# BAB II

# LANDASAN TEORI

Bab ini menjelaskan mengenai teori-teori yang akan digunakan dalam membangun sistem.

## 2.1. Basis Data

### 2.1.1. Definisi Basis Data

Basis data terdiri atas dua kata [1], yaitu *basis* dan *data*. *Basis* kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang/berkumpul. Sedangkan *data* adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, kedaan, dan sebagainya, yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya.

Basis data dapat didefinisikan dalam sejumlah sudut pandang seperti [1]:

- Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan dengan cepat dan mudah.
- Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.

- Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

### 2.1.2. Operasi Dasar Basis Data

Didalam sebuah *disk*, basis data dapat diciptakan dan dapat pula ditiadakan. Dalam sebuah basis data, kita dapat menempatkan satu atau lebih file/tabel. Pada tabel inilah sesungguhnya data disimpan/ditempatkan. Setiap basis data umumnya dibuat untuk mewakili sebuah semesta data yang spesifik, misalnya basis data kepegawaian, basis data akademik, basis data inventori (pergudangan), dan sebagainya.

Operasi-operasi yang dapat dilakukan berkenaan dengan basis data dapat meliputi:

- Pembuatan basis data baru (*create databse*).
- Penghapusan basis data (*drop database*).
- Pembuatan file/tabel baru (*create table*).
- Penghapusan file/tabel dari suatu basis data (*drop table*).
- Penambahan/pengisian data baru ke sebuah file/tabel pada sebuah basis data (*insert*).
- Menampilkan data dari sebuah file/tabel (*select*).
- Pengambilan data dari sebuah file/tabel (*retrieve*/*search*).
- Pengubahan data dari sebuah file/tabel (*update*)
- Penghapusan data dari sebuah file/tabel (*delete*).

## 2.2. Pemodelan Analisis

Model analisis, yang sebenarnya merupakan serangkaian model, merupakan representasi teknis yang pertama dari sistem. Tetapi saat ini ada dua yang mendominasi landasan pemodelan analisis. Yang pertama, analisis terstruktur, adalah metode pemodelan klasik, dan analisis berorientasi objek.

Analisis terstruktur adalah aktivitas pembangunan model. Dengan menggunakan notasi yang sesuai dengan prinsip analisis operasional dapat menciptakan model yang menggambarkan muatan dan aliran informasi (data kontrol), membagi sistem secara fungsional dan secara *behavioral*, dan menggambarkan esensi dari apa yang harus dibangun.

*Entity-Relationship Diagram* (ERD) adalah notasi yang digunakan untuk melakukan aktivitas pemodelan data. Atribut dari masing-masing objek data yang ditulis pada ERD dapat digambarkan dengan menggunakan deskripsi objek data. Sedangakan *data flow diagram* (DFD) memberikan informasi tambahan yang digunakan selama analisis domain informasi dan berfungsi sebagai dasar bagi pemodelan fungsi.

### 2.2.1. Entity Relationship Diagram (ERD)

ERD hanya berfokus pada data, dengan menunjukkan “jaringan data” yang ada untuk suatu sistem yang diberikan. ERD sangat berguna bagi aplikasi di mana data dan hubungan yang mengatur data sangatlah kompleks. ERD pada mulanya diusulkan oleh **Peter Chen** untuk desain sistem *database* relasional dan telah dikembangkan oleh yang lainnya. Serangkaian komponen utama diindentifikasikan untuk ERD: objek data, atribut, hubungan, dan berbagai tipe

indikator. Tujuan utama dari ERD adalah untuk mewakili objek data dan hubungan mereka.

### 2.2.2. Data Flow Diagram (DFD)

*Data Flow Diagram* (DFD –DAD/Diagram Alir Data) memperlihatkan hubungan fungsional dari nilai yang dihitung oleh sistem, termasuk nilai masukan, nilai keluaran, serta tempat penyimpanan internal. DAD adalah gambaran grafis yang memperlihatkan aliran data dari sumbernya dalam objek kemudian melewati proses yang mentransformasinya ke tujuan yang lain, yang ada pada objek lain. DAD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangan lingkungan fisik dimana data tersebut mengalir. DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur (*structured analysis and design*). DFD merupakan alat yang cukup populer sekarang ini, karena dapat menggambarkan arus data di dalam sistem dengan terstruktur jelas.

