

## BAB 3 METODOLOGI PENELITIAN

### 3.1 Metodologi Penelitian

Metodologi penelitian yang digunakan untuk merancang dan membangun aplikasi adalah sebagai berikut :

1. Studi Literatur

Studi literatur dilakukan dengan mempelajari teori-teori yang mendukung topik penelitian. Tujuan dilakukan studi literatur ini adalah untuk mengkaji lebih dalam teori yang mendukung penelitian yaitu kotlin, metode gamifikasi, *Octalysis*, HMSAM, dan skala likert.

2. Analisis Kebutuhan

Analisis kebutuhan berguna untuk menemukan hal-hal yang dibutuhkan untuk membuat aplikasi berdasarkan studi literatur yang telah dilakukan sebelumnya. Analisa ini juga menggunakan laporan penelitian aplikasi pembelajaran mobile lain dan website serta buku pembelajaran Kotlin sebagai referensi.

3. Perancangan Aplikasi

Tahap perancangan aplikasi dilakukan dengan memperhatikan hasil analisis kebutuhan yang telah dilakukan sebelumnya. Unsur-unsur aplikasi seperti tampilan, alur, aset, diagram, dan lain sebagainya akan dirancang dengan mempertimbangkan metode gamifikasi.

4. Implementasi Aplikasi

Implementasi aplikasi dilakukan dengan merealisasikan aplikasi yang telah dirancang sebelumnya menjadi bentuk nyata yang bisa dipakai oleh pengguna. Pada tahap ini, proses coding akan dilakukan menggunakan Android Studio.

5. Pengujian Aplikasi

Pengujian aplikasi dilakukan setelah implementasi aplikasi selesai. Tujuannya adalah melakukan uji coba terhadap aplikasi yang sebelumnya sudah jadi dengan memberikan kuesioner kepada pengguna berkaitan dengan aplikasi tersebut.

## 6. Evaluasi Pengujian Aplikasi

Setelah data kuesioner berhasil dikumpulkan, dilakukan tahap evaluasi pengujian aplikasi untuk melakukan pengolahan data kuesioner dengan skala likert untuk mengukur tingkat fungsi aplikasi.

## 7. Penyusunan Laporan

Tahap penyusunan laporan dilakukan dengan tujuan mendokumentasi semua tahap dan hasil penelitian yang telah dilakukan secara terstruktur sesuai dengan template laporan penelitian.

### 3.2 Perancangan Aplikasi

Perancangan aplikasi dilakukan dengan merancang sistem aplikasi dengan penerapan metode gamifikasi *Octalysis*, *flowchart*, *mockup* dan aset yang diperlukan dalam pembuatan aplikasi.

#### 3.2.1 Perancangan Gamifikasi

Perancangan gamifikasi dilakukan dengan menentukan elemen yang akan diterapkan dari kedelapan *core drive* gamifikasi *Octalysis*. Elemen-elemen yang akan digunakan adalah sebagai berikut.

##### a. *Epic Meaning & Calling*

Elemen-elemen gamifikasi yang digunakan dari *core drive* ini adalah sebagai berikut:

##### 1. *Narrative*

Elemen *narrative* diterapkan dengan penyampaian narasi cerita yang menceritakan bahwa pengguna baru saja bergabung ke dalam Kotlin Lab sebagai salah satu peneliti antivirus dari virus KT-19. *Narrative* ditampilkan dalam bentuk *alert dialog box* pada saat pertama kali pengguna melakukan login dan setiap pengguna menyelesaikan *stage*.

##### 2. *Free Lunch*

*Free Lunch* diterapkan dengan memberikan *freebies* kepada pengguna baru pada saat pertama kali pengguna melakukan login berupa *item game*, *coin* dan *default avatar*.

b. *Development & Accomplishment*

Elemen-elemen gamifikasi yang digunakan dari *core drive* ini adalah sebagai berikut:

1. *Status Points*

*Status Points* diterapkan dengan sistem *point* pada aplikasi. *Point* bisa didapatkan dari akumulasi penyelesaian tes pada setiap *stage*, *achievement*, dan hadiah *login*. *Point* ini akan digunakan untuk menentukan posisi pengguna pada menu *leaderboard*.

2. *Badges*

*Badges* diterapkan pada fitur *achievement*. Dimana pengguna akan menerima *badges* ketika pengguna telah menyelesaikan *achievement* tertentu.

3. *Leaderboard*

*Leaderboard* diterapkan dengan menampilkan 10 pengguna yang memiliki *point* tertinggi. Pengguna pada posisi pertama, kedua dan ketiga akan ditampilkan pada bagian atas disertai dengan medali.

c. *Empowerment of Creativity & Feedback*

Elemen gamifikasi yang digunakan dari *core drive* ini adalah *Milestone Unlocks* berupa sistem *stage* pada aplikasi. Hanya ada satu dari enam *stage* yang terbuka pada awal aplikasi, untuk membuka *stage* selanjutnya, maka pengguna harus menyelesaikan tes pada *stage* sebelumnya. Untuk setiap *stage* baru yang terbuka, tingkat kesulitan *stage* akan semakin bertambah.

d. *Ownership & Possession*

Elemen-elemen gamifikasi yang digunakan dari *core drive* ini adalah sebagai berikut:

1. *Exchangeable Points*

*Exchangeable Points* diterapkan dengan sistem *coin* dan fitur *shop* pada aplikasi. Pengguna dapat mendapatkan *coin* melalui hadiah *login*, akumulasi hasil tes setiap *stage* dan hadiah *achievement*. Nantinya *coin* ini bisa ditukar dengan *avatar* melalui *shop*.

2. *Avatar*

*Avatar* diterapkan dengan *avatar* yang disediakan melalui *shop*. Pengguna juga bisa melakukan *equip avatar* melalui menu *profile*. Total

*avatar* yang ada adalah 18, yaitu 2 *avatar* gratis pada awal permainan dan 16 *avatar* yang hanya bisa didapatkan melalui pembelian dari shop.

e. *Social Influence & Relatedness*

Elemen-elemen gamifikasi yang digunakan dari *core drive* ini adalah sebagai berikut:

1. *Friending*

*Friending* diterapkan dengan sistem pertemanan, pengguna dapat berteman dengan pengguna lain pada aplikasi. Pengguna bisa melakukan pencarian *username* dari menu *social* dan mengirimkan permintaan pertemanan terhadap pengguna tersebut. Hubungan pertemanan akan terbentuk apabila pengguna lain tersebut menerima permintaan pertemanan.

2. *Social Treasure/Gifting*

*Social Treasure/Gifting* diterapkan dengan pemberian hadiah yang hanya bisa dilakukan antar satu pengguna ke pengguna lainnya yang sudah berteman. Hadiah yang bisa diberikan berupa *beaker*, yaitu item *game* yang berguna saat pengerjaan tes setiap *stage*.

3. *Tout Flags*

*Tout Flags* diterapkan pada fitur *leaderboard*, dimana pengguna dapat melihat prestasi dan medali dari pengguna lain yang memiliki *point* tinggi tanpa pengguna lain harus memberitahu terlebih dahulu.

f. *Scarcity & Impatience*

Elemen-elemen gamifikasi yang digunakan dari *core drive* ini adalah sebagai berikut:

1. *Magnetic Caps*

*Magnetic Caps* diterapkan dengan pembatasan penggunaan item *beaker* untuk mendapatkan petunjuk untuk soal-soal yang muncul saat pengambilan tes setiap *stage*. Pengguna hanya bisa menggunakan 1 *beaker* untuk satu soal tes.

2. *Prize Pacing*

*Prize Pacing* diterapkan dengan pemberian hadiah berupa *beaker*, coin dan *point* pada akhir pengerjaan tes *stage*. Hadiah ini akan di akumulasi

dari jumlah soal yang benar dan waktu yang dibutuhkan pengguna untuk menyelesaikan tes.

3. *Countdown Timer*

*Countdown Timer* diterapkan dengan perhitungan waktu yang diperlukan pengguna untuk menyelesaikan tes dari setiap *stage*. Semakin cepat pengguna menyelesaikan tes, semakin banyak poin dan *coin* yang akan didapat oleh pengguna.

4. *Big Burn*

*Big Burn* diterapkan dengan pemberian harga *avatar* terbagus yang mahal dan salah satu *achievement* yang hanya dapat diraih setelah menyelesaikan semua *stage* aplikasi.

g. *Unpredictability & Curiosity*

Elemen gamifikasi yang digunakan dari *core drive* ini adalah *Random Rewards* berupa fitur hadiah login dan hadiah *achievement* yang akan didapatkan oleh pengguna. Dimana pengguna tidak dapat melihat isi dari hadiah login atau hadiah *achievement* tersebut sehingga rasa penasaran pengguna lebih meningkat.

h. *Loss & Avoidance*

Elemen gamifikasi yang digunakan dari *core drive* ini adalah *Progress Loss* berupa peringkat dan medali pada *leaderboard* yang dapat hilang apabila pengguna tidak mempertahankannya dengan mendapatkan *point* tinggi.

### 3.2.2 Unified Modeling Language (UML)

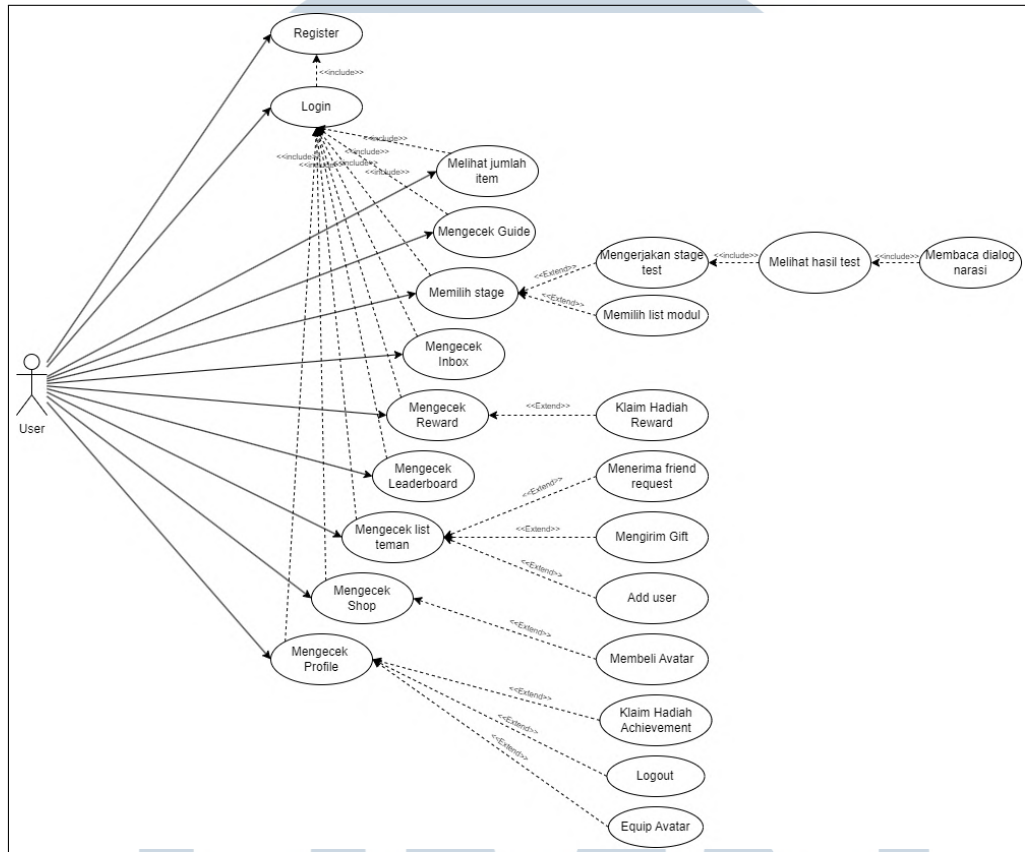
*Unified Modeling Language* (UML) digunakan untuk menggambarkan perancangan alur sistem. Perancangan UML terdiri dari *use case diagram*, *activity diagram* dan *class diagram*.

a. *Use Case Diagram* Sistem

*Use Case Diagram* pada aplikasi ini memiliki satu aktor tunggal yaitu pengguna. Terdapat beberapa fungsi dalam aplikasi yang bisa dilakukan oleh pengguna meliputi *register*, *login*, melihat jumlah *item* aplikasi, mengecek *guide*, memilih *stage*, mengecek *inbox*, mengecek *reward*, mengecek *leaderboard*, mengecek daftar teman, mengecek *shop* dan mengecek *profile*. Penerapan metode gamifikasi *Octalysis* terdapat pada fungsi melihat jumlah



item, memilih stage, mengecek reward, mengecek leaderboard, mengecek list teman, mengecek shop, dan mengecek profile. Use Case Diagram sistem dapat dilihat pada Gambar 3.1.



