

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Peralatan Pendukung**

Penelitian membutuhkan berbagai perangkat untuk mendukung berjalannya perancangan dan implementasi *website*, antara lain:

##### **3.1.1 Perangkat Keras**

Dalam perancangan dan implementasi *website* raport online SMP Negeri 5 Purwanegara, penulis menggunakan perangkat keras berupa *Personal Computer (Laptop)*. Berikut adalah spesifikasi perangkat keras yang digunakan.

- a. 64 bit *architectre processor*
- b. 6 GB *Random Access Memory* (RAM)

##### **3.1.2 Perangkat Lunak**

- a. Sistem Operasi *Windows 8.0*.
- b. Web Editor : *Sublime Text*
- c. *Web Server : Apache*
- d. Bahasa Pemrograman : PHP, HTML, CSS, Javascript
- e. *Database Server : MySQL*
- f. *Web Browser*

## **3.2 Bahan Penelitian**

### **3.2.1 Studi Literatur**

Proses studi literatur dilakukan dengan mempelajari bahan – bahan literatur tentang rekapitulasi nilai dan pembuatan laporan nilai siswa yang didapat melalui *internet* dan buku – buku dari perpustakaan. Sehingga bahan literatur yang telah dipelajari dapat diterapkan untuk membuat sistem pengelolaan nilai raport yang baik. Selain itu, penulis mempelajari berbagai teknologi yang sesuai dengan kebutuhan sistem dan kebutuhan pengguna yang menjadi acuan pembangunan sistem yang lebih baik.

### **3.2.2 Wawancara**

Wawancara dilakukan dengan cara mewawancarai langsung pihak – pihak terkait, yang berguna untuk mendapatkan informasi maupun data – data yang dibutuhkan untuk perancangan dan pembangunan yang akan dibuat yaitu:

1. Wawancara terhadap pihak sekolah SMP Negeri 5 Puwanegara dengan melakukan wawancara kepada Ibu Wartini sebagai kepala sekolah yang berguna untuk mendapatkan data maupun informasi.
2. Wawancara terhadap 2 orang guru dan siswa sebagai pihak pengguna yang berguna untuk menganalisis kebutuhan pengguna terhadap sistem yang dibuat.

Hasil dari wawancara ini berguna untuk mendukung bahasan penelitian ini dalam hal melakukan perancangan dan membangun sistem yang dapat menghasilkan rancangan sistem.

### 3.2.3 Observasi

Tujuan dari observasi adalah dengan mendeskripsikan *setting* yang diamati, tempat kegiatan orang-orang yang berpartisipasi dalam kegiatan tersebut dan makna apa yang diamati menurut perspektif pengamat (Patton, 1990 : 202). Proses observasi didasarkan pada pengamatan langsung dan mencatat perilaku atau kejadian seperti keadaan yang sebenarnya.

Pada metode pengamatan (observasi), ini dilakukan peninjauan dan penelitian langsung di lapangan untuk memperoleh dan mengumpulkan data yang dibutuhkan. Pengamatan ini dilakukan pada tanggal 18 sampai dengan 19 Juli 2016, di SMP Negeri 5 Purwanegara.

## 3.3 Langkah Penelitian

Penelitian tentang Sistem Pengelolaan Nilai dan Raport Online di SMP Negeri 5 Purwanegara menggunakan metode *System Development Life Cycle (SDLC)* model *waterfall*. *SDLC* adalah metodologi umum dalam siklus pengembangan sistem dan penyusunan suatu sistem yang baru untuk menggantikan sistem yang lama atau memperbaiki sistem yang sudah ada.

*SDLC* merupakan pola yang diambil untuk mengembangkan sistem perangkat lunak, yang terdiri dari tahap – tahap : perencanaan sistem (*planning*), analisa (*analysis*), desain (*design*), implementasi

(*implementation*), pengujian (*testing*) dan pengelolaan (*maintenance*). Dalam rekayasa perangkat lunak, konsep *SDLC* mendasari berbagai jenis metodologi pengembangan perangkat lunak.

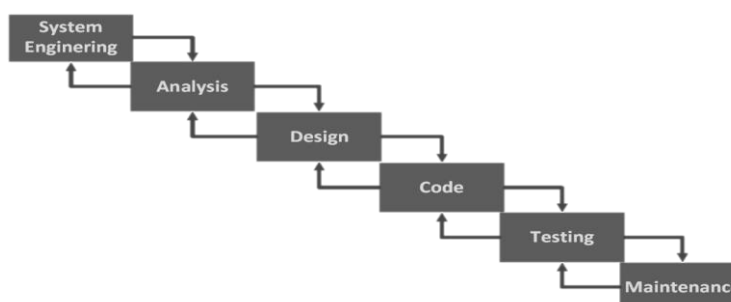
Fase – fase pada *System Development Life Cycle (SDLC)* meliputi:

1. Perencanaan Sistem (*Systems Planning*).
2. Analisis Sistem (*System Analysis*).
3. Perancangan Sistem (*Systems Design*).
4. Implementasi Sistem (*Systems Maintenance*).

Model *Waterfall* merupakan model klasik yang sederhana dengan aliran sistem yang linier. Output dari setiap tahap merupakan input bagi tahap berikutnya. Model ini pertama kali diperkenalkan oleh *Winston Royce* tahun 1970, sekarang model ini lebih dikenal dengan *Linear Sequential Model*. Karakteristik dari metodologi *waterfall* ini meliputi beberapa bagian, yaitu:

- Aktivitas mengalir dari satu fase ke fase lainnya secara berurutan.
- Setiap fase dikerjakan terlebih dahulu sampai selesai, jika sudah selesai baru mulai menuju fase berikutnya.

Tahapan model *Waterfall* dapat digambarkan pada gambar 3.1 di bawah ini.



**Gambar 3.1** Metode SDLC model *Waterfall*

Penjelasan mengenai tahapan metode SDLC model *Waterfall* adalah sebagai berikut.

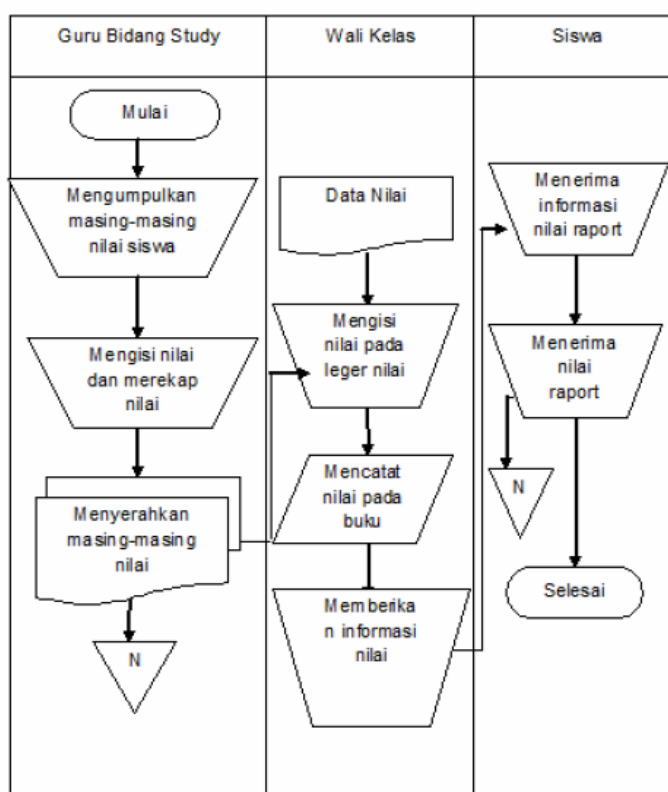
1. *System Engineering* merupakan tahapan untuk menetapkan segala hal yang diperlukan dalam pelaksanaan proyek.
2. *Analysis* merupakan tahapan untuk menganalisis hal-hal yang diperlukan untuk pembuatan atau pengembangan perangkat lunak dan melakukan pengumpulan data serta informasi yang diperlukan melalui observasi (pengamatan) dan wawancara.
3. *Design* merupakan tahap penerjemahan dari keperluan atau data yang telah dianalisis ke dalam bentuk yang mudah dimengerti oleh programmer. Tiga atribut yang penting dalam proses perancangan yaitu : struktur data, arsitektur perangkat lunak dan prosedur rinci / algoritma.
4. *Code* merupakan tahapan untuk menerjemahkan data yang telah dirancang / algoritma ke dalam bahasa pemrograman yang telah ditentukan.
5. *Testing* merupakan tahapan uji coba terhadap program telah dibuat.
6. *Maintenance* merupakan perubahan atau penambahan program sesuai dengan permintaan user.

