

BAB III

ANALISA DAN DESAIN SISTEM

III.1. Analisis Masalah

Dalam merancang sebuah aplikasi memahami arti Al-Qur'an Berbasis Android dibutuhkan komponen atau persiapan seperti proses pengumpulan data melalui buku pembelajaran Bahasa Arab, serta perangkat-perangkat aplikasi yang diperlukan. Agar aplikasi ini dapat bekerja sesuai dengan yang diinginkan.

Di dalam skripsi ini penulis merancang sebuah media aplikasi memahami arti Al-Qur'an menggunakan *Android Studio* yang bertujuan untuk membantu masyarakat dalam menghafal dan memahami arti ayat Al-Qur'an. Karena kurangnya kesempatan waktu untuk menghafal menggunakan Al-Qur'an dan buku dapat di lihat bahwasanya masyarakat sekarang lebih sering memegang *Smartphone*.

III.1.1. Analisis Kebutuhan Fungsional

Kebutuhan fungsional adalah jenis kebutuhan yang menjelaskan fungsi-fungsi yang nantinya dapat dilakukan pada aplikasi memahami arti Al-Qur'an yang penulis buat. Adapun kebutuhan fungsional adalah sebagai berikut:

1. Aplikasi dapat menampilkan surah-surah Al-Qur'an.
2. Aplikasi dapat menampilkan pemahaman arti Al-Qur'an dengan metode Tamyiz.
3. Aplikasi memiliki fitur Reminder / pengingat.

III.1.2. Analisis Kebutuhan Non Fungsional

Kebutuhan *Non Fungsional* adalah jenis kebutuhan yang menjelaskan tentang kebutuhan diluar fungsional aplikasi memahami arti Al-Qur'an.

1. Perangkat Keras (*Hardware*)

Spesifikasi *hardware* yang digunakan untuk mengimplementasikan sistem agar dapat berjalan dengan baik adalah sebagai berikut:

- a. *Processor* minimal *Core i3* 2,0 GHz
- b. *Memory* minimal 8 GB
- c. *Hardisk* 250 GB
- d. *Mouse* dan *Speaker* Aktif

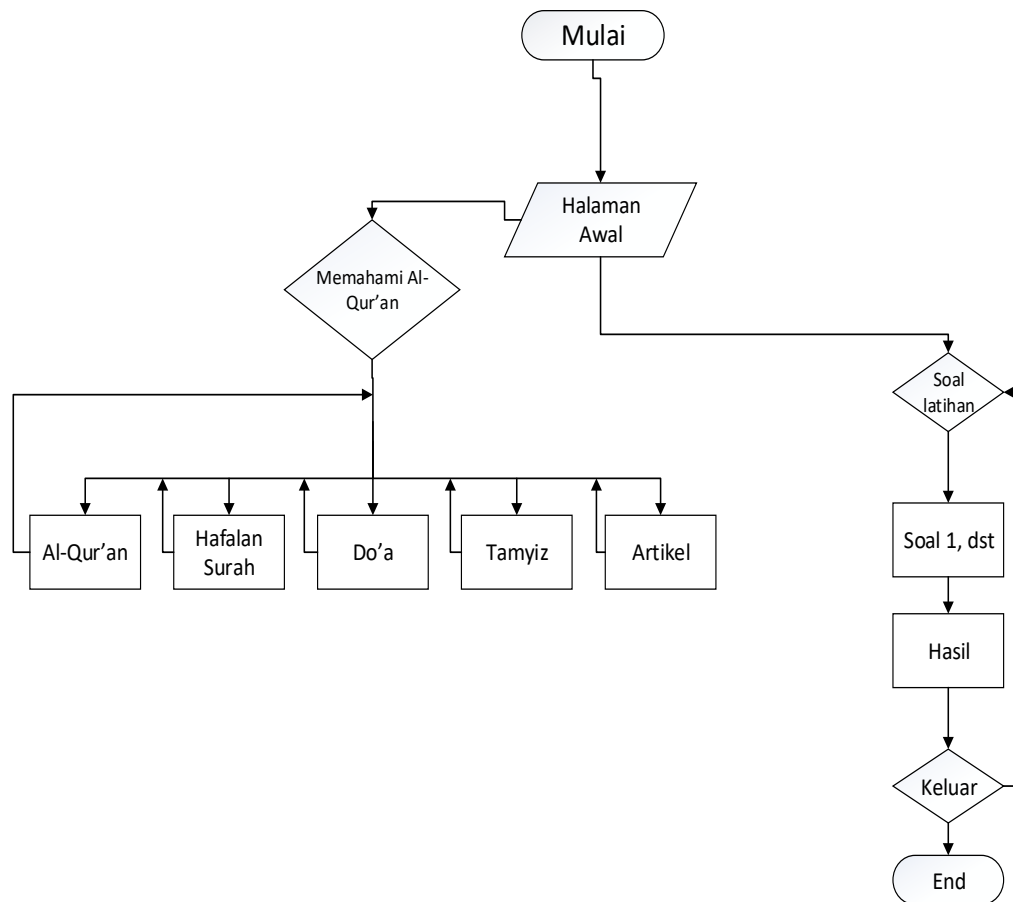
2. Perangkat Lunak (*Software*)

Perangkat lunak (*software*) merupakan suatu komponen didalam suatu sistem data berupa program atau instruksi untuk mengontrol suatu sistem. Dalam hal ini, perangkat lunak yang digunakan penulis untuk aplikasi perancangan adalah:

- a. Sistem Operasi Windows 10
- b. *Android Studio*

III.2 Penerapan Metode *Fisher-Yates*

Simulasi Pengacakan soal adalah suatu proses mengacak soal-soal untuk membentuk paket paket soal. Soal-soal diacak secara random menggunakan algoritma Fisher-Yates. Algoritma Fisher-Yates adalah adalah sebuah algoritma yang menghasilkan permutasi acak dari suatu himpunan terhingga, dengan kata lain untuk mengacak suatu himpunan tersebut. Berikut merupakan *flowchart system* metode fisher yates adalah sebagai berikut.



Gambar III.1. Flowchart

Jika di implementasikan dengan benar maka hasil dari algoritma ini tidak akan berat sebelah sehingga setiap permutasi memiliki kemungkinan yang sama. Langkah-langkah yang digunakan untuk menghasilkan suatu permutasi acak untuk soal 1 sampai N adalah sebagai berikut :

1. Tuliskan soal dari soal no1 sampai soal no N
2. Pilih sebuah soal acak K diantara 1 sampai dengan jumlah soal yang belum dicoret.
3. Dihitung dari bawah, coret soal K yang belum dicoret, dan tuliskan soal tersebut di lain tempat.

4. Ulangi langkah 2 dan langkah 3 sampai semua soal sudah tercoret.
 5. Urutan soal yang dituliskan pada langkah 3 adalah permutasi acak dari soal awal.
- Dari pengumpulan contoh data soal diatas ada 3 atribut yang diambil yaitu no soal yang disimbolkan dengan q1 untuk soal nomor 1 dan seterusnya hingga sampai pada q15 untuk soal nomor 15, urutan soal, urutan pilihan jawaban.

Tahap selanjutnya setelah menentukan atribut dari 15 (lima belas) soal yang akan dijadikan sebagai contoh maka proses yang pertama dilakukan adalah memasukkan atribut soal kedalam scratch (daftar soal yang belum terpilih), Lalu membuat range (jumlah soal yang belum terpilih) kemudian dilakukan proses pengacakan, Selanjutnya melihatkan roll (untuk sebuah soal yang teripilih dari semua jumlah soal yang ada) kemudian hasil soal yang sudah terpilih dimasukkan kedalam result (hasil dari seluruh soal yang telah dilakukan pengacakan) Proses algoritma Fisher-Yates dalam pengacakan sebanyak 15 (lima belas) buah soal yang dicontohkan dapat digambarkan pada tabel III.1 sebagai berikut :

Tabel III.1

Range	Roll	Scratch	Result
		1,2,3,4,5,6,7,8,9,10,11,12,13,14,15	
1-15	6	1,2,3,4,5,7,8,9,10,11,12,13,14,15	6
1-14	10	1,2,3,4,5,7,8,9,10,12,13,14,15	11,6
1-13	3	1,2,4,5,7,8,9,10,12,13,14,15	3,11,6
1-12	1	2,4,5,7,8,9,10,12,13,14,15	1,3,11,6
1-11	6	2,4,5,7,8,10,12,13,14,15	9,1,3,11,6
1-10	4	2,4,5,8,10,12,13,14,15	7,9,1,3,11,6
1-9	7	2,4,5,8,10,12,14,15	13,7,9,1,3,11,6
1-8	7	2,4,5,8,10,12,15	14,13,7,9,1,3,11,6

1-7	2	2,5,8,10,12,15	4,14,13,7,9,1,3,11,6
1-6	4	2,5,8,12,15	10,4,14,13,7,9,1,3,11,6
1-5	5	2,5,8,12	15,10,4,14,13,7,9,1,3,11,6
1-4	4	2,5,8	12,15,10,4,14,13,7,9,1,3,11,6
1-3	3	2,5	8,12,15,10,4,14,13,7,9,1,3,11,6
1-2	2	2	5,8,12,15,10,4,14,13,7,9,1,3,11,6
			2,5,8,12,15,10,4,14,13,7,9,1,3,11,6
			1,3,11,6

III.2.1 Pseudocode Metode Pengacakan *Fisher-Yates*

Untuk mengetahui bagaimana Pseudocodenya algoritma Fisher-Yates dapat dilihat pada tabel III.2.

