BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem

Telah diketahui bahwa informasi merupakan hal yang sangat penting bagi manajemen dalam pengambilan keputusan. Suatu sistem dapat dijelaskan dengan sederhana sebagai *suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu*. Pendekatan sistem lebih menekankan pada elemen atau komponennya, mendefinisikan sistem itu sebagai kumpulan dari elemen-elemen yang berinteraksi satu dengan yang lain. Informasi yang merupakan salah satu pembangun sistem dalam suatu organisasi sangatlah dibutuhkan, sehingga diperlukan suatu perencanaan yang matang untuk mendapatkan dan menyebarkannya.

2.1.1 Sistem

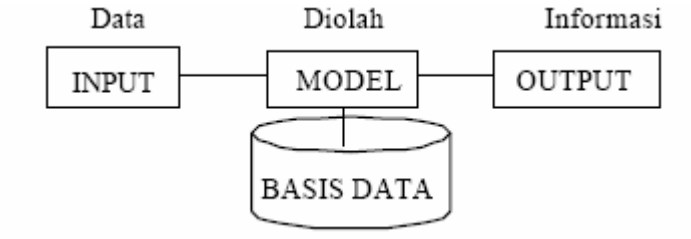
Pengertian dari sistem adalah kumpulan dari beberapa komponen untuk mencapai suatu tujuan yang sama.

2.1.2 Informasi

Pengertian dari informasi adalah “data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. Untuk mengolah data menjadi suatu informasi yang akurat, relevan dan tepat waktu maka diperlukan adanya sistem tertentu.” (*Drs. Thomas Wiyasa Bratawidjaja, 1995a, 1995b).*

2.1.3 Sistem Informasi

Pengertian dari sistem informasi adalah Sistem ysng mengolah sekumpulan data yang saling berinteraksi untuk diolah menjadi bentuk yang lebih berguna untuk mencapai tujuan yang diinginkan.



Gambar 2.1 Siklus Pengolahan data

Siklus Pengolahan Data

Sistem informasi mempunyai komponen :

a. Input merupakan data yang masuk.

b. Model menunjukkan suatu proses.

c. Output berupa informasi yang dihasilkan.

d. Basis Data adalah kumpulan data.

2.1.4 Sistem Informasi Manajemen

Sistem informasi manajemen *(Management Information System)* merupakan sistem manusia/mesin yang menyediakan informasi untuk mendukung operasi, manajemen dan fungsi pengambilan keputusan dari suatu organisasi. Tujuan penerapan sistem informasi di dalam organisasi salah satunya adalah untuk mendukung pembentukan informasi-informasi yang dibutuhkan oleh semua tingkatan manajemen.

2.2 Siklus Hidup Pengembangan Sistem

Siklus Hidup Sistem (*System Development Life Cycle/SDLC*) merupakan proses evolusioner yang diikuti dalam menerapkan sistem atau subsistem informasi berbasis komputer. Siklus Hidup Sistem terdiri dari sekumpulan tugas yang erat serta mengikuti langkah–langkah pendekatan sistem. Karena tugas–tugas tersebut mengikuti suatu pola yang teratur dan dapat dilakukan secara *topdown*. Salah satu metoda dalam SDLC adalah *Waterfall* yaitu siklus pengembangan sistem yang dapat dilakukan melalui tahapan berikut :

2.2.1 Tahap Perencanaan

Tahapan ini mendefinisikan permasalahan yang ada pada sistem. Dari pendefinisian masalah ini dapat menghasilkan suatu tujuan pengembangan sistem . Langkah – langkah dalam tahap perencanaan adalah sebagai berikut :

- Identifikasi masalah,
- Mendefinisikan masalah,
- Menentukan kebutuhan sistem,
- Mengidentifikasi kendala–kendala sistem,
- Membuat studi kelayakan,
- Mempersiapkan usulan penelitian sistem,
- Menetapkan mekanisme pengendalian.

2.2.2 Tahap Analisis

Tahap ini bertujuan untuk menganalisa sistem yang berjalan sehingga dapat diketahui jika ada kekurangan dan dapat diusulkan perbaikannya. Kesalahan pada tahap ini dapat berakibat sistem yang akan dibangun tidak dapat dipakai atau tidak dapat memenuhi kebutuhan user. Adapun langkah–langkahnya adalah sebagai berikut :

- Mendefinisikan kebutuhan informasi.
- Mendefinisikan kriteria kinerja sistem.
- Evaluasi sistem berjalan.
- Menyiapkan usulan rancangan.