### **3.4 Sistem yang Sedang Berjalan di SMP Negeri 5 Purwanegara**

Di dalam sistem pengelolaan nilai pada SMP Negeri 5 Purwanegara terdapat beberapa kendala dalam proses pencatatan dan pelaporan data nilai. Dikarenakan setiap siswa memiliki nilai sendiri – sendiri dan berbeda-beda

setiap semesternya. Proses pencarian data nilai siswa yang memerlukan cukup banyak waktu dan tenaga sehingga kinerja dari sistem menjadi kurang efektif dan menyita waktu yang sangat lama. Kurang efektifnya pengolahan nilai sehingga masih terjadi adanya kesalahan dalam memberikan informasi nilai bagi siswa maupun wali siswa.

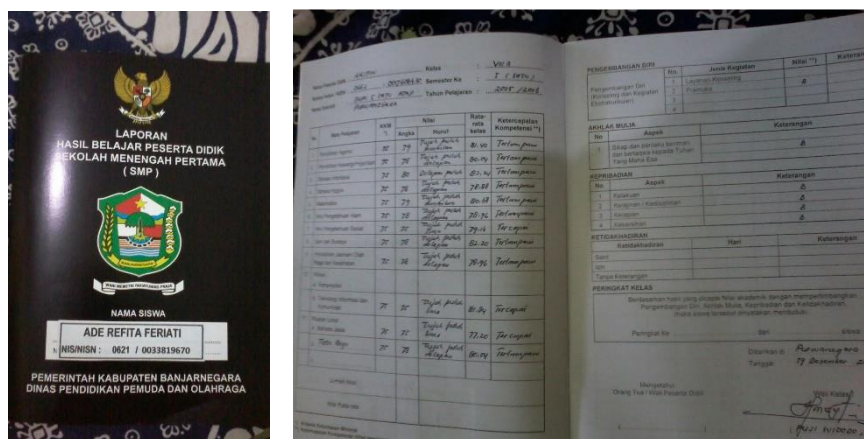
Sistem yang berjalan di sekolah saat ini dapat digambarkan pada gambar 3.2.



**Gambar 3.2** Sistem yang berjalan

Perlu adanya sistem terkomputerisasi untuk mengoptimalkan pengolahan data nilai sehingga dihasilkan informasi yang cepat, relevan, juga waktu pengaksesan cepat dan akurat Saat ini sistem informasi yang digunakan dalam pengolahan data nilai siswa masih bersifat konvensional,

sehingga kurang optimal dan membutuhkan waktu yang lama dalam pencatatan, sehingga selalu terhambat dalam pengumpulan nilai. Sistem yang berjalan saat ini kurang efektif dan efisien dimana masih manual cara pengolahan nilainya masih dituliskan ke dalam buku raport (Gambar 3.3) sehingga prosesnya sangat lama.



**Gambar 3.3** Raport Siswa SMP Negeri 5 Purwanegara

### 3.5 Analisis Kebutuhan Fungsional dan Non Fungsional

#### 3.5.1 Kebutuhan Fungsional

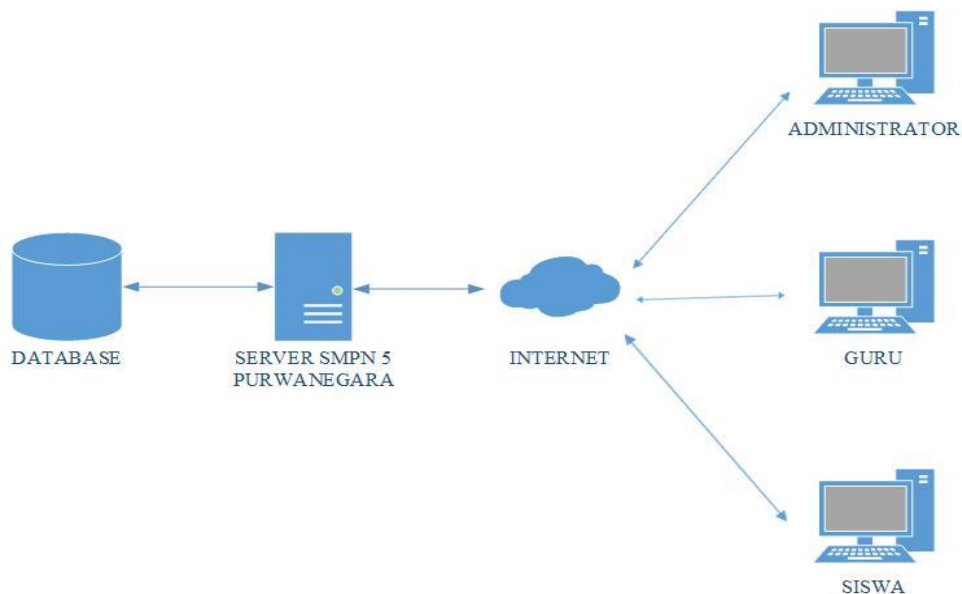
1. Data Siswa
2. Data Mata Pelajaran
3. Data Guru
4. Data Nilai
5. Laporan Nilai Siswa

#### 3.5.2 Kebutuhan Non Fungsional

1. Laporan nilai siswa ditampilkan hanya kepada siswa yang bersangkutan.
2. Sistem dilengkapi dengan hak akses.

### 3.6 Arsitektur Sistem

Pada tahap ini dilakukan penentuan perancangan sistem yang akan digunakan. Berikut ini merupakan perancangan sistem yang diusulkan:



**Gambar 3.4** Arsitektur Sistem yang Diusulkan

Gambar 3.4 merupakan rancangan arsitektur sistem yang diusulkan untuk membuat *website* Raport Online SMP Negeri 5 Purwanegara. *Database server* yang digunakan pada aplikasi *MySQL* dan menggunakan *Apache* sebagai *web server*. Komunikasi antara pengguna dan *web server* menggunakan *internet* dan *web browser* pada perangkat pengguna. Saat pengguna mengakses aplikasi, *web server* memuat antarmuka dan melakukan pengambilan data yang diperlukan dari *database server*. Melalui antarmuka yang dimuat *web server* pengguna bisa menyimpan data ke *database server*.

