# BAB II

# LANDASAN TEORI

## 2.1 Konsep Dasar

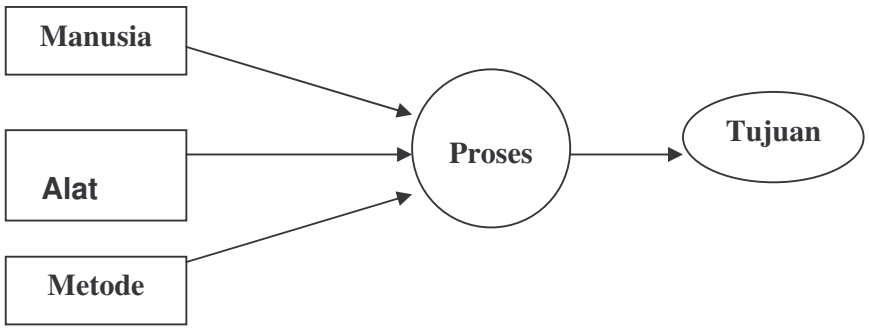
### 2.1.1 Sistem

Pengertian sistem secara pendekatan manual menurut **Cole/Neuchel** dibagi dalam dua bagian yang saling menunjang yaitu : "sistem" dan "prosedur" sebagai berikut :

> *"Sistem adalah suatu jaringan pekerjaan yang berhubungan dengan prosedur-prosedur yang erat hubungannya satu sama lain yang dikembangkan menjadi suatu skema untuk melaksanakan sebagian besar aktivitas perusahaan."*
>
> (H. La Midjan dan Azhar Susanto,1996:4 )

> *"Prosedur adalah suatu urut-urutan pekerjaan tata usaha (clerical operations) yang biasanya melibatkan beberapa petugas di dalam suatu bagian atau lebih yang diadakan untuk menjamin pelaksanaan yang seragam dari transaksi-transaksi yang berulang-ulang dalam perusahaan."*
>
> (H. La Midjan dan Azhar Susanto,1996:4 )

Berdasarkan uraian di atas pengertian sistem menurut pendekatan secara manual terdiri dari prosedur-prosedur dan apabila dikaji lebih jauh lagi maka suatu prosedur berintikan kegiatan tata usaha yang terdiri dari menulis, membuat kode, menghitung dan lain-lain dengan berlandaskan formulir-formulir dan menggunakan metode " pen " dan " ink " atau mesin. Sedangkan sistem menurut pendekatan sistem secara EDP (*Electronic Data Processing*) sebagaimana gambar 2.1 di bawah ini :




**Gambar 2.1 Sistem menurut pendekatan sistem EDP**

Perbedaan utama kedua pendekatan sistem tersebut terletak pada:

1) Pendekatan sistem secara manual dalam mengolah data sebagian besar lebih berintikan perangkat manusia dan mungkin dibantu oleh mesin, sedangkan pendekatan secara EDP sebagian besar telah dilaksanakan oleh mesin elektronik atau komputer.
2) Dalam mengolah data menciptakan informasi pendekatan sistem secara manual lebih berintikan prosedur secara visual dan dibantu oleh kertas

kerja berupa dokumen-dokumen. Sedangkan pendekatan secara EDP keseluruhan prosedur telah dirangkum dalam program-program dan tidak secara visual dan sebagian besar tanpa dokumen.

3) Dalam pemecahan masalah pendekatan sistem secara EDP lebih memiliki kemampuan daripada secara manual mengandung banyak hambatan utama adalah faktor manusia.

Adapun pengertian sistem menurut **Robert G. Murdick** adalah sebagai berikut :

*"Sistem dapat didefinisikan, adalah suatu kumpulan dari komponen-komponen yang bekerja sama untuk mencapai suatu tujuan."*

(H. La Midjan dan Azhar Susanto,1996:1 )

Pengertian sistem dalam kontek masalah data menghasilkan informasi sebagaimana yang dikemukakan oleh *The American National Standards Committee* adalah sebagai berikut :

*"Sistem, dalam pengolahan data, suatu kumpulan dari manusia, mesin dan metode yang terorganisir untuk memenuhi seperangkat fungsi."*

(H. La Midjan dan Azhar Susanto,1996:2 )

Suatu sistem mempunyai karakteristik atau sifat-sifat yang tertentu, yaitu :

a. **Komponen sistem (*components*)**

   Komponen-komponen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem yang saling berinteraksi artinya saling bekerja sama membentuk satu kesatuan.

b. **Batas sistem (*boundary*)**

   Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya.

c. **Lingkungan luar sistem (*environments*)**

   Lingkungan luar dari suatu sistem adalah apapun di luar batas dari sistem yang mempengaruhi operasi sistem.

d. **Penghubung sistem (*interface*)**

   Penghubung merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya.

e. **Masukan sistem (*input*)**

   Masukan adalah energi yang dimasukkan ke dalam sistem, dapat berupa masukan perawatan ( *maintenance input*) dan masukan sinyal (*signal input*).

f. **Keluaran sistem (*output*)**

   Keluaran adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

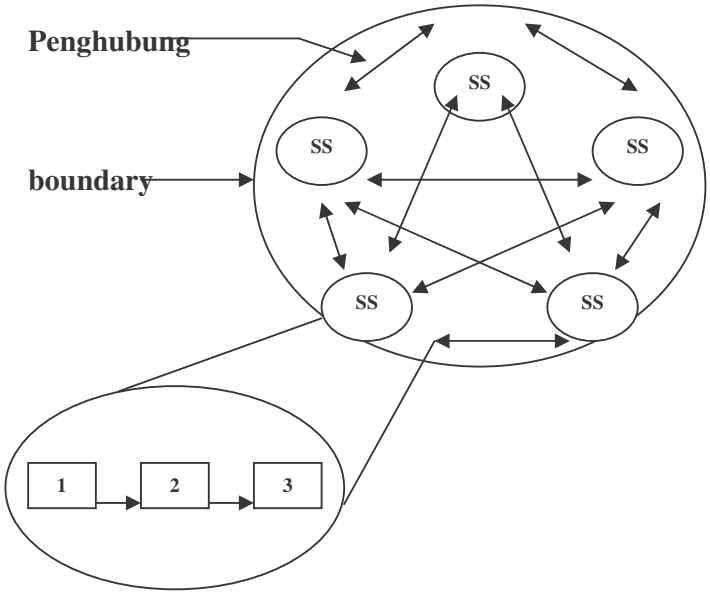
g. **Pengolah sistem (*process*)**

   Pengolah sistem berfungsi untuk mengubah masukan menjadi keluaran.

**h. Sasaran sistem/tujuan (*objectives/goal*)**

Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem.

Karakteristik sistem (penghubung, *boundary*, subsistem, *input*, proses, *output* ) seperti yang telah dijelaskan di atas dapat dilukiskan dengan gambar berikut ini :




**Gambar 2.2 Karakteristik suatu sistem**

Keterangan :

SS = Subsistem

1 = input

2 = proses

3 = output

Sistem diklasifikasikan dari beberapa sudut pandangan, di antaranya :

**a. Sistem abstrak dan sistem fisik (*abstract system and physical system*)**

*Sistem abstrak* adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Misalnya sistem teologia, sistem yang berupa pemikiran-pemikiran hubungan manusia dengan Tuhan.

*Sistem fisik* merupakan sistem yang ada secara fisik, misal sistem komputer.

**b. Sistem alamiah dan sistem buatan manusia (natural system and human made system)**

*Sistem alamiah* adalah sistem yang terjadi melalui proses alam, misalnya sistem perputaran bumi. *Sistem buatan manusia* adalah sistem yang dirancang oleh manusia.

**c. Sistem tertentu dan sistem tak tentu (deterministic system and probabilistic system)**

*Sistem tertentu* beroperasi dengan tingkah laku yang sudah dapat diprediksi. *Sistem tak tentu* adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

**d. Sistem tertutup dan sistem terbuka (closed system and open system)**

*Sistem tertutup* merupakan sistem yang tidak berhubungan dengan lingkungan luarnya. *Sistem terbuka* adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya.

