

BAB III

METODE PENELITIAN

3.1 Alat Dan Bahan

Pada saat penelitian dibutuhkan beberapa alat dan bahan untuk mendukung dalam perancangan sistem maupun implementasi aplikasi.

3.1.1 Alat

Dalam penelitian pembuatan Sistem Informasi Akademik Sekolah Dasar, alat yang digunakan adalah sebagai berikut:

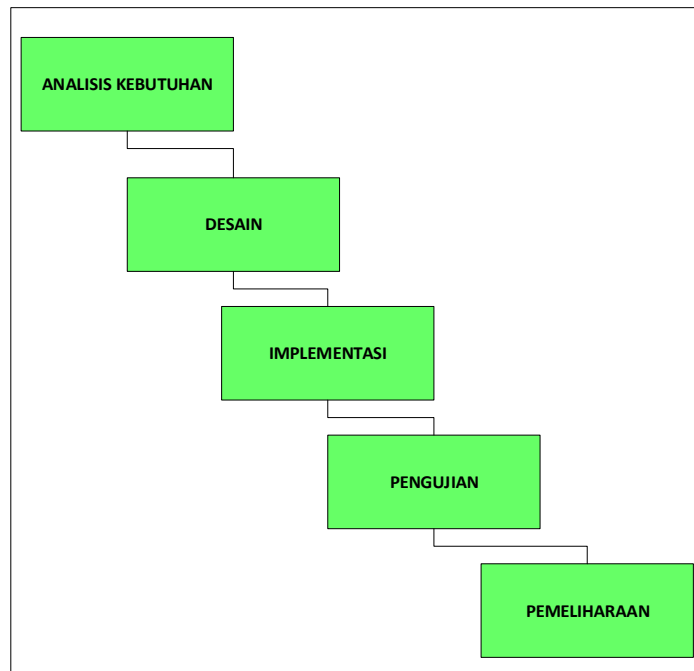
1. Perangkat Keras
 - a. Perangkat Komputer atau Laptop.
32/64 bit architecture processor,
minimal 2 GB Random Access Memory (RAM),
Sistem Operasi Windows XP/7/8.
 - b. *Printer* Dokumen untuk mencetak data.
2. Perangkat lunak
 - a. *Balsamiq Mockups 3.*
 - b. *Microsoft Visual Studio 2013.*
 - c. *Microsoft Visio 2013.*
 - d. *SQL Server 2014 Management Studio.*

3.1.2 Bahan

Bahan yang digunakan untuk penelitian adalah data-data yang didapatkan dari hasil wawancara langsung dengan pihak SD Negeri 06 Koto Salak. Dari wawancara tersebut dapat diketahui apa saja kebutuhan yang diperlukan untuk membangun aplikasi.

3.2 Alur Penelitian

Metode yang digunakan dalam penelitian ini yaitu metode *SDLC Waterfall* menurut Roger S. Pressman. Dapat dilihat pada Gambar 3.1.



Gambar 3.1 Metode SDLC *Waterfall*

1. Analisis kebutuhan

Langkah pertama adalah mencari kebutuhan untuk membangun sebuah aplikasi. Pada penelitian ini analisis kebutuhan didapatkan melalui wawancara dan survei.

2. Desain

Pada proses ini peneliti mulai mendesain *user interface* yang didalamnya terdapat kebutuhan aplikasi yang didapatkan pada metode sebelumnya.

3. Implementasi

Setelah mendesain *user interface*, langkah selanjutnya adalah melakukan pengkodean. Proses koding ini bertujuan agar desain aplikasi tersebut dapat di mengerti oleh mesin, dalam hal ini biasa disebut dengan bahasa pemrograman. Bahasa pemrograman yang digunakan adalah C#.

4. Pengujian

Setelah proses pengkodean selesai, aplikasi yang telah dibuat perlu dilakukan pengujian *software*. Tujuannya adalah agar aplikasi terbebas dari *error*. Hasil dari aplikasi tersebut harus sesuai dengan kebutuhan yang didapatkan pada analisis kebutuhan.

5. Pemeliharaan

Pemeliharaan pada aplikasi sangat diperlukan, termasuk didalamnya adalah pemeliharaan dalam memperbaiki kesalahan yang tidak ditemukan pada tahapan sebelumnya.

3.3 Analisis Kebutuhan Perangkat Lunak

3.3.1 Kebutuhan Proses

Kebutuhan proses dalam aplikasi sistem informasi akademik sekolah dasar, yaitu:

1. Proses pencarian data siswa, data guru, nilai siswa, data kelas, dan mata pelajaran.
2. Proses menyimpan *file* ke dalam bentuk dokumen *word (.docx)*.

3.3.2 Kebutuhan *Input*

Kebutuhan *input* atau masukan dari aplikasi sistem informasi akademik sekolah dasar, yaitu sebagai berikut:

1. *Input* Data siswa.
2. *Input* Data guru.
3. *Input* Data kelas.
4. *Input* Nilai siswa.
5. *Input* Mata Pelajaran.

