# BAB III

# PEMBAHASAN

## 3.1. Analisis Sistem

Analisis sistem dilakukan bertujuan untuk memberikan gambaran rancangan, sehingga mempermudah dalam membangun aplikasi media center. Analisis sistem dilakukan dengan beberapa metode seperti observasi, wawancara, studi pustaka, data sampling dan di kerangkai dengan metode pengembangan *prototype*. Hasil analisis sistem dituangkan dalam bentuk rancangan seperti diagram konteks, rancangan tabel, dan rancangan antarmuka.

## 3.2. Analisis Kebutuhan Sistem

Analisis sistem dilakukan bertujuan untuk memberikan gambaran rancangan, sehingga mempermudah dalam membangun aplikasi media center. Analisis sistem dilakukan dengan beberapa metode seperti observasi, wawancara, studi pustaka, data sampling dan di rangkai dengan metode pengembangan *prototype*. Hasil analisis sistem dituangkan dalam bentuk rancangan seperti diagram konteks, rancangan tabel, dan rancangan antarmuka.

## 3.3. Analisis Kebutuhan Non-Fungsional

Analisis non fungsional adalah sebuah tahap dimana seorang pembangun perangkat lunak menganalisis sumber daya yang akan menggunakan perangkat lunak yang

dibangun, sehingga dapat ditentukan kompatibilitas aplikasi yang dibangun terhadap sumber daya yang ada.

Analisis kebutuhan non fungsional dilakukan untuk mengetahui spesifikasi kebutuhan yang dibutuhkan oleh sistem. Spesifikasi kebutuhan melibatkan analisis perangkat keras/ *hardware*, analisis perangkat lunak/ *software*, analisis pengguna/ *user*.

### 3.3.1. Perangkat Keras yang digunakan

Spesifikasi perangkat keras yang digunakan dalam pembangunan aplikasi otomatisasi data atmosfer pada Pusat Sains dan Teknologi Atmosfer adalah tercantum pada Tabel 3.1 adalah sebagai berikut :

**Tabel 3 1 Spesifikasi Kebutuhan Perangkat Keras**

| No | Perangkat Keras | Spesifikasi |
|---|---|---|
| 1 | *Processor* | *Dual Core 2,4 Ghz* |
| 2 | *VGA* | *512 MB* |
| 3 | *RAM* | *2 GB* |
| 4 | *Hard Disk* | *160 GB* |
| 5 | *Operating System* | *Win XP/7* |
| 6 | *LCD* | *24 Inch* |

Penggunaan perangkat keras seperti yang telah disebutkan di atas, sudah cukup untuk mengakses *website* yang akan dibangun

### 3.3.2. Perangkat Lunak yang digunakan

Dalam pembuatan aplikasi ini menggunakan perangkat lunak diantaranya :

1. *PHP* sebagai bahasa pemrograman
2. *Adobe Dream Weaver CS6*
3. *XAMPP Server V 1.7.3*
3. *SQLyog Community 9.63*
4. *Browser Internet Explorer, Mozilla Firefox, Google Chrome.*
5. Sistem Operasi *Windows XP/ 7*

## 3.4. Analisis User

*Website* ini di dikelola oleh pihak instansi yang berada di Pusat sains dan teknologi atmosfer dan ditujukan untuk masyarakat yang ingin mendapatkan informasi mengenai data keatmosferan. Adapun *user* atau pengguna yang terdapat pada *website* ini adalah sebagai berikut :

1. *Administrator*istrator web

   *Administratoristrator web* adalah orang yang bertanggung jawab dalam penyajian informasi dan melakukan *maintenance* pada *website*.
2. Pengunjung atau *user* biasa

   Pengunjung adalah seseorang yang dapat mengakses informasi melalui situs web dengan batasan hak akses *user* biasa.

## 3.5. Analisis Sistem yang Sedang Berjalan

Analisis prosedur atau proses sistem memberikan gambaran tentang sistem yang saat ini sedang berjalan. Analisis sistem bertujuan untuk mengetahui lebih jelas bagaimana cara kerja sistem tersebut, sehingga kelebihan dan kekurangan sistem dapat diketahui.

Setelah melakukan pengamatan pada sistem yang sedang berjalan, dapat diketahui bahwa penyampaian informasi yang ada saat ini masih bersifat *offline* dengan menggunakan media penyimpanan berupa disket, *flash drive*, dan *CD*.

Pihak LAPAN belum menyediakan fasilitas berupa *website* yang bisa di akses oleh masyarakat yang ingin mengetahui informasi keatmosferan. Sehingga masyarakat harus datang langsung ke LAPAN untuk mendapatkan informasi keatmosferan.

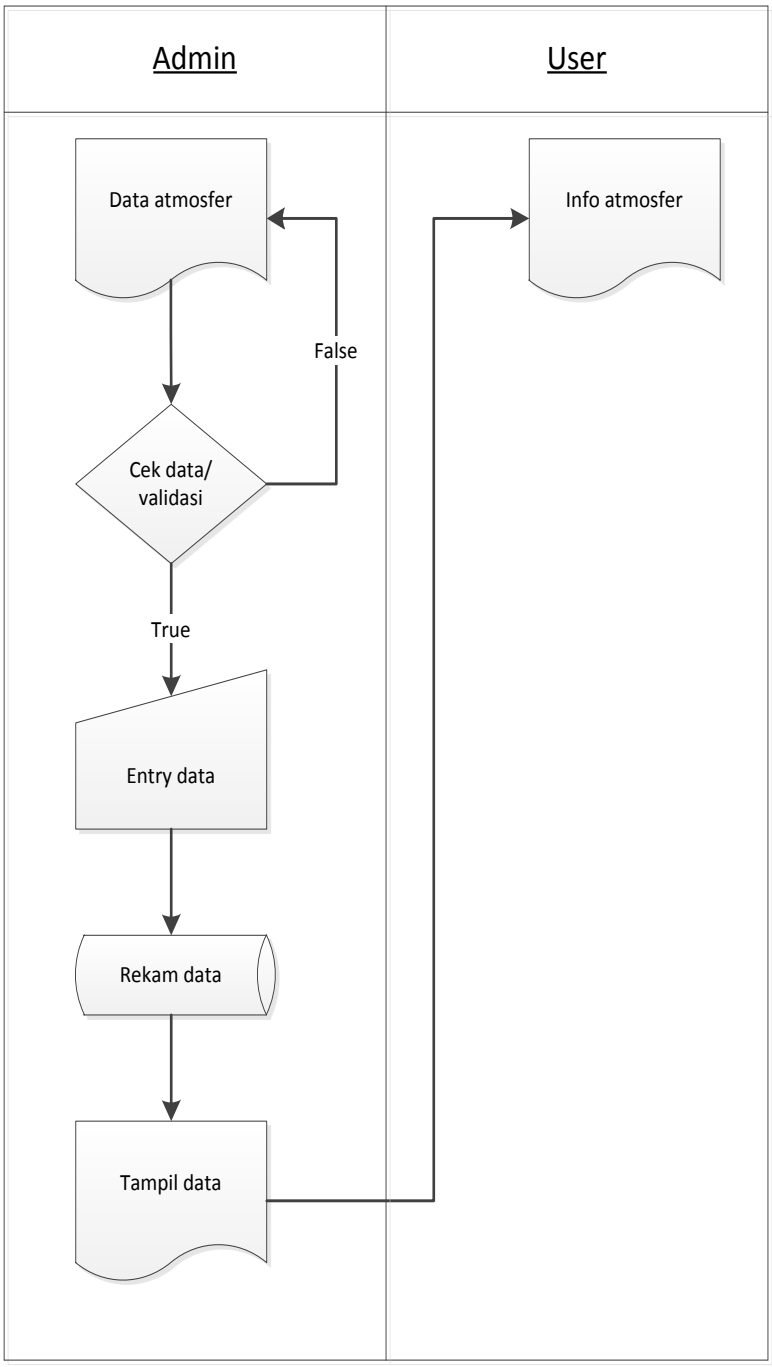
## 3.6. Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional adalah suatu gambaran dari informasi yang berlangsung pada sistem yang bersangkutan. Adapun perancangan aplikasi ini meliputi: *Flowmap, Diagram Context, Data Flow Diagram* (*DFD*) , *Entity Relationship Diagram* (*ERD*), dan spesifikasi proses.

