

BAB III

METODOLOGI PENELITIAN

3.1. Alat dan Bahan Penelitian

Dalam penelitian dibutuhkan beberapa alat dan bahan untuk mendukung berjalannya perancangan dan implementasi *aplikasi web*.

3.1.1. Alat

Alat yang digunakan berupa perangkat keras (*hardware*) dan perangkat lunak (*software*).

3.1.1.1. Perangkat Keras (*Hardware*) *Personal Computer* (PC) atau laptop.

- a. Interl® Core™ i5 *processor*,
- b. 8.00 GB RAM,
- c. 64-bit *Operating System*,
- d. Windows 10 *pro*. *Printer* dokumen untuk mencetak laporan.

3.1.1.2. Perangkat Lunak (*Software*)

- a. *Microsoft Visio 2016*
- b. *Xampp*
- c. *Visual Studio Code*

3.1.2. Bahan

Data yang diperoleh dari penelitian ini melalui studi literatur dengan mencari referensi teori yang relevan dengan kasus atau permasalahan yang ditemukan. Referensi teori yang diperoleh dengan jalan penelitian studi literatur dijadikan sebagai fondasi dasar dalam mengembangkan aplikasi.

3.2. Teknik Pengumpulan Data

3.2.1. Observasi

Observasi adalah metode pengumpulan data kualitatif yang dianjurkan untuk mendapatkan data-data deskriptif. Metode pengumpulan data observasi berasal dari kata *observation* yang berarti pengamatan. Metode pengumpulan data observasi cocok digunakan untuk penelitian yang bertujuan untuk mempelajari aplikasi-aplikasi yang memiliki kemiripan fungsi dan kemudian dilakukan analisa terhadap keunggulan dan kelemahan aplikasi tersebut agar dapat dijadikan sebuah referensi.

3.2.2. Wawancara

Wawancara merupakan teknik pengumpulan data yang dilakukan melalui tatap muka dan tanya jawab antara peneliti dan narasumber secara lisan baik langsung maupun tidak langsung. Lexy J Moleong mendefinisikan wawancara sebagai percakapan dengan maksud tertentu.

Dalam hal ini peneliti akan menggunakan metode wawancara langsung dengan subjek informan dari *TPA-KB-TKIT Nurul Ittihad*. Peneliti menggunakan wawancara atau interview tak terstruktur yaitu wawancara yang bentuk pertanyaannya bebas

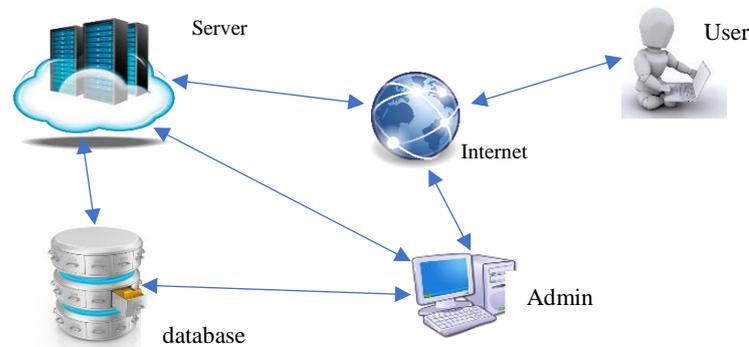
3.3. Gambaran Umum Sistem Usulan

Sistem yang akan dibangun dalam tugas akhir ini adalah sebuah Sistem Informasi *TPA-KB-TKIT Nurul Ittihad* berbasis web yang dapat diakses melalui perangkat apapun termasuk *smartphone* dan laptop. Dari sistem informasi ini bertujuan untuk menampilkan informasi-informasi tentang profil sekolah, blog sekolah, data pengajar dan informasi tentang jumlah siswa.

Pengguna sistem ini akan dibagi menjadi 2 bagian. Bagian yang pertama adalah administrator sistem yang memiliki tugas untuk mengelola data, kemudian bagian yang kedua adalah pengguna (*user*), yaitu orang yang menggunakan aplikasi untuk mencari informasi tentang sekolah tersebut.

3.4. Analisis Penelitian

3.3.1. Arsitektur



Gambar 3. 1 Arsitektur Website *TPA-KB-TKIT Nurul Ittihad*

Sesuai pada gambar 3.1 dalam penggunaan *database* yang diterapkan pada sistem adalah MySQL. Untuk bahasa pemrograman yang digunakan adalah PHP dengan menggunakan framework *CodeIgniter*. Pada saat pengguna akan mengakses *website*, maka digunakanlah *internet* dan *web browser* sebagai perantara untuk menampilkan atau memuat antar muka yang diambil serta dibutuhkan dari *database*. Melalui tampilan yang sudah dimuat, pengguna dapat mendapatkan informasi yang dibutuhkan.

3.3.2. Analisis Pengguna

Website yang dibuat dapat digunakan oleh *admin* dan pengguna. *Admin* berperan untuk mengelola data, seperti menambahkan data, hapus data, *edit* dan update data di antaranya pada halaman *Blog*, data guru dan data siswa. Sedangkan pengguna hanya dapat melihat data informasi tentang sekolah. Berikut ini merupakan analisis kebutuhan pada *website*:

a. Kebutuhan Pengguna

1. Melihat informasi tentang *website*
2. Menghubungi pemilik *website*
3. Mendapatkan informasi terbaru di halaman blog
4. Melihat profil sekolah, berupa sambutan dan visi misi
5. Melihat program sekolah
6. Melihat fasilitas sekolah
7. Melihat data guru

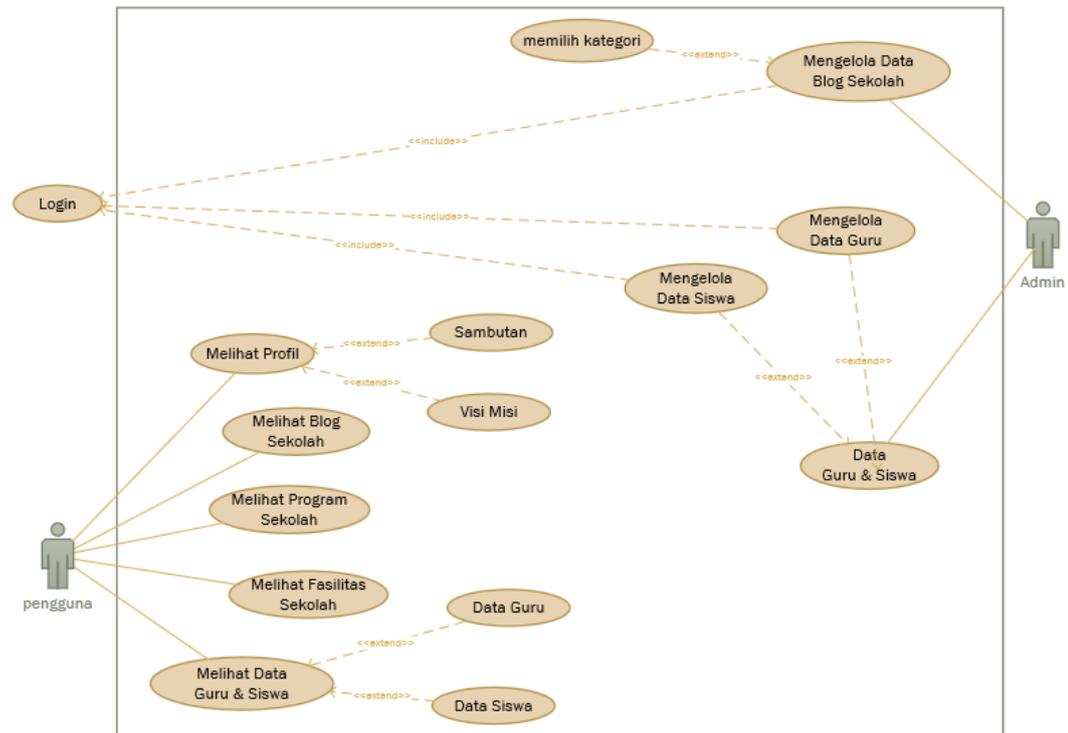
8. Melihat data siswa, berupa jumlah siswa setiap tahun ajaran
- b. Kebutuhan Administrator
1. Melakukan *login* ke dalam sistem
 2. Mengelola data blog
 3. Mengelola kategori
 4. Mengelola data guru
 5. Mengelola data siswa, seperti menambah tahun ajaran, menambahkan kelas baru, dan mengubah jumlah siswa setiap kelas untuk 1 tahun ajaran.
 6. Melakukan *logout*

3.5. Desain Sistem

Setelah melakukan analisis terhadap sistem, sesuai dengan metode SDLC maka hal berikutnya yang harus kita lakukan adalah membuat rancangan atau desain sistem. Dalam pengembangan aplikasi ini beberapa rancangan yang dibuat adalah *use case diagram*, *entity relationship diagram*, *activity diagram*, *class diagram* dan desain *user interface*.

3.5.1 Use Case Diagram

Use case diagram merupakan sebuah teknik yang digunakan dalam pengembangan sebuah sistem informasi untuk menangkap kebutuhan fungsional dari sistem yang bersangkutan. *Use case diagram* menjelaskan interaksi yang terjadi antara *user* dengan sistem yang direpresentasikan dengan urutan langkah yang sederhana. *Use case diagram* dalam laporan ini akan melibatkan 2 aktor, yaitu: pengguna akhir dan administrator sistem.



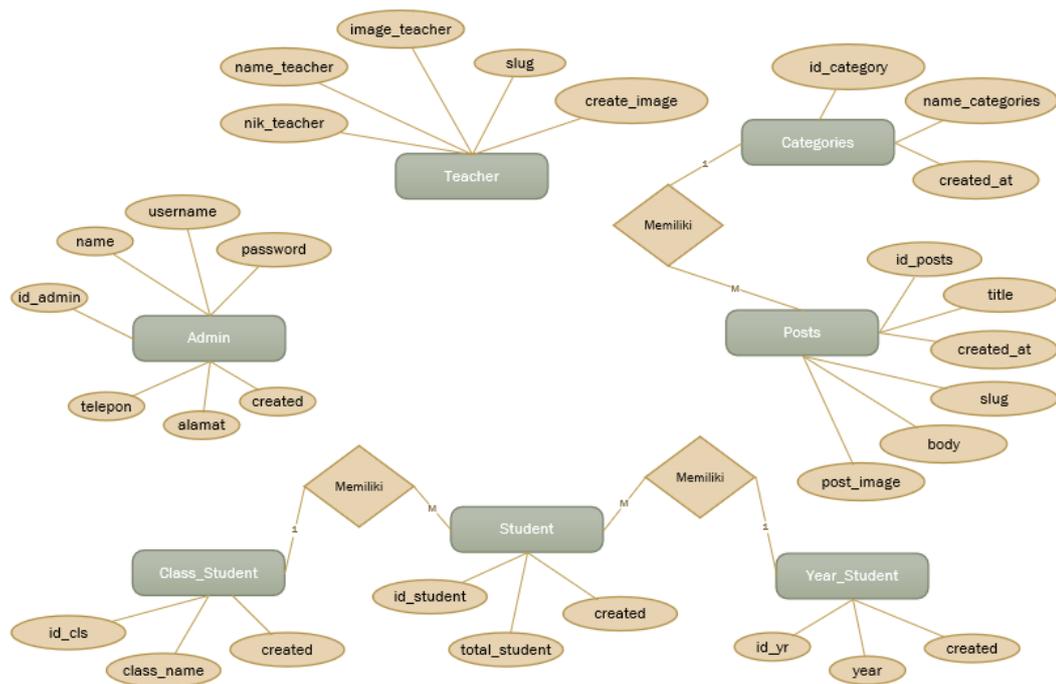
Gambar 3. 2 Use Case Diagram

Use case diagram pengguna akhir memiliki aktor utama pengguna akhir yang dilambangkan dengan simbol orang dengan label pengguna dibawahnya. Dari diagram diatas diperlihatkan bahwa pengguna dapat melakukan beberapa tindakan seperti: melihat profil (sambutan & visi misi), melihat blog sekolah, melihat program sekolah, melihat fasilitas sekolah, melihat data guru & siswa (data guru, data siswa).