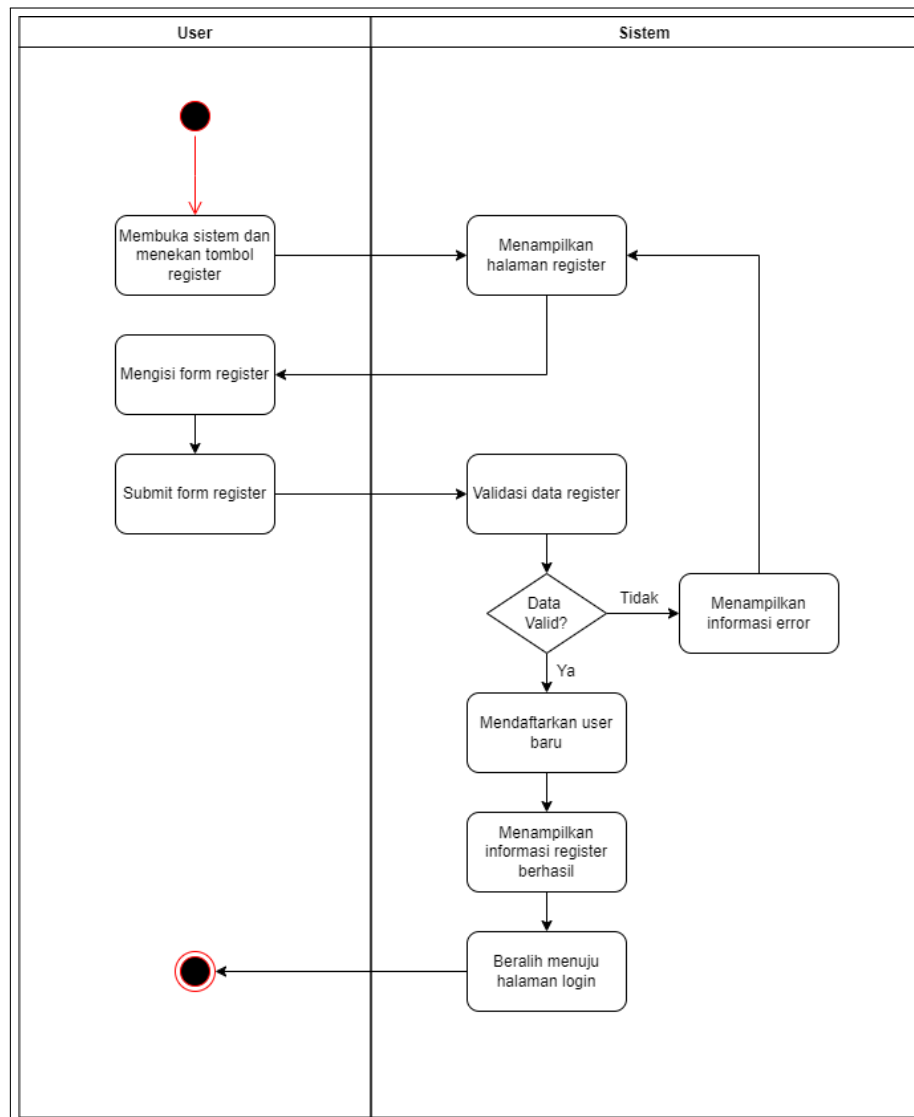
Gambar 3.1. Use Case Diagram

## b. Activity Diagram

Terdapat 11 *activity diagram* dalam perancangan aplikasi pembelajaran bahasa pemrograman Kotlin menggunakan metode gamifikasi. *Activity diagram* berfungsi untuk menjelaskan fungsi yang berada di dalam use case.

### 1. Activity Diagram Register

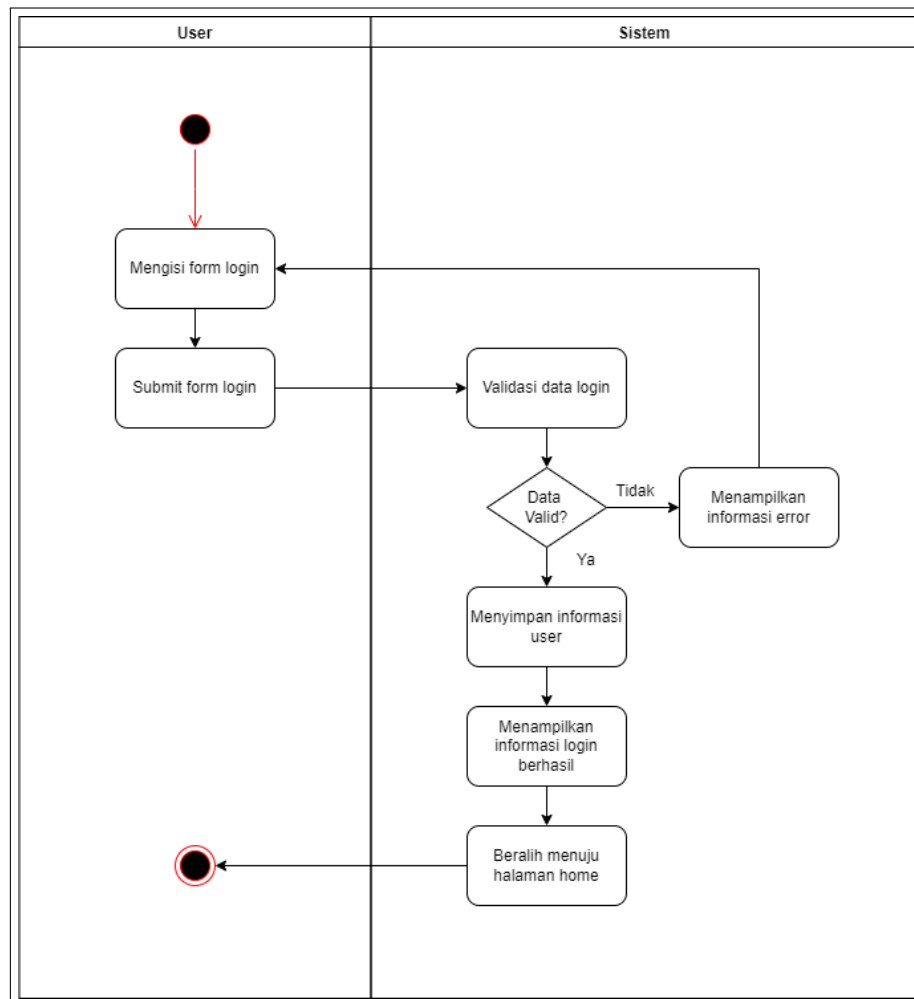
Registrasi pada halaman *register* dimulai dengan mengisi data yang diperlukan sesuai dengan form *register*. Apabila data yang dimasukkan pengguna sesuai dengan kriteria, maka data pengguna akan disimpan pada database dan sistem akan mengarahkan pengguna menuju halaman *login*. Jika data pengguna tidak sesuai kriteria, maka akan ditampilkan informasi *error*. *Activity diagram register* dapat dilihat pada Gambar 3.2.



Gambar 3.2. Activity Diagram Register

## 2. Activity Diagram Login

Setelah melakukan register, pengguna akan diarahkan menuju halaman *login*. Pengguna dapat melakukan *login* dengan memasukkan data yang diperlukan pada form *login*. Setelah itu, sistem akan melakukan validasi pada data yang dimasukkan. Apabila data *login* sesuai dengan data pada database, maka sistem akan menyimpan data pengguna aktif dan pengguna akan diarahkan menuju halaman *home*. Apabila data *login* tidak valid, maka akan ditampilkan informasi *error*. Activity diagram *login* dapat dilihat pada Gambar 3.3.

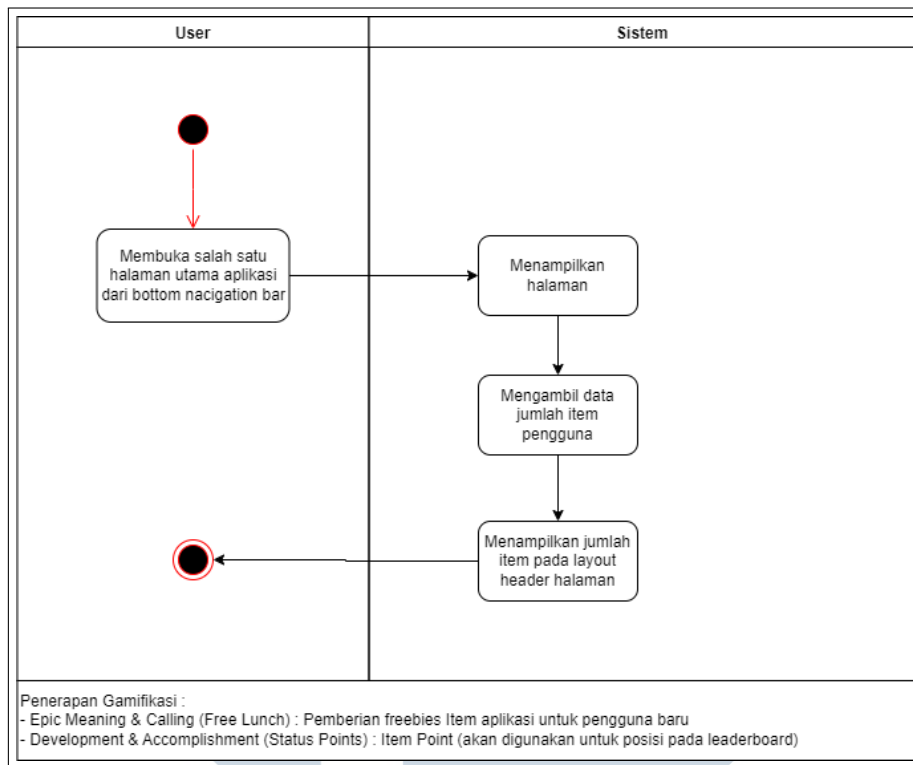


Gambar 3.3. Activity Diagram Login

### 3. Activity Diagram Melihat Jumlah Item

Terdapat beberapa *item* pada aplikasi ini. Apabila pengguna berada pada halaman *home* atau halaman-halaman utama lainnya pada *bottom navigation bar*, maka pengguna dapat melihat jumlah *item* pada bagian *header* halaman aplikasi. Terdapat penerapan *core drives epic meaning & calling* gamifikasi yaitu *free lunch* berupa pemberian *freebies item* aplikasi kepada pengguna baru dan penerapan *core drives development & accomplishment* gamifikasi yaitu *status points* berupa salah satu *item* poin yang digunakan sebagai tolak ukur posisi pengguna pada *leaderboard*. Activity diagram melihat jumlah *item* dapat dilihat pada Gambar 3.4.

