



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

METODOLOGI PENELITIAN

3.1 Metode Penelitian

Langkah-langkah yang dilakukan dalam penelitian untuk menyelesaikan skripsi ini adalah :

1. Studi Literatur

Studi Literatur dilakukan dengan mencari dan memahami bahan-bahan pustaka seperti jurnal, buku, dan bahan lainnya yang berkaitan dengan rumusan masalah dan metode STAD serta penerapannya dalam pembelajaran kooperatif untuk sistem yang dirancang.

2. Pengumpulan Data

Pengumpulan data dilakukan dengan mengumpulkan data-data dibutuhkan terkait dengan *cooperative learning*

3. Analisis Data

Analisis Data dilakukan dengan melakukan pencarian jurnal-jurnal, serta informasi tentang hal atau masalah yang terkait dengan *cooperative learning*.

4. Perancangan Program

Dari hasil analisis yang dilakukan selanjutnya dibuat rancangan untuk membangun aplikasi. Rancangan yang dibuat meliputi *interface*, proses dan alur program

5. Pembuatan Program

Pembuatan program dilakukan dengan mengaplikasikan rancangan yang telah dibuat sebelumnya dengan bahasa pemrograman dan *tools* yang dipilih.

6. Uji Coba dan Evaluasi

Setelah aplikasi selesai dibuat maka akan dilakukan uji coba untuk mengetahui kinerja sistem serta mengetahui kemungkinan terjadinya kesalahan.

7. Penulisan Skripsi

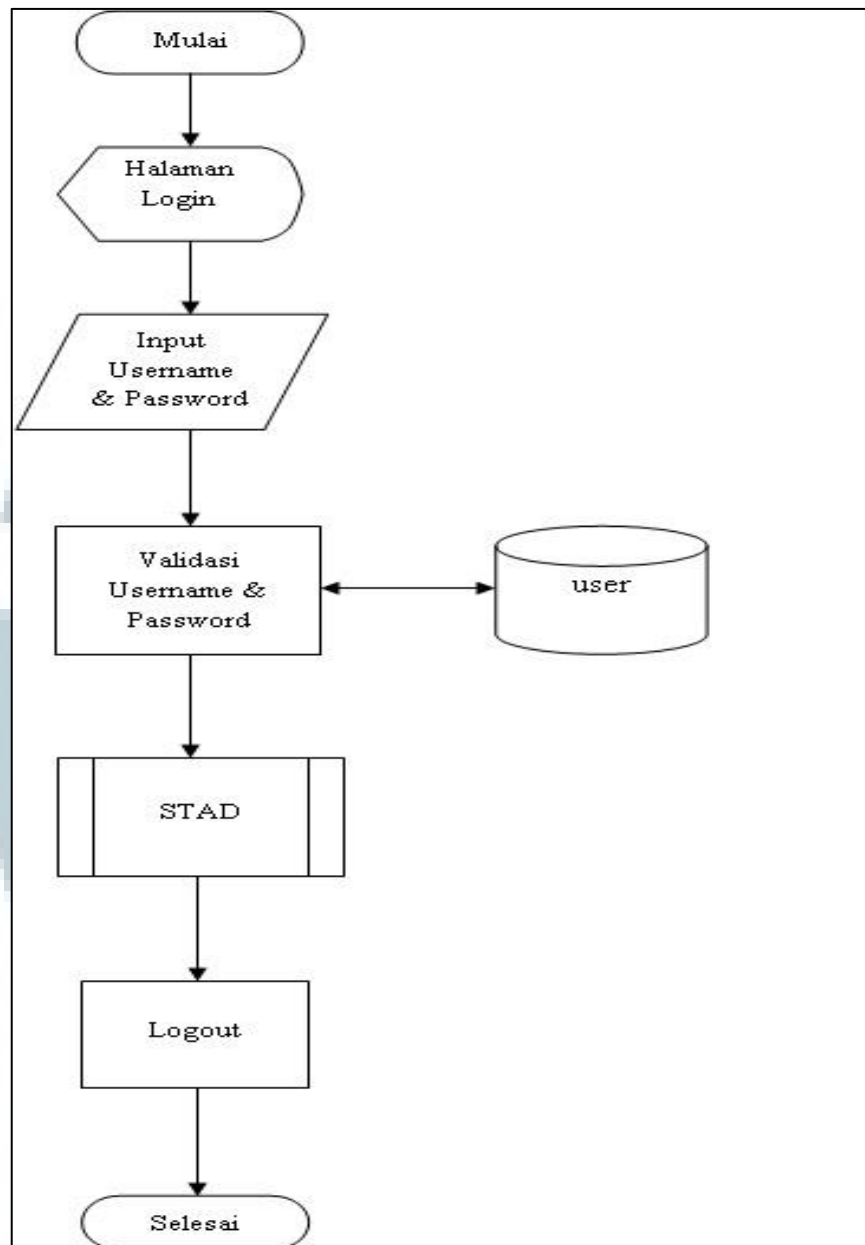
Setelah semua langkah tersebut selesai dilakukan, maka langkah selanjutnya adalah penyusunan laporan sebagai dokumentasi.

3.2 Perancangan Aplikasi

Perancangan aplikasi yang dibuat dalam sistem ini adalah *flowchart* diagram yang menggambarkan bagaimana alur proses jalannya aplikasi, *data flow* diagram yang menggambarkan alur data dari satu proses ke proses yang lain, dan *entity relationship* diagram yang menggambarkan hubungan antar tabel dalam aplikasi.

3.2.1 Flowchart

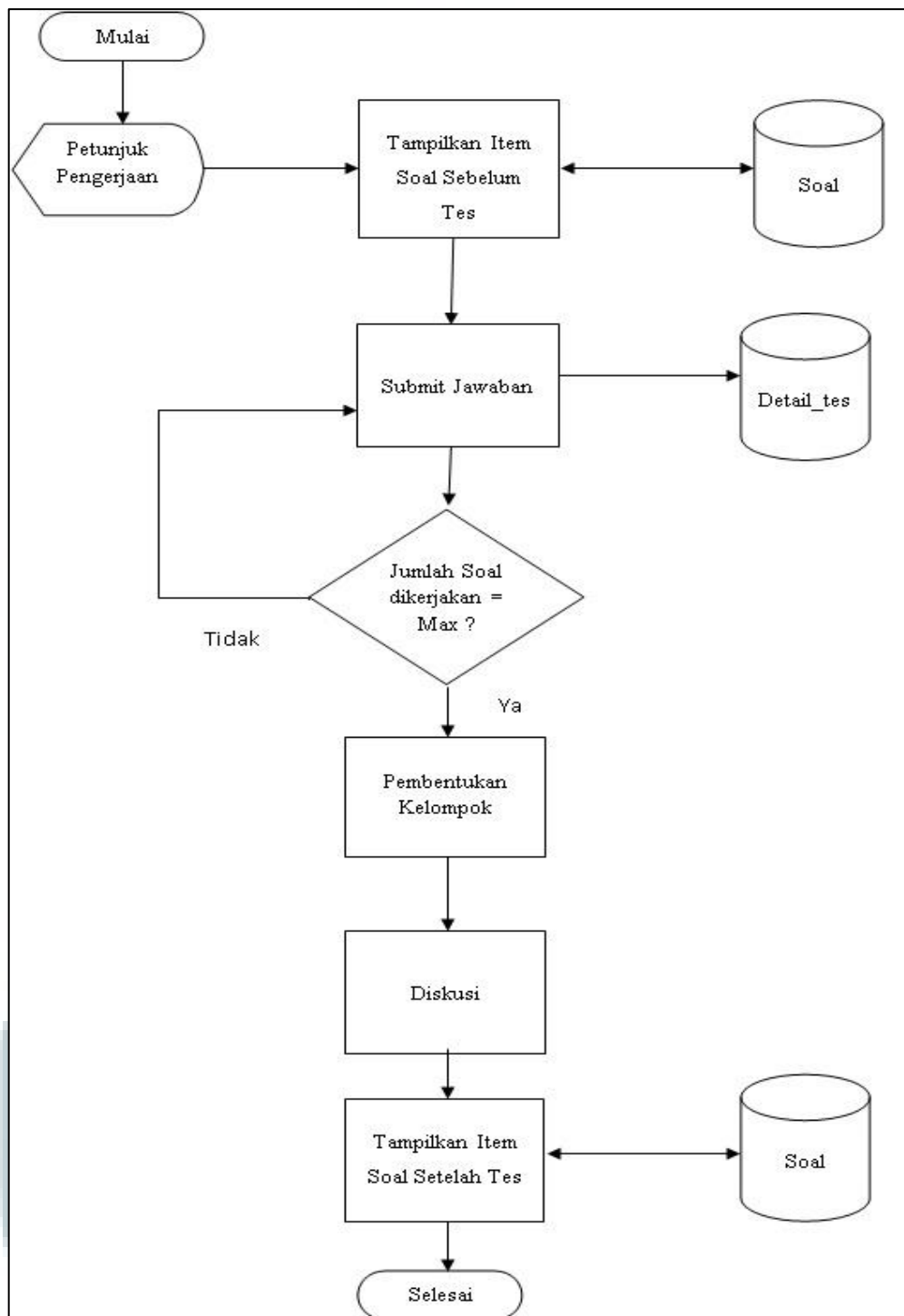
Aplikasi ini terdiri dari *front-end* dan *back-end*. Bagian *front-end* digunakan murid sedangkan bagian *back-end* digunakan *admin/guru* untuk melakukan *management content*. Berikut akan dibahas *flowchart* pada bagian *front-end*. Pertama-tama, murid harus melakukan *login* terlebih dahulu dengan meng-*input username* dan *password* yang dimilikinya. Sistem akan mengecek ke *database* apakah *username* dan *password* sesuai dengan yang ada di *database*. Setelah berhasil melakukan *login*, maka murid dapat melaksanakan proses pembelajaran kooperatif STAD.



Gambar 3.1 *Flowchart* aplikasi bagian *front-end*

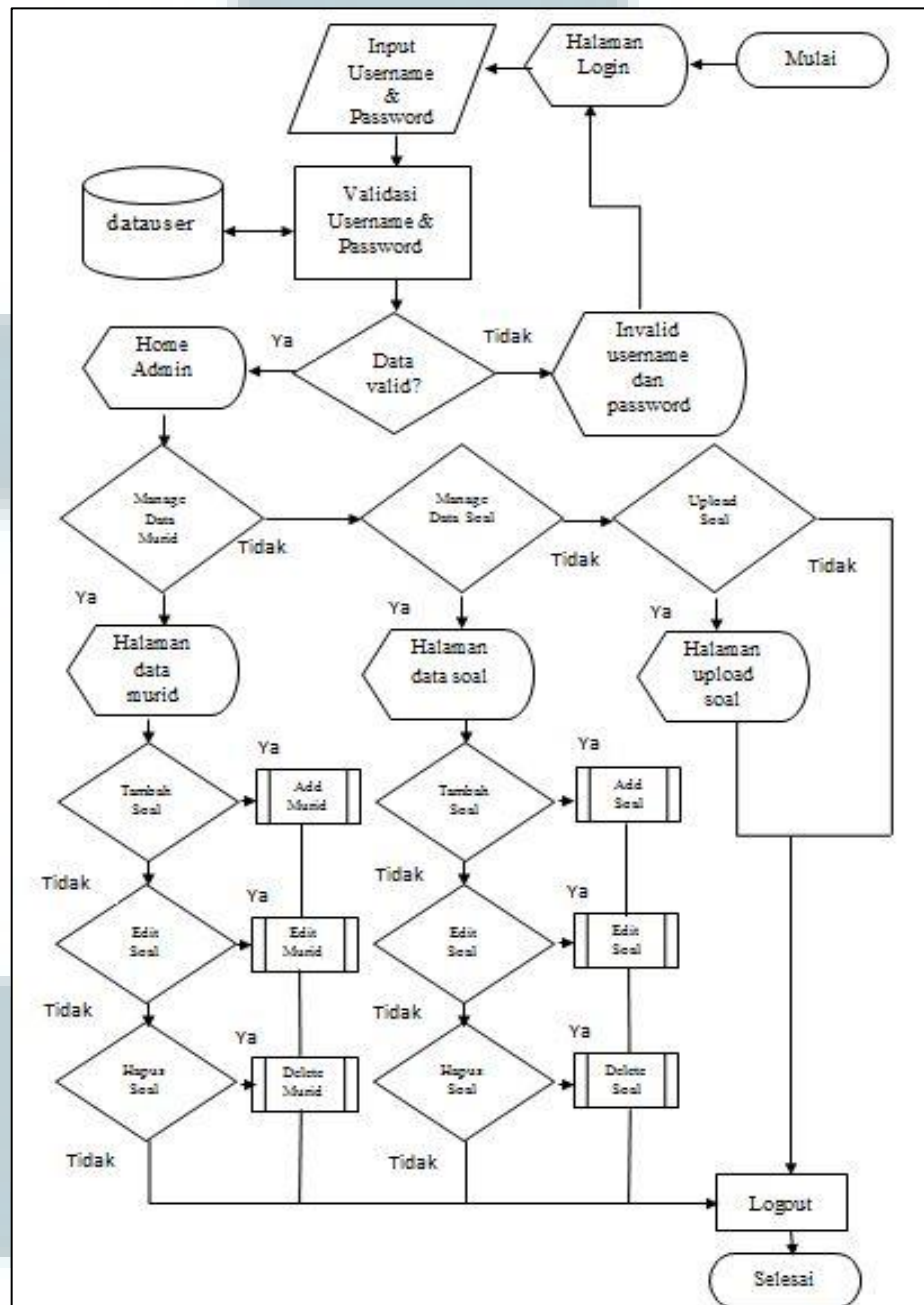
Setelah berhasil *login*, murid akan masuk ke halaman petunjuk pengerjaan. Kemudian murid akan mengerjakan tes soal sesi pertama. Sesi pertama bertujuan untuk mencari nilai acuan pembentukan kelompok. Setelah selesai mengerjakan, maka akan masuk ke proses pembentukan kelompok. Di dalam kelompok tersebut, murid-murid melakukan diskusi, baik dengan guru ataupun dengan teman-teman kelompoknya. Setelah selesai berdiskusi, maka murid mengerjakan

soal tes sesi kedua. Sesi kedua dijadikan sebagai perbandingan nilai dengan pengerjaan soal pada sesi pertama.



