

SIM SIMULATOR

**Disusun untuk memenuhi tugas mata kuliah
IF3111 Pengembangan Aplikasi pada Platform Khusus**

Kelompok 9 – TheSIM

Oktavianus Handika – 13515035

Felix Limanta – 13515065

Rionaldi Chandraseta – 13515077



**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2017**

Bab 1

Pendahuluan

1.1 Latar Belakang

Di Indonesia, Surat Izin Mengemudi (SIM) adalah bukti registrasi dan identifikasi yang diberikan oleh Polri kepada seseorang yang telah memenuhi persyaratan administrasi, sehat jasmani dan rohani, memahami peraturan lalu lintas dan terampil mengemudikan kendaraan bermotor. Setiap orang yang mengemudikan Kendaraan Bermotor di Jalan wajib memiliki Surat Izin Mengemudi sesuai dengan jenis Kendaraan Bermotor yang dikemudikan.

Untuk mendapatkan Surat Izin Mengemudi sebagaimana dimaksud dalam Pasal 77, setiap orang harus memenuhi persyaratan usia, administratif, kesehatan, dan lulus ujian. Syarat lulus ujian meliputi:

1. ujian teori;
2. ujian praktik; dan/atau
3. ujian keterampilan melalui simulator.

Per tanggal 1 Januari 2017, biaya administrasi pembuatan SIM A baru adalah Rp120.000,00. Meskipun demikian, mengingat sulitnya ujian praktik pembuatan SIM A baru, terdapat kemungkinan besar orang yang baru belajar menyetir mobil akan mengambil ujian ini berkali-kali. Misalnya saja, beberapa tahun yang lalu di Sragen, jumlah pemohon yang lulus ujian pembuatan SIM hanya 17%, sedangkan 83% sisanya gagal.

Untuk menghindari membuang waktu untuk mengambil ujian pembuatan SIM A baru berkali-kali, terdapat calo-calo yang menawarkan jasa pembuatan SIM A tanpa tes. Untuk kemudahan ini, harga calo-calo tersebut juga jauh lebih mahal dari biaya administrasi pembuatan SIM A secara resmi, yaitu sekitar Rp600.000,00 sampai Rp700.000,00. Menurut survei yang diambil untuk SIM C, sekitar 50% pemohon memperoleh SIM dengan calo.

Jika calon pengambil ujian tidak ingin membayar mahal untuk tidak mengambil ujian dan/atau tidak ingin melanggar undang-undang yang berlaku, pengambil ujian harus menjalani latihan keras agar cukup sekali saja mengambil ujian. Meskipun demikian, untuk menjalani latihan, tentunya harus ada mobil yang tersedia untuk dikemudikan calon pengambil ujian. Tidak semua orang, terutama orang dari kelas menengah ke bawah, memiliki akses yang mudah terhadap mobil yang bisa digunakan untuk latihan. Selain itu, latihan dengan mobil di dunia nyata juga memiliki risiko kecelakaan.

Untuk meminimalkan biaya dan risiko yang berpotensi dikeluarkan, harus terdapat suatu lingkungan latihan untuk pelamar SIM A dapat mengasah kemampuan menyetirnya. Lingkungan latihan tersebut harus bebas dari risiko kecelakaan dan tidak memerlukan pelamar merogoh kantongnya. Solusi yang kami tawarkan adalah sistem bernama SIM Simulator. Sistem ini terdiri dari aplikasi Android untuk menyimulasikan pengambilan ujian teori, sebuah simulator *online* untuk menyimulasikan pengambilan ujian praktik, dan sebuah model mobil untuk memberikan umpan balik yang lebih responsif ke *user* sistem.

1.2 Tujuan dan Manfaat

Pengembangan SIM Simulator bertujuan untuk membantu masyarakat yang ingin berlatih untuk menghadapi ujian SIM A yang dapat dikatakan lebih ketat dari tahun-tahun sebelumnya. Pelamar SIM A pertama-tama harus menyelesaikan ujian teori sebelum melanjutkan ke ujian praktik. Ujian praktik saat ini menempatkan pelamar ke dalam beberapa skenario berbeda yang cukup sulit untuk diselesaikan. Dengan adanya sistem SIM Simulator, diharapkan masyarakat yang ingin berlatih untuk lulus ujian SIM A dapat melakukannya dengan mudah melalui sistem ini.

Manfaat dari adanya sistem SIM Simulator adalah mempermudah para pelamar SIM A yang mengalami kesulitan dalam ujian baik teori maupun praktik. Dari segi keamanan, penggunaan SIM Simulator juga mengeliminasi kemungkinan terjadinya kecelakaan yang mungkin terjadi ketika berlatih menggunakan mobil di dunia nyata. Selain itu, dari segi lingkungan, juga mengurangi konsumsi BBM (Bahan Bakar Minyak) yang digunakan untuk berlatih. Ditambah lagi dengan berlatih menggunakan sistem ini, para pelamar SIM A dapat menghemat biaya yang mungkin dikeluarkan untuk fasilitas latihan.