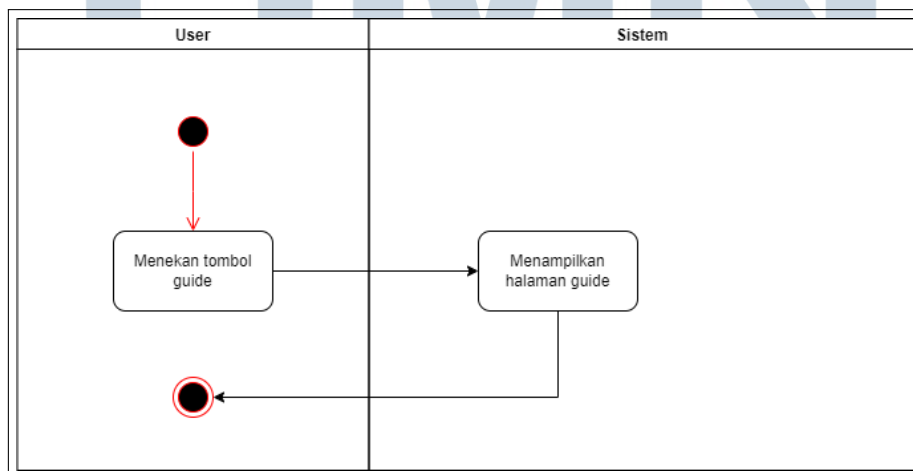




Gambar 3.4. Activity Diagram Melihat Jumlah Item

#### 4. Activity Diagram Mengecek Guide

Halaman *guide* dapat diakses menggunakan tombol pada halaman *home*. Sistem akan menampilkan *guide* berisi informasi dan petunjuk seputar aplikasi. *Activity diagram* mengecek *guide* dapat dilihat pada Gambar 3.5.



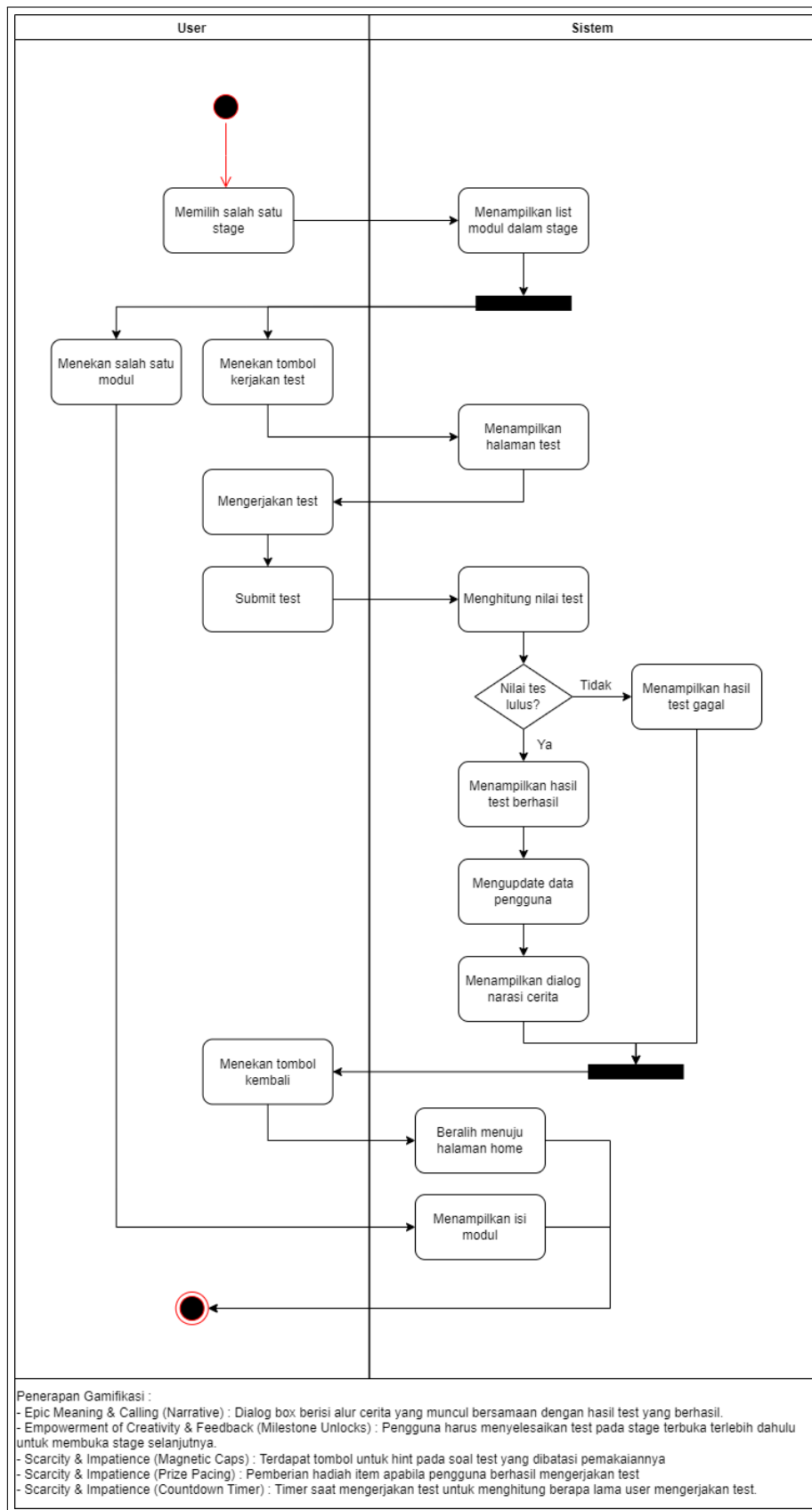
Gambar 3.5. Activity Diagram Mengecek Guide

##### 5. *Activity Diagram* Memilih *Stage*

Terdapat beberapa *stage* pada aplikasi yang dapat dipilih melalui halaman *home*. Setiap *stage* aplikasi akan memuat materi Kotlin dengan kesulitan yang meningkat seiring *stage*. Pengguna dapat memilih dan membaca salah satu modul yang terdaftar pada *stage* atau mengerjakan tes. Bergantung terhadap nilai tes pengguna, sistem akan menampilkan informasi tes gagal apabila nilai tes pengguna tidak lulus atau menampilkan informasi tes berhasil apabila nilai tes pengguna lulus. Sistem juga akan memberikan *item* terhadap pengguna yang dihitung melalui waktu dan jumlah jawaban benar pengguna, sistem juga akan menampilkan dialog berisi narasi jalan cerita aplikasi pada akhir kelulusan tes pengguna.

Terdapat penerapan *core drives epic meaning & calling* yaitu *narrative* berupa dialog narasi membentuk kesatuan cerita yang disampaikan pada akhir kelulusan tes *stage*, penerapan *core drives empowerment of creativity & feedback* yaitu *milestone unlock* berupa pengguna yang harus menyelesaikan *stage* terbuka terlebih dahulu untuk membuka *stage* selanjutnya, dan penerapan *core drives scarcity & impatience* yaitu *magnetic caps* berupa pembatasan pemakaian *hint* saat mengerjakan tes, *prize pacing* berupa pemberian hadiah *item* pada akhir pengerjaan tes dan *countdown timer* berupa timer waktu yang berjalan saat pengguna mengerjakan tes. *Activity diagram* memilih *stage* dapat dilihat pada Gambar 3.6.

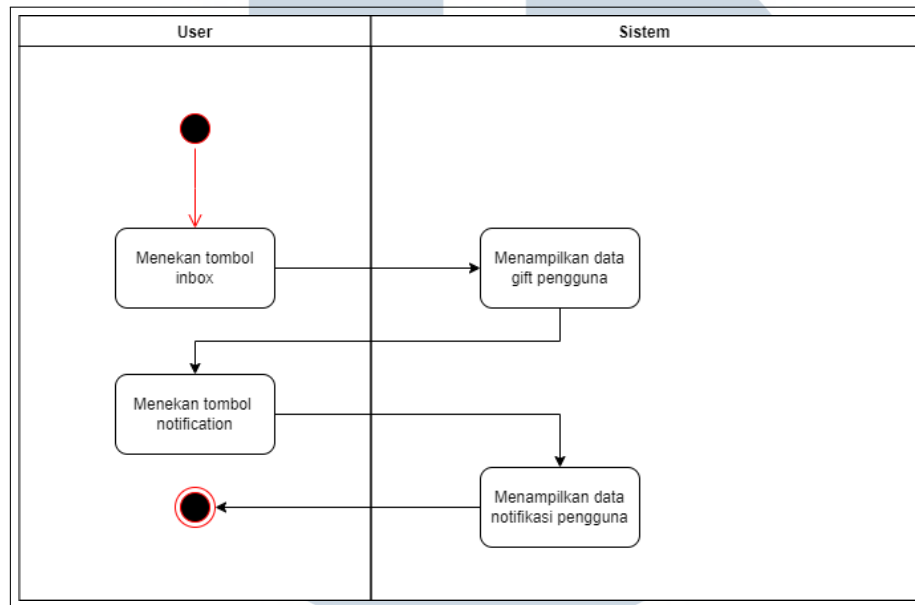
U I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



Gambar 3.6. Activity Diagram Memilih Stage

### 6. Activity Diagram Mengecek Inbox

Halaman *inbox* yang dapat diakses pengguna menampilkan informasi daftar *gift* dan daftar *notification* yang masuk ke akun pengguna. *Activity diagram* mengecek *inbox* dapat dilihat pada Gambar 3.7.

