

BAB 3 METODOLOGI PENELITIAN

3.1 Studi Literatur

Studi literatur dilakukan dengan menelusuri dan mempelajari berbagai referensi serta teori yang akan digunakan dalam proses penelitian. Sumber referensi dan teori yang digunakan berasal dari artikel, jurnal, dan buku yang berkaitan dengan *ReactJS*, Gamifikasi, *Octalysis*, *Hedonic Motivation System Adoption Model (HMSAM)*, dan Skala Likert.

3.2 Perancangan Gamifikasi

Website pembelajaran *library ReactJS* akan dibangun menggunakan metode gamifikasi dengan *framework octalysis*. Elemen-elemen permainan yang akan digunakan berdasarkan delapan pendorong inti (*8 core drives*) dari *framework octalysis* adalah sebagai berikut:

1. *Epic Meaning & Calling*

Elemen permainan yang diambil dari *Epic Meaning & Calling* adalah elemen *Narrative*. Narasi atau alur cerita yang dirancang menggunakan tema dunia *pixel RPG*, sehingga menciptakan suasana fiktif yang membuat pengguna merasa dirinya adalah tokoh utama dalam sebuah misi pembelajaran yang penting.

2. *Development & Accomplishment*

Elemen permainan yang diambil dari *Development & Accomplishment* adalah *Badges* dan *Progress Bar*. Kedua elemen ini diterapkan pada sistem *reward* sebagai simbol pencapaian yang diberikan kepada pengguna setelah berhasil menyelesaikan sebuah *quest*. Dengan adanya visual dalam bentuk *badge* dan *progress bar*, pengguna dapat secara langsung melihat kemajuannya.

3. *Empowerment of Creativity & Feedback*

Elemen permainan yang diambil dari *Empowerment of Creativity & Feedback* adalah *Milestone Unlocks*. Pengguna dapat membuka/*unlock* sebuah *quest* apabila pengguna telah berhasil menyelesaikan *quest* sebelumnya. Elemen

ini memberikan struktur progresif yang jelas dan menyediakan umpan balik secara langsung atas keberhasilan pengguna.

4. *Ownership & Possession*

Elemen permainan yang diambil dari *Ownership & Possession* adalah *Exchangeable Points, Avatar, dan Pet Companion*. Pengguna dapat mengumpulkan sejumlah poin melalui sistem *reward* yang tersedia pada daftar *quest*. Pada setiap kuis yang berhasil diselesaikan, pengguna akan mendapatkan poin sesuai tingkat kesulitan kuis. Poin yang berhasil dikumpulkan dapat digunakan untuk membeli *avatar* yang tersedia pada fitur *shop*. Pengguna juga bisa mendapatkan sebuah *pet* melalui fitur yang tersedia pada *easter egg*. Penerapan elemen ini memberikan rasa kepemilikan dan kontrol atas pencapaian pengguna serta dapat meningkatkan keterlibatan pengguna.

5. *Social Influence & Relatedness*

Elemen permainan yang diambil dari *Social Influence & Relatedness* adalah *Trophy Shelf*. Elemen permainan ini diterapkan melalui sistem *reward* yang terdapat pada bagian daftar *quest*. Pengguna dapat memperoleh *thropy* ketika pengguna berhasil menyelesaikan sebuah kuis. Pengguna juga dapat melihat koleksi *thropy* yang dimiliki pada bagian *home*. Penerapan elemen ini digunakan sebagai sebuah penghargaan simbolis kepada pengguna.

6. *Scarcity & Impatience*

Elemen permainan yang diambil dari *Scarcity & Impatience* adalah *Count Down Timer*. Elemen ini diterapkan pada bagian kuis dengan menggunakan hitungan mundur. Pengguna akan mendapatkan *bonus point* apabila berhasil menyelesaikan kuis dalam waktu tertentu. Hal ini membuat pengguna merasa termotivasi untuk menyelesaikan kuis sebelum *timer* menyentuh waktu yang telah ditentukan.

7. *Unpredictability & Curiosity*

Elemen permainan yang diambil dari *Unpredictability & Curiosity* adalah *Easter Eggs*. Elemen ini diterapkan melalui *icon easter egg* yang dapat ditemukan pada halaman *home*. Isi dari *easter egg* ini tidak diketahui sebelumnya dan pengguna hanya bisa mendapatkan satu *item* secara acak dari *easter egg* tersebut. Elemen ini menciptakan elemen kejutan

yang menyenangkan selama pembelajaran dan mendorong pengguna untuk menjelajahi fitur-fitur yang tersedia secara lebih mendalam.

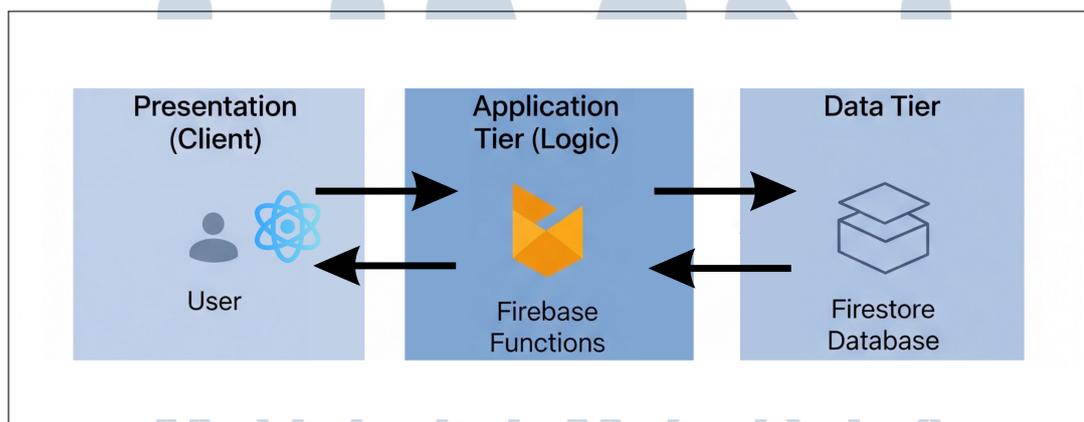
8. *Loss & Avoidance*

Elemen permainan yang diambil dari *Loss & Avoidance* adalah *Progress Loss*. Elemen *progress loss* diterapkan melalui fitur kuis yang terdapat pada daftar *quest*. Elemen ini hanya aktif apabila pengguna membuat sebuah kesalahan ketika sedang mengerjakan kuis. Kesalahan yang dilakukan oleh pengguna membuat *progress level* terhambat dan kehilangan sebagian *exp* yang dibutuhkan untuk melakukan *level up*. Elemen ini digunakan untuk meminimalisir sikap ceroboh pengguna yang belum terlalu memahami sebuah materi tetapi ingin menyelesaikan kuis secara lebih cepat.

3.3 Perancangan Aplikasi

Proses perancangan aplikasi meliputi beberapa tahapan, seperti perancangan model aplikasi, perancangan *flowchart*, perancangan antarmuka pengguna, dan perancangan aset.

