BAB III
ANALISIS DAN PERANCANGAN SISTEM

III.1. Analisis Sistem

Analisis sistem digunakan untuk mengindentifikasi masalah yang akan timbul pada saat pembangunan sistem, hal ini berguna untuk membantu pada saat proses selanjutnya, yaitu proses perancangan sistem. Dalam analisis ini terdapat beberapa bagian di antaranya, yaitu:

1. Analisis masalah
2. Analisis prosedur yang sedang berjalan
3. Analisis aturan bisnis
4. Analisis arsitektur sistem
5. Analisis *file sharing*
6. Analisis kebutuhan non fungsional
7. Analisis kebutuhan fungsional

III.1.1. Analisis Masalah

Permasalahan dari penelitian ini adalah bagaimana mempermudah distribusi materi pembelajaran lalu dari pengajar kepada siswa di setiap sekolah di kabupaten Karawang, Kemudian setiap sekolah di kabupaten Karawang dapat menggunakan layanan ini.

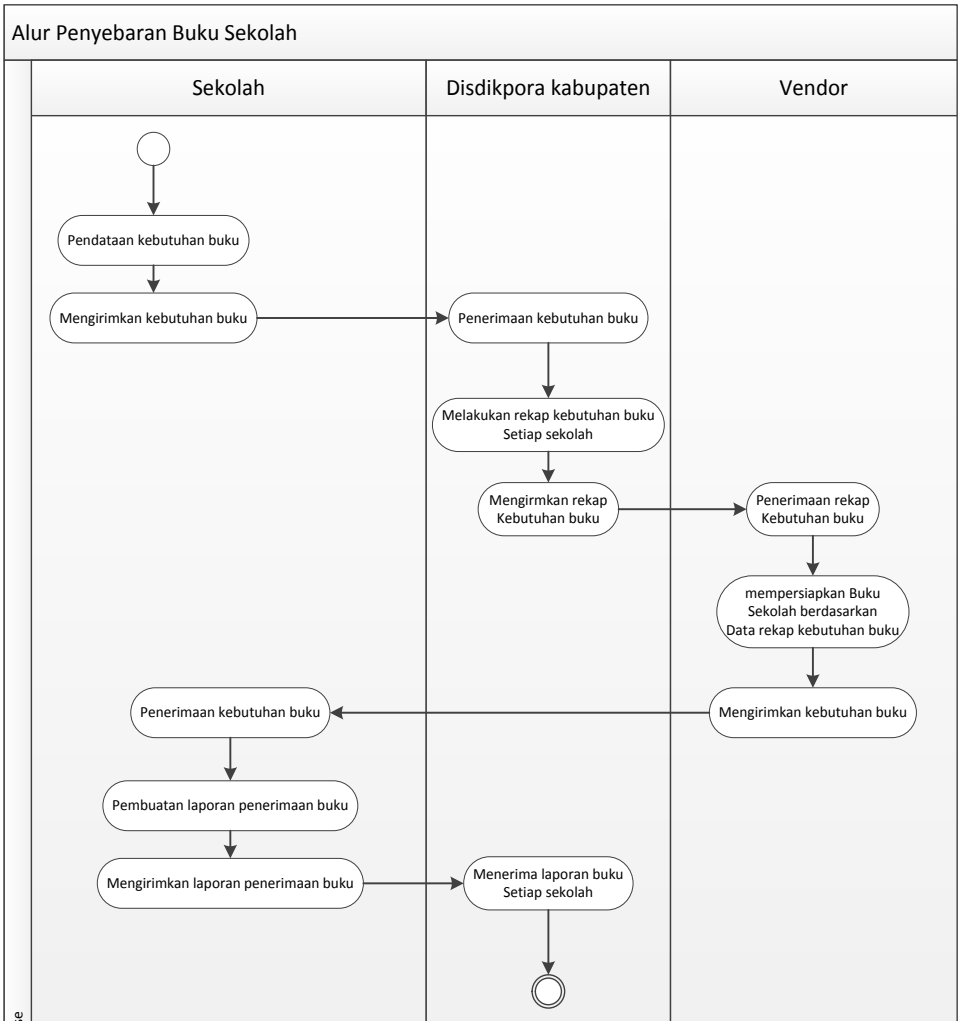
III.1.2. Analisis Prosedur yang Sedang Berjalan

Analisis prosedur yang sedang berjalan di Disdikpora berguna untuk mengetahui sistem penyebaran materi yang ada pada Disdikpora kabupaten Karawang kepada setiap sekolah yang berada di bawah naungan Disdikpora kabupaten Karawang.

1. Alur penyebaran buku sekolah

Alur penyebaran buku yang ada di Disdikpora kabupaten Karawang bermula dari setiap sekolah mendata kebutuhan buku untuk siswa yang ada di sekolah.

Setelah itu sekolah menyerahkan dokumen kebutuhan buku kepada dinas pendidikan. Selanjutnya Disdikpora mengumpulkan atau merekap kebutuhan buku untuk setiap sekolah, hasil dari rekap tersebut kemudian diserahkan kepada vendor atau perusahaan yang menangani pencetakan buku sekolah yang telah memenangkan tender sebelumnya. Setelah vendor menerima dokumen kebutuhan buku setiap sekolah, kemudian mereka mengirimkan buku sekolah kepada setiap sekolah di kabupaten Karawang sesuai dengan dokumen kebutuhan yang telah diterima. Setelah sekolah menerima buku, maka sekolah mengirimkan laporan penerimaan buku kepada dinas pendidikan. Alur dari penyebaran buku sekolah dapat dilihat pada Gambar III-1.

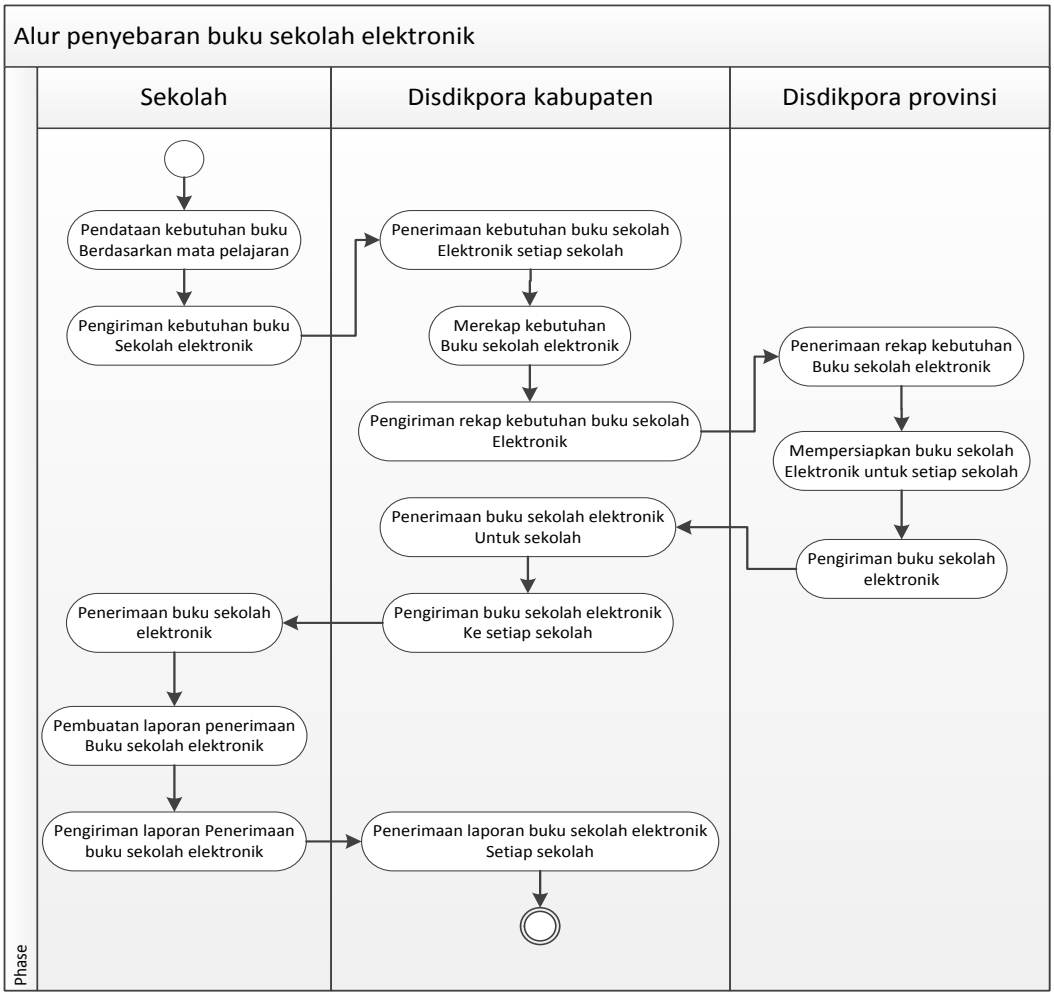


Gambar III-1 Prosedur Penyebaran Buku Sekolah

2. Alur penyebaran Buku Sekolah Elektronik

Alur penyebaran Buku Sekolah Elektronik dimulai dari setiap sekolah mendata kebutuhan buku berdasarkan mata pelajaran yang ada di sekolah tersebut. Setelah itu sekolah mengirimkan dokumen kebutuhan kepada Disdikpora. Selanjutnya Disdikpora akan menyampaikan kebutuhan buku kepada Disdikpora provinsi. Setelah disdikpora provinsi menerima kebutuhan Buku Sekolah Elektronik, maka Disdikpora provinsi akan mengirimkan Buku Sekolah Elektronik kepada Disdikpora kabupaten/kota. Setelah mendapatkan Buku Sekolah Elektronik maka Disdikpora kabupaten akan mengirimkan Buku Sekolah Elektronik dalam media CD/DVD kepada setiap sekolah berdasarkan kebutuhan sekolah tersebut. *Flowchart* dari penyebaran Buku Sekolah dapat dilihat pada

Gambar **III-2**.



Gambar III-2 Alur Pendistribusian Buku Sekolah Elektronik

III.1.3. Analisis Aturan Bisnis

Analisis aturan bisnis digunakan untuk menentukan aturan atau batasan yang pada sistem yang akan dibuat berdasarkan aturan yang telah ada sebelumnya dengan beberapa penyesuaian.

1. Aturan penyebaran materi buku sekolah elektronik

Buku sekolah elektronik yang digunakan disetiap sekolah di kabupaten Karawang berasal dari Disdikpora pusat yang disebarkan melalui Disdikpora kabupaten/kota. Setiap sekolah mendapatkan CD/DVD yang berisi buku sekolah elektronik berdasarkan kebutuhan buku yang telah diberikan kepada dinas pendidikan sebelumnya.

2. Aturan pendaftaran Sekolah dalam sistem

Sekolah yang akan menggunakan sistem harus sudah terdaftar di Disdikpora kabupaten Karawang dan sekolah tersebut harus memiliki NPSN *(Nomor Pokok Sekolah Nasional)* agar lebih mudah dalam melakukan pengelolaan dengan data yang telah ada. Jika nama sekolah tidak ada dalam daftar tetapi telah memiliki NPSN maka sekolah tersebut dapat menghubungi admin sistem untuk menambahkan data sekolah mereka dengan mengirimkan NPSN.

3. Aturan pembuatan kelas dalam sistem

Setiap sekolah dapat membuat kelas sesuai kebutuhan dengan sekolah tersebut. Setiap kelas harus mempunyai deskripsi dan tujuan dari pembuatan kelas tersebut. Setiap kelas hanya bisa memiliki satu pengajar.

4. Aturan *upload* materi oleh pengajar

Materi yang akan di *upload* harus sesuai dengan format yang telah ditentukan. Isi materi harus sesuai dengan silabus yang diberikan oleh Disdikpora dan juga harus sesuai dengan deskripsi yang ada pada kelas dimana materi tersebut di *upload.*

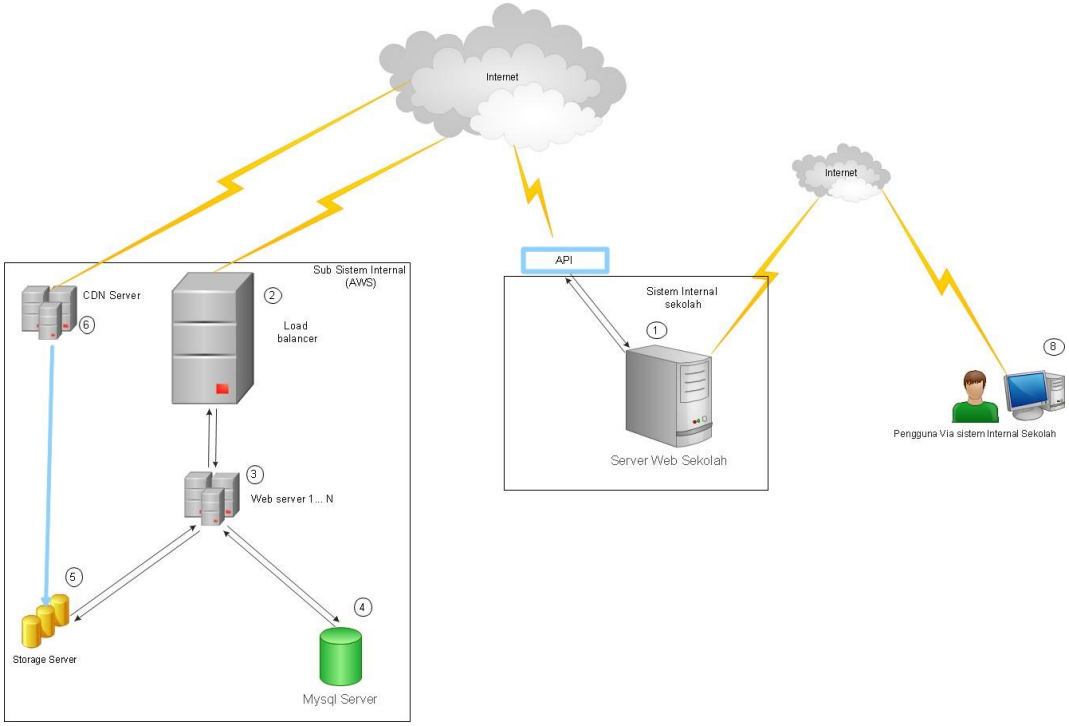
5. Aturan penyebaran materi dalam sistem

Materi yang telah di *upload* dapat disebarkan atau di-*download* untuk digunakan oleh siswa. Siswa juga dapat mencari materi dari sekolah lain jika dibutuhkan.

III.1.4. Analisis Arsiterktur Sistem

Analisis Arsitektur sistem berguna untuk menggambarkan bagaimana arsitektur sistem yang akan dibangun dan juga bisa menggambarkan interaksi antar komponen arsitektur di dalamnya.

Arsitektur sistem menggambarkan interaksi antara perangkat lunak yang sedang dibangunan dengan sistem yang digunakan di sekolah. Arsitektur sistem dapat dilihat pada Gambar III-3.



Gambar III-3 Arsitektur Sistem Integrasi Dengan Web Sekolah

a. Sistem internal sekolah berkomunikasi dengan Sistem utama menggunakan API. API tersebut digunakan untuk mengambil data dari sistem utama ataupun untuk mengirim data dan berkas ke sistem utama. Sistem internal sekolah dapat berupa aplikasi web, desktop atau mobile.
b. Pengguna *via* sistem internal sekolah mengakses sistem utama melalui sistem internal yang ada di sekolah. Semua data berasal dari sistem utama tetapi distribusi konten melewati sistem internal sekolah.

c. *Load balancer* menerima permintaan dari sistem internal sekolah dan akan meneruskan permintaan tersebut pada web server x. pemilihan web server berdasarkan trafik yang masuk pada web server tersebut. Permintaan akan diteruskan kepada web server dengan trafik yang tidak terlalu tinggi.
d. Web server akan memproses permintaan yang diteruskan dari *load balancer*. Permintaan tersebut akan diproses berdasarkan kebutuhan pengguna. Dalam hal ini pengguna akan mengunduh sebuah berkas.
e. Web server akan mencari alamat berkas yang akan diunduh didalam Mysql server. Jika berkas ditemukan, maka alamat dari berkas tersebut akan dicek oleh web server.
f. *Storage server* akan memberikan respon kepada web server. Respon yang diberikan berkaitan dengan ada tidaknya berkas yang diminta.
g. Web server akan mengirimkan alamat berkas kepada sistem internal sekolah, selanjutnya pengguna sistem web dapat mengunduh berkas tersebut melalui CDN *(Content Delivery Network)* Server. CDN server akan mengakses berkas berdasarkan alamat yang diminta oleh pengguna sistem sekolah.

III.1.5. Analisis File Sharing

Analisis *file sharing* menjelaskan tentang manajemen berkas, format berkas yang bisa diunggah ke dalam sistem, standarisasi untuk setiap berkas dari sekolah, dan keamanan dari sistem yang akan dibangun itu sendiri.

1. Manajemen Berkas

Manajemen berkas berguna untuk pengelompokan berkas yang telah diunggah oleh para pengguna ke dalam sistem, selain daripada itu manajeman berkas juga berguna untuk mengurangi duplikasi berkas yang ada. Manajemen berkas juga menjelelaskan seberapa besar ukuran maksimal sebuah berkas yang bisa diunggah dan bagaimana cara mendistribusikan berkas yang telah diunggah.

a. Ukuran berkas

Setiap berkas memiliki ukuran yang berbeda-beda. Besar kecilnya ukuran sebuah berkas ditentukan oleh seberapa banyak karakter yang ada dalam berkas

tersebut. Ukuran sebuah dokumen umumnya lebih kecil dibandingkan dengan sebuah video. Batas maksimum untuk sebuah berkas yang dapat diunggah ke dalam sistem adalah sebesar 300 MB (Mega Byte), hal tersebut berdasarkan dari analisa terhadap beberapa Buku Sekolah elektronik dan juga beberapa video tutorial yang ada. Berdasarkan hasil analisis terhadap beberapa Buku Sekolah Elektronik, didapat hasil bahwa ukuran Buku Sekolah Elektronik berkisar antara 4MB sampai dengan 30MB dan untuk ukuran sebuah video tutorial berkisar antara 10MB sampai dengan 200MB. Dari hasil analisa tersebut didapat kesimpulan bahwa kapasitas maksimum unggah berkas yang ditetapkan dapat melayani kebutuhan para pengguna. Detail dari batas unggah berkas dapat dilihat pada Tabel III-1.

Tabel III-1 Jenis dan Ukuran Berkas

Jenis Berkas	Ukuran Standar	Maksimal Unggah
Dokumen	4MB sampai 30MB	300MB
Video	10MB sampai 200MB	300MB

b. *File Hashing*

File hashing adalah sebuah metode yang digunakan untuk mengetahui nilai *hash* dari sebuah file. Setiap berkas akan menghasilkan nilai *hash* yang berbeda walau dokumen tersebut hanya berbeda satu karakter. *File hashing* memiliki banyak manfaat salah satu manfaatnya adalah menelusuri perubahan yang terjadi pada sebuah berkas, hal itu berguna jika ingin mengecek orisinilitas sebuah berkas. Selain untuk menelusuri perubahan yang ada pada berkas, *file hashing* juga bisa digunakan untuk mengurangi duplikasi berkas, setiap berkas yang sama persis pasti menghasilkan nilai *hash* yang sama pula, walau dokumen tersebut berbeda dalam nama dan waktu pembuatannya. Sebagai contoh terdapat dua buah berkas teks dengan nama dan waktu pembuatan yang berbeda tetapi memiliki isi dengan karakter yang sama, maka *hash* yang akan dihasilkan adalah sama.

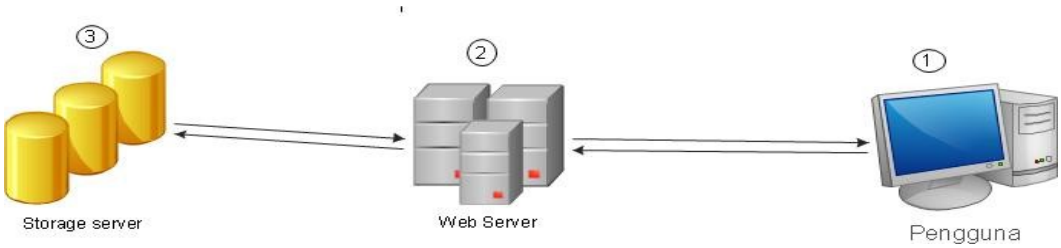
Salah satu metode *file hashing* yang sering digunakan adalah MD5 *(Message-Digest algorithm 5)*. Metode ini menghasilkan sebuah nilai dengan panjang 128 bit (16 Byte) yang dituangkan ke dalam *hexadecimal* sebanyak 32 digit. Penggunaan *hashing* dalam pembangunan perangkat lunak ini ditujukan

untuk mengurangi duplikasi berkas yang akan diunggah dan juga untuk menjaga bahwa berkas yang diunggah tidak rusak atau *corrupt*. Validasi dari sisi pengguna sebelum proses unggah menghasilkan sebuah nilai *hash* yang selanjutnya akan diunggah bersama berkas, Sehingga jika nilai *hash* yang telah dikirim sudah terdapat pada basis data sistem berkas tersebut tidak akan disimpan dalam *storage server*. Apabila nilai *hash* yang dikirim tidak terdapat dalam basis data sistem maka akan dilakukan pengujian ulang terhadap berkas yang diunggah. Pengujian dilakukan dengan cara mencocokan nilai *hash* yang dikirim dengan nilai *hash* yang ada pada berkas, jika nilai hash sama maka sistem akan menyimpan berkas tersebut dan menyimpan *hash* dari berkas ke dalam basis data, jika *hash* yang diuji menghasilkan nilai berbeda maka sistem akan mengirimkan pesan kesalahan.

c. Penyimpanan Berkas

Penyimpanan berkas yang telah diunggah pengguna menggunakan server khusus yaitu *storage server.* Web server hanya menampung sementara berkas yang diunggah lalu selanjutnya web server akan mengirimkan berkas tersebut ke *storage server.* Hal ini ditujukan untuk meminimalisir kemungkinan berkas terhapus secara tidak sengaja ketika ada *maintenance* terhadap web server sistem. Penyimpanan berkas dalam server khusus juga ditujukan agar pencarian berkas lebih mudah, karena jika berkas disimpan dalam server yang sama dengan web server maka akan sangat sulit melakukan pencarian karena arsitektur sistem menggunakan web server lebih dari satu server.

Storage server yang digunakan dalam sistem yang akan dibangun memiliki kemampuan untuk menambah kapasitas tanpa harus menkonfigurasi ulang sistem, hal ini sangat berguna terutama jika penggunaan tempat penyimpanan tidak bisa diprediksi. Alur penyimpanan berkas dari pengguna sampai dengan *storage server* dapat dilihat pada Gambar III-4.



Gambar III-4 Alur Penyimpanan Berkas

1. Pengguna mengunggah sebuah dokumen melalui web server.
2. setelah pengguna selesai mengunggah dokumen tersebut web server akan memindahkan dokumen tersebut ke dalam *storage server.*
3. *Storage server* akan menyimpan berkas yang dikirimkan oleh web server.

