BAB 3
METODOLOGI PENELITIAN

Pada penelitian ini, metodologi yang digunakan adalah Waterfall yang terdiri dari 5 tahap yaitu analisis kebutuhan, perancangan aplikasi, pengembangan/implementasi aplikasi (implementasi), ujicoba aplikasi (verifikasi), dan pemeliharaan aplikasi.

3.1 Analisis Kebutuhan Sistem

Pertama, akan dilakukan telaah literatur mengenai teori-teori dasar yang dibutuhkan dalam rumusan masalah di penelitian ini. Setelah telaah literatur, akan dilakukan proses perancangan aplikasi dimulai dari diagram alir untuk melakukan visualisasi alur kerja pengembangan aplikasi yang dibuat, perencanaan dan observasi terhadap arsitektur dan *tools* yang akan dipakai untuk pelaksanaan penilitian ini. Hasil perencanaan dan observasi tersebut akan digunakan sebagai simulasi penerapan metode keamanan dalam proses pembayaran menggunakan *Time-Based One-Time Password* dengan *Challenge Response Authentication Mechanism* berbasis Web. Simulasi akan dilakukan pada aplikasi Android dan Web, aplikasi simulasi ini memiliki beberapa kebutuhan dalam pengembangannya yang dibagi menjadi 2 bagian yaitu kebutuhan fungsional dan non fungsional.

3.1.1 Kebutuhan Fungsional

Kebutuhan fungsional berisikan mengenai fitur-fitur dan fungsi yang dibutuhkan dalam Aplikasi Store. Dalam proses penentuan kebutuhan fungsional ini dilakukan dengan cara observasi dan diskusi dengan masyrakat yang aktif dalam melakukan transaksi secara digital. Berikut adalah fitur-fitur dan fungsi dari hasil observasi tersebut:

1. Pengguna dapat melakukan *login* dan *register*.
2. Pengguna dapat melihat produk.
3. Pengguna dapat melihat detail informasi dari produk yang dilihat.
4. Pengguna dapat melakukan transaksi.

5. Pengguna dapat mengakses kode TOTP mereka pada web yang diberikan.
6. Pengguna dapat melakukan validasi TOTP mereka untuk menyelesaikan transaksi.

3.1.2 Kebutuhan Non Fungsional

Kebutuhan non fungsional berisikan mengenai spesifikasi kebutuhan perangkat dan sistem yang akan digunakan dalam perancangan dan pembangunan sistem dalam penelitian ini. Berikut adalah spesifikasi yang digunakan dalam perancangan dan pembangunan sistem ini:

A Perangkat Keras

(a) Processor Intel® core™ i7-7500U CPU @ 2.70GHZ

(b) RAM 12288 MB

(c) Hard Disk 1 TB

B Perangkat Lunak

(a) Operating system Windows 10 (64-bit)

(b) Android Studio

(c) Visual Studio Code

(d) Firebase

(e) Figma

3.2 Perancangan Aplikasi

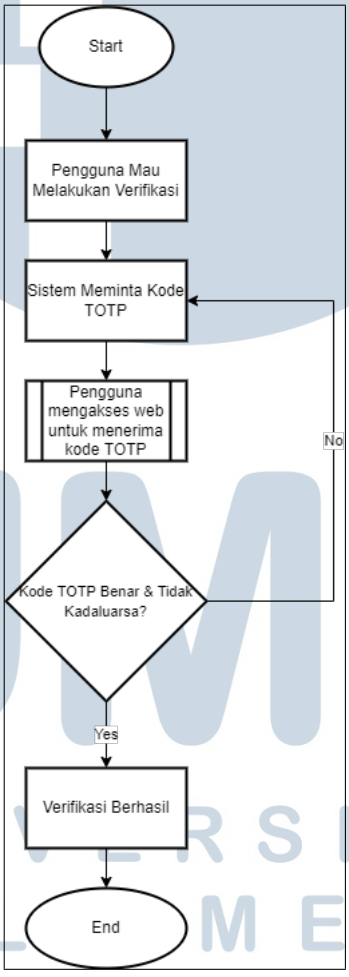
Perancangan aplikasi ini dibuat untuk menjadi dasar acuan arsitektur dalam pengerjaan pengembangan aplikasi yang didasari dari metode Waterfall.

3.2.1 Diagram Alir

A Challenge Response Authentication Mechanism

Pada penelitian ini akan menggunakan 2 jenis pendekatan *Challenge Response Authentication Mechanism* (CRAM) yaitu *static* dan *dynamic* yang akan diimplementasikan pada aplikasi (*dynamic* melalui *Time-Based One-Time Password*) dan web (*static*).

A.1 Dynamic

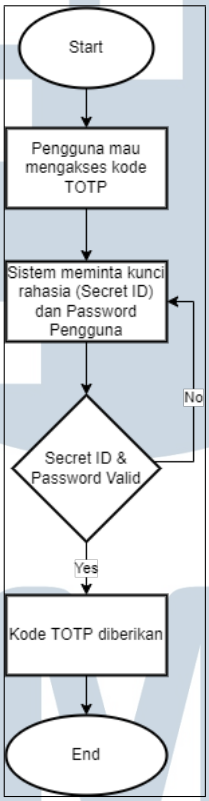


Gambar 3.1. Diagram Alir Alur *Dynamic*

Gambar 3.1 adalah alur proses otentikasi menggunakan metode *Challenge Response Authetication Mechanism* (CRAM) jenis *dynamic*. Pengguna yang ingin melakukan verifikasi atau autentikasi akan diberikan tantangan berupa

memasukkan kode *Time-Based One-Time Password* yang bisa dilihat pengguna dengan mengakses halaman web otentikator. Kode TOTP hanya bisa digunakan satu kali dengan waktu kadaluarsa 90 detik setelah diciptakan. Kemudian kode TOTP yang *valid* akan menyelesaikan otentikasi atau verifikasi. Alir ini yang akan digunakan menjadi landasan dari pembuatan diagram alir halaman TOTP aplikasi.

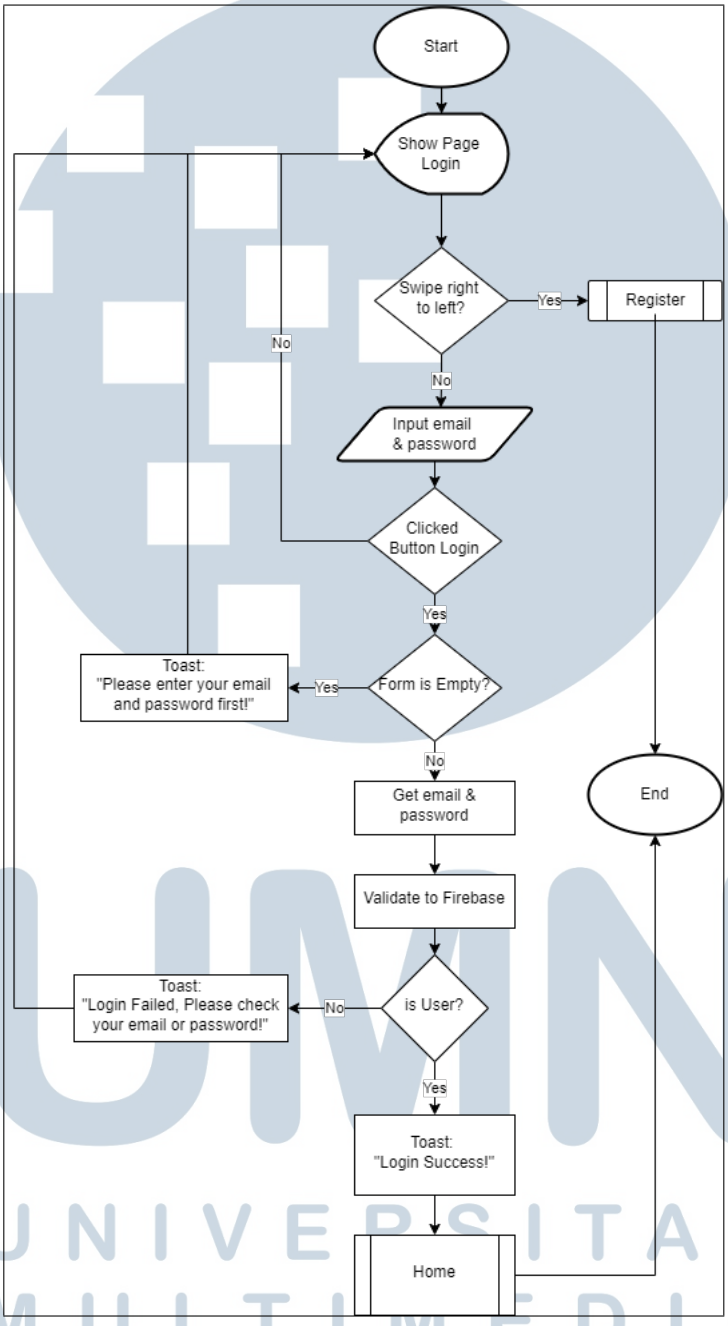
A.2 Static



Gambar 3.2. Diagram Alir Alur *Static*

Gambar 3.2 adalah alur proses otentikasi menggunakan metode *Challenge Response Authetication Mechanism* (CRAM) jenis *static*. Pengguna yang ingin mengakses kode TOTP perlu melakukan otentikasi dengan memasukkan kunci rahasia (*Secret ID*) dan *Password* pengguna. Berbeda dengan pendekatan *Dynamic* yang menggunakan kunci rahasia yang berubah-ubah (*One-Time Password*), pendekatan *static* ini memiliki kunci tetap yang tidak berubah-ubah yaitu berupa *Secret ID*. Kemudian apabila otentikasi berhasil, pengguna dapat melihat kode TOTP yang dibutuhkan untuk melakukan verifikasi. Alir ini yang akan digunakan menjadi landasan dari pembuatan diagram alir halaman TOTP web.

