

LAPORAN AKHIR

GAME EDUKASI “QUIZ” BERTEMA SEJARAH INDONESIA BERBASIS PYTHON

Oleh :

ABDUL MALIK NIM. 1931730002

DOSEN PEMBIMBING:

RINANZA ZULMY ALHAMRI, S.Kom., M.Kom.

NIDN. 0710049004



PROGRAM STUDI D-III MANAJEMEN INFORMATIKA

PSDKU POLINEMA DI KOTA KEDIRI

POLITEKNIK NEGERI MALANG

KEDIRI

2022

LAPORAN AKHIR

GAME EDUKASI “QUIZ” BERTEMA SEJARAH INDONESIA BERBASIS PYTHON

Laporan Akhir ini disusun untuk memenuhi syarat memperoleh gelar A.Md.

Program Studi D-III Manajemen Informatika

Politeknik Negeri Malang

Oleh :

ABDUL MALIK NIM. 1931730002



PROGRAM STUDI D-III MANAJEMEN INFORMATIKA

PSDKU POLINEMA DI KOTA KEDIRI

POLITEKNIK NEGERI MALANG

KEDIRI

2022

HALAMAN PENGESAHAN

GAME EDUKASI “QUIZ” BERTEMA SEJARAH INDONESIA BERBASIS PYTHON



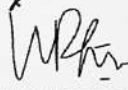
Disusun oleh :

ABDUL MALIK NIM. 1931730002

Laporan Akhir ini telah diuji pada tanggal 19 Juli 2022

Disetujui oleh:

Dewan Penguji:

- | | | |
|--|--|--------------|
| 1. Rinanza Zulmy Alhamri, S.Kom., M.Kom.
NIDN. 0710049004 | 
..... | (Pembimbing) |
| 2. Benni Agung Nugroho, S.Kom., M.Cs.
NIDN. 0724068102 | 
..... | (Penguji 1) |
| 3. Irfan Sandra Asti, S.ST., M.T
NIDN. 0027069502 | 
..... | (Penguji 2) |

Mengetahui,

Koordinator


PSDKU Polinema di Kota Kediri,



Menyetujui,

Ketua Program Studi

DIII-Manajemen Informatika


Fadelis Sukya, S.Kom., M.Cs
NIDN. 0730038201

PERNYATAAN

Dengan ini saya menyatakan bahwa Laporan Akhir ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar Ahli Madya / Sarjana Terapan di Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Kediri, 19 Juli 2022



Abdul Malik

ABSTRAK

Malik, Abdul. “*Game* Edukasi “*QUIZ*” Bertema Sejarah Indonesia Berbasis *Python*”. Pembimbing: **Rinanza Zulmy Alhamri, S.Kom., M.Kom.**

Laporan Akhir, Program Studi D-III Manajemen Informatika, Jurusan Teknologi Informasi, PSDKU Politeknik Negeri Malang di Kota Kediri, 2022.

Sejarah merupakan suatu kejadian-kejadian, peristiwa-peristiwa yang berhubungan dengan manusia, yakni menyangkut perubahan yang nyata di dalam kehidupan manusia. Selama ini untuk mengetahui tentang sejarah diperlukannya membuka buku yang dirasakan bosan karena perlu waktu yang panjang untuk menghabiskan waktu membaca buku tebal sejarah. Perihal tersebut mengakibatkan minimnya kehausan tentang sejarah pada generasi muda saat ini.

Oleh karena itu perlu adanya cara memahami sejarah dengan cara yang *fun* (menyenangkan) dan juga *rewarding* (bermanfaat) maka di buatlah *game* edukasi berupa *quiz* yang mengambil tema sejarah. *Game* ini dirancang menggunakan *Python*, *Sqllite*, dan *Pygame*. *Game* ini juga telah diuji coba oleh beberapa pemain (*user*) dan hasilnya setiap pemain (*user*) memainkan *game* dengan tingkat kesusahan yang dipilih oleh pemain (*user*) dan menerima nilai (*score*) setelah menjawab, untuk *admin* dapat melakukan kelola *user* dan kelola pertanyaan (*quiz*) melalui menu admin.

Kata Kunci : Game Quiz, Sejarah

ABSTRACT

Malik, Abdul. *“Educational Game “QUIZ” History of Indonesia Theme Based in Python”*. **Counseling Lecturer: Rinanza Zulmy Alhamri, S.Kom., M.Kom.**

Final Report, D-III Informatics Management Study Programme, Department of Information Technology Jurusan Teknologi Informasi, State Polytechnic of Malang PSDKU Kediri, 2022.

History is an events that related to man, real changes in the life of mankind. To this day to understand about history thick history book is needed to be read, which is boring to read it because it takes a long time to spend reading history books. Because of it, interest in history is on minimum in young generation nowadays.

Therefore it is necessary to understand history in a fun and rewarding way, so we create an educational quiz game that use history as its theme. This game is designed using Python, Sqlite, and Pygame. This game also have been tested by gamer (user) and the result is gamer (user) playing the game with their selected difficulty and gamer (user) will receive score after answering it, as for admin they can manage user and manage question (quiz) from admin menu.

Keywords : Game Quiz, History

KATA PENGANTAR

Puji Syukur kami panjatkan kehadiran Allah SWT atas segala rahmat dan hidayah-Nya penulis dapat menyelesaikan laporan akhir dengan judul “*Game Edukasi “Quiz” Bertema Sejarah Indonesia Berbasis Python*”. Laporan akhir ini penulis susun sebagai persyaratan untuk menyelesaikan studi program Diploma III Program Studi Manajemen Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang.

Kami menyadari tanpa adanya dukungan dan kerja sama dari berbagai pihak, kegiatan laporan akhir ini tidak akan dapat berjalan baik. Untuk itu, kami ingin menyampaikan rasa terima kasih kepada:

1. Bapak Dandung Novianto, S.T,M.T., selaku Koordinator PSDKU Polinema Kediri.
2. Bapak Fadelis Sukya, S.Kom.,M.Cs., selaku ketua program studi Manajemen Informatika.
3. Bapak Rinanza Zulmy Alhamri, S.Kom.,M.Kom selaku dosen pembimbing tugas akhir.
4. Seluruh dosen program studi Manajemen Informatika PSDKU Polinema Kediri.
5. Kedua orang tua, keluarga, saudara dan teman-teman yang telah memberikan semangat dan doa.
6. Semua teman-teman mahasiswa program studi Manajemen Informatika.
7. Dan seluruh pihak yang telah membantu dan mendukung lancarnya pembuatan Laporan Akhir dari awal hingga akhir yang tidak dapat penulis sebutkan satu persatu.

Penulis menyadari bahwa dalam penyusunan laporan akhir ini, masih banyak terdapat kekurangan dan kelemahan yang dimiliki penulis baik itu sistematika penulisan maupun penggunaan bahasa. Untuk itu penulis mengharapkan saran dan kritik dari berbagai pihak yang bersifat membangun demi penyempurnaan laporan ini. Semoga laporan ini berguna bagi pembaca secara umum dan penulisan secara khusus. Akhir kata, penulis ucapkan banyak terima kasih.

Kediri, 19 Juli 2022

Penulis

DAFTAR ISI

HALAMAN SAMPUL	i
HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN.....	ii
PERNYATAAN.....	iii
ABSTRAK	iv
<i>ABSTRACT</i>	v
KATA PENGANTAR	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR	xi
BAB I PENDAHULUAN	1
1.1. Latar Belakang Masalah	1
1.2. Batasan Masalah	3
1.3. Rumusan Masalah	3
1.4. Tujuan Penelitian.....	3
1.5. Manfaat.....	4
1.6. Sistematika Penulisan.....	4
BAB II LANDASAN TEORI	6
2.1. Penelitian yang Relevan	6
2.2. Teori Penelitian	8
2.2.1. Game Quiz	8
2.2.2. Sejarah Kemerdekaan Indonesia (20 Mei 1908 – 17 Agustus 1945)	9
2.2.3. SQLite	10
2.2.4. UML (Unified Modeling Language).....	11
2.2.5. Python	14
2.2.6. Pygame.....	15
2.2.7. Pillow (<i>Python Image Library</i>)	16
BAB III ANALISIS DAN PERANCANGAN	17
3.1. Analisis Permasalahan.....	17
3.2. Analisis Pemecahan Masalah	18
3.2.1. Implementasi Algoritma pada Game <i>Quiz</i>	18
3.2.2. Metode Pengambilan Data (user requirement)	20
3.2.3. Metode Pengembangan Sistem	21

3.2.4.	Analisis Kebutuhan Sistem (fungsional dan non fungsional system)	23
3.3.	Perancangan Sistem.....	25
3.3.1.	<i>Use Case Diagram</i>	25
3.3.2.	Skenario <i>Use Case Diagram</i>	27
3.3.3.	Activity Diagram.....	33
3.3.4.	Diagram Sequence	38
3.3.5.	Diagram Class	43
3.3.6.	Rancangan <i>Database</i>	44
3.3.7.	Arsitektur Sistem.....	46
3.3.8.	Rancangan Antarmuka	47
3.4.	Rancangan Pengujian	50
BAB IV IMPLEMENTASI		52
4.1.	Implementasi Proses.....	52
4.1.1.	Tampilan <i>Login</i> Pemain (<i>User</i>).....	52
4.1.2.	Tampilan Registrasi (<i>User</i>)	53
4.1.3.	Tampilan <i>Play Game</i>	54
4.1.4.	Tampilan Melihat Nilai	57
4.1.5.	Tampilan Kelola Pertanyaan (<i>Admin</i>).....	59
4.2.	Implementasi Data.....	64
4.2.1.	Tabel User	64
4.2.2.	Tabel Quiz	64
4.2.3.	Tabel Score.....	65
4.2.4.	Tabel Admin.....	66
BAB V PENGUJIAN DAN PEMBAHASAN		67
5.1.	Pengujian Proses.....	67
5.1.1	Pengujian Registrasi.....	67
5.1.2	Pengujian Login	69
5.1.3	Pengujian Play Game	70
5.1.4	Pengujian Menampilkan Nilai.....	72
5.1.5	Pengujian Kelola Pertanyaan	73
5.2.	Analisis Hasil Pengujian	75
BAB VI KESIMPULAN		77
6.1.	Kesimpulan.....	77
6.2.	Saran	77

DAFTAR PUSTAKA	78
LAMPIRAN.....	79

DAFTAR TABEL

Tabel 2. 1 Perbandingan Studi Literatur	7
Tabel 3. 1 Definisi Aktor.....	27
Tabel 3. 2 Skenario <i>Use Case Diagram</i> Registrasi.....	28
Tabel 3. 3 Skenario <i>Use Case Diagram Login</i>	29
Tabel 3. 4 Skenario <i>Use Case</i> Menjalankan Game.....	30
Tabel 3. 5 Skenario <i>use case</i> melihat nilai	31
Tabel 3. 6 Skenario <i>use case</i> kelola pertanyaan	32
Tabel 3. 7 Tabel user	44
Tabel 3. 8 Tabel quiz.....	45
Tabel 3. 9 Tabel score	45
Tabel 3. 10 Tabel admin.....	45
Tabel 3. 11 Keterangan Arsitektur Sistem	47
Tabel 3. 12 Tabel Rancangan Pengujian.....	51
Tabel 5. 1 Tabel Pengujian.....	76

DAFTAR GAMBAR

Gambar 2. 1 Rapat PPKI.....	10
Gambar 2. 2 Logo SQLite.....	11
Gambar 2. 3 Penjelasan simbol-simbol pada <i>use case diagram</i>	12
Gambar 2. 4 Penjelasan simbol-simbol pada <i>activity diagram</i>	13
Gambar 2. 5 Penjelasan simbol-simbol pada <i>sequence diagram</i>	13
Gambar 2. 6 Penjelasan simbol-simbol pada <i>class diagram</i>	14
Gambar 2. 7 Contoh perintah <i>print</i> pada python.....	15
Gambar 2. 8 Perintah untuk install pygame pada python	15
Gambar 2. 9 Implementasi pygame untuk membuat window tab.....	16
Gambar 3. 1 Proses bisnis belajar sejarah Indonesia.....	17
Gambar 3. 2 Flowchart algoritma pada <i>game quiz</i>	19
Gambar 3. 3 <i>Chart</i> keseluruhan dari penerapan algoritma pada <i>game quiz</i>	19
Gambar 3. 4 Metode Waterfall.....	21
Gambar 3. 5 <i>Use Case Diagram</i>	26
Gambar 3. 6 <i>Activity Diagram</i> Registrasi	33
Gambar 3. 7 <i>Activity Diagram</i> Login.....	34
Gambar 3. 8 <i>Activity Diagram</i> Play Game	35
Gambar 3. 9 <i>Activity Diagram</i> Melihat Nilai.....	36
Gambar 3. 10 <i>Activity Diagram</i> Kelola Pertanyaan.....	37
Gambar 3. 11 <i>Sequence Diagram</i> Registrasi	38
Gambar 3. 12 <i>Sequence Diagram</i> Login	39
Gambar 3. 13 <i>Sequence Diagram</i> Play Game	40
Gambar 3. 14 <i>Sequence Diagram</i> Melihat Nilai.....	41
Gambar 3. 15 <i>Sequence Diagram</i> Kelola Pertanyaan.....	42
Gambar 3. 16 <i>Diagram Class</i>	43
Gambar 3. 17 Relasi Antar Tabel.....	46
Gambar 3. 18 Arsitektur Sistem.....	47
Gambar 3. 19 Rancangan Antarmuka <i>Login</i>	48
Gambar 3. 20 Rancangan Antarmuka Registrasi	48
Gambar 3. 21 Rancangan Antarmuka Play Game	49
Gambar 3. 22 Rancangan Antarmuka Game.....	49
Gambar 3. 23 Rancangan Antarmuka Akhir Game	50
Gambar 4. 1 Tampilan <i>Login & Registrasi</i>	52
Gambar 4. 2 Tampilan Menu Registrasi	53
Gambar 4. 3 Tampilan Menu Memilih Tingkat Kesulitan.....	55
Gambar 4. 4 Tampilan Menu <i>Game</i>	56
Gambar 4. 5 Tampilan Melihat Nilai	57
Gambar 4. 6 Tampilan <i>Scoreboard</i>	58
Gambar 4. 7 Tampilan Menu Admin Kelola <i>Quiz</i>	59
Gambar 4. 8 Tampilan Tabel Kelola <i>Quiz</i>	60
Gambar 4. 9 Tampilan Edit <i>Quiz</i>	61
Gambar 4. 10 Tampilan Edit Foto <i>Quiz</i>	62
Gambar 4. 11 Implementasi Tabel <i>User</i>	64
Gambar 4. 12 Implementasi Tabel <i>Quiz</i>	65

Gambar 4. 13 Implementasi Tabel <i>Score</i>	65
Gambar 4. 14 Implementasi Tabel Admin	66
Gambar 5. 1 Pengujian Registrasi	68
Gambar 5. 2 Hasil Pengujian Registrasi	68
Gambar 5. 3 Hasil Pengujian Registrasi Gagal	68
Gambar 5. 4 Pengujian <i>Login</i>	69
Gambar 5. 5 Hasil Pop-Up Login	69
Gambar 5. 6 Hasil <i>Login</i> Masuk Ke <i>Menu</i> Utama	70
Gambar 5. 7 Hasil <i>Pop-Up Login</i> Gagal	70
Gambar 5. 8 Pengujian <i>Play Game</i> Memilih Tingkat Kesulitan	71
Gambar 5. 9 Hasil Pengujian <i>Play Game</i> Memainkan <i>Quiz</i>	71
Gambar 5. 10 Hasil Pengujian <i>Play Game Total Score</i>	72
Gambar 5. 11 Pengujian Menampilkan Nilai	72
Gambar 5. 12 Hasil Pengujian Menampilkan Nilai	73
Gambar 5. 13 Pengujian Kelola Pertanyaan Memilih <i>Log In As Admin</i>	73
Gambar 5. 14 Pengujian Kelola Pertanyaan <i>Login</i>	74
Gambar 5. 15 Pengujian Kelola Pertanyaan <i>Admin Menu</i>	74
Gambar 5. 16 Hasil Pengujian Kelola Pertanyaan bagian <i>Quiz</i>	75
Gambar 5. 17 Hasil Pengujian Kelola Pertanyaan Gambar	75

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Sejarah merupakan suatu kejadian-kejadian, peristiwa-peristiwa yang berhubungan dengan manusia, yakni menyangkut perubahan yang nyata di dalam kehidupan manusia (Ali, 2005). Sejarah adalah kejadian-kejadian lampau yang terjadi, sejarah sendiri perlu untuk dipahami untuk memahami tujuan-tujuan perilaku manusia di zaman sekarang dan masa depan yang akan datang. Terutama sejarah Indonesia yang dimulai dari kerajaan-kerajaan, masa penjajahan, hingga masa menuju kemerdekaan. Di sejarah Indonesia sendiri terdapat banyak cerita hingga tokoh-tokoh yang menarik untuk dipelajari.

Karena teramat banyaknya sejarah yang terjadi di Indonesia membuat untuk mempelajarinya sulit apabila menggunakan metode belajar dengan menghafal materi. Metode belajar yang digunakan selama ini yaitu membaca dan menghafal kemudian pelajar diuji dengan menghadapi soal yang diberikan oleh pembimbing pelajaran, sewaktu pelajar menghadapi soal banyak yang merasa tegang dan panik karena tekanan (*pressure*) yang dihadapi dimana harus mengerjakan soal dengan waktu yang ditentukan sehingga membuat akhir nilai dari pelajar kacau karena tekanan (*pressure*) yang dihadapi. Dengan demikian adanya keinginan untuk menggunakan metode belajar selain menghafal dan mengerjakan soal, contohnya menggunakan *game* (permainan) edukasi bertema sejarah Indonesia. *Game* dalam fungsi untuk pembelajaran dapat digunakan untuk menjelaskan konsep dan memperkuat pembelajaran dan tidak menekan (*pressure*) kepada pelajar. *Game* “*Quiz*” adalah *game* kuis atau tebak menebak dimana di dalam *game* ini terbentuk dengan permainan atau pikiran proses pemain (sebagai individu) berusaha untuk menjawab pertanyaan dengan benar. *Game Quiz* tersebut juga memiliki penilaian singkat yang digunakan dalam bidang pendidikan dan mirip dengan mengukur pertumbuhan dalam pengetahuan, kemampuan atau keterampilan.

Pembuatan *game* kuis bertema sejarah Indonesia ini sudah pernah dilakukan dengan judul “Aplikasi Trivia *Game* Mata Pelajaran Sejarah Berbasis Android” (Ningrum & Maslan, 2020). Di dalam pembuatan *game* kuis bertema sejarah

Indonesia ini Ningrum dan Maslan menggunakan aplikasi *game engine Construct 2* untuk melakukan pembuatan aplikasi trivia game mata pelajaran sejarah berbasis android. Kemudian di tahun 2017 dilakukan penelitian berjudul “Pembuatan Aplikasi Game Kuis “Pontianak Punye” Berbasis Android” (Rafiqin & Saputra, 2017). Di dalam pembuatan game ini Rafiqin dan Saputra juga menggunakan *game engine Construct 2*. Dari kedua penelitian diatas, peneliti menggunakan *engine Construct 2* sebagai aplikasi untuk membuat game kuis dan masing-masing memiliki konsep cara bermain yang hampir sama dimana pengguna memilih jawaban yang benar untuk mendapatkan skor (nilai) tinggi. Pada tahun 2018 dilakukan penelitian dengan judul “Pembuatan Game Kuis Siapa Pintar” (Barros et al., 2018). Pada penelitian ini Barros, Marisa, dan Wijaya membuat sebuah game kuis menggunakan algoritma *Fisher-Yates* untuk melakukan pengacakan (*random*) soal sehingga soal yang muncul disetiap pengguna kemungkinan berbeda.

Berdasarkan penelitian terdahulu dapat diketahui bahwa pembuatan game kuis edukasi dapat dijadikan metode belajar baru dengan memberikan informasi mengapa jawaban dari soal adalah demikian sehingga pemain dapat sekaligus mempelajari soal yang diberikan. Oleh sebab itu pembuatan game *Quiz* pada penelitian ini akan menggunakan python menggunakan module Pygame yang dimana di module ini kita dapat mengembangkan game di Python.

Game Quiz ini akan dikembangkan menggunakan bahasa pemrograman python karena python sendiri dapat berjalan secara *multiplatform* dan juga python sendiri dapat dijalankan dengan mudah, kita cukup menambahkan sebuah modul pada python seperti pygame yang dimana modul tersebut akan membantu pengembang untuk mempermudah mengembangkan game dengan SDL Library yang tersedia pada modul pygame seperti halnya adalah memberi sebuah antarmuka sehingga pengguna dengan mudah dapat mengendalikan game dengan menggunakan antarmuka yang tampil pada *display* pengguna. Kemudian pengguna akan dapat memilih tingkat kesulitan dalam menghadapi soal dimana nanti soal yang didapat juga berbeda di setiap tingkat kesulitannya, maka dari itu digunakannya metode algoritma forward chaining. Maka pada proposal ini diusulkan penelitian tentang Game Edukasi “Quiz” Bertema Sejarah Indonesia Berbasis Python.

1.2. Batasan Masalah

Terdapat batasan – batasan masalah dalam penelitian ini antara lain adalah sebagai berikut:

1. Game Quiz yang akan dibuat menggunakan studi kasus sejarah Indonesia hingga kemerdekaan.
2. Sistem kelola yang akan diterapkan adalah mempersunting pada bagian pertanyaan dan foto pertanyaan dari *quiz*.
3. Soal pada kuis game Quiz adalah sejarah kemerdekaan Indonesia hingga kemerdekaan Indonesia yang pada game ini gunakan adalah kronologi yang terjadi dari tahun 20 Mei 1908 hingga terjadinya proklamasi pada 17 Agustus 1945.

1.3. Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana membangun game sejarah kemerdekaan Indonesia berbasis python ?
2. Bagaimana penerepan pemrograman python dan *library* pada python untuk game quiz ?

1.4. Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Membangun game sejarah kemerdekaan Indonesia berbasis python.
2. Penerapan pemrograman python dan *library* pada python untuk game quiz.
3. Memberikan sebuah pilihan yang memiliki hasil yang berbeda pada game quiz akan diterapkan pada pemilihan tingkat kesulitan sehingga soal yang dihadapi berbeda dan nilai yang didapat sesuai dengan tingkat kesulitan yang dipilih.

1.5. Manfaat

Berikut ini adalah manfaat yang berdampak positif akibat dari tercapainya tujuan penelitian;

1. Pelajar dimudahkan mempelajari sejarah kemerdekaan Indonesia dengan memainkan game Quiz.
2. Hasil nilai dari kuis dapat digunakan sebagai skala ukur kemampuan pengguna.
3. Mendalami peristiwa-peristiwa penyusun kemerdekaan Indonesia.

1.6. Sistematika Penulisan

Sistematika penulisan dalam laporan tugas akhir ini, disusun sebagai berikut:

BAB I. PENDAHULUAN

Bab ini berisi latar belakang masalah, rumusan masalah, tujuan penelitian, batasan masalah, manfaat, dan sistematika penulisan.

BAB II. LANDASAN TEORI

Bab ini berisi tentang teori-teori yang digunakan pada penelitian dan penelitian terdahulu yang relevan dengan penelitian yang akan dilaksanakan.

BAB III. ANALISA DAN PERANCANGAN

Bab ini menjelaskan tentang variabel penelitian, metode pengumpulan data, metode pengembangan sistem, analisa kebutuhan sistem, dan perancangan sistem.

BAB IV. IMPLEMENTASI

Bab ini menjelaskan bagaimana proses dan alur kerja aplikasi yang dibangun.

BAB V. PENGUJIAN DAN PEMBAHASAN

Bab ini akan menjelaskan serta sekaligus menguji coba aplikasi yang dibangun.

BAB VI. KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dan saran tentang aplikasi *game* “*Quiz*”.

DAFTAR PUSTAKA

Pada daftar pustaka memuat pustaka yang diacu dalam laporan proyek akhir.

BAB II

LANDASAN TEORI

2.1. Penelitian yang Relevan

Pada sub bab ini akan di jelaskan penelitian terdahulu yang dilakukan oleh Ningrum dan Maslan dengan judul “Aplikasi Trivia Game Mata Pelajaran Sejarah Berbasis Android” begitu pula penelitian yang dilakukan oleh Rafiqin dan Saputra dengan judul “Pembuatan Aplikasi Game Kuis “Pontianak Punye” Berbasis Android” dan yang terakhir penelitian yang dilakukan oleh Barros, Marisa, dan Wijaya dengan judul “Pembuatan Game Kuis Siapa Pintar”.

Tahun 2020, telah dilakukan penelitian oleh Ningrum dan Maslan dengan judul “Aplikasi Trivia Game Mata Pelajaran Sejarah Berbasis Android”. Tujuan penelitian tersebut adalah membuat game trivia (tebak-menebak) dengan studi kasus pelajaran sejarah yang dikembangkan di android menggunakan game engine construct 2 untuk melawan minimnya pemahaman dan pengetahuan tentang sejarah kemerdekaan Indonesia kepada siswa. Penelitian tersebut menghasilkan sebuah game yang dapat digunakan untuk melatih siswa dalam pemahaman tentang sejarah kemerdekaan Indonesia (Ningrum & Maslan, 2020).