Bab 2

Deskripsi Sistem secara Keseluruhan

SIM Simulator merupakan alat simulasi untuk membantu masyarakat yang ingin mengikuti ujian pengambilan SIM (Surat Izin Mengemudi) kategori A (SIM A), baik dalam ujian teori maupun ujian praktik. Sistem ini merupakan gabungan dari tiga subsistem yang dibangun dengan tiga platform berbeda, yaitu Android, Unity, dan Arduino.

Dalam subsistem Android, sistem dapat menentukan kemampuan teori user dalam berlalu lintas. Nilai yang didapat oleh user dalam simulasi ujian teori akan dimasukkan ke dalam cloud server agar dapat diakses oleh subsistem lainnya. Selain itu, sistem juga dapat menentukan apakah user dinyatakan layak mendapatkan SIM A berdasarkan gabungan nilai ujian teori dan nilai ujian praktik yang dilakukan dengan Unity.

Dalam subsistem Unity, sistem dapat menyimulasikan ujian praktik untuk pengambilan SIM A. Ketika user ingin melakukan ujian praktik, sistem akan memeriksa terlebih dahulu apakah nilai ujian teori user mencukupi. Hanya user yang lulus ujian teori yang dapat mengambil ujian praktik dalam Unity. Kemajuan user akan dicatat dan dimasukkan ke dalam cloud server.

Dalam subsistem Arduino, sistem dapat menyimulasikan pergerakan mobil virtual di subsistem Unity pada dunia nyata. Sebuah model mobil yang ditenagai Arduino dapat dikontrol melalui model mobil di Unity untuk simulasi yang lebih nyata, di mana ketika user menggerakkan mobil virtual di Unity, model mobil fisik Arduino juga akan ikut bergerak mengikuti pergerakan mobil virtual tersebut. Sesuai dengan mobil virtual di Unity, model mobil tersebut dapat bergerak maju dan mundur, serta dapat berbelok ke kiri dan ke kanan. Untuk keamanan, model mobil dilengkapi sensor ultrasonik untuk mendeteksi jarak model mobil ke lingkungannya, serta limit switch untuk mendeteksi jika model mobil mengalami benturan dengan lingkungannya.

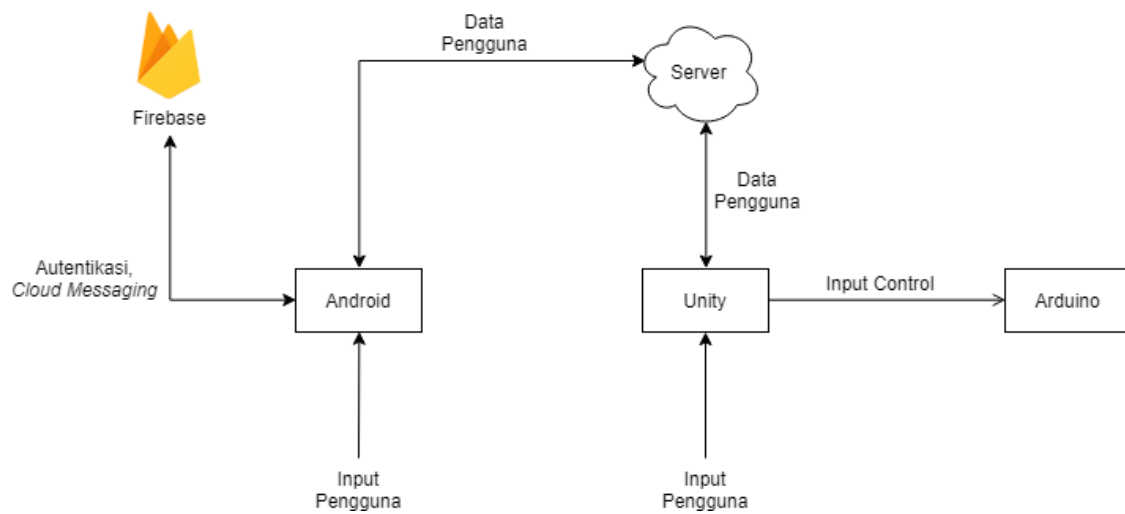
Bab 3

Rancangan Solusi

Dalam subsistem Android, sistem dapat menentukan kemampuan teori user dalam berlalu lintas. Nilai yang didapat oleh user dalam simulasi ujian teori akan dimasukkan ke dalam cloud server agar dapat diakses oleh subsistem lainnya. Selain itu, sistem juga dapat menentukan apakah user dinyatakan layak mendapatkan SIM A berdasarkan gabungan nilai ujian teori dan nilai ujian praktik yang dilakukan dengan Unity.