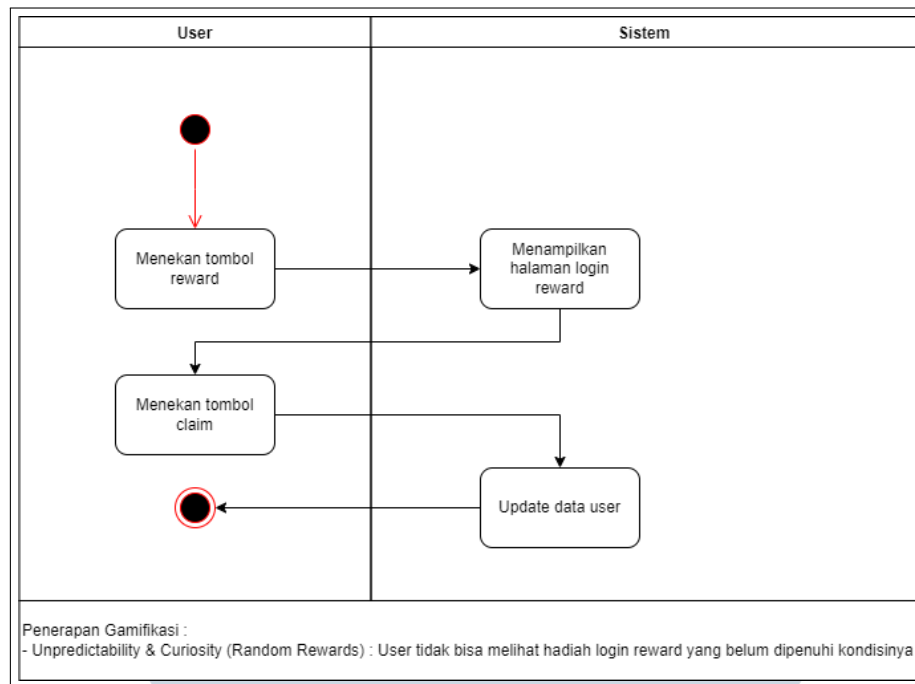


Gambar 3.7. Activity Diagram Mengecek Inbox

### 7. Activity Diagram Mengecek Reward

Halaman *reward* yang dapat diakses pengguna menampilkan informasi daftar hadiah *login reward* yang bisa pengguna *claim*. Terdapat penerapan *core drives unpredictability & curiosity* gamifikasi yaitu *random rewards* berupa daftar hadiah *reward* yang tertutup apabila pengguna belum memenuhi kondisi *login* pada hari tersebut. *Activity diagram* mengecek *reward* dapat dilihat pada Gambar 3.8.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

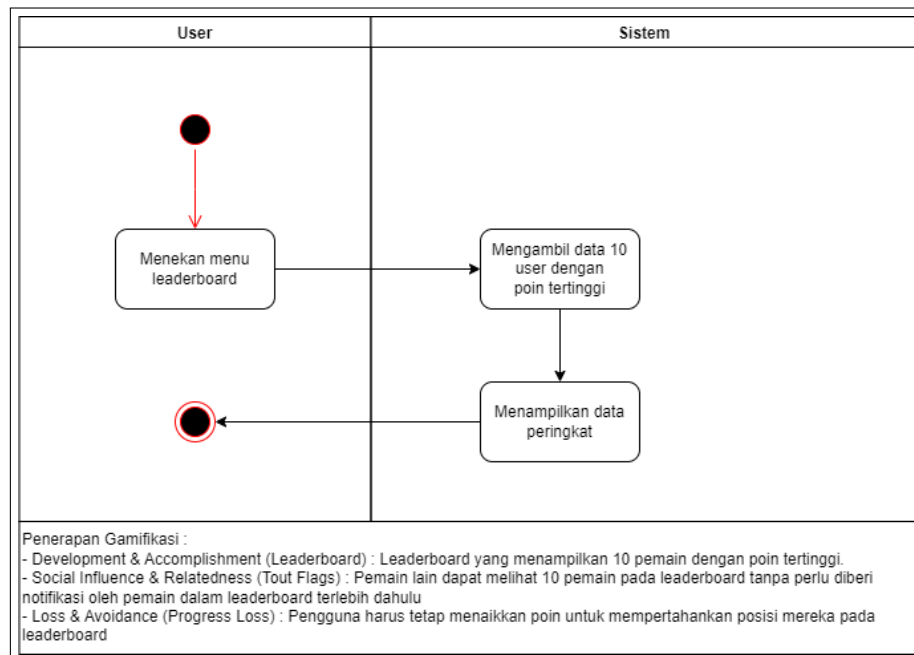


Gambar 3.8. *Activity Diagram Mengecek Reward*

#### 8. *Activity Diagram Mengecek Leaderboard*

Halaman *leaderboard* dapat diakses pengguna melalui *bottom navigation bar* aplikasi, halaman ini menampilkan informasi 10 pengguna dengan poin tertinggi. Terdapat penerapan *core drives development & accomplishment* gamifikasi yaitu *leaderboard*, penerapan *core drives social influence & relatedness* yaitu *tout flags* berupa pengguna lain dapat melihat pengguna pada *leaderboard* tanpa perlu diberi notifikasi oleh pengguna dalam *leaderboard*, dan penerapan *core drives loss & avoidance* yaitu *progress loss* berupa pengguna yang harus menjaga poin untuk mempertahankan posisi mereka pada *leaderboard*. *Activity diagram* mengecek *leaderboard* dapat dilihat pada Gambar 3.9.

UNIVERSITAS  
 MULTIMEDIA  
 NUSANTARA



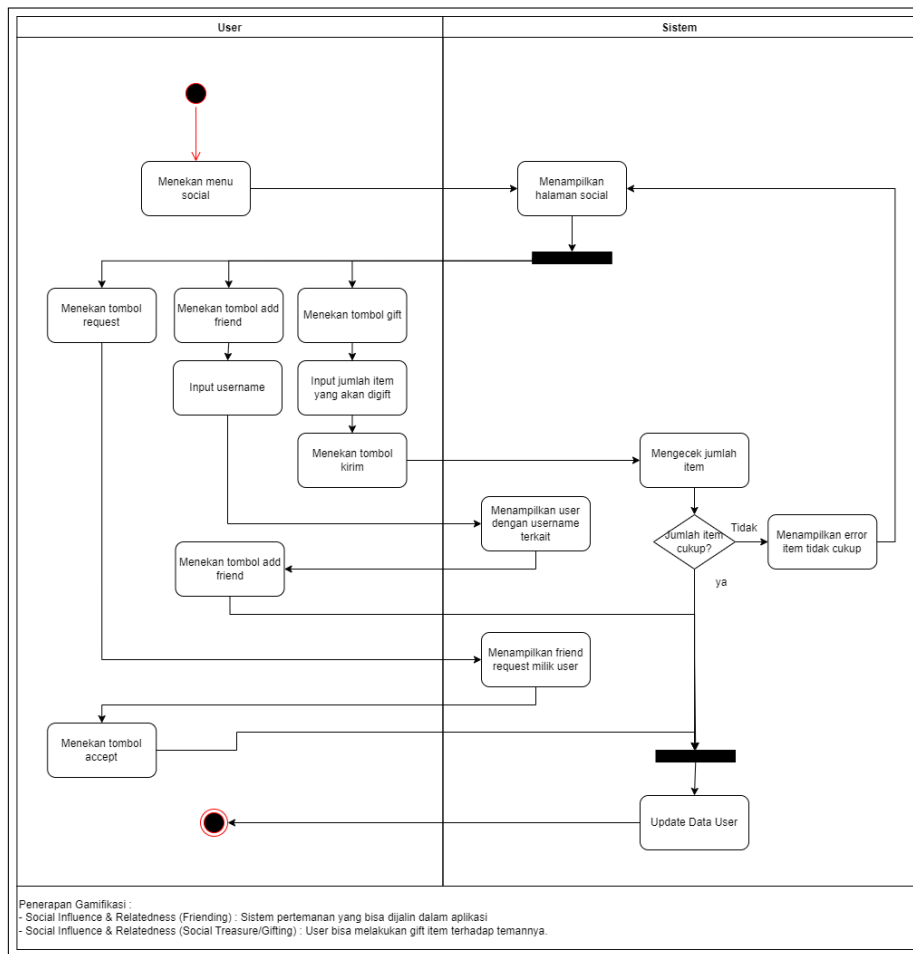
Gambar 3.9. Activity Diagram Mengecek Leaderboard

#### 9. Activity Diagram Mengecek List Teman

Dalam halaman *list* teman, pengguna bisa melakukan beberapa interaksi, yaitu melakukan *gift* terhadap daftar temannya, menerima *friend request* yang masuk dan mencari pengguna lain dan melakukan *add friend* terhadap pengguna tersebut. Terdapat penerapan *core drives social influence & relatedness* gamifikasi yaitu *friending* berupa sistem pertemanan pada aplikasi dan *social treasure/gifting* berupa pengguna yang bisa melakukan *gift* terhadap temannya. Activity diagram mengecek *list* teman dapat dilihat pada Gambar 3.10.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

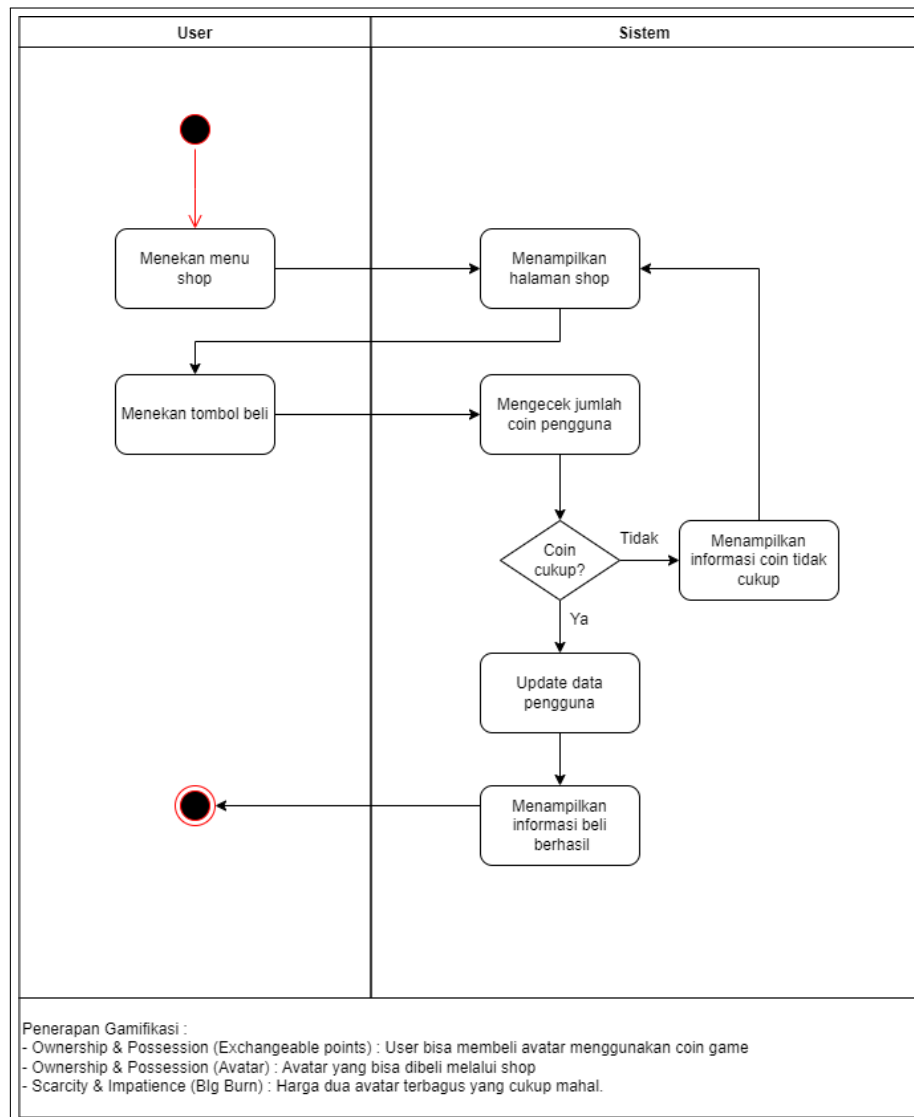




Gambar 3.10. Activity Diagram Mengecek List Teman

#### 10. Activity Diagram Mengecek Shop

Dalam halaman *shop*, pengguna bisa melihat daftar *avatar* yang dijual dan melakukan pembelian terhadap *avatar* tersebut dengan *item* yang dimiliki pengguna. Terdapat penerapan *core drives ownership & possession* yaitu *exchangeable points* berupa *avatar* yang bisa dibeli menggunakan *item* dan *avatar* berupa semua *avatar* yang dijual di *shop* serta penerapan *core drives scarcity & impatience* yaitu *big burn* berupa harga mahal yang diberikan kepada *avatar* paling bagus. Activity diagram mengecek *shop* dapat dilihat pada Gambar 3.11.