Pada tahun sebelumnya (2017) pernah dilakukan penelitian oleh Rafiqin dan Saputra dengan judul “Pembuatan Aplikasi Game Kuis “Pontianak Punye” Berbasis Android”. Tujuan penelitian tersebut adalah membuat game quiz sebagai media untuk memperkenalkan dengan luas keanekaragaman khas kota Pontianak yang minim dimengerti masyarakat. Game tersebut juga dikembangkan dengan game engine yang sama yaitu construct 2. Penelitian yang dilakukan menghasilkan sebuah game quiz yang dapat memperkenalkan ciri khas keanekaragaman kota Pontianak dan juga memperkenalkan sejarah sejarah begitu pula tempat wisata kota Pontianak (Rafiqin & Saputra, 2017).

Pada tahun selanjutnya yaitu pada tahun 2018 dilakukan penelitian oleh Barros, Marisa dan Wijaya dengan judul “Pembuatan Game Kuis Siapa Pintar”. Tujuan penelitian dilaksanakan adalah membuat game sebagai media pembelajaran untuk memancing minat anak terhadap materi pelajar sambil memainkan game.

Hasil dari penelitian adalah sebuah game seperti cerdas cermat yang menggunakan algoritma Fisher-Yates untuk melakukan acak (*randomize*) pada soal yang dihadapi pengguna, sehingga pengguna satu dengan yang lain memiliki kemungkinan menghadapi soal yang berbeda (Barros et al., 2018).

Pada tabel 2.1 menunjukkan ringkasan penelitian terdahulu untuk menjelaskan perbedaan penelitian terdahulu dan penelitian yang akan dilaksanakan.

Tabel 2. 1 Perbandingan Studi Literatur

No	Judul	Tujuan	Fitur	Hasil
1	Aplikasi Trivia Game Mata Pelajaran Sejarah Berbasis Android	Tujuan penelitian tersebut adalah membuat game trivia (tebak-menebak) dengan studi kasus pelajaran sejarah yang dikembangkan di android	1. Registrasi User 2. Login User 3. Menampilkan Nilai 4. Penjelasan materi-materi sejarah kemerdekaan Indonesia	Menghasilkan sebuah game yang dapat digunakan untuk melatih siswa dalam pemahaman tentang sejarah kemerdekaan Indonesia
2	Pembuatan Aplikasi Game Kuis “Pontianak Punye” Berbasis Android	Membuat game quiz sebagai media untuk memperkenalkan dengan luas keanekaragaman khas kota Pontianak yang minim dimengerti masyarakat	1. Cerita sejarah kota Pontianak 2. Permainan soal tentang kota Pontianak	Menghasilkan sebuah game quiz yang dapat memperkenalkan ciri khas keanekaragaman kota Pontianak dan juga memperkenalkan sejarah sejarah begitu pula tempat wisata kota Pontianak
3	Pembuatan Game Kuis Siapa Pintar	Membuat game sebagai media pembelajaran untuk memancing minat anak terhadap materi pelajar sambil memainkan game.	1. Sound effect 2. Pengacakan (<i>randomize</i>) soal menggunakan algoritma Fisher-Yates 3. Menampilkan Nilai	Sebuah game seperti cerdas cermat yang menggunakan algoritma Fisher-Yates untuk melakukan acak (<i>randomize</i>) pada soal yang dihadapi pengguna, sehingga pengguna satu dengan yang lain memiliki kemungkinan

				menghadapi soal yang berbeda
4	Game Edukasi “Quiz” Bertema Sejarah Indonesia Berbasis Python	Membuat sebuah game edukatif sejarah Indonesia.	<ol style="list-style-type: none"> 1. Fitur memilih tingkat kesusahan (<i>difficulty</i>). 2. Registrasi User. 3. Login User. 4. Menampilkan nilai 	

Game Edukasi “Quiz” Bertema Sejarah Indonesia Berbasis Python memiliki perbedaan dengan penelitian terdahulu yaitu, pembuatan game menggunakan bahasa pemrograman python yang menggunakan module pygame.

2.2. Teori Penelitian

Sub bab ini akan menjelaskan tentang dasar teori penelitian yang akan digunakan pada penelitian ini.

2.2.1. Game Quiz

Game dapat diartikan sebuah aktivitas yang bertujuan untuk mendapatkan kesenangan. Meskipun tujuan utama dari game adalah untuk kesenangan, game juga dapat memiliki tujuan tertentu sesuai dengan game yang dibuat misalnya bertujuan untuk pendidikan atau menambah wawasan. Game dijabarkan ke dalam berbagai bentuk dan salah satunya adalah video game, yaitu permainan elektronik yang dimainkan mengendalikan gambar di layar video (Isammudin et al., n.d.)

Quiz atau kuis menurut KBBI (Kamus Besar Bahasa Indonesia) adalah ujian lisan atau tertulis yang singkat, acara hiburan atau televisi yang berupa perlombaan adu cepat menjawab, cepat tepat, dan cerdas cermat. Dapat diartikan bahwa kuis adalah sebuah ujian atau ajang lomba atau permainan menjawab pertanyaan secara cepat dan cermat. Dalam konteks kali ini kita akan menggunakan kuis adalah sebuah ajang lomba atau permainan menjawab pertanyaan secara cepat dan cermat.

Jadi jika dua arti tersebut digabungkan game quiz atau game kuis adalah sebuah permainan edukatif yang menyenangkan yang dapat melatih kecermatan dan menambah wawasan sang pemain. Menyenangkannya sebuah permainan

diambil dari berapa banyak nilai atau skor yang didapatkan oleh pemain. Maka dari itu sebuah game quiz jika ingin dianggap menyenangkan maka diharuskannya memiliki sebuah sistem *scoring*.

Pada penelitian ini, akan dilakukannya pembuatan sebuah game quiz bertema sejarah kemerdekaan Indonesia. Game quiz ini akan menerapkan sistem *scoring* untuk membuat game quiz ini untuk menghitung nilai pengguna.

2.2.2. Sejarah Kemerdekaan Indonesia (20 Mei 1908 – 17 Agustus 1945)

Berikut-berikut ini adalah peristiwa-peristiwa penyusun kemerdekaan Indonesia yang akan menjadi soal pada game Quiz. Peristiwa ini adalah peristiwa-peristiwa yang terjadi pada atau diantara 20 Mei 1908 sampai dengan 17 Agustus 1945.

Bangsa Indonesia memperingati Hari Kebangkitan Nasional pada tanggal 20 Mei setiap tahunnya. Tanggal tersebut bertepatan dengan tanggal lahir organisasi Budi Utomo atau dengan ejaan lama Boedi Oetomo pada tahun 1908 atau 37 tahun sebelum Indonesia Merdeka. Hari lahir Boedi Oetomo tersebut dianggap sebagai langkah awal menuju lahirnya semangat kebangsaan Indonesia (Purwanto, 2021).

Tanggal 20 Mei disepakati bahwa merupakan hari kebangkitan nasional yang dimana merupakan hari didirikannya organisasi Boedi Oetomo yang merupakan pencetus kemerdekaan Indonesia menurut Soekarno pada dokumen Kesatuan Nasional tahun 1948.

Tetapi sebelum ditetapkannya oleh Soekarno sebelum pada tahun tersebut terjadi peristiwa-peristiwa penyusun proklamasi kemerdekaan. Semisal halnya sidang BPUPKI (Badan Penyelidik Usaha-usaha Persiapan Kemerdekaan Indonesia) dan PPKI (Panitia Persiapan Kemerdekaan Indonesia) yang didirikan oleh penjajah Jepang. Fungsi dari BPUPKI adalah menjadi saran memperoleh gambaran tentang bentuk negara, sistem pemerintahan, dan dasar hukum negara yang merdeka. Sedangkan PPKI menurut Jepang badan tersebut bertindak sebagai badan yang mempersiapkan penyerahan kekuasaan pemerintah dari tantara Jepang kepada badan tersebut, badan ini juga wajib menyelesaikan dan mengesahkan

Rancangan Undang-Undang Dasar dan Falsafah Negara Indonesia Merdeka yang sudah disiapkan oleh BPUPKI (Rendra Sanjaya, 2021).



Gambar 2. 1 Rapat PPKI (Sumber: kompaspedia.kompas.id)

Kemudian dengan desakan dari golongan muda maka sepakat untuk segera memerdekakan Indonesia, naskah kemerdekaan disusun oleh Soekarno, Mohammad Hatta, dan Soebardjo di rumah Laksamana Maeda kemudian susunan naskah tersebut diketik oleh Sayuti Melik dengan meminjam Kolonel Kandeler untuk mengetik naskah proklamasi. Pada 17 Agustus 1945, pukul 10.00 WIB, di Jalan Pegangsaan Timur No. 56 dibacakannya naskah proklamasi (Dwi Latifatul Fajri, 2021).

2.2.3. SQLite

SQLite adalah *database* yang tertanam pada suatu program tanpa harus menjalankan proses sendiri. Mengapa demikian, karena sqlite berdampingan dengan ruang proses suatu *program*. Tetapi didalamnya tetap ada sebuah *database* yang sedang bekerja didalam *program* tersebut (Allen & Owens, 2010).



Gambar 2. 2 Logo SQLite (Sumber: sqlite.org)

SQLite ini memiliki kelebihan yaitu dia adalah sebuah *database* yang tertanam didalam sebuah program. Tetapi dengan demikian SQLite juga membutuhkan *resource* yang diambil dari RAM (*Random Access Memory*) dari *hardware* pengguna. Menurut Allen & Owens perhitungan pengambilan RAM adalah 256 bytes setiap 1 MB *database*.

Pada penelitian ini dikarenakan akan menggunakan bahasa pemrograman python maka diperlukannya sebuah *library* bernama *pysqlite*. Kemudian setelah memasang *library*, maka akan dirancangnya sebuah tabel yang akan dibahas pada penelitian ini.

SQLite juga memiliki bahasa SQL yang dimengerti oleh SQLite. Bahasa SQLite gunakan adalah bahasa standar SQL yang dapat ditemukan pada SQL selain SQLite. Tidak hanya bahasa standar SQL tetapi SQLite juga memiliki fitur tersendiri yang hanya dapat ditemukan pada SQLite.

2.2.4. UML (Unified Modeling Language)

Unified Modeling Language (UML) adalah sebuah bahasa yang berisikan koleksi konsep skematik (*semantics*) dengan notasi (*syntax*) dan aturan-aturan yang digunakan untuk mengatur konsep dan notasi (Sinan & Alhir, 1999).

UML memiliki fungsi untuk menjelaskan atau mendokumentasikan bagaimana cara kerja suatu proses. Semisalnya pada penelitian ini adalah menjelaskan proses kerja suatu program. UML sendiri memiliki fase yaitu *use case diagram*, *activity diagram*, *sequence diagram*, dan *class diagram*.









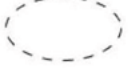

Pada penelitian ini, diagram yang akan digunakan adalah *use case*, *activity*, *sequence*, dan *class diagram*. UML juga memiliki aturan-aturan untuk pembuatan sebuah diagram sehingga pengertiannya *universal* atau dapat dimengerti banyak

orang. Penerapan UML sendiri akan dicontohkan pada penelitian ini pada bab III dimana akan dilakukannya perancangan suatu sistem.

UML memiliki simbol yang sudah memiliki arti yang paten. Simbol-simbol dari UML sendiri adalah aktor, proses, *decision*, start, dan end. Berikut akan dijelaskan simbol dan penjelasan setiap simbol-simbolnya:

1. Simbol Use Case Diagram







Berikut ini adalah simbol-simbol yang digunakan oleh *use case diagram* dan juga penjelasan setiap simbolnya:

SIMBOL	NAMA	KETERANGAN
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
	<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (<i>sinergi</i>).
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Gambar 2. 3 Penjelasan simbol-simbol pada *use case diagram* (Sumber: caraguna.com)

2. Simbol Activity Diagram








Berikut ini adalah simbol-simbol yang digunakan oleh *activity diagram* dan juga penjelasan setiap simbolnya:

Simbol	Nama	Keterangan
	Status awal	Sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan / Decision	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan / Join	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
	Swimlane	Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Gambar 2. 4 Penjelasan simbol-simbol pada *activity diagram* (Sumber: decoding.com)

3. Simbol *Sequence Diagram*








Berikut ini adalah simbol-simbol yang digunakan oleh *sequence diagram* dan juga penjelasan setiap simbolnya:

No	Simbol	Nama	Fungsi
1		Object	Komponen utama Sequence Diagram
2		Actor	Menggambarkan orang yang sedang berinteraksi dengan sistem
3		Entity Class	Menggambarkan hubungan kegiatan yang akan dilakukan
4		Boundary Class	Menggambarkan sebuah penggambaran dari form
5		Control Class	Menggambarkan penghubung antara boundary dengan tabel
6		Life Line	Menggambarkan tempat mulai dan berakhirnya sebuah message
7		Message	Menggambarkan pengiriman pesan

Gambar 2. 5 Penjelasan simbol-simbol pada *sequence diagram* (Sumber: devyferdiansyah.com)

4. Simbol Class Diagram

Berikut ini adalah simbol-simbol yang digunakan oleh *class diagram* dan juga penjelasan setiap simbolnya:

SIMBOL CLASS DIAGRAM			
NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

Gambar 2. 6 Penjelasan simbol-simbol pada *class diagram* (Sumber: pinhome.co.id)

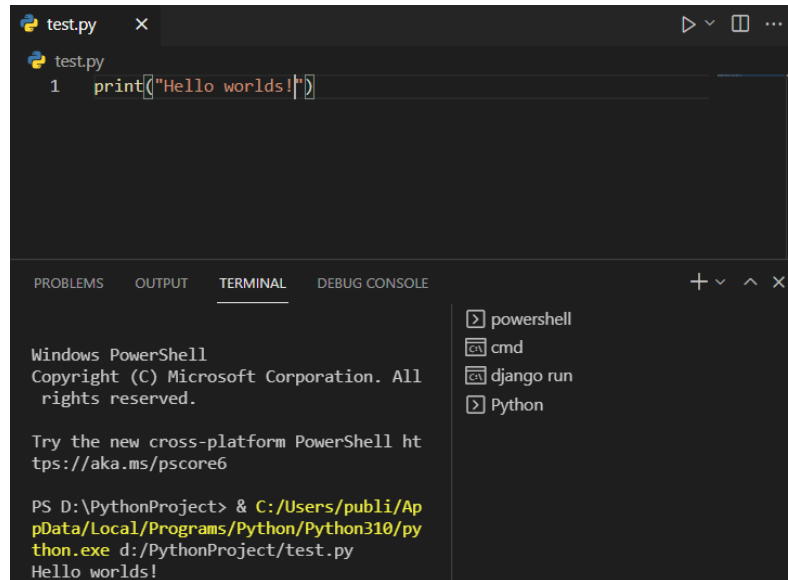
2.2.5. Python

Python merupakan suatu bahasa pemrograman yang *interpreted*, berbasis-objek (*object-oriented*), *high-level programing language with dynamic semantics*. *High level* yang terbangun pada struktur data digabung dengan *dynamic typing* dan *dynamic binding* membuatnya menarik bagi *rapid application development* (Python, n.d.).

Bahasa pemrograman python ini berfungsi untuk melakukan pengembangan game Quiz yang akan dilakukan pada penelitian ini. Diambil bahasa python dikarenakan python dapat berjalan di hampir semua perangkat, contohnya adalah raspberry pi. Versi python yang akan digunakan pada penelitian ini adalah versi python 3.10 keatas.

Bahasa pemrograman python untuk dapat bekerja diperlukannya sebuah IDE yang memiliki *support* (dukungan) python contohnya adalah visual studio. Setelah itu dilakukan pemrograman pada IDE. Kemudian ketika akan melakukan distribusi aplikasi dapat menggunakan sebuah library yang tersedia pada python.

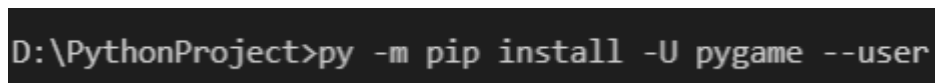
Python sendiri memiliki kemipiran *syntax* dengan *C++* dan *Java*. Python sendiri memiliki banyak *library* contohnya yang akan digunakan pada penelitian ini adalah *pygame*. Berikut adalah contoh implementasi pemrograman pada python yang akan ditunjukkan pada gambar dibawah ini:



Gambar 2. 7 Contoh perintah *print* pada python

2.2.6. Pygame

Pygame adalah sebuah module dari python yang di desain untuk merancang video games. Pygame menambahkan *functionality* dengan SDL library yang baik. Fitur tersebut memudahkan pengguna untuk memfiturkan program game dan multimedia pada bahasa python (Pete Shinnars, n.d.).



Gambar 2. 8 Perintah untuk install pygame pada python

Pygame adalah sebuah *library* ekstensi dari python yang berfungsi sebagai *library* untuk pengembangan game pada python. Pygame sendiri dapat menerima *input* (masukan) dari *joystick* dan dapat mengatur penggunaan *resource* pada sistem *hardware* pengguna sehingga performa akan terasa optimal. Pygame yang mendukung python versi 3.x adalah pygame 1.9.2 keatas, jadi waktu penelitian dilaksanakan akan menggunakan versi pygame diatas versi 1.9.2.

Penggunaan pygame sendiri dapat dengan mudah dilakukan, cukup dengan memanggil *pip install pygame* maka *library* pygame akan tersedia pada python. Untuk dapat melaksanakan perintah-perintah yang ada pada pygame cukup melakukan *import* pada *line* awal dari *file* py. Pygame sendiri terdapat modul yang dapat digunakan sebagai sebuah dasar pengembangan game ataupun *library game*.

Pygame sendiri dapat menerapkan seperti tampilan GUI (*Graphical User Interface*) untuk mempermudah pengguna mengendalikan *game* yang dibuat. Tidak hanya GUI saja tetapi dalam pygame dapat diterapkan sebuah database, pada penelitian ini contohnya menggunakan sqlite sebagai database game. Berikut adalah contoh dari implementasi pygame pada pemrograman python:

```

1  import pygame
2  from pygame.locals import *
3
4  pygame.init()
5  vec = pygame.math.Vector2 # 2 for two dimensional
6
7  HEIGHT = 450
8  WIDTH = 400
9  ACC = 0.5
10 FRIC = -0.12
11 FPS = 60
12
13 FramePerSec = pygame.time.Clock()
14
15 displaysurface = pygame.display.set_mode((WIDTH, HEIGHT))
16 pygame.display.set_caption("Game")

```

Gambar 2. 9 Implementasi pygame untuk membuat window tab

2.2.7. Pillow (Python Image Library)

Pillow (Python Image Library) merupakan suatu *library* pada *python* untuk memproses *image* (gambar) sehingga pada *program python* dapat menampilkan gambar (Clark, 2019).

```
$ python pip install Pillow
```

Gambar 2. 10 Perintah untuk install pillow pada python

Penggunaan *pillow* sendiri dilakukan, cukup dengan memanggil *pip install pillow* maka *library pillow* akan tersedia pada *python*. Untuk dapat melaksanakan perintah-perintah yang ada pada *pillow* cukup melakukan *import* pada *line* awal dari *file* py.

```

from tkinter import *
from PIL import ImageTk, Image

root = Tk()
root.title('Learn To Code at Codemy.com')
root.iconbitmap('c:/gui/codemy.ico')

```

Gambar 2. 11 Implementasi pillow untuk menampilkan image icon

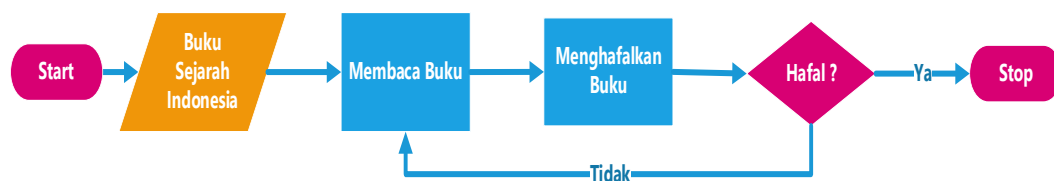
BAB III

ANALISIS DAN PERANCANGAN

3.1. Analisis Permasalahan

Analisis permasalahan bertujuan untuk mengetahui kebutuhan sistem yang akan dibangun. Analisis sistem juga diperlukan untuk mempelajari apa dan bagaimana operasi dari sistem yang ada sebelum mencoba untuk menganalisa permasalahan, kelemahan dan kebutuhan pemakai sistem sehingga dapat memberikan rekomendasi penyelesaian permasalahan tersebut. Sebelum dilakukan penelitian ini, untuk mempelajari sejarah Indonesia diperlukannya sebuah buku yang berisi tentang sejarah Indonesia, terkadang merasa bosan membaca buku teramat tebal yang berisi tentang sejarah Indonesia.

Selama ini untuk mempelajari sejarah tentang kemerdekaan Indonesia diperlukannya mencari sebuah buku yang membahas tentang sejarah Indonesia. Setelah buku tentang pembahasan sejarah Indonesia ini ditemukan, maka diperlukannya untuk membaca buku tebal tentang sejarah Indonesia. Kemudian hasil bacaan dari buku ini dihafalkan, yang padahal untuk menghafalkan tokoh yang banyak pada sejarah Indonesia dan juga peristiwa-peristiwa membutuhkan untuk membaca ulang sehingga menghafalkan sejarah Indonesia. Jika digambarkan dalam sebuah *flowchart* maka proses bisnis belajar sejarah tentang kemerdekaan Indonesia adalah sebagai berikut:



Gambar 3. 1 Proses bisnis belajar sejarah Indonesia

Dengan demikian timbulnya ketidak-minatan kepada sejarah Indonesia karena metode yang digunakan adalah menggunakan buku yang teramat tebal sehingga bosan untuk membacanya padahal menurut R. Moh. Ali Sejarah merupakan suatu kejadian-kejadian, peristiwa-peristiwa yang berhubungan dengan manusia, yakni menyangkut perubahan yang nyata di dalam kehidupan manusia (Ali, 2005).

Dengan masalah-masalah yang timbul maka diperlukannya sebuah metode belajar sejarah Indonesia yang *fun* (menyenangkan) dan menarik yang dimana pengguna nanti tidak merasa bosan ketika menjalankan metode tersebut, maka diusulkannya sebuah game berupa kuis yang berisi mengenai sejarah Indonesia.

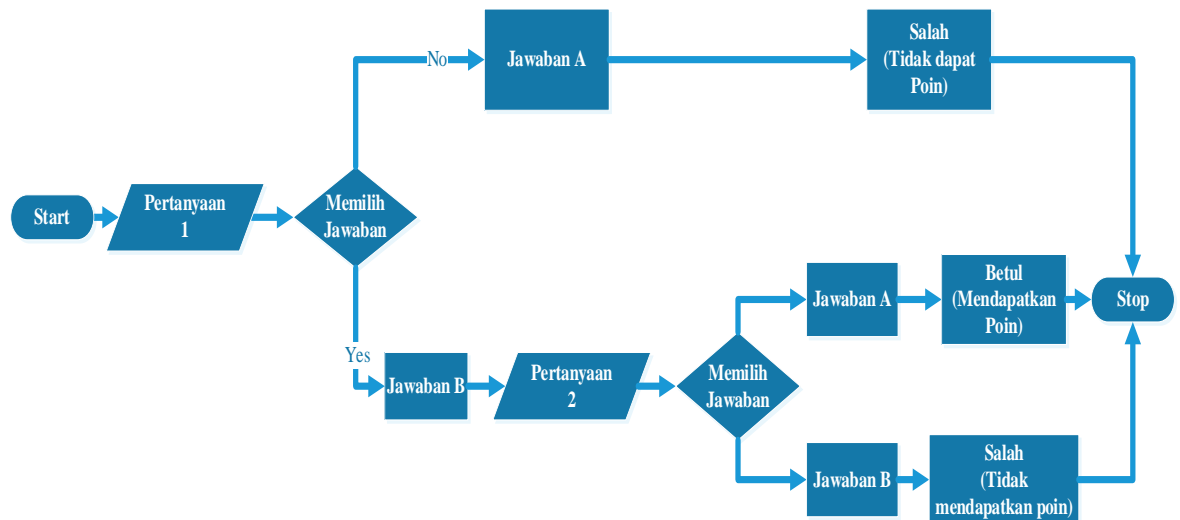
3.2. Analisis Pemecahan Masalah

Berdasarkan analisis permasalahan yang ada, maka dapat diusulkan suatu pembuatan game berupa kuis yang bertema sejarah Indonesia menggunakan bahasa python yang kemudian dijalankan pada sebuah konsol raspberry pi berbasis linux. Sistem game yang diusulkan digunakan oleh satu user. Sistem game Quiz ini digunakan untuk media pembelajaran yang *fun* (menyenangkan) dan menarik. Sistem game ini memiliki fitur-fitur yaitu memilih tingkat kesulitan, registrasi user, login user, dan menampilkan nilai. Sistem game Quiz ini dibangun menggunakan bahasa pemrograman python dengan module pygame.

