



KERJA PRAKTIK - EF234603

**Pengembangan dan Implementasi Skenario Permainan Kuis
Gamelab dengan Kerangka Kerja Phaser**

PT. Educa Sisfomedia Indonesia

Jl. Gilingrejo No.10, Gendongan, Kec. Tingkir, Kota Salatiga,
Jawa Tengah 50743

Periode: 15 Januari 2024 - 5 April 2024

Oleh:

Beryl 5025201029

Brian Akbar Wicaksana 5025201207

Pembimbing Jurusan

Dr. Dwi Sunaryono, S.Kom., M.Kom.

Pembimbing Lapangan

Samsul Huda

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2024



KERJA PRAKTIK - EF234603

**Pengembangan dan Implementasi Skenario Permainan Kuis
Gamelab dengan Kerangka Kerja Phaser**

PT. Educa Sisfomedia Indonesia

Jl. Gilingrejo No.10, Gendongan, Kec. Tingkir, Kota Salatiga, Jawa
Tengah 50743

Periode: 15 Januari 2024 - 5 April 2024

Oleh:

Beryl 5025201029

Brian Akbar Wicaksana 5025201207

Pembimbing Jurusan

Dr. Dwi Sunaryono, S.Kom., M.Kom.

Pembimbing Lapangan

Samsul Huda

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2024

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

| | |
|---|------|
| DAFTAR ISI | iv |
| DAFTAR GAMBAR | viii |
| DAFTAR KODE SUMBER | x |
| LEMBAR PENGESAHAN | xii |
| KATA PENGANTAR | xvi |
| BAB I PENDAHULUAN | 1 |
| 1.1. Latar Belakang | 1 |
| 1.2. Tujuan | 1 |
| 1.3. Manfaat | 1 |
| 1.4. Rumusan Masalah | 2 |
| 1.5. Lokasi dan Waktu Kerja Praktik | 2 |
| 1.6. Metodologi Kerja Praktik | 2 |
| 1.6.1. Perumusan Masalah | 2 |
| 1.6.2. Studi Literatur | 2 |
| 1.6.3. Analisis dan Perancangan Game | 3 |
| 1.6.4. Implementasi Game | 3 |
| 1.6.5. Pengujian dan Evaluasi | 3 |
| 1.6.6. Kesimpulan dan Saran | 3 |
| 1.7. Sistematika Laporan | 3 |
| 1.7.1. Bab I Pendahuluan | 3 |
| 1.7.2. Bab II Profil Perusahaan | 3 |
| 1.7.3. Bab III Tinjauan Pustaka | 3 |

| | | |
|--|--|-----------|
| 1.7.4. | Bab IV Analisis dan Perancangan Game | 4 |
| 1.7.5. | Bab V Implementasi Game | 4 |
| 1.7.6. | Bab VI Pengujian dan Evaluasi..... | 4 |
| 1.7.7. | Bab VII Kesimpulan dan Saran | 4 |
| BAB II PROFIL PERUSAHAAN | | 6 |
| 2.1. | Profil Gamelab..... | 6 |
| 2.2. | Lokasi..... | 6 |
| 2.3. | Program | 6 |
| BAB III TINJAUAN PUSTAKA..... | | 8 |
| 3.1. | Phaser | 8 |
| 3.2. | HTML5 | 8 |
| 3.3. | Javascript..... | 9 |
| 3.4. | WebGL..... | 9 |
| BAB IV ANALISIS DAN PERANCANGAN GAME..... | | 11 |
| 4.1. | Analisis Game | 11 |
| 4.1.1. | Menghitung Ceri..... | 11 |
| 4.1.2. | Menghitung Penumpang | 12 |
| 4.2. | Perancangan Game | 12 |
| 4.2.1. | Menghitung Ceri | 12 |
| 4.2.2. | Menghitung Penumpang..... | 18 |
| BAB V IMPLEMENTASI GAME | | 27 |
| 5.1. | Teknologi | 27 |
| 5.2. | Implementasi Logika Game | 27 |

| | | |
|--------------------------------|------------------------------------|-----------|
| 5.2.1. | Menghitung Ceri | 27 |
| 5.2.2. | Menghitung Penumpang..... | 36 |
| BAB VI | PENGUJIAN DAN EVALUASI..... | 42 |
| 6.1. | Tujuan Pengujian | 42 |
| 6.2. | Skenario Pengujian..... | 42 |
| BAB VII | KESIMPULAN DAN SARAN | 44 |
| 7.1. | Kesimpulan | 44 |
| 7.2. | Saran..... | 44 |
| DAFTAR PUSTAKA | | 46 |
| BIODATA PENULIS I | | 48 |

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

| | |
|--|----|
| Gambar 4.1. Game acuan Menghitung Ceri | 11 |
| Gambar 4.2. Game acuan Menghitung Penumpang..... | 12 |
| Gambar 4.3 Flowchart Menghitung Ceri | 14 |
| Gambar 4.4 Desain pertanyaan penjumlahan | 15 |
| Gambar 4.5 Desain saat memilih jawaban | 16 |
| Gambar 4.6 Desain saat monster meminta ceri..... | 16 |
| Gambar 4.7 Desain pertanyaan pengurangan..... | 17 |
| Gambar 4.8 Desain setelah menjawab pertanyaan pengurangan | 18 |
| Gambar 4.9 Flowchart Menghitung Penumpang..... | 20 |
| Gambar 4.10 Desain pertanyaan menghitung jumlah | 21 |
| Gambar 4.11 Desain saat memilih jawaban | 22 |
| Gambar 4.12 Desain pertanyaan penjumlahan..... | 23 |
| Gambar 4.13 Desain selesai putaran | 24 |
| Gambar 4.14 Desain akhir permainan | 25 |
| Gambar 5.1 Tampilan website gamelab saat memasukkan aset | 27 |

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

| | |
|---|----|
| Kode Sumber 5.1. Implementasi Pertanyaan Penjumlahan | 29 |
| Kode Sumber 5.2. Implementasi Permintaan Monster..... | 32 |
| Kode Sumber 5.3. Implementasi Pertanyaan Pengurangan | 36 |
| Kode Sumber 5.4. Implementasi Pertanyaan Menghitung | 37 |
| Kode Sumber 5.5. Implementasi Tampilan Akhir Permainan | 40 |

[Halaman ini sengaja dikosongkan]

**LEMBAR PENGESAHAN
KERJA PRAKTIK**

**Pengembangan dan Implementasi Skenario Permainan
Kuis Gamelab dengan Kerangka Kerja Phaser**

Oleh:

| | |
|-----------------------|------------|
| Beryl | 5025201029 |
| Brian Akbar Wicaksana | 5025201207 |

Disetujui oleh Pembimbing Kerja Praktik:

1. Dr. Dwi Sunaryono, S.Kom.,
M.Kom.
NIP. 197205281997021001



(Pembimbing Departemen)

2. Samsul Huda



(Pembimbing Lapangan)

[Halaman ini sengaja dikosongkan]

Pengembangan dan Implementasi Skenario Permainan Kuis Gamelab dengan Kerangka Kerja Phaser

Nama Mahasiswa : Beryl
NRP : 5025201029
Nama Mahasiswa : Brian Akbar Wicaksana
NRP : 5025201207
Departemen : Teknik Informatika FTEIC-ITS
Pembimbing Departemen : Dr. Dwi Sunaryono, S.Kom.,
M.Kom.
Pembimbing Lapangan : Samsul Huda

ABSTRAK

Gamelab adalah platform magang, pelatihan, kelas industri SMK, dan karir yang didirikan oleh Educa Studio. Dalam kerja praktik ini, kami diminta untuk mengembangkan game edukasi berbasis kuis yang dapat memperkaya stok produk game Educa. Proses pengembangan meliputi analisis kebutuhan, perancangan alur permainan dan aset visual, implementasi menggunakan bahasa pemrograman JavaScript dan game engine Phaser, serta pengujian oleh trainer untuk memastikan fungsionalitas dan kinerja game. Game yang telah dibuat berhasil mencapai tujuan dan ketentuan yang diharapkan oleh pihak Gamelab dan memiliki potensi untuk dikembangkan lebih lanjut dengan menambahkan fitur-fitur tambahan dan meningkatkan performa..

