

## APLIKASI PEMBELAJARAN RAMBU-RAMBU LALU LINTAS MENGUNAKAN METODE *FINITE STATE MACHINE* (FSM) BERBASIS WEB

Dahlan Abdullah<sup>1</sup>, Erlina<sup>2</sup>, Cut Ita Erliana<sup>3</sup>, Muhammad Zarlis<sup>4</sup>, Elviwani<sup>5</sup>, Hartono<sup>6</sup>

<sup>1,2</sup> Program Studi Teknik Informatika, Universitas Malikussaleh, Aceh, Indonesia

<sup>3</sup> Program Studi Teknik Industri, Universitas Malikussaleh, Aceh, Indonesia

<sup>4,5</sup> Program Studi Ilmu Komputer, Universitas Sumatera Utara, Medan, Indonesia

<sup>6</sup> Program Studi Ilmu Komputer, STMIK IBBI, Medan, Indonesia

<sup>1</sup>dahlan@unimal.ac.id, <sup>3</sup>cutitha@unimal.ac.id, <sup>4</sup>mzarlis2000@yahoo.com, <sup>5</sup>elviwani@usu.ac.id,

<sup>6</sup>hartonoibbi@gmail.com,

### ABSTRAK

Banyak kecelakaan yang terjadi saat ini disebabkan oleh kurangnya pemahaman masyarakat mengenai rambu-rambu lalu lintas. Rambu lalu lintas merupakan salah satu perlengkapan jalan yaitu lambang, huruf, angka kalimat atau perpaduan antaranya. Rambu sangat penting bagi pemakai jalan sebagai alat untuk petunjuk dan peringatan pada pengguna di jalan. Untuk itu perlu diberikan pemahaman kepada masyarakat tersebut. Cara yang paling mudah adalah dengan memberikan pembelajaran berbentuk *game* tentang pembelajaran rambu-rambu lalu lintas. Kegiatan Penelitian ini mengembangkan *game* pembelajaran Rambu Lalu Lintas Berbasis Web dengan menggunakan bahasa pemrograman PHP dan MySQL sebagai basis data. Dan menerapkan metode *Finite State Machine* untuk proses alur kontrol permainan yang terdiri dari 4 level, dimana *player* akan melengkapi jawaban dari setiap *level* agar dapat melanjutkan ke *level* berikutnya. Hasil dari *game* ini dapat memberikan pengetahuan tentang rambu-rambu lalu lintas yang lebih menarik untuk dipelajari dan menyadari pentingnya menaati rambu-rambu lalu lintas.

**Kata kunci** : PHP, MySQL, *Finite State Machine*.

## 1. PENDAHULUAN

### 1.1 Latar Belakang

Rambu lalu lintas adalah salah satu dari perlengkapan jalan, berupa lambang, huruf, angka kalimat atau perpaduan. Diantaranya rambu peringatan, rambu perintah, rambu larangan, rambu petunjuk bagi pemakai jalan. Rambu lalu lintas digunakan sebagai pengguna di jalan yang baik. Rambu sangat penting bagi pemakai jalan sebagai alat untuk petunjuk dan peringatan pada pengguna di jalan.

Rambu lalu lintas ini memberi definisi kepada masyarakat bagaimana isi, dan arti dari simbol rambu lalu lintas dan peraturan yang sudah di tetapkan dalam undang-undang lalu lintas. Akan tetapi sebagaimana masyarakat di jalan banyak yang melanggar rambu lalu lintas. Hal ini karena kurangnya kesadaran masyarakat untuk menaati peraturan lalu lintas. Tingkat kepatuhan secara stereotip juga berbeda-beda. Bukan hanya di sebabkan oleh faktor lingkungan dan budaya, akan tetapi juga oleh karena pendidikan yang berbeda. Dari tingkat kesadaran hukum yang relatif rendah menyebabkan kewajiban hukum dan perundang-undang di bidang lalu lintas. (Irwan Andriyanto : 2016).

Menyadari akan pentingnya kegunaan sebuah rambu lalu lintas. Maka dari itu peneliti akan membangun sebuah aplikasi pembelajaran rambu-rambu lalu lintas. Yang memudahkan

masyarakat untuk mengetahui rambu-rambu lalu lintas berbasis web.

Pembelajaran rambu-rambu lalu lintas berbasis web ini bisa diakses oleh semua kalangan sehingga nantinya lebih mengenal dan memahami rambu-rambu lalu lintas saat berkendara. Aplikasi ini dibuat menggunakan metode *Finite State Machine* (FSM) untuk proses alur control permainan *game* pembelajaran ini. Berdasarkan sifatnya, metode FSM ini sangat cocok digunakan sebagai basis perancangan perangkat lunak pengendalian yang bersifat reaktif dan *real time*. Salah satu keuntungan nyata penggunaan FSM adalah kemampuannya dalam mendekomposisi aplikasi yang relative besar dengan hanya menggunakan sejumlah kecil item *state*. Selain untuk bidang kontrol, penggunaan metode ini pada kenyataannya juga umum digunakan sebagai basis untuk perancangan protokol-protokol komunikasi, perancangan perangkat lunak *game*, aplikasi *Web* HTML dan sebagainya. Sehingga nantinya akan menjadikan sebuah *game* yang berkonten berpendidikan berbasis web dan sebagai media pembelajaran untuk meningkatkan hasil belajar dengan menyajikan *game* pertanyaan dalam permainan. Hal ini diharapkan dapat menjadi media suatu pembelajaran yang menyenangkan saat mendapat materi dan memotifasi dalam belajar lebih banyak lagi rambu-rambu lalu lintas agar menjadi lebih

mematuhi peraturan berlalu lintas. (Irwan Andriyanto : 2016).