Didalam penelitian ini, penerapan yang dilakukan adalah memberikan opsi untuk memilih tingkat kesulitan yang berpengaruh kepada soal kuis yang dihadapi pengguna. Sehingga dapat melatih pengguna dan tetap membuat game kuis ini menyenangkan untuk dimainkan, maka dari itu juga diterapkan sebuah skoring berupa nilai yang didapat setelah menjawab soal kuis yang dihadapi dan setiap tingkat kesulitan juga menghasilkan nilai yang berbeda pula.

3.2.1. Implementasi Algoritma pada Game Quiz

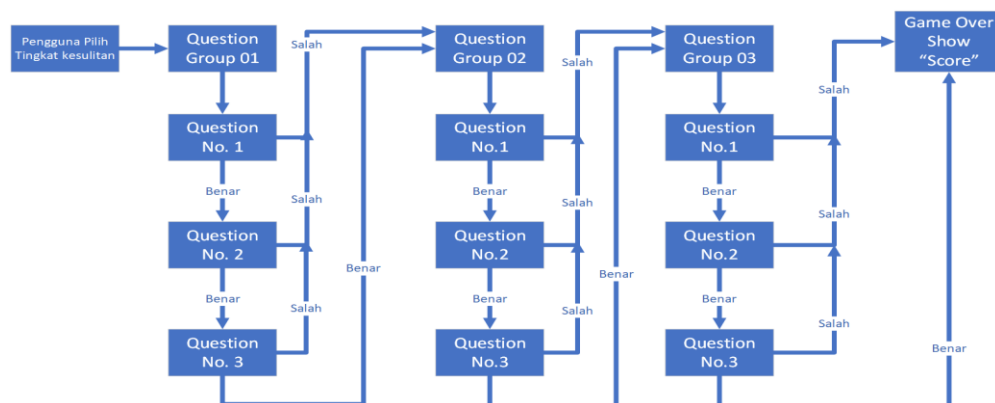
Algoritma yang diimplementasikan pada *game* ini mulai bekerja dengan data yang tersedia dan menggunakan aturan-aturan inferensi untuk mendapatkan data yang lain sampai sasaran atau kesimpulan didapatkan. Setelah dilakukannya aturan-aturan inferensi maka akan menghasilkan satu dari *antecedent* (*IF-THEN*). Ketika aturan tersebut ditemukan maka mesin pengambil keputusan dapat membuat kesimpulan, atau konsekuensi (*THEN*), yang menghasilkan informasi tambahan yang baru dari data yang disediakan. Mesin akan mengulang melalui proses ini sampai sasaran ditemukan. Berikut ini adalah penggambaran algoritma yang akan diterapkan pada penelitian:



Gambar 3. 2 Flowchart algoritma pada *game quiz*

Pengguna setelah memulai akan menerima pertanyaan 1 yang kemudian pengguna dapat memilih jawaban dari pertanyaan 1 yang disediakan (Jawaban A dan B) yang kemudian akan memberikan hasil yang berbeda jika menjawab dengan benar maka pengguna akan menerima pertanyaan 2, apabila dijawab salah maka pengguna tidak dapat poin dan *game* akan berakhir. Pada pertanyaan 2 dilakukan penerapan yang sama dengan pertanyaan 1, yang membedakan adalah apabila dijawab benar maka poin akan dibawa ke akhir dari *game* dan jika salah maka poin dari pertanyaan terakhir tidak akan dibawa ke akhir *game*.

Gambar berikut akan menjelaskan lebih lengkap mengenai bagaimana implementasi algoritma yang akan diterapkan pada *game quiz* edukasi ini. Keseluruhan cara sistem aplikasi memberikan *output* atas *input* dari user akan ditampilkan pada flowchart ini:



Gambar 3. 3 Chart keseluruhan dari penerapan algoritma pada *game quiz*

Cara kerjanya adalah sebagaimana ketika pengguna atau *player* dihadapkan sebuah pertanyaan yang dimana pengguna diharuskan menjawab salah satu pilihan jika pilihan jawaban benar maka pengguna akan dihadapkan ke pertanyaan selanjutnya yang kemudian akan dihadapkan sebuah kondisi jika salah menjawab maka pengguna tidak mendapatkan poin jika menjawab dengan benar maka pengguna akan mendapatkan poin.

3.2.2. Metode Pengambilan Data (user requirement)

Metode pengambilan data adalah langkah-langkah yang digunakan untuk memperoleh data yang diperlukan untuk menyusun laporan tugas akhir. Metode yang dilakukan adalah studi pustaka, dilakukan untuk mengumpulkan informasi yang dibutuhkan untuk merancang dan membuat game kuis bertema sejarah Indonesia menggunakan bahasa pemrograman python. Studi Pustaka dilakukan dengan pencarian referensi-referensi yang terkait dengan hal tersebut melalui internet, jurnal dan juga buku-buku yang menyangkut perancangan game kuis berbahasa program python dengan modul pygame yang menggunakan algoritma forward chaining untuk berjalan pada konsol raspberry pi yang berbasis linux. Referensi yang digunakan sebagai sumber informasi dapat dilihat pada daftar Pustaka diakhir proposal.

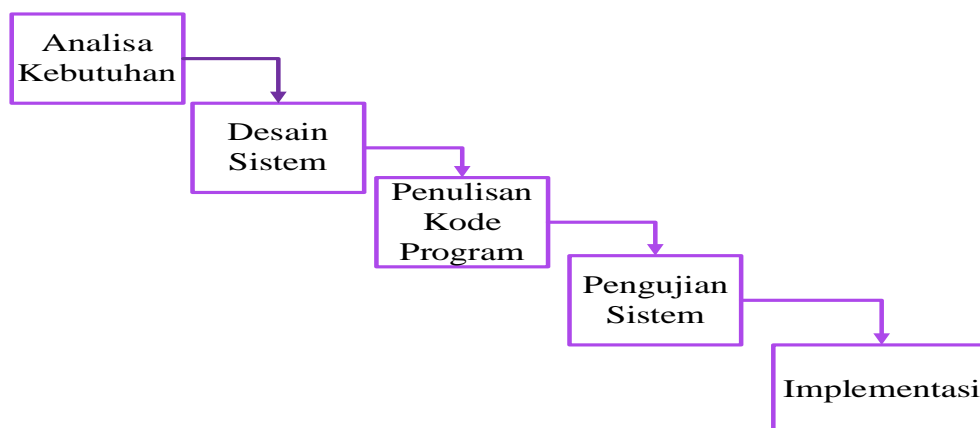
Pengambilan data sejarah Indonesia diambilkan dari peristiwa-peristiwa yang terjadi pada 20 Mei 1908 disaat dibangunnya suatu organisasi bernama Boedi Oetomo yang dianggap penggagas kemerdekaan Indonesia. Hingga terjadinya peristiwa proklamasi kemerdekaan Indonesia pada 17 Agustus 1945. Jalan menuju kemerdekaan Indonesia sendiri tidak dihadapi dengan jalan yang mulus, sehingga peristiwa-peristiwa yang terjadi dapat dijadikan bahan pembelajaran. Bahan pembelajaran ini kemudian akan dijadikan sebuah soal kuis yang akan tertanam pada pemrograman python.

Pengambilan detail data yang ada pada database SQLite adalah pengambilan data pengguna dan juga data nilai yang dihasilkan pengguna. Saat pengguna melakukan registrasi maka data yang diminta oleh program akan disimpan pada database sehingga sewaktu login akan mengenali pengguna. Kemudian disaat menjalankan game maka pengguna akan dihadapi kuis yang dapat menghasilkan

sebuah nilai dan nilai ini akan disimpan pada database kemudian ditampilkan nilai tertinggi yang diterima pengguna.

3.2.3. Metode Pengembangan Sistem

Metode waterfall merupakan metode yang sering digunakan oleh penganalisa sistem pada umumnya karena metode ini bersifat sistematis dan sekuensial, artinya setiap tahapan dalam metode ini dilakukan secara berurutan dan berkelanjutan. Tahapan utama dari metode waterfall mencerminkan aktifitas pengembangan dasar. Terdapat 5 tahapan pada metode waterfall, yaitu analisa kebutuhan, desain sistem, penulisan kode program, pengujian program, dan implementasi dan perawatan. Inti dari metode waterfall adalah pengerjaan dari suatu sistem dilakukan secara berurutan atau secara linear. Jadi jika langkah ke-1 belum dikerjakan, maka langkah 2 tidak dapat dikerjakan. Jika langkah ke-2 belum dikerjakan maka langkah ke-3 juga tidak dapat dikerjakan, begitu seterusnya. Secara otomatis langkah ke-3 akan bisa dilakukan jika langkah ke-1 dan ke-2 sudah dilakukan.



Gambar 3. 4 Metode Waterfall

Gambar 3.4 Metode Waterfall secara garis besar menggambarkan metode waterfall yang mempunyai 5 tahapan yaitu: Analisa, Desain, Penulisan, Pengujian dan Implementasi. Detail tahapan metode waterfall adalah sebagai berikut:

a. Analisa Kebutuhan

Langkah ini merupakan analisa terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini bisa melakukan sebuah penelitian, wawancara atau studi literatur.

Kebutuhan yang dibutuhkan untuk melakukan penelitian ini adalah studi pustaka yang terkait dengan pengembangan game pada python kemudian implementasi algoritma pada game. Untuk membangun sebuah game quiz diperlukannya sebuah soal pertanyaan, maka dari itu diambilkan pustaka yang berisi mengenai sejarah Indonesia pada tempo 20 mei 1908 hingga 17 agustus 1945 untuk dijadikan sebuah soal pertanyaan pada kuis.

b. Desain Sistem

Tahapan dimana dilakukan penuangan pikiran dan perancangan sistem terhadap solusi dari permasalahan yang ada dengan menggunakan perangkat pemodelan sistem serta struktur dan bahasan data.

Desain yang akan diterapkan pada sistem akan menggunakan database yang digunakan untuk menyimpan data pengguna yang berupa id, password, dan nilai yang diterima ketika selesai mengerjakan kuis.

c. Penulisan Kode Sistem

Penulisan kode program atau coding merupakan penerjemahan design dalam bahasa yang bisa dikenali oleh komputer. Dilakukan oleh programmer yang akan menerjemahkan transaksi yang diminta oleh user. Tahapan ini lah yang merupakan tahapan secara nyata dalam mengerjakan suatu sistem. Dalam artian penggunaan komputer pemikiran logika akan dimaksimalkan dalam tahapan ini.

Penulisan kode pada penelitian ini akan dilakukan pada bahasa pemrograman python yang menggunakan modul *library* pygame untuk mengakses SDL Library berupa GUI dan lain-lain yang tersedia pada pygame.

d. Pengujian Sistem

Tahapan dimana sistem yang baru diuji kemampuan dan ke-efektifannya sehingga didapatkan kekurangan dan kelemahan sistem yang kemudian dilakukan pengkajian ulang dan perbaikan terhadap aplikasi menjadi lebih baik dan sempurna. Tujuan pengujian adalah menemukan kesalahan-kesalahan terhadap sistem tersebut dan kemudian bisa diperbaiki. Pada tahap ini dilakukan setiap pengujian fitur yang sudah dibuat.

e. Implementasi

Perangkat yang sudah disampaikan kepada klien kemungkinan akan mengalami perubahan. Perubahan tersebut terjadi karena dalam implementasinya sistem mengalami ketidaksesuaian dengan lingkungan atau bisa juga karena pelanggan membutuhkan perkembangan fungsional.

Tahapan ini akan dilakukan testing menjalankan game Quiz yang dijalankan oleh pengguna sehingga pengguna dapat merasakan bagaimana menjalankan game Quiz.

3.2.4. Analisis Kebutuhan Sistem (fungsional dan non fungsional system)

Analisis Kebutuhan diperlukan untuk mengidentifikasi dan mengevaluasi permasalahan, kesempatan, dan kebutuhan yang diperlukan sehingga dapat membuat game kuis.

a. Kebutuhan Fungsional

Pada tahap ini dilakukan analisis kebutuhan fungsional sistem untuk membangun Game Edukasi “Quiz” Bertema Sejarah Indonesia Berbasis Python. Hasil dari analisis kebutuhan fungsional sebagai berikut:

1) Player, dapat melakukan aksi berikut:

a) Registrasi

Pengguna dapat melakukan registrasi yang kemudian *id* dan *password* ini dapat digunakan untuk *login*.

b) Login

Pengguna melakukan *login* dengan menggunakan *id* dan *password* yang dibuat melalui registrasi. Kemudian pengguna akan menerima tampilan *menu* utama dari aplikasi.

c) Menjalankan Game

Setelah pengguna melakukan *login* kemudian masuk kedalam *menu* utama yang dimana pengguna dapat memainkan game dengan memilih tingkat kesulitan dan juga dapat melihat *scoreboard* nilai dari setiap tingkat

kesulitan. Pada saat memainkan *game* yang menggunakan metode *forward-chaining* pengguna menerima skor apabila benar, kemudian total skor tersebut masuk ke dalam *scoreboard* dari Melihat Nilai.

d) Melihat Nilai

Disini pengguna dapat melihat *scoreboard* dari setiap tingkat kesulitan yang diurutkan berdasarkan nilai tertinggi dan tanggal/jam yang paling dahulu mendapatkan skor tersebut.

2) Admin, dapat melakukan aksi berikut:

a) Login

Admin melakukan *login* yang kemudian admin akan menerima tampilan utama dari menu admin, disini admin dapat melakukan Kelola Pertanyaan.

b) Kelola Pertanyaan

Disini admin dapat melakukan kelola lihat dan edit pertanyaan dan foto pertanyaan dari setiap *quiz* pertanyaan.

b. Kebutuhan Non Fungsional

Dalam membuat Game Edukasi “Quiz” Bertema Sejarah Indonesia Berbasis Python, dibutuhkan beberapa komponen utama yang mencakup perangkat keras (*hardware*) untuk membangun game (*developing*) dan untuk perangkat keras untuk menjalankannya dan perangkat lunak (*software*).

1. Kebutuhan Perangkat Keras (*Hardware*)

Spesifikasi perangkat keras (*Hardware*) yang digunakan dalam pembuatan game Quiz ini adalah sebagai berikut :

a) Perangkat keras (*Hardware*) untuk membangun (*Developing*)

- 1) Processor : Intel Pentium G4560 3.50Ghz
- 2) RAM : 8GB DDR4 2400 MHz
- 3) Hardisk : 1000 GB (1 TB)

b) Perangkat keras untuk menjalankan game quiz

- 1) Processor : 2 Core Processor
- 2) RAM : 2 GB

2. Kebutuhan Perangkat Lunak (*Software*)

Hardware tidak dapat menyelesaikan masalah tanpa adanya *software*. *Software* merupakan komponen di dalam sistem data berupa program atau intruksi untuk mengontrol suatu sistem. Perangkat lunak yang diperlukan untuk perancangan *game* edukasi “quiz” adalah sebagai berikut :

a) Perangkat lunak untuk mengembangkan (*developing*)

- 1) Visual Studio Code versi 1.63.2.
- 2) Sistem operasi Windows 10.
- 3) Python 3 dengan *library Pygame* dan *Pillow*.

b) Perangkat lunak untuk menjalankan *game quiz*

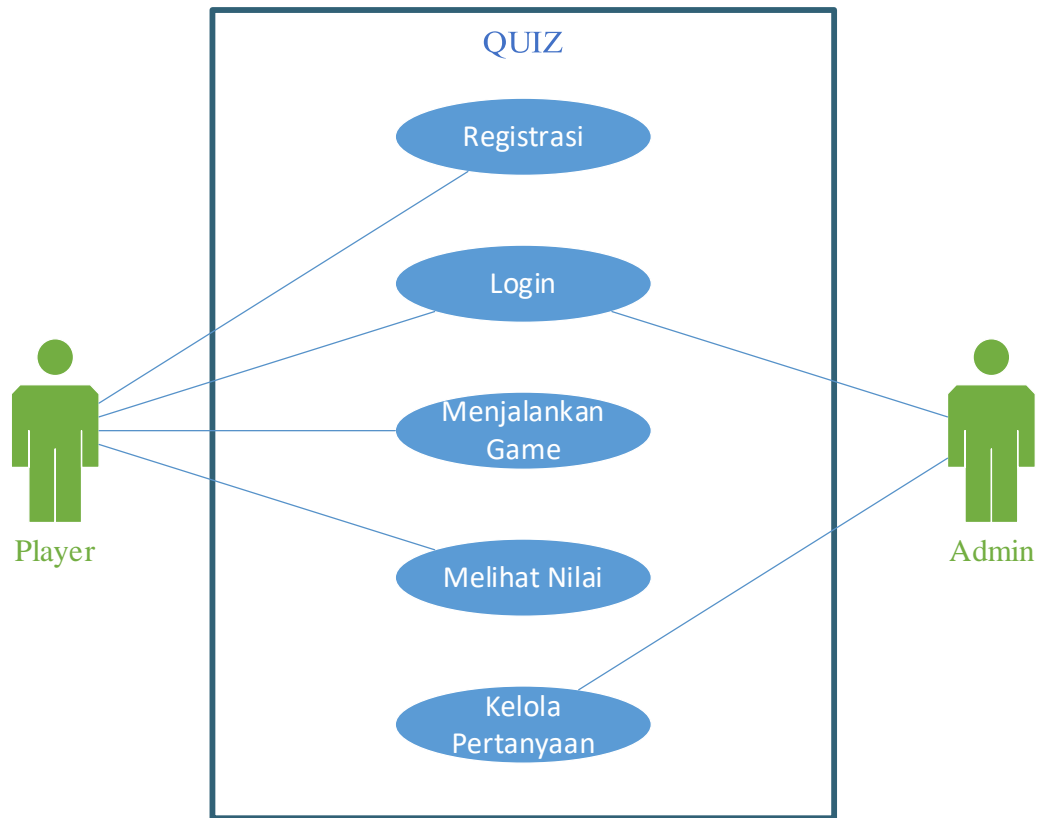
- 1) Sistem Operasi : MacOS, Windows, *Linux based System Operation*.
- 2) *Software* : Python 3 dengan *library Pygame* dan *Pillow*.

3.3. Perancangan Sistem

Perancangan sistem dilakukan untuk mengidentifikasi use case sistem, sistem, relasi antar tabel dalam sistem, arsitektur diagram dan desain user Interface secara lebih terinci, sehingga proses pengembangan dan implementasi bisa mendapatkan gambaran yang jelas mengenai sistem yang baru.

3.3.1. Use Case Diagram

Use case diagram digunakan untuk menggambarkan fungsionalitas yang diharapkan dari sistem, yaitu bagaimana cara user berinteraksi dengan sistem. Sehingga user dapat mengetahui Tindakan apa saja yang dapat dilakukan saat menggunakan aplikasi ini. Berikut ini adalah tampilan *use case diagram* dari Game Edukasi “Quiz” Bertema Sejarah Indonesia Berbasis Python



Gambar 3.5 Use Case Diagram

Gambar 3.5 menggambarkan *use case diagram* Game Edukasi “Quiz” Bertema Sejarah Indonesia Berbasis Python. Dilihat dari use case yang telah dirancang, dapat diambil kesimpulan fitur-fitur yang ada pada game yang dibangun. Fitur tersebut antara lain:

1) Player dapat :

- a) Registrasi
- b) Login
- c) Menjalankan Game
- d) Melihat Nilai

2) Admin dapat :

- a) Login
- b) Kelola Pertanyaan

A. Definisi Aktor

Tabel 3. 1 Definisi Aktor

No	Aktor	Deskripsi
1.	Player	<p>Player adalah aktor yang dapat melakukan aksi sebagai berikut:</p> <ol style="list-style-type: none"> 1. Registrasi. 2. Login. 3. Menjalankan Game. 4. Melihat Nilai.
2.	Admin	<p>Admin adalah aktor yang dapat melakukan aksi sebagai berikut:</p> <ol style="list-style-type: none"> 1. Login. 2. Kelola Pertanyaan.

3.3.2. Skenario *Use Case Diagram*

Skenario *Use Case* mendeskripsikan aktor-aktor yang melakukan prosedur dalam sistem, serta menjelaskan respon yang ditanggapi oleh sistem tersebut terhadap prosedur yang dilakukan oleh aktor. Pada perancangan sistem ini terdapat *use case* yaitu registrasi, *login*, menjalankan game, dan melihat nilai yang dapat dijalankan oleh pengguna (*player*) kemudian untuk *use case* yang dapat dilakukan oleh admin adalah *login* dan kelola pertanyaan. Berikut ini adalah skenario *use case* yang akan diimplementasikan pada game “*Quiz*”:

a. Skenario *Use Case Registrasi*

Berikut ini akan dijelaskan skenario *use case diagram* dari registrasi yang akan dijelaskan pada Tabel 3.2.

Tabel 3. 2 Skenario *Use Case Diagram* Registrasi

Use Case	:	Registrasi
Deskripsi	:	<i>Use case</i> ini menangani verifikasi data dan pendaftaran pengguna baru pada sistem.
Aktor	:	<i>Player</i> .
Kondisi Awal	:	Sistem menampilkan form registrasi.
Kondisi Akhir	:	Menerima <i>pop-up</i> berhasil registrasi.
Skenario Normal		
Aksi Aktor		Reaksi Sistem
1. <i>Use case</i> ini dimulai ketika pengguna memulai aplikasi kemudian pada <i>form</i> registrasi pengguna input <i>username</i> dan <i>password</i> .		
		2. Sistem melakukan check data input dari pengguna dan melakukan check data apakah <i>username</i> identik pada database.
		3. Menyimpan data <i>username</i> dan <i>password</i> ke dalam <i>database</i> .
		4. Sistem menampilkan <i>pop-up</i> registrasi berhasil
Skenario Gagal		
Aksi Aktor		Reaksi Sistem
1. <i>Use case</i> ini dimulai ketika pengguna memulai aplikasi kemudian pada <i>form</i> registrasi pengguna input <i>username</i> dan <i>password</i> .		
		2. Sistem melakukan check data input dari pengguna dan melakukan check data apakah <i>username</i> identik pada database.
		3. Sistem menampilkan <i>pop-up</i> registrasi gagal dikarenakan <i>username</i> sudah ada.

b. Skenario Use Case Login

Berikut ini akan dijelaskan skenario *use case diagram* dari *login* yang akan dijelaskan pada Tabel 3.3.

Tabel 3. 3 Skenario Use Case Diagram Login

Use Case	:	Login
Deskripsi	:	<i>Use case</i> ini menangani verifikasi data dan hak akses untuk masuk sebagai pengguna.
Aktor	:	<i>Player</i> , Admin
Kondisi Awal	:	Sistem menampilkan form login.
Kondisi Akhir	:	Menampilkan menu utama / menu admin.
Skenario Normal		
Aksi Aktor		Reaksi Sistem
1. <i>Use case</i> ini dimulai ketika pengguna memulai aplikasi kemudian pada <i>form login</i> pengguna input <i>username</i> dan <i>password</i> .		
		2. Sistem melakukan verifikasi <i>login</i> dari input <i>username</i> dan <i>password</i> yang dimasukkan pengguna.
		3. Menampilkan menu utama / menu admin
Skenario Gagal		
Aksi Aktor		Reaksi Sistem
1. <i>Use case</i> ini dimulai ketika pengguna memulai aplikasi kemudian pada <i>form login</i> pengguna input <i>username</i> dan <i>password</i> .		
		2. Sistem melakukan verifikasi <i>login</i> dari input <i>username</i> dan <i>password</i> yang dimasukkan pengguna.
		3. Menampilkan menu utama / menu admin

c. Skenario Use Case Menjalankan Game

Berikut ini akan dijelaskan skenario *use case diagram* dari menjalankan game yang akan dijelaskan pada Tabel 3.4.

Tabel 3. 4 Skenario Use Case Menjalankan Game

Use Case	:	Menjalankan Game
Deskripsi	:	<i>Use case ini menangani game yang akan dijalankan player</i>
Aktor	:	<i>Player</i>
Kondisi Awal	:	Sistem menampilkan pemilihan tingkat kesulitan <i>game</i> .
Kondisi Akhir	:	Menampilkan <i>game</i> sesuai dengan tingkat kesulitan yang dipilih
Skenario Normal		
Aksi Aktor		Reaksi Sistem
1. Pengguna memilih tingkat kesulitan <i>game</i> (Pemula / Sedang / Ahli) pada menu utama.		
		2. Sistem mengambil data pertanyaan berdasarkan tingkat kesulitan yang dipilih.
		3. Sistem menampilkan <i>game</i> sesuai dengan tingkat pertanyaan dipilih.
Skenario Gagal		
Aksi Aktor		Reaksi Sistem
1. Pengguna memilih tingkat kesulitan <i>game</i> (Pemula / Sedang / Ahli) pada menu utama.		
		2. Sistem mengambil data pertanyaan berdasarkan tingkat kesulitan yang dipilih.
		3. Sistem gagal menampilkan <i>game</i> sesuai dengan tingkat pertanyaan dipilih.

d. Skenario Use Case Melihat Nilai

Berikut ini akan dijelaskan skenario *use case diagram* dari melihat nilai yang akan dijelaskan pada Tabel 3.5.

