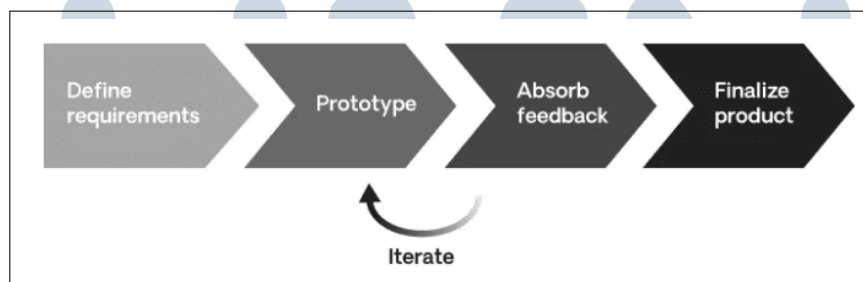


BAB 3 METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Metodologi penelitian yang digunakan adalah *Rapid Application Development*. Metode ini adalah metode yang dimana pengembangan sebuah sistem dilakukan dengan waktu yang cepat yang dimana pengembangan suatu sistem normal dilakukan dengan waktu minimal 180 hari sedangkan metode *Rapid Application Development* dilakukan hanya dengan waktu 30 – 90 hari saja. Metode *Rapid Application Development* menghasilkan suatu sistem dengan cepat serta dapat memenuhi keinginan dari pengguna karena selama metode pengembangan ini dijalankan, pengguna menjadi bagian dari sistem dan segala keputusan mengacu kepada pengguna sehingga waktu pengembangan dapat berjalan lebih cepat karena tidak harus menunggu *feedback* dari pengguna [21]. Dengan terlibatnya pengguna secara langsung selama pengembangan juga memperkecil persentase kesalahan dan ketidaksesuaian aplikasi dengan ekspektasi pengguna [22]. Terdapat beberapa tahap pada metode *Rapid Application Development*, yaitu *Define Requirements*, *Prototype and Absorb Feedback* dan *Finalize Product* [23] seperti yang ditunjukkan pada gambar 3.1.



Gambar 3.1. Bagan *Rapid Application Development*

Sumber: [23]

Detail dari bagan *rapid application development* akan dijelaskan sebagai berikut:

1. *Define Requirements*

Merupakan tahap dimana pengembang aplikasi dan pengguna melakukan identifikasi kebutuhan yang diperlukan pada sebuah aplikasi yang akan dikembangkan dan memahami informasi kebutuhan pengguna terhadap

aplikasi [21] [23]. Pada penelitian ini, kebutuhan yang perlu disiapkan adalah sumber materi dasar terkait *framework* Springboot yang didapat melalui *website* spring.io dan www.baeldung.com. Setelah mendapat sumber materi, akan dilanjutkan dengan proses pembuatan alur aplikasi menggunakan flowchart dan pembuatan alur pergerakan data pada sistem menggunakan *Data Flow Diagram* atau DFD.

(a) Informasi Materi Pembelajaran

Materi pembelajaran aplikasi diambil berdasarkan *website* spring.io dan www.baeldung.com yang merupakan *website* pendukung terbesar untuk *framework* Spring Boot. Materi pembelajaran yang disediakan oleh aplikasi ini hanya terbatas pada materi - materi dasar sebagai pondasi untuk pengembangan aplikasi menggunakan *framework* Spring Boot. Materi pembelajaran dibagi menjadi 4 bab, diantaranya adalah Perkenalan Spring Boot, Memulai dengan Spring Boot, Database dan Spring Boot, dan CRUD pada Spring Boot. Detail dari masing - masing bab adalah sebagai berikut.

i. Perkenalan Spring Boot

Bab ini memiliki 3 buah materi dan pertanyaan terkait dasar-dasar Springboot mulai dari apa itu Springboot, bahasa pemrograman yang didukung, penggunaan springboot pada pengembangan microservices dan kelebihan penggunaan framework Springboot dibandingkan framework lain.

ii. Memulai dengan Spring Boot

Bab ini memiliki 3 buah materi dan pertanyaan terkait Springboot mulai dari dependency pada Springboot, lokasi pengelolaan dependency pada framework Springboot, penggunaan annotation atau anotasi untuk memudahkan developer dalam pengembangan aplikasi serta bagaimana membuat dan menghasilkan sebuah framework Springboot yang siap digunakan.

iii. Database dan Spring Boot

Bab ini memiliki 4 buah materi dan pertanyaan terkait hubungan antara framework Springboot dan database. Materi akan dimulai dari menyiapkan dependency terkait database pada Springboot, melakukan konfigurasi awal untuk koneksi ke database, membuat class Entity untuk mengelola data yang terhubung dengan database,

dan membuat repository class yang merupakan perantara antara entity class dan service class.

iv. CRUD pada Spring Boot

Bab ini memiliki 4 buah materi dan pertanyaan terkait penerapan metode CRUD (Create, Read, Update, Delete) pada database menggunakan framework Springboot. Materi akan menjelaskan terkait penggunaan fungsi otomatis dari class repository, dan penggunaan anotasi yang dibutuhkan saat ingin menghubungkan repository dan service.

(b) Cara Penggunaan Aplikasi

Penerapan gamifikasi pada aplikasi berarti menerapkan elemen *game* pada aplikasi yang dimana sebuah *game* memiliki cara bermain masing - masing seperti halnya aplikasi gamifikasi. Cara penggunaan aplikasi gamifikasi ini akan dijelaskan sebagai berikut:

i. Level

A. Aplikasi memiliki 4 buah level (chapters) yang memiliki tingkat kesulitan yang berbeda-beda yang dimana semakin tinggi level (chapters) maka semakin tinggi pula tingkat kesulitan yang akan dihadapi.

B. Masing - masing level (chapters) harus dilewati secara berurutan karena masing - masing level memiliki keterkaitan satu sama lain.

C. Chapter 1 terdiri dari 3 buah soal, chapter 2 terdiri dari 3 buah soal, chapter 3 terdiri dari 4 buah soal, dan chapter 4 terdiri dari 4 buah soal.

D. *Reward* yang didapat pada setiap level (chapters) berbeda.

ii. Kesempatan

A. Setelah pengguna membuat akun melalui halaman "Register", masing - masing pengguna akan mendapatkan 3 buah kesempatan untuk menjawab soal dengan salah.

B. Setiap 1 soal yang dijawab dengan salah, maka kesempatan pengguna akan berkurang sebanyak 1.

C. Kesempatan dapat ditambah dengan cara membeli baik pada halaman pembelajaran maupun halaman profil pengguna dengan membayar seharga 20 gold untuk 1 buah kesempatan.

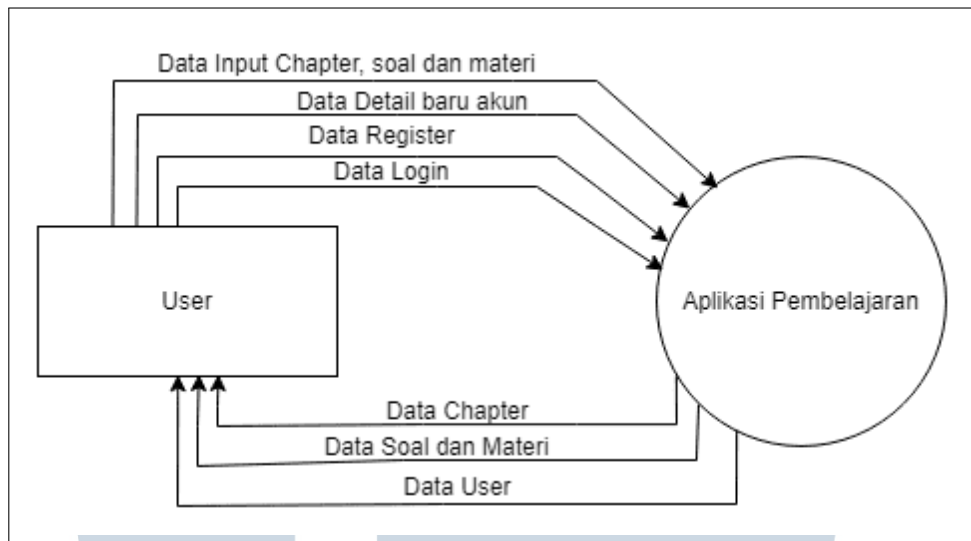
- D. Apabila kesempatan pengguna habis setelah menjawab soal, maka pengguna tidak dapat melanjutkan menjawab soal dan harus membeli kesempatan pada halaman profil atau menunggu pergantian hari.
- E. Kesempatan akan di-reset menjadi 3 kembali setiap hari pada pukul 00.00.