Kata Kunci : Game, JavaScript, Phaser, Kuis

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT atas penyertaan dan karunia-Nya sehingga penulis dapat menyelesaikan salah satu kewajiban penulis sebagai mahasiswa Departemen Teknik Informatika ITS yaitu Kerja Praktik yang berjudul: Pengembangan dan Implementasi Skenario Permainan Kuis Gamelab dengan Kerangka Kerja Phaser.

Penulis menyadari bahwa masih banyak kekurangan baik dalam melaksanakan kerja praktik maupun penyusunan buku laporan kerja praktik ini. Namun penulis berharap buku laporan ini dapat menambah wawasan pembaca dan dapat menjadi sumber referensi.

Melalui buku laporan ini penulis juga ingin menyampaikan rasa terima kasih kepada orang-orang yang telah membantu menyusun laporan kerja praktik baik secara langsung maupun tidak langsung antara lain:

1. Kedua orang tua penulis.
2. Bapak Dr. Dwi Sunaryono, S.Kom., M.Kom. selaku dosen pembimbing kerja praktik sekaligus koordinator kerja praktik.
3. Bapak Samsul Huda selaku pembimbing lapangan selama kerja praktik berlangsung.
4. Teman-teman penulis yang senantiasa memberikan semangat ketika penulis melaksanakan KP.

Surabaya, 20 Juni 2024
Beryl dan Brian Akbar Wicaksana

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

1.1. Latar Belakang

Gamelab merupakan platform yang lengkap untuk meningkatkan kompetensi lulusan yang siap kerja dan siap wirausaha melalui program pelatihan berbasis proyek (PBL), magang online yang bersertifikat, dan sertifikasi industri. Gamelab menggunakan kurikulum berbasis industri yang dipadukan dengan metode PBL (Project-based Learning) sehingga pelatihan menjadi lebih berkualitas. Pelatihan di Gamelab juga didukung oleh pengajar ahli di bidangnya serta teknologi yang khusus sehingga belajar tidak hanya sebatas menonton video, melainkan juga melaksanakan praktik langsung. Gamelab menyediakan beberapa kategori magang untuk bidang pemrograman, di antaranya adalah Web Programming, Game Programming, dan App Programming. Pada kerja praktik ini, kami memilih bidang pemrograman kategori gim untuk memenuhi kewajiban kerja praktik sebesar 4 sks.

1.2. Tujuan

Tujuan dari pelaksanaan kerja praktik ini adalah untuk memenuhi persyaratan akademis sebesar 4 SKS dan untuk meningkatkan kompetensi kami melalui penerapan metode Pembelajaran Berbasis Proyek agar sesuai dengan tuntutan industri.

1.3. Manfaat

Kerja praktik ini memberikan sejumlah manfaat yang signifikan. Selain memenuhi persyaratan akademis yang merupakan bagian penting dari program studi, kerja praktik ini juga memberikan kesempatan untuk mengembangkan keterampilan praktis melalui penerapan metode Pembelajaran Berbasis Proyek, serta memperluas pengetahuan dan keterampilan yang relevan. Interaksi

langsung dengan dunia industri juga memberikan pemahaman yang lebih dalam tentang praktik terkini, tren, dan standar industri yang relevan.

1.4. Rumusan Masalah

Rumusan masalah dari kerja praktik ini adalah sebagai berikut:

1. Bagaimana pengembangan permainan kuis sesuai dengan ketentuan dari trainer gamelab?
2. Bagaimana proses pengembangan dan implementasi permainan kuis?

1.5. Lokasi dan Waktu Kerja Praktik

Kerja praktik ini dilaksanakan pada waktu dan tempat sebagai berikut:

Lokasi : Online
Waktu : 15 Januari – 5 April 2024
Hari Kerja : Senin – Jumat
Jam Kerja : 08.00 - 17.00

1.6. Metodologi Kerja Praktik

Metodologi dalam pembuatan buku kerja praktik meliputi :

1.6.1. Perumusan Masalah

Untuk mengetahui kebutuhan game, trainer dari Gamelab memberikan deskripsi dan ketentuan untuk game yang akan dikembangkan. Deskripsi yang diberikan mencakup tujuan proyek, ketentuan hasil dan contoh, langkah-langkah, cara pelaporan dan pengumpulan progress, serta poin penilaian. Kami juga diberikan contoh game acuan yang dijadikan standar minimal kualitas untuk game yang akan kami kembangkan. Kami diminta untuk membuat game dengan desain dan aset yang mirip dengan game acuan tersebut.

1.6.2. Studi Literatur

Setelah mendapatkan gambaran tentang game tersebut, kami diberi ketentuan untuk menggunakan

Phaser sebagai framework dan JavaScript sebagai bahasa pemrograman untuk implementasi game tersebut.

1.6.3. Analisis dan Perancangan Game

Setelah ketentuan diberikan, kami memainkan game acuan tersebut, kemudian menganalisis mekanik dan aset yang diperlukan untuk game yang akan dikembangkan.

1.6.4. Implementasi Game

Implementasi merupakan realisasi dari tahap perancangan. Pada tahap ini kami melakukan pengembangan game di website Gamelab.

1.6.5. Pengujian dan Evaluasi

Selama proses pengembangan, game yang dibuat akan diuji dan dievaluasi secara berkala oleh trainer. Jika game yang dikembangkan belum sesuai, trainer akan memberikan masukan mengenai fitur atau bug yang perlu diperbaiki.

1.6.6. Kesimpulan dan Saran

Pengujian yang dilakukan ini telah memenuhi syarat yang diinginkan, dan berjalan dengan baik dan lancar.

1.7. Sistematika Laporan

1.7.1. Bab I Pendahuluan

Bab ini berisi latar belakang, tujuan, manfaat, rumusan masalah, lokasi dan waktu kerja praktik, metodologi, dan sistematika laporan.

1.7.2. Bab II Profil Perusahaan

Bab ini berisi gambaran umum Gamelab mulai dari profil, lokasi perusahaan.

1.7.3. Bab III Tinjauan Pustaka

Bab ini berisi dasar teori dari teknologi yang digunakan dalam menyelesaikan proyek kerja praktik.

1.7.4. Bab IV Analisis dan Perancangan Game

Bab ini berisi mengenai tahap analisis game dalam menyelesaikan proyek kerja praktik.

1.7.5. Bab V Implementasi Game

Bab ini berisi uraian tahap - tahap yang dilakukan untuk proses implementasi game.

1.7.6. Bab VI Pengujian dan Evaluasi

Bab ini berisi hasil uji coba dan evaluasi dari trainer untuk game yang telah dikembangkan selama pelaksanaan kerja praktik.

1.7.7. Bab VII Kesimpulan dan Saran

Bab ini berisi kesimpulan dan saran yang didapat dari proses pelaksanaan kerja praktik.

