

# Aplikasi Manajemen Keuangan Pribadi Pada Platform IOS Menggunakan Bahasa Pemrograman Swift

Maulana Muhammad Yusuf Frasha<sup>1</sup>, Agustine Hana Masitoh<sup>2</sup>

<sup>1,2</sup> *Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Gunadarma*

<sup>1</sup> *Jl. Kalibata Selatan No:90C, Pancoran Jakarta Selatan*

<sup>1</sup>[frashaa01@gmail.com](mailto:frashaa01@gmail.com)

<sup>2</sup> *Jl. Margonda Raya No:100, Pondok Cina Depok-16424*

<sup>2</sup>[agustinehana@staff.gunadarma.ac.id](mailto:agustinehana@staff.gunadarma.ac.id)

**Intisari**— Di era globalisasi yang membuat segalanya menjadi mudah, salah satunya adalah bertransaksi. Segala kemudahan tersebut dapat dilakukan hanya dengan menggunakan smartphone dan internet. Dengan segala kemudahan tersebut menyebabkan sifat konsumtif pada masyarakat. Maka pentingnya pengendalian uang yang terencana agar setiap transaksi yang kita lakukan sehari-hari dapat diatur dengan baik. Supaya transaksi sehari-hari dapat diatur dengan baik maka dibuatlah aplikasi mobile yang dapat digunakan pengguna untuk memudahkan pendataan transaksi setiap saat dimana saja dan kapan saja. Metode pembuatan aplikasi menggunakan metode R&D. Untuk pembuatan aplikasi menggunakan bahasa Swift, serta SQLite untuk databasenya. Hasil dari pembuatan aplikasi tersebut akan diuji coba dengan memasukan sejumlah data transaksi pada aplikasi. **Kata kunci**— Letakkan 5 – 6 kata kunci— Manajemen Keuangan, Xcode, iOS, Swift, SQLite, Aplikasi Mobile.

**Abstract**— In the globalization era that make everything easy, one of that is transactioning. All these conveniences can be done only by using a smartphone and the internet. With all these conveniences lead to consumptive behavior in society. The importance of well-planned money control is for every transaction that we do every day can be managed properly. So that daily transactions can be managed properly, it is necessary to create a mobile application that can be used by users to store transaction list at anytime and anywhere. The application development method uses the R&D. Swift programming language and SQLite for databases is used for making the application. The results of making the application will be tested by entering some transaction data into the application.

**Keywords**— Financial Management, Xcode, iOS, Swift, SQLite, Mobile Apps.

## I. PENDAHULUAN

Pada era globalisasi sekarang ini membuat segala hal menjadi lebih mudah, salah satunya adalah bertransaksi. Banyak sekali segala jenis transaksi yang dapat dilakukan, mulai dari transaksi perbankan hingga transaksi non-tunai atau e-wallet yang mana sangat memudahkan konsumen dalam melakukan segala macam transaksi hanya dengan bermodalkan smartphone dan internet.

Berdasarkan hasil survei bertema 'eChannel Fintech eCommerce & eLifestyle', GoPay menempati peringkat pertama sebagai e-money yang paling banyak digunakan, dipilih 81 persen responden, posisi kedua ditempati OVO sebanyak 71 persen. Selanjutnya ada ShopeePay menempati posisi ketiga dengan 44 persen, lalu Dana di posisi keempat dengan 41 persen [1].

Dengan data tersebut dapat dibuktikan adanya banyak pengguna e-wallet di Indonesia yang mana menyebabkan perilaku konsumtif di dalam masyarakat. Maka pentingnya pengendalian uang yang terencana adalah bentuk dari kita

memanajemen keuangan pribadi agar pendapatan dan pengeluaran sehari-hari bisa diatur dengan baik.

Untuk dapat mengatur keuangan dengan mendetail diperlukan bantuan aplikasi agar memudahkan dalam menghitung keluar masuknya uang. Dengan aplikasi manajemen keuangan pribadi diharapkan dapat membantu pengguna untuk mengatur pemasukan dan pengeluaran uang dimana saja dan kapan saja [2].

