BAB II
LANDASAN TEORI

2.1. Konsep Dasar Sistem

Sistem adalah kumpulan dari komponen-komponen yang saling terkait dan bekerja bersama ke arah pencapaian tujuan dengan menerima *input* dan menghasilkan *output* di dalam sebuah proses transformasi atau perubahan yang terorganisir (O'Brien, 2002, p8).

Sistem pada dasarnya adalah:

1. Sekelompok *user* yang erat berhubungan satu dengan yang lainnya, yang berfungsi bersama-sama untuk mencapai tujuan tertentu (Mulyadi, 1997, p2).
2. Merupakan sekumpulan sumber daya yang berhubungan untuk mencapai tujuan bersama (Bodnar dan Hopwood, 2000, p1).

Di lain pihak ada dua kelompok dalam mendefinisikan sistem, yaitu yang menekankan pada prosedurnya dan yang menekankan pada komponennya.

Pendekatan sistem yang lebih menekankan pada prosedurnya mendefinisikan sistem sebagai berikut:

Suatu sistem adalah jaringan kerja dari beberapa prosedur yang saling berhubungan, berkumpul bersama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu.[1]

Pendekatan sistem yang lebih menekankan pada komponen atau elemennya mendefinisikan sistem sebagai berikut:

Sistem adalah sekelompok elemen yang terintegrasi dan berinteraksi dengan maksud yang sama untuk mencapai suatu tujuan tertentu.[1]

Sistem itu sendiri memiliki karakterisitik atau beberapa sifat tertentu, yaitu mempunyai komponen (*components*), batas sistem (*boundary*), lingkungan luar sistem (*environments*), penghubung (*interface*), masukan (*input*), keluaran (*output*), pengolah (*proses*), dan sasaran suatu tujuan (*goal*). Adapun penjelasan dari karateristik dari suatu sistem adalah sebagai berikut:

a. Komponen sistem (*Components*)

   Bagian sistem yang saling berinteraksi dan membentuk satu kesatuan. Komponen sistem dapat berupa subsistem atau beberapa bagian sistem.

b. Batas sistem (*Boundary*)

   Daerah yang membatasi antara suatu sistem dengan lingkungannya atau dengan sistem lainnya. Batas sistem inilah yang membuat sistem dipandang sebagai satu kesatuan.

c. Lingkungan Luar Sistem (*Environments*)

   Segala sesuatu yang berada diluar sistem yang mempengaruhi sistem. Lingkungan luar sistem dapat bersifat menguntungkan sistem atau merugikan sistem.

d. Penghubung Sistem (*Interface*)

   Merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Penghubung inilah yang menyebabkan beberapa subsistem berintegrasi dan membentuk satu kesatuan.

e. Masukan Sistem (*Input*)

Sesuatu yang dimasukkan ke dalam sistem yang berasal dari lingkungan.

f. Keluaran Sistem (*Output*)

Suatu hasil dari proses pengolahan sistem yang dikeluarkan ke lingkungan.

g. Pengolah Sistem (Proses)

Bagian dari sistem yang mengubah masukan (input) menjadi keluaran (output).

h. Sasaran Sistem (*Objectives*) atau Tujuan (*Goal*)

Sasaran sistem adalah sesuatu yang menyebabkan mengapa sistem itu dibuat atau ada. Suatu sistem dikatakan berhasil bila tepat sasaran atau tujuannya.

Adapun definisi sistem menurut Van de Poel yang diterjemahkan oleh Winardi adalah: sebagai sekumpulan elemen dimana terdapat adanya hubungan-hubungan. Kerapkali dalam literatur dapat diketemukan kata-kata tambahan elemen-elemen mana ditunjukan ke arah pencapaian sasaran umum tertentu.[2]

Dari pengertian diatas dapat diuraikan bahwa suatu sistem terdiri dari sekumpulan elemen yang saling berhubungan satu sama lain yang memiliki pencapaian sasaran yang telah ditentukan,

Sedangkan Shrode dan Voich dalam Lukman mengatakan bahwa "*system is whole compounded of several parts...*" yang dapat diartikan, *sistem adalah suatu keseluruhan yang tersusun dari sekian banyak bagian*. Senada dengan pendapat diatas maka Awad menyatakan bahwa *system is an organized, funtioning, relationship among unit or components*". Yang dapat diartikan, *sistem adalah hubungan yang berlangsung diantara satuan-satuan atau komponen secara teratur[2].*

Riggs dalam Winardi mengemukan pengertian *sistem sebagai suatu kelompok yang terdiri dari beberapa komponen atau bagian yang saling berinteraksi dalam hubungan yang dinamis.[2]*