Tabel III.2.Pseudocode Algoritma Fisher-Yates

No	Algoritma	Pseudocode
1	Nama Fungsi	function fyAcak(\$array)
2	Jumlah Array	\$i = count(\$array);
3	Perintah perulangan sepanjang jumlah array	while(--\$i)
4	Membangkitkan bilangan random	\$j = mt_rand(0, \$i);
5	Membandingkan nilai itidak sama dengan j	if (\$i != \$j)
6	Simpan nilai j ke tmp	\$tmp = \$array[\$j];
7	Masukan nilai i ke j	\$array[\$j] = \$array[\$i];
8	Masukan nilai tmp ke array i	\$array[\$i] = \$tmp;
9	Nilai return	return \$array;

III.3. Desain Sistem

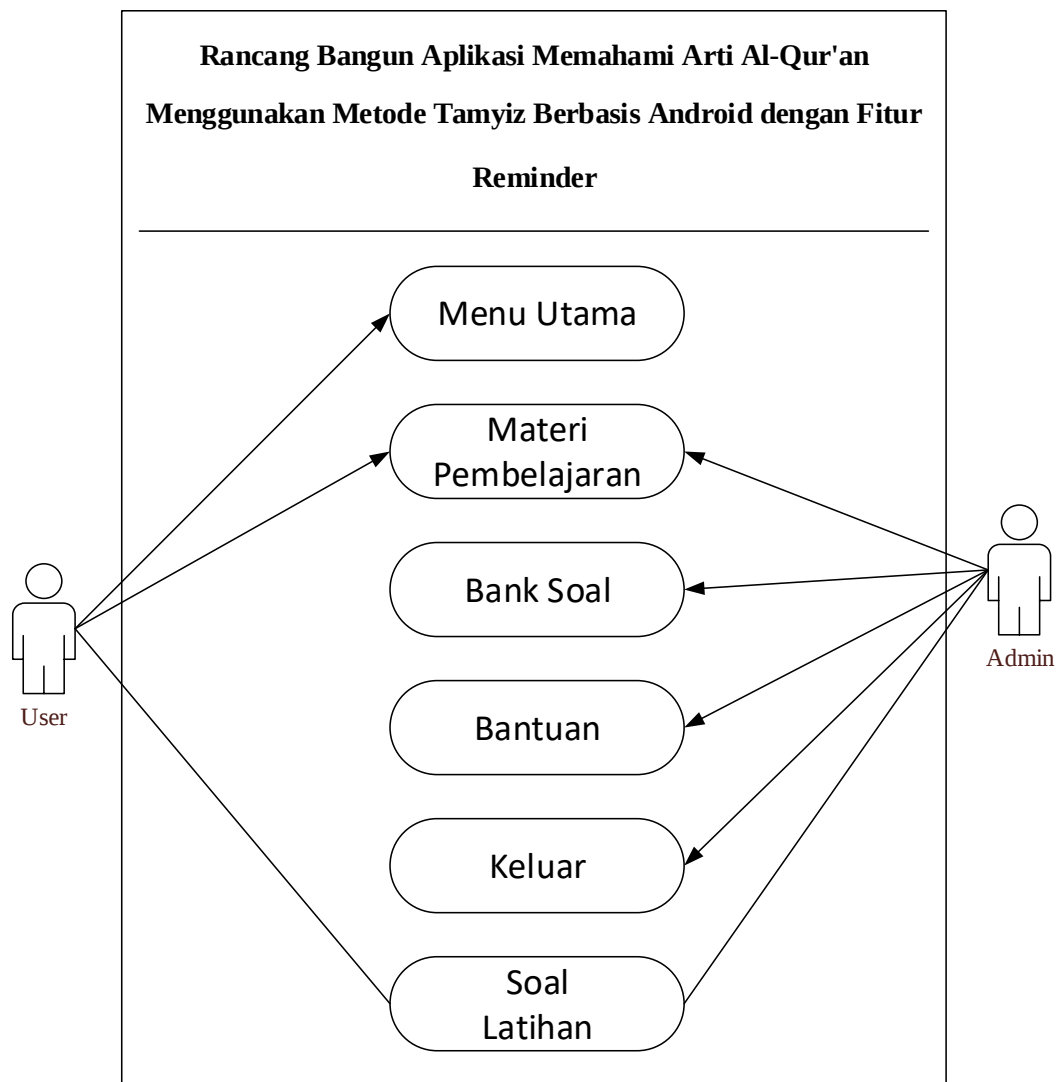
Desain sistem merupakan gambaran perancangan yang akan dilakukan dan yang dihasilkan. Desain sistem dibutuhkan sebagai gambaran langkah-langkah

desain dan bagian-bagian yang dibutuhkan agar aplikasi dapat berjalan sesuai perancangan.

Perancangan yang digunakan untuk merancang Aplikasi memahami arti Al-Qur'an dengan metode Tamyiz berbasis android dengan fitur Reminder adalah metode perancangan terstruktur menggunakan *Unified Modeling Language* (UML). *Unified Modeling Language* (UML) pada dasarnya merupakan konsep perancangan untuk mengetahui alurnya aplikasi yang akan dibuat oleh penulis. *Unified Modeling Language* (UML) yang digunakan meliputi perancangan *Use Case Diagram*, *Story Board*, *Sequence Diagram*, *Activity Diagram*, dan *Flowchart*.

III.3.1. *Use Case Diagram*

Use case menjelaskan urutan kegiatan yang dilakukan aktor dan sistem untuk mencapai suatu tujuan tertentu. Sebuah *usecase* mempresentasikan sebuah interaksi antara aktor dengan sistem dan menggambarkan fungsional yang diharapkan dari sebuah sistem perbandingan. Diagram *usecase* tersebut dapat dilihat pada gambar III.1.



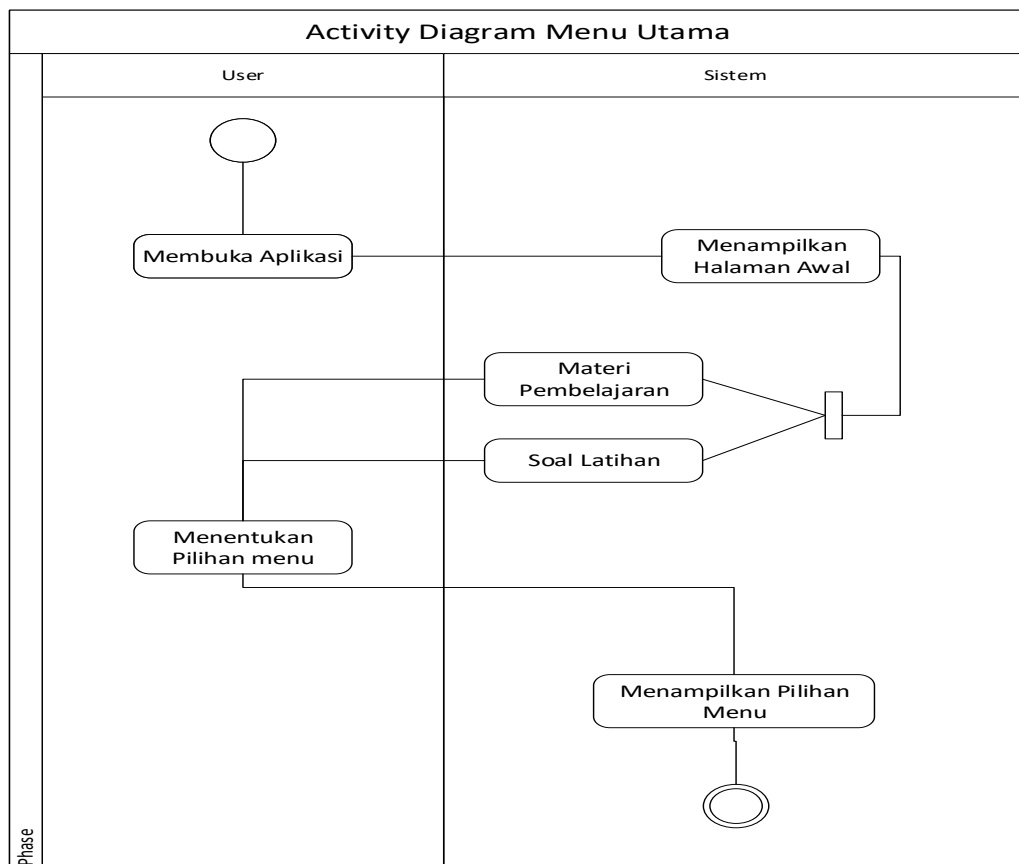
Gambar III.2. Use Case Diagram

III.3.2 . Activity Diagram

Activity Diagram merupakan gambaran berbagai aliran aktivitas yang terjadi dalam aplikasi yang sedang dirancang.

