



### Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

#### **BAB III**

#### ANALISIS DAN PERANCANGAN SISTEM

#### 3.1. Metodologi Penelitian

Metode penelitian yang digunakan untuk melakukan penelitian ini antara lain sebagai berikut:

#### 3.1.1 Studi Literatur

Dalam studi literatur dilakukan pembelajaran terhadap pembaruan yang ada pada bahasa pemrograman Swift yaitu Swift 4, pembelajaran artikel dan *paper* mengenai penggunaan metode *fuzzy time series*, peramalan dan karakteristik dari pengeluaran personal, dokumentasi cara penggunaan Shiny R, proyek yang menyerupai, serta situs-situs web yang mendukung untuk dijadikan pedoman penelitian.

#### **3.1.2 Pencarian Sampel**

Sampel yang didapatkan harus berada dalam kriteria populasi mahasiswa rantau, atau yang belum bisa mengatur pengeluarannya, pada Fakultas Teknik dan Informatika Universitas Multimedia Nusantara, dan bersedia memberikan data transaksi harian selama satu bulan penuh.

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

#### 3.1.3 Pengumpulan Data

Data yang diperlukan adalah data pengguna dan data pengeluaran bulanan. Data pengguna berisi data yang ada saat mendaftarkan diri pada Wordpress, seperti *username, email*, kata sandi, nama depan, dan nama belakang. Kemudian data pengeluaran bulanan yang berisi data transaksi yang yang dikumpulkan dari sampel dan dimasukkan dalam beberapa kategori.

Pengumpulan data dari sampel dilakukan dengan menggunakan Google Sheets, kolom yang ada dalam Google Sheets tersebut merupakan tanggal, jenis transaksi, nama transaksi, kategori, dan total dari transaksi tersebut.

Setelah web sudah siap, data digabungkan dari seluruh sampel ke dalam *database* yang ada pada *server*. Selain tabel *default* milik *WordPress*, dua buah tabel ditambahkan untuk menyimpan jumlah uang pengguna saat ini dan seluruh transaksi.

#### 3.1.4 Perancangan Aplikasi dan Web

Pembuatan sistem ini dibagi menjadi pembuatan aplikasi iOS dan pembuatan web. Beberapa diagram dibuat dalam perancangan ini, yaitu diagram blok sistem, diagram alur, *mock-up* aplikasi dan diagram menu.

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



Gambar 3.1. Diagram Blok Sistem.

Pengguna berinteraksi langsung dengan aplikasi iOS, kemudian aplikasi berkomunikasi dengan *server* dengan mengirimkan HTTP *request* ke *end-point* yang diatur oleh WP REST API dan menerima HTTP *response* berupa JSON. Pada server, terdapat WordPress dan *database* MySQL. Kemudian *server* berhubungan dengan *server* Shiny R untuk menjalankan metode *fuzzy time series* dengan menggunakan FTP untuk pertukaran *file* CSV.

# UNIVERSITAS MULTIMEDIA NUSANTARA



saldo, menambah pemasukan dan pengeluarannya, melihat history transaksinya, melihat hasil prediksi, mengubah passwordnya, dan logout dari aplikasi.

Sedangkan apabila pengguna hanya sebatas pengguna aplikasi, maka pengguna tersebut dapat membuat akun atau menekan tombol lupa password.



Alur pada halaman *check login* dimulai dari pengambilan token dari dalam UserDefaults, atau penyimpanan *internal* pada *device*. Apabila tidak ada token pada *device*, maka akan diarahkan ke halaman *login* untuk melakukan *login* kembali.

Apabila terdapat token, token tersebut akan divalidasi melalui tautan jwt/validate. Apabila token tersebut valid, maka akan diarahkan ke halaman *chart*, namun apabila tidak valid, baik karena token tersebut salah, maupun karena sudah kadaluarsa, maka akan diarahkan kembali ke halaman *login*.