Use case diagram administrator sistem memiliki aktor utama administrator sistem yang dilambangkan dengan simbol orang dengan label *admin* dibawahnya. Diagram diatas menggambarkan tentang apa saja tindakan yang dapat dilakukan oleh administrator sistem, yaitu: mengelola data blog sekolah, data guru dan siswa (mengelola data guru, mengelola data siswa), dimana semua proses tersebut hanya dapat dilakukan setelah administrator sistem melakukan proses login terlebih dahulu.

3.5.2 Entity Relationship Diagram (ERD)

Entity Relationship Diagram atau disingkat ERD adalah suatu model yang berfungsi untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi.



Gambar 3. 3 Entity Relationship Diagram

Gambar 3.3 adalah *entity relationship diagram* TPA-KB-TKIT Nurul Ittihad yang menunjukkan entitas dengan atribut dan relasinya. Entitas ini yang nantinya akan menjadi rancangan tabel pada *database* TPA-KB-TKIT Nurul Ittihad. Hubungan antar *entity* adalah sebagai berikut:

1. *Entitas Relationship Diagram* aplikasi TPA-KB-TKIT Nurul Ittihad memiliki 7 buah entitas yaitu:
 - a. *Admin*
 - b. *Teacher*
 - c. *Posts*
 - d. *Categories*
 - e. *Student*
 - f. *Class_student*

g. *Year_student*

2. Entitas *admin* merupakan entitas yang kuat karena entitas *admin* tidak bergantung pada entitas lain.
3. Entitas *teacher* merupakan entitas yang kuat karena entitas *admin* tidak bergantung pada entitas lain.
4. Entitas *categories* dengan entitas *posts* menggunakan relasi *one-to-many*.
5. Entitas *class_student* dengan entitas *student* menggunakan relasi *one-to-many*.
6. Entitas *student* dengan entitas *year_student* menggunakan relasi *many-to-one*.

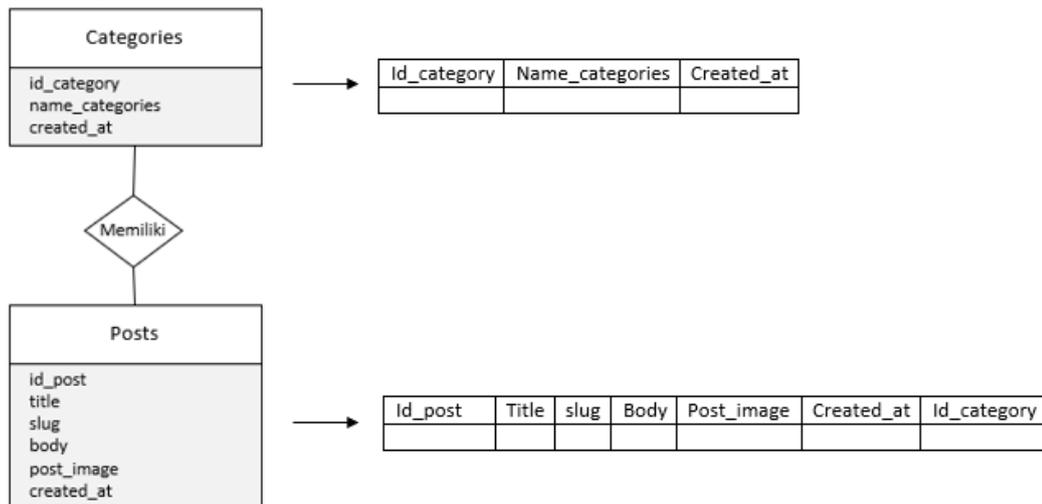
3.5.3 Konversi ERD ke bentuk Tabel

Konversi ERD ke bentuk tabel adalah proses memasukkan entitas menjadi nama tabel dan *attributes* menjadi nama kolom beserta relasinya ke dalam setiap tabel yang saling berhubungan.

1. Konversi dengan relasi *one to many*

Konversi entitas yang memiliki relasi *one to many* (1-M) adalah atribut kunci pada entitas yang memiliki kardinal satu (*one*), akan menjadi *foreign key* pada entitas yang memiliki kardinal banyak (*many*). Pada ERD yang ditunjukkan pada gambar 3.3 ditunjukkan terdapat relasi *one to many* yaitu antara entitas *categories* dengan entitas *posts*, entitas *class_stident* dengan entitas *student* dan *year_student* dengan entitas *student*. Dari entitas tersebut akan direpresentasikan kedalam tabel, dapat dilihat pada gambar 3.4 sebagai berikut.

Entitas *Categories* dengan Entitas *Posts*

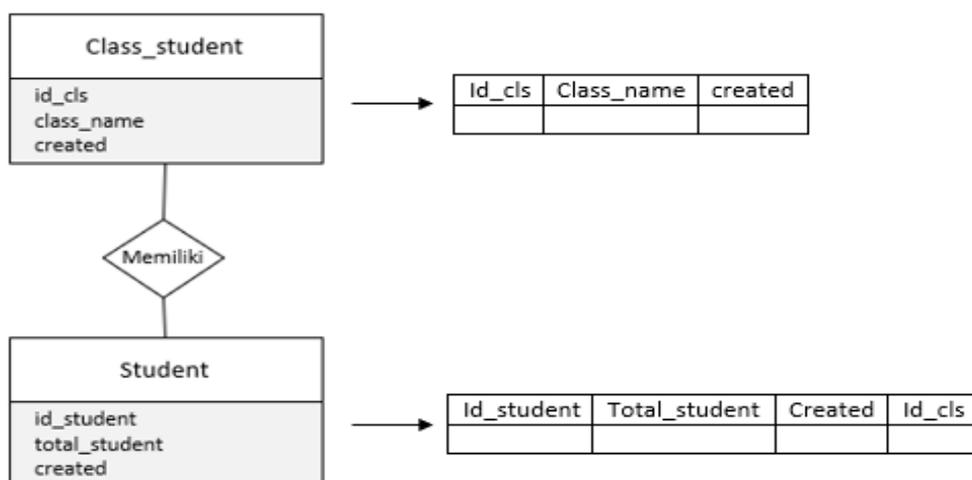


Gambar 3. 4 Relasi Entitas *Categories* dengan Entitas *Posts*

Penjelasan konversi relasi entitas *categories* dengan entitas *posts* pada gambar 3.4 adalah sebagai berikut.

Entitas *categories* memiliki relasi *one to many* dengan entitas *posts* yang artinya satu *categories* dapat memiliki lebih dari satu *posts* maka kolom `id_category` yang ada pada tabel *categories* perlu ditambahkan ke tabel *posts* yang berfungsi sebagai *foreign key* pada tabel *posts*.

Entitas *Class_Student* dengan Entitas *Student*

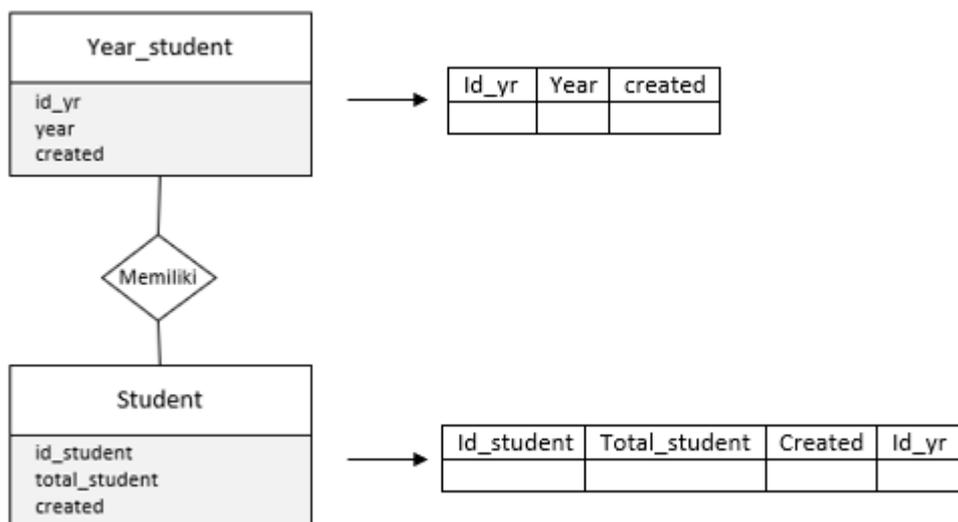


Gambar 3. 5 Entitas *Class_Student* dengan Entitas *Student*

Penjelasan konversi relasi entitas *class_student* dengan entitas *student* pada gambar 3.5 adalah sebagai berikut.

Entitas *class_student* memiliki relasi *one to many* dengan entitas *student* yang artinya satu *class_student* dapat memiliki lebih dari satu *student* maka kolom *id_cls* yang ada pada tabel *class_student* perlu ditambahkan ke tabel *student* yang berfungsi sebagai *foreign key* pada tabel *student*.

Entitas *Year_Student* dengan Entitas *Student*



Gambar 3. 6 Entitas *Year_Student* dengan Entitas *Student*

Penjelasan konversi relasi entitas *year_student* dengan entitas *student* pada gambar 3.6 adalah sebagai berikut.

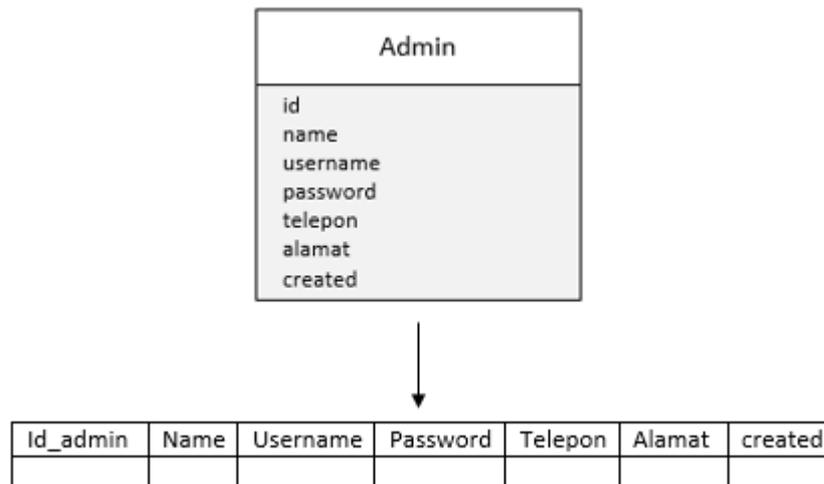
Entitas *year_student* memiliki relasi *one to many* dengan entitas *student* yang artinya *year_student* dapat memiliki lebih dari satu *student* maka kolom *id_yr* yang ada pada tabel *year_student* perlu ditambahkan ke tabel *student* yang berfungsi sebagai *foreign key* pada tabel *student*.