### 1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah diuraikan di atas, maka permasalahan yang penulis diidentifikasi di dalam penelitian ini, yaitu :

1. Bagaimana membuat Aplikasi Pembelajaran Rambu-Rambu Lalu Lintas Berbasis Web?
2. Bagaimana menerapkan metode *Finite State Machine* (FSM) dalam Aplikasi Pembelajaran Rambu-rambu Lalu Lintas ?

### 1.3 Batasan Masalah

Dari latar belakang dan identifikasi masalah yang telah dijelaskan diatas, maka batasan masalah pada penelitian ini adalah sebagai berikut :

1. Aplikasi yang dibuat berbasis Web.
2. Pembelajaran rambu-rambu lalu lintas ini dibangun dengan menggunakan metode *Finite State Machine* (FSM).
3. Pembelajaran ini membahas jenis rambu-rambu lalu lintas, yaitu rambu peringatan, rambu perintah, rambu larangan dan rambu petunjuk.

## 2. ANALISA DAN PERANCANGAN

### 2.1 Analisa

Analisa sistem bertujuan untuk mengidentifikasi permasalahan-permasalahan yang ada pada sistem, dimana aplikasi yang dibangun meliputi lingkungan operasi, *user* dan elemen-elemen yang terkait. Analisa terhadap sistem diperlukan sebagai dasar untuk tahapan perancangan sistem, yaitu meliputi desain sistem, perancangan dan implementasi sistem.

### 2.2 Cara Kerja FSM

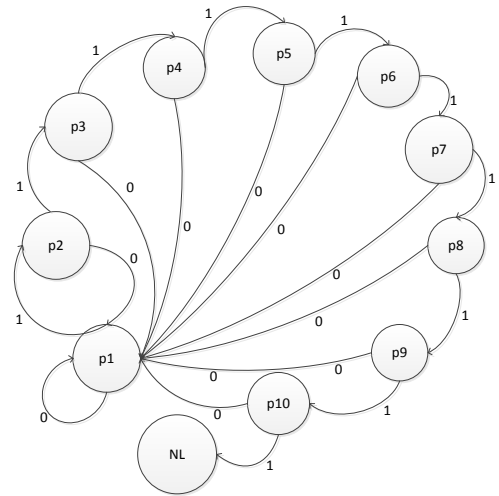
Dalam FSM masing-masing karakter menempati satu state. Biasanya, tindakan atau perilaku yang terkait dengan masing-masing state. Jadi selama karakter tetap dalam keadaan itu, ia akan terus melakukan tindakan yang sama. State terhubung bersama oleh transition. Setiap transition mengarah dari satu state ke state lain yang biasanya state tujuan *state* target ini disebut dengan action dan masing-masing memiliki seperangkat kondisi yang terkait. Jika permainan menentukan bahwa kondisi transition terpenuhi, maka karakter berubah dari state ke state target (action) melalui transition itu.

Dalam Game ini proses *Finite State Machine* (FSM) berjalan dengan mendapatkan aturan benar dan salah, ketika kondisi benar kemudian akan ditampilkan lanjutan state atau pertanyaan lanjutan dan selain dari jawaban benar maka akan melakukan pengulangan state / mengulangi pertanyaan. Berikut diagram rancangan game dengan *Finite State Machine* (FSM) :

Inisialisasi input :

Salah : 0 → Kembali

Benar : 1 → Next State



Gambar 1. Diagram *Finite State Machine* (FSM)

Dari diagram FSM diatas dapat dituliskan ke dalam tabel transisi sebagai berikut :

Tabel 1. Transisi *Finite State Machine* (FSM)

Current State	Next State	
	0	1
P1	P1	P2
P2	P1	P3
P3	P1	P4
P4	P1	P5
P5	P1	P6
P6	P1	P7
P7	P1	P8
P8	P1	P9
P9	P1	P10
P10	P1	NL
NL	-	-

Penjelasan tabel :

1. P1 merupakan pertanyaan 1 : jika pertanyaan 1 dapat dijawab dengan benar, maka *next state* adalah pertanyaan 2 dan jika pertanyaan 1 dijawab dengan jawaban salah maka akan kembali ke pertanyaan 1.
2. P2 merupakan pertanyaan 2 : jika pertanyaan 2 dapat dijawab dengan benar, maka *next state* adalah pertanyaan 3 dan jika pertanyaan 2 dijawab dengan jawaban salah maka akan kembali ke pertanyaan 1.
3. P3 merupakan pertanyaan 3 : jika pertanyaan 3 dapat dijawab dengan benar, maka *next state* adalah pertanyaan 4 dan jika pertanyaan 3 dijawab dengan jawaban salah maka akan kembali ke pertanyaan 1.
4. P4 merupakan pertanyaan 4 : jika pertanyaan 4 dapat dijawab dengan benar, maka *next state* adalah pertanyaan 5 dan jika pertanyaan 4