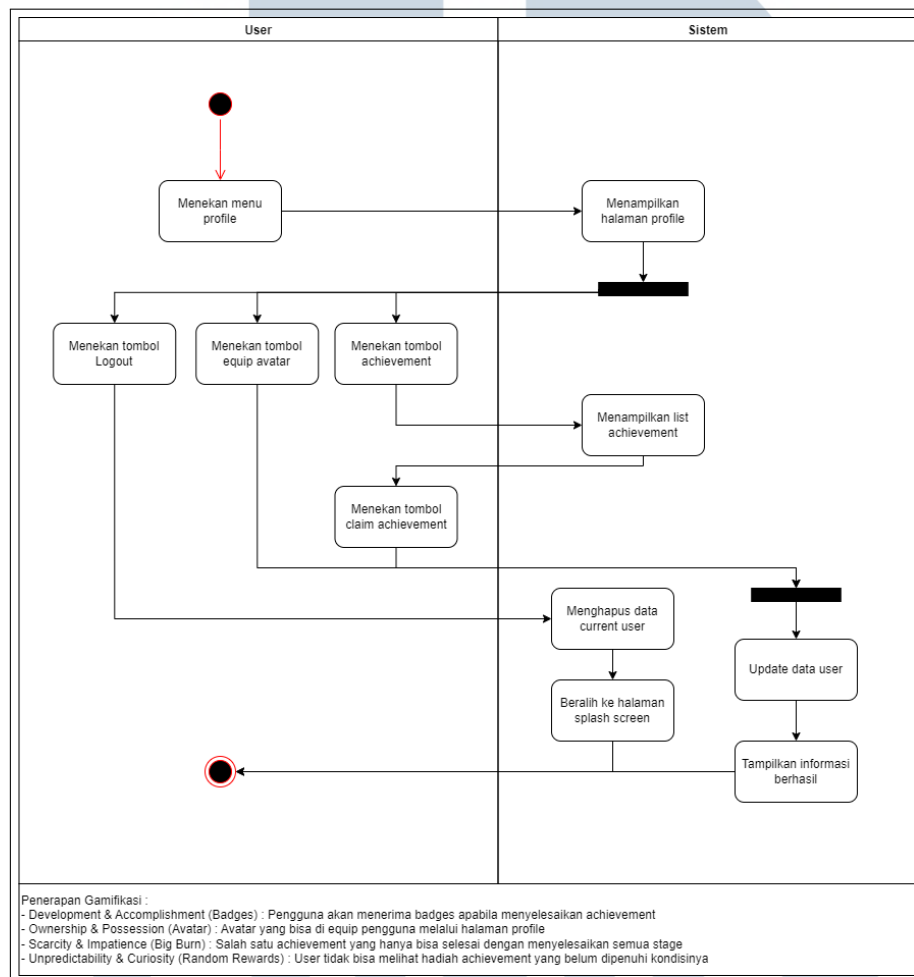


Gambar 3.11. Activity Diagram Mengecek Shop

### 11. Activity Diagram Mengecek Profile

Dalam halaman *profile*, pengguna dapat melakukan *equip* terhadap *avatar* yang dimiliki, melakukan *claim* terhadap hadiah *achievement* yang terpenuhi dan melakukan *logout* dari aplikasi. Terdapat penerapan *core drives development & accomplishment* gamifikasi yaitu *badges* berupa pengguna yang akan menerima *badges* dari *achievement* yang telah diselesaikan, penerapan *core drives ownership & possession* yaitu *avatar* berupa *avatar* milik pengguna yang bisa *diequip* melalui halaman *profile*, penerapan *core drives scarcity & impatience* yaitu *big burn* berupa salah satu *achievement* yang hanya bisa dibuka dengan

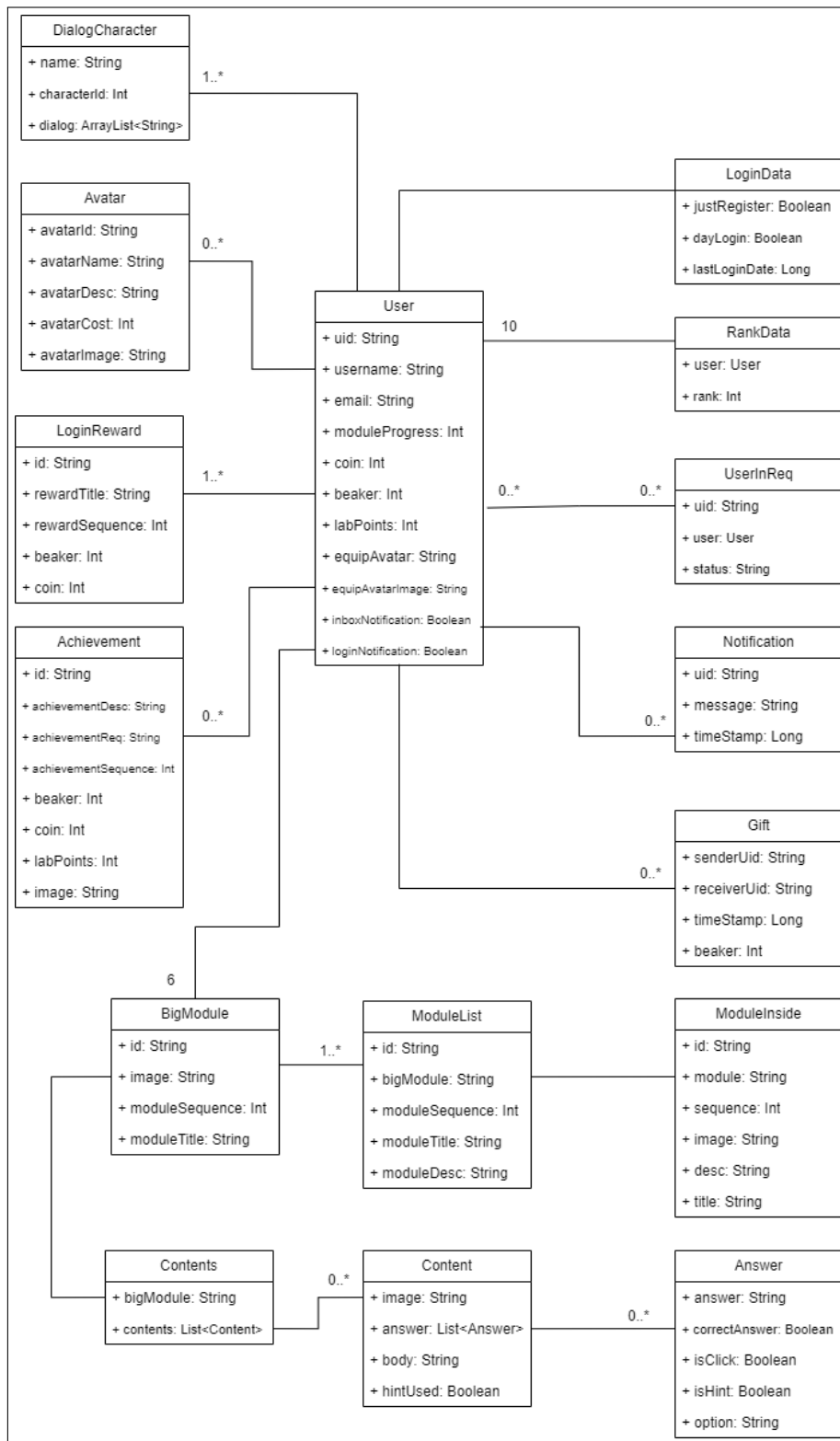
menyelesaikan semua *stage* dan penerapan *core drives unpredictability & curiosity* yaitu *random rewards* berupa pengguna yang tidak bisa melihat hadiah *achievement* yang belum terpenuhi kondisinya. *Activity diagram* mengecek *profile* dapat dilihat pada Gambar 3.12.



Gambar 3.12. Activity Diagram Mengecek Profile

### c. Class Diagram

*Class Diagram* dibuat untuk merancang *class* yang akan digunakan dalam aplikasi. Terdapat beberapa *class* yang digunakan dan sebagian besar *class* berhubungan dengan *class user*. *Class diagram* dapat dilihat pada Gambar 3.13.

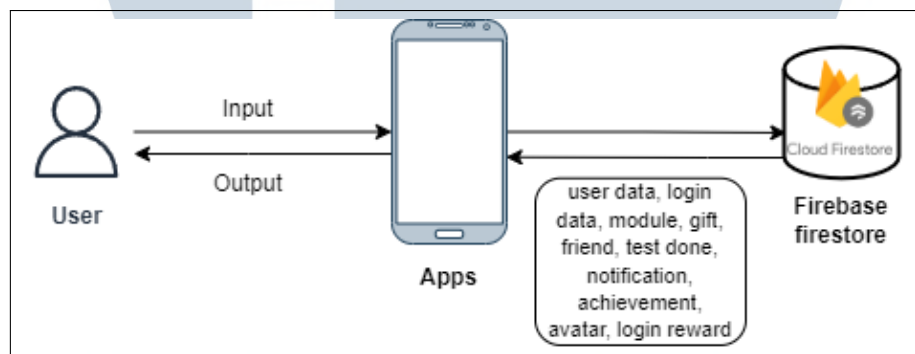


Gambar 3.13. Class Diagram

### 3.2.3 Model Aplikasi

Aplikasi pembelajaran Kotlin ini akan memanfaatkan *Firebase Firestore* sebagai database untuk menyimpan dan menampilkan data-data aplikasi tertentu. Melalui aplikasi ini, pengguna dapat mempelajari materi bahasa pemrograman Kotlin yang tingkat kesusahannya akan meningkat apabila *stage* pengguna semakin tinggi. Untuk melewati *stage*, maka pengguna harus lulus tes yang diberikan setiap *stage*, tes ini akan menguji pengetahuan pengguna akan *stage* yang sedang dihadapi.