2. Konversi Entitas Kuat

Konversi entitas kuat (*strong entity*) adalah entitas yang dicantumkan secara jelas dalam ERD, dan tidak memiliki ketergantungan terhadap

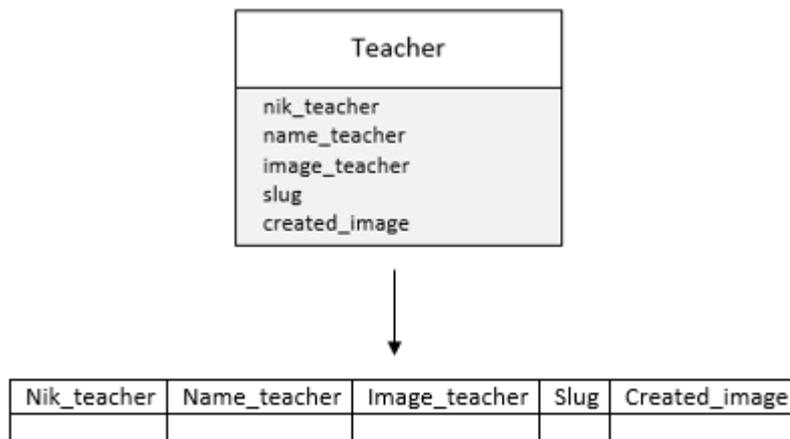
entitas lain. Pada gambar 3.3 terdapat entitas kuat yaitu entitas *admin* dan entitas *teacher* yang dapat dilihat pada gambar 3.7 dan gambar 3.8.

a. Entitas Kuat *Admin*



Gambar 3. 7 Entitas Kuat *Admin*

b. Entitas Kuat *Teacher*

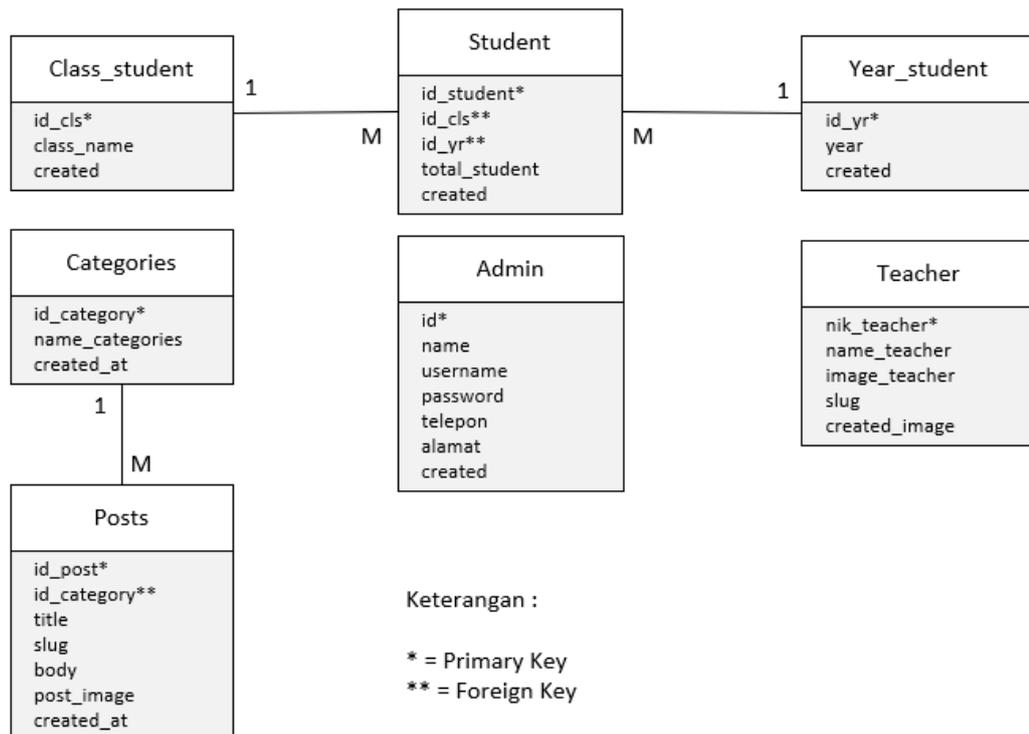


Gambar 3. 8 Entitas Kuat *Teacher*

Pada proses konversi entitas *admin* dan *teacher* dirubah ke dalam bentuk tabel pada gambar 3.7 dan gambar 3.8 dijelaskan bahwa entitas menjadi nama tabel dan *attributes* menjadi nama kolom.

3.5.4 Relasi Antar Tabel

Setelah dilakukan konversi ERD ke bentuk tabel maka akan didapatkan relasi antar tabel yang dapat dilihat pada gambar 3.9 sebagai berikut.



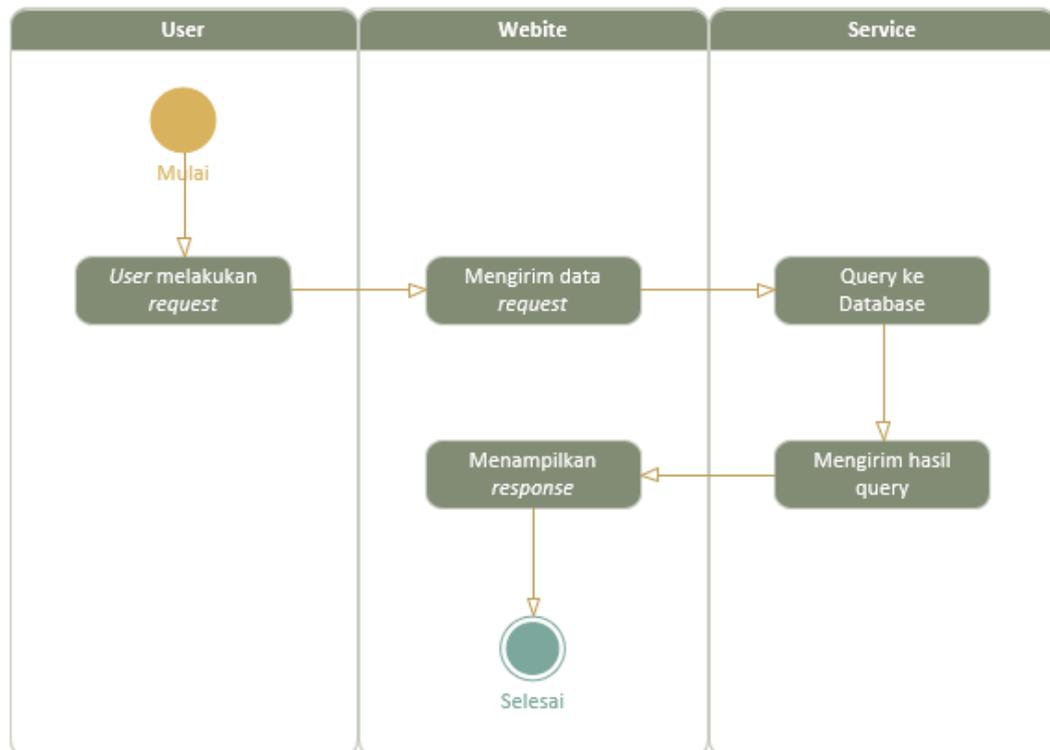
Gambar 3. 9 Relasi Antar Tabel

Penjelasan relasi antar tabel yang telah digambarkan seperti pada gambar 3.18 diatas adalah sebagai berikut:

- Tabel *class_student* memiliki relasi *one to many* dengan tabel *student* untuk menghubungkan tabel *class_student* dengan tabel *student* maka kolom *Id_cls* harus ditambahkan sebagai *foreign key* pada tabel *student*.
- Tabel *year_student* memiliki relasi *one to many* dengan tabel *student* untuk menghubungkan tabel *year_student* dengan tabel *student* maka kolom *id_yr* harus ditambahkan sebagai *foreign key* pada tabel *student*.
- Tabel *categories* memiliki relasi *one to many* dengan tabel *posts* untuk menghubungkan tabel *categories* dengan tabel *posts* maka kolom *id_categories* harus ditambahkan sebagai *foreign key* pada tabel *posts*.

3.5.5 Activity Diagram

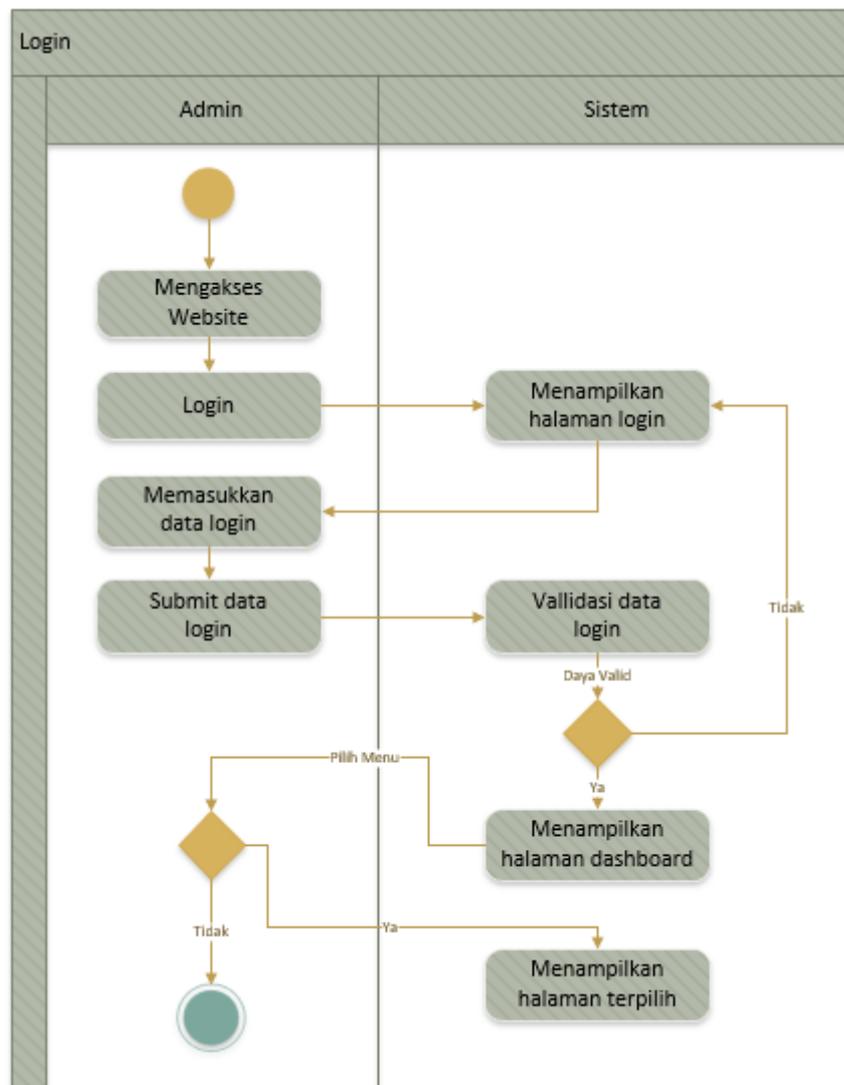
Activity diagram menjelaskan aktivitas-aktivitas yang dilakukan secara sistematis, yang sedang dirancang dan bagaimana masing-masing aliran berawal, keputusan yang mungkin terjadi, dan bagaimana aktivitas tersebut berakhir. Diagram ini mirip dengan *flowchart*.