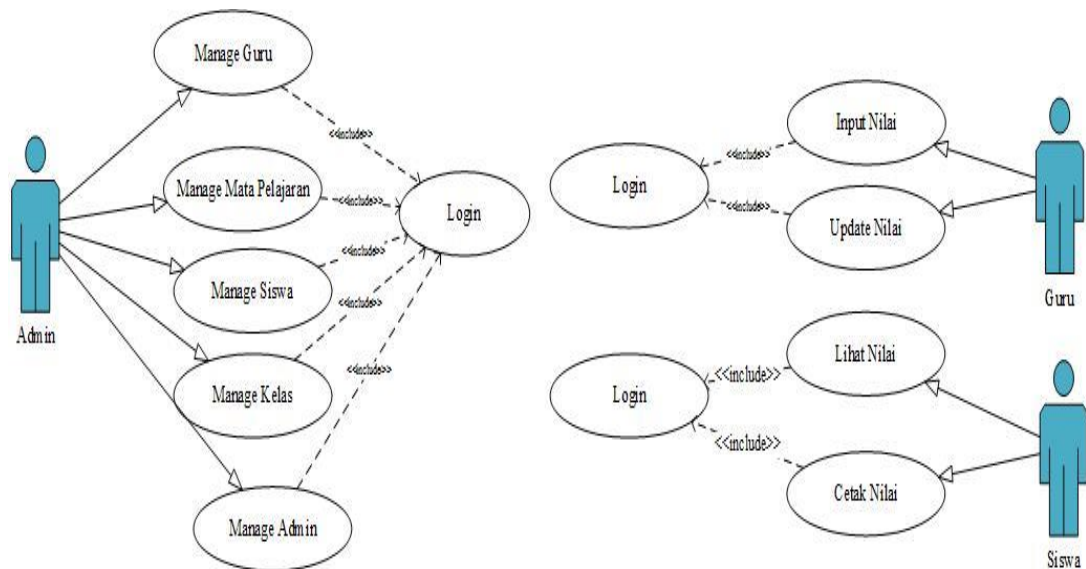


### 3.7 Perancangan Sistem

Dalam pembuatan aplikasi dilakukan perancangan *database* menggunakan bantuan *Diagram ER*. Metode perangan lain yang digunakan dalam aplikasi adalah *United Markup Language (UML)* yang dipakai dalam pengembangan aplikasi yaitu model *Use Case Diagram*, *Activity Diagram*, dan *Class Diagram*.

#### 3.7.1 Use Case Diagram Aplikasi

Gambaran *Use Case Diagram* yang digunakan dalam aplikasi dapat dilihat pada Gambar 3.5.



**Gambar 3.5** Use Case Diagram Aplikasi

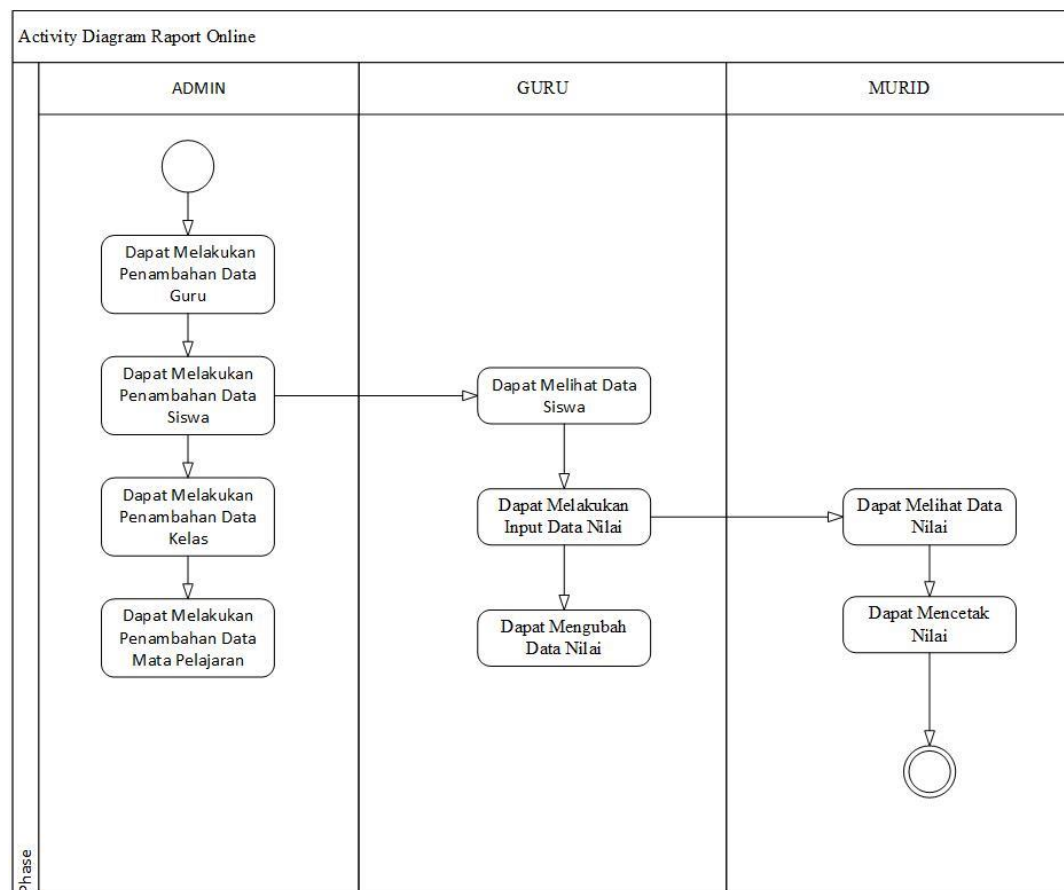
Berikut penjelasan tentang Gambar 3.5:

1. Terdapat 3 aktor pada *use case diagram* aplikasi yakni admin, guru dan siswa.

2. Admin berhubungan langsung pada *manage* data guru, data mata pelajaran, data siswa, data kelas dan data admin. Admin berhak menambah, mengurangi ataupun merubah data di dalamnya.
3. Guru dapat melakukan *input* nilai dan merubahnya sesuai dengan nilai siswa yang bersangkutan.
4. Siswa dapat melihat dan mencetak nilai sesuai dengan NIS masing – masing.

### 3.7.2 Activity Diagram Aplikasi

Gambar *Activity Diagram* yang digunakan dalam aplikasi dapat dilihat pada Gambar 3.6.



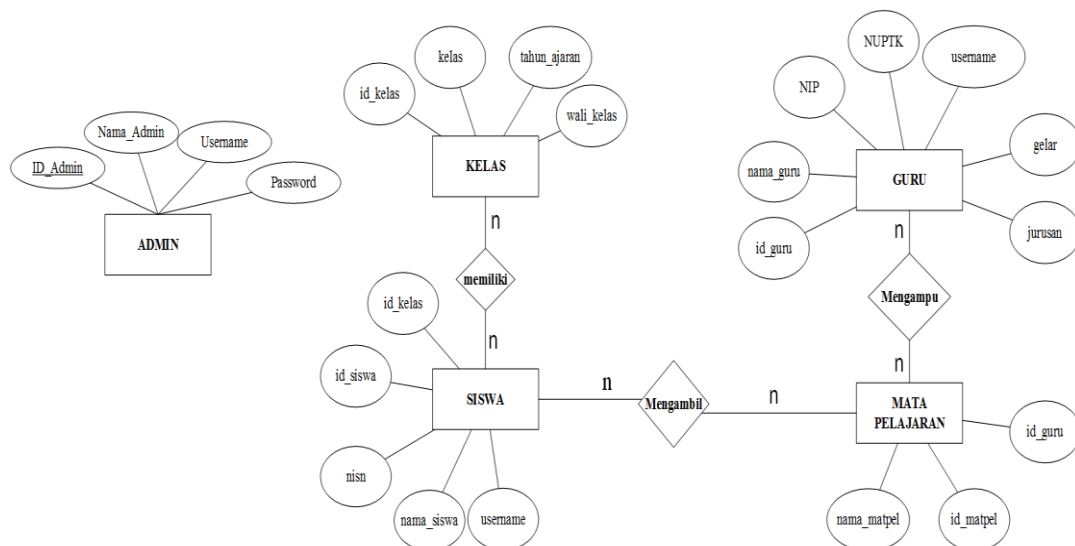
**Gambar 3.6** Activity Diagram Aplikasi

Gambar 3.6 menunjukkan *Activity Diagram* pada kegiatan alur proses raport online. Admin dapat melakukan penambahan data guru, data siswa, data kelas maupun data mata pelajaran. Selanjutnya guru dapat melihat data siswa dan melakukan *input nilai* dan mengubahnya sesuai dengan nilai siswa yang terbaru. Kemudian siswa dapat melihat dan mencetak nilai sesuai dengan NIS masing – masing.