## II. BACKGROUND/LATAR BELAKANG

Keuangan Pribadi Danes (2012) berpendapat belajar mengelola keuangan (money management) adalah hal yang penting dimiliki oleh seseorang. Pentingnya mengelola keuangan pribadi dapat membantu setiap orang untuk mencapai kepuasan ekonomi.

Dikarenakan kondisi ekonomi tiap orang berbeda, maka pentingnya pengelolaan keuangan yang baik sangat dibutuhkan untuk memenuhi kebutuhan dan tujuan yang telah direncanakan.

Kesimpulan yang dapat diambil adalah mengelola keuangan pribadi merupakan kegiatan yang terstruktur, mulai dari melakukan perencanaan, penganggaran, dan pemeriksaan untuk tiap pemasukan dan pengeluaran uang.

Dengan pengelolaan uang yang baik dapat meningkatkan kesejahteraan karna segala sesuatu yang berkaitan dengan uang sudah sesuai dengan rencana atau tujuan yang diinginkan. Untuk mencapai tujuan yang diinginkan maka diperlukan pengelolaan keuangan yang teliti dan rutin setiap hari. [3] menyampaikan bahwa kemampuan seseorang dalam mengelola keuangan adalah salah satu faktor penting untuk mencapai kesuksesan.

Untuk dapat mengelola keuangan dengan baik diperlukan 2 hal utama yang harus diperhatikan, yaitu pendapatan dan pengeluaran. Bahasa pemrograman berjenis high-level programming (syntax dan struktur mudah dipahami karena menggunakan bahasa yang dimengerti manusia). Bersifat client-side (hanya membutuhkan browser untuk menguji Javascript).

Berorientasi object (cocok buat Anda yang ingin masuk ke dalam konsep pemrograman berbasis objek). Bersifat loosely typed (tidak membutuhkan deklarasi variabel terlebih dulu).

Swift merupakan bahasa pemrograman berbentuk objek fungsional (Functional – Object) yang dikembangkan Apple Inc. Untuk mengembangkan aplikasi di dalam platform khusus buatan Apple seperti Mac, Apple TV, Apple Watch, dan iOS. Swift bersifat sumber terbuka (open source) dan dirancang agar dapat bekerja dengan Objective-C sehingga dapat meminimalisir kesalahan kode dalam program. Diperkenalkan oleh Apple pertama kali pada tahun 2014 pada acara tahunan Apple bernama WWDC (Worldwide Developer Conference) [4].

Swift dibangun dengan compiler LLVM (Low Level Virtual Machine) yang sudah termasuk di dalam Xcode 6 beta. Sekarang Swift berada pada versi 5.5 dengan Xcode versi 13. Sebagai bahasa pemrograman baru, Swift memiliki banyak kesamaan dalam hal baik dengan bahasa pemrograman lain dan juga menghindari kekurangan yang ada pada bahasa pemrograman lainnya. Swift dapat di kompilasi (compiled) dengan baik dan cepat daripada bahasa pemrograman lain dikarenakan Swift tidak diterjemahkan (interpreted) atau dikelola (managed). Metode kompilasi pada Swift menghasilkan native application (program asli) yang berjalan langsung pada mesin dan sistem operasi secara langsung tanpa bantuan penerjemah (interpreter) maupun VM (virtual machine) [5].

Karna bersifat open source bahasa Swift dapat digunakan pada perangkat lain dan tidak terbatas hanya untuk pengguna Mac. Beberapa sistem operasi umum yang didukung untuk menggunakan bahasa Swift selain macOS diantaranya Windows 10, Ubuntu 16, Ubuntu 18, dan Ubuntu 20. Swift juga memiliki banyak fitur yang baik untuk sebuah bahasa pemrograman modern digunakan untuk membuat aplikasi berbasis mobile, baik itu Android maupun iOS. React Native adalah sebuah framework berbasis JavaScript yang merupakan sekumpulan library berbasis JavaScript yang dikembangkan oleh Facebook. Syntax dari React Native

merupakan gabungan antara JavaScript dan XML yang dapat disebut JSX [6].

### III. METODOLOGI PENELITIAN

Tipe Metode Research And Development (R&D) merupakan metode yang digunakan untuk menghasilkan produk tertentu dan menguji keefektifan produk tersebut. Produk tersebut tidak selalu berbentuk benda atau perangkat keras (Hardware), seperti buku, alat tulis, dan alat pembelajaran lainnya. Akan tetapi, dapat pula dalam bentuk perangkat lunak (Software) [7].