3.3.3 Kebutuhan *Output*

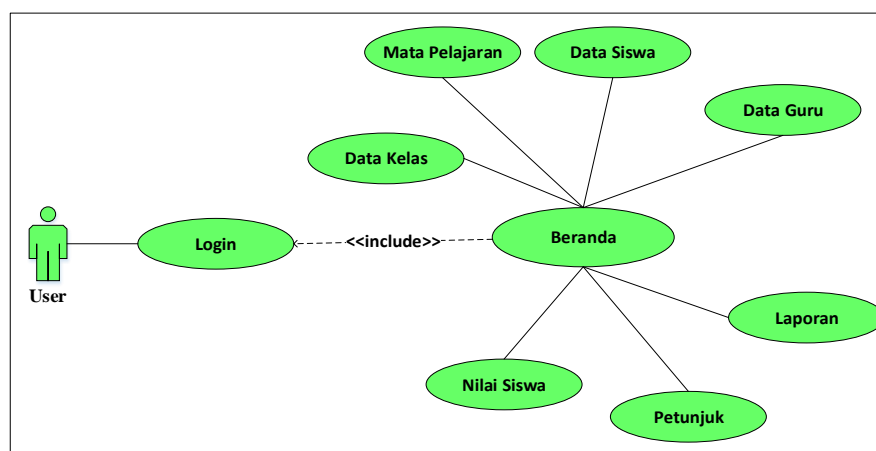
1. Cetak dokumen Data siswa.
2. Cetak dokumen Data guru.
3. Cetak dokumen Data kelas.
4. Cetak dokumen Nilai Siswa.
5. Cetak dokumen Mata Pelajaran.

3.4 Rancangan Sistem

Tahap perancangan sistem dengan menggunakan diagram UML (*Unified Modeling Language*) meliputi *use case* diagram, *class* diagram, *sequence* diagram, dan *activity* diagram.

3.4.1 Use Case Diagram

Use case diagram untuk aplikasi sistem informasi akademik sekolah dasar dapat dilihat pada Gambar 3.2.



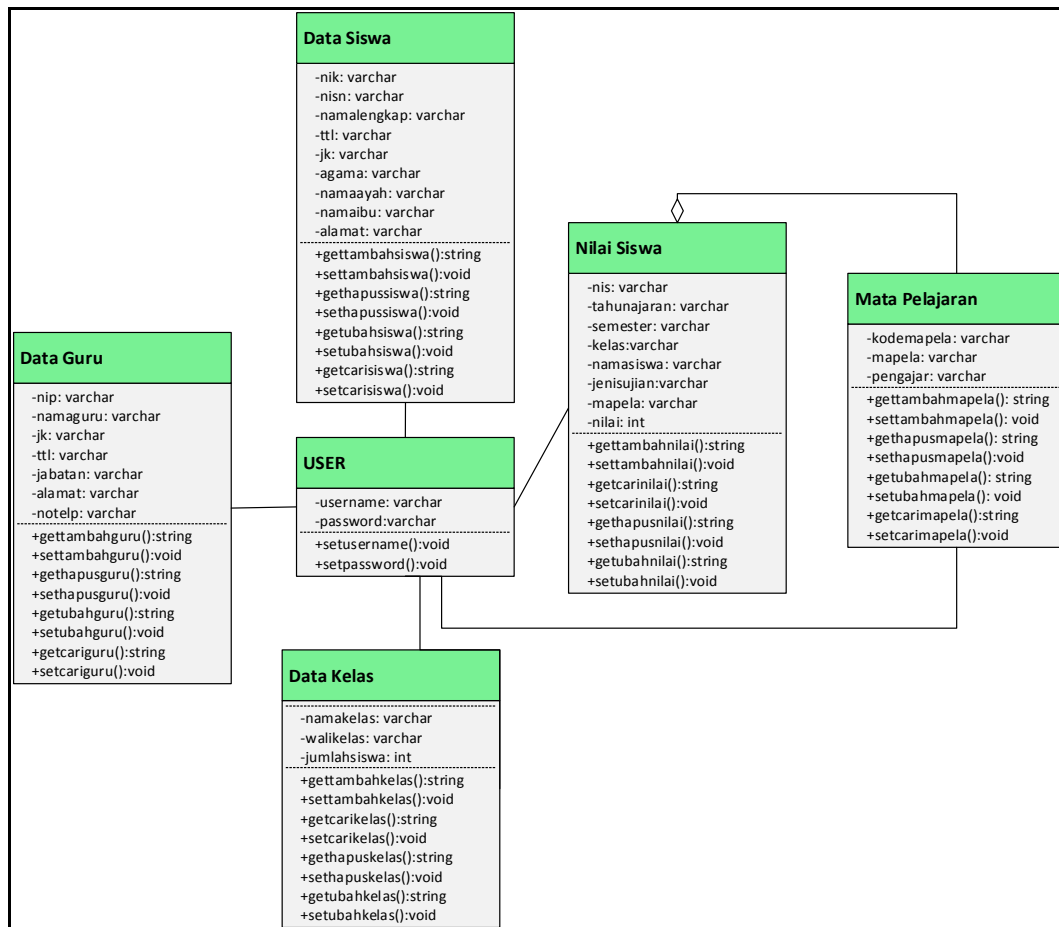
Gambar 3.2 Use Case diagram Aplikasi

Berikut penjelasan tentang Gambar 3.2:

1. Terdapat 1 aktor pada *use case* diagram yaitu *user*, dalam lingkup sekolah yaitu Tata Usaha sekolah.
2. *User* berhubungan langsung dengan *use case* diagram *login* seperti Beranda.
3. Pada *use case* beranda berhubungan langsung kepada data kelas, mata pelajaran, data siswa, data guru, nilai siswa, laporan, dan petunjuk.