### 3.8 Perancangan Basis Data

#### 3.8.1 ER Diagram Aplikasi

Gambar 3.7 berikut ini menjelaskan hubungan data dalam basis data yang terdapat di dalam sistem Raport Online SMP Negeri 5 Purwanegara:



**Gambar 3.7** ER Diagram Aplikasi

Berikut penjelasan tentang Gambar 3.7:

1. Terdapat 5 entitas yang masing – masing memiliki atribut dan relasi.

2. Entitas Kelas memiliki relasi *many-to-many* ke entitas Siswa, artinya sebuah kelas memiliki banyak siswa di dalamnya, dan sebaliknya Siswa mempunyai banyak kelas.
3. Entitas Guru memiliki relasi *many-to-many* ke entitas Mata Pelajaran, artinya seorang guru dapat mengampu banyak mata pelajaran, dan sebaliknya sebuah mata pelajaran dapat diampu oleh banyak guru.
4. Entitas Siswa memiliki relasi *many-to-many* ke entitas Mata Pelajaran, artinya siswa dapat mengambil banyak mata pelajaran, dan sebaliknya mata pelajaran dapat diambil oleh banyak siswa.
5. Entitas Admin tidak memiliki relasi ke entitas lainnya.

### 3.8.2 Kamus Data

Kamus data merupakan daftar rancangan basis data dan tabel yang akan dibuat. Dari sebuah *ER Diagram* dapat dikonversi menjadi tabel – tabel pada kamus data dengan lebih lengkap dan terperinci. Pada *ER Diagram* yang digambarkan pada gambar 3.7, terdapat entitas Admin, Kelas, Siswa, Mata Pelajaran dan Guru. Masing – masing entitas dapat dikonversikan menjadi sebuah tabel pada kamus data.

Berikut ini merupakan kamus data dalam basis data yang diperlukan dalam perancangan sistem:

1. Tabel Admin

Tabel Admin terbentuk dari entitas Admin pada *ER Diagram*. Tabel Admin mempunyai beberapa kolom. Masing –

masing kolom mempunyai fungsi yang akan dijelaskan pada tabel di bawah ini.

**Tabel 3. 1** Kamus Data Tabel Admin

| No. | Nama     | Tipe Data | Panjang | Keterangan  |
|-----|----------|-----------|---------|---|
| 1.  | Id       | Int       | -       | Kolom yang berfungsi sebagai <i>primary key</i> pada tabel admin. |
| 2.  | Nama     | Varchar   | 50      | Kolom yang menunjukkan nama admin.                                |
| 3.  | Username | Varchar   | 10      | Kolom yang menunjukkan identitas admin saat <i>login</i> .        |
| 4.  | Password | Varchar   | 8       | Kolom yang menunjukkan password saat admin <i>login</i> .         |

## 2. Tabel Kelas

Tabel Kelas terbentuk dari entitas Kelas pada *ER Diagram*.

Tabel Kelas mempunyai beberapa kolom. Masing – masing kolom mempunyai fungsi yang akan dijelaskan pada tabel di bawah ini.

**Tabel 3.2** Kamus Data Tabel Kelas

| No. | Nama     | Tipe Data | Panjang | Keterangan  |
|-----|----------|-----------|---------|---|
| 1.  | Id_kelas | Int       | -       | Kolom yang berfungsi sebagai <i>primary key</i> pada tabel Kelas. |
| 2.  | Kelas    | Varchar   | 10      | Kolom yang menunjukkan nama kelas.                                |
| 3.  | Tahun    | Varchar   | 30      | Kolom yang menunjukkan tahun ajaran.                              |
| 4.  | Wk       | Varchar   | 20      | Kolom yang menunjukkan nama wali kelas.                           |

### 3. Tabel Siswa

Tabel Siswa terbentuk dari entitas Siswa pada *ER Diagram*.

Tabel Siswa mempunyai beberapa kolom. Masing – masing kolom mempunyai fungsi yang akan dijelaskan pada tabel di bawah ini.

Tabel 3.3 Kamus Data Tabel Siswa

| No. | Nama     | Tipe Data | Panjang | Keterangan  |
|-----|----------|-----------|---------|---|
| 1.  | Id       | Int       | -       | Kolom yang berfungsi sebagai <i>primary key</i> pada tabel Siswa.                   |
| 2.  | Nisn     | Varchar   | 10      | Kolom yang menunjukkan NISN (Nomor Induk Siswa Nasional)                            |
| 3.  | Nama     | Varchar   | 50      | Kolom yang menunjukkan nama siswa.  |
| 4.  | Id_kelas | Int       | -       | Kolom yang menunjukkan id_kelas yang merupakan <i>foreign key</i> dari tabel Kelas. |
| 5.  | Username | Varchar   | 10      | Kolom yang menunjukkan identitas siswa saat <i>login</i> .                          |

|    |          |         |    |  |
|----|----------|---------|----|--|
| 6. | Password | Varchar | 8  | Kolom yang menunjukkan <i>password</i> siswa saat <i>login</i> . |
| 7. | Ortu     | Varchar | 50 | Kolom yang menunjukkan nama orang tua siswa.                     |

#### 4. Tabel Guru

Tabel Guru terbentuk dari entitas Guru pada *ER Diagram*.

Tabel Guru mempunyai beberapa kolom. Masing – masing kolom mempunyai fungsi yang akan dijelaskan pada tabel di bawah ini.

**Tabel 3.4** Kamus Data Tabel Guru

| No. | Nama    | Tipe Data | Panjang | Keterangan   |
|-----|---------|-----------|---------|--|
| 1.  | Id_guru | Int       | -       | Kolom yang berfungsi sebagai <i>primary key</i> pada tabel Guru. |
| 2.  | Nama    | Varchar   | 50      | Kolom yang menunjukkan nama guru.                                |
| 3.  | Nip     | Varchar   | 18      | Kolom yang menunjukkan NIP                                       |



|    |           |         |    |   |
|----|-----------|---------|----|---|
|    |           |         |    | (Nomor Induk Pegawai) guru.                               |
| 4. | Nuptk     | Varchar | 16 | Kolom yang menunjukkan NUPTK guru.                        |
| 5. | Username  | Varchar | 10 | Kolom yang menunjukkan identitas guru saat <i>login</i> . |
| 6. | Password  | Varchar | 8  | Kolom yang menunjukkan <i>password</i> guru saat login.   |
| 7. | Status_wk | Varchar | 1  | Kolom yang menunjukkan status wali kelas.                 |
| 8. | Gelar     | Varchar | 7  | Kolom yang menunjukkan gelar guru.                        |

#### 5. Tabel Mata Pelajaran

Tabel Mata Pelajaran terbentuk dari entitas Mata Pelajaran pada *ER Diagram*. Tabel Mata Pelajaran mempunyai beberapa

kolom. Masing – masing kolom mempunyai fungsi yang akan dijelaskan pada tabel di bawah ini.

**Tabel 3.5** Kamus Data Tabel Mata Pelajaran

| No. | Nama    | Tipe Data | Panjang | Keterangan   |
|-----|---------|-----------|---------|--|
| 1.  | Id      | Int       | -       | Kolom yang berfungsi sebagai <i>primary key</i> pada tabel Mata Pelajaran.               |
| 2.  | Id_guru | Int       | -       | Kolom yang menunjukkan identitas guru yang merupakan <i>foreign key</i> dari tabel Guru. |
| 3.  | Matpel  | Varchar   | 20      | Kolom yang menunjukkan nama mata pelajaran.  |

#### 6. Tabel Nilai

Tabel Nilai terbentuk dari hasil relasi *many-to-many* antara entitas Siswa dengan entitas Mata Pelajaran pada *ER Diagram*. Tabel Nilai mempunyai beberapa kolom. Masing – masing kolom mempunyai fungsi yang akan dijelaskan pada tabel di bawah ini.

