

**LAPORAN PRAKTIKUM**  
**GRAFIKA DAN KOMPUTASI VISUAL**



**Disusun untuk Memenuhi Tugas Praktikum**  
**pada Mata Kuliah Grafika dan Komputasi Visual**

**DISUSUN OLEH :**

1. Attaf Riski Putra Ramadhan (24060121140103)
2. Labiba Adinda Zahwana (24060121140111)
3. Muhamad Ridwan Ash'shidqi (24060121130075)
4. Safril Isnaini (24060121130057)

**PROGRAM STUDI S1 INFORMATIKA**  
**FAKULTAS SAINS DAN MATEMATIKA**  
**UNIVERSITAS DIPONEGORO**  
**SEMARANG**

**2022**

## **KATA PENGANTAR**

Segala puji dan syukur kami panjatkan kepada Allah Subhanahu Wa Ta'ala yang telah melimpahkan rahmat, hidayah, dan inayah-Nya karena telah memberikan kesempatan dan kelancaran kepada kami dalam penyelesaian tugas pembuatan objek tiga dimensi dan pembuatan laporan praktikum sehingga tugas-tugas tersebut dapat terselesaikan tepat waktu. Penyusunan laporan praktikum ini merupakan bentuk penyelesaian salah satu tugas dalam praktikum mata kuliah Gravika dan Komputasi Visual di Program Studi Informatika Universitas Diponegoro.

Dalam penulisan laporan ini, penyusun telah mendapatkan banyak bantuan dari berbagai pihak sehingga mempermudah dan memperlancar pembuatan laporan. Oleh karena itu, penyusun hendak menyampaikan terima kasih kepada :

1. Asisten praktikum Grafika dan Komputasi Visual
2. Teman-teman sekalian yang mengikuti praktikum GKV
3. Keluarga dari masing-masing penyusun
4. Semua pihak yang telah ikut membantu dalam penyelesaian laporan praktikum yang tidak bisa penyusun sebutkan satu persatu.

Dalam penyusunan laporan praktikum ini penyusun menyadari sepenuhnya bahwa masih terdapat kesalahan dan kekurangan baik di segi teknis maupun materi yang disampaikan. Maka dari itu, kritik dan saran yang membangun dari berbagai pihak kami harapkan demi perbaikan laporan praktikum ini. Akhir kata, penyusun harapkan agar makalah ini dapat memberikan manfaat maupun inspirasi kepada para pembaca.

Semarang, 22 Mei 2022

Tim Penyusun

## DAFTAR ISI

<b>HALAMAN SAMPUL .....</b>	<b>i</b>
<b>KATA PENGANTAR.....</b>	<b>ii</b>
<b>DAFTAR ISI.....</b>	<b>iii</b>
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang... ..	1
1.2 Rumusan Masalah .....	2
1.3 Tujuan Pembuatan .....	2
<b>BAB II DASAR TEORI .....</b>	<b>3</b>
2.1 Grafika 3 Dimensi.....	3
2.2 Pengertian Game .....	3
2.3 Perkembangan Game.....	4
2.4 Karakteristik Game.....	5
2.5 Jenis-jenis gendre Game.....	7
2.6 Tahapan Pembuatan Game .....	8
<b>BAB III PEMBAHASAN .....</b>	<b>10</b>
3.1 Rancangan Game... ..	10
3.1.1 Gendre Game .....	10
3.1.2 Tool.....	10
3.1.3 Game Play.....	10
3.1.4 Grafis.....	11
3.1.5 Suara.....	11
3.1.6 Pembuatan.....	11
3.1.7 Publishing.....	12
3.2 Pengembangan Game Truk.....	12
3.2.1 Mekanik-mekanik dalam pembuatan Game Truk.....	12
3.2.2 Objek-objek pada Game Truk.....	16
3.2.3 User Interface pada Game Truk.....	18
3.3 Pembagian Tugas.....	18
<b>BAB IV PENUTUP .....</b>	<b>19</b>
<b>DAFTAR PUSTAKA .....</b>	<b>20</b>
<b>LAMPIRAN.....</b>	<b>21</b>

# **BAB I**

## **PENDAHULUAN**

### **1. Latar Belakang**

Digital Image Processing, Komputer Grafik, Analisis dan Computer telah dikembangkan dan diaplikasikan dengan mengesankan selama beberapa dekade. Dimana perkembangan aplikasi-aplikasi yang menggunakan disiplin ilmu telah memimpin teknologi di beberapa bidang seperti komunikasi digital dan internet, penyiaran (broadcasting), alat kedokteran, sistem multimedia, biologi, ilmu pengetahuan material, robot dan manufaktur, sistem intelligent sensing, remote sensing, seni grafik dan proses print. Pertumbuhan yang pesat ini direfleksikan dengan diterbitkannya paper di jurnal ilmiah internasional setiap tahunnya dan diluncurkannya buku-buku tentang Pemrosesan Image Digital dan Komputer Grafik.

Zaman dahulu karya grafis Zaman dahulu karya grafis diolah memakai teknologi analog. Dalam teknologi diolah memakai teknologi analog. Dalam teknologi ini gelombang direkam atau ini gelombang direkam atau dipakai dalam wujud aslinya. Sebagai contoh, dalam dunia fotografi konvensional kamera menangkap image yang berasal dari sinar dunia fotografi konvensional kamera menangkap image yang berasal dari sinar yang dipantulkan oleh objek. Sinar pantul ini merupakan yang dipantulkan oleh objek. Sinar pantul ini merupakan gelombang analog dan jejak intensitas cahaya yang tertinggal/terekam pada film juga gelombang analog. Jejak analog ini bisa dibaca, ditransfer ke media khusus peka cahaya sehingga menghasilkan karya grafis. Dewasa ini pengolahan grafis cenderung memakai teknologi digital. Dalam teknologi ini image analog yang ditangkap kamera digital dibagi-bagi menjadi unsur-unsur gambar paling kecil lalu diubah menjadi bit dan disimpan dalam perangkat digital.

Istilah grafis komputer mencakup hampir semua yang ada di komputer yang bukan hanya teks atau suara. Grafik komputer dan gambar yang dihasilkan komputer menyentuh semua aspek. Komputer grafis yang aspek. Komputer grafis yang dibangun dengan baik dapat menyajikan grafik statistik kompleks dalam bentuk yang lebih muda untuk memahami dan menafsirkannya. Berbagai alat dan fasilitas yang tersedia pada komputer grafis memungkinkan

pengguna dapat memvisualisasikan data-data mereka dan juga komputer grafis banyak digunakan dalam digunakan dalam banyak ilmu.

Grafik komputer dua dimensi (2D) adalah pembuatan objek gambar dengan menggunakan dua titik sebagai acuannya, yaitu sumbu x dan y. Karakteristik pada grafik komputer 2D adalah terlihat dari tampilan objeknya yang datar, tidak beruang, berupa struktur garis dan tampilan frame layarnya terbatas. Dalam grafika komputer, objek 2D memiliki keunggulan di antaranya adalah kemudahan dalam membuat gambar dan tidak memerlukan waktu banyak untuk meresource.

Selain itu, objek 2D juga memiliki kelemahan, yaitu gambar terlihat biasa saja, belum mendukung bentuk-bentuk 3D atau fenomena yang bersifat tiga dimensi, seperti pencahayaan, bayangan, pantulan, refraksi, dan sebagainya. Macam-macam objek 2D di antaranya, line, circle, arc, polygon, text, section, rectangle, ellips, donut, star, dan helix.

## **2. Rumusan Masalah**

- a. Bagaimana pembuatan interaksi objek 3D dengan menggunakan OoenGL?
- b. Bagaimana pengembangan yang dilakukan pada game simulasi truk?
- c. Apa saja mekanik-mekanik yang ada pada game simulasi truk?
- d. Apa saja objek – objek yang ada pada game simulasi truk?
- e. Bagaimana user interface pada game simulasi truk?

## **3. Tujuan Pembuatan**

- a. Menjelaskan code dalam pembuatan interaksi objek 3D dengan OpenGL
- b. Menjelaskan pengembangan yang dilakukan pada game simulasi truk
- c. Menjelaskan mekanik pada permainan simulasi truk
- d. Menjelaskan objek-objek yang ada pada permainan simulasi truk
- e. Menjelaskan antarmuka pengguna permainan simulasi truk

## **BAB II**

### **DASAR TEORI**

#### **2.1 Grafika Tiga Dimensi**

Grafik tiga dimensi merupakan teknik penggambaran yang berpatokan pada titik koordinat sumbu x(datar), sumbu y(tegak), dan sumbu z (miring).<sup>1</sup> Dalam sistem 2D hanya digunakan dua koordinat X dan Y tetapi dalam 3D, koordinat Z tambahan ditambahkan.<sup>2</sup> Grafik adalah penyajian informasi dalam bentuk gambar,bukan dalam bentuk teks.<sup>3</sup> Tiga dimensi adalah ilusi yang menyajikan kedalaman atau jarak yang berbeda.<sup>4</sup> Dengan berbagai pengertian di atas dapat disimpulkan bahwa grafik tiga dimensi adalah penyajian informasi dalam bentuk gambar atau sejenis yang menggunakan koordinat X, Y dan Z sehingga tercipta kedalaman atau jarak.

#### **2.2 Pengertian Game**

Pengertian game jika diartikan secara bahasa (berasal dari bahasa Inggris) adalah permainan. Dengan bantuan teknologi, kini game dapat diartikan secara lebih luas lagi. Jadi, secara istilah pengertian game adalah permainan yang diprogram pada suatu perangkat yang dapat dijalankan secara offline maupun online.

Berdasarkan cara bermainnya, game dapat dijalankan pada perangkat yang tidak terhubung ke internet (offline) dan harus terhubung ke internet (online) untuk dapat mengakses suatu server yang terpusat. Oleh karena itu, pengguna tidak dapat bermain game online apabila perangkatnya tidak terhubung ke internet.