Dalam subsistem Unity, sistem dapat menyimulasikan ujian praktik untuk pengambilan SIM A. Ketika user ingin melakukan ujian praktik, sistem akan memeriksa terlebih dahulu apakah nilai ujian teori user mencukupi. Hanya user yang lulus ujian teori yang dapat mengambil ujian praktik dalam Unity. Kemajuan user akan dicatat dan dimasukkan ke dalam cloud server.

Dalam subsistem Arduino, sistem dapat menyimulasikan pergerakan mobil virtual di subsistem Unity pada dunia nyata. Sebuah model mobil yang ditenagai Arduino dapat dikontrol melalui model mobil di Unity untuk simulasi yang lebih nyata, di mana ketika user menggerakkan mobil virtual di Unity, model mobil fisik Arduino juga akan ikut bergerak mengikuti pergerakan mobil virtual tersebut. Sesuai dengan mobil virtual di Unity, model mobil tersebut dapat bergerak maju dan mundur, serta dapat berbelok ke kiri dan ke kanan. Untuk keamanan, model mobil dilengkapi sensor ultrasonik untuk mendeteksi jarak model mobil ke lingkungannya, serta limit switch untuk mendeteksi jika model mobil mengalami benturan dengan lingkungannya.



Gambar 3.0.1: Diagram sistem secara keseluruhan

Bab 4

Rincian Subsistem

4.1 Subsistem Android

4.1.1 Tujuan dan Fungsi

Tujuan dari subsistem Android pada Sim Simulator adalah menyimulasikan ujian teori dalam proses pembuatan SIM A. Aplikasi Android akan menyajikan sejumlah pertanyaan kepada *user*. Aplikasi kemudian akan menentukan apakah *user* lulus ujian teori atau tidak berdasarkan nilai yang diperolehnya.

Selain itu, terdapat juga fitur *leaderboard*, di mana nilai dari semua *user* dicatat untuk dibandingkan dengan sesama, baik berdasarkan nilai tertinggi ataupun waktu tercepat.

4.1.2 Daftar Fitur

1. **Ujian dalam bentuk pilihan ganda**
Simulasi ujian teori pada aplikasi dibuat dalam bentuk soal pilihan ganda dengan jumlah soal yang dapat diatur oleh *user*.
2. ***Login* dan *register***
Aplikasi menyediakan *register* akun untuk *user* agar profilnya tercatat dalam *database server* aplikasi.
3. **Logo aplikasi berubah sudut sebagai *easter egg* aplikasi**
Ketika *user* melakukan klik pada tulisan “SIM SIMULATOR” sebanyak 5 kali, maka *easter egg* akan muncul berupa logo melakukan rotasi berdasarkan *accelerometer sensor*.

4. Perubahan warna latar pada aplikasi sebagai *easter egg* aplikasi

Warna latar pada aplikasi berubah ketika *user* meng-*hold* logo pada *main menu*, *navigation drawer*, ataupun pada *fragment help*.

5. Mendeteksi lokasi *user*

Aplikasi dapat mendeteksi lokasi *user* berada dengan menggunakan sensor GPS.

6. Mencari kantor polisi terdekat dengan *user*

Aplikasi dapat menentukan kantor polisi yang terdekat dengan *user* berdasarkan letak *user* berada.

4.1.3 Daftar Sensor yang Digunakan**1. *Accelerometer sensor***

Accelerometer sensor digunakan untuk mengukur percepatan gerak perangkat. Percepatan yang terukur pada sensor digunakan pada *easter egg* aplikasi.

2. GPS

GPS digunakan untuk menentukan letak pengguna perangkat berada. Letak *user* ini akan dikirim ke Google Location Service untuk diproses dalam mencari tempat ujian mengemudi yang berada pada jarak tertentu dengan *user*.

4.1.4 Daftar Service yang Digunakan**1. Google Location Service**

Aplikasi menggunakan Google Location Service untuk menampilkan lokasi *user* berada dan kantor polisi terdekat dengan *user*.

2. Notification Service

Aplikasi menampilkan notifikasi kepada *user* ketika *user* telah menyelesaikan ujian dan admin memberi pesan kepada *user*.

4.2 Subsistem Unity**4.2.1 Tujuan dan Fungsi**

Tujuan dari subsistem Unity pada SIM Simulator adalah menyimulasikan ujian praktik dalam proses pembuatan SIM A. *User* dapat mengemudikan sebuah mobil pada aplikasi kami. Selain itu, tersedia beberapa level berbeda di mana *user* harus berhasil mengendalikan mobil melalui berbagai rintangan. Rintangan-rintangan ini disesuaikan dengan rintangan pada ujian mengemudi sesungguhnya.

4.2.2 Daftar Fitur

1. Autentikasi *user*

Program menyediakan fitur autentikasi *user* dan dapat mengecek apakah *user* telah lulus ujian teori. Jika *user* lulus ujian teori sebelumnya, maka *user* dapat menjalankan simulasi. *user*.