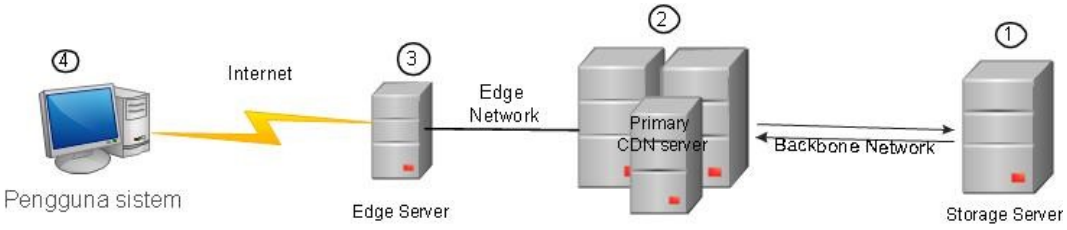
d. Penamaan Berkas

Setiap berkas yang telah diunggah ke server mengalami perubahan penamaan, tetapi untuk ekstensi berkas tidak mengalami perubahan. Penamaan berkas yang telah diunggah ke dalam server menggunakan nilai unik pada berkas tersebut. Sebagai contoh jika sebuah berkas yang akan diunggah ke dalam server mempunyai nama berkas "materi.pdf" maka setelah berkas tersebut diunggah ke dalam server, maka berkas tersebut berubah nama menjadi "cBe29_materi.pdf ". hal itu dimaksudkan untuk mencegah berkas dengan nama yang sama saat upload akan terhapus pada level penyimpanan di *storage server*. Penamaan berkas tersebut berlaku untuk semua format baik itu dokumen ataupun video. Pada saat berkas diunduh oleh pengguna nama berkas tersebut akan kembali ke nama semula yaitu dalam contoh ini adalah "materi.pdf".

e. Distribusi Berkas

Semua berkas yang telah diunggah tentunya harus bisa didistribusikan dengan baik kepada para pengguna. Cara yang digunakan untuk mendistribusikan berkas tentunya harus bisa memenuhi kebutuhan pengguna sistem. Salah satu masalah yang akan timbul jika cara untuk mendistribusikan tidak dapat memenuhi kebutuhan pengguna yang mengunduh berkas tersebut adalah pengguna gagal

setiap kali akan mengunduh dokumen tetapi server sedang sibuk. Sistem yang akan dibangun menggunakan sistem CDN *(Content Delivery Network)* dalam mendistribusikan berkas kepada pengguna yang akan mengunduh berkas tersebut. Penggunaan CDN dalam distribusi berkas yang akan diunduh dapat memaksimalkan distribusi, hal itu dikarenakan sistem CDN akan menggunakan rute terpendek dari lokasi pengunduh berkas tersebut, hal tersebut dapat meningkatkan kecepatan unduh karena delay yang kecil antara server dan pengguna. Layanan CDN juga menggunakan server yang berbeda dengan server yang digunakan oleh sistem yang akan dibangun, sehingga walaupun banyak pengguna yang sedang mengunduh berkas layanan sistem utama tidak akan terganggu ataupun menurun. Arsitektur dari sistem CDN dapat dilihat pada Gambar III-5.



Gambar III-5 Arsitektur CDN Server

1. *Storage server* menyimpan semua berkas yang telah diunggah. Berkas tersebut dapat diakses atau diunduh oleh pengguna yang memiliki alamat berkas di dalam *storage server*.
2. Distribusi berkas yang akan diunduh melalui CDN server. CDN server akan meng-*cache* atau menyimpan sementara berkas yang akan diunduh.
3. Setelah berkas berhasil di-*cache* oleh server utama CDN berkas tersebut didistribusikan melalui *edge server* yang terdekat denga lokasi pengunduh berkas tersebut.
4. Setelah berkas yang diunduh berada di *edge server* yang terdekat maka pengguna yang mengunduh berkas tersebut dapat mengunduh dengan kecepatan terbaik.

2. Format Berkas

Setiap berkas memilki format dan atribut yang berbeda. Berbeda sistem operasi atau editor yang digunakan untuk maka berbeda pula format dan atribut berkas tersebut. Pengaturan format dan atribut berkas yang akan diunggahke dalam sistem yang akan dibangun berguna untuk mencegah adanya berkas yang tidak dikenali sistem atau berkas yang berbahaya bagi sistem yang akan dibangun. Format berkas dan atribut berkas lebih jelasnya akan dibahas pada poin selanjutnya.

a. Format Dokumen

Dokumen yang bisa diunggah ke dalam sistem dibatasi untuk kebutuhan pendidikan saja, tidak semua format dokumen dapat diunggah ke dalam sistem yang akan dibangun. Beberapa dokumen dalam bentuk terkompresi seperti .exe (*executable*) tidak diperbolehkan untuk diunggah ke dalam sistem, hal tersebut diterapkan guna keamanan sistem yang akan dibangun. Berdasarkan analisa terhadap beberapa kasus, dokumen tersebut rawan akan terkena virus yang dapat membahayakan komputer pengguna sistem yang akan dibangun. Daftar format dokumen yang dapat diunggah ke dalam sistem dapat dilihat pada Tabel III-2.

Tabel III-2 Format Dokumen yang Diperbolehkan

Ekstensi Berkas	Keterangan
PDF	*Portable Document Format*
DOC	Microsoft Word 97-2003 *Document Format*
DOCX	Microsoft Word *Document Format*
PPT	Microsoft Power Point 97-2003 *Document Format*
PPTX	Microsoft Power Point *Document Format*
PPS	Power Point Show 97-2003
PPSX	Power Point Show
XLS	Microsoft Excel 97-2003 *Document Format*
XLSX	Microsoft Excel *Document Format*
ODT	OpenOffice Writer *Document Format*
ODS	OpenOffice Calc *Document Format*
ODP	OpenOffice Impress *Document Format*
CHM	Compiled HTML File

Format dokumen yang ada pada Tabel III-2 dapat bertambah sesuai kebutuhan dan perkembangan perangkat lunak pengolah dokumen yang akan datang.

b. Format Video

Seperti pada dokumen, berkas untuk video juga memiliki batasan mengenai video dengan format tertentu yang bisa diunggah ke dalam sistem. Berdasarkan analisa terhadap beberapa video tutorial yang ada maka beberapa format video tersebut sering digunakan untuk merekam sebuah video tutorial. Selain daripada itu beberapa format video memiliki kompresi yang cukup baik. Format video tersebut memiliki ukuran berkas yang cukup kecil namum dengan kualitas yang baik. Format video yang diperbolehkan diunggah ke dalam sistem dapat dilihat pada Tabel III-3.

Tabel III-3 Format Video yang Diperbolehkan

Format Video	Keterangan
MP4	MPEG-4 Part 14 *Video Format*
MOV	Quick Time *Video Format*
WMV	Windows Media Video Format
FLV	Flash *Video Format*
SWF	Shockwafe File Format

Format video yang diperbolehkan untuk diunggah ke dalam sistem dapat bertambah sesuai kebutuhan dan perkembangan teknologi video.

c. Format Arsip

Format berkas arsip digunakan untuk mengunggah berkas yang terdiri dari lebih satu berkas dijadikan satu kesatuan berkas. Format berkas arsip yang dapat diunggah kedalam sistem ini dapat dilihat pada.

Tabel III-4 Format Arsip yang Diperbolehkan

Format Video	Keterangan
RAR	Winrar Archive
ZIP	Winzip Archive
7Z	7zip Archive

d. Atribut Berkas

Setiap berkas harus memiliki atribut yang baik dan sesuai standar yang telah ditetapkan oleh sistem, hal itu dimaksudkan agar dokumen tersebut dapat dikenali dan dimanajemen dengan baik dalam lingkungan sistem. Sistem yang akan dibangun bertujuan untuk melayani distribusi berkas materi pembelajaran, sehingga atribut berkas akan sangat membantu jika seorang pengguna terutama di kalangan siswa sedang mencari sebuah materi.

3. Keamanan

Keamanan merupakan sebuah keharusan untuk setiap sistem yang akan dibangun. Faktor keamanan menjadi faktor utama dalam sebuah sistem. Sebuah sistem yang aman akan menjamin kerahasiaan data pengguna seperti *password* dan data pribadi lainnya. Dalam pembangunan sistem ini keamanan ditujukan untuk membatasi siapa saja yang dapat mengunggah dokumen, proteksi password dan batasan pada akses API yang disediakan oleh sistem.

a. *Password Hashing*

Password hashing yang akan digunakan dalam sistem yang akan dibangun berguna untuk mengacak *input password* yang dimasukan oleh pengguna sebelum dimasukan ke dalam basis data. Penggunaan *hashing* pada *password* berguna untuk mencegah pembobolan sistem jika data dalam sistem basis data diretas oleh pihak yang lain yang bukan bagian dari sistem yang akan dibangun. Metode *hashing* yang digunakan pada *password hashing* berbeda dengan metode yang digunakan pada *file hashing*. Pada *password hashing* selain algoritma *hashing* yang digunakan berbeda teknik dalam melakukannya berbeda pula. Dalam *password hashing* terdapat beberapa tahap *hashing* yang dilakukan terhadap *password* yang dimasukan oleh pengguna sistem. Berikut ini adalah tahap dalam melakukan *password hashing*:

1. Men-*generate string* acak menggunakan fungsi mcrypt_create_iv.
2. Menggabungkan *plain password* yang dimasukan user dengan *string* acak yang telah di-*generate* sebelumnya.

3. Menggunakan fungsi *hash* dengan algoritma SHA256 pada *string* yang telah digabungkan sebelumnya.

Dengan menggunakan metode yang telah dijelaskan sebelumnya akan menghasilkan sebuah hash yang cukup kuat untuk dijadikan sebuah password. Teknik *password hashing* yang dijelaskan sebelumnya adalah teknik *salting*. Teknik tersebut menambahkan karakter acak pada password yang dimasukan pengguna sebelum dilakukan *hashing.* Teknik *salting* cukup untuk memperkuat algoritma *hash* yang kadang bisa dibobol dengan cara *brute force, dictionary attack* atau dengan *rainbow table.* Penggunaan algoritma *hash* SHA256 (*Secure Hash Algorithm*) juga untuk mencegah celah keamanan yang terjadi pada algoritma *hash* MD5 dan SHA1. Selain untuk mencegah celah keamanan, pertimbangan lain juga adalah mengenai kecepatan.

b. API *Key*

API *key* digunakan jika pengguna sistem berupa sistem lain seperti sistem yang ada pada web sekolah atau sebuah aplikasi mobile. API *key* berguna untuk meminimalisir *farming* terhadap data yang ada pada sistem utama. Selain daripada hal itu API *key* juga dapat digunakan untuk memvalidasi sistem mana yang mengakses data. API *key* untuk setiap sistem lain yang akan mengakses haruslah unik, jadi tidak ada yang sama untuk dua sistem yang berbeda. Autentikasi terhadap akses melalui API masih menggunakan autentikasi dasar. Pada sistem yang akan dibangun tidak menggunakan teknik autentikasi seperti *Oauth* pada sistem API-nya. Autentikasi hanya berupa pencocokan API *key* dengan sistem sekolah yang mengakses, jika autentikasi berjalan sukses maka sistem akan memberikan data yang dimaksud, namun jika gagal sistem akan memberikan respon kesalahan. Pada Tabel III-5 memperlihatkan contoh API *key* dan alamat dari sistem lain yang bisa mengakses kepada sistem yang akan dibangun.

Tabel III-5 Contoh Data API *Key*

Nama Sekolah	Alamat *Domain* Sistem yang Akan Mengakses	API *Key*
SMK Negeri 1 Karawang	site.smkn1karawang.net	CoNtohAPIkEy
SMK Negeri 2 Karawang	smkn2-krw.sch.id	haNyaAPIKEy2

API *key* sendiri terdiri dari 12 karakter acak dari kumpulan huruf besar, huruf kecil dan angka. Penggunaan karakter acak dalam API *key* dimaksudkan untuk mempersulit orang dalam menebak karakter tersebut.

4. Mekanisme *Sharing*

Mekanisme *sharing* menjelaskan tatacara atau prosedur yang dijalankan ketika sistem akan saling bertukar informasi ataupun data lain. Mekanisme *sharing* sendiri terdiri dari komunikasi antara dua sistem yang ada, yaitu sistem utama dan sistem yang ada disekolah. Sistem sekolah akan mengirimkan informasi berkas yang akan diunggah kedalam sistem. Informasi pertama yang dikirim adalah nilai hash dari berkas yang akan diunggah dan nama berkas itu sendiri. Setelah nilai hash dan nama berkas diterima sistem, sistem akan mengecek nilai hash tersebut apakah sudah pernah ada di basis data. Jika nilai ditemukan di basis data, maka sistem akan mengirimkan alamat berkas tersebut ke sistem sekolah yang sebelumnya mengirimkan permintaan. Jika nilai hash yang dimaksud tidak ditemukan di basis data maka sistem akan memrintahkan sistem sekolah untuk mengunggah berkas tersebut, setelah proses unggah selesai sistem akan mengirimkan alamat tempat berkas tersebut disimpan.

1. *Application Programming Interface*

Aplication Programming Interface adalah sebuah cara yang digunakan untuk berkomunikasi antara sistem yang akan dibangun dengan sistem yang ada di sekolah.

a. Pengecekan berkas

Pengecekan berkas adalah proses yang dilakukan agar berkas yang sama tidak diunggah dua kali ke server. Pengecekan berkas menggunakan metode *file hashing*

yang ada pada bahasan sebelumnya. Sistem sekolah mengirimkan nilai hash dari berkas beserta nama berkas tersebut ke sistem yang akan dibangun, kemudia sistem yang akan dibangun mengecek apakah berkas yang sama pernah diunggah sebelumnya. Jika berkas sudah pernah diunggah sebelumnya maka sistem yang akan dibangun merespon dengan format API sebagai berikut:

```
{
 "check_hash":"sukses",
 "berkas":{
     "id_berkas":"id_berkas",
     "nama_berkas":"nama_berkas",
     "ukuran":123456,
     "format":"format",
    "href":"http://osindonesia.org/dw.php?id=id_berkas"
    }
}
```

Jika berkas belum pernah diunggah sebelumnya ke sistem akan merespon dengan format API sebagai berikut:

```
{
    "check_hash":"gagal",
    "hash":"0",
    "pesan":"silakan unggah berkas anda"
}
```

b. Unggah berkas

Unggah berkas adalah proses pengiriman berkas dari sistem sekolah ke sistem yang akan dibangun. Jika pada tahap pengecekan berkas yang telah dibahas sebelumnya menghasilkan nilai “check_hash” : “gagal” maka berkash tersebut akan diunggah ke sistem yang akan dibangun. Data yang dikirim pada saat mengunggah berkas adalah berkas itu sendiri. Jika proses unggah berhasil dan tidak terjadi kesalahan maka sistem akan memberikan respon dengan format API sebagai berikut:

```
{
 "unggah_berkas":"sukses",
 "berkas":{
     "id_berkas":"id_berkas",
     "nama_berkas":"nama_berkas",
     "ukuran":123456,
     "format":"format",
    "href":"http://osindonesia.org/dw.php?id=id_berkas"
    }
}
```

Jika terjadi sebuah kesalahan pada saat unggah berkas, seperti kesalahan ukuran berkas yang terlalu besar atau format berkas yang tidak didukung oleh sistem yang akan dibangun maka sistem akan memberikan respon dengan format API sebagai berikut:

```
{
    "unggah_berkas":"gagal",
    "pesan":"Pesan Kesalahan yang terjadi"
}
```

III.1.6. Analisis Spesifikasi Kebutuhan Perangkat Lunak

Berdasarkan analisis masalah, analisis prosedur yang sedang berjalan, analisis aturan bisnis dan analisis *file sharing* maka didapat spesifikasi kebutuhan untuk perangkat lunak *file sharing* materi pembelajaran. Terdapat dua bagian penting dalam spesifikasi kebutuhan yaitu kebutuhan fungsional dan kebutuhan non fungsional. Kebutuhan fungsional dapat dilihat pada Tabel III-6:

a. Spesifikasi kebutuhan fungsional administrator dapat dilihat pada Tabel III-6.

Tabel III-6 Spesifikasi Kebutuhan Fungsional Administrator

Kode SKPL	Spesifikasi Kebutuhan Perangkat Lunak
SKPL-F-1-001	Sistem menyediakan fungsionalitas *login*
SKPL-F-1-002	Sistem menyediakan fungsionalitas untuk menambah pengguna
SKPL-F-1-003	Sistem menyediakan fungsionalitas untuk menghapus pengguna
SKPL-F-1-004	Sistem menyediakan fungsionalitas untuk menambah data sekolah
SKPL-F-1-005	Sistem menyediakan fungsionalitas untuk menghapus data sekolah
SKPL-F-1-006	Sistem menyediakan fungsionalitas untuk melihat data sekolah
SKPL-F-1-007	Sistem menyediakan fungsionalitas untuk melihat pengguna

b. Spesifikasi kebutuhan fungsional Admin Sekolah dapat dilihat pada Tabel III-7.

Tabel III-7 Spesifikasi Kebutuhan Fungsional Admin Sekolah

Kode SKPL	Spesifikasi Kebutuhan Perangkat Lunak
SKPL-F-2-001	Sistem menyediakan fungsionalitas *login*
SKPL-F-2-002	Sistem menyediakan fungsionalitas untuk menghapus materi pembelajaran
SKPL-F-2-003	Sistem menyediakan fungsionalitas untuk merubah *password*
SKPL-F-2-004	Sistem menyediakan fungsionalitas untuk melihat detail sekolah masing-masing
SKPL-F-2-005	Sistem menyediakan fungsionalitas untuk mengubah data sekolah masing-masing
SKPL-F-2-006	Sistem menyediakan fungsionalitas untuk mengunduh materi pembelajaran
SKPL-F-2-007	Sistem menyediakan fungsionalitas untuk melihat daftar berkas yang diunggah

c. Spesifikasi kebutuhan fungsional siswa dapat dilihat pada Tabel III-8.

Tabel III-8 Spesifikasi Kebutuhan Fungsional Sistem Sekolah

Kode SKPL	Spesifikasi Kebutuhan Perangkat Lunak
SKPL-F-3-001	Sistem menyediakan fungsionalitas untuk mengunggah materi
SKPL-F-3-001	Sistem menyediakan fungsionalitas untuk mengunduh materi

Kebutuhan Non fungsional dari sistem yang akan dibangun dapat dilihat pada Tabel III-9.

Tabel III-9 Kebutuhan non Fungsional

Kode SKPL	Spesifikasi Kebutuhan Perangkat Lunak
SKPL-NF-1-001	Sistem yang akan dibangun berbasiskan web
SKPL-NF-1-002	Sistem yang akan dibangun menggunakan server dengan teknologi *Amazon Web Service*
SKPL-NF-1-003	Berkas materi pembelajaran disimpan di dalam *cloud* di AWS
SKPL-NF-1-004	Perangkat keras yang digunakan untuk menjalankan sistem harus sesuai kebutuhan minimal sistem
SKPL-NF-1-005	Sistem yang akan dibangun menyediakan API untuk pertukaran data dengan sistem lain
SKPL-NF-1-006	Sistem menggunakan Standar JSON untuk format pertukaran datanya

III.1.7. Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional menjelaskan spesifikasi yang terperinci mengenai hal-hal yang dilakukan di dalam proses. Analisis kebutuhan non fungsional pada sistem *file sharing* materi pembelajaran ini terdiri dari analisis kebutuhan perangkat lunak, analisis kebutuhan perangkat keras dan analisis kebutuhan perangkat pikir.

1. Analisis Kebutuhan Perangkat Lunak

Spefisikasi perangkat lunak yang digunakan dalam pembangunan sistem ini adalah sebagai berikut:

a. Sistem operasi windows 8.1
b. JetBrains PhpStorm 7.1.3
c. PHP 5.4.7
d. Apache HTTP server 2.4
e. Mysql server 5.5
f. Browser terbaru yang mendukung teknologi javascript dan HTML 5

2. Analisis Kebutuhan Perangkat Keras

Spesifikasi perangkat keras agar dapat menjalankan sistem ini dengan baik adalah sebagai berikut:

a. Intel Core 2 Duo 2.6Mhz
b. RAM 2Gb
c. HDD 160Gb
d. Monitor dengan resolusi 1024*768
e. Mouse
f. Keyboard
g. Modem 1Mbps

3. Analisis Pengguna

Analisis kebutuhan perangkat pikir digunakan untuk menganalisa karakter daripada pengguna yang akan menggunakan perangkat lunak yang akan dibangun. Adapun karakteristik pengguna secara rinci diuraikan pada bahasan berikut ini**.**

a. Analisis pengguna berdasarkan tingkat kemampuan dan pengalaman dapat dilihat pada Tabel III-10.

Tabel III-10 Pengguna Berdasarkan Kemampuan dan Pengalaman

Jenis Pengguna	Kemampuan pengguna	
Administrator	Keterampilan menggunakan komputer	pernah mengoperasikan komputer minimal tiga bulan
	Pengetahuan sistem	Mempunyai pengetahuan tentang sistem pengolahan data berbasis web
	Pengalaman menggunakan aplikasi	Mempunyai pengalaman dalam menggunakan aplikasi manajemen berkas berbasis web
	Pendidikan	Minimum perguruan tinggi
	Tingkat kemampuan membaca	Mempunyai kemampuan membaca tingkat sedang sampai dengan tinggi
	Keterampilan mengetik	Memunyai keterampilan menulis atau mengetik rata-rata yaitu 60 kata setiap menitnya.
	Kemampuan berbahasa	Dapat berbahasa Indonesia
Admin Sekolah	Keterampilan menggunakan komputer	Pernah menggunakan komputer
	Pengetahuan sistem	Memiliki pengetahuan untuk menjalankan aplikasi web
	Pengalaman menggunakan aplikasi	Pernah menggunakan aplikasi unggah berkas
	Pendidikan	Minimum perguruan tinggi
	Tingkat kemampuan membaca	Mempunyai kemampuan membaca sedang sampai tinggi

Jenis Pengguna	Kemampuan pengguna	
	Keterampilan mengetik	Memunyai keterampilan menulis atau mengetik rata-rata yaitu 30 kata setiap menitnya.
	Kemampuan berbahasa	Dapat berbahasa Indonesia

b. Analisis pengguna berdasarkan karakteristik kebutuhan pengguna, tugas dan pekerjaan dapat dilihat pada Tabel III-11.

Tabel III-11 Karakteristik Berdasarkan Kebutuhan, Tugas dan Pekerjaan

Jenis Pengguna	Kebutuhan/Tugas/Pekerjaan	
Administrator	Jenis pengguna dalam sistem	Pengelola berkas dan pengguna lain yang ada dalam sistem
	Intensitas penggunaan	Pengelolaan berkas dilakukan setiap saat
	Kebutuhan dalam sistem	Pengelola berkas yang diunggah dalam sistem
	Pelatihan	Dibutuhkan pelatihan dalam mengelola berkas yang telah diunggah
Admin Sekolah	Jenis pengguna dalam sistem	Menjadi sumber pengisi materi dalam sistem
	Intensitas penggunaan	Penggunaan aplikasi dapat dilakukan berdasarkan kebutuhan pengguna
	Kebutuhan dalam sistem	Pengunggah berkas materi dalam sistem

c. Analisis pengguna berdasarkan karakteristik fisik dapat dilihat pada Tabel III-12.

