yaitu lagu berelasi dengan artis dan lagu juga berelasi dengan album. Selanjutnya artis juga bereasi dengan album sehingga perlu dihilangkan salah satu relasi yang tidak penting, yaitu relasi antara artis dan album. Contohnya redundan data tesebut adalah:

- a. Paterpan ----- Bintang di Surga
- b. Kukatakan dengan indah ----- Bintang di Surga
- c. Kukatakan dengan indah ----- Paterpan (Redundan)

Ingat bahwa dalam merancang databases sebaiknya relasi yang memiliki relasi many-to-many dijadikan relasi one-to-many. Pada contoh tersebut, lagu dan artis mempunyai relasi many-to-many sehingga perlu dilakukan normalisasi ulang sehingga membentuk relasi one-to-many. Untuk itu diperlukan sebuah tabel baru yang bisa menghubungkan antar dua tabel tersebut, misalnya tabel penyanyi. [14]

b. Implementasi



Gambar 2. 8 Relationship Antartabel

Tabel-tabel yang diperlukan adalah seperti pada diagram di atas. Anda bisa membuat tabel-tabel tersebut dengan Ms.Access. [14]