2. Simulasi mobil

Program dapat menyimulasikan sebuah mobil dengan pergerakan fisik mobil, suara mesin, rotasi setir, serta spion. Selain itu *environment* juga mendukung gravitasi sehingga pergerakan mobil dalam simulasi ini mendekati kehidupan nyata.

3. Rintangan ujian

Program menyediakan berbagai rintangan yang relevan dengan ujian SIM A dan latihan mengemudi pada umumnya.

4. Deteksi *goal* dan *failure*

Program dapat mendeteksi apakah *user* berhasil mengemudikan mobil hingga mencapai tempat yang benar dan dinyatakan lulus suatu rintangan. Selain itu, program dapat mendeteksi tabrakan yang terjadi antara *user* dengan objek lain. Tabrakan ini dapat mengakibatkan *user* dinyatakan gagal dalam menyelesaikan rintangan.

4.2.3 Ide Dasar

Inspirasi pembuatan simulasi menggunakan Unity ini adalah karena munculnya pemberitaan mengenai sulitnya ujian SIM A, terutama pada bagian mengemudikan mobil mundur pada rintangan zig-zag tanpa mengenai *cone*. Pada dasarnya, saat ujian SIM A sesungguhnya pengemudi harus melalui rintangan dan menuju *goal* tanpa melakukan kesalahan. Ketika pengemudi melakukan satu kesalahan saja maka pengemudi dapat dinyatakan tidak lulus ujian SIM A. Hal ini membuat kami ingin meniru rintangan pada ujian SIM A dan membuat simulasinya agar dapat berlatih untuk ujian sesungguhnya.

4.2.4 *Gameplay*

Permainan yang kami desain adalah *first-person driving simulator*, dimana *user* ditempatkan di dalam sebuah mobil dan dapat mengendalikan mobil tersebut dalam tampilan *first-person*.

Pada setiap *scene*, *user* dinyatakan berhasil ketika dapat mengemudikan mobil menuju setiap *checkpoint* pada suatu rintangan. Pada sebagian besar *scene*, *user* harus memarkirkan mobil tepat di dalam *checkpoint* yang tersedia secara tepat agar dinyatakan berhasil.

User dinyatakan gagal ketika menabrak benda-benda (*obstacles*) yang tersebar pada setiap *scene*. Tidak ada sistem *scoring* dalam simulasi ujian mengemudi ini, sistem yang digunakan

dalam simulasi ini adalah sistem nyawa. Ketika *user* melakukan kesalahan dalam suatu rintangan maka *user* dinyatakan gagal.

Pengendalian mobil dilakukan oleh *user* dengan menggunakan tombol WASD maupun tanda panah. Selain itu, untuk rem tangan dapat digunakan dengan menekan tombol “Space”. *User* juga dapat melihat sekelilingnya dengan menggerakkan *mouse*. Terdapat beberapa perintah lain yang dapat diaktifkan ketika *user* telah berhasil atau gagal dalam suatu rintangan, seperti tombol R untuk mengulang *scene*.

4.2.5 Assets

1. **Vehicle Tools** <https://www.assetstore.unity3d.com/en/#!/content/83660>
2. **Edy’s Vehicle Physics** <https://drive.google.com/file/d/0B27xYiUULyKlSHppV0lTRXp1Q1E/view>
3. **Stylized and Modular City** <https://www.assetstore.unity3d.com/en/#!/content/67029>
4. **Free HDR Sky** <https://www.assetstore.unity3d.com/en/#!/content/61217><https://www.assetstore.unity3d.com/en/#!/content/61217>
5. **MarkLight** <https://www.assetstore.unity3d.com/en/#!/content/37466>
6. **Street Asset** <https://www.assetstore.unity3d.com/en/#!/content/4565>

4.3 Subsistem Arduino

4.3.1 Tujuan dan Fungsi

Tujuan dari subsistem Arduino pada SIM Simulator adalah menyimulasikan pergerakan mobil virtual di subsistem Unity pada dunia nyata. Sebuah model mobil yang ditenagai Arduino dapat dikendalikan melalui model mobil virtual di Unity untuk simulasi yang lebih nyata.

4.3.2 Daftar Fitur

1. **Bergerak**
Sistem dapat bergerak maju, mundur, dan berbelok sesuai dengan masukan *user*.

2. Deteksi Maju/Mundur Mobil

Deteksi pergerakan mobil apakah sedang maju, diam, atau mundur. Hasil pergerakan akan ditampilkan di 7-segment.

3. Deteksi Jarak

Deteksi jarak pada model mobil menggunakan sensor ultrasonik. Data dari deteksi jarak ini akan dikirim ke LCD untuk ditampilkan nilainya.

4. Deteksi Tabrakan

Deteksi apakah bagian depan mobil bertabrakan dengan sesuatu, menggunakan *limit switch*. Mobil akan berhenti total ketika bertabrakan.

