BAB II

TINJAUAN PUSTAKA

2.1 Konsep Dasar Sistem

Suatu sistem dapat didefinisikan sebagai suatu kesatuan yang terdiri dari dua atau lebih komponen atau sub sistem yang berinteraksi untuk mencapai suatu tujuan. Suatu sistem dapat terdiri dari sistem-sistem bagian (sub sistem). Sub sistem-sub sistem saling berinteraksi dan saling terhubung membentuk satu kesatuan sehingga tujuan atau sasaran sistem tersebut dapat tercapai. Interaksi dari sub sistem-sub sistem sedemikian rupa, hingga dicapai suatu kesatuan yang terpadu atau terintegrasi *(intergrated*). Jika seandainya sistem komputer masing-masing komponennya saling bekerja sendiri-sendiri tidak terintegrasi, maka tujuan dari sistem komputer tersebut tidak tercapai.

2.1.1 Pengertian Sistem

Setiap kegiatan dalam suatu organisasi tentu memiliki suatu tujuan yang ingin dicapai, kegiatan ini tidak terlepas dari peran suatu sistem. Jogianto, HM, (1989), [2], sistem adalah:

"Kumpulan/group dari sub sistem/bagian/komponen apapun baik fisik atau non fisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai suatu tujuan tertentu ".

Secara umum sistem memiliki pengertian sebagai unsur-unsur yang saling berhubungan dan berinteraksi dalam mencapai tujuan bersama. Sistem yang dimaksudkan adalah sistem yang memiliki *input* dari lingkungan dan menghasilkan *output* pada lingkungan sistem itu sendiri sebagai pemproses yang mengubah *input* menjadi *output.*

2.1.2 Ciri-Ciri Sistem

Dengan definisi diatas kita bisa menggambarkan sistem dengan menentukan bagian-bagiannya dan ciri-ciri sistem dari tujuan yang harus dicapai.

Sistem memiliki ciri-ciri yang secara umum terdiri dari :

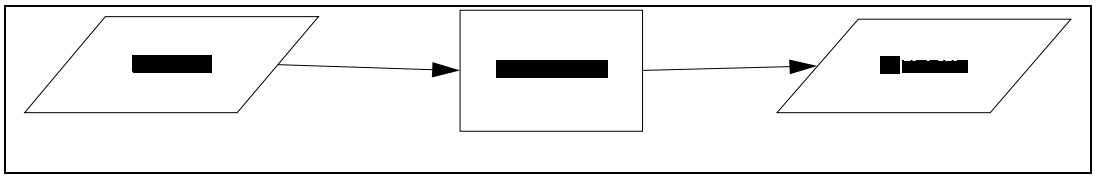
1. Tujuan sistem merupakan target atau sasaran akhir yang dicapai oleh suatu sistem.
2. Batas sistem yang garis abstraksi yang memisahkan antara sistem dan lingkungannya.
3. Sub sistem merupakan komponen atau bagian dari suatu sistem.
4. Hubungan sistem adalah hubungan yang terjadi antara sub sistem dengan sub sistem lainnya yang setingkat.
5. *Input-Proses-Output* merupakan tiga komponen dari fungsi atau sub sistem.
6. Lingkungan sistem merupakan faktor-faktor yang ada di luar sistem yang mempengaruhi sistem. Lingkungan sistem ada dua macam, yaitu lingkungan *eksternal* (di luar sistem) dan lingkungan *internal* (di dalam sistem).

2.2 Konsep Pengolahan Data

Pengolahan data merupakan serangkaian kegiatan yang dilakukan yang menggunakan masukan berupa data dan menghasilkan informasi yang bermanfaat untuk mencapai tujuan sesuai dengan yang telah ditetapkan.

