|  |
| --- |
| SOFTWARE DEVELOPMENT PROJECT  APLIKASI PENGELOLAAN KEUANGAN PRIBADI  “OURMONEY” |
| LOGO ISTTS TRANS PUTIH |
|  |
| Oleh:  Lukky Hariyanto – 219116856  Ray Vitto Nugroho – 219116858  Steven Liem – 219116860 |
| PROGRAM SARJANA  PROGRAM STUDI TEKNIK INFORMATIKA  FAKULTAS SAINS DAN TEKNOLOGI  INSTITUT SAINS DAN TEKNOLOGI TERPADU SURABAYA  SURABAYA  2021 |

TUGAS AKHIR/TESIS

APLIKASI PENGELOLAAN KEUANGAN

PRIBADI “OURMONEY”

Mengetahui/Menyetujui

Dosen Pembimbing

(Kevin Setiono, S.Kom)

FAKULTAS SAINS DAN TEKNOLOGI

PROGRAM STUDI TEKNIK INFORMATIKA

INSTITUT SAINS DAN TEKNOLOGI TERPADU SURABAYA

SURABAYA

JANUARI 2022

KATA PENGANTAR

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa, oleh karena berkat dan rahmat-Nya kami boleh menyelesaikan proyek aplikasi seluler kami. Kami juga turut berterima kasih kepada semua rekan-rekan beserta pihak yang turut memberikan dukungan demi tercapainya kelengkapan proyek kami.

Pada kesempatan ini, penulis ingin mengucapkan terima kasih sebesar-besarnya kepada bapak Kevin Setiono, selaku dosen mata kuliah Software Development Project, yang telah memberikan inspirasi dan saran bagi pembentukan program ini.

Penulis menaruh harapan agar dengan dibentuknya proyek "APLIKASI PENGELOLAAN KEUANGAN PRIBADI OURMONEY", dapat menambah pengetahuan dan memperluas wawasan pembaca sekalian. Penulis juga berharap bahwa proyek ini dapat memberikan inspirasi bagi pembaca dalam penelitian dan pengabdian pada ilmu pengetahuan.

Penulis menyadari bahwa dalam penulisan karya tulis ini terdapat kekurangan dan kesalahan. Karena itu, penulis memohon pemahaman dari pembaca sekalian atas kesalahan tersebut. Penulis juga terbuka untuk menerima koreksi dan saran dari pembaca yang dapat menyempurnakan karya tulis ilmiah ini.