2.2.3 Tahap Perancangan

Tahap perancangan dimaksudkan untuk mentranformasikan kebijakan user dan objek karakter menjadi suatu bentuk spesifikasi yang terstruktur sehingga menghasilkan esensial model. Adapun langkah–langkahnya adalah sebagai berikut :

- Menyiapkan rancangan sistem yang terinci,
- Mengidentifikasi berbagai alternatif konfigurasi sistem,
- Mengevaluasi berbagai alternatif konfigurasi sistem,
- Memilih konfigurasi terbaik,
- Menyiapkan usulan yang terbaik,

2.2.4 Tahap Implementasi

Tahap implementasi atau tahap penerapan merupakan sebuah tahap yang menerapkan hasil rancangan yang telah dibuat dalam lingkungan arsitektur software tertentu. Adapun langkah–langkahnya adalah sebagai berikut :

- Merencanakan penerapan,
- Mendapatkan sumber daya perangkat keras,
- Mendapatkan sumber daya perangkat lunak,
- Menyiapkan basis data,
- Menyiapkan fasilitas fisik,
- Mendidik peserta dan pemakai,
- Menyiapkan usulan cutover,
- Test Sistem

2.3 Kategori Model

Model mempunyai peran penting dalam pengembangan sistem. Model dapat merupakan rujukan bagi kelengkapan dan kualitas hasil. Dalam bidang pengembangan sistem, dikenal tiga kategori umum model .

Adapun kategori itu adalah sebagai berikut :

Kategori	Tujuan
Model Fungsional	Merepresentasikan fungsi-fungsi dalam sistem.
Model Behavioral	Merepresentasikan "kelakuan" sistem yang "*Time dependent"*.
Model Data	Merepresentasikan "*Real world*" melalui data, sekaligus berfungsi sebagai kerangka organisasi data.

Tabel 2.1 Kategori Model

Tools pemodelan yang dapat digunakan dalam pekerjaan analisis dan desain diantaranya:

Nama Model	Kategori	Penggunaan
Diagran Alur Data	Fungsional	Menggambarkan suatu sistem sebagai fungsi transformasi data
Diagram Konteks (*Contex Diagram*)	Fungsional	Menggambarkan interaksi (sistem sebagai *black-box*) dengan lingkungannya, mendeskripsikan fungsi sebuah sistem
Daftar dan Struktur kejadian (*Event List*)	*Behavioral*	Mendeskripsikan *event* sistematik beserta strukturnya
Kamus Data (*Data Dictionary*)	Data	Mendeskripsikan isi dan semantik dari data, laporan, dan formulir tsb

Entity Relationship Diagram	Data	Merepresentasikan "*Real Word*" melalui datanya, membuat kerangka, konseptual bagi organisasi data
Skema Data Relational	Data	Merepresentasikan hubungan matematis atau logis antar data, menyediakan kerangka logis bagi organisasi dan manajemen data
Skema Struktur Data	Data	Merepresentasikan karakteristik variabel yang dipergunakan oleh program aplikasi
Skema Struktur perangkat lunak	Fungsional	Merepresentasikan "keterhubungan" antar modul *software* atau disebut sebagai arsitektur/organisasi *software*
Algoritma	Fungsional dan behavioral	Mendeskripsikan logika proses

Tabel 2.2 Model analisis dan desain

2.3.1 Lingkungan (*Environmental*) Model

Environmental model yaitu model yang mendefinisikan batas antara sistem dengan lingkungan luarnya. Dalam model *environmental* ini dapat dibuat pemodelan dengan Tools :

a. Context Diagram

Pada context diagram dapat dilihat kelompok pemakai, organisasi atau sistem lain yang berhubungan dengan sistem yang dikembangkan " *Context Diagram* juga memperlihatkan input maupun output dari sistem lain dan juga penyimpanan yang digunakan bersama dengan sistem lain. Dalam Contex Diagram, sistem direpresentasikan melaui sebuah *bubble*". *(Modern Struktured, Edward Yourdon, 339)*

b. Event List

Event List yaitu daftar kejadian yang terjadi diluar sistem yang mampu merangsang sistem yang dikembangkan untuk memberikan respon sehubungan rangsangan yang dterima.