### 3.6.1. *Flowmap*

*Flowmap* adalah penggambaran secara grafik dari langkah - langkah dan urutan-urutan prosedur dari suatu program. *Flowmap* menolong analis dan *programmer* untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif - alternatif lain dalam pengoperasian. *Flowmap* biasanya mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut.

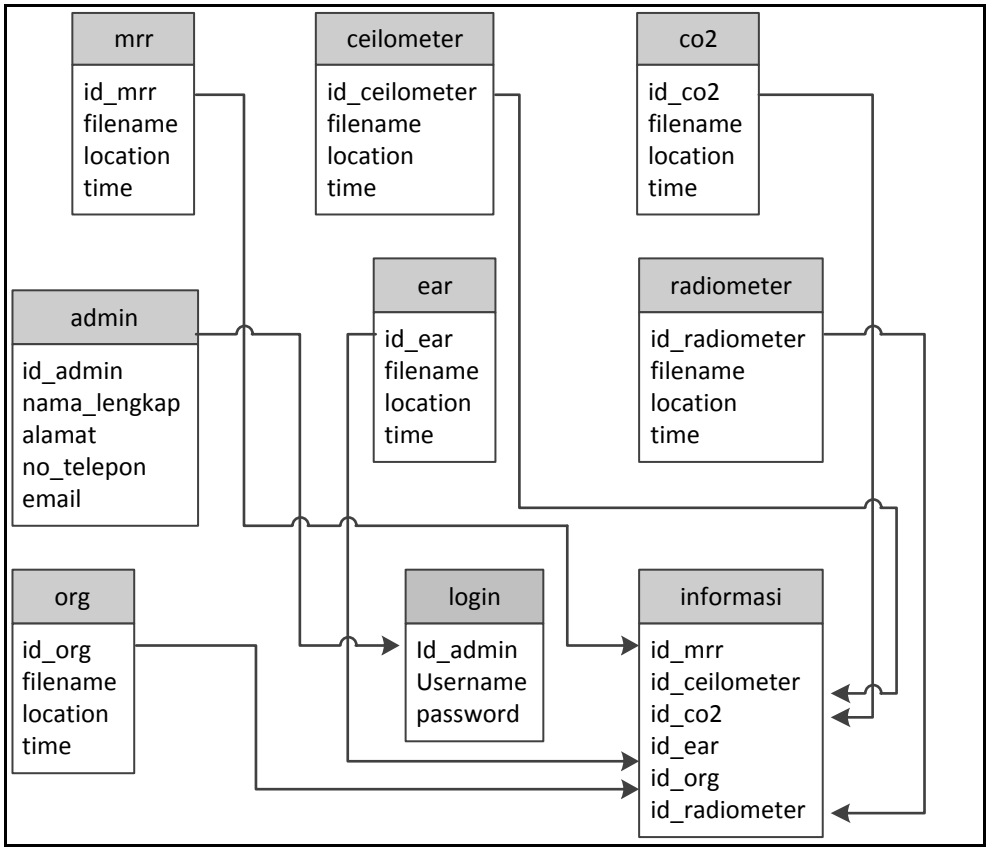
Berikut ini adalah *flowmap* penyampaian informasi:




**Gambar 3 1 *Flowmap* Proses Penyampaian Informasi**

### 3.6.2. Skema Relasi

Skema relasi merupakan rangkaian hubungan antara dua tabel atau lebih pada sistem database. Gambar berikut ini merupakan penjelasan rangkaian *database* pada *website* yang akan dibangun