Sistem pengolahan data memelihara dan mengurus *record* operasi perusahaan secara lengkap dan menghasilkan dokumen yang menjelaskan operasi tersebut. Dokumen tersebut digunakan oleh *Manager* dan non *manager* dalam perusahan dan oleh semua elemen lingkungan, kecuali pesaing. Bagi yang menerima dokumen menganggap bahwa dokumen tersebut berisi informasi.

Pengolahan data terdiri dari tiga langkah utama yaitu *input*, proses, dan *output* seperti digambarkan dibawah :



Gambar 2.1 Siklus Pengolahan Data

Penjelasan dari langkah tersebut adalah sebagai berikut :

1. Masukan (*Input*)

   Tahap ini merupakan proses memasukkan data ke dalam proses komputer lewat alat *input*

2. Proses

   Data *input* diubah, biasanya dikombinasikan dengan informasi yang lain untuk menghasilkan data yang telah bermanfaat.

3. Keluaran (*Output*)

   Tahap ini merupakan proses menghasilkan *output* dari hasil pengolahan data ke alat *output*, yaitu berupa informasi.

2.3 Basis Data (*Database*)

Secara sederhana *database* (basis data) dapat diungkapkan sebagai suatu pengorganisasian data dengan bantuan komputer yang memungkinkan data dapat diakses dengan mudah dan cepat. Dalam hal ini, pengertian akses dapat mencakup pemerolehan data maupun pemanipulasian data, seperti menambah dan menghapus data.

Manajemen modern mengikutsertakan informasi sebagai sumber daya penting yang setara dengan sumber daya manusia, uang, mesin, dan material. Informasi adalah suatu bentuk penyajian data yang melalui mekanisme pemrosesan, yang berguna bagi pihak tertentu, misalnya manajer. Bagi pihak manajemen, informasi merupakan bahan untuk pengambilan keputusan.

Keuntungan yang didapatkan dengan menggunakan *database*, diantaranya:

1. Mereduksi redudansi yang akibatnya mengurangi inkonsistensi.
2. Data dapat di*share* antar aplikasi.
3. Standarisasi data dapat dilakukan.
4. Batasan *security* dapat diterapkan.
5. Mengelola integritas (keterjaminan akurasi) data.
6. Menyeimbangkan kebutuhan yang saling konflik.

2.3.1 DBMS (*Database Management System*)

DBMS (*Database Management System*) merupakan perangkat lunak atau program komputer yang dirancang secara khusus untuk memudahkan pengelolaan *database*. Salah satu macam DBMS yang populer dewasa ini berupa RDBMS (*Relational Database Management System*), yang menggunakan model basis data relasional atau dalam bentuk tabel-tabel yang saling berhubungan.

DBMS memiliki 4 keunggulan, yaitu:

1. Kepraktisan
2. Kecepatan
3. Mengurangi kejemuan
4. Kekinian

Komponen utama DBMS, yaitu:

1. Data yang disimpan dalam basis data.
2. *Hardware* : *storage*, *processor*, *memory*.
3. *Software* : DBMS, *Report-writer*, *design arts*, dan lain-lain.
4. Pengguna, yang terdiri dari: pengguna awam (*naive user*), pengguna biasa (*casual user*), *programmer* dan *administrator*.

2.3.2 *Entity Relationship Diagram* (ERD)

ERD (*Entity Relationship Diagram*) merupakan sebuah diagram yang menggambarkan model relasi rancangan data tersimpan (*file*). ERD juga diperlukan untuk menggambarkan struktur data dan relasi antar data.

ERD memiliki komponen-komponen utama, yaitu:

1. Entitas (*Entity*)

   Entitas adalah sebuah barang atau objek yang dapat dibedakan dari objek lain. Entitas direpresentasikan dengan empat persegi panjang.

2. Atribut (*Attribute*)

   Atribut adalah properti deskriptif yang dimiliki oleh setiap anggota dari himpunan entitas. Atribut direpresentasikan dengan bentuk *elips*.