- dijawab dengan jawaban salah maka akan kembali ke pertanyaan 1.
5. P5 merupakan pertanyaan 5 : jika pertanyaan 5 dapat dijawab dengan benar, maka *next state* adalah pertanyaan 6 dan jika pertanyaan 5 dijawab dengan jawaban salah maka akan kembali ke pertanyaan 1.
  6. P6 merupakan pertanyaan 6 : jika pertanyaan 6 dapat dijawab dengan benar, maka *next state* adalah pertanyaan 7 dan jika pertanyaan 6 dijawab dengan jawaban salah maka akan kembali ke pertanyaan 1.
  7. P7 merupakan pertanyaan 7 : jika pertanyaan 7 dapat dijawab dengan benar, maka *next state* adalah pertanyaan 8 dan jika pertanyaan 7 dijawab dengan jawaban salah maka akan kembali ke pertanyaan 1.
  8. P8 merupakan pertanyaan 8 : jika pertanyaan 8 dapat dijawab dengan benar, maka *next state* adalah pertanyaan 9 dan jika pertanyaan 8 dijawab dengan jawaban salah maka akan kembali ke pertanyaan 1.
  9. P9 merupakan pertanyaan 9 : jika pertanyaan 9 dapat dijawab dengan benar, maka *next state* adalah pertanyaan 10 dan jika pertanyaan 9 dijawab dengan jawaban salah maka akan kembali ke pertanyaan 1.
  10. P10 merupakan pertanyaan 10 : jika pertanyaan 10 dapat dijawab dengan benar, maka *next state* adalah level selanjutnya dan jika pertanyaan 10 dijawab dengan jawaban salah maka akan kembali ke pertanyaan 1.
  11. NL merupakan *next level*

Pada game ini ada beberapa pertanyaan, setiap pertanyaan akan bernilai dengan jumlah benar dibagi jumlah pertanyaan kemudian dikali 100 seperti rumus berikut:

$$\frac{\text{Jumlah pertanyaan benar}}{\text{Jumlah pertanyaan}} \times 100 = \text{nilai}$$

Contoh :  $\frac{6}{10} \times 100 = 60$

atau  $\frac{4}{10} \times 100 = 40$

Note :

Jawaban Benar > Jawaban Salah = Level Selanjutnya

Jawaban Benar < Jawaban Salah = Tetap di level yang sama

Keterangan :

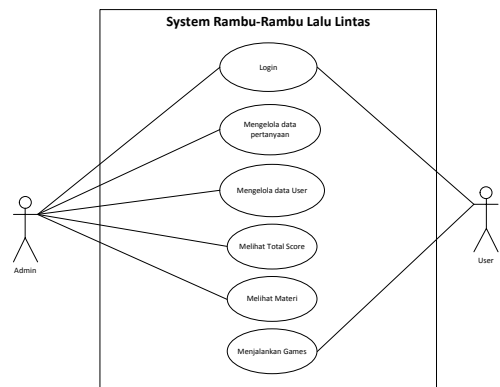
Jika jawaban pertanyaan benar lebih banyak dari jumlah jawaban pertanyaan yang salah maka sistem akan melanjutkan ke level selanjutnya, namun apabila jawaban dari pertanyaan benar lebih sedikit dari jawaban dari pertanyaan salah maka sistem akan mengulangi ke pertanyaan

pertama yang berarti sistem tidak dapat melanjutkan ke level selanjutnya.

### 2.3. Perancangan Sistem

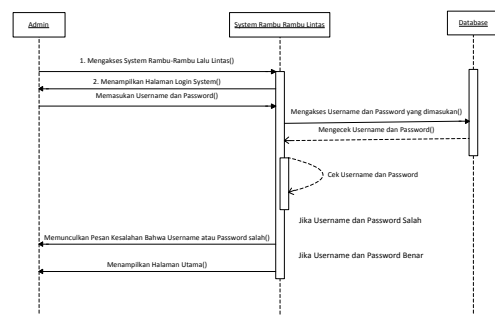
Perancangan perangkat lunak penyelesaian aplikasi pengenalan dan pembelajaran bangun ruang ini menggunakan Eclipse yang merupakan peranti pengembangan aplikasi yang menggunakan bahasa pemrograman *java*. Untuk *database* yang berfungsi sebagai tempat penyimpanan digunakan *SQLite Manager*.

Berikut merupakan rancangan menggunakan *usecase diagram* yang akan menjelaskan secara sederhana bagaimana aplikasi ini berjalan:



Gambar 2. Diagram usecase

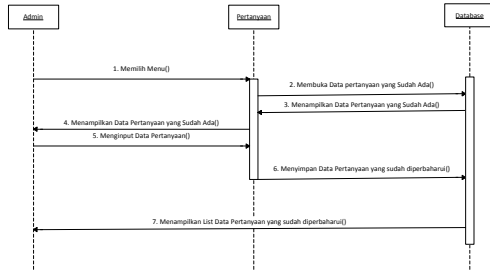
Diagram *use case* menunjukkan cara kerja sistem secara umum. Dalam diagram ini ditampilkan dua buah *actor* yaitu *admin* dan *user*. *Admin* berinteraksi dengan fungsi *login*, mengelola data pertanyaan, mengelola data user, melihat total score, melihat materi dan *logout*. *User* dapat berinteraksi dengan fungsi *login*, melihat materi, menjalankan games dan *logout*.



Gambar 3. Sequence Diagram login

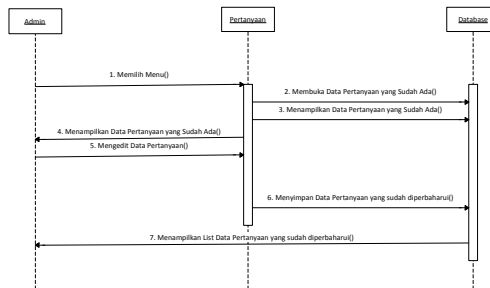
Berdasarkan gambar, dapat dilihat bahwa proses dimulai dengan *admin* mengakses sistem Pembelajaran Rambu-Rambu Lalu Lintas, kemudian sistem akan menampilkan halaman *login*. Lalu *admin* memasukkan *username* dan *password* dan sistem pun akan mengecek *username* dan *password* ke dalam *database*. Jika

*username* dan *password* yang dimasukan salah maka sistem akan memunculkan pesan kesalahan bahwa *username* dan *password* salah dan sistem akan kembali menampilkan halaman *login*. Jika *username* dan *password* benar maka sistem akan menampilkan halaman utama.