### 2.1.2 Data

Istilah **data** adalah suatu istilah majemuk dari kata **datum**, yang berarti fakta atau bagian dari fakta yang mengandung arti yang dihubungkan dengan kenyataan, simbol-simbol, gambar-gambar, kata-kata, angka-angka, huruf-huruf atau simbol-simbol yang menunjukkan suatu ide, objek, kondisi atau situasi dan lain-lain. Jelasnya data itu dapat berupa apa saja dan dapat ditemui di mana saja. Kegunaan daripada data ialah sebagai bahan dasar yang objektif (relatif) di dalam proses penyusunan kebijaksanaan dan keputusan oleh pimpinan organisasi.

Pengertian data menurut **Barry E. Cushing** adalah sebagai berikut :

> *"Data dapat dianggap bahwa terdiri dari sekumpulan karakter yang diterima sebagai masukan (input) untuk suatu sistem informasi dan disimpan serta diolah."* (H. La Midjan dan Azhar Susanto,1996:7 )

Apabila kata data dihubungkan dengan pengolahan maka pengertian pengolahan data adalah segala macam pengolahan terhadap data atau kombinasi dari bermacam-macam pengolahan terhadap data untuk membuat data itu berguna sesuai dengan hasil yang diinginkan, sehingga dapat segera dipakai. Dalam pengolahan data digunakan alat-alat yang dapat mempercepat jalannya pengolahan ini, tentunya memerlukan suatu prosedur pengolahan, beserta staf yang mampu melaksanakan seluruh faset dalam rangka pengolahan data.

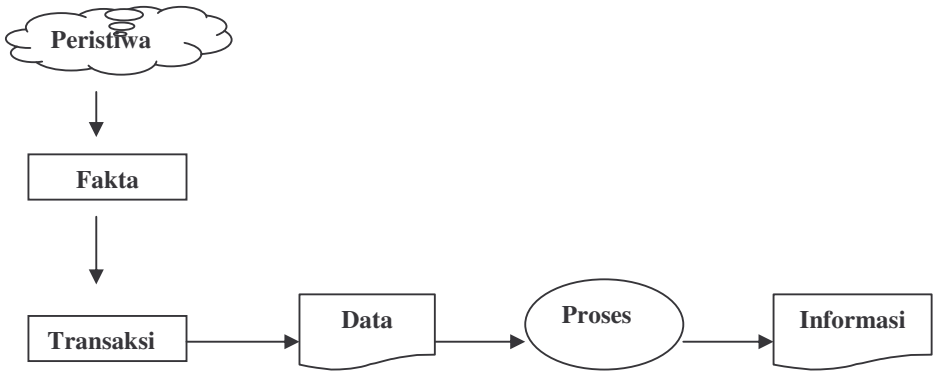
### 2.1.3 Informasi

Informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang

menggambarkan suatu kejadian-kejadian (*event*) yang nyata (*fact*) yang digunakan untuk pengambilan keputusan.

Pengertian informasi menurut **Barry E. Cushing** adalah sebagai berikut :

> *"Informasi diartikan sebagai keluaran (output) suatu pengolahan data yang telah diorganisir dan berguna bagi orang yang menerima."*
>
> (H. La Midjan dan Azhar Susanto, 1996:7 )




**Gambar 2.3 Proses data menjadi informasi**

**Kualitas informasi** tergantung dari tiga hal, yaitu :

1) **Akurat,** berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan, harus jelas mencerminkan maksudnya.
2) **Tepat pada waktunya,** berarti informasi yang datang pada penerima tidak boleh terlambat.
3) **Relevan,** berarti informasi tersebut mempunyai manfaat untuk pemakaiannya.

Informasi dikatakan bernilai bila manfaatnnya lebih efektif dibandingkan dengan biaya mendapatkannya. Kegunaan Informasi adalah untuk mengurangi hal ketidakpastian di dalam proses pengambilan keputusan tentang sesuatu keadaan.

### 2.1.4 Sistem Informasi

Menurut **John F. Nash/Martin B. Robert**, memberikan pengertian Sistem Informasi sebagai berikut :

> *"Sistem Informasi merupakan kombinasi dari manusia, fasilitas atau alat teknologi, media, prosedur dan pengendalian yang bermaksud menata jaringan komunikasi yang penting, proses atau transaksi-transaksi tertentu dan rutin, membantu manajemen dan pemakai intern dan ekstern dan menyediakan dasar pengambilan keputusan yang tepat (intelligent)".* (H. La Midjan dan Azhar Susanto,1996:8 )

Informasi dihasilkan oleh suatu proses sistem informasi dan bertujuan menyediakan informasi untuk membantu pengambilan keputusan manajemen, operasi perusahaan dari hari ke hari dan informasi yang layak untuk pihak luar perusahaan.

Sumber dari sistem informasi di dalam suatu perusahaan, bersumber dari alat pemrosesan data adalah sebagai berikut :

1. *Manual Information System*, bersumber dari proses manual.
2. *Automated Information System*, bersumber dari proses peralatan.

3. *Computer-based Information Systems*, bersumber dari proses Electronic Data Processing (EDP).

**Komponen-komponen sistem informasi, di antaranya :**

1. **Blok masukan (*input block*)**

   Input mewakili data yang masuk ke dalam sistem informasi. Input di sini termasuk metode-metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

2. **Blok model (*model block*)**

   Blok ini terdiri dari kombinasi prosedur, logika dan model matematika yang akan memanipulasi data input dan data yang tersimpan di dasar data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.

3. **Blok keluaran (*output block*)**

   Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. **Blok teknologi (*technology block*)**

   Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem keseluruhan.

5. **Blok dasar data (*database block*)**

   Dasar data merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.

**6. Blok kendali (*controls block*)**

Supaya sistem informasi dapat berjalan sesuai dengan yang diinginkan, maka perlu diterapkan pengendalian-pengendalian di dalamnya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi.

## 2.2 Analisis Sistem (Systems Analysis)

Suatu sistem pengolahan data dapat didefinisikan secara garis besar sebagai sistem yang **menerima, menghubungkan, menyimpan, mengolah** dan **menyediakan data** yang dibutuhkan untuk operasi/ketatalaksanaan suatu organisasi.

Sistem pengolahan data terdiri dari semua personil, perlengkapan, peralatan dan prosedur yang diperlukan untuk melaksanakan fungsi organisasi. Penentuan daripada tingkat di mana sistem yang berjalan dapat menjawab kebutuhan organisasi dengan informasi yang relevan dan tepat waktu serta sistem dapat berjalan secara efisien adalah keseluruhan objektif analisa sistem.

Analisis sistem yaitu penganalisaan masalah, kebutuhan, sumber data, perencanaan teknik pengolahan data yang akan digunakan, analisa peralatan dan penentuan informasi-informasi yang dikehendaki.

Untuk mengadakan evaluasi yang luas dan logis terhadap sistem pengolahan data yang berjalan, maka analisis sistem hendaknya dilakukan dalam beberapa tahapan, yaitu :

1) **Menentukan objektif/tujuan sistem**

Analisis sistem harus me-*review* kebijaksanaan dan prosedur organisasi yang berjalan untuk menentukan pengaruhnya terhadap objektif pengolahan data, menanyakan kepada pimpinan serta *user* sehubungan dengan kebutuhan informasi mereka, mewawancara pejabat-pejabat garis horizontal dan menentukan batasan-batasan di dalam memenuhi kebutuhan informasi.

2) **Mempelajari organisasi**

Dalam hubungannya dengan penentuan objektif daripada sistem, analis sistem harus mengumpulkan dan mempelajari mengenai organisasi, seperti struktur organisasi serta tugas-tugas dan wewenangnya, mempelajari aliran data antara setiap unit dalam organisasi dan mempelajari situasi organisasi informilnya.

