BAB III
ANALISIS DAN PERANCANGAN SISTEM

3.1. Diagram Blok Sistem

Diagram blok sistem aplikasi *Try out* ujian nasional tingkat SMA menggunakan PHP ini terdiri dari tiga bagian, yaitu bagian *Server*, *Router*, dan *Smartphone*. Gambar 3.1 merupakan diagram blok sistem secara umum, berikut adalah diagram blok sistem Aplikasi *Try out* Ujian Nasional tingkat SMA menggunakan PHP :



Gambar 3.1 Diagram Blok Aplikasi *Try out* Ujian Nasional tingkat SMA menggunakan PHP

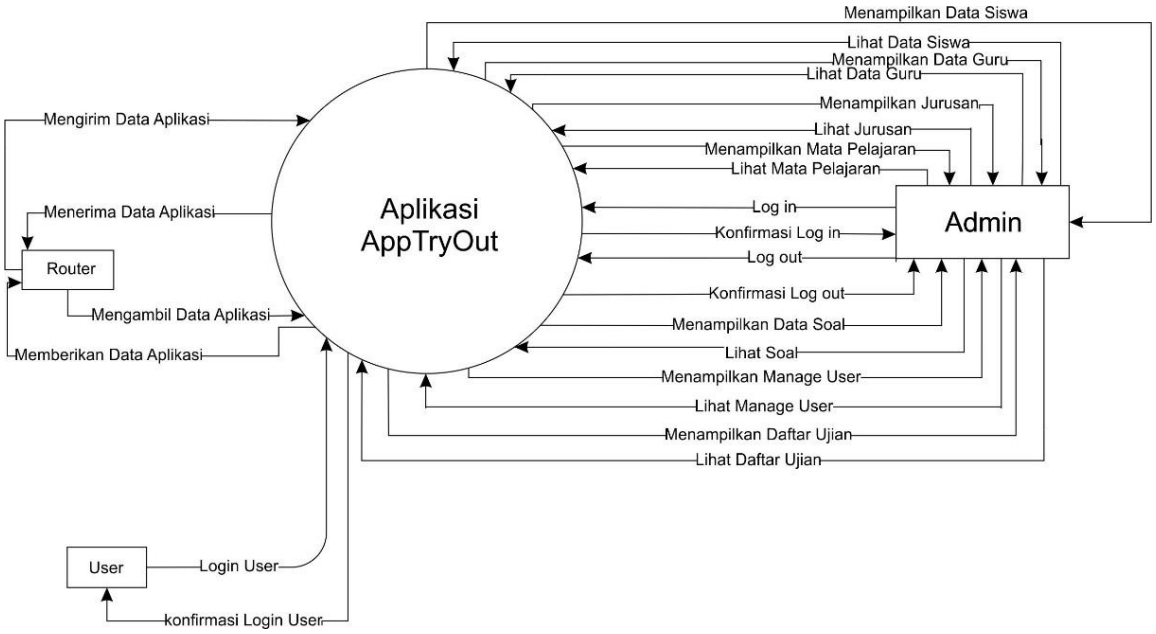
Pada Gambar 3.1 Diagram Blok Aplikasi *Try out* Ujian Nasional tingkat SMA menggunakan PHP dapat digambarkan sebagai berikut :

1. *Laptop* atau Komputer sebagai *Server* dimana *database* soal, daftar *user* maupun pembahasan dari soal disimpan ke dalam *database* MySQL
2. *Router* berfungsi sebagai pengirim paket atau sinyal *database* melalui jaringan yang tersambung ke *server* dengan meneruskan sinyal berupa *database* yang dapat tersambung dengan *Smartphone.*
3. *Smartphone* berfungsi sebagai *client, database* soal yang ada di dalam *server* dihubungkan melalui *router* dan diterima oleh *user* melalui jaringan *Hotspot* dari *Smartphone user* yang sudah diaktifkan, sehingga *user* dapat mengerjakan soal yang sudah ada dalam *database server*.

3.2. Perancangan Sistem

3.2.1. Data Context Diagram

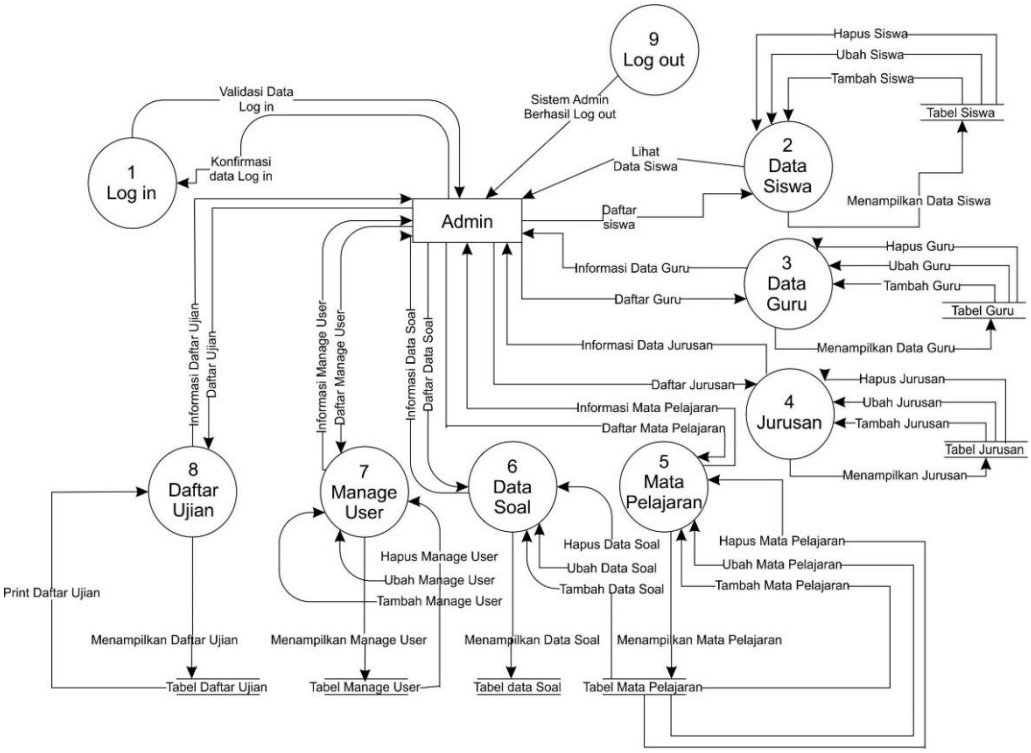
Data Context Diagram Aplikasi *Try out* Ujian Nasional tingkat SMA menggunakan PHP, Gambar 3.2 memperlihatkan DCD Aplikasi *Try out* Ujian Nasional tingkat SMA menggunakan PHP.



Gambar 3.2 DCD Aplikasi *Try out* Ujian Nasional tingkat SMA menggunakan PHP

3.2.2. DFD Level 1

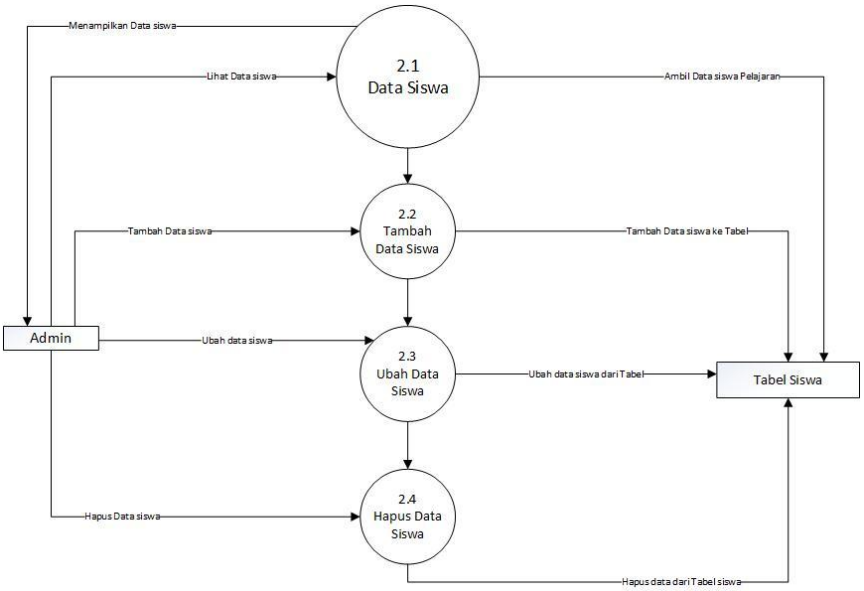
DFD Level 1 Aplikasi *Try out* Ujian Nasional tingkat SMA menggunakan PHP, Gambar 3.3 memperlihatkan DFD Level 1 Aplikasi *Try out* Ujian Nasional tingkat SMA menggunakan PHP.