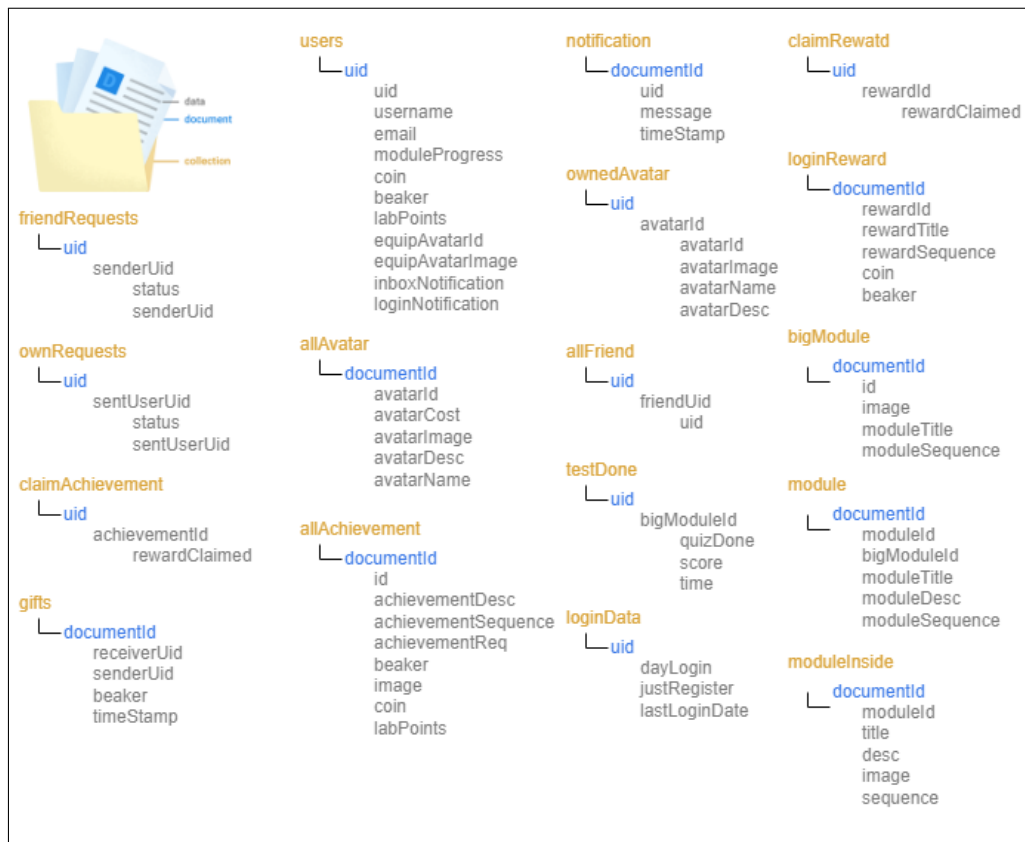
Terdapat juga fitur-fitur lain seperti *friend system*, hadiah login, *achievement*, avatar yang bisa dibeli dan *diequip*, dan lain sebagainya. Fitur-fitur ini dibuat dengan mempertimbangkan *core drive Octalysis* yang ada. Model aplikasi yang digunakan dapat dilihat pada Gambar 3.14.



Gambar 3.14. Model Aplikasi Pembelajaran Bahasa Pemrograman Kotlin

Database aplikasi pembelajaran Kotlin pada *Firebase Firestore* akan terdiri dari 17 *collection*. *Collection users* adalah *collection* utama untuk menyimpan informasi masing-masing akun pengguna. Kebanyakan dari *collection-collection* lain berhubungan dengan *collection users* karena menyimpan informasi mengenai pengguna, contohnya *collection owned avatar* berisi semua data *avatar* yang dimiliki pengguna atau *collection claimed reward* berisi semua data *login reward* yang sudah di *claim* oleh pengguna dan masih banyak lagi. *Collection users* dan *collection* yang berhubungan dengan *users* menggunakan *uid* pengguna untuk memisahkan masing-masing *document* data pengguna. Selain *collection users* dan *collection-collection* yang berhubungan dengan *users*, terdapat juga *collection* untuk menyimpan informasi semua *avatar* yang dijual di *shop*, *collection login reward*, *collection achievement*, dan *collection* modul. Struktur database aplikasi pembelajaran bahasa pemrograman Kotlin dapat dilihat pada Gambar 3.15, tulisan berwarna oranye menandakan *collection*, tulisan berwarna biru menandakan

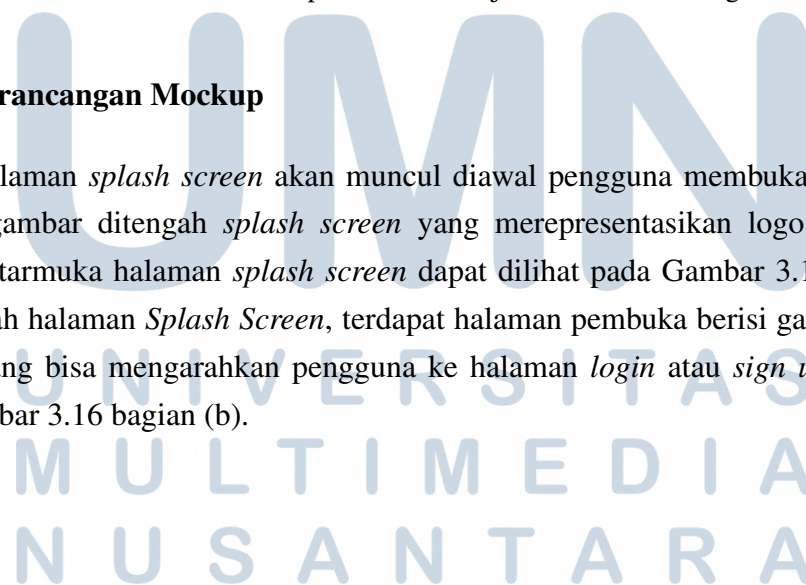
*document* dan tulisan warna abu menandakan isi data *document*.



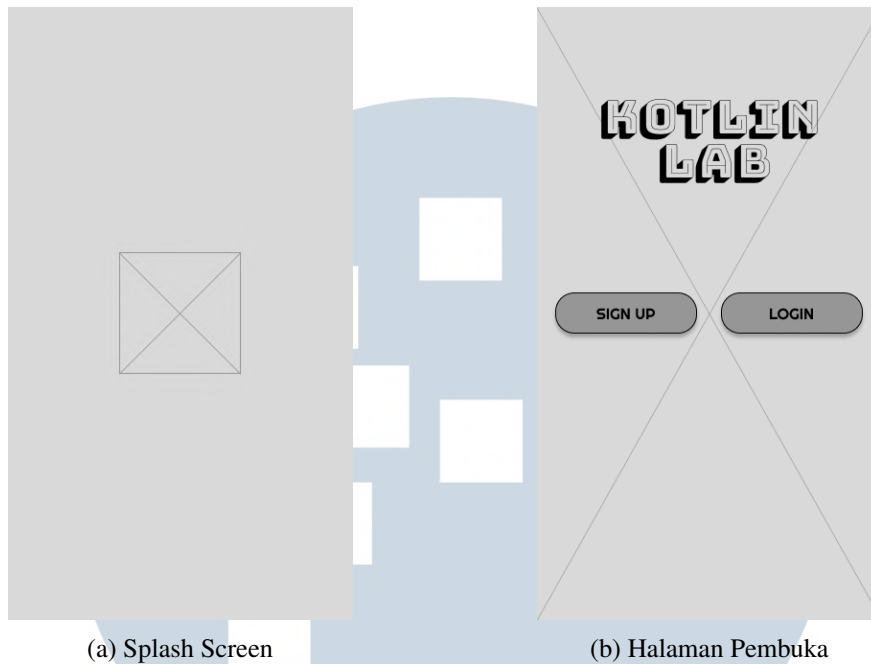
Gambar 3.15. Struktur Database Aplikasi Pembelajaran Bahasa Pemrograman Kotlin

### 3.2.4 Perancangan Mockup

Halaman *splash screen* akan muncul diawal pengguna membuka aplikasi, terdapat gambar ditengah *splash screen* yang merepresentasikan logo aplikasi. Desain antarmuka halaman *splash screen* dapat dilihat pada Gambar 3.16 bagian (a). Setelah halaman *Splash Screen*, terdapat halaman pembuka berisi gambar dan tombol yang bisa mengarahkan pengguna ke halaman *login* atau *sign up* seperti pada Gambar 3.16 bagian (b).

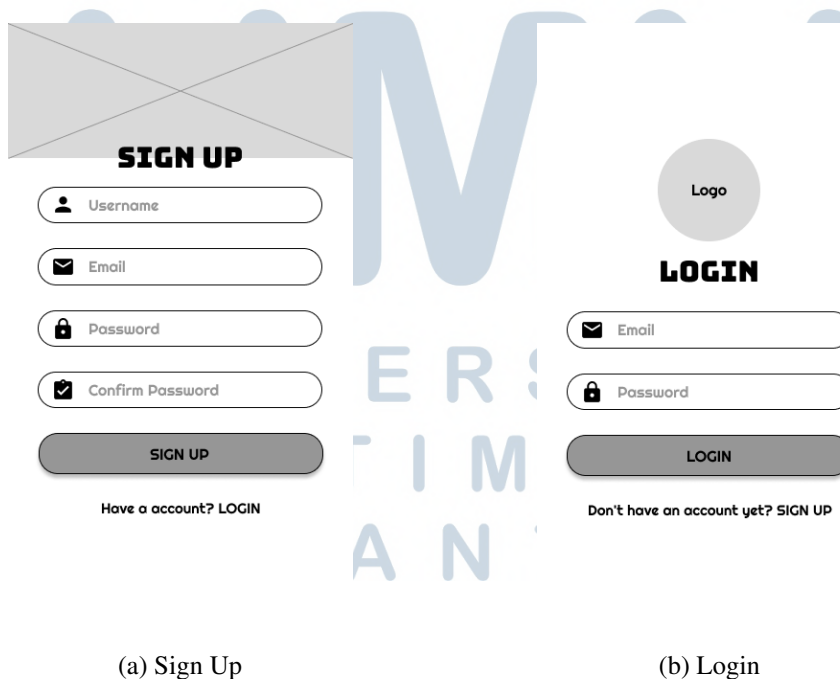






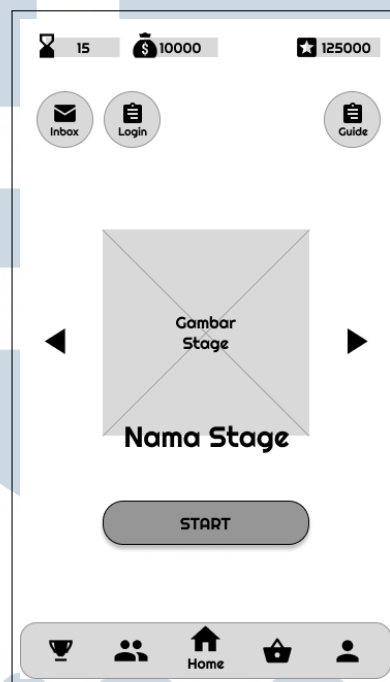
Gambar 3.16. Desain Antarmuka Pembuka Aplikasi

Desain antarmuka halaman *sign up* dapat dilihat pada Gambar 3.17 bagian (a), terdapat gambar pada bagian atas diantara tulisan *sign up* sebagai penghias. Desain antarmuka halaman *login* dapat dilihat pada Gambar 3.17 bagian (b), akan ditampilkan gambar di atas tulisan *login*.



Gambar 3.17. Desain Antarmuka *Sign Up* dan *Login* Aplikasi

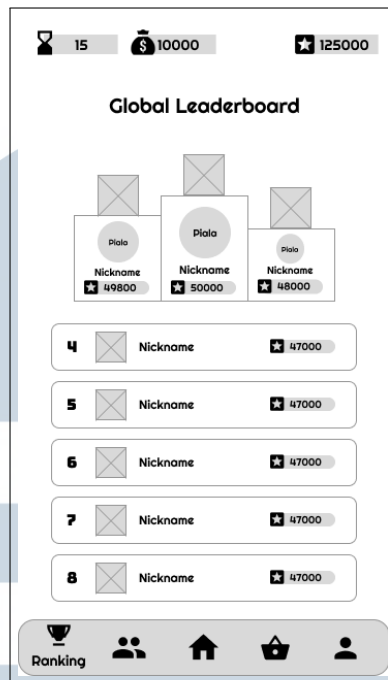
Desain antarmuka halaman *home* aplikasi dapat dilihat pada Gambar 3.18. Terdapat *bottom navigation bar* pada bagian bawah halaman dengan 5 pilihan menu yaitu *leaderboard*, *social*, *home*, *shop* dan *profile*. *Bottom navigation bar* ini akan muncul pada kelima halaman menu utama tersebut. Pada bagian atas terdapat jumlah *item beaker*, *coin* dan *point*. Terdapat juga tombol menuju *inbox*, *login rewards*, dan *guide* dan pengguna bisa memilih *stage* yang terbuka.