3.4.2 Class Diagram

Class diagram yang digunakan untuk aplikasi sistem informasi akademik sekolah dasar dapat dilihat pada Gambar 3.3.



Gambar 3.3 Class Diagram Aplikasi

Berikut penjelasan *class* diagram pada Gambar 3.3:

1. Pada *class user*, terdapat fungsi *username* dan *password* yang digunakan untuk validasi pada *class login*.
2. Pada *class data kelas*, terdapat fungsi untuk memasukkan data kelas berupa nama kelas, wali kelas, dan jumlah siswa.
3. Pada *class data guru*, terdapat fungsi untuk memasukkan data guru berupa nip, nama guru, jenis kelamin, tempat dan tanggal lahir, jabatan, alamat, dan nomor telepon.
4. Pada *class data siswa*, terdapat fungsi untuk memasukkan data siswa berupa nik, nis, nama lengkap, tempat dan tanggal lahir, jenis kelamin, agama, nama ayah, nama ibu, dan alamat.

5. Pada *class* nilai siswa, terdapat fungsi untuk memasukkan data nilai siswa berupa nis, tahun ajaran, semester, kelas, nama siswa, jenis ujian, mata pelajaran, dan nilai siswa.
6. Pada *class* mata pelajaran, terdapat fungsi untuk memasukkan data mata pelajaran berupa kode mata pelajaran, mata pelajaran, dan pengajar.

Class user memiliki *association* dengan *class* data kelas. *User* dapat menambah data kelas melalui *method* *tambahkelas()*, *user* dapat mencari data kelas melalui *method* *carikelas()*, *user* dapat mengubah data kelas melalui *method* *ubahkelas()*, dan *user* dapat menghapus data kelas melalui *method* *hapuskelas()*.

Class user memiliki *association* dengan *class* data guru . *User* dapat menambah data guru melalui *method* *tambahguru()*, *user* dapat mencari data guru melalui *method* *cariguru()*, *user* dapat mengubah data guru melalui *method* *ubahguru()*, dan *user* dapat menghapus data guru melalui *method* *hapusguru()*.

Class user memiliki *association* dengan *class* data siswa. *User* dapat menambah data siswa melalui *method* *tambahsiswa()*, *user* dapat mencari data siswa melalui *method* *carisiswa()*, *user* dapat mengubah data siswa melalui *method* *ubahsiswa()*, dan *user* dapat menghapus data siswa melalui *method* *hapussiswa()*.

Class user memiliki *association* dengan *class* nilai siswa. *User* dapat menambah nilai siswa melalui *method* *tambahnilai()*, *user* dapat mencari nilai siswa melalui *method* *carinilai()*, *user* dapat mengubah nilai siswa melalui *method* *ubahnilai()*, dan *user* dapat menghapus nilai siswa melalui *method* *hapusnilai()*.

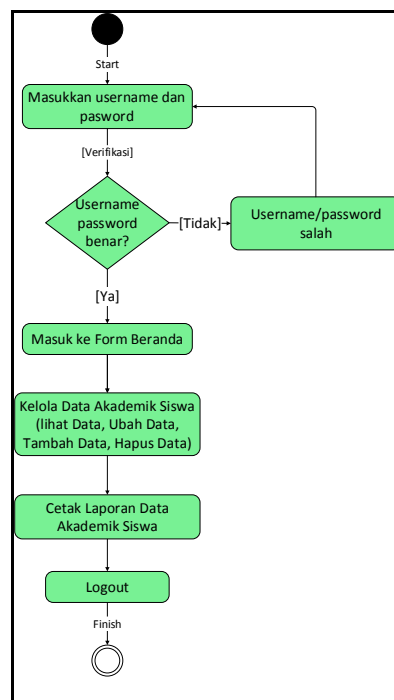
Class user memiliki *association* dengan *class* mata pelajaran. *User* dapat menambah mata pelajaran melalui *method* *tambahmapela()*, *user* dapat mencari data mata pelajaran *method* *carimapela()*, *user* dapat mengubah mata pelajaran melalui *method* *ubahmapela()*, dan *user* dapat menghapus mata pelajaran melalui *method* *hapusmapela()*.

Class nilai siswa memiliki *aggregation* dengan *class* mata pelajaran. Artinya *class* mata pelajaran menjadi atribut bagi *class* nilai siswa. Jika *class* mata

pelajaran tidak ada, maka atribut pada *class* nilai siswa yaitu mata pelajaran tidak akan tampil atau data kosong.