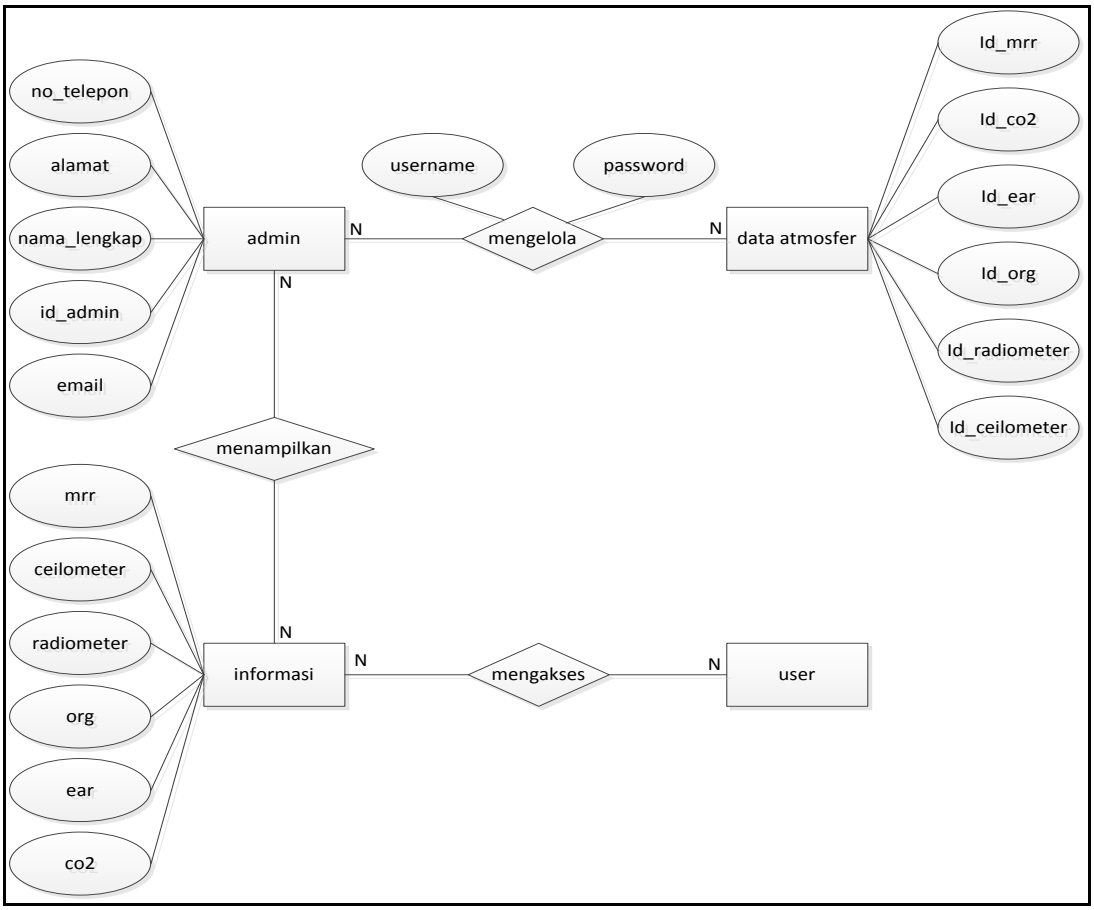



**Gambar 3 2 Skema Relasi *Website* Otomatisasi Data LAPAN**

### 3.6.3. Entity Relationship Diagram (ERD)

*Entity Relationship Diagram* (*ERD*) adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak. Penyusunan basis data (*database*) selalu didahului dengan pekerjaan pemodelan data. Model data E-R (*Entity Relationship*) didasarkan pada persepsi terhadap dunia nyata yang tersusun atas kumpulan objek-objek dasar yang disebut entitas dan relasi.

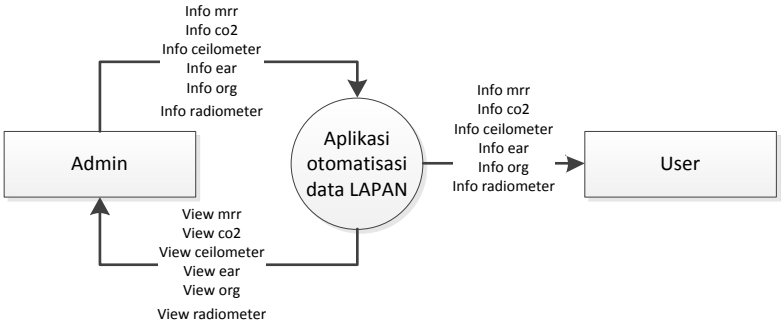
Berikut ini adalah *ERD* otomatisasi data LAPAN:




**Gambar 3 *3 Entity Diagram Relationship* Otomatisasi Data LAPAN**

## 3.7. Diagram Konteks

Diagram konteks merupakan gambaran secara umum mengenai sebuah sistem yang dirancang secara global, yaitu suatu diagram yang mempersentasikan atau mengambarkan hubungan antara sistem dengan luar lingkungan luar sistem yang mempengaruhi operasi sistem. Sistem ditunjukan dalam satu lingkungan yang mengambarkan keseluruhan proses dalam sistem dan hubungannya dengan entitas.

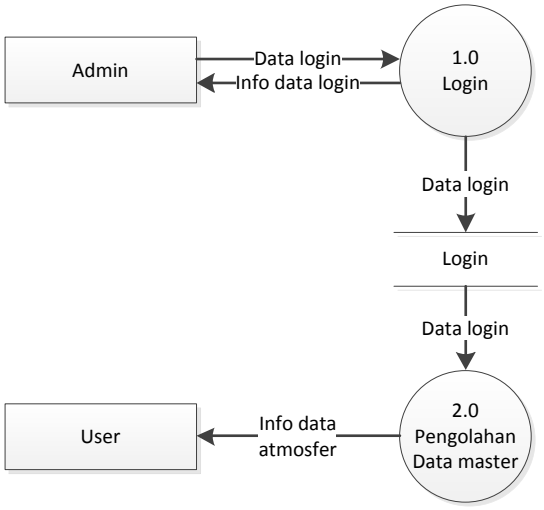



**Gambar 3 4 Diagram Konteks**

## 3.8. *Data Flow Diagram*

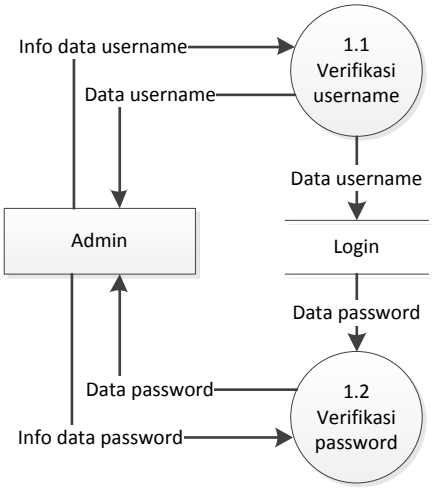
*Data Flow Diagram* (*DFD*) merupakan diagram yang mengunakan notasi-notasi atau symbol-simbol untuk menggambarkan sistem jaringan kerja antara fungsi-fungsi yang berhubungan satu sama lain dengan aliran dan penyimpanan data (Adi Nugroho, 2011).