iii. *Reward*

- A. Chapter 1 masing - masing soal mendapatkan *reward* 5 gold dan 10 exp.
- B. Chapter 2 masing - masing soal mendapatkan *reward* 6 gold dan 12 exp.
- C. Chapter 3 masing - masing soal mendapatkan *reward* 8 gold dan 15 exp.
- D. Chapter 4 masing - masing soal mendapatkan *reward* 10 gold dan 20 exp.
- E. Pengguna akan mendapat badge Bronze apabila exp pengguna berada diantara 0 hingga 29.
- F. Pengguna akan mendapat badge Silver apabila exp pengguna berada diantara 30 hingga 64.
- G. Pengguna akan mendapat badge Gold apabila exp pengguna berada diantara 65 hingga 124.
- H. Pengguna akan mendapat badge Platinum apabila exp pengguna berada diatas 124.

(c) Data Flow Diagram (DFD)

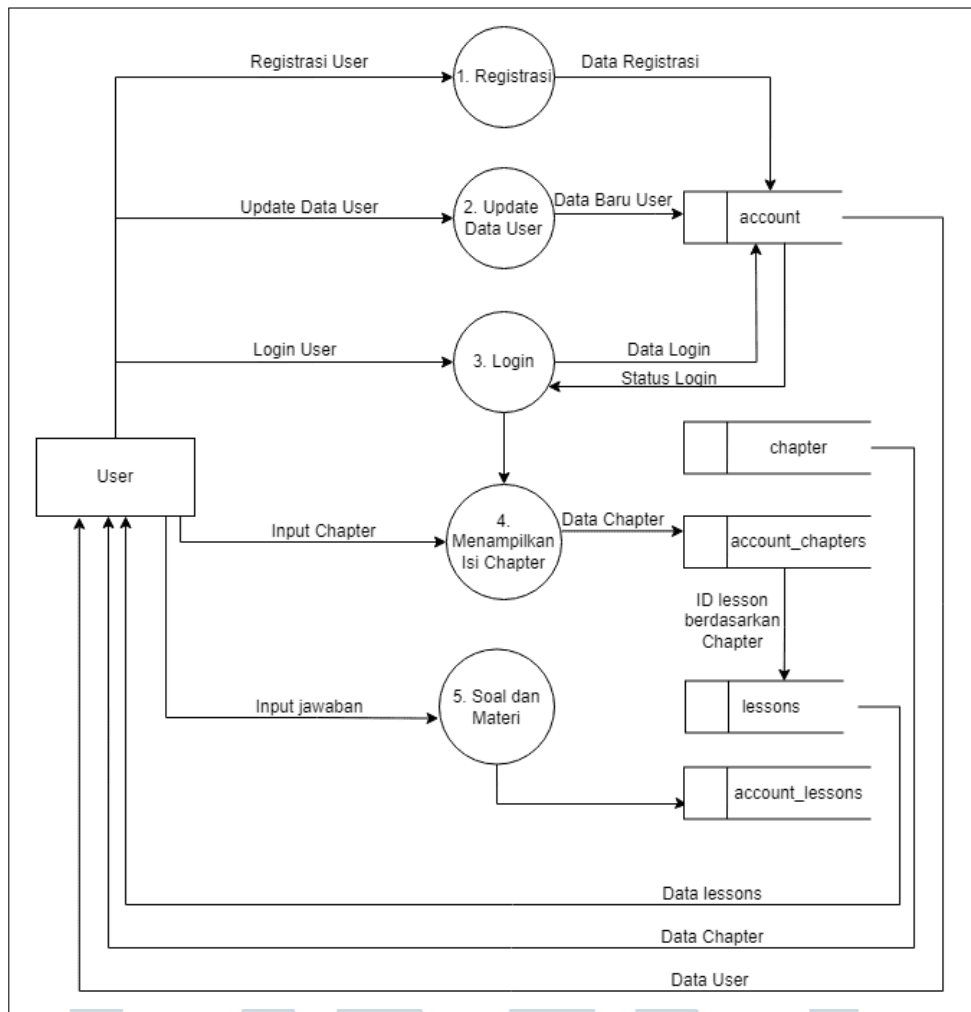
Data Flow Diagram atau DFD merupakan sebuah diagram untuk menggambarkan aliran data yang melewati proses - proses tertentu mulai dari input yang diberikan, bagaimana data tersebut disimpan dan output yang dihasilkan [24] [25]. Gambar 3.2 akan menunjukkan Data Flow Diagram (DFD) level 0.



Gambar 3.2. Data Flow Diagram Level 0

DFD level 0 merupakan diagram dasar yang hanya menggambarkan hubungan antara satu proses dengan beberapa entitas [26]. Diagram ini menunjukkan input yang masuk ke dalam proses berupa data login, register, detail baru akun, chapter, soal dan materi yang berasal dari user dan output yang keluar dari proses berupa data chapter, user, soal dan materi. Selain level 0, terdapat pula DFD level 1 yang menggambarkan detail lebih lanjut dari alur yang digambarkan pada DFD level 0 [26] seperti yang terlihat pada gambar 3.3.

U M I 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



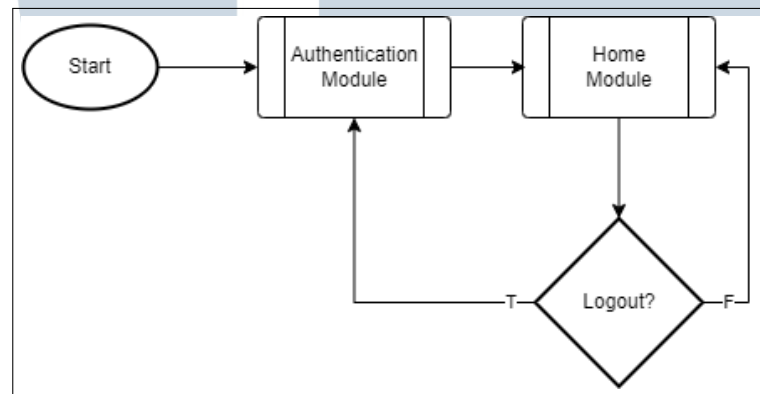
Gambar 3.3. Data Flow Diagram Level 1

DFD level 1 memiliki 5 buah proses, yaitu registrasi, update data user, login, menampilkan isi chapter, dan soal dan materi. Untuk proses registrasi, user akan memasukkan data registrasi user mulai dari nama, email, username, dan password. Data registrasi tersebut akan masuk ke tabel "account" untuk disimpan. Pada proses update data user, user akan memberikan input berupa data baru dan data tersebut akan dialirkan ke tabel "account" untuk dilakukan pembaruan data. Pada proses login, user akan memberikan input berupa username dan password yang dimana data input tersebut akan dialirkan ke tabel "account" untuk dilihat apakah data input ada pada database. Pada proses menampilkan isi chapter, user akan memberikan input dengan cara memilih chapter yang diinginkan dan input tersebut akan dialirkan

ke tabel "account_chapters" yang dimana dari tabel "account_chapters" akan mengalirkan data ke tabel "lessons" dan dari tabel "lessons" akan memberikan output berupa data lessons kepada user. Pada proses soal dan materi, pengguna akan memberikan input berupa pilihan jawaban yang dimana input tersebut akan dialirkan ke tabel "account_lessons". Tabel "chapter" akan memberikan output berupa data chapter kepada user agar user dapat memilih.