Kemudian Johnson, Katzs dan Rosenzwig dalam Lukman mendefinisikan *sistem sebagai suatu kebulatan atau keseluruhan yang kompleks atau terorganisir; suatu himpunan atau perpaduan hal-hal atau bagian – bagian yang membentuk suatu kebulatan atau keseluruhan yang kompleks atau utuh.[2]*

Selanjutnya menurut Winardi menyatakan bahwa pada intinya *sistem menunjuk pada dua hal, yaitu pertama menunjukan suatu identitas, suatu wujud benda (abstrak maupun konkrit, termasuk yang konseptual) yang memiliki susunan struktural dari bagian-bagiannya ; kedua menunjukan suatu metode atau tata cara yang menunjuk pada suatu rencana metode, alat atau tata cara untuk mencapai sesuatu.[2]*

Sejalan dengan pendapat di atas, Lukman mengartikan *sistem sebagai wujud (entitas).* Suatu sistem seringkali dianggap sebagai suatu himpunan bagian yang saling berhubungan sehingga merupakan suatu kesatuan. Contoh wujud (entitas) adalah lembaga pemerintah yang berkewajiban memberi pelayanan yang berkualitas (pelayanan prima) kepada pelanggannya.[2]

2.2. Konsep Dasar Informasi

Informasi dalam sebuah organisasi sangat penting peranannya. Suatu sistem yang kurang mendapatkan informasi akan menjadi lemah dan akhirnya berakhir. Informasi itu sendiri dapat didefinisikan sebagai berikut:

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya.[1]

Sumber dari informasi adalah data. Data merupakan bentuk jamak dari bentuk tunggal data atau data item. Data adalah kenyataan yang menggambarkan sesuatu yang terjadi pada saat tertentu.

2.2.1. Konsep Dasar Sistem Informasi

Informasi dapat diperoleh dari sistem informasi (*Information system*) atau disebut juga dengan *processing system* atau *information processing system* atau *information genereting system*. Sistem informasi didefinisikan oleh Robert A. Leitch dan K. Roscoe Davis sebagai berikut:

Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan yang diperlukan.[1]

Sistem informasi sendiri memiliki sejumlah komponen tertentu. Seperti yang dikemukakan oleh Robert dan Donald Symanzky, bahwa sistem informasi terdiri dari beberapa komponen yang berbeda yaitu, manusia, data, hardware, dan software. Sebagai suatu sistem, setiap komponen tersebut berinteraksi satu dengan lainnya membentuk satu kesatuan untuk mencapai sasarannya.

2.2.2. Konsep Dasar Basisdata

Istilah basis data banyak menimbulkan interpretasi yang berbeda. Pada saat maraknya perangkat lunak dBase II dan dBase II plus, sebuah berkas (dengan

Ekstansi .DSF) biasa disebut basis data. Istilah yang tidak tepat ini, meskipun telah merasuk ke sejumlah pemrogram, akhirnya diluruskan kembali oleh pencipta perangkat lunak basis data yang lain. George Tsu-der Chow, mendefinisikan basis data sebagai kumpulan informasi bermanfaat yang yang diorganisasikan kedalam tatacara yang khusus. Menurut Anthoni J. Fabbri dan A. Robert Schwab, basis data adalah sistem berkas terpadu yang di rancang terutama untuk memilnimalkan pengulangan data. Menurut C. J. Date, sistem basis data pada dasarnya adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara informasi dan membuat informasi tersebut tersedia saat dibutuhkan.

Ada juga yang berpendapat bahwa basis data yaitu suatu kumpulan data yang dapat di*share*, yang berhubungan secara logikal, dan deskripsi dari data tersebut, dirancang untuk memenuhi kebutuhan informasi suatu organisasi (Connolly dan Begg, 2002, p14).

Basis data adalah kumpulan data yang digunakan untuk menghadirkan informasi dari suatu sistem informasi yang menarik (Atzeni, 2003, p2).

Menurut Date (2000, p10), basis data merupakan kumpulan data *persistent* yang digunakan oleh sistem aplikasi pada beberapa perusahaan. Sementara O'Brien (2002, p469) mendefinisikan basis data sebagai kumpulan yang terintegrasi dari *record* atau objek yang terkait secara logikal.

Sedangkan pengertian sistem basis data adalah komputerisasi sistem penyimpanan data yang bertujuan menyimpan dan memelihara informasi

serta mengizinkan *user* untuk melakukan peng-*update*-an atau pengambilan data yang dibutuhkan (Date, 2000, p5).

Pengguna sistem basis data bisa melakukan berbagai operasi, antara lain:

a. Menambahkan file baru ke sistem basis data
b. Mengosongkan berkas
c. Menyisipkan data ke suatu berkas
d. Mengambil data yang ada pada suatu berkas
e. Mengubah data pada suatu berkas
f. Menghapus data pada suatu berkas
g. Menyajikan suatu informasi yang di ambil dari sejumlah berkas.