Gambar 3.5. Flow chart halaman login.

Alur pada halaman *login* dimulai dari pengambilan input dari pengguna yang berupa *username* dan *password*. Setelah pengguna menekan tombol *login*, maka akan ada validasi dari sistem mengenai apakah ada *field* yang masih kosong atau tidak sesuai. Setelah input tervalidasi, maka *username* dan *password* akan dikirim sebagai HTTP POST *request* ke halaman jwt/token.

Apabila keterangan *login* pengguna sudah benar, maka *response* dari WP REST API akan berupa sebuah token dan disimpan ke dalam UserDefaults untuk digunakan untuk pertukaran data selanjutnya di dalam aplikasi. Apabila *login* gagal, maka akan muncul pesan kesalahan, dan halaman aktif akan dikembalikan ke halaman *login*.



Alur pada halaman *chart* dimulai dari mengambil token pada UserDefaults kemudian mengirimkan token tersebut sebagai *parameter* pada HTTPS POST *request* ke tautan pulltrans untuk mengambil transaksi milik pengguna yang sedang aktif, dan menerima transaksi tersebut sebagai JSON. Kemudian HTTPS POST *request* juga dilakukan ke tautan *balance* untuk mengambil jumlah uang yang dimiliki oleh pengguna saat ini, dan menerima saldo tersebut juga sebagai JSON.

Transaksi yang diterima disimpan ke dalam *array*, kemudian diolah menjadi persentase per kategori, kemudian ditampilkan pada *chart*, beserta legendanya sebagai keterangan warna dan kategori. Kemudian saldo pengguna juga ditampilkan pada label *balance* yang ada.



tombol yang ditekan di sisi label *balance*. Kemudian pengguna memasukkan transaksi yang telah dilakukannya. Sistem melakukan validasi apakah *input* tersebut ada yang kosong atau tidak.

Apabila *input* sudah tervalidasi, kemudian token diambil dari dalam UserDefaults, kemudian transaksi dari formulir tersebut dikirimkan ke tautan addtrans bersama token sebagai parameternya. Setelah transaksi tersimpan maka layar akan kembali lagi ke halaman *chart* dengan data *chart* dan *balance* yang telah ditambahkan.



Gambar 3.8. Flow chart prediksi pada web.

Alur saat membuat prediksi di web adalah *server* yang menerima *request* pada tautan createcsv membuat *array* pengeluaran dari satu pengguna yang sedang login, kemudian isi *array* tersebut dimasukkan ke dalam *file* csv. Kemudian sebuah *script* yang berisi fungsi *fuzzy time series* pada *server* Shiny R akan dijalankan dengan cara membuka sebuah tautan pada shinyapps.io. Tahap pertama adalah *server* Shiny membaca *file* csv yang disediakan kemudian memodifikasi data tanggal, mencari *determine of constant* yang sesuai dengan data yang dimiliki,

kemudian menjalankan fungsi *fuzzy time series* 2 bertipe Abbasov-Mamedova. Setelah hasil prediksi didapatkan, *server* Shiny melakukan FTP pada *server* untuk membuat sebuah *file* csv berisi hasil prediksi tersebut dan diberi nama sesuai nama pengguna.

Alur pada halaman prediksi dimulai dengan mengambil token dari UserDefaults dan mengirimkan token tersebut melalui HTTPS POST *request* ke tautan readcsv yang berfungsi untuk membuka file csv milik pengguna yang sedang login. *Response* yang didapatkan adalah hasil prediksi yang dikirimkan sebagai JSON, kemudian ditampilkan ke label hasil prediksi. Begitupun alur pada halaman *history*, mengirim token ke tautan pulltrans dan responsenya ditampilkan pada tabel.