1. Activity Diagram Menu Utama User

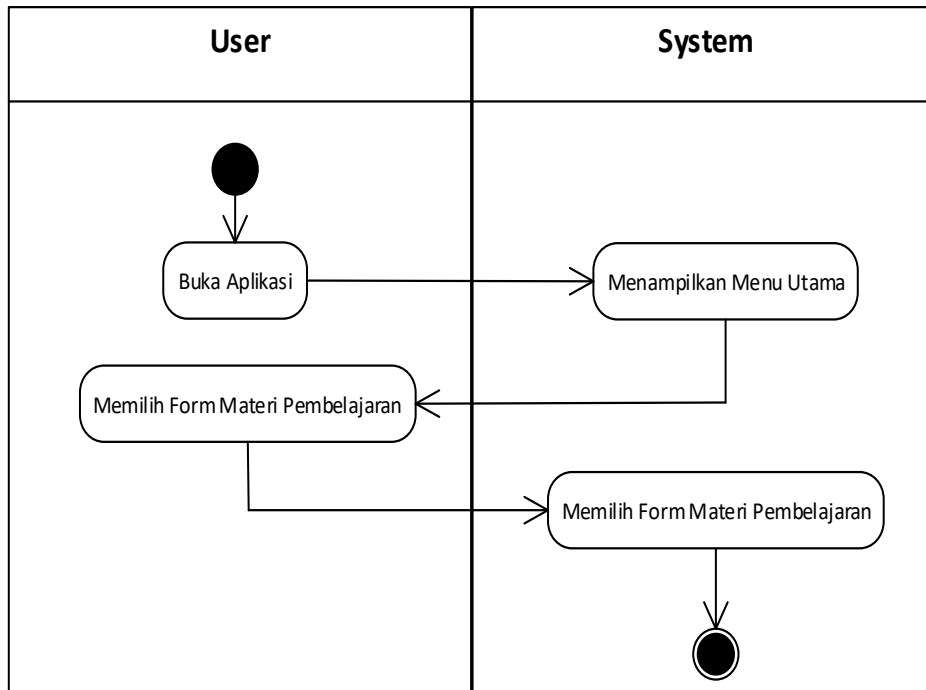
Activity diagram menu utama merupakan activity diagram saat memilih menu utama pada aplikasi. Activity diagram menu utama ditunjukkan pada gambar III.3.



Gambar III.3. Activity Diagram Menu Utama

2. Activity Diagram Materi Pembelajaran User

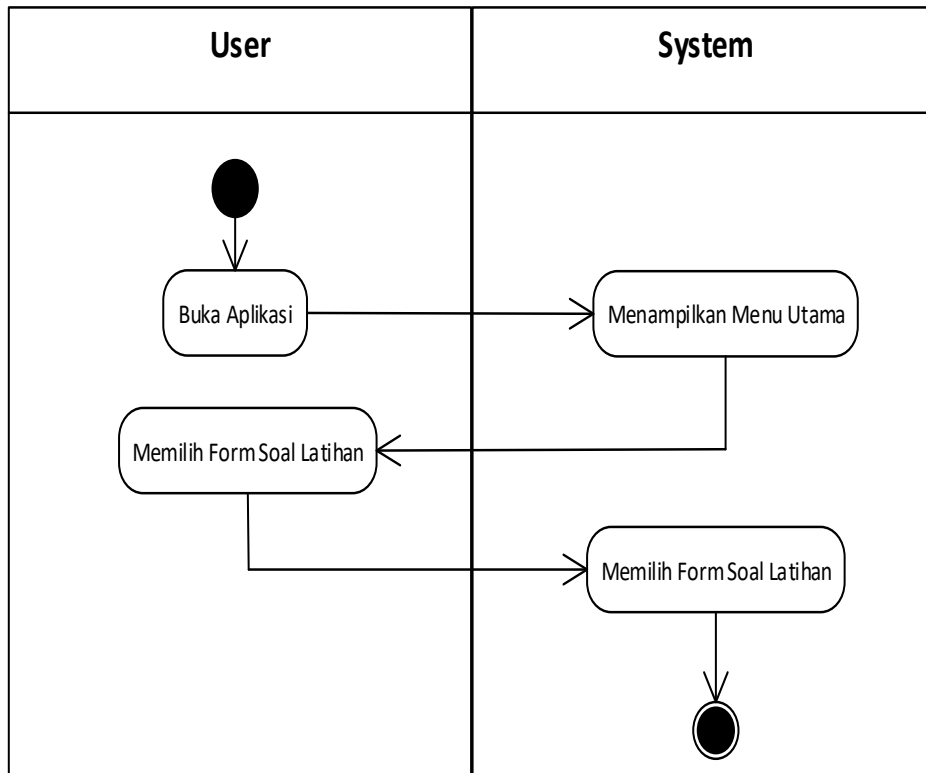
Activity diagram materi pembelajaran merupakan activity diagram yang dapat dikelola oleh user. Activity diagram materi pembelajaran ditunjukkan pada gambar III.4.



Gambar III. 4 Activity Diagram Materi Pembelajaran User

3. Activity Diagram Soal Latihan

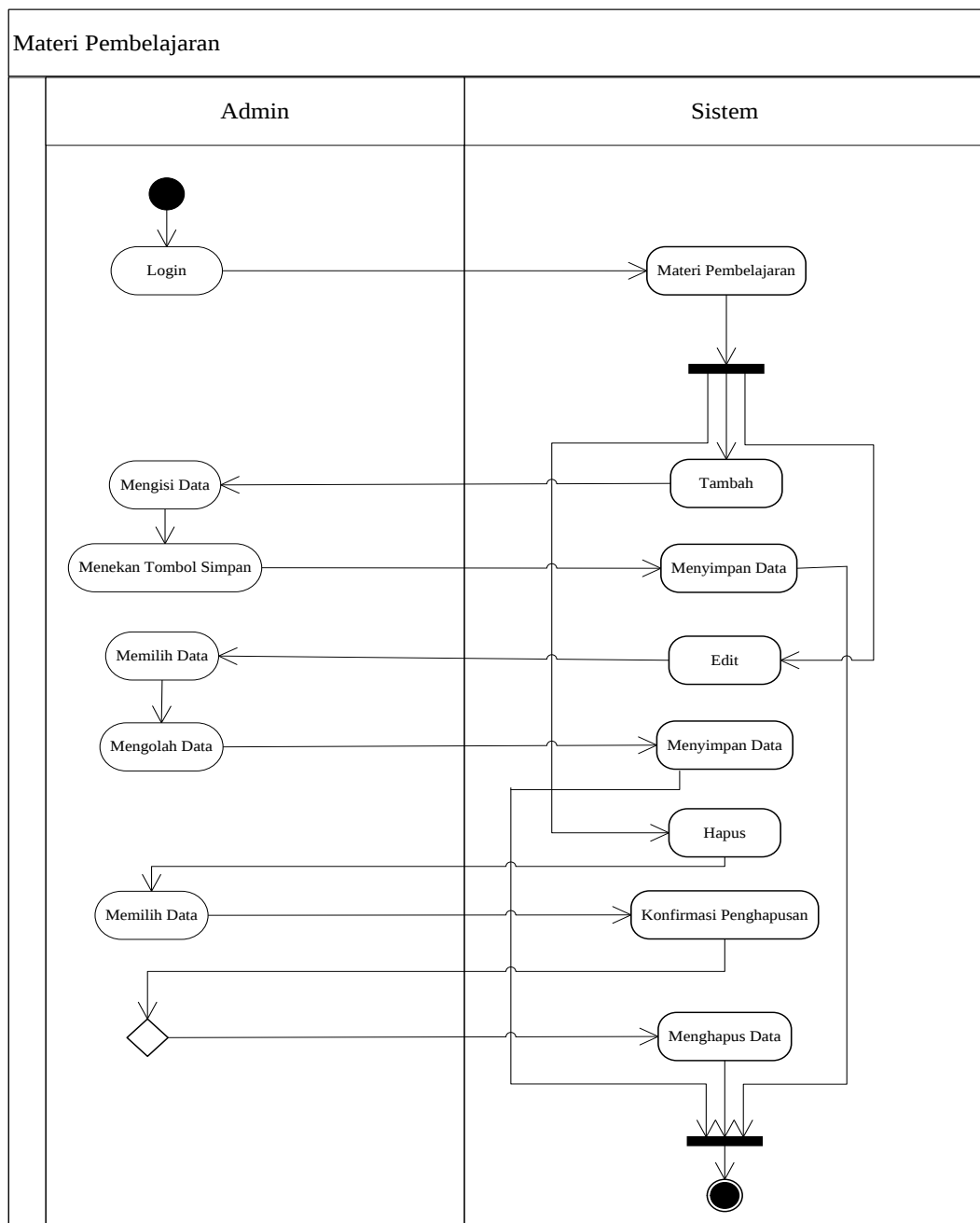
Activity diagram soal latihan merupakan *activity diagram* yang dapat dikelola oleh user. *Activity diagram* materi pembelajaran ditunjukkan pada gambar III.5.