B Halaman Login Aplikasi

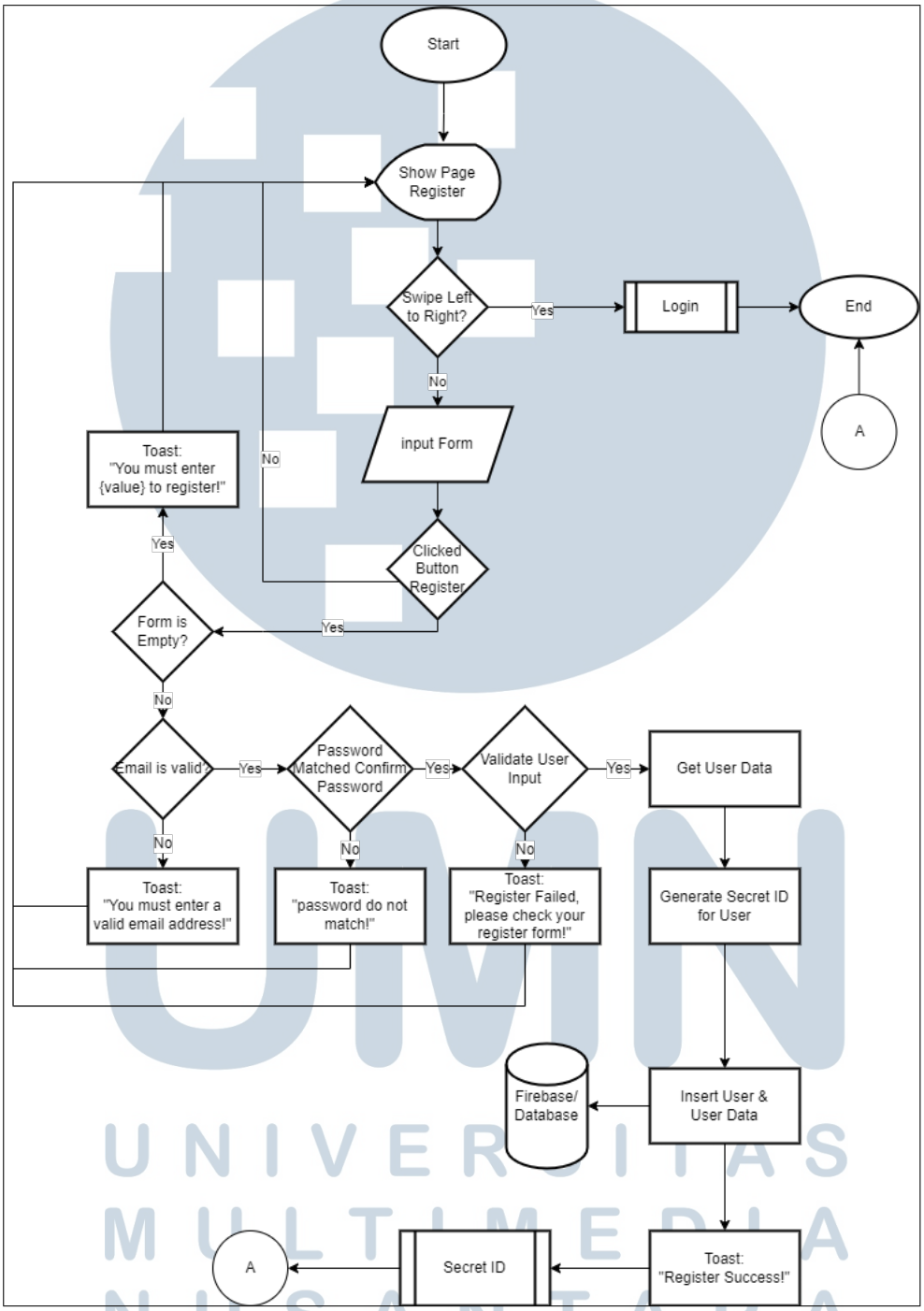


Gambar 3.3. Diagram Alir Halaman *Login* Aplikasi

Gambar 3.3 adalah proses diagram alir halaman *login*. halaman ini akan ditampilkan sebagai halaman awal ketika pengguna membuka aplikasi. pengguna dapat menggeser ke arah kanan untuk mengakses halaman pendaftaran apabila pengguna belum memilikin akun atau belum terdaftar. Kemudian pengguna yang

sudah memiliki akun dapat memasukan kredensial berupa *email* dan *password* dari akun pengguna. Pada halaman ini, sistem akan melakukan pengecekan dari kredensial yang dimasukkan pengguna. Pengecekan dimulai dari apakah kredensial sudah terisi atau tidak, jika terdapat kredensial yang kosong maka, pengguna akan mendapatkan sebuah pesan untuk memasukkan *email* dan password mereka. Setelah itu dilakukan verifikasi melalui firebase terhadap kredensial yang terisi, jika data pengguna sudah benar dan terverifikasi, pengguna akan mendapatkan pesan *login* berhasil serta diarahkan menuju halaman utama. Namun apabila data pengguna tidak sesuai atau terverifikasi maka pengguna akan mendapatkan pesan *login* gagal, harap melakukan pengecekan kembali terhadap kredensial yang dimasukkan.

C Halaman Register Aplikasi

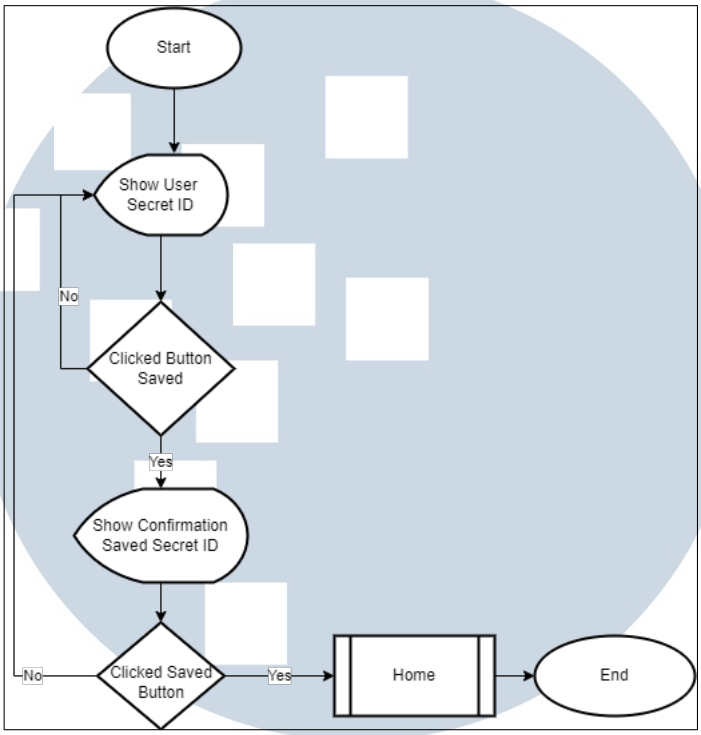


Gambar 3.4. Diagram Alir Halaman *Register* Aplikasi

Gambar 3.4 adalah proses diagram alir halaman *register*. Halaman ini adalah halaman dimana pengguna dapat melakukan pendaftaran akun untuk dapat