3.3.1 Model Aplikasi

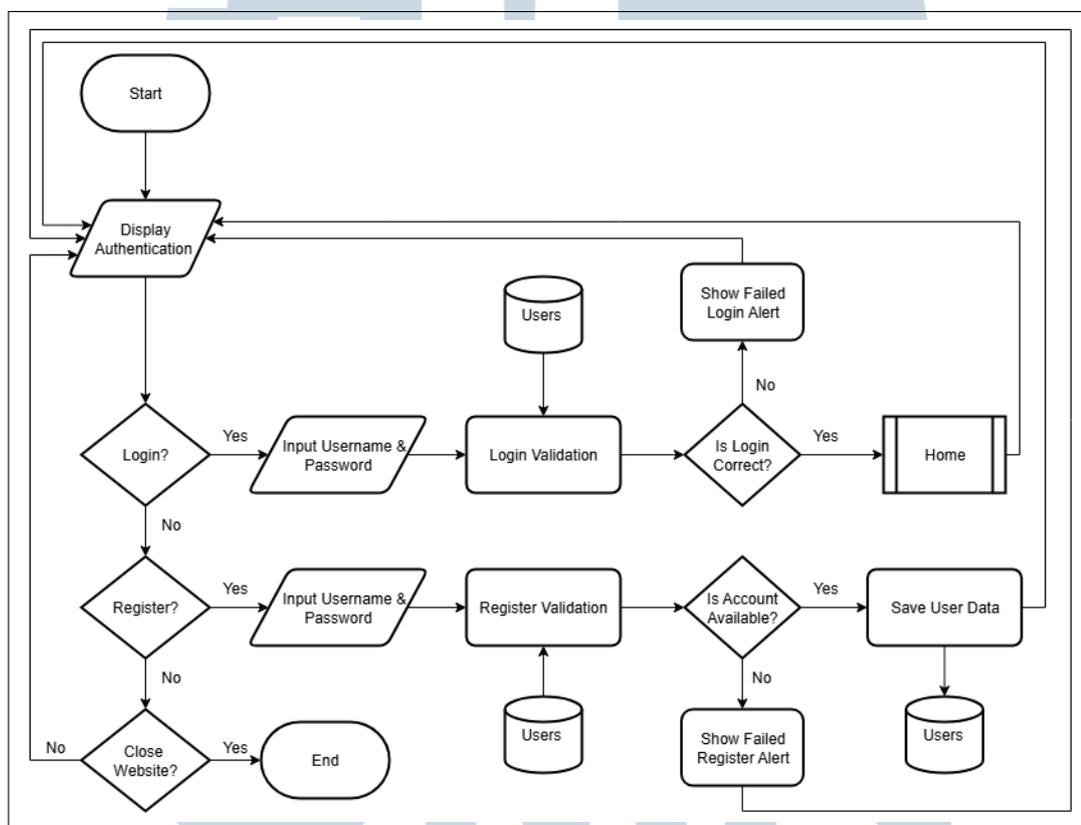


Gambar 3.1. Arsitektur Website React Realm

Gambar 3.1 menunjukkan *software architecture pattern* yang digunakan dalam model pengembangan aplikasi, yaitu *3-Tier Architecture*. Arsitektur ini mencakup tiga tingkatan, seperti *Presentation Tier*, *Application Tier*, dan *Database Tier*. Dalam struktur ini, *Application Tier (API)* berfungsi sebagai perantara untuk

operasi *CRUD* data antara *Presentation Tier* dan *Data Tier*. *Presentation Tier* dibangun menggunakan bahasa pemrograman *Javascript* dengan *framework React JS*, *Application Tier* memanfaatkan *library* dari *Firebase Functions* untuk membuat *API*, dan *Data Tier* menggunakan layanan *Cloud Firestore* yang disediakan oleh *Firebase* untuk penyimpanan data.

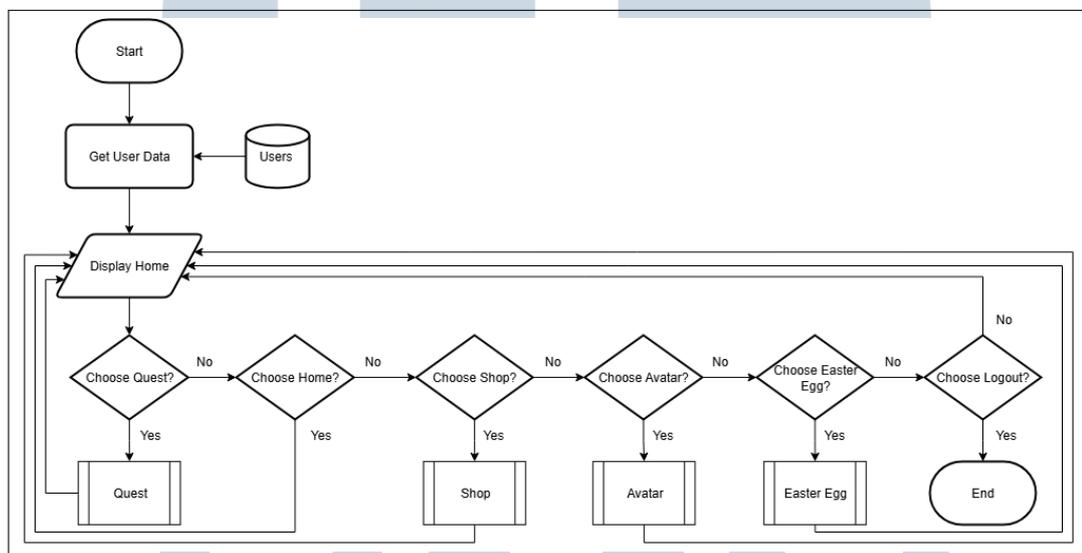
3.3.2 Perancangan *Flowchart*



Gambar 3.2. *Flowchart* Fitur *Authentication*

Gambar 3.2 menunjukkan *flowchart* untuk fitur autentikasi. Sistem akan menampilkan halaman autentikasi yang memberikan pilihan kepada pengguna untuk masuk (*login*) atau mendaftar (*register*). Apabila pengguna memilih untuk melakukan *login*, pengguna akan memasukkan *username* dan *password* pada sebuah *form* untuk divalidasi oleh sistem. Apabila data sesuai, maka proses validasi berhasil dan pengguna akan diarahkan ke halaman *home*. Namun, jika data yang dimasukkan salah, maka sistem akan memberikan notifikasi bahwa proses *login* gagal dan pengguna akan diarahkan ke halaman autentikasi. Apabila

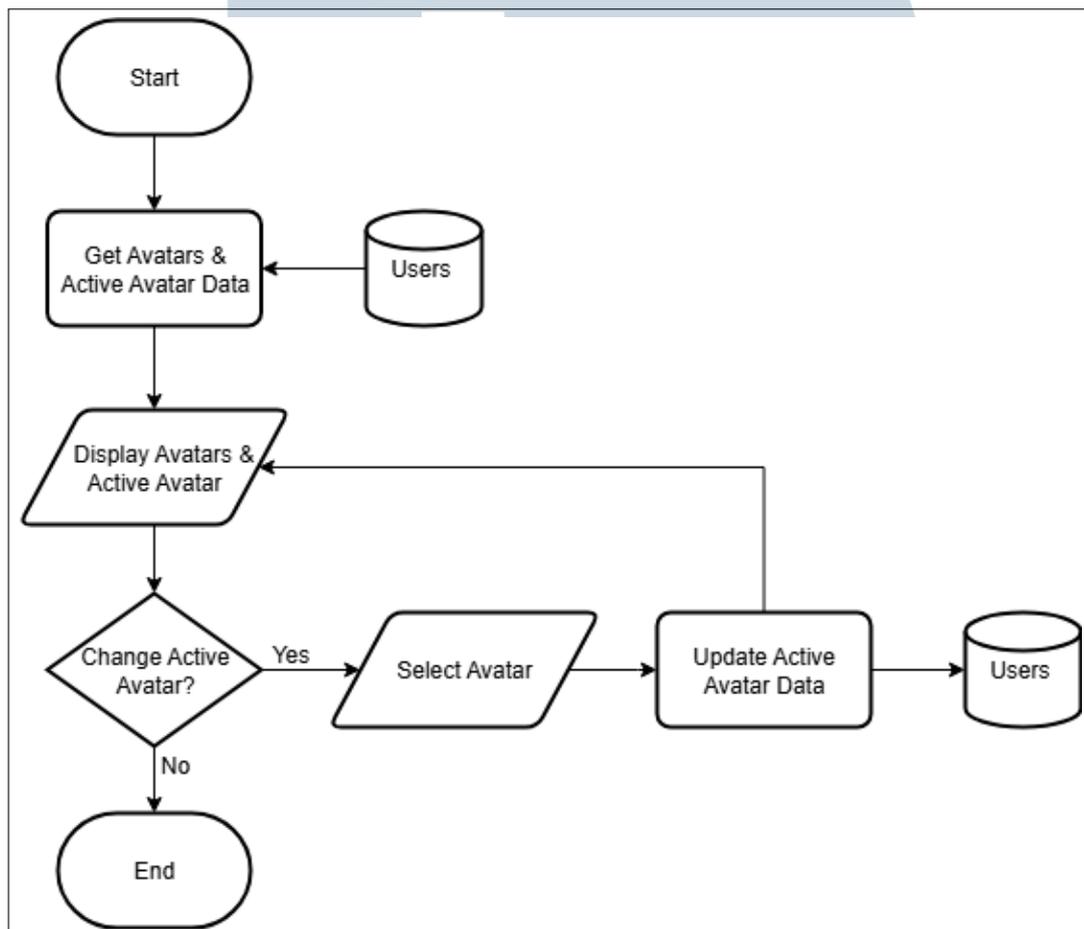
pengguna memilih untuk melakukan registrasi, pengguna akan diberikan pilihan untuk melanjutkan proses pendaftaran atau tidak. Jika pengguna ingin mendaftar, maka *input* dari data pengguna baru akan divalidasi terlebih dahulu. Apabila akun tersedia maka data akan disimpan oleh sistem ke dalam *database users* dan akan diarahkan kembali ke halaman autentikasi. Apabila akun tidak tersedia, pengguna akan mendapatkan notifikasi registrasi gagal dan akan diarahkan kembali ke halaman autentikasi. Jika pengguna tidak ingin melanjutkan proses pendaftaran dan ingin menutup *website* maka proses akan berakhir. Jika tidak, pengguna akan tetap berada pada halaman autentikasi.