3) **Menganalisa output yang sudah ada dan menentukan kegunaannya dalam sistem.**

Analis sistem harus mengumpulkan dan menilai laporan-laporan, formulir-formulir, catatan-catatan yang sedang digunakan dan perlu mewawancara sumber dan penerima daripada setiap jenis dokumen.

4) **Menyelidiki sistem dan prosedur yang berjalan**

Selama proses penyelidikan sistem yang berjalan, analis sistem harus mengumpulkan semua data kuantitatif yang berkenaan dengan sistem, yaitu masalah waktu, biaya, kebutuhan tenaga manusia, perlengkapan dan bila perlu kebutuhan waktu mesin.

5) **Mengenai kebutuhan input**

Analis sistem harus menentukan bagi sistem yang berjalan mengenai asal/sumber daripada data/*input*, media dan format dokumen, *volume* dan frekuensi input.

6) **Evaluasi efektifitas sistem yang berjalan**

Setelah analis sistem mempersiapkan prosedur yang tidak kaku serta mengusulkan penyempurnaan sistem maka laporannya diserahkan kepada pimpinan untuk penilaian. Adapun alat-alat yang biasa digunakan dalam menganalisa system di antaranya seperti yang kita lihat dalam uraian berikut :

### 2.2.1 Diagram Aliran Data

Selama tahap analisa dan proses desain, analis sistem mengumpulkan data yang belum terstruktur atau data mentah yang diperoleh dari hasil wawancara, penulisan memo, manual dokumen, catatan-catatan dan lain-lain. Data ini kemudian harus diikhtisarkan untuk membantu berbagai tujuan selain memudahkan berkomunikasi dengan *user*, membantu mengembangkan sistem yang akan datang.

Diagram aliran data merupakan suatu model logika dalam sistem dan tidak tergantung oleh perangkat keras, perangkat lunak, struktur data atau arsip organisasi dan merupakan gambar grafik dari logika sistem. Dengan melihat diagram aliran data akan mempermudah *user* non-teknik untuk mengerti jalannya

aktivitas dan menjadi alat komunikasi yang baik. Dapat disimpulkan diagram aliran data merupakan titik awal bagi analis sistem dalam mendesain sistem.

### 2.2.2 Kamus Data

Kamus data sangat penting dalam mendefinisikan dan menjelaskan secara semestinya elemen-elemen data di dalam suatu system sehingga semua pihak (pemakai maupun analis) dapat mengerti semua *input, output* dan bagian simpanan. Pusat penyimpanan informasi data-data dari *database* yang memuat "skema *database*", yang mana nama dari setiap *item* dalam *database* serta deskripsi dan definisi atribut-atributnya, yang merujuk pada "data standar".

Kamus data berisi informasi lokasi *database* pada *file database*, aturan mengakses data, pengamanan data serta kondisi informasi data. DBMS (*Data Base Management System*) menerima permintaan dari suatu program dan mengakses kamus *database* untuk melihat apakah program memiliki data yang dicari dan memastikan lokasi data pada *database*. Berikut ini notasi yang akan dipakai dalam penulisan Kamus Data :

**Tabel 2.1 Notasi Kamus Data**

| Notasi | Arti |
|---|---|
| =<br>+<br>( ) | Terdiri dari atau sama dengan<br>Dan<br>Optional (elemen data dalam tanda kurung bisa ada bisa tidak) |

| | |
|---|---|
| { } | Iteras |
| [ ] | Pilih salah satu dari beberapa pilihan |
| * | Komentar atau keterangan |
| @ | Identitas atau field kunci |
| \| | Sama dengan tanda [] |

### 2.2.3 Model Data

Pemodelan data adalah suatu teknik untuk menganalisa dan menjelaskan data yang dibutuhkan oleh pemakai sistem. Dalam menganalisa data, difokuskan pada pengertian terhadap konsep struktur data. Penjelasan tentang data harus dibuat secara singkat, akurat dan dapat dibaca oleh pemakai, *programmer* dan spesialis teknis lainnya, karena model data adalah merupakan cetak biru dari rancangan *database*.

Model *Entitas Relationship Diagram* adalah metode perancangan *database* yang berbasis pada dunia nyata yang terdiri dari sekumpulan obyek yang logis serta adanya hubungan antar obyek-obyek tersebut .komponen-komponen model data dapat digunakan untuk menjelaskan rancangan basis data yang diimplementasikan pada berbagai jenis perancangan *database*. Komponen-komponen tersebut yaitu:

*1)* Entitas

Entitas merupakan sesuatu hal dalam bentuk apapun yang datanya dapat dikumpulkan dan dapat berupa obyek, orang, konsep, abstrak maupun kejadian yang bersifat unik. Definisi entitas harus menekankan pada pemakai suatu entitas dan karena entitas merupakan suatu abstraksi maka entitas berlabel kata benda.

Aturan untuk membentuk entitas :

1. Nama suatu entitas hanya dapat muncul sekali dalam sebuah model data.
2. Masing-masing entitas harus diberi sebuah identitas unik.
3. Nilai dari suatu atribut tidak boleh disertakan dalam model data.

2) Atribut

Atribut merupakan karakteristik dari suatu entitas. Definisi suatu atribut termasuk informasi tentang :"Arti suatu atribut untuk pemakai dan karakter, teknis suatu atribut bila atribut tersebut diimplementasikan sebagai bagian *database* untuk sistem (panjang karakter suatu atribut dan tipe data, seperti *numeric*, *alphabet*)".

Aturan untuk membuat atribut:

1. Nama suatu atribut hanya dapat muncul sekali dalam sebuah model data.
2. Suatu atribut harus berhubungan atau dikelompokan kepada entitas.
3. Nilai dari suatu atribut tidak boleh disertakan dalam model data.

Ada 4 macam entitas atau atribut kunci, yaitu:

1. ***Primary Key*** **(Kunci Utama)**

   *Primary key* adalah satu atribut yang tidak hanya mengidentifikasikan secara unik suatu kejadian spesifik, tapi juga mewakili setiap kejadian dari suatu entitas.

2. ***Candidate Key (*** **Kunci Kandidat)**

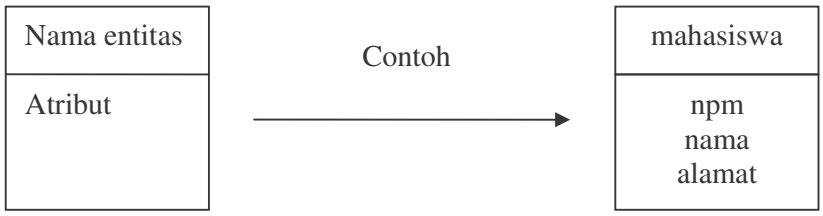
   Kunci kandidat adalah satu atau lebih atribut yang mengidentifikasikan secara unik suatu kejadian spesifik dari suatu entitas atau tabel. Jika satu atribut dinyatakan sebagai kunci kandidat, maka kunci lainnya disebut sebagai kunci komposit (*composite key*).

3. ***Alternate Key*** **(Kunci Alternatif)**

   *Alternate key* adalah kunci kandidat yang tidak dipakai sebagai *primary key*. Kunci alternatif biasanya sering dipakai sebagai kunci untuk pengurutan data.

4. ***Foreign Key*** **(Kunci Tamu)**

   Kunci tamu adalah satu atribut atau lebih yang melengkapi suatu relasi atau hubungan yang menunjukkan ke induknya. Kunci tamu ditempatkan pada entitas anak yang sama dengan entitas kunci utama yang direlasikan pada entitas induk.




**Gambar 2.4 Entitas dan Atribut**

*3) Relationship*

*Relationship* merupakan hubungan/asosiasi antara satu entitas dengan dirinya sendiri atau dengan entitas lainnya. *Relationship* digambarkan sebagai garis lurus yang menghubungkan entitas-entitas yang dipandang memiliki hubungan satu dengan yang lainnya.

Ada dua macam *relationship* yaitu *mandatory relationship* yang digambarkan dengan garis penuh dan *optional relationship* yang digambarkan dengan garis putus-putus. Pada ujung *relationship* itu terdapat derajat hubungan.