3.4.3 Activity Diagram

Activity diagram untuk proses *login* oleh *user* pada aplikasi sistem informasi akademik sekolah dasar dapat dilihat pada Gambar 3.4.



Gambar 3.4 Activity Diagram *Login user*

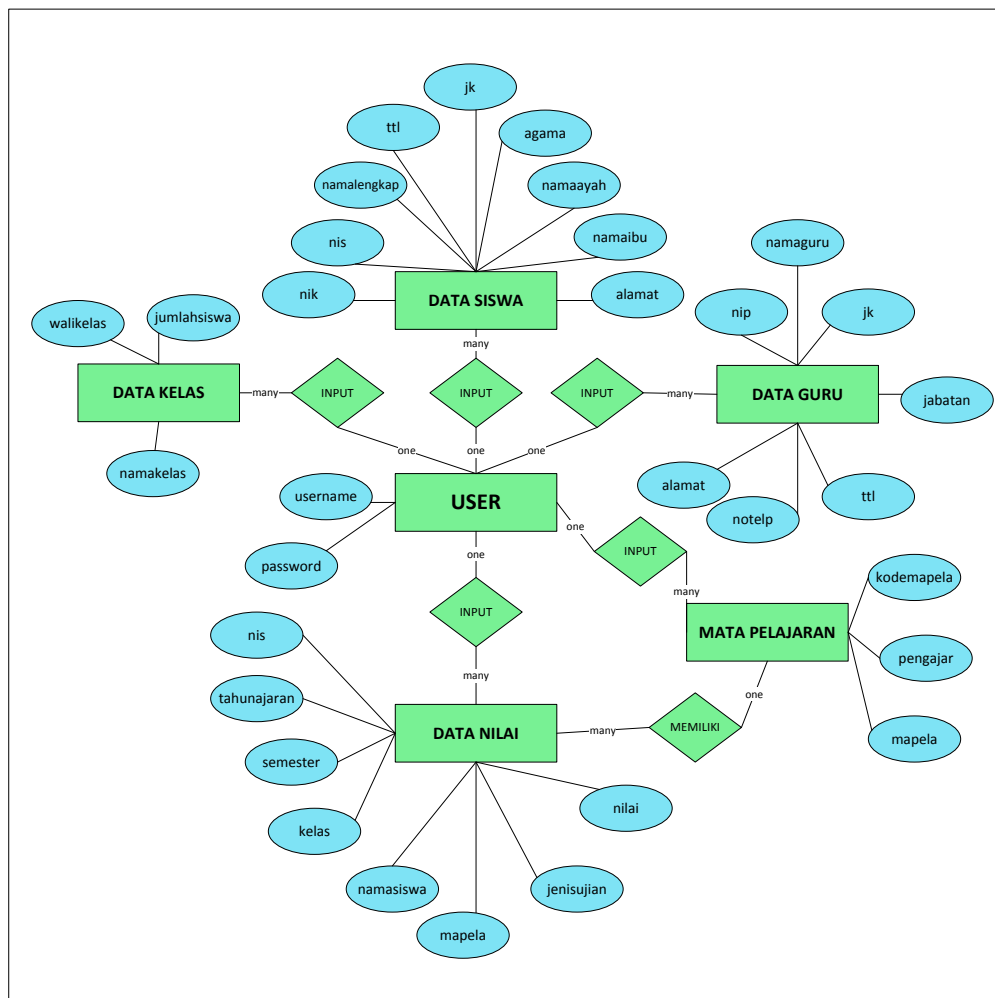
Berikut penjelasan tentang Gambar3.4:

1. *User* melakukan proses login dengan memasukkan *username* dan *password*.
2. Jika *username* atau *password* yang *user* masukkan salah, maka akan kembali ke *login*.
3. Jika *username* atau *password* yang *user* masukkan benar, maka akan masuk ke beranda.
4. *User* dapat mengelola data, dalam hal lihat data, tambah data, ubah data, dan hapus data.

3.5 Rancangan Basis Data

Pengembangan rancangan basis data pada aplikasi adalah menggunakan ER Diagram. ER Diagram merupakan model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar yang mempunyai hubungan antar relasi.

Tujuan ER Diagram adalah untuk memberikan gambaran umum tentang sistem yang dikembangkan dan memudahkan dalam merancang basis data. ER Diagram untuk aplikasi sistem informasi akademik sekolah dasar dapat dilihat pada Gambar 3.5.