Gambar 3.3. *Flowchart Halaman Home*

Gambar 3.3 menunjukkan *flowchart* pada halaman *home*. Proses *flowchart* pada Gambar 3.3 dapat dimulai apabila pengguna sudah berhasil melakukan *login*. Ketika pengguna mengakses halaman ini, sistem akan mengambil *detail* data pengguna (*Get User Data*) dari *database Users*. *Database* tersebut memberikan seluruh informasi yang dibutuhkan halaman *home*. Setelah data berhasil didapatkan, sistem akan menampilkan halaman *home* (*Display Home*). Setelah halaman *home* ditampilkan, sistem akan menunggu tindakan dari pengguna. Sistem akan memeriksa apakah pengguna memilih untuk bernavigasi ke halaman *Quest*. Jika ya, maka sistem akan menampilkan halaman *Quest*. Jika tidak, alur berlanjut ke pengecekan berikutnya, yaitu apakah pengguna memilih untuk kembali ke halaman *Home*. Jika ya, sistem akan menampilkan halaman *Home* (atau melakukan *refresh*). Jika tidak, pengecekan berlanjut ke pilihan navigasi *Shop*. Jika pengguna memilih *Shop*, maka halaman *Shop* akan ditampilkan. Jika tidak,

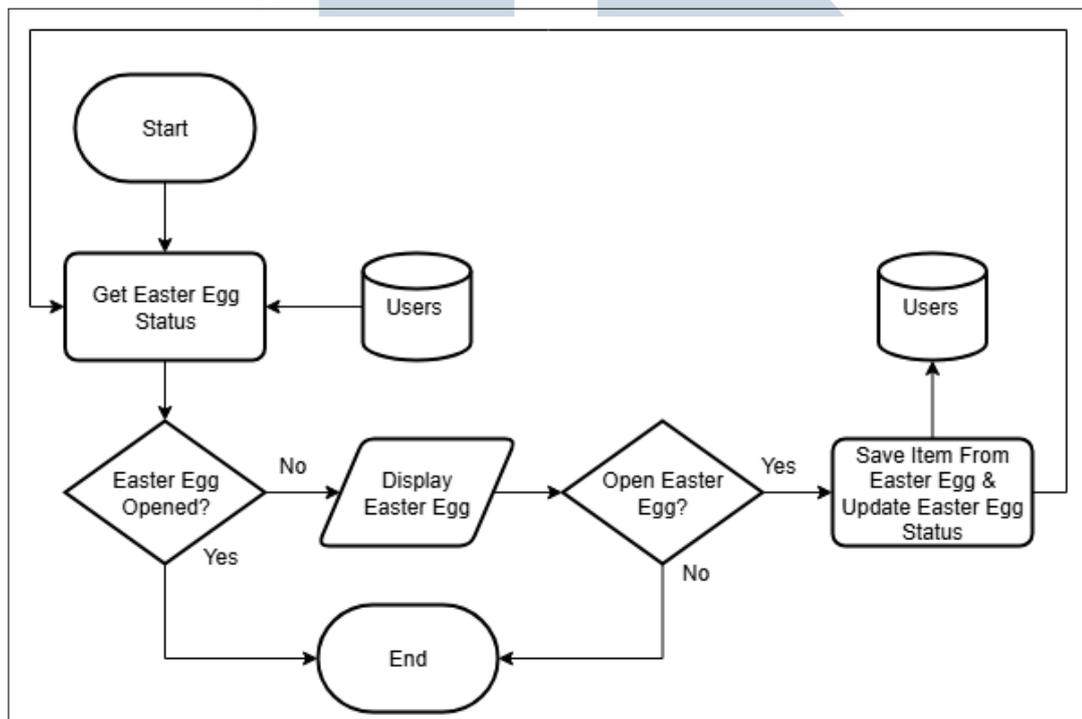
pengecekan berlanjut ke fitur *avatar*. Jika pengguna memilih fitur *avatar*, maka fitur *avatar* akan dijalankan. Jika tidak, pengecekan berlanjut ke fitur *easter egg*. Jika pengguna memilih fitur *easter egg*, maka fitur *easter egg* akan berjalan. Jika tidak, pengecekan berlanjut ke pilihan navigasi *logout*. Jika pengguna memilih untuk melakukan *logout*, maka pengguna akan diarahkan ke halaman autentikasi dan proses akan berakhir. Jika tidak, maka *website* akan kembali menampilkan halaman *home*.



Gambar 3.4. Flowchart Fitur Avatar Selection

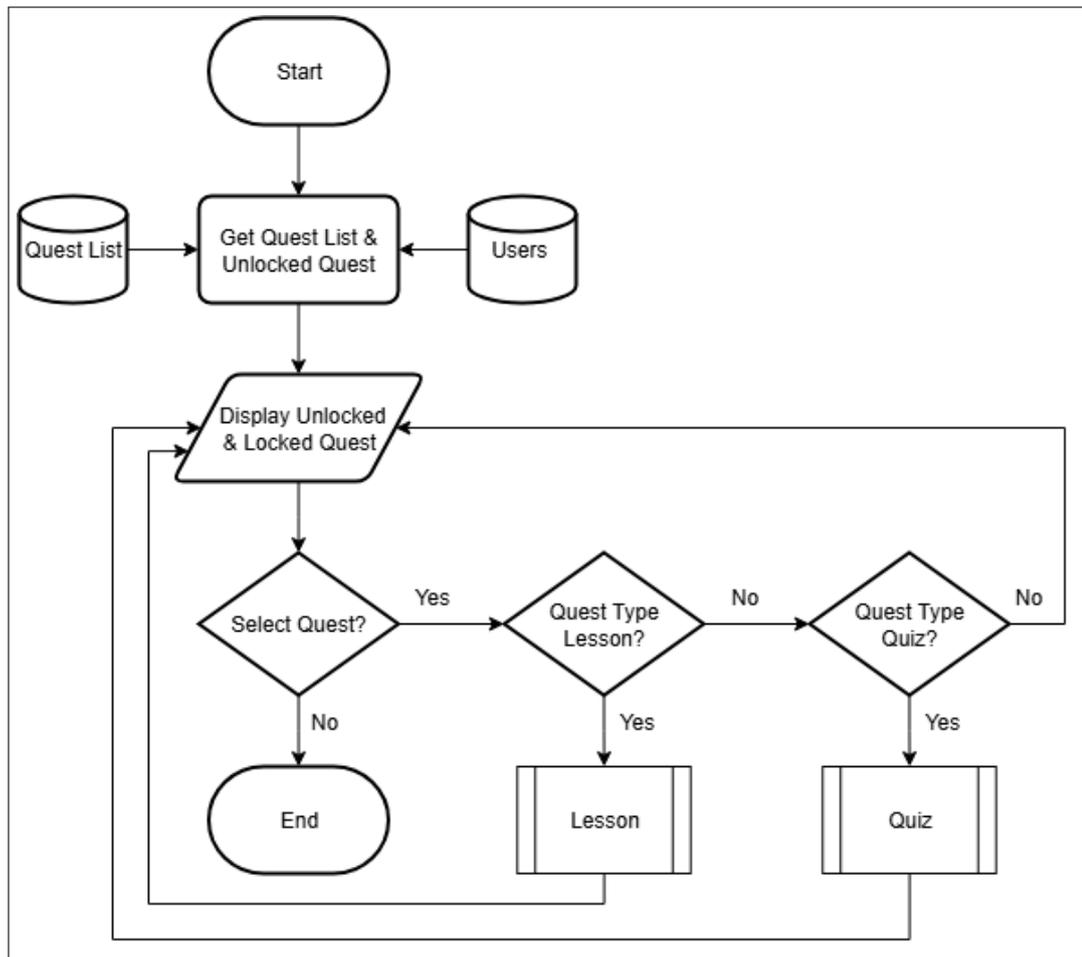
Gambar 3.4 menunjukkan *flowchart* fitur *avatar selection* yang terdapat pada halaman *home*. Ketika pengguna mengakses fitur ini, sistem akan mengambil data *avatar* yang dimiliki pengguna dan *avatar* yang sedang aktif digunakan (*Get Avatars* dan *Active Avatar Data*) dari *database Users*. Sistem kemudian menampilkan data tersebut pada halaman *home* (*Display Active Avatar*). Setelah itu, sistem akan menunggu tindakan pengguna. Jika pengguna memilih untuk tidak melakukan perubahan pada *avatar*, maka tidak akan ada perubahan pada *avatar* dan