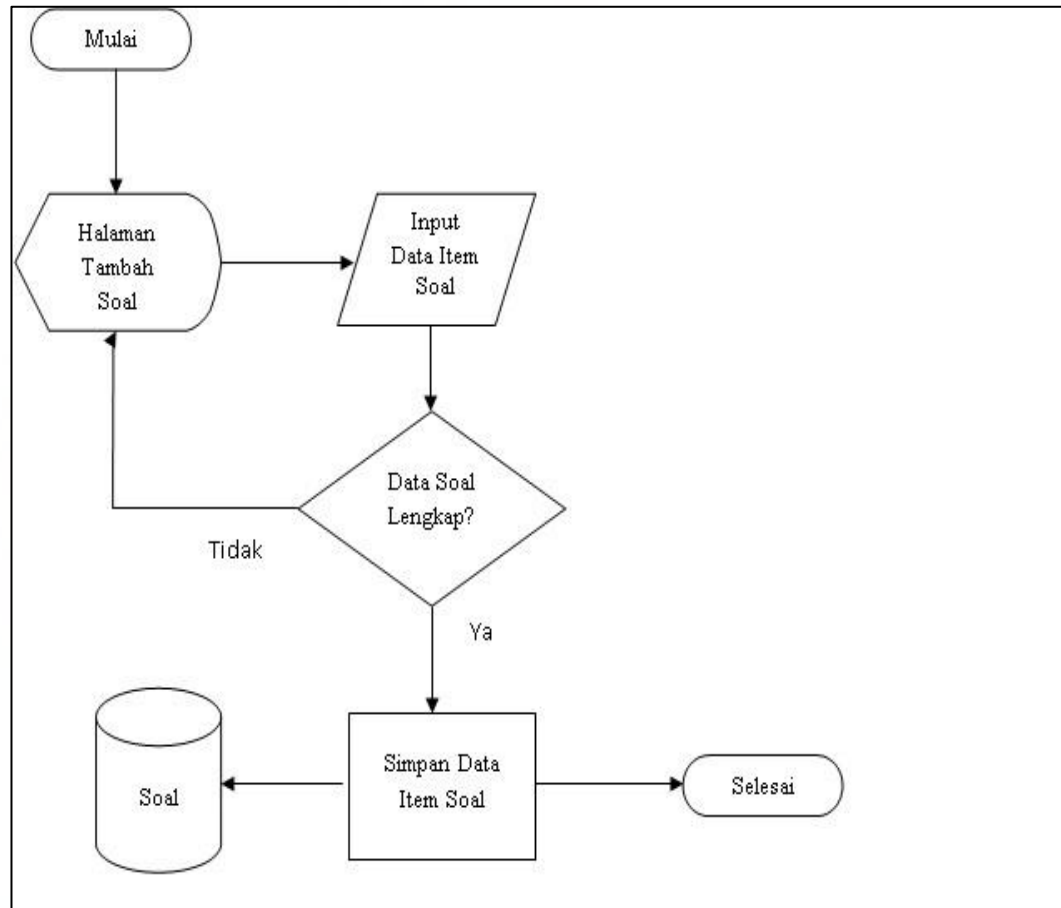
Gambar 3.2 *Flowchart* proses STAD

Untuk *flowchart* bagian *back-end*, maka *admin/guru* melakukan *login* terlebih dahulu. Setelah berhasil melakukan *login*, maka *admin/guru* dapat melakukan *insert/add*, *update*, dan *delete* untuk data soal dan murid serta dapat mengunggah materi pembelajaran.



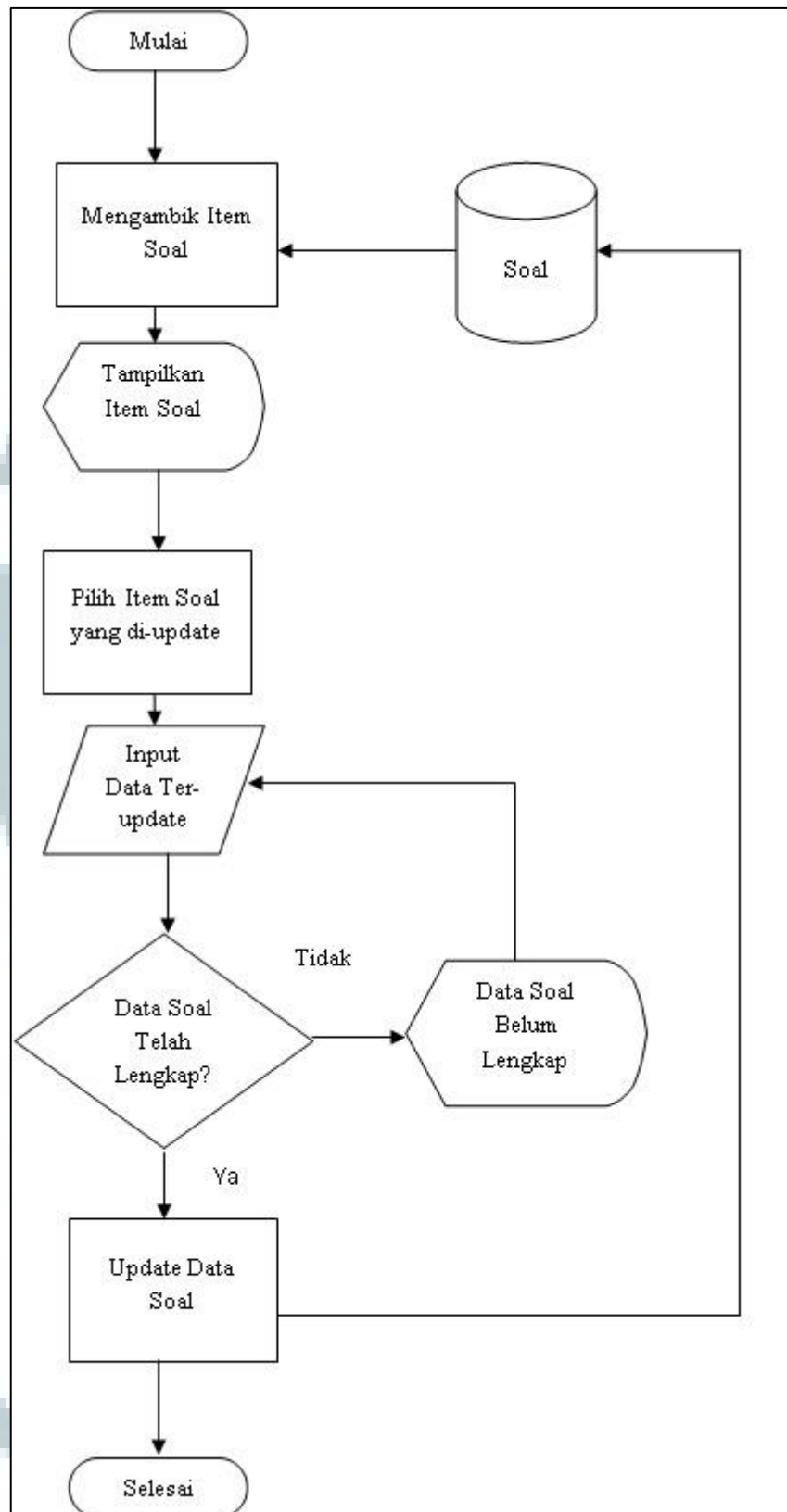
Gambar 3.3 *Flowchart* aplikasi bagian *back-end*

Pada halaman *add soal*, *admin/guru* harus melengkapi data yang berupa pertemuan, sesi, pertanyaan, pilihan jawaban, serta kunci jawaban untuk dimasukkan ke dalam *database*.



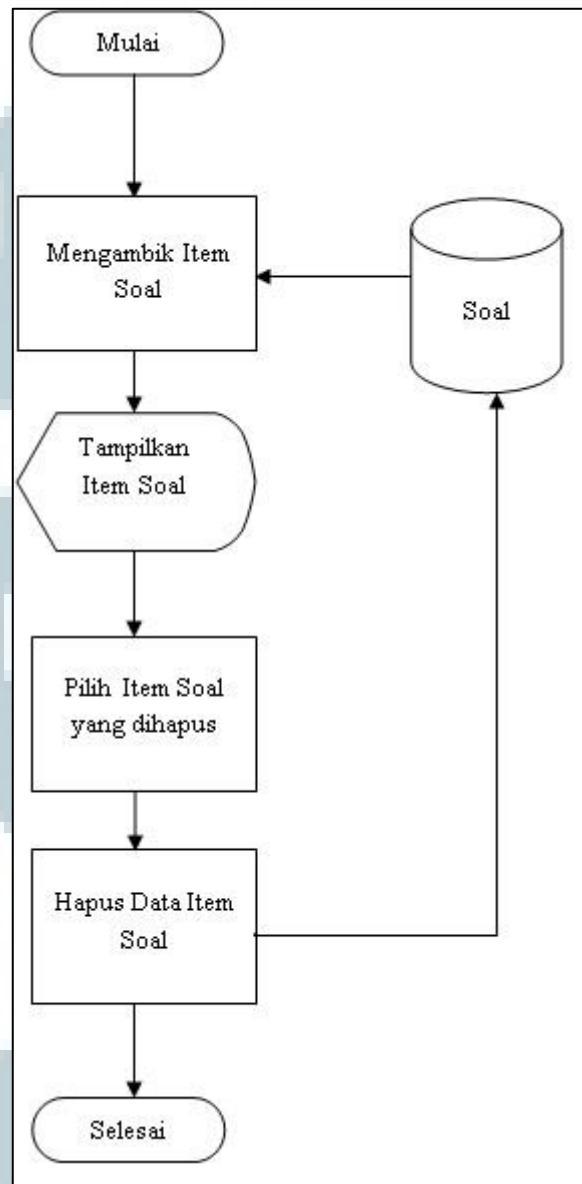
Gambar 3.4 *Flowchart add soal*

Jika *admin/guru* ingin melakukan perubahan data soal, maka *admin/guru* harus menentukan pertemuan dan sesi dimana letak soal itu berada. Setelah itu pilih soal yang akan diubah dan masukkan perubahan yang diinginkan.



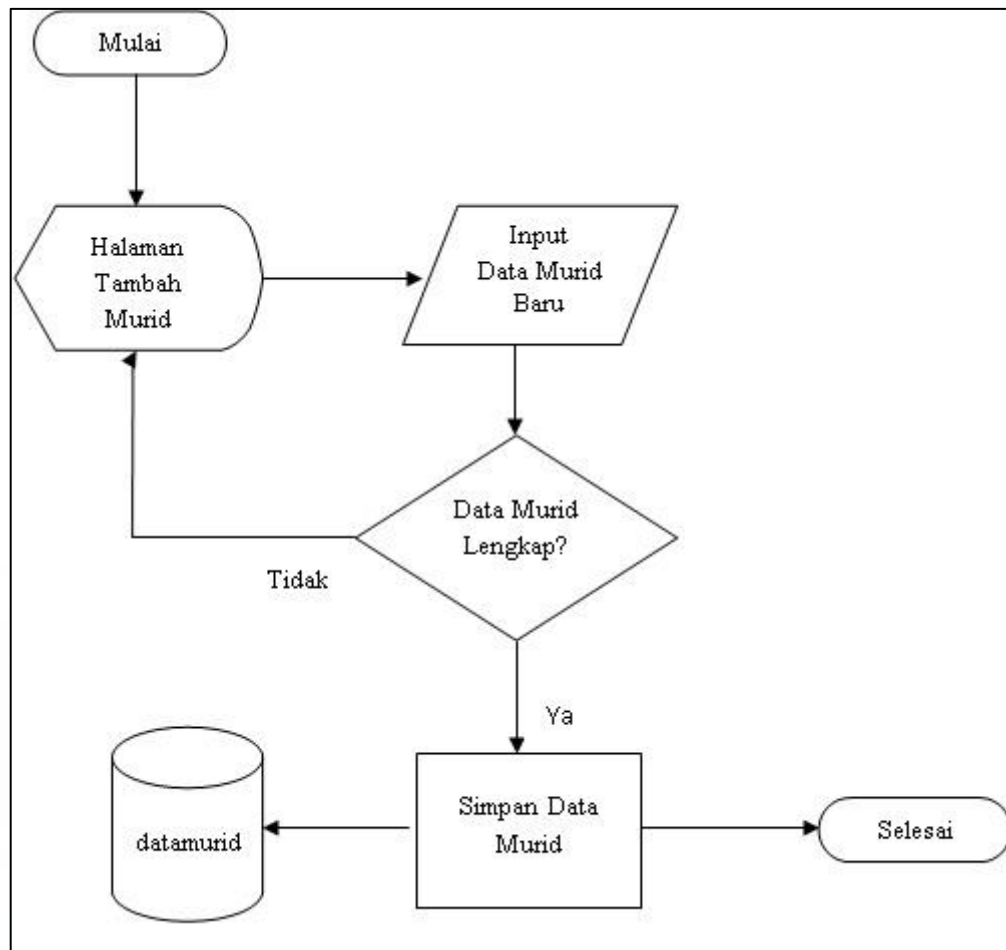
Gambar 3.5 Flowchart edit soal

Untuk melakukan *delete* soal, maka *admin/guru* dihadapkan pada tabel yang berisi data-data soal. Kemudian *admin/guru* memilih soal yang akan dihapus dari *database*.