masuk ke dalam aplikasi Store. Pada halaman ini, apabila pengguna menggeser ke arah kiri maka, pengguna akan diarahkan ke halaman *login*. Pada halaman *register* ini terdapat 5 *input form* yaitu *username*, *email*, nomor telepon, *password*, dan konfirmasi *password* yang harus dilengkapi/diisi pengguna sebagai data pengguna. Sistem akan melakukan pengecekan terhadap *input form* yang diisi pengguna, apabila *input form* kosong maka, pengguna akan mendapatkan pesan untuk memasukkan data sesuai *form*. Kemudian sistem akan pengecekan terhadap *email* yang dimasukkan apakah *email* tersebut merupakan alamat *email* yang benar, jika alamat email tidak benar maka pengguna akan mendapatkan pesan untuk memasukkan alamat *email* yang benar. Kemudian sistem akan melakukan pengecekan terhadap *password* pengguna, apakah *password* tersebut sama dengan konfirmasi *password*, jika berbeda maka pengguna akan mendapatkan pesan bahwa kedua *password* tersebut tidak sama. Setelah semua *input form* sudah terisi dengan benar maka, sistem akan melakukan pengecekan/validasi ke Firebase, untuk memastikan data tersebut sudah pernah terdaftar atau tidak, jika terdapat kesalahan maka, pengguna akan mendapatkan pesan untuk melakukan pengecekan kembali terhadap *input form* tersebut. Data pengguna yang berhasil melewati tahap validasi akan disimpan ke dalam Firebase Firestore Database dan kemudian pengguna akan mendapatkan pesan *register* berhasil dilakukan, lalu pengguna akan otomatis melakukan *login* dan diarahkan menuju halaman Secret ID.

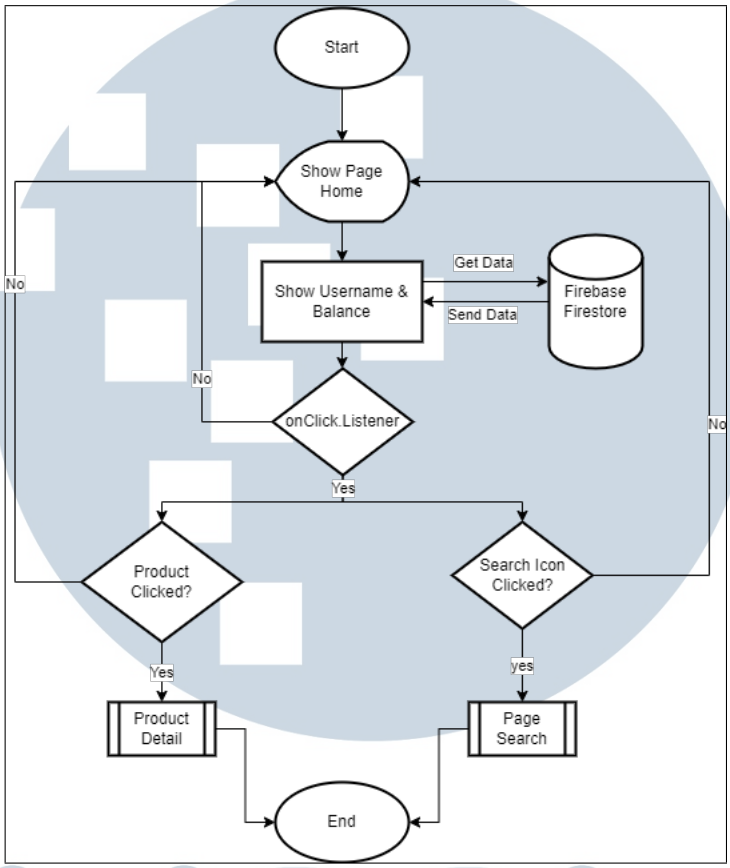
D Halaman Secret ID Aplikasi



Gambar 3.5. Diagram Alir Halaman *Secret ID* Aplikasi

Gambar 3.5 adalah proses diagram alir halaman *Secret ID*. Halaman ini akan menampilkan *id* rahasia pengguna serta teks mengenai fungsi *id* rahasia tersebut. Pada bagian bawah akan terdapat tombol yang akan memunculkan notifikasi konfirmasi serta pesan untuk memastikan pengguna sudah menyimpan *id* rahasia tersebut. Jika sudah pengguna akan diarahkan menuju halaman utama aplikasi.

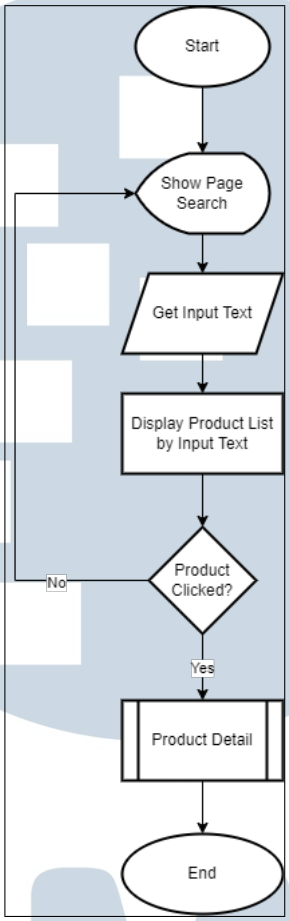
E Halaman Utama Aplikasi



Gambar 3.6. Diagram Alir Halaman *Home* Aplikasi

Gambar 3.6 adalah proses diagram alir halaman utama. Halaman ini akan menampilkan *username* pengguna serta saldo yang dimiliki dari Firebase Firestore. Pada halaman utama ini terdapat beberapa pilihan yang dapat dilakukan yaitu melakukan klik terhadap kumpulan produk yang ditampilkan, teks *hair*, teks *face*, dan teks *skin* yang akan mengarahkan pengguna ke halaman detail produk sesuai dengan produk yang dipilih oleh pengguna. Kemudian terdapat juga ikon *search* yang akan mengarahkan pengguna ke halaman *search* apabila di klik.

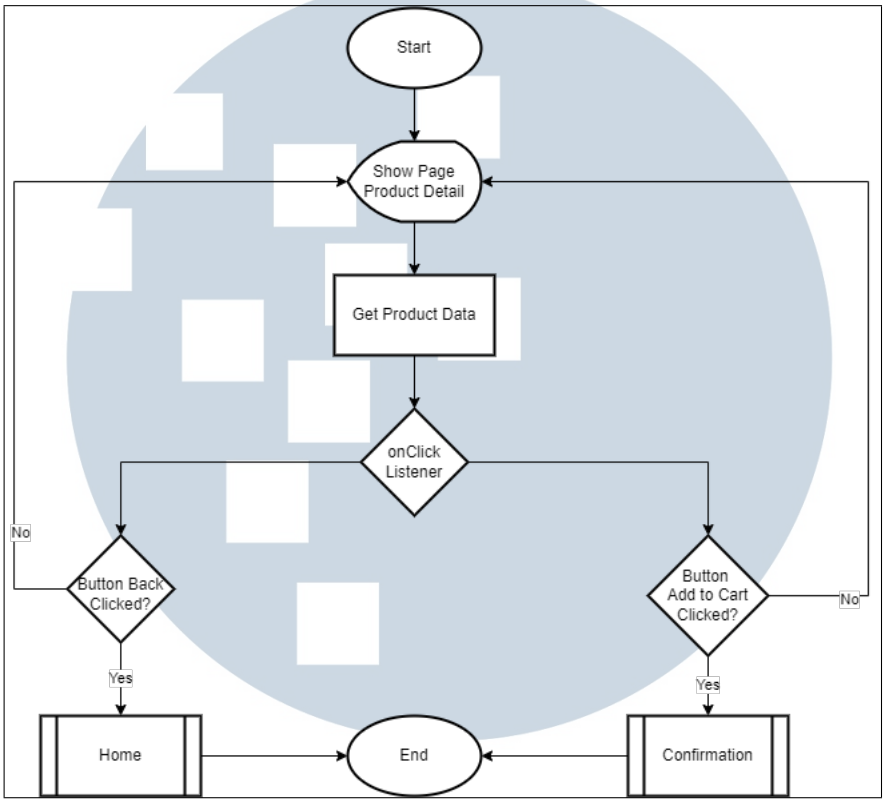
F Halaman Search Aplikasi



Gambar 3.7. Diagram Alir Halaman *Search* Aplikasi

Gambar 3.7 adalah proses diagram alir halaman *search*. Halaman ini akan menampilkan kumpulan dari produk-produk yang ada pada aplikasi Store pada bagian awal, pengguna dapat memasukkan teks pada kolom pencarian untuk mencari produk yang diinginkan lalu menampilkannya ke pengguna. Produk yang dipilih akan mengarahkan pengguna ke halaman detail produk sesuai pilihan pengguna.