Kelebihan Basis Data

1. Kecepatan dan kemudahan (*speed*)

   Dengan menggunakan basis data pengambilan informasi dapat dilakukan dengan cepat dan mudah.

2. Kebersamaan pemakai (*Sharability*)

   Sebuah basis data dapat digunakan oleh banyak user dan banyak aplikasi.

3. Pemusatan kontrol data

   Karena cukup dengan satu basis data untuk banyak keperluan, pengontrolan terhadap data juga cukup dilakukan di satu tempat saja.

4. Efisiensi ruang penyimpanan (*space*)

   Dengan pemakaian bersama, kita tidak perlu menydiakan tempat penyimpanan diberbagai tempat, tetapi cukup satu saja sehingga ini akan menghemat ruang penyimpanan yang dimiliki oleh sebuah organisasi.

5. Keakuratan (*accuracy*)

   Penerapan secara ketat aturan tipe data, domain data, keunikan data, hubungan antara data dan lain-lain, dapat menekan ketidakakuratan dalam pemasukan/ penyimpanan data.

6. Ketersediaan (*availability*)

   Dengan basis data kita dapat membackup data, memilah-milah data mana yang masih diperlukan dan data mana yang perlu kita simpan ketempat lain.

7. Kelengkapan (*completeness*)
8. Keamanan (*security*)

   Kebanyakan DBMS dilengkapi dengan fasilitas manajemen pengguna.

9. Kemudahan dalam pembuatan program aplikasi baru

   Penggunaan basis data merupakan bagian dari perkembangan teknologi. Dengan adanya basis data pembuatan aplikasi bisa memanfaatkan kemampuan dari DBMS.

10. Pemakaian secara langsung

    Basis data memiliki fasilitas untuk melihat datanya secara langsung dengan tool yang disediakan oleh DBMS.

11. Kebebasan Data (*Data Independence*)

    Jika sebuah program telah selesai dibuat, dan ternyata ada perubahan isi/struktur data. Maka dengan basis data, perubahan ini hanya perlu dilakukan pada level DBMS tanpa harus membongkar kembali program aplikasi.

12. User view

    Basis data menyediakan pandangan yang berbeda-beda untuk tiap-tiap pengguna.

2.2.3. Database Management System (DBMS)

Berdasarkan pendapat James O'Brien (2002, p146), DBMS berperan sebagai penghubung antara *user* dengan basis data. DBMS membantu *user* dalam mengakses data dalam basis data. DBMS menggunakan suatu *software* manajemen basis data untuk mengatur pembuatan basis data, penggunaannya, dan pemeliharaannya untuk menyediakan informasi yang diperlukan oleh *user* dan organisasi.

Menurut Connolly dan Begg (2002, p16), DBMS adalah suatu *software* yang memperbolehkan pemakai untuk mendefinisikan (*define*), membuat, memelihara (*maintain*), dan mengendalikan akses ke basis data. Selain itu, DBMS adalah suatu *software* yang menghubungkan antara pemakai program aplikasi dengan basis data.

Secara umum DBMS menyediakan beberapa fasilitas:

1. Memperbolehkan pemakai mendefinisikan (*define*) basis data, biasanya melalui *Data Definition Language* (DDL). DDL memperbolehkan pemakai untuk menspesifikasikan tipe, struktur data, dan jenis-jenis data yang dimasukkan dalam basis data.

2 Memperbolehkan pemakai untuk melakukan *insert, update, delete*, dan *retrieve* data dari basis data, biasanya menggunakan *Data Manipulation Language* (DML). Adanya penyimpanan (*repository*) pusat untuk semua data beserta deskripsinya, memperbolehkan DML untuk menyediakan sebuah fasilitas *inquery* umum, yang disebut *query language.*

3 Menyediakan pengendalian akses ke basis data. Contohnya :

   a. Sistem keamanan, di mana menghindari pemakai yang tidak memiliki izin (*unauthorized*) untuk mengakses basis data.
   b. Sistem integritas, di mana memelihara konsistensi data yang dimasukkan.
   c. *Concurrency Control System*, di mana memperbolehkan untuk pembagian akses ke basis data.
   d. Sistem pengendali pemulihan (*recovery control system*), di mana berfungsi untuk mengembalikan basis data ke kondisi sebelumnya jika terjadi kegagalan pada *software* atau *hardware.*
   e. *User accessible catalog*, di mana berisikan deskripsi dari data yang ada di basis data.

DBMS melakukan tiga aktivitas dasar :

1. Meng-*update* dan memelihara basis data umum untuk transaksi bisnis baru dan kegiatan lainnya yang memerlukan perubahan pada data organisasi.
2. Menyediakan informasi yang dibutuhkan oleh setiap aplikasi dengan menggunakan program aplikasi yang menggunakan data pada basis data umum. *User* dan *programmer* tidah harus tahu dimana dan bagaimana data tersebut disimpan.