### 3.8.1. *Data Flow Diagram (DFD) Level* 1




**Gambar 3 5 *DFD Level* 1**

### 3.8.2. *Data Flow Diagram (DFD) Level* 2 Proses *Login*

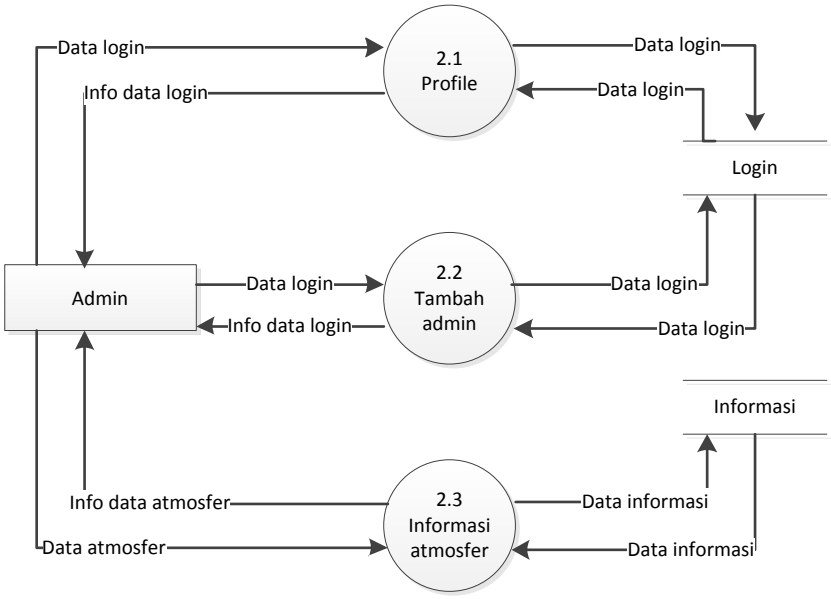



**Gambar 3 6 *DFD Level* 2 Proses *Login***

**Tabel 3 2 Proses Identifikasi 1.1**

| Proses Identifikasi | |
|---|---|
| No. | 1.1 |
| Nama | Verifikasi *username* |
| Tujuan | Memverifikasi *username* |
| Deskripsi | Proses verifikasi *username* |
| Aktor | Sistem |

### 3.8.3. *Data Flow Diagram (DFD) Level* 2 *Administrator*




**Gambar 3 7 *DFD Level* 2 *Administrator***

**Tabel 3 3 Proses Identifikasi 2.1**

| Proses Identifikasi | |
|---|---|
| No. | 2.1 |
| Nama | Proses profile *administrator* |
| Tujuan | *Administrator* memilih profile *administrator* |
| Deskripsi | Proses menampilkan profile *administrator* |
| Aktor | *Administrator* |

**Tabel 3 4 Proses Identifikasi 2.2**

| Proses Identifikasi | |
|---|---|
| No. | 2.2 |
| Nama | Proses create new account |
| Tujuan | *Administrator* Memilih Create New Account |
| Deskripsi | Proses menambah account baru untuk *administrator* |
| Aktor | *Administrator* |

**Tabel 3 5 Proses Identifikasi 2.3**

| Proses Identifikasi | |
|---|---|
| No. | 2.3 |
| Nama | Proses memuat informasi |
| Tujuan | *Administrator* memuat data atmosfer |
| Deskripsi | Proses memuat data atmosfer ke dalam database |
| Aktor | *Administrator* |

## 3.9. Implementasi Database

*Database* dibangun menggunakan *SQLyog Community* yang merekam *Query MySQL* sehingga muncul pada log ketika *database* dibangun.

### 3.9.1. Tabel

Berikut ini adalah *query* pembuatan *database* beserta tabel-tabel yang berada didalamnya :

- *Create Database*

```
CREATE DATABASE `KOTOTABANG`;
```

- *Create Table Administrator*

```
Create Table

CREATE TABLE `admin` (
  `id_admin` int(9) NOT NULL,
  `nama_lengkap` varchar(50) DEFAULT NULL,
  `alamat` varchar(200) DEFAULT NULL,
  `no_telepon` varchar(12) DEFAULT NULL,
  `email` varchar(20) DEFAULT NULL,
  `username` varchar(30) DEFAULT NULL,
  `password` int(200) DEFAULT NULL,
  PRIMARY KEY (`id_admin`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
```

**Gambar 3 8 Proses *Create* Tabel *Administrator***

- *Create Table Login*

```
Create Table

CREATE TABLE `login` (
  `username` varchar(20) NOT NULL,
  `password` varchar(20) default NULL,
  PRIMARY KEY  (`username`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

**Gambar 3 9 Proses *Create* Tabel *Login***

- *Create Table MRR*

```
Create Table

CREATE TABLE `mrr` (
  `id_mrr` int(100) NOT NULL,
  `filename` varchar(255) NOT NULL,
  `location` varchar(255) NOT NULL,
  `time` datetime NOT NULL,
  PRIMARY KEY (`id_mrr`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
```

**Gambar 3 10 Proses *Create* Tabel *MRR***

- *Create Table Ceilometer*

```
Create Table

CREATE TABLE `ceilometer` (
  `id_ceilometer` int(100) NOT NULL,
  `filename` varchar(255) NOT NULL,
  `location` varchar(255) NOT NULL,
  `time` datetime DEFAULT NULL,
  PRIMARY KEY (`id_ceilometer`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
```

**Gambar 3 11 Proses *Create* Tabel *Ceilometer***

- *Create Table* $CO^2$

```
Create Table

CREATE TABLE `co2` (
  `id_co2` int(100) NOT NULL,
  `filename` varchar(255) NOT NULL,
  `location` varchar(255) NOT NULL,
  `time` datetime DEFAULT NULL,
  PRIMARY KEY (`id_co2`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
```

**Gambar 3 12 Proses *Create* Tabel $CO^2$**

- *Create Table EAR*

```
Create Table

CREATE TABLE `ear` (
  `id_ear` int(100) NOT NULL,
  `filename` varchar(255) NOT NULL,
  `location` varchar(255) NOT NULL,
  `time` datetime DEFAULT NULL,
  PRIMARY KEY (`id_ear`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
```

**Gambar 3 13 Proses *Create* Tabel *EAR***

- *Create Table ORG*

```
Create Table

CREATE TABLE `org` (
  `id_org` int(100) NOT NULL,
  `filename` varchar(255) NOT NULL,
  `location` varchar(255) NOT NULL,
  `time` datetime DEFAULT NULL,
  PRIMARY KEY (`id_org`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
```

**Gambar 3 14 Proses *Create* Tabel *ORG***

- *Create Table Radiometer*