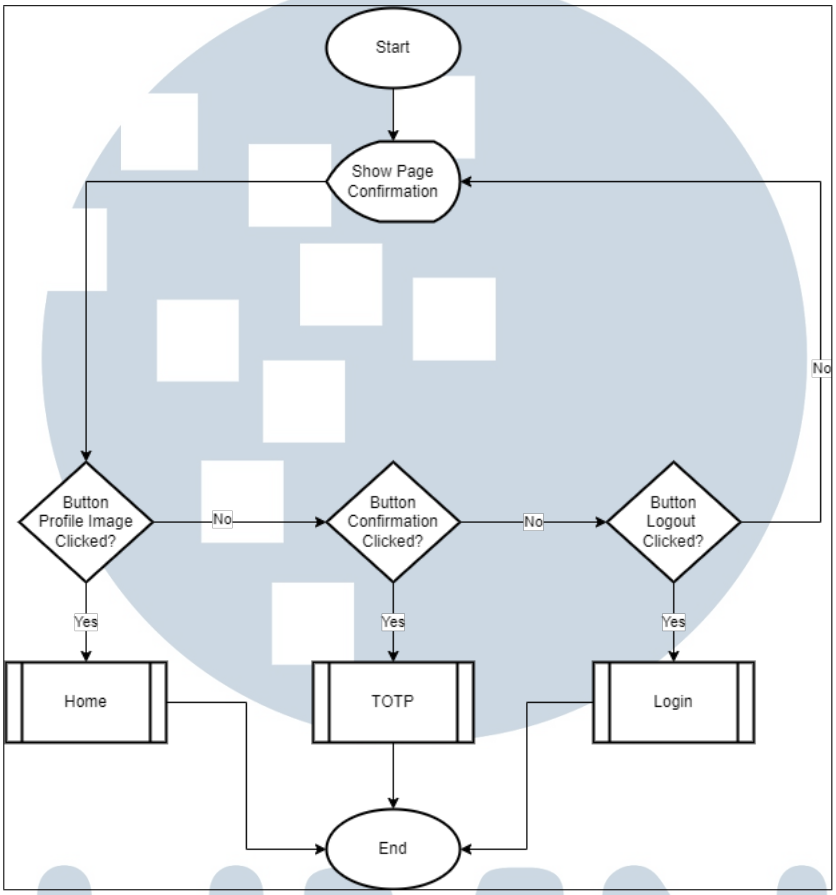
G Halaman Detail Produk Aplikasi



Gambar 3.8. Diagram Alir Halaman *Detail* Produk Aplikasi

Gambar 3.8 adalah proses diagram alir halaman detail produk. Pada halaman ini, pengguna dapat melihat informasi mengenai produk yang mereka pilih dengan lengkap. Halaman ini juga memiliki 2 tombol yaitu tombol *back* dan tombol *Add to Cart*. Tombol *back* akan mengarahkan pengguna kembali ke halaman utama, sedangkan tombol *Add to Cart* akan mengarahkan pengguna ke halaman konfrimasi.

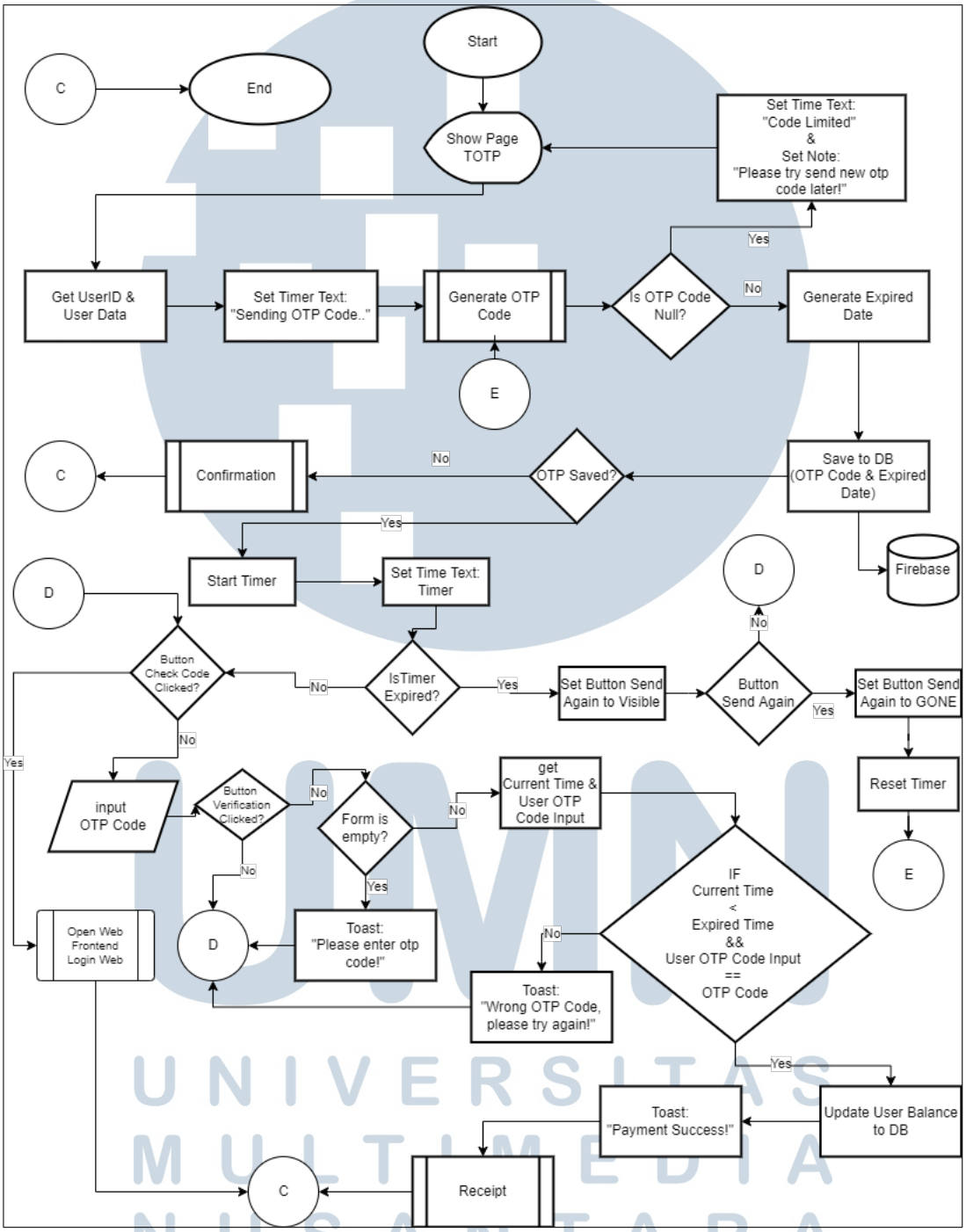
H Halaman Konfirmasi Aplikasi



Gambar 3.9. Diagram Alir Halaman Konfirmasi Aplikasi

Gambar 3.9 adalah proses diagram alir halaman konfirmasi. Halaman ini akan menampikan pesan teks dalam aplikasi kepada pengguna dilengkapi dengan tombol untuk melakukan konfirmasi pembayaran tersebut yang apabila di klik akan menampilkan halaman TOTP. Pengguna juga dapat kembali ke halaman utama dengan melakukan klik pada gambar profil, atau melakukan *logout* dengan tombol *logout* yang akan mengarahkan pengguna ke halaman *login*.