Gambar 3.3 DFD Level 1 Aplikasi *Try out* Ujian Nasional tingkat SMA menggunakan PHP

3.2.3. DFD Level 2 Proses 2 (Data Siswa)

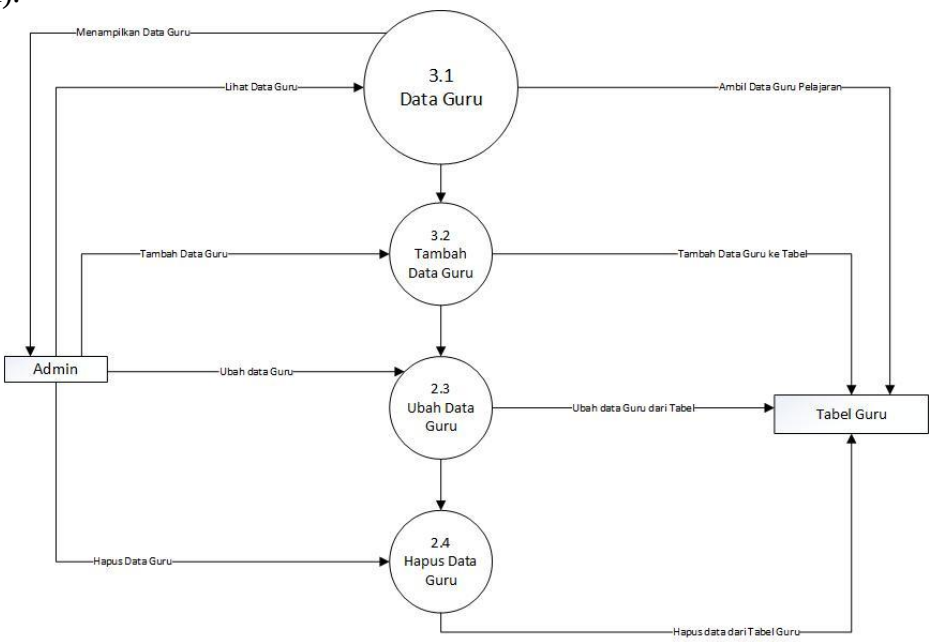
DFD Level 2 proses 2 Aplikasi *Try out* Ujian Nasional tingkat SMA menggunakan PHP, Gambar 3.4 memperlihatkan DFD Level 2 proses 2 (Data siswa).



Gambar 3.4 DFD Level 2 proses 2 (Data siswa)

3.2.4. DFD Level 2 Proses 3 (Data Guru)

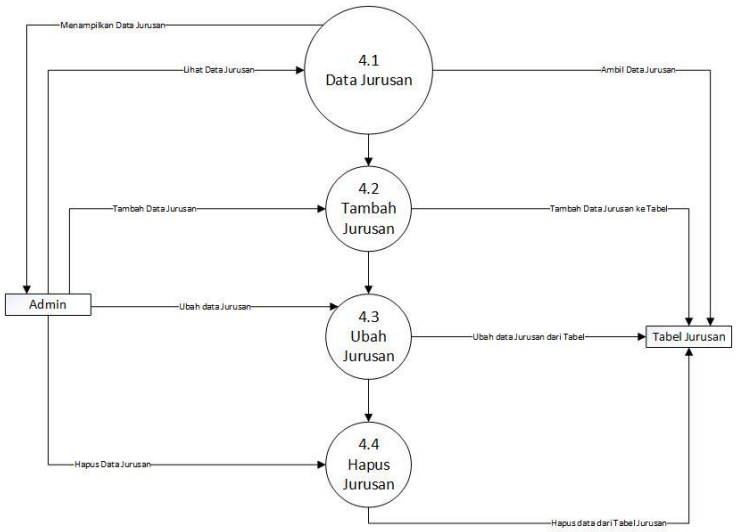
DFD Level 2 proses 3 Aplikasi *Try out* Ujian Nasional tingkat SMA menggunakan PHP, Gambar 3.5 memperlihatkan DFD Level 2 proses 3 (Data Guru).



Gambar 3.5 DFD Level 2 proses 3 (Data Guru)

3.2.5. DFD Level 2 Proses 4 (Data Jurusan)

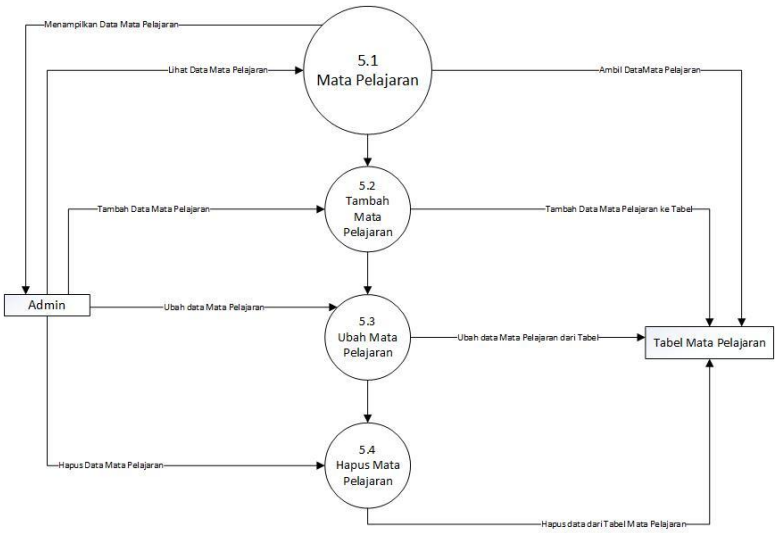
DFD Level 2 proses 4 Aplikasi *Try out* Ujian Nasional tingkat SMA menggunakan PHP, Gambar 3.6 memperlihatkan DFD Level 2 proses 4 (Data Jurusan).



Gambar 3.6 DFD Level 2 proses 4 (Data Jurusan)

3.2.6. DFD Level 2 Proses 5 (Data Mata Pelajaran)

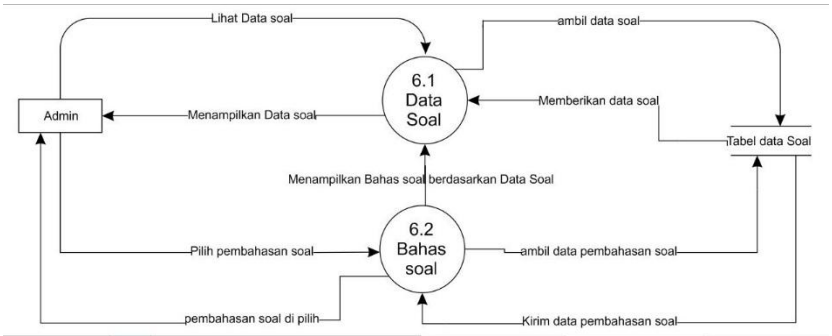
DFD Level 2 proses 5 Aplikasi *Try out* Ujian Nasional tingkat SMA menggunakan PHP, Gambar 3.7 memperlihatkan DFD Level 2 proses 5 (Mata Pelajaran).



Gambar 3.7 DFD Level 2 proses 5 (Mata Pelajaran)

3.2.7. DFD Level 2 Proses 6 (Data Soal)

DFD Level 2 proses 6 Aplikasi *Try out* Ujian Nasional tingkat SMA menggunakan PHP, Gambar 3.8 memperlihatkan DFD Level 2 proses 6 (Data Soal).



Gambar 3.8 DFD Level 2 proses 6 (Data Soal)

3.3 Deskripsi Proses Program Berbasis Web

3.3.1 Proses 1 (Login)

Input : *Login*, Konfirmasi *login.*

Output : Validasi data *login*, Konfirmasi *login.*

Algoritma :

1. *Admin* melakukan *login* ke proses *login* Aplikasi AppTryOut.
2. Proses *login* melakukan validasi data *login* ke *database* tabel *admin.*
3. *Database* tabel *admin* memberikan konfirmasi data *login* ke proses *login.*
4. Proses *login* memberikan konfirmasi *login* ke *admin.*

3.3.2 Proses 2 (Data Siswa)

Input : Data Siswa, Tambah data siswa, Ubah data siswa, Hapus data siswa.

Output : Tampilan data siswa, Tabel siswa.

Algoritma :

1. Koneksi *Database.*
2. *Admin* melihat daftar siswa yang menuju ke proses.
3. *Admin* input data siswa ke proses data siswa yang ada di *server.*
4. *Admin* ubah data siswa ke proses data siswa yang ada di *server.*

5. *Admin* hapus data ke proses data siswa yang ada di *server.*
6. Menampilkan data siswa.
7. Menampilkan tabel siswa dari proses data siswa.

3.3.3 Proses 3 (Data Guru)

Input : Data guru, Tambah data guru, Ubah data guru, Hapus data guru.

Output : Tampilan data guru, Tabel guru.

Algoritma :

1. Koneksi *Database.*
2. *Admin* melihat daftar guru menuju ke proses data guru.
3. *Admin* input data guru ke proses data guru yang ada di *server.*
4. *Admin* ubah data guru ke proses data guru yang ada di *server.*
5. *Admin* hapus data guru ke proses data guru yang ada di *server.*
6. Menampilkan daftar guru.
7. Menampilkan informasi tabel guru.