Surabaya, Januari 2022

Penulis

DAFTAR ISI

Halaman

HALAMAN JUDUL i

HALAMAN PENGESAHAN ii

KATA PENGANTAR iii

DAFTAR ISI iv

DAFTAR GAMBAR vi

DAFTAR ALGORITMA viii

DAFTAR SEGMEN PROGRAM ix

BAB I PENDAHULUAN 1

1.1 Latar Belakang 1

1.2 Tujuan 1

1.3 Ruang Lingkup 2

1.4 Teori Pendukung 4

1.5 Sistematika Pembahasan 6

BAB II ANALISA 8

2.1 Analisa Sistem 8

2.1.1 Canvas 3

2.1.2 3D 5

2.2 Kebutuhan Aplikasi 19

2.3 Sistem yang Akan Dibuat 21

BAB III SISTEM DESAIN 24

3.1 Desain Interface 24

3.2 Desain Database 38

3.3 Desain Procedural 40

BAB IV IMPLEMENTASI 46

4.1 Program Pembuatan Report 46

4.2 Program Penyedia Data Transaksi Home 52

4.3 Program Ekspor dan Impor Data 54

BAB V UJI COBA 57

5.1 Uji Coba Mengubah Isi File Save Data 57

5.2 Uji Coba Mengubah Transaksi Ketika Target Aktif 59

5.3 Uji Coba Melihat Report 59

BAB VI PENUTUP 61

6.1 Kesimpulan 61

6.2 Saran 61

DAFTAR PUSTAKA 63

DAFTAR GAMBAR

Gambar Halaman

1.1 Arsitektur Sistem 2

1.2 Wireframe 4

2.1 Tampilan wallet 8

2.2 Tampilan aliran dana 9

2.3 Tampilan tambah transaksi 10

2.4 Tampilan kategori (foto kiri) dan tambah kategori (foto kanan) 10

2.5 Report pengeluaran 11

2.6 Report aliran dana 12

2.7 Report perbandingan pemasukan dan pengeluaran dalam bentuk summary dan detail 12

2.8 Layar Splash dan Layar Home OurMoney 13

2.9 Layar - Layar Transaksi OurMoney 14

2.10 Layar Report Summary OurMoney 15

2.11 Layar Report Detail Wallet 15

2.12 Layar Penambahan Wallet dan Transfer OurMoney 16

2.13 Tampilan Target Menabung OurMoney 17

2.14 Layar Tambah Kategori dan Edit Kategori 18

2.15 Layar Profil OurMoney 19

2.16 Diagram Use Case 22

3.1 Tampilan navigation bar OurMoney 24

3.2 Tampilan Input Nama OurMoney 25

3.3 Tampilan HomePage OurMoney 26

3.4 Tampilan Tambah Transaksi OurMoney 27

3.5 Tampilan Detail Transaksi OurMoney 28

3.6 Tampilan Menabung OurMoney 29

3.7 Tampilan Target Baru OurMoney 30

3.8 Tampilan Laporan OurMoney 31

3.9 Tampilan detail laporan OurMoney 32

3.10 Tampilan profil OurMoney 33

3.11 Tampilan Halaman Wallet OurMone 33

3.12 Tampilan Tambah Wallet OurMoney 34

3.13 Tampilan Edit Wallet OurMoney 35

3.14 Tampilan Halaman Kategori OurMoney 36

3.15 Tampilan Halaman Tambah/Edit Kategori OurMoney 36

3.16 Tampilan Halaman Ekspor/Impor Data OurMoney 37

3.17 Conceptual Data Model 38

3.18 Physical Data Model 38

DAFTAR ALGORITMA

Algoritma Halaman

3.1 Algoritma Menampilkan Report 40

3.2 Algoritma Add/Update Wallet/Kategori 41

3.3 Algoritma Ekspor Data 42

3.4 Algoritma Impor Data 43

3.5 Algoritma Tambah/Edit Transaksi 44

DAFTAR SEGMEN PROGRAM

Segmen Program Halaman

4.1 Class GetTransactionAndCategories 46

4.2 Method setdata 47

4.3 Program utama membuat report 50

4.4 Class GetTransactionAsync 52

4.5 Procedure SetupRV untuk menampilkan data pada home 53

4.6 Procedure doWriteSerialize untuk membuat save file 54

4.7 Procedure doReadSerialize untuk membaca save file 56

BAB I

PENDAHULUAN

Pada bab ini dipaparkan pendahuluan proyek. Bab ini akan menjelaskan latar belakang dari proyek kami beserta masalah-masalah yang akan dianalisa. Latar belakang dan analisis tersebut akan menjadi fondasi proyek ini. Dalam bab ini juga dijelaskan tujuan dan batasan proyek.

1. Latar Belakang

Menabung adalah salah satu dari sekian banyak metode pengelolaan keuangan dengan tujuan menyisihkan sejumlah dana untuk digunakan di masa depan. Dana yang terkumpul itu kelak dapat digunakan untuk membeli barang keinginan, memenuhi kebutuhan pokok, atau bahkan sekedar menyisihkan dana cadangan untuk berjaga-jaga. Kegiatan menabung sudah menjadi suatu kebiasaan finansial yang dikenal oleh masyarakat Indonesia. Banyak kalangan muda yang berminat untuk menumbuhkan kebiasaan menabung sebagai sebuah inisiatif untuk mempersiapkan diri untuk masa depan.

Implementasi pengelolaan uang dalam bentuk aplikasi digital dapat memudahkan banyak orang untuk menggunakannya. Dengan demikian, kalangan muda yang memahami teknologi akan tertarik untuk menumbuhkan minat mengelola keuangan dan kebiasaan menabung. Kami menggagaskan sebuah aplikasi pengelolaan keuangan yang dapat berjalan pada perangkat yang paling banyak dipakai, yaitu ponsel Android, untuk mencapai tujuan tersebut. Tidak hanya itu, kami juga berencana membuat aplikasi kami mendukung Android versi lama sehingga meningkatkan jangkauan aplikasi kami terhadap calon pengguna.

1. Tujuan

Tujuan dari perancangan dan pembuatan aplikasi “OurMoney” dijabarkan sebagai berikut

* Merancang sebuah aplikasi pengelolaan uang dalam bentuk mobile application yang dapat dijalankan pada platform yang banyak digunakan.
* Memudahkan banyak orang menumbuhkan kebiasaan menabung melalui aplikasi mobile.

1. Ruang Lingkup

Kami memberi nama aplikasi kami “OurMoney”, dengan filosofi bahwa uang kita adalah prioritas dan asset yang berharga. Program ini ditujukan agar pengguna dapat mencatat pemasukan/pengeluaran dengan mudah dan di mana saja. Oleh karena itu kami memilih untuk mengembangkan aplikasi ini sebagai mobile app demi mendukung penggunaan yang lebih luas. Pada aplikasi ini user dapat menambah catatan transaksi dan memberi kategori berdasarkan jenis transaksinya.