proses akan berakhir. Jika pengguna memilih untuk mengganti *avatar*, maka sistem akan memproses *avatar* yang dipilih pengguna dan memperbarui data *avatar* aktif pengguna (*Update Active Avatar Data*) ke dalam *database Users*. Setelah proses *update* selesai, maka *avatar* akan kembali disegarkan (*refresh*).



Gambar 3.5. Flowchart Fitur Easter Egg

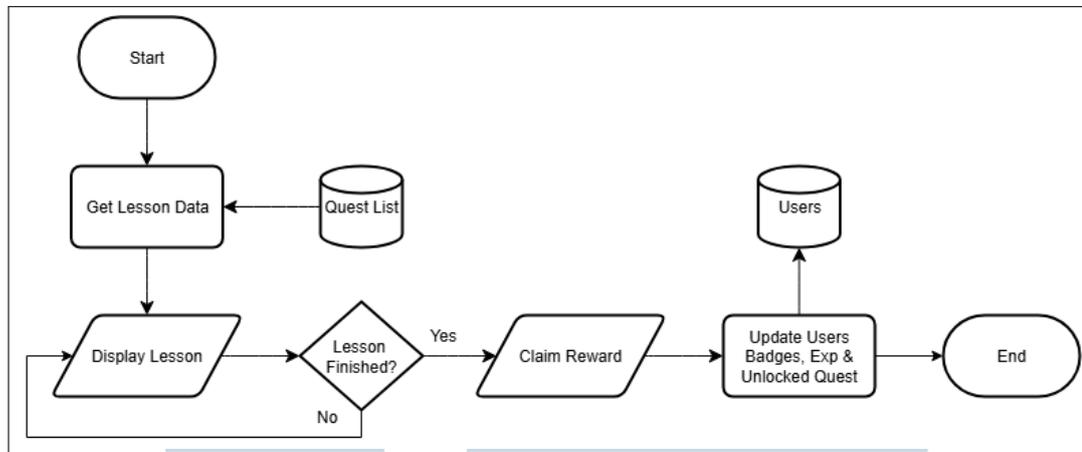
Gambar 3.5 menunjukkan *flowchart* fitur *easter egg* yang terdapat pada halaman *home*. Ketika pengguna mengakses fitur ini, sistem akan memeriksa status *easter egg* dari pengguna dengan mengambil data dari *database Users* (*Get Easter Egg Status*). Setelah melakukan pengecekan (*Is Easter Egg Opened?*), jika *easter egg* sudah pernah dibuka maka sistem akan menghapus tampilan dari *easter egg* dan proses akan berakhir. Jika pengguna belum pernah membuka *easter egg*, maka sistem akan menampilkan antarmuka *easter egg* (*Display Easter Egg*). Apabila pengguna ingin membuka *easter egg*, maka sistem akan menyimpan hadiah dari *easter egg* tersebut ke akun pengguna (*Save Item From Easter Egg*) dan memperbarui status dari *easter egg* (*Update Easter Egg Status*) ke *database Users*. Setelah data *easter egg user* berhasil diperbarui, maka antarmuka *easter egg* akan disegarkan (*refresh*). Jika pengguna memilih untuk tidak membuka *easter egg* maka proses akan berakhir.



Gambar 3.6. Flowchart Halaman Quest

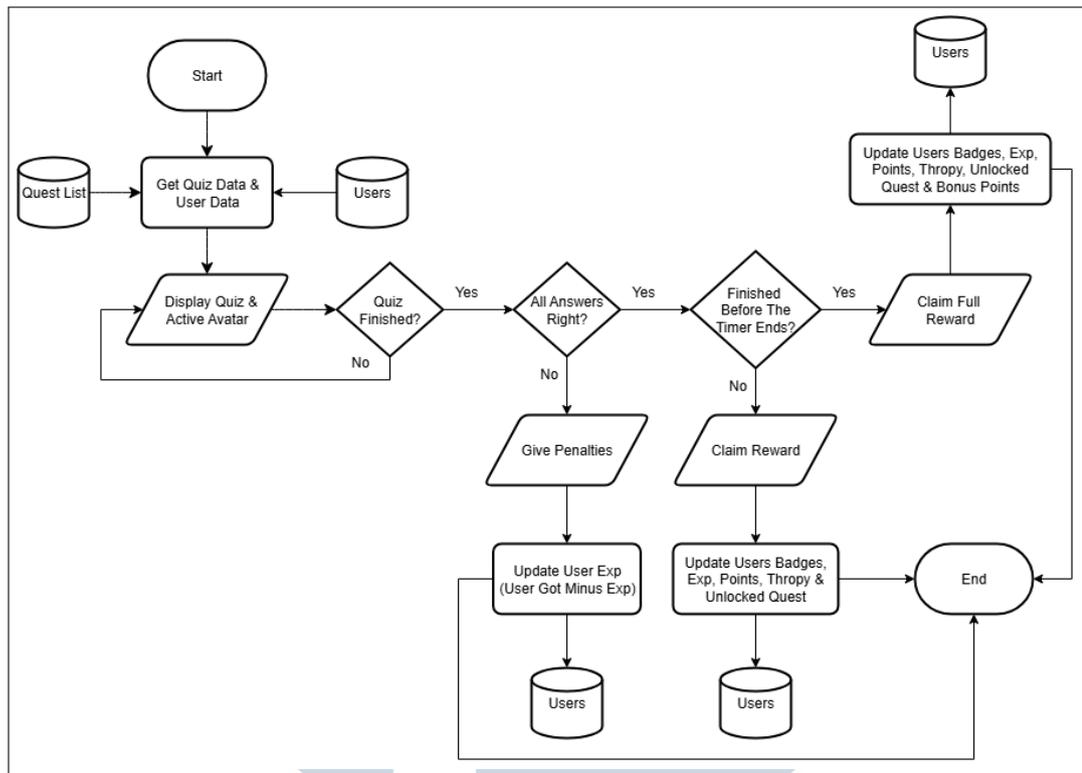
Gambar 3.6 menunjukkan *flowchart* pada halaman *quest*. Ketika pengguna mengakses halaman ini, sistem akan mengambil data dari dua *database* yaitu, data daftar seluruh *quest* dari *database quest list* dan data *quest* yang sudah terbuka dari *database users*. Kedua data tersebut kemudian digunakan untuk menampilkan daftar *quest* yang sudah terbuka dan yang belum terbuka (*Display Unlocked dan Locked Quest*). Setelah daftar *quest* ditampilkan, sistem akan menunggu tindakan dari pengguna. Apabila pengguna memilih untuk menjalankan *quest* maka sistem akan melakukan pengecekan terhadap tipe *quest* yang dipilih oleh pengguna. Jika tipe misi tersebut adalah pelajaran (*Is Quest Type Lesson?*), maka pengguna akan diarahkan ke modul pelajaran (*Lesson*). Jika tidak, sistem akan memeriksa apakah tipe misi tersebut adalah sebuah kuis (*Is Quest Type Quiz?*). Jika ya, maka pengguna akan diarahkan untuk memulai modul kuis (*Quiz*). Jika tidak, maka pengguna akan diarahkan untuk memilih *quest* yang tersedia. Apabila pengguna tidak ingin

memilih *quest* maka proses akan berakhir.