Tabel III-12 Pengguna Berdasarkan Karakter Fisik

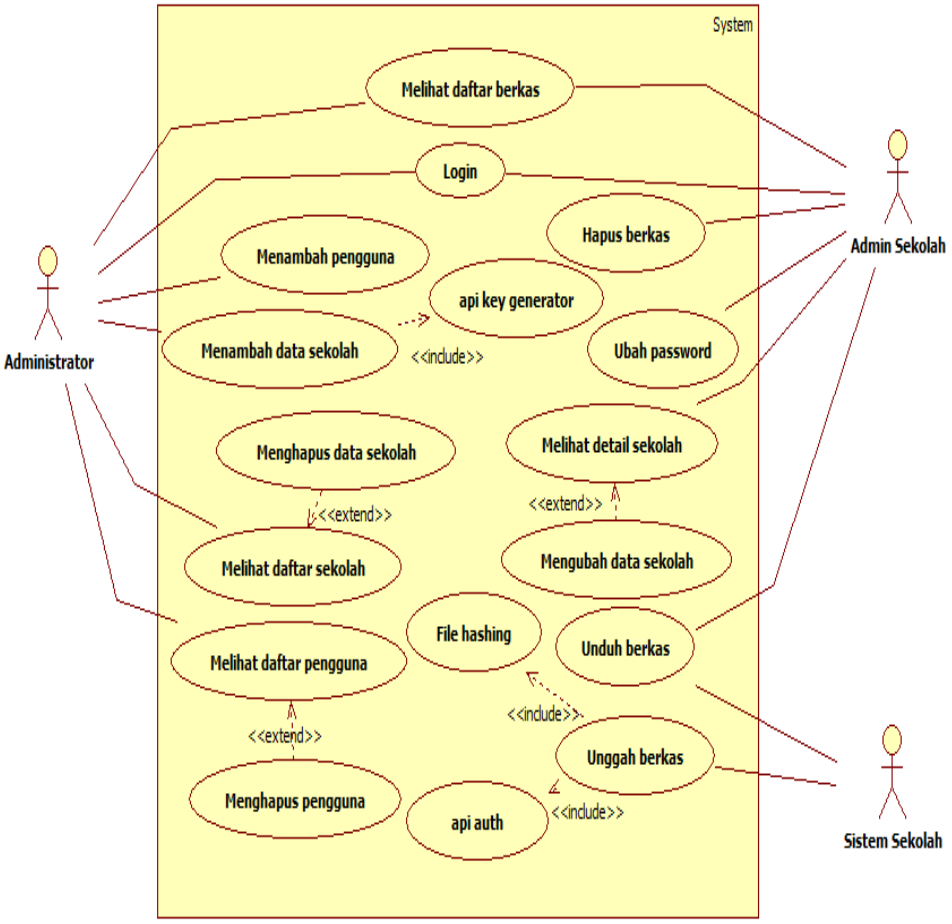
Jenis Pengguna	Karakteristik Fisik	
Administrator	Umur	Lebih dari 25 tahun
	Handedness	Dapat menggunakan kedua tangannya dengan baik
	Disabilities	Tidak mempunyai kelainan anggota tubuh terutama mata dan tangan
Admin Sekolah	Umur	Lebih dari 21 tahun
	Handedness	Dapat menggunakan kedua tangannya dengan baik
	Disabilities	Tidak mempunyai kelainan anggota tubuh terutama mata dan tangan

III.1.8. Analisis Kebutuhan Fungsional

Di dalam analisis kebutuhan fungsional ini digambarkan proses yang akan ada pada sistem yang akan dibangun. Analisis sistem yang akan dibangun menggunakan pendekatan analisis berorientasi objek yang menggunakan pemodelan UML yang terdiri dari *use case diagram, activity diagram, class diagram* dan *sequence diagram.*

1. Use Case Diagram

Use case diagram menggambarkan fungsionalitas terhadap sistem yang akan dibangun. Pada sistem yang akan dibangun saat ini terdiri dari beberapa *use case* yang dapat dilihat pada Gambar III-6.



Gambar III-6 *Use Case Diagram*

2. Definisi Aktor

Berikut ini adalah definisi aktor yang ada pada *use case diagram*:

Tabel III-13 Definisi Aktor

No	Aktor	Deskripsi
1	Administrator	Pengguna yang mempunyai fungsi untuk memanage sistem yang akan berjalan berdasarkan usecase yang ada.
2	Admin Sekolah	Pengguna yang mempunya fungsi yang berkaitan dengan sekolah masing-masing
3	Sistem Sekolah	Sistem yang berkomunikasi dengan sistem yang akan dibangun

3. Definisi *Use Case Diagram*

Bagian ini menjelaskan mengenai difinisi dari *use case diagram* yang ada pada sistem. Definisi *use case diagram* dapat dilihat pada Tabel III-14.

Tabel III-14 Definisi *Use Case*

No	*Use Case*	Deskripsi
1	*Login*	Sistem menyediakan fungsionalitas untuk masuk ke dalam sistem
2	Menambah pengguna	Sistem menambahkan pengguna ke dalam basis data
3	Melihat daftar pengguna	Sistem menampilkan *list* pengguna
4	Menghapus pengguna	Sistem menghapus data pengguna dari sistem
5	Melihat daftar sekolah	Sistem menampilkan data *list* sekolah yang ada pada sistem
6	Menambah data sekolah	Sistem menambah data sekolah ke basis data
7	Mengubah data sekolah	Sistem mengubah data sekolah yang sudah ada di basis data
8	Menghapus data sekolah	Sistem Menghapus data sekolah yang telah ada di dalam basis data
9	*File hashing*	Sistem mengecek hash dari materi yang diunggah ke dalam sistem
10	Melihat daftar berkas	Sistem menampilkan berkas materi yang telah diunggah ke dalam sistem
11	Hapus berkas	Sistem menghapus berkas materi yang telah diunggah
12	Mengubah *password*	Sistem dapat mengganti atau merubah *password* pengguna
13	Melihat Detail Sekolah	Melihat informasi dasar sekolah
14	Api *auth*	Otentikasi Sistem API
15	Unduh berkas	Mengunduh berkas
16	Unggah berkas	Mengunggah berkas
17	Api *key generator*	Men-*generate* Api *key*

4. Skenario *Use Case*

Skenario *Use Case* digunakan untuk menjelaskan interaksi aktor dengan dengan *use case*. Interaksi aktor dengan *use case* yang terjadi dari awal hingga akhir proses. Berikut ini adalah skenario *use case* pada sistem yang akan dibangun:

a. Skenario *Use Case Login*

Skenario *use case login* yang ada pada sistem digunakan ketika pengguna akan masuk ke dalam sistem utama. Skenario *use case login* dapat dilihat pada Tabel III-16:

Tabel III-15 *Requirment* SKPL-F-1-001

Requirment SKPL-F-1-001
Sistem menyediakan fungsionalitas *login*

Tabel III-16 Skenario *use case login*

Use Case Name	*Login*	
Related Requirement	*Requirement* SKPL-F-1-001	
Goal in Context	Melakukan *login* ke dalam sistem	
Precondition	Sistem menyediakan *form login*	
Succes End Condition	Masuk ke halaman utama web	
Failed End Condition	Menampilkan pesan kesalahan *login*	
Primary Actors	Administrator, Admin Sekolah	
Secondary Actors	-	
Trigger	Pengguna menekan tombol *login*	
Included Cases	-	
Mainflow	Step	Action
	1	Pengguna memilih menu *login*
	2	Pengguna memasukan *username* dan *password* di *form*
	3	Sistem akan memvalidasi data *username* dan *password*
	4	Sistem menampilkan halaman utama aplikasi
Extensions	Step	Branching Action
	3.1	Jika data *username* dan *password* tidak valid maka sistem akan menampilkan pesan kesalahan login

b. Skenario *Use Case* Menambah Pengguna

Skenario *use case* menambah pengguna menjelaskan alur ketika sistem menerima data yang telah dimasukan pengguna ke dalam *form* pendaftaran lalu

memprosesnya untuk disimpan ke dalam basis data. Berikut ini adalah skenario *use case* menambah pengguna yang dapat dilihat pada Tabel III-18.

Tabel III-17 *Requirment* SKPL-F-1-002

Requirment SKPL-F-1-002
Sistem menyediakan fungsionalitas untuk menambah pengguna

Tabel III-18 Skenario *Use Case* Menambah Pengguna

Use Case Name	Menambah pengguna	
Related Requirement	*Requirement* SKPL-F-1-002	
Goal in Context	Memasukan data pendaftaran pengguna ke dalam basis data	
Precondition	Pengguna mengirimkan data pendaftaran kepada sistem	
Succes End Condition	Data pendaftaran pengguna berhasil disimpan ke dalam basis data	
Failed End Condition	Menampilkan pesan kesalahan data gagal disimpan ke dalam basis data	
Primary Actors	Administrator	
Secondary Actors	-	
Trigger	Sistem menerima data pendaftaran	
Included Cases	-	
Mainflow	Step	Action
	1	Sistem menerima data pendaftaran
	2	Sistem memvalidasi data pendaftaran
	3	Sistem menyimpan data pendaftaran pengguna ke dalam basis data
	4	Pengguna berhasil mendaftar ke dalam sistem
Extensions	Step	Branching Action
	3.1	Jika data pendaftaran tidak valid maka sistem akan menampilkan pesan kesalahan

c. Skenario *Use Case* Melihat Daftar Pengguna

Skenario *use case* melihat daftar pengguna menjelaskan alur ketika pengguna ingin melihat daftar pengguna yang ada di sistem. Adapun skenario dari *use case* melihat daftar pengguna dapat dilihat pada Tabel III-20.

Tabel III-19 Requirment SKPL-F-1-007

Requirment SKPL-F-1-007
Sistem menyediakan fungsionalitas untuk melihat pengguna

Tabel III-20 Skenario *Use Case* Melihat Daftar Pengguna

Use Case Name	Melihat daftar pengguna	
Related Requirement	*Requirement* SKPL-F-1-007	
Goal in Context	Menampilkan daftar pengguna yang ada di sistem	
Precondition	Pengguna memilih menu pengguna	
Succes End Condition	Data berhasil ditampilkan	
Failed End Condition	Data pengguna tidak dapat ditampilkan	
Primary Actors	Administrator	
Secondary Actors	-	
Trigger	Pengguna memilih menu pengguna	
Included Cases	-	
Mainflow	Step	Action
	1	Pengguna memilih menu daftar pengguna
	2	Sistem menampilkan daftar pengguna
Extensions	Step	Branching Action
	2.1	Sistem tidak dapat menampilkan daftar pengguna

d. Skenario *Use Case* Menghapus Pengguna

Skenario *use case* menghapus pengguna menjelaskan alur dari penghapusan seorang pengguna yang sudah terdaftar dari sistem. Adapun skenario dari *use case* hapus pengguna dapat dilihat pada Tabel III-22.

Tabel III-21 *Requirment* SKPL-F-1-003

Requirment SKPL-F-1-003
Sistem menyediakan fungsionalitas untuk menghapus pengguna

Tabel III-22 Skenario *Use Case* Menghapus Pengguna

Use Case Name	Hapus pengguna	
Related Requirement	*Requirement* SKPL-F-1-003	
Goal in Context	Menghapus pengguna yang telah terdaftar dalam sistem	
Precondition	Memilih menu daftar pengguna	
Succes End Condition	Berhasil menghapus pengguna	
Failed End Condition	Menampilkan kesalahan tidak bisa menghapus pengguna	
Primary Actors	Administrator	
Secondary Actors	-	
Trigger	Pengguna memilih menu hapus	
Included Cases	-	
Mainflow	Step	Action
	1	Pengguna memilih menu tampilkan pengguna

	2	Sistem menampilkan daftar pengguna yang ada di sistem
	3	Pengguna memilih menu hapus pada list pengguna yang ada
	4	Sistem berhasil mengahapus pengguna
Extensions	Step	Branching Action
	4.1	Sistem gagal menghapus pengguna

e. Skenario *Use Case* Menambah Data Sekolah

Skenario *use case* menambah data sekolah menjelaskan alur ketika pengguna akan menambah data sekolah ke dalam sistem. Adapun skenario *use case* menambah data sekolah dapat dilihat pada Tabel III-24.

Tabel III-23 *Requirment* SKPL-F-1-004

Requirment SKPL-F-1-004
Sistem menyediakan fungsionalitas untuk menambah data sekolah

Tabel III-24 Skenario *Use Case* Menambah Data Sekolah

Use Case Name	Menambah data sekolah	
Related Requirement	*Requirement* SKPL-F-1-004	
Goal in Context	Menambah data sekolah ke dalam sistem	
Precondition	Memilih menu tambah sekolah	
Succes End Condition	Berhasil menambahkan data sekolah ke dalam basis data	
Failed End Condition	Menampilkan kesalahan tidak dapat menambahkan data sekolah	
Primary Actors	Administrator	
Secondary Actors	-	
Trigger	Pengguna memilih menu tambah sekolah	
Included Cases	Api *generator*	
Mainflow	Step	Action
	1	Pengguna memilih menu tambah sekolah
	2	Sistem menampilkan form tambah sekolah
	3	Pengguna memasukan data sekolah ke dalam form tambah sekolah
	4 *Include* api *generator*	Pengguna menekan tombol tambah sekolah
	5	Sistem memvalidasi data sekolah yang dimasukan
	6	Sistem menyimpan data sekolah ke dalam basis data
Extensions	Step	Branching Action
	5.1	Sistem memberikan pesan kesalahan data tidak valid

	6.1	Sistem gagal menyimpan data sekolah ke dalam basis data

f. Skenario *Use Case* Melihat Daftar Sekolah

Use case melihat data sekolah digunakan ketika pengguna ingin melihat data sekolah yang ada pad sistem. Skenario *use case* melihat data sekolah menjelaskan alur dari *use case* tersebut. Berikut ini adalah skenario *use case* melihat data sekolah yang dapat dilihat pada Tabel III-26.

Tabel III-25 *Requirment* SKPL-F-1-006

Requirment SKPL-F-1-006
Sistem menyediakan fungsionalitas untuk melihat data sekolah

Tabel III-26 Skenario *Use Case* Melihat Data Sekolah

Use Case Name	Melihat data sekolah	
Related Requirement	*Requirement* SKPL-F-1-006	
Goal in Context	Menampilkan data sekolah	
Precondition	Memilih menu sekolah	
Succes End Condition	Menampilkan data sekolah	
Failed End Condition	Data sekolah tidak tampil	
Primary Actors	Administrator	
Secondary Actors	-	
Trigger	Pengguna memilih menu sekolah	
Included Cases	-	
Mainflow	Step	Action
	1	Pengguna memilih menu sekolah
	2	Sistem menampilkan data sekolah
Extensions	Step	Branching Action
	2.1	Sistem gagal menampilkan data sekolah

g. Skenario *Use Case* Mengubah Data Sekolah

Use case mengubah data sekolah memiliki fungsi untuk mengubah data sekolah yang ada pada sistem. Adapun skenario dari *use case* mengubah data sekolah adalah untuk menjelaskan alur dari pengguna yang akan mengubah data sekolah. Berikut ini adalah skenario *use case* mengubah data sekolah yang dapat dilihat pada Tabel III-28.

Tabel III-27 *Requirment* SKPL-F-2-005

Requirment SKPL-F-2-005
Sistem menyediakan fungsionalitas untuk mengubah data sekolah masing-masing

Tabel III-28 Skenario *Use Case* Merubah Data Sekolah

Use Case Name	Merubah data sekolah	
Related Requirement	*Requirement* SKPL-F-2-005	
Goal in Context	Merubah data sekolah yang ada dalam sistem	
Precondition	Memilih menu ubah data sekolah	
Succes End Condition	Berhasil merubah data sekolah di basis data	
Failed End Condition	Menampilkan kesalahan tidak dapat merubah data sekolah	
Primary Actors	Admin sekolah	
Secondary Actors	-	
Trigger	Pengguna memilih menu ubah	
Included Cases	-	
Mainflow	Step	Action
	1	Pengguna memilih menu ubah sekolah
	2	Sistem menampilkan form ubah sekolah
	3	Pengguna memasukan data sekolah yang baru pada form ubah sekolah
	4	Pengguna menekan tombol simpan
	5	Sistem menyimpan data sekolah ke dalam basis data
Extensions	Step	Branching Action
	5.1	Sistem memberikan pesan kesalahan data tidak bisa dirubah

h. Skenario *Use Case* Menghapus Data Sekolah

Skenario *use case* menghapus data sekolah menjelaskan alur dari pengguna yang akan menghapus data sekolah dari basis data. Berikut ini adalah skenario *use case* mengapus data sekolah yang dapat dilihat pada Tabel III-30.

Tabel III-29 *Requirment* SKPL-F-1-005

Requirment SKPL-F-1-005
Sistem menyediakan fungsionalitas untuk menghapus data sekolah

Tabel III-30 Skenario *Use Case* Menghapus Data Sekolah

Use Case Name	Menghapus data sekolah
Related Requirement	*Requirement* SKPL-F-1-005
Goal in Context	Menghapus data sekolah yang ada dalam sistem
Precondition	Memilih menu hapus data sekolah
Succes End Condition	Berhasil menghapus data sekolah di basis data
Failed End Condition	Menampilkan kesalahan tidak dapat menghapus data sekolah
Primary Actors	Administrator

Secondary Actors	-	
Trigger	Pengguna memilih menu hapus	
Included Cases	-	
Mainflow	Step	Action
	1	Pengguna memilih menu sekolah
	2	Sistem menampilkan list data sekolah
	3	Pengguna memilih menu hapus
	4	Sistem menghapus data sekolah dari basis data
Extensions	Step	Branching Action
	4.1	Sistem memberikan pesan kesalahan data tidak bisa dihapus

i. Skenario *Use Case File Hashing*

Skenario *use case file hashing* menjelaskan alur dari proses *file hashing* yang dilakukan sistem terhadap berkas yang diunggah pengguna. Berikut ini adalah skenario *use case file hashing* yang dapat dilihat pada Tabel III-31.

Tabel III-31 Skenario *Use Case File Hashing*

Use Case Name	*File hashing*	
Related Requirement	*Requirement* SKPL-F-3-001	
Goal in Context	Mengecek nilai *hash* dari berkas	
Precondition	Memilih berkas	
Succes End Condition	Sistem berhasil mengecek nilai *hash* berkas	
Failed End Condition	Sistem tidak bisa mengecek nilai *hash* berkas	
Primary Actors	Sistem sekolah	
Secondary Actors	-	
Trigger	Memilih berkas yang akan diunggah	
Included Cases	-	
Mainflow	Step	Action
	1	Pengguna memilih berkas yang akan diunggah
	2	Sistem mengecek *hash* dari berkas tersebut dari sisi *client*
	3	Sistem mengirim hasil *hash* berkas kepada server
	4	Sistem menerima hasil *hash* dari server
Extensions	Step	Branching Action
	4.1	Sistem tidak menerima hasil *hash* dari server

j. Skenario *Use Case* melihat daftar berkas

Skenario *use case* melihat daftar berkas menjelaskan alur ketika pengguna akan melihat daftar berkas yang pernah diunggah olah sekolahnya. Berikut ini adalah skenario *use case* melihat daftar berkas yang dapat dilihat pada .

Tabel III-32 Requirment SKPL-F-2-007

Requirment SKPL-F-2-007
Sistem menyediakan fungsionalitas untuk melihat daftar berkas yang diunggah

Tabel III-33 Skenario *Use Case* Melihat Daftar Berkas

Use Case Name	Melihat daftar berkas	
Related Requirement	*Requirement* SKPL-F-2-007	
Goal in Context	Menampilkan daftar berkas yang ada	
Precondition	Pengguna memilih menu berkas	
Succes End Condition	Data berhasil ditampilkan	
Failed End Condition	Data tidak dapat ditampilkan	
Primary Actors	Admin sekolah	
Secondary Actors	-	
Trigger	Pengguna memilih menu daftar berkas	
Included Cases	-	
Mainflow	Step	Action
	1	Pengguna memilih menu daftar berkas
	2	Sistem menampilkan daftar berkas
Extensions	Step	Branching Action
	2.1	Sistem tidak dapat menampilkan daftar berkas

k. Skenario *Use Case* hapus berkas

Skenario *use case* hapus berkas menjelaskan alur ketika pengguna akan menghapus berkas yang pernah diunggahnya. Pengguna hanya bisa menghapus berkas materi yang pernah diunggahnya. Berikut ini adalah skenario *use case* menghapus berkas materi yang dapat dilihat pada Tabel III-35.

Tabel III-34 *Requirment* SKPL-F-2-002

Requirment SKPL-F-2-002
Sistem menyediakan fungsionalitas untuk menghapus berkas materi pembelajaran

Tabel III-35 Skenario *Use Case* Menghapus Berkas

Use Case Name	Menghapus berkas
Related Requirement	*Requirement* SKPL-F-2-002

Goal in Context	Menghapus berkas materi yang telah diunggah	
Precondition	Memilih menu hapus di daftar berkas	
Succes End Condition	Berhasil menghapus berkas yang telah diunggah dari sistem	
Failed End Condition	Gagal menghapus berkas materi yang ada di sistem	
Primary Actors	Admin sekolah	
Secondary Actors	-	
Trigger	Memilih menu hapus	
Included Cases	-	
Mainflow	Step	Action
	1	Pengguna memilih berkas yang akan dihapus
	2	Pengguna menekan tombol hapus
	3	Sistem menghapus berkas yang dipilih
Extensions	Step	Branching Action
	3.1	Sistem menampilkan pesan kesalahan tidak dapat menghapus berkas

l. Skenario *Use Case* ubah *Password*

Skenario *use case* ubah *password* menjelaskan alur ketika pengguna akan mengubah password mereka. Berikut ini adalah skenario *use case* mengubah *password* yang dapat dilihat pada Tabel III-37.

Tabel III-36 *Requirment* SKPL-F-2-005

Requirment SKPL-F-2-003
Sistem menyediakan fungsionalitas mengubah *password*

Tabel III-37 Skenario *Use Case* Ubah *Password*

Use Case Name	Ubah *password*	
Related Requirement	*Requirement* SKPL-F-2-003	
Goal in Context	Mengganti password pengguna	
Precondition	Memilih menu ubah password	
Succes End Condition	*Password* berhasil dirubah	
Failed End Condition	Gagal mengubah *password*	
Primary Actors	Admin sekolah	
Secondary Actors	-	
Trigger	Memilih menu ubah *password*	
Included Cases	-	
Mainflow	Step	Action
	1	Pengguna memilih menu ubah *password*
	2	Sistem menampilkan form ubah *password*

	3	Pengguna memasukan data *password* lama dan *password* baru
	4	Mencocokan password lama
	5	Sistem menyimpan perubahan *password* ke dalam basis data
Extensions	Step	Branching Action
	5.1	Sistem menampilkan kesalahan *password* tidak bisa dirubah

m. Skenario *Use Case* Unggah Berkas

Skenario *Use Case* unggah berkas menjelaskan tentang alur ketika sistem sekolah akan mengunggah berkas ke sistem yang ada. Berikut ini adalah skenario dari *use case* unggah berkas yang dapat dilihat pada Tabel III-39.

Tabel III-38 Requirment SKPL-F-3-001

Requirment SKPL-F-3-001
Sistem menyediakan fungsionalitas untuk mengunggah materi

Tabel III-39 Skenario *Use Case* Unggah Berkas

Use Case Name	Unggah Berkas	
Related Requirement	*Requirement* SKPL-F-3-001	
Goal in Context	Materi pembelajaran dapat diunggah	
Precondition	Memilih materi yang akan diunggah	
Succes End Condition	Materi dapat diunggah oleh pengguna	
Failed End Condition	Pengguna tidak dapat mengunduh materi	
Primary Actors	Sistem Sekolah	
Secondary Actors	-	
Trigger	Memilih berkas yang akan diunggah	
Included Cases	*File hashing, api auth*	
Mainflow	Step	Action
	1	Pengguna memilih berkas yang akan diunggah
	2 *Include file hashing*	Sistem mengecek nilai *hash* berkas tersebut
	3 *Include api auth*	Sistem mengecek hash berkas ke server utama
	4	Sistem mengunggah berkas ke server
Extensions	Step	Branching Action
	4.1	Berkas tidak sesuai aturan, unggah berkas gagal

n. Skenario *Use Case Api Auth*

Skenario *use case api auth* menjelaskan alur ketika sistem sekolah akan berkomunikasi dengan sistem yang akan dibangun. Berikut ini adalah skenario dari *use case api auth* yang dapat dilihat pada Tabel III-40.