Gambar 3. 10 Activity Diagram Umum

Activity diagram seperti diatas yang ditunjukkan pada gambar 3.10, dimana user dapat melakukan sebuah *request web* ke halaman *website*. Kemudian hasil *request* tersebut akan di publikasikan ke *website* tadi dan akan menampilkan sebuah *response* dari hasil *request* user tersebut.

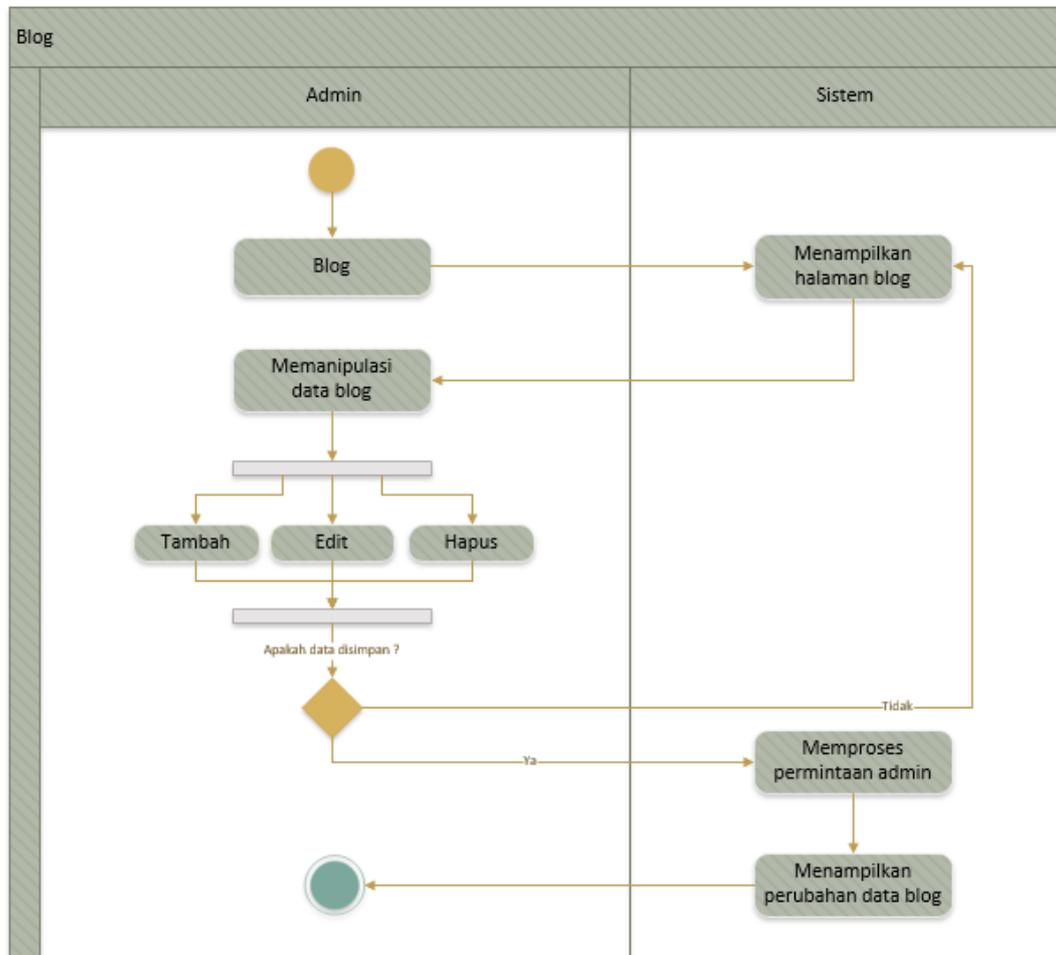
3.5.6 Activity Diagram Login



Gambar 3. 11 Activity Diagram Login

Gambar 3.11 menunjukkan *activity diagram* untuk *login*. Aktor utama dalam diagram aktivitas ini adalah administrator sistem yang selanjutnya akan disebut *admin*. Setelah *admin* mengakses *website* dan memilih halaman login, maka sistem akan menampilkan halaman login terlebih dahulu dilanjutkan dengan *admin* memasukkan data login dan kemudian melakukan submit data. Setelah itu sistem akan melakukan validasi data. Jika data valid maka sistem akan menampilkan halaman dashboard, dan dapat mengakses menu-menu yang ada didalamnya. Sedangkan jika login gagal, maka sistem kembali menampilkan halaman login.

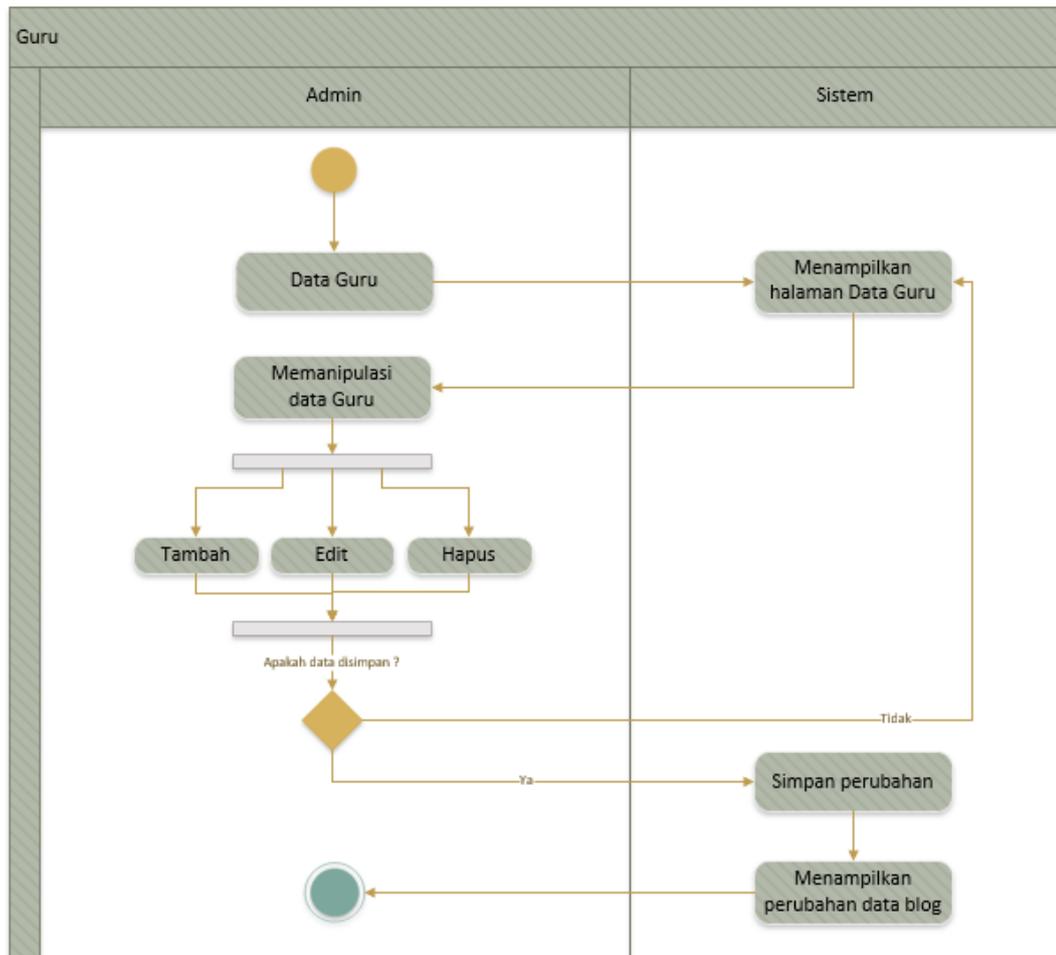
3.5.7 Activity Diagram Blog



Gambar 3. 12 Activity Diagram Blog

Gambar 3.12 menunjukkan *activity diagram* untuk blog. Aktor utama dalam diagram aktivitas ini adalah administrator sistem yang selanjutnya akan disebut *admin*. Setelah *admin login* dan masuk ke halaman blog, *admin* dapat melihat halaman data blog dan *admin* juga dapat mengelola data blog seperti tambah data, *edit* data dan menghapus data blog sesuai dengan kehendak *admin*. Jika proses mengelola data berlanjut maka akan ada 2 pilihan yaitu membatalkan proses dan kembali ke halaman blog atau menyimpan data dilanjutkan perubahan data dihalaman blog.

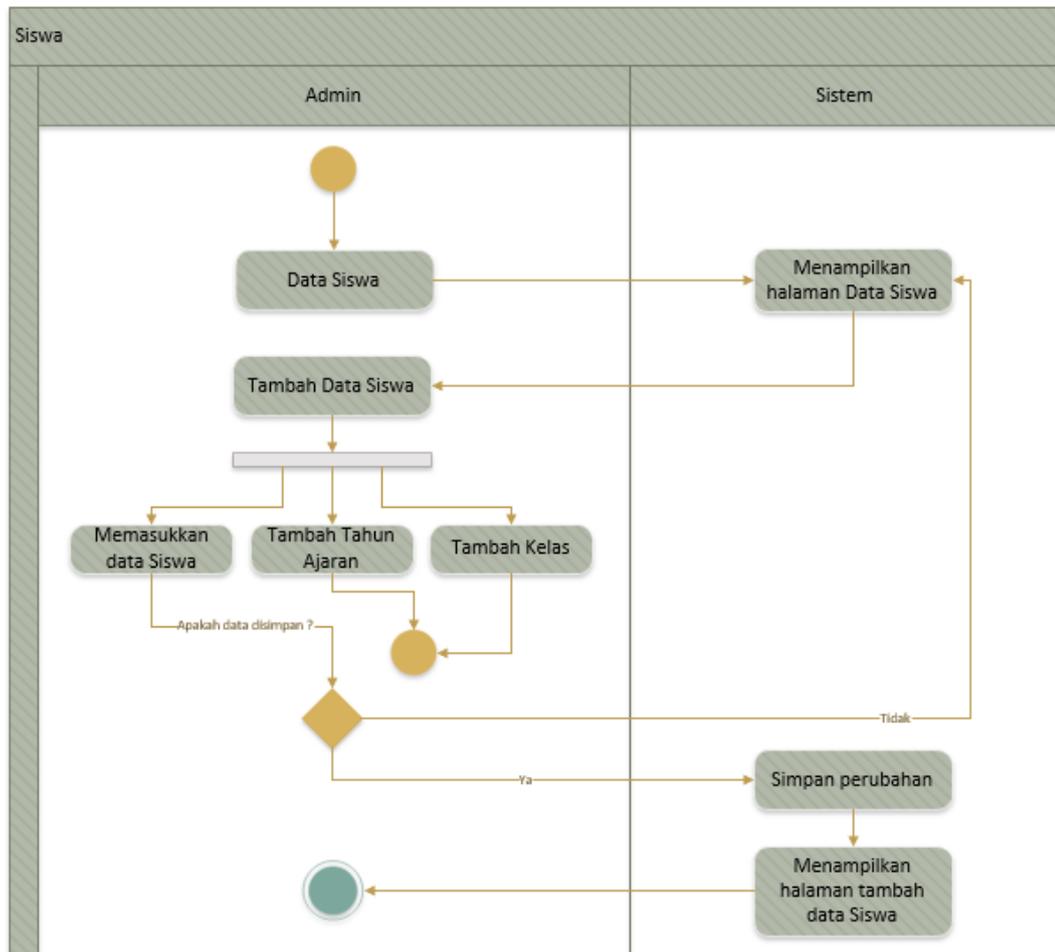
3.5.8 Activity Diagram Data Guru



Gambar 3.13 Activity Diagram Data Guru

Gambar 3.13 menunjukkan *activity diagram* data guru. Aktor utama dalam diagram aktivitas ini adalah administrator sistem yang selanjutnya akan disebut *admin*. Setelah *admin login* dan masuk ke halaman data guru, *admin* dapat melihat halaman data guru dan juga dapat mengelola data guru seperti tambah data, *edit* data dan menghapus data guru sesuai dengan kehendak *admin*. Jika proses mengelola data berlanjut maka akan ada 2 pilihan yaitu membatalkan proses dan kembali ke halaman data guru atau menyimpan data dilanjutnya perubahan dihalaman blog.