Gambar 3.7. *Flowchart* Fitur *Lesson*

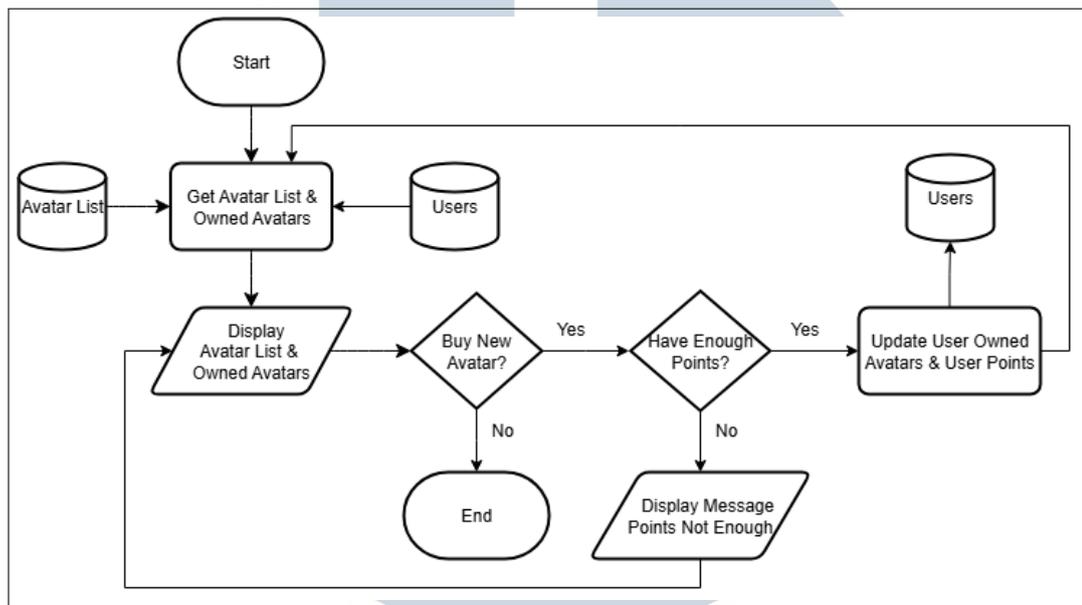
Gambar 3.7 menunjukkan *flowchart* fitur *lesson* yang terdapat pada halaman *quest*. Ketika pengguna mengakses fitur tersebut, sistem akan mengambil data atau materi pelajaran secara spesifik untuk misi tersebut (*Get Lesson Data*) dari *database Quest List*. Setelah materi berhasil didapatkan, sistem akan menampilkan antarmuka dari *lesson (Display Lesson)*. Selanjutnya, sistem akan masuk ke dalam pengecekan untuk menentukan apakah pengguna telah menyelesaikan pelajaran tersebut (*Is Lesson Finished?*). Apabila pengguna belum menyelesaikan pelajaran, pengguna akan tetap berada di halaman pelajaran untuk melanjutkan proses belajar. Jika pengguna telah menyelesaikan pelajaran alur akan berlanjut ke tahap klaim hadiah (*Claim Reward*). Pada tahap ini, sistem akan menampilkan notifikasi keberhasilan dan tombol untuk mengambil hadiah. Setelah hadiah diklaim, sistem akan melakukan pembaruan data pada akun pengguna di dalam *database Users*. Pembaruan ini mencakup penambahan lencana (*badges*), poin pengalaman (*Exp*), dan akses untuk *quest* berikutnya. Setelah data pengguna berhasil diperbarui, maka proses akan berakhir.



Gambar 3.8. Flowchart Fitur Quiz

Gambar 3.8 menunjukkan *flowchart* fitur *quiz* yang terdapat pada halaman *quest*. Ketika pengguna mengakses fitur ini, sistem akan mengambil seluruh data yang diperlukan dengan mengakses *database Quest List*, serta *database Users* untuk mengambil data *avatar* aktif pengguna. Selanjutnya, sistem menampilkan antarmuka kuis beserta *avatar* pengguna yang sedang aktif digunakan. Apabila pengguna telah selesai menyelesaikan kuis, sistem akan masuk ke tahap evaluasi. Jika terdapat satu atau lebih jawaban yang salah, maka sistem akan memperbarui data pengguna dengan memberikan pengurangan poin pengalaman (*Update Users Exp By Giving Minus Exp*) sebagai bentuk penalti. Setelah data pengguna diperbarui, maka proses akan berakhir. Jika seluruh jawaban yang diberikan pengguna benar (*All Answers Right?*), maka sistem akan melakukan pengecekan kedua untuk menentukan apakah pengguna akan mendapatkan bonus. Jika pengguna berhasil menyelesaikan kuis sebelum waktu habis (*Is Finished Before Timer Ends?*), maka pengguna akan menerima bonus poin. Setelah itu, sistem akan memperbarui data pengguna dengan memberikan lencana, poin pengalaman, piala, membuka misi selanjutnya, dan juga poin bonus tambahan. Setelah data pengguna diperbarui, maka proses akan berakhir. Apabila pengguna menjawab semua dengan

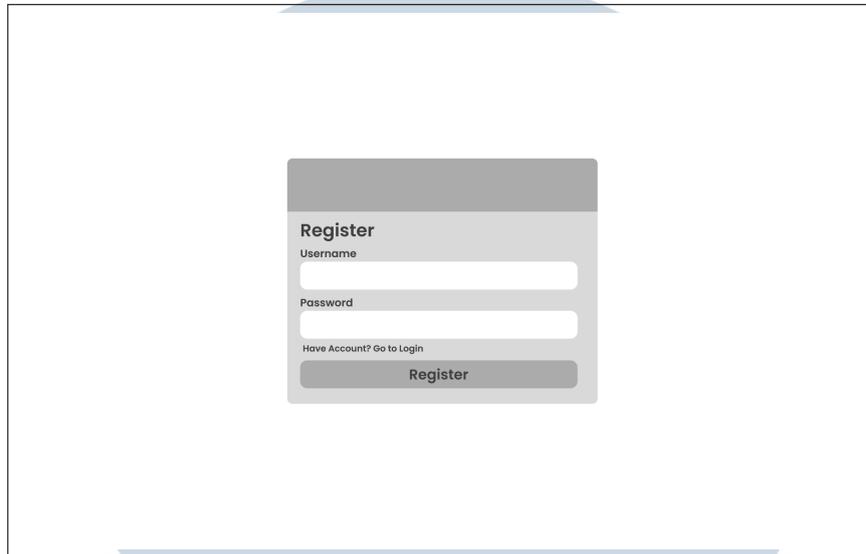
benar tetapi gagal menyelesaikan dalam batas waktu, maka pengguna tetap akan menerima hadiah standar penyelesaian misi seperti lencana, poin pengalaman, piala, dan terbukanya misi baru, tetapi tanpa poin bonus. Setelah data pengguna diperbarui, maka proses akan berakhir.



Gambar 3.9. Flowchart Halaman Shop

Gambar 3.9 menunjukkan *flowchart* pada halaman *shop*. Ketika pengguna mengakses halaman ini, sistem akan mengambil data dari kedua *database*, yaitu daftar *avatar* yang tersedia dari *database Avatar List* dan *database Users* untuk mengetahui *avatar* yang sudah dimiliki oleh pengguna serta jumlah poin pengguna. Data dari kedua *database* tersebut kemudian digunakan untuk menampilkan *list avatar* dan *avatar* yang sudah dimiliki pada antarmuka *shop*. Setelah *shop* ditampilkan, sistem akan menunggu tindakan dari pengguna. Apabila pengguna memilih untuk tidak membeli *avatar* baru, maka proses akan berakhir. Namun, jika pengguna memilih sebuah *avatar* untuk dibeli, maka sistem akan melakukan proses validasi. Apabila poin pengguna tidak cukup, sistem akan menampilkan pesan poin tidak cukup dan antarmuka *shop* akan ditampilkan kembali. Apabila poin pengguna mencukupi, transaksi akan dilakukan dan sistem akan memperbarui data pengguna di *database Users* dengan menambahkan *avatar* yang baru dibeli serta mengurangi jumlah poin pengguna sesuai harga *avatar* tersebut. Setelah pembaruan data dan proses transaksi selesai, halaman *shop* akan disegarkan kembali (*refresh*).

3.3.3 Perancangan Antarmuka Pengguna



A mockup of a registration form titled "Register". It features a header with the title "Register", followed by two input fields labeled "Username" and "Password". Below the fields is a link that says "Have Account? Go to Login" and a "Register" button at the bottom.