Gambar 3.5 ER Diagram

Pada Gambar 3.5 menunjukkan *Entity Relationship Diagram* yang terdapat 6 buah entitas, yaitu:

- a. Entitas *User*
- b. Entitas Data Kelas
- c. Entitas Data Siswa
- d. Entitas Data Nilai
- e. Entitas Data Guru
- f. Entitas Mata Pelajaran

Berikut penjelasan dari Gambar 3.5:

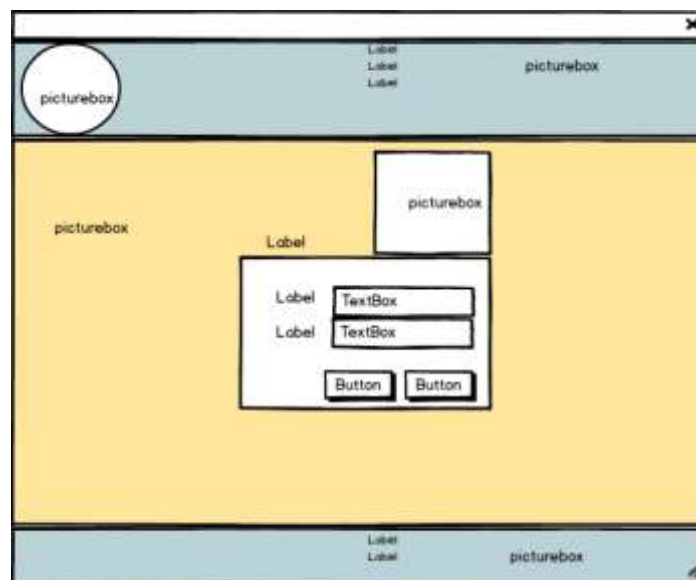
1. Entitas *user* memiliki relasi dengan entitas data kelas *one to many*. Artinya *user* dapat menambahkan banyak data kelas.
2. Entitas *user* memiliki relasi dengan entitas data siswa *one to many*. Artinya *user* dapat menambahkan banyak data siswa.
3. Entitas *user* memiliki relasi dengan entitas data nilai *one to many*. Artinya *user* dapat menambahkan banyak data nilai.
4. Entitas *user* memiliki relasi dengan entitas data guru *one to many*. Artinya *user* dapat menambahkan banyak data guru.
5. Entitas *user* memiliki relasi dengan entitas mata pelajaran *one to many*. Artinya *user* dapat menambahkan banyak mata pelajaran.
6. Entitas mata pelajaran memiliki relasi dengan entitas data nilai *one to many*. Artinya satu mata pelajaran mempunyai banyak data nilai.
7. *Username* dan *password* merupakan suatu atribut yang mendeskripsikan entitas *user*.
8. Nama kelas, wali kelas, dan jumlah siswa adalah atribut yang mendeskripsikan entitas Data Kelas.
9. Kode mapela, mapela, dan pengajar adalah atribut yang mendeskripsikan entitas Mata Pelajaran.
10. NIK, NIS, nama lengkap, ttl, jk, agama, nama ayah, nama ibu, dan alamat merupakan suatu atribut yang mendeskripsikan entitas Data Siswa.

11. NIS, Tahun ajaran, semester, kelas, nama siswa, jenis ujian, mapela, dan nilai adalah atribut yang mendeskripsikan entitas Data Nilai.
12. NIP, nama guru, jk, ttl, jabatan, alamat, dan notelp adalah atribut yang mendeskripsikan entitas Data Guru.

3.6 Rancangan Antarmuka

3.6.1 Rancangan antarmuka *Form Login*

Rancangan antarmuka *form login* merupakan tampilan utama dari aplikasi yang nantinya digunakan oleh *user*. Gambar dari rancangan antarmuka *form login* dapat dilihat pada Gambar 3.6.

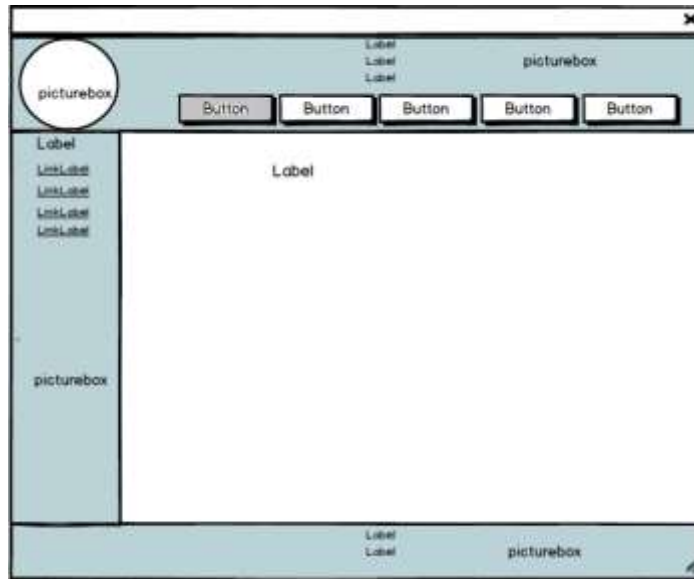


Gambar 3.6 Rancangan Antarmuka *Form Login*

3.6.2 Rancangan Antarmuka *Form Beranda*

Rancangan antarmuka *form beranda* merupakan tampilan utama dari aplikasi setelah login. Pada rancangan antarmuka *form beranda* terdapat beberapa *button*, dan beberapa *linklabel*.

Pada rancangan antarmuka *form beranda* terdapat menu-menu yang apabila diklik akan menampilkan tampilan sesuai dengan fungsinya masing-masing. Rancangan antarmuka *form beranda* dapat dilihat pada Gambar 3.7.