Gambar 1.1  
Arsitektur sistem

Aplikasi ini akan berjalan pada platform Android, sistem operasi yang paling banyak digunakan dalam perangkat mobile. Oleh karena itu aplikasi kami akan menggunakan bahasa pemrograman Java. Kami akan mengembangkan OurMoney dengan IDE Android Studio yang sangat memudahkan pengembangan aplikasi mobile. Penyimpanan data dari aplikasi OurMoney nantinya akan menggunakan SQLite yang sudah terintegrasi pada aplikasi Android pada umumnya.

Pada Gambar 1.1 ditampilkan visualisasi arsitektur sistem OurMoney. Data yang disimpan bersifat *offline* (luar jaringan) dan disimpan dalam SQLite. Pengguna dapat melakukan ekspor data SQLite ke format CSV dan melakukan impor data dari *file* CSV. Perlu diingat bila transfer File hanya bisa dilakukan oleh sesama pengguna aplikasi OurMoney.

Fitur-fitur yang kami ajukan untuk diimplementasikan pada sistem kami adalah:

* Wallet ganda dalam satu perangkat. Satu user pada satu perangkat dapat memiliki wallet penyimpanan lebih dari satu.
* Catat, ubah, dan hapus pengeluaran & pemasukan dalam setiap wallet. Fungsi utama dari aplikasi kami.
* Mengkategorikan transaksi berdasarkan jenisnya dan menambahkan jenis. Terdapat beberapa kategori yang disediakan dari aplikasi, tetapi user dapat menambahkan kategori baru sesuai kebutuhan atau keinginan mereka.
* Pembuatan target menabung. Satu user dapat memiliki satu target.
* Mengatur peringatan pengeluaran. Peringatan bersifat global, sama untuk setiap wallet.
* Data bisa dipindah ke perangkat baru dalam bentuk serializable. Untuk berpindah ke perangkat baru, perlu dilakukan pemindahan data dengan file.
* Pembuatan report pengeluaran & pemasukan per bulan dalam bentuk grafik atau chart.
* Pembuatan report berapa kali melewati batas pengeluaran.



Gambar 1.2  
Wireframe

Pada Gambar 1.2 ditampilkan wireframe antarmuka dari aplikasi OurMoney. Wireframe sebelah kiri menunjukkan tampilan beranda aplikasi yang pertama kali dilihat ketika membuka aplikasi.Sementara wireframe kanan menunjukkan tampilan report berdasarkan data keuangan yang ada pada aplikasi. Ada juga halaman untuk melihat fitur lain seperti manambah transaksi dsb.

1. Teori Pendukung
   * + **Java**

Java adalah bahasa pemrograman yang banyak digunakan. Java dapat dijalankan di berbagai jenis komputer. Baik komputer meja, komputer tablet, ponsel pintar, komputer server, dan lainnya. Bahasa ini diciptakan James Gosling dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis dari bahasa C, namun sintaksis model objek Java cenderung lebih sederhana.

Kelebihan dari bahasa pemrograman Java ialah:

* Banyak *platform* mendukung.
* Berorientasi obyek.
* Memiliki banyak *class library*.
* Serupa dengan C++.
* Pengumpulan sampah (*garbage collection*) yang dilakukan otomatis.
  + - **Android**

Android adalah sistem operasi yang dirancang untuk perangkat genggam layer sentuh dan berbasis Linux. Pada mulanya, sistem operasi Android dikembangkan Android Inc. dengan dukungan dari Google. Pada tahun 2005, Google membeli Android sepenuhnya. Kemudian Android dirilis secara resmi pada tahun 2007 di saat Open Handset Alliance didirikan bersamaan. Ponsel Android pertama dijual pada Oktober 2008.

* + - **SQL**

SQL, atau kepanjangannya adalah *“Structured Query Language”*, adalah bahasa yang biasa digunakan untuk melakukan akses data dalam basis data relasional. Bahasa ini adalah bahasa standar yang digunakan untuk mengelola sebuah basis data relasional. SQL berakar pada ide Jhonny Oracle untuk membuat basis data relasional pada Juni 1970. Kala itu, bahasa SQL dinamai SEQUEL (Structured English Query Language). Di akhir tahun 1970-an, perusahaan Oracle muncul dan membuat server basis data yang bernama sama dengan nama perusahaannya.

* + - **Android Studio**

Android Studio adalah sebuah *Integrated Development Environment* (IDE), atau dalam bahasa Indonesianya adalah Lingkungan Pengembangan Terintegrasi, untuk sistem operasi *mobile* Android. Android Studio dibangun di atas perangkat lunak IntelliJ IDEA, sebuah IDE Java, dengan desain yang dikhususkan untuk pengembangan aplikasi Android. IDE ini merupakan pengganti dari Eclipse Android Development Tools (ADT) yang sebelumnya merupakan IDE utama untuk pengembangan aplikasi android. Dengan itu banyak fitur dari Android Studio yang akan berguna dalam pengembangan aplikasi Android.