Komponen–komponen sistem yang digunakan pada *event list* adalah sebagai berikut :

No. *Event*	Nomor *Event*
Nama *Event*	Nama suatu kejadian
Deskripsi	Penjelasan singkat dari suatu kejadian
Frekuensi	Waktu rata–rata suatu kejadian
Respon	Respon dari sistem atas suatu kejadian
Input	Data input yang digunakan
Output	Data output yang digunakan
Pelaku	Pelaku yang terlibat di dalamnya

Tabel 2.3 Komponen–komponen sistem yang digunakan pada event list

c. Data Flow Diagram

DFD merupakan penurunan dari model diagram konteks yang merupakan suatu tools yang menunjukan aliran informasi atau data sumber/tujuan data, proses dan tempat penyimpanan data. “DFD menggambarkan sistem sebagai sebuah jaringan proses yang dihubungkan satu sama lainnya melalui suatu aliran data dan data store”. (*EdwardYourdon, Modern Strujtured Anaysis, Prantice Hall Inc, 1989).*

2.3.2 Behavioral Model

Behavioral model adalah model yang mendefinisikan aktifitas-aktifitas yang perlu ada di dalam sistem untuk berinteraksi dengan lingkungan luar sistem. Behavioral model ini dibagi menjadi dua bagian, yaitu :

- *Behavioral Model Awal*

  Merupakan aktifitas pengidentifikasian respon sistem terhadap kejadian/event di luar sistem yang harus direspon oleh sistem.

- *Behavioral Model Akhir*

  Tahap behavioral model akhir ini lebih lanjut akan dibahas dalam tahap design detail. Hasil yang diharapkan dari tahap ini adalah arsitektur umum sistem informasi, arsitektur umum *software* dan *hardware*, arsitektur umum *hardware* dan jaringan, rencana implementasi. Sedangkan kegiatan yang dilakukan adalah pemodelan proses s/d level fungsional, pemodelan data s/d level logika, pemilihan

sistem komputer sistem dan jaringan, pembuatan rencana implementasi.

Tahap perancangan detail merupakan tahap behavioral model akhir. Pada tahap ini diharapkan dapat dihasilkan file/data, program, format masukan/keluaran, spesifikasi sistem komputer dan jaringan, sistem prosedure. Dan kegiatan yang dilakukan dalam tahap ini adalah pembuatan file (*database*), pembuatan format masukan/keluaran, pembuatan program, penyusunan panduan dan acuan penyiapan data, pelatihan, uji coba dan modifikasi.

Dalam tahap ini dapat digunakan tools permodelan, yaitu:

Penyelesaian pemodelan proses, meliputi pembuatan :

a. Data Flow Diagram
b. Spesifikasi Proses

Merupakan tools yang digunakan untuk menjelaskan kejadian yang terjadi pada setiap proses yang terdapat pada DFD level yang paling bawah.

Spesifikasi proses mendefinisikan kegiatan yang harus dilakukan untuk mengubah input menjadi output. *(Modern Structured Analysis , Edward Yourdon, 203)*. Tujuannya adalah untuk memperjelas input atau proses, event kerja dalam proses serta kelebihan dan kelemahannya.

Nama	Uraian
No. proses	Urutan proses
Nama proses	Nama proses berdasarkan dfd pada level terbawah beserta nomornya.
Proses yang dipanggil	Proses lain yang dipanggil oleh proses ini
Proses yang memanggil	Proses lain yang memanggil proses ini
Input	Masukan aliran data ke proses.
Output	Keluaran aliran data dari suatu proses.
Logika proses	Berisi narasi/logika setiap proses pada level terbawah.

Tabel 2.4 Spesifikasi Proses

Penyelesaian pemodelan data, meliputi:

a. Data dictionary (kamus data)

Kamus data merupakan bentuk dokumentasi yang mendukung diagram aliran data (DFD) dalam sistem yang dirancang. Kamus data ini berupa kumpulan data yang mendefinisikan data-data yang tersimpan serta menghubungkan aliran data dalam sistem yang dikembangkan.

Kamus data terdiri dari dua macam yaitu kamus data komposit dan kamus data elementer.