Alur pada saat *log out* dimulai dengan menghapus token dari UserDefaults sehingga saat aplikasi dibuka lagi tidak akan ada token untuk dicek validasinya pada *device*. Kemudian aplikasi akan kembali ke halaman *login*, agar selanjutnya pengguna dapat *login* kembali dan mendapatkan token baru, atau mengganti pengguna yang aktif, atau sekedar menutup aplikasi. Selain token, *username* dan *last login* juga dihapus.

## UNIVERSITAS MULTIMEDIA NUSANTARA

3	1.4.2. Mock-up Aplikasi	
4	Label Username Label	Username Row
	Password Label	Password Row
	Login button	
	Logo Image	

Gambar 3.9. Mock-up halaman login.

Pada halaman *login* akan ada label yang menerangkan *field* yang berada di sisinya, yaitu *field username* dan *password* yang harus diisi agar dapat masuk ke dalam halaman lain pada aplikasi. Akan ada sebuah tombol yang digunakan untuk validasi *input* dan sebagai *trigger* untuk mengirim keterangan *login* ke *server*. Pada halaman ini juga akan ditampilkan logo aplikasi pada sebuah *image view* di bawah tombol *login*.

## UNIVERSITAS MULTIMEDIA NUSANTARA



Gambar 3.10. Mock-up halaman chart.

Pada halaman *chart* akan ada satu *chart view* yang berisi pembagian persentase pengeluaran pengguna pada setiap kategori pengeluaran. Kemudian di bawahnya akan ada legenda yang menerangkan warna dari setiap kategori yang tidak tertulis.

Kemudian di bawah legenda akan ada label *balance* yang berisi saldo atau jumlah uang yang dimiliki oleh pengguna. Di sisinya terdapat dua buah tombol, yaitu tombol tambah *income* dan tambah transaksi pengeluaran. Pada bagian bawah halaman, ada *bar button* yang bertuliskan *Chart*, *Prediction*, dan *User*, berguna untuk berpindah ke halaman *prediction* dan halaman *user*, *bar button* ini juga dimiliki oleh halaman *prediction* dan halaman *user* untuk saling berpindah.



### Gambar 3.11. *Mock-up* halaman prediksi.

Pada halaman *prediction* akan ada dua *text view* yang bertuliskan prediksi keuangan pengguna pada bulan yang berlangsung, dan prediksi pengeluaran pengguna pada bulan selanjutnya.



Gambar 3.12. Mock-up halaman user.

Pada halaman *user* akan ada tombol *history* untuk berpindah ke halaman *history*. Kemudian akan ditampilkan label *login as* untuk memperlihatkan pengguna yang sedang aktif pada *device* tersebut. Kemudian ada sebuah *tombol log out* untuk menghapus token dari dalam *device*, dan kembali ke halaman *login*.

JUSANTAR



Gambar 3.13. Mock-up halaman history.

Pada halaman *history*, akan ada sebuah tabel yang menampilkan pengeluaran dan pemasukan pengguna yang sudah berlalu. Karena *trigger* halaman ini adalah sebuah *button*, maka untuk kembali ke halaman *user*, pengguna dapat menekan tombol kembali. Apabila menekan *bar button* untuk pindah ke halaman lain dan kembali ke halaman *user*, layar akan tetap pada halaman *history* karena *navigation bar* masih menyimpan posisi layar di halaman *history*.



Gambar 3.14. Diagram Menu.

Hubungan antar menu atau antar halaman yang ada pada aplikasi adalah halaman *check login* dapat berpindah ke halaman *login* atau halaman *chart* sesuai dengan kondisi token yang dimiliki *device*. Kemudian halaman *login* dapat berpindah ke halaman *chart* apabila pengguna sukses melakukan *login*. Kemudian halaman *chart* dapat berpindah ke halaman *prediction* dan halaman *user* karena berada pada satu *navigation bar controller*. Kemudian halaman *user* dapat berpindah ke halaman *history* dengan *trigger button history*, atau kembali ke halaman *login* dengan *trigger button log out*.

SANTA