3.3.4 Proses 4 (Data Jurusan)

Input : Data jurusan, Tambah jurusan, Ubah jurusan, Hapus jurusan.

Output : Tampilan daftar jurusan, informasi tabel jurusan.

Algoritma :

1. Koneksi *Database.*
2. *Admin* melihat daftar jurusan yang menuju ke proses jurusan.
3. *Admin* input data ke proses data jurusan yang ada *server.*
4. *Admin* ubah data ke proses data jurusan yang ada *server.*
5. *Admin* hapus data ke proses data jurusan yang ada *server.*
6. Menampilkan daftar jurusan.
7. Menampilkan informasi data jurusan.

3.3.5 Proses 5 (Mata Pelajaran)

Input : Daftar mata pelajaran, Tambah mata pelajaran, Ubah mata pelajaran, Hapus mata pelajaran.

Output : Tampilan tabel mata pelajaran, informasi mata pelajaran.

Algoritma :

1. Koneksi *Database.*

2. *Admin* melihat mata pelajaran.
3. *Admin* input data ke proses data mata pelajaran yang ada *server.*
4. *Admin* ubah data ke proses data mata pelajaran yang ada *server.*
5. *Admin* hapus data ke proses data mata pelajaran yang ada *server.*
6. Menampilkan informasi mata pelajaran.
7. Menampilkan mata pelajaran.

3.3.6 Proses 6 (Data Soal)

Input : Daftar data soal, Tambah data soal, Ubah data soal, Hapus data soal.

Output : Tampilan data soal, informasi data soal.

Algoritma :

1. Koneksi *Database.*
2. *Admin* melihat data soal.
3. *Admin* input data soal ke proses data soal yang ada *server.*
4. *Admin* ubah data soal ke proses data soal yang *server.*
5. *Admin* hapus data soal ke proses data soal yang *server.*
6. Menampilkan informasi data soal untuk *Admin.*
7. Menampilkan daftar data soal untuk *Admin.*

3.3.6.1 Proses 6.1 (Data Soal)

Input : Memberikan data soal, Lihat data soal.

Output : ambil data soal, informasi data soal, menampilkan bahas soal berdasarkan data soal

Algoritma :

1. Koneksi *Database.*
2. *Admin* lihat data soal.
3. Ambil data soal.
4. menampilkan bahas soal berdasarkan data soal ke *server.*
5. Menampilkan data soal.

3.3.6.2 Proses 6.2 (Bahas Soal)

Input : Pilih pembahasan soal, Kirim data pembahasan.

Output : Tampilan tabel data pembahasan soal, pembahasan soal dipilih, ambil data pembahasan soal.

Algoritma :

1. Koneksi *Database.*
2. Ambil data soal ke tabel data soal.
3. Tabel data soal memberikan data ke proses data soal.
4. Bahas soal menampilkan berdasarkan data soal.
5. Ambil data pembahasan soal ke tabel data soal .
6. Tabel data soal mengirim data pembahasan ke proses bahas soal.
7. Pembahasan soal dipilih admin.

3.3.7 Proses 7 (Manage User)

Input : Daftar *manage user*, Tambah *manage user*, Ubah *manage user*, Hapus *manage user*.

Output : Tampilan *manage user*, informasi *manage user*.

Algoritma :

1. Koneksi *Database.*
2. *Admin* melihat daftar manage user .
3. *Admin* input data manage user ke proses *manage user.*
4. *Admin* ubah data manage user ke proses *manage user.*
5. *Admin* hapus data manage user ke proses *manage user.*
6. Menampilkan informasi data *manage user*.
7. Menampilkan daftar data *manage user*.

3.3.8 Proses 8 (Daftar Ujian)

Input : Daftar ujian, Print data daftar ujian.

Output : Tampilan daftar ujian, informasi daftar ujian.

Algoritma :

1. Koneksi *Database.*
2. *Admin* melihat daftar daftar ujian.
3. *Admin* melakukan print daftar ujian dari tabel daftar ujian.
4. Menampilkan informasi data daftar ujian dari proses daftar ujian.

3.3.9 Proses 9 (Log out)

Input : Sistem *admin* berhasil *log out* .

Output : Tampilan *Log out.*

Algoritma :

1. Koneksi *Database.*
2. Sistem *admin* berhasil *logout*.

3.4 Perancangan *Use case diagram*

Tahapan yang akan dilakukan dalam membuat *Use Case Diagram* adalah menentukan *actor*, menentukan kebutuhan *Use Case* yang berguna untuk mengidentifikasikan kebutuhan *use case* dalam Aplikasi *Try out* Ujian Nasional tingkat SMA menggunakan PHP, dan interaksi antara *actor-actor* terhadap *use case–use case* yang telah didefinisikan melalui pembuatan use case diagram. Berikut merupakan *use case* pada aplikasi yang dibuat :

1. *Actor*

   Actor-actor yang terdapat dalam sistem ini adalah

   • *Administrator/admin* : mempunyai hak akses untuk melakukan *input*, *update* dan *delete* ujian, soal, jawaban, nilai, dan *user*.

   • *User* : dapat melakukan *Try out* dan melihat nilai yang didapat dari *Try out* yang dilakukan oleh *user*.

2. Kebutuhan *Use Case*

   Setelah mengidentifikasi *actor*, tahap selanjutnya adalah menentukan kebutuhan fungsi *use case* yang dibutuhkan oleh sistem dalam interaksinya dengan *actor-actor* tersebut.

Tabel 3.1 memperlihatkan Kebutuhan *Use Case* pada perancangan *use case* diagram Aplikasi *Try Out* ujian nasional tingkat SMA menggunakan PHP.

Tabel 3.1 Kebutuhan *Use Case*

No	Kebutuhan	Aktor	Use case
1	Sebelum masuk sistem *admin* harus *login*	*Admin*	*login*
2	*Admin* dapat menambah, mengedit dan menghapus soal ujian	*Admin*	Mengelola soal
3	*Admin* dapat menghapus menambah jawaban	*Admin*	Mengelola jawaban
4	*Admin* dapat menambah dan menghapus *user*	*Admin*	Mengelola *user*
5	Sebelum masuk sistem *user* harus *login*	*User*	*login*
6	*User* dapat memilih jurusan	*User*	Memilih jurusan
7	*User* memilih ujian yang dikerjakan	*User*	Memilih ujian
8	*User* dapat melihat hasil ujian	*User*	Memilih hasil
9	*User* dapat melihat rangking dari beberapa *user* yang sudah melakukan ujian	*User*	Memilih rangking
10	*Admin* dan *user* dapat keluar sistem	*Admin, user*	*logout*

Tabel 3.2. memperlihatkan Skenario *Use Case* pengelolaan soal *Try out* pada perancangan Skenario *Use Case* pengelolaan soal *Try out* pada Aplikasi *Try Out* ujian nasional tingkat SMA menggunakan PHP.

Tabel 3.2. Skenario *Use Case* pengelolaan soal *Try out*

Nama Use Case	Pengelolaan soal *Try out*
Aktor	*Admin*
Tujuan	Memperbarui Data Soal *Try out*
Keadaan awal	Sistem menampilkan Halaman *Login*
Step	***Action***
1	Membuka Aplikasi *Server*
2	Menampilkan Halaman *Login*
3	Mengisi Form *Login* dengan memasukan *username* dan *password* yang sesuai
4	Memeriksa kesesuaian antara *username* dan *password*
5	Jika *username* dan *password* benar maka sistem akan menampilkan halaman menu utama
6	*Admin* memilih menu soal
7	Menampilkan Halaman soal
8	*Admin* Menambah, Edit, Hapus soal
9	Menambah atau Mengedit, Menghapus Data soal pada *database*
10	Kondisi Akhir Menampilkan Halaman Soal

Perancangan Skenario *Use Case* pengelolaan isi pembahasan materi Pelajaran pada Aplikasi *Try Out* ujian nasional tingkat SMA menggunakan PHP, Tabel 3.3 memperlihatkan Skenario *Use Case* pengelolaan isi pembahasan materi Pelajaran.