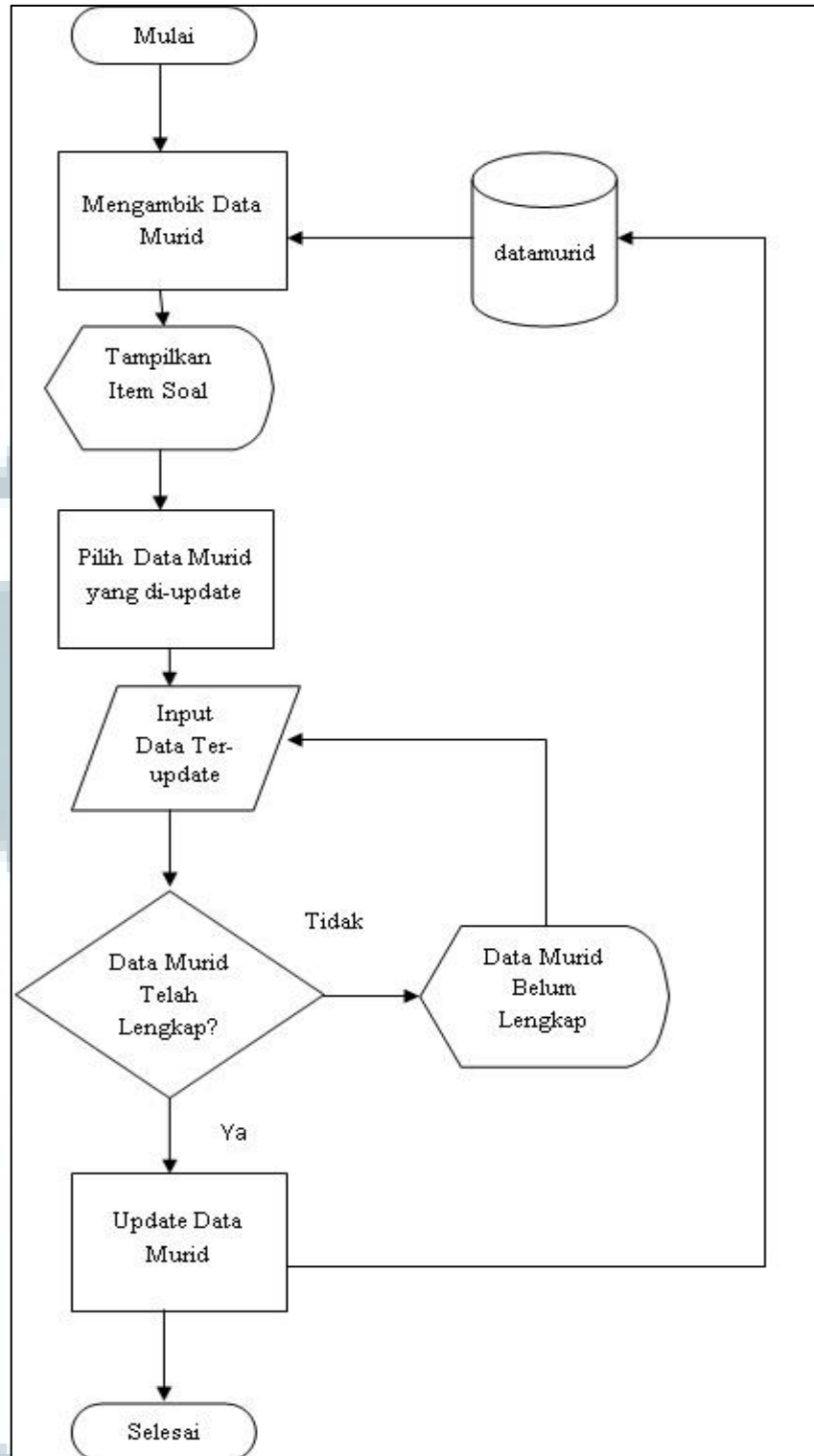
Gambar 3.6 *Flowchart delete soal*

Untuk *management content* murid, *admin* juga dapat melakukan *insert/add*, *edit*, dan *delete*. Untuk penambahan murid, maka *admin/guru* harus mengisi data-data lengkap murid sebelum dimasukkan ke dalam *database*.

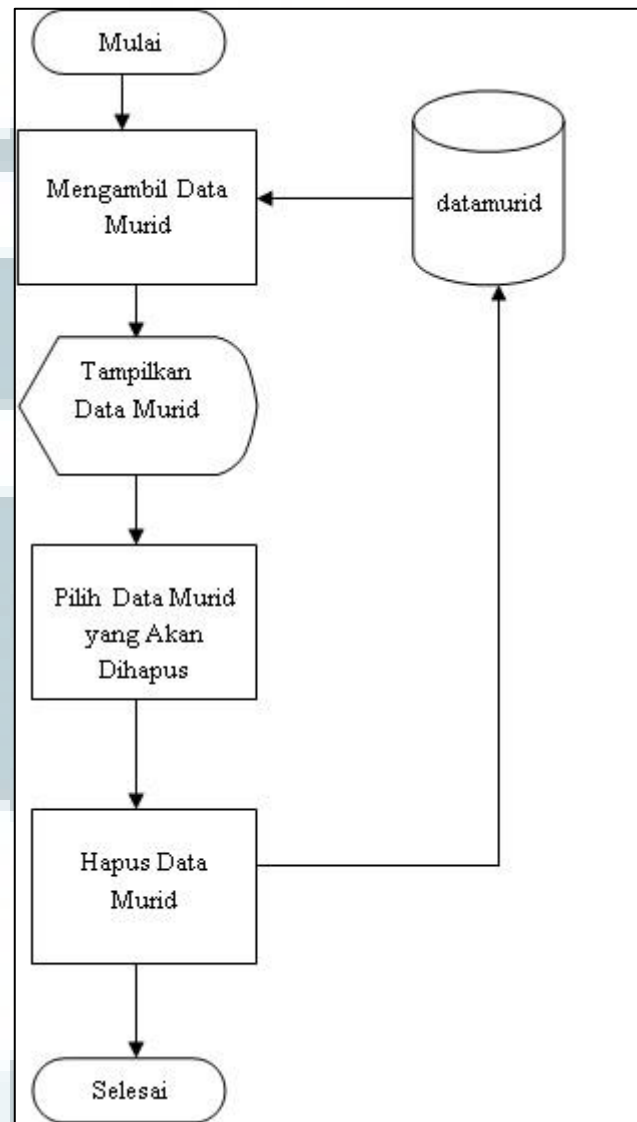


Gambar 3.7 *Flowchart add murid*

Jika *admin/guru* ingin melakukan perubahan pada data-data murid, maka *admin/guru* dapat melakukan perubahan dengan memilih murid yang akan diubah datanya. Kemudian *admin/guru* memasukkan data lengkap yang *ter-update* untuk disimpan ke dalam *database*.

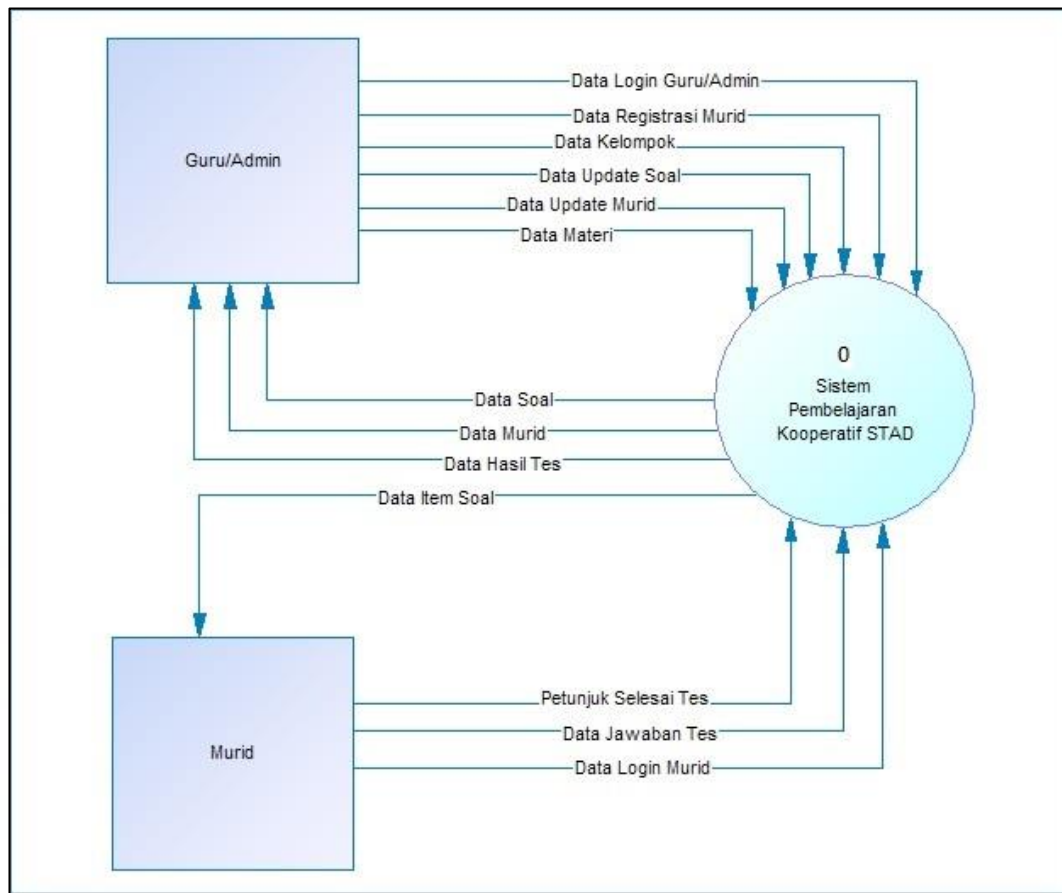
Gambar 3.8 *Flowchart edit murid*

Untuk menghapus data murid, maka *admin/guru* akan dihadapkan pada tabel yang berisi daftar murid. Kemudian *admin/guru* memilih murid yang akan dihapus datanya dari *database*.



Gambar 3.9 Flowchart delete murid

3.2.2 Data Flow Diagram



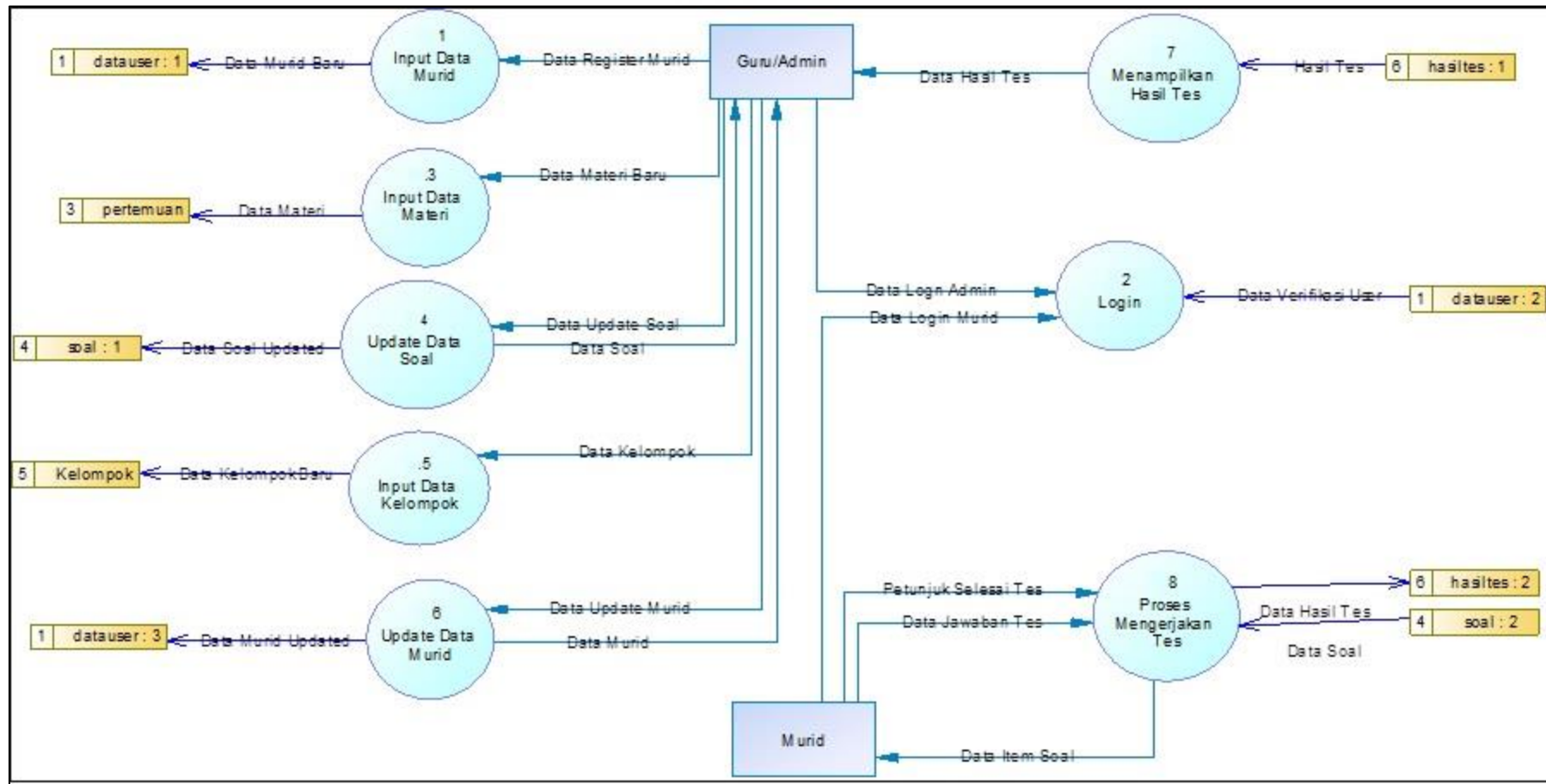
Gambar 3.10 *Context* diagram atau level 0

Dalam sistem yang dirancang, *context* diagram digambarkan seperti gambar 3.1 dimana terdapat sebuah proses utama STAD dan dua buah entitas *external* yang terlibat dalam sistem ini yaitu guru/admin dan murid.