Tabel III-40 Skenario *Use Case Api Auth*

Use Case Name	*Api auth*	
Related Requirement	*Requirement* SKPL-F-3-001	
Goal in Context	Otentikasi berhasil	
Precondition	Mengirim data otentikasi	
Succes End Condition	Pengguna bisa masuk ke sistem api	
Failed End Condition	Pengguna tidak dapat masuk ke sistem api	
Primary Actors	Sistem Sekolah	
Secondary Actors	-	
Trigger	Mengirim data otentikasi	
Included Cases		
Mainflow	Step	Action
	1	Pengguna mengirim data api key
	2	Sistem mengecek data api key
	3	Pengguna bisa berkomunikasi dengan sistem api
Extensions	Step	Branching Action
	2.1	Otentikasi gagal

o. Skenario *Use Case Api key Generator*

Use Case Name	*Api key generator*	
Related Requirement	*Requirement* SKPL-F-1-004	
Goal in Context	Api *key* berhasil dibuat	
Precondition	Nama Sekolah	
Succes End Condition	Sekolah berhasil mendapatkan data api *key*	
Failed End Condition	Api *key* gagal dibuat	
Primary Actors	Administrator	
Secondary Actors	-	
Trigger	Menambah data sekolah	
Included Cases		
Mainflow	Step	Action
	1	Pengguna menambah data sekolah
	2	Sistem men-*generate* api *key* untuk sekolah tersebut
Extensions	Step	Branching Action
	2.1	Api *key* gagal di-*generate*

p. Skenario *Use Case* Unduh Berkas

Skenario *use case* unduh berkas menjelaskan tentang alur ketika pengguna akan mengunduh materi yang ada di sebuah kelas. Berikut ini adalah skenario *use case* mengunduh materi yang dapat dilihat pada Tabel III-42.

Tabel III-41 *Requirment* SKPL-F-3-001

Requirment SKPL-F-2-006
Sistem menyediakan fungsionalitas untuk mengunduh materi pembelajaran

Tabel III-42 Skenario *Use Case* Unduh Berkas

Use Case Name	Unduh Berkas	
Related Requirement	*Requirement* SKPL-F-2-006	
Goal in Context	Materi pembelajaran dapat diunduh	
Precondition	Memilih materi yang akan diunduh	
Succes End Condition	Materi dapat diunduh oleh pengguna	
Failed End Condition	Pengguna tidak dapat mengunduh materi	
Primary Actors	Admin Sekolah, Sistem Sekolah	
Secondary Actors	-	
Trigger	Memilih menu unduh	
Included Cases	-	
Mainflow	Step	Action
	1	Pengguna masuk ke dalam kelas
	2	Pengguna memilih materi yang akan diunduh
	3	Pengguna menekan tombol unduh
	4	Sistem memberikan alamat berkas yang akan diunduh
	5	Pengguna mengunduh berkas
Extensions	Step	Branching Action
	4.1	Berkas tidak ditemukan, unduh berkas gagal

q. Skenario *Use Case* Melihat Detail Sekolah

Skenario *use case* melihat detail sekolah menjelaskan alur ketika pengguna akan melihat sekolah mana saja yang ada pada sistem. Beikut ini adalah skenario *use case* melihat info sekolah yangd dapat dilihat pada Tabel III-44.

Tabel III-43 *Requirment* SKPL-F-3-004

Requirment SKPL-F-3-004
Sistem menyediakan fungsionalitas untuk melihat info sekolah

Tabel III-44 Skenario *Use Case* Melihat Detail Sekolah

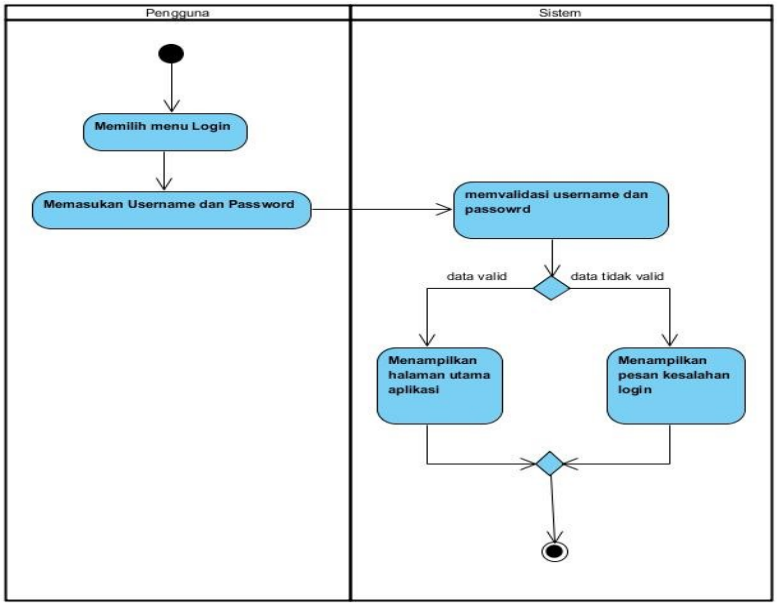
Use Case Name	Melihat detail sekolah	
Related Requirement	*Requirement* SKPL-F-3-004	
Goal in Context	Menampilkan info sekolah	
Precondition	Memilih menu sekolah	
Succes End Condition	Menampilkan info sekolah	
Failed End Condition	Data sekolah tidak tampil	
Primary Actors	Admin sekolah	
Secondary Actors	-	
Trigger	Pengguna memilih menu sekolah	
Included Cases	-	
Mainflow	Step	Action
	1	Pengguna memilih menu sekolah
	2	Sistem menampilkan data sekolah
Extensions	Step	Branching Action
	2.1	Sistem gagal menampilkan data sekolah

5. *Activity Diagram*

Activity diagram menggambarkan aktivitas yang terjadi atau interaksi antara aktor dengan aplikasi yang akan dibangun.

a. *Activity Diagram Login*

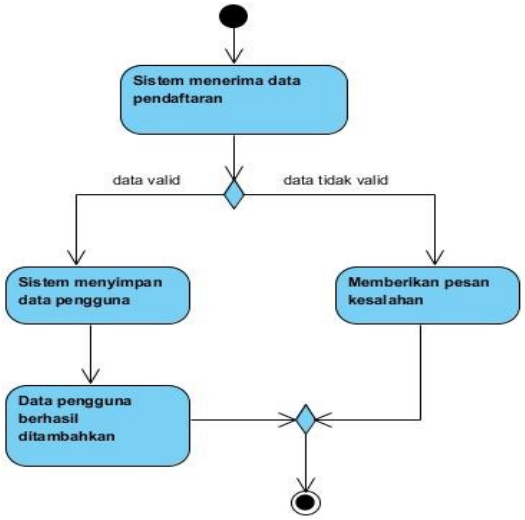
Berikut ini adalah *activity diagram login* yang dapat dilihat pada Gambar III-7.



Gambar III-7 *Activity Diagram Login*

b. *Activity Diagram* Menambah Pengguna

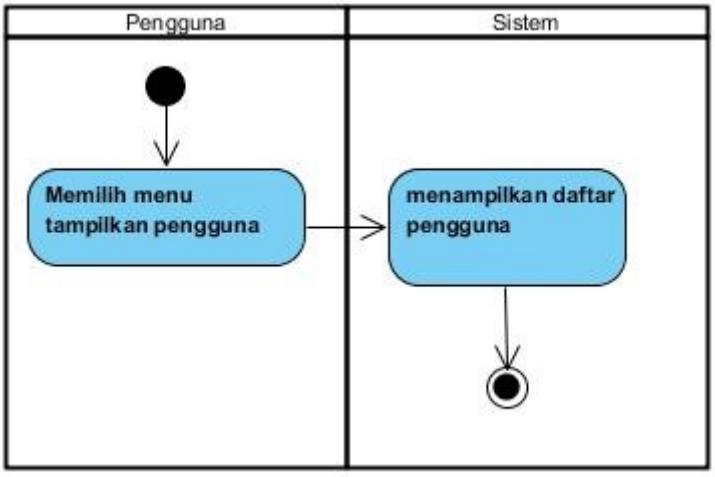
Berikut ini adalah *activity diagram* menambah pengguna yang dapat dilihat pada Gambar III-8.



Gambar III-8 Activity Diagram Menambah Pengguna

c. *Activity Diagram* Melihat Daftar Pengguna

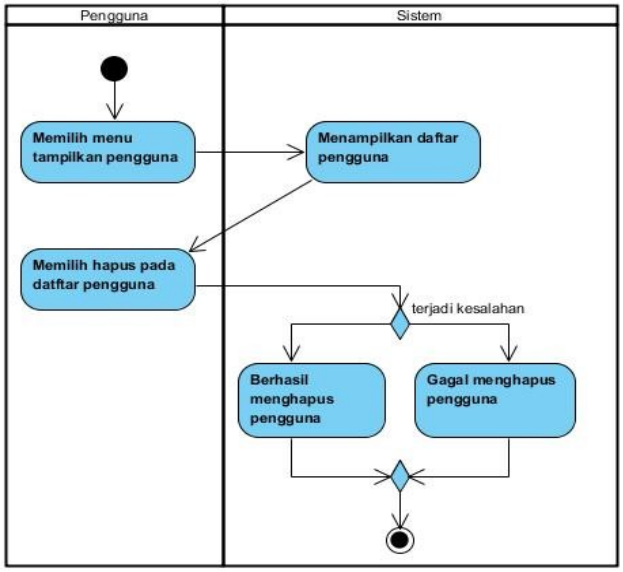
Berikut ini adalah *activity diagram* melihat daftar pengguna yang dapat dilihat pada Gambar III-9.



Gambar III-9 *Activity Diagram* Melihat Pengguna

d. *Activity Diagram* Menghapus Pengguna

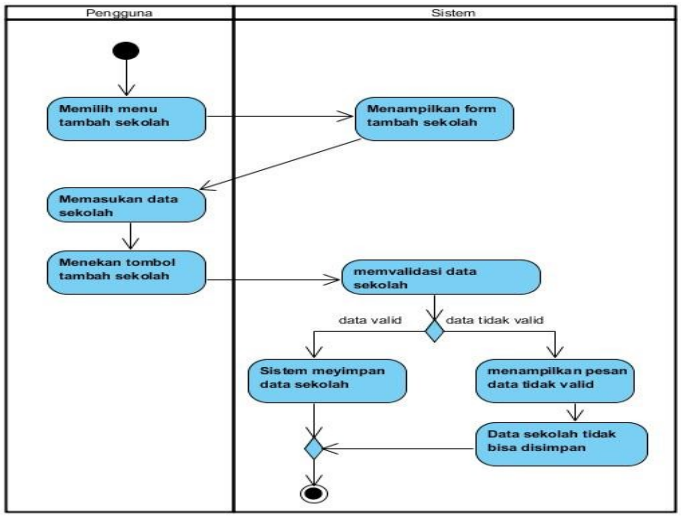
Berikut ini adalah *activity diagram* menghapus pengguna yang dapat dilihat pada Gambar III-10.



Gambar III-10 *Activity Diagram* Menghapus Pengguna

e. *Activity Diagram* Menambah Data Sekolah

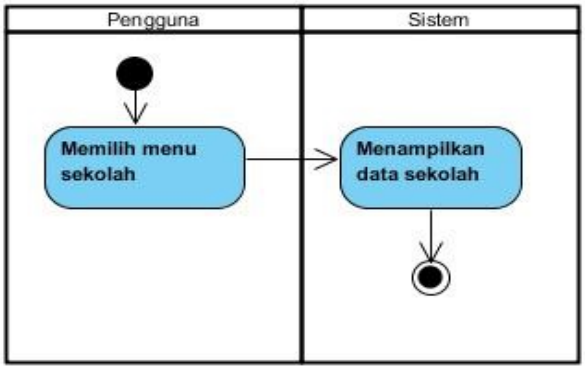
Berikut ini adalah *activity diagram* menambah data sekolah yang dapat dilihat pada Gambar III-11.



Gambar III-11 *Activity Diagram* Menambah Data Sekolah

f. *Activity Diagram* Melihat Daftar Sekolah

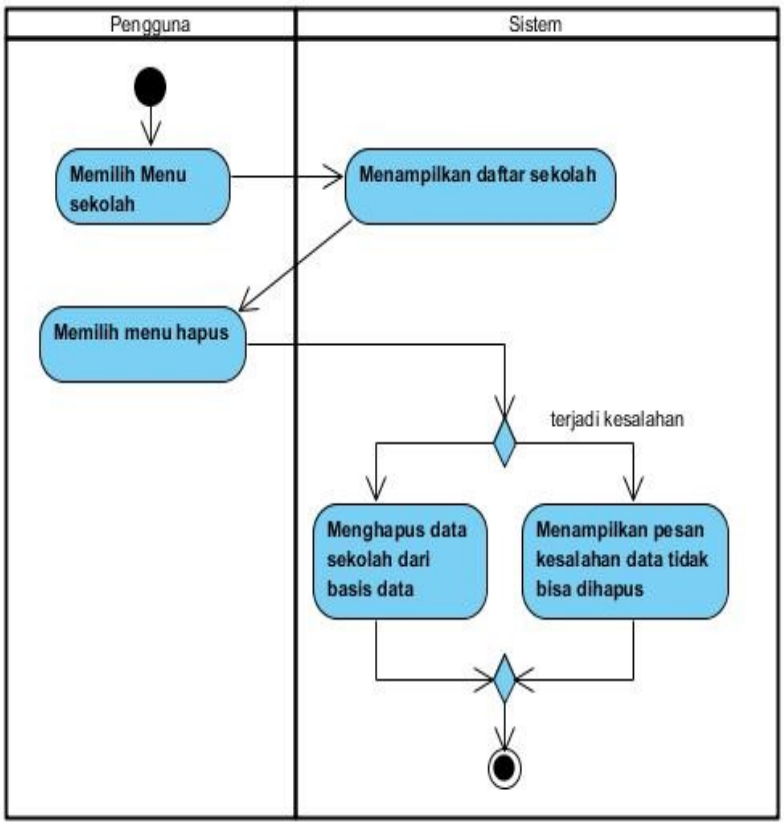
Berikut ini adalah *activity diagram* melihat daftar sekolah yang dapat dilihat pada Gambar III-12.



Gambar III-12 *Activity Diagram* Melihat Data Sekolah

g. *Activity Diagram* Mengubah Data Sekolah

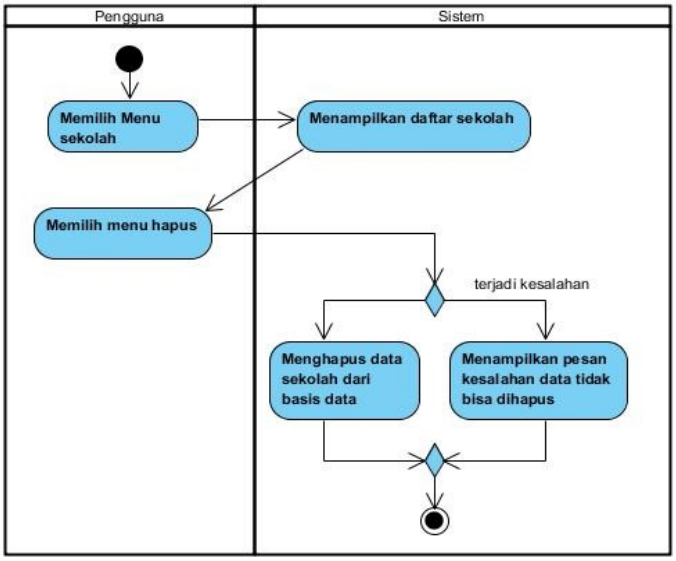
Berikut ini adalah *activity diagram* mengubah data sekolah yang dapat dilihat pada Gambar III-13.



Gambar III-13 *Activity Diagram* Mengubah Data Sekolah

h. *Activity Diagram* Menghapus Data Sekolah

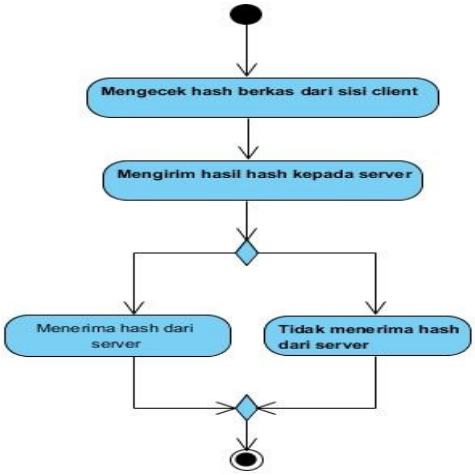
Berikut ini adalah *activity diagram* menghapus data sekolah yang dapat dilihat pada Gambar III-14.



Gambar III-14 *Activity Diagram* Menghapus Data Sekolah

i. *Activity Diagram File Hashing*

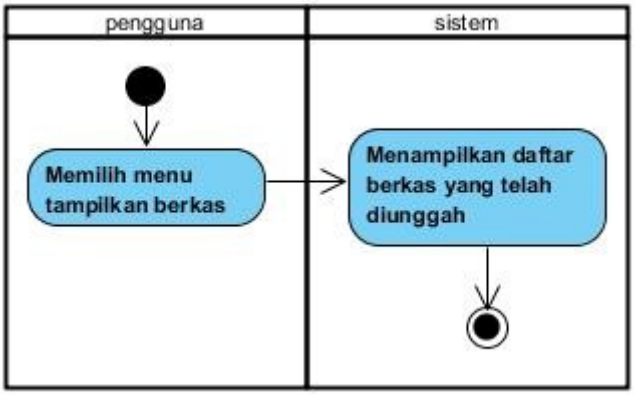
Berikut ini adalah *activity diagram file hashing* yang dapat dilihat pada Gambar III-15.



Gambar III-15 *Activity Diagram File Hashing*

j. *Activity Diagram* Melihat Daftar Berkas

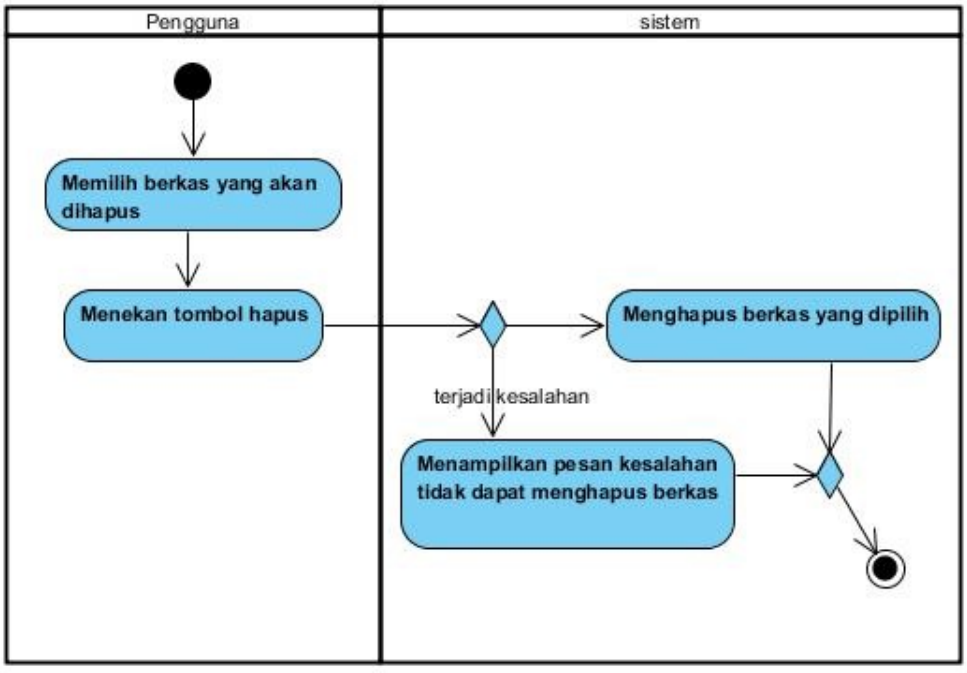
Berikut ini adalah *activity diagram* melihat daftar berkas yang dapat dilihat pada Gambar III-16.



Gambar III-16 *Activity Diagram* Melihat Berkas Materi

k. *Activity Diagram* Menghapus Berkas

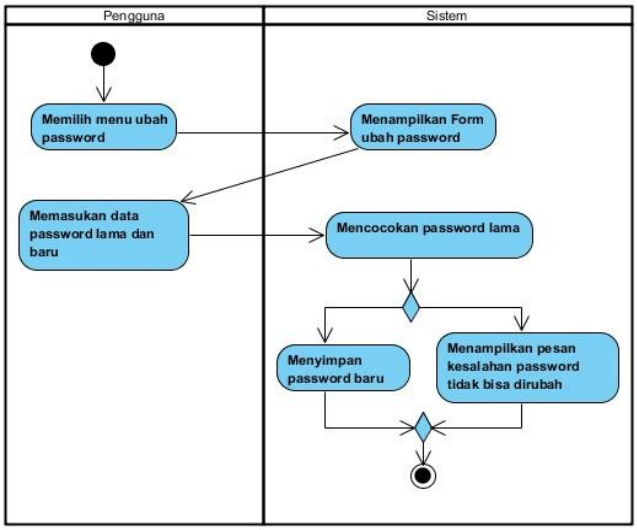
Berikut ini adalah *activity diagram* menghapus berkas yang dapat dilihat pada Gambar III-17.



Gambar III-17 *Activity Diagram* Menghapus Berkas Materi

l. *Activity Diagram* Mengubah *Password*

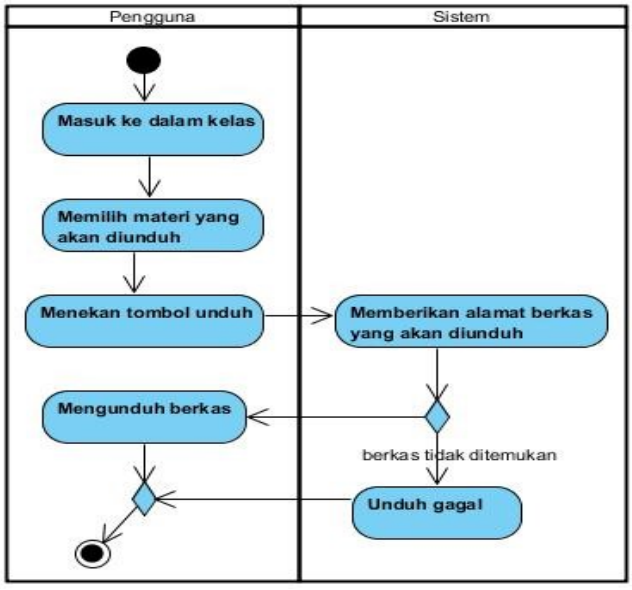
Berikut ini adalah *Activity diagram* mengubah *password* yang ada pada Gambar III-18.



Gambar III-18 *Activity Diagram* Mengubah *Password*

m. *Activity Diagram* Mengunduh Materi

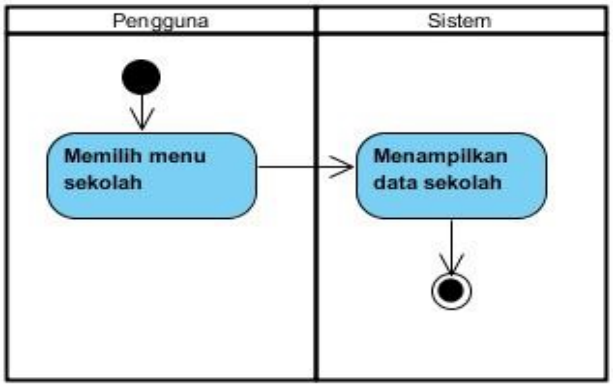
Berikut ini adalah *activity diagram* mengunduh materi yang dapat dilihat pada Gambar III-19.