3. Relasi (*Relationship*)

   Relasi adalah hubungan antara suatu himpunan entitas dengan himpunan entitas yang lainnya. Relasi direpresentasikan dengan jajaran genjang. Relasi antar tabel dapat dikategorikan menjadi tiga macam, yaitu :

   1. *One to One Relationship*

      Hubungan *one to one relationship* adalah antara *file* pertama dengan *file* kedua adalah satu banding satu.

   2. *One to Many Relationship*

      Hubungan *one to many relationship* adalah sebuah relasi merupakan tahap dimana hubungan antara *file* pertama dengan *file* kedua adalah satu banding banyak.

   3. *Many to Many Relationship*

      Hubungan *many to many relationship* adalah sebuah relasi antara suatu *file* satu dengan *file* yang kedua memepunyai relasi banyak berbanding banyak.

4. Kardinalitas

   Kardinalitas adalah angka yang menunjukkan banyaknya kemunculan suatu objek terkait dengan kemunculan objek lainnya pada suatu relasi.

2.3.3 Normalisasi

Normalisasi merupakan suatu proses untuk mengubah suatu tabel yang memiliki masalah tertentu ke dalam dua buah tabel atau lebih yang tak lagi memiliki masalah tersebut. Normalisasi bisa digunakan oleh para perancang *database* untuk melakukan verifikasi terhadap tabel-tabel yang telah dibuat sehingga tidak menimbulkan masalah saat data diperbaharui maupun saat data dihapus.

Aturan-aturan normalisasi dinyatakan dalam istilah bentuk normal. Bentuk normal adalah suatu aturan yang dikenakan pada tabel-tabel dalam *database* dan harus dipenuhi oleh tabel-tabel tersebut pada level-level normalisasi. Suatu tabel dikatakan berada dalam bentuk normal tertentu jika memenuhi kondisi-kondisi tertentu.

Bentuk normal yang digunakan pada normalisasi, yaitu:

1. Bentuk normal pertama (1NF)

   Suatu tabel dikatakan dalam bentuk normal pertama hanya jika setiap kolom bernilai tunggal untuk setiap baris.

2. Bentuk Normal Kedua (2NF)

   Suatu tabel berada dalam bentuk normal kedua jika:

   a. Tabel berada dalam bentuk normal pertama.

b. Semua kolom bukan kunci primer tergantung sepenuhnya terhadap kunci primer.

3. Bentuk Normal Ketiga (3NF)

   Suatu tabel berada dalam bentuk normal ketiga jika:

   a. Tabel berada dalam bentuk normal kedua.
   b. Setiap kolom bukan kunci primer tidak memiliki ketergantungan secara transitif terhadap kunci primer.

4. *Boyce-Codd Normal Form* (BCNF)

   Suatu tabel berada dalam bentuk BCNF jika dan hanya jika setiap atribut kunci pada suatu relasi adalah kunci kandidat (atribut yang bersifat unik).

5. Bentuk Normal Keempat (4NF)

   Suatu tabel berada dalam bentuk normal keempat jika suatu relasi ada dalam bentuk BCNF dan tidak mengandung *multivalue dependencies* (kebergantungan banyak nilai).

6. Bentuk Normal Kelima (5NF)

   Suatu relasi berada dalam 5NF jika dan hanya jika setiap dependensi gabungan dalam R tersirat oleh kunci kandidat relasi R. Secara praktis dapat dikatakan bahwa suatu relasi R berada dalam 5NF jika data yang ada padanya tak dapat lagi didekomposisi menjadi relasi-relasi yang lebih kecil dengan kunci kandidat relasi-relasi yang lebih kecil ini tidak sama dengan kunci kandidat relasi.

2.4 Alat Pengembangan Sistem

Dalam pengembangan sistem diperlukan suatu alat dan teknik untuk proses pelaksanaannya. Alat-alat pengembangan sistem akan dibahas secara mendalam dalam subbab-subbab di bawah ini.