4.3.3 Komponen yang Digunakan beserta Fungsinya**1. Arduino Uno**

Mikrokontroler sebagai penggerak sistem

2. Sensor Ultrasonik HC-SR04

Mendeteksi jarak sistem ke lingkungan di depannya

3. *Limit Switch*

Mendeteksi apakah mobil bersentuhan dengan suatu objek atau tidak

4. Resistor 10000 Ω

Bekerja sebagai *pulldown resistor*, sehingga arus tidak semua mengalir ke *ground*

5. LCD 16x2 *Blue on White*

Menampilkan jarak model mobil dengan objek di depannya serta kecepatan gerak kedua motor (skala 0-255)

6. I2C *Backpack* untuk LCD

Menyambungkan LCD ke Arduino dengan I2C

7. *Common Anode 7-segment*

Menampilkan mobil dalam keadaan maju, mundur, atau berhenti. Ketika mobil maju, maka 7-segment akan menampilkan huruf 'd'. Ketika mobil mundur, 7-segment akan menampilkan huruf 'r' dan saat mobil berhenti, 7-segment akan menampilkan huruf 'P'.

8. *Serial-In/Parallel-Out* IC 74HC595

Mengonversi data dari Arduino ke format parallel, yang diteruskan ke 7-segment

9. Resistor 330 Ω

Membatasi arus yang lewat ke setiap segmen pada 7-segment.

10. Resistor 1000 Ω

Bekerja sebagai *pullup resistor* untuk mem-*buffer* perubahan arus dan/atau tegangan pada VCC.

11. **Motor DC + Roda (1 pasang)**
Menggerakkan model mobil
12. **Roda Bebas**
Menyangga bagian depan sistem agar dapat bergerak bebas
13. **H-bridge Motor Driver IC L293D**
Memungkinkan pergerakan 2 motor, masing-masing dalam dua arah (CW/CCW), menggunakan 6 pin dan 1 sumber daya
14. **Kabel *jumper* (MM, MF, FF)**
Menghubungkan komponen-komponen ke Arduino dan *breadboard*
15. **Breadboard**
Sebagai tempat meletakkan dan menghubungkan komponen-komponen
16. **Kabel USB Type-B MM**
Menghubungkan komputer dengan Arduino dan sebagai sumber daya Arduino
17. **Baterai 9V**
Sumber daya motor

4.3.4 Mekanisme Komunikasi dengan Platform Lain

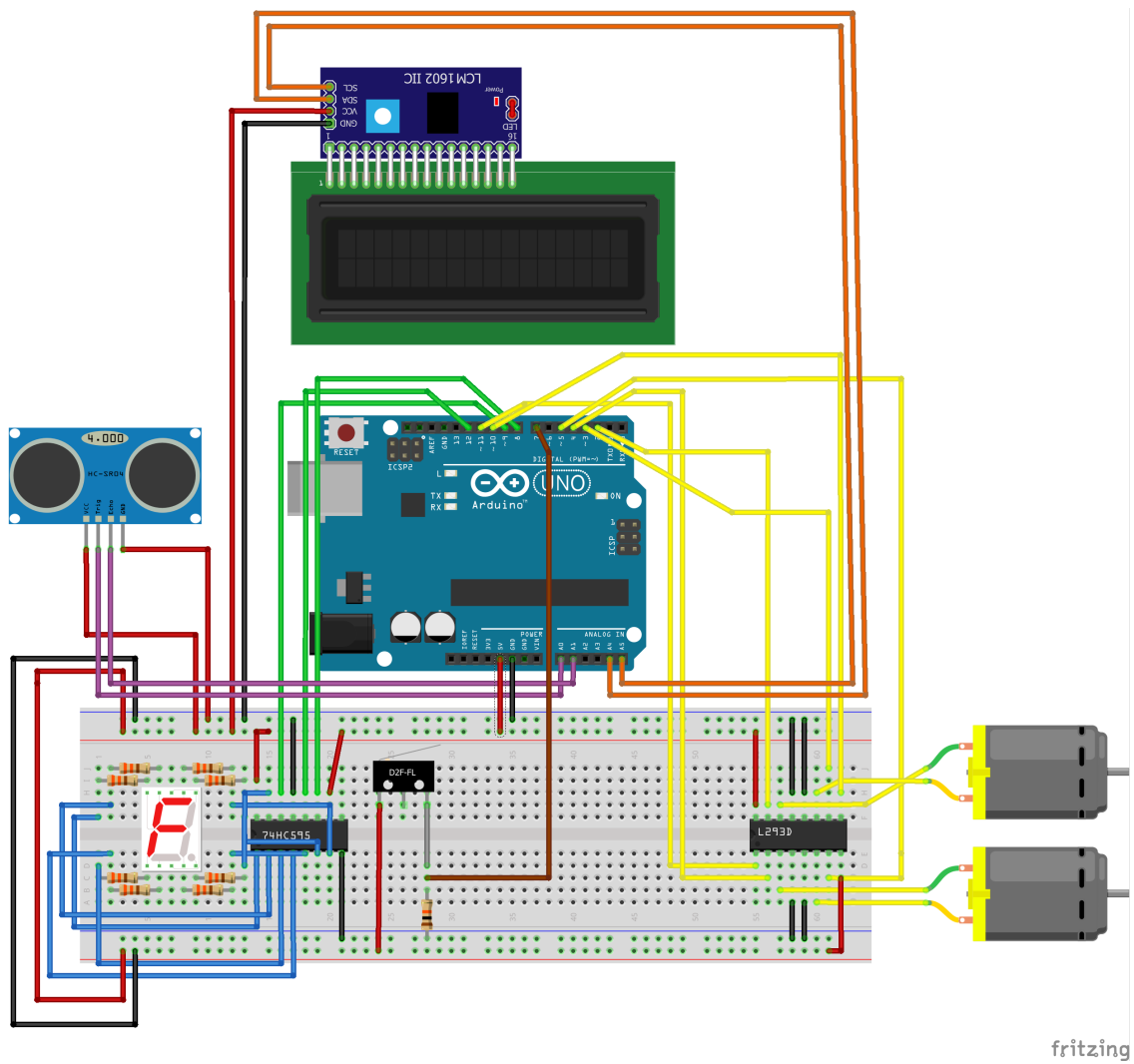
Ketika *user* menggerakkan mobil virtual di Unity, model mobil fisik Arduino juga akan ikut bergerak mengikuti pergerakan mobil virtual tersebut. Sesuai dengan mobil virtual di Unity, model mobil tersebut dapat bergerak maju dan mundur, serta dapat berbelok ke kiri dan ke kanan sesuai pergerakan mobil yang dikendalikan di platform Unity.