```
Create Table

CREATE TABLE `radiometer` (
  `id_radiometer` int(100) NOT NULL,
  `filename` varchar(255) NOT NULL,
  `location` varchar(255) NOT NULL,
  `time` datetime DEFAULT NULL,
  PRIMARY KEY (`id_radiometer`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
```

**Gambar 3 15 Proses *Create* Tabel *Radiometer***

## 3.10. Struktur Tabel

Dalam penggambaran diagram-diagram sebelumnya, disebutkan bahwa sistem memiliki beberapa tabel.

Berikut adalah perancangan struktur tabel dari *website* ini :

- Struktur tabel *administrator*

Di dalam tabel *administrator* berisi *id administrator*, nama lengkap, alamat, nomor telepon, email, *username* dan *password*. Agar lebih jelas dapat dilihat pada table di bawah ini

**Tabel 3 6 Tabel *Administrator***

| | Field | Type | Comment |
|---|---|---|---|
| 🔑 | id_admin | int(9) NOT NULL | |
| | nama_lengkap | varchar(50) NULL | |
| | alamat | varchar(200) NULL | |
| | no_telepon | varchar(12) NULL | |
| | email | varchar(20) NULL | |
| | username | varchar(30) NULL | |
| | password | int(200) NULL | |

- Struktur tabel *login*

**Tabel 3 7 Tabel *Login***

| Create Table |
|---|
| CREATE TABLE `login` (<br>  `username` varchar(20) NOT NULL,<br>  `password` varchar(20) default NULL,<br>  PRIMARY KEY  (`username`)<br>) ENGINE=InnoDB DEFAULT CHARSET=latin1 |

- Struktur tabel *MRR*

**Tabel 3 8 Tabel *MRR***

| | Field | Type | Comment |
|---|---|---|---|
| 🔑 | id_mrr | int(100) NOT NULL | |
| | filename | varchar(255) NOT NULL | |