Langkah-langkah yang dilakukan untuk rancangan penelitian Research And Development (R&D) adalah sebagai berikut: (1)Potensi dan Masalah, (2) Pengumpulan Data, (3) Desain Produk, (4)Validasi Desain, (5)Revisi Desain, (6)Uji Coba Produk, (7)Revisi Produk, (8)Uji Coba Pemakaian, (9)Revisi Produk, (10) Produksi Masal.

Langkah-langkah penelitian yang dilakukan oleh penulis menurut metode penelitian yang dilakukan adalah sebagai berikut:

#### 1. Potensi dan Masalah

Penelitian ini berawal dari permasalahan yang ada di lokasi penelitian. Potensi dan Masalah merupakan segala sesuatu yang bila dimanfaatkan dengan baik akan mendapatkan nilai tambah dan masalah dapat terjadi apabila tujuan tidak sesuai dengan yang diharapkan. Masalah dapat diatasi melalui metode R&D dengan cara meneliti, sehingga dapat ditemukan solusi, pola atau penanganan yang efektif, dan dapat di gunakan untuk menyelesaikan masalah tersebut. Potensi dan masalah yang disebutkan pada penelitian ini mengenai pendataan inventaris barang yang masih manual, untuk masalah arsip dan pelaporan berupa kertas dan tidak tertata rapi, barang hilang dan tidak terawat, peminjaman dan pengembalian barang tidak teratur. Potensi dan masalah yang disebutkan diteliti agar dapat meningkatkan kinerja kepala tata usaha dalam menangani inventaris di sekolah dan mengurangi masalah-masalah yang ada.

#### 2. Pengumpulan Informasi

Setelah potensi dan masalah dapat ditunjukkan secara fakta, selanjutnya dikumpulkan berbagai informasi dari narasumber maupun ahli-ahli di bidang tersebut dan studi literatur yang dapat digunakan sebagai bahan untuk perancangan produk, diharapkan dapat mengatasi masalah yang ada. Studi ditunjukkan untuk menemukan konsep-konsep teori yang memperkuat suatu produk, khususnya mengenai inventaris, misalnya produk berbentuk model, program, sistem, pendekatan, sistem dan lain sebagainya. Studi literatur juga diperlukan untuk mengetahui langkah-langkah paling tepat dalam pengembangan produk.

#### 3. Desain Produk

Untuk menghasilkan sistem kerja baru, harus melakukan perancangan kerja baru berdasarkan penilaian terhadap kinerja sistem lama, sehingga dapat ditemukan kelemahan-kelemahan terhadap sistem tersebut. Disamping itu dilakukan penelitian terhadap bagian-bagian lainnya. Selain itu harus mengkaji referensi yang terkait dengan sistem kerja yang modern. Hasil akhir dari kegiatan ini berupa desain produk baru, desain yang

ingin dicapai adalah sistem yang mudah dipahami dan mampu mengoptimalkan sistem atas peminjaman serta pengembalian inventaris sekolah dan mengoptimalkan laporan yang lebih efisien [8].

Validasi Desain

Validasi desain merupakan kegiatan untuk menilai apakah rancangan produk akan lebih efektif di bandingkan dengan cara kerja sistem lama. Validasi di sini masih bersifat penilaian dengan pemikiran dan tanggapan secara pemikiran yang rasional, belum fakta secara lapangan, yang artinya produk masih berbentuk rancangan-rancangan sementara [9].

5. Perbaikan Desain

Setelah dilakukannya validasi melalui diskusi dengan unit-unit yang terkait, maka akan dapat diketahui kelemahan-kelemahannya. Setelah di ketahui kelemahannya maka selanjutnya peneliti mencoba untuk mengurangi atau menambahkan kekurangan pada desain tersebut yang kemudian produk di lakukan uji coba.