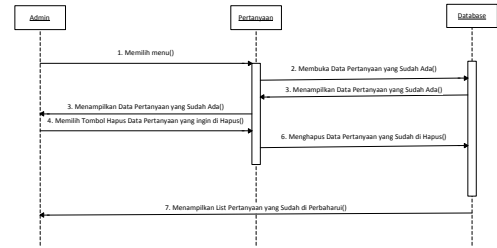
Gambar 4. *Sequence Diagram* Data Pertanyaan

Berdasarkan gambar di atas dapat dilihat bahwa proses dimulai dengan *admin* memilih menu, lalu sistem membuka data pertanyaan *database* menampilkan data pertanyaan, kemudian *admin* menginput data pertanyaan sistem menyimpan data pertanyaan ke dalam *database* dan proses berakhir dengan sistem menampilkan data pertanyaan yang sudah diperbaharui.



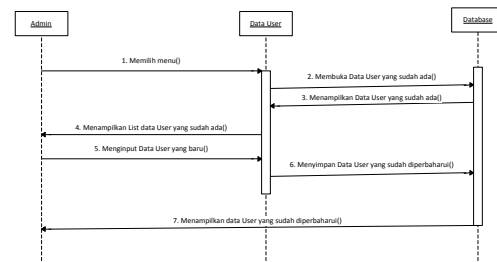
Gambar 5. *Sequence diagram* Edit Data Pertanyaan

Berdasarkan gambar diatas dapat dilihat bahwa proses dimulai dengan *admin* memilih menu, lalu sistem akan membuka data pertanyaan yang sudah disimpan sebelumnya dan sistem akan menampilkan data pertanyaan yang sudah tersimpan sebelumnya. Dilanjutkan dengan *admin* memilih menu edit pada data yang ingin di edit, setelah itu sistem akan menyimpan data yang sudah diperbaharui ke dalam *database*. Kemudian sistem akan menampilkan data pertanyaan yang sudah diperbaharui.



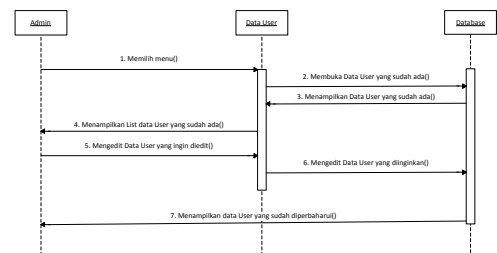
Gambar 6. *Sequence diagram* Hapus Data Pertanyaan

Berdasarkan gambar diatas dapat dilihat bahwa proses dimulai dengan *admin* memilih menu, lalu sistem akan membuka data pertanyaan yang sudah disimpan sebelumnya dan sistem akan menampilkan data pertanyaan yang sudah tersimpan sebelumnya. Dilanjutkan dengan *admin* memilih menu hapus pada data yang ingin di hapus, setelah itu sistem akan menyimpan data yang sudah diperbaharui ke dalam *database*. Kemudian sistem akan menampilkan data pertanyaan yang sudah diperbaharui.



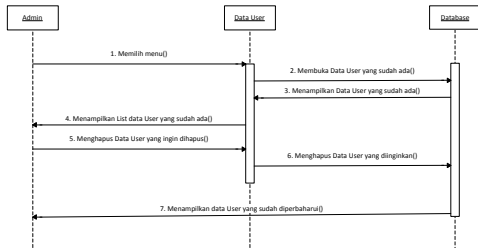
Gambar 7. *Sequence diagram* Data User

Berdasarkan gambar diatas dapat dilihat bahwa proses dimulai dengan *admin* memilih menu, lalu sistem akan membuka data *user* yang sudah disimpan sebelumnya dan sistem akan menampilkan data *user* yang sudah tersimpan sebelumnya. Dilanjutkan dengan *admin* memilih menu simpan pada data yang ingin diinput, setelah itu sistem akan menyimpan data yang sudah diperbaharui ke dalam *database*. Kemudian sistem akan menampilkan data *user* yang sudah diperbaharui.



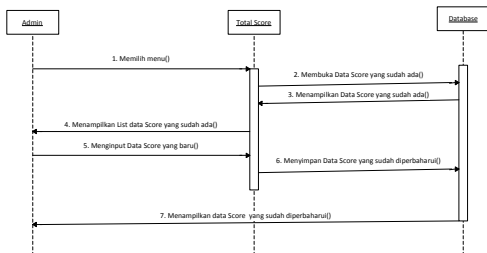
Gambar 8. *Sequence diagram* Edit Data User

Berdasarkan gambar diatas dapat dilihat bahwa proses dimulai dengan *admin* memilih menu, lalu sistem akan membuka data *user* yang sudah disimpan sebelumnya dan sistem akan menampilkan data *user* yang sudah tersimpan sebelumnya. Dilanjutkan dengan *admin* memilih menu edit pada data yang ingin diinput, setelah itu sistem akan menyimpan data yang sudah diperbaharui ke dalam *database*. Kemudian sistem akan menampilkan data *user* yang sudah diperbaharui.



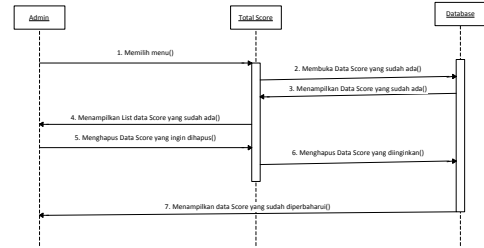
Gambar 9. *Sequence diagram* Hapus Data User

Berdasarkan gambar diatas dapat dilihat bahwa proses dimulai dengan *admin* memilih menu, lalu sistem akan membuka data *score* yang sudah ada dan sistem akan menampilkan *list* data *score* yang sudah disimpan sebelumnya. Dilanjutkan dengan *admin* memilih menu hapus pada data yang ingin di hapus, setelah itu sistem akan menyimpan data yang sudah diperbaharui ke dalam *database*. Kemudian sistem akan menampilkan data *user* yang sudah diperbaharui.