Gambar 3.10. *Mockup Register*

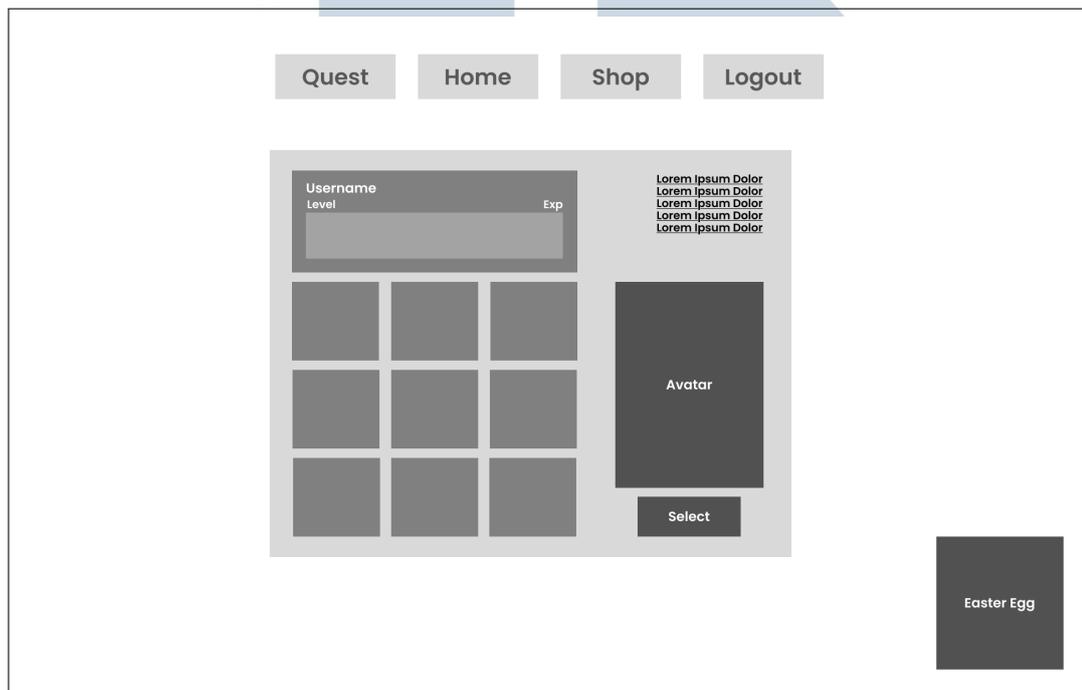


A mockup of a login form titled "Login". It features a header with the title "Login", followed by two input fields labeled "Username" and "Password". Below the fields is a link that says "New? Create Account" and a "Login" button at the bottom.

Gambar 3.11. *Mockup Login*

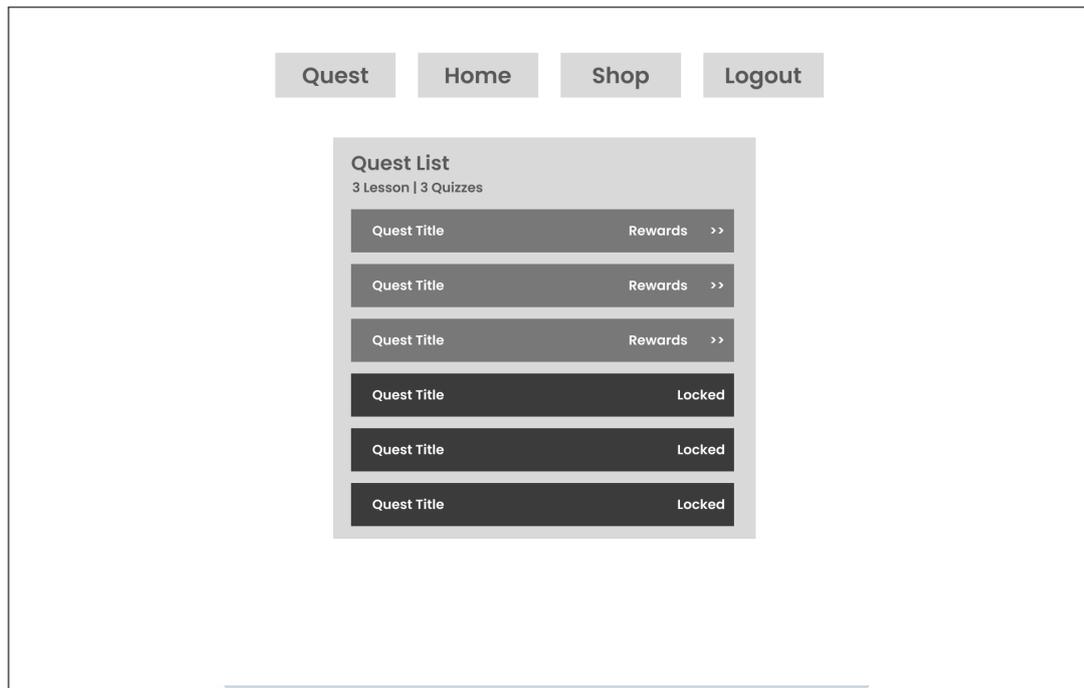
Gambar 3.10 dan gambar 3.11 menampilkan rancangan antarmuka untuk halaman autentikasi pengguna yang di dalamnya terdapat dua fitur, yaitu *Register* dan *Login*. Halaman autentikasi ditampilkan ketika pengguna pertama kali membuka *website*. Pada gambar 3.10 pengguna dihadapkan pada beberapa kolom *form*, yaitu *username* untuk menentukan nama akun dan *password* untuk membuat

kata sandi. Selain itu, terdapat tombol *register* yang berfungsi untuk mengirimkan data registrasi ke *database*. Pada gambar 3.11 terdapat *form login* dan tombol *login* pada *form* yang berfungsi untuk memvalidasi *username* dan *password* pengguna serta mengarahkan pengguna ke halaman *home*.



Gambar 3.12. *Mockup Home*

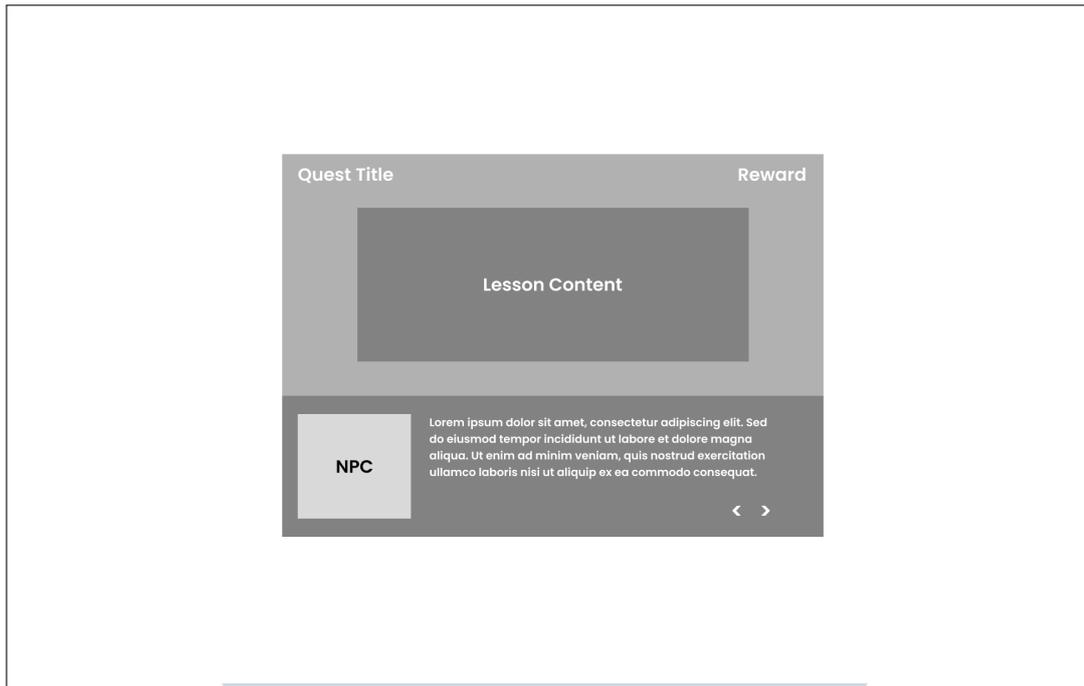
Gambar 3.12 menampilkan rancangan antarmuka untuk halaman *home*. Halaman ini akan menampilkan seluruh menu navigasi, detail status pengguna, fitur pemilihan *avatar*, serta fitur *easter egg* yang terdapat pada pojok kanan bawah halaman *home*. Pada bagian detail status pengguna, terdapat detail data dari *user*, seperti *username*, *level*, *exp*, *point*, *title*, *avatar collection*, *badges*, dan *thropy*. Pada fitur pemilihan *avatar*, bentuk persegi panjang yang terdapat tulisan *avatar* menunjukkan *avatar* yang sedang aktif digunakan. Sedangkan bentuk persegi-persegi kecil di bawah detail status pengguna menunjukkan koleksi *avatar* yang sudah dimiliki dan dapat dipilih sebagai *avatar* aktif pengguna. Pada bagian pojok kanan bawah halaman, terdapat *easter egg* yang dapat dibuka oleh pengguna. Apabila pengguna berhasil menemukan dan membuka *easter egg*, sistem akan memberikan *item* berupa 1 dari 3 hewan peliharaan (*pet*) secara acak kepada pengguna.