Gambar III-19 *Activity Diagram* Mengunduh Materi

n. *Activity Diagram* Melihat Detail Sekolah

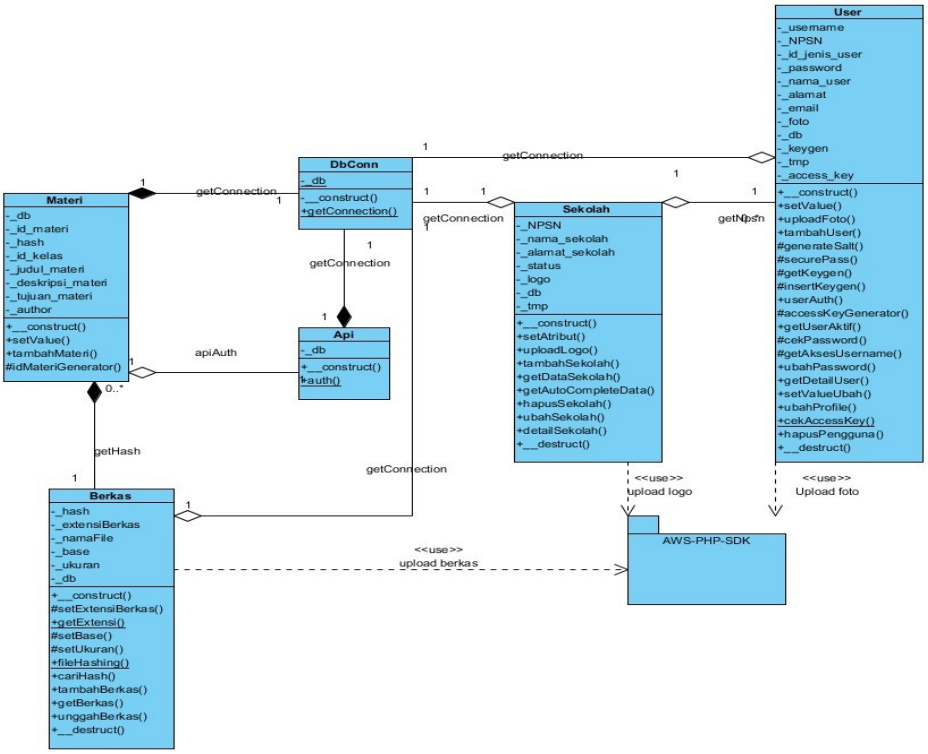
Berikut ini adalah *activity diagram* melihat info sekolah yang dapat dilihat pada Gambar III-20.



Gambar III-20 *Activity Diagram* Melihat Info Sekolah

6. *Class Diagram*

Berikut ini adalah class diagram dari perangkat lunak yang akan dibangun yang dapat dilihat pada Gambar III-21.

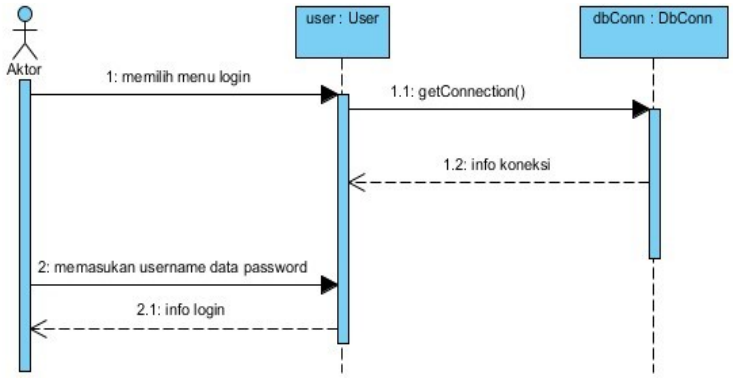


Gambar III-21 *Class Diagram*

7. *Sequence Diagram*

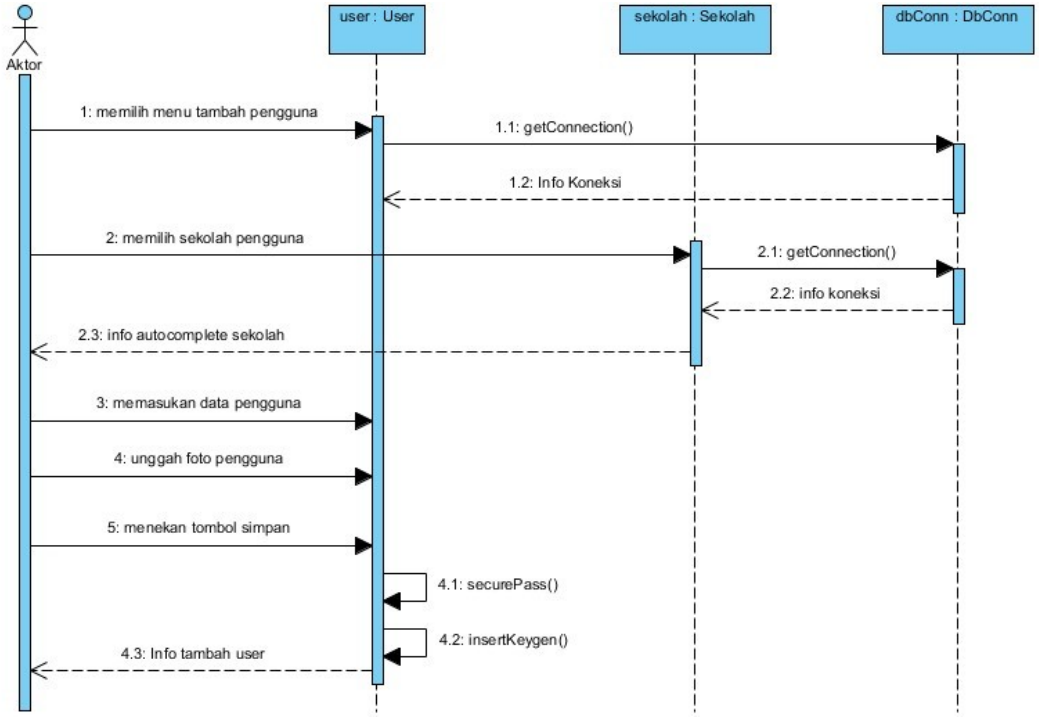
Berikut ini adalah *sequence diagram* pada sisten yang akan dibangun yang dapat dilihat pada .

a. *Use Case* : *Login*



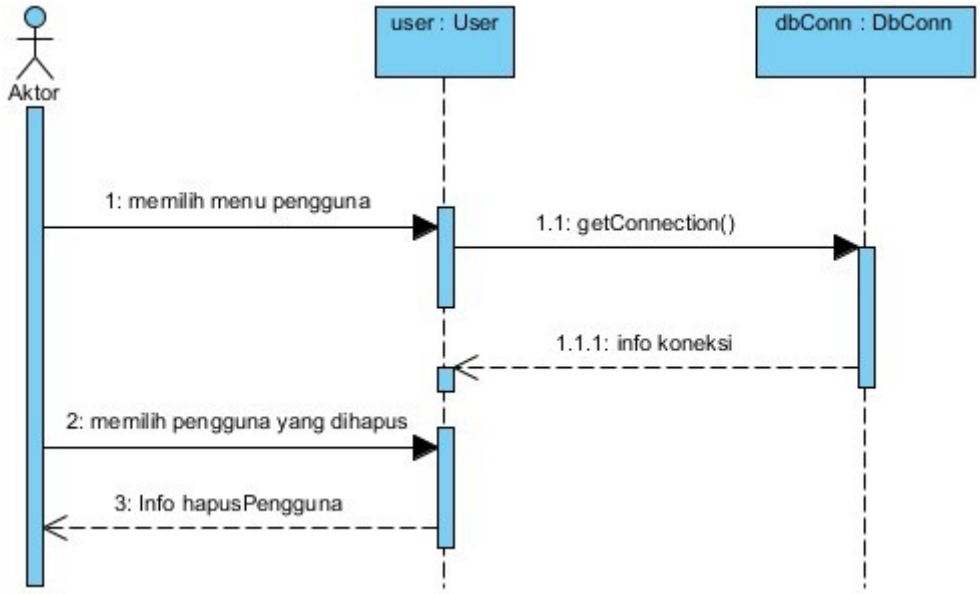
Gambar III-22 *Sequence Diagram Login*

b. *Use Case* : Menambah Pengguna



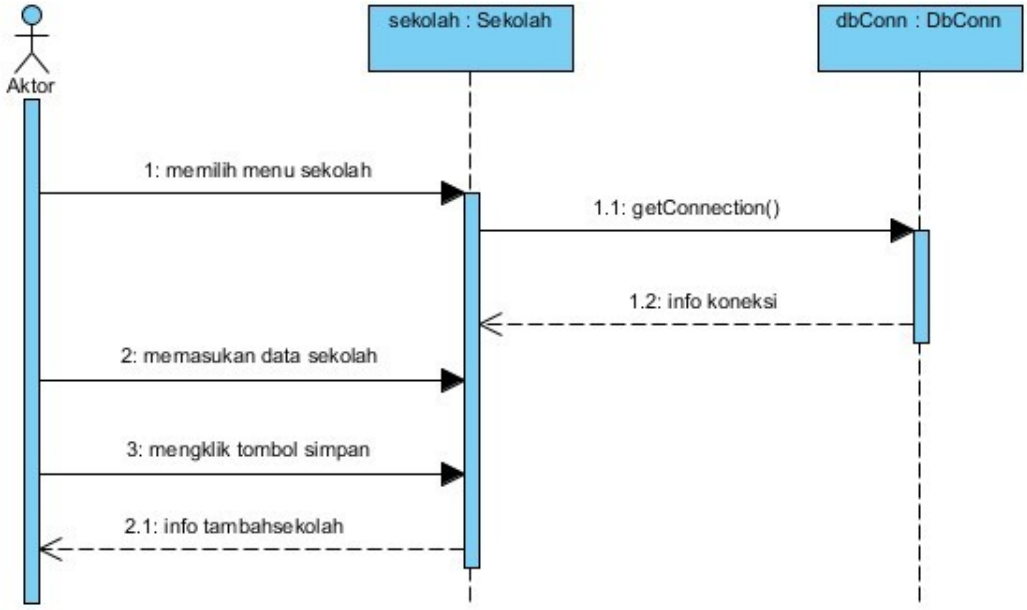
Gambar III-23 *Sequence Diagram* Pendaftaran Pengguna

c. *Use Case* : Menghapus Pengguna



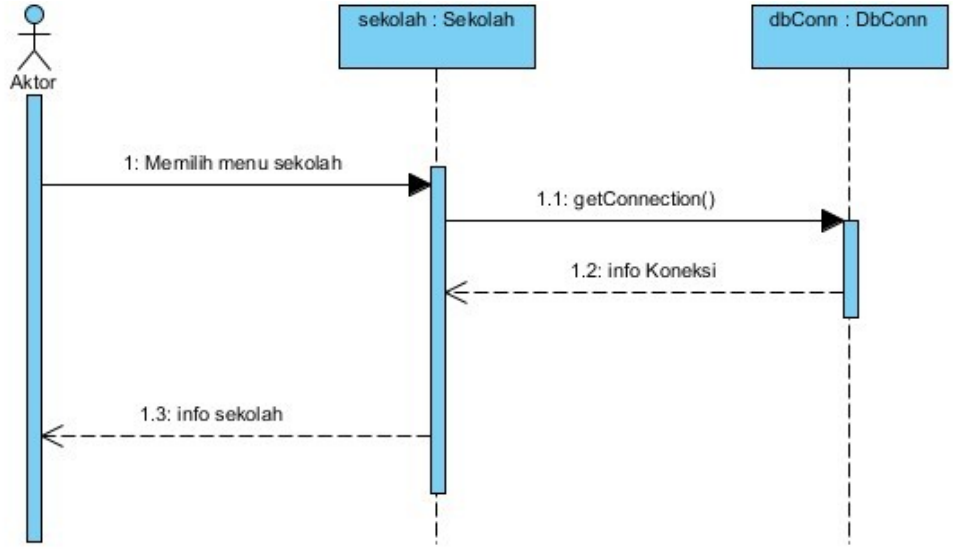
Gambar III-24 *Sequence Diagram* Menghapus Pengguna

d. *Use Case* : Menambah Data Sekolah



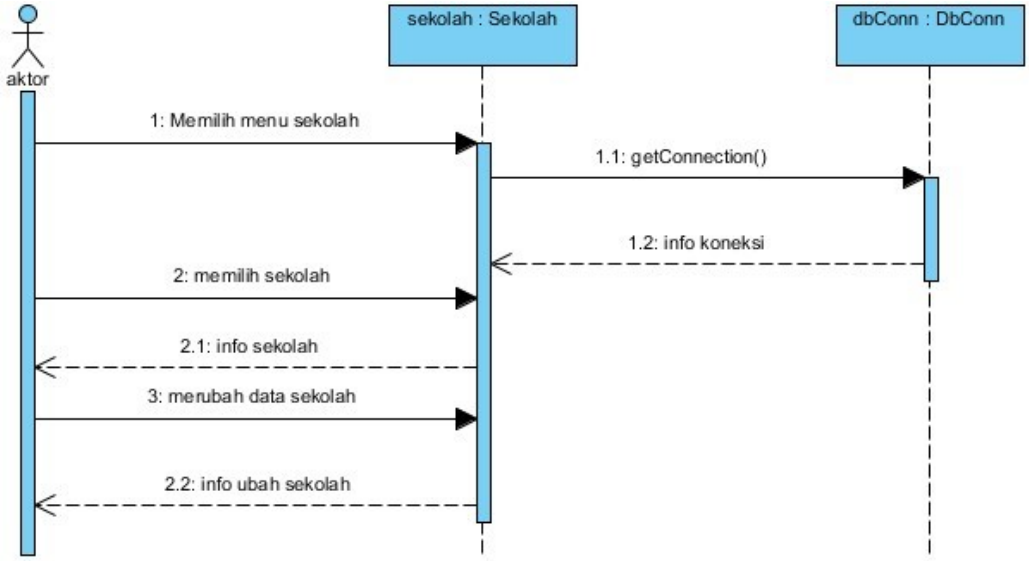
Gambar III-25 *Sequence Diagram* Menambah Data Sekolah

e. *Use Case* : Melihat daftar Sekolah



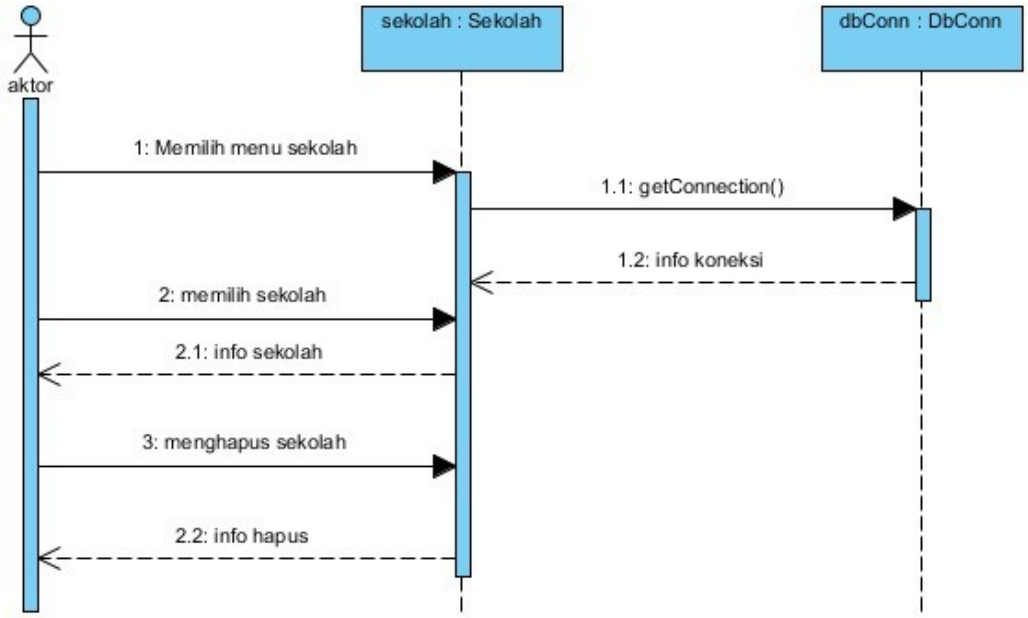
Gambar III-26 *Sequence Diagram* Melihat Info Sekolah

f. *Use Case* : Mengubah Data Sekolah



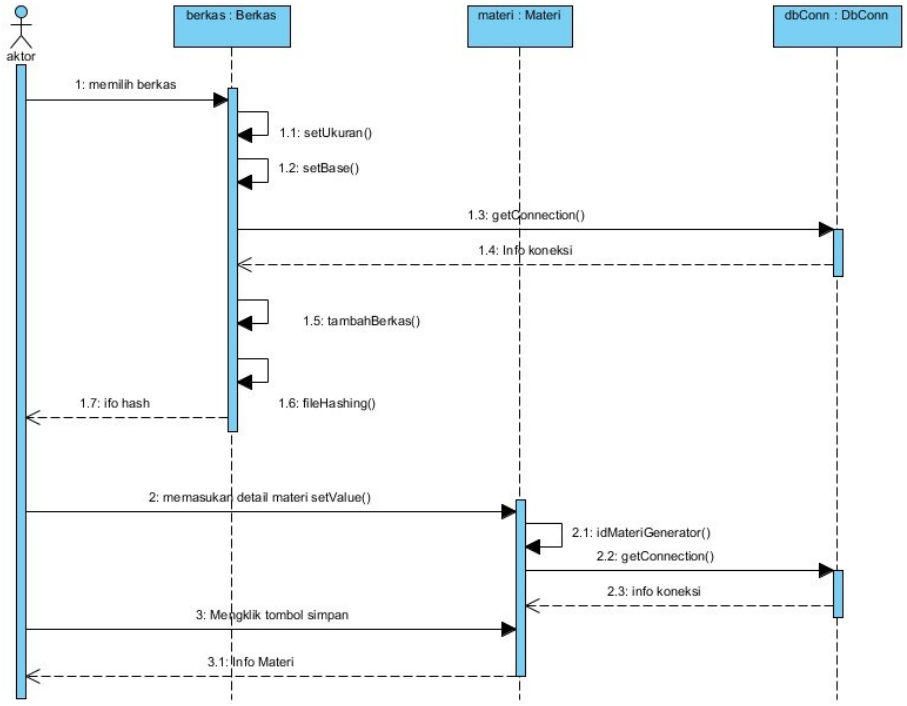
Gambar III-27 *Sequence Diagram* Mengubah Data Sekolah

g. *Use Case* : Menghapus Data Sekolah



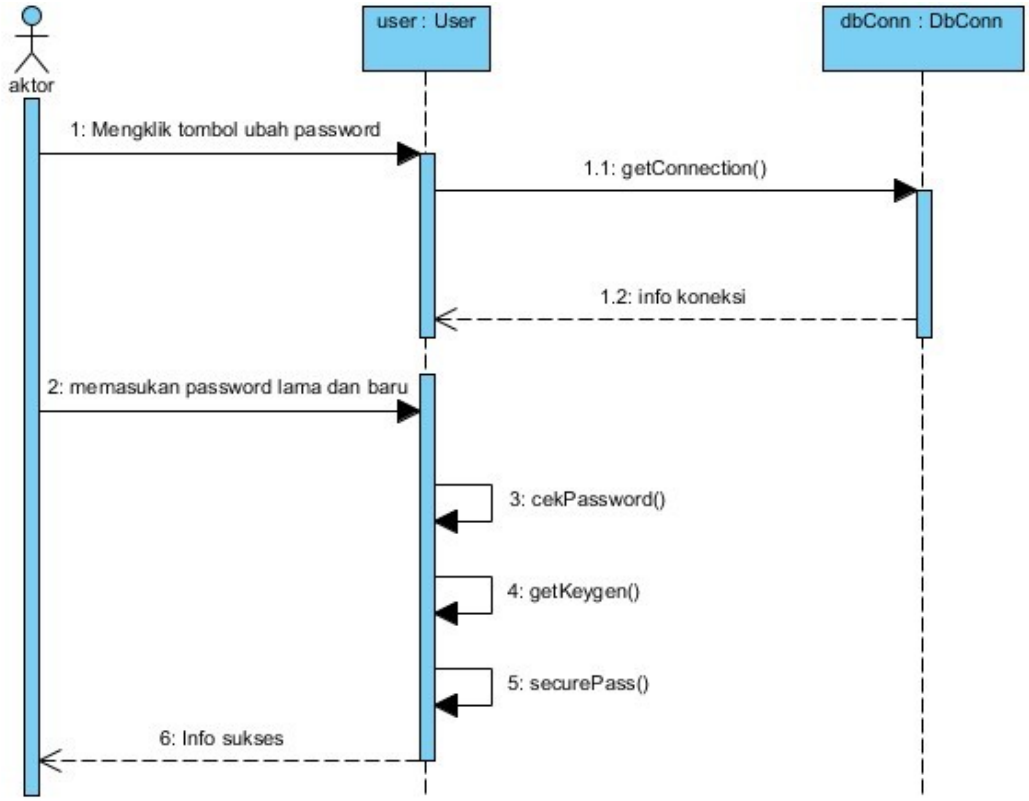
Gambar III-28 *Sequence Diagram* Menghapus Data Sekolah

h. *Use Case* : Unggah Berkas



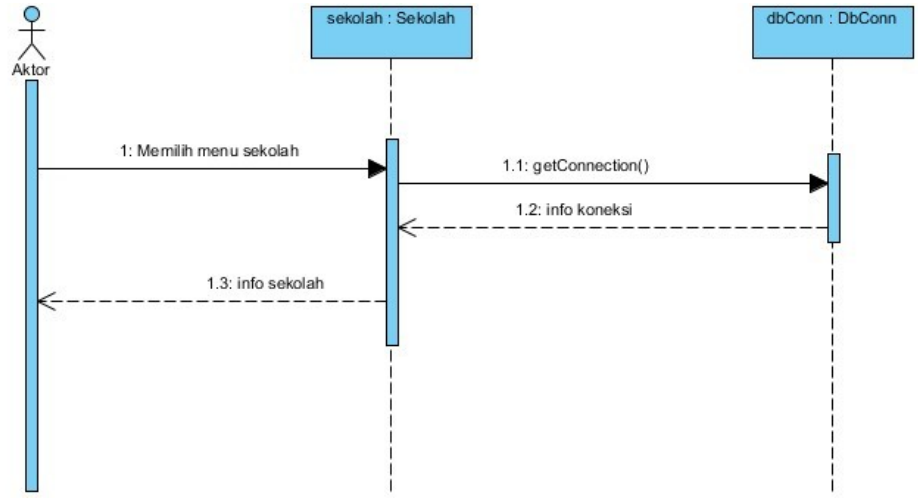
Gambar III-29 *Sequence Diagram* Unggah Berkas