**Tabel 3.6** Kamus Data Tabel Nilai

| No. | Nama     | Tipe Data | Panjang | Keterangan  |
|-----|----------|-----------|---------|---|
| 1.  | Id       | Int       | -       | Kolom yang berfungsi sebagai <i>primary key</i> pada tabel Nilai.                   |
| 2.  | Id_siswa | Varchar   | 10      | Kolom yang menunjukkan id_siswa yang merupakan <i>foreign key</i> dari tabel Siswa. |
| 3.  | Matpel   | Varchar   | 20      | Kolom yang menunjukkan nama mata pelajaran.   |
| 4.  | Semester | Int       | -       | Kolom yang menunjukkan semester yang ditempuh.                                      |
| 5.  | Nilai    | Int       | -       | Kolom yang menunjukkan nilai siswa tiap mata pelajaran.                             |

|    |              |         |   |  |
|----|--------------|---------|---|--|
| 6. | KKM          | Int     | - | Kolom yang menunjukkan batas minimal nilai yang harus dicapai. |
| 7. | Ketercapaian | Varchar | 8 | Kolom yang menunjukkan status ketercapaian siswa.              |

#### 7. Tabel Guru Matpel

Tabel Guru Matpel terbentuk dari hasil relasi *many-to-many* antara entitas Guru dengan entitas Mata Pelajaran pada *ER Diagram*. Tabel Guru Matpel mempunyai beberapa kolom. Masing – masing kolom mempunyai fungsi yang akan dijelaskan pada tabel di bawah ini.

**Tabel 3.7** Kamus Data Tabel Guru Matpel

| No. | Nama    | Tipe Data | Panjang | Keterangan  |
|-----|---------|-----------|---------|---|
| 1.  | Id      | Int       | -       | Kolom yang berfungsi sebagai <i>primary key</i> pada tabel Guru Matpel. |
| 2.  | Id_guru | Int       | -       | Kolom yang menunjukkan id_guru  |

|    |        |         |    |  |
|----|--------|---------|----|--|
|    |        |         |    | yang merupakan <i>foreign key</i> dari tabel Guru. |
| 3. | Matpel | Varchar | 50 | Kolom yang menunjukkan nama mata pelajaran.        |
| 4. | Kkm    | Int     | -  | Kolom yang menunjukkan KKM ().                     |

#### 8. Tabel Siswa Per Kelas

Tabel Guru Matpel terbentuk dari hasil relasi *many-to-many* antara entitas Siswa dengan entitas Kelas pada *ER Diagram*. Tabel Guru Matpel mempunyai beberapa kolom. Masing – masing kolom mempunyai fungsi yang akan dijelaskan pada tabel di bawah ini.

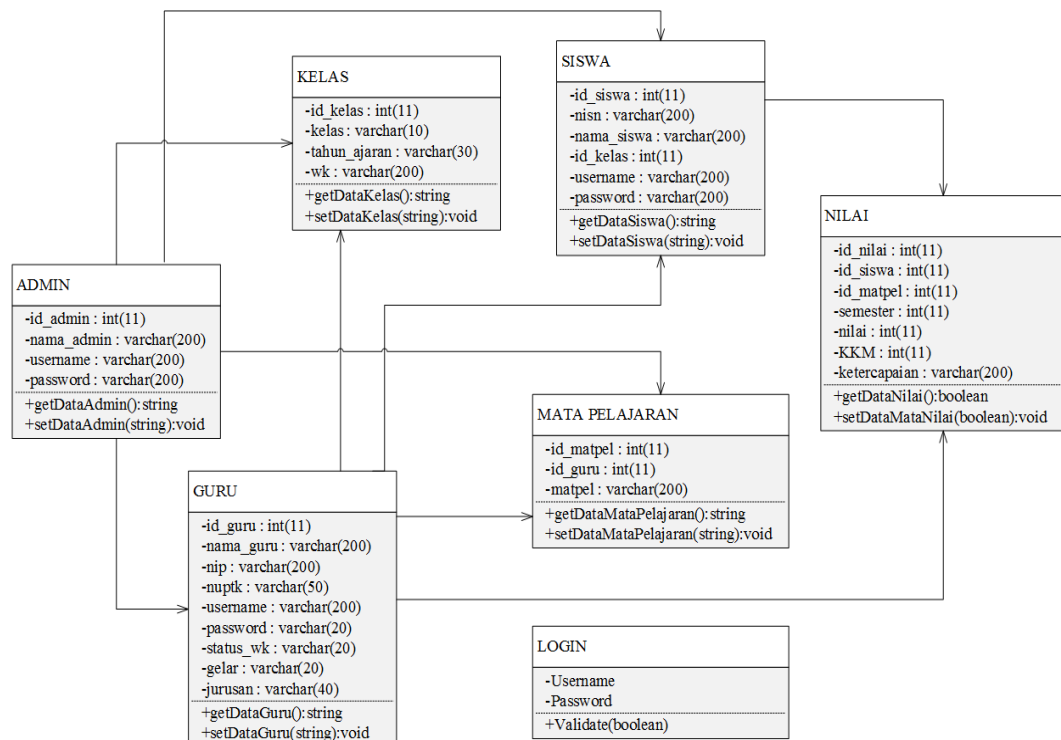
**Tabel 3. 8** Kamus Data Tabel Siswa Per Kelas

| No. | Nama | Tipe Data | Panjang | Keterangan  |
|-----|------|-----------|---------|---|
| 1.  | Id   | Int       | -       | Kolom yang berfungsi sebagai <i>primary key</i> pada tabel Siswa Per Kelas. |

|    |          |     |   |   |
|----|----------|-----|---|---|
| 2. | Id_siswa | Int | - | Kolom yang menunjukkan id_siswa yang merupakan <i>foreign key</i> dari tabel Siswa. |
| 3. | Id_kelas | Int | - | Kolom yang menunjukkan id_kelas yang merupakan <i>foreign key</i> dari tabel Kelas. |

### 3.8.3 Class Diagram Aplikasi

Gambaran *Class Diagram* yang digunakan dalam aplikasi dapat dilihat pada Gambar 3.8.



Gambar 3.8 *Class Diagram* Aplikasi

Berikut penjelasan fungsi dari masing masing kelas :

1. *Class Admin*

*Class Admin* berfungsi untuk memasukan dan menampilkan data Admin.

2. *Class Kelas*

*Class Kelas* berfungsi untuk memasukkan dan menampilkan data Kelas. *Class Kelas* juga memiliki fungsi sebagai penghubung *class* lain yang akan menampilkan nama Kelas.

3. *Class Guru*

*Class Guru* berfungsi untuk memasukkan dan menampilkan data Guru. *Class Kelas* juga memiliki fungsi sebagai penghubung *class* lain yang akan menampilkan nama Guru.

4. *Class Siswa*

*Class Siswa* berfungsi untuk memasukan data siswa. *Class Siswa* sangat berperan penting pada aplikasi. *Class* ini berfungsi sebagai sumber data siswa yang akan dihubungkan pada *class* lain yang menampilkan nama siswa.

5. *Class Mata Pelajaran*

*Class Mata Pelajaran* berfungsi untuk memasukkan dan menampilkan data Mata Pelajaran. *Class Mata Pelajaran* juga memiliki fungsi sebagai penghubung *class* lain yang akan menampilkan nama Mata Pelajaran.

## 6. *Class* Nilai

*Class* Nilai berfungsi untuk memasukkan data nilai para siswa. Pada *Class* Nilai dapat menampilkan nama siswa, mata pelajaran yang ditempuh, dan nilai tiap mata pelajaran.