I Halaman TOTP

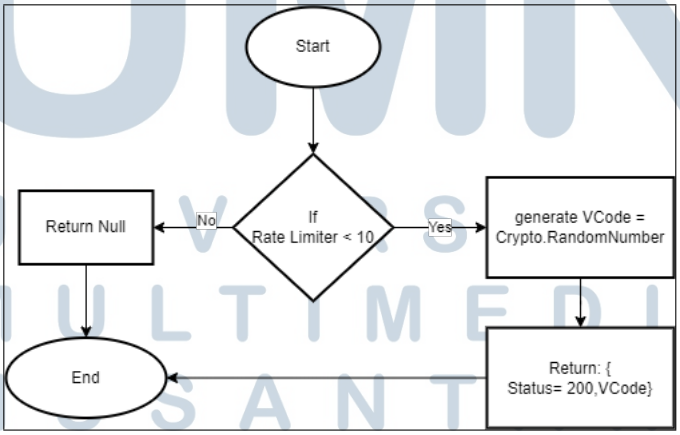


Gambar 3.10. Diagram Alir Halaman *TOTP* Aplikasi

Gambar 3.10 adalah proses diagram alir halaman TOTP. Kode One-Time Password akan di-*generate* untuk pengguna yang prosesnya dapat dilihat pada

diagram alir halaman *generate OTP Code*. Kemudian kode yang didapatkan dicek oleh sistem, jika kode yang diterima kosong maka pengguna akan mendapatkan pesan *code limited* dan pengguna diminta untuk melakukan verifikasi beberapa saat lagi. Kode yang tidak kosong akan diberikan durasi waktu kadaluarsa untuk kode tersebut lalu disimpan kedalam Firebase Firestore Database serta menampilkan teks perhitungan mundur waktu kadalurasa kode OTP kepada pengguna di halaman tersebut. Ketika durasi waktu otp sudah kadaluarsa, teks dan tombol untuk mengirimkan kode otp yang baru akan ditampilkan kepada pengguna yang akan hilang setelah diklik oleh pengguna serta menghentikan tampilan teks perhitungan mundur. Pengguna dapat mengecek kode yang didapat pada halaman web melalui tombol *check code* yang akan mengarahkan pengguna membuka *web browser* berisikan alamat web pengecekan kode. Setelah mendapatkan kode OTP, pengguna dapat memasukkan nya pada kolom *input text*. Kode OTP yang dimasukkan akan dicek oleh sistem, jika *input text* kosong maka pengguna akan mendapatkan teks untuk mengecek kembali kode OTP yang dimasukkan, jika terisi sistem akan melakukan verifikasi pada kode tersebut. Jika kode tidak *valid* pengguna akan mendapatkan pesan bahwa kode yang dimasukkan tidak *valid* dan diminta untuk mengecek kembali kode yang dimasukkan. Jika *valid*, saldo pengguna akan diperbarui serta mendapatkan teks bahwa transaksi berhasil kemudian pengguna diarahkan menuju halaman *receipt*.

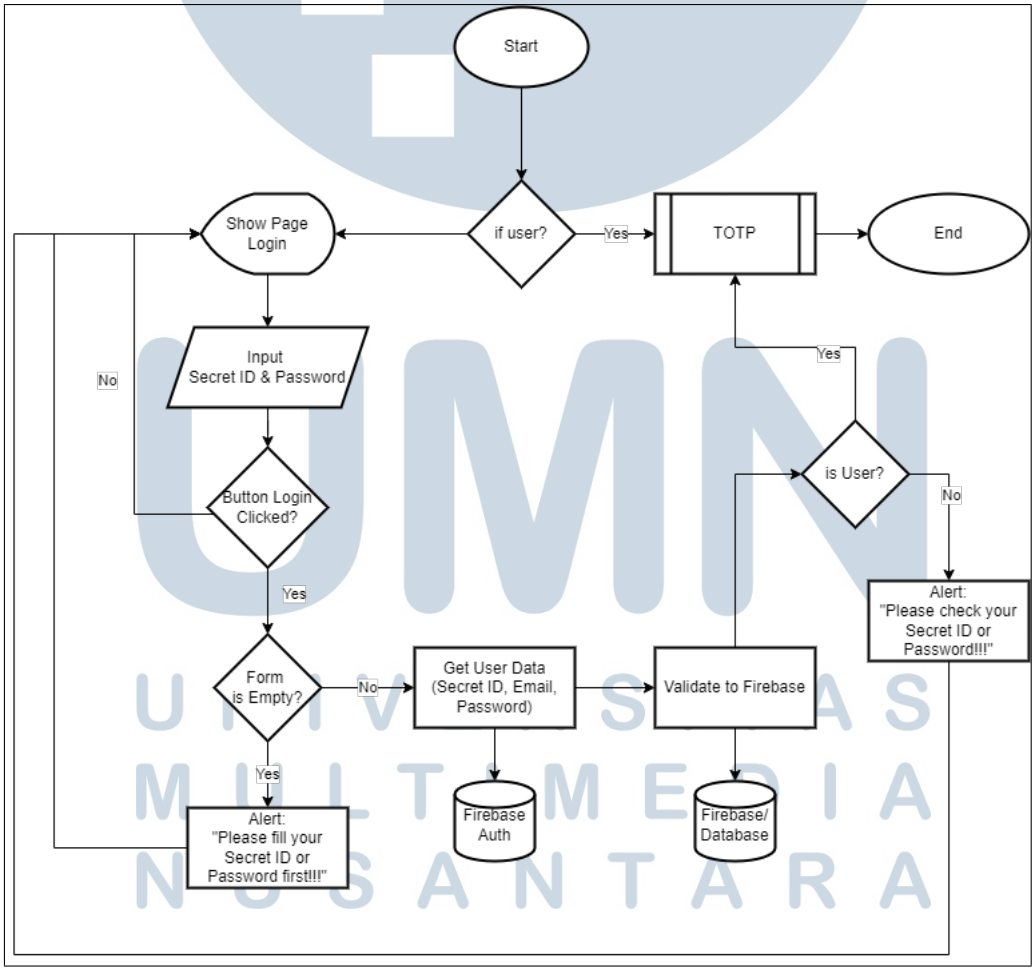
J Algoritma Generate OTP Code



Gambar 3.11. Diagram Alir Halaman *Generate OTP Code*

Gambar 3.11 adalah proses diagram alir halaman *generate OTP Code*. Halaman ini akan memproses permintaan untuk kode OTP, sebelum kode OTP diberikan, sistem akan melakukan pengecekan terhadap *IP Address* pengguna untuk meminimalisir terjadinya *spam* yang menyebabkan sistem menjadi lambat dan rusak dengan membatasi pengguna hanya dapat meminta maksimal 10 permintaan kode OTP dalam kurung waktu 15 menit. Kode OTP akan diciptakan dengan algoritma *random* menggunakan bantuan dari *library* bernama crypto sehingga meminimalisir prediksi kode OTP yang diciptakan. Kode OTP yang berhasil diciptakan akan diberikan kembali kepada aplikasi pengguna untuk digunakan pada proses selanjutnya.

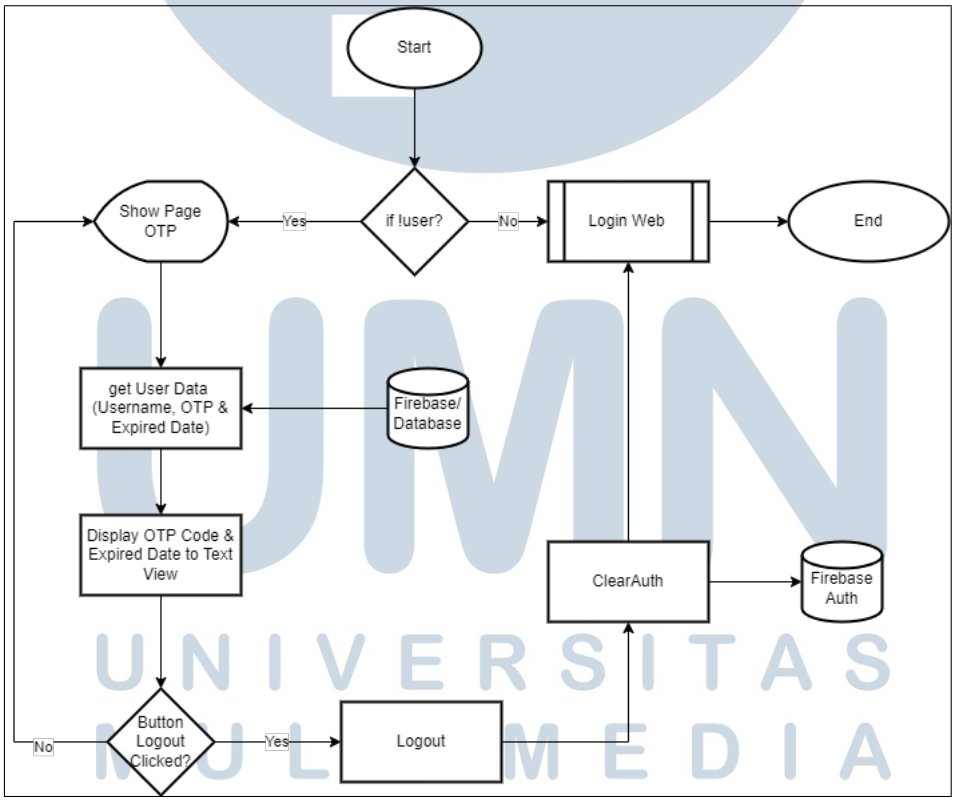
K Halaman Login Web



Gambar 3.12. Diagram Alir Halaman *Login* Web

Gambar 3.12 adalah proses diagram alir halaman *login web*. Sistem terlebih dahulu akan melakukan pengecekan dan mengklasifikasi mengakses web adalah pengguna yang terdaftar atau tidak. Jika bukan pengguna maka sistem akan menampilkan halaman *login*. Pada halaman *login* memiliki 2 kolom *input text* yaitu kolom *Secret ID* dan kolom *Password*. Kemudian dari kolom *input text* yang diisi akan dilakukan pengecekan oleh sistem apakah kolom tersebut kosong atau terisi, jika kosong maka sistem akan menampikan pesan untuk mengisi kolom yang kosong. Jika terisi maka sistem akan melanjutkan validasi pengguna kepada Firestore, validasi pengguna yang tidak berhasil akan mendapatkan teks untuk mengecek kembali terhadap kolom *Secret ID* dan kolom *Password*. Pengguna yang berhasil di validasi akan diarahkan menuju ke halaman TOTP Web.

