

Game Edukasi Rambu Lalu Lintas Berbasis Android

Ida Bagus Kresna Adi Jaya, Putu Wira Buana, A. A. K. Agung Cahyawan W.

Jurusan Teknologi Informasi, Fakultas Teknik, Universitas Udayana

Bukit Jimbaran, Bali, Indonesia, telp. +6285102853533

e-mail: kresnaadijaya@gmail.com, wbhuana@yahoo.com, a.cahyawan@yahoo.com

Abstrak

Jumlah kendaraan setiap tahun di Indonesia terus meningkat. Salah satu penyebab meningkatnya jumlah kendaraan di Indonesia adalah orang Indonesia lebih suka menggunakan kendaraan pribadi dibanding kendaraan umum. Konsekuensinya angka pelanggaran kendaraan bermotor di Indonesia meningkat tajam. Berdasarkan data pelanggaran dari KNKT dengan banyaknya pelanggaran yang dilakukan oleh pengendara kendaraan bermotor, maka dikembangkan penelitian yang berjudul Game Edukasi Rambu Lalu Lintas Berbasis Android. Game ini dibuat untuk membantu para pengendara kendaraan bermotor dalam mempelajari dan menghafal arti dari rambu lalu lintas. Metode Pengacakan Fisher-Yates digunakan dalam Game Edukasi Rambu Lalu Lintas yang bersifat Multiple-Choice. Soal yang terdapat dalam Game Edukasi Rambu Lalu Lintas sebanyak 84 soal, yang dibagi menjadi tiga level yang berbeda. Level Mudah untuk rambu yang sering dijumpai di jalan perkotaan di Indonesia. Level Sedang untuk rambu yang jarang ditemui di perkotaan, melainkan banyak berada di daerah pedesaan maupun pegunungan. Level Susah lebih fokus kepada rambu yang sifatnya lebih terkait kondisi tertentu yang tidak umum atau lebih mengkhusus. Hasil dari penelitian game ini menunjukkan bahwa responden yang menyukai aspek gameplay dari Game Edukasi Rambu Lalu Lintas sebesar 100%.

Kata kunci: Metode Pengacakan Fisher-Yates, Game, Android, Rambu Lalu Lintas.

Abstract

The number of motor vehicle operated in Indonesia is gradually increasing each year. One of the leading factors is that Indonesian prefers utilizing their private vehicle rather than make use of the public transports. Consequently, the number of motor vehicle related offences in Indonesia had been significantly increasing. Considering the violation data compiled from KNKT, author will conduct a research study entitled Android-Based Traffic Signs Education Game. The main objectives of this game are to assist motor vehicle users to learn and memorize traffic signs. The Fisher-Yates Shuffle Method is adapted to the traffic signs education game with multiple-choice questions. The traffic signs education game consists of 84 questions that is divided into three different levels: The Easy Level is focusing on the traffic signs that commonly seen in urban area in Indonesia; The Medium Level is focusing on the traffic signs that rarely can be seen in the urban area, however, commonly seen in rural area and/or highlands area; and The Hard Level is focusing on the specialized traffics signs. The result show that there are 100% respondents enjoy the gameplay from Android-Based Traffic Signs Education Game.

Keywords : Fisher Yates Shuffle Methods, Game, Android, Traffic Signs.

1. Pendahuluan

Setiap tahun jumlah kendaraan terus meningkat. Perkembangan jumlah kendaraan bermotor menurut jenis dari Tahun 1987-2012, jumlah kendaraan bermotor Tahun 2012 sebanyak 94.373.324 unit dibandingkan Tahun 2011 sebanyak 85.601.351 unit [1]. Kecelakaan lalu lintas dapat disebabkan oleh banyak hal, salah satunya adalah pengendara yang tidak mematuhi rambu-rambu lalu lintas. Polri mengakui angka pelanggaran meningkat tajam. Tahun 2012, tercatat ada 135.940 pelanggaran lalu lintas. Tahun 2013 pelanggaran lalu lintas meningkat signifikan sebesar 382.286, meningkat sekitar 181 persen dari tahun sebelumnya [2]. Data Investigasi Kecelakaan Angkutan Jalan Tahun 2007-2013 menyatakan bahwa jumlah

kecelakaan sampai tanggal 27 Desember 2013 adalah 85.662,21.375 orang meninggal dunia dan 121.173 orang lainnya mengalami luka ringan dan berat [3].