Proses ini kemudian dipecah menjadi proses yang lebih rinci pada DFD level 1 seperti pada gambar 3.2 dimana terdapat 8 buah proses. Pertama-tama guru/admin dan murid melakukan *login* terlebih dahulu pada halaman login. *Username* dan *password* akan divalidasi apakah sama dengan data yang ada di *database*. Kemudian sistem akan mengarahkan ke masing-masing *page* sesuai dengan *privileges* masing-masing. Apabila guru/admin, maka akan diarahkan ke halaman admin dimana guru/admin dapat melakukan proses *input* data murid,

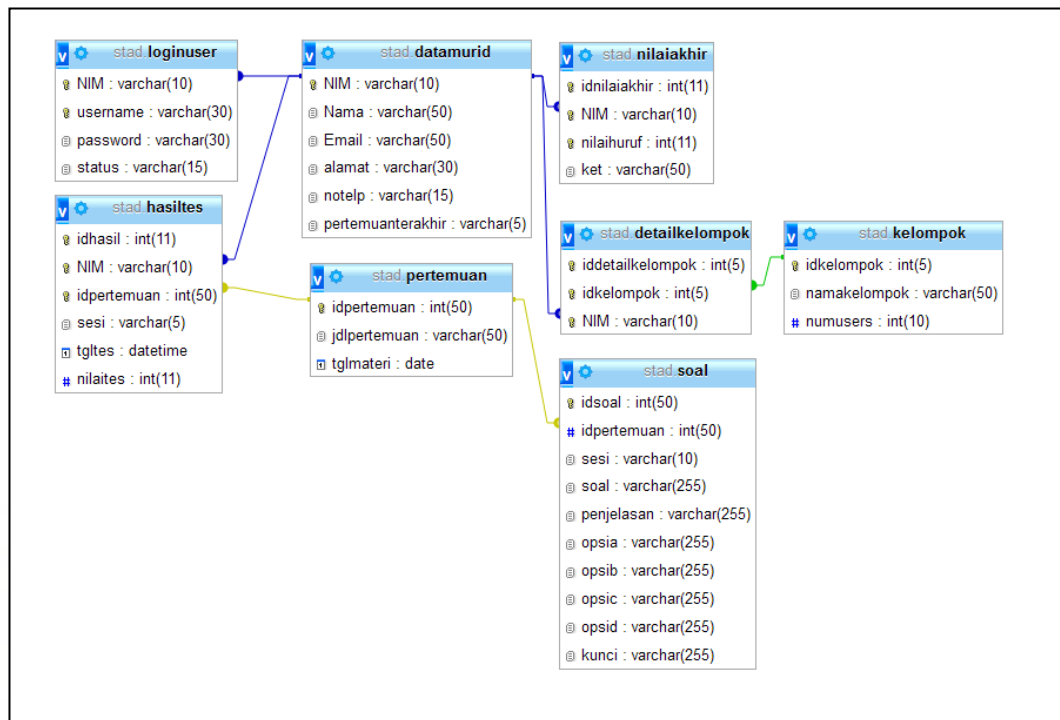
input data materi, *update* data soal, *input* data kelompok, *update* data murid, serta melihat hasil tes yang dikerjakan murid. Apabila murid, maka akan diarahkan ke halaman pengerjaan soal dan murid akan mengerjakan soal sesuai dengan waktu yang disediakan.





Gambar 3.11 DFD Level 1

3.2.3 Entity Relationship Diagram



Gambar 3.12 Entity Relationship Diagram

Dari gambar 3.3, dapat dilihat bahwa terdapat beberapa tabel. Hubungan antara tabel data murid dan tabel loginuser adalah *one to one* karena seorang murid hanya memiliki satu data *login*. Sedangkan antara tabel datamurid dan hasil tes adalah *one to one* karena seorang murid bisa memiliki satu *record* hasil tes. Begitu juga hubungan antara tabel nilaiakhir dan tabel detailkelompok. Sedangkan antara tabel pertemuan dan soal memiliki hubungan *one to many* karena dalam satu pertemuan bisa memiliki beberapa soal. Terdapat pula tabel datauser yang tidak memiliki hubungan sama sekali dengan tabel yang lainnya.

3.2.4 Struktur Tabel

Dalam membangun sistem *cooperative learning* ini, terdapat 11 tabel. Tabel-tabel tersebut adalah sebagai berikut.

Nama tabel : loginuser

Fungsi : digunakan untuk menyimpan data-data login murid dan admin

Primary key : NIM

Foreign key : -

Tabel 3.1 Struktur tabel *login user*

Nama Field	Tipe Data	Keterangan
NIM	varchar(10)	Nomor induk dari murid dan admin
Username	varchar(30)	Username untuk login
Password	varchar(30)	Password untuk login
Status	varchar(15)	Status dari user yang login

Nama tabel : datamurid

Fungsi : digunakan untuk menyimpan data-data pribadi murid

Primary key : -

Foreign key : NIM

Tabel 3.2 Struktur tabel murid

Nama Field	Tipe Data	Keterangan
NIM	varchar(10)	Nomor induk dari murid dan admin
Nama	varchar(50)	Nama murid
Email	varchar(50)	Email murid
Alamat	varchar(50)	Alamat murid
Notelp	varchar(15)	Nomor telepon murid
Pertemuanterakhir	varchar(5)	<i>Record</i> pertemuan terakhir

Nama tabel : pertemuan

Fungsi : digunakan untuk menyimpan data pertemuan murid

Primary key : idpertemuan

Foreign key : -

Tabel 3.3 Struktur tabel pertemuan

Nama Field	Tipe Data	Keterangan
Idpertemuan	int(10)	Nomor id pertemuan
Jdlpertemuan	varchar(50)	Judul dari pertemuan
Tglmateri	Date	Tanggal materi pertemuan

Nama tabel : soal

Fungsi : digunakan untuk menyimpan data-data soal

Primary key : idsoal

Foreign key : idpertemuan

Tabel 3.4 Struktur tabel soal

Nama Field	Tipe Data	Keterangan
Idsoal	int(50)	Nomor id soal
Idpertemuan	int(50)	Untuk menentukan jenis soal berdasarkan pertemuan

Tabel 3.4 Struktur tabel soal (lanjutan)

Nama Field	Tipe Data	Keterangan
Sesi	varchar(10)	Sesi tes
Soal	varchar(255)	Nomor soal
Penjelasan	varchar(255)	Soal
Opsia	varchar(255)	Opsi jawaban A
Opsib	varchar(255)	Opsi jawaban B
Opsic	varchar(255)	Opsi jawaban C
Opsid	varchar(255)	Opsi jawaban D
Kunci	varchar(2)	Kunci jawaban

Nama tabel : idhasil

Fungsi : digunakan untuk menyimpan data hasil tes

Primary key : idhasil

Foreign key : NIM, idpertemuan

Tabel 3.5 Struktur tabel hasil

Nama Field	Tipe Data	Keterangan
Idhasil	int(11)	Nomor id hasil tes
NIM	varchar(10)	Nomor induk dari murid
Idpertemuan	int(50)	Hasil tes berdasarkan pertemuan
Sesi	varchar(5)	Sesi tes
Tglugas	Date	Tanggal tes
Nilaites	int(11)	Nilai res

Nama tabel : kelompok

Fungsi : digunakan untuk menyimpan data-data kelompok

Primary key : idkelompok

Foreign key : -

Tabel 3.6 Struktur tabel kelompok

Nama Field	Tipe Data	Keterangan
Idkelompok	int(5)	Nomor id kelompok
Namakelompok	varchar(50)	Nama kelompok
Numusers	int(10)	Jumlah anggota kelompok

Nama tabel : detailkelompok

Fungsi : digunakan untuk menyimpan data detail per kelompok

Primary key : iddetailkelompok

Foreign key : idkelompok, NIM

Tabel 3.7 Struktur tabel detail kelompok

Nama Field	Tipe Data	Keterangan
Iddetailkelompok	int(5)	Nomor id detail kelompok
Idkelompok	Int(5)	Nomor id kelompok
NIM	varchar(10)	Nomor induk murid

Nama tabel : nilaiakhir

Fungsi : digunakan untuk menyimpan data nilai akhir dari murid

Primary key : idnilaiakhir

Foreign key : NIM

Tabel 3.8 Struktur tabel nilai akhir

Nama Field	Tipe Data	Keterangan
Idnilaiakhir	int(11)	Nomor id nilai akhir
NIM	varchar(10)	Nomor induk murid
Nilaihuruf	int(11)	Nilai huruf dalam huruf
Ket	varchar(50)	Keterangan dari nilai huruf

Nama tabel : chat_rooms

Fungsi : digunakan untuk menyimpan data *chat* kelompok

Primary key : id

Foreign key : -

Tabel 3.9 Struktur tabel *chat room*

Nama Field	Tipe Data	Keterangan
Id	tinyint(4)	Nomor id <i>chat room</i> kelompok
Name	varchar(20)	Nama <i>chat room</i>
Numofuser	int(10)	Jumlah <i>user</i> dalam <i>room</i>
File	varchar(30)	<i>File</i> untuk menyimpan <i>chat-records</i>

Nama tabel : chat_users

Fungsi : digunakan untuk menyimpan data *user* saat di *room*

Primary key : id

Foreign key : -

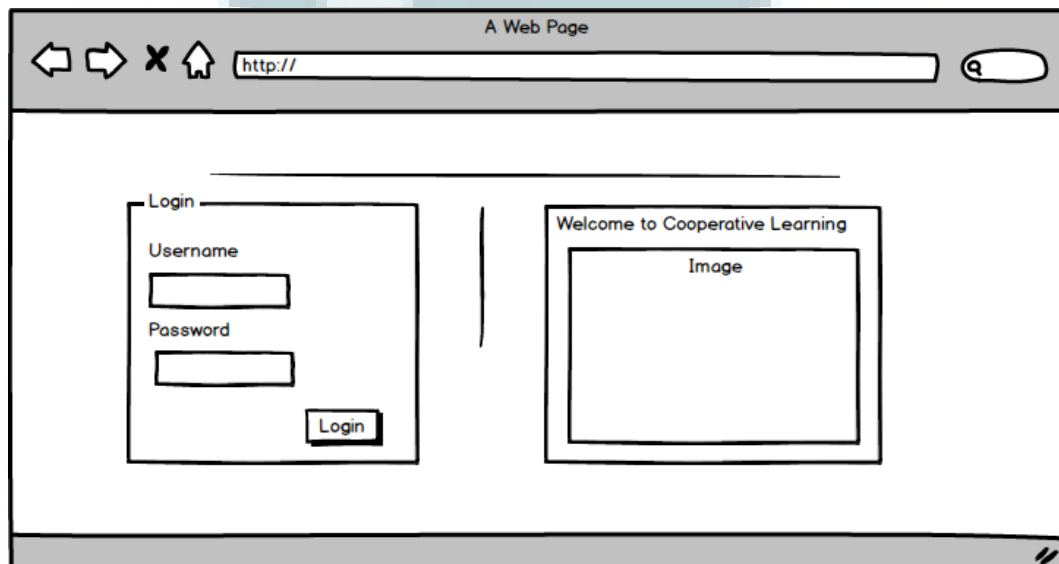
Tabel 3.10 Struktur tabel *chat user*

Nama Field	Tipe Data	Keterangan
Id	tinyint(10)	Nomor id status
Username	varchar(100)	Username dari <i>user</i>
Status	tinyint(1)	Status aktif dari <i>user</i>
Time_mod	int(100)	Waktu <i>user</i> saat berada di <i>room</i>