L Halaman TOTP Web

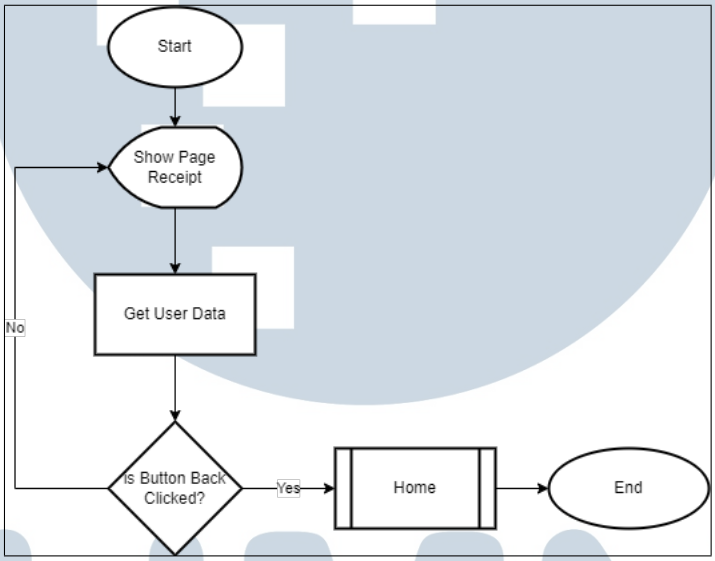


Gambar 3.13. Diagram Alir Halaman TOTP Web

Gambar 3.13 adalah proses diagram alir halaman *TOTP Web*. Sistem terlebih dahulu akan melakukan pengecekan dan mengklasifikasi mengakses web adalah pengguna yang terdaftar atau tidak. Jika bukan pengguna maka sistem

akan mengembalikan pengguna tersebut ke halaman *login*, pengguna yang berhasil *login* akan mendapatkan tampian halaman OTP. Sistem akan mengambil data pengguna terkait kode OTP dan durasi kadaluarsa melalui Firebase Firestore lalu menampilkannya dalam bentuk teks kepada pengguna. Halaman ini juga terdapat tombol *logout* yang berfungsi untuk menghapus sesi pengguna dan mengembalikan pengguna ke halaman *login*.

M Halaman Receipt Aplikasi



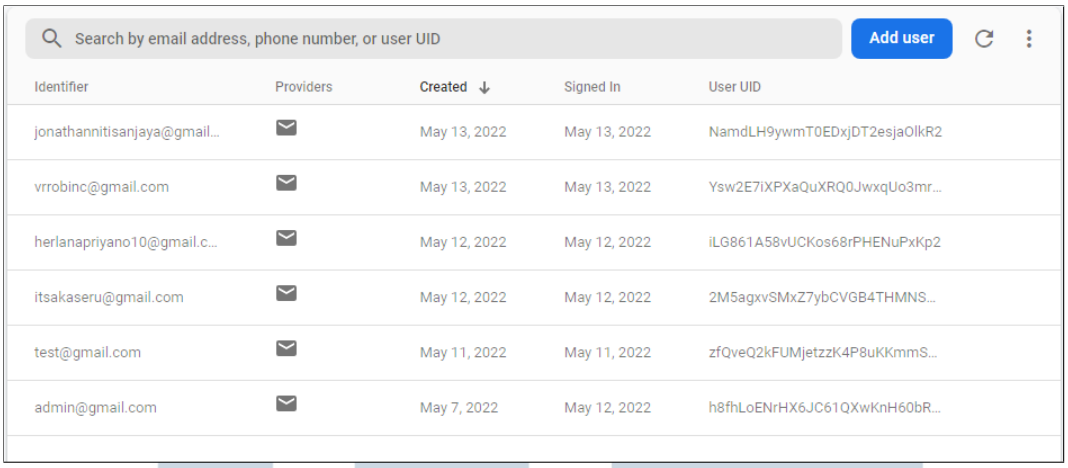
Gambar 3.14. Diagram Alir Halaman *Receipt* Aplikasi

Gambar 3.14 adalah proses diagram alir halaman *receipt* aplikasi. Sitem akan mengambil serta menampikan data terbaru dari pengguna terutama saldo yang dimiliki pengguna. Halaman ini juga memiliki tombol untuk kembali menuju halaman utama.

3.2.2 Struktur Database

A Firebase Authentication

Firebase Authentication digunakan untuk penyimpanan akun pengguna yang telah dibuat dalam perangkat lunak yang dilengkapi dengan berbagai macam cara *login* seperti menggunakan nomor telepon, alamat *email*, akun Google, dll.



Gambar 3.15. Firebase Authentication

Berikut ini penjelasan dari Firebase Authentication:

(a) Identifier, berisi email yang terdaftar.

(b) Providers, berisi info bahwa ID terdaftar sebagai suatu jenis (dalam penelitian ini menggunakan *email*).

(c) Created, berisi tanggal ID pengguna telah terdaftar.

(d) Signed In, berisi info terakhir ID masuk ke dalam perangkat lunak.

(e) User UID, ID dari pengguna yang juga digunakan sebagai kunci utama (*Primary Key*).

B Firebase Firestore Database/Cloud Firestore

Firebase Firestore Database/Cloud Firestore ini digunakan sebagai alat komunikasi antara *platform* (*mobile* dan web) untuk mengakses atau mengubah data secara *real time*.

B.1 Database User Data

Tabel 3.1. Firebase Firestore Database User Data

Nama Kolom	Tipe Data	Keterangan
user_name	String	nama panggilan pengguna
email	String	alamat email yang aktif untuk verifikasi *register* dan *login*
mobile_number	String	nomor telepon aktif pengguna
balance	Number	jumlah saldo pengguna dalam aplikasi

Tabel 3.1 berisikan data informasi pribadi pengguna.

B.2 Database User Second

Tabel 3.2. Firebase Firestore Database User Second

Nama Kolom	Tipe Data	Keterangan
email	String	alamat email yang aktif untuk verifikasi *register* dan *login*
expired_date	Number	waktu masa berlaku kode otp
otp_code	String	kode OTP yang diperlukan pengguna untuk melakukan verifikasi pembayaran

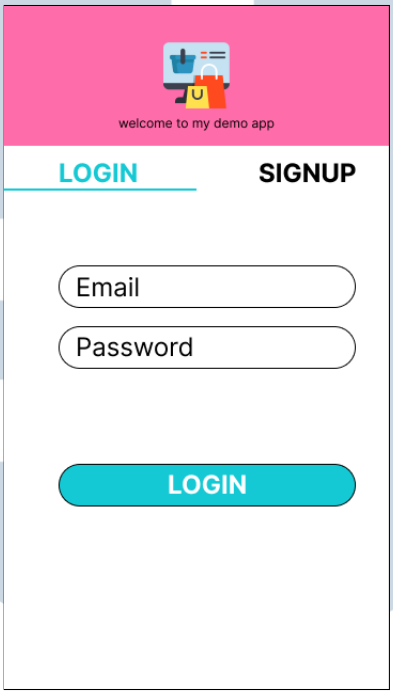
Tabel 3.2 berisikan data pengguna terkait proses verifikasi OTP yang dilakukan. Pada tabel ini, ID dari tabel merupakan *Secret ID* yang dibuat oleh pengembang serta digunakan pengguna untuk mengakses data OTP mereka.