[Halaman ini sengaja dikosongkan]

BAB II

PROFIL PERUSAHAAN

2.1. Profil Gamelab

Gamelab merupakan platform lengkap untuk meningkatkan kompetensi lulusan yang siap kerja dan siap wirausaha melalui program pelatihan berbasis proyek (PBL), magang online bersertifikat, dan sertifikasi industri. Gamelab menjembatani dunia pendidikan dan DUDIKA (Dunia Usaha & Dunia Industri Kerja) sehingga mempersiapkan lulusan SMA/SMK dan Perguruan Tinggi agar lebih kompeten dan siap Kerja siap Wirausaha.

2.2. Lokasi

Jl. Kalisombo No.18, Salatiga, Kec. Sidorejo, Kota Salatiga, Jawa Tengah, 50711

2.3. Program

Menggunakan kurikulum berbasis industri yang dipadukan dengan metode PBL (Project-based Learning) sehingga pelatihan menjadi lebih berkualitas. Pelatihan di Gamelab juga didukung oleh pengajar expert di bidangnya serta teknologi yang spesial sehingga belajar tidak hanya menonton video saja, tetapi benar-benar praktik.

[Halaman ini sengaja dikosongkan]

BAB III

TINJAUAN PUSTAKA

3.1. Phaser

Phaser adalah kerangka kerja (framework) open-source yang kuat dan fleksibel untuk pengembangan game 2D berbasis web yang dirancang dengan JavaScript. Phaser menyediakan alat dan fitur untuk menciptakan game yang dapat berjalan di browser menggunakan teknologi seperti HTML5, Canvas, dan WebGL. Hal ini memungkinkan pengembang untuk membuat berbagai jenis permainan, termasuk platformer, puzzle, arcade, dan RPG. Phaser menjadi pilihan populer di kalangan pengembang game indie dan studio game yang ingin membuat game yang menarik dan responsif dengan performa tinggi. [6]

3.2. HTML5

HTML5 adalah standar teknologi terbaru untuk pengembangan konten web, yang diadopsi secara luas untuk membangun situs web dan aplikasi web modern. HTML5 memperkenalkan berbagai elemen dan atribut baru yang memperkaya kemampuan markup dan memungkinkan integrasi yang lebih dalam dengan teknologi web lainnya seperti CSS dan JavaScript. Salah satu keunggulan utama HTML5 adalah dukungannya untuk multimedia tanpa memerlukan plugin tambahan, memungkinkan pemutaran video, audio, dan animasi langsung di browser.

Selain itu, HTML5 menyediakan kemampuan untuk pengembangan aplikasi web offline, interaktivitas yang lebih kompleks melalui canvas dan SVG, serta fitur-fitur yang mendukung aksesibilitas dan responsivitas. Hal ini menjadi peran penting dalam mendorong evolusi

pengalaman web yang lebih dinamis, interaktif, dan ramah perangkat seluler. [9]

3.3. Javascript

JavaScript adalah bahasa pemrograman tingkat tinggi yang sering digunakan untuk pengembangan aplikasi web. Sebagai bahasa scripting yang berjalan di sisi klien (client-side) pada browser, JavaScript memberikan kemampuan untuk membuat interaksi dinamis, animasi, validasi formulir, manipulasi DOM, dan pengolahan data secara langsung di dalam halaman web.

JavaScript memungkinkan pengembang untuk membangun aplikasi web yang kompleks dan responsif dengan berbagai fitur dan fungsionalitas. Selain itu, dengan kemampuannya yang semakin berkembang di luar lingkungan web melalui platform seperti Node.js, JavaScript juga digunakan secara luas untuk pengembangan server-side, IoT (Internet of Things), dan aplikasi desktop. [7]

3.4. WebGL

WebGL (Web Graphics Library) adalah API berbasis JavaScript yang memungkinkan rendering grafis 2D dan 3D dalam browser web tanpa memerlukan plugin tambahan. Dibangun di atas standar OpenGL ES, WebGL memberikan pengembang web akses langsung ke akselerasi grafis perangkat keras, memungkinkan konten grafis yang kaya dan kompleks untuk berjalan secara efisien di dalam browser. Fitur-fitur WebGL mencakup dukungan untuk shader, tekstur, dan rendering real-time, memberikan kebebasan kreatif untuk menciptakan efek visual yang canggih dan realistis. Selain itu, WebGL terintegrasi erat dengan teknologi web lainnya seperti HTML5 dan WebAssembly, memungkinkan pembuatan aplikasi web yang interaktif dan imersif dengan performa tinggi dan portabilitas yang luas. [8]

[Halaman ini sengaja dikosongkan]

BAB IV ANALISIS DAN PERANCANGAN GAME

4.1. Analisis Game

4.1.1. Menghitung Ceri

4.1.1.1. Definisi Umum

Game ini merupakan permainan kuis di mana pemain akan diberikan pertanyaan tentang penjumlahan dan pengurangan. Terdapat dua bagian dalam game ini. Pada bagian pertanyaan penjumlahan, pemain harus menjawab jumlah buah ceri yang ada di meja. Pada bagian pertanyaan pengurangan, monster akan memakan beberapa buah ceri yang ada di meja, dan pemain harus menebak berapa banyak buah ceri yang dimakan oleh monster. Gambar 4.1 menunjukkan gim acuan dari Menghitung Ceri.



Gambar 4.1. Game acuan Menghitung Ceri

4.1.2. Menghitung Penumpang

4.1.2.1. Definisi Umum

Game ini berupa kuis di mana pemain akan diberikan pertanyaan tentang penjumlahan. Pemain harus memerhatikan penumpang yang naik ke dalam bus pada bagian pertama dan kedua, dan pemain harus menjumlahkan total penumpang dari bagian pertama dan kedua. Gambar 4.2 menunjukkan gim acuan dari Menghitung Penumpang.



Gambar 4.2. Game acuan Menghitung Penumpang

4.2. Perancangan Game

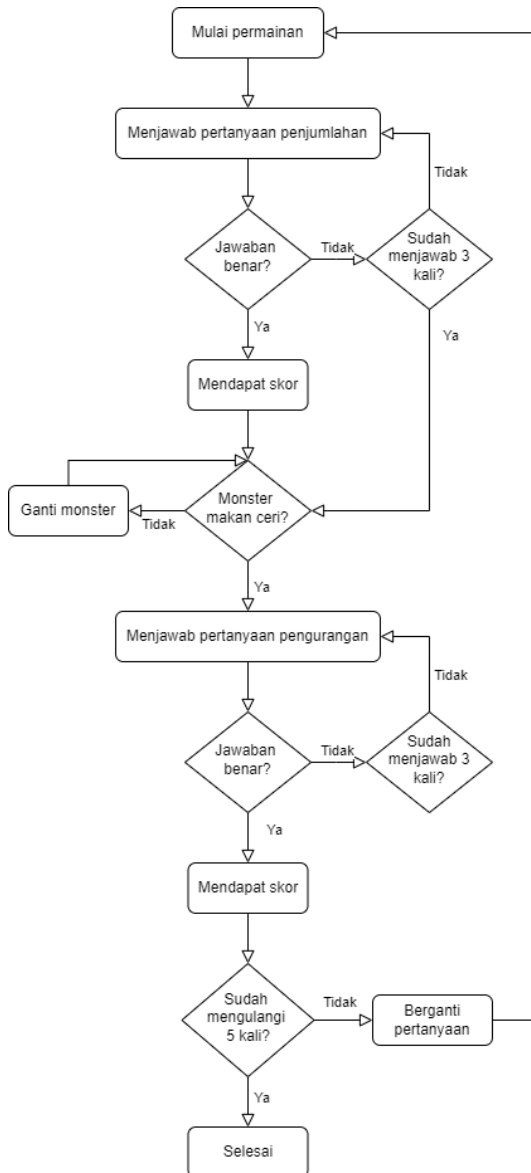
4.2.1. Menghitung Ceri

4.2.1.1. Mekanisme Gameplay

Permainan dimulai dengan pemain masuk ke sesi permainan dan diberikan pertanyaan jumlah ceri yang harus dijawab. Jika jawaban benar, pemain mendapatkan