| | location | varchar(255) NOT NULL | |
| | time | datetime NOT NULL | |

- Struktur tabel *Ceilometer*

**Tabel 3 9 Tabel *Ceilometer***

| | Field | Type | Comment |
|---|---|---|---|
| 🔑 | id_ceilometer | int(100) NOT NULL | |
| | filename | varchar(255) NOT NULL | |
| | location | varchar(255) NOT NULL | |
| | time | datetime NULL | |

- Struktur tabel *$CO^2$*

**Tabel 3 10 Tabel *$CO^2$***

| | Field | Type | Comment |
|---|---|---|---|
| 🔑 | id_co2 | int(100) NOT NULL | |
| | filename | varchar(255) NOT NULL | |
| | location | varchar(255) NOT NULL | |
| | time | datetime NULL | |

- Struktur tabel *EAR*

**Tabel 3 11 Tabel *EAR***

| | Field | Type | Comment |
|---|---|---|---|
| 🔑 | id_ear | int(100) NOT NULL | |
| | filename | varchar(255) NOT NULL | |
| | location | varchar(255) NOT NULL | |
| | time | datetime NULL | |

- Struktur tabel *ORG*

**Tabel 3 12 Tabel *ORG***

| | Field | Type | Comment |
|---|---|---|---|
| 🔑 | id_org | int(100)<br>NOT NULL | |
| | filename | varchar(255)<br>NOT NULL | |
| | location | varchar(255)<br>NOT NULL | |
| | time | datetime<br>NULL | |

- Struktur tabel *Radiometer*

**Tabel 3 13 Tabel *Radiometer***

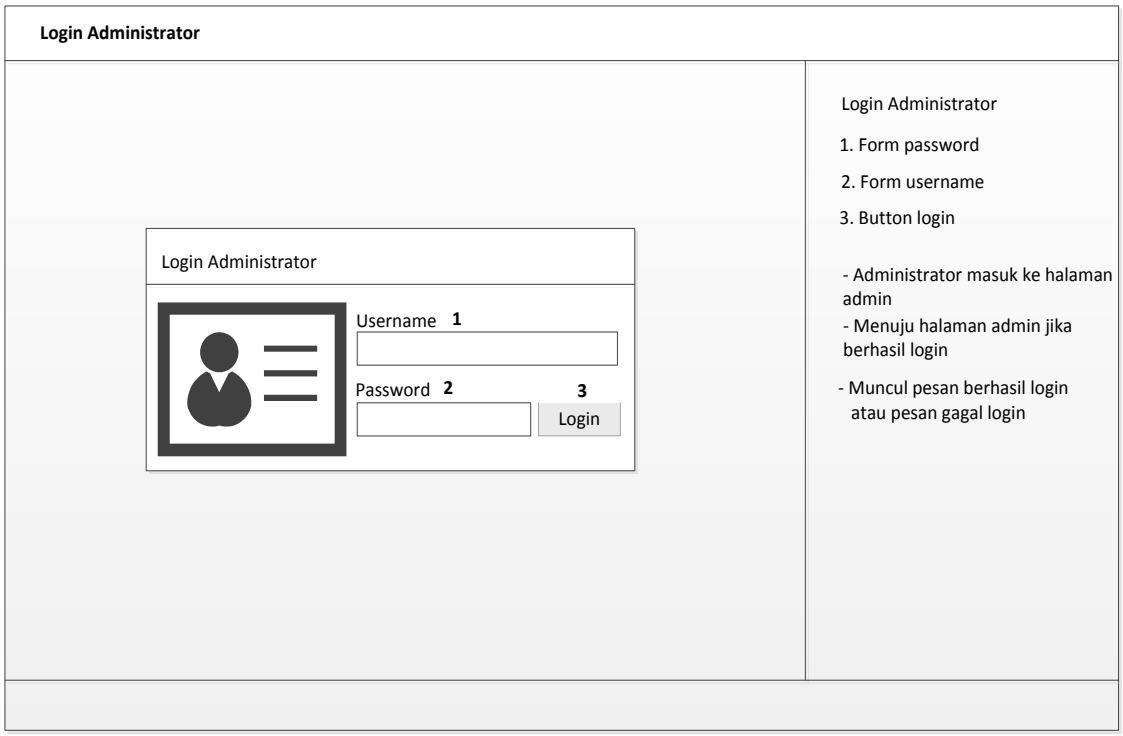
| | Field | Type | Comment |
|---|---|---|---|
| 🔑 | id_radiometer | int(100)<br>NOT NULL | |
| | filename | varchar(255)<br>NOT NULL | |
| | location | varchar(255)<br>NOT NULL | |
| | time | datetime<br>NULL | |

## 3.11. Perancangan Antar Muka

Perancangan Antarmuka digunakan untuk membantu dalam membentuk suatu tampilan yang sesuai dengan yang diharapkan.