i. *Use Case* : Mengubah *Password*



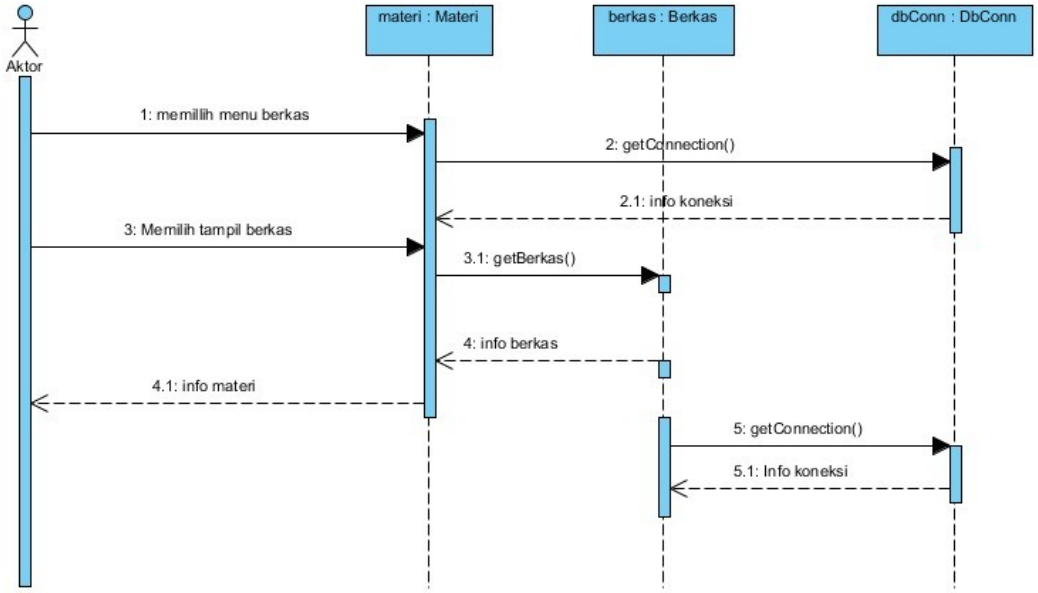
Gambar III-30 *Sequence Diagram* Mengubah *Password*

j. *Use Case* : Melihat Detail Sekolah



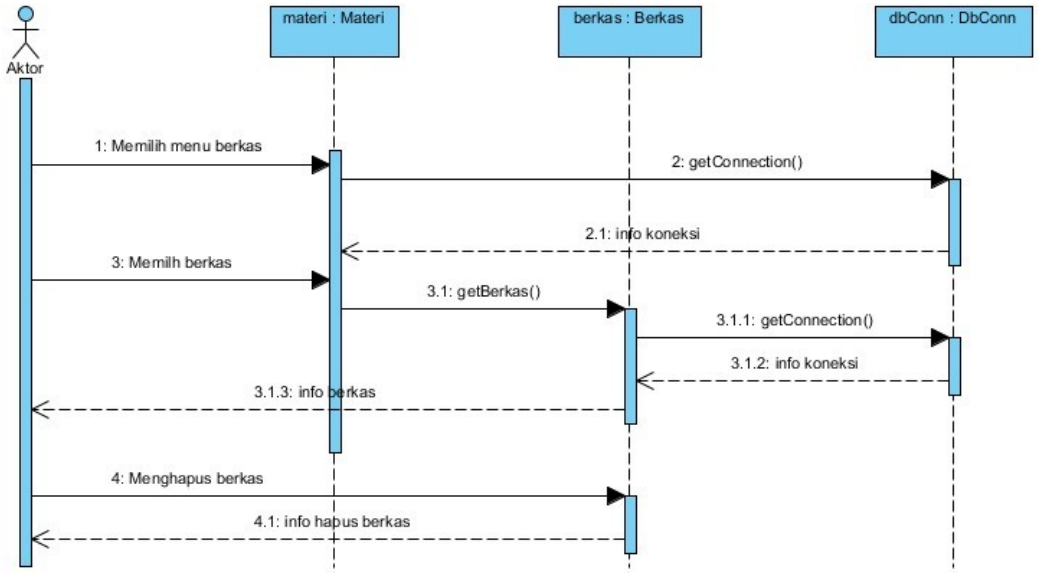
Gambar III-31 *Sequnence Diagram* Melihat Detail Sekolah

k. *Use Case* : Melihat Daftar Berkas



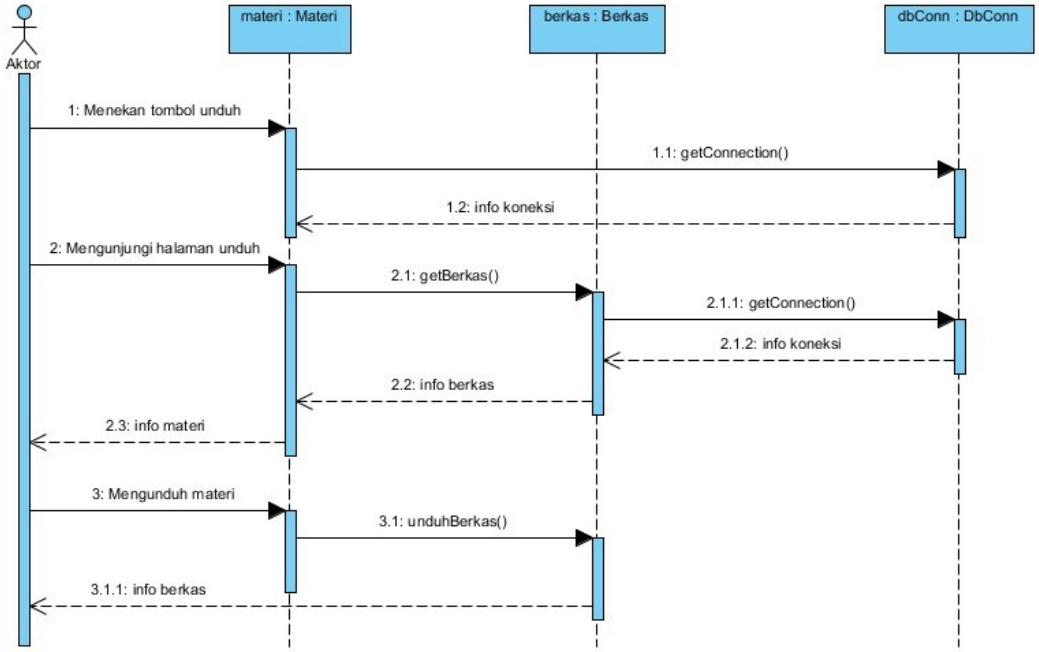
Gambar III-32 *Sequnence Diagram* Melihat Berkas Materi

l. *Use Case* : Menghapus Berkas



Gambar III-33 *Sequnence Diagram* Menghapus Berkas Materi

m. *Use Case* : Mengunduh Materi



Gambar III-34 *Sequnence Diagram* Mengunduh Materi

III.2. Perancangan Sistem

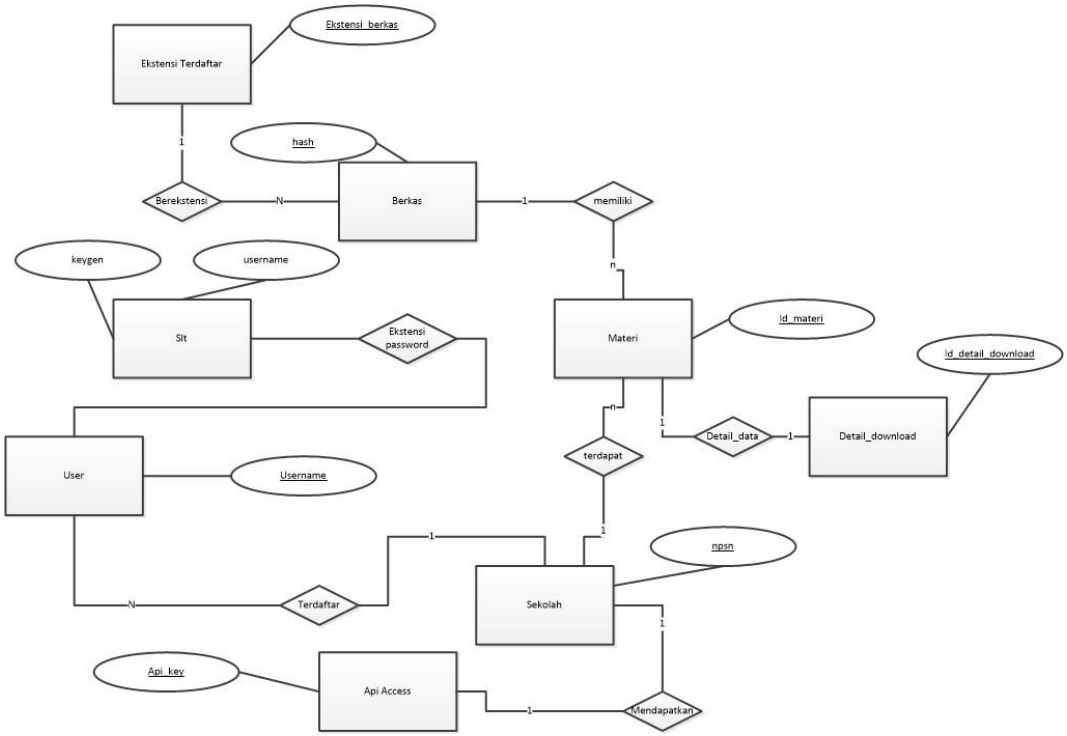
Tahap perancangan dalam pembangunan sistem ini meliputi perencanaan dan penggambaran yang meliputi beberapa elemen yang ada. Perancangan yang dibuat adalah perancangan data, perancangan arsitektur menu, perancangan antarmuka dan jaringan semantik.

III.2.1 Perancangan Data

Perancangan data pada aplikasi yang aka dibangun adalah dalam bentuk ERD *(Entity Relationship Diagram)* dan diagram relasi berikut dengan struktur tabel yang ada di dalamnya.

1. *Entity Relationship Diagram*

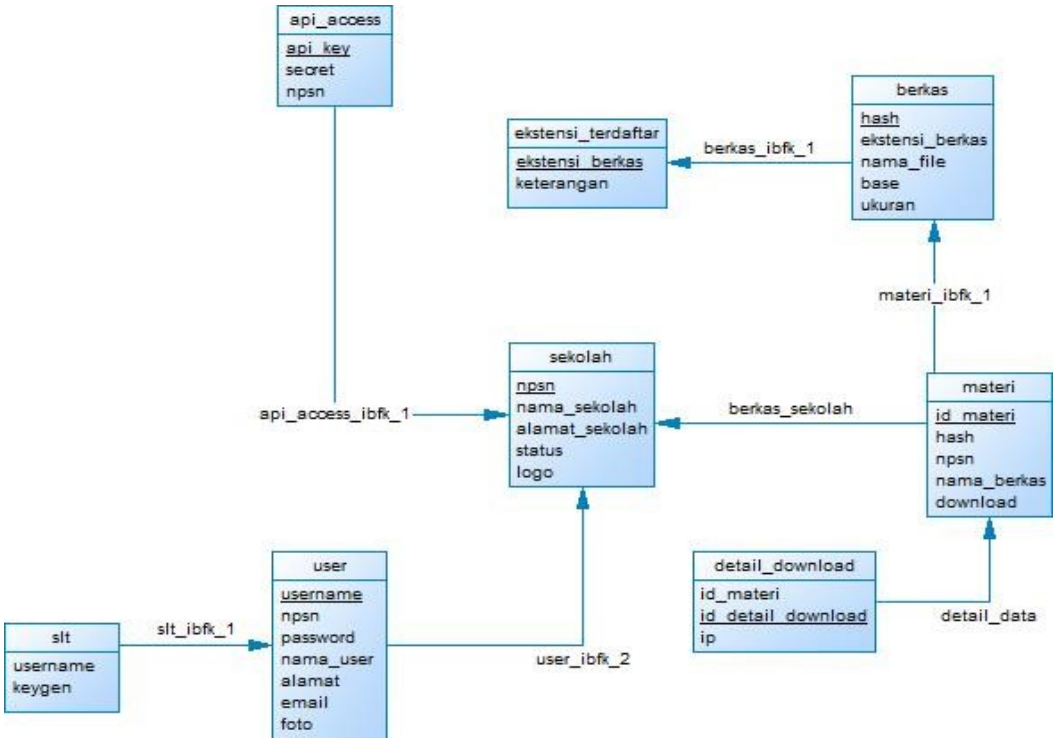
ERD *(Entity Relationship Diagram)* menggambarkan keterhubungan satu entitas data dengan entitas data lainnya, *Entity Relationship Diagram* dapat dilihat pada Gambar III-35.



Gambar III-35 *Entity Relationship Diagram*

2. Diagram Relasi

Berikut ini adalah diagram relasi dari aplikasi yang akan dibangun, diagram relasi dapat dilihat pada Gambar III-36.



Gambar III-36 Diagram Relasi

3. Struktur Tabel

Struktur tabel keseluruhan dari aplikasi yang akan dibangun dapat dilihat pada Tabel III-45 sampai dengan Tabel III-51.

a. Struktur tabel sekolah dapat dilihat pada Tabel III-45.

Nama berkas : sekolah.frm

Tempat penyimpanan : Hard Disk

Tabel III-45 Struktur Tabel Sekolah

Nama Field	Tipe Data	Ukuran	Key	Constraint
NPSN	CHAR	8	Primary key	Not Null
Nama_sekolah	VARCHAR	30	Unique	Not Null
Alamat_sekolah	VARCHAR	200		Not Null
Status	VARCHAR	8		Not Null
Logo	VARCHAR	100		Not Null

b. Struktur tabel api_access dapat dilihat pada Tabel III-46.

Nama berkas : api_access.frm

Tempat penyimpanan : Hard Disk

Tabel III-46 Struktur Tabel Api_access

Nama Field	Tipe Data	Ukuran	Key	Constraint
Api_key	VARCHAR	12	Primary key	Not Null
secret	VARCHAR	32		Not Null
NPSN	CHAR	8	Foreign Key sekolah(NPSN)	Not Null

c. Struktur tabel user dapat dilihat pada Tabel III-47.

Nama berkas : user.frm

Tempat penyimpnan : Hard Disk

Tabel III-47 Struktur Tabel User

Nama Field	Tipe Data	Ukuran	Key	Constraint
username	VARCHAR	25	Primary key	Not Null
NPSN	CHAR	8	Foreign Key sekolah(NPSN)	Not Null
Password	CHAR	64		Not Null
Nama_user	VARCHAR	35		Not Null
Alamat	VARCHAR	100		Not Null
Email	VARCHAR	30		Not Null
Foto	VARCHAR	100		Not Null

d. Struktur tabel slt dapat dilihat pada Tabel III-48.

Nama berkas : slt.frm

Tempat penyimpanan : Hard Disk

Tabel III-48 Struktur Tabel Slt

Nama Field	Tipe Data	Ukuran	Key	Constraint
Username	VARCHAR	25	Foreign Key user(username)	Not Null
Keygen	CHAR	16		Not Null

e. Struktur tabel materi dapat dilihat pada Tabel III-49.

Nama berkas : materi.frm

Tempat penyimpanan : Hard Disk

Tabel III-49 Struktur Tabel Materi

Nama Field	Tipe Data	Ukuran	Key	Constraint
Id_materi	VARCHAR	20	Primary key	Not Null
Hash	CHAR	32	Foreign Key berkas(hash)	Not Null
Npsn	CHAR	8	Foreign Key sekolah(npsn)	Not Null
Nama_berkas	VARCHAR	100		Not Null
Download	VARCHAR	200		Not Null

f. Struktur tabel berkas dapat dilihat pada Tabel III-50.

Nama berkas : berkas.frm

Lokasi penyimpanan : Hard Disk

Tabel III-50 Struktur Tabel Berkas

Nama Field	Tipe Data	Ukuran	Key	Constraint
Hash	CHAR	32	Primary Key	Not Null
Ekstensi_berkas	VARCHAR	6	Foreign Key Ekstensi_terdaftar(ekstensi_berkas)	Not Null
Nama_file	VARCHAR	30		Not Null
Base	VARCHAR	100		Not Null
Ukuran	INTEGER	11		Not Null

g. Struktur tabel ekstensi_terdaftar dapat dilihat pada Tabel III-51.

Nama berkas : ekstensi_terdaftar.frm

Tempat penyimpanan : Hard Disk

Tabel III-51 Struktur Tabel Ekstensi_terdaftar

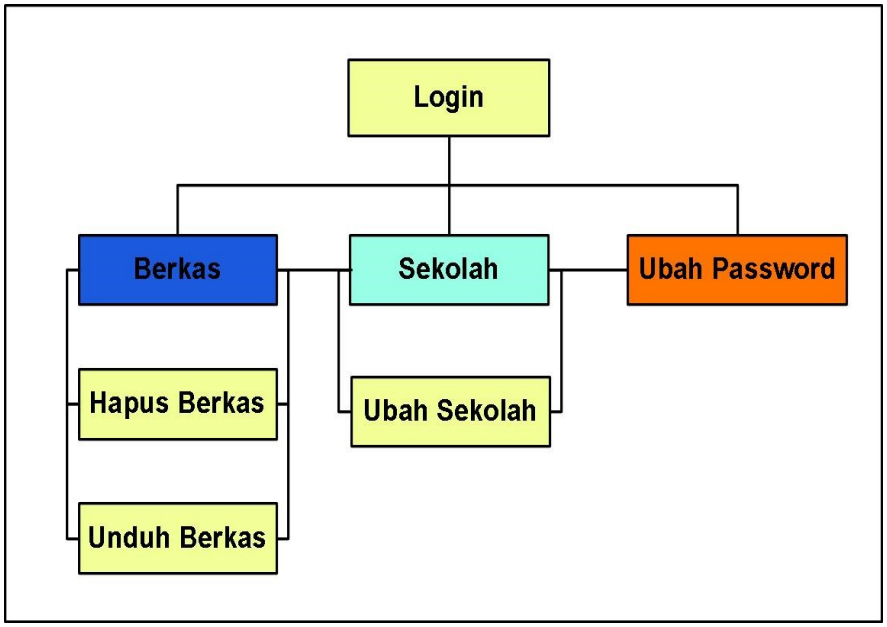
Nama Field	Tipe Data	Ukuran	Key	Constraint
Ekstensi_berkas	VARCHAR	6	Primary Key	Not Null
Keterangan	VARCHAR	100		Not Null

III.2.2 Perancangan Arsitektur Menu

Perancangan arsitektur menu pada sistem yang akan dibangun terdapat dua bagian yaitu, perancangan arsitektur menu admin sekolah dan perancangan arsitektur menu administrator.

1. Perancangan Arsitektur Menu Admin Sekolah

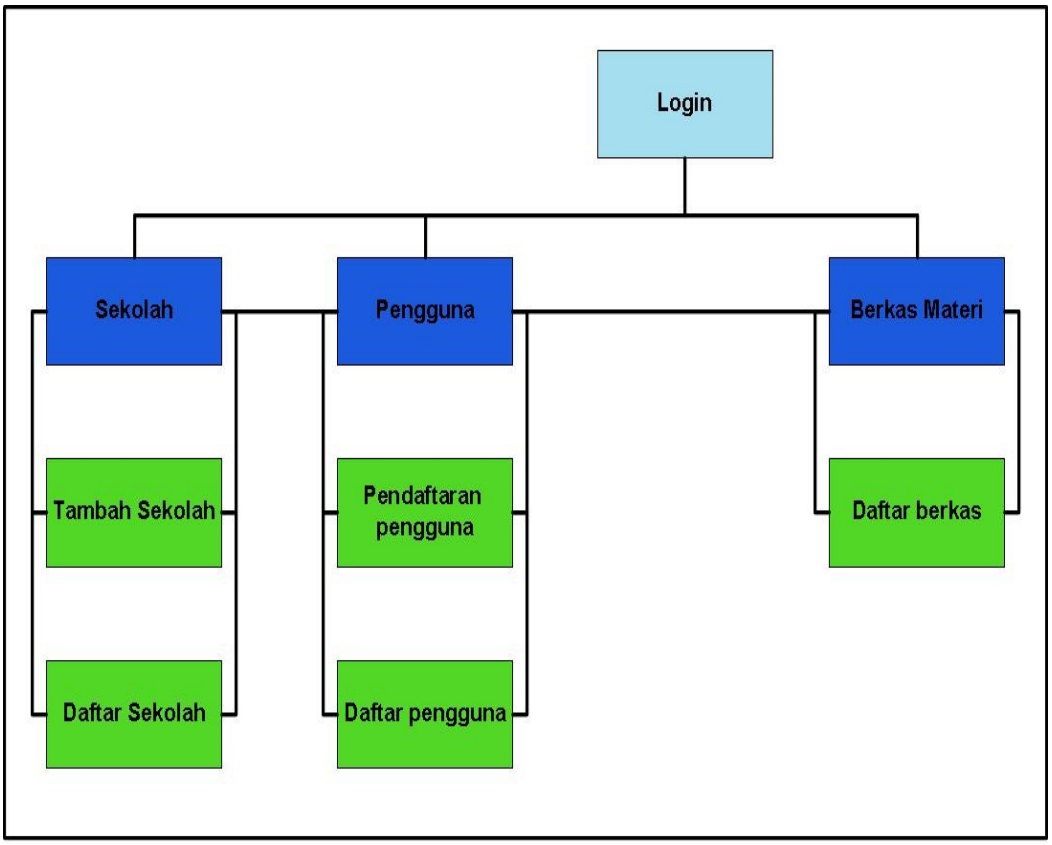
Berikut ini adalah perancangan arsitektur menu Admin Sekolah yang dapat dilihat pada Gambar III-37.



Gambar III-37 Arsitektur Menu Admin Sekolah

2. Perancangan Arsitektur Menu Administrator

Berikut ini adalah perancangan arsitektur menu administrator yang dapat dilihat pada Gambar III-38.



Gambar III-38 Arsitektur Menu Administrator

III.2.3 Perancangan Antarmuka

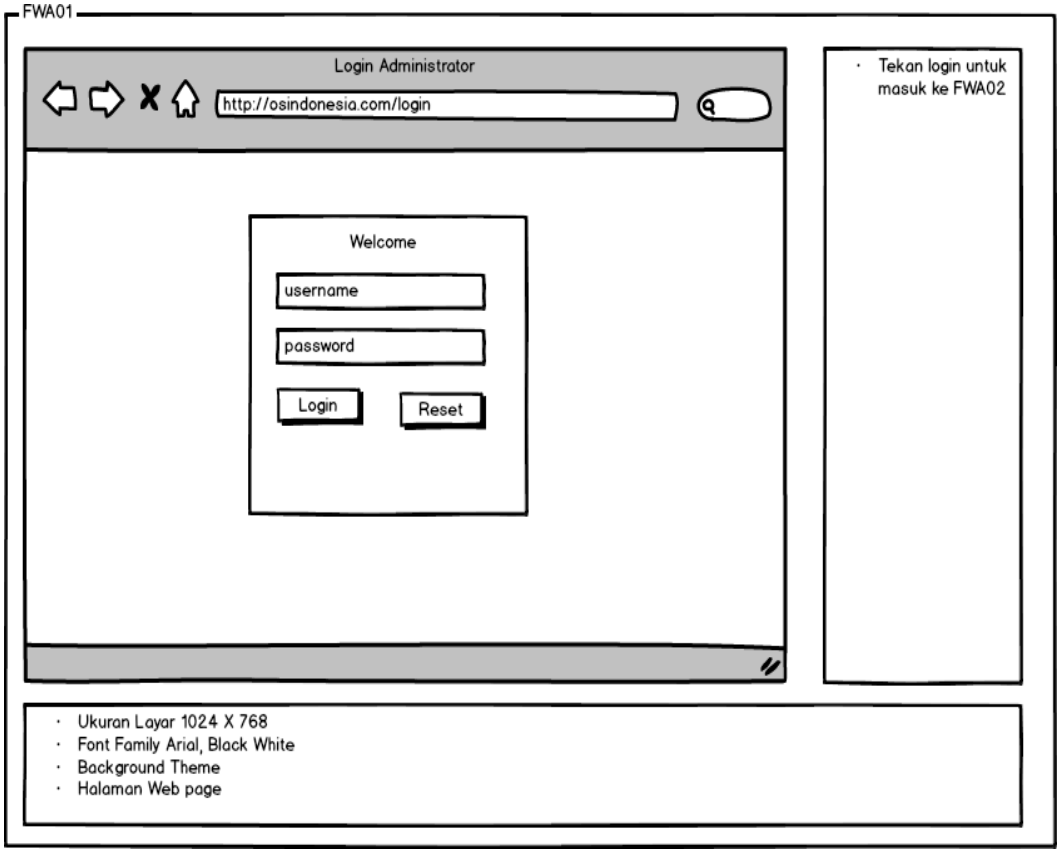
Perancangan antarmuka merupakan desain dari antarmuka aplikasi yang akan dibangun. Perancangan antarmuka pada aplikasi yang akan dibangun terdiri dari perancangan antarmuka untuk administrator dan perancangan antarmuka pengajar. Berikut ini adalah perancangan antarmuka administrator dan perancangan pengajar yang dapat dilihat pada bahasan selanjutnya.