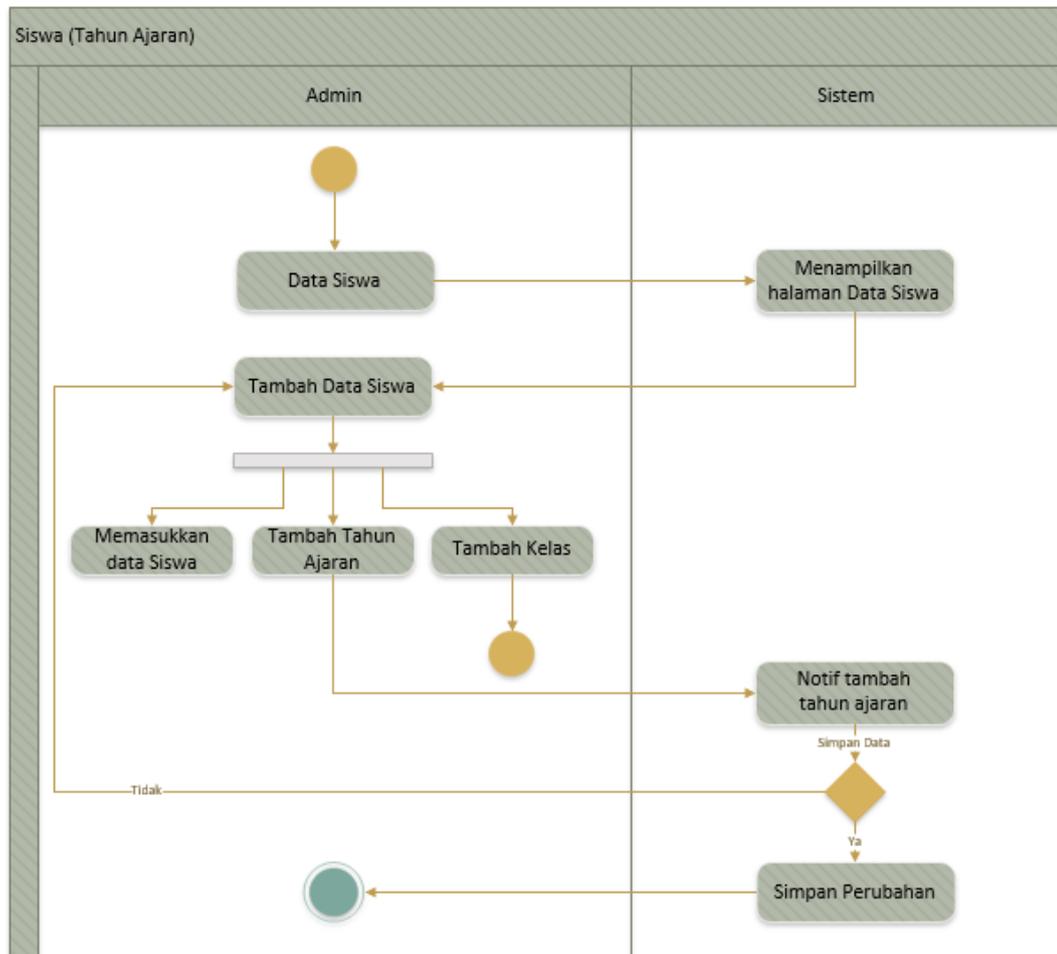
3.5.9 Activity Diagram Data Siswa



Gambar 3. 14 Activity Diagram Data Siswa

Gambar 3.14 menunjukkan *activity diagram* data siswa. Aktor utama dalam diagram aktivitas ini adalah administrator sistem yang selanjutnya akan disebut *admin*. Setelah *admin* login dan masuk ke halaman data siswa, *admin* dapat melihat halaman data siswa dan juga dapat tambah data siswa. Kemudian *admin* memasukkan data siswa hingga muncul pilihan menyimpan atau membatalkan. Jika batal, maka akan kembali ke halaman data siswa, dan jika menyimpan maka data akan di proses oleh sistem dilanjutkan perubahan data dihalaman data siswa.

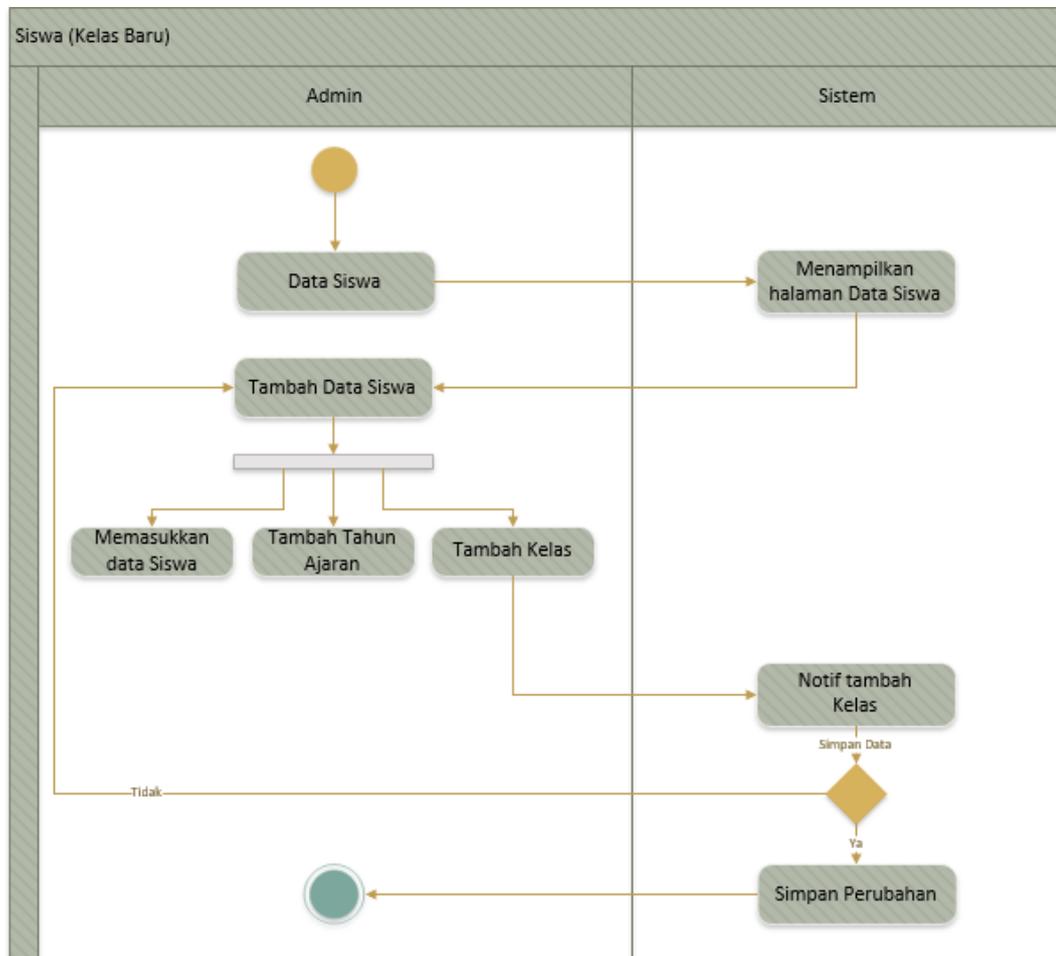
3.5.10 Activity Diagram Tahun Ajaran



Gambar 3. 15 Activity Diagram Tahun Ajaran

Gambar 3.15 menunjukkan *activity diagram* tahun ajaran. Aktor utama dalam diagram aktivitas ini adalah administrator sistem yang selanjutnya akan disebut *admin*. Setelah *admin* login dan masuk ke halaman data siswa, lalu *admin* pergi ke halaman tambah data siswa. Kemudian *admin* memasukkan tambah tahun ajaran hingga muncul notif pilihan menyimpan atau membatalkan. Jika batal, maka akan kembali ke halaman tambah data siswa, dan jika menyimpan maka data akan di proses oleh sistem dilanjutkan tambahan data tahun ajaran dihalaman data siswa.

3.5.11 Activity Diagram Kelas Baru

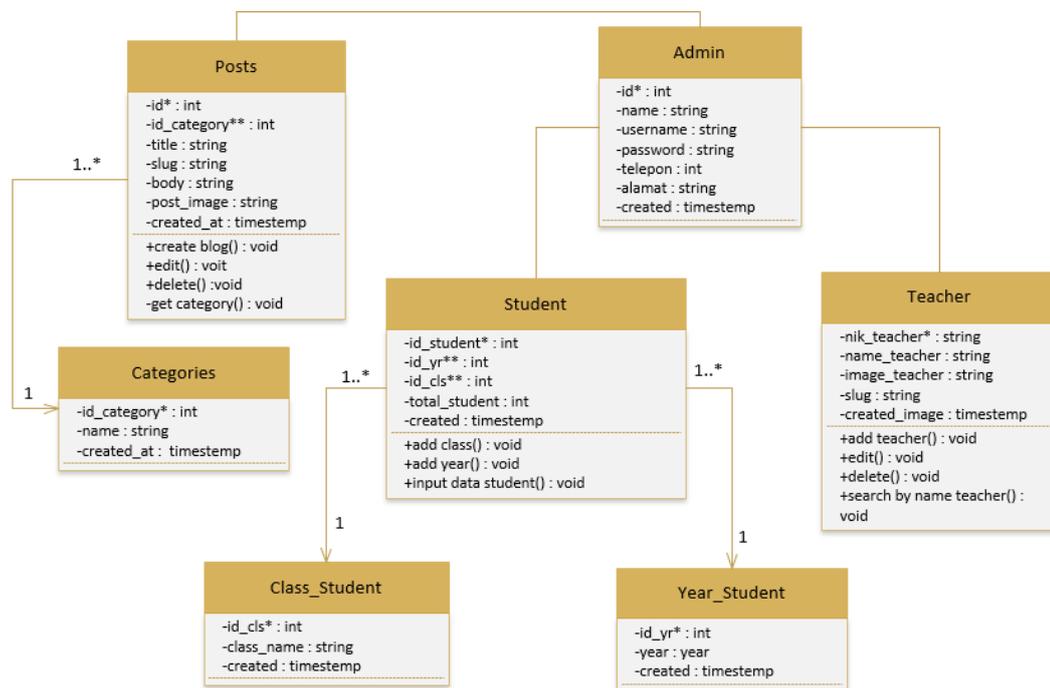


Gambar 3. 16 Activity Diagram Tambah Kelas

Gambar 3.16 menunjukkan *activity diagram* tambah kelas. Aktor utama dalam diagram aktivitas ini adalah administrator sistem yang selanjutnya akan disebut *admin*. Setelah *admin* login dan masuk ke halaman data siswa, lalu admin pergi ke halaman tambah data siswa. Kemudian *admin* memasukkan tambah kelas hingga muncul notif pilihan menyimpan atau membatalkan. Jika batal, maka akan kembali ke halaman tambah data siswa, dan jika menyimpan maka data akan di proses oleh sistem dilanjutkan tambahan data kelas dihalaman data siswa.