- Desain *login*

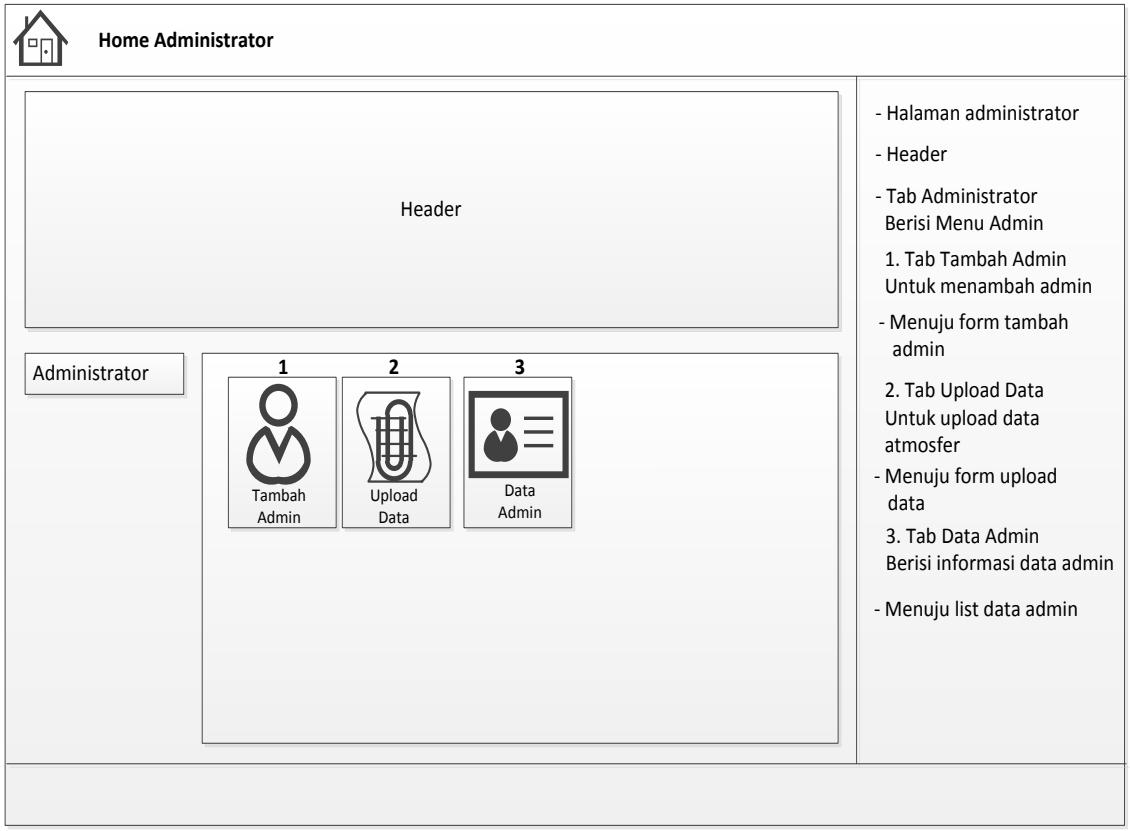
  Perancangan antarmuka pada halaman *login*, *administrator* harus memasukkan username dan password terlebih dahulu agar bias masuk ke halaman *administrator*.




**Gambar 3 16 Rancangan Antarmuka *Login Administrator***

- Desain Halaman *Administrator*istrator

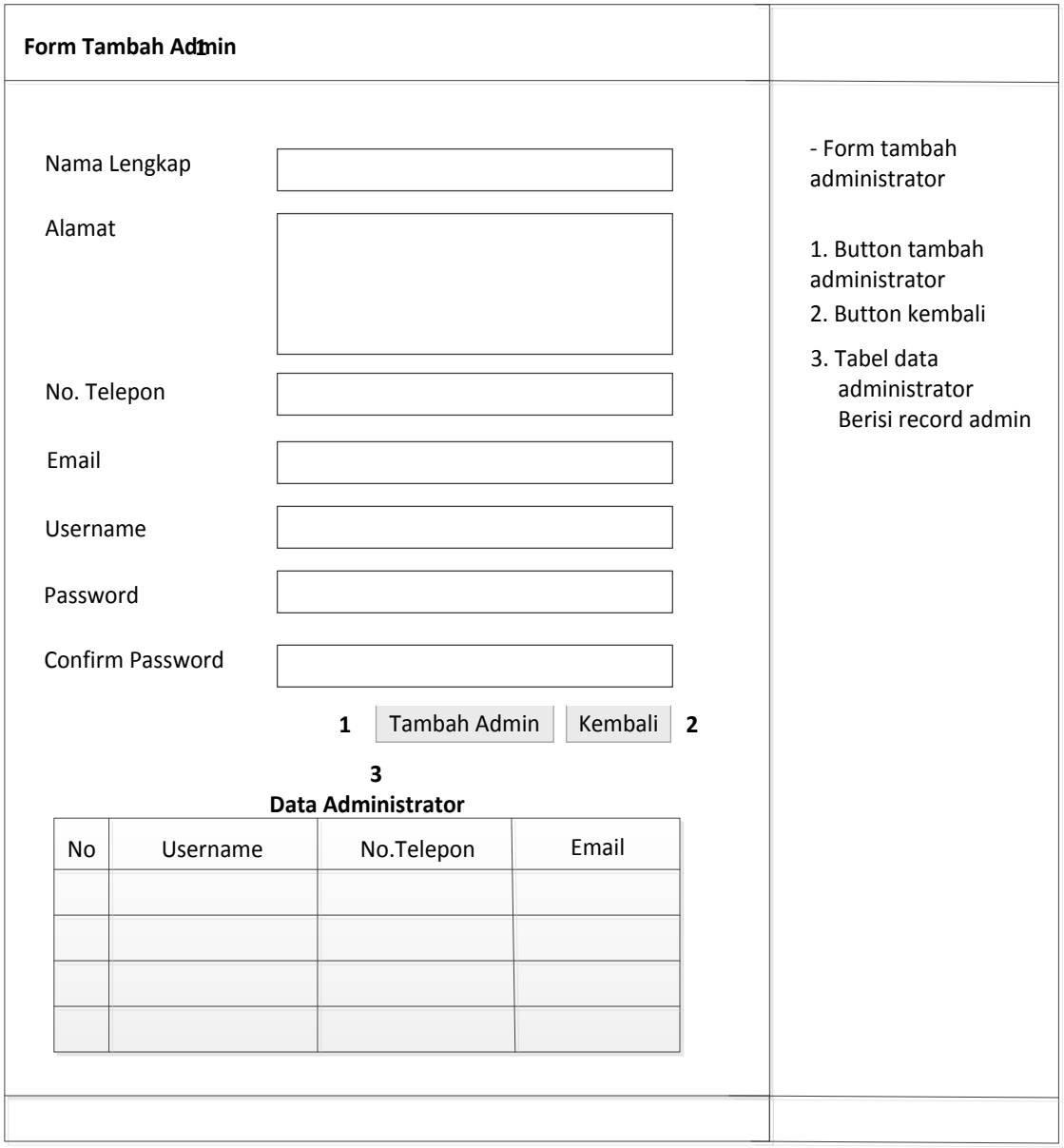
Perancangan antarmuka pada halaman ini, ditujukan untuk *administratoristrator*. Pada halaman ini *administratoristrator* bisa mengelola data *administratoristrator* dan data-data mengenai atmosfer.




**Gambar 3 17 Rancangan Antarmuka Halaman *Administrator***

- Desain tambah *administrator* (Manajemen *Administratoristrator*)

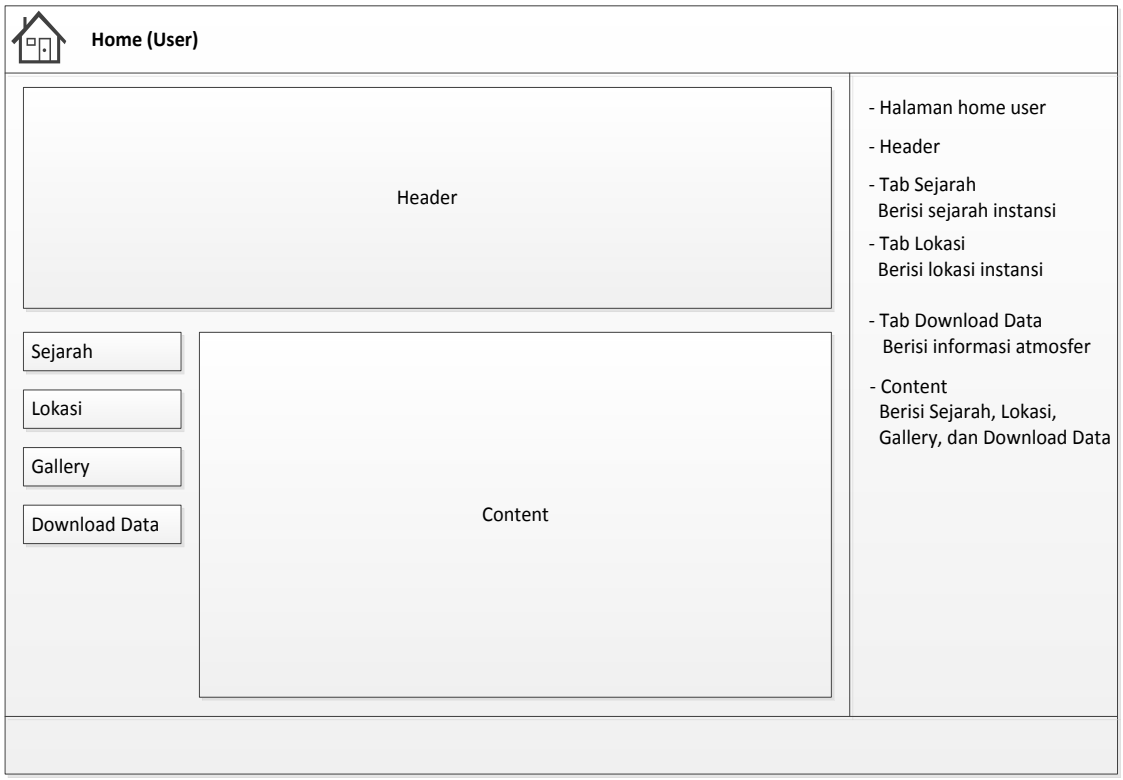
Perancangan antarmuka pada halaman ini, *administrator* bisa menambah account *administrator* dengan mengisi form-form yang telah disediakan. *Administrator* dapat juga melihat data *administrator* yang telah terdaftar sebelumnya




**Gambar 3 18 Rancangan Antarmuka Tambah *Administrator***

- Desain *Home (User)*

Perancangan antarmuka pada halaman home adalah tampilan awal setelah user mengakses *website*




**Gambar 3 19 Rancangan Antarmuka *Home User***

## 3.12. Implementasi

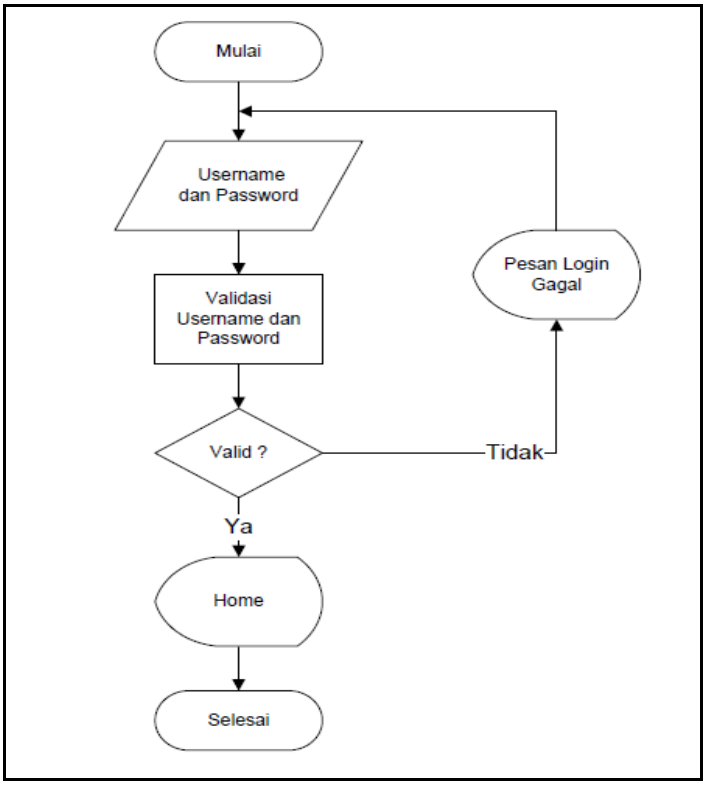
### 3.12.1. Perancangan Prosedural

Perancangan prosedural merupakan perancangan yang dilakukan untuk menetapkan detail algoritma yang akan dinyatakan ke dalam suatu program. Perancangan prosedural menjelaskan tentang bagaimana program aplikasi berjalan. Perancangan prosedural digambarkan dalam bentuk *flowchart*.