Berdasarkan data pada paragraf pertama, dibuatkan sebuah *Game* Edukasi Rambu Lalu Lintas Berbasis Android dengan memanfaatkan metode pengacakan Fisher-Yates. Manfaat metode pengacakan Fisher-Yates pada penelitian yang berjudul "*Perancangan Ujian Online pada STMIK GI MDP Berbasis Web*" oleh (Antony Susanto, 2013) untuk mengacak soal ujian sehingga setiap mahasiswa memiliki urutan soal ujian yang berbeda [4]. Ade-Ibijola pada tahun 2012 juga melakukan penelitian yang berjudul "*A Simulated Enhancement of Fisher-Yates Algorithm for Shuffling in Virtual Card Games using Domain-Specific Data Structures*" guna mengacak kartu agar memungkinkan setiap kartu pada iterasi selanjutnya berbeda [5]. Penerapan metode pengacakan Fisher-Yates pada *Game* Edukasi Rambu Lalu Lintas Berbasis Android adalah untuk mengacak setiap kuis yang ada. Sehingga jika *user* yang telah *installGame* Edukasi Rambu Lalu Lintas pada *smartphone* tidaklah sama urutan kuisnya.

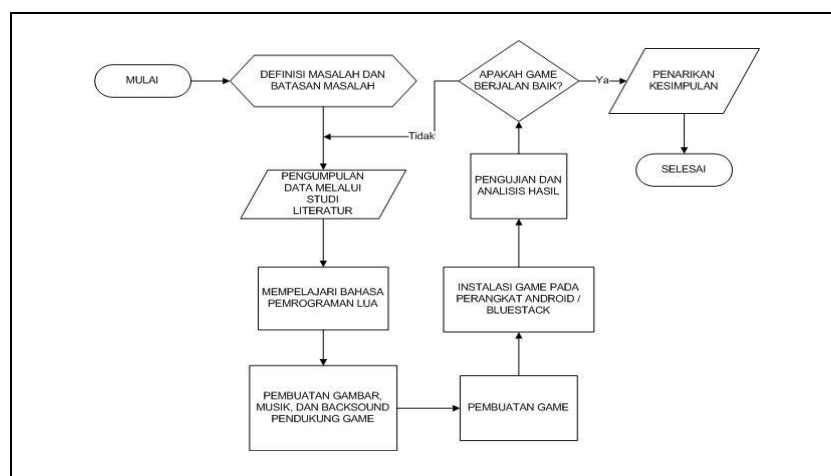
Game Edukasi Rambu Lalu Lintas Berbasis Android adalah *game* yang ber-genre *Multiple-Choice*. *User* diberikan 4 jawaban untuk setiap soal pada *game* yang dirancang, dimana hanya ada 1 jawaban benar. *User* harus menjawab 20 soal setiap bermain yang diacak menggunakan Metode Pengacakan Fisher-Yates. Metode Pengacakan Fisher-Yates digunakan untuk mengacak soal yang diberikan kepada *user* tiap kali bermain, sehingga soal yang muncul tidaklah sama urutan kemunculannya. *Game* Edukasi Rambu Lalu Lintas bertujuan untuk dapat membantu para pengendara kendaraan bermotor dalam mengenali, mempelajari dan menghafal arti dari rambu lalu lintas.

2. Metodologi Penelitian

Alur penelitian dari Rancang Bangun *Game* Edukasi Rambu Lalu Lintas pada platform Android yang dilakukan dalam penelitian ini adalah sebagai berikut:

1. Pendefinisian masalah dan batasan masalah dari *game* yang akan dibuat.
2. Pengumpulan data yang berhubungan dengan perancangan dan pembuatan *Game*, melalui studi literatur.
3. Mempelajari Bahasa Pemrograman Lua yang akan digunakan dalam pemrograman *Game*.
4. Pembuatan gambar-gambar yang diperlukan dalam *Game*, penentuan musik dan sound untuk latar *Game*.
5. Pembuatan *Game*.
6. Instalasi *Game* pada perangkat Android.
7. Pengujian terhadap *Game* yang telah dibuat dan terinstall pada perangkat Android kemudian dilakukan analisis hasil pengujian *Game*.

Berikut ini adalah diagram alur penelitian yang dilakukan :