3.5.12 Class Diagram

Class diagram atau diagram kelas merupakan inti pemrograman berbasis objek karena diagram ini memberikan pemetaan terhadap kelas-kelas serta paket-paket yang ada dalam sistem atau perangkat lunak yang sedang digunakan oleh suatu aplikasi. Berikut ini merupakan tabel dari kamus basis data (*database*) pada website *TPA-KB-TKIT Nurul Ittihad*:



Gambar 3. 17 *Class Diagram*

Gambar 3.17 diatas menunjukkan *class diagram* *TPA-KB-TKIT Nurul Ittihad*. Berikut penjelasan tentang *class diagram* diatas.

Penjelasan class diagram berdasarkan fungsi :

1. *Class admin* mempunyai fungsi untuk menyimpan seluruh data *admin* yang ada pada sistem. mengelola seluruh data di aplikasi seperti menambah, mengedit, dan menghapus data.
2. *Class teacher* mempunyai fungsi untuk menyimpan data guru pada sistem.
3. *Class posts* merupakan halaman blog dari *TPA-KB-TKIT Nurul Ittihad* yang berfungsi untuk memasukkan data tentang informasi kegiatan-

kegiatan yang akan dilakukan baik informasi akademik maupun non-akademik.

4. *Class categories* dipakai di *class posts* yang berfungsi untuk menentukan informasi akademik atau non-akademik di blog *TPA-KB-TKIT Nurul Ittihad*.
5. *Class student* mempunyai fungsi untuk memasukkan data informasi jumlah siswa setiap tahun ajaran.
6. *Class year_student* dipakai di *class student* yang berfungsi untuk menentukan tahun ajaran di *TPA-KB-TKIT Nurul Ittihad*.
7. *Class class_student* juga dipakai di *class student* yang berfungsi untuk menentukan kelas yang ada di *TPA-KB-TKIT Nurul Ittihad*.

Penjelasan *class diagram* berdasarkan relasi antar kelas :

1. *Class categories* dengan *class posts* menggunakan relasi *one-to-many* dengan simbol relasi asosiasi berarah (*directed association*).
2. *Class class_student* dengan *class student* menggunakan relasi *one-to-many* dengan simbol relasi asosiasi berarah (*directed association*).
3. *Class year_student* dengan *class student* menggunakan relasi *one-to-many* dengan simbol relasi asosiasi berarah (*directed association*).
4. *Class admin* dengan *class posts* memiliki relasi antar kelas bermakna umum (*association*), artinya *class admin* digunakan untuk dapat menjalankan *class posts*.
5. *Class admin* dengan *class student* memiliki relasi antar kelas bermakna umum (*association*), artinya *class admin* digunakan untuk dapat menjalankan *class student*.
6. *Class admin* dengan *class teacher* memiliki relasi antar kelas bermakna umum (*association*), artinya *class admin* digunakan untuk dapat menjalankan *class teacher*.

3.5.13 Kamus Database

Berikut ini merupakan tabel-tabel dari kamus *database* pada “Aplikasi Sistem Informasi Web *TPA-KB-TKIT Nurul Ittihad* Menggunakan Framework Codeigniter”.

1. Kamus Data *Student*

Kamus data tabel *student* yang digunakan dalam aplikasi desktop dapat dilihat pada tabel 3.1.

Tabel 3. 1 Kamus Data Tabel *Student*

Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
Id_Student	Int	11	<i>Primary Key</i>	ID Siswa
Id_Yr	Int	3	<i>Foreign Key</i>	ID Tahun
Id_Cls	Int	3	<i>Foreign Key</i>	ID Kelas
Total_students	Int	3		Total Siswa
Created	Timestamp	-		Waktu Pembuatan

2. Kamus Data *Year_Student*

Kamus data tabel *year_student* yang digunakan dalam aplikasi desktop dapat dilihat pada tabel 3.2.

Tabel 3. 2 Kamus Data Tabel *Year_Student*

Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
Id_Yr	Int	3	<i>Primary Key</i>	ID Siswa
Year_student	Year	4	-	Tahun Ajaran
Created	Timestamp	-		Waktu Pembuatan

3. Kamus Data *Class_Student*

Kamus data tabel *class_student* yang digunakan dalam aplikasi desktop dapat dilihat pada tabel 3.3.

Tabel 3. 3 Kamus Data Tabel *Class_Student*

Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
Id_Cls	Int	3	<i>Primary Key</i>	ID Siswa
Class_Name	Varchar	25	-	Nama Kelas
Created	Timestamp	-		Waktu Pembuatan

4. Kamus Data *Posts*

Kamus data tabel *posts* yang digunakan dalam aplikasi desktop dapat dilihat pada tabel 3.4.

Tabel 3. 4 Kamus Data *Tabel Posts*

Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
Id	Int	5	<i>Primary Key</i>	ID Siswa
Id_Category	Int	1	<i>Foreign Key</i>	Nama Kelas
Title	Varchar	30		Judul Blog
Slug	Varchar	30		Menyederhanakan URL Blog
Body	Text	-		Isi Blog
Post_Image	Varchar	255		Gambar Blog
Created	Timestamp	-		Waktu Pembuatan

5. Kamus Data *Categories*

Kamus data tabel *categories* yang digunakan dalam aplikasi desktop dapat dilihat pada tabel 3.5.

Tabel 3. 5 Kamus Data Tabel *Categories*

Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
Id_Category	Int	1	<i>Primary Key</i>	ID Kategori
Name	Varchar	25		Nama Kategori
Created	Timestamp	-		Waktu Pembuatan

6. Kamus Data *Teacher*

Kamus data tabel *teacher* yang digunakan dalam aplikasi desktop dapat dilihat pada tabel 3.6.

Tabel 3. 6 Kamus Data Tabel *Teacher*

Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
NIK_Teacher	Varchar	15	<i>Primary Key</i>	ID Kategori
Name_Teacher	Varchar	25		Nama Guru
Image_Teacher	Varchar	255		Gambar Guru
Slug	Varchar	25		Menyederhanakan URL Guru
Created_Image	Timestamp	-		Waktu Pembuatan

7. Kamus Data *Admin*

Kamus data tabel *admin* yang digunakan dalam aplikasi desktop dapat dilihat pada tabel 3.1.

Tabel 3. 7 Kamus Data Tabel *Admin*

Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
Id	Int	15	<i>Primary Key</i>	ID Admin
Name	Varchar	25		Nama Admin
Username	Varchar	25		Username Admin
Password	Varchar	10		Password Admin
Telepon	Varchar	12		Telepon Admin
Alamat	Text	-		Alamat Admin
Created	Timestamp	-		Waktu Pembuatan

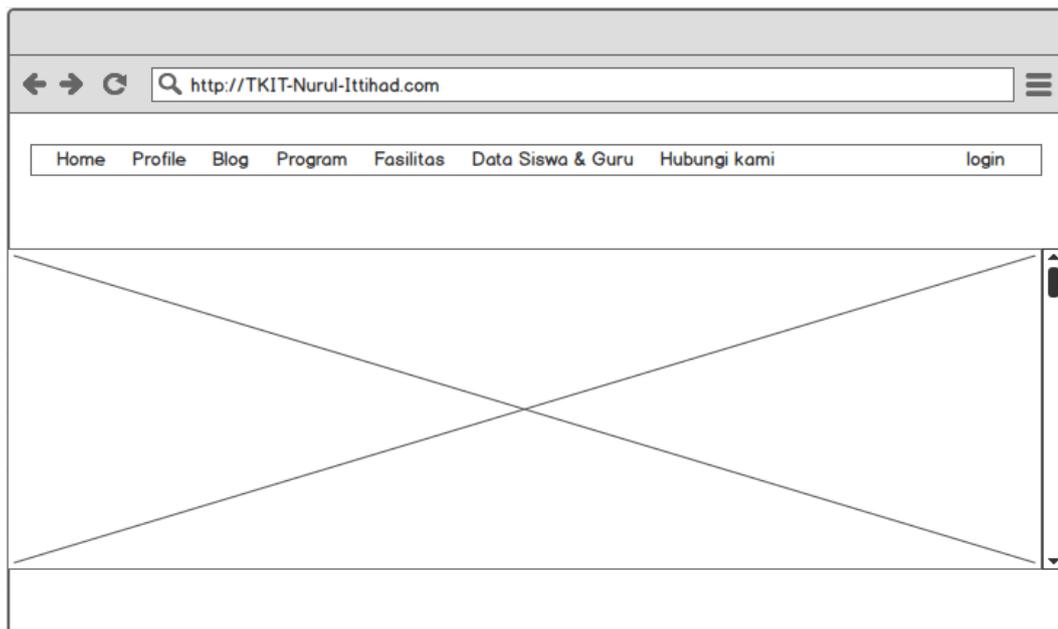
3.6. Desain User Interface

User interface merupakan bagian yang sangat penting pada suatu aplikasi, karena bagian ini yang menjembatani antarmuka antara pengguna dengan sistem. Desain user interface yang baik akan memudahkan pengguna dalam menggunakan aplikasi yang dibuat.

Dalam laporan ini desain *user interface* dibagi menjadi 2 bagian, yaitu: *user interface* pengguna akhir dan administrator sistem.

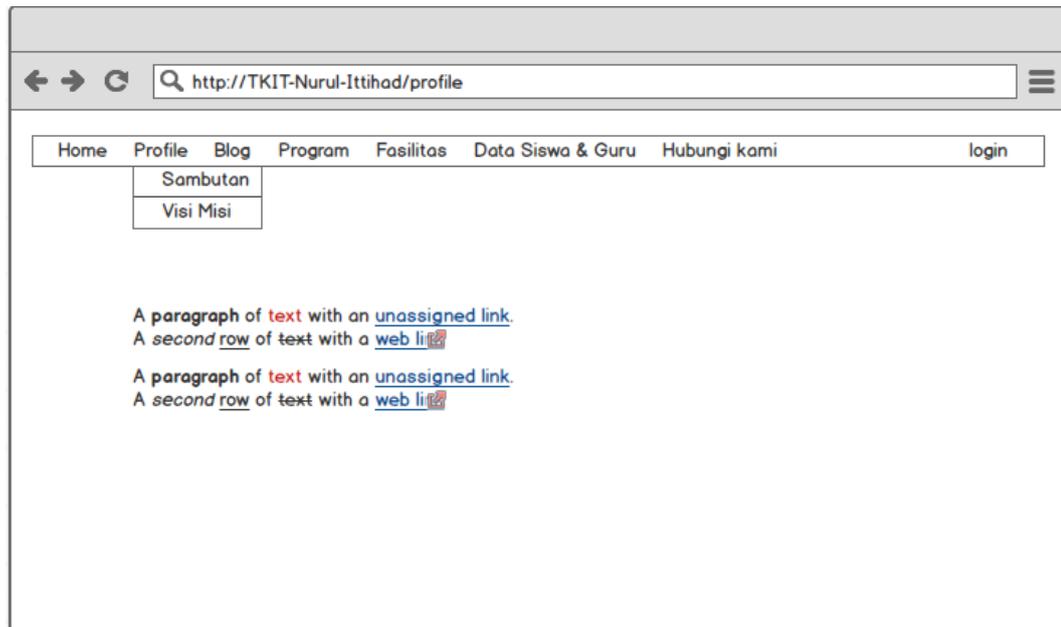
3.6.1 Desain *User Interface* Pengguna Akhir

Halaman awal adalah halaman utama yang menjadi landasan saat pengguna mengakses website *TPA-KB-TKIT Nurul Ittihad*. Berikut tampilan halaman utama dapat dilihat pada gambar dibawah ini.