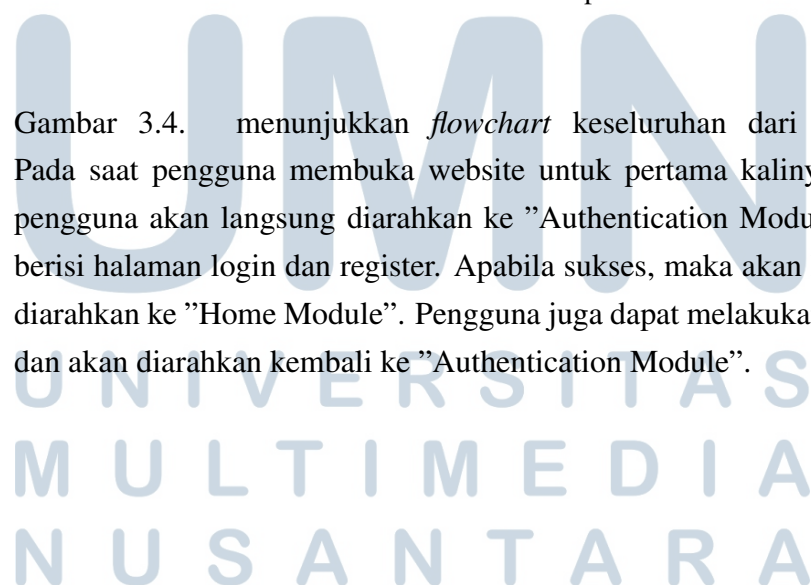
(d) Flowchart Aplikasi

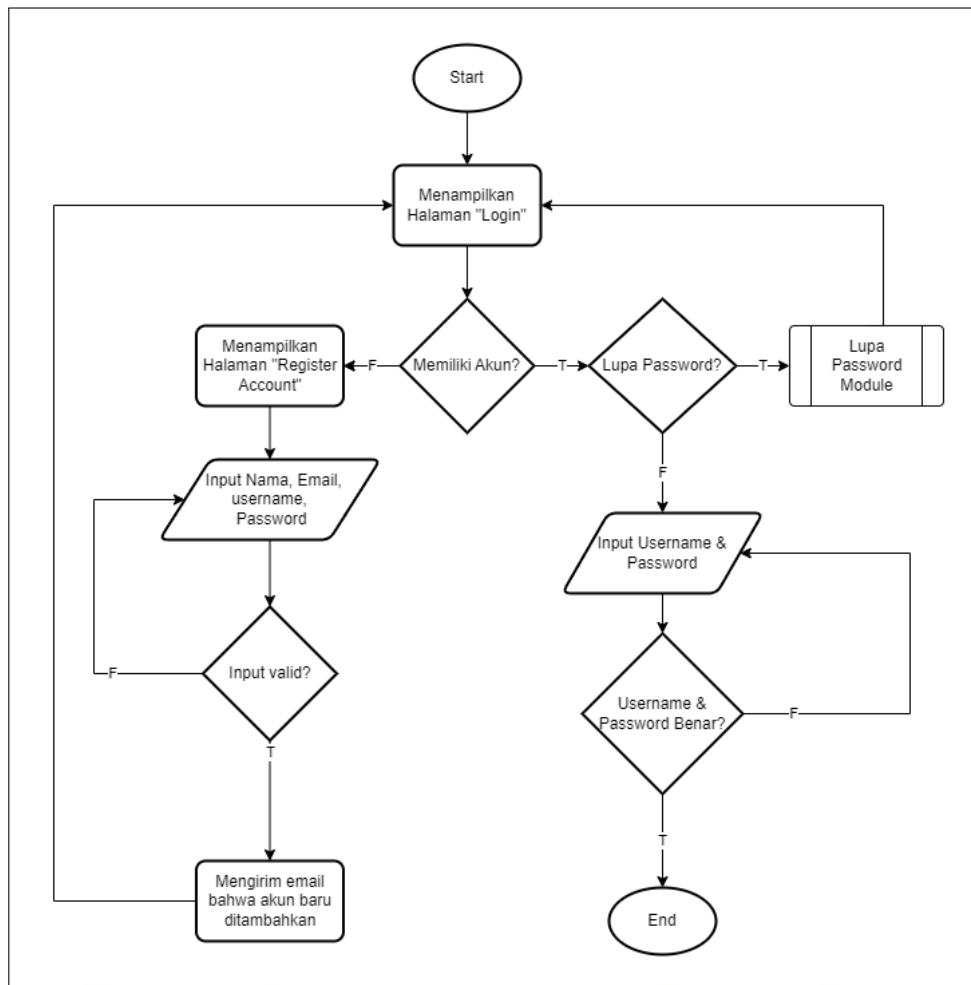
Berikut merupakan *flowchart* dari aplikasi pembelajaran *framework* Spring Boot dengan Metode gamifikasi berbasis website.



Gambar 3.4. Flowchart Keseluruhan Aplikasi

Gambar 3.4. menunjukkan *flowchart* keseluruhan dari aplikasi. Pada saat pengguna membuka website untuk pertama kalinya, maka pengguna akan langsung diarahkan ke "Authentication Module" yang berisi halaman login dan register. Apabila sukses, maka akan langsung diarahkan ke "Home Module". Pengguna juga dapat melakukan Logout dan akan diarahkan kembali ke "Authentication Module".

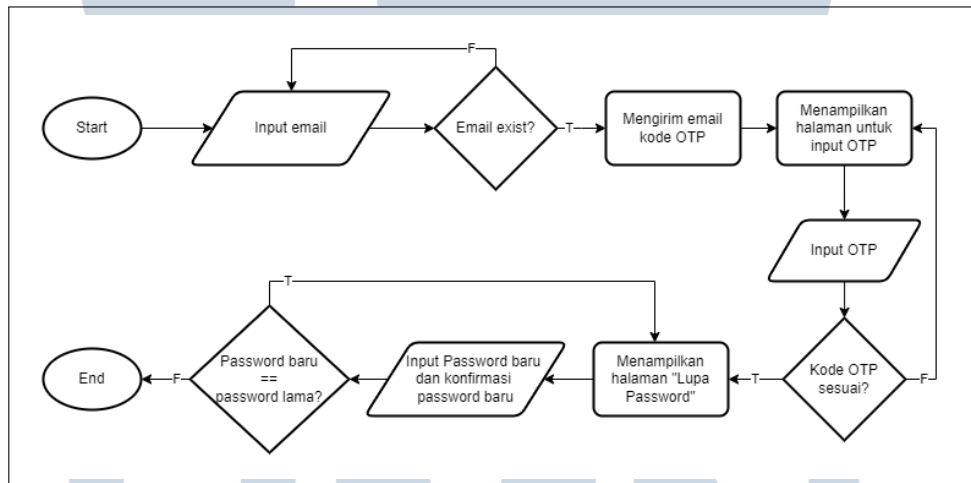




Gambar 3.5. Flowchart Autentikasi Aplikasi

Gambar 3.5. menunjukkan *flowchart* dari sistem autentikasi aplikasi yang akan dibuat. Alur autentikasi dimulai dari pengguna akan diarahkan ke halaman "Login" yang dimana pengguna akan dihadapkan dengan dua pilihan, yaitu memiliki akun atau tidak memiliki akun. Apabila pengguna tidak memiliki akun, maka pengguna dapat masuk ke halaman "Register Account" untuk melakukan pendaftaran akun. Pendaftaran akun dilakukan dengan memasukkan nama, email, username, dan password yang akan digunakan, kemudian akan dilanjutkan dengan validasi input. Apabila input yang dimasukkan pengguna tidak valid, maka pengguna akan diarahkan kembali ke halaman "Register Account" dengan *error message* yang sesuai. Apabila input tersebut valid, maka sistem akan mengirimkan notifikasi