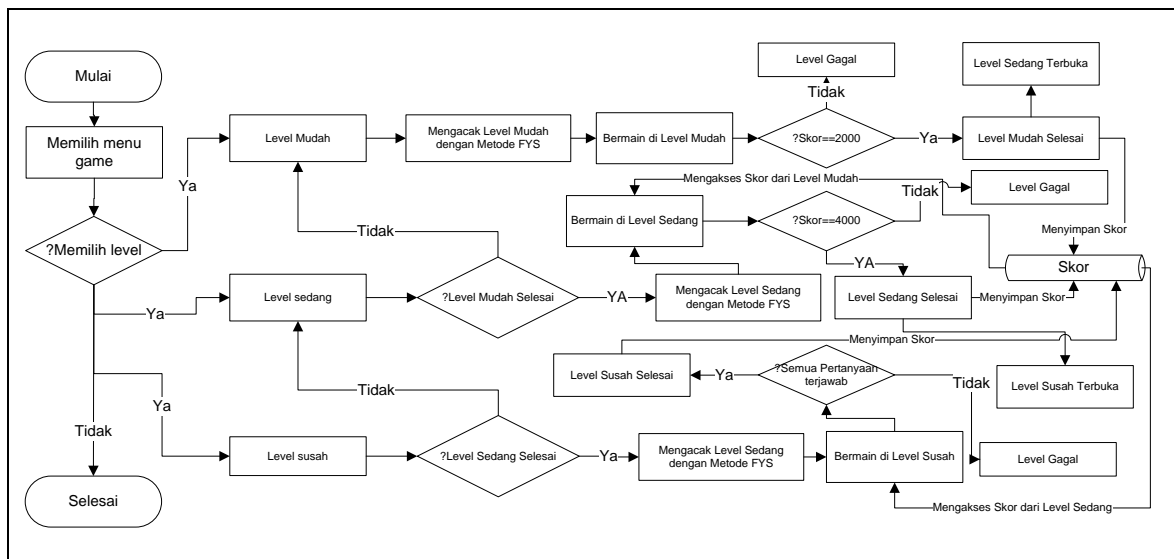


Gambar 1. Alur Penelitian *Game* Edukasi Rambu Lalu Lintas pada Platform Android

Gambar 1 diatas menjelaskan alur penelitian dimulai dari mendefinisikan masalah dan batasan masalah. Tahap kedua adalah pengumpulan data melalui studi literatur, tahap ketiga adalah mempelajari Bahasa Pemrograman Lua, kemudian proses pembuatan asset gambar, musik, dan *background* pendukung *Game*. Tahap kelima adalah pembuatan *Game*. Tahap keenam adalah proses instalasi *Game* pada perangkat Android. Kemudian *Game* ini diuji dan dianalisa hasilnya, jika *Game* berjalan dengan baik, maka *Game* ini layak untuk digunakan.

2.1 Gambaran Umum Sistem

Metode Pengacakan Fisher-Yates digunakan dalam *Game* Edukasi Rambu Lalu Lintas yang bersifat *Multiple-Choice*. Soal yang terdapat dalam *Game* Edukasi Rambu Lalu Lintas sebanyak 84 soal, yang dibagi menjadi tiga *level* yang berbeda. *Level* Mudah untuk rambu yang sering dijumpai di jalan perkotaan di Indonesia. *Level* Sedang untuk rambu yang jarang ditemui di perkotaan, melainkan banyak berada di daerah pedesaan maupun pegunungan. *Level* Susah lebih fokus kepada rambu yang sifatnya lebih mengkhhusus.



Gambar 2. *Flowchart* Gambaran Umum Sistem *Game* Edukasi Rambu Lalu Lintas Berbasis Android

Gambar 2 menunjukkan gambaran umum dari *Game* Edukasi Rambu Lalu Lintas Berbasis Android. Terdapat 30 soal di dalam *Level* Mudah, namun yang dikeluarkan oleh sistem sebanyak 20 soal saja. Setiap soal benar diberikan skor 100 poin, setiap soal salah skor dikurangi sebanyak 50 poin. Ketika poin sudah terkumpul sebanyak 200 poin, maka bantuan pertama berupa 50:50 terbuka. Jika *user* menggunakan bantuan 50:50, maka sistem menghilangkan dua jawaban salah dan poin *user* dikurangi 200 poin. Jika *user* memakai bantuan 50:50 dan menjawab benar, maka *user* diberikan poin selayaknya benar menjawab soal sebanyak 100 poin. Ketika poin telah terkumpul sebanyak 1000 poin, bantuan kedua berupa bantuan *auto-completed* terbuka. Bantuan *auto-completed* menghilangkan tiga jawaban salah dan otomatis *user* langsung menuju soal berikutnya dengan mengorbankan 1000 poin.

Level Sedang diisi dengan 20 soal dari total keseluruhan soal 30 buah. *User* menjawab benar maka memperoleh poin sebesar 100, jika *user* salah menjawab pertanyaan maka poin dikurangi 150 tiap melakukan kesalahan. *Level* Sedang memiliki *countdowntimer* seperti *Level* Mudah, yaitu selama 10 detik. *User* diberikan bantuan berupa jawaban 50:50 ketika poin tiap yang terkumpul lebih dari 500 poin. *User* juga diberikan bantuan berupa *auto-completed* ketika poin tiap yang terkumpul lebih dari 2000 poin. *User* dinyatakan berhasil dan dapat melanjutkan ke *level* berikutnya jika memperoleh poin sebanyak 4000 poin

Level Susah diisi dengan 20 soal dari total keseluruhan soal 24 buah. Setelah *user* benar menjawab soal maka memperoleh poin sebesar 200, jika *user* salah menjawab pertanyaan maka poin dikurangi 300 tiap melakukan kesalahan. *User* diberikan bantuan berupa

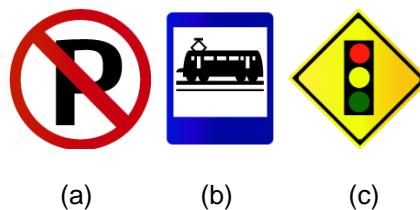
jawaban 50:50 ketika poin tiap yang terkumpul lebih dari 1000 poin. User juga diberikan bantuan berupa jawaban langsung ketika poin tiap yang terkumpul lebih dari 4000 poin.