Gambar 3.13. *Mockup Quest*

Gambar 3.13 menampilkan rancangan antarmuka untuk halaman *quest*. Halaman ini akan menampilkan seluruh daftar *quest* yang tersedia. Pada daftar *quest*, terdapat beberapa *quest* dengan warna lebih gelap. Hal ini menunjukkan *quest* belum bisa dibuka apabila pengguna belum menyelesaikan *quest* sebelumnya. Selain itu, pada halaman ini terdapat 2 jenis *quest* yang berbeda, yaitu *lesson* dan *quiz* yang membuat kedua jenis *quest* tersebut memiliki rancangan antarmuka yang berbeda. Pada setiap *quest*, *reward* yang akan diberikan kepada pengguna sudah pasti berbeda sesuai dengan tingkat kesulitannya.

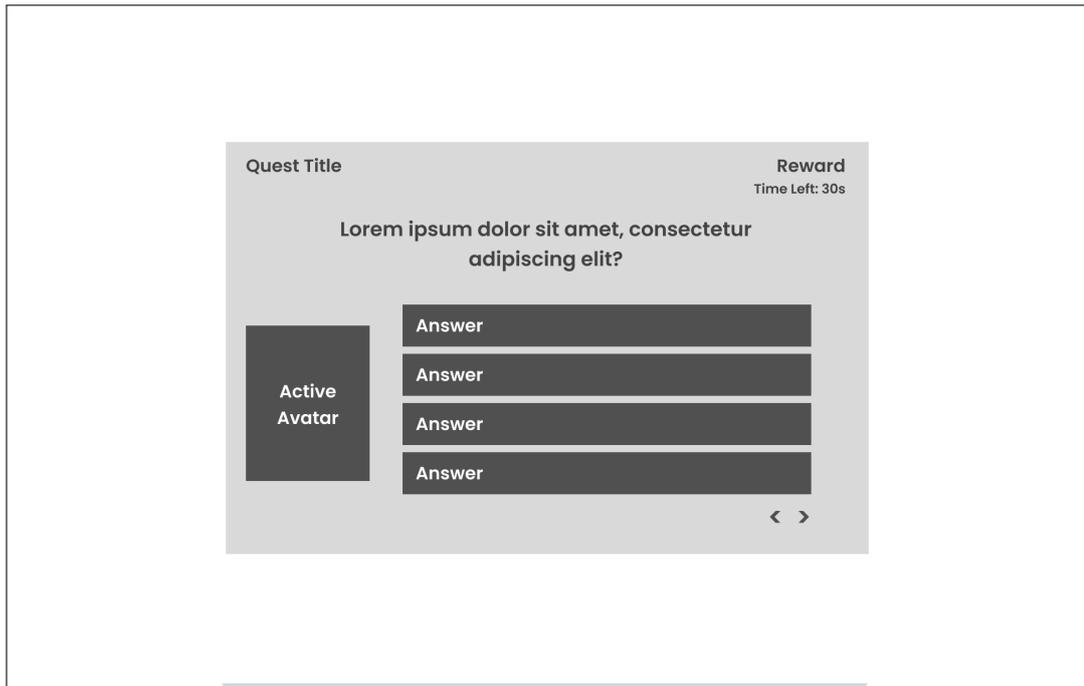
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.14. *Mockup Lesson*

Gambar 3.14 menampilkan rancangan antarmuka untuk halaman *lesson*. Halaman *lesson* terdiri dari beberapa materi pembelajaran yang sudah ditempatkan sesuai pada gambar 3.14. Ilustrasi materi akan ditempatkan pada bagian persegi panjang dengan tulisan *lesson content*, ilustrasi *master/guru* akan ditempatkan pada bagian persegi dengan tulisan *NPC*, dan penjelasan dari materi pembelajaran akan diletakkan di bawah *lesson content*. Pada bagian atas *lesson content* terdapat detail dari judul *quest* dan *reward* dari *quest* tersebut. Pada bagian bawah terdapat tombol navigasi untuk melihat materi selanjutnya/sebelumnya sesuai dengan tindakan yang ingin dilakukan pengguna.

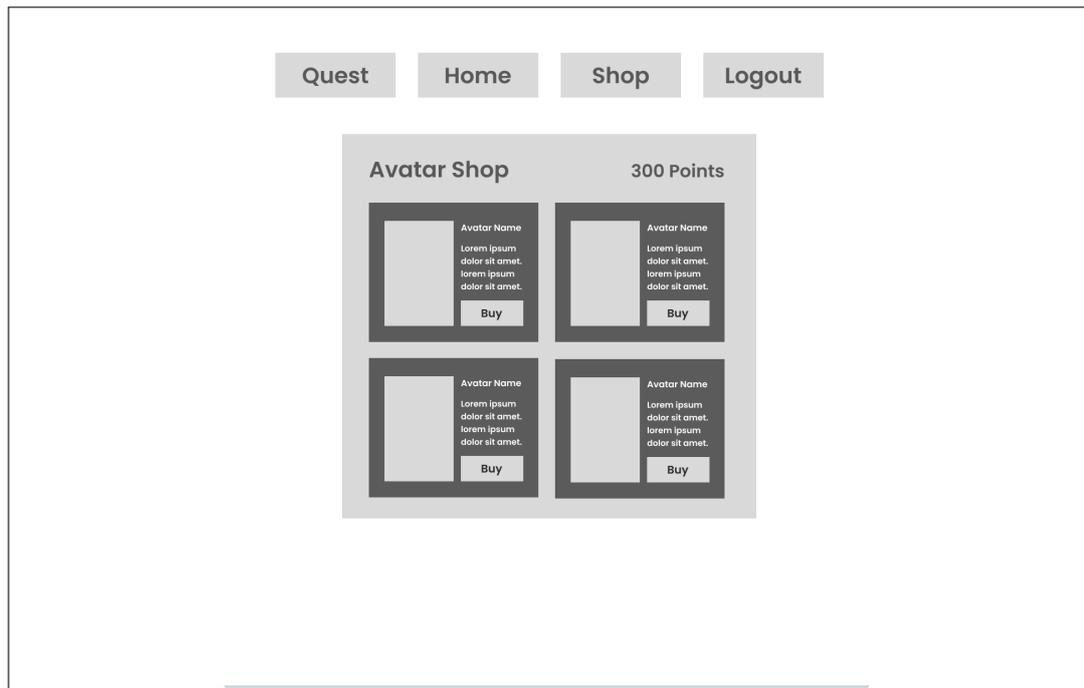
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.15. *Mockup Quiz*

Gambar 3.15 menampilkan rancangan antarmuka untuk halaman *quiz*. Pada bagian atas pertanyaan *lorem ipsum* terdapat detail dari judul *quest*, *reward quest*, dan waktu yang tersisa untuk mendapatkan bonus poin. Pada bagian inti/tengah dari fitur *quiz* terdapat pertanyaan, beberapa jawaban yang dapat dipilih, dan *avatar* yang sedang aktif digunakan. Pada bagian bawah *quiz* juga terdapat panah kiri dan kanan yang berfungsi sebagai navigasi untuk ke pertanyaan selanjutnya/sebelumnya.