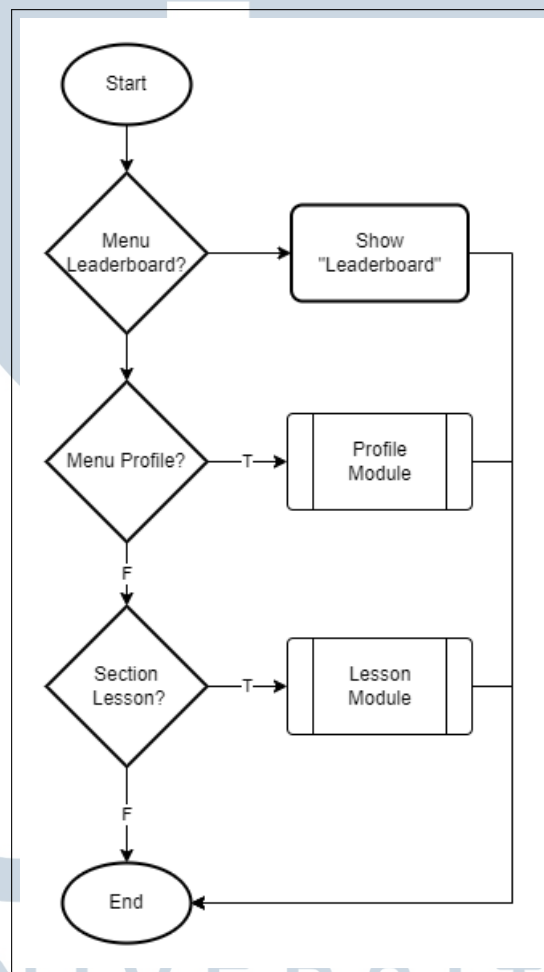
registrasi akun ke email yang telah didaftarkan sebelumnya dan langsung diarahkan ke halaman "Login". Apabila pengguna telah memiliki akun, maka pengguna dapat langsung memasukkan input berupa username dan password yang telah dimiliki kemudian sistem akan melakukan validasi apakah username dan password tersebut valid. Apabila username dan password tidak valid, maka pengguna akan diarahkan kembali ke halaman "Login" dengan *error message* yang sesuai dan apabila input valid, maka akan langsung diarahkan ke halaman "Home". Apabila pengguna lupa terhadap password yang mereka miliki, maka dapat langsung mengarah ke menu "Lupa Password" dan pengguna akan diarahkan ke "Lupa Password Module".



Gambar 3.6. Flowchart Lupa Password Aplikasi

Gambar 3.6. menunjukkan *flowchart* dari fitur "Lupa Password" pada aplikasi yang akan dibuat. Alur "Lupa Password" dimulai dari pengguna yang memberikan input berupa email dan akan dilakukan validasi apakah email tersebut terdaftar atau tidak. Apabila tidak terdaftar, maka pengguna akan diarahkan kembali ke halaman untuk memberikan input email. Apabila email tersebut terdaftar, maka aplikasi akan mengirimkan kode OTP ke email yang terdaftar dan pengguna dapat menggunakan kode OTP tersebut untuk mengisi input kode OTP pada halaman selanjutnya. Apabila kode OTP yang diinput oleh pengguna dan kode OTP yang dikirim melalui email tidak sesuai, maka pengguna akan diminta untuk melakukan input kode OTP kembali. Apabila kode

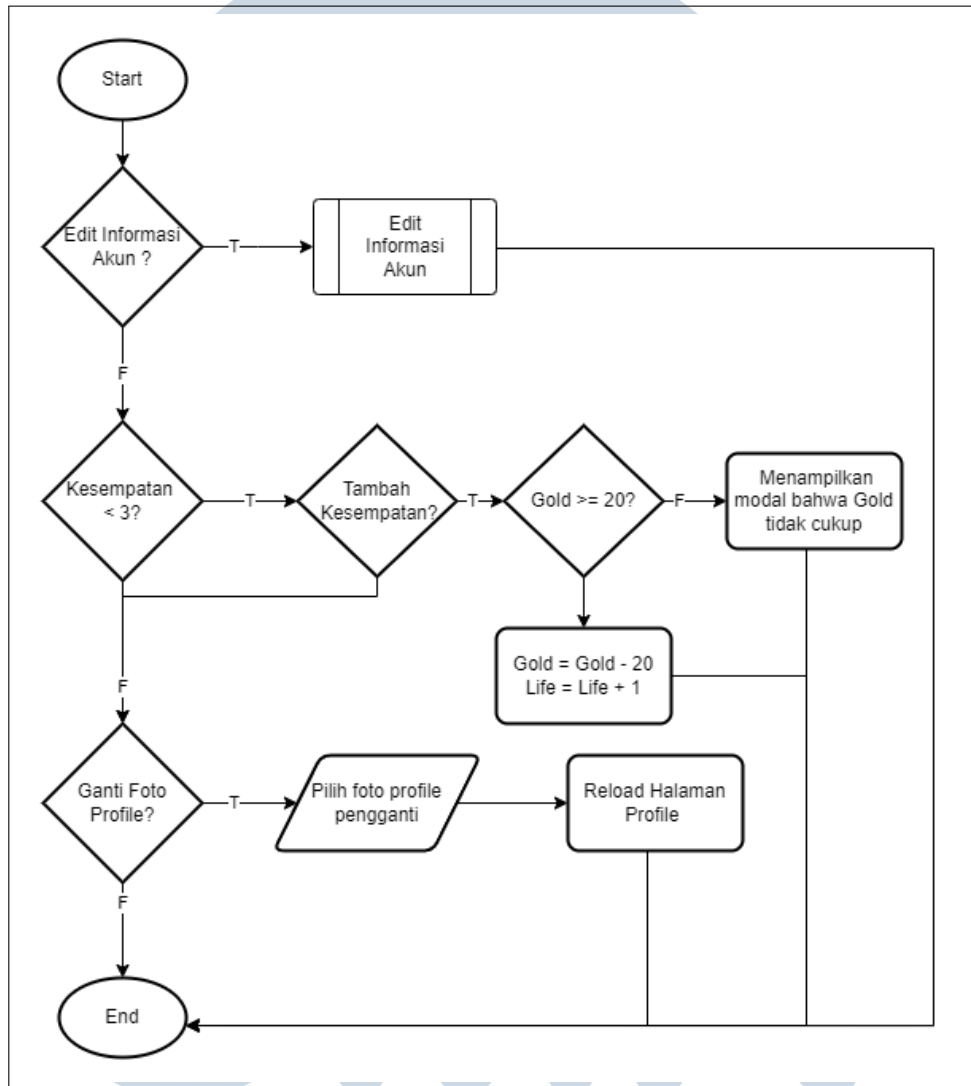
OTP yang diinput sesuai, maka pengguna akan diarahkan ke halaman untuk memberikan input password baru dan konfirmasi password. Apabila input password baru dan password lama sama, maka pengguna diarahkan kembali ke halaman input password baru untuk mengisi password yang berbeda dengan password lama. Apabila password baru dan password lama berbeda, maka pengguna berhasil melakukan pergantian password dan kembali ke halaman "Login".