2.4.1 Diagram Konteks

Diagram Konteks adalah diagram arus data yang berfungsi untuk menggambarkan suatu sistem informasi baik secara global atau menyeluruh mengenai keterkaitan aliran-aliran data antara sistem dengan kesatuan luar. Diagram Konteks menyorot sejumlah karakteristik sistem, yaitu:

1. Kelompok pemakai yaitu organisasi atau sistem lain dimana sistem melakukan komunikasi (sebagai terminator).
2. Data masuk yaitu data yang diterima sistem dari lingkungan dan harus diproses dengan cara tertentu.
3. Data keluar yaitu data yang dihasilkan sistem dan diberikan ke dunia luar.
4. Penyimpanan data yaitu digunakan secara bersama-sama antara sistem dengan terminator.
5. Batasan antara sistem dan lingkungan.

2.4.2 DFD (*Data Flow Diagram*)

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 disimpan.

DFD merupakan alat yang digunakan pada metodologi pengembangan sistem terstruktur (*structure analysis and design*). DFD merupakan dokumentasi dari sistem yang baik.

Sintak-sintak yang berlaku untuk pembuatan simbol-simbol di DFD, yaitu:

1. Entity luar

   Entity luar direpresentasikan dengan empat persegi panjang. *Entity* luar merupakan bagian luar dari sistem dimana aliran data yang dihubungkan dengan *entity* luar menunjukkan hubungan antara sistem dengan dunia luar.

2. Proses

   Proses menunjukkan transformasi dari masukan menjadi keluaran. Proses umumnya didefinisikan dengan kalimat sederhana. Nama lingkaran tersebut mendeskripsikan respon yang harus dilakukan sistem dalam menganalisa keadaan.

3. Penyimpanan Data (*Data Store*)

   Data store dipakai untuk memodelkan kumpulan data. Penyimpanannya direpresentasikan dengan garis lurus paralel.

4. Aliran Data

   Aliran data menggambarkan dari paket data atau informasi dari satu bagian ke bagian lain dari satu sistem. Kemudian ujung panah menunjukkan arah yang bergerak. Aliran data yang digambarkan sebagai panah dengan dua ujung menggambarkan terjadinya dialog.

2.4.3 Kamus Data

Salah satu bagian yang terintegrasi pada DBMS bertipe relasional adalah kamus data (*data dictionary*), yang menyimpan metadata, atau informasi-informasi mengenai basis data, termasuk di dalamnya nama-nama dan definisi-definisi untuk setiap tabel pada basis data.

Kamus data biasanya merupakan bagian dari katalog sistem (*system catalog*) yang dibentuk untuk setiap basis data. Katalog sistem mendeskripsikan semua objek basis data, termasuk data tentang tabel-tabel seperti nama tabel, pembuat tabel, pengguna yang berhak menggunakannya, nama dan tipe data, kunci tamu (*foreign key*) dan kunci primer (*primary key*), berkas indeks, pengguna yang berhak, dan sebagainya. Katalog sistem diciptakan oleh DBMS dan informasinya disimpan di tabel sistem, yang padanya dapat dilakukan *query* seperti tabel data biasa, jika pengguna memiliki hak yang memadai.

2.4.4 Diagram Alir Dokumen (*Flow Map*)

Diagram aliran dokumen adalah diagram yang menggambarkan aliran informasi dan menggambarkan perubahan bentuk mulai dari *input* data sampai dengan *output*.

Diagram aliran dokumen dimaksudkan untuk mengetahui aliran dokumen yang digunakan dalam sistem pengelolaan data yang sedang berjalan di suatu perusahaan.

Kegunaan dari diagram aliran dokumen adalah:

a. Menggambarkan aktivitas yang berjalan.

b. Menjabarkan aliran dokumen yang terlibat.
c. Memperjelas hubungan-hubungan data dan informasi antara bagian dalam aktivitas tersebut.