- Hubungan satu ke satu (*mandatory*)
- Hubungan banyak ke banyak (*optional*)
- Hubungan satu ke banyak (*mandatory*)

Pada model data *relational*, hubungan antar *file-file* direlasikan dengan kunci relasi (*relation key*), yang merupakan kunci utama masing-masing *file*. Perancangan *database* yang tepat membuat program bekerja secara optional.

Aturan untuk membuat *relationship*:

1. Setiap *relationship* harus diberi label
2. Bila memungkinkan, hindari garis yang saling bersilangan
3. Sertakan derajat dari *relationship* pada setiap ujung garis

4. Setiap garis harus menghubungkan tepat dua entitas.

4) *Identifier*

*Identifier* adalah sebuah atau kumpulan atribut yang mempunyai nilai unik untuk setiap nilai yang terdapat pada atribut atau entitas.

Aturan untuk membuat *identifier*:

1. Suatu nama *identifier* hanya dapat digunakan sekali dalam sebuah model data.
2. Suatu *identifier* harus berasosiasi dengan atau dikelompokkan pada sebuah entitas.
3. Nilai dari suatu *identifier* tidak boleh dimunculkan dalam model data.
4. Sebuah *identifier* memiliki sebuah nilai unik untuk setiap kejadian.

## 2.3 Konsep Basis Data

Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.

Definisi database menurut **George M. Scott** adalah :

> *"Database adalah sistem file komputer yang menggunakan cara pengorganisasian file tertentu, dimaksudkan untuk mempercepat pembaruan masing-masing record, serta pembaruan secara serempak atas record terkait, juga untuk mempermudah dan mempercepat akses terhadap seluruh record lewat program aplikasi, serta akses yang cepat terhadap data yang tersimpan yang harus digunakan secara bersama-sama untuk*

*dibaca guna penyusunan laporan-laporan rutin atau khusus ataupun untuk penyelidikan."*

(George M. Scott, 1997:349)

Unsur-unsur konsep pembangun *database*, adalah:

1. *Field* atau atribut

   *Field* atau atribut adalah entitas yang mewakili satu jenis data. Misalnya *field* nama pada tabel mahasiswa.

2. *Record* atau *tuple*

   *Record* adalah kumpulan elemen yang saling terkait yang menginformasikan tentang suatu entitas secara lengkap.

3. *File*

   *File* yaitu suatu kumpulan *record* sejenis yang mempunyai panjang elemen dan atribut yang sama.

4. Tabel

   Tabel adalah sebuah *file* yang menampung data-data dalam kelompok tertentu.

### 2.3.1 Keuntungan Sistem *Database*

1. *Mengurangi redudansi*

Penyimpanan di beberapa tempat untuk data yang sama disebut *redudansi*. Redudansi ini akan mengakibatkan pemborosan ruang penyimpanan dan biaya untuk mengakses data akan lebih tinggi. Penyimpanan data yang sama dengan

kunci yang sama secara berulang-ulang pada beberapa *file* akan menyebabkan inkonsistensi data. Biasanya, kedua hal tadi terjadi ketika proses entri dan *update* data. Dan kedua hal tadi dapat dihindari dengan penggunaan *database*.

2. *Integrity*

*Database* berisi *file* yang saling berkaitan yaitu dengan adanya *field* kunci pada suatu *file* yang menghubungkan pada *file* lain, dengan tujuan untuk menjamin agar elemen dalam suatu *file* yang menunjuk ke suatu pengenal unik pada *file* lain benar-benar menunjuk pada suatu nilai yang memang ada.

3. *Menghindarkan inkonsistensi*

Karena redundansi berkurang, sehingga umumnya *update* data hanya sekali.

4. *Penggunaan data bersama*

Data yang sama dapat diakses oleh beberapa *user* pada saat bersamaan. Hal ini menggunakan sistem *database*, setiap aplikasi mempunyai *file* tersendiri yang berarti suatu data tunggal dalam basis data dapat digunakan untuk beberapa kegunaan.

5. *Standardisasi*

*Database* dapat melakukan kontrol terpusat untuk memastikan bahwa standard untuk nama data, pengguna data, format data atau dokumentasi diikuti seragam dalam suatu organisasi, sehingga keseragaman data dan informasi dapat disajikan untuk memenuhi kebutuhan *user* atau pengguna.

6. *Jaminan sekuriti*

*Database* dapat mendefinisikan prosedur otoritas untuk memastikan bahwa data hanya dapat diakses oleh *user* yang berhak dan dapat mengatur pembagian akses data sesuai dengan tipe *user* yang mengakses.

7. *Menyeimbangkan kebutuhan*

*Database* dapat ditentukan prioritas suatu operasi, misal antara *Update* dengan *Retrieval*.

### 2.3.2 Kerugian Sistem *Database*

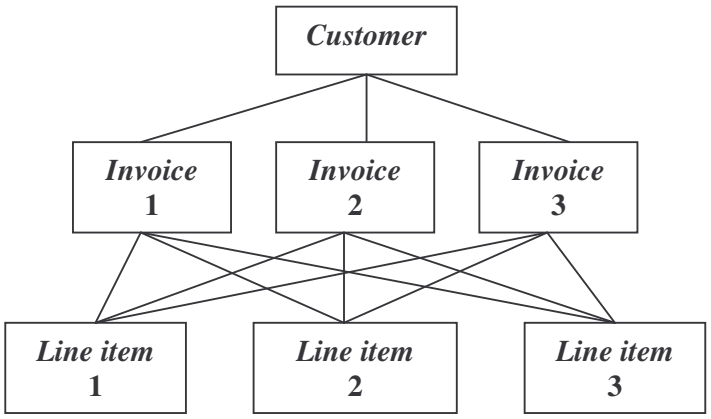
1. *Diperlukan hardware tambahan* : CPU yang lebih kuat, terminal yang lebih banyak, alat untuk komunikasi.
2. *Biaya performance yang lebih besar* :listrik, Personil yang lebih tinggi klasifikasinya, biaya telekomunikasi antar lokasi/kota.
3. *Sistem tampak lebih komplek* : banyaknya aspek yang harus diperhatikan
4. *Rawannya keberhasilan operasi,* karena gangguan listrik, gangguan komunikasi.

### 2.3.3 Model-model *Database*

Dikenal tiga model yang digunakan untuk mengembangkan struktur logis dari data dalam *database*, yang umum digunakan pada waktu sekarang adalah:

**a. Model hierarki**

Unsur data telah disusun dalam bentuk *tree structure* di mana setiap unsur data pada tingkat tertentu memiliki unsur-unsur data yang lebih rendah. Sebagai contoh dalam pengolahan order penjualan, unsur langganan memiliki unsur-unsur faktur dan setiap unsur faktur memiliki unsur penjualan jenis barang (gambar 2.5).



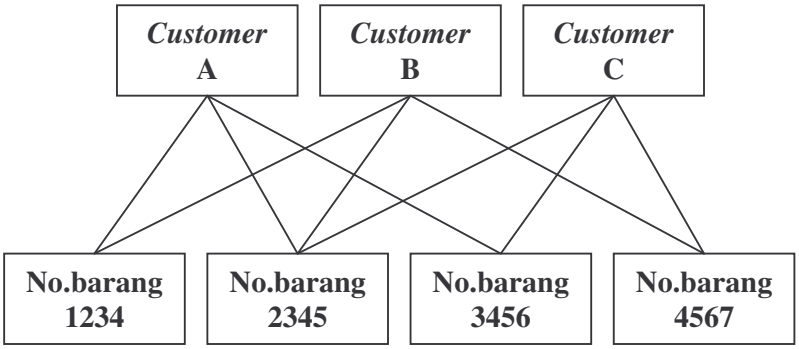

**Gambar 2.5 Model hierarki**

**b. Model *Network***

Model jaringan pada dasarnya tidak berbeda dengan model hierarki, hanya setiap data memiliki hubungan dengan lebih unsur data pada tingkat di atasnya. Keterbatasan pada model ini adalah harus konsisten secara intern, sebagai contoh, unsur data dapat memiliki "anak" dan kemungkinan "cucu" tetapi "cucu" tidak dapat sekaligus memiliki "kakek" yang sama.