3. Kajian Pustaka

Pengumpulan teori yang didapatkan dari buku atau internet maupun jurnal yang menunjang pembuatan aplikasi ini.

3.1 Rambu Lalu Lintas

Rambu lalu lintas yang resmi di Indonesia diatur dalam Keputusan Menteri Perhubungan Nomor 61 Tahun 1993 tentang Rambu Lalu Lintas [6]. Terdapat 3 kategori utama rambu lalu lintas di Indonesia. Rambu yang didominasi oleh warna biru merupakan rambu perintah. Rambu yang didominasi warna kuning merupakan rambu peringatan. Rambu yang didominasi oleh warna merah merupakan rambu larangan. Sumber referensi rambu lalu lintas yang ada didalam *Game Edukasi Rambu Lalu Lintas Berbasis Android* didapat dari buku yang berjudul "Tertib Berlalu Lintas" karangan Rinto Raharjo [7].



Gambar 3. (a) Rambu Larangan, (b) Rambu Perintah, (c) Rambu Peringatan

Gambar 3(a) merupakan rambu larangan untuk parkir. Gambar 3(b) merupakan rambu perintah/himbauan ada pemberhentian trem disini. Gambar 3(c) merupakan rambu peringatan bahwa anda melintasi lampu lalu lintas.

3.2 Metode Pengacakan Fisher-Yates

Metode Pengacakan Fisher-Yates diadopsi dari nama penemunya, yaitu Ronald Fisher dan Frank Yates, yang pertama kali diperkenalkan pada tahun 1934 dan kemudian di revisi kembali pada tahun 1948 [5]. Metode Pengacakan Fisher-Yates adalah sebuah metode untuk menghasilkan suatu permutasi acak dari suatu himpunan terhingga, dengan kata lain untuk mengacak suatu himpunan tersebut. Metode Pengacakan Fisher-Yates jika diimplementasikan dengan benar, maka hasil dari algoritma ini tidak akan berat sebelah, sehingga setiap permutasi memiliki kemungkinan yang sama.

Misalkan dalam permainan kartu *bridge* ada 10 kartu yang diacak, maka array-nya adalah urutan Kartu = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]. Array tersebut dimasukkan kedalam prosedur pengacakan dimana proses pengacakan terjadi, kemudian didapat panjang array yang kemudian dimasukkan ke dalam variabel m ($m=10$).

Langkah-langkah pengacakan soal dengan Metode Pengacakan Fisher-Yates adalah sebagai berikut [5] :

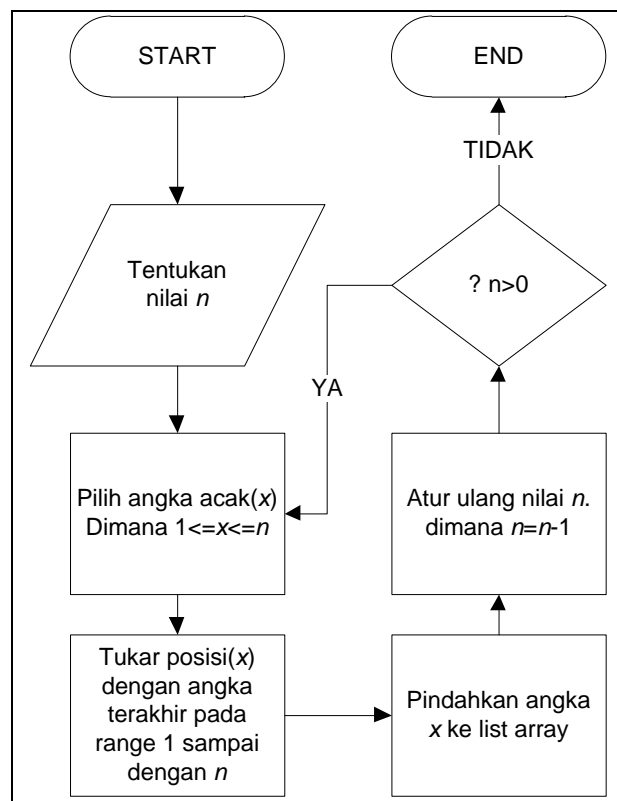
1. Ambil satu elemen secara acak dari elemen yang tersisa. Pengambilan elemen acak adalah berdasarkan elemen yang tersisa. Misalkan jika $m=10$, maka elemen acak yang boleh diambil adalah $10(array[0...9])$.
2. Tukar dengan elemen saat ini. Penukaran dilakukan dengan memasukkan elemen saat ini ke dalam suatu variabel baru bernama t . Elemen saat ini ($array[m]$) diisi nilai dari elemen acak tadi ($array[i]$) dan elemen acak ($array[i]$) diisi nilai dari variabel t .
3. Ulangi selama masih ada elemen yang tersisa