Tabel 3.3 Skenario *Use Case* pengelolaan isi pembahasan materi Pelajaran

Nama Use Case	Pengelolaan isi pembahasan materi pelajaran
Aktor	*Admin*
Tujuan	Memperbarui data pembahasan Materi pelajaran
Keadaan awal	Sistem menampilkan Halaman *Login*

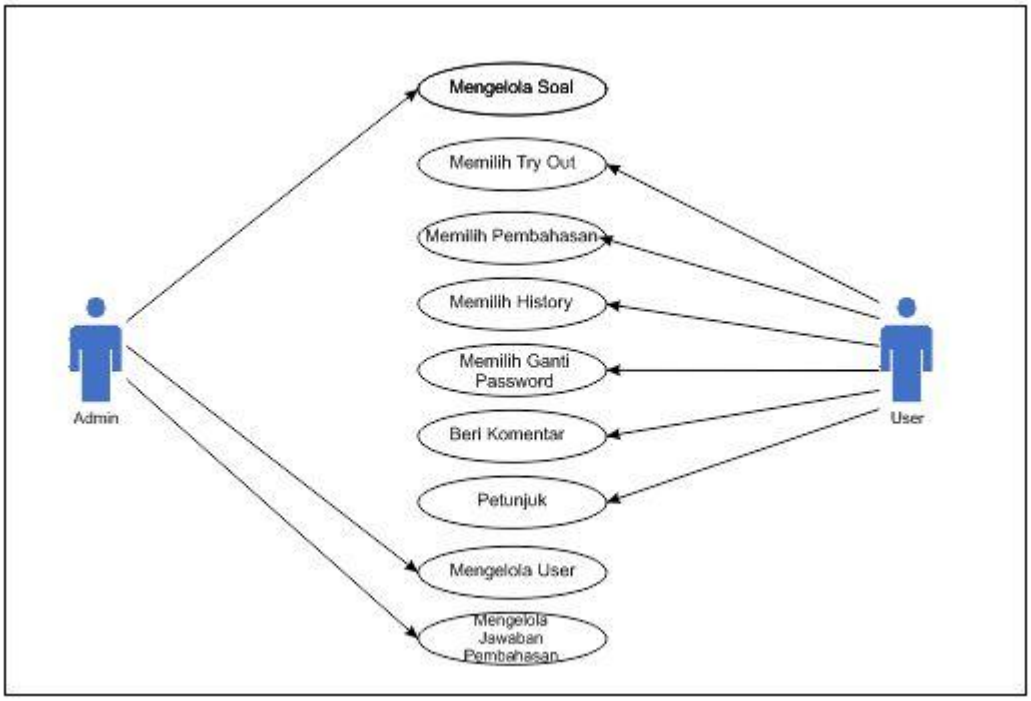
Step	***Action***
1	Membuka Aplikasi *Server*
2	Menampilkan Halaman *Login*
3	Mengisi Form *Login* dengan memasukan *username* dan *password* yang sesuai
4	Memeriksa kesesuaian antara *username* dan *password*
5	Jika *username* dan *password* benar maka sistem akan menampilkan halaman menu utama
6	*Admin* memilih menu materi pembahasan
7	Menampilkan Halaman pembahasan
8	*Admin* Menambah, Edit, Hapus pembahasan materi
9	Menambah atau Mengedit, Menghapus Data pembahasan materi pada *database*
10	Kondisi Akhir Menampilkan Halaman pembahasan

Tabel 3.4 memperlihatkan Skenario *Use Case* pengerjaan soal *Try out* pada perancangan pengerjaan soal pada Aplikasi *Try Out* ujian nasional tingkat SMA menggunakan PHP.

Tabel 3.4 Skenario *Use Case* pengerjaan soal *Try out*

Nama Use Case	Pengerjaan soal *Try out*
Aktor	*User*
Tujuan	Mengerjakan soal *Try out*
Keadaan awal	Aplikasi menampilkan Halaman *Login*
Step	***Action***
1	Membuka Aplikasi
2	Menampilkan Halaman *Login*
3	Mengisi Form *Login* dengan memasukan *username* dan *password* yang sesuai
4	Memeriksa kesesuaian antara *username* dan *password*
5	Jika *username* dan *password* benar maka sistem akan menampilkan halaman menu utama
6	*user* memilih menu ujian
7	Menampilkan Halaman pilih jurusan
8	*User* memilih jurusan yang diinginkan
9	Menampilkan halaman pilih mata pelajaran
10	*User* memilih mata pelajaran yang diinginkan
11	Menampilkan soal *Try out*
12	*User* mengerjakan soal
13	Kondisi Akhir User selesai mengerjakan soal, maka sistem akan menampilkan hasil

Gambar 3.9 merupakan perancangan *Use Case Diagram* pada Aplikasi *Try Out* ujian nasional tingkat SMA menggunakan PHP.

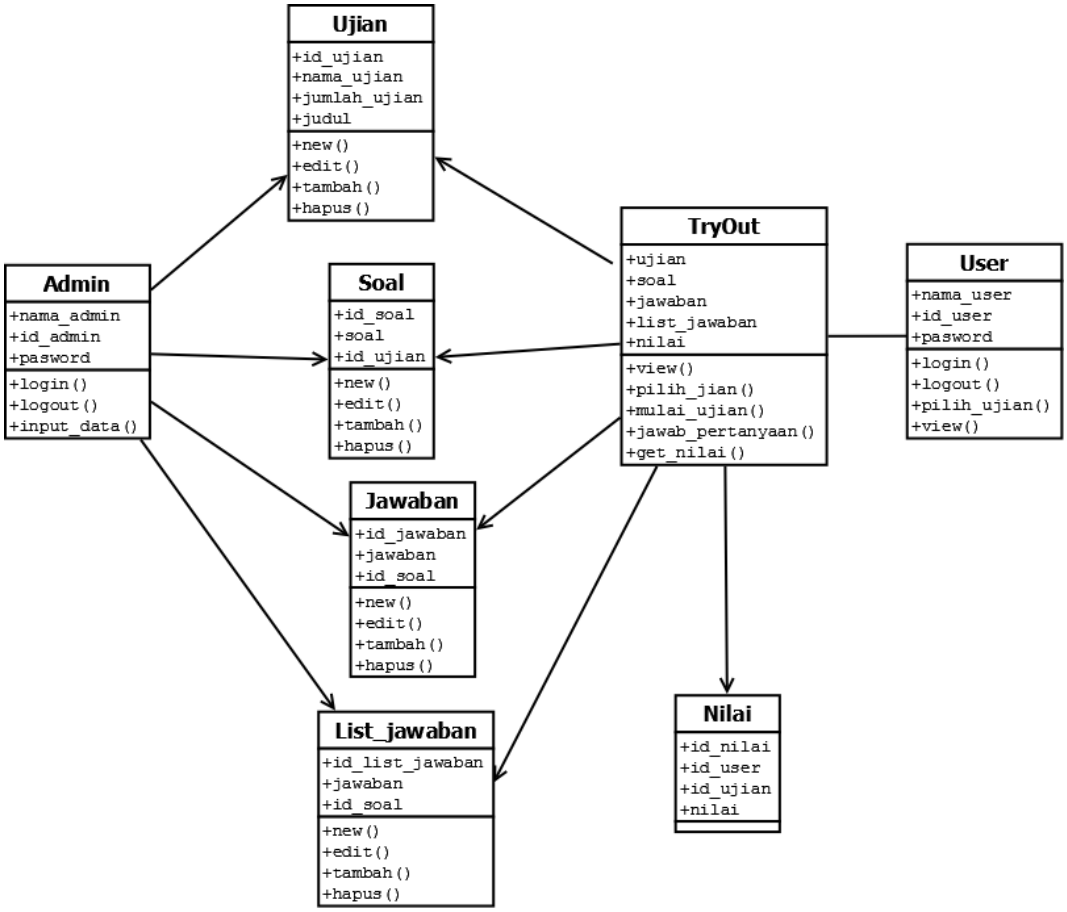


Gambar 3.9 *Use Case Diagram*

3.5 Perancangan *Class diagram*

Class diagram merupakan diagram yang menunjukkan kebutuhan *class-class* dalam suatu sistem program, dimana *class-class* tersebut mengandung atribut-atribut dan *operation-operation* yang dibutuhkan. *Class* juga merupakan pembentuk utama dari sistem berorientasi obyek karena *class* menunjukkan kumpulan obyek yang memiliki atribut dan operasi yang sama. *Class* digunakan untuk mengimplementasikan *interface*. Dalam satu *class* terdapat atribut dan *operation* yang berkaitan dengan *class* tersebut.

Gambar 3.10 merupakan perancangan *Class Diagram* pada Aplikasi *Try Out* ujian nasional tingkat SMA menggunakan PHP.



Gambar 3.10 *Class Diagram*

3.6 Perancangan *Sequence diagram*

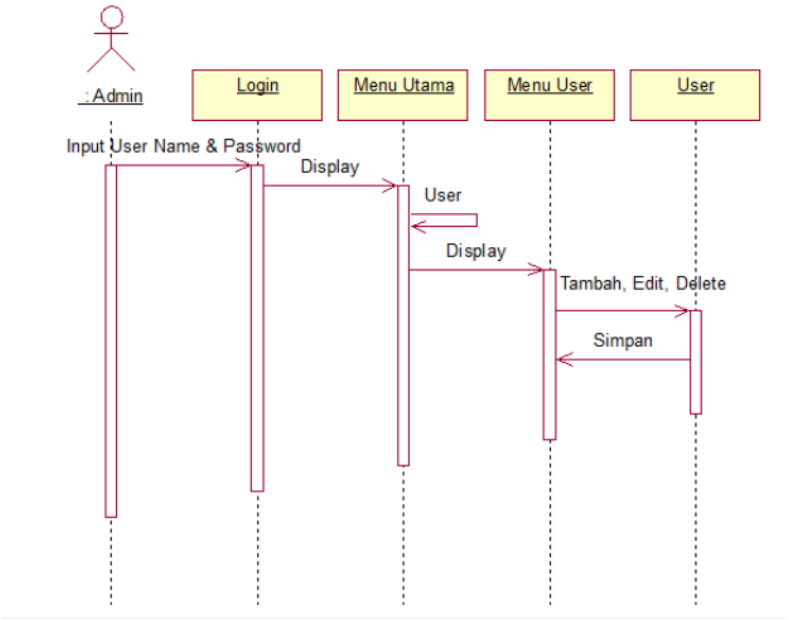
Sequence diagram menggambarkan pola hubungan antara sekumpulan obyek yang saling mempengaruhi menurut urutan waktu.