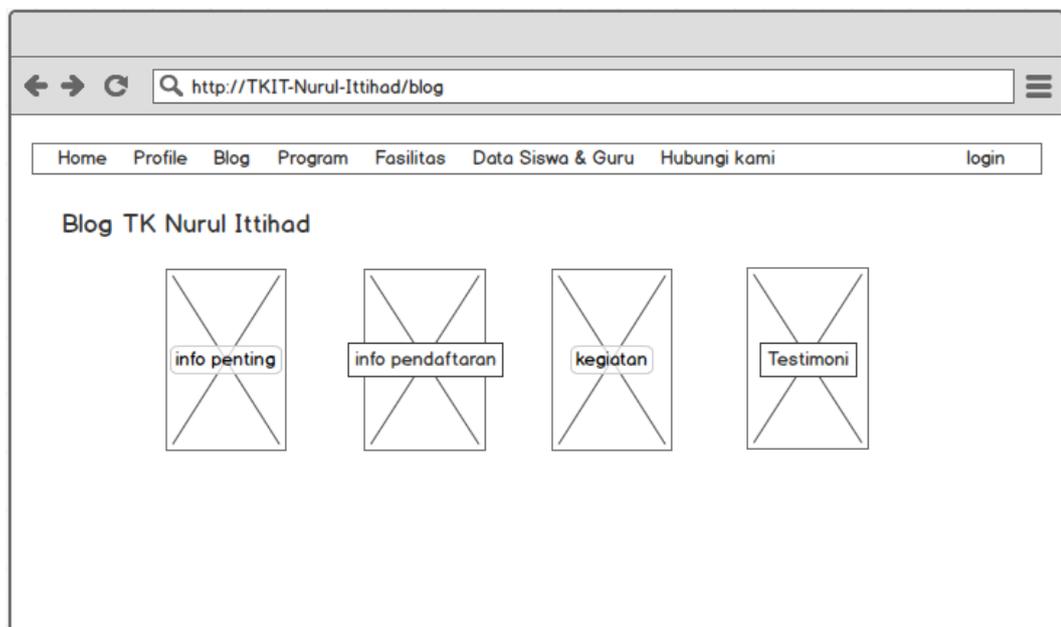
Gambar 3. 18 Desain Halaman Awal

Kemudian halaman selanjutnya adalah halaman *profile* (sambutan dan visi misi) yang memiliki struktur dimulai dari *header*, isi dan *footer*.



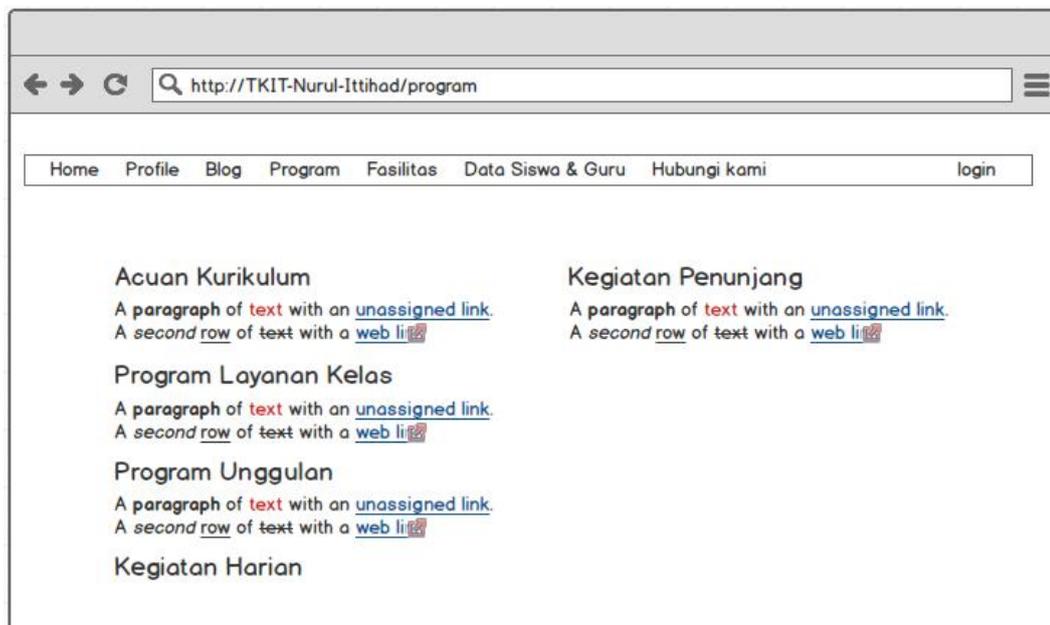
Gambar 3. 19 Rancangan Tampilan *Profile*

Setelah halaman *profile* (sambutan dan visi misi) ada halaman blog yang memiliki struktur dimulai dari *header*, blog TK berupa gambar dan deskripsi serta *footer*.



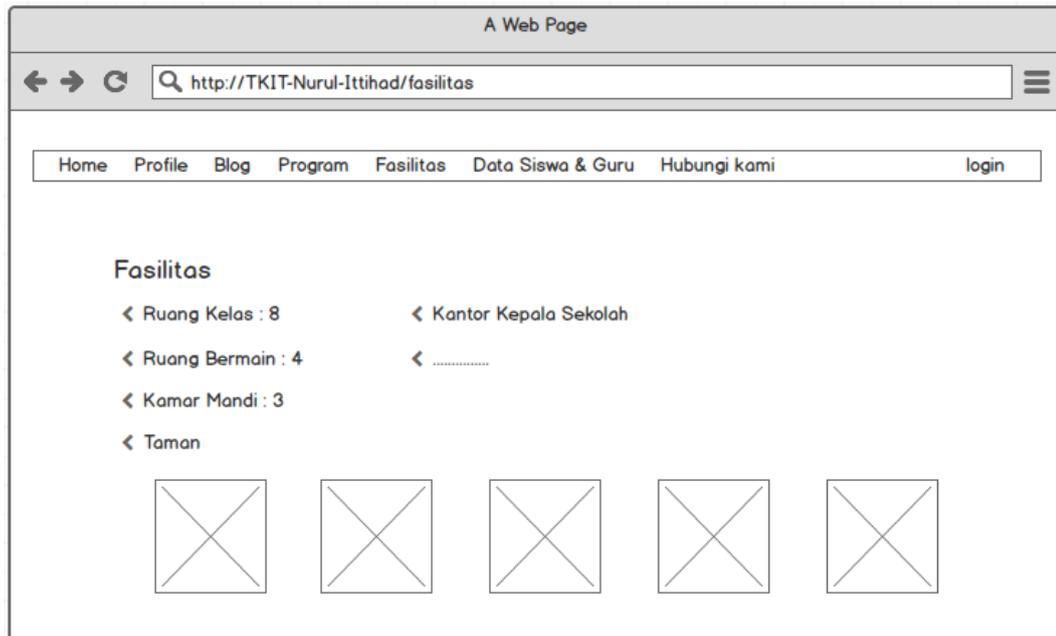
Gambar 3. 20 Desain Blog Sekolah

Halaman selanjutnya adalah halaman program yang memuat informasi-informasi tentang program sekolah. Struktur halaman ini dimulai dari *header*, isi teks dan *footer*.



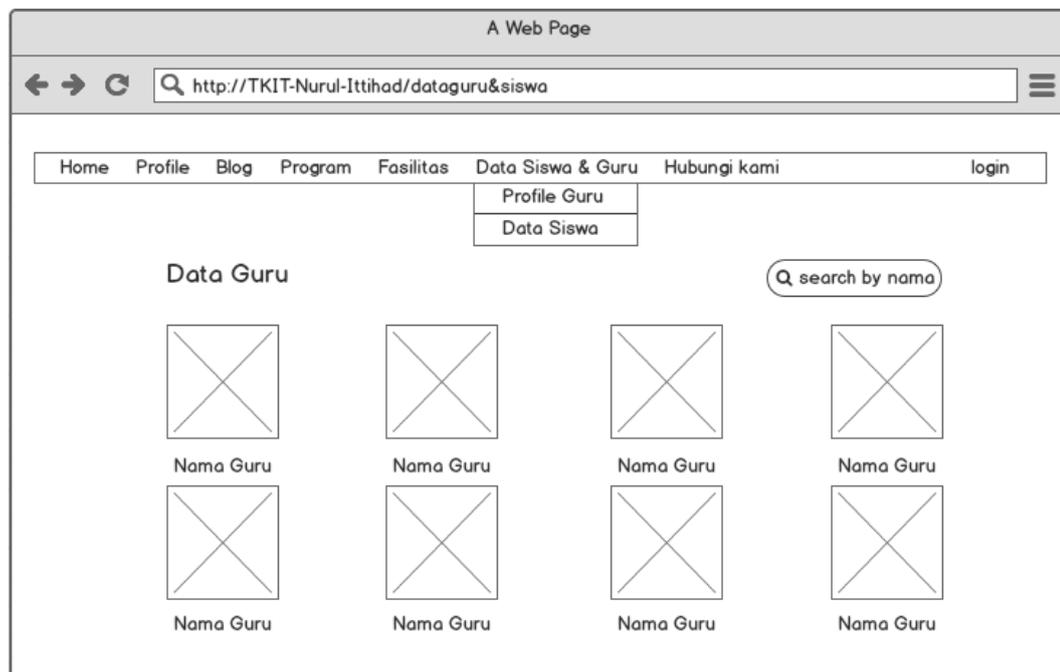
Gambar 3. 21 Desain Program Sekolah

Desain *user interface* selanjutnya adalah desain halaman fasilitas yang dimulai dari *header*, isi teks, gambar dan *footer*.



Gambar 3. 22 Desain Fasilitas Sekolah

Selanjutnya yaitu halaman data guru yang berisi informasi-informasi data guru berupa gambar dan foto. Strukturnya dimulai dari *header*, teks dan gambar, serta *footer*.