Beberapa simbol yang digunakan dalam *Data Flow Diagram* (DFD) antara lain:

1. ***External Entity*** **(kesatuan luar) atau** ***boundary*** **(batas sistem)**

   Setiap sistem pasti mempunyai batas sistem (*boundary*) yang memisahkan suatu sistem dengan lingkungan luarnya. Sistem akan menerima *input* dan menghasilkan *output* kepada lingkungan luarnya. Kesatuan luar (*external entity*) merupakan kesatuan (*entity*) di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem

lainnya yang berada di lingkungan luarnya yang akan memberikan *input* atau menerima *output* dari sistem.

2. ***Data Flow*** **(arus data)**

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

3. ***Process*** **(proses)**

   Suatu proses adalah kegiatan atau kerja yang dilakukan oleh orang, mesin atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses. Setiap proses harus diberi penjelasan yang lengkap meliputi identifikasi proses, nama proses dan pemroses.

4. ***Data Store*** **(simpanan luar)**

   Simpanan data (*data store*) merupakan simpanan dari data yang dapat berupa, yaitu suatu file atau *database* di sistem komputer, suatu arsip atau catatan manual, suatu kotak tempat data di meja seseorang, suatu tabel acuan manual, dan suatu agenda atau buku.

## 2.3. Diagram Konteks

Diagram konteks menggambarkan hubungan antara sistem dengan entitas luarnya. Diagram konteks berfungsi sebagai transformasi dari satu proses yang

melakukan transformasi data *input* menjadi data *output*. Entitas yang dimaksud adalah entitas yang mempunyai hubungan langsung dengan sistem.

Suatu diagram konteks selalu mengandung satu dan hanya satu proses saja. Proses ini mewakili proses dari seluruh sistem. Diagram konteks ini menggambarkan hubungan *input* atau *output* antara sistem dengan dunia luarnya (kesatuan luar).

## 2.4. Kamus Data

Kamus data (KD) atau *data dictionary* (DD) atau disebut juga dengan istilah *system data dictionary* adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Dengan menggunakan kamus data, analisis sistem dapat mendefinisikan data yang mengalir di sistem dengan lengkap. Kamus data dibuat pada tahap analisis sistem dan digunakan baik pada tahap analisis maupun pada tahap perancangan sistem.

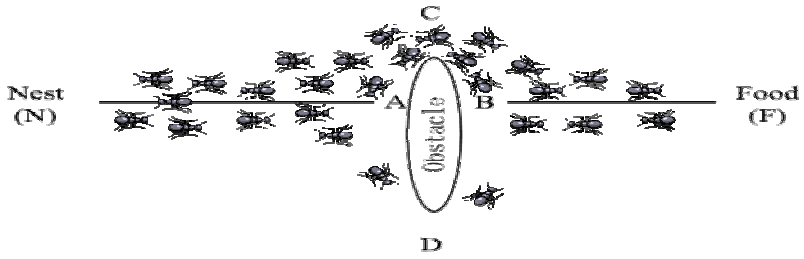
Pada tahap analisis, kamus data dapat digunakan sebagai alat komunikasi antara analisis sistem dengan pemakai sistem tentang data yang mengalir di sistem, yaitu tentang data yang masuk ke sistem dan tentang informasi yang dibutuhkan oleh pemakai sistem.

Pada tahap perancangan, kamus data digunakan untuk merancang *input*, merancang laporan-laporan dan *database*. Kamus data dibuat berdasarkan arus data yang ada di DFD. Arus data di DFD sifatnya global, hanya ditunjukkan nama arus datanya saja. Keterangan lebih lanjut tentang struktur data dari arus data di DFD secara lebih rinci dapat dilihat di kamus data. Kamus data harus dapat

mencerminkan keterangan yang jelas tentang data yang dicatatnya, maka kamus data harus memuat nama arus data, alias, bentuk data, arus data, penjelasan, periode, volume, dan struktur data.[2]

## 2.5. Algoritma *Ant Colony System*

*Ant Colony Optimization* (ACO) pertama kali diperkenalkan oleh Marco Dorigo, ACO itu sendiri terinspirasi oleh koloni-koloni semut dalam mencari makan. Semut-semut tersebut meninggalkan zat (*pheromone*) di jalan yang mereka lalui. Algoritma ACO ini merupakan algoritma pencarian berdasarkan probabilistik, di mana probabilistik yang digunakan merupakan probabilistik dengan bobot sehingga butir pencarian dengan bobot yang lebih besar akan berakibat memiliki kemungkinan terpilih yang lebih besar pula.