Adapun antara game satu dengan lainnya memiliki fungsi logika yang berbeda-beda. Hal ini pula yang membuat setiap game memiliki perintah dan data yang berbeda pula. Pada dasarnya, pengertian game ini bersifat sangat global. Dengan kata lain, game tidak hanya permainan yang dapat dijalankan oleh perangkat saja, melainkan juga dapat dimainkan secara tradisional.

---

<sup>1</sup> Ikhsanudin, "Cara Membuat Grafik 3D di GeoGebra", diakses dari <https://googebra.blogspot.com/2021/09/cara-membuat-grafik-3d-di-geogebra.html>, pada tanggal 22 Mei 2022, pukul 20.11 WIB

<sup>2</sup> Tutorialspoint. *Computer Grapics* (-:Tutorialspoint,2015), hlm. 41.

<sup>3</sup> grafik. 2016. Pada KBBI Daring. Diambil 22 Mei 2022, dari <https://kbbi.kemdikbud.go.id/entri/grafik>

<sup>4</sup> tiga dimensi. 2016. Pada KBBI Daring. Diambil 22 Mei 2022, dari <https://kbbi.kemdikbud.go.id/entri/tiga>dimensi

## 2.3 Perkembangan Game

Saat ini game menjadi salah satu aplikasi yang paling banyak dicari oleh berbagai kalangan terutama anak-anak hingga dewasa. Meski terlihat sekedar permainan, namun sebenarnya di dalamnya ada beragam manfaat. Berikut ini akan dijelaskan tahapan pengembangannya:

### 1. Tahap Riset dan Penyusunan Konsep Dasar

Sebelum memulai pengembangannya, tentu hal pertama yang harus dilakukan adalah melakukan riset untuk mendapatkan konsep paling sesuai. Tahapan satu ini juga bermanfaat agar dapat mengetahui mulai dari peluang, risiko, serta perkiraan dana yang akan dibutuhkan nantinya.

Ada beberapa hal yang perlu dipertimbangkan dalam tahapan ini yakni mulai dari ide dasar, tema, objek, target audience, teknologi, platform, dan batasan-batasan yang telah ditentukan. Perlu diperhatikan, tanpa adanya riset sebelumnya bisa saja akan membuat pengembangan game lebih sulit dari perkiraan.

### 2. Perumusan Gameplay

Fokus utama dalam tahapan ini adalah lebih terfokus pada seluruh karakter yang akan dimasukkan dalam game nantinya. Gameplay sendiri merupakan aturan, pola, atau mekanisme yang akan mengatur bagaimana nantinya game akan beroperasi untuk membuat pengguna mendapatkan pengalaman bermain sesuai ekspektasi.

Penyusunan gameplay akan menentukan bagaimana kesan yang akan didapatkan para pemain saat menggunakan game. Bagian ini juga tidak boleh dipandang sebelah mata karena berkaitan dengan tingkat kepuasan. Apabila terlalu membosankan umumnya user akan melakukan uninstall atau meninggalkan platform tersebut.

### 3. Penyusunan Aset dan Level Design

Ada beberapa hal yang perlu dimasukkan dalam tahap satu ini yakni mulai dari penyusunan berbagai aset misalnya saja tokoh, level, tema, dan sebagainya. Anda juga perlu menentukan tokoh-tokoh serta bentuknya sekaligus. Pastikan untuk memilih dengan desain menarik serta tidak membuat orang cepat bosan.

Saat melakukan proses desain latar tempat, silahkan mengelompokkan berdasarkan level serta tingkat kesulitan yang akan dihadapi para pemain. Tujuan utamanya untuk memberikan kesan menarik pada game, Anda bisa melakukan tahapan pengonsepan secara menyeluruh.

### 4. Test Play (Prototyping)

Setelah menjalankan tahapan di atas, bagian penting lain yang tidak boleh dilewatkan adalah melakukan test play yang bertujuan untuk mengetahui jika adanya kesalahan.

Pemeriksaan ini terbilang cukup penting guna meminimalisir kesalahan setelah platform resmi dirilis.

Pada tes play penting untuk melakukan pengujian secara keseluruhan mulai dari per satu level maupun secara konsep keseluruhan beserta konsepnya. Dengan begitu jika masih terdapat kekurangan maka proses perbaikan bisa dilakukan secara tepat sasaran.

#### 5. Development dan Alpha/ Close Beta Test

Pada tahapan ini seluruh konsep yang disusun sebelumnya mulai dikembangkan secara menyeluruh. Proses paling penting adalah penggabungan seluruh elemen untuk menjadi satu kesatuan yang salig melengkapi. Setel proses ini berlanjut pada tahapan untuk melakukan close beta test.

Close beta test sendiri memiliki fokus utama pada untuk mengetahui apakah seluruh komponen utama dari game telah memberikan pengalaman pengguna layaknya yang diharapkan. Manfaat lainnya yakni untuk mendeteksi jika terjadi sebuah masalah teknis misalnya saja pengoperasiannya sedikit tersendat.

#### 6. Perilisan

Setelah pemeriksaan secara keseluruhan, hal yang perlu dilakukan adalah melakukan perilisan. Meski nantinya telah dirilis, namun umumnya developer akan tetap melakukan pembaharuan untuk memberikan pengalaman bermain pada ara user lebih maksimal misalnya saja dengan menambahkan fitur terkini.

Setelah perilian bukan berarti tahap pengembangan telah berakhir, Anda bisa mengoptimalkan seluruh komponen yang ada di dalamnya. Jika perlu dapat pula menambahkan level dengan tampilan grafis lebih segar. Sebelum itu pastikan untuk selalu memperkirakan pendanaan serta estimasi keuntungan apabila mengizinkannya digunakan oleh publik.

## 2.4 Karakteristik Game

Ada beberapa unsur sistematis umum yang menjadikan sebuah permainan bisa disebut sebagai permainan. Menurut Elias (2012:71), karakteristik umum dari sebuah permainan antara lain:

#### 1. Permainan memiliki aturan

Semua permainan memiliki setidaknya aturan, yaitu suatu ketentuan aksi apa yang dapat dilakukan oleh pemain serta akibatnya, baik secara langsung maupun tidak langsung. Terdapat dua derajat aturan dalam sebuah permainan. Aturan derajat satu



merupakan aturan dasar yang harus dipahami untuk memahami sebuah permainan, seperti gerakan-gerakan pada pion catur. Sementara itu, aturan derajat dua merupakan aturan tambahan yang harus ada, akan tetapi tidak semua pemain harus mengetahuinya. Misalnya, aturan pelanggaran pada permainan bola basket. Aturan terkadang dapat menjadi penghalang bagi pemain baru untuk mulai memainkan suatu game.

## 2. Permainan memiliki suatu standar

Menguasai sebuah aturan yang baru bisa menjadi hal yang sulit. Suatu permainan dapat mengakali hal ini dengan menggunakan standar, yaitu pola yang tidak asing dan dapat diterima oleh para pemain. Misalnya, penggunaan dadu pada permainan papan untuk menggerakkan karakter, atau tombol panah untuk menggerakkan karakter pada sebuah video game. Game yang berada dalam satu genre biasanya memiliki standar yang sama.

## 3. Permainan memberikan hasil

Kebanyakan permainan memiliki hasil yang jelas antara menang dan kalah. Jika permainan memiliki karakteristik tersebut, maka permainan disebut ortogonal. Terdapat juga permainan yang tidak memberikan hasil yang jelas antara menang dan kalah, disebut permainan tidak ortogonal. Hasil dari permainan ini ditentukan sendiri oleh pemain, baik secara individu maupun bersama. Misalnya, skor tertinggi atau waktu tercepat.

## 4. Permainan menghasilkan kondisi akhir

Baik permainan ortogonal maupun tidak ortogonal memiliki kondisi akhir yang akan didapat oleh pemain. Pada permainan ortogonal, kondisi akhirnya adalah jelas keadaan menang, kalah, atau seri. Bagi permainan tidak ortogonal, kondisi akhir yang didapat oleh pemain dapat berupa jumlah skor.

## 5. Permainan menyediakan ketidaksimetrisan

Hampir semua permainan memiliki keadaan awal yang berbeda bagi para pemainnya. Ketidaksimetrisan itu bisa datang dari aturan permainan, lingkungan, atau pemain itu sendiri. Ketidaksimetrisan ini dihadirkan agar pemain mendapat sebuah tantangan dan permainan memiliki nilai perulangan yang tinggi.

## 6. Permainan memberikan umpan balik sensori

Bermain bukan hanya pertemuan antara otak manusia dan algoritma dari aturan permainan yang keduanya abstrak. Suatu permainan memberikan informasi audio, visual, atau taktil kepada pemainnya. Pemain juga memberikan masukan kepada sebuah permainan melalui medium-medium tersebut.

## 2.5 Jenis-Jenis Genre Game

Hingga saat ini genre-genre game terus bermunculan, ini menandakan bahwa game tetap tumbuh hingga saat ini. Perkembangan ini didasarkan permintaan pasar dan juga kreatifitas dari para pengembang game yang mencampurkan berbagai jenis game sehingga tercipta genre baru dan tidak terduga sebelumnya. Berikut ini adalah beberapa genre yang paling banyak disukai saat ini.<sup>5</sup>

### 1. Sandbox

Sandbox adalah genre game yang berhubungan dengan pilihan dari pemain, lingkungan yang terbuka, dan alur game yang tidak linear. Di genre ini dibanding para pemain terlalu mengikut narasi dari game, pemain dibebaskan untuk melakukan berbagai tugas sesuai keinginan pemain dengan berbagai cara yang disukai pemain juga.