3.6.1 *Sequence diagram Try out*

Sebelum masuk ke menu utama, *user* diharuskan *login* dengan memasukan *username* dan *password* terlebih dahulu, lalu *user* memilih untuk melakukan ujian *Try out* yang telah di sediakan, setelah memilih *Try out* sistem akan memperlihatkan soal-soal yang harus dikerjakan, apabila *user* sudah selesai

mengerjakan soal-soal ujian *Try out*, sistem akan memberikan nilai yang didapatkan dari pengerjaan soal-soal ujian *Try out*.

Gambar 3.11 memperlihatkan perancangan *Sequence Diagram Try Out* pada Aplikasi *Try Out* ujian nasional tingkat SMA menggunakan PHP.

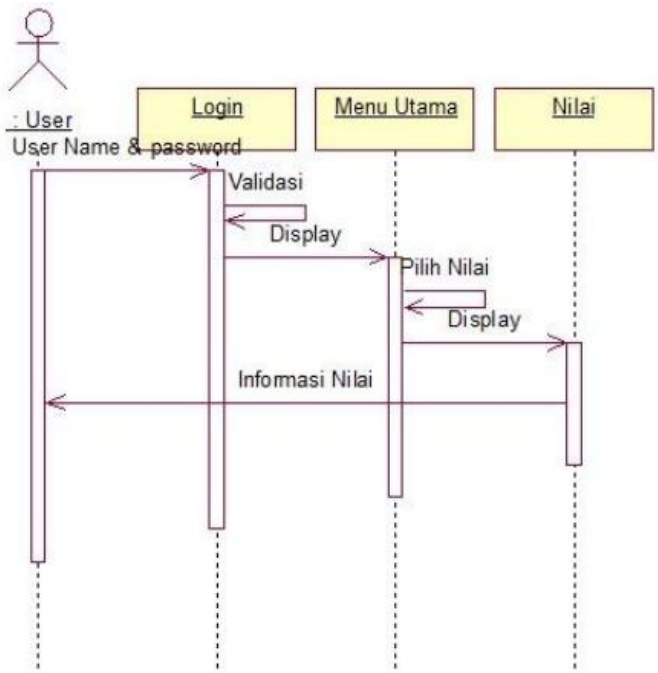


Gambar 3.11 *Sequence Diagram Try Out*

3.6.2 *Sequence diagram view* nilai

User diharuskan melakukan *login* dengan memasukan *username* dan *password* terlebih dahulu lalu *user* memilih menu nilai pada menu utama, kemudian sistem akan menampilkan daftar nilai.

Gambar 3.12 memperlihatkan perancangan *Sequence Diagram View* Nilai pada Aplikasi *Try Out* ujian nasional tingkat SMA menggunakan PHP.

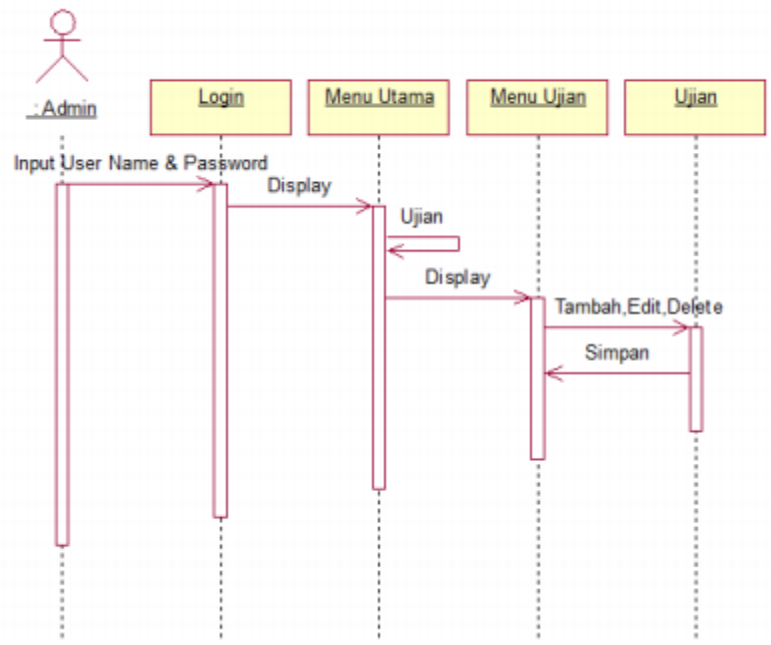


Gambar 3.12 *Sequence Diagram View* Nilai

3.6.3 *Sequence diagram* mengelola ujian

Sequence diagram mengelola ujian memperlihatkan proses *administrator* melakukan *update* ujian. Untuk masuk kedalam menu utama *administrator* harus melakukan *login* dengan memasukan *username* dan *password* pada menu *login*, setelah *login* akan menuju proses menu utama dan pilih menu *update* ujian. Setelah selesai *update* ujian maka data tersimpan pada *database* ujian.

Gambar 3.13 memperlihatkan perancangan *Sequence Diagram* Mengelola Ujian pada Aplikasi *Try Out* ujian nasional tingkat SMA menggunakan PHP.



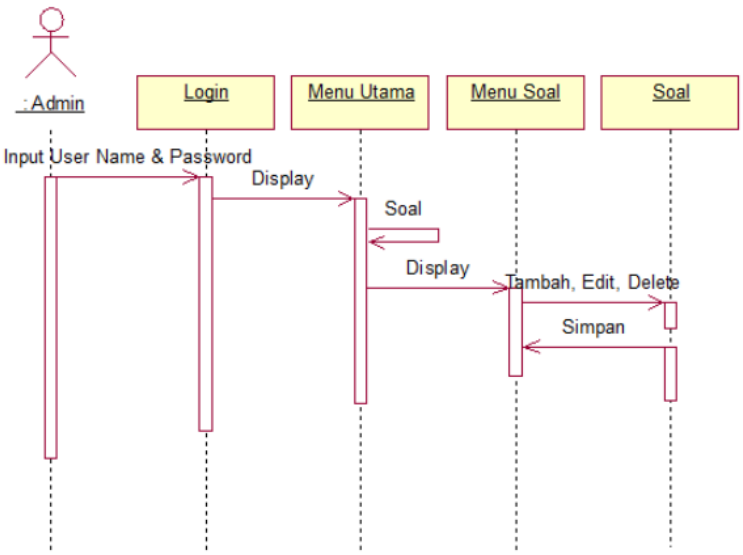
Gambar 3.13 *Sequence Diagram*

Mengelola Ujian

3.6.4 *Sequence diagram* mengelola soal

Sequence diagram mengelola soal melihatkan proses *administrator* melakukan *update* soal. Untuk masuk kedalam menu utama *administrator* harus melakukan *login* dengan memasukan *username* dan *password* pada menu *login*, setelah *login* akan menuju proses menu utama dan pilih menu *update* soal. Setelah selesai *update* soal maka data tersimpan pada *database* soal.

Gambar 3.14 memperlihatkan perancangan *Sequence Diagram* Mengelola Soal pada Aplikasi *Try Out* ujian nasional tingkat SMA menggunakan PHP.

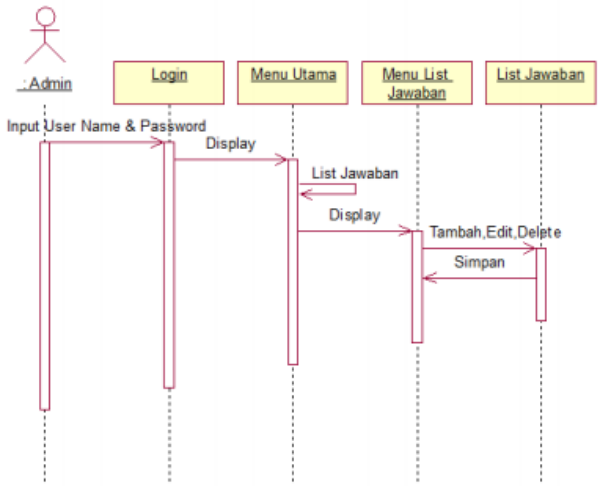


Gambar 3.14 *Sequence Diagram* Mengelola Soal

3.6.5 *Sequence diagram* mengelola jawaban

Sequence diagram mengelola jawaban melihatkan proses *administrator* melakukan *update* Jawaban. Untuk masuk kedalam menu utama *administrator* harus melakukan *login* dengan memasukan *username* dan *password* pada menu *login*, setelah *login* akan terlihat menu utama dan pilih menu *update* Jawaban. Setelah selesai *update* Jawaban maka data tersimpan pada *database* Jawaban.