**Gambar 2.1.** Perilaku nyata semut ketika mencari makan

Dari gambar 2.1 di atas terlihat dalam perjalanannya, semut-semut dari sarang menuju ke tempat makanan ketika menemui hambatan maka semut-semut tersebut akan memilih jalan yang berbeda, jalur A-C-B merupakan jalur yang paling banyak dilalui semut daripada jalur A-D-B sehingga kandungan *pheromone* A-C-B lebih besar daripada kandungan *pheromone* A-D-B, dengan demikian alur A-C-B akan memiliki kemungkinan untuk dilalui kembali.

*Ant Colony Optimation* itu sendiri dapat diterapkan untuk menyelesaikan berbagai macam masalah diantaranya *Traveling Salesman Problem, Minimum Spanning Tree Problem, Scheduling Problems* dan lain-lain. Algoritma umum dari ACO adalah sebagai berikut:

```
masukan data permasalahan
while not selesai do
      bangkitkan semut
      bangkitkan solusi
      update pheromone
      hancurkan semut
end while
keluarkan solusi
```

**Gambar 2.2** Algoritma umum ACS

Algoritma ACO terdiri atas dua versi besar, yaitu :

1. ACS (Ant Colony System)
2. MMAS (*MAX-MIN Ant System*)

Perbedaan umum dari kedua versi itu adalah pada bagian *update pheromone*. Dan pada fungsi probabilistik ACS menggunakan fungsi heuristik sedangkan MMAS tidak menggunakan fungsi heuristik. Selain itu, pada MMAS terdapat parameter *pheromone maximum* dan *pheromone minimum* yang tidak diterapkan pada ACS. Alasan penggunaan batasan maksimum dan minimum tersebut adalah untuk memberikan keseimbangan level intensifikasi dan diversifikasi pencarian. Penggunaan nilai maksimum yang terlalu tinggi akan berakibat semua semut menghasilkan solusi yang sama sehingga algoritma

menjadi stagnan. Penggunaan nilai minimum yang terlalu rendah bahkan sama dengan nol akan berakibat jalur tersebut semakin tidak berpeluang untuk dipilih sehingga akan membatasi ruang solusi.

Dalam peghitungan nilai probabilistik untuk setiap kemungkinan pada ACS menggunakan nilai heuristik sedangkan pada MMAS heuristik tersebut dihilangkan. Selain hal tersebut perbedaan antara ACS dan MMAS dapat terlihat dari persamaan *update pheromone*, pada ACS *update pheromone* dilakukan dua kali yaitu setiap semut melewati node (*update pheromone* lokal) dan setiap satu koloni semut melewati node (*update pheromone* global), sedangkan pada MMAS hanya dilakukan *update pheromone* global, tetapi pada MMAS digunakan parameter $T_{\max}$ (*pheromone* maksimum) dan $T_{\min}$ (*pheromone* minimum).

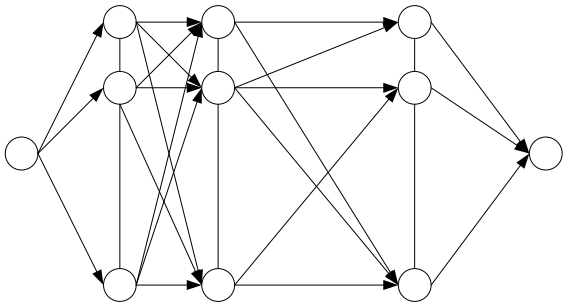
**Tabel 2.1** Parameter yang digunakan dalam ACO

| Nama parameter | Keterangan |
|---|---|
| $\tau_{(e,t)}$ | *Pheromone* dari event *e* dan *timeslot t* |
| $\rho$ | Derajat penguapan *pheromone* |
| g | Faktor skala |
| $q(C_{global_best})$ | Nilai kecocokan solusi dari solusi terbaik global |
| $\tau_{\min}$ | *Pheromone* minimum |
| $\tau_{\max}$ | *Pheromone* maksimum |