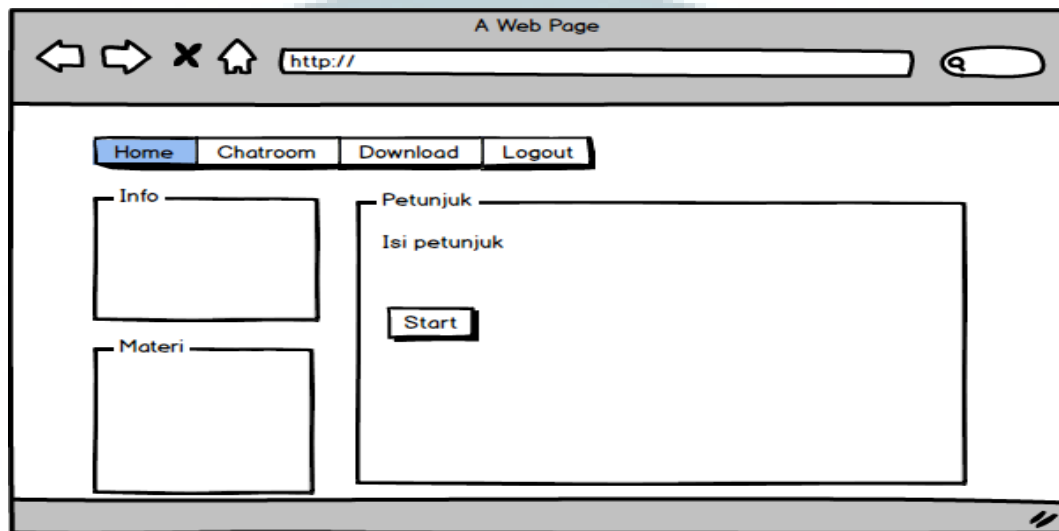
3.2.5 Rancangan Antarmuka

Sistem atau aplikasi ini memiliki beberapa rancangan antarmuka yang terdiri dari rancangan antarmuka untuk murid (*front-end*) dan rancangan antarmuka untuk *admin* atau guru (*back-end*).

Halaman *login* untuk *front-end* terdiri dari sebuah *box* untuk mengisi *username* dan *password* yang dibutuhkan untuk masuk ke dalam sistem seperti pada gambar 3.13.

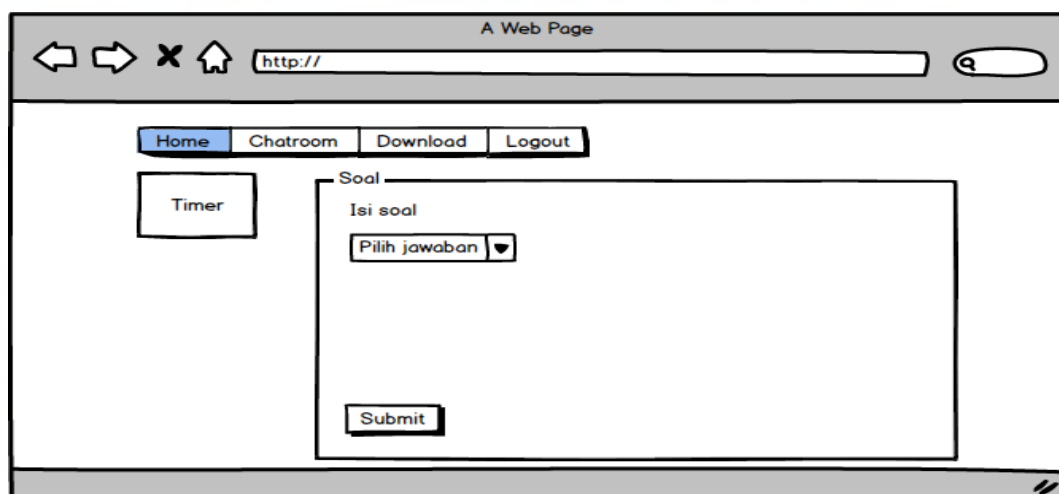
Gambar 3.13 Rancangan Antarmuka *login front-end*

Setelah murid berhasil masuk, murid akan melihat petunjuk pengerjaan soal dan sebuah *button* 'Start' untuk mulai mengerjakan soal. Pada halaman ini juga terdapat *box* yang berisi informasi murid dan materi yang akan dipelajari pada saat itu.



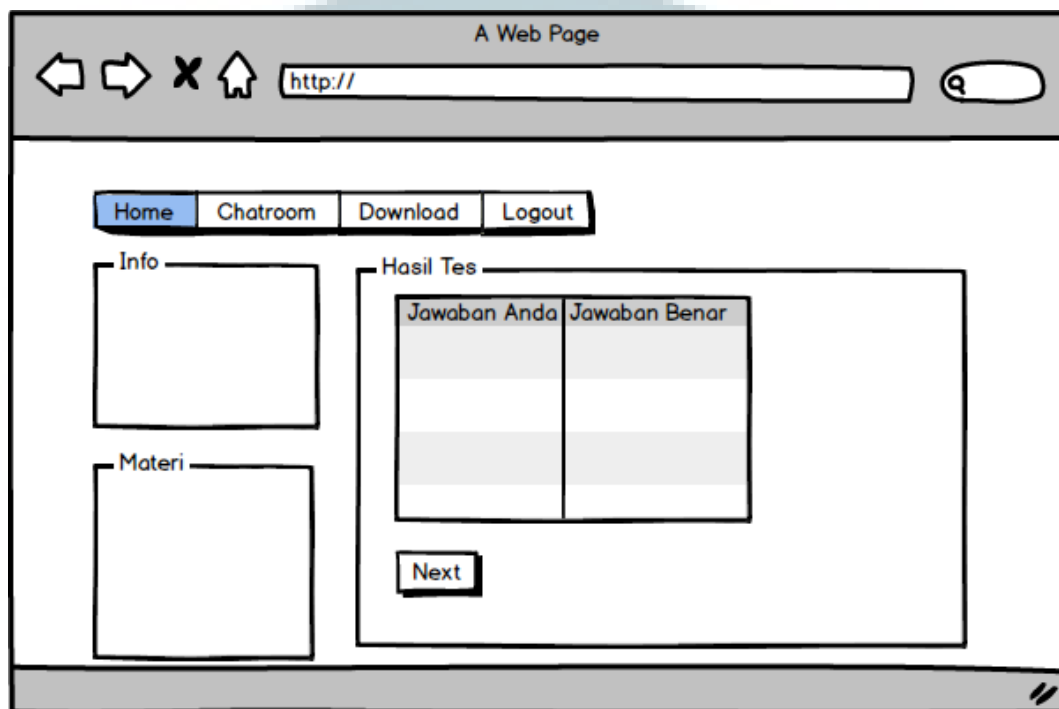
Gambar 3.14 Rancangan antarmuka halaman *index*

Pada halaman tes soal, terdapat sebuah *timer* untuk menentukan batas waktu tes dan soal-soal yang harus dikerjakan oleh murid. Untuk memilih jawaban, murid dapat memilih lewat *combo box* yang telah disediakan.



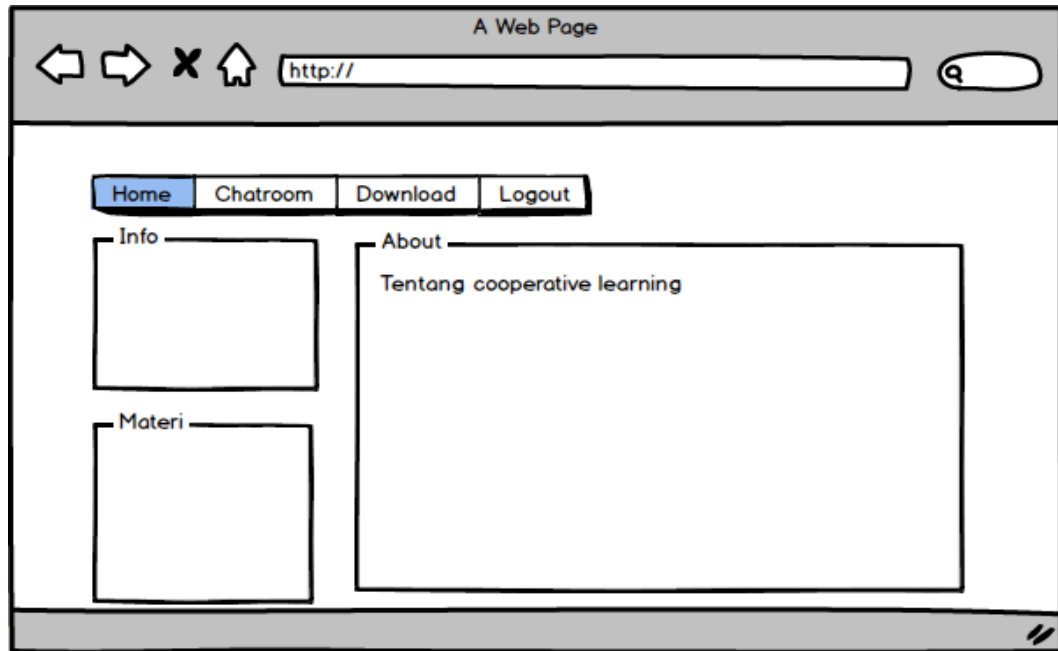
Gambar 3.15 Rancangan antarmuka halaman tes

Setelah selesai mengerjakan soal, maka murid dapat melihat hasil dari tes tersebut. Murid dapat membandingkan jawabannya dengan jawaban benar dari soal-soal yang telah dikerjakannya dalam sebuah tabel. Terdapat sebuah *button* 'Next' untuk lanjut ke halaman berikutnya.



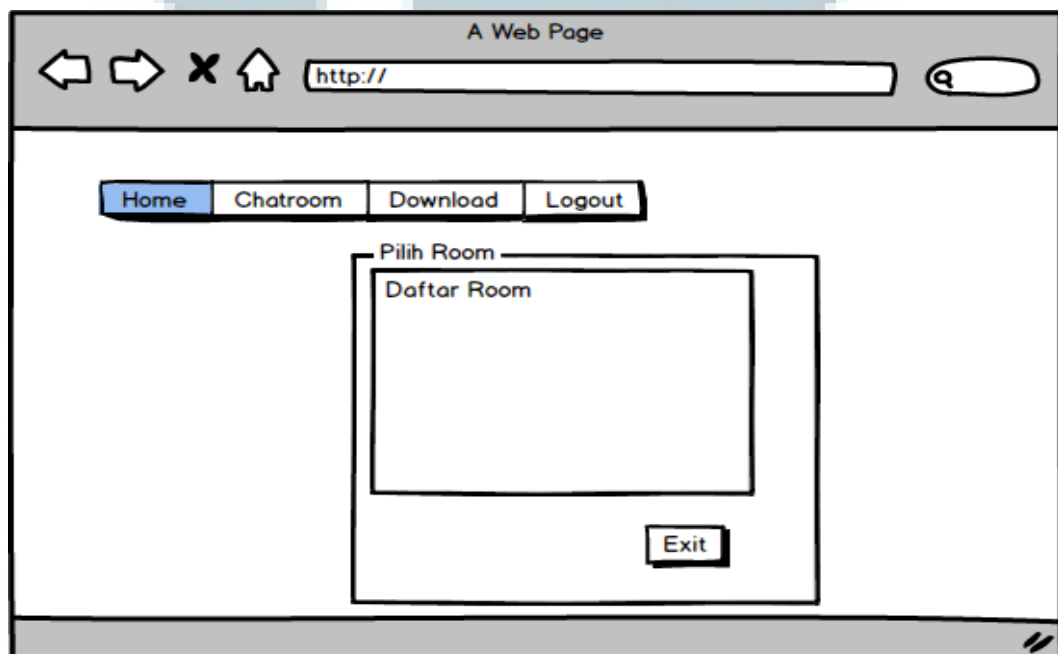
Gambar 3.16 Rancangan antarmuka halaman cek hasil

Setelah murid melihat hasil tes, mereka masuk ke halaman utama seperti gambar 3.17. Di halaman utama ini, murid dapat masuk ke *chat room* ataupun *download* materi yang telah di-*upload*.