Gambar 3.15 memperlihatkan perancangan *Sequence Diagram* Mengelola Jawaban pada Aplikasi *Try Out* ujian nasional tingkat SMA menggunakan PHP.

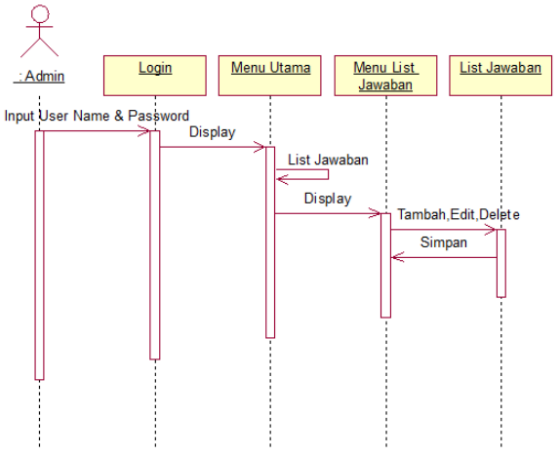


Gambar 3.15 *Sequence Diagram* Mengelola Jawaban

3.6.6 *Sequence diagram* mengelola list jawaban

Sequence diagram mengelola list Jawaban memperlihatkan proses *administrator* melakukan *update* list Jawaban. Untuk masuk kedalam menu utama *administrator* harus melakukan *login* dengan memasukan *user name* dan *password* pada menu *login*, setelah *login* berhasil akan terlihat menu utama dan pilih menu *update* list Jawaban. Setelah selesai *update* list jawaban maka data tersimpan pada *database* list Jawaban.

Gambar 3.16 memperlihatkan perancangan *Sequence Diagram* Mengelola List Jawaban pada Aplikasi *Try Out* ujian nasional tingkat SMA menggunakan PHP.



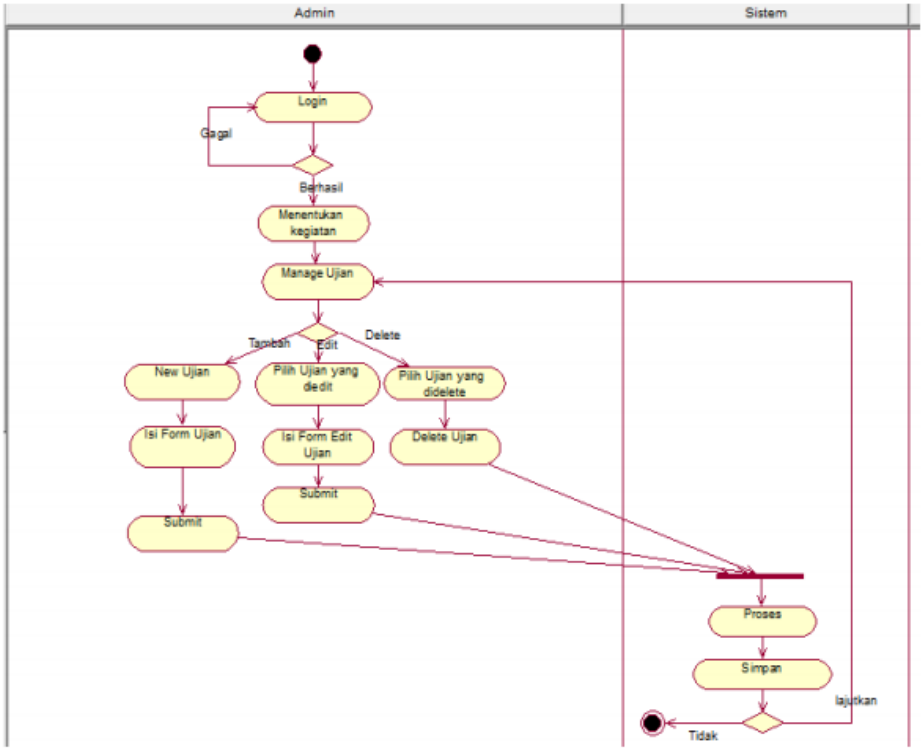
Gambar 3.16 *Sequence Diagram* Mengelola List Jawaban

3.7 Perancangan *activity diagram*

Activity diagram merupakan salah satu cara untuk memodelkan *event-event* yang terjadi dalam suatu *use case*. Memperlihatkan aliran kendala dari satu aktivitas ke aktivitas lainnya. Pada diagram di bawah ini menjelaskan sistem secara keseluruhan, dimana dapat dilihat aliran-aliran aktivitas atau proses yang terjadi di dalam sistem.

3.7.1 *Activity Diagram Manage* Ujian

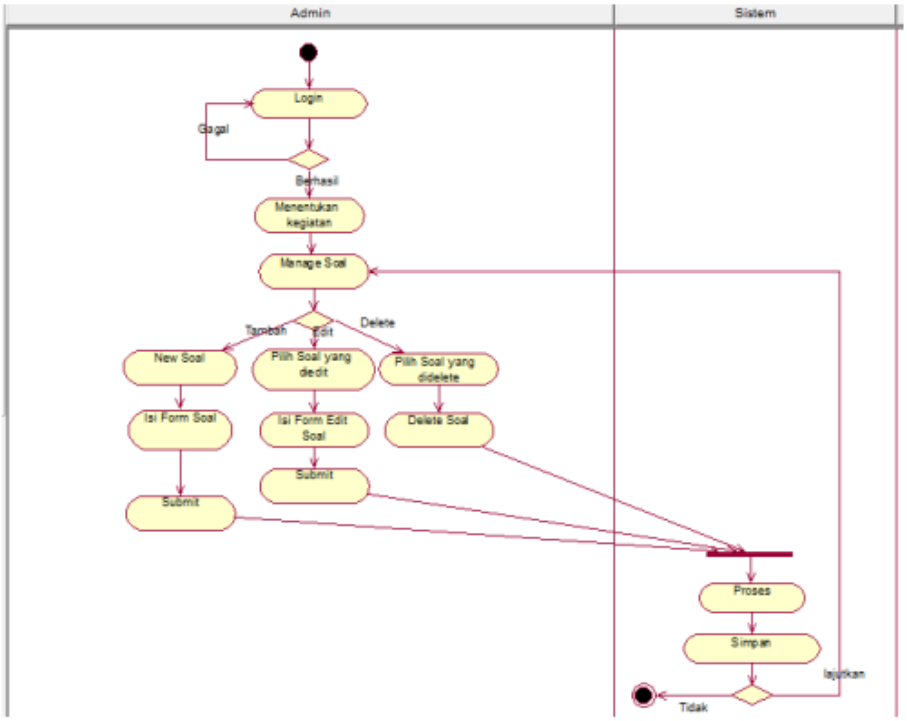
Activity Diagram Manage ujian merupakan pemodelan dari aktivitas ke aktivitas lainnya pada proses *Manage* Ujian. Gambar 3.17 memperlihatkan *Activity Diagram Manage* Ujian.



Gambar 3.17 *Activity Diagram Manage* Ujian

3.7.2 *Activity Diagram Manage* Soal

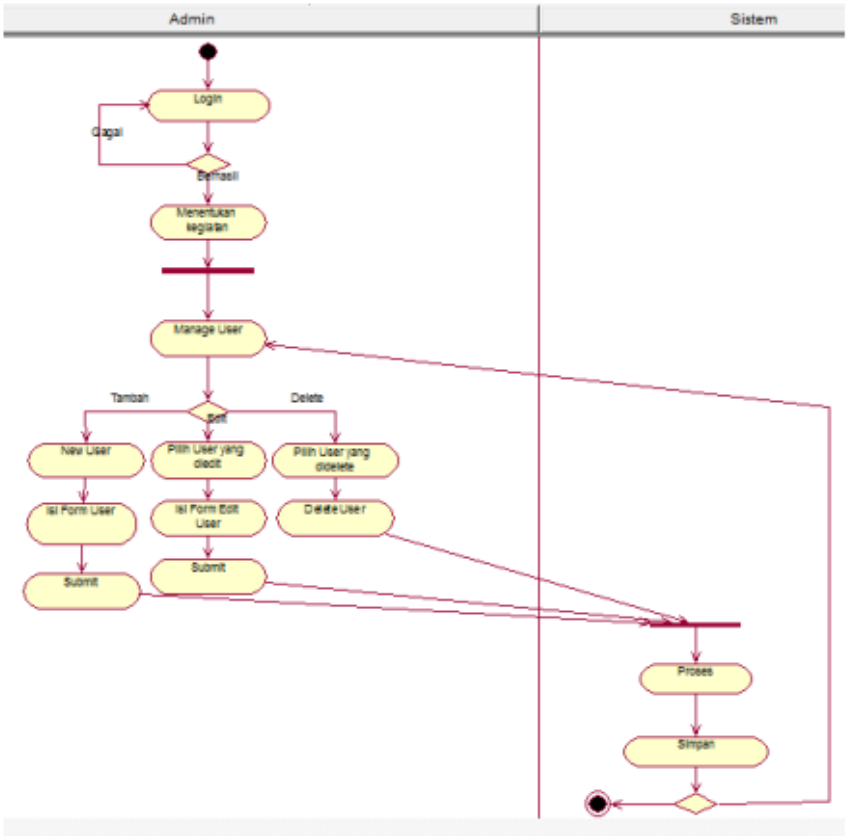
Activity Diagram Manage soal merupakan pemodelan dari aktivitas ke aktivitas lainnya pada proses *Manage* Soal. Gambar 3.18 memperlihatkan *Activity Diagram Manage* Soal.