Tabel 3. 5 Skenario *use case* melihat nilai

Use Case	:	Melihat Nilai
Deskripsi	:	<i>Use case</i> ini menangani lihat nilai dari <i>game</i>
Aktor	:	<i>Player</i>
Kondisi Awal	:	Sistem menampilkan pemilihan lihat nilai berdasarkan tingkat kesulitan <i>game</i> .
Kondisi Akhir	:	Menampilkan nilai <i>game</i> berdasarkan <i>username</i> sesuai dengan nilai tingkat kesulitan yang dipilih
Skenario Normal		
Aksi Aktor		Reaksi Sistem
1. Pengguna memilih nilai yang dibagi dengan tingkat kesulitan <i>game</i> (Pemula / Sedang / Ahli) pada menu utama.		
		2. Sistem mengambil data nilai berdasarkan nilai per tingkat kesulitan yang dipilih.
		3. Sistem menampilkan nilai <i>game</i> berdasarkan <i>username</i> sesuai dengan tingkat nilai kesulitan dipilih.
Skenario Gagal		
Aksi Aktor		Reaksi Sistem
1. Pengguna memilih nilai yang dibagi dengan tingkat kesulitan <i>game</i> (Pemula / Sedang / Ahli) pada menu utama.		
		2. Sistem mengambil data nilai berdasarkan nilai per tingkat kesulitan yang dipilih.
		3. Sistem gagal menampilkan nilai <i>game</i> berdasarkan <i>username</i> sesuai dengan tingkat nilai kesulitan dipilih.

e. Skenario Use Case Kelola Pertanyaan

Berikut ini akan dijelaskan skenario use case diagram dari kelola pertanyaan yang akan dijelaskan pada Tabel 3.6.

Tabel 3. 6 Skenario use case kelola pertanyaan

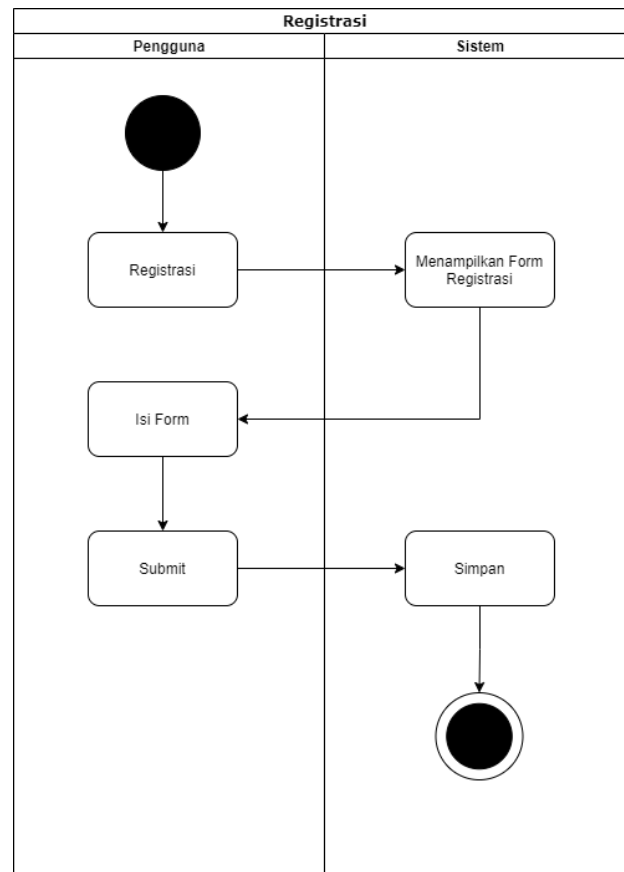
Use Case	:	Kelola Pertanyaan
Deskripsi	:	<i>Use case</i> ini menangani kelola pertanyaan
Aktor	:	Admin
Kondisi Awal	:	Sistem menampilkan menu kelola pertanyaan berdasarkan tingkat kesulitan.
Kondisi Akhir	:	Data pertanyaan tersimpan pada database
Skenario Normal		
Aksi Aktor		Reaksi Sistem
1. Admin memilih edit pertanyaan / edit foto pertanyaan berdasarkan tingkat kesulitan yang dipilih		
		2. Sistem menampilkan data pertanyaan / foto pertanyaan sesuai dengan tingkat kesulitan yang dipilih admin.
3. Admin melakukan mengelola data pertanyaan / foto pertanyaan view, edit.		
		4. Sistem menampilkan data sukses disimpan.
Skenario Gagal		
Aksi Aktor		Reaksi Sistem
1. Admin memilih edit pertanyaan / edit foto pertanyaan berdasarkan tingkat kesulitan yang dipilih		
		2. Sistem menampilkan data pertanyaan / foto pertanyaan sesuai dengan tingkat kesulitan yang dipilih admin.
3. Admin melakukan mengelola data pertanyaan / foto pertanyaan view, edit.		
		4. Sistem menampilkan data gagal disimpan.

3.3.3. Activity Diagram

Menunjukkan bagaimana aktivitas yang terjadi saat user menjalankan retro konsol. Merupakan workflow atau pemodelan aliran aktivitas yang terjadi pada use case. Berikut adalah activity diagram yang ada pada Game Edukasi “Quiz” Bertema Sejarah Indonesia Berbasis Python

a. Activity Diagram Registrasi

Pada bagian ini akan dijelaskan *activity diagram* dari registrasi yang diimplementasikan pada Game Edukasi “Quiz” Bertema Sejarah Indonesia Berbasis Python yang dijelaskan pada Gambar 3.6.

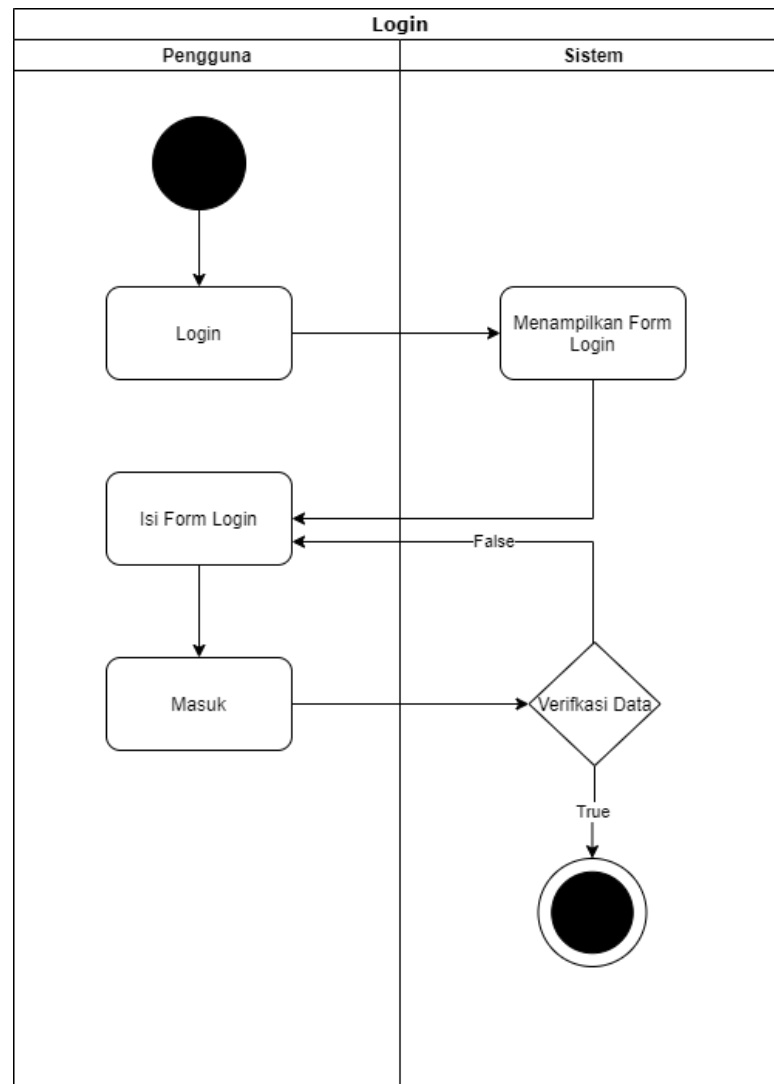


Gambar 3. 6 Activity Diagram Registrasi

Gambar 3.6 menjelaskan tentang alur registrasi yang dilakukan oleh pengguna game quiz. Langkah pertama pengguna harus melakukan proses registrasi kemudian akan dikirimkan form oleh sistem yang kemudian diisi oleh pengguna, kemudian setelah mengisi form pengguna submit data kemudian sistem akan menyimpan data.

b. Activity Diagram Login

Pada bagian ini akan dijelaskan activity diagram dari login yang diimplementasikan pada Game Edukasi “Quiz” Bertema Sejarah Indonesia Berbasis Python yang dijelaskan pada Gambar 3.7.

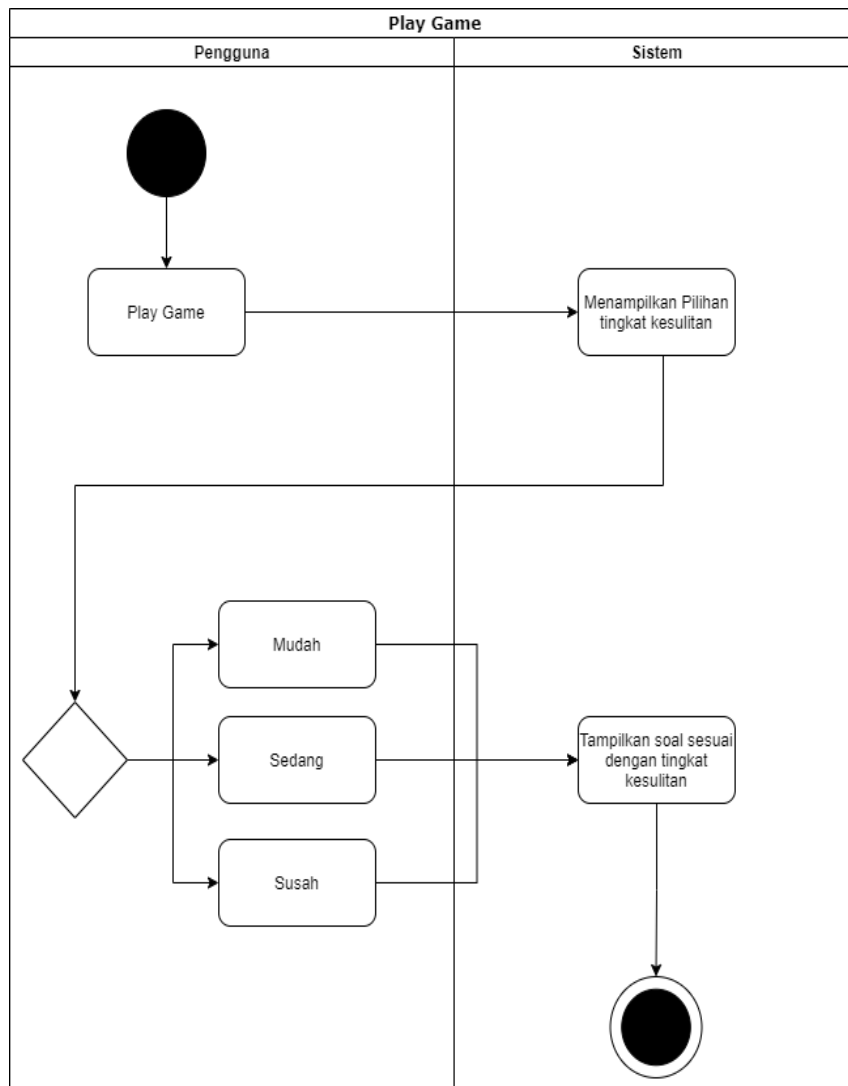


Gambar 3. 7 Activity Diagram Login

Gambar 3.7 menjelaskan alur login yang dilakukan oleh pengguna game quiz. Langkah pertama pengguna melakukan proses login kemudian sistem akan mengirimkan form login kepada pengguna lalu pengguna mengisi form login setelah itu pengguna melakukan proses masuk maka sistem akan verifikasi data yang di kirim user apakah benar atau salah, jika benar maka proses login berhasil.

c. Activity Diagram Play Game

Pada bagian ini akan dijelaskan activity diagram dari *play game* yang diimplementasikan pada Game Edukasi “Quiz” Bertema Sejarah Indonesia Berbasis Python yang dijelaskan pada Gambar 3.8.

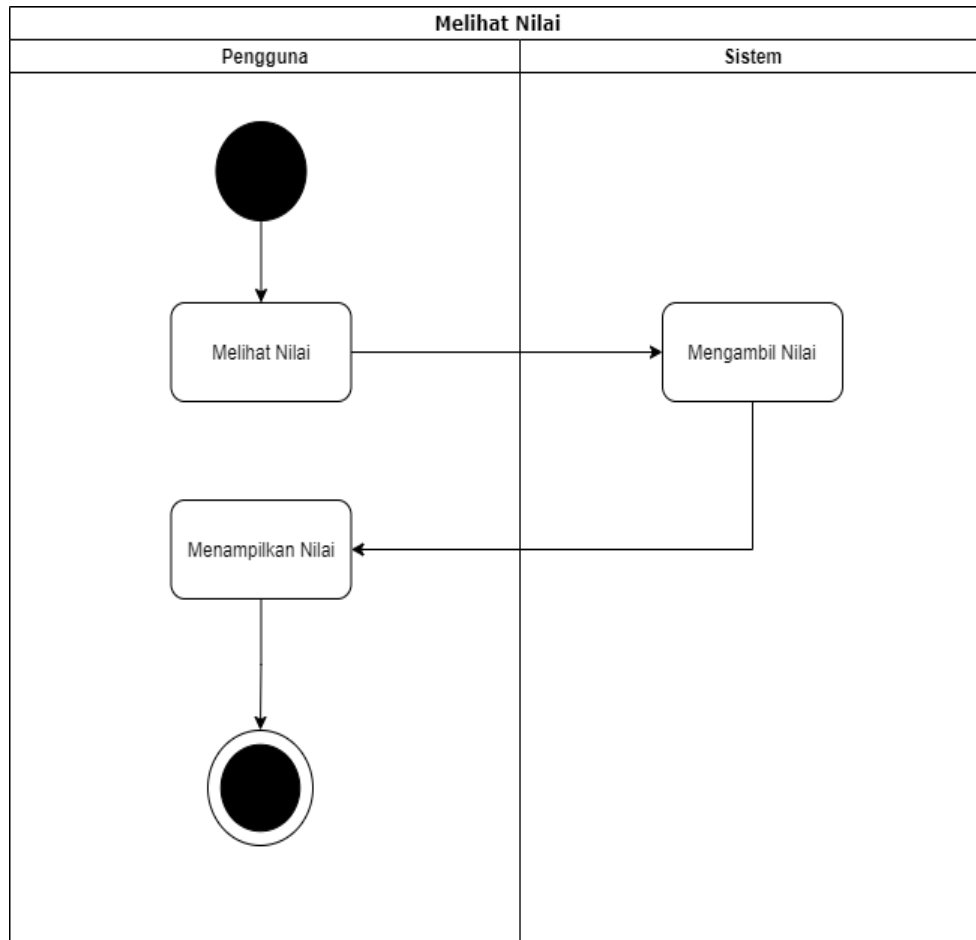


Gambar 3. 8 Activity Diagram Play Game

Gambar 3.8 menjelaskan alur play game yang dilakukan oleh pengguna game quiz. Langkah pertama pengguna melakukan proses play game kemudian sistem akan menampilkan sebuah pilihan tingkat kesulitan kemudian menampilkan soal sesuai dengan tingkat kesulitan.

d. *Activity Diagram Melihat Nilai*

Pada bagian ini akan dijelaskan activity diagram dari login yang diimplementasikan pada Game Edukasi “Quiz” Bertema Sejarah Indonesia Berbasis Python yang dijelaskan pada Gambar 3.9.

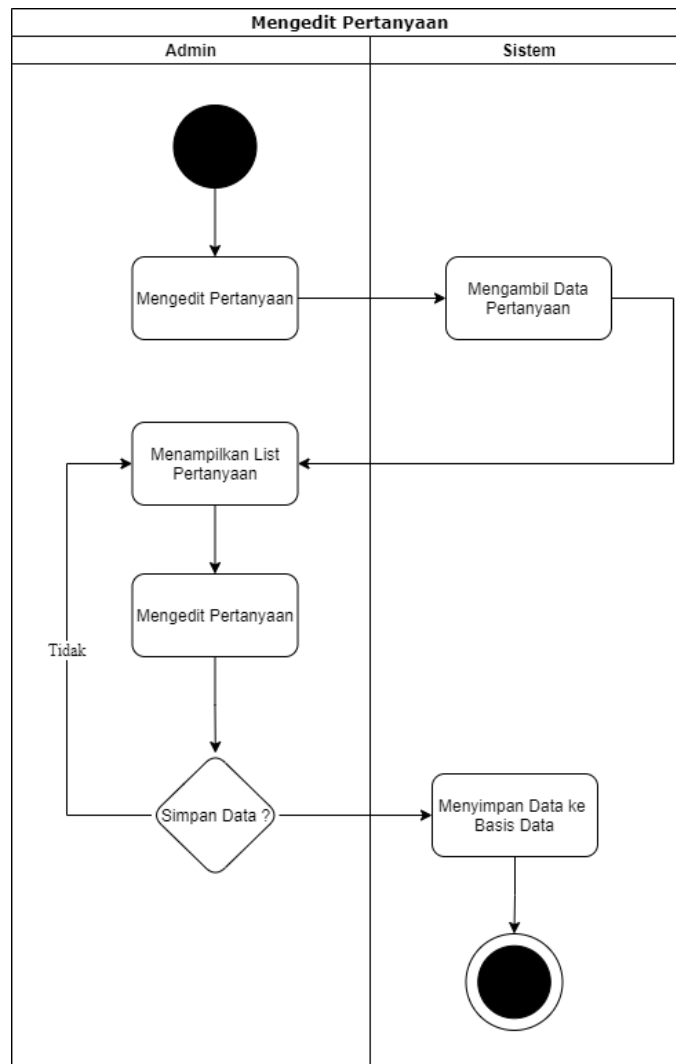


Gambar 3. 9 Activity Diagram Melihat Nilai

Gambar 3.9 menjelaskan alur melihat nilai yang dilakukan oleh pengguna (*player*) game quiz. Langkah pertama pengguna melakukan proses melihat nilai kemudian sistem akan mengambil data nilai yang tersimpan kemudian pengguna akan menerima tampilan nilai yang dimana data tersebut diambil dari basis data.

e. Activity Diagram Kelola Pertanyaan

Pada bagian ini akan dijelaskan activity diagram dari login yang diimplementasikan pada Game Edukasi “Quiz” Bertema Sejarah Indonesia Berbasis Python yang dijelaskan pada Gambar 3.10.



Gambar 3. 10 Activity Diagram Kelola Pertanyaan

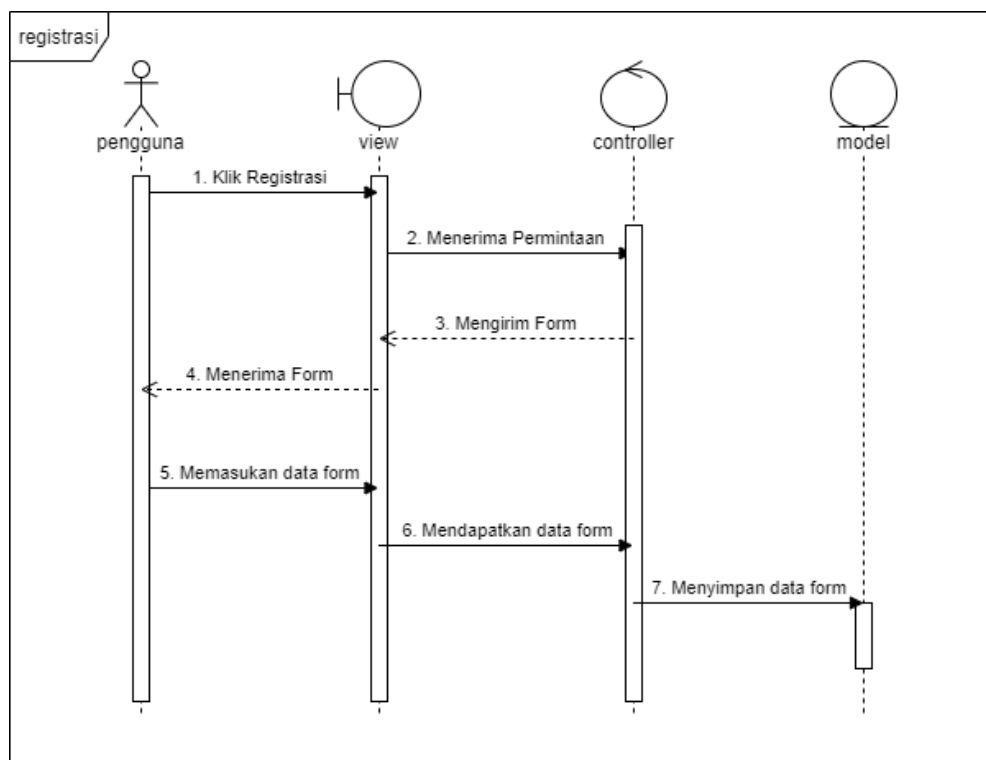
Gambar 3.10 menjelaskan alur Kelola Pertanyaan yang dapat dilakukan oleh admin game *quiz*. Langkah pertama adalah admin melakukan proses Kelola Pertanyaan kemudian sistem akan mengambilkan data pertanyaan dari basis data kemudian ditampilkan kepada admin, kemudian admin melakukan proses Kelola Pertanyaan setelah melakukan proses tersebut akan muncul *decision* simpan data, jika iya maka data akan disimpan oleh sistem ke basis data apabila tidak maka admin akan dikembalikan ke tampilan list pertanyaan.

3.3.4. Diagram Sequence

Berikut adalah diagram sequence dari Game Edukasi “Quiz” Bertema Sejarah Indonesia Berbasis Python:

1. Registrasi

Proses sebelum login adalah dilakukannya proses registrasi dengan maksud untuk membuat akun yang digunakan untuk login. Didalamnya terdapat proses isi form data pengguna dan ketika berhasil akan diteruskan ke halaman awal.



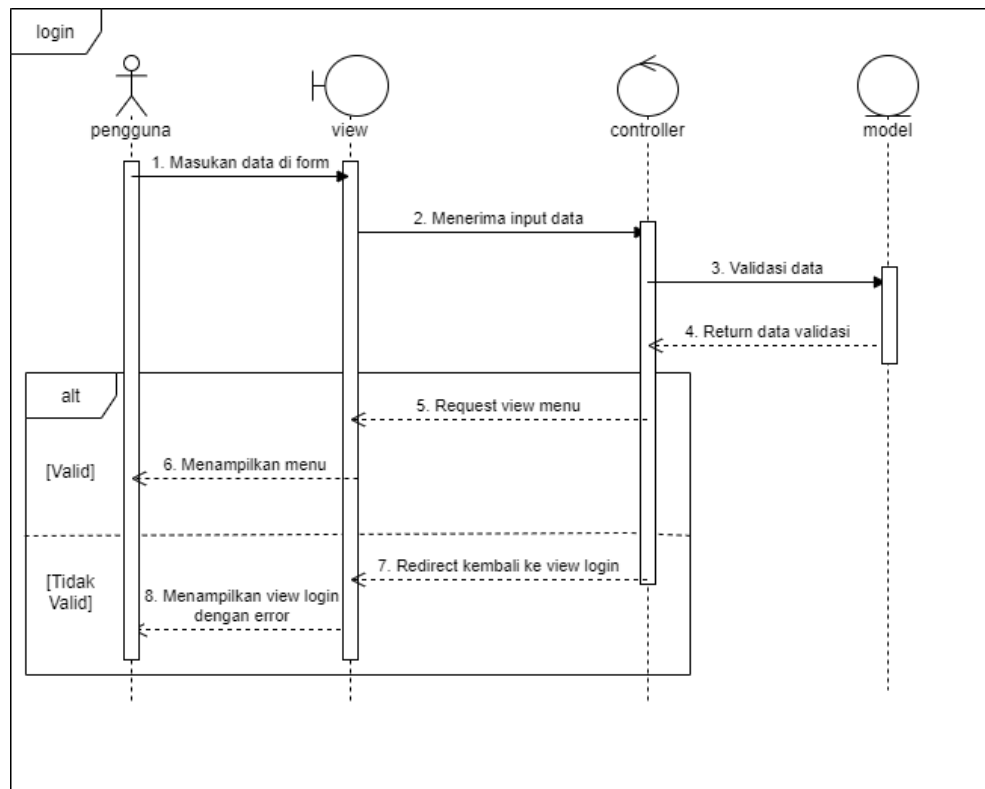
Gambar 3. 11 Sequence Diagram Registrasi

Pada Gambar 3.11 merupakan *sequence diagram* mengenai proses untuk registrasi ke sistem yang dapat dijelaskan sebagai berikut:

- 1) **Message 1 – 4**, pengguna melakukan proses dengan klik tombol registrasi kemudian permintaan akan diproses oleh *controller* kemudian *controller* mengirim *form* ke pengguna untuk diisi.
- 2) **Message 5 – 7**, pengguna melakukan input form registrasi kemudian data tersebut akan diproses oleh *controller* kemudian masuk ke dalam *model* untuk disimpan ke *database*.