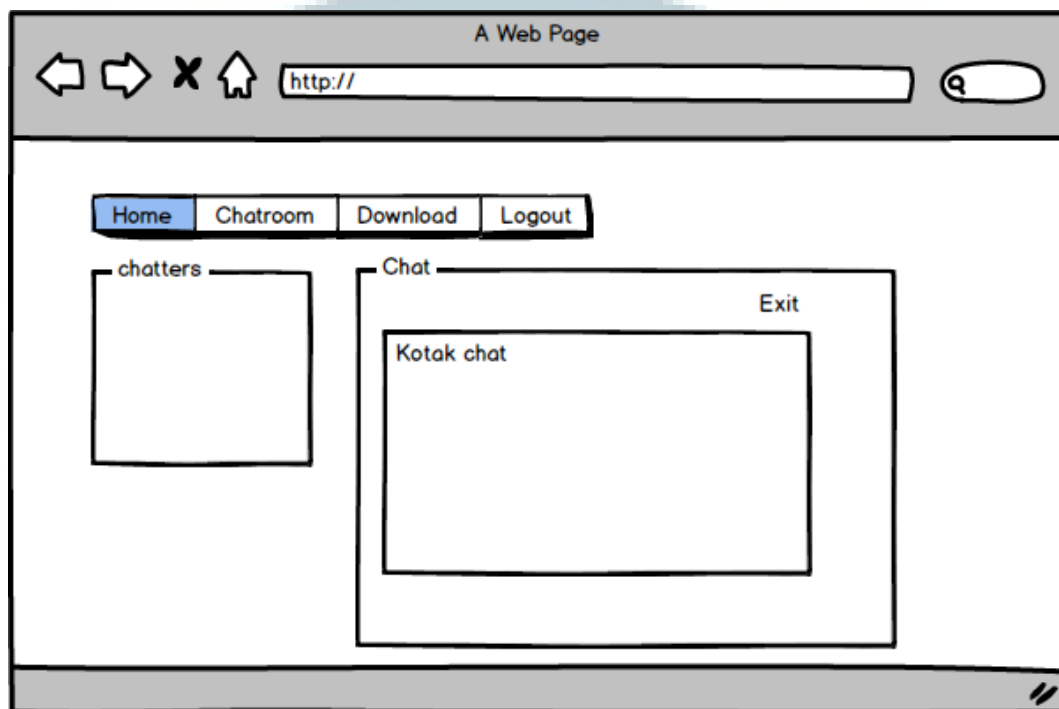
Gambar 3.17 Rancangan antarmuka halaman utama

Jika murid masuk ke dalam chat *room*, maka murid dapat memilih *room* yang akan dipilih sebagai tempat diskusi. Murid dapat masuk ke dalam *room* sesuai dengan kelompoknya.



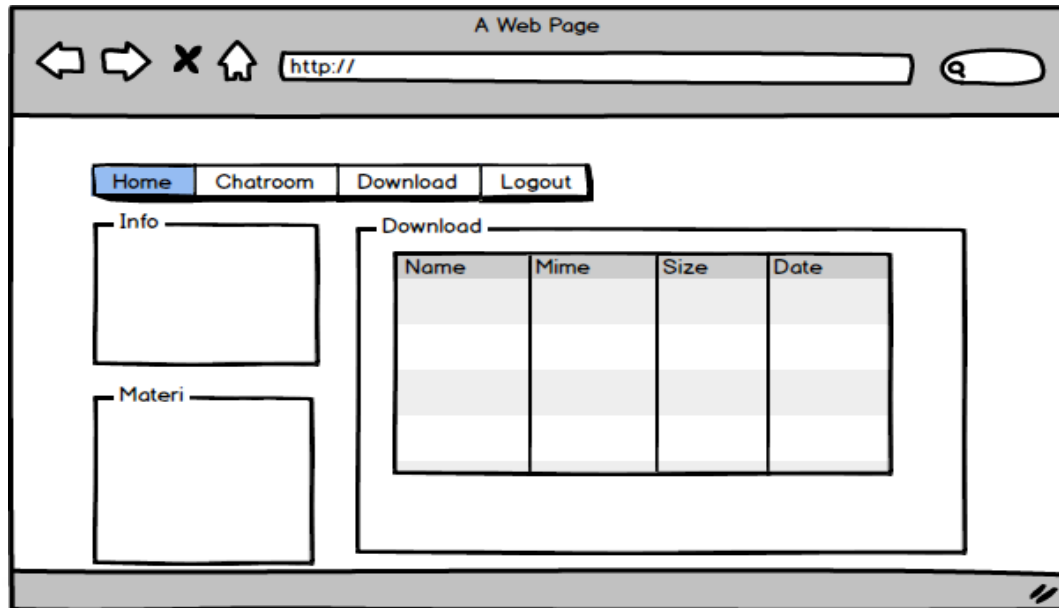
Gambar 3.18 Rancangan antarmuka halaman *chat room*

Setelah memilih untuk *join room*, maka murid akan masuk ke halaman *chatting*. Di halaman ini murid dapat berinteraksi dengan guru ataupun dengan teman-teman kelompoknya. Di sebelah kiri terdapat orang-orang yang masih aktif di dalam *room chatting*.



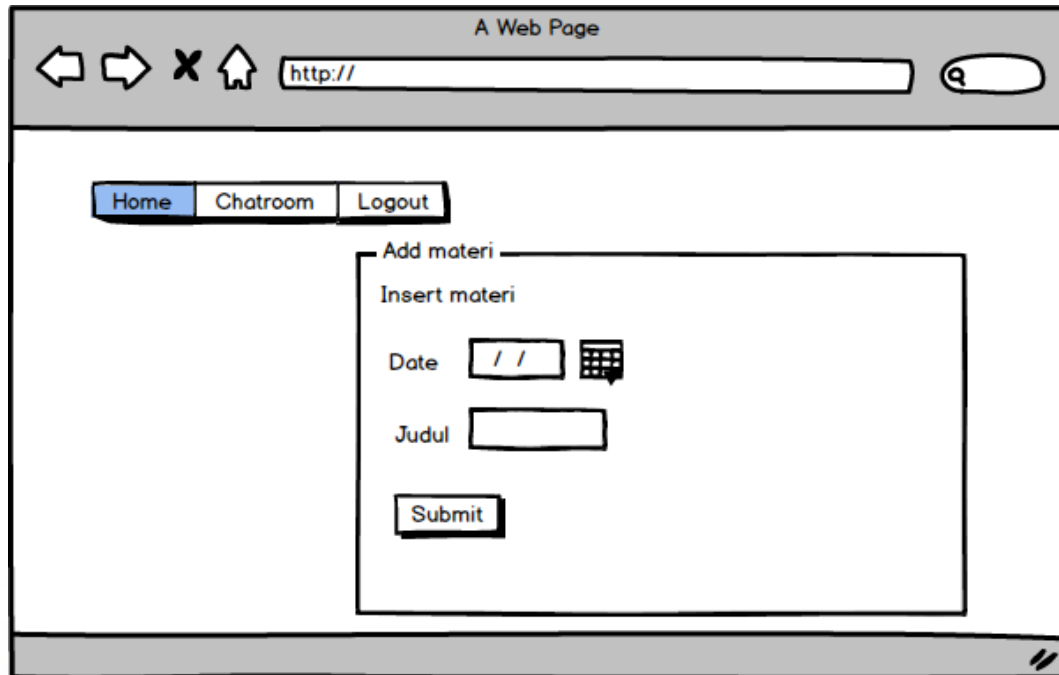
Gambar 3.19 Rancangan antarmuka halaman *room chatting*

Pada halaman *download*, murid dapat meng-*unduh* materi-materi yang telah di-*upload* oleh *admin*. Daftar *file* yang dapat di-*unduh* terdapat dalam sebuah tabel. Halaman *download* tampak seperti pada gambar 3.20.



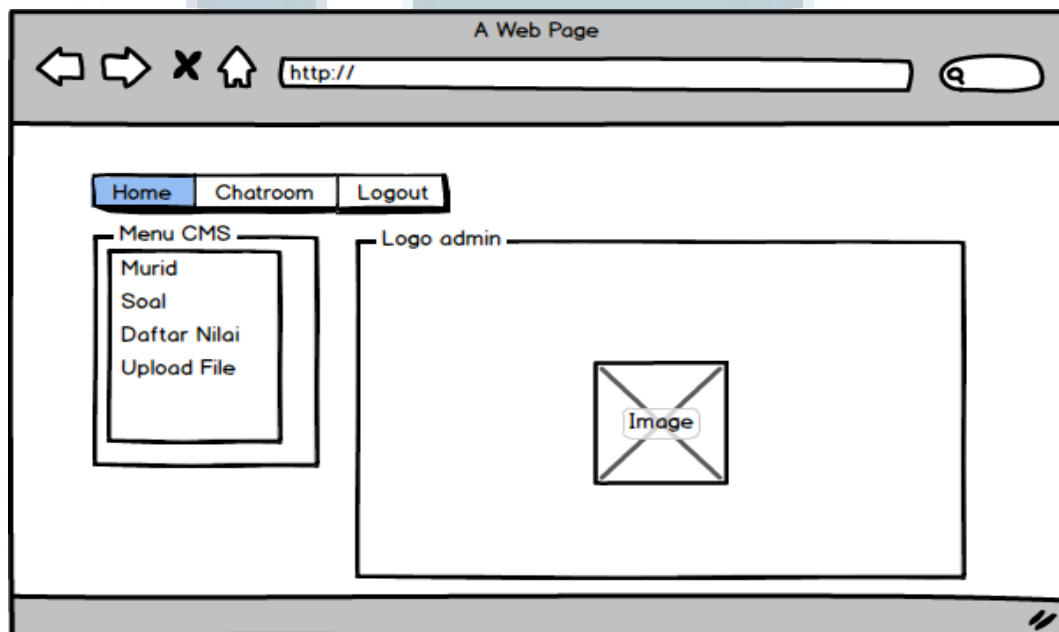
Gambar 3.20 Rancangan antarmuka halaman *download*

Halaman *back-end* dirancang agar *admin/guru* dapat melakukan *management content* untuk data-data di dalam sistem. *Admin/guru* melakukan login terlebih dahulu untuk dapat mengakses halaman *admin*. *Login* dilakukan di halaman *login* seperti pada gambar 3.13. Setelah berhasil melakukan *login*, maka *admin* akan tiba di halaman *insert lesson* yang tampak seperti pada gambar 3.21. Di halaman ini *admin/guru* harus menentukan judul pertemuan yang akan dibahas. Terdapat sebuah *date time picker* dan *input text* untuk menentukan tanggal dan judul pertemuan. Setelah itu ada *button* 'Submit' untuk pindah ke halaman selanjutnya.



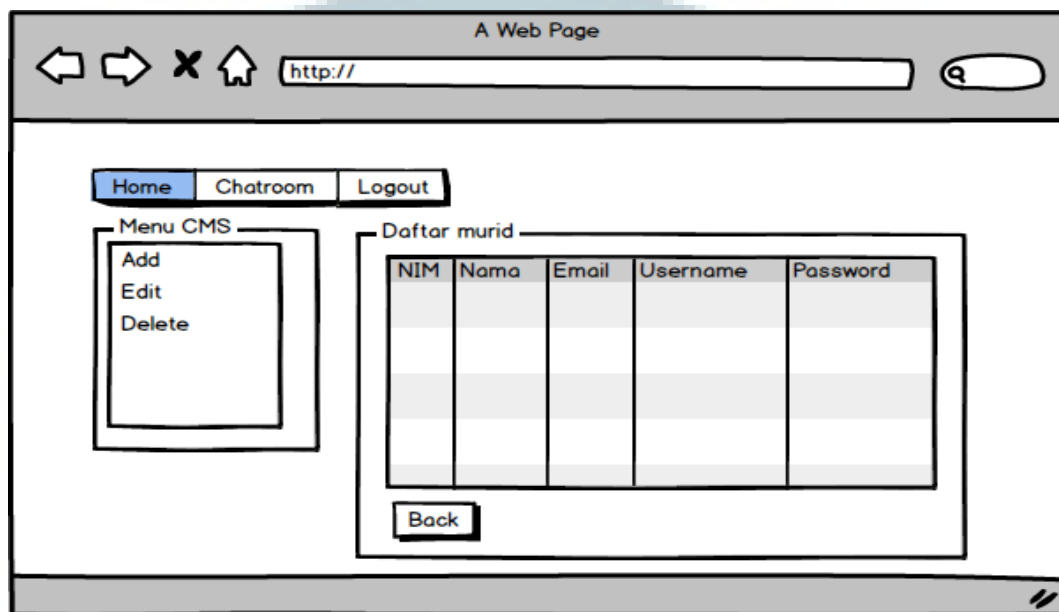
Gambar 3.21 Rancangan antarmuka halaman *insert materi*

Setelah melakukan *insert materi*, maka *admin/guru* akan masuk ke halaman *home*. Di sini *admin/guru* dapat memilih *menu* untuk melakukan *management content* di sebelah kiri.



Gambar 3.22 Rancangan antarmuka halaman *home admin*

Jika memilih murid, *admin/guru* masuk ke halaman *admin* murid. Di sebelah kiri terdapat *menu* untuk tambah murid, *edit* murid, dan hapus murid. Di sebelah kanan terdapat daftar data-data murid dan sebuah *button* 'Back' untuk kembali ke halaman sebelumnya.