2.5 Pengertian Data dan Pengolahan Data

Data dan inforamasi memiliki keterkaitan antara satu dengan yang lainnya. Berikut ini akan diuraikan tentang data dan informasi dalam buku JOG [4] :

"Data adalah fakta-fakta atau pesan-pesan yang belum dievaluasi atau bahan mentah dari informasi yang melalui pengolahan tertentu dibentuk menjadi suatu informasi".

Data dapat meliputi kata-kata, bagan, grafik ataupun simbol yang menggambarkan suatu ide, objek, kondisi atau situasi tertentu.

Adapun pengertian pengolahan data itu sendiri merupakan masa atau waktu yang digunakan untuk mendiskripsikan perubahan bentuk data menjadi pengolahan data, dibutuhkan sistem pengolahan data yang bersifat otomatis yang menyangkut penambahan data, penghapusan data, pengeditan data, pengurutan data, pencarian terhadap data tertentu, perhitungan-perhitungan yang dilakukan terhadap data, penyimpanan terhadap data yang telah akurat dan lainnya sesuai dengan yang diinginkan pemakai. Pengolahan data dan informasi saling berhubungan satu sama lainnya yang terdiri dari perangkat keras (*hardware*), perangkat lunak (*software*), pemakai (*brainware*). Data itu sendiri dan lainnya berhubungan dengan suatu pengolahan data dan informasi.

Operasi-operasi yang dilakukan dalam pengolahan data itu sendiri antara lain :

1. Input Data
   a. *Recording*

      Transaksi data kesebuah pengolahan data medium.

   b. *Coding*

      Transaksi data kedalam bentuk lain.

   c. *Storing*

      Data atau informasi untuk membantu dalam pengambilan keputusan.

2. Transformasi Data
   a. *Calcuting*

      Operasi aritmatika terhadap data field.

   b. *Summarizing*

      Proses akumulasi beberapa data.

   c. *Classifying*

      mengelompokan data kedalam group-group tertentu.

      1. *Catugorizing* data kedalam group berdasarkan karakteristik tertentu.
      2. *Sorting* data kedalam bentuk berurutan.
      3. *Merging* untuk dua atau lebih set data berdasarkan kriteria tertentu.
      4. *Matcing* data berdasarkan keinginan pemakai terhadap group data.

3. Informasi *Output*
    a. *Displaying Result*, menampilkan informasi yang dibutuhkan pemakai melalui monitor atau cetakan.
    b. *Reproducing*, penyimpanan data yang digunakan pemakai lain yang membutuhkan.
    c. *Telecommunicating*, penyimpanan data secara elektronik melalui seluruh komunikasi.

2.6 Perangkat Lunak yang Digunakan

Program ini penulis buat dengan menggunakan *software* pemrograman Borland Delphi 7.0. Sedangkan untuk *database*-nya, penulis menggunakan MySQL.

2.6.1 Borland Delphi 7.0

Delphi merupakan bahasa pemrograman yang mempunyai cakupan kemampuan yang luas dan sangat canggih. Berbagai jenis aplikasi dapat dibuat dengan menggunakan Delphi, termasuk aplikasi untuk mengolah teks, grafik, angka, *database*, web, dan *interface* dengan *hardware*.

Secara umum, kemampuan Delphi adalah menyediakan komponen-komponen dan bahasa pemrograman yang andal, sehingga memungkinkan untuk membuat program aplikasi sesuai dengan keinginan, tampilan dan kemampuan yang canggih.

Delphi menyediakan fasilitas yang luas mulai dari fungsi untuk membuat form hingga fungsi untuk menggunakan beberapa format *file* basis data yang populer (*dBase, Paradoks,* dll).

Komponen-komponen Delphi yang dipakai dalam membuat program aplikasi antara lain:

1. *Project*

   Project adalah sekumpulan form, unit, dan beberapa hal lain dalam program aplikasi. Singkatnya *project* adalah aplikasi itu sendiri.