### 2. Real-Time Strategy (RTS)

RTS adalah cabang atau pecahan dari game bergenre strategi yang digemari oleh banyak pecinta game. Biasanya ciri dari genre ini adalah dapat mengendalikan banyak unit dengan *dragging* atau *clicking*.

### 3. Shooter (FPS dan TPS)

Perbedaan FPS dan TPS adalah perspektif. FPS merupakan perspektif penglihatan manusia sedangkan TPS adalah penglihatan dibelakang atas karakter. Shooter memiliki inti alur permainan adalah menembaki musuh atau target sesuai misi.

---

<sup>5</sup> Dwight Pavlovic, "Video Game Genres: Everything You Need to Know", diakses dari <https://www.hp.com/us-en/shop/tech-takes/video-game-genres>, pada tanggal 22 Mei 2022, pukul 20.54 WIB

#### 4. Multiplayer Online Battle Arena (MOBA)

MOBA merupakan game strategi mirip dengan RTS namun MOBA hanya mengendalikan satu karakter berbeda dengan RTS yang mengendalikan banyak objek. Biasanya genre MOBA memiliki perspektif *top-down*.

#### 5. Role-Playing Games (RPG)

RPG memiliki ciri membuat atau mengambil alih seorang karakter dan seorang pemain bisa menaikan level melewati poin pengalaman (XP). RPG memiliki banyak sub genre.

#### 6. Simulasi dan Olahraga

Genre ini memiliki ciri alur game seperti di dunia asli sehingga apabila disederhakan genre ini adalah membuat game komputer berdasarkan permainan olahraga di dunia nyata.

#### 7. Arkade

Arkade biasanya membutuhkan reflek pemain yang bagus dan sedikit menggunakan kemampuan strategi atau pemikiran kompleks dari pemain. Genre ini juga yang diambil di tugas besar praktikum GKV.

### 2.6 Tahapan Pembuatan Game

Berikut ini adalah tahapan-tahapan dalam pembuatan sebuah game :

##### 1. Genre Game

Pertama kali adalah memikirkan genre game yang akan dibuat. Pemilihan jenis game yang sesuai dengan kemampuan yang dimiliki, ketertarikan, , tingkat kemudahan dan cepat dalam pembuatannya.

##### 2. Tool

Menentukan tool yang akan digunakan. Ini bagian yang paling penting, menggunakan apa kita akan membuat game tersebut. Misalkan menggunakan software tertentu untuk mempermudah proses pembuatan game yang dikehendaki.

### 3. Gameplay

Gameplay adalah sistem alur game tersebut, mulai dari menu, area permainan, game over, storyline, mission success, mission failed, cara bermain dan sistem lain yang harus ditentukan. Buatlah gameplay yang mudah dipahami maupun dimainkan para pengguna.

### 4. Grafis

Penjelasan tentang grafis yang akan digunakan. Jenis grafis dibagi menjadi tiga, yaitu jenis grafis kartun, semi realis dan realis. Pilihlah jenis grafis yang sesuai dengan kebutuhan dari jenis game yang akan dibuat.

### 5. Suara (Audio)

Interface atau tampilan akan tampak membosankan tanpa adanya suara.

Pemilihan karakter suara harus sesuai dengan genre dan juga bagian-bagian dari game tersebut.

### 6. Pembuatan

Persiapan yang matang memudahkan pembuat game untuk segera membuat game yang sesuai dengan waktu yang sudah direncanakan sebelumnya.

### 7. Publishing

Ketika segala proses pembuatan game telah selesai, publish game menjadi setup jika game harus diinstal terlebih dahulu sebelum dimainkan.

## **BAB III**

### **PEMBAHASAN**

#### **3.1 Rancangan Game**

Dalam pembuatan game yang telah kelompok kami lakukan ada beberapa tahapan seperti pemilihan genre game, untuk alat yang digunakan karena sudah ditentukan sehingga lebih memudahkan untuk pengerjaan dan beberapa perancangan lainnya yang perlu diperhatikan.<sup>6</sup>

##### **3.1.1 Genre Game**

Genre adalah pengkategorian permainan yang memiliki karakteristik serupa. Sebagai perancang game, penentuan genre akan mempengaruhi target pengguna. Di google play terdapat genre yang diantaranya adalah Arkade, Balapan, Kartu, Kasino, Kata, Laga, Musik, dan lain lain.

Melalui game yang telah kami buat, memiliki genre bertipe arkade. Permainan dengan genre arkade biasanya adalah permainan yang membutuhkan refleks pemain, dan biasanya hanya menampilkan sedikit teka-teki atau pemikiran yang kompleks atau keahlian strategi.

##### **3.1.2 Tool**

Peralatan utama yang digunakan dalam pembuatan game ini adalah modul library grafik OpenGL. Untuk menggunakan library ini secara praktis, pengembangan permainan menggunakan modul GLUT (Graphical Library Utility Tool). Pengembangan dari permainan ini menggunakan bahasa pemrograman C++. Alasan dari penggunaan C++ adalah untuk mendapatkan performa yang tinggi. Sementara itu, aplikasi IDE yang digunakan untuk menulis kode sumber dari permainan ini adalah Dev C++.

##### **3.1.3 Gameplay**

Pada game yang kami kerjakan mempunyai gameplay layaknya game arcade endless runner yaitu mengutamakan refleks yang baik dari pemain dan mencari skor sebanyak-banyaknya.

---

<sup>6</sup> Helva Silvianita, 2019. "Pengertian Game Beserta Sejarah, Manfaat, serta Jenis-Jenis Game Lengkap!", [www.nesabamedia.com](http://www.nesabamedia.com). Diakses pada 23 Mei 2022.  
<https://www.nesabamedia.com/pengertian-game/>

Pada awal permainan pemain akan memiliki nyawa sebanyak lima dan skor masih nol. Nyawa pemain akan berkurang apabila pemain menabrakkan truk ke mobil yang lewat dan skor akan bertambah seiring lamanya waktu hidup dari truk pemain. Permainan berakhir apabila nyawa dari truk hanya nol dan terdapat menu main lagi dengan menekan tombol spasi.

#### **3.1.4 Grafis**

Dikarenakan penggunaan library GLUT untuk mengembangkan game ini, maka diputuskan bahwa direksi artistik yang digunakan adalah simpel. Namun kompleks.

Maksudnya adalah penggunaan berbagai primitif-primitif 3D simpel yang diimplementasikan pada library GLUT untuk membangun kota yang terdapat objek-objek menarik.

Alasan digunakannya direksi artistik seperti ini antara lain:

- Keterbatasan waktu

Membuat objek dengan sangat detail memerlukan waktu yang sangat lama.

- Performance

Objek yang sangat detail perlu banyak vertex atau sudut yang ditambah dengan object count yang banyak pada kota akan menurunkan performa game.

- Kecocokan

Objek dengan detail terlalu tinggi akan tampak out of place dengan cara render GLUT yang terbilang simpel. Memang pengimplementasian shader dan bayangan bisa saja dilakukan namun selain akan merumitkan pengembangan game (yang telah sangat terbatas waktu) nantinya juga mengurangi performa game.

#### **3.1.5 Suara**

Dalam pengembangan game ini tidak ditambahkan suara.

#### **3.1.6 Pembuatan**

Pembuatan game ini dilakukan secara daring. Masing-masing anggota kelompok melakukan kontribusi untuk masing-masing komponen game. Source code dari game ini kemudian diunggah ke repository Github. Dalam melaksanakan pengembangan game, dilakukan beberapa kali

pertemuan untuk membahas komponen-komponen dalam game. Pengembangan game dilakukan selama 5 minggu dengan alokasi waktu individu 5 jam per minggu.

### **3.1.7 Publishing**

Setelah tahap pembuatan kelompok kami memutuskan untuk mempublish hasil karya kami menjadi proyek *open source* di github. Hal itu memungkinkan para pemain dapat mengunduh game kami dengan gratis sekaligus mendapatkan *source code* dari game kami. Untuk memenuhi tugas praktikum juga kami mengupload hasil review proyek oleh kelompok kami di Youtube.

## **3.2 Pengembangan Game Truk**

Arahan dalam pengembangan game truk adalah pengembangan yang dilakukan secara bersama-sama menggunakan github agar dapat berkolaborasi kode bersama-sama sesuai dengan bagian-bagiannya masing-masing. Setiap anggota kelompok saling bantu membantu untuk pengembangan game truk ini agar tercipta kolaborasi yang baik dan harmonis. Game truk ini dikonsep oleh kelompok kami agar tetap ringan dengan menggunakan objek yang tidak banyak namun tetap ramai, caranya dengan menggunakan proses perulangan yaitu ketika objek yang sudah tidak terlihat lagi maka tidak akan dihancurkan namun tetap akan digunakan lagi dengan cara proses translasi ke arah depan lagi. Serta berbagai macam objek yang dibentuk baik menggunakan objek bawaan library GLUT seperti `glSolidDodecahedron`, `glSolidCube`, dan lain-lain serta pembuatan objek menggunakan objek-objek dasar seperti kotak, titik, dan lain-lain.

### **3.2.1 Mekanik Dalam Game Truk**

Terdapat beberapa mekanik game yang kami buat sejauh ini, beberapa diantaranya adalah:

#### **a. Movement Truk**

Untuk mensimulasikan pengendaraan truk, diperlukan pemahaman bagaimana suatu truk itu berjalan. Sebuah truk berjalan maju, dan bila kita menekan `a` maka akan ke kiri dan jika kita menekan `d` akan ke kanan.

Implementasi dari movement truk dalam game ini dilakukan dengan cara menggerakkan truk dengan memindah koordinatnya. Diputuskan bahwa dalam game ini pergerakan truk tidak akan ada naik dan turun, sehingga tidak akan ada perubahan koordinat `y`.