Gambar 3.23 Rancangan antarmuka halaman *admin* murid

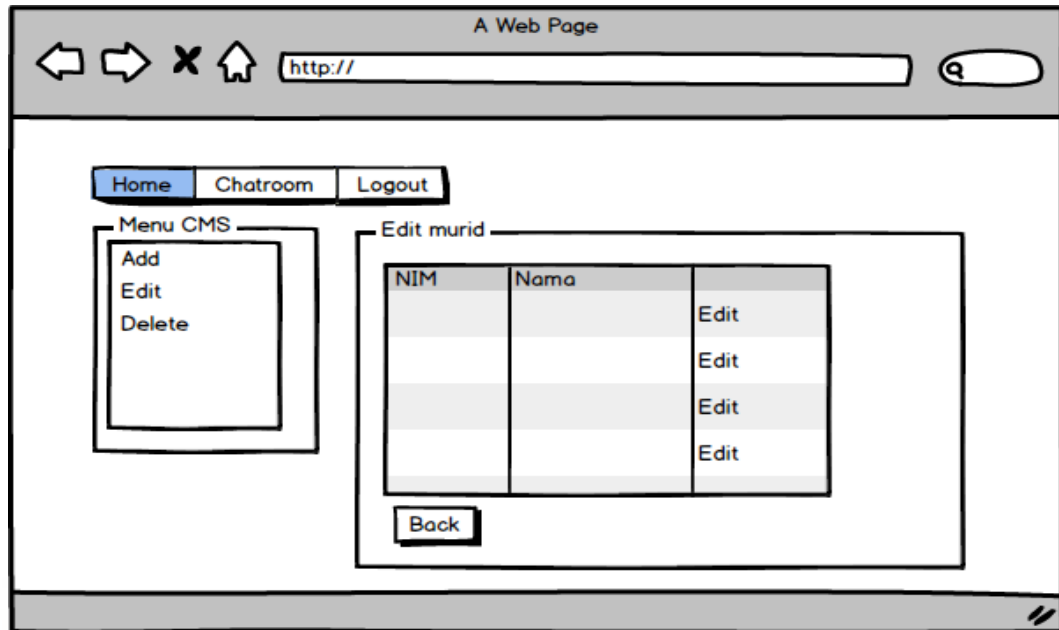
Jika *admin/guru* ingin menambah murid, maka akan muncul *input text* untuk mengisi data-data murid yang berupa nama, *email*, alamat, nomor telepon, *username*, dan *password* dari murid yang bersangkutan.

U
M
M
N

The image shows a web browser window titled "A Web Page" with a search bar containing "http://". Below the browser window is a web page layout. At the top, there are three navigation buttons: "Home", "Chatroom", and "Logout". Below these is a "Menu CMS" box containing three options: "Add", "Edit", and "Delete". To the right of the menu is a form titled "Add murid". The form contains several input fields: "Nama", "Email", "Alamat", "No Telp", "Username", and "Password". Below the "Password" field is an "Insert" button.

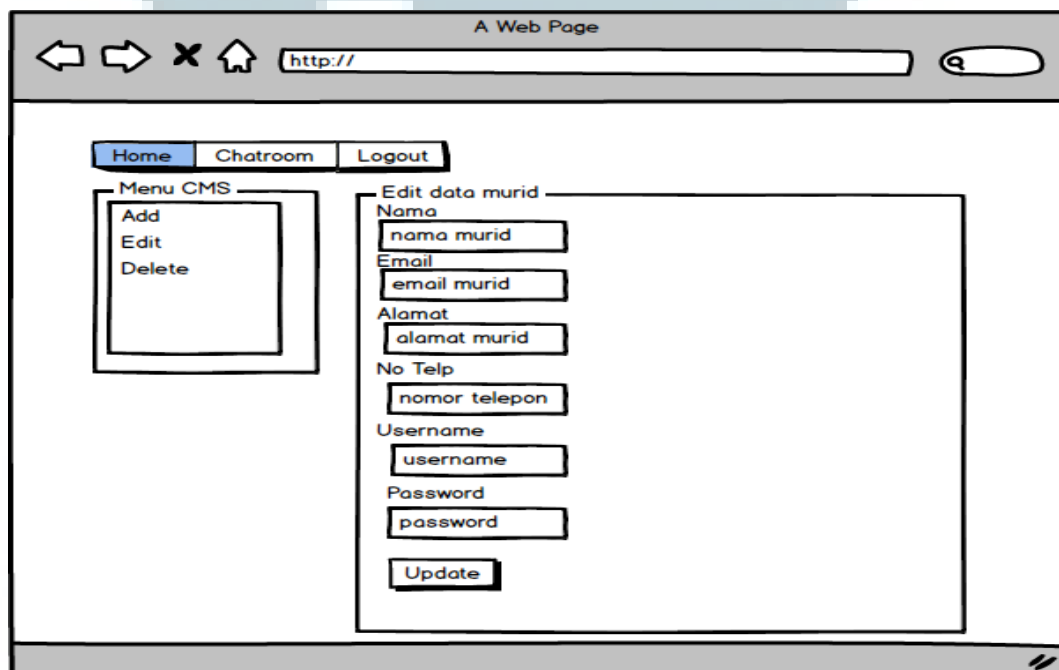
Gambar 3.24 Rancangan antarmuka halaman *add* murid

Jika *admin/guru* ingin meng-*edit* data-data dari murid, maka akan muncul tabel yang berisi nama-nama murid dan terdapat pilihan 'Edit' di kolom paling kanan dalam tabel seperti pada gambar 3.25.



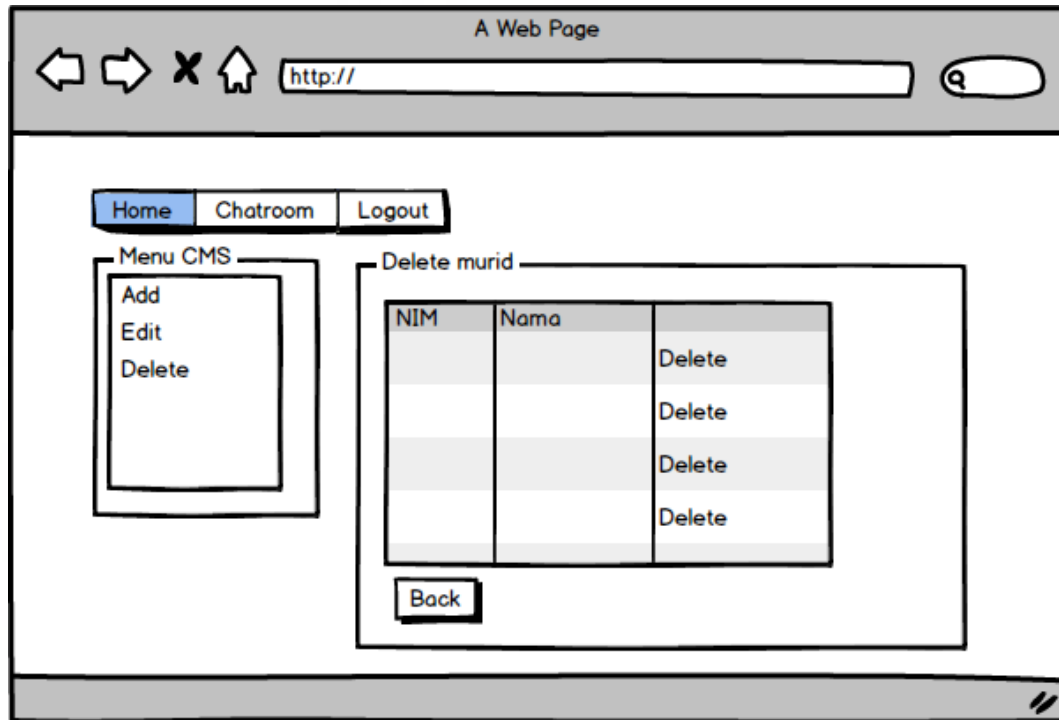
Gambar 3.25 Rancangan antarmuka halaman *edit* murid

Setelah memilih murid yang akan di-*edit* datanya, maka akan muncul *input text* yang telah terisi oleh data-data murid yang siap diubah. Setelah data-data murid telah diganti, maka klik *button* 'Update' di bagian bawah.



Gambar 3.26 Rancangan antarmuka halaman *edit* data murid

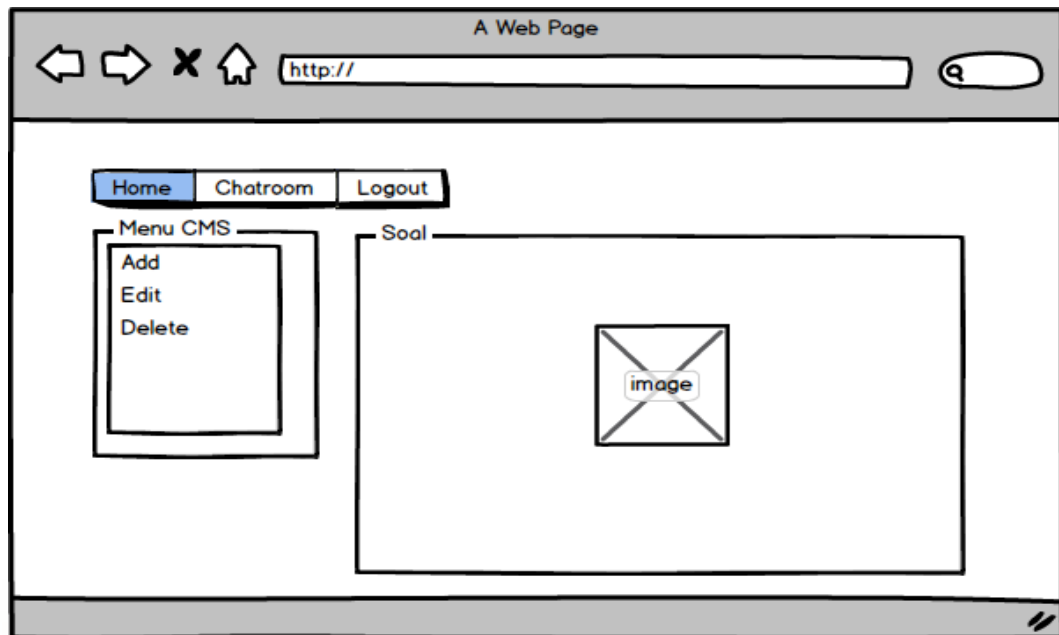
Jika *admin/guru* ingin memilih hapus murid, maka akan muncul tabel yang berisi nama-nama murid yang tercantum dalam *database*. Pada kolom paling kanan terdapat pilihan ‘Delete’ untuk menghapus murid yang dipilih.



Gambar 3.27 Rancangan antarmuka halaman hapus murid

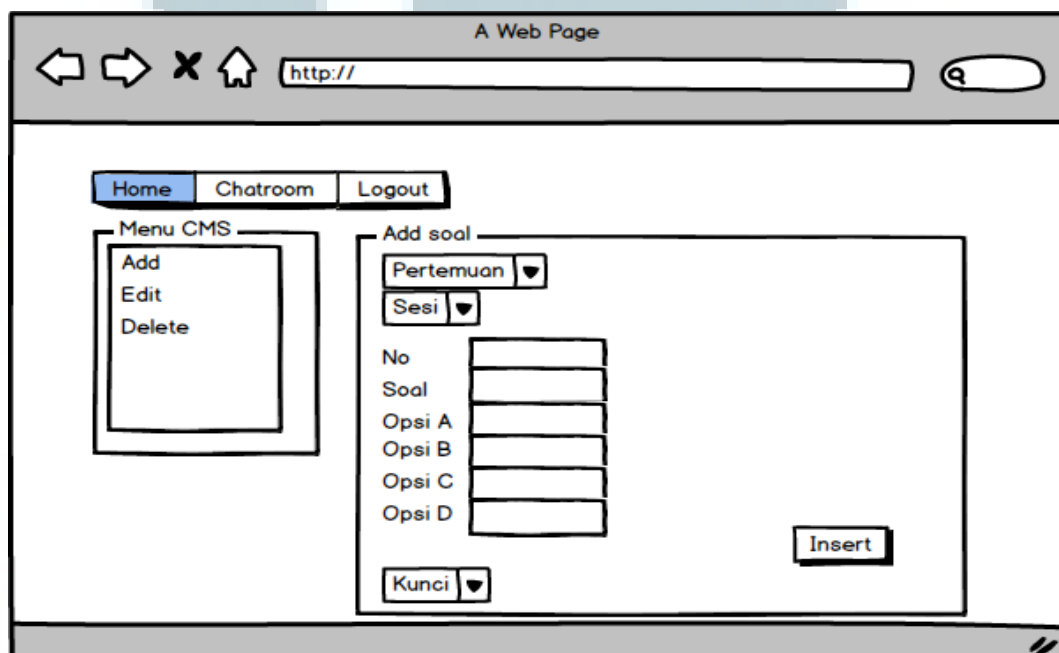
Kembali ke halaman home *admin*, untuk *menu* soal, maka *admin/guru* akan masuk ke menu *admin* soal. Pada halaman ini terdapat pilihan untuk *add* soal, *edit* soal, dan *delete* soal.