Model ini sangat tepat digunakan pada semua bentuk hubungan unsur data termasuk bentuk hubungan banyak dengan banyak unsur data. Pada gambar 2.6

memperlihatkan aplikasi model jaringan pengolahan order penjualan menyangkut hubungan langganan dengan persediaan suku cadang. Dengan model ini dapat disajikan daftar pemesanan suku cadang untuk tiap langganan begitu pula daftar langganan untuk tiap nomor suku cadang yang dipesan.




**Gambar 2.6 Model *Network***

**c. Model Relasional**

Model relasional disebut juga *plat file approach*, pemakai tidak usah menaruh perhatian terhadap kompleksitas hubungan antara unsur data. Malahan semua data sama disusun secara terstruktur dalam suatu format tabel yang sederhana berdasarkan kunci *field* yang diperlukan dan menyatakan sebuah ilusi dari suatu *file* yang jelas kepada pemakai. Data dapat ditampakkan lebih dari satu tabel sehubungan dengan kunci yang telah diseleksi yang memperhatikan bentuk tabulasi dalam model yang berhubungan.

Pada tabel yang pertama kuncinya berdasarkan pemilihan atas dasar langganan, sedangkan tabel kedua kuncinya berdasarkan pemilihan atas dasar nomor suku cadang.

**Tabel 2.2 Model Relasional**

| *Customer* | *Part no.* | *Other Data* |
|---|---|---|
| A | 1234 | ……. |
| | 2345 | ……. |
| | 3456 | ……. |
| B | 1234 | ……. |
| | 2345 | ……. |
| | 4567 | ……. |
| C | 1234 | ……. |
| | 2345 | ……. |
| | 4567 | ……. |

| *Part no.* | *Customer* | *Other data* |
|---|---|---|
| 1234 | A | …….. |
| | B | …….. |
| 2345 | A | …….. |
| | B | …….. |
| | C | ……. |
| 3456 | A | ……. |
| | C | ……. |
| 4567 | B | ……. |
| | C | ……. |

### 2.3.4 Normalisasi *Database*

Proses normalisasi adalah proses pengelompokan data elemen menjadi tabel-tabel yang menunjukan entitas dan relasinya. Normalisasi banyak digunakan untuk mengubah bentuk *database* struktur pohon jaringan menjadi struktur hubungan.

Bentuk-bentuk normalisasi *database* :

**a. Bentuk tidak normal (*Unnormalized Form*)**

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

**b. Bentuk normal kesatu (*First Normal Form*)**

Bentuk normal pertama mempunyai ciri yaitu setiap data dibentuk dalam *flat file* (*file* datar atau rata), data dibentuk dalam satu *record* dan nilai dari *field* berupa "*atomic value*". Tidak ada set atribut yang berulang-ulang atau atribut bernilai ganda. Tiap *field* hanya satu pengertian, bukan merupakan kumpulan kata yang mempunyai arti ganda atau sebuah relasi di mana irisan antara tiap kolom dan baris terdapat satu dan hanya mempunyai satu nilai.

**c. Bentuk normal kedua (*Second Normal Form*)**

Bentuk normal kedua mempunyai syarat yaitu bentuk data telah memenuhi kriteria bentuk normal pertama dan setiap atribut kunci hanya tergantung pada *primary key*. Maka untuk membentuk normal kedua setiap *file* harus mempunyai *field-field* kunci terlebih dahulu dan *field* kunci harus unik dan dapat mewakili atribut-atribut lain menjadi anggotanya.

**d. Bentuk normal ketiga (*Third Normal Form*)**

Untuk bentuk normal ketiga, relasi harus dalam bentuk normal kedua. Semua atribut bukan kunci harus bergantung pada *primary key* secara menyeluruh.

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

Sebuah relasi sudah dikatakan dalam bentuk *Boyce-Codd Normal* jika dan hanya jika setiap determinan adalah kunci kandidat.

## 2.4 Konsep SMS (*Short Message Service*)

SMS merupakan salah satu layanan pesan teks yang dikembangkan dan distandarisasi oleh suatu badan yang bernama ETSI (*European Telecomunication Standards Institute*) sebagai bagian dari pengembangan GSM *Phase* 2, yang terdapat pada dokumentasi GSM 03.40 dan GSM 03.38. Fitur SMS ini memungkinkan perangkat Stasiun Seluler Digital (*Digital Cellular Terminal,* seperti ponsel) untuk dapat mengirim dan menerima pesan-pesan teks dengan panjang sampai dengan 160 karakter melalui jaringan GSM. (ETSI, 1996) SMS dapat dikirimkan ke perangkat Stasiun Seluler Digital lainnya hanya dalam beberapa detik selama berada pada jangkauan pelayanan GSM. Lebih dari sekedar pengiriman pesan biasa, layanan SMS memberikan garansi SMS akan sampai pada tujuan meskipun perangkat yang dituju sedang tidak aktif yang dapat disebabkan karena sedang dalam kondisi mati atau berada di luar jangkauan layanan GSM. Jaringan SMS akan menyimpan sementara pesan yang belum terkirim, dan akan segera mengirimkan ke perangkat yang dituju setelah adanya tanda kehadiran dari perangkat di jaringan tersebut.

Dengan fakta bahwa layanan SMS (melalui jaringan GSM) mendukung jangkauan/jelajah nasional dan internasional dengan waktu keterlambatan yang sangat kecil, memungkinkan layanan SMS cocok untuk dikembangkan sebagai

aplikasi-aplikasi seperti: *pager*, *e-mail*, dan notifikasi *voice mail*, serta layanan pesan banyak pemakai (*multiple users)*. Namun pengembangan aplikasi tersebut masih bergantung pada tingkat layanan yang disediakan oleh operator jaringan. (Gupta, 2003).

### 2.4.1 PDU (*Protocol Data Unit*) SMS

Dalam proses pengiriman atau penerimaan pesan pendek (SMS), data yang dikirim maupun diterima oleh stasiun bergerak menggunakan salah satu dari 2 mode yang ada, yaitu: mode teks, atau mode PDU (*Protocol Data Unit*) (Wavecom, 2000). PDU berisi bilangan-bilangan heksadesimal yang mencerminkan bahasa I/O. PDU terdiri dari atas beberapa *Header*. *Header* unutk kirim SMS ke SMS-*Centre* berbeda dengan SMS yang diterima dari SMS-*Centre*.

Maksud dari bilangan heksadesimal adalah bilangan yang terdiri atas 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F. Sebagai contoh, untuk angka decimal 1000, bilangan heksadesimalnya adalah 3E8.

Cara mengonversikannya :

1000 : 16 = 62 sisa 8 → 8

62 : 16 = 3 sisa 14 → E

3 : 16 = 0 sisa 3 → 3

#### 2.4.1.1 PDU untuk kirim SMS ke SMS-*Centre*

PDU untuk kirim SMS terdiri atas delapan *header*, sebagai berikut:

1. **Nomor SMS-*Centre***

   *Header* pertama ini terbagi atas tiga *subheader*, yaitu:

a. Jumlah Pasangan Heksadesimal SMS-*Centre* dalam bilangan heksa.

*b. National/International Code*

- Untuk *National*, kode *subheader*-nya yaitu 81
- Untuk *International*, kode *subheader*-nya yaitu 91

c. No SMS-*Centre*-nya sendiri, dalam pasangan heksa dibalik-balik. Jika tertinggal satu angka heksa yang tidak memiliki pasangan, angka tersebut akan dipasangkan dengan huruf F di depannya.