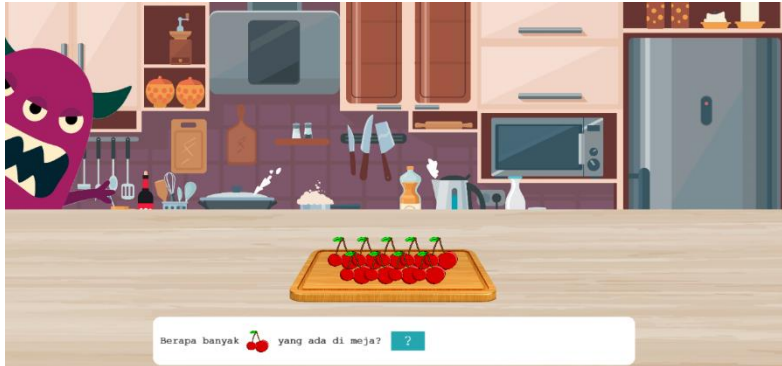
skor dan permainan lanjut ke tahap berikutnya untuk memeriksa apakah monster makan ceri. Jika jawaban salah, pemain harus menjawab ulang hingga tiga kali sebelum pertanyaan diganti dengan yang baru. Jika pemain mengizinkan monster makan ceri, permainan berlanjut ke pertanyaan pengurangan yang juga harus dijawab oleh pemain. Jika jawaban benar, pemain mendapatkan skor dan proses berlanjut untuk memeriksa apakah sudah mengulangi lima kali. Jika jawaban salah, pemain harus menjawab ulang hingga tiga kali sebelum pertanyaan diganti.

Siklus ini terus berlanjut hingga pemain telah menjawab pasangan pertanyaan penjumlahan dan pengurangan sebanyak 5 kali. Pada saat itu, pemain akan diberi skor akhir dan permainan selesai. Gambar 4.3 menunjukkan flowchart permainan Menghitung Ceri.



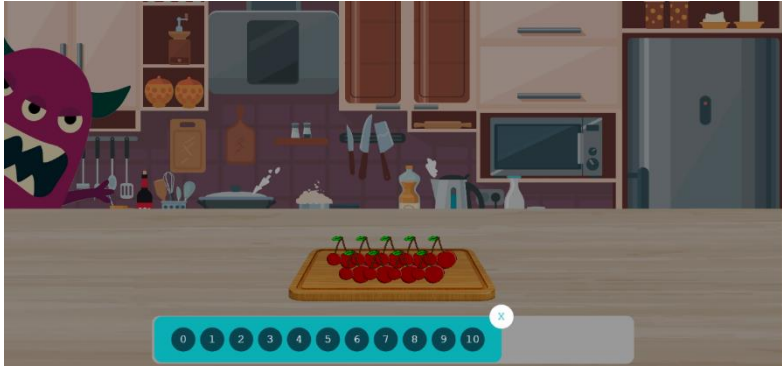
Gambar 4.3 Flowchart Menghitung Ceri

4.2.1.2. Antarmuka Pengguna



Gambar 4.4 Desain pertanyaan penjumlahan

Gambar 4.4 menunjukkan tampilan saat pemain diberi pertanyaan penjumlahan. Tampilan antarmuka pengguna ini menampilkan sebuah permainan kuis dengan latar belakang dapur. Di sisi kiri layar, terdapat karakter monster berwarna ungu. Di tengah meja, tersusun beberapa buah ceri berwarna cerah yang harus dihitung oleh pemain sebagai bagian dari pertanyaan penjumlahan.



Gambar 4.5 Desain saat memilih jawaban

Gambar 4.5 menunjukkan tampilan saat pemain memilih jawaban. Tampilan ini menampilkan desain saat pemain hendak memilih jawaban. Dapat dilihat pemain dapat memilih jawaban dari 0 sampai 10.



Gambar 4.6 Desain saat monster meminta ceri

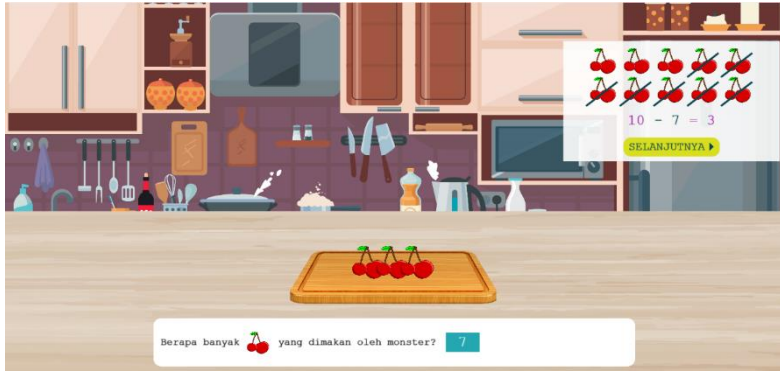
Gambar 4.6 menunjukkan tampilan saat monster meminta ceri. Tampilan ini menampilkan desain saat monster meminta pemain untuk memakan ceri. Pemain dapat memberikan ceri ataupun menolak permintaan

monster. Jika pemain menolak, monster itu akan pergi dan digantikan dengan monster yang lain. Jika pemain memberi ceri, monster akan memakan ceri dan pemain akan masuk ke tahap selanjutnya.



Gambar 4.7 Desain pertanyaan pengurangan

Gambar 4.7 menunjukkan tampilan saat pemain diberi pertanyaan pengurangan. Tampilan ini menampilkan saat pemain berada di tahap pertanyaan pengurangan. Pemain harus menebak berapa buah ceri yang dimakan oleh monster. Jika pemain tidak ingat, pemain dapat menekan tombol "Tonton Lagi" untuk menonton ulang monster memakan ceri.



Gambar 4.8 Desain setelah menjawab pertanyaan pengurangan

Gambar 4.8 menunjukkan tampilan setelah pemain menjawab pertanyaan pengurangan. Tampilan ini menampilkan desain setelah pemain menjawab pertanyaan pengurangan. Jika pemain menjawab salah, pemain akan diberi 3 kali kesempatan untuk menjawab benar sebelum ditunjukkan jawaban yang benar.