Gambar 3.18 *Activity Diagram Manage* Soal

3.7.3 *Activity Diagram Manage user*

Activity Diagram Manage user merupakan pemodelan dari aktivitas ke aktivitas lainnya pada proses *Manage user*. Gambar 3.19 memperlihatkan *Activity Diagram Manage user*.



Gambar 3.19 *Activity Diagram Manage User*

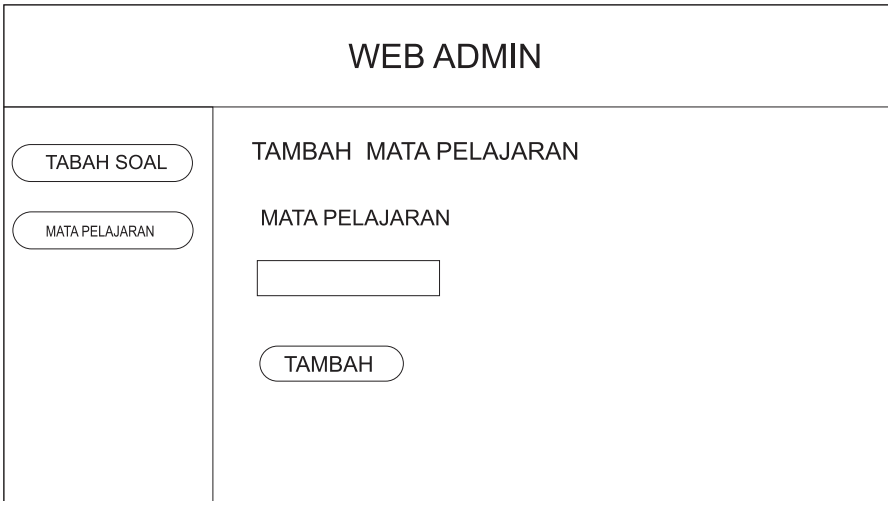
3.8 Perancangan *Interface* (antarmuka)

Interface (antarmuka) pemakai merupakan sarana komunikasi antara *user* dengan sistem yang digunakan atau dengan kata lain antarmuka merupakan keluaran atau *output* dari sistem.

3.8.1 *Interface* admin

3.8.1.1 Menu tambah mata pelajaran

Dalam menu ini *admin* dapat menambahkan mata pelajaran yang akan diujikan, dapat dilihat pada Gambar 3.20 Menu tambah mata pelajaran.



Gambar 3.20 Menu tambah mata pelajaran

Keterangan Fungsi dan fitur pada *web admin*

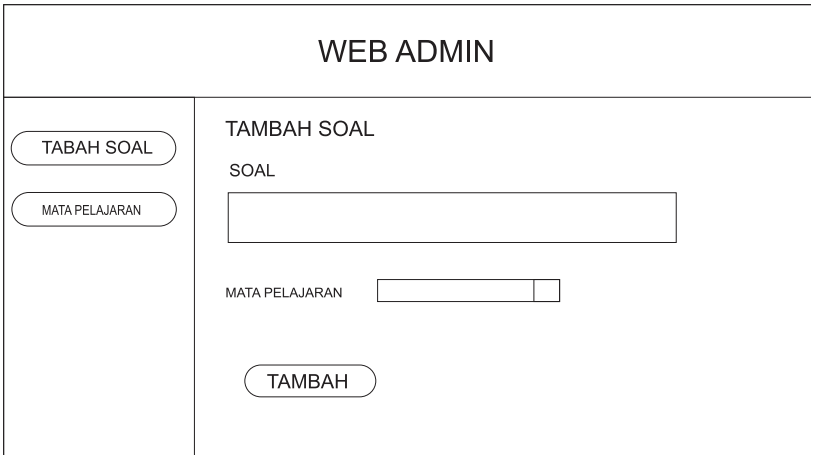
Tabel 3.5 merupakan keterangan fungsi dan fitur web *admin.*

Tabel 3.5 Keterangan fungsi dan fitur web *admin*

No	Nama Fitur	Keterangan
1	Tambah Soal	Penambahan soal pada mata pelajaran
2	Mata Pelajaran	Penambahan Mata Pelajaran

3.8.1.2 Menu tambah soal

Dalam menu ini *admin* dapat menambahkan soal yang akan diujikan, Gambar 3.21 Menu merupakan *interface* tambah soal mata pelajaran.



Gambar 3.21 Menu tambah soal mata pelajaran

Keterangan Fungsi dan fitur penambahan soal pada *web admin*

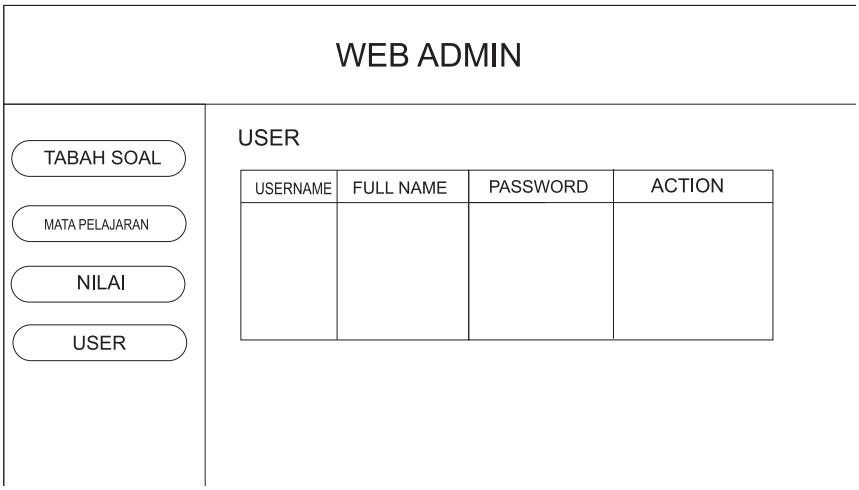
Tabel 3.6 merupakan keterangan fungsi dan fitur web *admin* tambah soal.

Tabel 3.6 Keterangan fungsi dan fitur web *admin* tambah soal

No	Nama Fitur	Keterangan
1	Tambah Soal	Penambahan soal beserta pilihan ganda,pembahasan
2	Button Tambah	Penambahan soal

3.8.1.3 Menu mengelola *user*

Dalam menu ini *Admin* dapat menambah, mengedit, dan menghapus *user*, Gambar 3.22 merupakan Halaman untuk info dari *user*.



Gambar 3.22 Halaman untuk info dari *user*

Keterangan Fungsi dan fitur *manage user* pada *web admin*

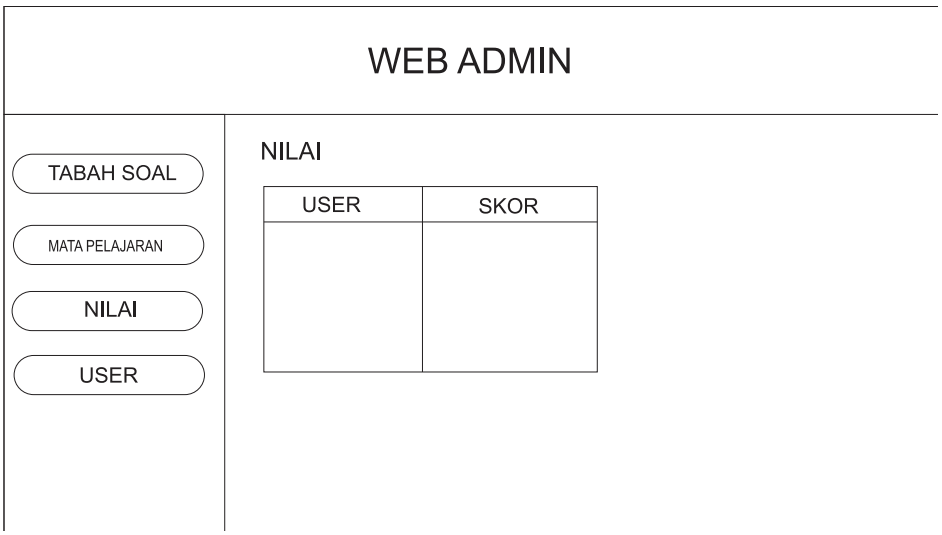
Tabel 3.7 merupakan keterangan fungsi dan fitur *manage user* web *admin*.