Tahap ketiga dilakukan pengulangan sebanyak 10 kali untuk mengacak urutan soal tersebut. Iterasinya bisa dilihat pada Tabel 1

Tabel 1. Contoh perhitungan dengan Metode Pengacakan Fisher-Yates

<i>m</i>	<i>i</i>	<i>array[i]</i>	<i>t=array[m]</i>	Array yang sudah diacak	Isi array setelah ditukar
10	6	6	9	6	0, 1, 2, 3, 4, 5, 9, 7, 8, 6
9	0	0	8	0, 6	8, 1, 2, 3, 4, 5, 9, 7, 0, 6
8	1	1	7	1, 0, 6	8, 7, 2, 3, 4, 5, 9, 1, 0, 6
7	2	2	9	2, 1, 0, 6	8, 7, 9, 3, 4, 5, 2, 1, 0, 6
6	2	9	5	9, 2, 1, 0, 6	8, 7, 5, 3, 4, 9, 2, 1, 0, 6
5	1	7	4	7, 9, 2, 1, 0, 6	8, 4, 5, 3, 7, 9, 2, 1, 0, 6
4	1	4	3	4, 7, 9, 2, 1, 0, 6	8, 3, 5, 4, 7, 9, 2, 1, 0, 6
3	1	3	5	3, 4, 7, 9, 2, 1, 0, 6	8, 5, 3, 4, 7, 9, 2, 1, 0, 6
2	1	5	5	5, 3, 4, 7, 9, 2, 1, 0, 6	8, 5, 3, 4, 7, 9, 2, 1, 0, 6
1	0	8	8	8, 5, 3, 4, 7, 9, 2, 1, 0, 6	8, 5, 3, 4, 7, 9, 2, 1, 0, 6

Hasil yang didapatkan adalah 8 5 3 4 7 9 2 1 0 6. Kolom *m* menunjukkan *index* elemen saat ini. Kolom *i* menunjukkan *index* dari elemen yang ditukar, nilai *i* tersebut diambil secara acak dari range yang diperbolehkan (*m*). Kolom *array[i]* menunjukkan nilai yang terdapat pada *array* ke-*i*. Kolom *t* menunjukkan nilai yang terdapat pada *array* ke-*m* (elemen saat ini). *array* yang sudah fix menunjukkan *array* yang sudah tidak boleh ditukar kembali nilainya, jumlahnya bertambah seiring bertambahnya iterasi. Kemudian kolom *array* yang sudah di *swap* menunjukkan isi *array* setelah elemennya ditukar/*swap* ada setiap iterasi. Untuk flowchart dari Metode Pengacakan Fisher-Yates dapat dilihat pada Gambar 5.



Gambar 4. Flowchart Metode Pengacakan Fisher-Yates

Flowchart pada Gambar 4 terlihat bahwa pengacakan selesai jika seluruh *array* telah diacak. Metode Pengacakan Fisher-Yates menghasilkan urutan *array* yang acak. Pseudocodenya dapat dilihat pada Tabel 2.

Tabel 2. *Pseudocode* Metode Pengacakan Fisher-Yates

No	Algoritma	<i>Pseudocode</i>
1	Ketikan masih ada elemen tersisa untuk diacak	While (m) {
2	Ambil elemen secara acak dari elemen yang tersisa	i=Math.floor(Math.random()*m- -);
3	Kemudian tukar dengan elemen saat ini	t = array [m]; array[m] = array[i]; array[i] = t; }

3.3. Android

Android adalah sistem operasi mobile berbasis Linux kernel. User interface Android berbasis *directmanipulation*. Android diprioritaskan untuk smartphone yang menggunakan *touchscreen* sebagai mobilitas utamanya dan komputer tablet. User Interface Android didasari atas interaksi manusia dengan komputer. Android menggunakan input sentuhan tangan manusia seperti mengusap, menyentuh, menngatup dan mengusap kearah berlawanan dengan carabersamaan [8]. *Game* Edukasi Rambu Lalu Lintas menggunakan Android sebagai platform dasar pembuatan *Game*.

3.4. Corona SoftwareDevelopmentKit (SDK)

Corona SDK dibuat untuk memungkinkan para programmer membangun game pada Platform iOS, Android, Windows Phone 8 dan Kindle yang dibuat oleh Walter Luh yang merupakan founder dari Corona Labs Inc pada Tahun 2008 [9]. Corona mendukung perangkat dengan processor minimal ARMv7 dan OS Android mulai dari versi 2.2 (Froyo) dan iOS mulai dari versi 4.3 [10].