Gambar 3. 23 Desain Tampilan Data Guru

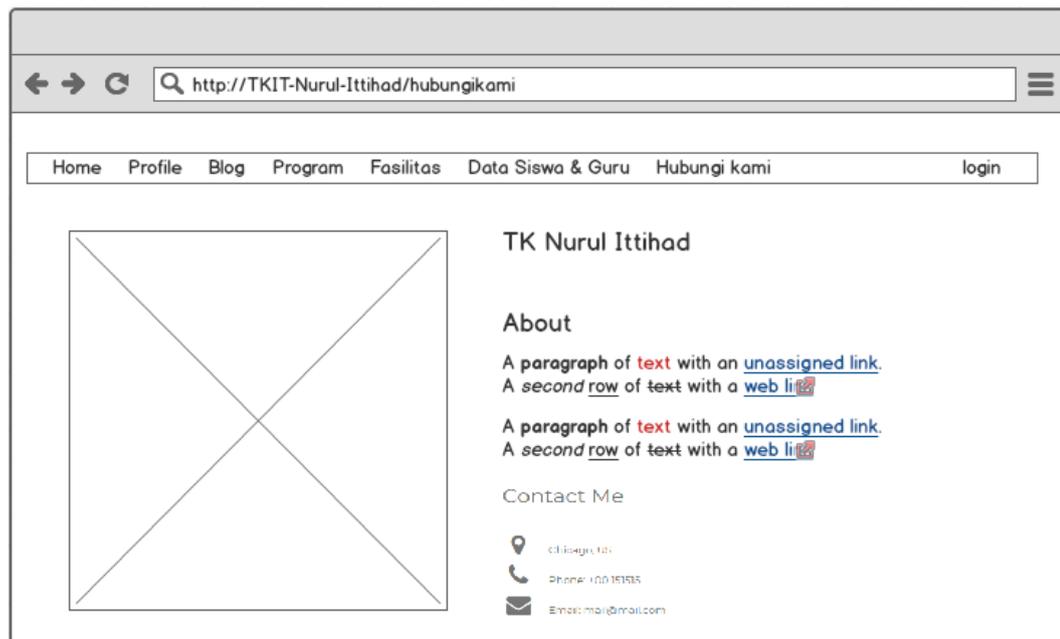
Setelah halaman data guru ada halaman data siswa yang berisi informasi-informasi jumlah siswa setiap tahun angkatan. Strukturnya dimulai dari *header* dan informasi siswa.

The screenshot shows a web browser window with the URL `http://TKIT-Nurul-Ittihad/dataguru&siswa`. The navigation menu includes: Home, Profile, Blog, Program, Fasilitas, Data Siswa & Guru, Hubungi kami, and login. Under 'Data Siswa & Guru', there are sub-menus for 'Data Guru' and 'Data Siswa'. The main content area is titled 'Data Siswa' and features a list of 'Tahun Ajaran' (Academic Years) with radio buttons: 2017/2018 (selected), 2016/2017, 2015/2016, 2014/2015, and 2013/2014. To the right, a table displays 'Data Siswa tahun ajaran 2017/2018'.

TPA	KB1	KB2	TKA1	TKA2	TKB1	TKB2	TKB3
9	9	18	13	14	16	16	16

Gambar 3. 24 Desain Tampilan Data Siswa

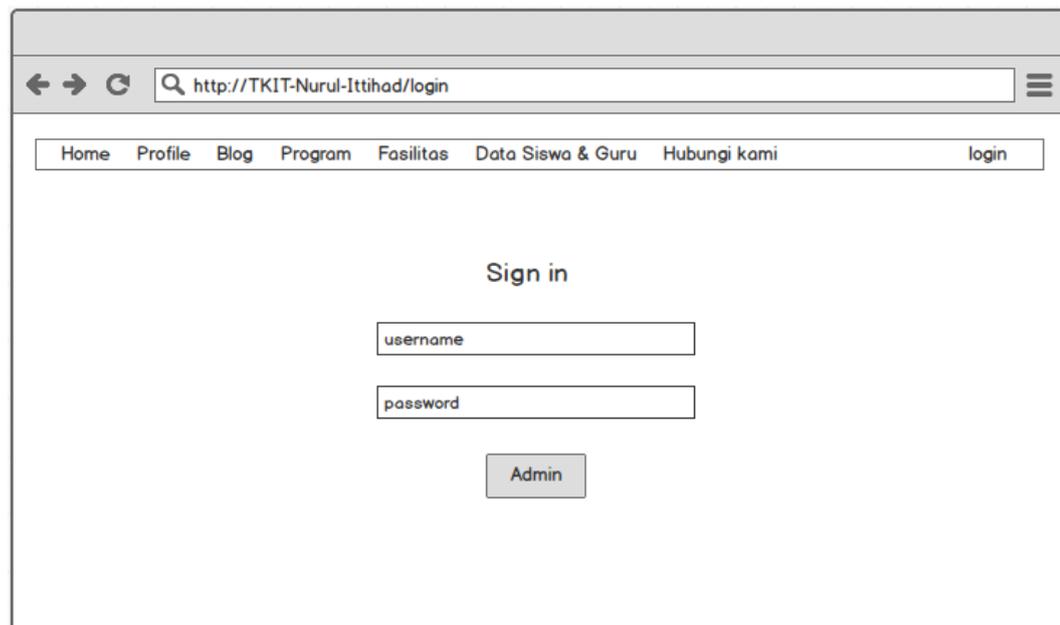
Halaman terakhir adalah halaman hubungi kami yang berisi informasi-informasi tentang situs. Strukturnya dimulai dari *header*, blok informasi dan *footer*.



Gambar 3. 25 Rancangan Tampilan Hubungi Kami

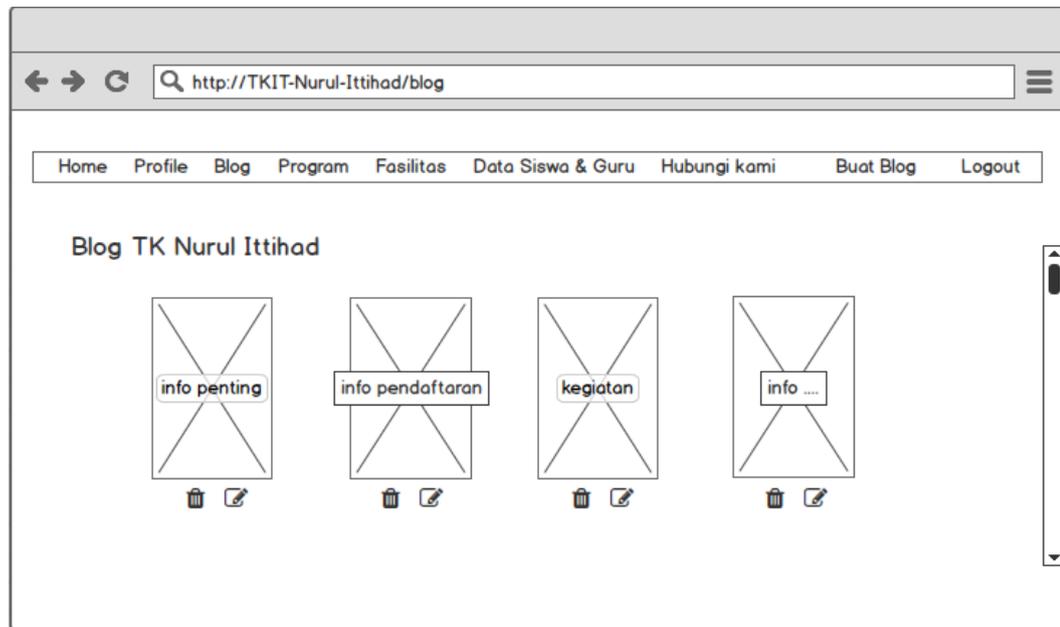
3.6.2 Desain *User Interface* Administrator Sistem

Untuk *user interface admin* terdiri dari beberapa bagian, dimulai dari halaman login, halaman blog, halaman data guru dan halaman data siswa. Berikut tampilan halaman dapat dilihat pada gambar dibawah ini.



Gambar 3. 20 Desain Halaman *Login*

Gambar diatas adalah desain halaman login yang berisi *field* untuk memasukkan *username*, *password*, serta tombol *admin*. Desain selanjutnya adalah halaman blog yang gambarnya sebagai berikut:



Gambar 3. 26 Desain Halaman Mengelola Blog

Desain berikutnya adalah halaman mengelola blog dimana *admin* dapat menambahkan blog baru, *edit* data blog dan hapus data blog.

Gambar 3. 27 Desain Halaman Blog Baru

Halaman blog baru berfungsi untuk menambahkan data blog baru ke halaman blog yang terdiri dari *title*, *body*, kategori, *upload* gambar dan *button submit*.

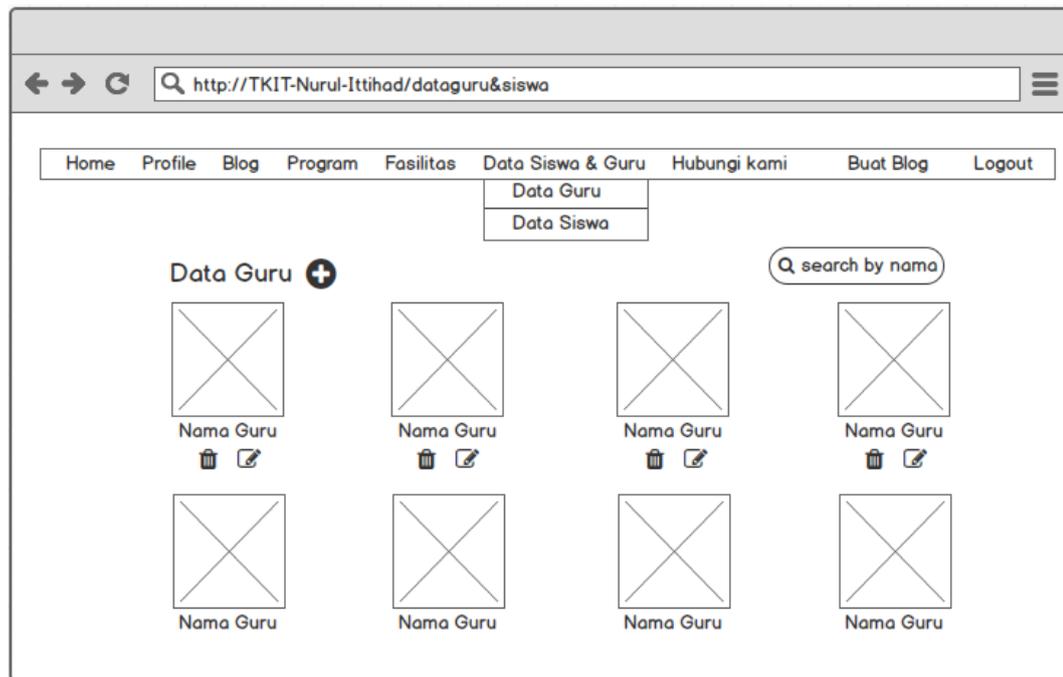
TPA	KB1	KB2	TKA1	TKA2	TKB1	TKB2	TKB3
9	9	18	13	14	16	16	16

Gambar 3. 28 Desain Halaman Mengelola Data Siswa

Gambar diatas adalah halaman data siswa yang berfungsi untuk mengelola data siswa, sesuai dengan tahun angkatan.

Gambar 3. 29 Desain Halaman Tambah Data Siswa

Halaman tambah data siswa adalah halaman yang akan tampil apabila *admin* mengklik icon tambah data pada halaman data siswa. Terdapat 2 *button* di atas, yang satu untuk menambahkan kelas baru dan yang kedua untuk menambah tahun angkataan. Sedangkan di bawahnya terdapat 2 *ComboBox*, yang pertama untuk menampilkan data tahun angkatan yang sudah tersimpan ke *database*, yang kedua untuk menampilkan data kelas yang telah tersimpan di *database* dan pilihan yang ketiga adalah *text input* untuk memasukkan jumlah siswa. *Button submit* paling bawah berfungsi untuk menyimpan data siswa kedalam *database*.



Gambar 3. 30 Desain Halaman Data Guru Setelah *Login*

Gambar diatas adalah halaman terakhir yang bisa dikelola oleh *admin*. Strukturnya ialah terdapat *icon +* berfungsi menambahkan data guru, *search box* untuk mencari data guru, *icon edit* dan yang terakhir *icon hapus*.