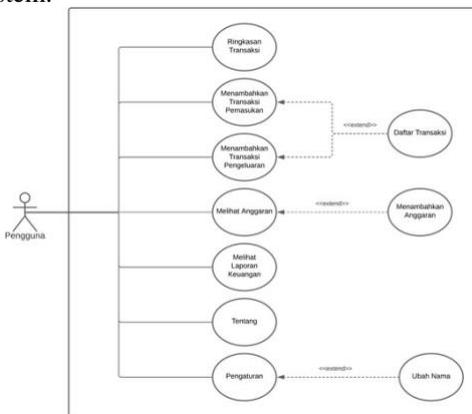
6. Ujicoba Produk

Desain produk yang telah dibuat tidak bisa langsung dapat diuji coba, akan tetapi harus dibuat terlebih dahulu, menghasilkan produk, dan pada akhirnya produk tersebut diuji coba [10] :

Perancangan aplikasi dengan menggunakan UML sangat diperlukan untuk memberikan gambaran jelas bagaimana aplikasi bekerja. Diagram UML yang akan digunakan yaitu Use Case Diagram, Sequence Diagram, dan Class Diagram dengan rancangan sebagai berikut.

1. Use Case Diagram

Use Case diagram dibuat untuk memberikan gambaran visual bagaimana actor bernama pengguna dan aplikasi dapat berinteraksi. Relasi yang digunakan adalah association tanpa panah, karena pengguna akan secara aktif berinteraksi dengan sistem.



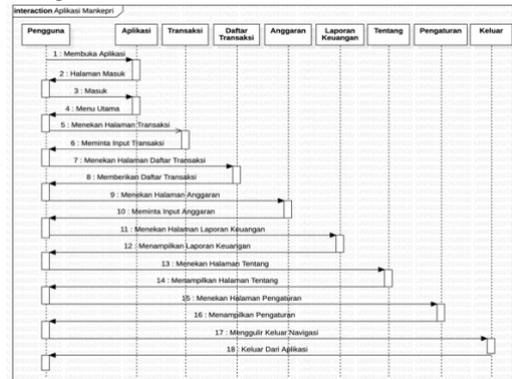
Gambar 1. Use Case Diagram Aplikasi

2. Sequence Diagram

Perancang Sequence diagram dapat memberikan gambaran sederhana bagaimana operasi pada aplikasi terjadi. Objek yang akan berinteraksi diantaranya adalah Pengguna, Menu Utama, Transaksi, Anggaran, Laporan Keuangan, Tentang.

Disaat pengguna membuka aplikasi, pengguna akan diberikan ringkasan transaksi yang pengguna telah buat pada

Menu Utama. Jika pengguna memilih menu Transaksi di toolbar maka pengguna akan diminta untuk memasukan transaksi pemasukan atau pengeluaran dan pengguna dapat melihat daftar transaksinya. Selanjutnya jika pengguna menekan tombol Anggaran, maka akan ditunjukkan data anggaran dan pengguna dapat menambahkan anggaran baru. Pada menu Laporan Keuangan pengguna akan diberikan data laporan mengenai jumlah pemasukan dan pengeluaran yang telah dilakukan selama menggunakan aplikasi. Terakhir pada menu Tentang pengguna diperlihatkan informasi singkat mengenai detail aplikasi dari aplikasi. Berikut desain dari Sequence diagram.

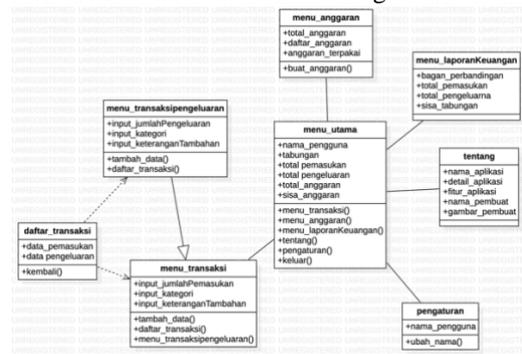


Gambar 2. Sequence Diagram Aplikasi

3. Class Diagram

Class Diagram merupakan diagram yang menggambarkan kelas-kelas dalam sebuah sistem dan hubungan diantaranya. Class sendiri merupakan istilah yang mendeskripsikan sekelompok objek yang semuanya memiliki peran serupa dalam sistem

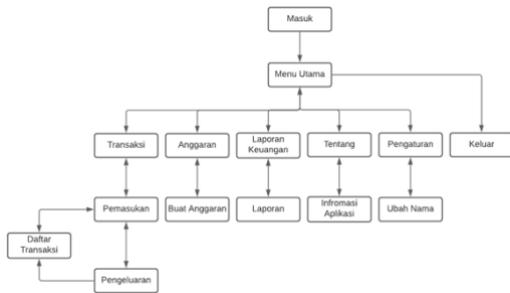
Sekelompok objek ini terdiri atas fitur struktural yang mendefinisikan apa yang diketahui class dan fitur operasional yang mendefinisikan apa yang bisa dilakukan oleh suatu class. Berikut class diagram.