Ini dilakukan dengan komunikasi serial antara subsistem Unity dan subsistem Arduino. Ketika tombol *keyboard* untuk menggerakkan mobil di Unity ditekan ('W', 'A', 'S', 'D'), Unity akan mengirim pesan *serial* berupa sebuah *byte* ke *serial port* di mana Arduino berada. Arduino kemudian akan menerjemahkan pesan *byte* tersebut menjadi arahan gerak yang sesuai, serta mengubah 7-segment sesuai dengan arah gerak tersebut.

Saat Unity mendeteksi bahwa salah satu tombol *keyboard* yang sesuai ditekan, Unity akan mengirim satu *byte* untuk menandakan mulai gerakan. Arduino akan terus bergerak dengan arah yang sesuai dan kecepatan maksimum (PWM 255 *full duty cycle*). Ketika tombol di *keyboard* dilepas, Unity akan mengirim *byte* yang menandakan bahwa pergerakan berakhir. Arduino akan mulai berhenti perlahan-lahan dengan mengurangi kecepatan PWM motor hingga 0.

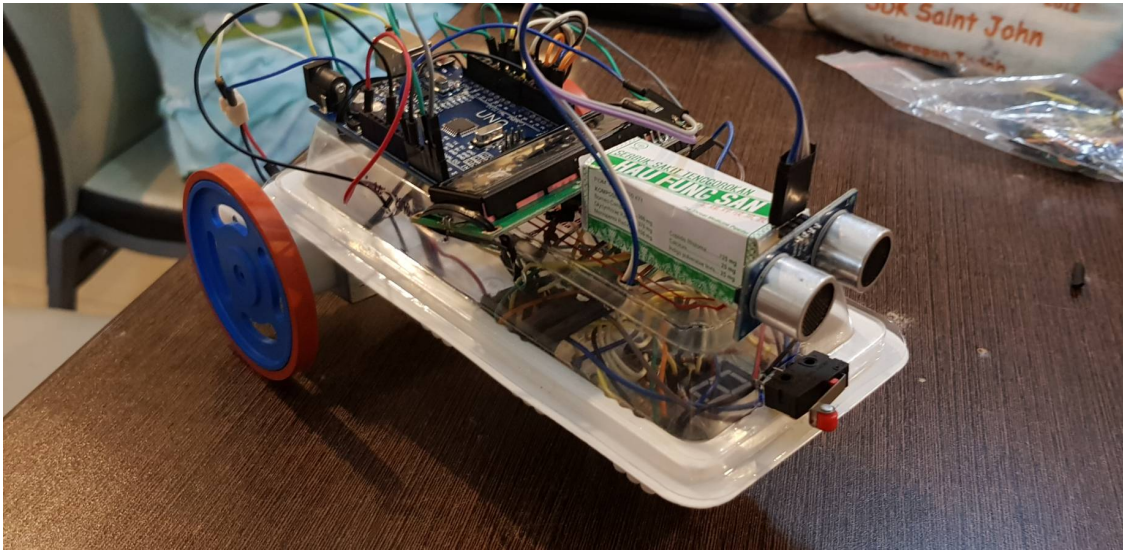
Arduino tidak akan menerima perintah apapun ketika *limit switch* ditekan. Selain itu, mobil hanya dapat digerakan pada *sandbox* di menu utama Unity.