Untuk menggerakkan truk diubah koordinat x dan z nya dengan cara menambahkan atau mengurangi dengan kecepatan. Digunakan modifier yang akan dikalikan dengan kecepatan truk dalam mengubah koordinat x dan z masing-masing. Misalnya bila truk memiliki perputaran 0 derajat, maka koordinat z akan memiliki modifier 1 dan koordinat x akan memiliki modifier

0. Bila perputaran 90 derajat maka koordinat z akan memiliki modifier 0 sedangkan koordinat x akan memiliki modifier 1. Lalu bila perputaran 45 derajat maka kedua koordinat x dan z akan memiliki modifier 0.5. Modifier tersebut akan dihitung dengan melibatkan rotasi truk. Hal ini akan memastikan bahwa truk akan bergerak maju atau mundur sesuai rotasinya. Berikut merupakan penghitungan modifier yang digunakan:

$$\text{ModX} = (90.0 - \text{delta}) / -90;$$

$$\text{ModZ} = -\sin(\text{putaran} * M\_PI / 180) * (1 - \text{abs}((90.0 - \text{delta}) / -90))$$

Dimana :

delta = jarak antara rotasi truk saat ini dan rotasi truk saat berbalik badan atau 180 derajat. Maka dari itu  $\text{delta} = \text{abs}(180 - \text{putaran})$

Selanjutnya tentang pergerakan atau setir truk. Dikarenakan pergerakan truk telah melibatkan rotasi truk saat ini, maka perputaran truk dapat dihitung dengan mudah, yaitu menambah atau mengurangi rotasi truk sesuai dengan arah setir (kanan atau kiri).

#### a. Movement Camera

Kamera pada game ini diposisikan tepat dibelakang truk. Dalam mengikuti pergerakan truk kamera akan ikut bergerak sesuai dengan rotasi dan posisi truk saat ini. Maka dari itu, untuk menghitung posisi dari kamera melibatkan koordinat x dan z truk serta rotasi truk saat ini. Posisi kamera didapatkan dengan menambahkan posisi truk saat ini dengan konstan jarak kamera dengan truk yang dikalikan dengan modifier posisi kamera sesuai rotasi. Berikut merupakan persamaan yang digunakan untuk menentukan posisi kamera:

$$\text{CameraX} = \text{TrukX} + \text{jarak} * \text{ModX} \quad \text{CameraZ} = \text{TrukZ} + \text{jarak} * \text{ModZ} \quad \text{CameraY} = \text{TrukY} + \text{jarak}$$

$$\text{ModX} = \sin(i * M\_PI / 180) \quad \text{ModZ} = \cos(i * M\_PI / 180)$$

Dimana :

jarak = konstan jarak antara kamera dengan truk



i = rotasi truk

Dalam memindahkan posisi kamera, tidak dilakukan secara langsung, namun diberikan sedikit *lag* pada input rotasi truk untuk fungsi gerakan kamera. Ini dimaksudkan agar kamera tidak langsung berada di posisinya secara instan namun melakukan putaran secara smooth. Hal ini juga memungkinkan player untuk melihat sisi kanan, kiri, serta depan truk.

#### b. Collision (Tabrakan)

Mekanik collision pada game ini sangatlah sederhana. Pertama-tama pada bagian depan dan bagian belakang truk akan dipasang sebuah box. Box ini tidak terlihat dan berfungsi sebagai collision areanya. Tidak hanya pada truk saja, setiap objek yang memungkinkan untuk ditabrak, misalnya mobil juga memiliki collision area.

Jika koordinat collision area truk memasuki atau berdempetan dengan koordinat collision area pada objek, maka game akan mendeteksi kejadian tersebut sebagai suatu tabrakan. Jika terjadi tabrakan pada depan truk, maka truk tidak bisa bergerak maju. Begitu sebaliknya.



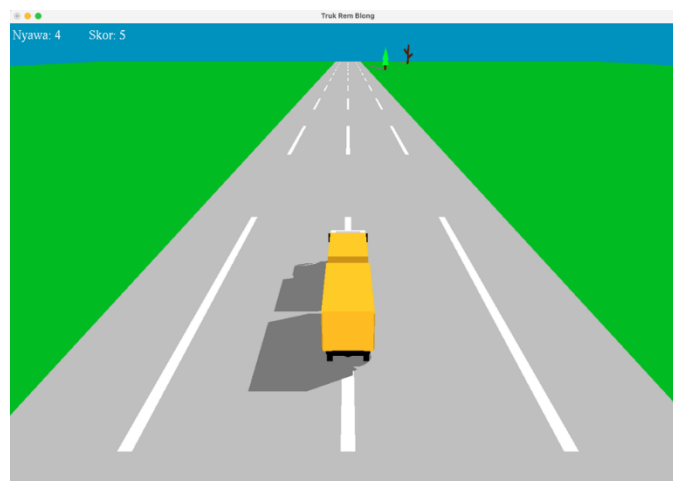
Collision box pada objek truk

#### c. Sistem Score dan Random Coin

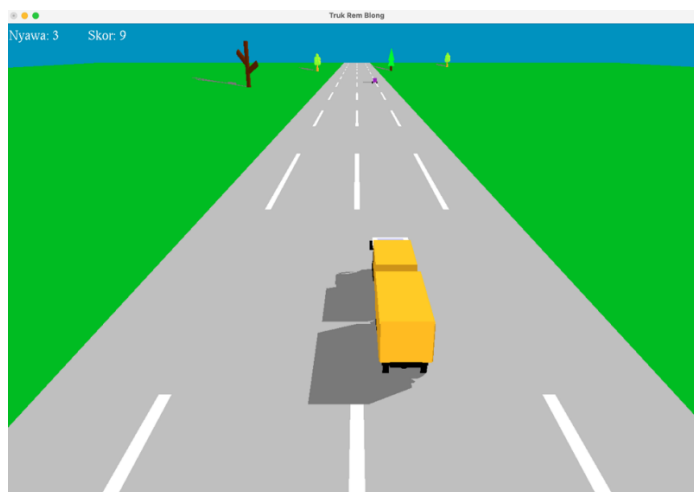
Pada awal permainan truk ini akan diberikan nyawa 5, skor akan bertambah seiring dengan bertambahnya waktu di game. Jika truk menabrak mobil maka nyawa akan berkurang.

Karena game ini adalah game arkade, maka skor hanya bisa bertambah atau tetap. Tidak ada suatu event/kejadian yang dapat membuat skor berkurang. Selain itu tidak ada perubahan kesulitan/level serta batas waktu pada game ini.

Sistem game ini dengan bertambahnya waktu, semakin lama bertahan semakin banyak skor yang didapat. Namun saat truk mengenai mobil, nyawa akan berkurang. Seperti yang terlihat pada gambar dibawah ini.



(I)



(II)

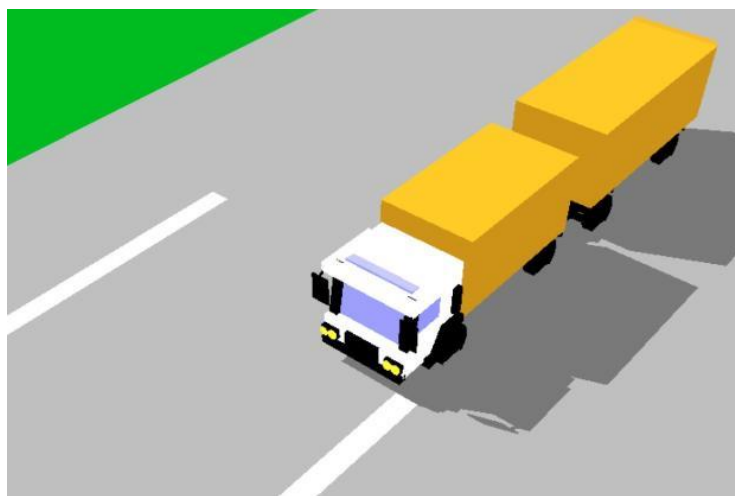
### 3.2.2 Objek Pada Game Truk

objek-objek yang dibuat untuk memberikan suasana yang hidup dan ramai yang ada pada game yang kami buat tersebut adalah :

#### A. Truk

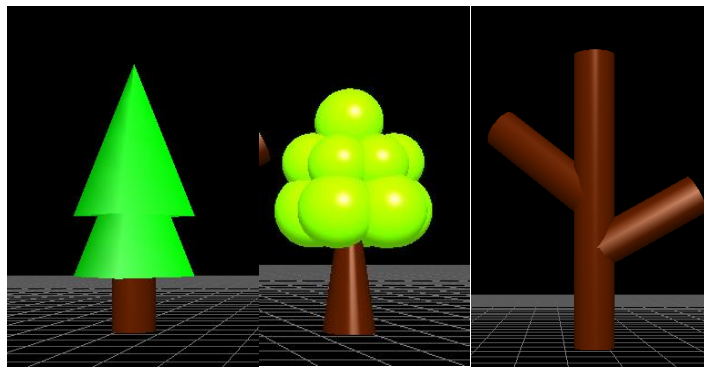
Objek Truk dibuat dengan menggunakan beberapa primitif seperti `GlutSolidCube`, `gluDisk`, `gluSphere`, dan `gluCylinder`. Pembuatan Truk dimulai dengan membuat Prosedur chassis, Wheel, cabin dan lain sebagainya. Di dalam Prosedur chassis terdapat fungsi yang nantinya akan digunakan untuk membuat roda pada bagian depan dan belakang truk. Dasar Pembuatan Chassis Menggunakan Primitif `GluCylinder`.

Pada prosedur Chassis juga terdapat panggilan untuk Prosedur `Wheelcover`, gardan dan sumbu roda dan powertrain. Prosedur `WheelCover` digunakan fungsi `glutSolidCube` untuk membuat roda sedemikian rupa. prosedur gardan dan sumbu roda depan belakang truk digunakan fungsi `gluCylinder` dan yang terakhir prosedur powertrain digunakan fungsi `gluSolidCube`. Selanjutnya terdapat bagian body yang terdiri dari base, half cabin, roof, pillar kiri&kanan, dan lain sebagainya yang memanggil fungsi-fungsi seperti `glutSolidCube`, `gluCylinder`, `gluDisk`. dan yang terakhir terdapat bagian chassis trailer yang terdiri dari gandengan kanan&kiri, frame utama, wheel, `wheelCover`, kontainer, dan lain sebagainya.