Gambar 3. Sequence Diagram Aplikasi

Struktur navigasi yang akan dibuat adalah struktur navigasi campuran dengan tujuan untuk menggambarkan bagaimana alur dari aplikasi dan hubungan dengan tiap menunya. Menu utama merupakan halaman utama yang akan ditampilkan setelah pengguna masuk ke dalam

aplikasi yang kemudian didalamnya terdapat toolbar navigasi ke menu-menu yang lain dibawahnya.



Gambar Sequence Diagram Aplikasi

4. dsasd

1. Tabel Transaksi

Tabel pemasukan dibuat untuk menyimpan data pemasukan uang pada aplikasi. Struktur dari table pemasukan sebagai berikut.

Tabel 1. Tabel Transaksi

Nama Field	Tipe Data	Keterangan
idtransaksi	INTEGER(10)	Primary key untuk penanda.
uang	INTEGER(10)	Jumlah uang yang akan didata
kategori	TEXT(25)	Untuk kategori pendapatan
keterangan	TEXT(30)	Informasi tambahan
tanggal	TEXT(40)	Tanggal data diinput
jenis	TEXT(2)	Jenis transaksi

2. Tabel Pengguna

Tabel pengguna yang hanya

Nama Field	Tipe Data	Keterangan
iduser	INTEGER(2)	Primary key untuk penanda
nama	TEXT(15)	Nama pengguna

Tabel 2. Tabel Pengguna

Nama Field	Tipe Data	Keterangan
iduser	INTEGER(2)	Primary key untuk penanda
nama	TEXT(15)	Nama pengguna

3. Tabel Anggaran

Tabel anggaran dibuat untuk menampung jumlah anggaran yang akan digunakan nanti. Berikut struktur tabel anggaran.

IV. HASIL DAN PEMBAHASAN

Hasil

Splash screen tidak memerlukan pemrograman didalamnya, saat aplikasi dijalankan maka splash screen akan muncul dan berlanjut ke halaman masuk. Hanya cukup merubah imageview dengan logo aplikasi dan merubah label dengan tulisan MANKEPRI.



Gambar 1. Splash Screen

aplikasi akan meminta pengguna untuk menggunakan Face ID, Touch ID, atau Passcode supaya bisa masuk ke menu utama.



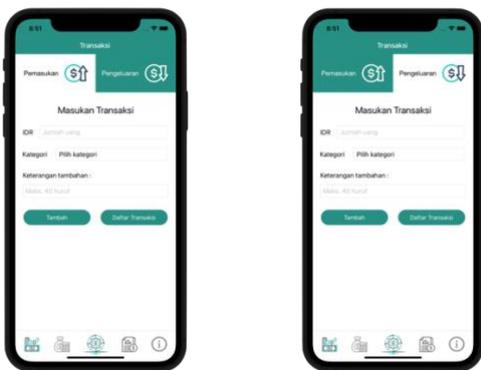
Gambar 2. Halaman Login

Menu utama mengambil data dari tabel transaksi dan tabel anggaran yang sudah tersimpan didalam database SQLite. Setelah semua data terpanggil dilakukan perhitungan sederhana untuk mencari nilai tabungan dan sisa anggaran menampilkan hasilnya pada menu utama.



Gambar 3. Menu Utama

Menu transaksi pemasukan dan pengeluaran yang memiliki tampilan sama persis, tetapi berbeda sedikit pada bagian kategori, dan berbeda sedikit di bagian program untuk menginput data kedalam tabel transaksi.