U
M
M
N



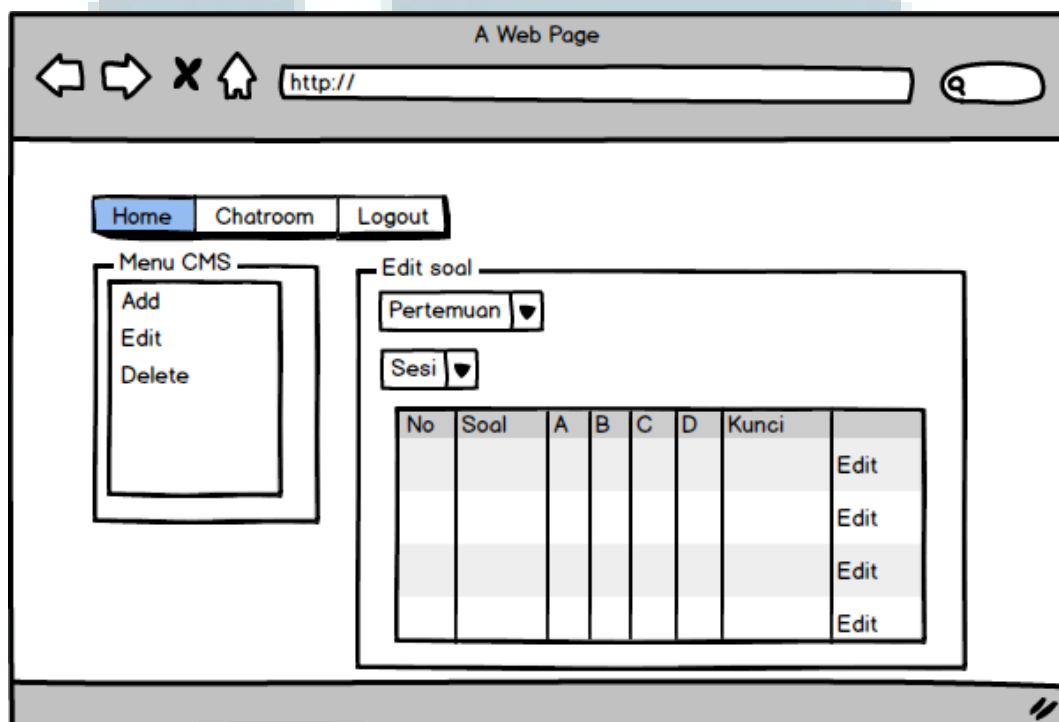
Gambar 3.28 Rancangan antarmuka halaman *admin soal*

Pada halaman *add soal*, maka akan muncul *dropdown list* untuk menentukan soal akan dimasukkan ke pertemuan dan sesi yang diinginkan serta *input text* untuk mengisi nomor urut, soal, jawaban opsi, serta kunci jawaban.



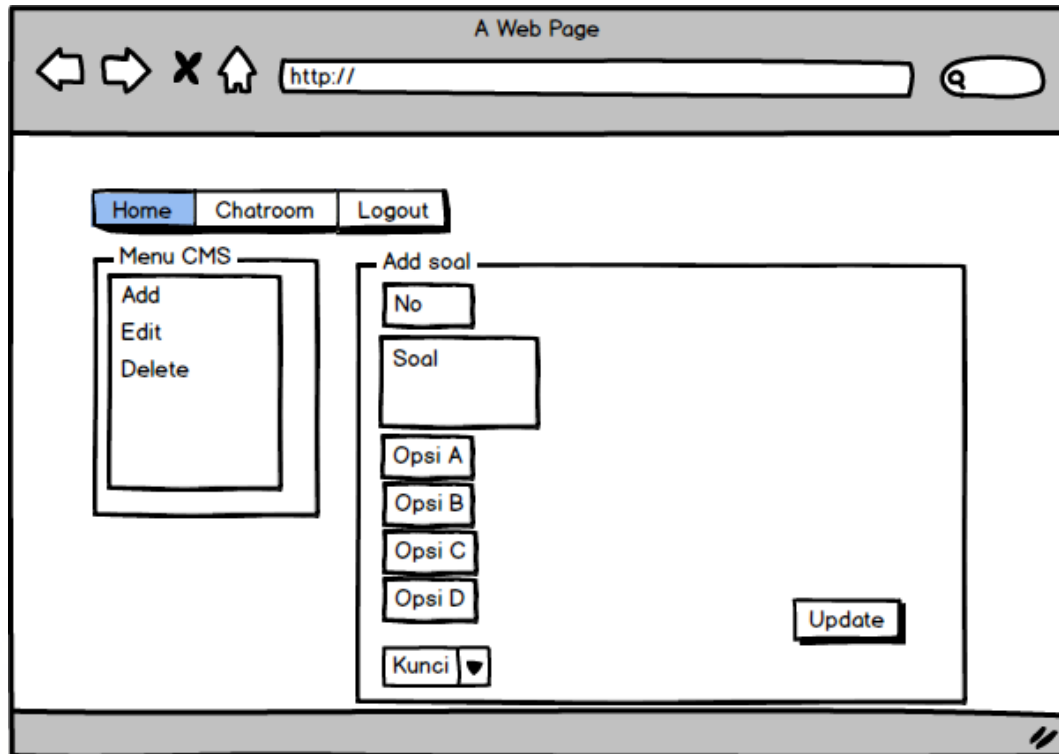
Gambar 3.29 Rancangan antarmuka halaman *add soal*

Pada halaman *edit* soal, maka akan muncul dua buah *dropdown list* untuk memilih pertemuan dan sesi soal yang akan ditampilkan. Setelah *admin/guru* memilih, maka akan muncul tabel yang berisi data-data soal sesuai dengan pertemuan dan sesi yang dipilih. Pada kolom paling kanan terdapat pilihan ‘Edit’ untuk meng-*edit* soal. Juga terdapat *button* ‘Back’ untuk kembali ke halaman sebelumnya.



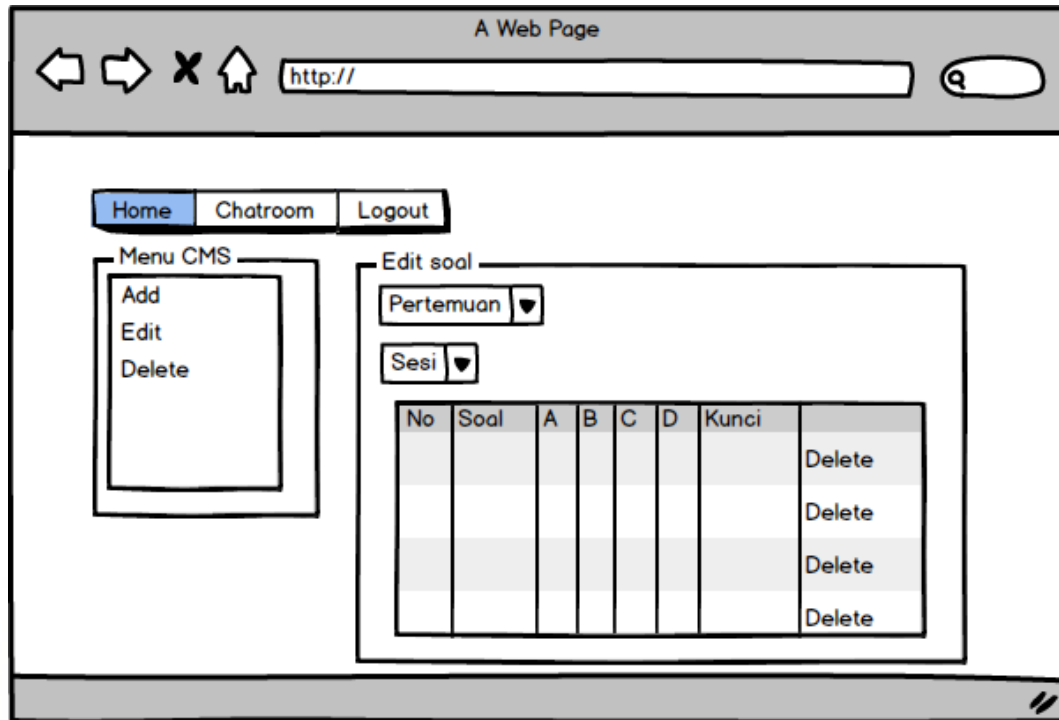
Gambar 3.30 Rancangan antarmuka halaman *edit* soal

Setelah *admin/guru* memilih soal yang akan di-*edit*, maka *admin/guru* akan masuk ke halaman *edit* data soal. Pada halaman ini, *admin/guru* dapat melakukan *edit* pada *input text* yang telah tersedia. Jika sudah selesai, *admin/guru* dapat menekan *button* ‘Update’.



Gambar 3.31 Rancangan antarmuka halaman *edit* data soal

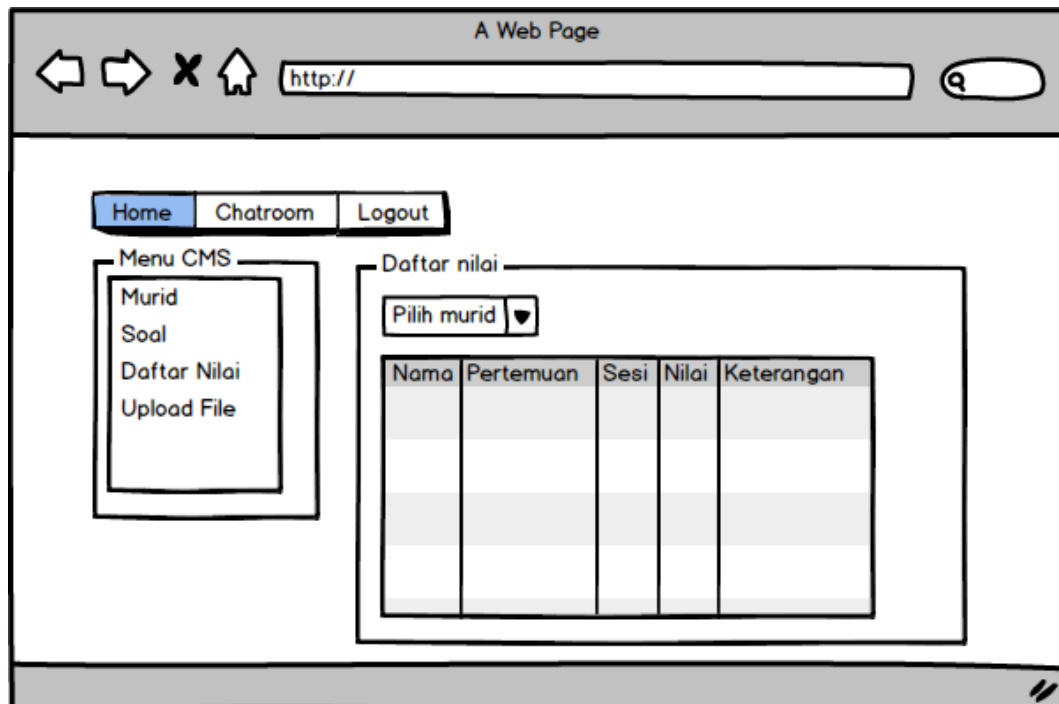
Pada halaman *delete* soal, *admin/guru* dapat menghapus soal yang tersimpan dalam *database*. Terdapat dua buah *dropdown list* untuk memilih pertemuan dan sesi dari soal yang ada. Setelah dipilih, maka akan muncul tabel yang berisi data-data soal. Pada kolom paling kanan terdapat pilihan ‘Delete’ untuk menghapus soal. Halaman *delete* soal tampak seperti pada gambar 3.32.



Gambar 3.32 Rancangan antarmuka halaman *delete* soal

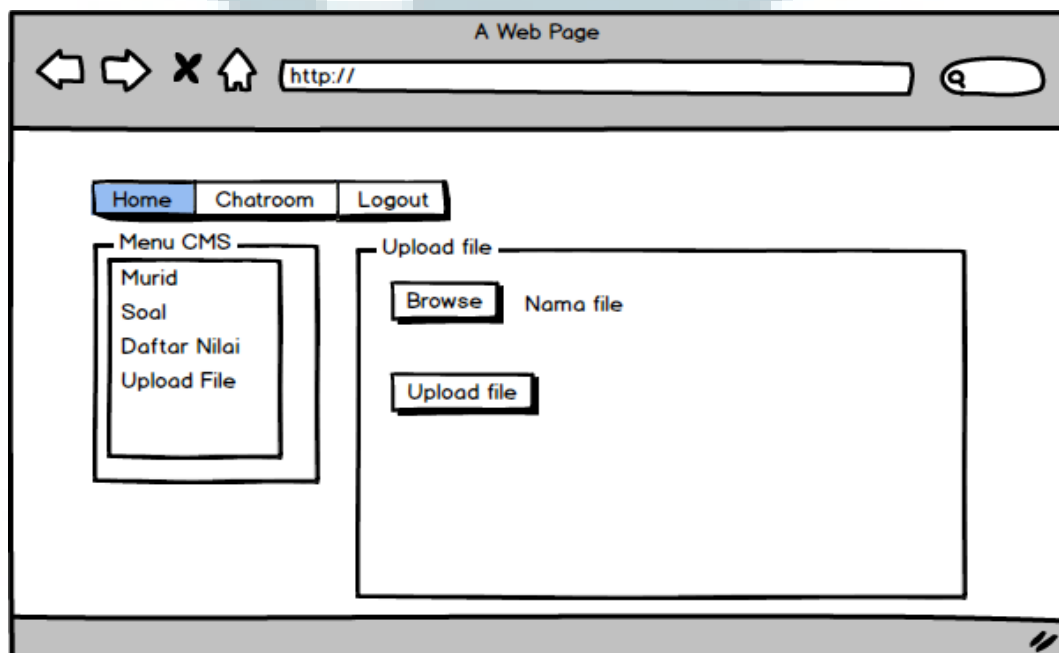
Kembali ke halaman home *admin*, jika admin/guru memilih untuk melihat daftar nilai, maka akan muncul *dropdown list* untuk memilih murid yang akan dilihat daftar nilainya. Setelah NIM dipilih, maka akan muncul tabel yang berisi daftar nilai masing-masing siswa sesuai daftar NIM-nya.

U
M
M
N



Gambar 3.33 Rancangan antarmuka halaman daftar nilai

Untuk menu *upload file*, maka akan ada dua buah *button*. *Button* pertama adalah 'Browse' untuk memilih file yang akan di-*upload* dan *button* yang kedua adalah 'Upload' untuk *action upload*.



Gambar 3.34 Rancangan antarmuka halaman unggah