## B. Batang pohon

Pada Batang Pohon terdapat 3 Objek Pohon dibuat dengan menggabungkan beberapa fungsi primitif utama, yaitu `gluCylinder`, `glutSolidCone`, `glutSolidSphere`. pada pohon 1 menggunakan `gluCylinder` untuk membuat batang pohon, dan menggunakan fungsi `glutSolidCone` untuk membuat daun pohon. Pada pohon 2 menggunakan fungsi `glutSolidCone` untuk membuat batang pohon dan menggunakan fungsi `glutSolidSphere` untuk membuat daun pohon. Pada pohon 3 menggunakan fungsi `gluCylinder` untuk membuat batang pohon



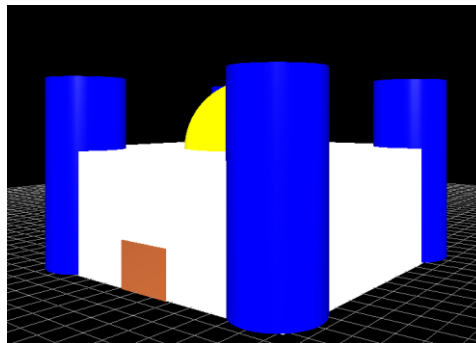
Pohon 1

pohon 2

pohon 3

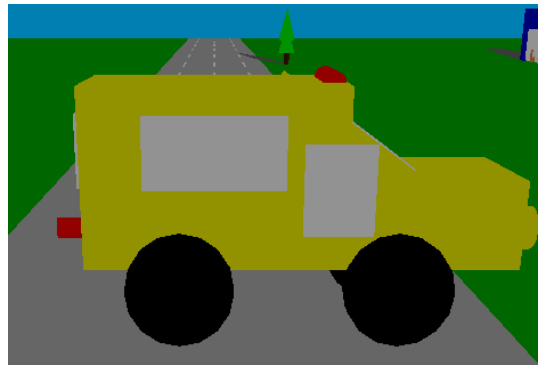
## C. Masjid

Objek masjid dibuat dengan menggabungkan beberapa komponen primitif utama, yaitu `glutSolidCube`, `glutSolidSphere`, `gluCylinder`, dan `GL_POLYGON`. Primitif `glutSolidCube` digunakan untuk membuat badan masjid, `glutSolidSphere` digunakan untuk membuat kubah, `gluCylinder` digunakan untuk membuat 4 meranapada masjid, dan `GL_POLYGON` digunakan untuk membuat sebuah pintu pada masjid



#### D. Mobil

Objek mobil dibuat dari 2 fungsi yaitu cylinder dan blok. Objek mobil pada game ini digunakan sebagai halangan pada truk



#### 3.2.3 User Interface Pada Game Truk

Perangkat utama yang digunakan untuk kontrol dari game ini adalah keyboard. Sebagaimana yang sudah dijelaskan di bagian mekanik game, kontrol dalam game ini dikendalikan oleh tombol panah, huruf ‘A’, dan huruf ‘D’ pada keyboard. Tombol panah kanan dan kiri berturut-turut digunakan untuk mengarahkan pandangan kamera ke arah kanan dan kiri. Sementara itu, tombol ‘A’ dan ‘D’ digunakan untuk mengarahkan truk ke arah kiri dan kanan secara berturut-turut.

Pada jendela program, terdapat tampilan jalan, truk, serta objek-objek lainnya yang menjadi inti dari permainan ini. Di bagian atas jendela, terdapat teks “Skor” dan “Nyawa”. Teks Skor ini akan menunjukkan berapa skor yang telah didapat oleh pemain. Skor pemain akan bertambah setelah truk yang dijalankan oleh pemain berhasil menghindari mobil yang berjalan pada permainan. Kemudian, “Nyawa” akan menunjukkan berapa nyawa yang masih dimiliki oleh pemain. Pada awalnya, diberikan nyawa sejumlah 5. Nyawa pemain akan berkurang jika pemain menabrak objek lain di jalan seperti mobil atau pohon. Apabila nyawa pemain sudah mencapai 0, layar akan mengeluarkan tulisan berbunyi “Game Over, Pencet ‘spasi’ untuk mengulang”. Jika pemain menekan tombol spasi, nyawa pemain akan kembali menjadi 5 dan skor akan kembali menjadi 0. Artinya, pemain telah mengulangi permainan.

#### 3.3 Pembagian Tugas

Pembagian tugas disertakan pada bagian lampiran.

## **BAB IV**

### **PENUTUP**

Game truk ini adalah game arkade berjenis *endless runner* dengan karakter utama adalah sebuah truk dengan rem blong yang berusaha tidak menabrak mobil-mobil yang truk itu lewati. Game ini mengutamakan reflek pemain yang bagus dan tidak perlu berpikir rumit untuk memainkannya. Pada awal permainan pemain mempunyai nyawa sebanyak lima yang akan berkurang apabila pemain menabrak mobil-mobil di depannya. Permainan akan berakhir apabila nyawa dari truk berjumlah nol karena sudah menabrak mobil.

Pengembangan game truk ini dilakukan secara bersama-sama menggunakan bantuan github untuk berkolaborasi secara daring. Game ini memiliki arahan agar dibuat tetap ringan walaupun mempunyai banyak objek di dalamnya secara bersamaan dan menerapkan berbagai macam proses translasi, rotasi, dll setiap kali iterasinya. Dengan bantuan library GLUT, proyek ini menjadi lebih mudah di kerjakan karena kelompok kami menjadi tidak perlu membuat objek-objek secara manual dari awal karena beberapa objek sudah disediakan oleh library GLUT. Namun demikian, GLUT bukanlah sebuah *game engine* seperti cocos2dx ataupun Godot sehingga beberapa fitur penting fisika seperti collision harus dibuat manual pada GLUT dengan memanfaatkan nilai posisi dari masing-masing objek yang ingin diawasi.

Pembagian tugas dilakukan dengan baik dengan membagi-bagi komponen dalam game sehingga masing-masing anggota kelompok memiliki porsi pekerjaannya masing-masing. Dengan bantuan github, masing-masing file yang terpisah tersebut digabung bersama sehingga menjadi satu-kesatuan proyek yang dapat dikatakan sebagai sebuah game arkade truk endless runner.

## DAFTAR PUSTAKA

- Adiani, Fani. 2014. "Laporan Praktikum Komputer Grafik Menggunakan openGL".  
[www.scribd.com](http://www.scribd.com) Diakses pada 23 Mei 2022  
<https://www.scribd.com/doc/229129786/Laporan-Praktikum-Komputer-Grafik-menggunakan-openGL>,
- Elias, George Skaff, dkk. 2012. *Characteristics of Games*. Massachusetts: The MIT Press.
- Googebra.com (2022, 3 September). "Cara Membuat Grafik 3D di Geogebra". Diakses pada 22 Mei 2022, dari <https://googebra.blogspot.com/2021/09/cara-membuat-grafik-3d-di-geogebra.html>
- Grafik. 2016. Pada KBBI Daring. Diambil 22 Mei 2022, dari <https://kbbi.kemdikbud.go.id/entri/grafik>
- Helva Silvianita, 2019. "Pengertian Game Beserta Sejarah, Manfaat, serta Jenis-Jenis Game Lengkap!", [www.nesabamedia.com](http://www.nesabamedia.com). Diakses pada 23 Mei 2022.  
<https://www.nesabamedia.com/pengertian-game/>
- Tiga Dimensi. 2016. Pada KBBI Daring. Diambil 22 Mei 2022, dari [https://kbbi.kemdikbud.go.id/entri/tiga\\_dimensi](https://kbbi.kemdikbud.go.id/entri/tiga_dimensi)
- Tutorialspoint. (2015). *Computer Grapics*. -: Tutorialspoint.

## LAMPIRAN

### Lampiran 1 : Pembagian Tugas

No	Nama	NIM	Alokasi Waktu	Uraian Tugas
1	Attaf Riski Putra Ramadhan	24060121120005	5 Jam per Minggu	<ol style="list-style-type: none"> <li>1. Objek mobil</li> <li>2. Mekanisme bayangan</li> <li>3. String Nyawa dan skor</li> <li>4. Collision</li> <li>5. Membuat Laprak sesuai bagian</li> <li>6. Koordinator Kolaborasi github</li> <li>7. Bug Hunter</li> </ol>
2	Labiba Adinda Zahwana	24060121140111	5 Jam per Minggu	<ol style="list-style-type: none"> <li>1. Jalan</li> <li>2. Garis Jalan</li> </ol>
3	Muhamad ridwan ash'shidqi	24060121130075	5 jam per Minggu	<ol style="list-style-type: none"> <li>1. Pohon</li> <li>2. Masjid</li> <li>3. Ide Game Truk</li> </ol>
4	Safril Isnaini	24060121130057	5 jam per minggu	<ol style="list-style-type: none"> <li>1. Model dan warna truk</li> <li>2. Tekstur pada truk</li> <li>3. Membuat Laprak sesuai bagian</li> <li>4. Integrasi truk dengan seluruh bagian kode sumber</li> </ol>