Gambar 3.7. Flowchart Home Aplikasi

Gambar 3.7. menunjukkan *flowchart* dari halaman "Home" pada aplikasi yang akan dibuat. Pengguna akan dihadirkan 3 opsi saat tiba pada halaman "Home", yaitu Leaderboard, Profile, dan Chapters. Apabila pengguna memilih menu "Leaderboard", maka pengguna akan diarahkan ke menu "Leaderboard", apabila pengguna memilih menu

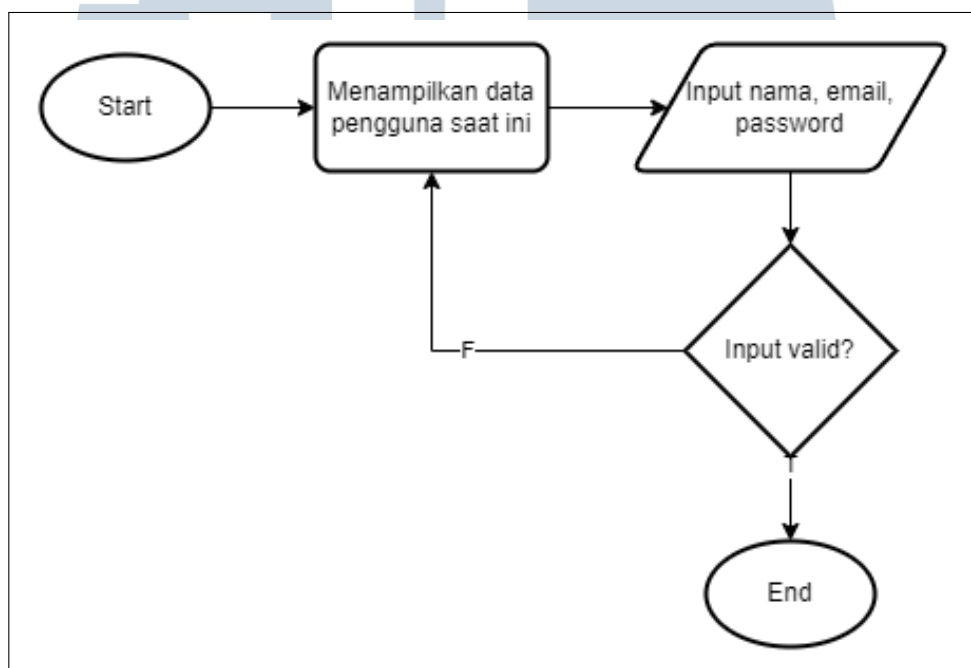
”Profile”, maka pengguna akan diarahkan ke menu ”Profile”, dan apabila pengguna memilih ”Chapter” maka pengguna akan diarahkan ke masing-masing ”lesson” dari ”Chapter” yang dipilih.



Gambar 3.8. Flowchart Profile Aplikasi

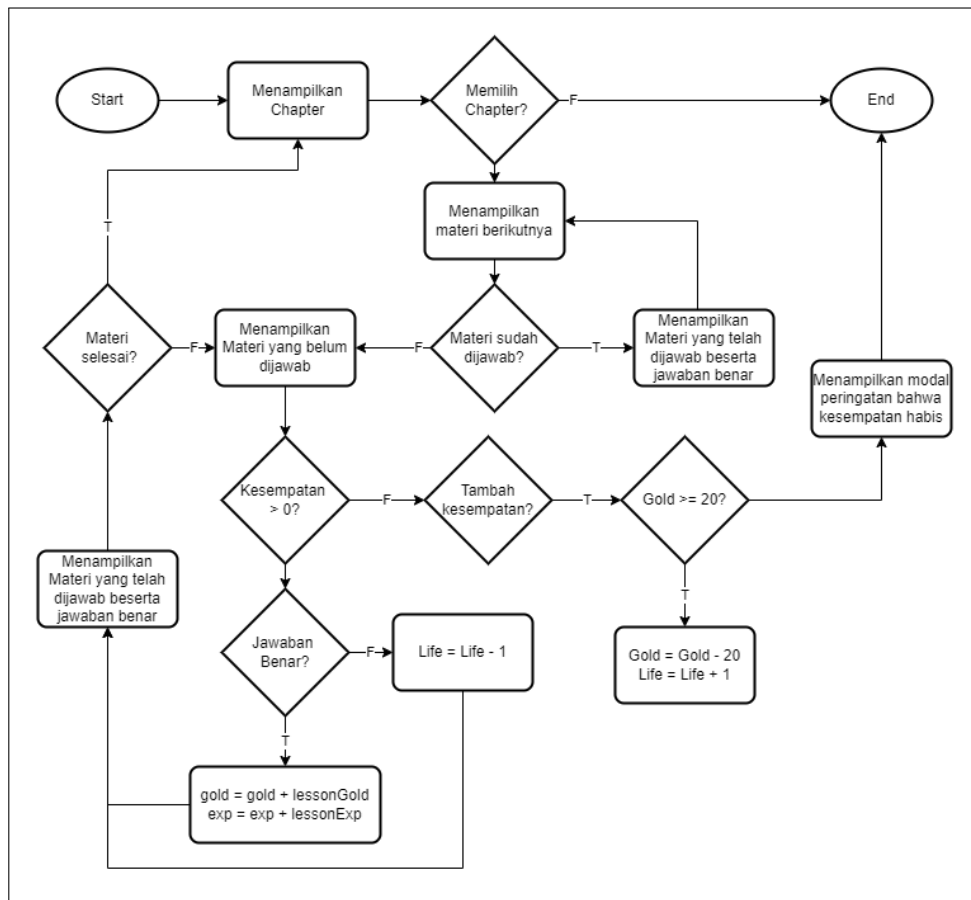
Gambar 3.8. menunjukkan *flowchart* dari halaman ”Profile” pada aplikasi yang akan dibuat. Apabila pengguna mengedit informasi akun sedang login, maka akan diarahkan ke ”Edit Informasi Akun Module”. Apabila kesempatan pengguna kurang dari 3, maka pengguna dapat melakukan klik pada tombol ”Tambah Kesempatan” kemudian aplikasi akan melakukan pengecekan apakah gold pengguna lebih dari atau sama dengan 20 karena harga 1 kesempatan adalah 20 gold.

Apabila gold mencukup, maka proses transaksi akan berjalan dan gold pengguna akan berkurang sebesar 20 dan kesempatan pengguna akan bertambah sebanyak 1. Apabila gold pengguna tidak mencukupi, maka akan muncul modal pop up yang menyatakan bahwa gold pengguna tidak cukup untuk melakukan pembelian kesempatan. Setelah itu, apabila pengguna ingin mengganti foto profil, maka pengguna dapat langsung memilih foto profil pengganti halaman akan di-refresh untuk menyimpan perubahan tersebut.



Gambar 3.9. Flowchart Edit Informasi Akun Aplikasi

Gambar 3.9. menunjukkan *flowchart* dari halaman "Edit Informasi Akun" pada aplikasi yang akan dibuat. Pengguna akan diarahkan ke halaman dengan field form yang sudah terisi dengan data-data pengguna yang lama. Pengguna dapat mengganti field yang ingin diganti baik nama, email, maupun password kemudian aplikasi akan melakukan validasi terkait input tersebut dan apabila valid, maka data pengguna akan diperbarui dan apabila input tidak valid, maka pengguna akan kembali diarahkan ke halaman "Edit Informasi Akun" dengan data - data lama.



Gambar 3.10. Flowchart Lesson Aplikasi

Gambar 3.10. menunjukkan *flowchart* dari menu "Lessons" yang menjadi inti dari aplikasi ini. Pengguna akan ditampilkan beberapa "Chapter" yang tersedia dan pengguna harus menyelesaikan "Chapter" secara berurutan dari awal hingga akhir. Saat memilih "Chapter", aplikasi akan melakukan pengecekan apakah materi dan soal pertama dari "Chapter" tersebut telah terjawab atau belum. Apabila materi dan soal tersebut sudah terjawab, maka akan menampilkan materi dan soal yang terjawab tersebut beserta jawaban yang benar kemudian akan menampilkan materi dan soal berikutnya dan melakukan pengecekan kembali. Apabila materi dan soal tersebut belum dijawab maka pengguna dapat memilih jawaban dari soal tersebut. Apabila jawaban salah, maka kesempatan pengguna akan dikurangi 1 tetapi jika jawaban pengguna benar maka pengguna akan mendapat tambahan gold dan exp sesuai "Chapter". Selanjutnya, pengguna akan diarahkan ke

halaman materi dan soal tersebut dengan jawaban yang benar dan berlanjut ke soal berikutnya apabila masih ada. Apabila soal dan materi pada "Chapter" tersebut sudah habis, maka pengguna akan diarahkan ke halaman "Home" untuk memilih "Chapter" selanjutnya. Apabila kesempatan pengguna kurang dari sama dengan 0, maka pengguna dapat menambah kesempatan. Aplikasi akan melakukan pengecekan gold pengguna apakah lebih dari sama dengan 20 karena harga 1 kesempatan adalah 20 gold. Apabila gold mencukupi, maka gold pengguna akan berkurang sebesar 20 dan kesempatan akan bertambah 1. Apabila gold tidak mencukupi, maka aplikasi akan menampilkan modal pop up peringatan bahwa kesempatan pengguna telah habis.