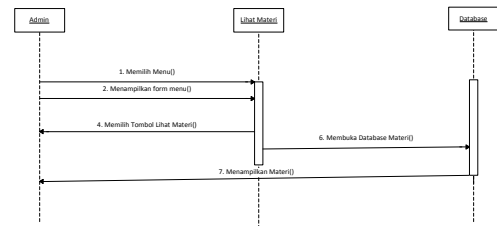
Gambar 10. *Sequence diagram* Total Score

Berdasarkan gambar dapat dilihat bahwa proses dimulai dengan *admin* memilih menu, lalu sistem akan membuka data *score* yang sudah ada dan sistem akan menampilkan *list* data *score* yang sudah disimpan sebelumnya. Dilanjutkan dengan *admin* memilih menu simpan untuk menyimpan *score* yang baru, setelah itu sistem akan menyimpan *score* yang sudah diperbaharui ke dalam *database*. Kemudian sistem akan menampilkan data *score* yang sudah diperbaharui.



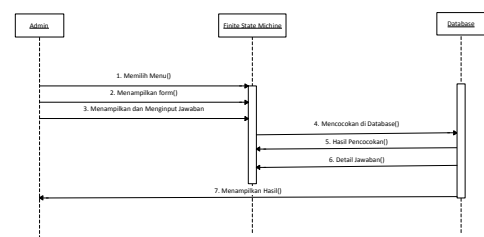
Gambar 11. *Sequence diagram* Hapus Total Score

Berdasarkan gambar dapat dilihat bahwa proses dimulai dengan *admin* memilih menu, lalu sistem akan membuka data *score* yang sudah ada dan sistem akan menampilkan *list* data *score* yang sudah disimpan sebelumnya. Dilanjutkan dengan *admin* memilih menu hapus untuk menyimpan *score* yang baru, setelah itu sistem akan menghapus *score* yang sudah diperbaharui ke dalam *database*. Kemudian sistem akan menampilkan data *score* yang sudah diperbaharui.



Gambar 12. *Sequence diagram* Materi

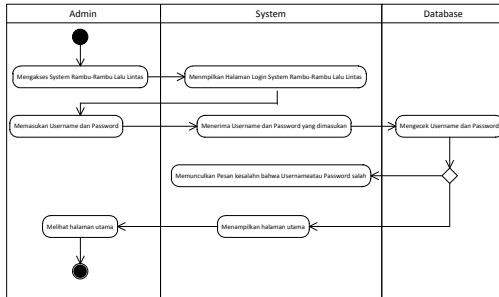
Berdasarkan gambar dapat dilihat bahwa proses dimulai dengan *admin* memilih menu lalu sistem akan membuka menu materi. System akan membuka *database*, kemudian sistem akan menampilkan materi. Materi ini dapat di lihat oleh kedua *actor* yaitu *admin* dan *user*



Gambar 13. *Sequence diagram* Games

Berdasarkan gambar dapat dilihat bahwa proses dimulai dengan *admin* memilih menu untuk menampilkan form dan menginput jawaban lalu sistem akan mencocokkan jawaban di dalam *database* setelah hasil pencocokkan di periksa dan ditemukan jawaban sistem akan menampilkan hasil.

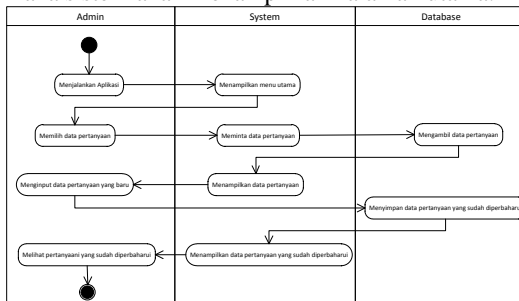
Activity diagram menggambarkan aktivitas dari aplikasi pembelajaran rambu-rambu lalu lintas yang akan dibangun. Diagram ini bermanfaat untuk membantu memahami proses dari aplikasi pembelajaran rambu-rambu lalu lintas ini secara keseluruhan. Berikut merupakan rancangan menggunakan activity diagram dalam pembuatan perangkat lunak ini:



Gambar 14. Activity Diagram Login

Keterangan :

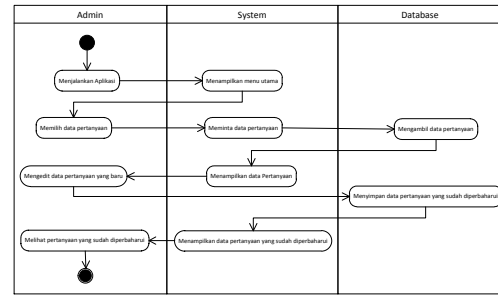
Proses login dimulai dengan admin mengakses sistem rambu-rambu lalu lintas dan sistem pun akan menampilkan halaman login. Lalu admin memasukkan username dan password, sehingga sistem akan mengecek username dan password tersebut ke dalam database. Jika username dan password yang dimasukkan admin salah sistem akan memunculkan pesan kesalahan dan akan memasukkan kembali username dan password hingga benar. Jika username dan password benar maka sistem akan menampilkan halaman utama.