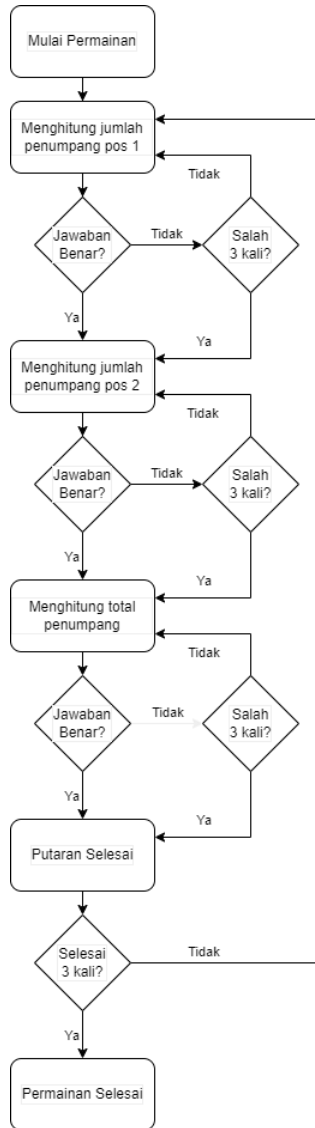
4.2.2. Menghitung Penumpang

4.2.2.1. Mekanisme Gameplay

Permainan dimulai ketika pemain menekan tombol start dan pemain diberikan pertanyaan berupa penumpang yang ada di dalam bus. Pada bagian pertama dan kedua, pemain diberikan tugas untuk menghitung penumpang yang masuk pada masing masing pos. Pada bagian ketiga, pemain harus menjumlahkan penumpang yang naik pada pos pertama dan kedua. Jika jawaban yang dipilih tepat, pemain akan diberikan poin dan dapat melanjutkan ke putaran selanjutnya. Namun, jika pemain memilih jawaban yang salah sebanyak tiga kali, permainan akan

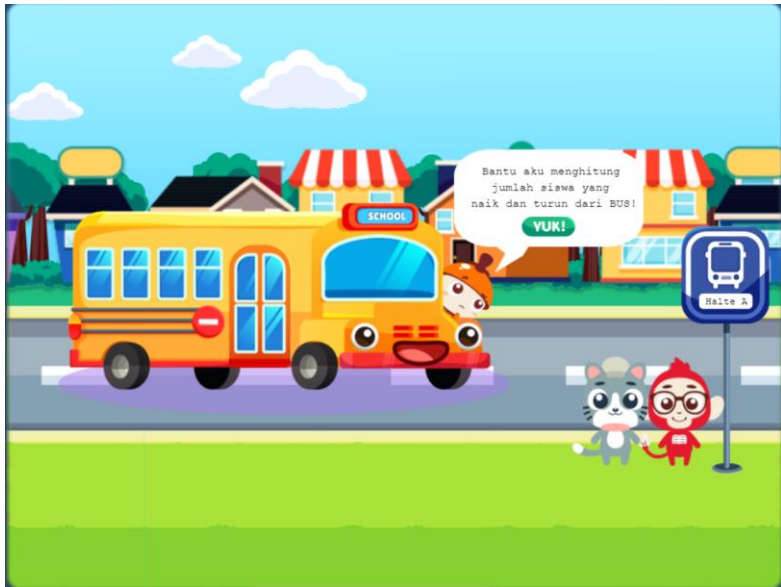
memperlihatkan Kembali berapa penumpang yang ada didalam bus dan memperbolehkan pemain untuk melanjutkan ke bagian berikutnya.

Permainan akan terus berulang hingga pemain menjawab set pertanyaan penjumlahan sebanyak 3 kali. Pada saat itu, pemain akan diberi skor akhir dan permainan selesai. Gambar 4.9 menunjukkan flowchart permainan Menghitung Penumpang.



Gambar 4.9 Flowchart Menghitung Penumpang

4.2.2.2. Antarmuka Pengguna



Gambar 4.10 Desain pertanyaan menghitung jumlah

Gambar 4.10 menunjukkan tampilan saat pemain diberi pertanyaan menghitung jumlah. Tampilan antarmuka pengguna ini menampilkan sebuah permainan kuis dengan latar belakang kota. Di tengah layar, terdapat sebuah bus. Di sisi kanan, terdapat sebaris penumpang yang harus dihitung oleh pemain sebagai bagian dari pertanyaan menghitung.



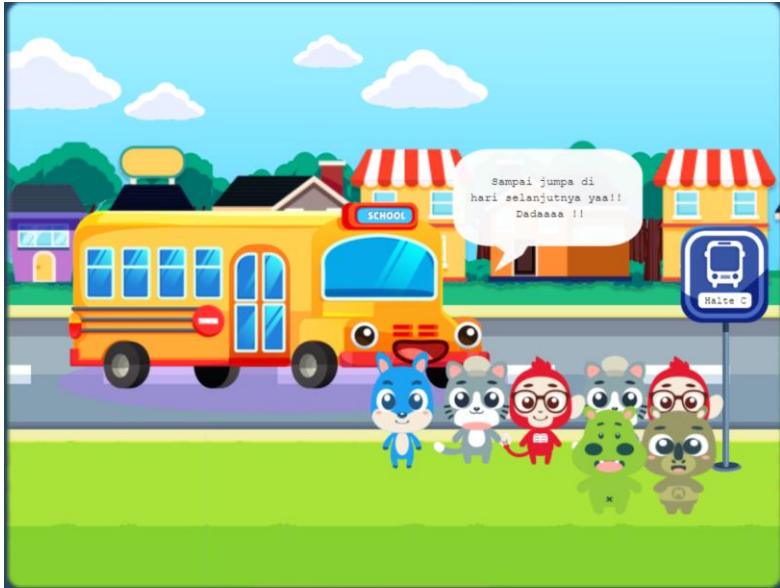
Gambar 4.11 Desain saat memilih jawaban

Gambar 4.11 menunjukkan tampilan saat pemain memilih jawaban. Tampilan ini menampilkan desain saat pemain hendak memilih jawaban. Dapat dilihat pemain dapat memilih jawaban dari 1 sampai 10.



Gambar 4.12 Desain pertanyaan penjumlahan

Gambar 4.12 menunjukkan tampilan saat pemain diberi pertanyaan penjumlahan. Tampilan ini menampilkan saat pemain berada di tahap pertanyaan penjumlahan. Pemain harus menghitung berapa jumlah penumpang yang ada di dalam bus.



Gambar 4.13 Desain selesai putaran

Gambar 4.13 menunjukkan tampilan setelah selesai putaran. Tampilan ini menampilkan desain setelah pemain berhasil menjawab semua pertanyaan pada putaran saat itu. Jumlah penumpang ditampilkan pada layar.



Gambar 4.14 Desain akhir permainan

Gambar 4.14 menunjukkan tampilan saat akhir permainan. Tampilan ini menampilkan desain setelah pemain berhasil melewati 3 putaran. Skor akhir berikut tombol untuk Kembali bermain ditampilkan pada layar.

[Halaman ini sengaja dikosongkan]

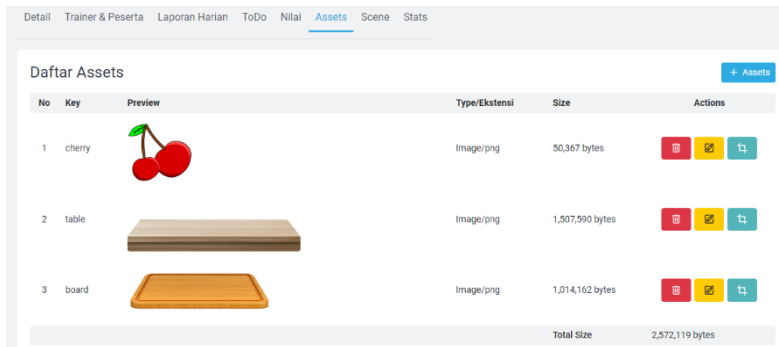
BAB V













IMPLEMENTASI GAME

Bab ini menjelaskan proses implementasi dari game yang telah dirancang, mulai dari teknologi yang digunakan, pengelolaan aset, kode program, hingga game dapat dijalankan. Penjelasan ini mencakup detail teknis mengenai cara game diimplementasikan dan dioptimalkan untuk memastikan kinerja yang baik.