2. *Prototype and Absorb Feedback*

Merupakan tahap dimana pengembang menyiapkan *prototype* berdasarkan informasi kebutuhan yang telah didapat pada tahap sebelumnya dan meminta pengguna untuk memberikan *feedback* terkait *prototype* tersebut [23]. Setelah *feedback* didapatkan, pengembang akan memperbaiki kembali sesuai dengan *feedback* pengguna dan akan kembali meminta pengguna memberikan *feedback*. Proses ini akan terus terjadi secara terus menerus hingga pengguna puas dengan *prototype* yang telah dibuat [21] [23]. Pembuatan *prototype* aplikasi ini menggunakan aplikasi Figma. Pemilihan aplikasi Figma untuk membuat *prototype* karena aplikasi Figma dapat digunakan dan diakses dimana saja asal perangkat yang digunakan terhubung ke jaringan internet [27] [28].

3. *Finalize Product*

Merupakan tahap dimana pengembang melakukan pengkodean terhadap desain yang sudah dibuat sebelumnya [21]. Pengkodean ini dilakukan menggunakan HTML, CSS, dan Javascript untuk tampilan terhadap pengguna, dan menggunakan *framework* Spring Boot yang berbasis Java untuk sisi *back end* dan PostgreSQL untuk database pengguna dan materi. Penggunaan HTML dalam pengembangan *website* karena HTML merupakan perantara untuk menampilkan konten yang telah dibuat kepada *browser* tetapi konten yang ditampilkan tidak menarik karena tidak memiliki *styling*. Untuk penambahan *styling* diperlukan CSS atau *Cascading Style Sheets*. Penambahan CSS hanya sebatas memberikan *style* pada *website* tetapi tidak menambahkan interaksi untuk pengguna. Oleh karena itu, penambahan

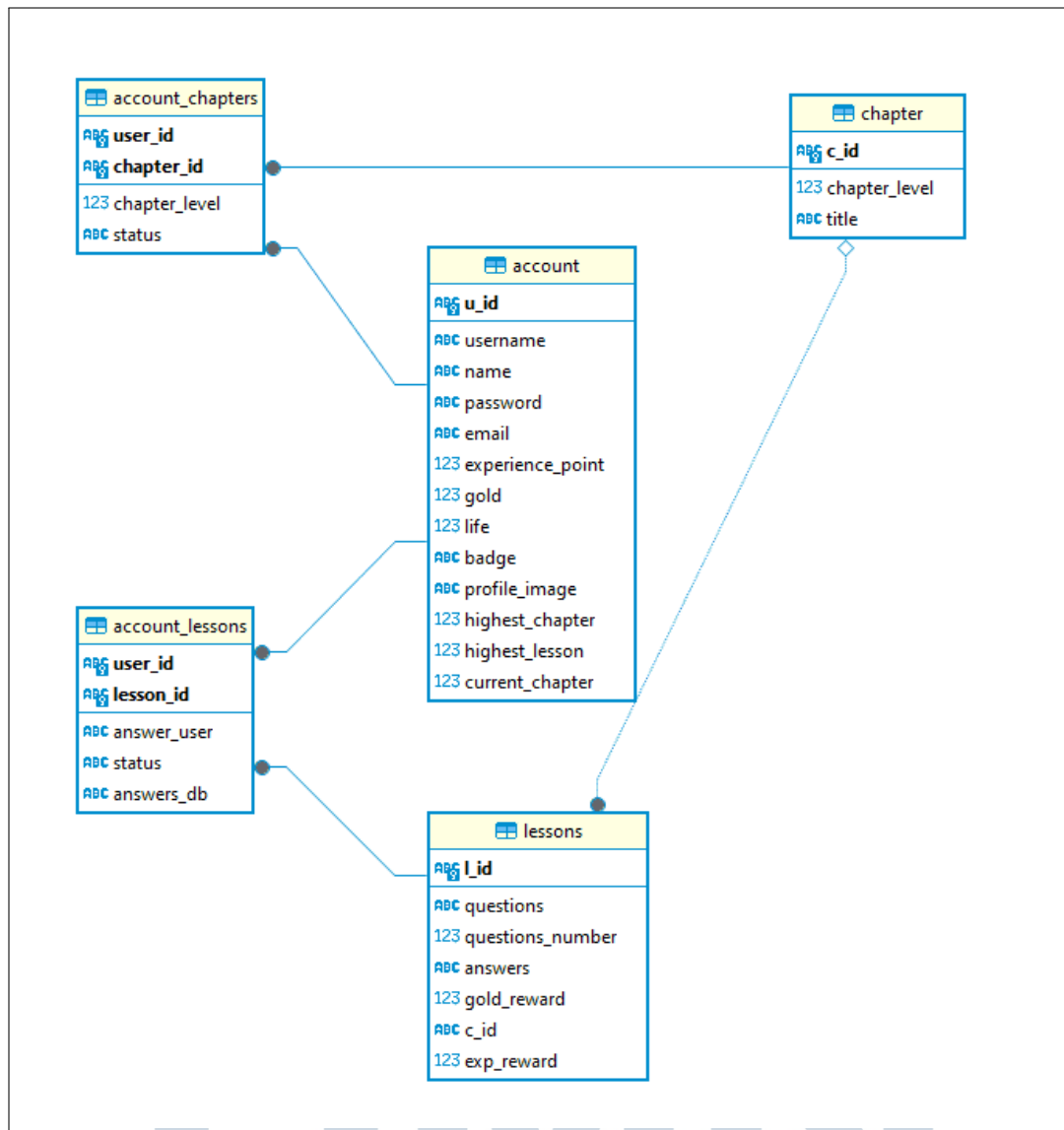
interaksi pengguna akan menggunakan JavaScript [3]. Pada sisi *back end* akan menggunakan bahasa pemrograman Java dengan *framework* Spring Boot karena *framework* Spring Boot memiliki beberapa kelebihan dibandingkan *framework* lain, yaitu memiliki server Tomcat sendiri sehingga aplikasi hanya perlu di-*run* tanpa perlu melakukan *deployment*, memiliki anotasi yang memudahkan dan mempercepat pengembangan dan dapat melakukan konfigurasi *dependencies* secara otomatis [29]. Setelah selesai melakukan pengembangan, pengembang akan melakukan pengujian untuk melihat apakah ada kegagalan dalam sistem atau tidak. Pada tahap ini juga, pengguna akan dilibatkan untuk melihat kepuasan pengguna terhadap program yang telah dibuat melalui penggunaan kuesioner dengan beberapa pertanyaan terkait penggunaan aplikasi dan akan dihitung kepuasan secara keseluruhan menggunakan kuesioner SUS (*System Usability Scale*)

3.2 Perancangan Aplikasi

3.2.1 Database Aplikasi

Database aplikasi pembelajaran *framework* Spring Boot ini dirancang menggunakan *database* PostgreSQL dan aplikasi pengelolaan *database* menggunakan aplikasi Dbeaver. *Database* aplikasi ini terdiri dari 5 buah tabel, diantaranya adalah tabel "account", "account_chapters", "account_lessons", "chapter", "lessons". Untuk ERD (*Entity Relationship Diagram* dari *database* aplikasi ini dapat dilihat pada Gambar 3.19.