Gambar 15. Activity Diagram Data Pertanyaan

Keterangan :

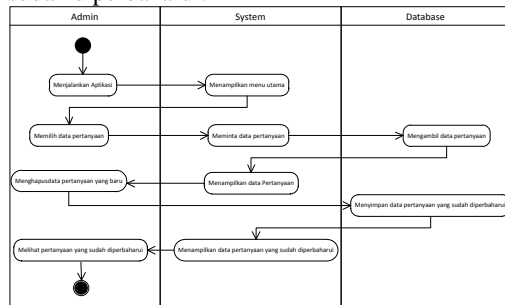
Apabila admin ingin menambah data pertanyaan maka admin dapat memilih menu simpan yang tersedia pada form data pertanyaan. Kemudian admin dapat menginput data pertanyaan yang baru pada form simpan dan sistem akan menyimpan data yang telah diinput tersebut ke dalam database selanjutnya admin dapat melihat list data pertanyaan yang sudah diperbaharui.



Gambar 16. Activity Diagram Edit Data Pertanyaan

Keterangan :

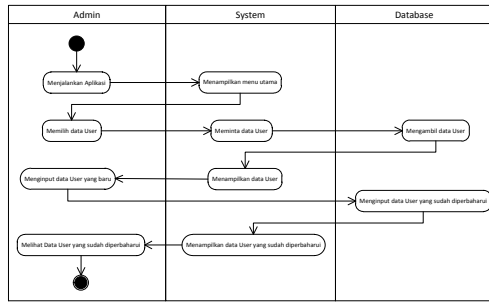
Apabila admin ingin menambah data pertanyaan maka admin dapat memilih menu edit yang tersedia pada form data pertanyaan. Kemudian admin dapat mengedit data pertanyaan yang baru pada form edit dan sistem akan menyimpan data yang telah diedit tersebut ke dalam database selanjutnya admin dapat melihat pertanyaan yang sudah diperbaharui.



Gambar 17. Activity Diagram Hapus Data Pertanyaan

Keterangan :

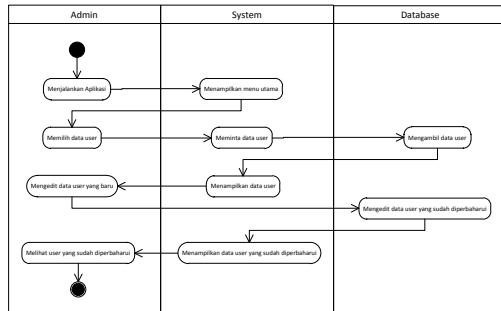
Apabila admin ingin menghapus data pertanyaan maka admin dapat memilih menu hapus yang tersedia pada form data pertanyaan. Kemudian admin dapat menghapus data pertanyaan yang baru pada form hapus dan sistem akan menyimpan data yang telah dihapus tersebut ke dalam database selanjutnya admin dapat melihat pertanyaan yang sudah diperbaharui.



Gambar 18. Activity Diagram Simpan Data User

Keterangan :

Apabila *admin* ingin menambah data *user* maka *admin* dapat memilih menu simpan yang tersedia pada *from* data *user*. Kemudian *admin* dapat menginput data *user* yang baru pada form simpan dan sistem akan menyimpan data yang telah diinput tersebut ke dalam *database* selanjutnya *admin* dapat melihat *list* data *user* yang sudah diperbaharui.

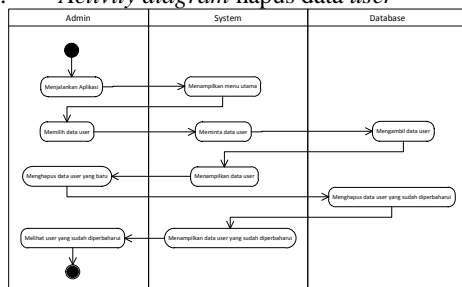


Gambar 19. Activity Diagram Edit Data User

Keterangan :

Apabila *admin* ingin mengedit data *user* maka *admin* dapat memilih menu edit yang tersedia pada *from* data *user*. Kemudian *admin* dapat mengedit data *user* yang baru pada form edit dan sistem akan menyimpan data yang telah diinput tersebut ke dalam *database* selanjutnya *admin* dapat melihat *list* data *user* yang sudah diperbaharui.

1. Activity diagram hapus data user

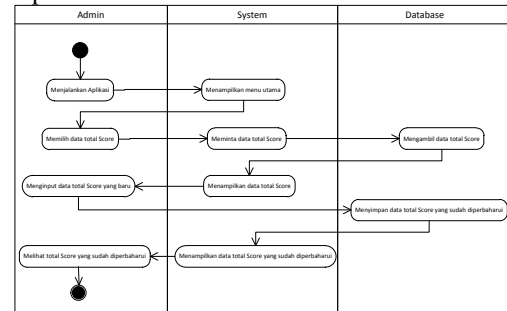


Gambar 20. Activity Diagram Hapus Data User

Keterangan :

Apabila *admin* ingin menghapus data *user* maka *admin* dapat memilih menu hapus yang tersedia

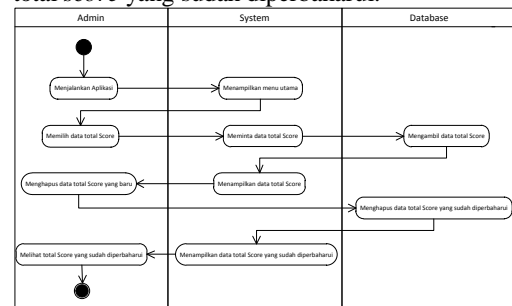
pada *from* data *user*. Kemudian *admin* dapat menghapus data *user* yang baru pada form hapus dan sistem akan menyimpan data yang telah dihapus tersebut ke dalam *database* selanjutnya *admin* dapat melihat *list* data *user* yang sudah diperbaharui.