2. Login

Proses setelah registrasi adalah login untuk dapat memasuki game. Didalamnya terdapat proses validasi akun dan ketika berhasil akan diteruskan ke halaman utama game.



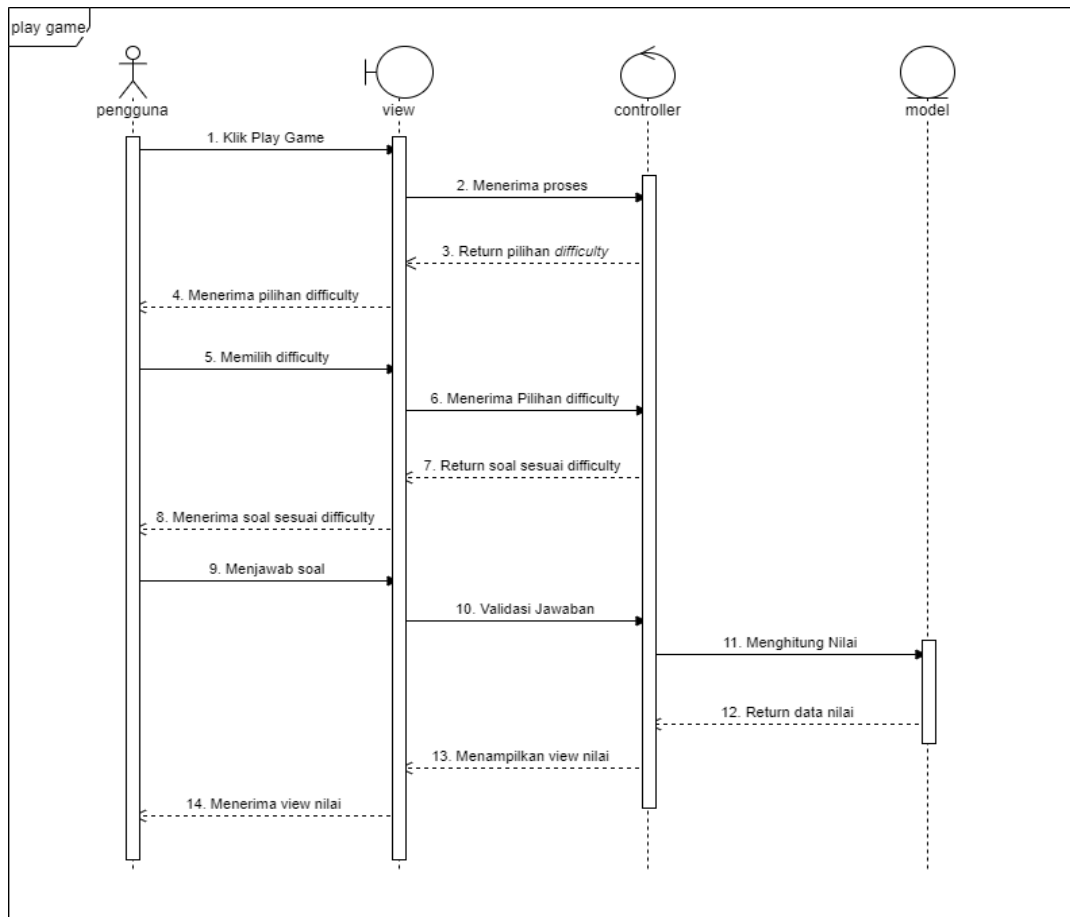
Gambar 3. 12 Sequence Diagram Login

Pada Gambar 3.12 merupakan *sequence diagram* mengenai proses untuk *login* ke sistem yang dapat dijelaskan sebagai berikut:

- 1) **Message 1 – 4**, pengguna memasukkan data di form yang ada pada halaman login dan kemudian data tersebut diproses di dalam *controller*. Data tersebut dicocokkan dengan data yang ada di dalam *database* melalui *model*.
- 2) **Message 5 – 6**, apabila data yang dimasukkan pengguna valid maka *controller* akan mengalihkan ke *menu*.
- 3) **Message 7 – 8**, apabila data yang dimasukkan tidak valid maka akan dikembalikan ke *view login*.

3. Play Game

Proses berikutnya adalah play game dimana proses ini pengguna dapat memilih tingkat kesulitan kemudian memainkan game.



Gambar 3. 13 Sequence Diagram Play Game

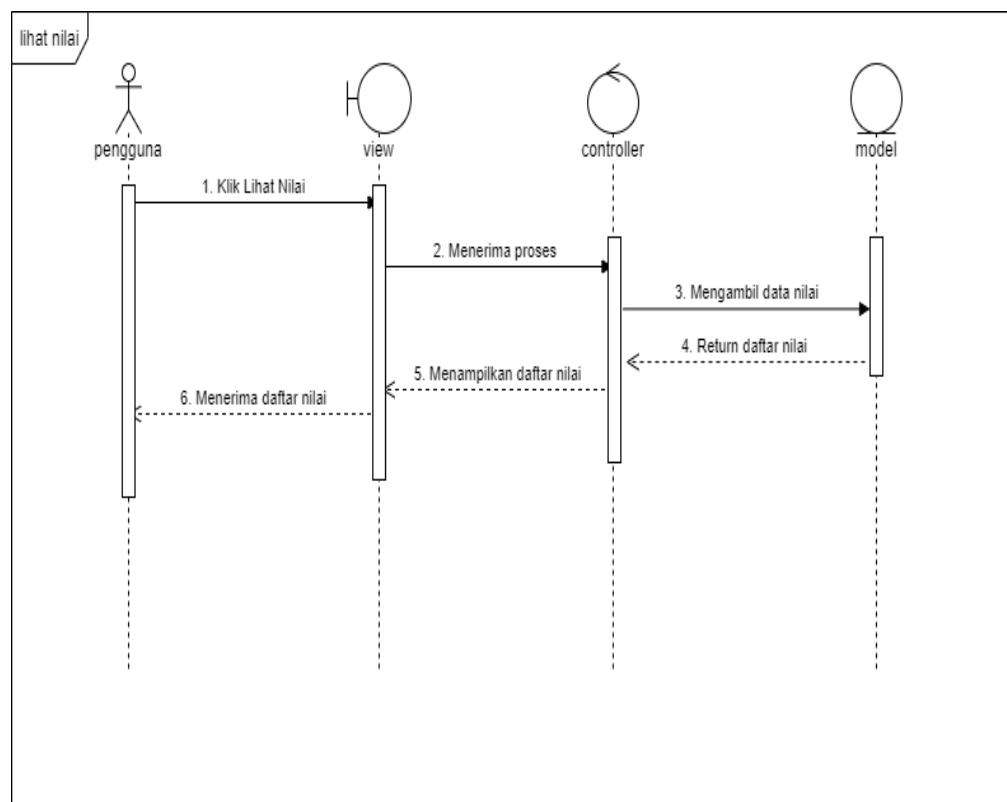
Pada Gambar 3.13 merupakan *sequence diagram* mengenai proses untuk *play game* yang dapat *dijelaskan* sebagai berikut:

- 1) **Message 1 – 4**, merupakan proses untuk memulai *play game* yang dimana proses dimulai oleh pengguna kemudian proses tersebut diolah oleh *controller* kemudian akan mengembalikan opsi kepada pengguna untuk memilih tingkat kesulitan (*difficulty*).
- 2) **Message 5 – 8**, adalah proses dimana pengguna akan memilih *difficulty* sehingga nanti pilihan akan diproses oleh *controller* untuk menampilkan soal sesuai dengan tingkat *difficulty*.

- 3) **Message 9 – 14**, proses yang dimana pengguna menjawab soal kemudian jawaban akan di validasi oleh *controller* kemudian hasil dari validasi akan berupa nilai yang dihitung pada *model database* kemudian nilai ini akan ditunjukkan kepada pengguna.

4. Melihat Nilai

Proses berikutnya adalah melihat nilai dimana proses ini pengguna dapat melihat daftar nilai.



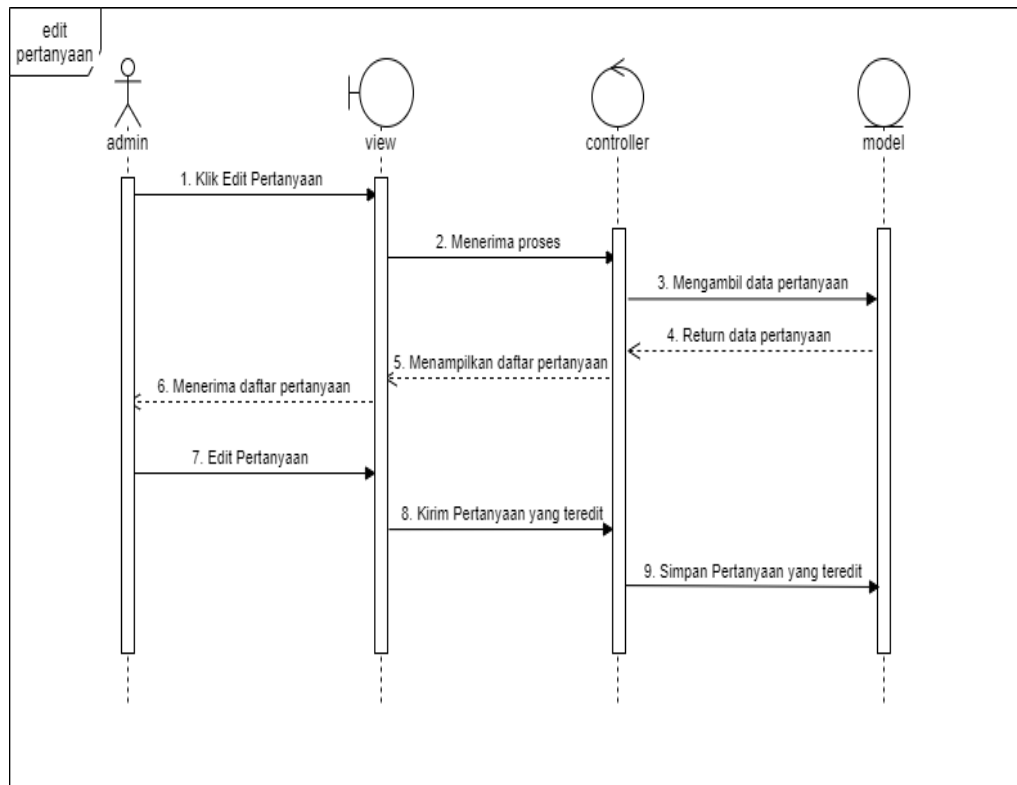
Gambar 3. 14 Sequence Diagram Melihat Nilai

Pada gambar 3.14 merupakan *sequence diagram* mengenai proses melihat nilai yang dapat dijelaskan sebagai berikut:

- 1) **Message 1 – 3**, merupakan proses untuk memulai melihat nilai yang dimana proses tersebut mengambilkan data dari *model*.
- 2) **Message 4 – 6**, merupakan proses dimana setelah data nilai diambil dari *model* maka akan menuju *controller* untuk menampilkannya pada *view* sehingga sampai kepada pengguna untuk menerima daftar nilai.

5. Kelola Pertanyaan

Proses berikutnya Kelola Pertanyaan adalah dimana admin dapat Kelola Pertanyaan.



Gambar 3. 15 Sequence Diagram Kelola Pertanyaan

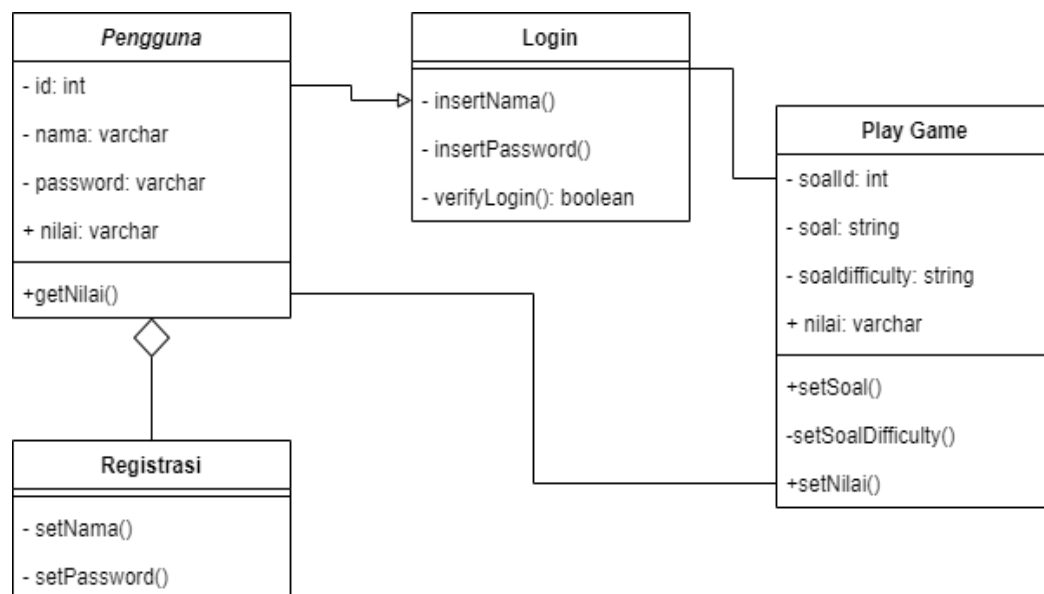
Pada gambar 3.15 merupakan *sequence diagram* dari Kelola Pertanyaan yang dapat dijelaskan sebagai berikut:

- 1) **Message 1 – 3**, adalah dimana proses admin melakukan proses edit pertanyaan kemudian *controller* menerima perintah proses edit pertanyaan kemudian mengambil data dari *model*.
- 2) **Message 4 – 6**, adalah proses dimana data yang diambil dari *model* dikirimkan ke *controller* setelah itu admin akan menerima daftar pertanyaan.
- 3) **Message 7 – 9**, adalah proses dimana admin Kelola Pertanyaan kemudian disimpan pada basis data di *model*.

3.3.5. Diagram Class

Class diagram atau diagram kelas adalah salah satu jenis diagram struktur pada UML yang menggambarkan dengan jelas struktur serta deskripsi *class*, atribut, metode, dan hubungan dari setiap objek. Ia bersifat statis dalam artian diagram kelas bukan menjelaskan apa yang terjadi jika kelas-kelasnya berhubungan, melainkan mnjelaskan hubungan apa yang terjadi.

Berikut ini merupakan implementasi *diagram class* dari Game Edukasi “Quiz” Bertema Sejarah Indonesia Berbasis Python yang akan diterapkan pada sistem aplikasi:



Gambar 3. 16 Diagram Class

Pada Gambar 3.16 merupakan *diagram class* Game Edukasi “Quiz” Bertema Sejarah Indonesia Berbasis Python. Pada *diagram class* ini terdapat 4 class yang dimana registrasi dan *login* berhubungan dengan *class* Pengguna kemudian setelah melakukan *login* maka akan lanjut ke *class* Play Game kemudian dari *class* Play Game menghasilkan nilai yang kemudian akan dimasukkan ke dalam *class* Pengguna.

3.3.6. Rancangan Database

Rancangan *database* adalah proses untuk menentukan isi dan pengaturan data yang dibutuhkan untuk mendukung berbagai rancangan sistem. Tujuan perancangan *database* untuk memenuhi informasi yang berisikan kebutuhan-kebutuhan *user* secara khusus dan aplikasi-aplikasinya sehingga data data tersebut dapat saling berelasi. Pada pengembangan aplikasi ini digunakan *database* menggunakan *database sqlite* bernama *database.db* yang akan ditempatkan pada satu direktori dengan aplikasi. Berikut ini adalah rancangan tabel dan relasi dari database

3.3.6.1. Rancangan Tabel

Dalam sistem ini terdapat tabel dalam database yaitu tabel user, tabel quiz, tabel score, dan tabel admin. Adapun rancangan antar tabel pada masing-masing tabel akan dijabarkan dalam tabel berikut:

a. Tabel user

Tabel user adalah tabel yang berisi data akun dari user. Struktur tabel user dijabarkan pada Tabel 3.7 dibawah ini.

Tabel 3. 7 Tabel user

No.	Kolom	Tipe Data	Ukuran Field	Keterangan
1.	Id	Integer Primary Key Auto Increment	20	Untuk menyimpan data <i>id</i> .
2.	Username	Text Unique	50	Menyimpan data <i>username</i> .
3.	Password	Text	50	Menyimpan data <i>password</i> dari <i>username</i> .
4.	Scr	Integer	20	Menyimpan data <i>score</i> per- <i>username</i> .
5.	Datetime	Integer	20	Menyimpan tanggal pencapaian <i>score</i> per- <i>username</i>

b. Tabel quiz

Tabel *quiz* adalah tabel yang berisi data quiz pertanyaan dari game *quiz*. Struktur tabel user dijabarkan pada Tabel 3.8 dibawah ini.

Tabel 3. 8 Tabel quiz

No.	Kolom	Tipe Data	Ukuran Field	Keterangan
1.	quiz_id	Integer	20	Menyimpan <i>id</i> dari pertanyaan <i>quiz</i>
2.	Quiz	Text	150	Menyimpan data pertanyaan.
3.	answer_tohelp	Text	20	Menyimpan data jawaban pertanyaan.
4.	qz_photo	Image	1000	Menyimpan data foto dari pertanyaan.

c. Tabel score

Tabel score adalah tabel yang berisi data score dari game quiz. Struktur tabel user dijabarkan pada Tabel 3.9 dibawah ini.

Tabel 3. 9 Tabel score

No.	Kolom	Tipe Data	Ukuran Field	Keterangan
1.	Id	Text	20	Menyimpan <i>id</i> dari <i>score</i> berdasarkan tingkat kesulitannya.
2.	Score	Text	20	Menyimpan <i>score</i> dari <i>game quiz</i> .

d. Tabel admin

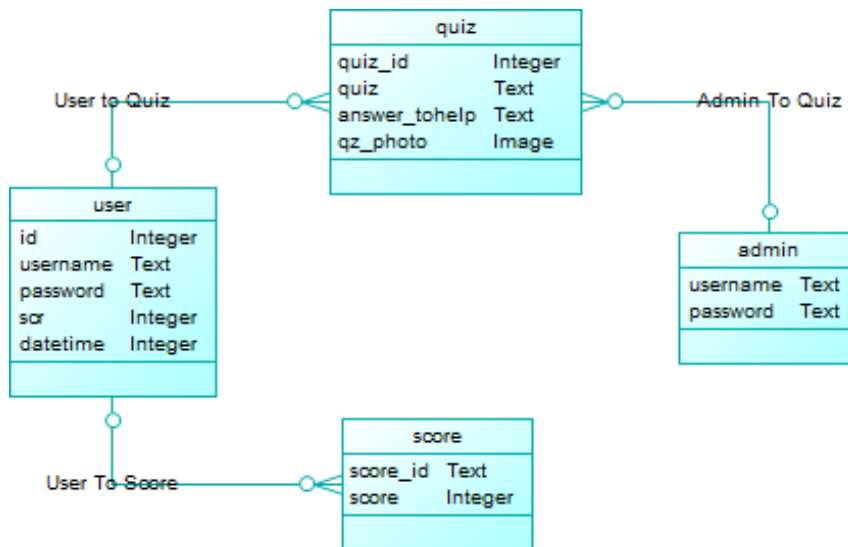
Tabel admin adalah tabel yang berisi data akun admin dari game quiz. Struktur tabel user dijabarkan pada Tabel 3.10 dibawah ini.

Tabel 3. 10 Tabel admin

No.	Kolom	Tipe Data	Field Size	Keterangan
1.	Username	Text	20	Menyimpan data <i>username</i> dari admin.
2.	Password	Text	20	Menyimpan data <i>password</i> dari admin.

3.3.6.2. Relasi Antar Tabel

Dari rancangan tabel yang telah dijelaskan dalam bentuk tabel tersebut, maka terbentuk relasi antar tabel masing-masing tabel yang terdapat di dalam basisdata yaitu pada Gambar 3.17 sebagai berikut:



Gambar 3. 17 Relasi Antar Tabel

Pada Gambar 3.17 merupakan rancangan relasi antar tabel Game Edukasi “Quiz” Bertema Sejarah Indonesia Berbasis Python. Pada rancangan tersebut terdapat 4 (empat) tabel. Tabel pertama adalah tabel pengguna (*user*) yang akan digunakan untuk menampung data pengguna dari game ini. Kemudian terdapat tabel *quiz* yang akan digunakan untuk menampung perihal yang berhubungan dengan soal kuis yang akan dikerjakan. Kemudian tabel *score* yang diperuntukkan menyimpan nilai dari game Quiz. Yang terakhir adalah

3.3.7. Arsitektur Sistem

Dalam pengembangan *Game* Edukasi “Quiz” Bertema Sejarah Indonesia Berbasis Python ini diperlukan adanya arsitektur sistem yang jelas, sehingga sistem dapat berjalan secara runtut dan benar. Didalam arsitektur sistem, pengguna dapat melakukan registrasi, *login*, meminta play / menampilkan pertanyaan, dan menampilkan nilai yang data tersebut diambil dari *database* sesuai dengan permintaan pengguna. Untuk jalannya arsitektur sistem, digambarkan pada diagram sebagai berikut. Diagram adalah bagian dari perancangan sistem yang menggambarkan bentuk komponen fisik dari sistem yang akan dibuat. Arsitektur pada Game Edukasi “Quiz” Bertema Sejarah Indonesia Berbasis Python dapat ditunjukkan pada Gambar 3.18 sebagai berikut :



Gambar 3. 18 Arsitektur Sistem

Keterangan dari *diagram* arsitektur sistem akan dijelaskan lebih detail, dari Gambar 3.18 akan dijelaskan pada Tabel 3.11 Keterangan Arsitektur Diagram berikut:

Tabel 3. 11 Keterangan Arsitektur Sistem

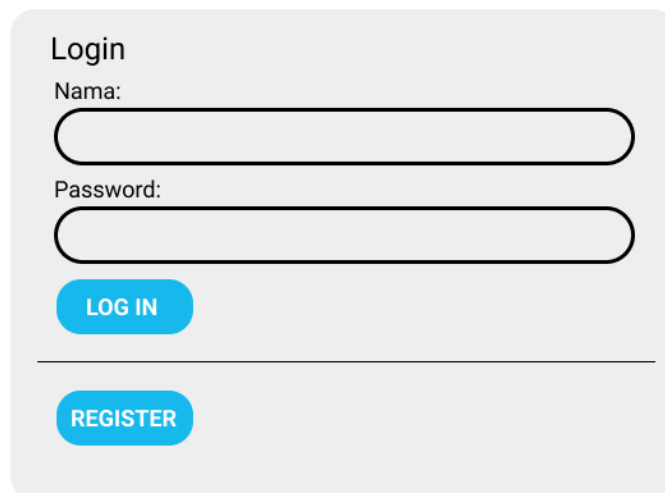
Simbol	Keterangan
A1	<ol style="list-style-type: none"> 1. Melakukan registrasi 2. Melakukan login 3. Meminta play game (menerima soal) 4. Melihat nilai
A2	<ol style="list-style-type: none"> 1. Menampilkan form registrasi 2. Menampilkan form login 3. Menampilkan soal play game 4. Menampilkan nilai
S1	<ol style="list-style-type: none"> 1. Data pengguna (data registrasi & login) 2. Data soal 3. Data nilai
S2	<ol style="list-style-type: none"> 1. Record data pengguna 2. Record data soal 3. Record data nilai

3.3.8. Rancangan Antarmuka

Antarmuka aplikasi merupakan rancangan antarmuka yang akan digunakan sebelum diimplementasikan pada aplikasi. Perancangan ini berfungsi sebagai acuan dalam pembuatan desain antarmuka aplikasi agar tidak menyimpang dari tujuan awal, sesuai dengan konsep kebutuhan serta memudahkan dalam pembuatan desain *interface* dalam sistem. Berikut merupakan rancangan antarmuka dari Game Edukasi “Quiz” Bertema Sejarah Indonesia Berbasis Python.

a. Desain Antarmuka *Login*

Rancangan antarmuka halaman *login* dapat ditunjukkan pada Gambar 3.19 sebagai berikut:

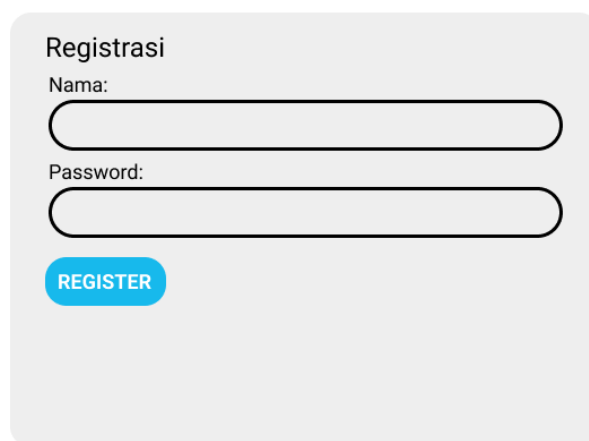
A login form with a light gray background. At the top, the word "Login" is written in a bold, black font. Below it, the label "Nama:" is followed by a white rounded rectangular input field. Underneath, the label "Password:" is followed by another white rounded rectangular input field. Below the password field is a blue rounded rectangular button with the text "LOG IN" in white. A horizontal line separates this section from the bottom section, which contains a blue rounded rectangular button with the text "REGISTER" in white.