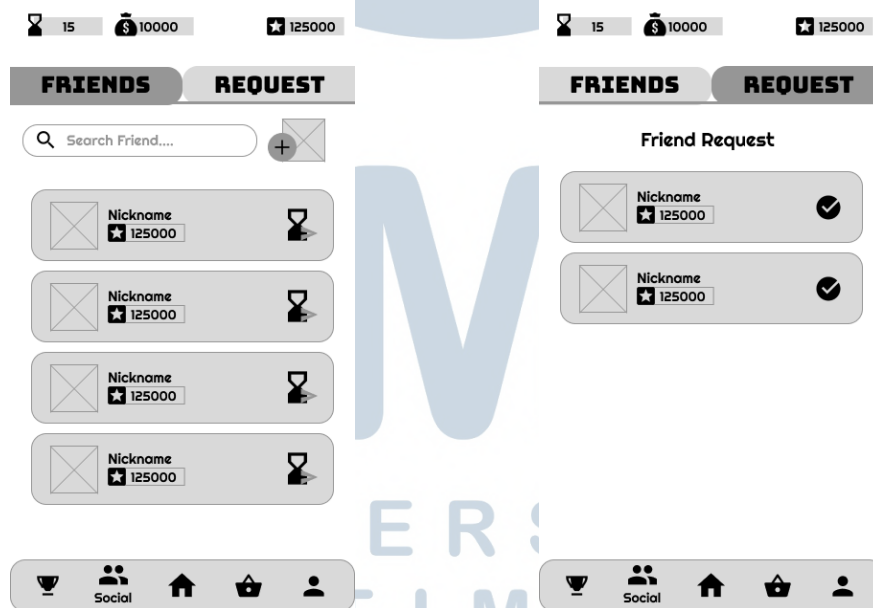
Gambar 3.18. Desain Antarmuka Home

Desain antarmuka halaman *leaderboard* aplikasi dapat dilihat pada Gambar 3.19. Akan ditampilkan 10 pengguna dengan peringkat tertinggi, dengan 3 peringkat tertinggi dibagian atas dengan gambar medali. Sedangkan sisa peringkat lainnya akan ditampilkan dalam bentuk *list* di bawah peringkat 3 besar.

Desain antarmuka halaman *social (friend)* dapat dilihat pada Gambar 3.20 bagian (a), akan ditampilkan list *friend* pengguna dengan tombol gambar *gift* yang akan berfungsi untuk memberi teman pengguna hadiah. Desain antarmuka halaman *social (friend request)* dapat dilihat pada Gambar 3.20 bagian (b), akan ditampilkan *friend request* yang ditujukan pada pengguna dan terdapat tombol gambar untuk menerima *friend request* tersebut.



Gambar 3.19. Desain Antarmuka Leaderboard



(a) Desain Antarmuka Social (Friend)

(b) Desain Antarmuka Social (Friend Request)

Gambar 3.20. Desain Antarmuka *Social* Aplikasi

Desain antarmuka halaman *shop* dapat dilihat pada Gambar 3.21, akan ditampilkan gambar, nama, deskripsi dan harga *avatar* dalam bentuk *list*.



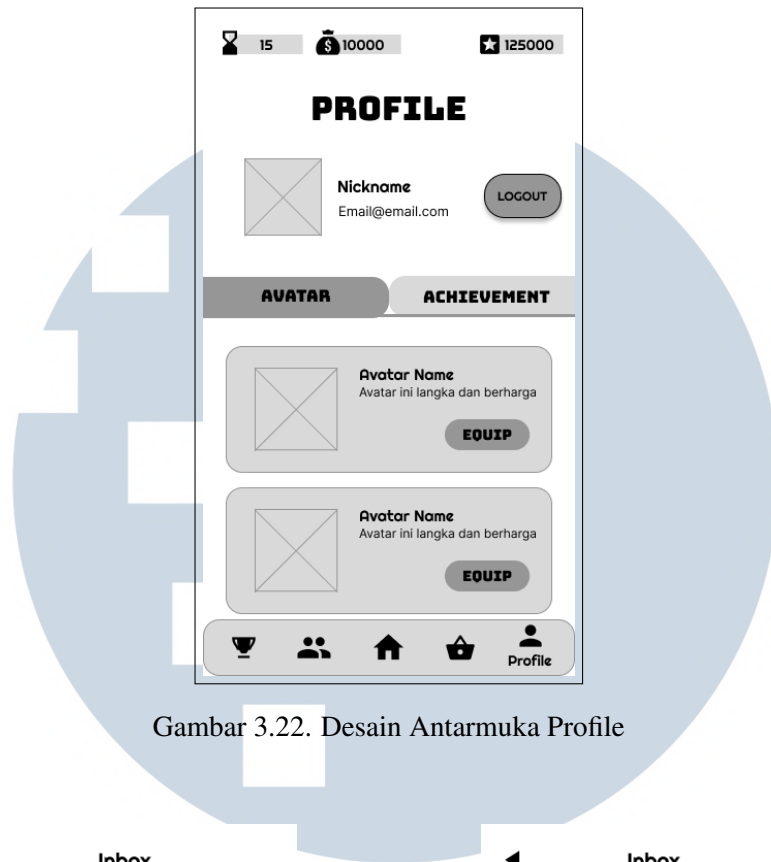
Gambar 3.21. Desain Antarmuka Shop

Desain antarmuka halaman *profile* dapat dilihat pada Gambar 3.22, akan ditampilkan *avatar*, nama, dan *email* pengguna. Di bawahnya terdapat dua *tab layout*, yaitu *avatar* dan *achievement*. *Tab layout* avatar akan menampilkan *avatar* yang dimiliki pengguna dan *avatar* bisa di *equip* melalui *tab avatar*. *Tab layout achievement* akan berisi daftar *list achievement* yang bisa di *claim* oleh pengguna.

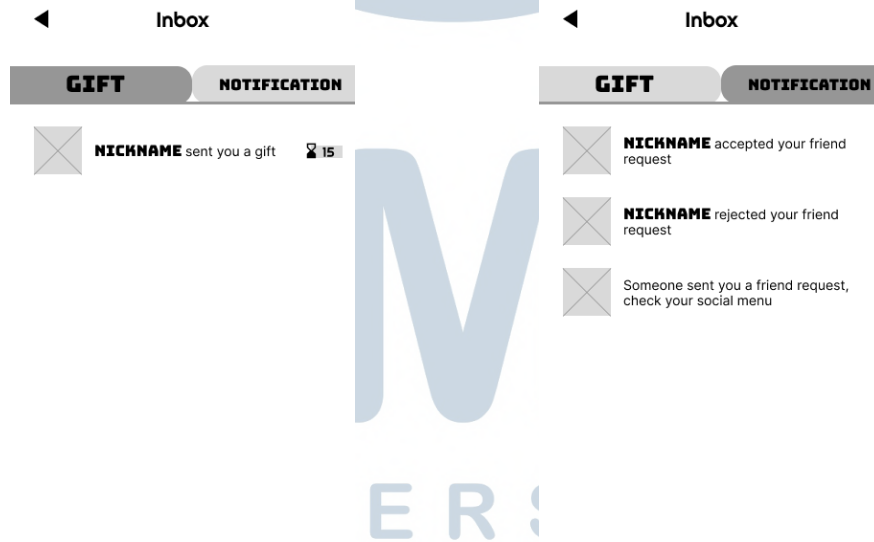
Desain antarmuka halaman *inbox (gift)* dapat dilihat pada Gambar 3.23 bagian (a). Desain antarmuka halaman *inbox (notification)* dapat dilihat pada Gambar 3.23 bagian (b). Keduanya akan ditampilkan dalam bentuk list.

Desain antarmuka halaman *login reward* dapat dilihat pada Gambar 3.24. Akan ditampilkan *list login reward*, apabila *reward* belum terbuka maka isi *reward* akan disembunyikan terlebih dahulu. Apabila *reward* sudah terbuka, maka akan ada gambar jumlah *reward* dan tombol *claim*.

Desain antarmuka halaman *stage* dapat dilihat pada Gambar 3.25 bagian (a), akan ditampilkan *list* isi modul yang diajarkan pada *stage* ini. Desain antarmuka halaman modul *stage* dapat dilihat pada Gambar 3.25 bagian (b), akan ditampilkan isi materi dalam bentuk tulisan dan gambar.



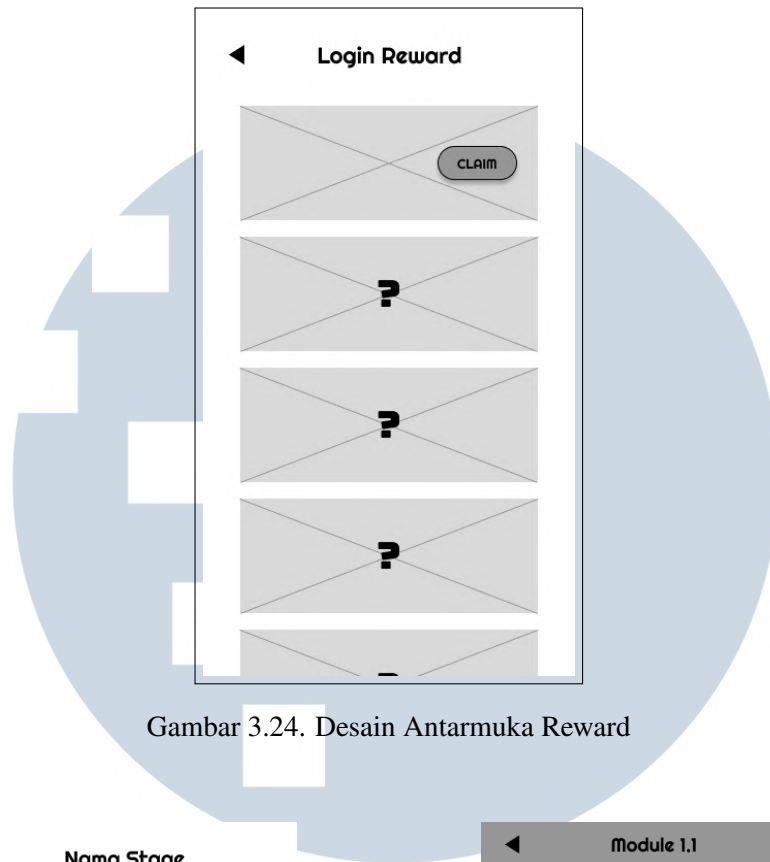
Gambar 3.22. Desain Antarmuka Profile



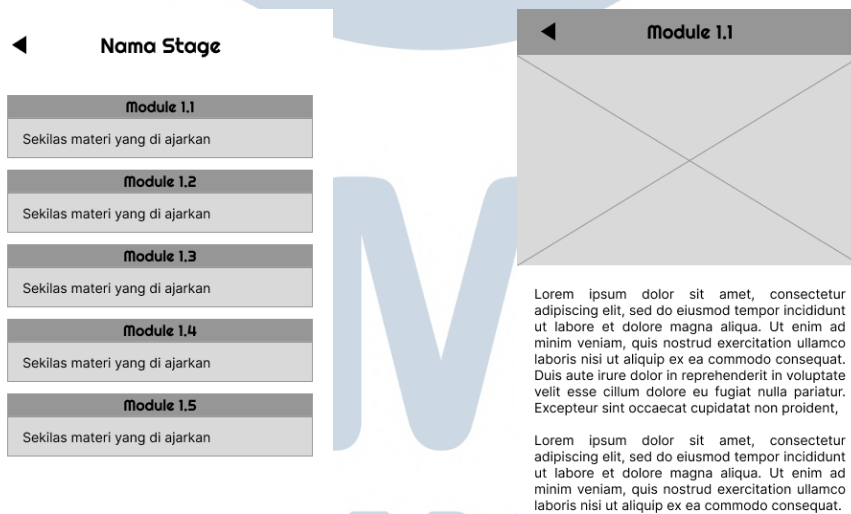
(a) Desain Antarmuka Inbox (Gift)