U M M N
UNIVERSITAS
MULTIMEDIA
NUSANTARA



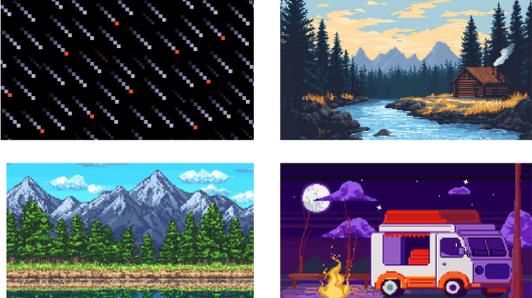
Gambar 3.16. *Mockup Shop*

Gambar 3.16 menampilkan rancangan antarmuka untuk halaman *shop*. Halaman ini terdiri dari jumlah poin pengguna, *avatar* yang sudah dimiliki, dan *avatar* yang dapat dibeli. Pada *avatar shop*, terdapat beberapa *list avatar* yang dapat dibeli. Setiap persegi dengan warna yang lebih gelap memiliki nama dari *avatar*, deskripsi *avatar*, tampilan *avatar*, dan tombol *buy* untuk membeli *avatar* tersebut. Apabila pengguna sudah memiliki *avatar*, antarmuka dari tombol *buy* akan diganti dengan bentuk persegi panjang dengan tulisan *owned* yang menandakan *avatar* sudah dimiliki dan tidak dapat dibeli kembali. Tampilan jumlah poin pengguna diletakkan pada bagian kanan atas dari *avatar shop*.

3.3.4 Perancangan Aset

Daftar aset yang digunakan dalam proses perancangan *website React Realm* terdapat pada tabel 3.1.

Tabel 3.1. Daftar aset *react realm*

Gambar	Keterangan	Sumber Gambar
	<p>Logo website</p>	<p>www.flaticon.com</p>
	<p>Latar belakang halaman autentikasi, home, quest, dan shop</p>	<p>www.freepik.com</p>
	<p>Button navigasi untuk berpindah halaman</p>	<p>Desain pribadi</p>
	<p>List avatar yang terdapat pada halaman shop</p>	<p>Desain pribadi</p>
<p>Lanjut ke halaman berikutnya</p>		

Tabel 3.1 Daftar aset *react realm* (lanjutan)

Gambar	Keterangan	Sumber Gambar
	<p>List badge dan trophy untuk achievement dan reward pada halaman quest</p>	<p>Desain pribadi</p>
	<p>List pet yang bisa didapatkan dari easter egg</p>	<p>Desain pribadi</p>
	<p>Asset panah dan locked quest pada halaman quest</p>	<p>www.flaticon.com</p>

3.4 Pembangunan Aplikasi

Pembangunan aplikasi ini melibatkan serangkaian perangkat keras dan perangkat lunak tertentu. Berikut adalah rincian spesifikasi sistem yang digunakan selama proses pengembangan.

1. Perangkat Keras

- (a) Prosesor: *Intel Core i5-10400F 2.9GHz*
- (b) Memori: *16 GB*
- (c) SSD: *Midasforce SSD Superlightning 128 GB*
- (d) Graphics Card: *NVIDIA GeForce GTX 1650*

2. Perangkat Lunak

- (a) Aplikasi IDE: *Visual Studio Code*
- (b) Bahasa Pemrograman: *Javascript*

- (c) *Library Front End: React JS*
- (d) *CSS Framework: Tailwind CSS*
- (e) *Component Framework: Ant Design*
- (f) *Database: Firebase Firestore*
- (g) *Routing: React Router DOM*
- (h) *Frontend Build Tool: Vite*
- (i) *Package Manager: NPM (Node Package Manager)*
- (j) *Version Control System: Git dan Github*
- (k) *Hosting: Vercel*

3.5 Uji Coba dan Evaluasi

Uji coba dilakukan kepada mahasiswa dan pelajar sebagai target utama dari sistem pembelajaran yang dirancang. Evaluasi dilakukan menggunakan metode *HMSAM* dan pengukuran tingkat kepuasan pengguna menggunakan metode *End-User Computing Satisfaction (EUCS)* [36], di mana setiap pertanyaan dinilai menggunakan skala penilaian dari skala Likert. Pertanyaan-pertanyaan dari metode *HMSAM* telah diterjemahkan dan disesuaikan dengan konteks penggunaan *website React Realm*. Rincian pertanyaan yang diajukan kepada responden dapat dilihat pada Tabel 3.2. Tanda * pada *item* menandakan skala dengan nilai terbalik.

U M M N
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

Tabel 3.2. Pertanyaan Evaluasi *React Realm*

Construct	Items
<i>Perceived ease of use</i>	<p>PEOU1. Interaksi ketika menggunakan React Realm jelas dan mudah dimengerti.</p> <p>PEOU2. Berinteraksi dengan React Realm tidak memerlukan banyak usaha mental dari saya.</p> <p>PEOU3. Saya merasa React Realm bebas dari masalah.</p> <p>PEOU4. Saya merasa mudah untuk membuat React Realm melakukan apa yang saya inginkan.</p> <p>PEOU5. Mempelajari cara menggunakan React Realm mudah bagi saya.</p> <p>PEOU6. Mudah untuk melakukan apa yang saya inginkan dengan React Realm.</p> <p>PEOU7. Mudah bagi saya untuk mahir dalam menggunakan React Realm.</p> <p>PEOU8. Saya merasa React Realm mudah untuk digunakan.</p>
<i>Perceived usefulness</i>	<p>PU1. React Realm mengurangi stres saya.</p> <p>PU2. React Realm membantu saya mengisi waktu dengan lebih baik.</p> <p>PU3. React Realm menyediakan pelarian yang bermanfaat.</p> <p>PU4. React Realm membantu saya berpikir lebih jernih.</p> <p>PU5. React Realm membantu saya merasa segar kembali.</p>
<i>Curiosity</i>	<p>CUR1. Pengalaman menggunakan React Realm membangkitkan rasa ingin tahu saya.</p> <p>CUR2. Pengalaman menggunakan React Realm membuat saya penasaran.</p> <p>CUR3. Pengalaman menggunakan React Realm memicu imajinasi saya.</p>
Lanjut ke halaman berikutnya	

Tabel 3.2 Pertanyaan Evaluasi *React Realm* (lanjutan)

<i>Construct</i>	<i>Items</i>
<i>Joy</i>	<p>JOY1. Saya merasa bermain React Realm menyenangkan.</p> <p>JOY2. Saya bersenang-senang saat menggunakan React Realm.</p> <p>*JOY3. Menggunakan React Realm membosankan.</p> <p>*JOY4. React Realm sangat mengganggu saya.</p> <p>JOY5. Pengalaman bermain React Realm dapat dinikmati.</p> <p>*JOY6. React Realm membuat saya tidak puas.</p>
<i>Control</i>	<p>CTL1. Saya memiliki banyak kendali.</p> <p>CTL2. Saya bisa dengan bebas memilih apa yang ingin saya lihat atau lakukan.</p> <p>*CTL3. Saya memiliki sedikit kendali atas apa yang bisa saya lakukan.</p> <p>CTL4. Saya memegang kendali.</p> <p>*CTL5. Saya tidak memiliki kendali atas interaksi saya.</p> <p>CTL6. Saya diizinkan untuk mengontrol interaksi saya.</p>
<i>Behavioral intention to use</i>	<p>BIU1. Saya berencana untuk menggunakan React Realm di masa yang akan datang.</p> <p>BIU2. Saya berniat untuk terus menggunakan React Realm di masa yang akan datang.</p> <p>BIU3. Saya berharap akan terus menggunakan React Realm di masa yang akan datang.</p>
<i>Focused immersion</i>	<p>FI1. Saya dapat terlepas dari distraksi yang ada ketika menggunakan React Realm</p> <p>FI2. Saya terserap kedalam penggunaan React Realm.</p> <p>FI3. Saya terjerumus kedalam penggunaan React Realm.</p> <p>*FI4. Perhatian saya sangat mudah teralihkan oleh hal lain ketika menggunakan React Realm.</p> <p>FI5. Perhatian saya tidak mudah teralihkan ketika menggunakan React Realm</p>

3.6 Dokumentasi

Penulisan laporan penelitian bertujuan untuk mendokumentasikan keseluruhan proses pada setiap tahapan penelitian yang telah dilakukan. Laporan berisikan hasil analisis, landasan teori yang digunakan, pembahasan hasil penelitian dari rumusan masalah, serta kesimpulan dan saran.