Gambar III.5 Activity Diagram Soal Latihan

4. Activity Diagram Materi Pembelajaran Admin

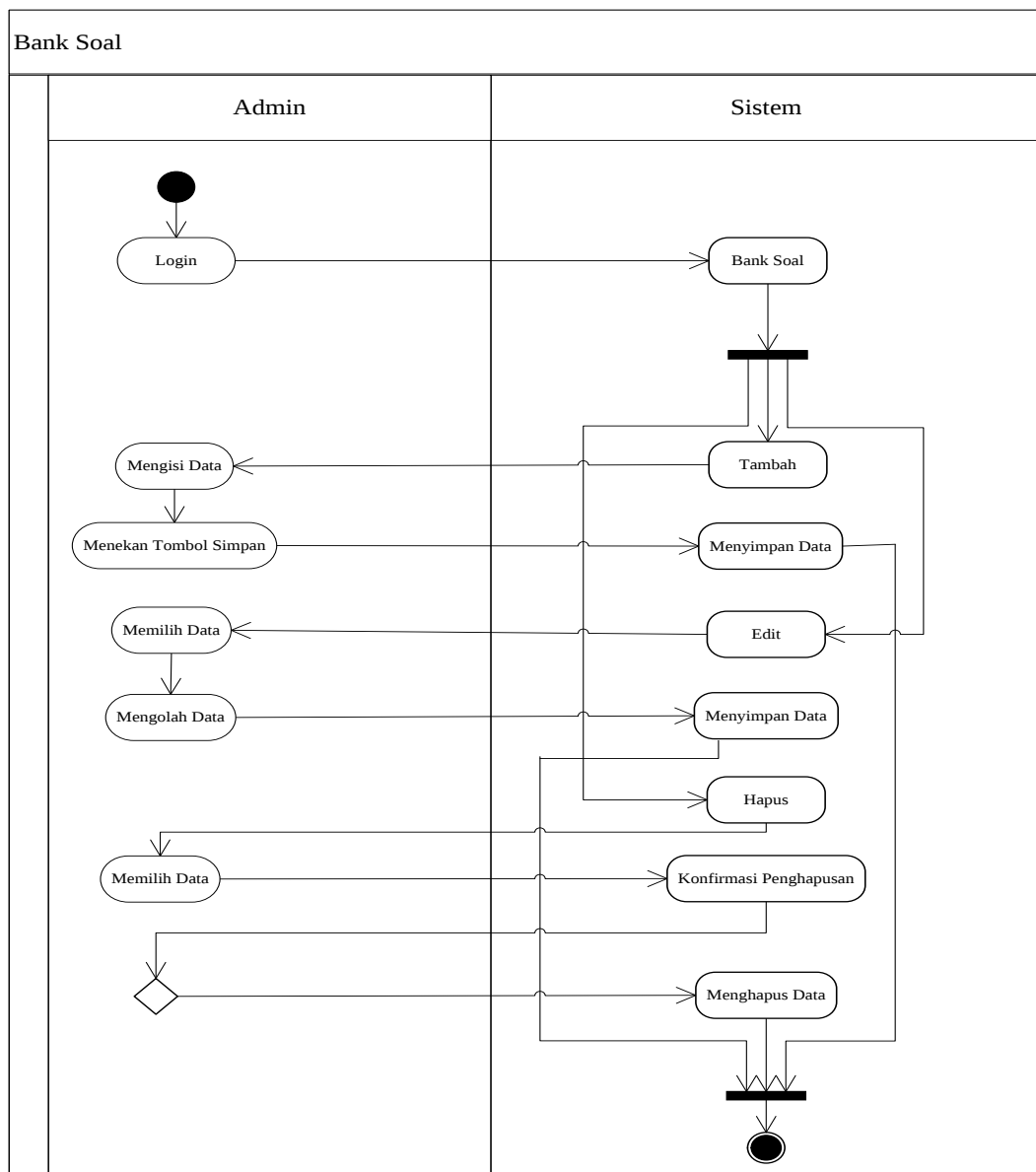
Activity diagram materi pembelajaran merupakan *activity diagram* yang dapat dikelola oleh admin. *Activity diagram* materi pembelajaran ditunjukkan pada gambar III.6.



Gambar III.6 Activity Diagram Materi Pembelajaran Admin

5. Activity Diagram Bank Soal Admin

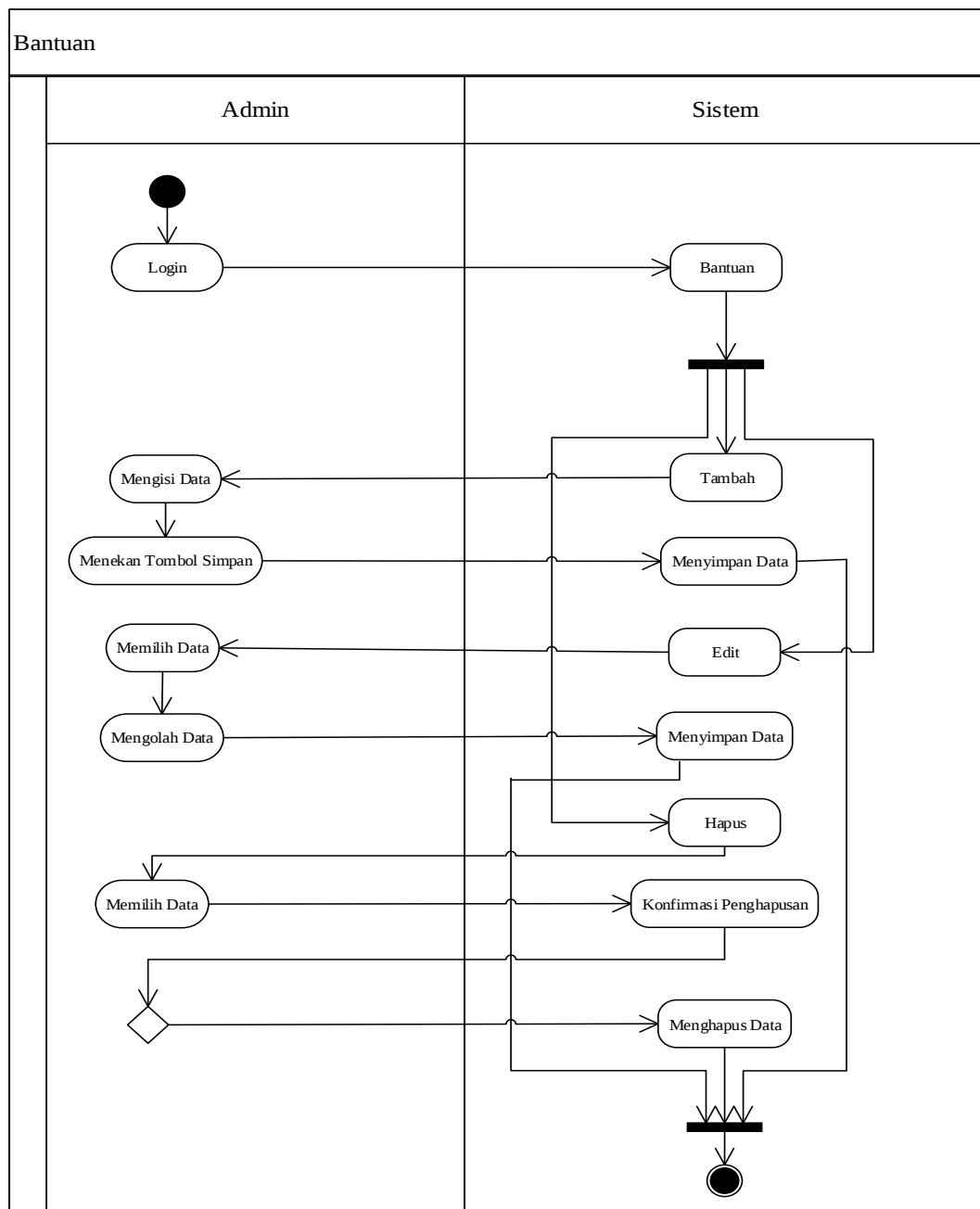
Activity diagram Bank Soal merupakan activity diagram yang dapat dikelola oleh admin. Activity diagram Bank Soal ditunjukkan pada gambar III.7.