Gambar Halaman Transaksi

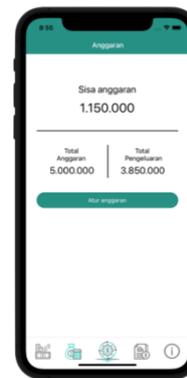
Halaman daftar transaksi memberikan daftar seluruh transaksi yang pernah dimasukan sebelumnya dengan

menekan button Daftar Transaksi pada halaman transaksi pemasukan maupun transaksi pengeluaran. Data transaksi dibedakan dengan warna dan gambar yang berbeda. Pada data transaksi pemasukan maka teks pada label jumlah menjadi hijau dan gambar uang keatas, jika data transaksi pengeluaran maka teks pada label jumlah menjadi merah dan gambar uang kebawah.



Gambar 5. Halaman Daftar Transaksi

Halaman anggaran sangat sederhana. Halaman ini hanya berisi beberapa label dan sebuah button. label menyimpan data yang dipanggil dari database dan mengolahnya dengan perhitungan sederhana. Untuk button atur anggaran sendiri akan memunculkan pop up ketika ditekan dan angka yang dimasukan akan tersimpan kedalam database.



Gambar 6. Halaman Anggaran

Halaman laporan keuangan berisi dengan beberapa label dan sebuah bagan lingkaran. Label sendiri menggunakan data yang sudah tersimpan didalam database dan hanya tinggal memanggilnya. Untuk bagan lingkaran dibuat dengan menggunakan data dari transaksi pemasukan dan transaksi pengeluaran, data lalu diolah dan dirubah dalam bentuk persentase. Setelah diolah, data persentase tersebut dimasukan

kedalam program bagan lingkaran dan bagan lingkaran akan otomatis terbentuk.



Gambar 7. Halaman Laporan Keuangan

Halaman pengaturan terdapat pilihan mengganti nama pengguna. Dengan menekan button akan memunculkan pop up untuk merubah nama sesuai keinginan. Setelah nama dibuat maka nama akan masuk kedalam tabel pengguna.



Gambar 8. Halaman Pengaturan

Pengujian aplikasi akan dilakukan pada tahap ini menggunakan metode Black Box dan hasil pengujian akan ditampilkan dalam bentuk tabel dan tampilan dari aplikasi saat dijalankan pada perangkat iPhone XR.

Tabel 3. Pengujian Halaman Masuk

Skenario Pengujian	Hasil Yang Diharapkan	Hasil
Membuka aplikasi.	Tampil <i>splash screen</i> dan berlanjut ke halaman masuk	Sukses
Menekan tombol MASUK pada halaman masuk dan memberikan	Masuk ke menu utama.	Sukses

akses biometrik.		
Menekan tombol <i>icon</i> transaksi dan masuk ke halaman transaksi.	Halaman transaksi terbuka.	Sukses
Tambah daftar transaksi pemasukan dan pengeluaran.	Tampil <i>pop up</i> kecil di bawah yang menandakan transaksi telah terdata.	Sukses
Melihat data transaksi pada halaman daftar transaksi dengan menekan tombol Daftar Transaksi.	Halaman daftar transaksi terbuka dan tampil seluruh daftar transaksi yang pernah dibuat.	Sukses
Menekan tombol <i>icon</i> anggaran dan masuk ke halaman anggaran.	Halaman anggaran terbuka.	Sukses
Menambahkan jumlah anggaran pada halaman anggaran dengan menekan tombol Atur anggaran.	Berbuahnya data dari total anggaran yang telah dibuat dan muncul <i>pop up</i> kecil di bawah bertuliskan anggaran berhasil ditambah.	Sukses
Menekan tombol <i>icon</i> laporan keuangan dan masuk ke halaman laporan keuangan.	Halaman laporan keuangan terbuka beserta data-data transaksi didalamnya.	Sukses
Menekan tombol <i>icon</i> tentang dan masuk ke halaman tentang.	Halaman tentang terbuka.	Sukses
Menekan tombol <i>icon</i> pengaturan dan masuk ke	Halaman pengaturan terbuka.	Sukses

V. KESIMPULAN

Berdasarkan hasil penulisan ilmiah mengenai pembuatan Aplikasi Manajemen Keuangan Pribadi Pada Platform iOS Menggunakan Bahasa Pemrograman Swift didapatkan kesimpulan. Bahwa Aplikasi Manajemen Keuangan Pribadi ini memiliki beberapa fitur diantaranya pencatatan pemasukan dan pengeluaran uang, jumlah anggaran yang ingin dibuat, serta laporan keuangan dengan menyimpan data dari input pengguna sebagai inti dari aplikasi.