U M I 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



Gambar 3.11. *Entity Relationship Diagram* dari *Database* aplikasi pembelajaran *framework* Spring Boot

Pada Gambar 3.11. ditunjukkan bahwa tabel "account" memiliki relasi *one-to-many* dengan tabel "account.chapters" yang berarti satu data pada tabel "account" memiliki beberapa data berupa *progress* dari pengguna terhadap chapter - chapter yang ada. Tabel "account" juga memiliki relasi *one-to-many* dengan tabel "account_lessons" yang berarti satu data pada tabel "account" memiliki beberapa data berupa *progress* dari pengguna terhadap lesson - lesson yang ada. Tabel "chapter" memiliki relasi *one-to-many* dengan tabel "lessons" yang berarti satu data pada tabel "chapters" memiliki beberapa data pada tabel "lessons". Tabel "chapter" juga memiliki relasi *one-to-many* dengan tabel "account_chapters" yang berarti

satu data pada tabel "chapter" memiliki beberapa data berupa *progress* dari setiap chapter yang ada pada tabel "account_chapters". Tabel "lessons" juga memiliki relasi *one-to-many* dengan tabel "account_lessons" yang berarti satu data pada tabel "lessons" memiliki beberapa data berupa *progress* dari setiap lesson yang ada pada tabel "account_lessons".

Tabel 3.1. Struktur Tabel "account" pada Database

account	
u_id	String
username	String
name	String
password	String
email	String
experience_point	Integer
gold	Integer
life	Integer
badge	String
profile_image	String
highest_chapter	Integer
highest_lesson	Integer
current_chapter	Integer

Tabel 3.1. menunjukkan struktur dari tabel "account" yang terdiri dari beberapa atribut, diantaranya adalah u_id, username, name, password, email, experience_point, gold, life, badge, profile_image, highest_chapter, highest_lesson, dan current_chapter. u_id merupakan *primary key* dari tabel "account" yang bertipe data String atau VARCHAR. Atribut u_id berisi id untuk setiap data yang dihasilkan secara otomatis. Atribut username memiliki nilai berupa input String dari pengguna melalui aplikasi saat melakukan registrasi akun yang dimana username ini dapat digunakan untuk login ke aplikasi. Atribut name memiliki nilai berupa input String dari pengguna melalui aplikasi saat melakukan registrasi akun yang dimana atribut name ini akan ditampilkan pada bar navigasi aplikasi. Atribut password memiliki nilai berupa input String dari pengguna melalui aplikasi saat melakukan registrasi akun, reset password, maupun saat pengguna mengedit informasi akun. Nilai atribut password akan dienkripsi menggunakan metode "Bcrypt" sehingga password asli tidak akan terlihat langsung. Atribut email memiliki nilai berupa input String dari pengguna melalui aplikasi saat melakukan registrasi akun maupun saat pengguna mengedit informasi akun. Atribut email digunakan untuk pengiriman notifikasi registrasi akun dan kode OTP saat pengguna melakukan reset password.

Atribut *experience_point* dan *gold* memiliki nilai berupa integer yang didapat setiap pengguna menyelesaikan soal yang tersedia dengan benar. Nilai yang didapat berbeda - beda tergantung dari level chapter yang dikerjakan. Atribut *life* memiliki nilai *default* sebesar 3 yang dimana akan berkurang setiap kali pengguna menjawab soal dengan salah dan dapat bertambah apabila pengguna melakukan penambahan kesempatan. Atribut *life* juga akan di-reset kembali menjadi 3 saat pergantian hari. Atribut *badge* memiliki nilai berupa String yang didapat berdasarkan nilai *experience_point* yang didapat. Atribut *profile_image* memiliki nilai *default* yaitu *Avatar10*. *Avatar10* merujuk pada nama file pada penyimpanan lokal untuk foto profil *default*. Nilai dapat berubah sesuai dengan pilihan foto profil yang dipilih oleh pengguna. Atribut *highest_chapter*, *highest_lesson* memiliki nilai integer yang akan terisi saat pengguna masuk ke chapter tertentu. Atribut *current_chapter* memiliki nilai integer yang akan terisi saat pengguna masuk ke chapter tertentu dan nilai atribut ini akan digunakan pada logika halaman "Home".

Tabel 3.2. Struktur Tabel "chapter" pada Database

chapter	
c_id	String
chapter_level	Integer
title	String

Tabel 3.2. menunjukkan struktur dari tabel "chapter" yang terdiri dari 3 atribut, diantaranya adalah *c_id*, *chapter_level*, dan *title*. Atribut *c_id* merupakan *primary key* dari tabel "chapter" yang bertipe data String atau VARCHAR. Atribut *c_id* berisi id untuk setiap data yang dihasilkan secara otomatis. Atribut *chapter_level* memiliki nilai berupa integer untuk merujuk pada level chapter dan diisi secara manual. Atribut *title* memiliki nilai berupa String untuk merujuk pada judul setiap chapter dan diisi secara manual.

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 3.3. Struktur Tabel "lessons" pada Database

lessons	
l_id	String
questions	String
questions_number	Integer
answer	String
gold_reward	Integer
exp_reward	Integer
c_id	String

Tabel 3.3. menunjukkan struktur dari tabel "lessons" yang terdiri dari 7 atribut, diantaranya adalah l_id, questions, questions_number, answer, gold_reward, exp_reward dan c_id. Atribut l_id merupakan *primary key* sedangkan c_id merupakan *foreign key* dari tabel "chapter" yang bertipe data String atau VARCHAR. Atribut questions memiliki nilai berupa String yang berisi pertanyaan yang akan ditampilkan ke pengguna sedangkan atribut answer berisi opsi pilihan (a/b/c/d) yang menjadi jawaban dari pertanyaan tersebut. Atribut questions_number memiliki nilai berupa integer yang berisi nomor pertanyaan, sedangkan gold_reward dan exp_reward merupakan atribut yang berisi jumlah gold dan exp yang akan ditambah ke gold dan exp pengguna saat pengguna berhasil menjawab pertanyaan dengan benar. Atribut - atribut tersebut kecuali *primary key* dan *foreign key* diisi secara manual melalui query SQL.

Tabel 3.4. Struktur Tabel "account_chapters" pada Database

account_chapters	
user_id	String
chapter_id	String
chapter_level	Integer
status	String

Tabel 3.4. menunjukkan struktur dari tabel "account_chapters" yang terdiri dari 4 atribut, diantaranya adalah user_id, chapter_id, chapter_level dan status. Tabel ini menyimpan data terkait relasi antara tabel "Chapter" dan tabel "Account", yang dimana atribut user_id dan chapter_id menjadi *primary key* gabungan atau *composite primary key* sekaligus *Foreign key* untuk tabel "Chapters" dan tabel "Account". Atribut chapter_level memiliki nilai berupa integer untuk merujuk pada level chapter dan akan terisi otomatis setelah pengguna registrasi akun. Atribut status memiliki nilai *default* "Incomplete" untuk setiap user_id dan chapter_id. Nilai atribut status

akan berubah tergantung *progress* yang dikerjakan oleh pengguna. *Foreign key* dari tabel ini