5.1. Teknologi

Game ini dikembangkan menggunakan bahasa pemrograman JavaScript dengan memanfaatkan Phaser sebagai game engine. Desain grafis dan aset lainnya diambil dari aset gratis yang beredar di internet. Sementara pengelolaan proyek dan aset dilakukan di dalam website Gamelab. Gambar 5.1 menunjukkan tampilan website gamelab saat memasukkan aset.



| No | Key | Preview | Type/Ekstensi | Size | Actions |
|------------|--------|---|---------------|-----------------|---|
| 1 | cherry |  | image/png | 50,367 bytes |    |
| 2 | table |  | image/png | 1,507,590 bytes |    |
| 3 | board |  | image/png | 1,014,162 bytes |    |
| Total Size | | | | 2,572,119 bytes | |

Gambar 5.1 Tampilan website gamelab saat memasukkan aset

5.2. Implementasi Logika Game

5.2.1. Menghitung Ceri

5.2.1.1. Implementasi Pertanyaan Penjumlahan

Kode Sumber 5.1 menunjukkan implementasi pertanyaan penjumlahan. Bagian ini menjelaskan proses

implementasi dari fitur pertanyaan penjumlahan. Penjelasan ini mencakup detail teknis mengenai cara pertanyaan dihasilkan, ditampilkan, dan bagaimana interaksi pemain ditangani saat menjawab pertanyaan. Kode ini bertanggung jawab untuk menambahkan pertanyaan penjumlahan ke dalam container yang menampilkan pertanyaan kepada pemain. Fungsi `addSumQuestionToContainer` menerima beberapa parameter, yaitu `questionContainer` untuk menampung pertanyaan, `textString` sebagai teks pertanyaan, `padding` untuk menambahkan ruang sekitar pertanyaan, dan `cherryDelay` untuk menunda animasi buah ceri.

```
this.addSumQuestionToContainer = function(questionContainer,
textString = "Text", padding = 5, cherryDelay = 0) {
    this.next = function() {
        this.animateHideMonsterAtDoor(() =>
this.animateShowMonsterAtTable(() => {
            this.canInteractWithGame = true;

this.questionBoxContainer.getData("container").forEach(gameObject
t => gameObject.destroy());

this.addConfirmationQuestionToContainer(this.questionBoxContaine
r);
        }));
    };
    this.drawCountCherries = function(onComplete, delay = 0) {
        this.drawCountCherry = function() {
            let circle = this.drawCircle(this.cherries[i].x,
this.cherries[i].y, 22, 0xFFFFFFFF, 1.0);
            let text = this.addTextToButton(circle, i + 1, 24,
0x0A4550);
            container.push(circle, text);
            circle.alpha = 0;

```

```

        text.alpha = 0;
        this.alphaContainer(delay, 250, 1, () => {
            if (++i >= this.cherries.length)
onComplete(container);
            else this.drawCountCherry();
        }, [circle, text]);
    };
    let i = 0;
    let container = [];
    this.drawCountCherry();
};
return this.addGenericQuestionToContainer(questionContainer,
textString, padding, this.cherries.length, () => {
    this.hideContainer(0, 300, () => {
        this.next();
    });
}, (correctAnswer) => {
    this.canInteractWithGame = false;
    this.drawCountCherries((countContainer) => {
        this.animateCorrectAnswer(correctAnswer, () => {
            let nextButton = this.createNextButton(
                this.questionBoxContainer.getData('right') -
200 - 25,
                this.questionBoxContainer.getData('centerY')
- 40/2
            )
            nextButton.setData('pointerup', () => {
                this.SOUNDS['confirmClick'].play();
                nextButton.destroy();
                this.hideContainer(0, 300, () => {
                    this.canInteractWithGame = true;
                    this.next();
                    countContainer.forEach(element =>
element.destroy());
                });
            });
        });
    });
}, null, null, [], cherryDelay);
};

```

Kode Sumber 5.1. Implementasi Pertanyaan Penjumlahan

5.2.1.2. Implementasi Permintaan Monster

Kode Sumber 5.2 menunjukkan implementasi permintaan monster. Bagian ini bertanggung jawab untuk menambahkan pertanyaan konfirmasi kepada pemain apakah monster boleh memakan sejumlah ceri ke dalam sebuah kontainer pertanyaan. Fungsi `addConfirmationQuestionToContainer` menerima satu parameter, yaitu `questionContainer` untuk menampung pertanyaan.

```
this.addConfirmationQuestionToContainer =
function(questionContainer) {
  let depth = questionContainer.depth;

  const SPACING = 60;
  var container = [];

  let speechBubble = this.createSpeechBubble("Bolehkah aku
minta\n beberapa ceri?");

  questionContainer.setData("container", container);

  let buttonX = questionContainer.x +
questionContainer.getData('width')/2 - 40;
  let buttonY = questionContainer.y +
questionContainer.getData('height')/2 - 25;
  let yesConfirm = new RectButton(this, buttonX-SPACING,
buttonY, 0xD9DF20, 0x174F4D, 0x174F4D, 0xD9DF20, 80, 50, true,
"Ya", 21);
  yesConfirm.setDepth(depth);
  yesConfirm.deactivateOnGamePause = true;
  yesConfirm.setData("pointerup", () => {
    this.canInteractWithGame = false;
    this.SOUNDS['confirmClick'].play();
    this.hideContainer(0, 300, () => {
      speechBubble.forEach(element => element.destroy());
```

```

        this.SOUNDS['eat'].play();

this.questionBoxContainer.getData("container").forEach(gameObject => gameObject.destroy());
        this.previousMonster = this.monster.getData('key');
        this.animateTween(this.monster, 800, 600, '+=0',
this.YPOSITION.BOTTOM + 300);
        this.eatenCherries = Math.floor(Math.random() *
this.cherries.length) + 1;
        this.cherriesBeforeEaten = this.cherries.length;
        this.cherries.forEach(cherry => cherry.destroy());
        this.cherries =
this.getCherries(this.cherriesBeforeEaten - this.eatenCherries);
        this.setCherriesInteractable(this.cherries, false);
        this.time.addEvent({
            delay: 900,
            callback: () => {

this.addSubstractionQuestionToContainer(this.questionBoxContainer, "Berapa banyak cери yang dimakan oleh monster?", 15,
this.eatenCherries);
                this.showContainerSideway(200, 300, () => {
                    this.canInteractWithGame = true;

this.setCherriesInteractable(this.cherries, true);
                });
            }
        });
    });
    container.push(yesConfirm);

    let noConfirm = new RectButton(this, buttonX+SPACING,
buttonY, 0xD9DF20, 0x174F4D, 0x174F4D, 0xD9DF20, 80, 50, true,
"Tidak", 21);
    noConfirm.setDepth(depth);
    noConfirm.deactivateOnGamePause = true;
    noConfirm.setData("pointerup", () => {
        this.canInteractWithGame = false;
        this.SOUNDS['confirmClick'].play();
        this.previousMonster = this.monster.getData('key');
    });