1. Perancangan Antarmuka Administrator

Perancangan antarmuka administrator terdiri dari beberapa form dengan rancangan berbentuk web. Berikut ini adalah perancangan antarmuka administrator:

a. Perancangan *form* login

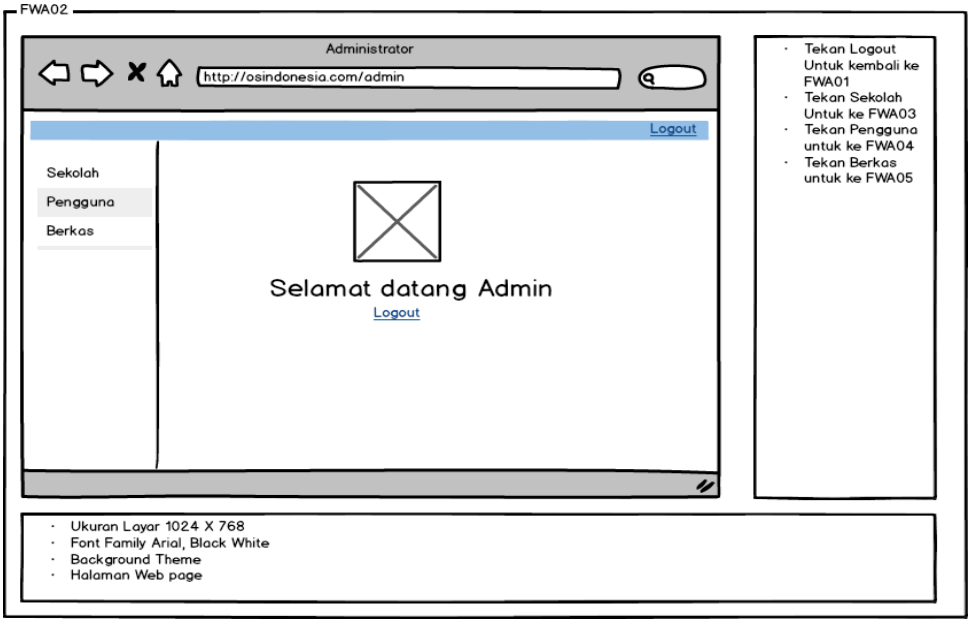
Berikut ini adalah perancangan *form* login yang dapat dilihat pada Gambar III-39.



Gambar III-39 Perancangan *Form* Login

b. Perancangan halaman admin

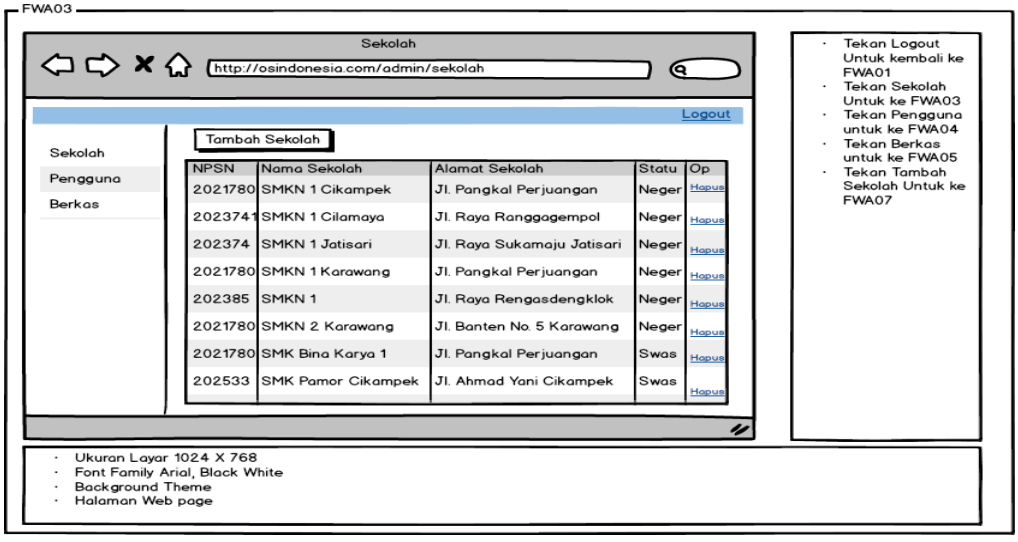
Berikut ini adalah perancangan halaman admin yang dapat dilihat pada Gambar III-40.



Gambar III-40 Perancangan Halaman Admin

c. Perancangan halaman sekolah

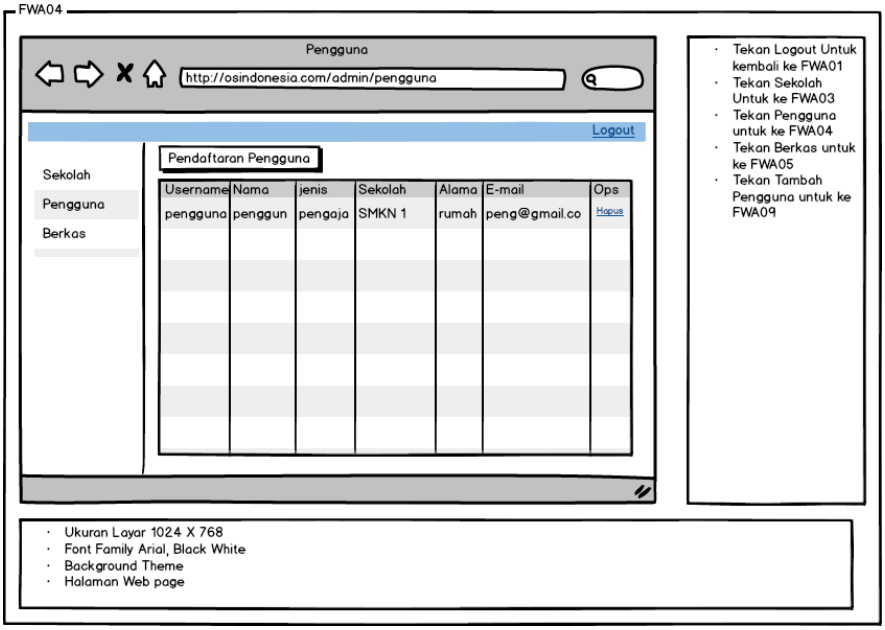
Berikut ini adalah perancangan halaman sekolah yang dapat dilihat pada Gambar III-41.



Gambar III-41 Perancangan Halaman Sekolah

d. Perancangan halaman pengguna

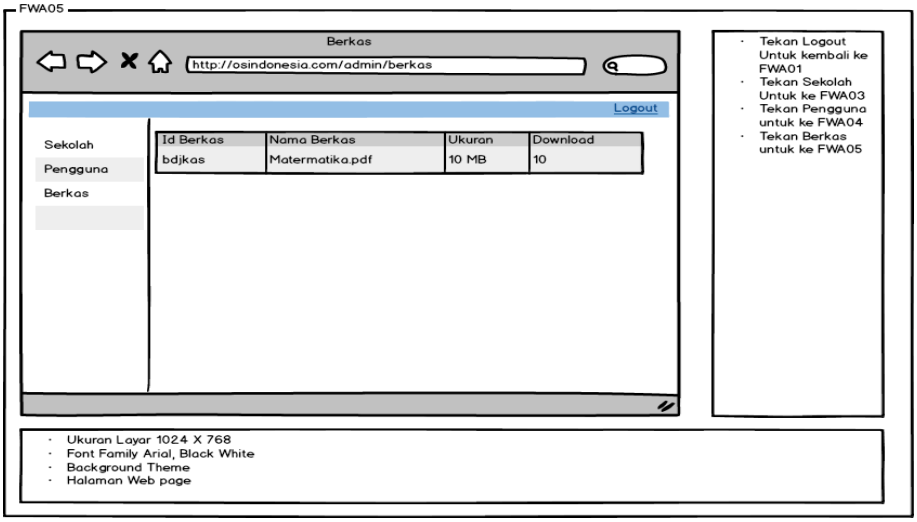
Berikut ini adalah perancangan halaman pengguna yang dapat dilihat pada Gambar III-42.



Gambar III-42 Perancangan Halaman Pengguna

e. Perancangan halaman berkas

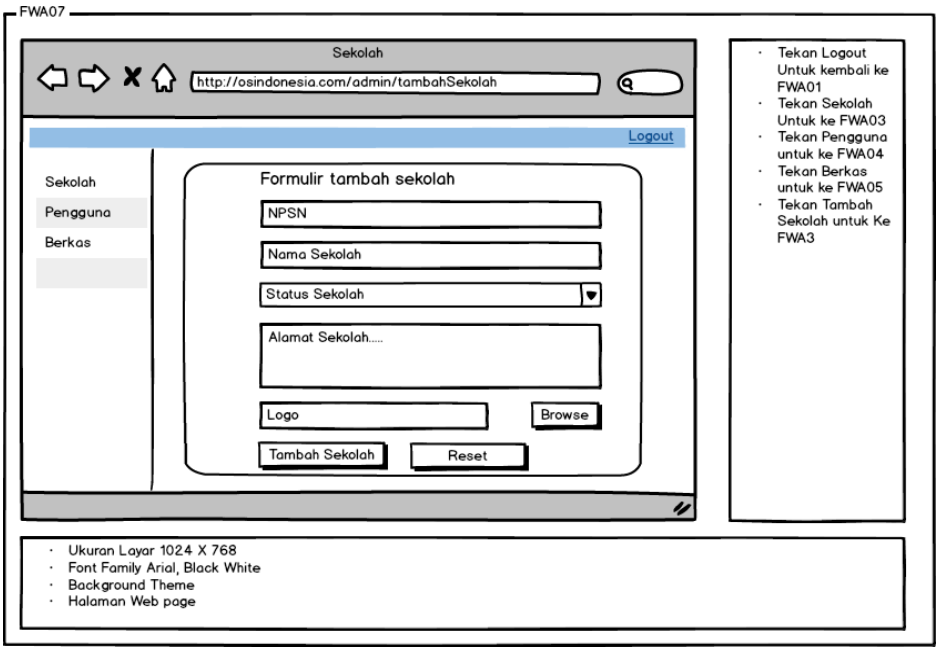
Berikut ini adalah perancangan halaman berkas yang dapat dilihat pada Gambar III-43.



Gambar III-43 Perancangan Halaman Berkas

f. Perancangan *form* tambah sekolah

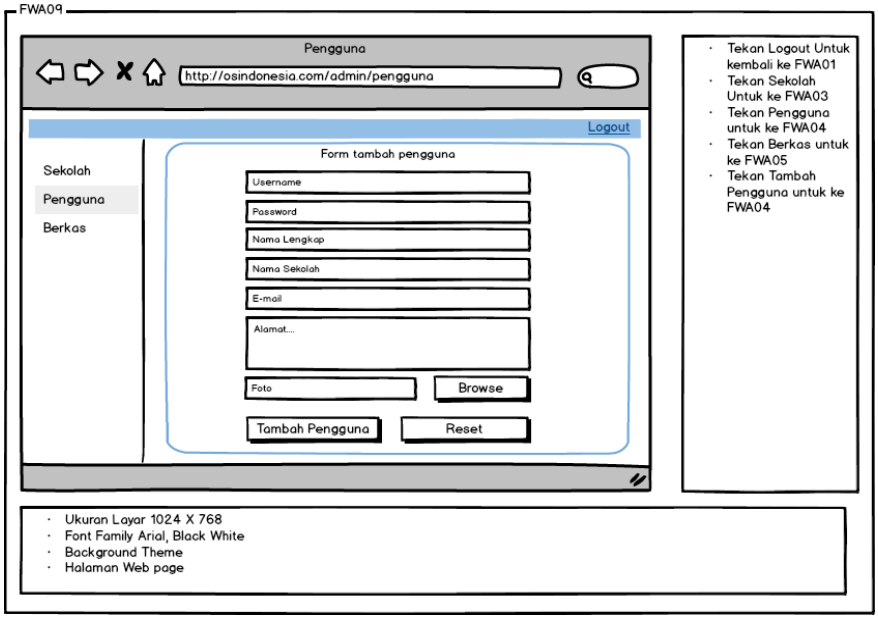
Berikut ini adalah perancangan *form* tambah sekolah yang dapat dilihat pada Gambar III-44.



Gambar III-44 Perancangan *Form* Tambah Sekolah

g. Perancangan *form* tambah pengguna

Berikut ini adalah perancangan *form* ubah pengguna yang dapat dilihat pada Gambar III-45.



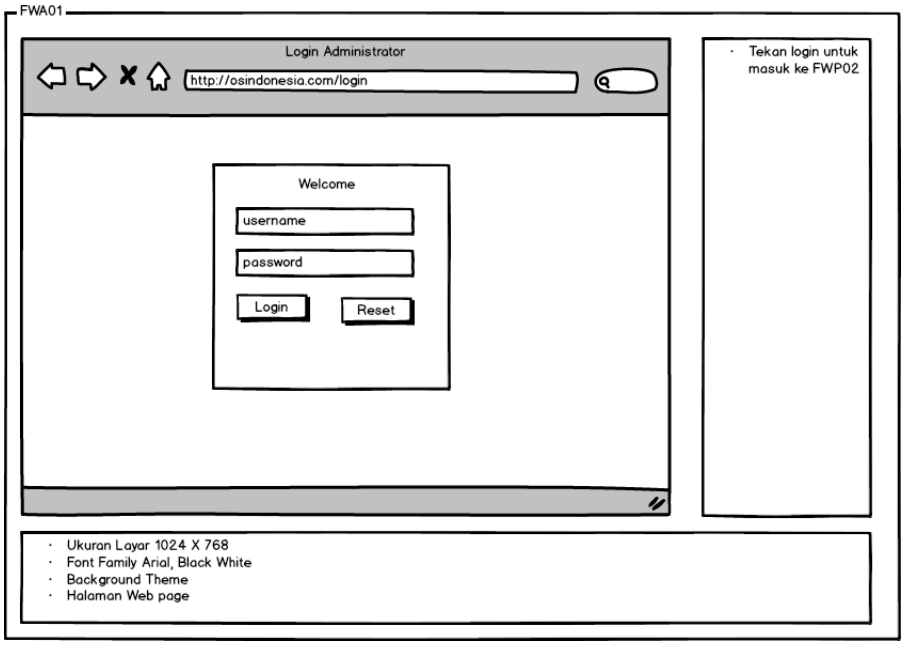
Gambar III-45 Perancangan *Form* Tambah Pengguna

2. Perancangan Antarmuka Admin Sekolah

Perancangan antarmuka Admin Sekolah terdiri dari beberapa halaman dan form untuk melakukan penambahan

a. Perancangan *form login*

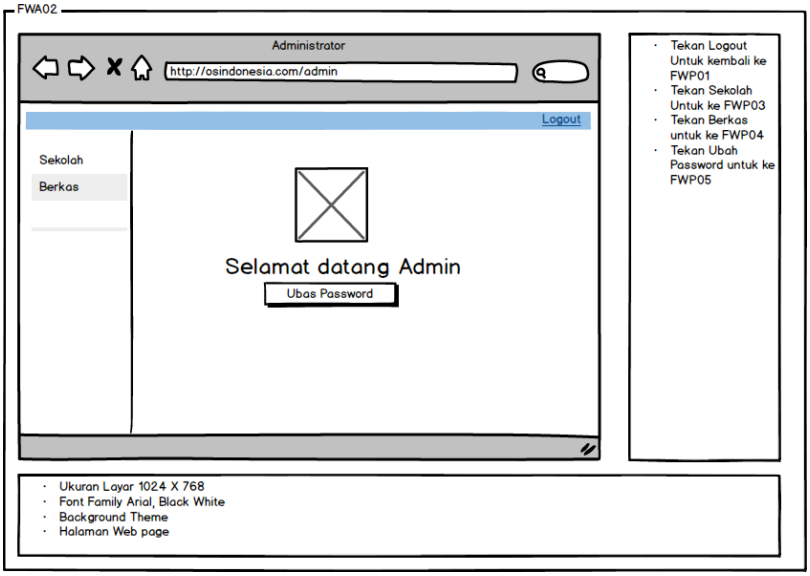
Berikut ini adalah perancangan antarmuka *form login* yang dapat dilihat pada Gambar III-46 .



Gambar III-46 Perancangan *Form Login*

b. Perancangan *dashboard* admin sekolah

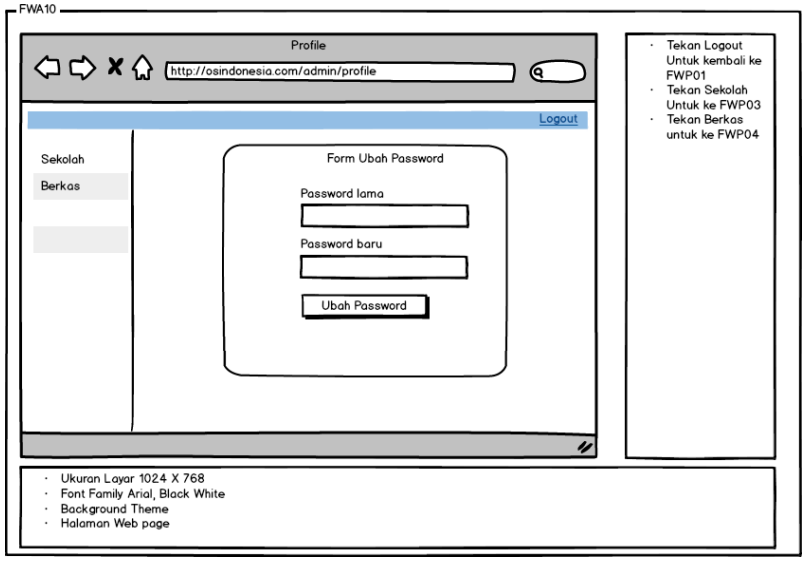
Berikut ini adalah perancangan antarmuka *dashboard* admin sekolah yang dapat dilihat pada Gambar III-47.



Gambar III-47 Perancangan *Dashboard* Admin Sekolah

c. Perancangan *form* ubah *password*

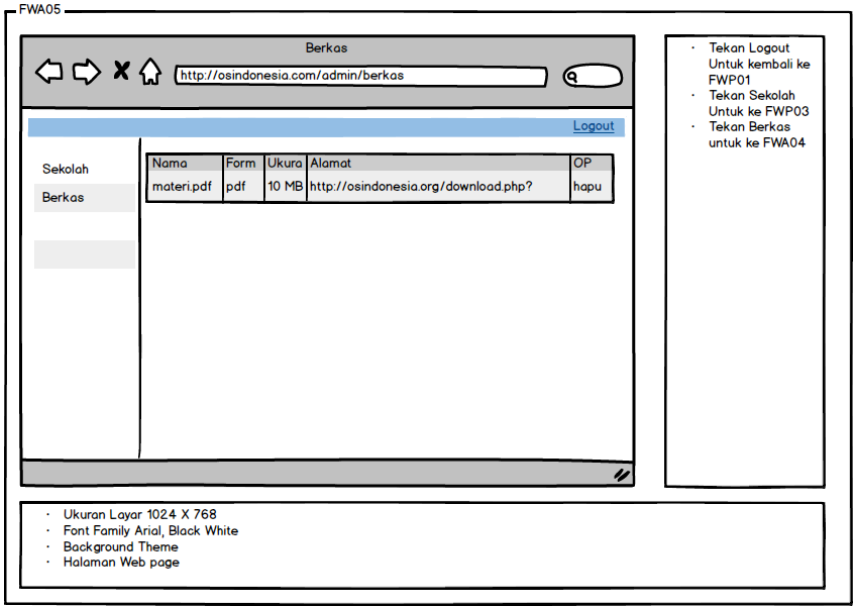
Berikut ini adalah perancangan antarmuka *form* ubah *password* yang dapat dilihat pada Gambar III-48.



Gambar III-48 Perancangan *Form* Ubah *Password*

d. Perancangan halaman berkas

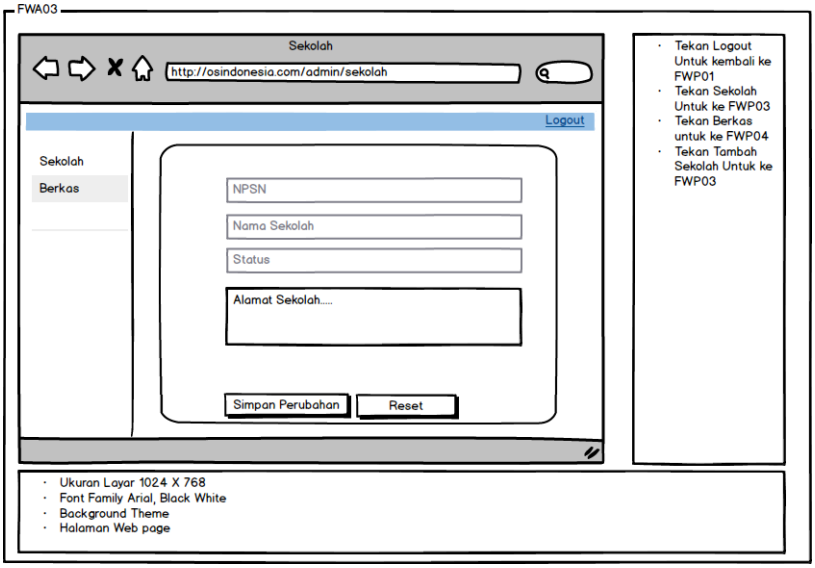
Berikut ini adalah perancangan antarmuka halaman berkas yang dapat dilihat pada Gambar III-49.



Gambar III-49 Perancangan Halaman Berkas

e. Perancangan halaman ubah sekolah

Berikut ini adalah perancangan antarmuka halaman sekolah yang dapat dilihat pada .



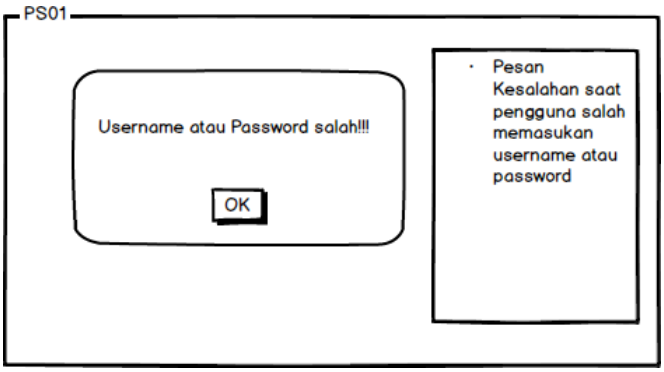
Gambar III-50 Perancangan Halaman Sekolah

3. Perancangan Pesan

Perancangan pesan yang ada pada sistem yang akan dibangun menampilkan pesan pada setiap proses yang membutuhkan bantuan, seperti pesan kesalahan pada saat menambah data ataupun pada saat *login* ke dalam sistem. Berikut ini adalah perancangan pesan pada sistem yang akan dibangun:

a. Perancangan pesan kesalahan pada saat *login*

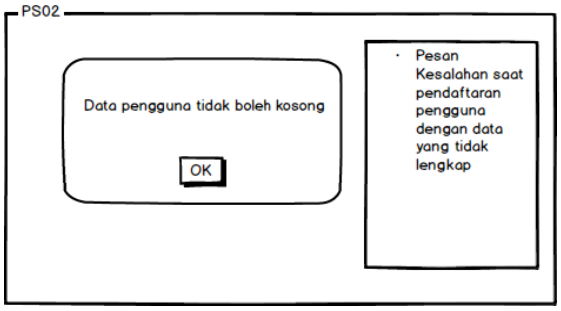
Berikut ini adalah pesan ketika pengguna salah memasukan *username* atau *password* pada saat *login* yang dapat dilihat pada Gambar III-51.