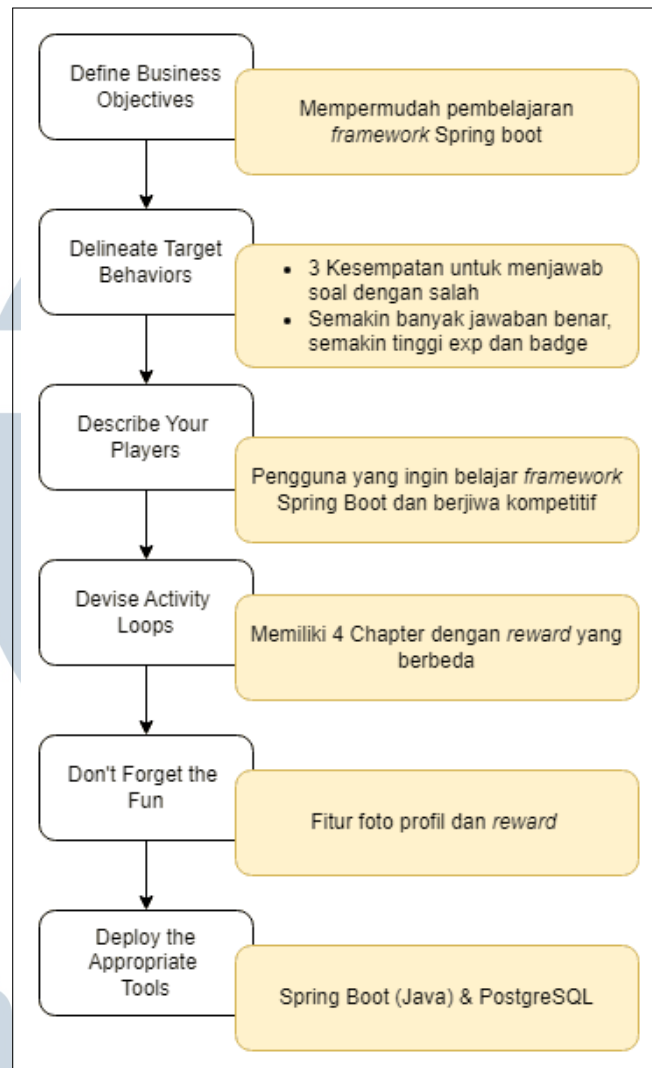
Tabel 3.5. Struktur Tabel "account_lessons" pada Database

account_lessons	
user_id	String
lesson_id	String
answer_user	String
status	String
answer_db	String

Tabel 3.5. menunjukkan struktur dari tabel "account_lessons" yang terdiri dari 5 atribut, diantaranya adalah user_id, lesson_id, answer_user, status dan answer_db. Tabel ini menyimpan data terkait relasi antara tabel "lessons" dan tabel "Account", yang dimana atribut user_id dan lesson_id menjadi *primary key* gabungan atau *composite primary key* sekaligus *Foreign key* untuk tabel "Chapters" dan tabel "Account". Atribut answer_user memiliki nilai berupa String yang akan terisi otomatis setelah pengguna memilih jawaban pada soal yang ada. Atribut status hanya memiliki 2 nilai, yaitu "True" dan "False" yang dimana akan terisi "True" apabila nilai pada atribut answer_user sama dengan nilai pada atribut answer_db dan akan bernilai "False" apabila nilai pada atribut answer_user berbeda dengan nilai pada atribut answer_db. Atribut answer_db berisi nilai yang didapat dari atribut "answer" pada tabel "lessons" berdasarkan nilai atribut lesson_id yang sesuai.

3.2.2 Penerapan Gamifikasi

Penerapan gamifikasi pada aplikasi pembelajaran *framework* Spring Boot ini dilakukan dengan menggunakan salah satu *framework* gamifikasi, yaitu *Six Steps to Gamification*. *Six Steps to Gamification* terbagi menjadi enam tahap, yaitu *Defines Business Objectives*, *Delineate Target Behaviors*, *Describe Your Players*, *Devise Activity Loops*, *Don't Forget the Fun* dan *Deploy the Appropriate Tools* seperti yang terlihat pada gambar 3.12.



Gambar 3.12. Bagan Penerapan *Framework* Gamifikasi "Six Steps to Gamification"

Detail dari masing - masing tahap akan dijelaskan sebagai berikut:

1. *Define Business Objectives*

Aplikasi ini dirancang dan dibangun dengan tujuan agar pengguna dapat lebih mudah untuk mempelajari *framework* Spring Boot. Dengan adanya unsur *game* yang didapat melalui metode gamifikasi, diharapkan pengguna dapat lebih nyaman dan menikmati pembelajaran *framework* Spring Boot ini.

2. *Delineate Target Behaviors*

Aplikasi ini dapat membuat pengguna terus menerus menggunakan aplikasi ini karena terdapat beberapa level yang dimana level selanjutnya dapat dijalankan setelah level sebelumnya diselesaikan. Terdapat juga 3 buah

kesempatan yang dimana pengguna harus berusaha agar tidak kehabisan kesempatan sebelum menyelesaikan seluruh soal dengan benar. Selain itu, semakin banyak jawaban benar yang diberikan pengguna, maka semakin tinggi pula *experience point* dan *badge* yang didapat. Oleh karena itu, semakin tinggi *experience point* pengguna, maka semakin mengerti pengguna tersebut terkait cara penggunaan *framework* Spring Boot yang tepat.

3. *Describe Your Players*

Aplikasi ini menargetkan pengguna yang tertarik pada bidang pemrograman atau bidang informatika. Selain itu, aplikasi ini juga menargetkan pengguna yang ingin memperluas pengetahuan terlebih untuk *framework* berbasis Java, yaitu Spring Boot. Terdapat juga fitur *leaderboard* untuk pengguna yang memiliki sifat kompetitif agar masing-masing pengguna dapat saling berkompetisi dengan melihat fitur *leaderboard* tersebut.

4. *Devise Activity Loops*

Aplikasi ini memberikan sebuah motivasi kepada pengguna melalui apresiasi berupa *gold*, *experience point*, dan *badge*. Setiap chapter memiliki tingkat kesulitan yang berbeda, sehingga apresiasi yang didapat berbeda pula. Untuk Chapter pertama, masing - masing pertanyaan mendapat *gold* sebesar 5 dan *experience point* sebesar 10. Untuk Chapter kedua, masing - masing pertanyaan mendapat *gold* sebesar 6 dan *experience point* sebesar 12. Untuk Chapter ketiga, masing - masing pertanyaan mendapat *gold* sebesar 8 dan *experience point* sebesar 15. dan Untuk Chapter keempat, masing - masing pertanyaan mendapat *gold* sebesar 10 dan *experience point* sebesar 20. Untuk *badge* yang akan diberikan kepada pengguna akan dikelompokkan berdasarkan jumlah *experience point*. Untuk *experience point* sebesar 0 hingga 29, maka *badge* yang didapat adalah "Bronze". Untuk *experience point* sebesar 30 hingga 64, maka *badge* yang didapat adalah "Silver". Untuk *experience point* sebesar 65 hingga 124, maka *badge* yang didapat adalah "Gold". Dan untuk *experience point* lebih besar dari 124, maka *badge* yang didapat adalah "Platinum".

5. *Don't Forget the Fun*

Aplikasi ini memberikan aspek menyenangkan bagi pengguna, seperti tantangan, elemen interaktif, dan apresiasi. Untuk tantangan yang diberikan adalah apabila kesempatan pengguna sudah habis, maka pengguna tidak dapat melanjutkan mengerjakan soal dan materi kecuali membeli kesempatan melalui halaman Profile atau menunggu pukul 00.00. Untuk elemen interaktif sendiri terdapat pada halaman Profile, tepatnya pada bagian memilih foto profile. Pengguna dapat melakukan klik pada foto profil yang diinginkan sehingga foto tersebut akan ter-*highlight* dan pengguna dapat klik "Simpan" untuk menjadikan foto tersebut menjadi foto profile. Pengguna juga mendapatkan apresiasi dalam bentuk *gold* yang dapat digunakan untuk membeli kesempatan, dan *experience point* untuk mendapatkan *badge*.

6. *Deploy the Appropriate Tools*

Untuk pembangunan sistem gamifikasi yang sesuai dengan langkah - langkah yang telah disiapkan sebelumnya, maka perancangan dan pembangunan aplikasi pembelajaran *framework* Spring Boot berbasis website dilakukan menggunakan bahasa pemrograman Java. Pengembangan aplikasi ini juga menggunakan *framework* Spring Boot sebagai kerangka kerja aplikasi. Untuk basis data atau *database* yang digunakan adalah PostgreSQL.