Gambar III.7 Activity Diagram Bank Soal Admin

6. Activity Diagram Bantuan Admin

Aktivitas bantuan yang dilakukan oleh admin untuk menambahkan bantuan penggunaan aplikasi dapat diterangkan dengan langkah-langkah *state* berikut :



Gambar III.8. Activity Diagram Bantuan Admin

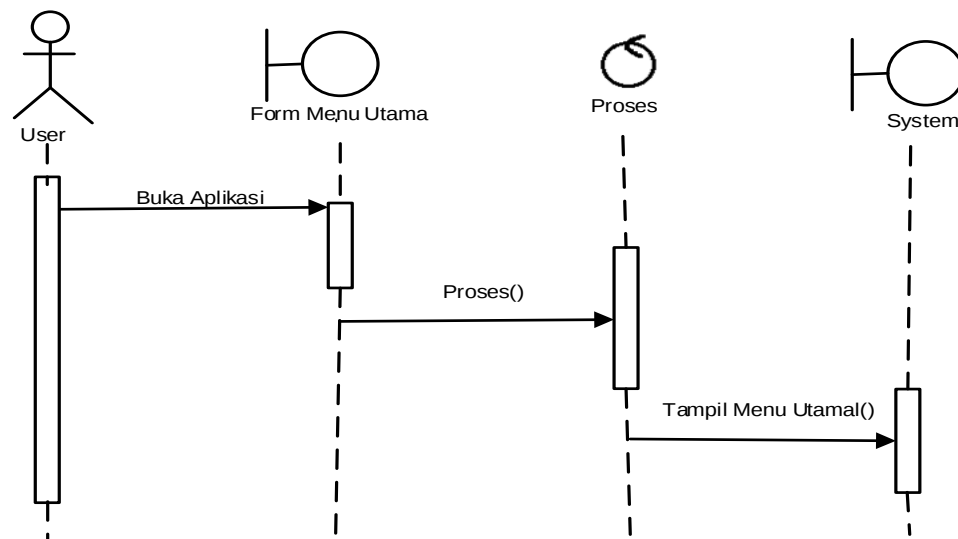
III.3.3. Sequence Diagram

Sequence Diagram menunjukkan interaksi objek yang diatur dalam urutan waktu. Diagram ini menggambarkan objek dan kelas yang terlibat dalam skenario dan urutan pesan yang dipertukarkan antara objek yang diperlukan untuk

melaksanakan fungsi skenario. Adapun beberapa *Sequence Diagram* pada aplikasi yang akan dibangun yaitu:

1. *Sequence Diagram* Menu Utama User

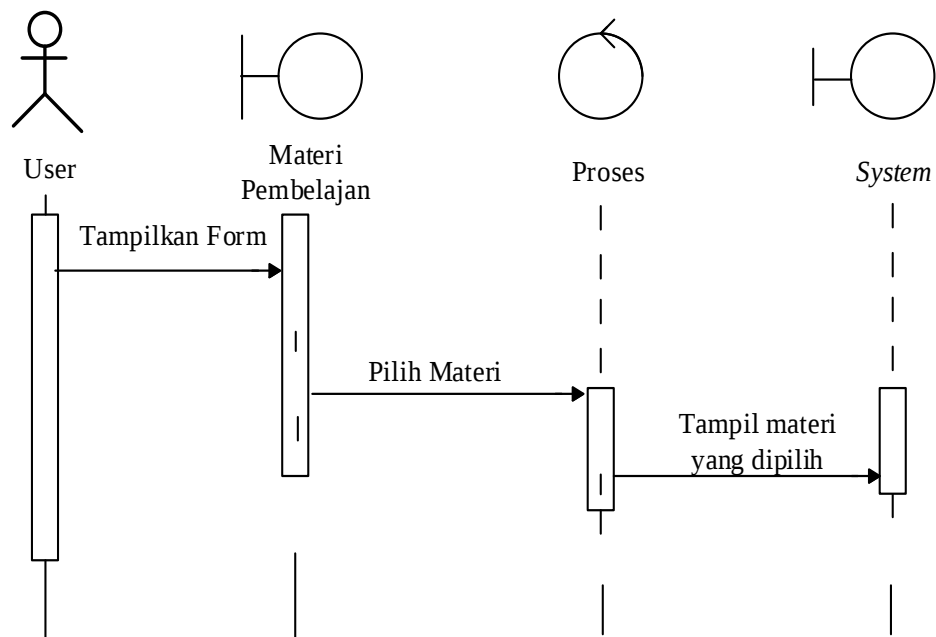
Serangkaian kegiatan *tampilan* menu utama yang dilakukan oleh *user* dapat memilih form telah disediakan oleh *system*, diterangkan dengan langkah-langkah *state* berikut.



Gambar III.9. *Sequence Diagram* Menu Utama User

2. *Sequence Diagram* Materi Pembelajaran User

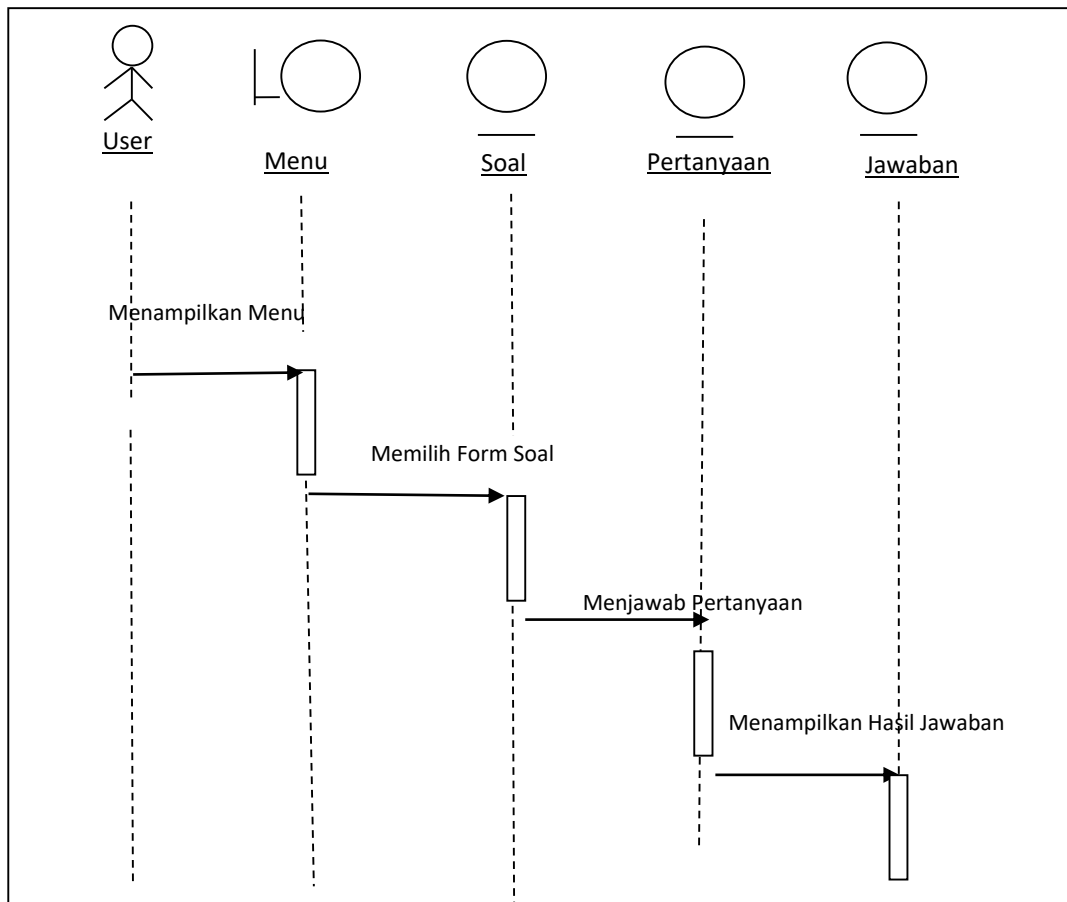
Serangkaian kegiatan Materi Pembelajaran yang dilakukan oleh *user* dapat diterangkan dengan langkah-langkah *state* berikut.