Gambar 21. Activity Diagram Total Score

Keterangan :

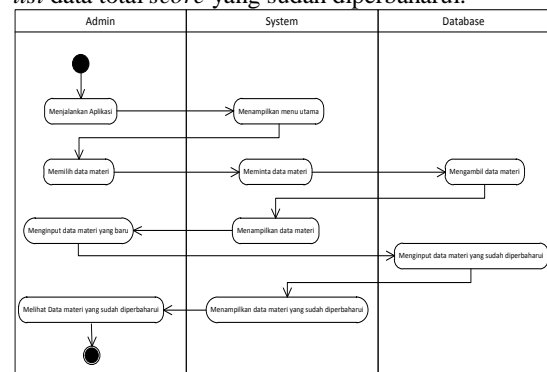
Apabila *admin* ingin menambahkan data total *score* maka *admin* dapat menginput data total *score* yang baru dan sistem akan menyimpan data dalam *database* selanjutnya *admin* dapat melihat total *score* yang sudah diperbaharui.



Gambar 22. Activity Diagram Hapus Total Score

Keterangan :

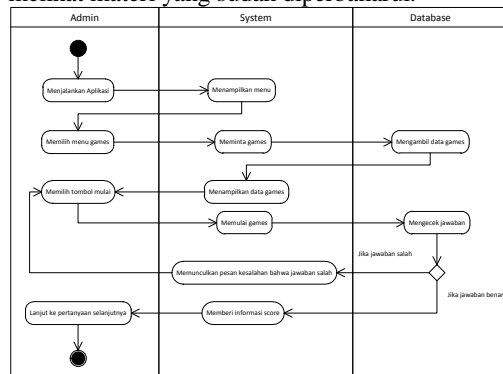
Apabila *admin* ingin menghapus data total *score* maka *admin* dapat memilih menu hapus yang tersedia pada *from* data total *score*. Kemudian *admin* dapat menghapus data total *score* yang baru pada form hapus dan sistem akan menyimpan data yang telah dihapus tersebut ke dalam *database* selanjutnya *admin* dapat melihat *list* data total *score* yang sudah diperbaharui.



Gambar 23. Activity Diagram Materi

Keterangan :

Apabila *admin* ingin menambahkan data materi maka *admin* dapat menginput data materi yang baru yang kemudian akan disimpan ke dalam *database* selanjutnya *admin* dan *user* dapat melihat materi yang sudah diperbaharui.



Gambar 24. Activity Diagram Materi

Keterangan :

Aplikasi di mulai dengan menjalankan aplikasi dan sistem akan menampilkan data *games* dan memilih tombol mulai untuk memulai jalannya *games* dan sistem akan mengecek jawaban ke dalam *database* apabila jawaban salah maka sistem akan memunculkan pesan kesalahan jawaban dan apabila jawaban benar maka sistem akan member informasi *score* dan sistem akan lanjut ke pertanyaan selanjutnya.

#### 2.4. User Interface

Pada *user interface* ditampilkan desain grafis antarmuka. Adapun *form* tampilan dalam aplikasi ini, antara lain tampilan *splash screen*, menu utama, menu data tanaman, menu data penyakit, menu kategori tanaman sayuran, menu kategori tanaman buah, menu kategori tanaman rempah, menu kategori penyakit dalam, menu kategori penyakit luar dan menu tentang.

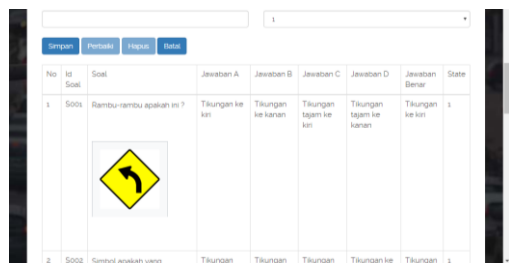
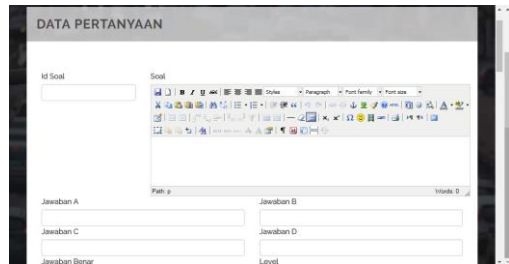


Gambar 25. Tampilan Splash Screen

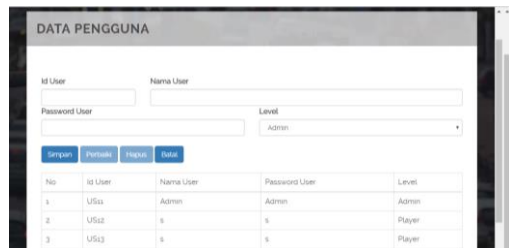
*Splash screen* merupakan layar pembuka saat pertama kali aplikasi dibuka. *Splash screen* juga memiliki fungsi sebagai *loader* (pemuat) *database* yang digunakan, sehingga ketika aplikasi dijalankan, aplikasi tersebut hanya perlu membuka *database* tanpa perlu memuatnya terlebih dahulu.