4.3.5 Board Diagram Sistem

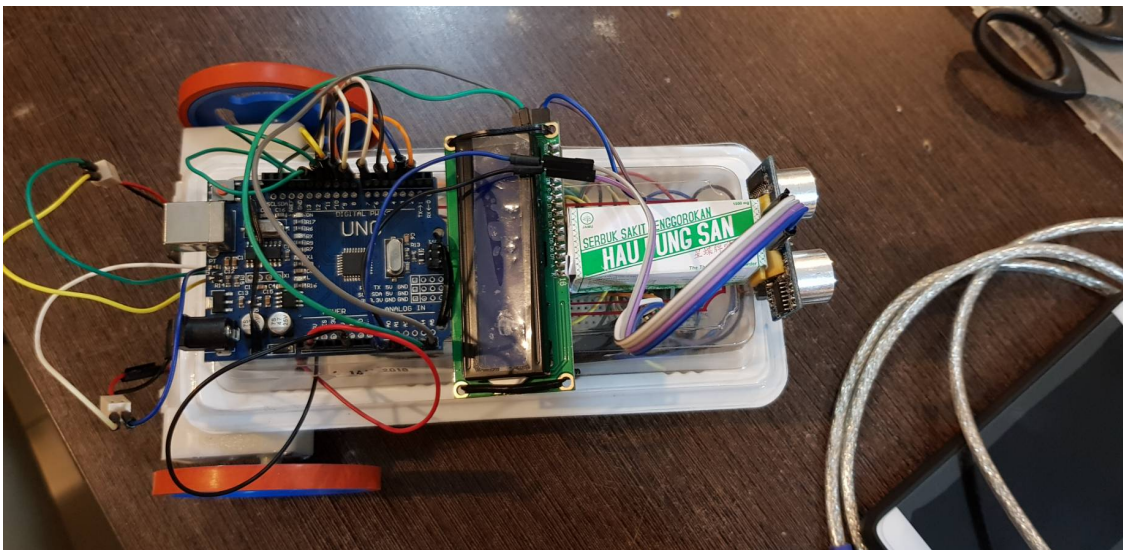


Gambar 4.3.1: Board Diagram sistem Arduino SIM Simulator

4.3.6 Foto Sistem



Gambar 4.3.2: Foto sistem Arduino SIM Simulator



Gambar 4.3.3: Tampak atas Arduino SIM Simulator

Bab 5

Log Activity

5.1 Pembagian Tugas

5.1.1 Android

No.	NIM	Nama	Tugas
1	13515035	Oktavianus Handika	Membuat <i>navigation drawer</i> pada menu utama, mengimplementasi SharedPreferences untuk pengaturan banyak soal dalam ujian, membuat <i>activity</i> pada <i>test</i> , <i>settings</i> , dan <i>result</i> , pembuatan laporan.
2	13515065	Felix Limanta	Pembuatan API server ke Heroku, basis data, koneksi aplikasi dengan basis data sistem, membuat AsyncTaskLoader, kelas NetworkUtils, pembuatan kelas adapter, membuat fungsionalitas utama untuk ujian teori, pembuatan laporan.
3	13515077	Rionaldi Chandraseta	Implementasi <i>easter egg</i> dan <i>geomagnetic field</i> , membuat autentikasi pengguna menggunakan Firebase, membuat <i>push notification</i> , desain logo aplikasi, implementasi Google Location Service untuk Google Maps API, pembuatan laporan.

5.1.2 Unity

No.	NIM	Nama	Tugas
1	13515035	Oktavianus Handika	Desain <i>scene</i> untuk ujian SIM, menambahkan <i>props</i> , <i>skybox</i> , menampilkan pesan
2	13515065	Felix Limanta	Konsep <i>scene</i> untuk ujian SIM, pengaksesan basis data di <i>server</i> , menu utama, kontrol kamera
3	13515077	Rionaldi Chandraseta	Pembuatan mobil, rotasi setir, cermin, <i>checkpoint</i> , logika untuk keberhasilan dan kegagalan ujian

5.1.3 Arduino

No.	NIM	Nama	Tugas
1	13515035	Oktavianus Handika	Laporan
2	13515065	Felix Limanta	Merancang dan merakit Arduino
3	13515077	Rionaldi Chandraseta	Membuat diagram <i>fritzing</i> dan kontroler Arduino