- Kamus data komposit digunakan untuk menjelaskan komposisi dari paket data yang kompleks yang dapat dipecah menjadi beberapa item-item elementer. (*Modern Structured Analysis, Edward Yurdon, Hal 189*).
- Kamus data elementer digunakan untuk menjelaskan data-data elementer, yang tidak dapat di pecah-pecah lagi. (*Modern Structured Analysis, Edward Yurdon, Hal 193*). Adapun simbol-simbol yang digunakan dalam kamus data, diperlihatkan pada tabel berikut ini :

No.	Simbol	Uraian
1.	=	terdiri dari, mendefinisikan, diuraikan menjadi, artinya
2.	+	Dan
3.	{ }	iterasi / pengulangan
4.	[ ]	seleksi dari beberapa alternatif
5.	-	hingga, biasanya merupakan elemen pendefinisi yang sequential seperti numerik atau alphabet, misalnya : A-Z, atau 0-9
6.	|	Pemisah alternatif / pilihan dalam [ ]

Tabel 2.5 Kamus Data

b. *Entity Relationship Diagram* (*ERD*)

ERD adalah suatu model network yang menjelaskan layout penyimpanan data pada suatu abstraksi level tinggi (*Modern Structured Analysis , Edward Yourdon, 233*).

Dengan kata lain, ERD merupakan diagram yang menggambarkan model data yang digunakan dalam suatu sistem secara keseluruhan.

Komponen-komponen ERD:

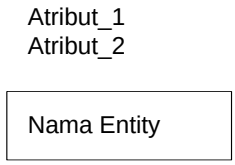
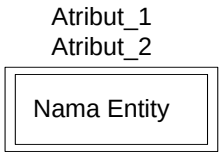
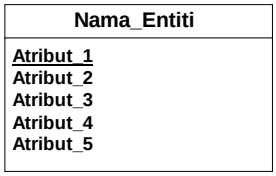
- **Entity/Object**

Entity adalah suatu objek yang merepresentasikan suatu himpunan atau sesuatu di dunia nyata yang mempunyai peranan dalam sistem yang sedang dibangun, dapat diidentifikasi secara unik dan dapat dijelaskan dengan satu atau lebih atribut.(Modern Structured Analysis, Edward Yourdon, Hal 70).

Entity memiliki ciri-ciri sebagai berikut:

- Memiliki identitas yang unik sehingga dapat dibedakan dengan objek yang lainnya.
- Memiliki peranan dalam sistem, sehingga sistem tidak dapat bekerja tanpa mengaksesnya.
- Dapat dilukiskan oleh satu atau lebih elemen.

Simbol Entity yang digunakan :

Simbol	Deskripsi
	Entity disamping ini merupakan strong entity mempunyai dua atribut dimana atribut_1 merupakan *key* dari entity tersebut
	Entity disamping merupakan weak entity karena keberadaannya tergantung pada entity parent-nya . Bisa memiliki key, bisa tidak.
	Entity disamping ini mempunyai lima buah atribut dan Atribut_1 merupakan *Key* dari entity tersebut

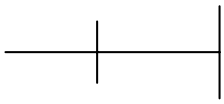
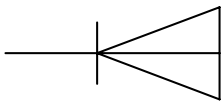
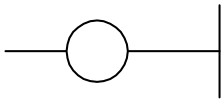
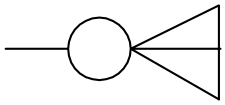
Tabel 2.6 Entity / Objek

- **Relationship**

  Relationalship adalah hubungan yang diberi nama diantara entity. *(Data Architect Users Guide, S-Designor, Hal 71).* Kardinaliti mengidentifikasikan jumlah keberadaan dari sebuah entity dalam relasinya dengan entity yang lain. *(Data Architect Users Guide, S-Designor, Hal 101).*

- ***Termination Points***

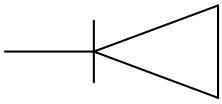
Termination points mengindikasikan kardinaliti pada setiap *relationalship*. Berikut ini daftar simbol dari *termination points*.

Termination Point	Eksistensi	Kardinaliti	Deskripsi
	Mandatory	One	Harus ada satu dan hanya satu
	Mandatory	Many	Harus ada satu dan Dapat lebih dari satu
	Optional	One	Dapat ada satu, atau tidak ada
	Optional	Many	Dapat ada satu, atau lebih dari satu, atau tidak ada

Tabel 2.7 Termination Point

- ***Dependent Relationalship***

Pada sebuah dependent *relationalship*, suatu entity teridentifikasi secara parsial oleh entity yang lainnya. (Data Architec Users Guide, S-Designor, Hal 104). Untuk simbol dari termination points dari *dependent relationalship* adalah sebagai berikut :