## 2.6. Model Matematis Graf sebagai Representasi *Hard Constraint*

*Constraint* adalah kendala yang didefinisikan agar penjadwalan yang dihasilkan merupakan jadwal yang baik, *constraint* itu sendiri secara umum dibagi menjadi dua kategori yaitu: *soft* (lembut) dan *hard* (keras). *Soft constraint* adalah kendala-kendala yang memungkinkan untuk dilanggar karena pelanggaran terhadap *soft constraint* tidak akan mempengaruhi kebenaran hasil penjadwalan, namun jika terjadi pelanggaran terhadap *soft constraint* ini maka jadwal yang dihasilkan bukan merupakan hasil yang nyaman untuk digunakan. Sedangkan *hard constraint* adalah kendala di mana tidak boleh terjadi pelanggaran terhadap kendala-kendala yang telah ditetapkan. Untuk merepresentasikan *hard constraint* yang ada, maka dibangun sebuah model matematis dalam bentuk graf.

Untuk masalah penjadwalan yang akan diselesaikan dengan menggunakan algoritma MMAS ini, graf yang dibangun merupakan graf berarah. Node-node yang terbentuk nantinya merupakan representasi dari event-event yang akan dijadwalkan pada waktu *t*. Sedangkan keterhubungan antar node-node yang terbentuk merupakan representasi dari *hard constraint* yang ada, sebagai contoh dapat dilihat pada gambar 2.3 di bawah ini.



**Gambar 2.3** Graf berarah yang merupakan pasangan *event* dan *timeslot*

## 2.7. *Short Message Service (SMS)*

### 2.7.1. Mekanisme Distribusi Pesan SMS

Terdapat empat macam mekanisme distribusi pesan SMS oleh aplikasi SMS, yaitu:

a. Pull, yaitu pesan yang dikirimkan ke pengguna berdasarkan permintaan pengguna.

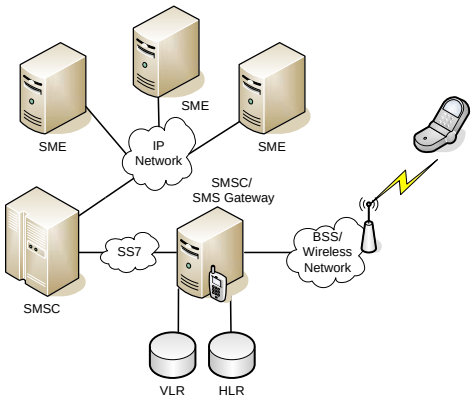
b. Push - Event based, yaitu pesan yang diaktivasi oleh aplikasi berdasarkan kejadian yang berlangsung.

c. Push - Schedule, yaitu pesan yang diaktivasi oleh aplikasi berdasarkan waktu yang telah terjadwal.

d. Push - Personal Profile, yaitu pesan yang diaktivasi oleh aplikasi berdasarkan profile dan preference dari pengguna.

### 2.7.2. Arsitektur dan Elemen Layanan Jaringan SMS

Untuk implementasi layanan SMS, operator menyediakan apa yang disebut SMS Center (SMSC). Secara fisik SMSC dapat berwujud sebuah PC biasa yang mempunyai interkonektivitas dengan jaringan GSM.




**Gambar 2.4** Arsitektur dasar jaringan SMS

### 2.7.3. Elemen dasar jaringan SMS

1. *Short Messaging Entities* (SME), suatu piranti yang dapat menerima dan mengirim pesan pendek.
2. *Short Message Service Center* (SMSC), kombinasi perangkat lunak dan perangkat keras yang brtanggung jawab memperkuat, menyimpan dan meneruskan pesan pendek antar SME dan piranti bergerak (*mobile phone).*
3. *Signaling System 7* (SS7), protokol *signalling* yang umum di gunakan dalam jeringan telepon seluler.
4. *Base Station System* (BSS), berfungsi mengendalikan satu atau lebih BTS dan bertanggung jawab dalam pemberian sumber data dan transmisi sinyal radio elektromagnetis antara MSC dan *mobile phone*.
5. *Home Location Register* (HLR), basis data yang digunakan untuk penyimpanan permanen, pengelolaan dan profil layanan.
6. *Visitor Location Register* (VLR), basis data yang berisi informasi temporal mengenai pelanggan yang berasal dari suatu HLR yang roaming ke HLR lainnya.