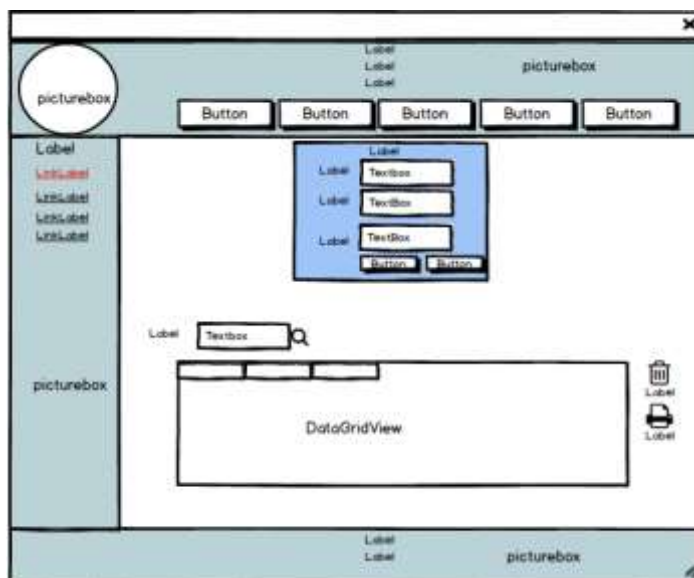


Gambar 3.7 Rancangan Antarmuka *Form* Beranda

3.6.3 Rancangan Antarmuka *Form* Data Kelas

Pada rancangan antarmuka *form* data kelas terdapat beberapa *label*, *textbox*, *button*, dan *DataGridView*. *User* dapat melakukan proses *input* pada *textbox* data kelas dan dapat melakukan proses pencarian data kelas.

DataGridView berfungsi untuk menampilkan tabel data kelas yang ada pada *database* secara keseluruhan. Rancangan antarmuka *form* data kelas dapat dilihat pada Gambar 3.8.

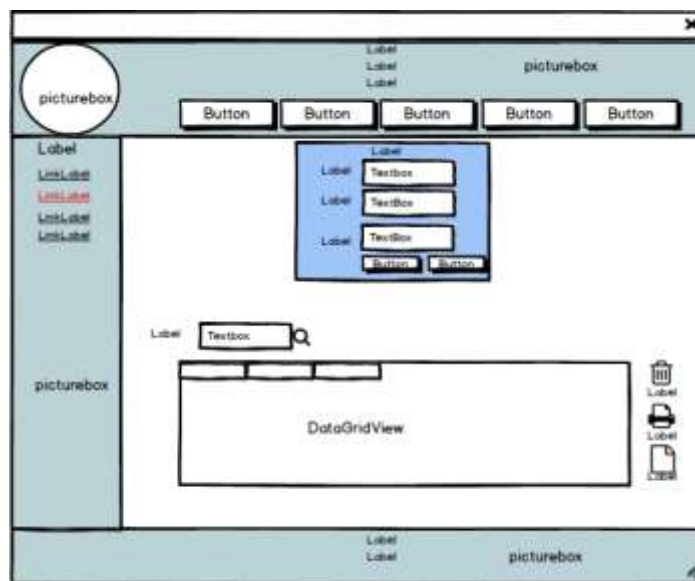


Gambar 3.8 Rancangan Antarmuka *Form* Data Kelas

3.6.4 Rancangan Antarmuka *Form* Mata Pelajaran

Rancangan antarmuka *form* mata pelajaran, berisi beberapa *label*, *textbox*, *button*, dan *DataGridView*. *User* dapat melakukan proses *input* mata pelajaran, dan dapat melakukan proses pencarian data mata pelajaran.

DataGridView pada *form* ini berfungsi untuk menampilkan tabel data mata pelajaran yang ada pada *database* secara keseluruhan. Rancangan antarmuka *form* mata pelajaran dapat dilihat pada Gambar 3.9.

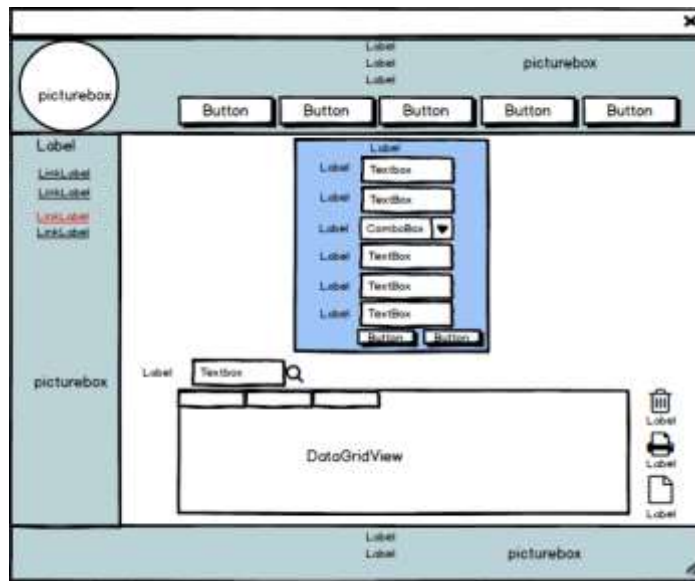


Gambar 3.9 Rancangan Antarmuka *Form* Mata Pelajaran

3.6.5 Rancangan Antarmuka *Form* Data Guru

Rancangan antarmuka *form* data guru, berisi beberapa *label*, *textbox*, *button*, dan *DataGridView*. *User* dapat melakukan proses *input* data guru pada *textbox* yang telah disediakan, dan *user* dapat melakukan pencarian data.