3.5. Adobe Illustrator CS6

Adobe Illustrator adalah software desain berbasis vektor yang dikembangkan dan dipasarkan oleh Adobe System [11]. Penggunaan Adobe Illustrator CS6 pada *Game* Edukasi Rambu Lalu Lintas Berbasis Android adalah untuk membuat model gambar rambu lalu lintas, model mobil, model rambu jalan dan keseluruhan gambar yang ada di *Game* Rambu Lalu Lintas Berbasis Android.

3.6. Sublime Text Portable

Sublime Text adalah *cross-platfomeditor* teks dan kode program, dengan user interface pemrograman Aplikasi Python (API). Sublime didistribusikan sebagai perangkat lunak bebas. Sublime Text tidak memiliki dialog pengaturan grafis dan sepenuhnya dikonfigurasi dengan mengedit file teks. Sublime Text juga mendukung beberapa bahasa pemrograman, didukung dalam hal ini adalah dimengerti dan diterjemahkan menjadi text oleh Sublime Text. Misalnya pada C++, fungsi-fungsinya dimasukkan kedalam daftar fungsi dan kata-katanya berubah warna sesuai dengan makna kata tersebut di C++ [12].

4. Hasil dan Pembahasan

Hasil dan pembahasan berisikan tentang pembahasan dari aplikasi yang telah dirancang dan analisis hasil yang didapat setelah melakukan pengujian terhadap *Game* Edukasi Rambu Lalu Lintas Berbasis Android.

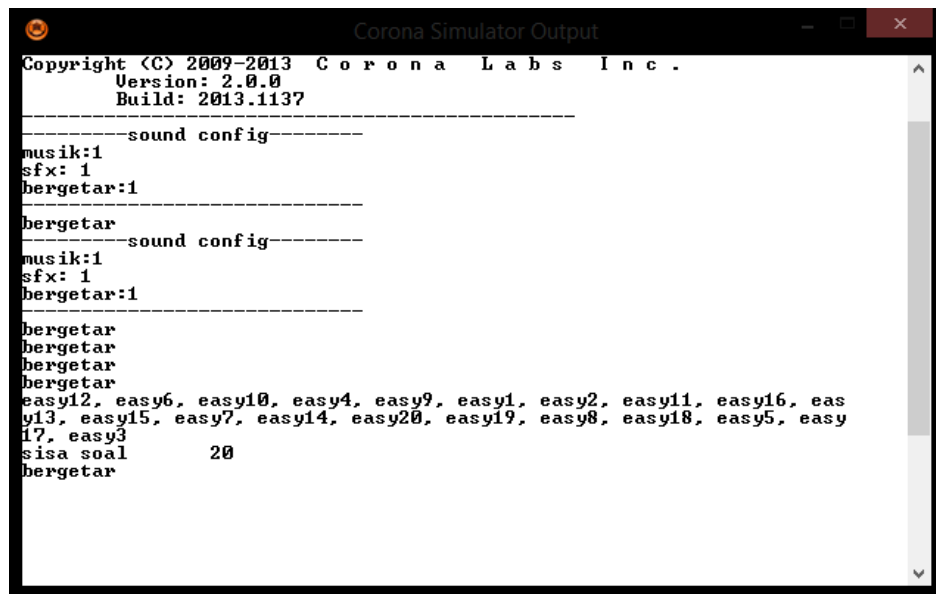
4.1. Hasil Perancangan

Scene Rambu mudah diisi dengan 20 soal dari total 30 soal. Pada tampilan *SceneLevel* Rambu mudah berisi 4 tombol yaitu tombol *replay*, tombol *pause*, tombol bantuan 50:50 dan bantuan *auto-completed*.



Gambar 5. SceneLevel Rambu Mudah

Berdasarkan Gambar 5, tampilan *SceneLevel* Rambu Mudah juga berisi penghitung waktu mundur dan skor yang di dapat. Gambar 6 menunjukkan hasil pengacakan menggunakan Metode Pengacakan Fisher-Yates.

Gambar 6. Implementasi Metode Pengacakan Fisher-Yates di *Level* Rambu Mudah

Pertanyaan *easy 12* berada di posisi pertama soal yang muncul, pertanyaan *easy 6* berada di posisi kedua soal yang muncul, dst. Setiap *user* bermain di *level* rambu mudah, maka pengacakan dilakukan. Urutan soal pada permainan pertama dan kedua dipastikan berbeda sesuai dengan hasil pengacakan menggunakan Metode Pengacakan Fisher-Yates. *User* diberikan bantuan 50:50 di *level* mudah ketika poin yang terkumpul lebih dari 200 poin. Ketika bantuan 50:50 dipilih, maka 2 jawaban salah dihilangkan dari permainan. Gambar 7 merupakan tampilan implementasi aturan 50:50 di *level* rambu mudah.