Gambar III-51 Perancangan Pesan Gagal *Login*

b. Perancangan pesan data pengguna tidak boleh kosong

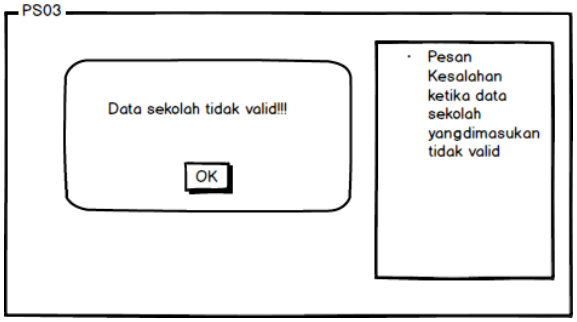
Berikut ini adalah pesan ketika data pengguna tidak lengkap pada saat menambah pengguna yang dapat dilihat pada Gambar III-52.



Gambar III-52 Perancangan Pesan Data Pengguna Tidak Lengkap

c. Perancangan pesan data sekolah tidak valid

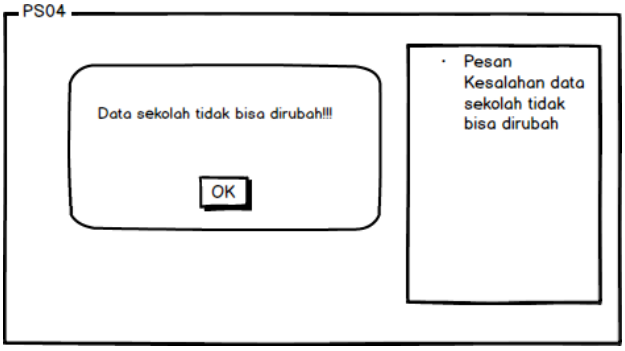
Berikut ini adalah pesan data sekolah tidak valid atau salah ketika menambahkan data sekolah ke dalam sistem yang akan dibangun yang dapat dilihat pada Gambar III-53.



Gambar III-53 Perancangan Pesan Data Sekolah Tidak Valid

d. Perancangan pesan data sekolah tidak bisa dirubah

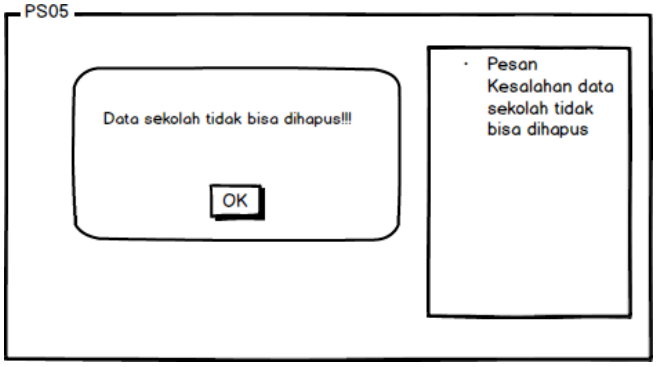
Berikut ini adalah pesan ketika data sekolah tidak dapat diubah atau terjadi kesalahan pada saat mengubah data sekolah yang dapat dilihat pada Gambar III-54.



Gambar III-54 Perancangan Pesan Data Sekolah Tidak Bisa Dirubah

e. Perancangan pesan data sekolah tidak bisa dihapus

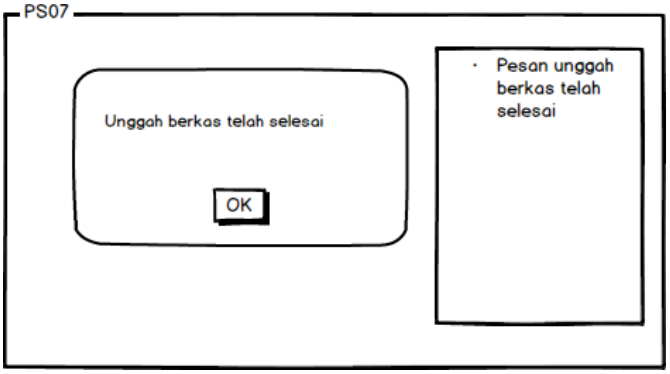
Berikut ini adalah pesan ketika data sekolah tidak dapat dihapus yang dapat dilihat pada Gambar III-55.



Gambar III-55 Perancangan Pesan Data Sekolah Tidak Bisa Dihapus

f. Perancangan pesan unggah berkas telah selesai

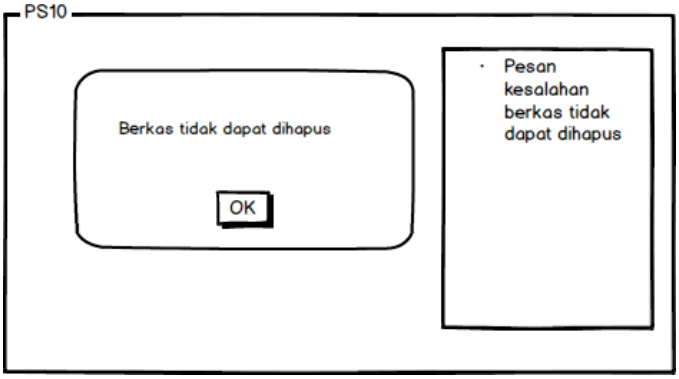
Berikut ini adalah pesan saat unggah berkas telah selesai yang dapt dilihat pada Gambar III-56.



Gambar III-56 Perancangan Pesan Unggah Berkas Telah Selesai

g. Perancangan pesan berkas tidak bisa dihapus

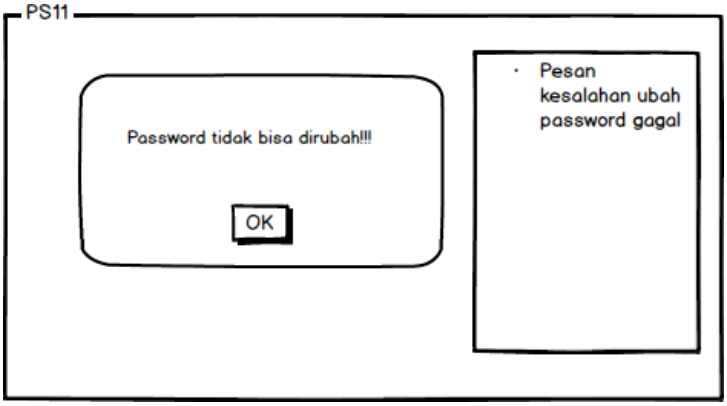
Berikut ini adalah pesan berkas yang tidak bisa hapus pada saat penghapusan berkas oleh pengguna yang dapat dilihat pada Gambar III-57.



Gambar III-57 Perancangan Pesan Berkas Tidak Dapat Dihapus

h. Perancangan pesan ubah *password* gagal

Berikut ini adalah pesan ketika pengguna gagal merubah password yangd apat dilihat pada Gambar III-58.

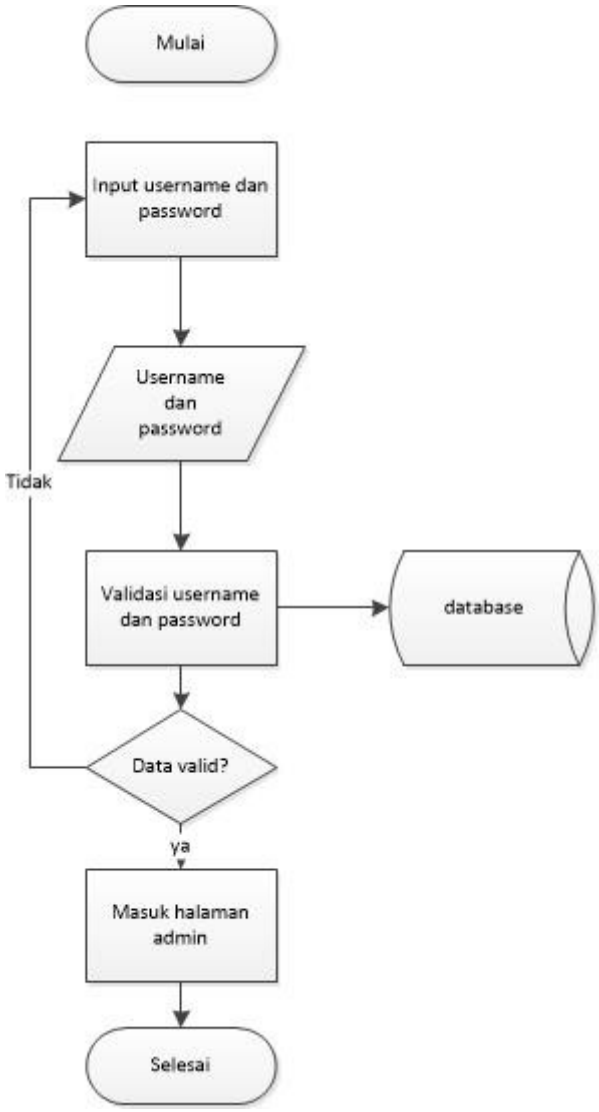


Gambar III-58 Perancangan Pesan *Password* Gagal Dirubah

III.2.4 Perancangan Method

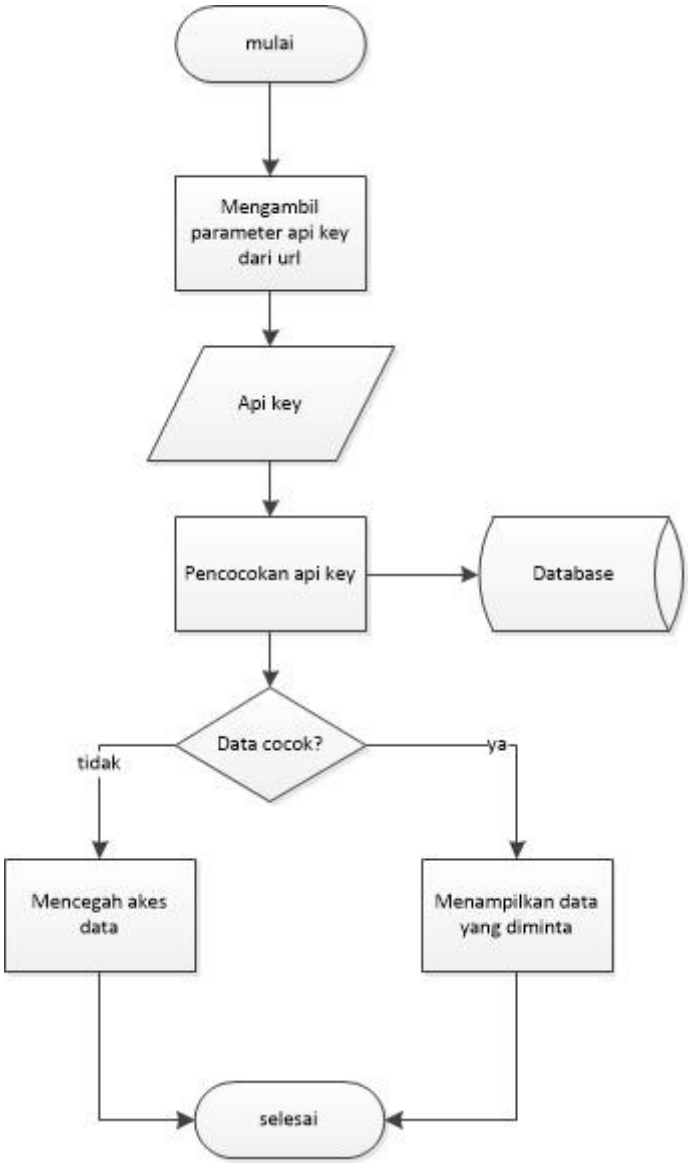
Perancangan method adalah sebuah tahap dari serangkaian tahap perancangan sistem yang ada. perancangan method menjelaskan alur dari sebuah method dalam sistem yang akan dibangun. Perancangan method dapat dilihat pada bahasan selanjutnya yang dapat dilihat pada Gambar III-59 sampai dengan Gambar III-62.

1. Login administrator



Gambar III-59 Perancangan Method Login admistrator

2. Validasi API Key



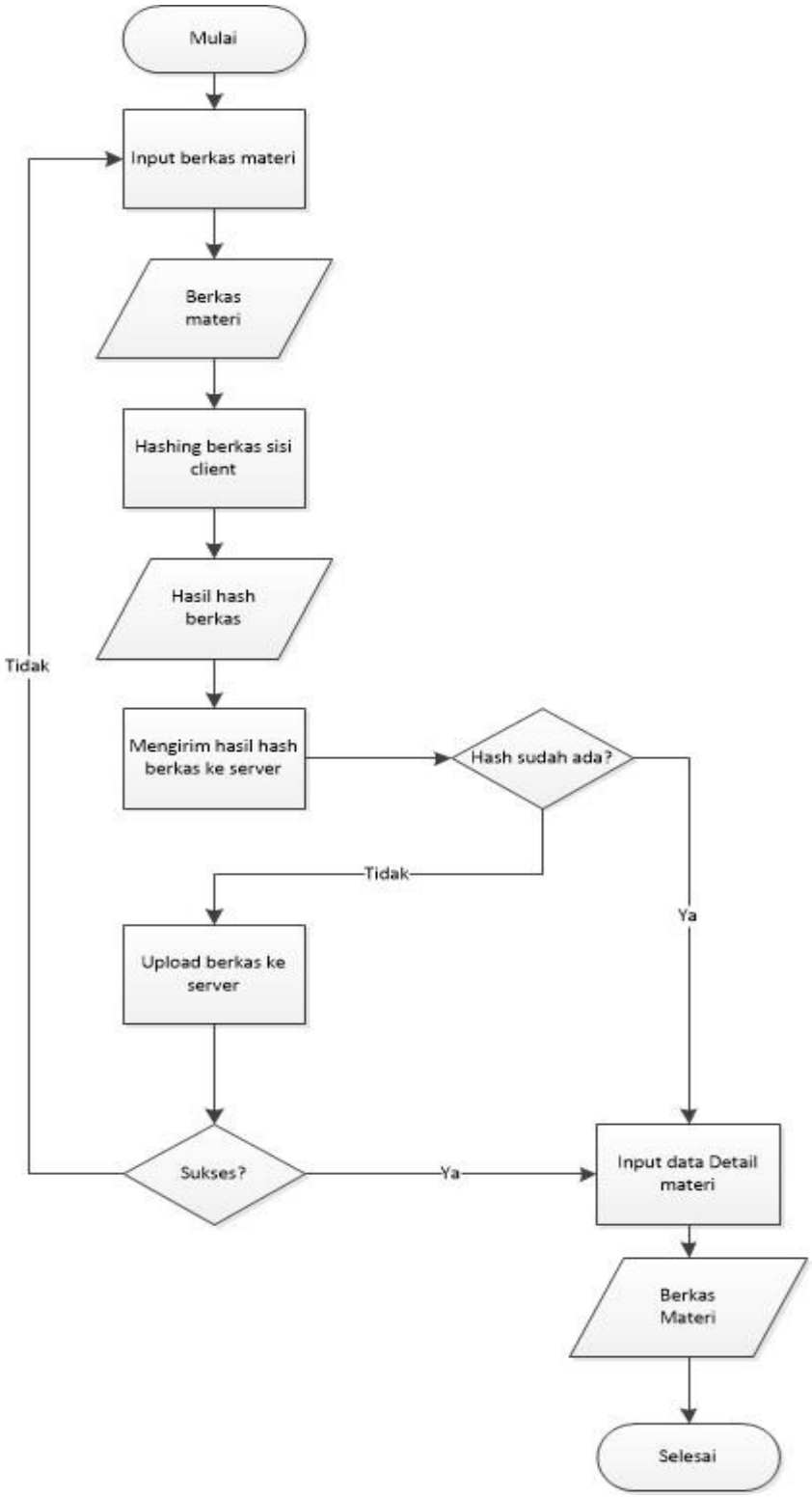
Gambar III-60 Perancangan Method Validasi API Key

3. Password hashing



Gambar III-61 Perancangan Method *Password Hashing*

4. Unggah berkas

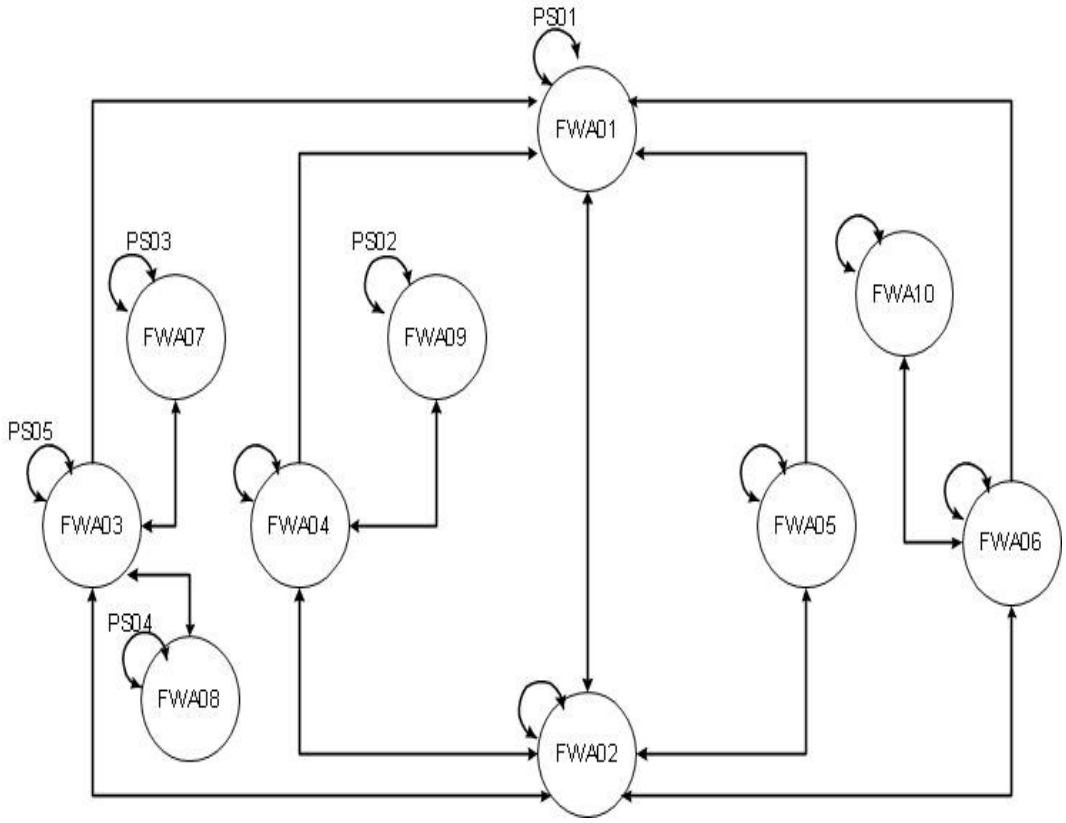


Gambar III-62 Perancangan Method Unggah Berkas

III.2.5 Jaringan Semantik

1. Jaringan semantik antarmuka administrator

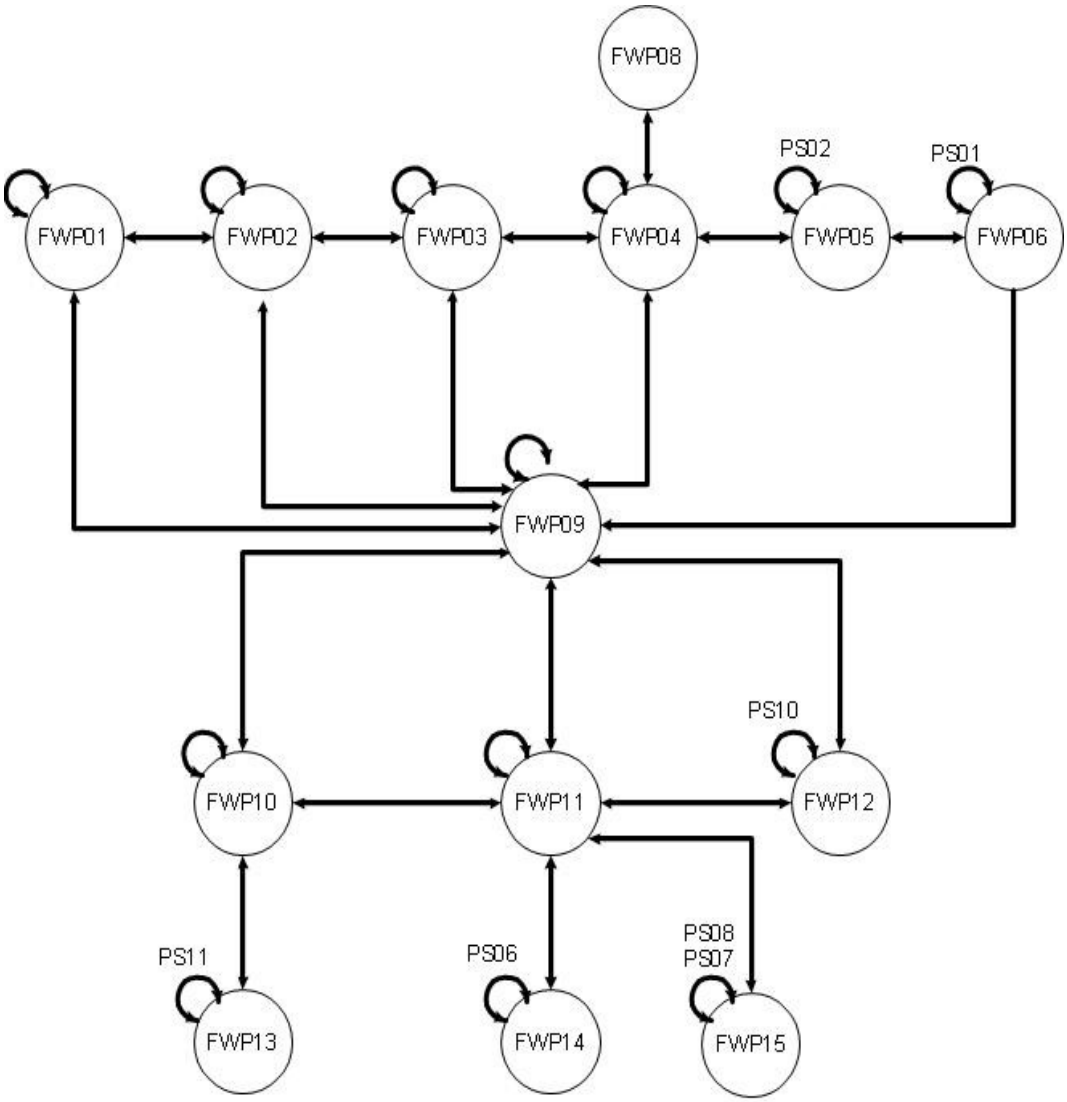
Berikut ini adalah jaringan semantik yang ada pada antarmuka administrator yang dapat dilihat pada Gambar III-63.



Gambar III-63 Jaringan Semantik Antarmuka Administrator

2. Jaringan semantik admin sekolah

Berikut ini adalah jaringan semantik yang ada antarmuka admin sekolah yang dapat dilihat pada Gambar III-64.



Gambar III-64 Jaringan Semantik Antarmuka Admin Sekolah