2. *Form*

   Form adalah suatu *object* yang dipakai sebagai tempat bekerja program aplikasi. Form berbentuk jendela dan dapat dibayangkan sebagai kertas atau meja kerja yang dapat digambari atau diletakkan *object-object* lain diatasnya.

3. *Unit*

   Unit adalah modul kode program. Satu *project* bila mempunyai satu unit atau lebih. Dalam Delphi, ada unit yang tak terpisah dari form yang disebut unit form. Setiap kali dibuat satu form, maka otomatis pula dibuat satu unit.

Manfaat penggunaan unit dalam program aplikasi antara lain seperti berikut ini:

a. Membagi program aplikasi yang besar dalam beberapa unit sehingga dapat mengedit unit-unit tertentu saja. Hal ini sangat memudahkan dalam perancangan, pengubahan, dan penyelesaian program aplikasi.
b. Membuat *library* atau daftar pustaka berupa *function* dan *procedure* sehingga memudahkan *sharing* antar program.

c. Unit dapat kompilasi terpisah dari program aplikasi, sehingga program aplikasi lain yang memerlukan unit, serta dapat menggunakannya tanpa harus menulis kembali *coding* programnya.

4. *Program*

   Program dibangun dari satu unit atau lebih. Pada Delphi, program akan otomatis dibuat dan di *update* oleh Delphi pada saat menambahkannya, mengedit, atau menghapus form dan unit.

5. *Properti*

   Properti digunakan untuk mendefinisikan atribut atau *setting* suatu objek. *Setting* properti akan menentukan cara kerja dari objek yang bersangkutan saat *project* dijalankan misalnya menentukan warna objek, ukuran objek, font yang dipakai, *database* yang diambil, dll.

6. *Event*

   Event adalah peristiwa atau kejadian yang diterima oleh suatu objek misalnya klik, drag, tunjuk, dll.

7. *Method*

   Method adalah prosedur atau perintah yang melekat pada suatu objek.

2.6.2 MySQL

MySQL merupakan salah satu contoh produk RDBMS (*Relational Database Management System*) yang sangat populer di lingkungan Linux, tetapi juga tersedia pada Windows. Banyak situs Web yang menggunakan *MySQL*

sebagai *database server* (server yang melayani permintaan akses terhadap *database*).

MySQL sebagai *database* server juga dapat diakses melalui program yang dibuat dengan menggunakan Borland Delphi. Dengan cara seperti ini *database* dapat diakses secara langsung melalui program *executable* yang kita buat sendiri.

Database MySQL merupakan *database* yang menjanjikan sebagai alternatif pilihan *database* yang dapat digunakan untuk sistem *database* personal atau organisasi.

Elemen dasar SQL, diantaranya yaitu:

1. Pernyataan SQL

   Pernyataan adalah perintah SQL yang meminta sesuatu tindakan kepada DBMS. Pernyataan dasar SQL meliputi : *INSERT*, *DELETE*, *CREATE*, *UPDATE*, dan lain-lain.

2. Nama

   Nama digunakan sebagai identitas bagi objek-objek pada DBMS. Contoh objek pada DBMS adalah tabel, kolom, dan pengguna.

3. Tipe Data

   Setiap data mempunyai tipe data. Tipe data standar adalah *CHAR*, *INTEGER, NUMERIC.*

4. Konstanta

   Konstanta menyatakan nilai yang tetap. Contoh : konstanta numerik dan konstanta string.

5. Ekspresi

   Ekspresi yaitu segala sesuatu yang menghasilkan nilai. Ekspresi digunakan untuk menghitung nilai. Contoh : (LABA/MODAL)*1000

6. Fungsi Bawaan

   Fungsi adalah sebuah sub program yang menghasilkan suatu nilai jika dipanggil. SQL mempunyai sejumlah fungsi bawaan, yaitu: *MIN*, *MAX*, *AVG*, dan lain-lain.