Melakukan tanggapan dan memberikan laporan melalui *software* DBMS sehingga *user* dapat menggunakan *web browser* dan *internet* atau *intranet* untuk mengakses, membuat laporan, dan menerima respon yang cepat terhadap permintaan informasi.

Keuntungan DBMS

1. Mengurangi pengulangan data.

2. Mencapai independensi data.

   a. Spesifikasi data disimpan dalam tiap program aplikasi.
   b. Perubahan dapat dibuat pada struktur data tanpa mempengaruhi program yang mengakses data.

3. Mengintegrasikan data dari beberapa file.

   Saat file dibentuk sehingga menyediakan kaitan logis, organisasi fisik tidak lagi menjadi kendala.

4. Mengambil data dan informasi secara cepat.

   Hubungan logis query language memungkinkan pemakai mengambil data dalam hitungan detik atau menit.

5. Meningkatkan keamanan.

   Baik DBMS mainframe maupun komputer mikro dapat menyertakan beberapa lapis keamanan seperti kata sandi, directori pemakai, dan bahasa sandi.

Kerugian DBMS

1. Memperoleh perangkat lunak yang mahal.
2. Memperoleh konfigurasi perangkat keras yang besar.
3. Memperkerjakan dan mempertahankan staf DBA.

2.2.4. Normalisasi

Normalisasi adalah teknik yang digunakan untuk menstrukturkan data sedemikian rupa sehingga mengurangi atau mencegah timbulnya masalah-masalah yang berhubungan dengan pengolahan basis data. Proses normalisasi didalam model basis data relasional menitik beratkan pada masalah penentuan struktur data yang paling sederhana untuk tabel-tabelnya.

Hasil proses normalisasi adalah data, records atau tabel-tabel yang konsisten secara logika, mudah dimengerti dan pemeliharaanya tidak sulit dan murah. Oleh karena itu, dalam perancangan skema basis data dalam bentuk normal.

Adapun konsep-konsep yang digunakan pada normalisasi, antara lain:

a. Kunci atribut (a*tribut key*), yaitu suatu kunci field yang memiliki record.
b. Kunci kandidat (*candidate key*), yaitu suatu atribut atau satu set atribut yang mendefinisikan secara unik dari suatu entity.
c. Kunci primer (*prymary key*), yaitu suatu atribut atau satu set atribut yang mendefinisikan secara unik dan mewakili setiap kejadian pada suatu entity.
d. Kunci alternatif (*alternative key*), yaitu kunci kandidate yang dipakai sebagai kunci primer.
e. Kunci tamu (*foreign key*), yaitu satu atribut atau satu set atribut dan melengkapi hubungan yang menunjukkan keinduknya

Bentuk dan tahapan dalam proses normalisasi adalah sebagai berikut:

1. Bentuk Tidak Normal (*Unnormalized Form*)

   Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti suatu format tertentu, dapat saja data tidak lengkap atau terduplikasi. Data dikumpulkan apa adanya sesuai dengan kedatangannya.

2. Bentuk Normal Pertama (1NF)

   Suatu tabel dapat disebut sebagai bentuk nomal pertama jika semua atributnya memiliki nilai yang atomik (atribut yang bersangkutan tidak dapat dibagi lagi menjadi atribut-atribut yang lebih kecil), tetapi masih mengandung redudanci (atribut yang tampil berulang).

3. Bentuk Normal Kedua (2NF)

   Bentuk normal kedua adalah suatu tabel bentuk normal pertama yang memenuhi syarat tambahan bahwa semua atribut yang bukan atribut kunci hanya bergantung pada kunci primer.

4. Bentuk Normal Ketiga (3NF)

   Bentuk normal ketiga adalah suatu bentuk normal kedua yang memenuhi syarat tambahan bahwa semua atribut yang bukan atribut kunci tidak memiliki ketergantungan transitif terhadap kunci primer.

5. Bentuk Normal Boyce-codd (BCNF)

   Bentuk normal Boyce-codd adalah suatu tabel yang memiliki semua field penentu yang merupakan kunci kandidat. Bentuk ini merupakan perbaikan dari bentuk normal ketiga.

2.2.5. Diagram ER (Entity Relationship Diagram)

Entity Relationship Diagram adalah sebuah diagram yang menggambarkan model relasi antar rancangan data tersimpan. Model relasi ini diperlukan untuk menggambarkan struktur data dari relasi antar data. Model data ERD dibentuk dari empat komponen dasar yaitu :

1. Entitas

   Entitas adalah segala sesuatu yang ada dan yang dapat dibedakan. Entitas dapat berupa orang, tempat, benda, peristiwa atau konsep yang bisa memberikan atau mengandung informasi.