Gambar III.10. Sequence Diagram Materi Pembelajaran User

3. Sequence Diagram Soal Latihan User

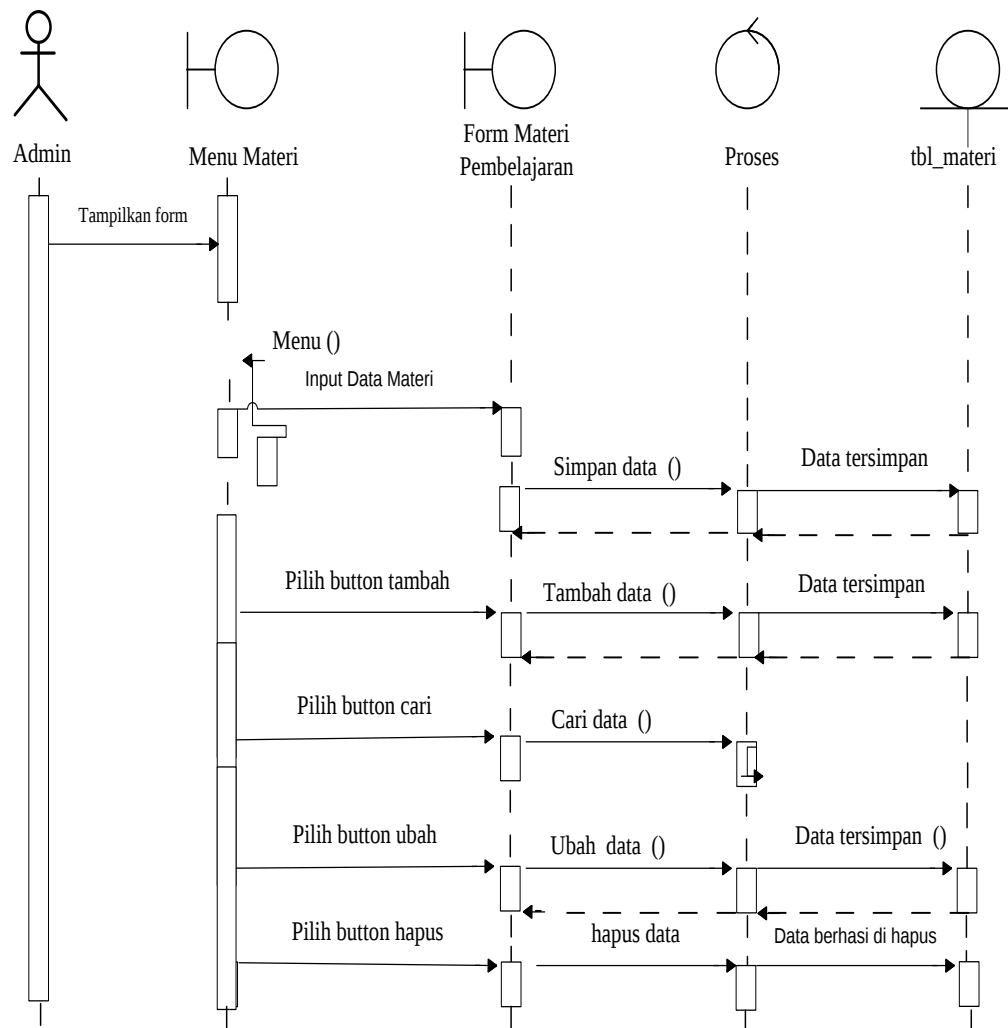
Activity Digram yang dilakukan oleh user pada saat menampilkan soal latihan serta menjawab soal dan mendapatkan hasil dari soal yang dijawab, dapat diterangkan dengan langkah-langkah *state* berikut.



Gambar III.11. Sequence Diagram Latihan Soal User

4. Sequence Diagram Materi Pembelajaran Admin

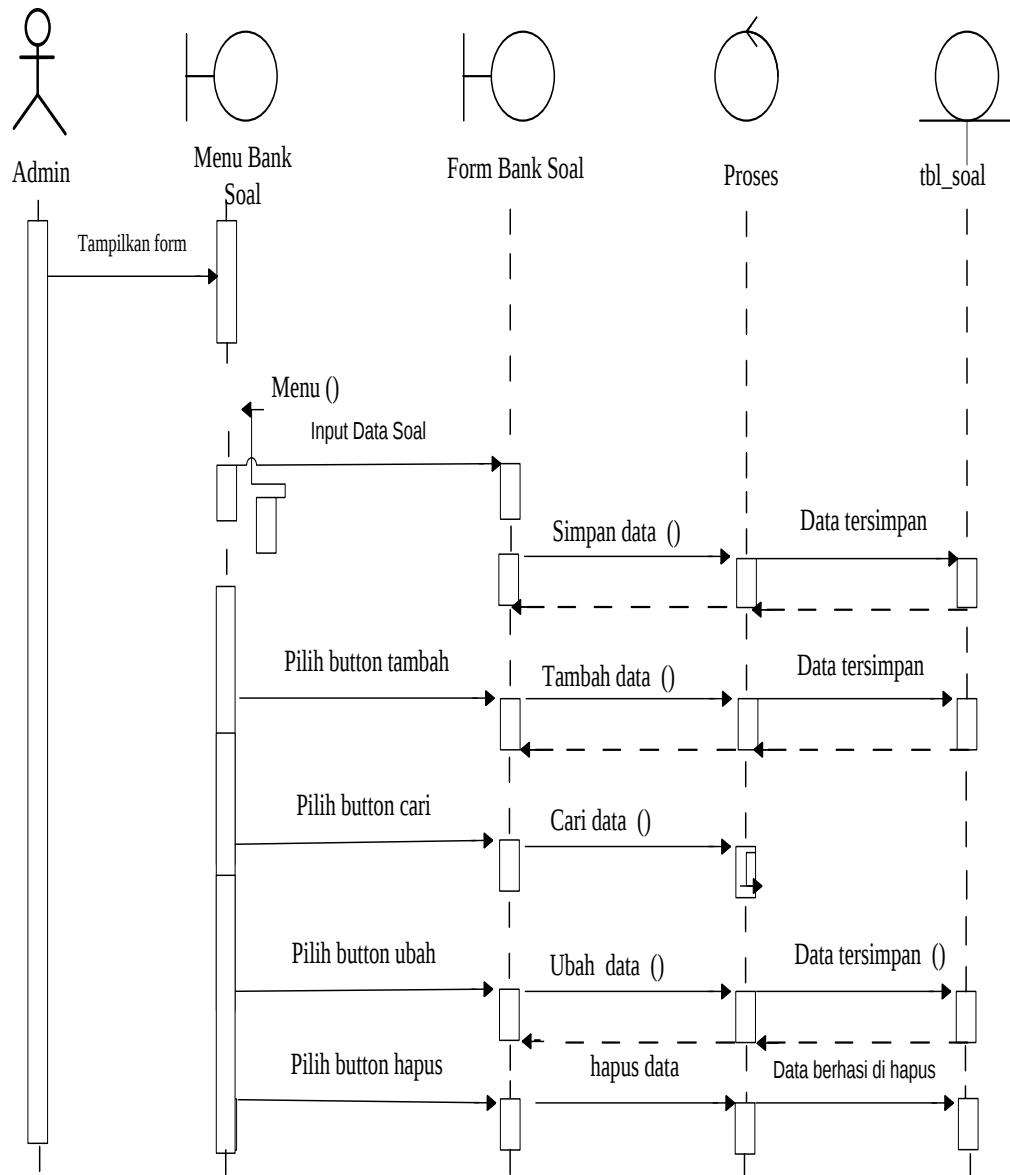
Serangkaian kegiatan Materi Pembelajaran yang dilakukan oleh admin dapat diterangkan dengan langkah-langkah *state* berikut.



Gambar III.12. Sequence Diagram Materi Pembelajaran Admin

5. Sequence Diagram Bank Soal Admin

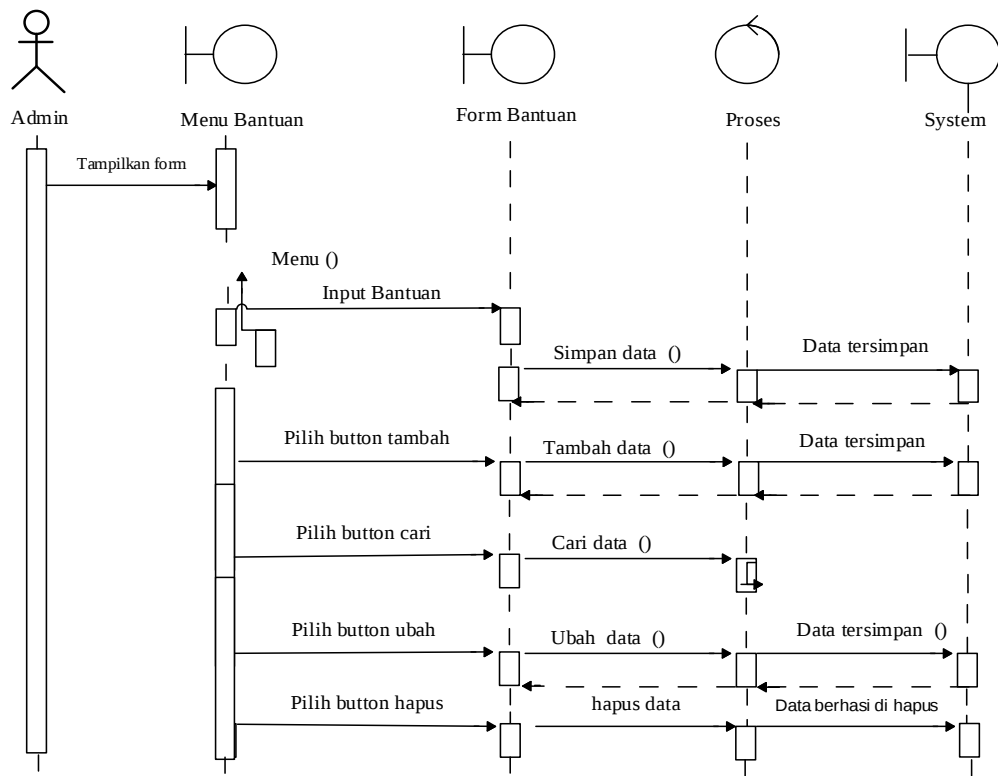
Serangkaian kegiatan Bank Soal yang dilakukan oleh admin dapat diterangkan dengan langkah-langkah *state* berikut.