## Lampiran 2 : Pembagian Pengerjaan Laporan

1. Attaf
2. dinda
3. shidki
4. safril

1. kata pengantar (1)
2. pendahuluan
  - a. latar belakang(2)
  - b. rumusan masalah(3)
  - c. tujuan(4)
3. dasar teori
  - a. grafika 3 dimensi(1)
  - b. pengertian game(2)
  - c. perkembangan game(3)
  - d. karakteristik game(4)
  - e. jenis-jenis genre game(1)
  - f. tahapan pembuatan game(2)
4. pembahasan
  - a. rancangan game ( 1 )
    - i. genre game(3)
    - ii. tool(4)
    - iii. gameplay(1)
    - iv. grafis(2)
    - v. suara(3)
    - vi. pembuatan(4)
    - vii. publishing(1)
  - b. pengembangan game truk(1)
    - i. mekanik-mekanik dalam game truk(2)
    - ii. objek-objek pada game truk(3)
    - iii. user interface pada game truk(4)
  - c. pembagian tugas (1,2,3,4)
5. penutup (1)
6. daftar pustaka (1,2,3,4)
7. lampiran (1)

### Lampiran 3 : Bagian yang dikerjakan Attaf Riski Putra Ramadhan (2406012112005)

```
//
// mobil.h
// TugBes
//
// Created by Attaf Riski Putra Ramadhan on 15/04/22.
//

#ifndef MOBIL_H
#define MOBIL_H
class Halangan {
public:
    float posisiZMobil;
    float posisiXMobil;
    void cylinder(float alas, float atas, float tinggi);
    void kerucut(float ngisor, float nduwur, float dowo);
    void blok(float tebal, int ratiol, int ratiop);
    void Mobil(float r, float g, float b);
    void Cone();
    void sisiCone();
};

#endif /* mobil_h */

#include <math.h>
#include "mobil.h"
#ifdef __APPLE__ // untuk program yang dijalankan di MacOS
#include <GLUT/glut.h>
#include <OpenGL/gl.h>
#else // untuk program yang dijalankan di Windows
#include <GL/glut.h>
#endif

void Halangan::Cone()
{
    // sisi satu
    sisiCone();

    // sisi dua
    glPushMatrix();
    glRotatef(-90, 0.0, 1.0, 0.0);
    glTranslatef(-2.5, -0.1, -1.);
    sisiCone();
    glPopMatrix();

    // sisi tiga
    glPushMatrix();
    glRotatef(90, 0.0, 1.0, 0.0);
    glTranslatef(-0.75, -0.1, 2.);
}
```

```

// pergerakan truk
// gambar bayangan
glPushMatrix();
glTranslatef(0, 0, 40);
glRotatef(-180, 0.0, 1.0, 0.0);
glScalef(1.5, 1.5, 1.5);
glTranslatef(perubahanXTruk, 0, 0);
glEnable(GL_LIGHTING);
glColor3f(0.0,0.0,0.8);
trukGandeng(_textureId);
glPopMatrix();
//sekarang gambar bayangan yang muncul
glPushMatrix();
glShadowProjection(1,e,n);
glTranslatef(0, 0, 40);
glRotatef(-180, 0.0, 1.0, 0.0);
glScalef(1.5, 1.5, 1.5);
glTranslatef(perubahanXTruk, 0, 0);
glDisable(GL_LIGHTING);
glColor3f(0.4,0.4,0.4);
truktanpawarna();
glPopMatrix();

void RenderScore() {
    // nyawa
    char s[100];
    glPushMatrix();
    sprintf(s,"Nyawa: %d", nyawa);
    glColor3f(1,1,1);
    renderBitmapString(5, 30, 0,s); //y axis inverted
    glPopMatrix();

    // skor
    char s2[100];
    glPushMatrix();
    sprintf(s2,"Skor: %d", skor);
    glColor3f(1,1,1);
    renderBitmapString(150, 30, 0,s2); //y axis inverted
    glPopMatrix();

    //gameover
    if(gameover){
        char s3[100];
        glPushMatrix();
        sprintf(s3,"Game Over");
        glColor3f(1,0,0);

```

---

```

if(deltaXTruk == -trukSpeed) // enabler ke kanan
{
    if(posisiXTrukKekanan >= batasTrukKanan) // mentok jalan kanan kanan
    {
        //collision nyawa
        posisiXTrukUntukCollision += 2;

        // collision jalan
        posisiXTrukKekanan += deltaXTruk;
        posisiXTrukKekiri += deltaXTruk; // agar bisa bergerak ke kiri walau
        trukKekananKekiri(posisiXTrukKekanan);
    }
}
else if (deltaXTruk == trukSpeed) // enabler ke kiri
{
    if(posisiXTrukKekiri <= batasTrukKiri) // mentok jalan kanan kiri
    {
        //collision nyawa
        posisiXTrukUntukCollision -= 2;

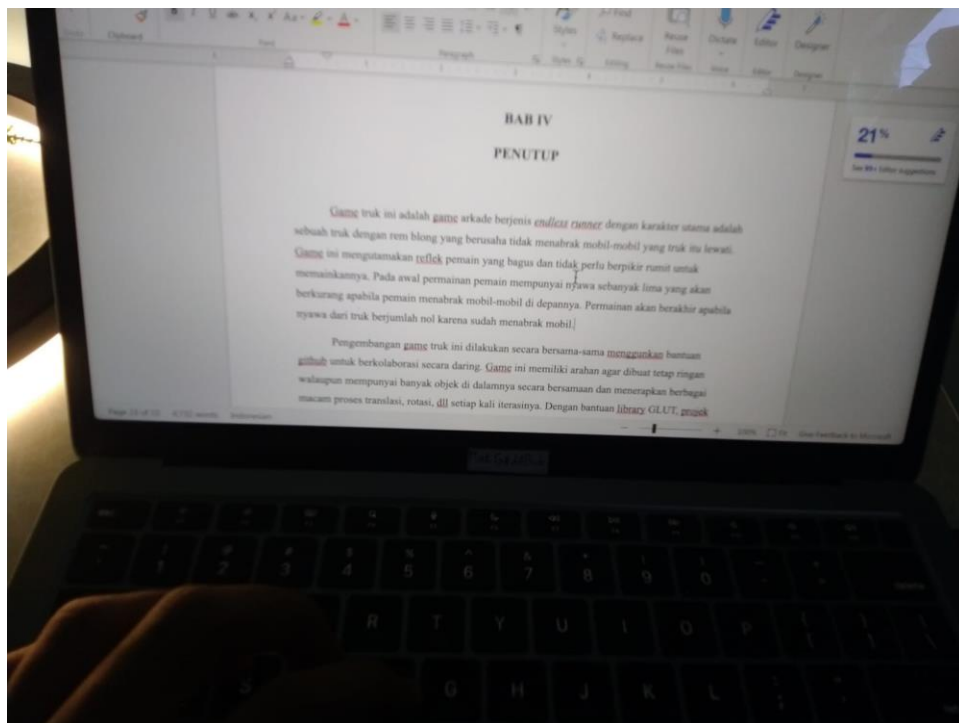
        // collision jalan
        posisiXTrukKekanan += deltaXTruk ; // agar bisa bergerak ke kanan wal
        posisiXTrukKekiri += deltaXTruk;
    }
}

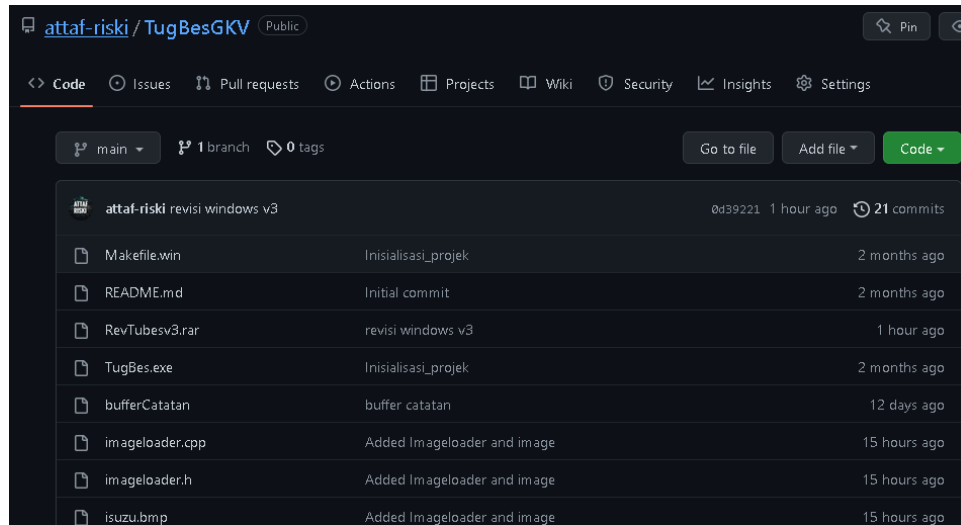
// collision antara truk dan mobil
// penjelasan
// lebar truk 5
// panjang truk perkiraan 20
// ketika mobil berada dalam area truk maka nyawa berkurang

if ( (halangan1.posisiZMobil >= 10 && halangan1.posisiZMobil <= 70)
{
    // biar nyawa gak terus-terusan habis
    lagiketabrak = 0;
    if(!gameover){
        nyawa--;
    }
}
// agar tidak terus-terusan nyawa habis
if(halangan1.posisiZMobil > 80)
{
    lagiketabrak = 1;
}

// pergerakan truk kanan kiri

```





### Lampiran 3 : Bagian yang dikerjakan Safril Isnaini (24060121130057)

```
// Nama File : truk.cpp
// Deskripsi : modul truk
// Pembuat : Safril Isnaini
// Tanggal : Jumat, 15 April 2022
// Lab : GKV B2

#include <math.h>

#ifdef __APPLE__ // untuk program yang dijalankan di MacOS
#include <GLUT/glut.h>
#include <OpenGL/gl.h>
#else // untuk program yang dijalankan di Windows
#include <GL/glut.h>
#endif

void pembuatSegiEmpat(
    float x1, float y1, float z1,
    float x2, float y2, float z2,
    float x3, float y3, float z3,
    float x4, float y4, float z4 )
{
    glBegin(GL_QUADS);
    glVertex3f(x1, y1, z1);
    glVertex3f(x2, y2, z2);
    glVertex3f(x3, y3, z3);
    glVertex3f(x4, y4, z4);
    glEnd();
}

void trukGandeng(GLuint _textureId)
{
    GLUQuadricObj * quadr;
    quadr = gluNewQuadric();

    //=====CHASSIS=====
    //=====CHASSIS=====