2. Atribut

   Setiap entitas memiliki atribut-atribut atau elemen data yang mencirikan entitas tersebut.

3. Kunci (*Key*)

   Sebuah atribut atau set atribut yang nilainya mengidentifikasikan entitas secara unik dalam set entitas.

4. Relasi

   Relasi menunjukan adanya hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda.

Hubungan antara entitas atau beberapa entitas jenis relasi yang ada di dalam basis data adalah sebagai berikut :

A. Relasi banyak ke satu (n-1)

   Misalkan direpresentasikan dalam pencantuman atribut kunci dari entitas B (berderajat 1) ke himpunan entitas A (berderajat n), ini berarti setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan B, tetapi tidak sebaliknya.

B. Relasi banyak ke banyak (n-m)

   Artinya setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B dan demikian juga sebaliknya.

C. Relasi satu ke banyak (1-n)

Misalkan di representasikan dalam bentuk pencantuman atribut kunci dari himpunan entitas A (berderajat 1) ke himpunan entitas B (berderajat n), ini berarti setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi tidak sebaliknya.

D. Relasi satu ke satu (1-1)

Ini berarti setiap entitas pada himpunan entitas A berhubungan dengan paling banyak satu entitas pada himpunan entitas B dan begitu juga sebaliknya.

2.2.6. Data Flow Diagram (DFD)

DFD adalah suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut.

DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau dimana data tersebut akan disimpan.

DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur. Kelebihan utama pendekatan aliran data, yaitu :

1. Kebebasan dari menjalankan implementasi teknis sistem.

2. Pemahaman lebih jauh mengenai keterkaitan satu sama lain dalam sistem dan subsistem.
3. Mengkomunikasikan pengetahuan sistem yang ada dengan pengguna melalui diagram aliran data.
4. Menganalisis sistem yang diajukan untuk menentukan apakah data-data dan proses yang diperlukan sudah ditetapkan.

Disamping itu terdapat kelebihan tambahan, yaitu :

1. Dapat digunakan sebagai latihan yang bermanfaat bagi penganalisis, sehingga bisa memahami dengan lebih baik keterkaitan satu sama lain dalam sistem dan subsistem.
2. Membedakan sistem dari lingkungannya dengan menempatkan batas-batasnya.
3. Dapat digunakan sebagai suatu perangkat untuk berinteraksi dengan pengguna.
4. Memungkinkan penganalisis menggambarkan setiap komponen yang digunakan dalam diagram.

DFD terdiri dari *context diagram* dan diagram rinci (DFD Levelled). *Context diagram* berfungsi memetakan model lingkungan (menggambarkan hubungan antara entitas luar, masukan dan keluaran sistem), yang direpresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem. DFD levelled menggambarkan sistem sebagai jaringan kerja antara fungsi yang berhubungan satu sama lain dengan aliran dan penyimpanan data, model ini hanya memodelkan sistem dari sudut pandang fungsi.

Dalam DFD level akan terjadi penurunan level dimana dalam penurunan level yang lebih rendah harus mampu merepresentasikan proses tersebut ke dalam spesifikasi proses yang jelas. Jadi dalam DFD levelled bisa dimulai dari DFD level 0 kemudian turun ke DFD level 1 dan seterusnya. Setiap penurunan hanya dilakukan bila perlu. Aliran data yang masuk dan keluar pada suatu proses di level x harus berhubungan dengan aliran data yang masuk dan keluar pada level x+1 yang mendefinisikan proses pada level x tersebut. Proses yang tidak dapat diturunkan/dirinci lagi dikatakan primitif secara fungsional dan disebut sebagai proses primitif.

DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data itu mengalir. DFD merupakan alat yang digunakan pada metodologi pembangunan sistem yang terstruktur. DFD merupakan alat yang cukup popular saat ini, karena dapat menggambarkan arus data didalam sistem dengan terstruktur dan jelas. DFD juga merupakan dokumentasi sistem yang baik. Simbol yang digunakan DFD :

a. Kesatuan Luar

Kesatuan luar (*external entity*) merupakan kesatuan dilingkungan luar sistem yang dapat berupa orang, organisasi, atau sistem lainnya yang akan memberikan input atau menerima output dari sistem, disimbolkan dengan suatu kotak.

b. Arus Data

Arus data (*data flow*) diberi simbol suatu panah. Arus data ini mengalir diantara proses (*process*), simpanan data (*data store*) dan kesatuan luar (*external entity*). Arus data ini menunjukan arus data yang dapat berupa masukkan untuk sistem atau hasil dari proses sistem.

c. Proses

Suatu proses adalah kegiatan atau kerja yang dilakukan oleh orang, mesin atau komputer dari hasil suatu arus data yang masuk kedalam proses untuk dihasilkan arus data yang akan keluar dari proses. Suatu proses dapat ditunjukkan dengan simbol empat persegi panjang bersudut-sudut tumpul atau sebuah lingkaran.