3.2.3 Mockup Aplikasi

Berikut ini adalah *mockup* dari aplikasi Store yang akan digunakan sebagai landasan utama dalam proses pengimplementasian dan pengembangan aplikasi. Beberapa desain yang telah dibuat, memiliki kemiripan dengan desain aplikasi terbuka lainnya.[22]

A Aplikasi Mobile

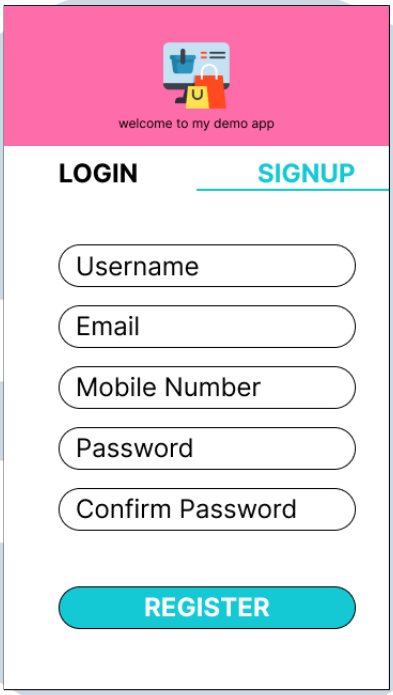
A.1 Halaman Login



Gambar 3.16. *Mockup* Halaman *Login*

Halaman ini akan ditampilkan pada awal atau saat pertama kali pengguna membuka aplikasi *mobile* store. Halaman *Login* ini, terdapat logo aplikasi disertai judul kecil mengenai aplikasi Store. Kemudian diikuti dengan 2 kolom teks untuk mengisi *email* dan *password* serta terdapat tombol *Login* untuk menjalankan fungsi *login* yang akan mengarah ke halaman utama (*home*) apabila pengguna berhasil melakukan *login*. Apabila pengguna belum mempunyai akun, pengguna dapat melakukan registrasi dengan menggeser halaman dari arah kanan ke kiri untuk menuju halaman registrasi.

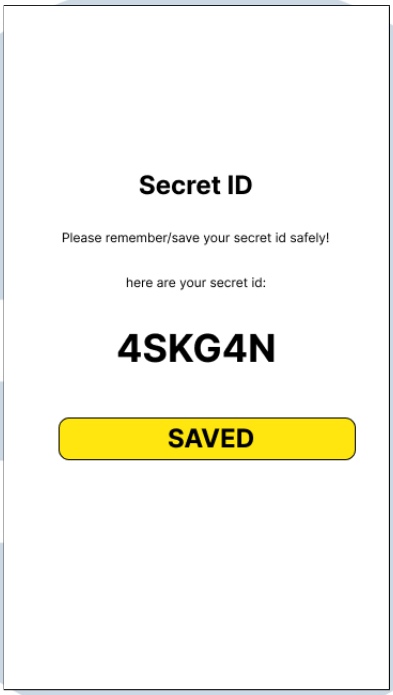
A.2 Halaman Register



Gambar 3.17. *Mockup* Halaman *Register*

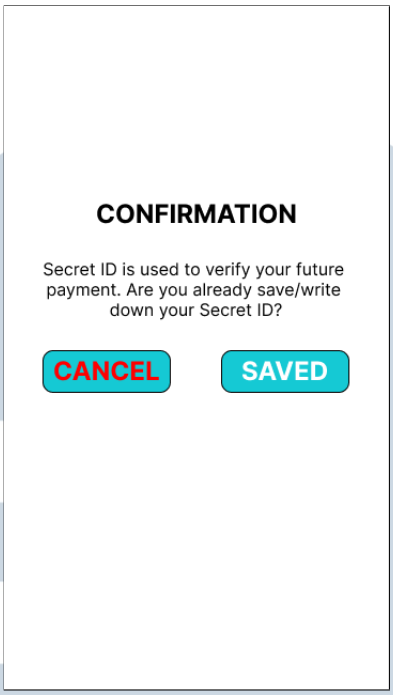
Sama seperti pada halaman *Login*, pada halaman ini terdapat logo aplikasi disertai judul kecil mengenai aplikasi Store. Kemudian diikuti dengan 5 kolom teks untuk mengisi *username, email, mobile number, password,* dan konfirmasi *password* serta tombol dengan teks *register* untuk menjalankan fungsi pendaftaran akun pengguna ke *database* aplikasi. Apabila berhasil terdaftar maka, pengguna akan otomatis ter-*login* dan menuju ke halaman Secret ID.

A.3 Halaman Secret ID



Gambar 3.18. *Mockup* Halaman *Secret ID*

Halaman ini berisikan Secret ID dari pengguna yang bersifat rahasia dan unik. Tombol dengan teks *saved* akan membawa pengguna membuka sebuah notifikasi untuk melakukan validasi apakah pengguna benar-benar sudah mengingat atau mencatat Secret ID mereka.



Gambar 3.19. *Mockup* Halaman Konfirmasi *Secret ID*

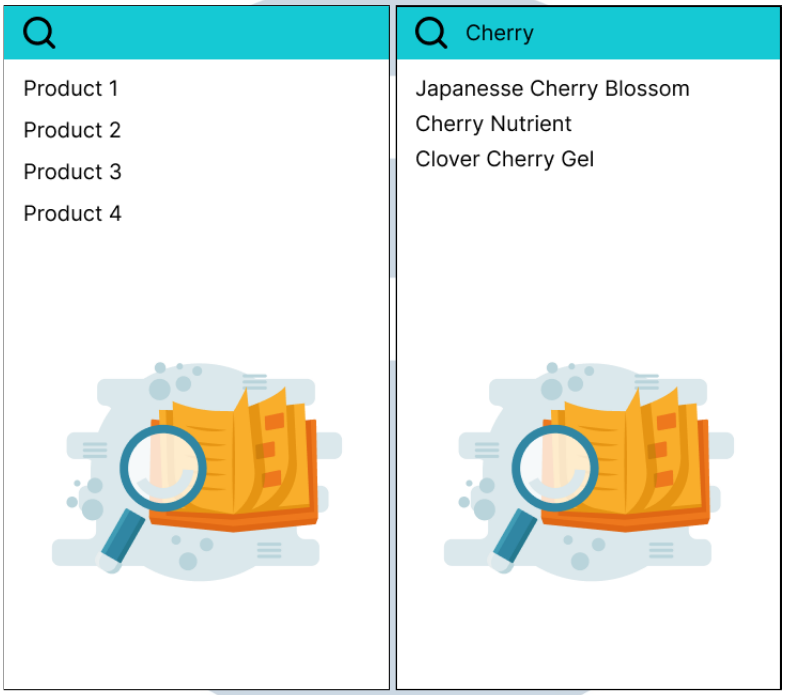
Notifikasi konfirmasi Secret ID ini berisikan pesan pengingat sekaligus konfirmasi kepada pengguna mengenai Secret ID mereka. Terdapat 2 tombol yaitu *cancel* dan *saved*, ketika tombol *cancel* ditekan maka pengguna akan dikembalikan ke halaman Secret ID untuk mencatat kembali kode rahasia tersebut. Lalu tombol *saved* akan mengarahkan pengguna menuju ke halaman utama dari aplikasi Store.

A.4 Halaman Home

Gambar 3.20. *Mockup* Halaman *Home*

Halaman *home* atau halaman utama akan menampilkan data pengguna dan produk yang ada. Pada bagian atas halaman, terdapat nama pengguna, foto profil, dan jumlah saldo yang bisa dipakai untuk berbelanja di dalam aplikasi. Kemudian terdapat kumpulan produk dari aplikasi Store beserta harga dan ukurannya yang dapat dilihat detail nya apabila di klik oleh pengguna yang akan mengarahkan pengguna ke halaman detail produk sesuai produk yang dipilih/tekan.

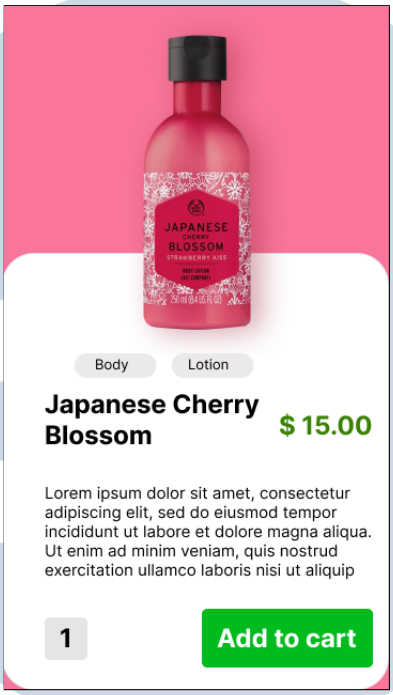
A.5 Halaman Search



Gambar 3.21. *Mockup* Halaman *Search*

Halaman ini akan menampilkan *list* dari produk yang ada di aplikasi Store dan filter berdasarkan masukkan pengguna di kolom pencarian. Apabila produk yang dipilih ditekan maka, pengguna akan di arahkan ke halaman detail produk sesuai produk yang dipilih.