```



```

        speechBubble.forEach(element => element.destroy());
        this.SOUNDS['monsterBellow'].play();
        this.hideContainer(0, 300, null);
        this.animateTween(this.monster, 0, 400, '+=40', '+=100',
    () => {
            this.animateTween(this.monster, 500, 800, '+=400',
'+=500', () => {
                this.monster.destroy();
                this.monster = this.getRandomMonster();
                this.monster.x = this.XPOSITION.CENTER;
                this.monster.y = this.YPOSITION.BOTTOM + 300;
                this.animateShowMonsterAtDoor(() =>
this.animateHideMonsterAtDoor(() => {
                    this.animateShowMonsterAtTable(() => {
                        this.showContainerSideway(400, 300, ()
=> this.canInteractWithGame = true);
                        speechBubble =
this.createSpeechBubble("Bolehkah aku minta\n beberapa ceri?");
                            }, 100)
                            }, 1500));
                    });
                });
            });
        container.push(noConfirm);

        questionContainer.setData("container", container);

        this.showContainerSideway(400, 300, null);

        return questionContainer;
    };

```

Kode Sumber 5.2. Implementasi Permintaan Monster

5.2.1.3. Implementasi Pertanyaan Pengurangan

Kode Sumber 5.3 menunjukkan implementasi pertanyaan pengurangan. Bagian ini bertanggung jawab untuk menambahkan pertanyaan pengurangan ceri yang

dimakan oleh monster ke dalam sebuah kontainer pertanyaan. Fungsi `addSubstractionQuestionToContainer` menerima beberapa parameter, yaitu `questionContainer` untuk menampung pertanyaan, `textString` sebagai teks pertanyaan, `padding` untuk menambahkan ruang sekitar pertanyaan, dan `correctAnswer` untuk jawaban benar dari pertanyaan.

Bagian ini memiliki beberapa fungsi didalamnya. Fungsi `drawRetryButton` digunakan untuk menggambar tombol "Tonton Lagi" yang akan ditampilkan di kontainer pertanyaan. Tombol ini memiliki fungsi untuk memberi pemain opsi untuk melihat kembali aksi monster memakan ceri. Fungsi `hideMonster` mengatur animasi mengatur animasi untuk menyembunyikan monster setelah memberikan petunjuk atau hint kepada pemain. Fungsi `showMonster` memunculkan monster untuk memberikan hint.

```
this.addSubstractionQuestionToContainer =
function(questionContainer, textString = "Text", padding = 5,
correctAnswer) {
  this.drawRetryButton = function() {
    const X = questionContainer.getData('right');
    const Y = questionContainer.getData('centerY');
    let depth = questionContainer.depth;
    const HEIGHT = 50;
    const WIDTH = 210;
    const MARGINX = 25;
    let button = new RectButton(this, X - WIDTH - MARGINX, Y
- HEIGHT/2, 0xD9DF20, 0x0A4550, 0x0A4550, 0xD9DF20, WIDTH,
HEIGHT, true, "Tonton Lagi", 21);
    button.onDrawContent = (backgroundColor, textColor) => {
      let restart = this.add.image(button.rect.centerX -
```

```

75, button.rect.centerY, 'restart');
    let text = this.addText(button.rect.centerX + 25,
button.rect.centerY, button.text, button.textSize, textColor);
    restart.setTint(textColor);
    restart.setScale(0.15);
    button.add(restart);
    button.add(text);
}
button.refresh();
button.setDepth(depth);
button.deactivateOnGamePause = true;
button.onPointerUp = () => {
    this.canInteractWithGame = false;

    speechBubble.forEach(element => element.destroy());
    active = false;

    this.setCherriesInteractable(this.cherries, false);
    button.destroy();
    this.SOUNDS['confirmClick'].play();

    this.drawFadeOut(0, 600, null, 0x000000);
    this.time.addEvent({
        delay: 600,
        callback: () => {
            this.drawFadeIn(0, 700, null, 0x000000);
            this.monster.setRotation(0);
            this.monster.x = this.XPOSITION.CENTER;
            this.monster.y = this.YPOSITION.BOTTOM -
450;

            this.cherries.forEach(cherry =>
cherry.destroy());
            this.cherries =
this.getCherries(this.cherriesBeforeEaten);
            this.setCherriesInteractable(this.cherries,
false);

            this.time.addEvent({
                delay: 2000,
                callback: ()=>{

```

```

        this.SOUNDS['eat'].play();
        this.animateTween(this.monster, 800,
600, '+=0', this.YPOSITION.BOTTOM + 150, () => {
            this.hideContainer(0, 300, () =>
{
                this.showContainerSideway(0,
300, () => {
                    this.canInteractWithGame
= true;
this.setCherriesInteractable(this.cherries, true);
                });
            });
        });
        this.cherries.forEach(cherry =>
cherry.destroy());
        this.cherries =
this.getCherries(this.cherriesBeforeEaten - this.eatenCherries);
this.setCherriesInteractable(this.cherries, false);
    }
    });
}
});
};

    return button;
};
    this.hideMonster = function(onComplete) {
        speechBubble.forEach(element => element.destroy());
        this.animateTween(this.monster, 0, 500, '+=0',
this.YPOSITION.BOTTOM + 100, onComplete);
        active = false;
    };
    this.showMonster = function(onComplete) {
        this.animateTween(this.monster, 0, 500, '+=0',
this.YPOSITION.BOTTOM - 250, onComplete);
    };
    this.next = function(isCorrect) {
        this.addSubstractionCountContainer(isCorrect);
        this.hideMonster();
    };

```

```

        speechBubble.forEach(element => element.destroy());
        if (button != null) button.destroy();
    };
    let button = this.drawRetryButton();
    let speechBubble = [];
    let active = false;
    let evt;
    return this.addGenericQuestionToContainer(questionContainer,
textString, padding, correctAnswer, () => {
        this.next(true);
    }, () => {
        this.next(false);
    }, () => {
        active = true;
        this.showMonster(() => {
            if (active == false) return;

            speechBubble =
this.createHintSpeechBubble(this.cherriesBeforeEaten, -250,
true);

            evt = this.time.addEvent({
                delay: 4000,
                callback: ()=>{
                    this.hideMonster();
                }
            })
        });
    }, () => {
        this.hideMonster();
        if (evt != null) evt.destroy();
    }, [button]);
};

```

Kode Sumber 5.3. Implementasi Pertanyaan Pengurangan

5.2.2. Menghitung Penumpang

5.2.2.1. Implementasi Pertanyaan Menghitung

Kode Sumber 5.4 menunjukkan implementasi pertanyaan menghitung. Bagian ini merupakan fungsi Bernama `addPenumpang` yang bertanggung jawab dalam

jumlah penumpang yang dimunculkan pada layar. Penumpang yang dikumpulkan dalam array akan dipanggil berdasarkan jumlah yang sesuai dengan jawaban benar pada bagian itu.

```
function addPenumpang(){
    let posX = 0;
    let posY = 0;
    shuffleArray(myScene.arrPen);

    myScene.arrPenumpang = [];
    if(myScene.jawabanBenar == 3){
        posX = myScene.bebe_chara.x;
        posY = myScene.bebe_chara.y;
        soal = parseInt(permainan1) + parseInt(permainan2);
    } else {
        posX = X_POSITION.CENTER + 650;
        posY = Y_POSITION.CENTER + 150;
        soal = Phaser.Math.Between(1, 5);
    }

    question1.setText("Berapa siswa yang naik di Halte" +
    indxHalte[myScene.jawabanBenar] + "?");
    textHalte.setText("Halte"
    +indxHalte[myScene.jawabanBenar]);

    for(let j = 0; j < soal; j++){
        let penumpangX = X_POSITION.CENTER + 370 - (j * 90);
        myScene.arrPosX.push(penumpangX);

        let pen = j % myScene.arrPen.length;

        let penumpang = myScene.add.image(posX, posY,
        "penumpang" + pen).setDepth(5.5).setScale(1.1);
        myScene.arrPenumpang.push(penumpang);
        myScene.arrLastPen.push(penumpang);
    }
}
```