Setiap proses harus diberi penjelasan yang lengkap meliputi:

i. Identifikasi proses

Identifikasi ini umumnya berupa suatu angka yang menunjukkan nomor acuan dari proses dan ditulis pada bagian atas simbol.

ii. Nama proses

Nama proses menunjukkan apa yang dikerjakan oleh proses tersebut.

d. Simpanan Data

Merupakan simpanan dari data.

Langkah-langkah penyusunan DFD:

1. Identifikasikan terlebih dahulu semua kesatuan luar yang terlibat di sistem.
2. Identifikasikan semua input dan output yang terlibat dengan kesatuan luar.
3. Gambarkan terlebih dahulu suatu *diagram context*.

4. Gambarkan bagan berjenjang untuk semua proses yang ada di sistem terlebih dahulu.
5. Gambarkan sketsa DFD untuk *overview diagram* (level 0) berdasarkan proses di bagan berjenjang.
6. Gambarkan DFD untuk level-level berikutnya, yaitu level satu dan seterusnya untuk tiap-tiap proses yang dipecah-pecah sesuai dengan bagan berjenjangnya.
7. Setelah semua level DFD digambar, berikutnya adalah menggambar DFD untuk pelaporan manajemen yang digambar terpisah. Setelah semua level DFD dan DFD untuk pelaporan manajemen digambar, maka semua DFD ini dapat digabungkan dalam satu diagram.

2.2.7. Metode Pengujian Sistem

1. White-Box

Pengujian *white box* dilakukan untuk menguji prosedur-prosedur yang ada. Lintasan lojik yang dilalui oleh setiap bagian prosedur diuji dengan memberikan kondisi/loop spesifik. Pengujian *white box* menjamin pengujian terhadap semua lintasan yang tidak bergantungan minimal satu kali, mencoba semua keputusan lojik dari sisi '*true*' dan '*false*', eksekusi semua *loop* dalam batasan kondisi dan batasan operasionalnya dan pengujian validasi data internal.

Konsep Pengujian Basis *Path*

a. Merupakan bagian dari pengujian *white-box* dalam hal pengujian prosedur-prosedur;

b. Mempergunakan notasi aliran *graph* (*node*, *link* untuk merepresentasikan *sequence*, *if*, *while*, *until* dll.);
c. Konsep kompleksitas *cyclomatic* antara lain cara perhitungan daerah tertutup pada graph planar dimana dapat menghubungkan batas atas jumlah pengujian yang harus direncanakan dan dieksekusi untuk menjamin pengujian seluruh *statement* program.
d. Memunculkan kasus-kasus yang akan diuji dengan membuat daftar lintasan kasus pengujian berdasarkan kompleksitas dan *cyclomatic* yang didapat.
e. Membuat alat bantu matrik graph yang membantu pengawasan pengujian.

2. *Black-Box*

Pengujian yang dilakukan untuk antarmuka perangkat lunak, pengujian ini dilakukan untuk memperlihatkan bahwa fungsi-fungsi bekerja dengan baik dalam arti masukan yang diterima dengan benar dan keluaran yang dihasilkan benar-benar tepat, pengintegrasian dari eksternal data berjalan dengan baik.

Metode pengujian *black-box* memfokuskan pada *requirement* fungsi dari perangkat lunak, pengujian ini merupakan komplenetari dari pengujian *white-box*. Pengujian *white-box* dilakukan terlebih dahulu pada proses pengujian, sedangkan pengujian *black-box* dilakukan pada tahap akhir dari pengujian perangkat lunak.

Proses yang terdapat dalam proses pengujian *black-box* antara lain sebagai berikut :

a. Pembagian kelas data untuk pengujian setiap kasus yang muncul pada pengujian *white-box*;
b. Analisis batasan nilai yang berlaku untuk setiap data.

2.3 Software Pendukung

1. Borland Delphi

Borland delphi (delphi) adalah paket pahasa pemrograman yang bekerja dalam sistem operasi Windows. Delphi merupakan bahasa pemrograman yang mempunyai cakupan kemampuan yang luas dan sangat canggih. Berbagai jenis aplikasi dapat anda buat dengan Delphi, termasuk aplikasi untuk mengolah teks, grafik, angka, database, dan aplikasi web.

Secara umum, kemampuan Delphi adalah menyediakan komponen-komponen dan bahasa pemrograman yang andal, sehingga memungkinkan anda untuk membuat program aplikasi yang canggih, sesuai dengan keinginan. Sampai dengan saat ini , Delphi rilis terakhir adalah Delphi 2007. dalam pembuatan program ini, penulis menggunakan Delphi 7.0.