Contoh: untuk nomor SMS-*Centre* Exelcom dapat ditulis dengan dua cara sebagai berikut:

***Cara 1:***

0818445009 diubah menjadi:

a. 06 → ada 6 pasang

b. 81 →1 pasang

} Total: 6 pasang

c. 80- 81-44-05-90 → 5 pasang

Digabung menjadi : **06818081440590**

***Cara 2:***

62818445009 diubah menjadi :

a. 07 →ada 7 pasang

b. 91 → 1 pasang

} Total: 7 pasang

c. 26-18-48-54-00-F9 → 6 pasang

Digabung menjadi : **07912618485400F9**

Berikut ini beberapa nomor SMS-*Centre* operator seluler di Indonesia.

***Cara 1:*** **Tabel 2.3 SMS *Centre* 1**

| No | Operator Seluler | SMS-*Centre* No | Kode PDU |
|---|---|---|---|
| 1 | Telkomsel | 0811000000 | 06818011000000 |
| 2 | Satelindo | 0816125 | 0581806121F5 |
| 3 | Exelcom | 0818445009 | 06818081440590 |
| 4 | Indosat-M3 | 0855000000 | 06818055000000 |

***Cara 2:*** **Tabel 2.4 SMS *Centre* 2**

| No | Operator Seluler | SMS-*Centre* No | Kode PDU |
|---|---|---|---|
| 1 | Telkomsel | 62811000000 | 07912618016666F0 |
| 2 | Satelindo | 62816125 | 059126181652 |
| 3 | Exelcom | 62818445009 | 07912618485400F9 |
| 4 | Indosat-M3 | 62855000000 | 07912658050000F0 |

**2. Tipe SMS**

Untuk *SEND* tipe SMS = 1. Jadi bilangan heksanya adalah **01**.

**3. Nomor Referensi SMS**

Nomor referensi ini dibiarkan dulu 0, jadi bilangan heksanya adalah **00**. Nanti akan diberikan sebuah nomor referensi otomatis oleh ponsel/alat *SMS-Gateway*.

**4. Nomor Ponsel Penerima**

Sama seperti cara menulis PDU *Header* untuk SMS-*Centre*, *Header* ini juga terbagi atas 3 bagian, sebagai berikut:

- Jumlah bilangan desimal nomor ponsel yang dituju dalam bilangan heksa.
- *National / International Code*.
  - Untuk *National*, kode *subheader*-nya: 81
  - Untuk *International*, kode *subheader*-nya: 91
- Nomor ponsel yang dituju dalam pasangan heksa dibalik-balik. Jika tertinggal satu angka heksa yang tidak memiliki pasangan, angka tersebut dipasangkan dengan huruf F didepannya.

Contoh:

Untuk nomor ponsel yang dituju = 628129573337 dapat ditulis dengan 2 cara sebagai berikut:

***Cara 1:*** 08129573337 diubah menjadi:

a. 0B → ada 11 angka

b. 81

c. 80-21-59-37-33-F7

Digabung menjadi : **0B818021593733F7**

***Cara 2:*** 628129573337 diubah menjadi:

d. 0C → ada 12 angka

e. 91

f. 26-18-92-75-33-73

Digabung menjadi : **0C91261892753373**

5. **Bentuk SMS,** antara lain:

   0 → 00 → dikirim sebagai SMS

   1 → 01 → dikirim sebagai *telex*

   2 → 02 → dikirim sebagai *fax*

   Dalam hal ini, untuk mengirim dalam bentuk SMS tentu saja kita memakai **00.**

6. **Skema Encoding Data I/O**

   Ada dua skema, yaitu:

   a. Skema 7 bit → ditandai dengan angka 0 → 00
   b. Skema 8 bit → ditandai dengan angka lebih besar dari 0 → diubah ke heksa.

   Kebanyakan ponsel/*SMS Gateway* yang ada di pasaran sekarang menggunakan skema 7 bit sehingga kita menggunakan kode **00**.

7. **Jangka Waktu Sebelum SMS *Expired***

   Jika bagian ini di-*skip*, itu berarti kita tidak membatasi waktu berlakunya SMS. Sedangkan jika kita isi dengan suatu bilangan *integer* yang kemudian diubah ke pasangan heksa tertentu, bilangan yang kita berikan tersebut akan mewakili jumlah waktu validitas SMS tersebut.

   Rumus untuk menghitung jangka waktu validitas SMS adalah sebagai berikut:

**Tabel 2.5 Validitas SMS**

| ***Integer* (INT)** | **Jangka Waktu Validitas SMS** |
|---|---|
| 0-143 | (INT + 1) x 5 menit (berarti: 5 menit s/d 12 jam) |
| 144-167 | 12 jam + ((INT -143) x 30 menit) |
| 168-196 | (INT – 166 ) x 1 hari |
| 197-255 | (INT – 192 ) x 1 minggu |

Agar SMS kita pasti terkirim sampai ke ponsel penerima, sebaiknya kita tidak memberikan batasan waktu validnya.

**8. Isi SMS**

*Header* ini terdiri atas dua *subheader*, yaitu:

a. Panjang isi ( jumlah huruf dari isi)

   Misalnya: untuk kata "*hello*" → ada 5 huruf → **05**

b. Isi berupa pasangan bilangan heksa

   - Untuk ponsel/*SMS Gateway* berskema *encoding* 7 bit, jika kita mengetikkan suatu huruf dari *keypad*-nya, berarti kita telah membuat 7 angka 1/0 berurutan.

Ada dua langkah yng harus kita lakukan untuk mengkonversikan isi SMS, yaitu:

Langkah pertama : mengubahnya menjadi kode 7 bit

Langkah kedua : mengubah kode 7 bit menjadi 8 bit, yang diwakili oleh pasangan heksa.

Contoh: untuk kata **"*hello*"**

Langkah Pertama:

| **Bit** | **7** | **1** |
|---|---|---|
| h | 110 | 1000 |

| | | |
|---|---|---|
| e | 110 | 0101 |
| l | 110 | 1100 |
| l | 110 | 1100 |
| o | 110 | 1111 |

| | | | | b7 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | b6 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| | | | | b5 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| b4 | b3 | b2 | b1 | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 0 | 0 | 0 | 0 | 0 | @ | Δ | SP | 0 | - | P | ¨ | p |
| 0 | 0 | 0 | 1 | 1 | | | ! | 1 | A | Q | a | q |
| 0 | 0 | 1 | 0 | 2 | $ | Φ | ˝ | 2 | B | R | b | r |
| 0 | 0 | 1 | 1 | 3 | | Γ | # | 3 | C | S | c | s |
| 0 | 1 | 0 | 0 | 4 | | Λ | | 4 | D | T | d | t |
| 0 | 1 | 0 | 1 | 5 | | Ω | % | 5 | E | U | e | u |
| 0 | 1 | 1 | 0 | 6 | | Π | & | 6 | F | V | f | v |
| 0 | 1 | 1 | 1 | 7 | | Ψ | ‘ | 7 | G | W | g | w |
| 1 | 0 | 0 | 0 | 8 | | Σ | ( | 8 | H | X | h | x |
| 1 | 0 | 0 | 1 | 9 | | Θ | ) | 9 | I | Y | i | y |
| 1 | 0 | 1 | 0 | 10 | LF | Ξ | * | : | J | Z | j | z |
| 1 | 0 | 1 | 1 | 11 | | | + | ; | K | Ä | k | ä |
| 1 | 1 | 0 | 0 | 12 | | | , | < | L | Ö | l | ö |
| 1 | 1 | 0 | 1 | 13 | CR | | - | = | M | | m | |
| 1 | 1 | 1 | 0 | 14 | | ß | . | > | N | Ü | n | ü |
| 1 | 1 | 1 | 1 | 15 | | | / | ? | O | | o | |

**Gambar 2.7 Skema 7 bit**

### 2.4.2 Jenis PDU SMS

Jenis PDU SMS yang akan digunakan adalah: SMS-Penerimaan (*SMS-DELIVER*) dan SMS-Pengiriman (*SMS-SUBMIT*).