Gambar 7. Implementasi Aturan 50:50 Di Level Rambu Mudah

Ketika *user* menggunakan bantuan 50:50 maka skor *user* dikurangi sebesar 200 poin. *User* diberikan bantuan *Auto Completed* di *level* mudah ketika poin yang terkumpul lebih dari 1000 poin. Ketika bantuan *Auto Completed* dipilih, maka 3 jawaban salah dihilangkan dari permainan. Gambar 8 merupakan tampilan implementasi aturan *Auto Completed* di *level* rambu mudah.



Gambar 8. Implementasi Aturan *Auto-Completed* di *Level* Rambu Mudah

Ketika *user* menggunakan bantuan *Auto-Completed* maka skor *user* dikurangi sebesar 1000 poin. *SceneStageCompletedLevel* Rambu mudah terbuka jika *user* telah menjawab keseluruhan soal atau jika *user* tidak menjawab sama sekali keseluruhan soal atau jika *user* salah menjawab disaat pertama kali soal diberikan. Tampilan *Scene Stage Completed Level* Rambu mudah terdapat pada Gambar 9.



Gambar 9. Scene Stage Completed Level Rambu mudah

Merujuk pada Gambar 9 terdapat dua tombol yaitu tombol *home* dan *replay*. Tombol *home* digunakan untuk kembali ke menu sebelumnya. Tombol *replay* digunakan untuk bermain kembali ke *level* rambu mudah. Scene Galeri terlihat setelah user menekan tombol "Galeri" pada scene main menu. Tampilan Scene galeri terlihat seperti Gambar 10.



Gambar 10. Tampilan Scene Galeri (1)

Setelah *user* bermain pada *level* rambu lalu lintas dan berhasil menjawab benar, maka tampilan scene galeri berubah menjadi seperti Gambar 11.



Gambar 11. Tampilan Scene Galeri (2)

Ketika *user* menyentuh gambar rambu dilarang parkir, maka tampil *scene* penjelasan lebih lanjut mengenai rambu dilarang parkir seperti yang terlihat di Gambar 12.



Gambar 12. Tampilan Scene Galeri Dilarang Parkir

Ketika *user* menyentuh gambar rambu berlambang 5T pada Gambar 11, maka tampil *scene* penjelasan lebih lanjut mengenai rambu 5T seperti yang terlihat di Gambar 13.

4.2. Pembahasan

Kerja aplikasi dihitung melalui hasil pengujian terhadap fitur aplikasi pada konten yang disajikan. Perhitungan dilakukan dengan mengumpulkan data hasil pengujian aplikasi yang dilakukan oleh responden melalui nilai kualitatif yang disajikan dalam tabel berikut.