(b) Desain Antarmuka Inbox (Notification)

Gambar 3.23. Desain Antarmuka Inbox



Gambar 3.24. Desain Antarmuka Reward



(a) Desain Antarmuka Stage

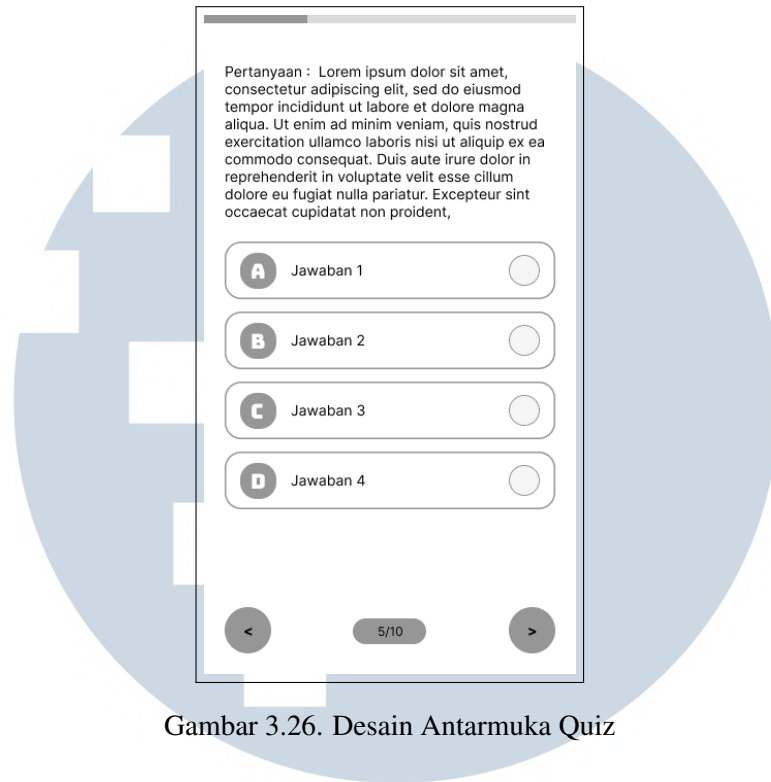
(b) Desain Antarmuka Modul Stage

Gambar 3.25. Desain Antarmuka Stage dan Modul Stage Aplikasi

Desain antarmuka halaman *quiz* dapat dilihat pada Gambar 3.26, akan terdapat *progress bar* pada atas halaman, dengan penampilan soal dalam bentuk tulisan dan beberapa pilihan jawaban, pergantian halaman dilakukan melalui kedua



anak panah pada bagian bawah halaman.

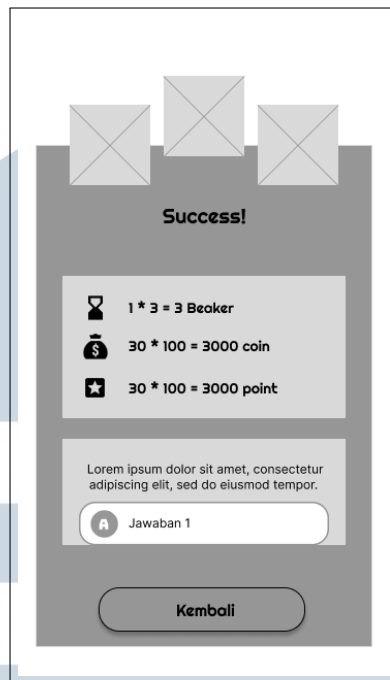


Gambar 3.26. Desain Antarmuka Quiz

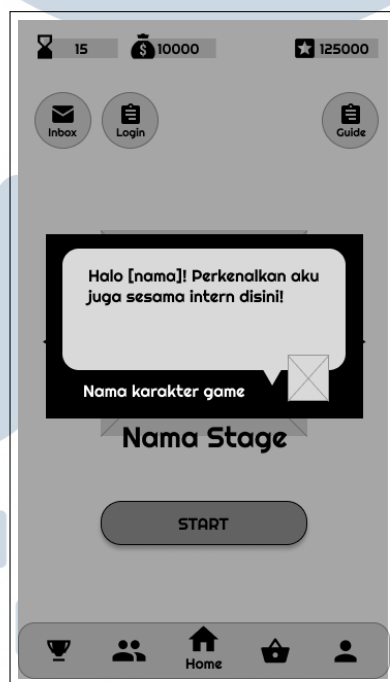
Desain antarmuka halaman *score* dapat dilihat pada Gambar 3.27, akan terdapat gambar bintang sesuai dengan nilai tes pengguna dan tulisan akumulasi *item beaker, coin* dan *point* yang didapatkan pengguna. Terdapat juga *list* jawaban tes yang benar dengan tombol kembali dibawahnya.

Desain antarmuka penampilan *dialog* pada aplikasi dapat dilihat pada Gambar 3.28. Penampilan *dialog* bertujuan untuk jalan cerita yang akan dibawakan dalam aplikasi, penampilan ini disertai dengan *dialog*, gambar dan nama tokoh yang ada di dalam aplikasi.

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 3.27. Desain Antarmuka Score

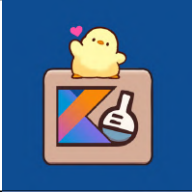

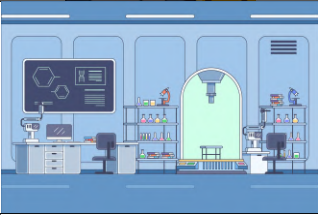





Gambar 3.28. Desain Antarmuka Dialog

### 3.2.5 Perancangan Aset










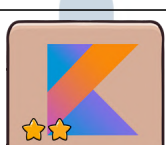
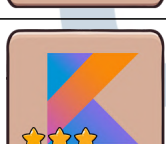
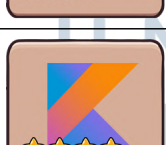
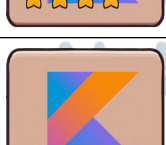
Berikut merupakan daftar aset yang akan digunakan dalam aplikasi, aset ini bertujuan untuk mempercantik tampilan aplikasi.

Tabel 3.1. Daftar Aset Aplikasi

Gambar	Penjelasan	Sumber
	Logo Aplikasi	Aset pribadi
	Background halaman pembuka aplikasi	Aset pribadi
	Background gambar auth, home, shop dan leaderboard	www.freepik.com
	Icon Bottom Navigation Bar	www.icons8.com
	Background halaman isi stage	www.freepik.com
	Icon Item Beaker	penzilla.itch.io



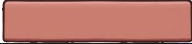








Lanjut pada halaman berikutnya

Tabel 3.1 Daftar Aset Aplikasi (lanjutan)

Gambar	Penjelasan	Sumber
	<i>Icon Item Point</i>	penzilla.itch.io
	<i>Icon Item Coin</i>	www.gamedevmarket.net
	<i>Icon Menu Inbox</i>	penzilla.itch.io
	<i>Icon Menu Login Rewards</i>	penzilla.itch.io
	<i>Icon Menu Guide</i>	penzilla.itch.io
	<i>Icon stage yang belum terbuka</i>	penzilla.itch.io
	<i>Icon tombol berpindah stage</i>	penzilla.itch.io
	<i>Icon tombol berpindah stage</i>	penzilla.itch.io
	Gambar <i>stage</i> pertama	Aset pribadi
	Gambar <i>stage</i> kedua	Aset pribadi
	Gambar <i>stage</i> ketiga	Aset pribadi
	Gambar <i>stage</i> keempat	Aset pribadi
	Gambar <i>stage</i> kelima	Aset pribadi











Lanjut pada halaman berikutnya

Tabel 3.1 Daftar Aset Aplikasi (lanjutan)

Gambar	Penjelasan	Sumber
	Gambar <i>stage</i> keenam	Aset pribadi
	<i>Icon list notification</i>	www.icons8.com
	<i>Background item dan text</i>	penzilla.itch.io
	<i>Background item dan text</i>	penzilla.itch.io
	<i>Background item kosong</i>	Aset pribadi
	<i>Icon tombol kembali</i>	penzilla.itch.io
	<i>Background bubble chat dialog karakter</i>	penzilla.itch.io
	<i>Icon karakter asisten Lia</i>	Aset pribadi
	<i>Icon karakter prof Chic</i>	Aset pribadi
	<i>Icon medali peringkat pertama</i>	www.pngtree.com
	<i>Icon medali peringkat kedua</i>	www.pngtree.com

Lanjut pada halaman berikutnya

Tabel 3.1 Daftar Aset Aplikasi (lanjutan)

Gambar	Penjelasan	Sumber
	<i>Icon medali peringkat ketiga</i>	www.pngtree.com
	<i>Icon add friend</i>	Aset pribadi
	<i>Icon gift friend</i>	penzilla.itch.io
	<i>Icon accept friend request</i>	www.gamedevmarket.net
	<i>Avatar default male</i>	Aset pribadi
	<i>Avatar default female</i>	Aset pribadi
	<i>Avatar basic male</i>	Aset pribadi
	<i>Avatar basic female</i>	Aset pribadi
	<i>Avatar cute male</i>	Aset pribadi
	<i>Avatar cute female</i>	Aset pribadi
Lanjut pada halaman berikutnya		



Tabel 3.1 Daftar Aset Aplikasi (lanjutan)

Gambar	Penjelasan	Sumber
	<i>Avatar boss male</i>	Aset pribadi
	<i>Avatar boss female</i>	Aset pribadi
	<i>Avatar hipster male</i>	Aset pribadi
	<i>Avatar hipster female</i>	Aset pribadi
	<i>Avatar formal male</i>	Aset pribadi
	<i>Avatar formal female</i>	Aset pribadi
	<i>Avatar kimono male</i>	Aset pribadi
	<i>Avatar kimono female</i>	Aset pribadi
	<i>Avatar lab male</i>	Aset pribadi

Lanjut pada halaman berikutnya

Tabel 3.1 Daftar Aset Aplikasi (lanjutan)

Gambar	Penjelasan	Sumber
	<p><i>Avatar lab female</i></p>	<p>Aset pribadi</p>
	<p><i>Avatar plague male</i></p>	<p>Aset pribadi</p>
	<p><i>Avatar plague female</i></p>	<p>Aset pribadi</p>
	<p><i>Icon tombol hint</i></p>	<p>Pengeditan dari aset <a href="https://penzilla.itch.io">penzilla.itch.io</a></p>
	<p><i>Icon tombol log out</i></p>	<p><a href="https://penzilla.itch.io">penzilla.itch.io</a></p>
	<p>Gambar hasil tes gagal</p>	<p>Aset pribadi</p>
	<p>Gambar hasil tes bintang satu</p>	<p>Aset pribadi</p>
	<p>Gambar hasil tes bintang dua</p>	<p>Aset pribadi</p>
	<p>Gambar hasil tes bintang tiga</p>	<p>Aset pribadi</p>