DataGridView berfungsi untuk menampilkan tabel data guru yang ada pada *database* secara keseluruhan. Rancangan antarmuka *form* data guru dapat dilihat pada Gambar 3.10.

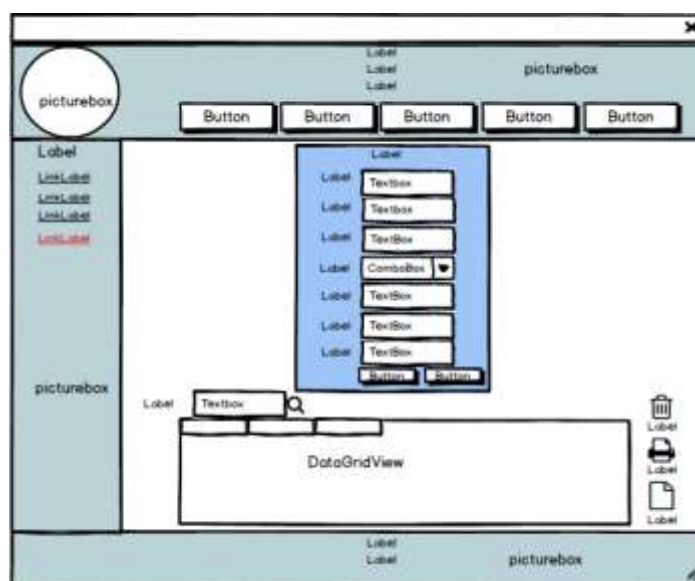


Gambar 3.10 Rancangan Antarmuka *Form* Data Guru

3.6.6 Rancangan Antarmuka *Form* Data Siswa

Rancangan antarmuka *form* data siswa, berisi beberapa *label*, *textbox*, *button*, dan *DataGridView*. Pada *form* ini *user* dapat melakukan proses *input* data siswa pada *textbox* yang telah disediakan, dan *user* dapat melakukan pencarian data.

DataGridView berfungsi untuk menampilkan tabel data siswa yang ada pada *database* secara keseluruhan. Rancangan antarmuka *form* data siswa dapat dilihat pada Gambar 3.11.

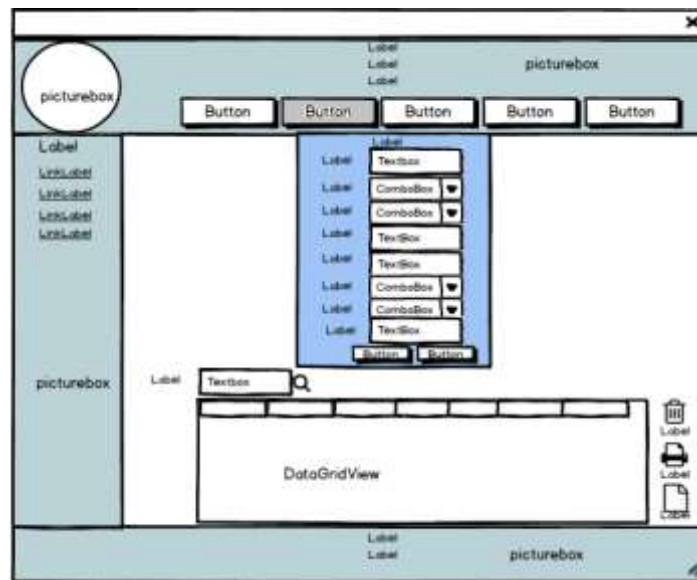


Gambar 3.11 Rancangan Antarmuka *Form* Data Siswa

3.6.7 Rancangan Antarmuka *Form* Nilai Siswa

Rancangan antarmuka *form* nilai siswa, berisi beberapa *label*, *textbox*, *button*, dan *DataGridView*. *User* dapat melakukan proses *input* data nilai siswa pada *textbox* yang telah disediakan, dan *user* dapat melakukan pencarian data.