Gambar 3. 19 Rancangan Antarmuka Login

Pada Gambar 3.19 menunjukkan mengenai desain rancangan antarmuka *login*. Di dalam halaman *login* tersebut terdapat *form* untuk *login* yang terdiri dari nama, password dan tombol untuk login. Dan terdapat tombol registrasi untuk melakukan registrasi pengguna baru.

b. Desain Antarmuka Registrasi

Rancangan antarmuka halaman registrasi dapat ditunjukkan pada Gambar 3.20 sebagai berikut:

A registration form with a light gray background. At the top, the word "Registrasi" is written in a bold, black font. Below it, the label "Nama:" is followed by a white rounded rectangular input field. Underneath, the label "Password:" is followed by another white rounded rectangular input field. Below the password field is a blue rounded rectangular button with the text "REGISTER" in white.

Gambar 3. 20 Rancangan Antarmuka Registrasi

Pada Gambar 3.20 menunjukan mengenai desain rancangan antarmuka registrasi. Di dalam halaman registrasi tersebut terdapat *form* untuk registrasi yang terdiri dari nama, *password* dan tombol registrasi untuk memproses registrasi.

c. Desain Antarmuka *Play Game*

Rancangan antarmuka *play game* dapat ditunjukkan pada Gambar 3.21 sebagai berikut:

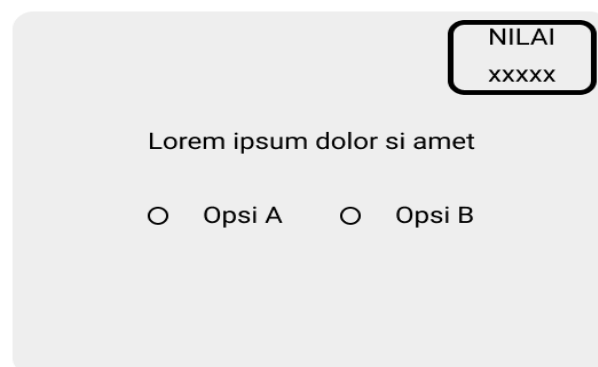


Gambar 3. 21 Rancangan Antarmuka Play Game

Pada Gambar 3.21 menunjukkan mengenai desain rancangan antarmuka *menu awal dari play game* yang dimana pengguna dapat memilih tingkat kesulitan kemudian terdapat tombol *Play Game* untuk lanjut ke tampilan game.

d. Desain Antarmuka Game

Rancangan antarmuka *game* dapat ditunjukkan pada Gambar 3.22 sebagai berikut :



Gambar 3. 22 Rancangan Antarmuka Game

Pada Gambar 3.22 menunjukkan desain rancangan antarmuka *game* yang dimana pengguna menerima soal dan kemudian menjawab dengan menggunakan *button* yang tersedia jika jawaban benar maka nilai pengguna akan bertambah.

e. Desain Antarmuka Akhir Game

Rancangan antarmuka akhir *game* dapat ditunjukkan pada Gambar 3.23 sebagai berikut:



Gambar 3. 23 Rancangan Antarmuka Akhir Game

Pada Gambar 3.23 menunjukkan mengenai desain akhir *game* yang menunjukkan nilai total yang diterima oleh pengguna dan memberikan tombol main kembali untuk pengguna sehingga dapat kembali ke menu *Play Game*.

3.4. Rancangan Pengujian

Rancangan pengujian merupakan sebuah rancangan tahapan yang akan dilakukan untuk menguji fungsional dari sistem yang dibangun. Pengujian sistem pada Implementasi Metode Forward Chaining pada Game Edukasi “Quiz” Bertema Sejarah Indonesia Berbasis Python ditunjukkan pada Tabel 3.12 Skenario Pengujian Sistem.

Tabel 3. 12 Tabel Rancangan Pengujian

No	Subjek Pengujian	Objek Pengujian	Skenario Pengujian	Hasil Yang Diharapkan
1.	Pengguna	Registrasi	Melakukan proses registrasi pada sistem.	Sistem berhasil mengeksekusi perintah menambah pengguna.
2.	Pengguna	Login	Melakukan proses login pada sistem.	Sistem berhasil mengeksekusi perintah dan memvalidasi pengguna untuk masuk kedalam game.
3.	Pengguna	Play Game	Melakukan proses memilih tingkat kesulitan soal dan memainkan game.	Sistem berhasil menampilkan soal yang sesuai dengan tingkat kesulitan yang dipilih oleh pengguna.
4.	Pengguna	Menampilkan nilai pengguna	Menampilkan nilai total dari hasil pengerjaan pengguna.	Sistem menunjukkan total nilai yang didapatkan oleh pengguna.
5.	Admin	Kelola Pertanyaan	Menampilkan pertanyaan dan dapat melakukan proses <i>Update</i> pertanyaan.	Sistem dapat melakukan <i>update</i> pertanyaan dan menampilkan seluruh pertanyaan.

BAB IV IMPLEMENTASI

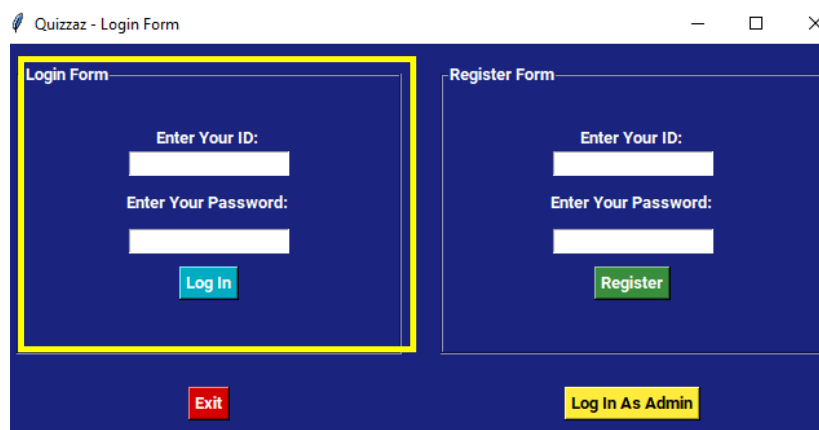
Kegiatan setelah melakukan perancangan sistem adalah kegiatan implementasi. Implementasi dibuat berdasarkan rancangan aplikasi yang telah dibuat, kemudian direalisasikan dalam bentuk aplikasi untuk mengetahui aplikasi yang diusulkan relevan atau tidak.

4.1. Implementasi Proses

Implementasi *user interface* atau antarmuka sistem pada *Game “Quiz”* bertema Sejarah Indonesia. Dapat diakses oleh pengguna, yaitu admin dan pemain (*user*).

4.1.1. Tampilan *Login Pemain (User)*

Tampilan *login* adalah halaman utama ketika pengguna mengakses aplikasi *game Quiz* dengan memasukkan *username* dan *password*. Setelah *login* pemain (*user*) akan diarahkan ke tampilan *user menu*. Tampilan *login* terdapat pada gambar 4.1.



Gambar 4. 1 Tampilan *Logins*

User harus memasukkan *username* dan *password*, dan akan diproses oleh *database* jika data benar maka system akan menampilkan halaman utama *user*, jika data salah maka sistem meminta untuk memasukkan data yang benar.

Dari implementasi tampilan pada Gambar 4.1 Tampilan *Login*, berikut adalah potongan *source code* untuk menampilkan tampilan *login*.

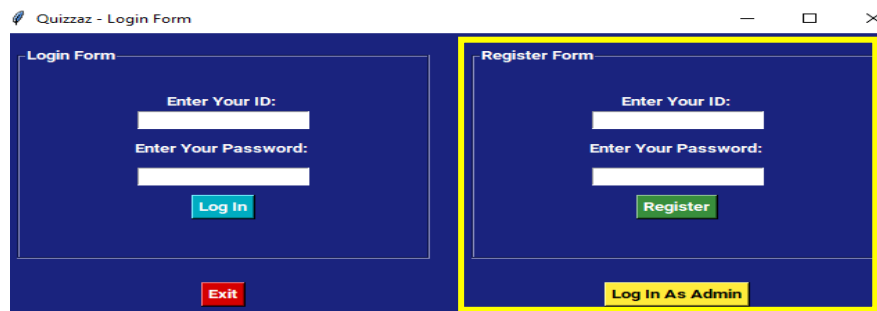
```
1. def LoginForm():
2.     window = Tk()
3.     window.title('Quizzaz - Login Form')
4.     window.configure(bg="#1a237e")
5.     global login_id
6.     global login_pass
7.     global regis_id
8.     global regis_pass
9.     login_id = tkinter.StringVar()
10.    login_pass = tkinter.StringVar()
11.    regis_id = tkinter.StringVar()
12.    regis_pass = tkinter.StringVar()
```

Berikut ini akan dijelaskan *source code* berdasarkan *line* yang ditampilkan diatas:

- 1) **Line 1**, merupakan *function* dari login form yang berfungsi sebagai tempat untuk kumpulan perintah-perintah dari *login* sehingga kumpulan perintah ini dapat dipanggil.
- 2) **Line 2 – 4**, merupakan perintah dari tkinter untuk membuat *window* dari *interface* yang diterapkan pada *game*.
- 3) **Line 5 – 8**, merupakan perintah *global* supaya data dari perintah yang di *global* dapat diambil oleh *function* lain.
- 4) **Line 9 – 12**, merupakan perintah mengambil data *input* pengguna sebagai string.

4.1.2. Tampilan Registrasi (*User*)

Tampilan registrasi digunakan untuk menambah / registrasi *user* baru supaya *user* tersebut dapat melakukan *login* untuk mengakses konten di dalam aplikasi *game* “Quiz”. Berikut ini adalah tampilan dari *menu* registrasi pada Gambar 4.4.



Gambar 4. 2 Tampilan Menu Registrasi

Pertama *user* memasukkan *input username* dan *password* kemudian *user* menekan tombol *Register* untuk melakukan proses registrasi. Setelah melakukan proses registrasi maka *user* akan menerima *pop-up* bahwa registrasi berhasil, sehingga dapat melanjutkan ke proses *login*. Aktivitas dimulai dengan *user* membuka aplikasi dan melakukan *input username* dan *password* kemudian menekan tombol registrasi, sistem membalas dengan memberi tampilan *pop-up* kepada *user* yang menyatakan bahwa berhasil registrasi.

Dari implementasi tampilan pada Gambar 4.2 Tampilan Menu Registrasi, berikut adalah potongan *source code* untuk menampilkan menu registrasi.

```

1. regis_frame = LabelFrame(window, text="Register Form", padx
2. regis_frame.grid(column=1, row=0, padx=15, pady=5, sticky=
3. text_id_regis = Label(regis_frame, text="Enter Your ID: "
4. text_id_regis.pack()
5. global regis_id_input
6. regis_id_input = Entry(regis_frame, textvariable=regis_id)
7. regis_id_input.pack()
8. text_pass_regis = Label(regis_frame, text="Enter Your Passw
9. text_pass_regis.pack(pady=10)
10. global regis_pass_input
11. regis_pass_input = Entry(regis_frame, show="*", textvariab
12. regis_pass_input.pack()
13. global regis_button
14. regis_button = Button(regis_frame, text="Register", command
15. regis_button.pack(pady=10)

```

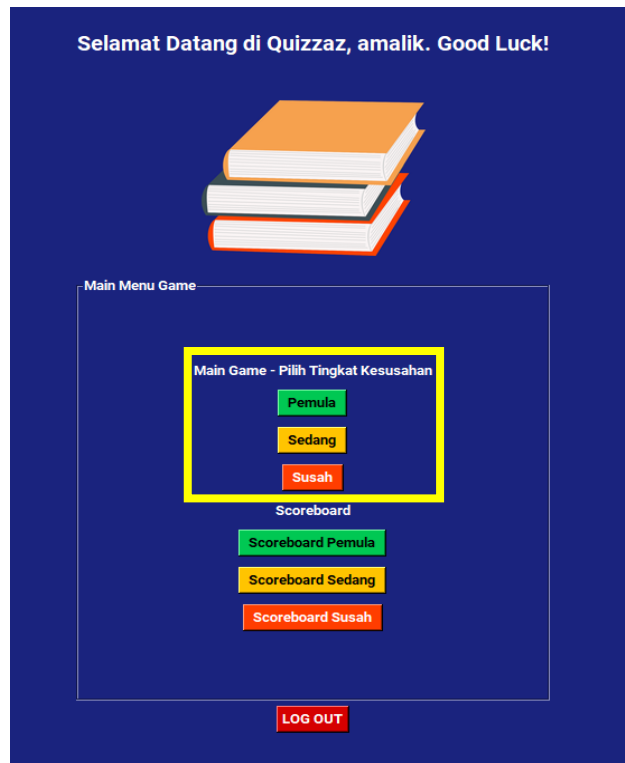
Berikut ini akan dijelaskan *source code* berdasarkan *line* yang ditampilkan diatas:

- 1) **Line 1 – 4**, merupakan perintah untuk menampilkan *frame* dan *text* pada *frame* registrasi.
- 2) **Line 5 – 9**, merupakan perintah untuk menampilkan *input box* pengguna sehingga pengguna dapat memasukkan data.
- 3) **Line 10 – 15**, merupakan perintah untuk menampilkan *button* untuk melakukan proses registrasi.

4.1.3. Tampilan *Play Game*

Tampilan *play game* adalah tampilan utama *user* ketika berhasil melakukan *login*. Disini *user* dapat memilih tingkat kesulitan yang *user* pilih sesuai dengan keinginan *user* dengan catatan semakin tinggi tingkat kesulitan yang dihadapi *user*

akan diberikan *reward* lebih tinggi dari pada tingkat kesulitan yang paling rendah. Berikut ini adalah tampilan *play game* yang akan ditunjukkan pada Gambar 4.3



Gambar 4. 3 Tampilan Menu Memilih Tingkat Kesulitan

Dari implementasi tampilan pada Gambar 4.3 Tampilan Menu Memilih Tingkat Kesulitan, berikut adalah potongan *source code* untuk menu memilih tingkat kesulitan:

```

1. txt_difficulty = Label(difficulty_frame, text="Main Game -
2. global btn_easy
3. global btn_medium
4. global btn_hard
5. btn_easy = Button(difficulty_frame, text="Pemula", comman
6. btn_medium = Button(difficulty_frame, text="Sedang", comma
7. btn_hard = Button(difficulty_frame, text="Susah", command=la
8. txt_scoreboard = Label(difficulty_frame, text="Scoreboard",
9. global btn_scoreboard_ez
10. global btn_scoreboard_md
11. global btn_scoreboard_hrd
12. btn_scoreboard_ez = Button(difficulty_frame, text="Scoreboa
13. btn_scoreboard_md = Button(difficulty_frame, text="Scoreboa
14. btn_scoreboard_hrd = Button(difficulty_frame, text="Scoreoa

```

Berikut ini akan dijelaskan *source code* berdasarkan *line* yang ditampilkan diatas:

- 1) **Line 1**, untuk menampilkan *frame* dari *difficulty* yang menampung *button* untuk memilih *difficulty*.
- 2) **Line 2 – 7**, perintah untuk menampilkan *button difficulty* kedalam *frame difficulty*.
- 3) **Line 8**, perintah untuk menampilkan *frame scoreboard* untuk menampung pilihan *scoreboard* berdasarkan tingkat kesulitan (*difficulty*).
- 4) **Line 9 – 14**, pada *line* ini digunakan untuk menampilkan perintah *button* untuk memilih *scoreboard* yang ditampilkan dalam *frame scoreboard*.

Setelah *user* selesai memilih tingkat kesulitan maka *user* akan menerima tampilan dari *game quiz* yang ditunjukkan pada Gambar 4.4.



Gambar 4. 4 Tampilan Menu Game

Setelah pemain (*user*) melakukan *login*, pemain (*user*) akan melihat tampilan dari memilih tingkat kesulitan, ketika pemain (*user*) sudah memilih tingkat kesulitan maka sistem akan mengambil *quiz* yang sesuai dengan tingkat kesulitan yang dipilih oleh pemain (*user*) lalu sistem menampilkan *quiz* tersebut. Dari implementasi tampilan pada Gambar 4.4 Tampilan Menu Game, berikut adalah potongan *source code* untuk menampilkan menu *game*.

```

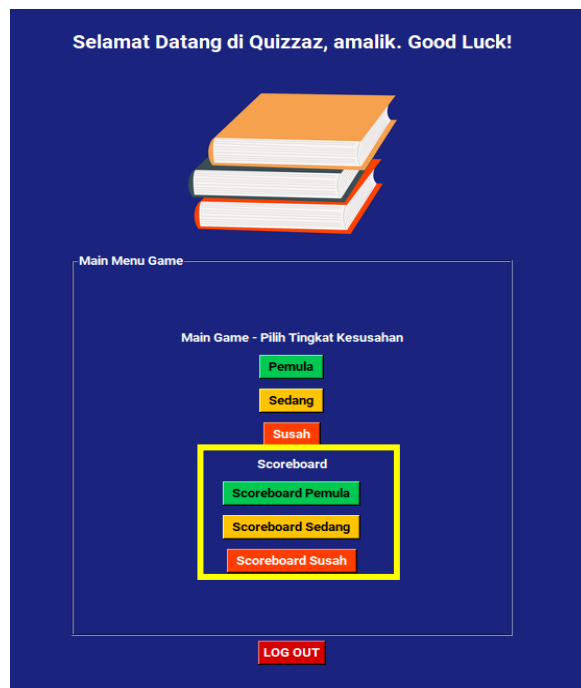
1. def qz_hrd_starter():
2.     quiz01 = Toplevel()
3.     quiz01.geometry("860x480")
4.     quiz01.title("Quizzaz - No. 1")
5.     quiz01.configure(bg="#1a237e")
6.     quiz01.wm_attributes("-fullscreen", 3)
7.     conn = sqlite3.connect('database.db')
8.     score_cursor = conn.cursor()
9.     score_frame = LabelFrame(quiz01, text="
10.     score_frame.pack(padx=50)
11.     txt_score = Label(score_frame, text="S
12.     score_cursor.execute("SELECT score FRO
13.     score = score_cursor.fetchone()
```

Berikut ini akan dijelaskan *source code* berdasarkan *line* yang ditampilkan diatas:

- 1) **Line 1**, merupakan *function* dari *game* yang menandakan level utama dari tingkat kesulitan yang dipilih.
- 2) **Line 2 – 6**, merupakan perintah untuk menampilkan *window interface* dari game.
- 3) **Line 7**, merupakan koneksi untuk terhubung ke *database*.
- 4) **Line 8**, merupakan *cursor* untuk menjalankan perintah *query* database.
- 5) **Line 9 – 11**, menampilkan frame dari *score*.
- 6) **Line 12 – 13**, perintah *execute* cursor dengan perintah *query* dan mengambil data dari database dengan *fetchone*.

4.1.4. Tampilan Melihat Nilai

Tampilan melihat nilai adalah tampilan nilai yang pemain (*user*) terima setelah menyelesaikan *quiz game* yang dimana pemain (*user*) akan diperlihatkan nilai (skor) yang mereka dapatkan. Berikut ini adalah tampilan dari nilai yang di tampilkan pada Gambar 4.5.



Gambar 4. 5 Tampilan Melihat Nilai

Dari implementasi tampilan pada Gambar 4.5 Tampilan Melihat Nilai, berikut adalah potongan *source code* untuk menampilkan nilai:

```

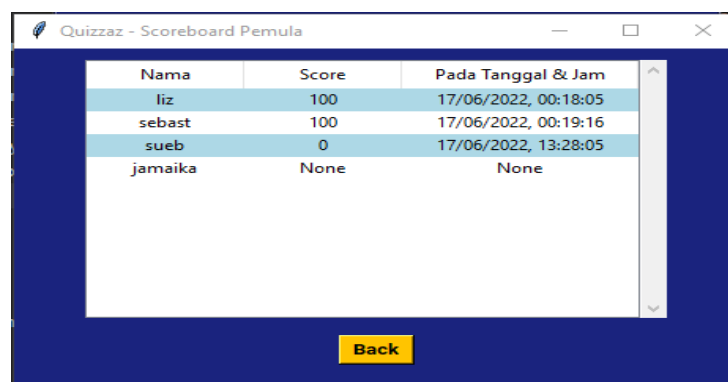
1. txt_difficulty = Label(difficulty_frame, text="Main Game -
2. global btn_easy
3. global btn_medium
4. global btn_hard
5. btn_easy = Button(difficulty_frame, text="Pemula", comman
6. btn_medium = Button(difficulty_frame, text="Sedang", comma
7. btn_hard = Button(difficulty_frame, text="Susah", command=la
8. txt_scoreboard = Label(difficulty_frame, text="Scoreboard",
9. global btn_scoreboard_ez
10. global btn_scoreboard_md
11. global btn_scoreboard_hrd
12. btn_scoreboard_ez = Button(difficulty_frame, text="Scoreboa
13. btn_scoreboard_md = Button(difficulty_frame, text="Scoreboa
14. btn_scoreboard_hrd = Button(difficulty frame, text="Scoreoa

```

Berikut ini akan dijelaskan *source code* berdasarkan *line* yang ditampilkan diatas:

- 5) **Line 1**, untuk menampilkan *frame* dari *difficulty* yang menampung *button* untuk memilih *difficulty*.
- 6) **Line 2 – 7**, perintah untuk menampilkan *button difficulty* kedalam *frame difficulty*.
- 7) **Line 8**, perintah untuk menampilkan *frame scoreboard* untuk menampung pilihan *scoreboard* berdasarkan tingkat kesulitan (*difficulty*).
- 8) **Line 9 – 14**, pada *line* ini digunakan untuk menampilkan perintah *button* untuk memilih *scoreboard* yang ditampung dalam *frame scoreboard*.

User setelah memilih *scoreboard* dari tingkat kesulitan masing-masing maka akan ditampilkan papan nilai yang berisikan nama, skor, dan tanggal & jam kapan di raihnya skor tersebut oleh *user* yang ditunjukkan pada Gambar 4.6.



Nama	Score	Pada Tanggal & Jam
liz	100	17/06/2022, 00:18:05
sebast	100	17/06/2022, 00:19:16
sueb	0	17/06/2022, 13:28:05
jamaika	None	None

Back

Gambar 4. 6 Tampilan Scoreboard

Pemain (*user*) akan melihat nilai setelah selesai menjawab pertanyaan *quiz* dengan benar maupun salah, setelah itu sistem membalas dengan mengambil nilai yang kemudian data nilai ini ditampilkan kepada pemain (*user*). Dari tampilan Gambar 4.6 Tampilan *Scoreboard*, berikut adalah potongan *source code* untuk menampilkan *scoreboard*.

```
1. def scoreboard_hrd():
2.     scrbrd_window = Tk()
3.     scrbrd_window.title("Quizzaz - Score
4.     scrbrd_window.configure(bg="#1a237e")
5.     scrbrd_window.wm_attributes("-fullsc
6.     conn = sqlite3.connect('database.db')
7.     c = conn.cursor()
8.     c.execute("SELECT username, scr_hrd,
```

Berikut ini akan dijelaskan *source code* berdasarkan *line* yang ditampilkan diatas:

- 1) **Line 1**, merupakan *function* dari lihat nilai yang menyimpan perintah perintah untuk menampilkan nilai per pengguna
- 2) **Line 2 – 5**, merupakan perintah untuk membuat *window*
- 3) **Line 6**, merupakan perintah koneksi ke database
- 4) **Line 7 – 8**, merupakan perintah *cursor* database untuk mengambil data dari *database*.

4.1.5. Tampilan Kelola Pertanyaan (*Admin*)

Tampilan mengedit (Kelola) pertanyaan digunakan *admin* untuk mengganti atau mengedit *quiz* atau pertanyaan yang akan tampil pada *game quiz*. Demikian ini adalah implementasi tampilan menu admin kelola *quiz* yang akan ditunjukkan pada Gambar 4.7.