Tabel 3.7 Keterangan fungsi dan fitur *manage user* web *admin*

No	Nama Fitur	Keterangan
1	User Name	Nama untuk akun pada aplikasi *Try out*
2	Nama lengkap	Nama lengkap siswa yang terdaftar
3	Password	Kode untuk akun pada aplikasi *Try out*
4	*Action*	Aktivitas yang dilakukan oleh *user* di aplikasi

3.8.1.4 Menu lihat nilai

Pada tampilan ini, *admin* dapat melihat nilai yang diperoleh *user* dari ujian *Try out*, Gambar 3.23 merupakan tampilan untuk nilai *user*.



Gambar 3.23 Tampilan untuk nilai user

Keterangan Fungsi dan fitur *Rangking* pada *web admin*

Tabel 3.8 merupakan keterangan fungsi dan fitur *Rangking* web *admin*.

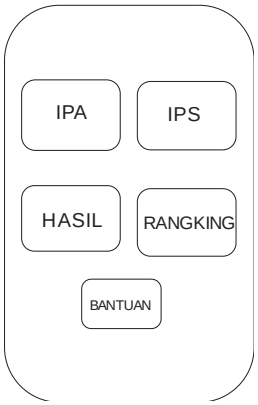
Tabel 3.8 Keterangan fungsi dan fitur *Rangking* web *admin*

No	Nama Fitur	Keterangan
1	Nilai *user*	Nama dari *user* yang sudah mengerjakan soal
2	Skor *user*	Nilai untuk *user* yang sudah mengerjakan soal

3.8.2 *Interface user*

3.3.2.1 Menu utama

Berisi pilihan mata pelajaran dan hasil serta ranking yang dapat dilihat oleh *user*, Gambar 3.24 memperlihatkan Menu utama pada aplikasi.



Gambar 3.24 Menu utama pada aplikasi

3.3.2.2 Menu mata pelajaran

Pada halaman mata pelajaran terdapat beberapa mata pelajaran yang akan diujikan. *User* dapat memilih jurusan IPA atau IPS yang terdapat rincian mata pelajaran, seperti pada Gambar 3.25 memperlihatkan Menu dari mata pelajaran IPA dan pada Gambar 3.26 memperlihatkan Menu dari mata pelajaran IPS.



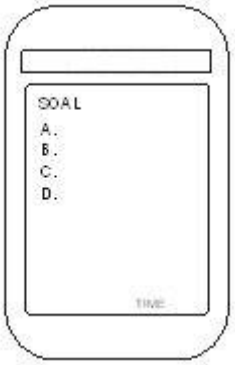
Gambar 3.25 Menu dari mata pelajaran IPA



Gambar 3.26 Menu dari mata pelajaran IPS

3.3.2.3 Menu soal

Pada halaman menu soal berisi soal mata pelajaran yang sudah dipilih dengan tipe soal pilihan ganda, Gambar 3.27 memperlihatkan Tampilan soal pilihan ganda.



Gambar 3.27 Tampilan soal pilihan ganda

3.3.2.4 Menu hasil

Menu hasil akan menampilkan nilai dari beberapa soal yang sudah dikerjakan, Gambar 3.28 memperlihatkan Tampilan dari hasil latihan *Try out*.

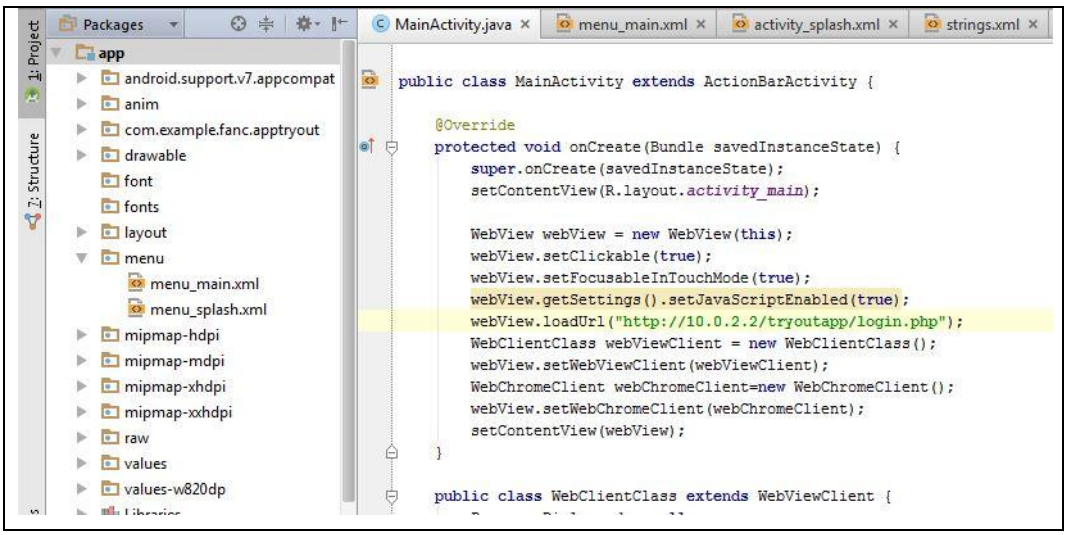


Gambar 3.28 Tampilan dari hasil latihan *Try out*

3.9 Perancangan Aplikasi Android

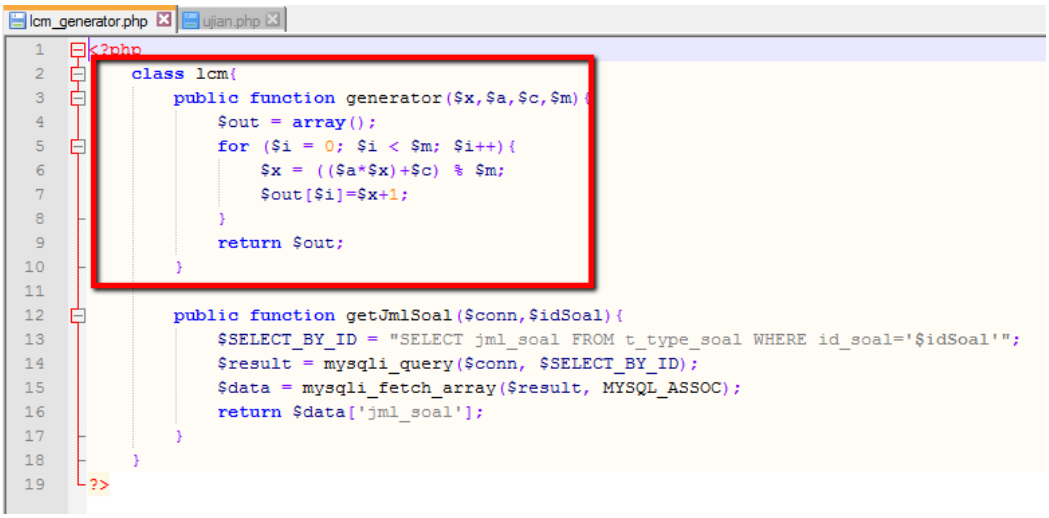
3.9.1 Perancangan Web view Client

WebView adalah sebuah *class* pada Android yang berfungsi sebagai semacam *sandbox* untuk menampilkan dan menjalan aplikasi *mobile* yang berbasiskan web, seperti HTML5, *JQuery Mobile*, dan sebagainya. Gambar 3.29 memperlihatkan Tampilan dari *Web View* pada aplikasi android.



Gambar 3.29 Tampilan dari *Web View* aplikasi android

3.9.2 Penerapan Metode Linear Congruent (LCM)



Gambar 3.30 Tampilan *coding* pengacakan soal

Fungsi generator digunakan untuk membuat Metode *Linear Congruent* yang hasilnya akan dimasukan ke dalam *array*. Gambar 3.30 memperlihatkan Tampilan *coding* pengacakan soal.
$x = bilangan acak sebelum nya

$a = faktor pengali

$c = increment

*$*m = modulus.



Gambar 3.31 Tampilan *coding* dan *looping* pengacakan soal

Cara penerapan metode *Linear Congruent* pada sistem, Gambar 3.31 memperlihatkan Tampilan *coding* dan *looping* pengacakan soal
Proses untuk pengacakan soal, yaitu :

1. Panggil fungsi generator dengan parameter:
   $x = 2

   $a = 1

   $c = 7

   *$*m = $jmlSoal → didapat dari fungsi getJmlSoal yang berada di file lcm_generator.php
2. Buat *looping* untuk menampilkan data soal berdasarkan no_soal = nilai hasil metode lcm.

3.9.3 Penerapan Metode Bubble Short

Pada aplikasi ini, metode berfungsi untuk menginputkan data kedalam variabel angka yang berbentuk *array* dengan perulangan *for*, seperti pada Gambar 3.32 Tampilan *coding bubble short*.

```
Scanner i = new Scanner(System.in);

int[] angka = new int[10];
```

Gambar 3.32 Tampilan *coding bubble short*

```
for(int a = 0; a <= 9; a++){

    System.out.print("Masukan angka ke-"+(a+1)+" : ");

    angka[a] = i.nextInt();

}
```

Gambar 3.33 Tampilan *coding bubble short index*

Gambar 3.33 Tampilan *coding bubble short*, dimana data yang telah di bandingkan pada satu *index* dengan *index* lain akan ditukar jika kondisi bernilai benar.