Kode Sumber 5.4. Implementasi Pertanyaan Menghitung

5.2.2.2. Implementasi Tampilan Akhir Permainan

Kode Sumber 5.5 menunjukkan implementasi tampilan akhir permainan. Bagian ini bertanggung jawab untuk menampilkan skor akhir berdasarkan penilaian pada seluruh pertanyaan sebelumnya.

```
function gameSelesai() {  
  
    var hasil = 0;  
    let result = [];  
    let posY = [170, 190, 170];  
    let jarak = 90;  
  
    let nilai = myScene.resPoint;  
  
    myScene.tweens.add({  
        targets: myScene.driveaway_sound,  
        volume: 0,  
        duration: 700,  
    });  
  
    switch (true) {  
        case nilai >= 0 && nilai <= 2:  
            result = ["kosong", "kosong", "kosong"];  
            hasil = 0;  
            break;  
        case nilai >= 3 && nilai <= 4:  
  
            result = ["kosong", "penuh", "kosong"];  
            hasil = 1;  
            break;  
        case nilai >= 5 && nilai <= 6:  
  
            result = ["penuh", "penuh", "kosong"];  
            hasil = 2;  
            break;  
        case nilai >= 7 && nilai <= 8:
```

```

        result = ["penuh", "penuh", "kosong"];
        hasil = 3;
        break;
    case nilai === 9:
        result = ["penuh", "penuh", "penuh"];
        hasil = 4;
        break;
    }

    let win = myScene.sound.add('sfxWin')

    if (hasil < 1) {
        win.play();
        city_music.setVolume(0.3);
        win.on('complete', function () {
            city_music.setVolume(1);
        });
    } else if (hasil < 2) {
        win.play();
        city_music.setVolume(0.3);
        win.on('complete', function () {
            city_music.setVolume(1);
        });
    } else if (hasil < 3) {
        win.play();
        city_music.setVolume(0.3);
        win.on('complete', function () {
            city_music.setVolume(1);
        });
    } else if (hasil <= 5) {
        win.play();
        city_music.setVolume(0.3);
        win.on('complete', function () {
            city_music.setVolume(1);
        });
    }

    let bebeRes = myScene.add.image(X_POSITION.CENTER,
    Y_POSITION.CENTER, 'bebeRes' +

```



```

hasil).setDepth(28).setScale(0.75);
    bebeRes.setScale(1.1);
    myScene.snd_gameover = myScene.sound.add('sfxRes' +
hasil).play();

    for (let i = 0; i <= 2; i++) {
        var resBintang = myScene.add.image(bebeRes.x + 50 +
jarak * (i - 1.5), bebeRes.y - posY[i], 'bintang_' +
result[i]).setDepth(28.5);
        arrResult.push(resBintang);
    }

    myScene.transparent.setAlpha(1);
    btnReplay.y = bebeRes.y + 180;
    btnReplay.setAlpha(1).setInteractive({ cursor:
'url(cursor-hand.png), pointer' });
    myScene.rayy.forEach(ray => {
        ray.setAlpha(1)
        arrResult.push(ray);
    });
    arrResult.push(bebeRes);
    arrResult.push(btnReplay);
    arrResult.push(myScene.transparent);

    setTimeout(function () {
        resultStars();
    }, 500);
}

```

Kode Sumber 5.5. Implementasi Tampilan Akhir Permainan

[Halaman ini sengaja dikosongkan]

BAB VI

PENGUJIAN DAN EVALUASI

Bab ini menjelaskan tahap uji coba terhadap aplikasi game yang telah dikembangkan. Pengujian dilakukan untuk memastikan fungsionalitas dan kesesuaian hasil implementasi game dengan analisis dan perancangan game.

6.1. Tujuan Pengujian

Pengujian dilakukan terhadap aplikasi game yang telah dikembangkan guna menguji kesesuaian fungsionalitas dan kebutuhan game.

6.2. Skenario Pengujian

Peserta kerja praktik diwajibkan menulis laporan setiap hari jam 11 - 12 siang dan jam 4 - 5 sore berisi pekerjaan yang dikerjakan pada hari itu. Kemudian, trainer akan memberikan tanggapan dan saran untuk memperbaiki dan mengembangkan game lebih lanjut.

[Halaman ini sengaja dikosongkan]

BAB VII

KESIMPULAN DAN SARAN

7.1. Kesimpulan

Kesimpulan yang didapat setelah melakukan pengembangan aplikasi game di gamelab adalah sebagai berikut :

- a. Aplikasi game yang telah dikembangan sudah sesuai dengan ketentuan dari trainer
- b. Proses pengembangan dimulai dengan analisis kebutuhan, diikuti oleh pengelolaan aset dan alur permainan. Implementasi dilakukan menggunakan bahasa pemrograman JavaScript.

7.2. Saran

Saran untuk pengembangan aplikasi game di gamelab adalah sebagai berikut :

- a. Optimalisasi kode dan penggunaan aset yang lebih efisien dapat meningkatkan performa game maupun keterbacaan kode.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- [1] J. Enterprise, HTML 5 MANUAL BOOK, Jakarta: PT Elex Media Komputindo, 2014.
- [2] D. Bartholomew, <<MariaDB vs. MySQL,>> Dostopano, 2014,2012.
- [3] IDCloudHost. 2020. Mengenal Apa itu Framework CodeIgniter | IDCloudHost. [ONLINE] Available at: <https://idcloudhost.com/panduan/mengenal-apa-itu-framework-codeigniter/>. [Diakses September 2020].
- [4] Niagahoster Blog. 2020. Apa Itu Nginx dan Cara Kerjanya - Niagahoster Blog. [ONLINE] Available at: <https://www.niagahoster.co.id/blog/nginx-adalah/>. [Diakses September 2020].
- [5] HAProxy - The Reliable, High Performance TCP/HTTP Load Balancer. 2020. HAProxy - The Reliable, High Performance TCP/HTTP Load Balancer. [ONLINE] Available at: <http://www.haproxy.org/>. [Accessed 22 December 2020].
- [6] Phaser. Getting Started with Phaser 3. [ONLINE] Available at: <https://phaser.io/tutorials/getting-started-phaser3>
- [7] Mozilla Developer Network. JavaScript. [ONLINE] Available at: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- [8] Mozilla Developer Network. WebGL API. [ONLINE] Available at: https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API
- [9] TechTarget. HTML5 Definition. [ONLINE] Available at: <https://www.techtarget.com/whatis/definition/HTML5>

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS I

Nama : Beryl
Tempat, Tanggal Lahir :
Jenis Kelamin : Laki - laki
Telepon : +6289603278658
Email :

AKADEMIS

Kuliah : Departemen Teknik Informatika –
FTEIC , ITS
Angkatan : 2020
Semester : 8 (Delapan)

BIODATA PENULIS II

Nama : Brian Akbar Wicaksana
Tempat, Tanggal Lahir : Surabaya, 28 Juli 2001
Jenis Kelamin : Laki - laki
Telepon : +6282132358820
Email : brianakbar@yahoo.com.sg

AKADEMIS

Kuliah : Departemen Teknik Informatika –
FTEIC , ITS
Angkatan : 2020
Semester : 8 (Delapan)