Berikut merupakan penjelasan relasi antar kelas pada *Class Diagram* yang digambarkan pada gambar 3.8:

1. Terdapat kelas pada diagram digunakan untuk inisialisasi ke dalam *table model* dan basis data aplikasi.
2. Pada *class* Admin memiliki *association* dengan *class* Kelas, artinya admin berhak melihat data Kelas dari *database* melalui *method* `getDataKelas()` dan mengembalikan nilai dengan tipe data *string*. Admin juga dapat menyimpan data ke dalam *database* menggunakan *method* `setDataKelas()`.
3. Pada *class* Admin memiliki *association* dengan *class* Guru, artinya admin berhak melihat data Guru dari *database* melalui *method* `getDataGuru()` dan mengembalikan nilai dengan tipe data *string*. Admin juga dapat menyimpan data ke dalam *database* menggunakan *method* `setDataGuru()`.
4. Pada *class* Admin memiliki *association* dengan *class* Siswa, artinya admin berhak melihat data Siswa dari *database* melalui *method* `getDataSiswa()` dan mengembalikan nilai dengan tipe data



*string*. Admin juga dapat menyimpan data ke dalam *database* menggunakan *method* `setDataSiswa()`.

5. Pada *class* Admin memiliki *association* dengan *class* Mata Pelajaran, artinya admin berhak melihat data Mata Pelajaran dari *database* melalui *method* `getDataMataPelajaran()` dan mengembalikan nilai dengan tipe data *string*. Admin juga dapat menyimpan data ke dalam *database* menggunakan *method* `setDataMataPelajaran()`.
6. Pada *class* Nilai mempunyai *composition* dengan *class* Siswa, artinya *class* Nilai tidak dapat terbentuk apabila *class* Siswa tidak ada. Siswa disini hanya dapat melihat data dari *class* Nilai melalui *method* `getDataNilai()` dan mengembalikan nilai dengan tipe data *string*.
7. Pada *class* Nilai mempunyai *composition* dengan *class* Mata Pelajaran, artinya *class* Nilai tidak dapat berdiri sendiri apabila *class* Mata Pelajaran tidak ada.
8. Pada *class* Guru mempunyai *association* dengan *class* Kelas, artinya Guru dapat melihat data Kelas di dalam *database*. Guru mengambil data dari *database* menggunakan *method* `getDataKelas()` dan mengembalikan nilai dalam tipe data *string*.
9. Pada *class* Guru mempunyai *association* dengan *class* Siswa, artinya Guru dapat melihat data Siswa di dalam *database*. Guru

mengambil data dari *database* menggunakan *method* *getDataSiswa()* dan mengembalikan nilai dalam tipe data *string*.

10. Pada *class* *Guru* mempunyai *association* dengan *class* *Mata Pelajaran*, artinya *Guru* dapat melihat data *Mata Pelajaran* di dalam *database*. *Guru* mengambil data dari *database* menggunakan *method* *getDataMataPelajaran()* dan mengembalikan nilai dalam tipe data *string*.

11. Pada *class* *Guru* mempunyai *association* dengan *class* *Nilai*, artinya *Guru* dapat melihat serta merubah data nilai di dalam *database*. *Guru* mengambil data dari *database* menggunakan *method* *getDataNilai()* dan mengembalikan nilai dalam tipe data *string* serta menyimpan data yang berupa tipe data *string* melalui *method* *setDataNilai(string)* dan mengembalikan dalam tipe data *void*.

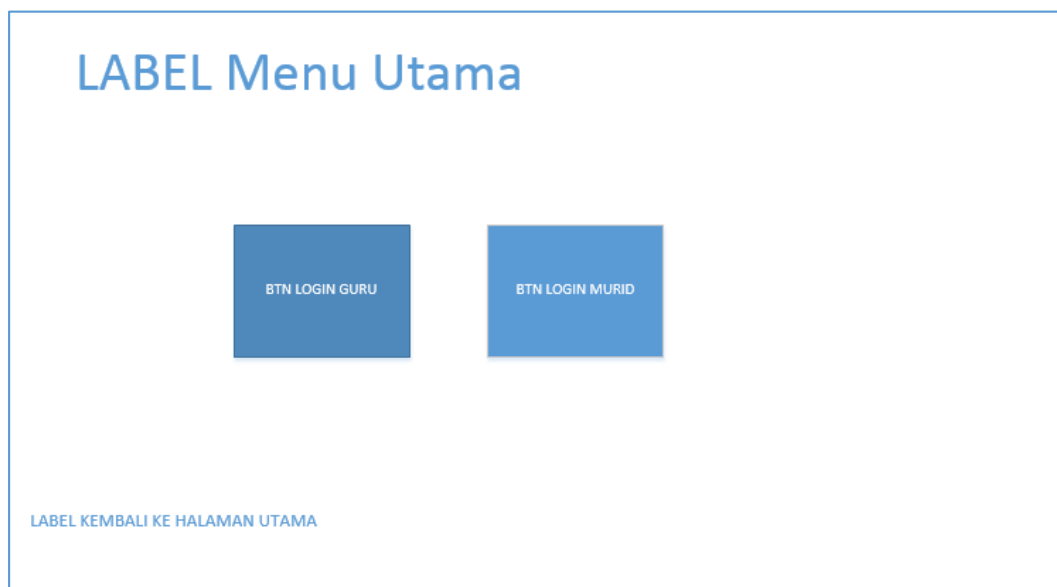
### **3.9 Perancangan Antarmuka**

Antarmuka sangatlah penting dalam suatu aplikasi karena merupakan bagian dari perangkat lunak yang menjadi sarana komunikasi antara pengguna dengan sistem serta dapat memberikan kemudahan bagi pengguna dalam melakukan aktivitasnya.

#### **3.9.1 Rancangan Antarmuka Menu Utama**

Rancangan antarmuka halaman utama merupakan tampilan utama dari aplikasi yang dapat dilihat oleh guru, murid dan admin.

Gambaran rancangan antarmuka halaman utama aplikasi dapat dilihat pada Gambar 3.9.



**Gambar 3.9** Rancangan antarmuka menu utama

Terdapat menu-menu yang dapat dibuka sesuai dengan pengguna. Jika pengguna sudah masuk pada halaman utama, semua menu akan aktif.

### 3.9.2 Rancangan Antarmuka Halaman *Login* Guru

Rancangan untuk halaman *login* Guru terdapat dua buah kolom dan satu *button* yaitu kolom enter *username*, *password* dan satu *button* masuk. Guru dapat *login* sesuai *username*, *password* yang telah dibuat oleh admin. Kemudian di bawahnya terdapat *link* untuk kembali ke halaman utama. Gambaran rancangan antarmuka halaman *login* guru dapat dilihat pada Gambar 3.10.

**Gambar 3.10** Rancangan Antarmuka *Login* Guru

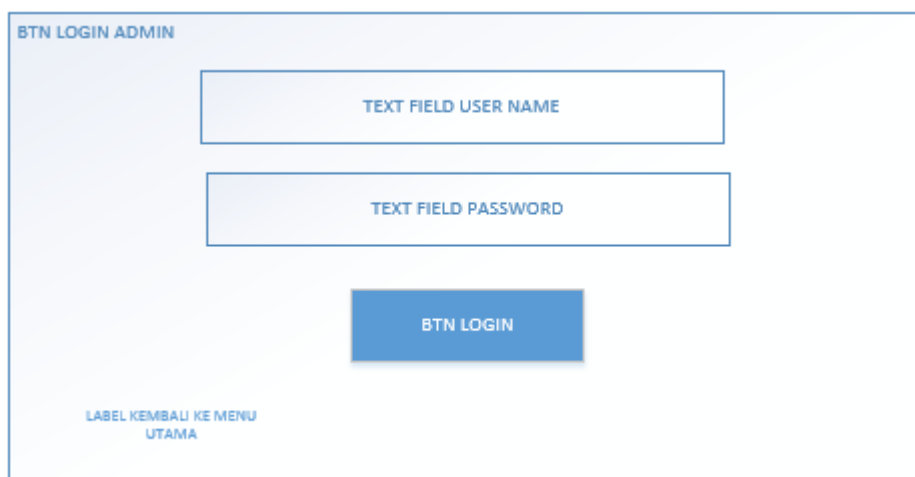
### 3.9.3 Rancangan Antarmuka Halaman Form Nilai Siswa

Rancangan untuk halaman Form Nilai Siswa terdapat beberapa label dan *button* yang berfungsi apabila kita mengeksekusi. Gambaran rancangan antarmuka halaman Form Nilai Siswa dapat dilihat pada Gambar 3.11.