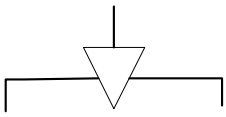
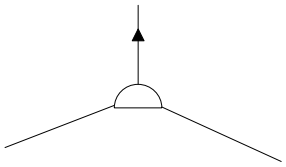
Termination Point	Eksistensi	Kardinaliti
	Mandatory	[One | Many]

Tabel 2.8 Dependent Relationalship

- ***Inheritance***

Inheritance adalah *relatiolalship* khusus yang digunakan untuk mendefinisikan sebuah entity sebagai sebuah kasus khusus dari sebuah entity yang lebih umum. Entity yang lebih umum disebut *Super Type (Parent),* dan entity yang lebih khusus disebut *Sub Type (Child).* (Data *Architec Users Guide, S-Designor, Hal 109*)

Berikut ini simbol dari Inheritance Link yang menghubungkan antara Parent dan Child-nya.

Simbol	Keterangan
	Simbol disamping ini merupakan simbol “IS A”
	Simbol disamping ini merupakan simbol dari *Inheritance Link*

Tabel 2.9 Inheritance

c. Skema Relasi ERD

Transformasi dari entity Relationship Diagram ke Skema Relational perlu dilakukan karena ERD belum merepresentasikan hubungan diantara file-file yang diperlihatkan oleh suatu bidang aplikasi. Skema relasi adalah daftar dari atribut-atribut dan domain-domain yang berhubungan dengan masing-masing atribut tersebut. *(Database System Concepts; 2nd Edition, Henry F. Korth, Abraham Silberschatz, Hal. 55).*

Normalisasi

Normalisasi merupakan tahap modifikasi dari skema relational database, berupa pengelompokan data elemen ke dalam suatu tabel yang menunjukkan entity dan relasinya.

Normalisasi adalah proses dekomposisi struktur data yang komplek ke dalam bentuk tabel yang berelasi. *(Jerry FitzGerald dan Andra F. FitzGerald, Fundamentals of System Analysis, Hal. 85)*

Tujuan dari dilakukannya normalisasi adalah untuk menghindarkan database dari :

- Redundansi data yang tidak perlu.
- Ketidakmampuan dalam merepresentasikan beberapa informasi tertentu.
- Kehilangan informasi.

(*Henry F. Korth dan Abraham Silberschatz, Database System Concepts, Hal. 173)*

Sebelum melangkah lebih lanjut, maka akan dijelaskan dulu mengenai definisi dari jenis-jenis key yang ada. Ada beberapa jenis key yang dikenal, diantaranya :

- *Superkey* : satu atau beberapa atribut yang dapat mengidentifikasi secara unik sebuah tuple dari tuple-tuple lainnya pada skema relasi
- *Candidate key* : *superkey* yang minimal, superkey yang subsetnya tidak mengandung superkey lagi
- *Primary key* : *candidate key* yang dipilih oleh designer database untuk mengidentifikasi sebuah baris dari baris-baris lainnya pada suatu table
- *Foreign key* : atribut pada suatu tabel yang nilainya bersesuaian dengan *primary key* pada tabel lain, berfungsi untuk merepresentasikan *relationship* antar tabel

Adapun bentuk normal yang digunakan pada desain ini antara lain :

Normalisasi bentuk pertama (1^{st} NF)

Suatu skema termasuk dalam bentuk normal ini jika dan hanya jika tidak terdapat pengulangan atribut (sudah atomik)

Normalisasi bentuk kedua (2^{nd} NF)

Suatu skema termasuk dalam bentuk normal ini, jika :

- skema berada dalam bentuk *1^{st} NF*, dan
- setiap atribut non-key pada skema ini tidak tergantung sebagian pada candidate key

*Normalisasi bentuk ketiga (*3rd NF)

Suatu skema termasuk dalam bentuk normal ini, jika :

- X → Y trivial, atau
- X → Y non-trivial, X adalah *superkey*, atau
- X → Y non-trivial, X bukan *superkey*, Y subset dari candidate key

Normalisasi Boyce Codd (BCNF)

Suatu skema termasuk dalam bentuk normal ini, jika :

- X → Y trivial, atau
- X → Y non-trivial, X adalah *superkey*

Integrity Constraint

Integrity constraint adalah suatu cara untuk menjamin konsistensi data pada database ketika terjadi transaksi insert, delete ataupun update *(James R. Groof dan Paul N. Weinberg, LAN Times; Guide to SQL, Hal. 261)*

Beberapa cara untuk menerapkan integrity constraint, diantaranya :