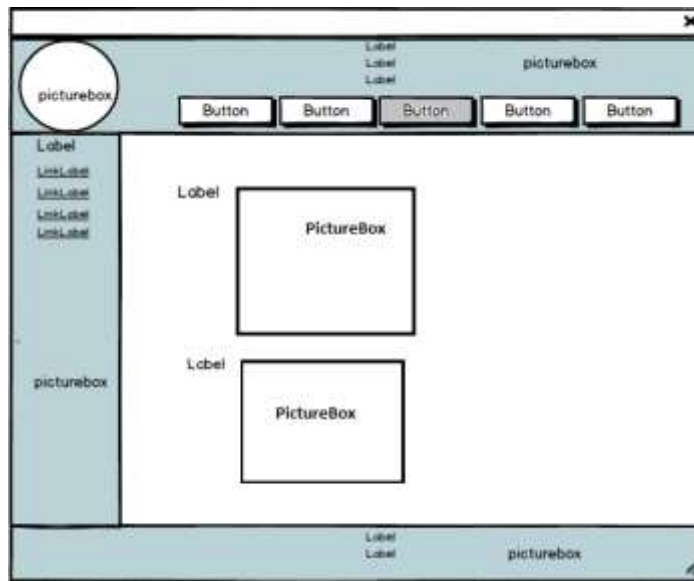
DataGridView berfungsi untuk menampilkan tabel data nilai siswa yang ada pada *database* secara keseluruhan. Rancangan antarmuka *form* nilai siswa dapat dilihat pada Gambar 3.12.



Gambar 3.12 Rancangan Antarmuka *Form* Nilai Siswa

3.6.8 Rancangan Antarmuka *Form* Laporan

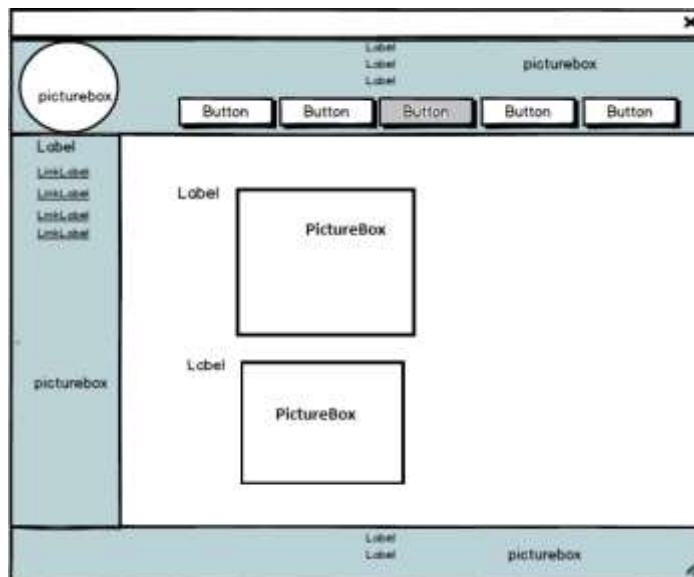
Rancangan antarmuka *form* Laporan menampilkan data secara keseluruhan di dalam sebuah tabel. *User* dapat mencari data dengan kategori yang lebih spesifik. Data dapat di simpan ke bentuk *file Document Microsoft Word* dan *file Microsoft Excel*. Rancangan antarmuka *form* Laporan dapat dilihat pada Gambar 3.13.



Gambar 3.13 Rancangan Antarmuka *Form* Petunjuk

3.6.9 Rancangan Antarmuka *Form* Petunjuk

Rancangan antarmuka *form* petunjuk berisi *label-label* dan *picturebox-picturebox*. Pada *form* petunjuk ini akan menampilkan langkah-langkah menggunakan aplikasi. Rancangan antarmuka *form* petunjuk dapat dilihat pada Gambar 3.14.



Gambar 3.14 Rancangan Antarmuka *Form* Petunjuk

3.7 Pengembangan Sistem

Dalam aplikasi, *user* hanya satu orang saja yaitu tata usaha sekolah. Berikut beberapa prosedur yang dilakukan dalam pengembangan sistem:

1. Menyiapkan perangkat keras dan perangkat lunak pendukung yang dibutuhkan.
2. Merancang *user interface* aplikasi sehingga diharapkan dapat memudahkan pengguna dalam menggunakan aplikasi.
3. Membuat basis data sesuai dengan kebutuhan aplikasi
4. Melakukan pengkodean di dalam *user interface* yang telah dibuat.
5. Membuat sistem yang dapat menampilkan dan mencetak laporan data akademik siswa.

3.8 Metode Pengujian

Pengujian perangkat lunak merupakan suatu kegiatan yang dilakukan untuk memperoleh informasi serta mengevaluasi kualitas dari layanan aplikasi yang sedang diuji. Tujuan pengujian dalam pengembangan aplikasi adalah untuk dapat memenuhi kebutuhan yang diperlukan oleh sekolah dasar.

Metode pengujian perangkat lunak yang digunakan adalah metode *Black Box*. Secara garis besar teknik pengujian ini tidak perlu memerlukan pengetahuan tentang cara kerja pada bagian dalam aplikasi, sehingga penguji hanya berfokus kepada spesifikasi fungsional dari aplikasi. Kemudian penguji hanya memeriksa *output* tanpa perlu mengetahui bagaimana proses *input* bekerja.

Hal-hal yang menjadi perhatian dalam pengujian adalah sebagai berikut:

1. Aplikasi dapat menambah, menghapus, mencari, dan mengubah data akademik siswa (data siswa, data guru, mata pelajaran, data kelas, dan data nilai).
2. Aplikasi dapat mencetak laporan-laporan data akademik siswa.
3. Aplikasi dapat menyimpan *file* dalam bentuk dokumen *word (.docx)* data akademik siswa.