**Gambar 3.11** Rancangan Antarmuka Halaman Form Nilai Siswa

### 3.9.4 Rancangan Antarmuka Halaman *Login Admin*

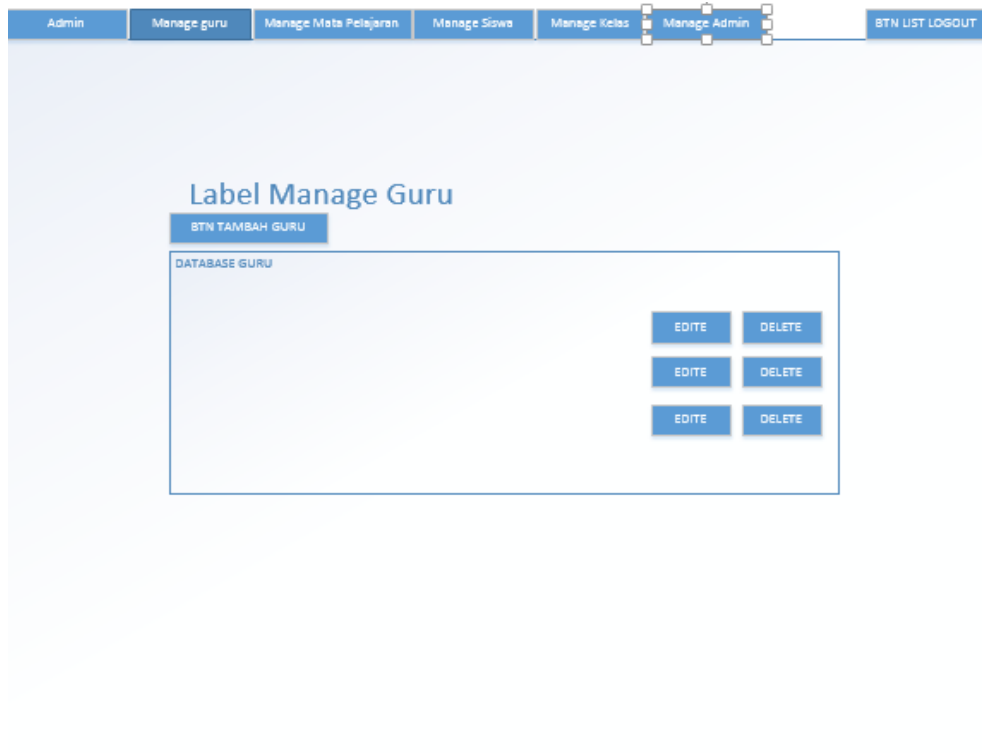
Rancangan untuk halaman *login Admin* terdapat dua buah kolom dan satu *button* yaitu kolom enter *username, password* dan satu *button* masuk. Admin dapat *login* sesuai *username, password*. Kemudian di bawahnya terdapat *link* untuk kembali ke halaman utama. Gambaran rancangan antarmuka halaman *login admin* dapat dilihat pada Gambar 3.12.



**Gambar 3.12** Rancangan Antarmuka *Login Admin*

### 3.9.5 Rancangan Antarmuka Halaman *Manage Data*

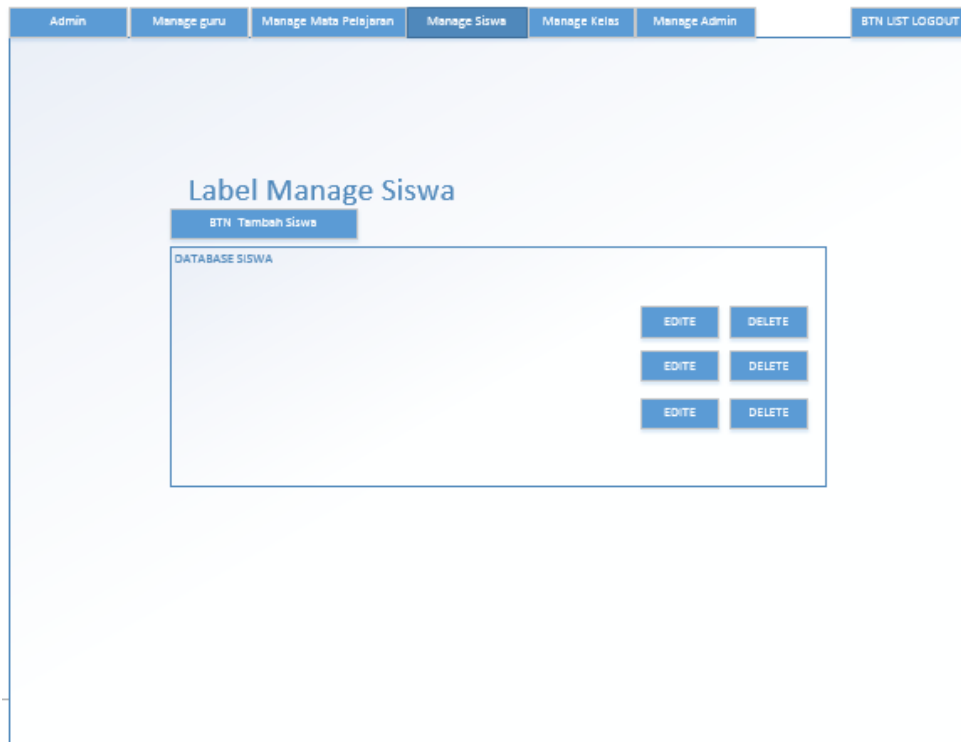
Rancangan antarmuka halaman *manage data* merupakan tampilan dari aplikasi yang hanya dapat dilihat oleh admin. Gambaran rancangan antarmuka halaman *manage data* aplikasi dapat dilihat pada Gambar 3.13 sampai dengan 3.17. Halaman – halaman ini berfungsi untuk mengatur dan merubah data sesuai dengan kuasa admin.



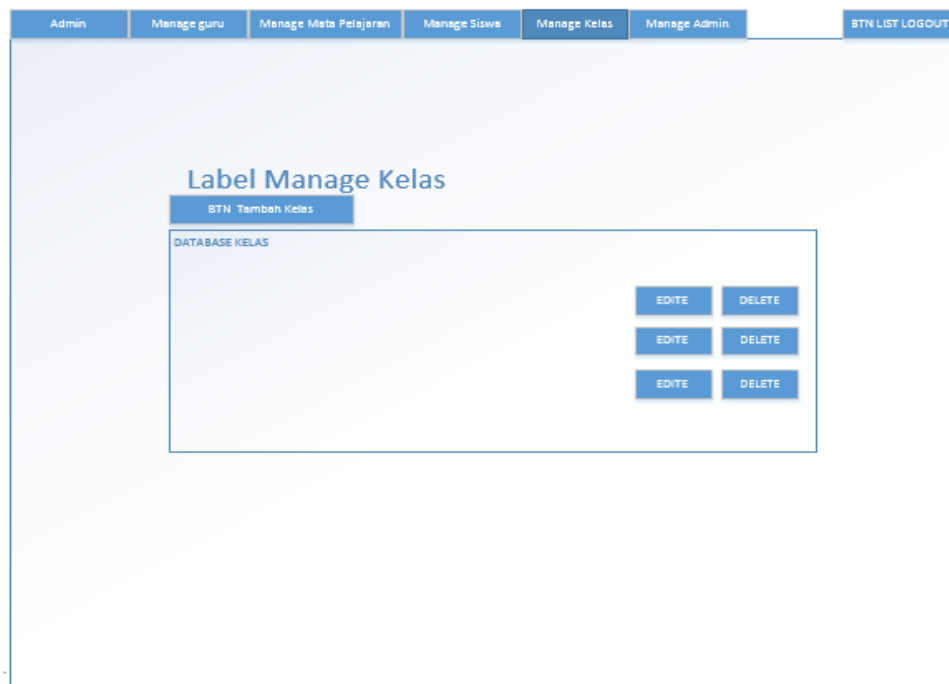
**Gambar 3.13** Rancangan Antarmuka *Manage Data Guru*