Gambar III.13. Sequence Diagram Bank Soal Admin

6. Sequence Diagram Bantuan Admin

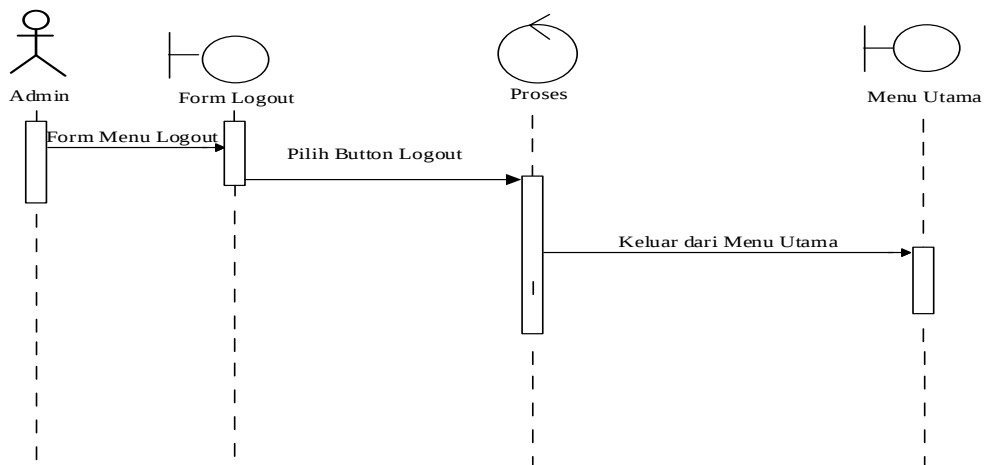
Serangkaian kegiatan bantuan yang dilakukan oleh admin dapat diterangkan dengan langkah-langkah *state* berikut :



Gambar III.14. Sequence Diagram Bantuan Admin

7. Sequence Diagram Keluar

Serangkaian kegiatan keluar yang dilakukan oleh *admin* dapat diterangkan dengan langkah-langkah *state* berikut.



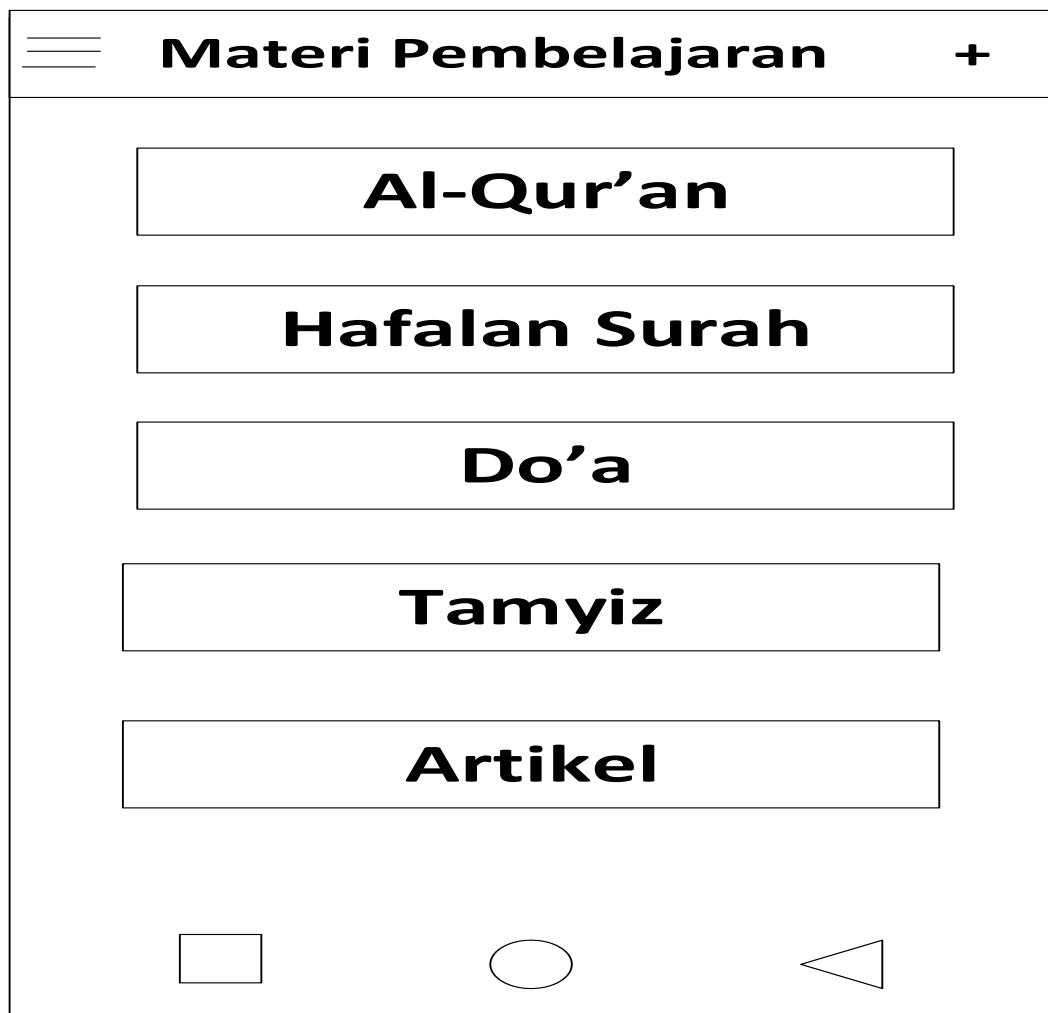
Gambar III.15. Sequence Diagram Keluar

III.4. Desain *User Interface*

Perancangan tampilan aplikasi dalam program ini sangat diperlukan karena ini merupakan bentuk tampilan aplikasi yang akan dirancang. Didalam aplikasi ini terdapat *scene* yang dirancang yaitu sebagai berikut.

1. Rancangan Tampilan Materi Pembelajaran Admin

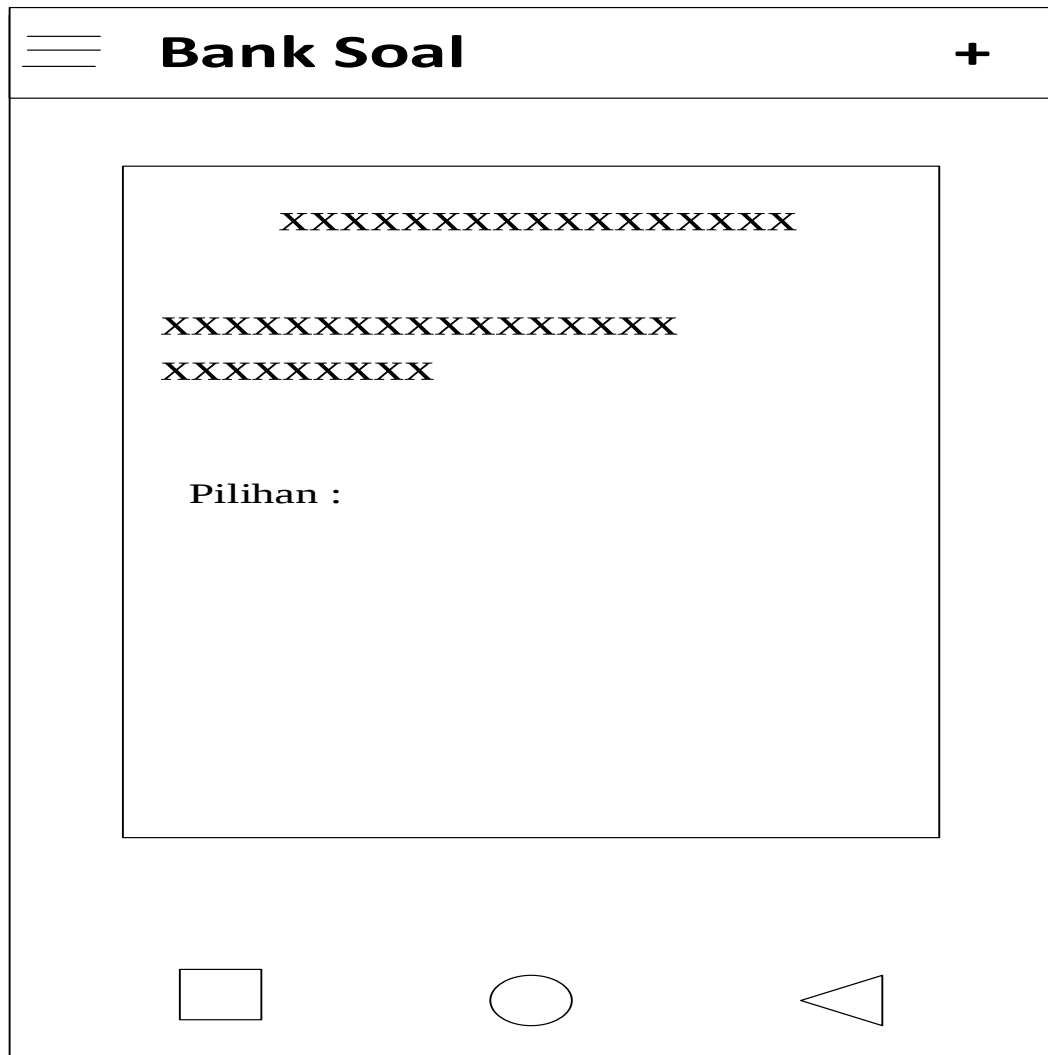
Tampilan materi pembelajaran merupakan tampilan saat aplikasi dijalankan dapat dilihat pada gambar III.16.