5.2 Log Activity

5.2.1 Android

No.	Tanggal	Anggota	Kegiatan
1	10 September 2017	13515035, 13515065, 13515077	Membahas dan membuat rancangan awal sistem.
2	18 September 2017	13515035, 13515065, 13515077	Melakukan revisi proposal sesuai dengan feedback asisten.
3	25 September 2017	13515035, 13515065, 13515077	Membuat <i>project</i> dan <i>repository</i> , menambahkan fungsionalitas autentikasi pengguna pada aplikasi.
4	26 September 2017	13515035, 13515065, 13515077	Menyatukan autentikasi pengguna menjadi satu aktivitas dengan banyak Fragment, memastikan data tidak hilang saat orientasi layar berubah.
5	27 September 2017	13515035, 13515065, 13515077	Membuat menu utama untuk aplikasi dengan menggunakan NavigationDrawer, menambahkan Activity awal untuk aplikasi, pembuatan basis data untuk <i>server</i> .
6	28 September 2017	13515035, 13515065, 13515077	Memastikan <i>user</i> harus <i>login</i> sebelum dapat menggunakan aplikasi, <i>routing</i> pada <i>server</i> , <i>deploy server</i> ke Heroku.
7	29 September 2017	13515035, 13515065, 13515077	Memperbaiki pengambilan gambar dari <i>server</i> , mencoba membuat <i>services</i> untuk aplikasi.
8	30 September 2017	13515035, 13515065, 13515077	Menambahkan <i>notification service</i> dan <i>location service</i> , menambahkan <i>easter egg</i> , menambahkan Preferences untuk <i>user</i> , mengubah tampilan, mengubah cara <i>hosting</i> gambar menjadi menggunakan Imgur.
9	1 Oktober 2017	13515035, 13515065, 13515077	Membenarkan <i>bug</i> pada <i>location service</i> , menambahkan fungsionalitas utama, NetworkUtils untuk mengakses basis data, melengkapi <i>fragment</i> yang masih kosong (Help, Profile), membuat laporan
10	25 November 2017	13515035, 13515065, 13515077	Memperbaiki hubungan aplikasi dengan <i>server</i> , memperbaiki <i>routing</i> dan operasi lainnya pada <i>server</i>

5.2.2 Unity

No.	Tanggal	Anggota	Kegiatan
1	23 Oktober 2017	13515035, 13515065, 13515077	Membahas dan membuat rancangan awal sistem.
2	24 Oktober 2017	13515077	Membuat dasar mobil untuk dikendalikan oleh <i>user</i> .
3	25 Oktober 2017	13515077	Memodifikasi mobil agar memiliki <i>collision</i> dan memiliki <i>physics</i> .
4	26 Oktober 2017	13515035, 13515065, 13515077	Memperbaiki mobil dengan menambahkan rotasi setir dan spion, membuat <i>scene</i> untuk ujian SIM, dan menambahkan menu utama.
5	27 Oktober 2017	13515035, 13515065, 13515077	Menambahkan rotasi kamera menggunakan <i>mouse</i> , menyelesaikan <i>conflict</i> yang terjadi karena <i>merge branch</i> .
6	28 Oktober 2017	13515035, 13515065, 13515077	Modifikasi dan penambahan <i>scene</i> untuk pengujian, penambahan <i>props</i> pada <i>scene</i> , penambahan <i>checkpoint</i> , penambahan logic untuk simulasi pengujian, dan penambahan lampu pada mobil.
7	29 Oktober 2017	13515035, 13515065, 13515077	Modifikasi <i>scene</i> pengujian, merapikan <i>assets</i> , penyatuan menu utama dengan <i>scene</i> lainnya, pembuatan laporan.
8	25 November 2017	13515065, 13515077	Integrasi Arduino dengan Unity

5.2.3 Arduino

No.	Tanggal	Anggota	Kegiatan
1	4 November 2017	13515065, 13515077	Mengetes sensor ultrasonik, <i>limit switch</i> , LCD, dan <i>7-segment</i>
2	9 November 2017	13515065	Merakit <i>7-segment</i> , motor.
3	10 November 2017	13515065	Menyolder kabel motor yang lepas
4	11 November 2017	13515065, 13515077	Menyatukan sistem, menguji sistem, menyatukan dengan <i>body</i> .
5	12 November 2017	13515035, 13515065, 13515077	Menyolder kabel motor yang lepas, membuat program untuk Arduino, membuat kontroler Python, <i>fritzing</i> .
8	25 November 2017	13515065, 13515077	Integrasi Arduino dengan Unity

Bibliography

- [1] *Android API Guides*. <https://developer.android.com/guide/index.html>. Diakses 28 September 2017. Android Open Source Project.
- [2] *Firebase Guides*. <https://firebase.google.com/docs/guides>. Diakses 28 September 2017. Google Developers.
- [3] *Fritzing*. <http://fritzing.org/learning/>. Diakses 11 November 2017. Friends-of-Fritzing.
- [4] *FusedLocationProviderClient*. <https://developers.google.com/android/reference/com/google/android/gms/location/FusedLocationProviderClient>. Diakses 28 September 2017. Google Developers.
- [5] *Learn Arduino*. <https://www.arduino.cc/en/Guide/HomePage>. Diakses 2 November 2017. Arduino Software.
- [6] Taylor Otwell. *Laravel Documentation*. <https://laravel.com/>. Diakses 28 September 2017.
- [7] *Unity User Manual*. <https://docs.unity3d.com/Manual/index.html>. Diakses 24 Oktober 2017. Unity Technologies.