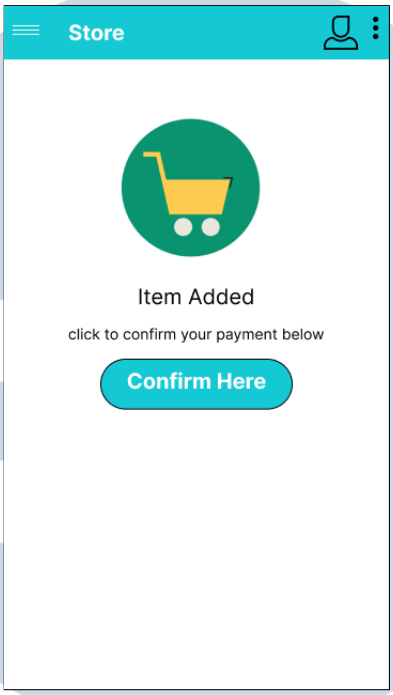
A.6 Halaman Detail Product



Gambar 3.22. *Mockup* Halaman Detail *Product*

Halaman ini berisikan detail produk secara lengkap seperti nama, penjelasan singkat, jenis produk, dan harga. Kemudian pada bagian bawah terdapat tombol *add to cart* yang berarti menambahkan produk ini ke dalam keranjang belanja, tombol ini juga akan mengarahkan pengguna ke halaman konfirmasi.

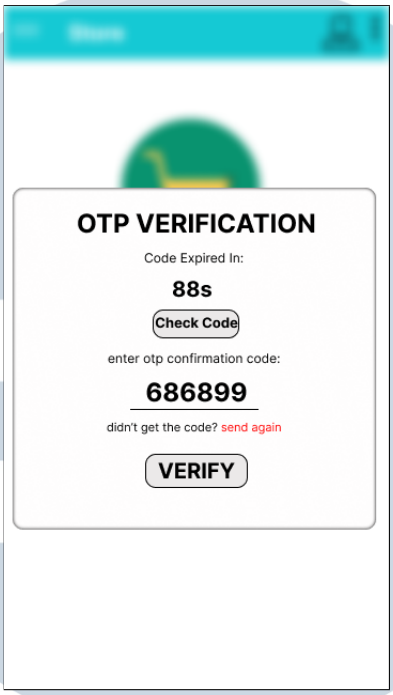
A.7 Halaman Konfirmasi



Gambar 3.23. *Mockup* Halaman Konfirmasi

Halaman ini akan menampilkan informasi terhadap produk yang sudah berhasi ditambahkan, pengguna kemudian akan diminta untuk melakukan konfirmasi pembayaran melalui tombol *confirm here*. Tombol *confirm here* akan memunculkan halaman OTP untuk melakukan verifikasi pembayaran oleh pengguna.

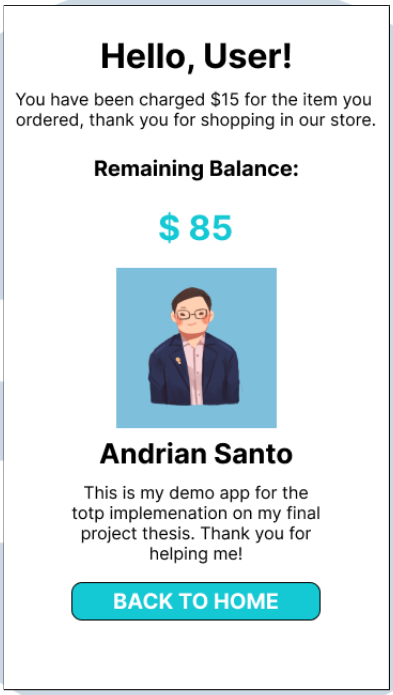
A.8 Halaman TOTP



Gambar 3.24. *Mockup* Halaman *OTP*

Halaman ini akan berisikan *timer* dari masa berlaku kode OTP yang diberikan, dibawahnya terdapat tombol yang mengarahkan pengguna untuk membuka web tempat mengecek kode OTP untuk melakukan verifikasi. Kemudian terdapat kolom masukan untuk kode OTP yang diterima dengan maksimal 6 jumlah karakter. Setelah kolom teks tersebut, terdapat tombol untuk meminta pengiriman kode OTP baru yang akan muncul setelah masa berlaku kode OTP sebelumnya habis/selesai. Pada bagian paling bawah terdapat tombol verifikasi yang akan menjalankan fungsi verifikasi terhadap kode OTP yang dimasukan, apabila kode otp benar maka pembayaran/transaksi telah berhasil, kemudian pengguna akan diarahakan ke halaman *receipt*.

A.9 Halaman Receipt

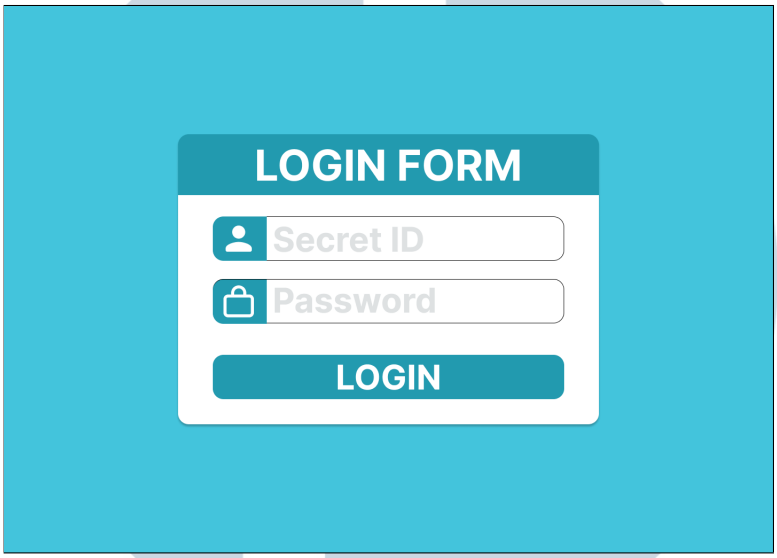


Gambar 3.25. *Mockup* Halaman *Receipt*

Halaman ini akan berisikan pesan mengenai transaksi yang telah berhasil dilakukan. Pengguna akan mendapat informasi mengenai sisa saldo di akun mereka. Halaman ini juga akan menampilkan kalimat terimakasih dari pengembang kepada pengguna yang telah membantu dalam penelitian ini, juga disertai tombol untuk kembali ke halaman utama.

B Aplikasi Web

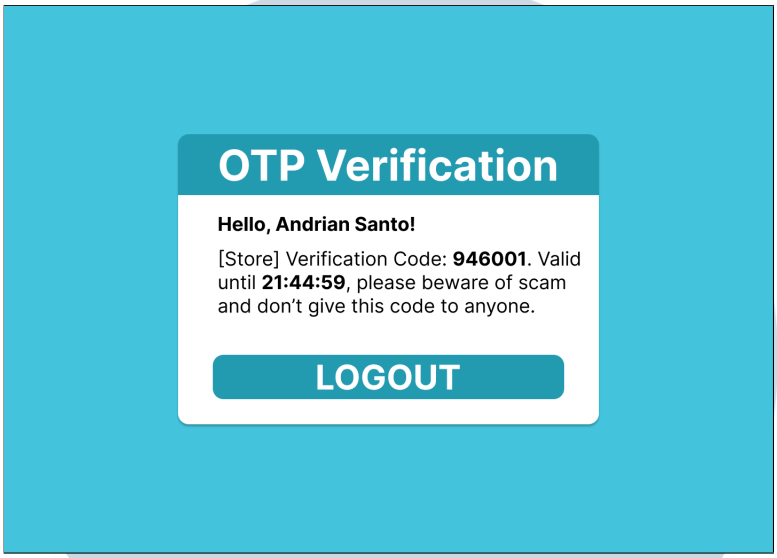
B.1 Halaman Login



Gambar 3.26. *Mockup* Halaman *Login*

Halaman *login* web ini berisikan 2 kolom teks yang harus diisi dengan Secret ID dan *Password*. Kemudian pada bagian bawah terdapat tombol *Login* dimana akan menjalankan fungsi *login* berdasarkan masukan pada 2 kolom teks tersebut. Jika berhasil melakukan *Login*, Pengguna akan diarahkan ke halaman OTP.

B.2 Halaman TOTP



Gambar 3.27. *Mockup* Halaman *TOTP*

Halaman ini pengguna dapat melihat *username* mereka serta kode OTP yang diberikan oleh aplikasi Store dengan waktu masa berlaku kode tersebut. Teks juga dilengkapi dengan peringatan kepada pengguna untuk tidak memberikan kode OTP ke siapapun. Pada bagian bawah terdapat tombol *Logout* yang akan menjalankan fungsi *logout* untuk mengeluarkan pengguna dari halaman tersebut.