#### 2.4.2.1 PDU Penerimaan (SMS-*Deliver*)

SMS Penerimaan (*SMS-Deliver*) adalah pesan yang diterima oleh terminal dari SMSC dalam bentuk PDU. PDU SMS-Penerimaan memiliki format seperti

pada Gambar 2.7 Pada PDU ini, terdapat beberapa meta-informasi yang dibawa, antara lain:

1. SCA (*Service Centre Address*),

   Berisi informasi SMS-*Centre*.

2. Tipe PDU (PDU *Type*),

   Berisi informasi jenis dari PDU tersebut

3. OA (*Originating Address*)

   Berisi informasi nomor pengirim.

4. PID (*Protocol Identifier*)

   Berisi informasi Identifikasi Protokol yang digunakan.

5. DCS (*Data Coding Scheme*)

   Berisi informasi skema pengkodean data yang digunakan.

6. SCTS (*Service Center Time Stamp*)

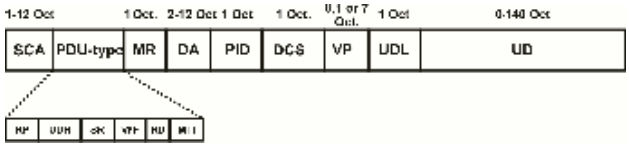
   Berisi informasi waktu.

7. UDL (*User Data Length*)

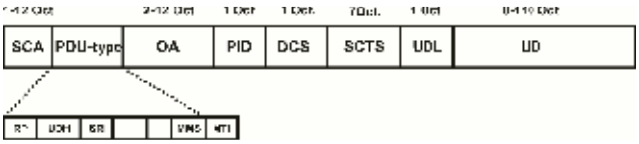
   Berisi informasi panjang dari data yang dibawa.

8. UD (*User Data*)

   Berisi informasi data-data utama yang dibawa.




**Gambar 2.8 Format PDU penerimaan**




**Gambar 2.9 Format PDU pengiriman**

### 2.4.2.2 PDU Pengiriman (SMS-*Submit*)

PDU Pengiriman memiliki informasi-informasi yang sama dengan PDU Penerimaan, sementara yang berbeda adalah berupa informasi (lihat Gambar 2.8):

1. MR (*Message Reference*),

Parameter yang mengindikasikan nomor referensi SMS-Pengiriman.

2. DA (*Destination Address*),

Berisi informasi nomor alamat yang dituju.

3. VP (*Validity Period*),

Berisi informasi jangka waktu validitas pesan pada jaringan.

### 2.4.3 Perintah AT (*AT COMMAND*)

Perintah AT (*Hayes* AT *Command*) digunakan untuk berkomunikasi dengan terminal (*modem*) melalui gerbang serial pada komputer. Dengan penggunaan perintah AT, dapat diketahui atau dibaca kondisi dari terminal, seperti mengetahui kondisi sinyal, kondisi baterai, mengirim pesan, membaca pesan, menambah *item* pada daftar telepon, dan sebagainya. Pada tabel 2.6 diperlihatkan beberapa jenis perintah AT yang berhubungan dengan penanganan pesan-pesan SMS.

**Tabel 2.6 Beberapa jenis perintah AT yang digunakan**

| *AT Command* | Fungsi |
|---|---|
| AT+CMGS | Mengirim pesan |
| AT+CMGR | Membaca pesan |
| AT+CMGF | Format pesan |
| AT+CMGD | Menghapus pesan |
| AT+CNMI | Prosedur indikasi pesan baru |
| AT+CPMS | Pemilihan target memori |
| AT+CSMS | Pemilihan layanan pesan |

## 2.5 Tinjauan Perangkat Lunak

Dalam pembuatan program aplikasi ini, penulis menggunakan perangkat lunak bahasa Pemrograman *Borland Delphi 7.0* dan *MySQL* sebagai perangkat lunak pengolah *database.*

### 2.5.1 *Borland Delphi 7.0*

Program aplikasi ini dibuat dengan menggunakan bahasa pemrograman *Borland Delphi 7.0* dengan pertimbangan bahwa *Borland Delphi 7.0* merupakan salah satu bahasa pemrograman atau pengembangan aplikasi yang sudah sangat dikenal dan mudah untuk membangun aplikasi berbasis *Window*. Selain itu produktifitas pengembangan perangkat lunaknya dibagi menjadi lima atribut penting, seperti :

- **Kualitas** dari lingkungan pengembangan visual.

- **Kecepatan** *compiler* dibandingkan dangan kompleksitasnya.
- **Kekuatan** dari bahasa pemrogaraman dibandingkan dengan kompleksitasnya.
- **Fleksibilitas** dari arsitektur basis data.
- **Pola desain dan pemakaian** yang diwujutkan oleh *framework*-nya.

Bahasa perograman *Borland Delphi 7.0* memberikan fasilitas untuk dua *platform*, yaitu *platform Windows* dan *Linux*. Namun disini penulis menggunakan *platform Windows*. *Delphi 7.0* menambahkan fitur-fitur baru dan memperbaiki fitur-fitur yang telah dimiliki oleh *Delphi 6.0.* perubahan-perubahannya dapat ditemukan pada kategori IDE, WEB, COM, *Database*, *Component Library, Runtime Library, Compiler, Rave reports, ModelMaker,* dan *Documentation*.

*Borland Delphi 7.0* merupakan salah satu aplikasi pemrograman visual yang dibuat oleh *Borland International Corporation* dam berjalan diatas *platform* (sistem operasi) *Windows*, dan sebagai pengetahuan yang berjalan diatas *platform Linux* adalah *Kylix*, yang merupakan saudara kembar pemrograman *Delphi*. Adapun rilis pertama (versi *Delphi* pertama) adalah tahun 1995, kemudian berlanjut sampai rilis ketujuh pada tahun 2002 dan akhirnya muncul rilis terbarunya yaitu *Borland Delphi 8.0* pada tahun 2005.

*Borland Delphi 7.0* menyediakan berbagai perangkat yang dapat digunakan untuk membuat program aplikasi, baik aplikasi kecil dan sederhana untuk keperluan sendiri hingga aplikasi untuk sistem *enterprise* yang besar dan rumit. *Borland Delphi 7.0* dapat menghasilkan program-program canggih, dimulai dari program multimedia, grafis sampai dengan pemrograman *database* yang

menggunakan jaringan. Bahkan dapat juga membuat program aplikasi yang dapat mengakses data *internet*.

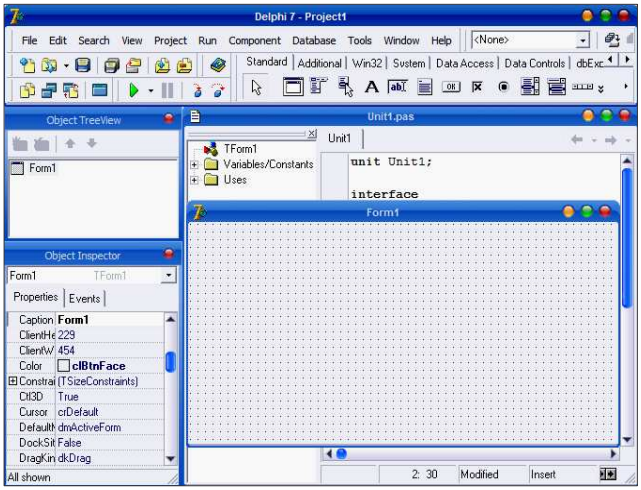
Antar muka yang dimiliki oleh *Borland Delphi 7.0* adalah antar muka yang berupa ruang kerja terpadu yang disebut IDE (*Integrated Development Environment*). IDE ini merupakan bagian dari *Borland Delphi 7.0* yang digunakan untuk menciptakan aplikasi. IDE inilah yang memungkinkan pemrograman secara visual merancang tampilan untuk para *user* (antarmuka pemakai) dan menuliskan listing program (kode).