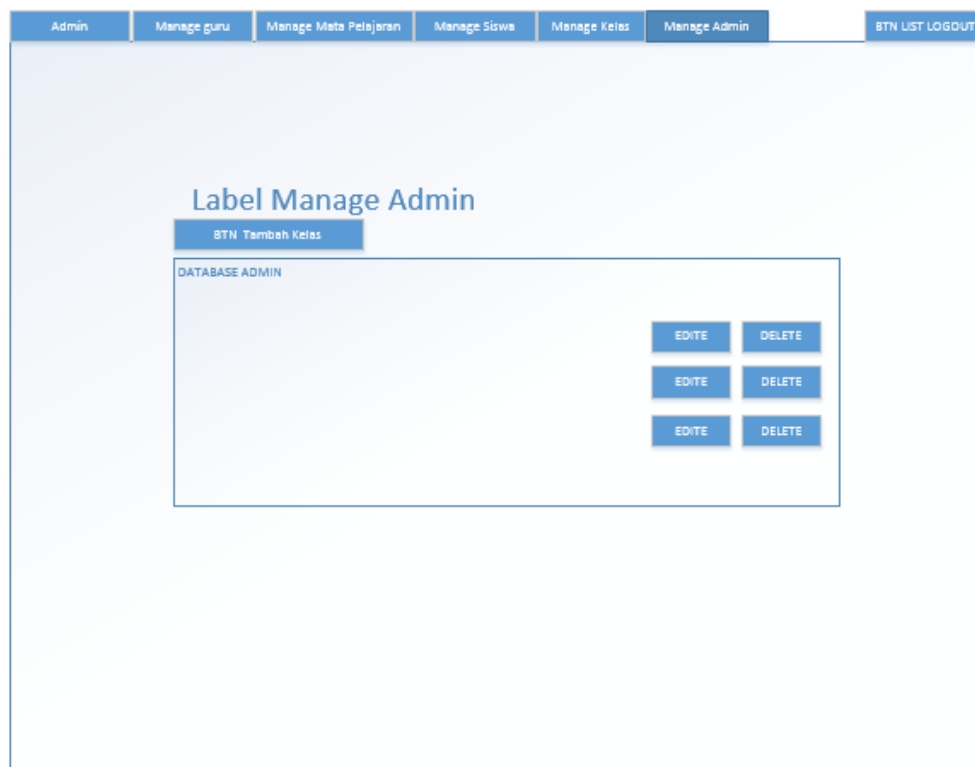
**Gambar 3.14** Rancangan Antarmuka *Manage Data Mata Pelajaran*



**Gambar 3.15** Rancangan Antarmuka *Manage Data Siswa*



**Gambar 3.16** Rancangan Antarmuka *Manage Data Kelas*

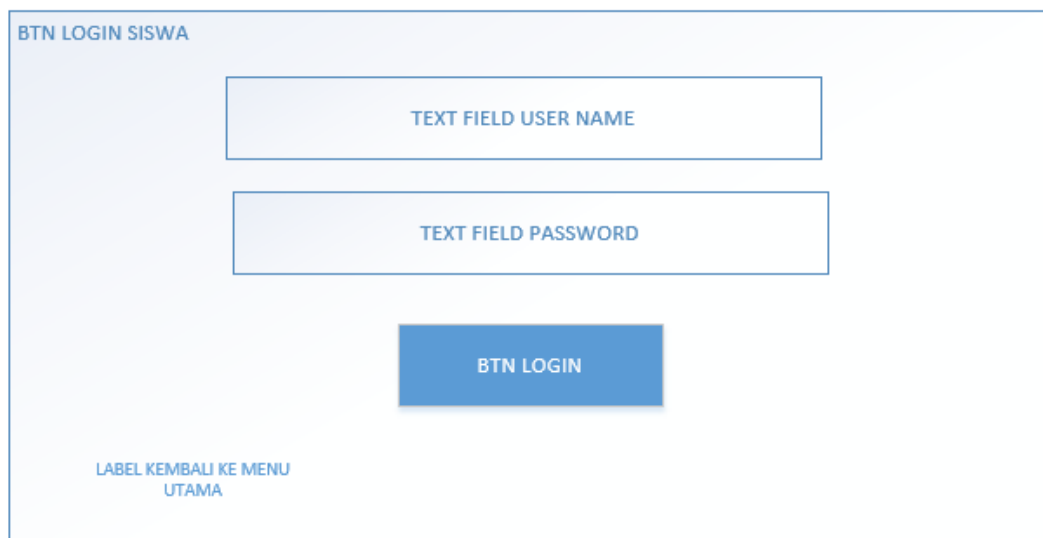


**Gambar 3.17** Rancangan Antarmuka *Manage Data Admin*

### **3.9.6 Rancangan Antarmuka Halaman *Login Siswa***

Rancangan untuk halaman *login* Siswa terdapat dua buah kolom dan satu *button* yaitu kolom enter *username*, *password* dan satu *button* masuk. Siswa dapat *login* sesuai *username*, *password* yang telah diberikan oleh Admin. Kemudian di bawahnya terdapat *link* untuk kembali ke halaman utama. Gambaran rancangan antarmuka halaman *login* admin dapat dilihat pada Gambar 3.18.





The image shows a login interface for a student. It features a light blue background with a white border. At the top left, the text "BTN LOGIN SISWA" is displayed. Below this, there are two white rectangular text input fields with blue borders. The first field is labeled "TEXT FIELD USER NAME" and the second is labeled "TEXT FIELD PASSWORD". Centered below these fields is a solid blue rectangular button labeled "BTN LOGIN". In the bottom left corner, there is a label "LABEL KEMBALI KE MENU UTAMA".

**Gambar 3.18** Rancangan Antarmuka *Login* Siswa

### 3.9.7 Rancangan Antarmuka Halaman Laporan Nilai Siswa

Rancangan untuk halaman laporan nilai siswa. Siswa dapat melihat dan mencetak laporan nilai siswa sesuai dengan NIS masing – masing. Gambaran rancangan antarmuka halaman laporan dapat dilihat pada Gambar 3.19.



The image shows a report interface for a student. It features a light blue background with a white border. At the top center, the text "LABEL LAPORAN HASIL BELAJAR SISWA" is displayed. On the left side, there is a vertical list of labels: "LABEL Nama peserta didik", "LABEL No. Induk/NISN", "LABEL Nama Sekolah", "LABEL Kelas", "LABEL Tahun Ajaran", and "LABEL SEMESTER". In the center of the page, there is a large white rectangular area with a blue border, labeled "DATA BASE HASIL RAPOT SISWA".

**Gambar 3.19** Rancangan Antarmuka Laporan Nilai Siswa

### **3.10 Metode Pengujian**

Metode pengujian yang dipakai dalam pengembangan aplikasi adalah tes fungsional yang dilakukan untuk memperoleh informasi serta mengevaluasi mengenai kualitas dari produk atau layanan yang sedang diuji. Pengujian menggunakan metode tes fungsional ialah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak tersebut.

Hal yang menjadi perhatian dalam pengujian adalah aplikasi dapat menyimpan dan mengubah data seluruh siswa, nilai siswa, kelas, guru dan mata pelajaran. Aplikasi dapat mengolah data mata pelajaran dan nilai yang didapat.