```

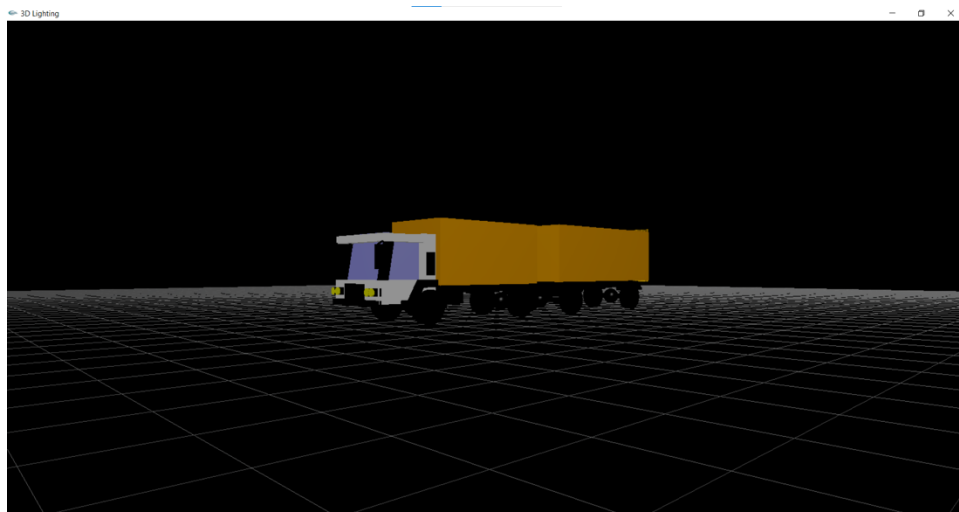
```

glPushMatrix();
//frame utama
glScalef(1.15,1.15,1.15);
glPushMatrix();
    glTranslatef(0.0,1.0,0.0);
    glColor3f(0,0,0);
    glScalef(1.5, 0.25, 5.0);
    glutSolidCube(1.0);
glPopMatrix();

glPushMatrix();
    glTranslatef(0.0,0.75,2.5);
    glScalef(1.5, 0.25, 2.0);
    glColor3f(0,0,0);
    glutSolidCube(1.0);
glPopMatrix();
// roda depan kiri
glPushMatrix();
    glTranslatef(0.9, 0.5, 2.5);
    glRotatef(90,0,1,0);
    glColor3f(0,0,0);
    gluCylinder(quadr, 0.5, 0.5, 0.3, 32, 32);
glPopMatrix();

glPushMatrix();
    glTranslatef(1.15, 0.5, 2.5);
    glRotatef(90,0,1,0);
    gluDisk(quadr, 0, 0.5, 32, 32);
glPopMatrix();
// roda depan kanan
glPushMatrix();
    glTranslatef(-1.15, 0.5, 2.5);
    glRotatef(90,0,1,0);
    glColor3f(0,0,0);
    gluCylinder(quadr, 0.5, 0.5, 0.3, 32, 32);
glPopMatrix();

```



Commits on May 25, 2022

Add files via upload

pensilbujel committed 2 days ago

Verified



ef09247



Added Imagerloader and image

pensilbujel committed 2 days ago

Verified



d5745db



Added image to the trailer's back

pensilbujel committed 2 days ago

Verified



590bd6a



Added colors to the truck

pensilbujel committed 2 days ago

Verified



eb8d348



Commits on May 14, 2022

Added void TrukGandeng()

pensilbujel committed 13 days ago

Verified



4a0ea13



## Lampiran 4 : Bagian yang dikerjakan Muhamad Ridwan Ash'shidqi (24060121130075)

[\*] pohon.h

```
1 // Nama file : pohon.h
2 // Deskripsi : Prototype untuk file pohon.cpp
3 // Pembuat : Muhamad Ridwan Ash'shidqi
4 // Tanggal : 28 April 2022
5 // Lab : Praktikum B2
6
7 #ifndef pohon_h
8 #define pohon_h
9
10 void pohon1();
11 void pohon2();
12 void pohon3();
13 void masjid();
14
15 #endif /* pohon_h */
```

[\*] pohon.cpp

```
1 #include "pohon.h"
2 #include <math.h>
3 #ifdef __APPLE__ // untuk program yang dijalankan di MacOS
4 #include <GLUT/glut.h>
5 #include <OpenGL/gl.h>
6 #else // untuk program yang dijalankan di Windows
7 #include <GL/glut.h>
8 #endif
9
10
11 void pohon1()
12 {
13
14 // POHON 1
15 //===== batang pohon =====
16 GLUQuadraticObj * quadratic;
17 quadratic = gluNewQuadric();
18 glPushMatrix();
19 glColor3ub(62,20,0);
20 glRotatef(90,-20,0,0);
21 glScalef(2,2,2);
22 gluCylinder(quadratic,0.35f,0.35f,2.0f,32,32);
23 glPopMatrix();
24
25 //=====daun pohon=====
26 // bawah
27 glColor3f(0,1,0);
28 glPushMatrix();
29 glTranslatef(0,1.8,0);
30 glRotatef(90,-20,0,0);
31 glutSolidCone(2,5,4,4);
32 glPopMatrix();
```



```

35 // atas
36 glPushMatrix();
37 glTranslatef(0,3.8,0);
38 glRotatef(90,-20,0,0);
39 glutSolidCone(2,5,4,4);
40 glPopMatrix();
41
42
43 }
44
45 void pohon2()
46 {
47     // pohon daun bulet
48     // batang
49     GLUQuadricObj * quadratic;
50     quadratic = gluNewQuadric();
51
52     glPushMatrix();
53     glColor3f(8.0, 0.5, 0);
54     glTranslatef(14,0,0);
55     glRotatef(90,-2,0,0);
56     glutSolidCone(0.7,5,40,40);
57     glPopMatrix();
58
59     // daun
60     //===== baris 1 =====
61     // baris 1 depan kanan
62     glColor3ub(124,252,0);
63
64     glPushMatrix();
65     glTranslatef(14.7,3.3,0.98);
66     glRotatef(90,-2,0,0);
67     glutSolidSphere(1,50,50);
68     glPopMatrix();

```

```

69 // baris 1 depan kiri
70 glPushMatrix();
71 glTranslatef(13.4,3.3,0.98);
72 glRotatef(90,-2,0,0);
73 glutSolidSphere(1,50,50);
74 glPopMatrix();
75 // baris 1 dbelakang kiiri
76 glPushMatrix();
77 glTranslatef(13.4,3.3,-0.98);
78 glRotatef(90,-2,0,0);
79 glutSolidSphere(1,50,50);
80 glPopMatrix();
81 // baris 1 belakang kanan
82 glPushMatrix();
83 glTranslatef(14.7,3.3,-0.98);
84 glRotatef(90,-2,0,0);
85 glutSolidSphere(1,50,50);
86 glPopMatrix();
87
88 //samping kiri
89 glPushMatrix();
90 glTranslatef(13,3.3,0.0);
91 glRotatef(90,-2,0,0);
92 glutSolidSphere(1,50,50);
93 glPopMatrix();
94
95 //samping kanan
96 glPushMatrix();
97 glTranslatef(15.2,3.3,0.0);
98 glRotatef(90,-2,0,0);
99 glutSolidSphere(1,50,50);
100 glPopMatrix();
101 //===== baris 2=====

```

```

101 //===== baris 2=====
102 // depan kanan
103 glPushMatrix();
104 glTranslatef(14.5,4.5,0.8);
105 glRotatef(90,-2,0,0);
106 glutSolidSphere(0.8,50,50);
107 glPopMatrix();
108 // depan kiri
109 glPushMatrix();
110 glTranslatef(13.5,4.5,0.8);
111 glRotatef(90,-2,0,0);
112 glutSolidSphere(0.8,50,50);
113 glPopMatrix();
114 // belakang kanan
115 glPushMatrix();
116 glTranslatef(14.7,4.5,-0.8);
117 glRotatef(90,-2,0,0);
118 glutSolidSphere(0.8,50,50);
119 glPopMatrix();
120 // belakang kiri
121 glPushMatrix();
122 glTranslatef(13.4,4.5,-0.8);
123 glRotatef(90,-2,0,0);
124 glutSolidSphere(0.8,50,50);
125 glPopMatrix();
126
127 //samping kiri
128 glPushMatrix();
129 glTranslatef(13,4.6,0.0);
130 glRotatef(90,-2,0,0);
131 glutSolidSphere(0.8,50,50);
132 glPopMatrix();
133

```

```

134 //samping kanan
135 glPushMatrix();
136 glTranslatef(15.2,4.6,0.0);
137 glRotatef(90,-2,0,0);
138 glutSolidSphere(0.8,50,50);
139 glPopMatrix();
140 // ===== baris 3 =====
141 glPushMatrix();
142 glTranslatef(14,5.7,0);
143 glRotatef(90,-2,0,0);
144 glutSolidSphere(0.9,50,50);
145 glPopMatrix();
146 }
147
148 void pohon3()
149 {
150     // POHON tandus
151     GLUQuadricObj * quadratic;
152     quadratic = gluNewQuadric();
153
154     // batang berdir
155     glPushMatrix();
156     glColor3ub(62,20,0);
157     glTranslatef(8,0,0);
158     glRotatef(90,-2,0,0);
159     glScalef(2,2,2);
160     gluCylinder(quadratic,0.35f,0.35f,5.0f,32,32);
161     glPopMatrix();
162 }