Gambar 4. 7 Tampilan Menu Admin Kelola *Quiz*

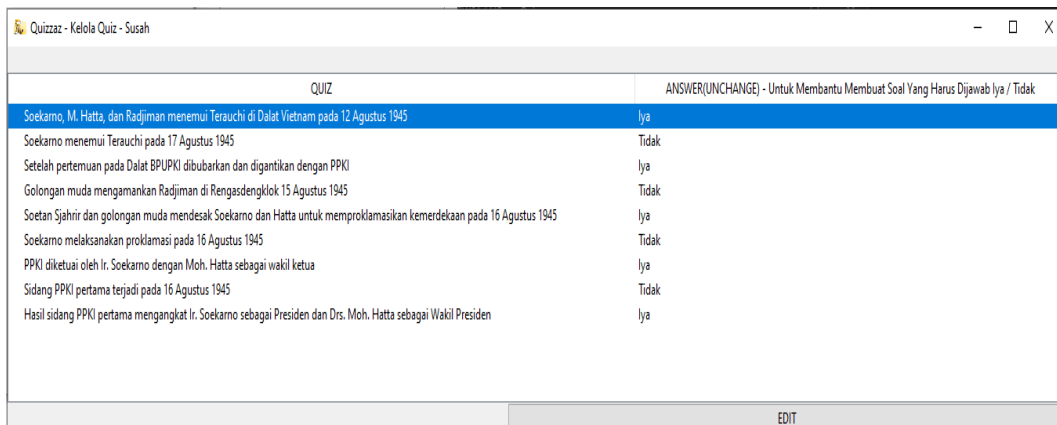
Dari implementasi tampilan pada Gambar 4.7 Tampilan Menu Admin Kelola Quiz, berikut adalah potongan source code untuk menampilkan halaman admin:

```
1. admin_mainmenu = Toplevel()
2. admin_mainmenu.title("Quizzaz - Admin Menu")
3. admin_mainmenu.configure(bg="#1a237e")
4. admin_mainmenu.wm_attributes("-topmost", 2)
5. menu_editquiz_frame = LabelFrame(admin_mainmenu, text="Admin
6. menu_editquiz_frame.grid(column=0, row=0, pady=15, padx=5)
7. crud_quizez_btn = Button(menu_editquiz_frame, text="Kelola Qu
8. crud_quizez_btn.bind("<Button>", lambda e: QuizEZForm(admin_
9. crud_quizez_btn.pack(pady=10)
10. crud_quizmedium_btn = Button(menu_editquiz_frame, text="Kelo
11. crud_quizmedium_btn.bind("<Button>", lambda e: QuizMediumFor
12. crud_quizmedium_btn.pack(pady=10)
13. crud_quizhard_btn = Button(menu_editquiz_frame, text="Kelola
14. crud_quizhard_btn.bind("<Button>", lambda e: QuizHardForm(
15. crud_quizhard_btn.pack(pady=10)
```

Berikut ini akan dijelaskan source code berdasarkan line yang ditampilkan diatas:

- 1) **Line 1 – 4**, perintah ini dipergunakan untuk menampilkan *window* dari *interface* menu admin.
- 2) **Line 5 – 6**, *line* ini digunakan untuk membuat *frame* dari menu *admin* untuk menampung *button-button* kelola.
- 3) **Line 7 – 15**, pada baris perintah ini digunakan untuk menampilkan *button* untuk memasuki *menu* kelola sehingga admin dapat menuju *window* kelola.

Gambar 4.8 adalah tampilan dari ketika *admin* sudah memilih salah satu kelola *quiz* yang kemudian akan tampil berupa *table view* (table) yang berisikan pertanyaan yang akan tampil dalam *game quiz*.



QUIZ	ANSWER(UNCHANGE) - Untuk Membantu Membuat Soal Yang Harus Dijawab Iya / Tidak
Soekarno, M. Hatta, dan Radjiman menemui Terauchi di Dalat Vietnam pada 12 Agustus 1945	Iya
Soekarno menemui Terauchi pada 17 Agustus 1945	Tidak
Setelah pertemuan pada Dalat BPUPKI dibubarkan dan digantikan dengan PPKI	Iya
Golongan muda mengamankan Radjiman di Rengasdengklok 15 Agustus 1945	Tidak
Soetan Sjahrir dan golongan muda mendesak Soekarno dan Hatta untuk memproklamasikan kemerdekaan pada 16 Agustus 1945	Iya
Soekarno melaksanakan proklamasi pada 16 Agustus 1945	Tidak
PPKI diketuai oleh Ir. Soekarno dengan Moh. Hatta sebagai wakil ketua	Iya
Sidang PPKI pertama terjadi pada 16 Agustus 1945	Tidak
Hasil sidang PPKI pertama mengangkat Ir. Soekarno sebagai Presiden dan Drs. Moh. Hatta sebagai Wakil Presiden	Iya

Gambar 4. 8 Tampilan Tabel Kelola Quiz

Dari implementasi tampilan pada Gambar 4.8 Tampilan Tabel Kelola Quiz, berikut adalah potongan *source code* untuk menampilkan halaman edit pertanyaan.

```
1. class QuizEZForm(Toplevel):
2.     db_name = 'database.db'
3.     def __init__(self, master = None):
4.         super().__init__(master=master)
5.         self.title('Quizzaz - Kelola Quiz - Pem
6.         self.configure(bg="#1a237e")
7.         self.wm_attributes("-topmost", 3)
8.         self.message = Label(self, text = '',
9.         self.message.grid(row = 3, column = 0,
10.        self.tree = ttk.Treeview(self, height
11.        self.tree.grid(row = 4, column = 0, co
12.        self.tree.column("#0", width=650 ,stre
13.        self.tree.column("#1", width=500 ,stre
```

Berikut ini akan dijelaskan *source code* berdasarkan *line* yang ditampilkan diatas:

- 1) **Line 1**, merupakan class dari kelola pertanyaan untuk menampung *function* – *function* dari kelola pertanyaan.
- 2) **Line 2**, adalah untuk menentukan database mana yang akan dipanggil kedalam kelola pertanyaan.
- 3) **Line 3 – 9**, merupakan perintah untuk mengelola *window interface* dari kelola pertanyaan sehingga *interface* dapat ditampilkan ke pengguna.
- 4) **Line 10 – 13**, adalah perintah untuk membuat tabel *view* (didalam tkinter bernama *treeview*) data untuk menampilkan data pertanyaan dari *game quiz* kedalam tabel.

Jika *admin* berkehendak untuk mengedit salah satu pertanyaan maka *admin* memilih satu pertanyaan kemudian tekan tombol “EDIT” kemudian akan muncul *pop-up* yang dimana *admin* dapat mengganti isi pertanyaan implementasinya ditunjukkan pada Gambar 4.9.

Gambar 4. 9 Tampilan Edit Quiz

Dari implementasi tampilan pada Gambar 4.9 Tampilan Edit Quiz, berikut adalah potongan source code untuk menampilkan halaman edit quiz:

```

1. self.message['text'] = ''
2. try:
3. self.tree.item(self.tree.selection())['values'][0]
4. except IndexError as e:
5. self.message['text'] = 'Please, select Record'
6. return
7. quiz = self.tree.item(self.tree.selection())['text']
8. answer = self.tree.item(self.tree.selection())['values'][0]
9. Entry(self.edit_wind, textvariable = StringVar(self.edit_wi
10. Label(self.edit_wind, text = 'Deskripsi Baru:').grid(row =
11. new_quiz = Entry(self.edit_wind, width=100)
12. new_quiz.grid(row = 1, column = 2)

```

Berikut ini akan dijelaskan *source code* berdasarkan *line* yang ditampilkan diatas:

- 1) **Line 1 – 6**, adalah perintah untuk mengambil data yang dipilih untuk dilakukan *edit*.
- 2) **Line 7**, merupakan perintah untuk menampilkan data pertanyaan (*quiz*) yang dipilih kedalam *treeview* (*table view*).
- 3) **Line 8**, *line* ini dipergunakan untuk menampilkan data jawaban (*answer*) dari pertanyaan yang dipilih kedalam *treeview*.
- 4) **Line 9 – 12**, adalah untuk membuat dan menampilkan *user input* dimana admin dapat memasukkan data baru kedalam *user input* ini yang kemudian akan disimpan kedalam database.

Berikut ini adalah tampilan dari kelola foto pertanyaan yang ditunjukkan pada Gambar 4.10, disini admin dapat melakukan kelola foto dari pertanyaan berdasarkan tingkat kesulitan yang dipilih.



Gambar 4. 10 Tampilan Edit Foto Quiz

Dari implementasi tampilan pada Gambar 4.10 Tampilan Edit Foto Quiz, berikut adalah potongan *source code* untuk menampilkan edit foto *quiz*:

```

1. def ez_photoedit():
2.     conn = sqlite3.connect('database.db')
3.     window = Toplevel()
4.     window.geometry("800x600")
5.     window.title("Admin - Edit Easy Quiz Photo")
6.     window.wm_attributes("-topmost", 3)
7.     window.configure(bg="#1a237e")
8.     qz_010_frame = LabelFrame(window, text="Quiz_ID_010", bg="#1
9.     qz_010_frame.grid(column=0, row=0, padx=15, pady=15)
10.    img_call = Image.open("quiz_photo/blank.png")
11.    resized = img_call.resize((150, 100), Image.ANTIALIAS)
12.    show_photo_ez_010= ImageTk.PhotoImage(resized)
13.    global show_blank_010
14.    show_blank_010 = Label(qz_010_frame, image=show_photo_ez_010
15.    show_blank_010.pack(anchor=CENTER, padx=5, pady=10)
16.    show_photo_010 = Label(qz_010_frame, image=get_data_010('qz_
17.    ez_010_edit = Button(qz_010_frame, text="Edit Foto", bg="#ff
18.    ez_010_edit.pack(anchor=S, padx=15)

```

Berikut ini akan dijelaskan *source code* berdasarkan *line* yang ditampilkan diatas:

- 1) **Line 1**, merupakan *function* untuk membungkus perintah-perintah dari *edit* foto *quiz*.
- 2) **Line 2**, merupakan koneksi ke database *sqlite* yang bernama *database.db* sehingga program terhubung ke database.
- 3) **Line 3 – 7**, adalah perintah untuk menampilkan *window* dari *edit* foto *quiz*.
- 4) **Line 8 – 9**, dipergunakan untuk membuat frame yang menampung foto dari *quiz*.
- 5) **Line 10**, digunakan untuk memanggil foto dengan memasukkan direktori foto.
- 6) **Line 11**, adalah perintah untuk mengatur ukuran foto yang akan tampil dan memberi filter *antialias* kepada foto.
- 7) **Line 12 – 18**, berfungsi untuk menampilkan foto kedalam *frame*.

Pertama *admin* masuk ke menu *admin* kemudian sistem membalas dengan memberikan tampilan menu *admin* yang setelah itu *admin* memilih salah satu tingkat kesusahan *quiz* untuk dapat mengedit *quiz* kemudian data *quiz* akan diambil dari *database* yang kemudian akan ditampilkan oleh sistem kepada *admin*. Kemudian setelah ditampilkan isi dari *quiz* kepada *admin*, *admin* dapat memilih

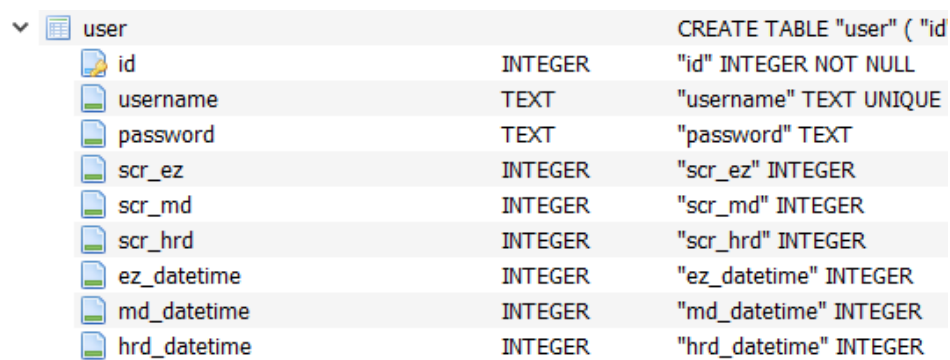
salah satu pertanyaan kemudian Kelola Pertanyaan tersebut menjadi pertanyaan yang baru.

4.2. Implementasi Data

Pada tahap ini dilakukan implementasi dari perancangan *database* yang telah dilakukan pada tahap sebelumnya. Berikut ini adalah tampilan implementasi *database* Game Edukasi “Quiz” Bertema Sejarah Indonesia Berbasis *Python*.

4.2.1. Tabel User

Tabel *user* berfungsi menyimpan data-data *user* yang telah dimasukan ke dalam *database*. Dalam tabel *user* terdapat *attribute* yaitu *id*, *username*, *password*, *scr_ez*, *scr_md*, *scr_hrd*, *ez_datetime*, *md_datetime*, *hrd_datetime*. Berikut adalah implementasi dari tabel *user* yang ditunjukkan pada Gambar 4.11.



user		CREATE TABLE "user" (
id	INTEGER	"id" INTEGER NOT NULL
username	TEXT	"username" TEXT UNIQUE
password	TEXT	"password" TEXT
scr_ez	INTEGER	"scr_ez" INTEGER
scr_md	INTEGER	"scr_md" INTEGER
scr_hrd	INTEGER	"scr_hrd" INTEGER
ez_datetime	INTEGER	"ez_datetime" INTEGER
md_datetime	INTEGER	"md_datetime" INTEGER
hrd_datetime	INTEGER	"hrd_datetime" INTEGER

Gambar 4. 11 Implementasi Tabel User

Berikut adalah SQL *query* untuk membuat tabel *user* yang akan ditunjukkan seperti berikut:

```
CREATE TABLE "user" ( "id" INTEGER NOT NULL, "username" TEXT UNIQUE, "password" TEXT, "scr_ez" INTEGER, "scr_md" INTEGER, "scr_hrd" INTEGER, "ez_datetime" INTEGER, "md_datetime" INTEGER, "hrd_datetime" INTEGER, PRIMARY KEY("id" AUTOINCREMENT) )
```

4.2.2. Tabel Quiz

Tabel *quiz* berfungsi untuk menyimpan data-data *quiz* yang telah dimasukan ke dalam *database*. Di dalam aplikasi ini terdapat 3 (tiga) tabel *quiz* yaitu *quiz_easy*, *quiz_medium*, *quiz_hard*. Setiap tabel *quiz* memiliki *attribute* yang sama sebagai berikut *id*, *quiz*, *answer_tohelp*, *qz_photo*. Berikut adalah implementasi dari tabel *quiz* yang ditunjukkan pada Gambar 4.12.

▼	quiz_easy		CREATE TABLE "quiz_easy" ("id" TEXT, "quiz" TEXT, "answer_tohelp" TEXT, "qz_photo" BLOB)
	id	TEXT	"id" TEXT
	quiz	TEXT	"quiz" TEXT
	answer_tohelp	TEXT	"answer_tohelp" TEXT
	qz_photo	BLOB	"qz_photo" BLOB
▼	quiz_hard		CREATE TABLE "quiz_hard" ("id" TEXT, "quiz" TEXT, "answer_tohelp" TEXT, "qz_photo" BLOB)
	id	TEXT	"id" TEXT
	quiz	TEXT	"quiz" TEXT
	answer_tohelp	TEXT	"answer_tohelp" TEXT
	qz_photo	BLOB	"qz_photo" BLOB
▼	quiz_medium		CREATE TABLE "quiz_medium" ("id" TEXT, "quiz" TEXT, "answer_tohelp" TEXT, "qz_photo" BLOB)
	id	TEXT	"id" TEXT
	quiz	TEXT	"quiz" TEXT
	answer_tohelp	TEXT	"answer_tohelp" TEXT
	qz_photo	BLOB	"qz_photo" BLOB

Gambar 4. 12 Implementasi Tabel Quiz

Berikut adalah SQL *query* untuk membuat tabel *quiz* yang akan ditunjukkan seperti berikut:

```
CREATE TABLE "quiz_easy" ( "id" TEXT, "quiz" TEXT, "answer_tohelp" TEXT, "qz_photo" BLOB )
CREATE TABLE "quiz_medium" ( "id" TEXT, "quiz" TEXT, "answer_tohelp" TEXT, "qz_photo" BLOB )
CREATE TABLE "quiz_hard" ( "id" TEXT, "quiz" TEXT, "answer_tohelp" TEXT, "qz_photo" BLOB )
```

4.2.3. Tabel Score

Tabel *score* berfungsi untuk mengambil data *score* yang akan diterima pemain (*user*) setelah menyelesaikan *game quiz*. Tabel *score* juga sama halnya dengan tabel *quiz*, dimana memiliki 3 (tiga) tabel dan memiliki *attribute* yang sama (dalam kasus ini nama dari tabelnya adalah; *score_easy*, *score_medium*, *score_hard*). Dalam tabel ini memiliki *attribute* berupa *id* dan *score*. Berikut ini adalah implementasi dari tabel *score* yang akan ditunjukkan pada Gambar 4.13.

score_easy		CREATE TABLE "score_easy" ("id" TEXT, "score" TEXT)
id	TEXT	"id" TEXT
score	TEXT	"score" TEXT
score_hard		CREATE TABLE "score_hard" ("id" TEXT, "score" TEXT)
id	TEXT	"id" TEXT
score	TEXT	"score" TEXT
score_medium		CREATE TABLE "score_medium" ("id" TEXT, "score" TEXT)
id	TEXT	"id" TEXT
score	TEXT	"score" TEXT

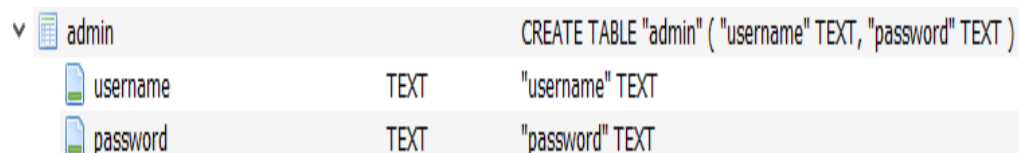
Gambar 4. 13 Implementasi Tabel Score

Berikut adalah SQL *query* untuk membuat tabel *score* yang akan ditunjukkan seperti berikut:

```
CREATE TABLE "score_easy" ( "id" TEXT, "score" TEXT)
CREATE TABLE "score_medium" ( "id" TEXT, "score" TEXT)
CREATE TABLE "score_hard" ( "id" TEXT, "score" TEXT)
```

4.2.4. Tabel Admin

Tabel *admin* berfungsi untuk menyimpan data kredensial admin supaya admin dapat melakukan *login* kedalam menu admin. Tabel admin ini terdiri 2 (dua) *column*, *attribute* dari *column* tersebut adalah *username* dan *password*. Berikut ini adalah tampilan implementasi tabel admin yang ditunjukkan pada Gambar 4.14.



admin	CREATE TABLE "admin" ("username" TEXT, "password" TEXT)
username	TEXT "username" TEXT
password	TEXT "password" TEXT

Gambar 4. 14 Implementasi Tabel Admin

Berikut adalah SQL *query* untuk membuat tabel admin yang akan ditunjukkan seperti berikut:

```
CREATE TABLE "admin" ( "username" TEXT, "password" TEXT )
```

BAB V

PENGUJIAN DAN PEMBAHASAN

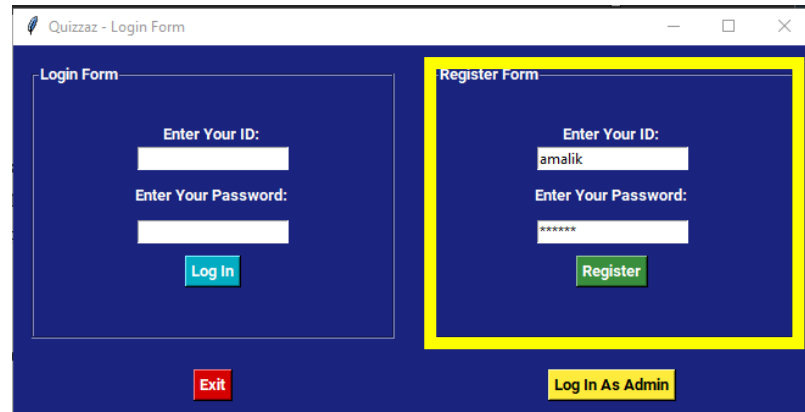
Pengujian sistem merupakan proses pengeksekusian sistem perangkat lunak untuk menentukan apakah sistem tersebut sesuai dengan spesifikasi sistem dan berjalan pada lingkungan yang diinginkan. Pengujian dilakukan dengan menguji setiap proses dan kemungkinan kesalahan yang terjadi untuk setiap proses.

5.1. Pengujian Proses

Pengujian sistem ini menggunakan pengujian *black box*. Aplikasi dinyatakan berhasil melewati pengujian apabila dalam percobaan-percobaan berikut aplikasi tidak mengalami error. Pada metode pengujian *black box*, aplikasi diberikan berbagai macam kondisi masukan, kemudian keluaran yang dihasilkan sistem akan dibandingkan dengan keluaran yang diharapkan. Berikut adalah hasil pengujian dari aplikasi game edukasi “*Quiz*” bertema sejarah Indonesia berbasis python.

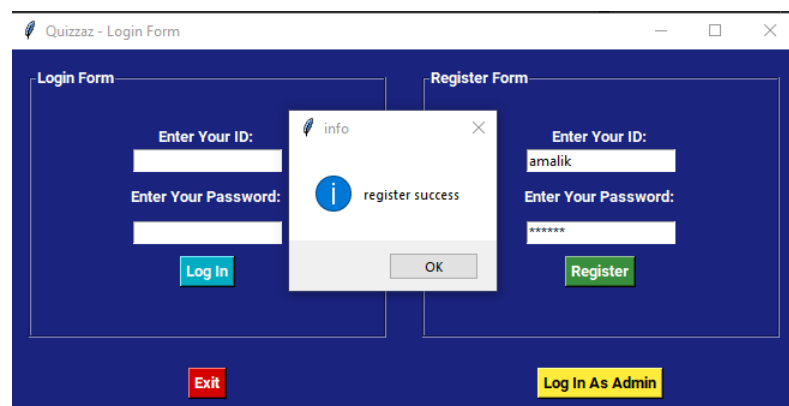
5.1.1 Pengujian Registrasi

Pengujian ini dilakukan untuk menguji tampilan awal dari aplikasi ini, yaitu dengan *login & registrasi* dimana halaman inilah dasar untuk memasuki halaman utama dari aplikasi *game* ini. Pada form ini pengujian dilakukan dengan cara memasukkan *id* dan *password* baru untuk disimpan kedalam *database*, ketika berhasil maka akan muncul *pop-up* bahwa registrasi berhasil dan *id* baru tersebut dapat digunakan sebagai *id* untuk *login*. Jika gagal, maka pengguna akan ditampilkan *pop-up* bahwa registrasi gagal. Berikut adalah hasil pengujian registrasi ditunjukkan pada Gambar 5.1.



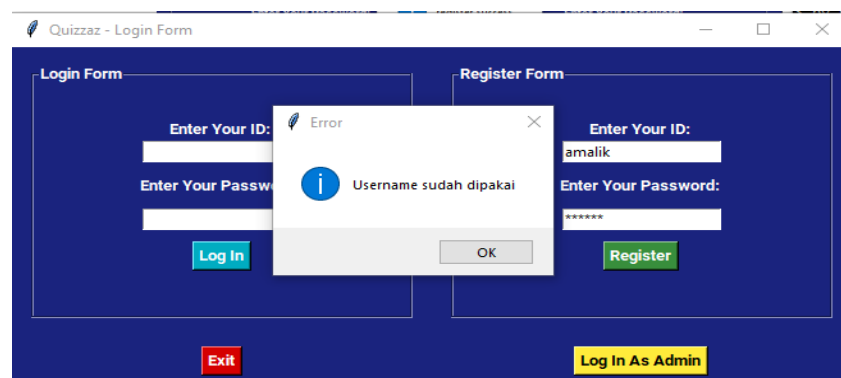
Gambar 5. 1 Pengujian Registrasi

Berikut adalah hasil dari proses registrasi dengan pengguna memiliki *id* dan *password* yang dapat digunakan untuk *login*, hasilnya akan ditunjukkan pada Gambar 5.2.



Gambar 5. 2 Hasil Pengujian Registrasi

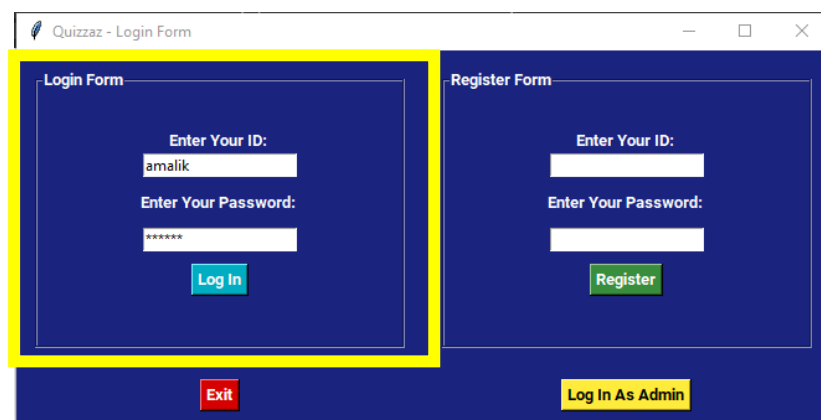
Berikut adalah hasil dari proses registrasi apabila gagal yang ditunjukkan pada Gambar 5.3.