Gambar III.16 Rancangan Materi Pembelajaran Admin

2. Rancangan Tampilan Bank Soal

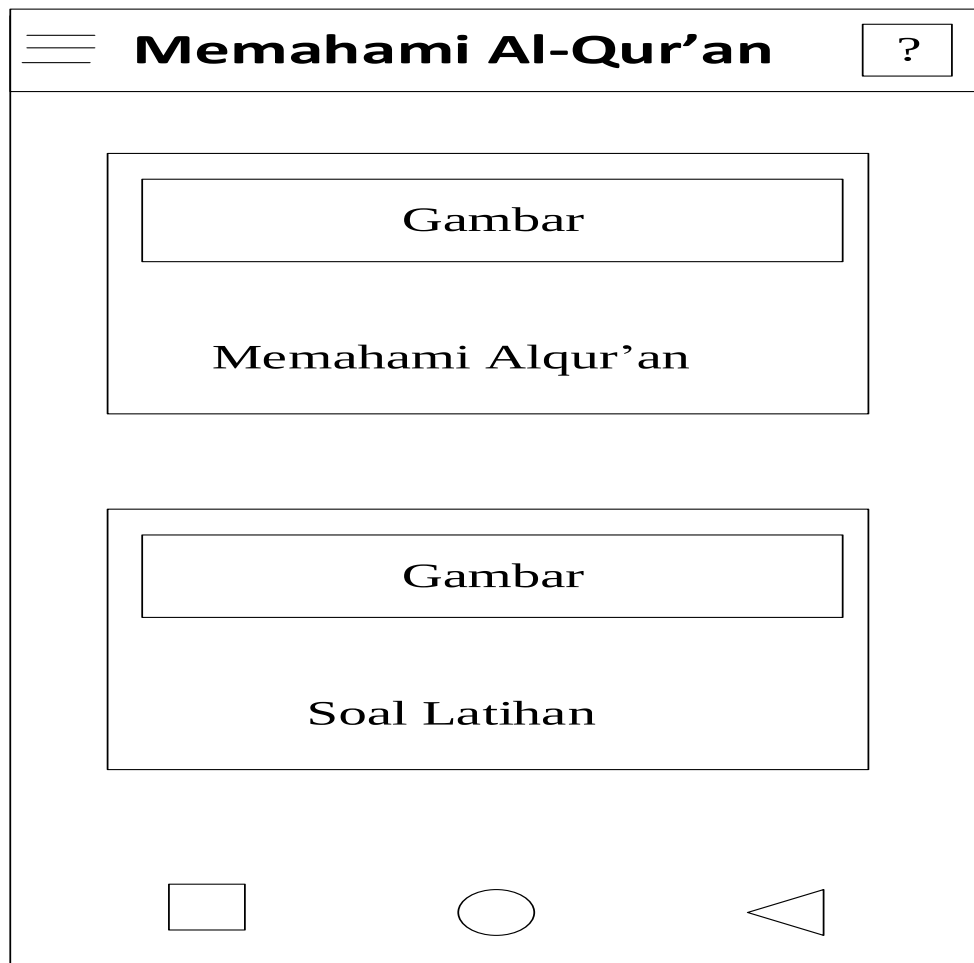
Tampilan dari bank soal merupakan tampilan saat aplikasi dijalankan dapat dilihat pada gambar III.17.



Gambar III.17 Rancangan Bank Soal

3. Rancangan Tampilan Bantuan

Tampilan dari bantuan merupakan tampilan saat aplikasi dijalankan dapat dilihat pada gambar III.18.



Gambar III.19 Rancangan Halaman Utama User

5. Rancangan Tampilan Materi Pembelajaran User

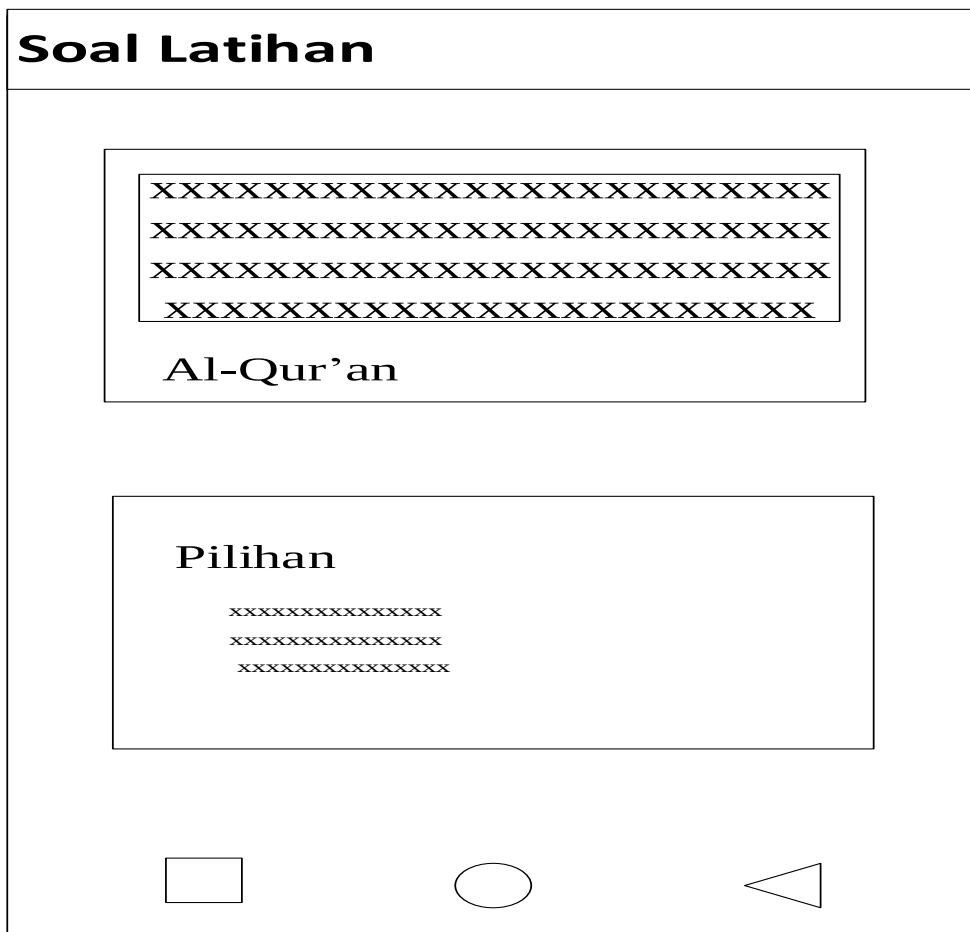
Tampilan dari materi pembelajaran user merupakan tampilan saat aplikasi dijalankan dapat dilihat pada gambar III.20.



Gambar III.20 Rancangan Halaman Materi Pembelajaran User

6. Rancangan Tampilan Soal Latihan User

Tampilan dari soal latihan user merupakan tampilan saat aplikasi dijalankan dapat dilihat pada gambar III.21.



Gambar III.21 Rancangan Halaman Soal Latihan User

7. Rancangan Tampilan Bantuan

Tampilan dari bantuan merupakan tampilan saat aplikasi dijalankan dapat dilihat pada gambar III.22.



Gambar III.22 Rancangan Bantuan