Perancangan prosedural pada *website* terdiri dari *flowchart login*, penambahan data, pengubahan data dan tampil data/pencarian data.

- Flowchart Login

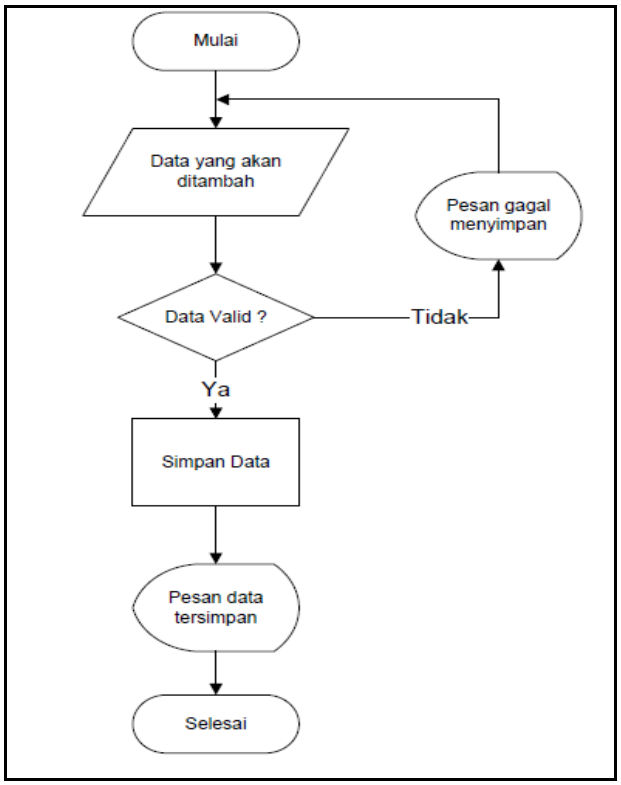
*Flowchart* login menggambarkan proses login dari awal sampai akhir. Berikut ini adalah gambaran dari *flowchart login*




**Gambar 3 20 Prosedur *Login***

- *Flowchart* Penambahan Data

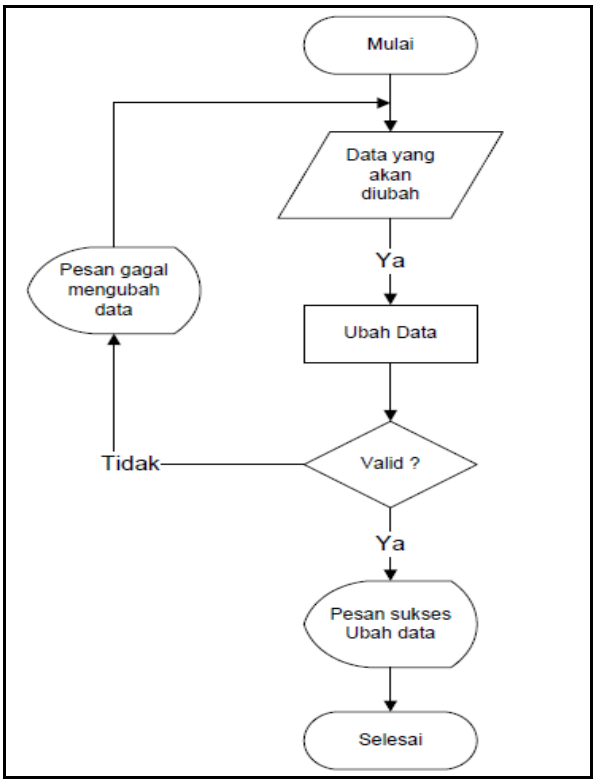
*Flowchart* penambahan data menggambarkan langkah-langkah yang dilakukan oleh pengguna untuk melakukan proses penambahan data pada aplikasi yang dibangun. Adapun gambaran *flowchart* dari proses penambahan data dapat dilihat pada gambar dibawah ini:




**Gambar 3 21 Prosedur Tambah Data**

- Flow Chat Pengubahan Data

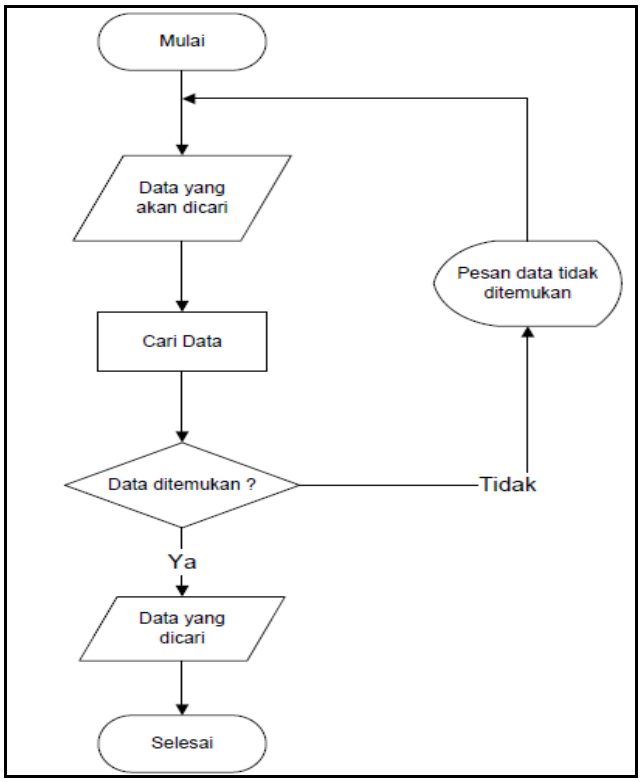
*Flowchart* pengubahan data menggambarkan langkah-langkah yang dilakukan oleh pengguna untuk melakukan penambahan data pada aplikasi yang dibangun. Adapun gambaran *flowchart* dari proses pengubahan data dapat dilihat pada gambar dibawah ini:




**Gambar 3 22 Prosedur Ubah Data**

- Flow Chart Pencarian Data

*Flowchart* Pencarian Data/ Tampil Data menggambarkan langkah-langkah yang dilakukan oleh pengguna untuk menampilkan data pada aplikasi yang dibangun. Adapun gambaran *flowchart* dari proses tampil data dapat dilihat pada gambar dibawah ini:




**Gambar 3 23 Prosedur Cari Data**