Untuk mempermudah pemrogram dalam membuat program applikasi, Delphi menyediakan fasilitas pemrograman yang sangat lengkap. Fasilitas pemrograman tersebut dibagi dalam dua kelompok, yaitu object dan bahasa pemrograman.

Secara ringkas, object adalah suatu komponen yang mempunyai bentu fisik dan biasanya dapat dilihat (visual). Object biasanya dipakai untuk melakukan

tugas tertentu dan mempunyai batasa-batasan tertentu. Sedangkan bahasa pemrograman secara singkat dapat disebut sebagai sekumpulan teks yang mempunyai arti tertentu dan disusun dengan aturan tertentu serta untuk menjalankan tugas tertentu.

Delphi menggunakan struktur bahasa pemrograman Object Pascalyang sudah sangat dikenal dikalangan pemrogram fropesional. Gabungan dari object-object dan bahasa pemrograman ini, sering disebut bahasa pemrograman berorientasi object atau Object Oriented Proramming (OOP).

Khusus untuk pemrograman database, Delphi menydiakan object yang sangat kuat, canggih dan lengkap, sehingga memudahkan pemrogram dalam merancang, membuat dan menyelesaikan aplikasi database yang diinginkan. Selain itu, Delphi juga dapat menangani data dalam berbagai format database, misalnya format MS-Access, SyBase, Oracle, FoxPro, Informix, DB2, SQL Server, dan lain-lain. Format database yang dianggap asli dari Delphi adalah Paradox dan dBase. Dalam membuat progran ini penulis menngunakan database SQL Server.

Borland Database Engine (BDE) adalah salah satunya teknologi yang ditawarkan untuk mengakses database paada beberapa versi pertama dari Delphi, borland mulai mengembangkan beberapa VCL yang berhubungan dengan akses database untuk memberikan beberapa solusi atau cara pengaksesan database. Berikut adalah penjelasan singkat tentang teknologi-tekonologi akses database yag ada di Delphi.

Mengakses Database

Berikut adalah penjelasan singkat tentang teknologi-teknologi akses databaseyang ada di Delphi:

***Borland Database Engine* (BDE)**

Kelebihan BDE:

1) Aplikasi dapat sangat portabel karena mudah untuk memindahkan database antar server.
2) Sangat terintegrasi dengan Delphi.
3) Elemen-elemenya terdokumentasi lengkap.

Kekurangan BDE:

1) Borland berhenti mengembagkannya lagi
2) Harus diinstal dan diatur dikomputer cilent.
3) BDE termasuk kelas berat, membutuhkan file instalasi dan memori yang besar.
4) Hanya tersedia untuk sistem operasi Windows.

***ActiveX Data Objects* (ADO)**

Kelebihan ADO:

1) Lebih kompatibel dan kualitas yang lebih baik dari rata-rata OLEDB Providers jika menggunakan Access atau MS SQL Serve karena driver berasal dari Microsoft.

2) Sangat berdayaguna.

Kekurangan ADO:

1) Hanya tersedia untuk Windows, tidak tersedia untuk sistem operasi lain.

dbExpress Library

Kelebihan *dbExpress Library*:

1) Karakteristiknya yang ringan dan portabel.

2) Dataset undirectional tanpa update langsung adalah standart untuk menghasilkan report.

Kekurangan *dbExpress Library*:

1) Hanya dapat mengaksess database server tertentu.
2) Tidak ada kemampuan *catch, dan hanya dapat mengakses data* unibidirectional.
3) Dapat bekerja dengan SQL query, tapi belum dapat melakukan statement generate SQL update.

***InterBase Express* (IBX)**

Kelebihannya

jika ingin mengganti database, dan ingin memperoleh kinerja dan kontrolyang baik dengan sifat fleksibel dan fortabelnya, Interbase dapat dijadikan pertimbagan.

Kekurangannya

Interbase membutuhkan kinerja extra dan kontro yang diperoleh terbatas.

2. MySQL

MySQL merupakan salah satu database relasional yang mendukung pemakaian *Structured Query Language* (SQL) dan dirancang untuk penggunaan aplikasi dengan arsitektur *client server*. MySQL memungkinkan pengguna untuk mengolah data di dalam *database* yang tersentral pada komputer pusat yang disebut sebagai *server*. Sedangkan informasi yang dihasilkan dapat digunakan bersama-sama oleh beberapa *user*. Semua komunikasi yang terjadi mendukung perintah-perintah SQL (*Query*).