### 2.7.4. Elemen Layanan

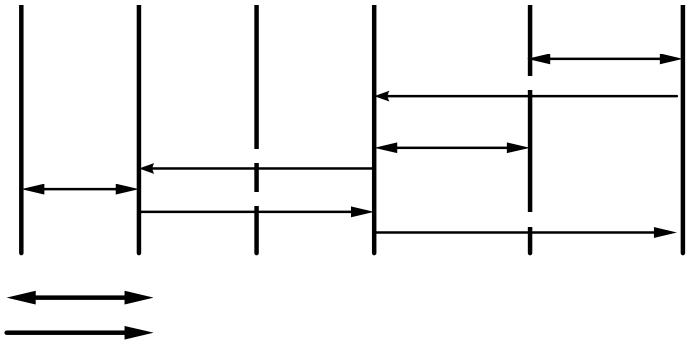
SMS terdiri dari beberapa elemen layanan yang relevan terhadap penerimaan dan pengiriman pesan pendek yaitu:

1. *Message Expiration*, SMSC akan menyimpan dan mencoba mengirimkan kembali pesan yang mengalami kegagalan sampai pengiriman tersebut berhasil.

2. *Priority*, Untuk memberi tanda pesan-pesan yang penting dan membedakannya dari pesan biasa.

Sistem SMS memiliki dua layanan dasar *point-to-point* bagi pelanggan yaitu:

a. *Mobile-Oriented* (MO) *Short Message*, dikirimkan dari *mobile phone* yang MO-*Capable* ke SMSC dan dapat ditujukan ke *mobile phone* lainnya. Pada layanan ini selalu ada laporan yang dikirimkan ke *mobile phone*, baik yang mengkonfirmasikan pengiriman pesan pendek ke SMSC ataupun mengkonfirmasikan kegagalan pengiriman dan mengidentifikasikan penyebabnya.



**Gambar 2.5** Skenario pengiriman MO-SM

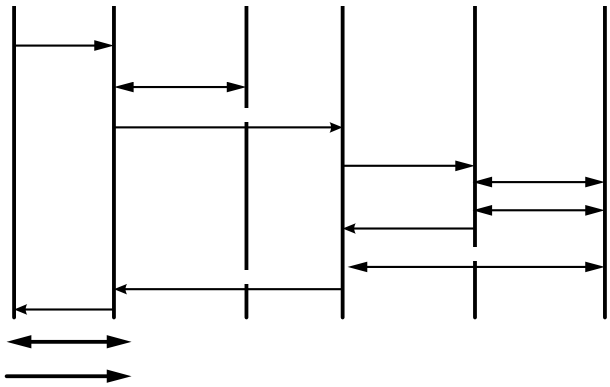
Keterangan :

1. MS mengirimkan SM ke MSC
2. MSC menginterogasi VLR utuk membuktikan bahwa pengiriman pesan tersebut tidak melanggar permintaan layanan pembatasan yang telah ditetapkan.
3. MSC mengirimkan pesan pendek ke SMSC dengan menggunakan operasi *Forward Short Message*.

SMC SMSC H

4. SMSC mengirimkan pesan pendek ke SME
5. SMSC memberitahu MSC mengenai keberhasilan operasi *Forward Short Message.*
6. MSC mengembalikan hasil dari operasi MO-SM ke MS.

b. *Mobile Terminated* (MT) *Short Message*, dikirimkan dari SMSC ke *mobile phone* dan dapat sampai ke SMSC dari *mobile phone* lainnnya melalui MO-SM, pada layanan ini juga terdapat laporan yang diberikan kepada SMSC yang isinya bisa berupa konfirmasi pengiriman pesn pendek ke *mobile phone* maupun informasi kegagalan pengiriman pesan.



**Gambar 2.6** Skenario pengiriman MT-SM

Keterangan :

a. Pesan pendek dikirimkan dari SME ke SMSC.
b. Setelah menyelesaikan pengolahan internalnya, SMSC menginterogasi HLR dan menerima informasi *routing* untuk pelanggan *mobile*.

SMC SMSC

c. SMSC mengirimkan pesan pendek ke MSC dengan menggunakan operasi *Forward Short Message*..
d. MSC mengambil informasi pelanggan dari VLR. Operasi ini dapat melibatkan prosedur autentifikasi.
e. MSC mengirimkan pesan pendek ke MS.
f. MSC mengembalikan hasil dari operasi *Forward Short Message* ke SMSC.
g. Jika diminta oleh SME, SMSC akan mengembalikan laporan status yang mengindikasikan pengiriman pesan pendek.