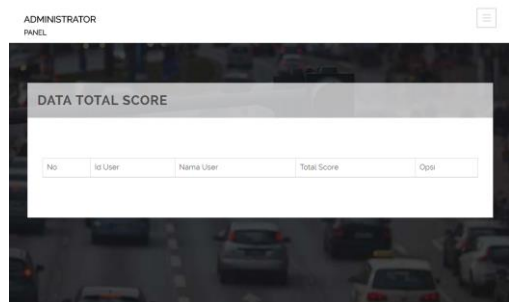
Gambar 3. Tampilan Menu Admin



Gambar 26. Tampilan Data Pertanyaan

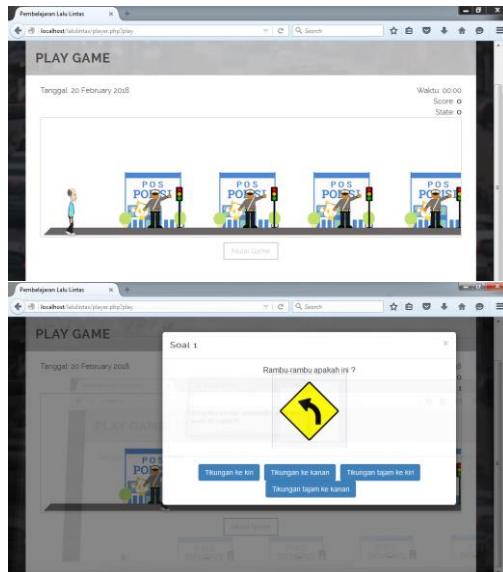


Gambar 27. Tampilan Data User

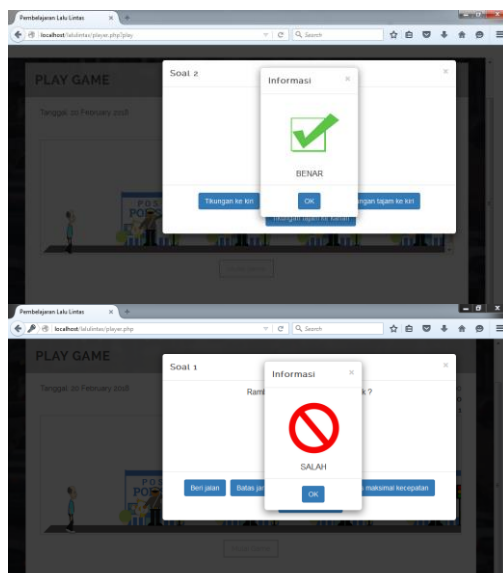


Gambar 28. Tampilan Score





Gambar 29. Tampilan Permainan



Gambar 30. Tampilan Jawaban

### 3. KESIMPULAN

Adapun kesimpulan yang diperoleh dari penelitian ini, adalah sebagai berikut:

1. Aplikasi berbasis Web ini bertujuan sebagai media pembelajaran kepada masyarakat dalam mengenal dan mempelajari arti rambu-rambu lalu lintas agar lebih mematuhi tata cara berlalu lintas yang baik dan benar.
2. Pembelajaran rambu-rambu lalu lintas ini di buat menggunakan *Finite State Machine* (FSM) dalam menyelesaikan permainan dengan menjawab pertanyaan.
3. Dalam aplikasi ini disisipkan soal-soal yang harus dijawab pengguna untuk mengukur pengetahuan dalam berlalu lintas.
4. Permainan akan berhasil jika telah menyelesaikan permainan dengan menjawab pertanyaan dalam setiap level.
5. Permainan ini dibuat dengan pertanyaan bergambar untuk menarik pengguna dalam menjawab semua pertanyaan namun aplikasi

akan mengulang keawal pertanyaan apabila terjadi kesalahan dalam menjawab pertanyaan.

### PUSTAKA

- [1] Abdullah, Dahlan., dkk. 2015, *Game Edukasi Berbasis Role Playing Game dengan Metode Finite State Macine (FSM)*, Informatika. Universitas Malikussaleh Lhoksemawe.
- [2] A.S., Rosa, Shalahuddin, M., 2013, *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*, Informatika, Bandung.
- [3] Andriyanto, Irwan., 2016, *Game Edukasi Pengenalan Rambu-Rambu Lalu Lintas Untuk Anak SD*, Skripsi. Universitas Muhammadiyah, Surakarta.
- [4] Huda, A.S., 2016, *Game Edukasi Cepat Tepat Dengan Metode Finite State Machine pada Smartphone*, Skripsi. Islam Negri Maulana Malik Ibrahim, Malang.
- [5] Muharraran, Maryam., 2014, *Aplikasi Pembelajaran Dasar Aksara Sunda Berbasis Android*, Program Studi Sistem Informasi, Universitas Komputer Indonesia, Bandung.
- [6] Pratama, Dargawan Aditia., 2014, *Aplikasi Edukasi Bagi Anak Usia Dini Berbasis Mobile Android*, Program Studi Sistem Informasi, Universitas Komputer Indonesia, Bandung.
- [7] Reza, Baihaqi., 2015, *Game Edukasi Berbasis Role Playing Game Dengan Metode Finite State Machine*, Skripsi. Universitas Malikussaleh, Lhoksemawe.
- [8] Rostianingsih, Silvia, dkk., 2011, *Game Simulasi Finite State Machine untuk Pertanian dan Perternakan*, Informatika, Kristen Petra, Surabaya.
- [9] Sidik, Fajri., 2016, *Implementasi Algoritma Multiplicative Congruential Random Number Generator Pada Try Out SMA (Sekolah Menengah Atas) Berbasis Web*, Skripsi, Universitas Malikussaleh, Lhoksemawe.
- [10] Sugiman, V.R, dkk., 2012, *Pengembangan Aplikasi Rambu-Rambu Lalu Lintas berbasis Android Menggunakan Metode Prototyping*, Sekolah Tinggi Teknologi, Garut vol 11.
- [11] Suyanto, Wardan., 2012, *Pengenalan Rambu-Rambu Lalu Lintas Bagi Siswa SMK Dalam Rangka Membentuk Prilaku Tertip Berlalu Lintas*, Yogyakarta.