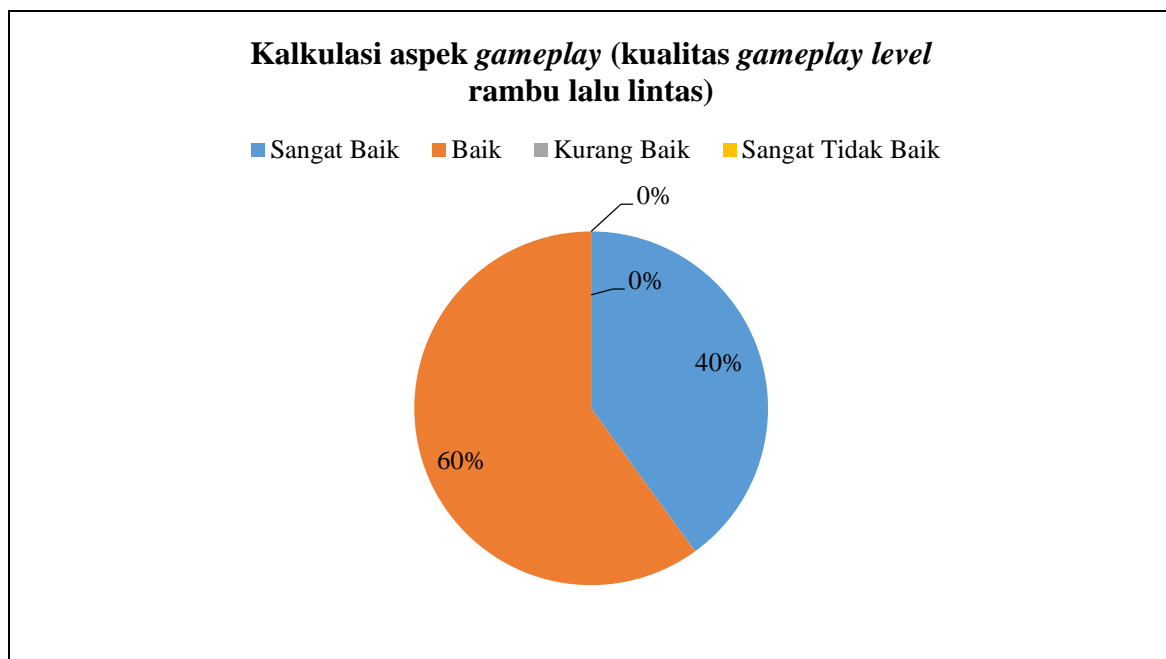
Tabel 3 Hasil Pengujian Konten Aplikasi

Nilai	Jumlah Responden (n)	Perhitungan Hasil $\frac{n}{t} \times 100\%$	Presentase Hasil
Sangat Tidak Baik	0	$\frac{0}{20} \times 100\%$	0%
Tidak Baik	0	$\frac{0}{20} \times 100\%$	0%
Baik	12	$\frac{12}{20} \times 100\%$	60%
Sangat Baik	8	$\frac{8}{20} \times 100\%$	40%

Tabel 3 diatas menjelaskan bahwa dapat ditarik kesimpulan sebagai berikut:

1. Jumlah responden sebanyak 20 orang.
2. Responden yang memilih setuju sebanyak 12 orang dan memiliki persentase sebesar $(12/20) \times 100\% = 60\%$
3. Responden yang memilih sangat setuju sebanyak 8 orang dan memiliki persentase sebesar $(8/20) \times 100\% = 40\%$

Berdasarkan persentase diatas, maka gambaran pada diagram dapat dilihat sebagai berikut:



Gambar 14. Kalkulasi Aspek *Gameplay* (Kualitas *Gameplay level* Rambu Lalu Lintas)

Gambar 14 aspek *gameplay* (kualitas *gameplay level* rambu lalu lintas), sebagian responden memberikan respon persentase baik sebanyak 60% dan sangat baik 40%. Berdasarkan hasil persentase yang sudah diperoleh, pilihan persentase terbanyak pada pilihan baik sehingga dapat disimpulkan bahwa konten atau isi *game level* rambu lalu lintas sudah sesuai.

5. Kesimpulan

Game Edukasi Rambu Lalu Lintas Berbasis Android adalah *game* yang ber-genre *Multiple-Choice*. *User* diberikan 4 jawaban untuk setiap soal pada game yang dihasilkan, dimana hanya ada 1 jawaban benar. *User* harus menjawab 20 soal setiap bermain yang diacak menggunakan Metode Pengacakan Fisher-Yates. Metode Pengacakan Fisher-Yates digunakan untuk mengacak soal yang diberikan kepada user tiap kali bermain, sehingga soal yang muncul tidaklah sama urutan kemunculannya. Berdasarkan aspek *gameplay* yang didapat dari penilaian responden, sebanyak 40% responden menilai Game Edukasi Rambu Lalu Lintas Berbasis Android sangat baik dan 60% responden menilai baik.

Daftar Pustaka

- [1] <http://www.bps.go.id/linkTabelStatis/view/id/1413>, diakses pada tanggal 4 Mei 2014.
- [2] <http://nasional.sindonews.com/read/816365/14/polri-akui-angka-pelanggaran-meningkat-tajam-1386847978>, diakses pada tanggal 4 Mei 2014.
- [3] http://kemhubri.dephub.go.id/knkt/ntsc_home/Media_Release/Media_Release_KNKT_2013.pdf, diakses pada tanggal 4 Mei 2014.
- [4] Antony Susanto, Hengky Honggo. Perancangan Ujian Online pada STMIK GI MDP Berbasis Web. 2013:6-7.
- [5] Ade-Ibijola, Abejide Olu. A Simulated Enhancement of Fisher-Yates Algorithm for Shuffling in Virtual Card Games using Domain-Specific Data Structures. *International Journal of Computer Applications*. 2012; 54(11): 24-28.
- [6] <http://hubdat.dephub.go.id/km/tahun-1993/121-km-61-tahun-1993>, diakses pada tanggal 17 November 2014.
- [7] Raharjo, R. Tertib Berjalan Lintas. Yogyakarta: Shafa Media. 2014.
- [8] https://en.wikipedia.org/wiki/Android_%28operating_system%29, diakses pada tanggal 17 November 2014.
- [9] https://en.wikipedia.org/wiki/Corona_%28software%29, diakses pada tanggal 17 November 2014.
- [10] <https://docs.coronalabs.com/guide/start/systemReqs/index.html>, diakses pada tanggal 17 November 2014.
- [11] https://en.wikipedia.org/wiki/Adobe_Illustrator, diakses pada tanggal 17 November 2014.
- [12] <http://www.sublimetext.com/>, diakses pada tanggal 17 November 2014.