Gambar 5. 3 Hasil Pengujian Registrasi Gagal

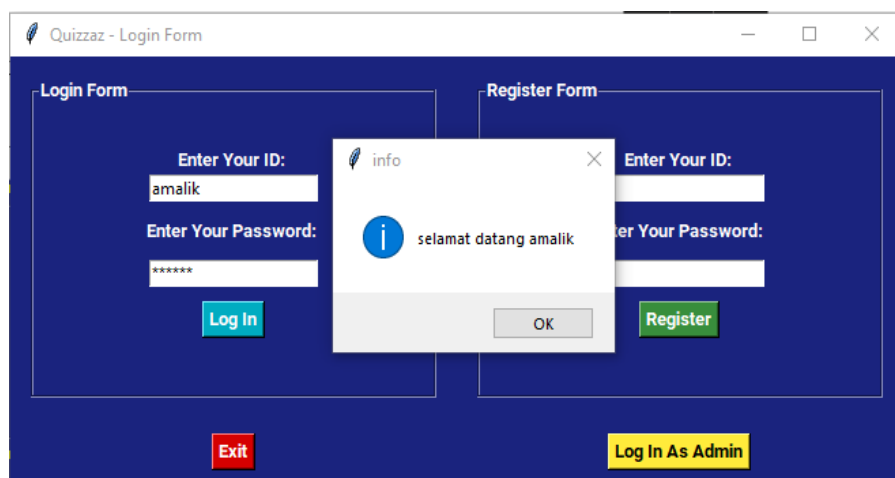
5.1.2 Pengujian Login

Pengujian ini dilakukan untuk menguji tampilan awal dari aplikasi, yaitu pengujian *login* pengguna dimana halaman inilah dasar untuk memasuki halaman utama dari aplikasi ini. Pada form *login* ini pengujian dilakukan dengan cara memasukkan *id* dan *password* yang telah tersimpan di *database*, ketika berhasil *login* maka pengguna akan menerima *pop-up login* berhasil kemudian masuk kedalam *menu* utama dari aplikasi. Jika gagal maka pengguna akan menerima *pop-up login* gagal. Berikut adalah hasil pengujian login yang akan ditunjukkan pada Gambar 5.4.



Gambar 5. 4 Pengujian Login

Berikut adalah hasil proses *login* berhasil maka pengguna akan menerima tampilan *pop-up* berhasil *login* kemudian akan masuk kedalam menu utama dari aplikasi. Berikut adalah contoh gambarnya:



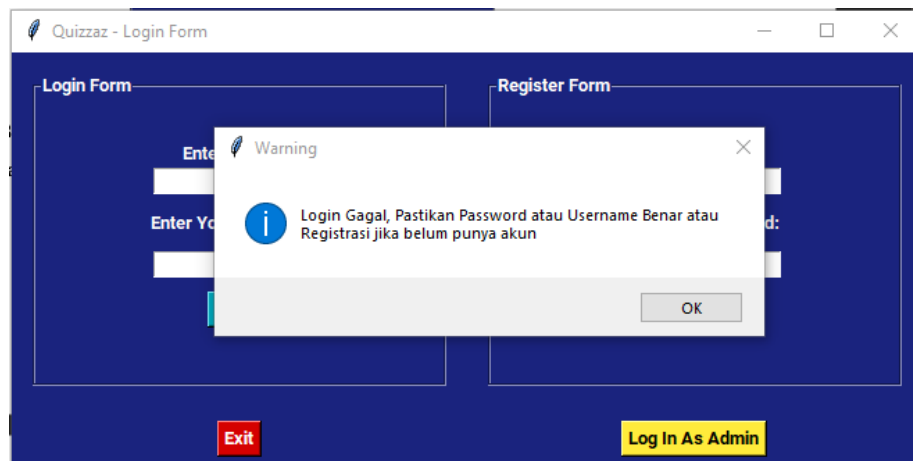
Gambar 5. 5 Hasil Pop-Up Login

Setelah *pop-up login* di ok oleh pengguna maka aplikasi akan menampilkan menu utama sebagai berikut:



Gambar 5. 6 Hasil Login Masuk Ke Menu Utama

Berikut adalah hasil dari pengujian *login* apabila pengguna gagal melakukan *login* yang ditunjukkan pada Gambar 5.7.

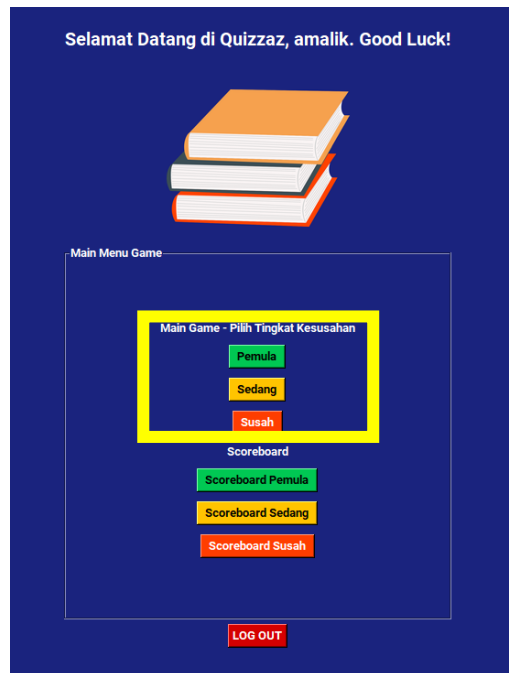


Gambar 5. 7 Hasil Pop-Up Login Gagal

5.1.3 Pengujian Play Game

Pengujian ini dilakukan untuk menguji tampilan dari *play game*. Pada tampilan ini pengguna dapat memilih tingkat kesulitan yang akan diambil oleh

pengguna, dimana setiap tingkat kesulitan ini memiliki pertanyaan dan nilai yang berbeda. Berikut ini akan ditampilkan pada Gambar 5.8

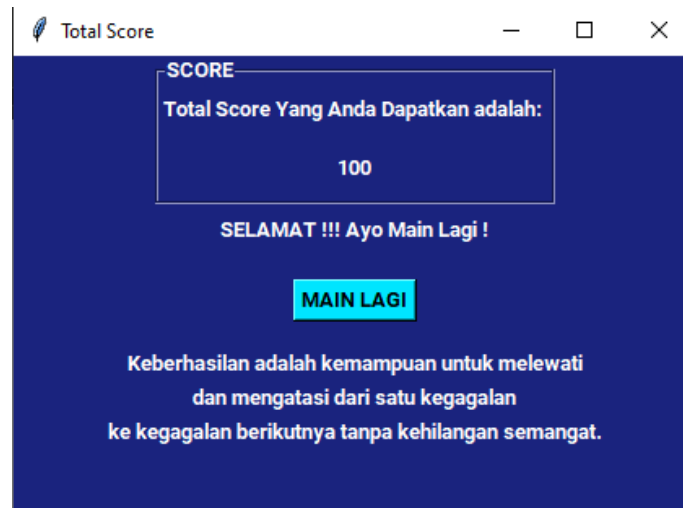


Gambar 5. 8 Pengujian *Play Game* Memilih Tingkat Kesulitan

Berikut adalah hasil proses setelah pengguna memilih tingkat kesulitan maka akan dialihkan pada *game* seperti pada Gambar 5.9 setelah *game* usai pengguna akan menerima tampilan *total score* yang ditampilkan pada Gambar 5.10.



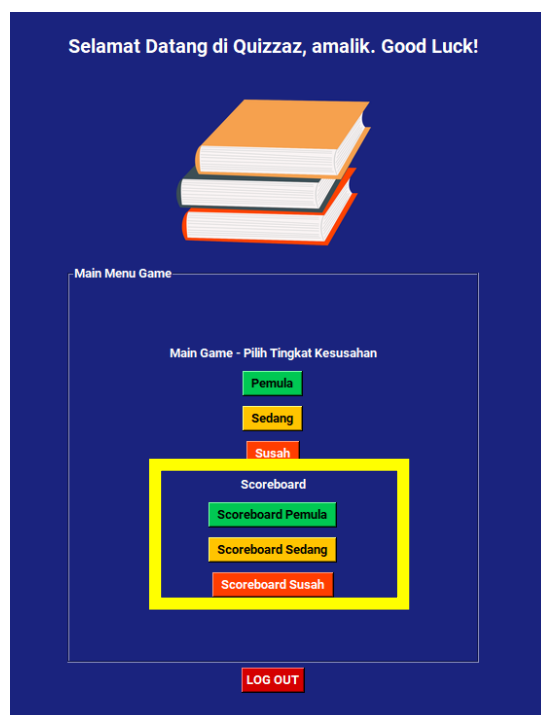
Gambar 5. 9 Hasil Pengujian *Play Game* Memainkan *Quiz*



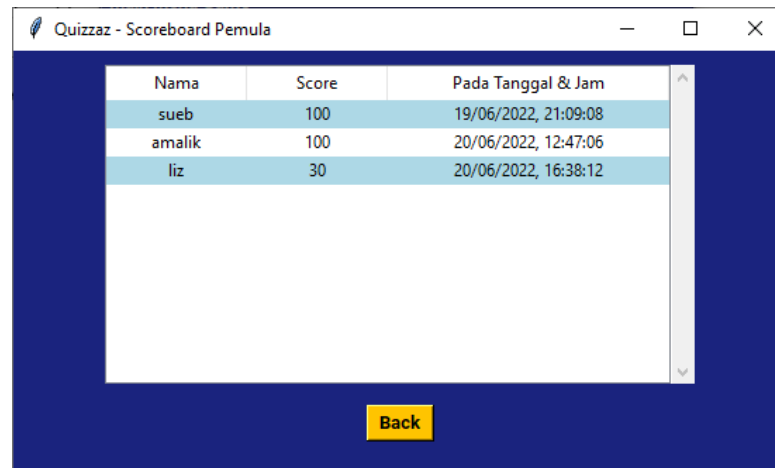
Gambar 5. 10 Hasil Pengujian *Play Game Total Score*

5.1.4 Pengujian Menampilkan Nilai

Pengujian menampilkan nilai dilakukan dengan cara menampilkan nilai yang didapatkan pengguna setelah melakukan proses *play game*. Pengguna memasuki *main menu* kemudian dapat memilih tampilan nilai *scoreboard* yang sesuai dengan tingkat kesulitan yang tersedia, data yang tampil dalam *scoreboard* adalah data yang disimpan di *database* setelah pengguna menyelesaikan proses *play game* seperti yang ditunjukkan pada Gambar 5.11 dan Gambar 5.12.



Gambar 5. 11 Pengujian Menampilkan Nilai



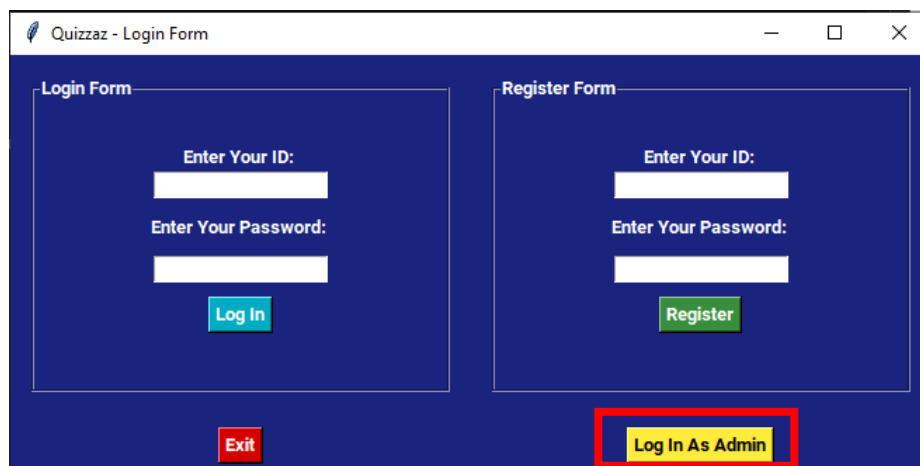
Nama	Score	Pada Tanggal & Jam
sueb	100	19/06/2022, 21:09:08
amalik	100	20/06/2022, 12:47:06
liz	30	20/06/2022, 16:38:12

Back

Gambar 5. 12 Hasil Pengujian Menampilkan Nilai

5.1.5 Pengujian Kelola Pertanyaan

Pengujian Kelola Pertanyaan dilakukan dengan cara menampilkan pertanyaan yang tersimpan dalam *database* kemudian ditampilkan setelah *admin* melakukan *login*. Pada menu utama *admin* melakukan proses *log in as admin* yang kemudian *admin* memasukkan *id* dan *passwordnya* yang kemudian sistem akan membalas dengan menampilkan *menu admin*. Di dalam *menu admin*, *admin* dapat melakukan *edit* pertanyaan yang di kategorikan menurut tingkat kesulitannya. Berikut adalah gambar dari pengujian



Quizzaz - Login Form

Login Form

Enter Your ID:

Enter Your Password:

Log In

Register Form

Enter Your ID:

Enter Your Password:

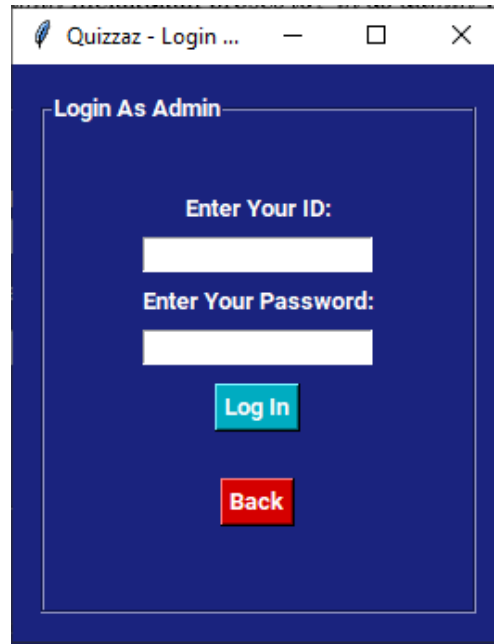
Register

Exit

Log In As Admin

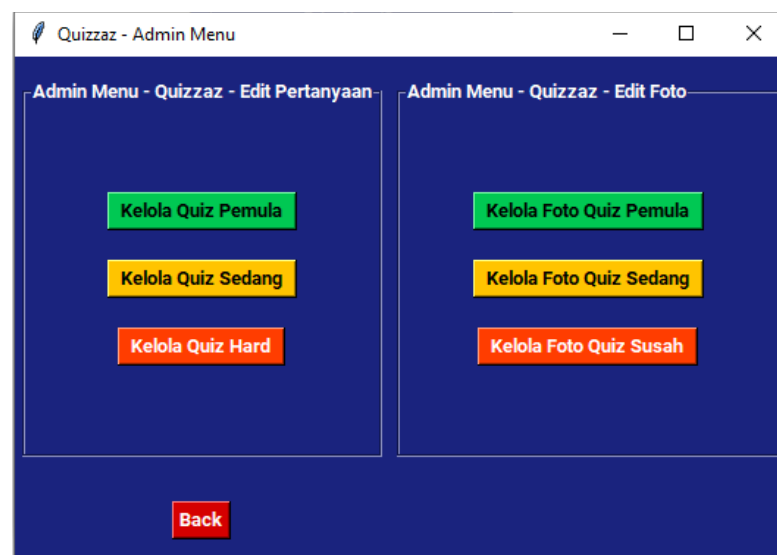
Gambar 5. 13 Pengujian Kelola Pertanyaan Memilih Log In As Admin

Setelah *admin* klik tombol “Log In As Admin” maka *admin* akan disuguhkan oleh tampilan *menu login admin* yang dimana *admin* dapat memasukkan *ID* dan *password* dari *admin*.



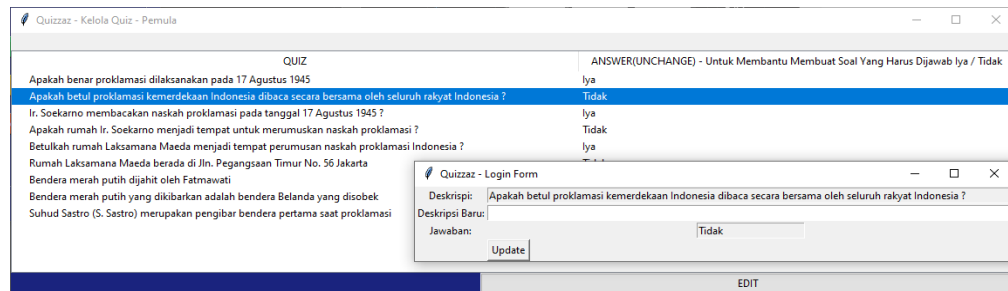
Gambar 5. 14 Pengujian Kelola Pertanyaan *Login*

Sesaat *admin* sudah melakukan *login* maka tampilan yang akan diberikan oleh sistem kepada *admin* adalah tampilan *menu* dari admin yang ditunjukkan pada Gambar 5.15.



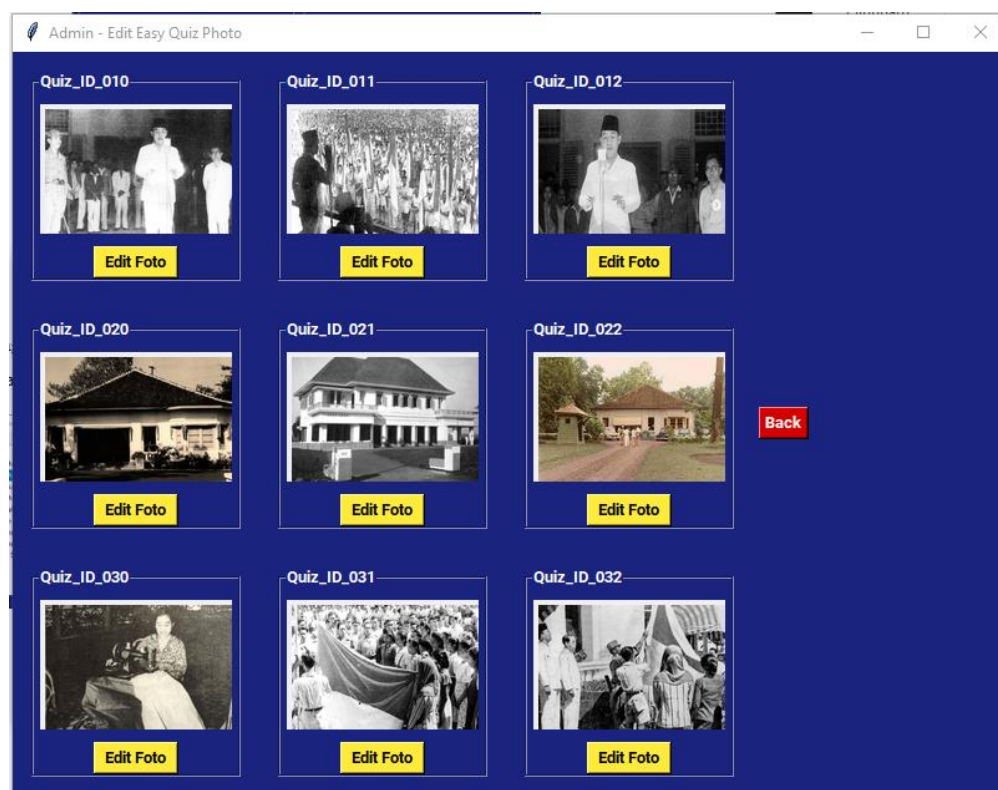
Gambar 5. 15 Pengujian Kelola Pertanyaan *Admin Menu*

Berikut adalah tampilan dari edit pertanyaan yang dimana *admin* dapat mengganti pertanyaan yang muncul di dalam *game quiz* seperti yang ditampilkan pada Gambar 5.16.



Gambar 5. 16 Hasil Pengujian Kelola Pertanyaan bagian Quiz

Berikut ini adalah hasil pengujian dari Kelola Pertanyaan bagian gambar yang ditunjukkan pada Gambar 5.17



Gambar 5. 17 Hasil Pengujian Kelola Pertanyaan Gambar

5.2. Analisis Hasil Pengujian

Dari hasil pengujian terhadap aplikasi *game* “Quiz”, dapat dilihat bahwa secara umum aplikasi mampu menjalankan setiap proses uji coba secara baik dan berhasil. Aplikasi *game* “Quiz” bertema sejarah Indonesia ini dapat digunakan oleh pengguna sebagai hiburan yang mengedukasi dalam hal peristiwa-peristiwa yang membangun negeri Indonesia. Berikut ini adalah data deskripsi dari hasil pengujian sistem ditunjukkan pada Tabel 5.1.

Tabel 5. 1 Tabel Pengujian

Deskripsi	Prosedur	Masukan	Hasil yang Diharapkan	Hasil Pengujian
Pengujian Registrasi	Memasukkan <i>username</i> dan <i>password</i> pada <i>form</i> registrasi	Tekan registrasi	Data <i>username</i> dan <i>password</i> tersimpan di <i>database</i>	Sesuai
Pengujian Login	Memasukkan <i>username</i> dan <i>password</i> pada <i>form</i> login	Tekan <i>login</i>	Masuk ke menu utama <i>game</i>	Sesuai
Pengujian Play Game	Memilih tingkat kesulitan <i>quiz</i> game	Memilih jawaban	Tampil pertanyaan, nilai, dan jika menjawab dengan benar mendapat nilai	Sesuai
Pengujian Menampilkan Nilai	Memilih <i>scoreboard</i>	Memilih <i>scoreboard</i>	Tampil papan nilai dengan nama <i>user</i> & waktu yang dicapai	Sesuai
Pengujian Kelola Pertanyaan	<i>Admin</i> memilih <i>edit</i> pertanyaan	Memilih <i>edit</i> menurut tingkat	Tampil seluruh pertanyaan dan dapat melakukan <i>edit</i> pertanyaan	Sesuai

BAB VI

KESIMPULAN

6.1. Kesimpulan

Dari penulisan Laporan Akhir yang berjudul “*Game* Edukasi “*Quiz*” Bertema Sejarah Indonesia Berbasis Python” maka dapat diambil kesimpulan sebagai berikut:

1. Berhasil membangun game sejarah kemerdekaan Indonesia berbasis python.
2. Berhasil menerapkan fungsional *user* dari game quiz yaitu registrasi yang dimana *user* dapat melakukan registrasi sehingga pengguna dapat melakukan registrasi, login sehingga pengguna dapat melakukan login, menjalankan game sehingga *user* dapat menjalankan game, kemudian melihat nilai sehingga *user* dapat melihat papan nilai *user*.
3. Berhasil menerapkan pemrograman python dan *library* pada python untuk game quiz.
4. Berhasil menerapkan fungsional admin dari game quiz yaitu login sehingga admin dapat melakukan login untuk masuk kedalam menu admin kemudian menerapkan fungsional kelola pertanyaan dimana admin dapat melakukan kelola pertanyaan *quiz* dan foto *quiz*.

6.2. Saran

Saran yang dapat diberikan untuk pengembangan lebih lanjut dari “*Game* Edukasi “*Quiz*” Bertema Sejarah Indonesia Berbasis Python” adalah kedepannya dapat di implementasikan menggunakan *interface* Kivy sehingga aplikasi memiliki *native support* pada setiap perangkat yang dapat menjalankan *python*. Hal ini untuk menghindari terjadinya perbedaan hasil tampilan halaman di setiap sistem operasi berbeda dikarenakan memiliki cara *rendering* yang berbeda pada setiap sistem operasi.

DAFTAR PUSTAKA

- Ali, R. M. (2005). *Pengantar Ilmu Sejarah Indonesia* (Cet. ke-1). LKiS.
- Allen, G., & Owens, M. (2010). The Definitive Guide to SQLite. In *The Definitive Guide to SQLite*. <https://doi.org/10.1007/978-1-4302-3226-1>
- Barros, B., Marisa, F., & Wijaya, I. D. (2018). Pembuatan Game Kuis Siapa Pintar. *J I M P - Jurnal Informatika Merdeka Pasuruan*, 3(1), 44–52. <https://doi.org/10.37438/jimp.v3i1.88>
- Clark, A. (2019). *Pillow (PIL Fork)*. 2019. <https://pillow.readthedocs.io/en/stable/?badge=latest>
- Dwi Latifatul Fajri. (2021). *Sejarah dan Isi Teks Proklamasi Kemerdekaan Indonesia*. Katadata.Co.Id. <https://katadata.co.id/safrezi/berita/61a0681ac32e6/sejarah-dan-isi-teks-proklamasi-kemerdekaan-indonesia>
- Isammudin, Pragantha, J., & Haris, D. A. (n.d.). Pembuatan Game Quiz 2D Pahlawan Masa Depan Pada Platform Pc. *Errr*.
- Ningrum, S., & Maslan, A. (2020). Aplikasi Trivia Game Mata Pelajaran Sejarah Berbasis Android. *Khazanah Ilmu Berazam*, 3(2), 327–336. <https://ejournal.ymbz.or.id/index.php/KIB/article/view/183>
- Pete Shinnars. (n.d.). *Pygame About*. Pygame.Com. <https://www.pygame.org/wiki/about?parent=>
- Purwanto, A. (2021). *Hari Kebangkitan Nasional: Latar Belakang, Polemik, dan Profil Budi Utomo*. Kompaspedia. <https://kompaspedia.kompas.id/baca/paparan-topik/hari-kebangkitan-nasional-latar-belakang-polemik-dan-profil-budi-utomo?>
- Python. (n.d.). *What is Python? Executive Summary*. <https://www.python.org/doc/essays/blurb/>
- Rafiqin, A., & Saputra, D. (2017). Pembuatan Aplikasi Game Kuis “Pontianak Punye” Berbasis Android. *Jurnal Khatulistiwa Informatika*, V(2), 71–84.
- Rendra Sanjaya. (2021). *BPUPKI dan PPKI: Mempersiapkan Kemerdekaan Indonesia*. Kompaspedia. <https://kompaspedia.kompas.id/baca/infografik/kronologi/bpupki-dan-ppki-mempersiapkan-kemerdekaan-indonesia>
- Sinan, B., & Alhir, S. (1999). *Understanding the Unified Modeling Language (UML)*. 1–8.

LAMPIRAN