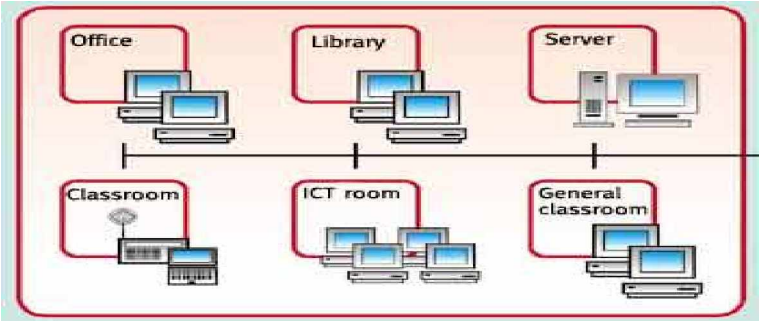
Secara umum, kelebihan MySQL adalah tersedia gratis untuk banyak sistem operasi dan hardware, memiliki kecepatan proses dan kemampuan menangani database dengan sangat baik di banyak tipe hardware dan sistem operasi yang digunakan. Selain itu MySQL juga mudah digunakan karena bahasa yang digunakannya termasuk simple, mendukung SQL, sebuah bahasa untuk semua database sistem yang modern, mampu bekerja dengan beban berat dengan banyak *client* yang terhubung dalam satu waktu secara bersamaan, dapat diakses dari mana saja dari *internet*, mudah diperoleh dan faktor keamanan yang terjamin.

2.4 Jaringan Komputer

Lokal Area Network (LAN)

Lokal Area Network ini digunakan untuk menghubungkan simpul yang Berada disuatu area dengan radius 100 meter sampai 2 Km tergantung jenis kabel yang digunakan apakah menggunakan kabel Unshielded Twisted Pair (UTP), Kabel Shielded Twiste Pair (STP), Coaxial cable, atau menggunakan Kabel Fiber Optik.

Kecepatan transfer data pada jaringan lokal ini sudah relatif tinggi yaitu antara 1 – 100 Mbps atau sekitar 125.000 – 12.500.000 karakter perdetik. Lokal Area Network tidak menggunakan Fasilitas dari perusahaan publik, seperti sambungan telepon atau Frekuensi radio, adapun topologi yang dapat dipakai dalam Lokal Area Network bisa digunakan Linear bus, Star, Ring, atau Tree, dengan model hubungan bisa Peer to Peer atau Client - Server .



Gambar 2.1 *Lokal Area Network*

Bilamana tersedia dua buah komputer yang memiliki Network Interface Card (NIC) atau kartu jaringan baik berupa Ethernet, Local Talk Connector atau Token Ring Card pada masing-masing komputer yang kemudian dihubungkan dengan kabel jaringan dan dilakukan instalasi sistem operasi jaringan seperti Novel Netware, Windows, atau Linux maka jadilah sebuah jaringan lokal, adapun untuk jumlah komputer lebih banyak dan terbagi kedalam beberapa network bisa menggunakan Bridges atau Router, sedangkan untuk memperkuat sinyal bisa menggunakan Repeater.

IP Address

Alamat dalam internet disebut IP Address atau disebut juga network Adress. Ip adress ini penting sekali, sebab Ip adress ini yang akan digunakan oleh sebuah

protokol yang mengatur rute dan mengidentifikasi komputer tujuan / asal internet, protokol tersebut adalah internet protokol (IP). Dengan IP Address ini, protokol IP dapat mengenali komputer dengan membuet rute untuk mengirim data agar sampai ke komputer tujuan yang berada di enta belahan dunia mana.

Sebuah IP Address berupa sebuah bilangan biner 32 bit. Sehingga total IP Address adalah 2^{32}. Jadi space sebanyak 429.967.296 dengan konsep IP Address 32 bit, atau disebut IP versi 4. Suatu saat IP Address ini akan habis terpakai. Tapi tidak perlu khawatir, sebab sudah ditemukan IP versi 6 yang spacenya jauh lebih tinggi dari IP versi 4. IP versi 4 ini dibagi kedalam 4 kelas dan Ip nya dibagi kedalam 4 segmen adalah bilangan bulat desimal setiap segmen panjangnya 8 bit sehingga memiliki range antara 0-255 alamt ini dipecahkan menjadi dua bagian yaitu bagian pertama menunjukkan host dalam network tersebut (tabel 2.1).

Tabel 2.1 *format IP Address*

Network Address	Host Address

Tabel 2.2 *Tipe IP Address*

Bit segmen pertama	Range bit segmen pertama	kelas	Network	Host	penggunaan
0...	0-127	A	Segmen 1	Segmen 2,3,4	Untuk Jaringan Bebas
10...	128-191	B	Segmen 1,2	Segmen 3,4	Untuk jaringan menengah
100...	192-223	C	Segmen 1,2,3	Segmen 4	Untuk Jaringan Kecil
1110...	244-247	D	Segmen 1,2,3	Segmen 4	Cadangan: IP Multicasting
1111…	248-255	E	Segmen 1,2,3	Segmen 4	Cadangan: u/ eksperimen