Perubahan umum pada IDE pada *Borland Delphi 7.0* dibanding dengan versi sebelumnya adalah terletak pada :

- *Menu View Additional Massage Info* menampilkan jendela *massage hint*, yaitu informasi tambahan tentang *compiler massage* yag dapat di-*download* dari *website Borland*.
- Perubahan pada *component pallete*.
- Jika membuat aplikasi baru untuk CLX, pada *page system* akan ditampilkan beberapa komponen yang berhubungan dengan *file* dan direktori.
- Penambahan *page Indy Intercepts* dan *Indy I/O Handler Open Souerce* untuk komponen *Internet Protocol* pada edisi *professional* dan *Enterprise.*
- Penambahan *page IW Stendart, IW Data, IW Clien Side, IW Control page* yang menjadikan kompoen-komponen *IntraWeb* untuk pengembangan aplikasi berbasis *web*.

- ❖ Penambahan *page Rave* yang menyediakan komponen-komponen untuk keperluan pembuatan *report* .
- ❖ *Code Completition* yang lebih cepat. *Customize code completition manager* menggunakan *OpenTools* API.
- ❖ Dan masih banyak lagi *Option* maupun penambahan fitur lainnya pada IDE.

Antar muka *Borland Delphi 7.0* dapat diatur sesuai selera dan kebutuhan pengguna. Namun biasanya, *Borland Delphi 7.0* memiliki tampilan antar muka seperti gambar di bawah ini.




**Gambar 2.10 Lingkungan IDE *Borland Delphi 7.0***

IDE dalam pemrograman *Borland Delphi 7.0* terdiri dari beberapa bagian-bagian. Setiap bagiannya memiliki komponen-komponen dan fungsi-fungsi tertentu, diantaranya : *MenuBar, ToolBar/SpeedBar, Component palette, Jendela Form, Jendela Unit, Object TreeView,* dan *Object Inspector*. Berikut ini penulis

akan memberikan penjelasan lebih lanjut serta fungsi-fungsi dari komponen-komponen tersebut.

- ***MenuBar*** berfungsi memilih tugas-tugas tertentu seperti memulai, memilih, dan mengakhiri suatu aplikasi. *MenuBar* berisi berbagai macam perintah untuk proses pembuatan program aplikasi.




**Gambar 2.11** ***Menu Bar***

- ***ToolBar/SpeedBar*** berisi ikon-ikon yang mewakili perintah dasar yang ada pada *menu bar*, penggunaan *ToolBar* lebih praktis dan cepat. Misalnya perintah menyimpan, cukup dilakukan dengan cara mengklik ikon bergambar disket.



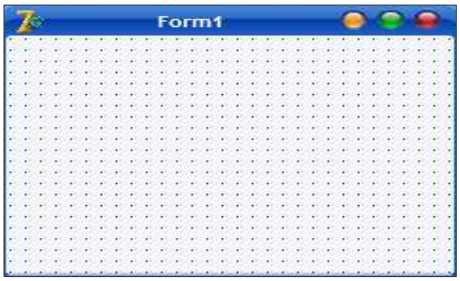
**Gambar 2.12** ***Tool Bar/ Speed Bar***

- ***Component Palette*** berisikan kumpulan-kumpulan komponen yang akan ditempelkan atau diletakkan dalam *form* dan digunakan untuk mendesain *form* sehingga membentuk *user interface*. Didalam *component palette* terdapat berbagai pilihan komponen, diantaranya: *Component Palette Standard, Win32, System, DataAcces, DataControls, dbExpress, DataSnap,* dan masih banyak lagi. Masing-masing memiliki banyak kontrol komponen disertai kegunaannya.



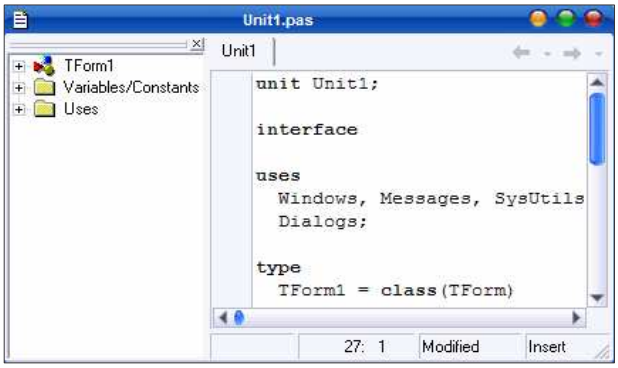

**Gambar 2.13** ***Component Palette***

❖ ***Jendela Form*** digunakan untuk merancang dan mendesain *user interface*. Disini pula kontrol-kontrol komponen dari *component palette* diletakkan. Dengan kata lain, *form* akan menjadi objek induk, semua objek-objek lain yang ada dakam *component palette*.



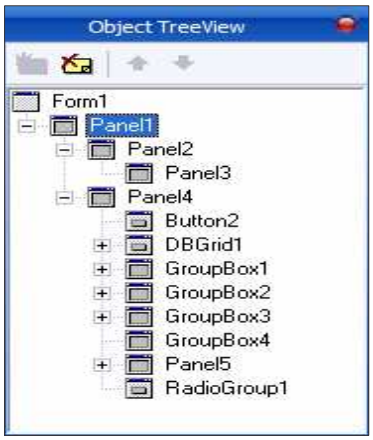

**Gambar 2.14** ***Jendela Form***

❖ ***Jendela Unit*** digunakan secara umum untuk menuliskan listing program dalam satu aplikasi. Didalamnya terdapat sebuah struktur unit.



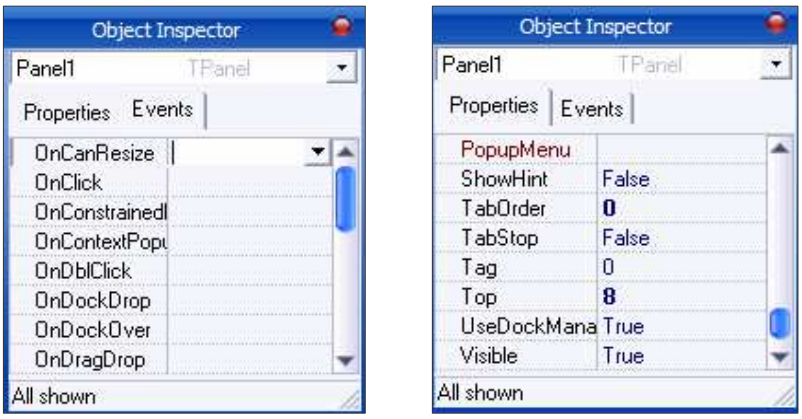

**Gambar 2.15** ***Jendela Unit***

❖ ***Object TreeView*** berfungsi untuk menampilkan keterangan atau menginformasikan semua objek atau kontrol program yang digunakan dalam membuat desain *form*. *Object TreeView* sangat berguna pada saat membuat program yang cukup besar.




**Gambar 2.16 *Object TreeView***

❖ ***Object Inspector*** digunakan untuk memanipulasi kontrol-kontrol yang sudah ada dalam *form*. *Object Inspector* memberikan informasi mengenai *properties* atau *attribute* objek yang sedang aktif. *Object Inspector* memiliki dua halaman, yang pertama disebut Properti (*Properties*) yang berfungsi untuk menampung *properties* masing-masing objek yang ada dalam *Delphi*. Yang kedua disebut Kejadian (*Event*) berfungsi menampung berbagai *event* yang ada dalam satu objek atau kontrol program yang sedang aktif




**Gambar 2.17** ***Object Inspector***

### *2.5.2 MySQL*

*MySQL* sebenarnya merupakan susunan salah satu konsep utama dalam database sejak lama, yaitu SQL (*Structure Query Language*). Keandalan suatu sistem *database* (DBMS) dapat diketahui dari cara kerja *optimizer*-nya dalam melakukan proses perintah-perintah SQL yang dibuat oleh *user* maupun program-program aplikasinya. Sebagai *database server*, *MySQL* dapat dikatakan lebih unggul dibandingkan *database server* lainnya dalam *query* data. *MySQL* adalah satu dari sekian banyak sistem *database*, merupakan terobosan solusi yang tepat dalam aplikasi *database*.