Aplikasi ini cocok digunakan untuk semua kalangan, terutama bagi pengguna yang ingin melakukan pendataan terhadap tiap transaksi yang dilakukan untuk memantau daftar arus pemasukan dan pengeluaran sehari-hari.

Aplikasi Manajemen Keuangan Pribadi dibuat dengan menggunakan bahasa Swift dan SQLite untuk programnya, serta aplikasi Figma untuk [11].

pengembangan desain dan perancangan aplikasi berikut juga UML untuk perancangan sistem aplikasinya.

Berdasarkan urutan tahapan dalam pembuatan, penulis dinyatakan berhasil karena telah sesuai dengan tujuan yang diharapkan oleh penulis, yaitu pembuatan aplikasi yang dimulai dari proses input data, hingga proses penyampaian informasi daftar transaksi dan laporan keuangan yang dapat disampaikan kepada pengguna.

Pada aplikasi berbasis iOS ini terdapat beberapa fitur yang dapat dikembangkan lebih lanjut untuk memenuhi kebutuhan para pengguna. Yaitu fitur untuk menyimpan dan menyebarkan daftar transaksi, fitur pembuatan anggaran dan dan fitur laporan keuangan sesuai jangka waktu tertentu, serta fitur untuk menghapus data transaksi sesuai jangka waktu tertentu.

Oleh karena itu diharapkan dalam pengembangan kedepannya dapat membuat aplikasi ini menjadi lebih baik dan dapat memberikan manfaat yang lebih besar bagi para pengguna.

REFERENSI

[1] Nidar, S. R., & Bestari, S. 2012. Personal Financial Literacy Among University Students. *World Journal of Social Sciences*.

[2] Sukirno, Sadono. 2006. *Teori Pengantar Mikro Ekonomi*. Jakarta: PT. Raja Grafindo Persada.

[3] Cummins, M., Haskel, J. H., & Jenkins, S. 2009. Financial Attitudes and Spending Habits Of University Fresmen. *Journal Of Economics and Economic Education Research*.

[4] Apple Inc. 2014. *The Swift Programming Language (Swift 5.4)*. URL: <https://books.apple.com/id/book/the-swift-programming-language-swift-5-4/id881256329> [Tanggal Akses: 16 Juli 2021].

[5] Education, Apple. 2021. *Develop in Swift Fundamentals Xcode* 12. URL:<https://books.apple.com/id/book/develop-in-swiftfundamentals/id1556365994> [Tanggal Akses: 13 Juli 2021].

[6] Ardiansyah, D. (2016). *Sistem Informasi Pendaftaran Event dengan PHP untuk Panduan Skripsi*. Cirebon, Jawa Barat: CV. ASFA Solution.

[7] Rouf, Abdul. 2012. *Pengujian Perangkat Lunak Dengan Menggunakan Metode White Box dan Black Box*. STMIK HIMSYA Jurnal Teknologi Informasi.

[8] Algani, Ghifarin. 2020. *Game SAGASU GOI! Sebagai Media Alternatif Pembelajaran Kosakata Bahasa Jepang Tingkat Dasar Berbasis Android*. Bandung: Universitas Komputer Indonesia.

[9] Wati, Embun Fajar, dan Arvin Anggoro Kusumo. 2016. *Penerapan Metode Unified Modeling Language (UML) Berbasis Desktop Pada Sistem Pengolahan Kas Kecil Studi Kasus Pada PT Indo Mada Yasa Tangerang*. UNSIKA Syntax Jurnal Informatika.

[10] Rejeki, Muslim Setyo, dan Ali Tarmuji. 2013. *Membangun Aplikasi Autogenerate Script Ke Flowchart Untuk Mendukung Business Process Reengineering*. Jurnal Sarjana Teknik Informatika.

[11] Kreibich, J. A. 2010. *Using SQLite*. California: O'Reilly Media, Inc.