- Deklarasi *key*. Suatu cara untuk mengidentifikasi secara unik sebuah baris dari baris-baris lainnya pada suatu tabel, telah diterapkan mulai ERD
- *Relationship*. Suatu cara untuk merepresentasikan hubungan antara entity, termasuk di dalamnya adalah kardinaliti, diterapkan mulai ERD seperti halnya *deklarasi key*
- Domain Constraints. Menjelaskan tipe data, panjang data, dan nilai-nilai yang valid untuk setiap atribut.
- *Referential Integrity*. Konsep mengenai penerapan *foreign key*, suatu atribut yang nilainya bersesuaian dengan *primary key* pada tabel lain

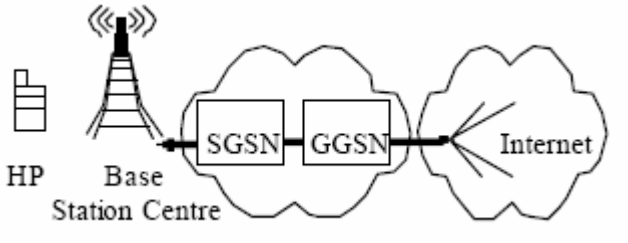
- *Trigge*r. Suatu statement yang dieksekusi secara otomatis oleh sistem sebagai efek samping dari modifikasi pada data. Konsep trigger ini dapat digunakan untuk menerapkan *rule* yang tidak dapat direpresentasikan dengan *constraint-constraint* lainnya

2.4 Struktur Tabel

Pada level konseptual atau logik, database pada model relational dipresentasikan dengan sebuah tabel. Pada level implementasi, digunakan pendekatan yang lebih dapat menggambarkan tabel-tabel tersebut menjadi gabungan record-record yang disimpan dalam suatu table, yaitu dengan menggambarkan struktur table yang merepresentasikan tabel-tabel yang dihasilkan dari model data relational.

2.5 Sekilas Tentang SMS *Gateway*

SMS *gateway* merupakan sistem informasi yang menghubungkan antara teknologi selular dengan teknologi komputer. Dimana kita dapat mengakses informasi yang dibutuhkan melalui sms ke sebuah operator. Data dari sms akan diolah oleh komputer untuk menghasilkan informasi yang dibutuhkan sesuai dengan keinginan orang yang melakukan sms.



Gambar2.2 Jaringan GPRS Dengan Internet

2.6 Konsep Dasar Komunikasi Data

- **Komunikasi**

  Pertukaran informasi antara 2 pihak dengan menggunakan simbol, suara, dll. Yang dapat difahami diantara keduanya melalui media perantara.

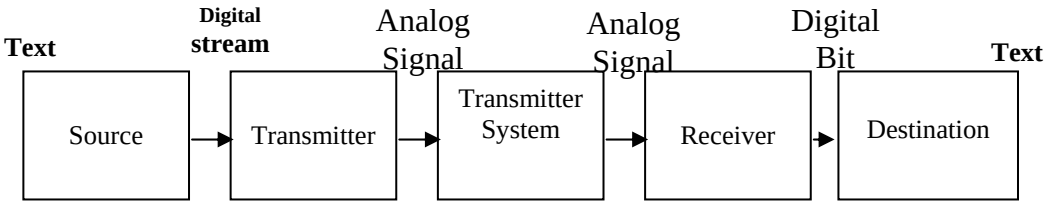
  Media perantara adalah media yang membolehkan informasi sampai pada tujuannya.

- **Komunikasi Data**

  Merupakan perpindahan data melalui media komputer yang diwakili oleh digit-digit biner dari satu tempat ke tempat lain dalam bentuk elektronik, gelombang atau cahaya.

  Elemen komunikasi data, antara lain:

- ✓ Source (Sumber): Alat ini membangkitkan data sehingga dapat ditransmisikan.
- ✓ Transmitter(Pengirim): Biasanya data yang dibangkitkan dari sistem sumber tidak ditransmisikan secara langsung dalam bentuk aslinya.
- ✓ Transmitter System(Sistem Transmisi): Berupa jalur transmisi yang menghubungkan antara sumber dengan tujuan.
- ✓ Receiver(Penerima): Receiver menerima sinyal dari sistem tranmisi dan menggabungkannya ke dalam bentuk tertentu yang dapat ditangkap oleh tujuan.
- ✓ Destination(Tujuan): Menangkap data yang dihasilkan oleh receiver.



Gambar 2.3 Komunikasi Data