```

```

163 //batang kanan
164 glPushMatrix();
165 glColor3ub(62,20,0);
166 glTranslatef(8,3.3,0);
167 glRotatef(-93,20,-20,8);
168 glScalef(2,2,2);
169 gluCylinder(quadratic,0.35f,0.35f,2.0f,10,10);
170 glPopMatrix();
171
172 //batang kiri
173 glPushMatrix();
174 glColor3ub(62,20,0);
175 glTranslatef(8,5,0);
176 glRotatef(180,30,-25,-30);
177 glScalef(2,2,2);
178 gluCylinder(quadratic,0.35f,0.35f,2.2f,10,10);
179 glPopMatrix();
180 }

```

```

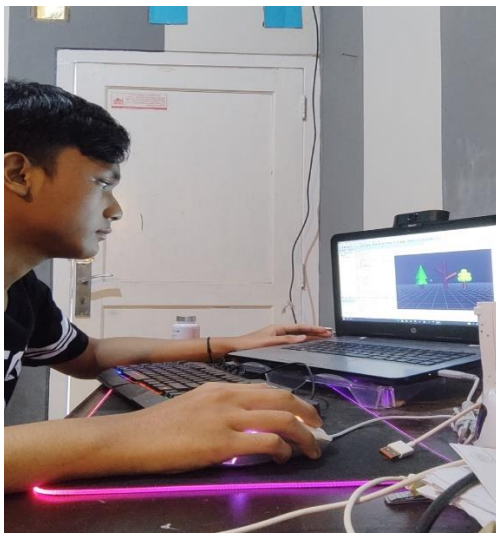
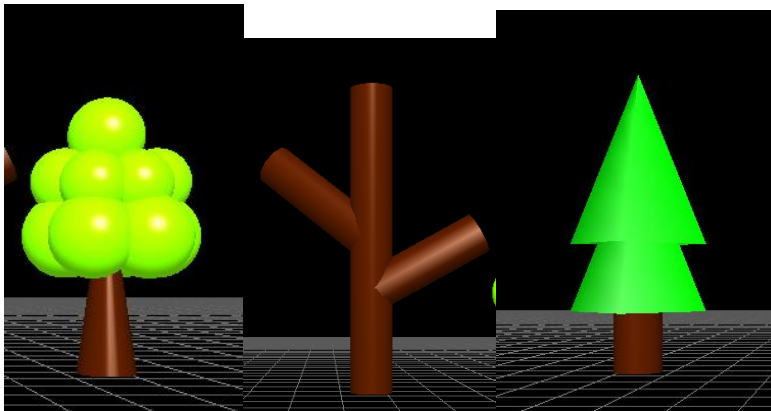
182 void masjid()
183 {
184     // ==== badan masjid =====
185     glPushMatrix();
186     glColor3f(1, 1, 1);
187     glTranslatef(3,4,3);
188     glScalef(6,2.4,6);
189     glutSolidCube(3.4);
190     glPopMatrix();
191
192     //===== kubah =====
193     glPushMatrix();
194     glColor3ub(255,255,0);
195     glTranslatef(3,8,3);
196     glutSolidSphere(5,40,40);
197     glPopMatrix();
198
199     // ==== menara =====
200     GLUQuadric * quadratic;
201     quadratic = gluNewQuadric();
202     // ==== kanan depan =====
203     glColor3f(0, 0, 0.8);
204     glPushMatrix();
205     glTranslatef(11.4,13,11.3);
206     glRotatef(90,20,0,0);
207     gluCylinder(quadratic,2.7,2.7,13,50,50);
208     glPopMatrix();
209     // ==== kiri depan =====
210     glPushMatrix();
211     glTranslatef(-5.4,13,11.3);
212     glRotatef(90,20,0,0);
213     gluCylinder(quadratic,2.7,2.7,13,50,50);
214     glPopMatrix();

```

```

215 // ==== kiri belakang =====
216 glPushMatrix();
217 glTranslatef(-5.4,13,-5.3);
218 glRotatef(90,20,0,0);
219 gluCylinder(quadratic,2.7,2.7,13,50,50);
220 glPopMatrix();
221 // ==== kanan belakang =====
222 glPushMatrix();
223 glTranslatef(11.4,13,-5.3);
224 glRotatef(90,20,0,0);
225 gluCylinder(quadratic,2.7,2.7,13,50,50);
226 glPopMatrix();
227
228 // ===== pintu =====
229 // depan
230 glPushMatrix();
231 glColor3ub(160,84,45);
232 glTranslatef(-1,-0.1,13.25);
233 glBegin(GL_POLYGON);
234 glVertex3f(2,0,0);
235 glVertex3f(6,0,0);
236 glVertex3f(6,3,0);
237 glVertex3f(2,3,0);
238 glEnd();
239 glPopMatrix();
240 }
241

```



## Lampiran 5 : Bagian yang dikerjakan Labiba Adinda Zahwana (24060121140111)

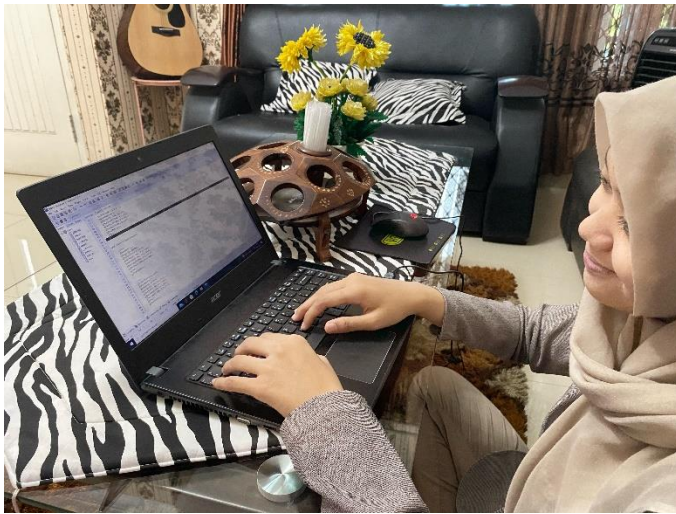
```
[*]jalan.h
1 // Nama file : jalan.h
2 // Deskripsi : Prototype untuk file jalan.cpp
3 // Pembuat : Labiba Adinda Zahwana
4 // Tanggal : 28 April 2022
5 // Lab : Praktikum B2
6
7
8 #ifndef jalan_hpp
9 #define jalan_hpp
10
11 #include <stdio.h>
12 class Jalan {
13 public:
14     void jalan();
15     void pinggirJalan();
16 };
17
18 #endif /* jalan_hpp */
19
```

```
[*]jalan.cpp
1 // Nama file : jalan.cpp
2 // Deskripsi : membuat jalan
3 // Pembuat : Labiba Adinda Zahwana
4 // Tanggal : 28 April 2022
5 // Lab : Praktikum B2
6
7
8 #include "jalan.h"
9 #ifdef __APPLE__ // untuk program yang dijalankan di MacOS
10 #include <GLUT/glut.h>
11 #include <OpenGL/gl.h>
12 #else // untuk program yang dijalankan di Windows
13 #include <GL/glut.h>
14 #endif
15
16
17 void Jalan::jalan() {
18     glColor3f(1.0,0.75,0.0);
19     glBegin(GL_QUADS);
20
21     //Bagian Kanan Jalan
22     glBegin(GL_QUADS);
23     glColor3f(0.7,0.7,0.7);
24     glPushMatrix();
25     glVertex3f(-100,-0.1,-10.0);
26     glVertex3f(-100,-0.1,0.10);
27     glVertex3f(100,-0.1,0.10);
28     glVertex3f(100,-0.1,-10.0);
29     glPopMatrix();
30     glEnd();
31
32
33
34
35     // Garis tengah jalan
36     glPushMatrix();
37     glColor3f(1.0,1.0,1.0);
38     glTranslatef(0, 0.1, 0);
39     glBegin(GL_QUADS);
40     glVertex3f(-100,-0.1,-1.0);
41     glVertex3f(-100,-0.1,1.);
42     glVertex3f(-50,-0.1,1.);
43     glVertex3f(-50,-0.1,-1.0);
44
45     glVertex3f(0,-0.1,1.);
46     glVertex3f(0,-0.1,-1);
47     glVertex3f(50,-0.1,-1);
48     glVertex3f(50,-0.1,1);
49     glEnd();
50     glPopMatrix();
51
52
53     glBegin(GL_QUADS);
54     //Bagian Kiri Jalan
55     glColor3f(0.7,0.7,0.7);
56     glPushMatrix();
57     glVertex3f(-100,-0.1,0);
58     glVertex3f(-100,-0.1,10);
59     glVertex3f(100,-0.1,10);
60     glVertex3f(100,-0.1,0);
61     glPopMatrix();
62     glEnd();
63
64
65 }
```

```

66
67 void Jalan::pinggiranJalan()
68 {
69     //kanan
70     glPushMatrix();
71     glBegin(GL_QUADS);
72     glColor3f(0.0,0.7,0.0);
73     glVertex3f(-200,-0.1,-500);
74     glVertex3f(-200,-0.1,-25.0);
75     glVertex3f(1000,-0.1,-25.0);
76     glVertex3f(1000,-0.1,-500);
77     glEnd();
78     glPopMatrix();
79
80
81     //kiri
82     glColor3f(0.0,0.7,0.0);
83     glPushMatrix();
84     glBegin(GL_QUADS);
85     glVertex3f(-200,-0.1,25.0);
86     glVertex3f(-200,-0.1,500);
87     glVertex3f(1000,-0.1,500);
88     glVertex3f(1000,-0.1,25.0);
89     glEnd();
90     glPopMatrix();
91 }

```



## Lampiran 6 : Link Video Youtube

<https://youtu.be/mfg4lQBNmgQ>