* + - **Aplikasi Seluler**

Aplikasi seluler atau aplikasi *mobile* adalah program computer yang dirancang untuk berjalan di atas *mobile device* atau perangkat seluler yang dapat dibawa-bawa seperti ponsel pintar, tablet, atau jam tangan. Aplikasi seluler sering kali dianggap sebagai kebalikan dari aplikasi desktop yang berjalan di komputer desktop, dan dengan aplikasi web yang berjalan di browser web perangkat. Aplikasi seluler berguna sebagai fitur utama dari setiap Smartphone yang ada. Sebab tanpa aplikasi seluler yang dimiliki maka Smartphone kita tidak akan popular di kalangan masyarakat.

* + - **Desain Material**

Desain Material (dalam bahasa inggris disebut “Material Design”) adalah bahasa desain yang dikembangkan oleh Google dan diumumkan pada konferensi Google I/O pada 25 Juni 2014. Memperluas pada motif "kartu" yang pertama kali terlihat pada aplikasi Google Now, adalah desain dengan peningkatan penggunaan layout berbasis grid, animasi dan transisi responsif, padding, dan efek kedalaman seperti pencahayaan dan bayangan. Desainer Matías Duarte menjelaskan bahwa, "tidak seperti kertas asli, materi digital kami dapat mengembang dan berubah secara cerdas. Materi memiliki permukaan dan tepi fisik. Jahitan dan bayangan memberikan makna tentang apa yang dapat Anda sentuh." Google menyatakan bahwa bahasa desain baru mereka didasarkan pada kertas dan tinta tetapi implementasinya dilakukan dengan cara yang canggih.

1. Sistematika Pembahasan

Proyek akan dibahas secara lengkap dan sistematis dalam 6 bab yang saling terkait. Sebelum memasuki bab pertama, terdapat halaman sampul, halaman pengesahan, kata pengantar, daftar isi, daftar gambar, daftar tabel, daftar algoritma, dan daftar segmen program. Selanjutnya maka memasuki pembahasan utama di antaranya:

* BAB I : PENDAHULUAN

Bab pendahuluan berisi penyampaian dasar dari pengerjaan proyek. Unsur-unsur yang mendasari pembuatan proyek adalah latar belakang, tujuan, ruang lingkup, beserta teori pendukung.

* BAB II : ANALISA

Bab kedua, yaitu bab Analisa, memuat seluruh analisis permasalahan. Sebuah subyek tolak ukur yang merupakan aplikasi seluler pengelolaan keuangan akan dianalisis juga. Serangkaian analisis dilaksanakan dalam bentuk dokumentasi, prototyping, perancangan fitur, dan perencanaan sistem yang akan dibuat.

* BAB III : SISTEM DESAIN

Pada bab ketiga, desain dari target proyek dijelaskan. Hasil proyek akan memiliki desain interface, desain database, dan desain procedural.

* BAB IV : IMPLEMENTASI

Bab keempat akan membahas mekanisme aplikasi hasil proyek dalam penggunaannya. Beberapa segmen program yang penting akan ditunjukkan dan dijelaskan.

* BAB V : UJI COBA

Serangkaian pengujian terhadap aplikasi hasil proyek akan dilakukan pada bab lima. Pada bab ini dirumuskan beberapa pengujian yang dilakukan pada aplikasi agar dapat mengukur kelengkapan dan kematangan aplikasi.

* BAB VI : PENUTUP

Bab keenam merupakan bab penutup. Bab penutup memuat kesimpulan dari proyek dan saran.

BAB II

ANALISA

Dalam bab ini, dijelaskan analisa permasalahan mengenai program yang akan dibuat. Dalam analisa permasalahan, kami memilih MoneyLover sebagai acuan dalam pembuatan fitur-fitur pada sistem yang akan kami buat. Hal yang akan dibahas meliputi cara kerja dari aplikasi acuan yang terpilih dan report apa saja yang dapat dibuat oleh aplikasi manajemen keuangan tersebut. Semua itu akan dipaparkan juga penjelasan aplikasi apa yang akan dibuat.

1. Analisa Sistem

Kami ingin membuat sebuah aplikasi pengelolaan keuangan. Sebagai acuan, kami mengambil referensi dari aplikasi MoneyLover. MoneyLover adalah aplikasi manajemen keuangan yang dikembangkan Finsify Technology. MoneyLover sudah memiliki banyak pengguna, bahkan menerima penghargaan dari badan jurnalisme ternama. Fitur-fitur MoneyLover mencakup sistem wallet, pencatatan pemasukan dan pengeluaran, pemilihan kategori pemasukan dan pengeluaran, pembuatan kategori buatan sendiri.



Gambar 2.1  
Tampilan wallet

Gambar 2.1 adalah tampilan wallet dari aplikasi MoneyLover. Dalam tempilannya terdapat daftar wallet apa saja yang dimiliki pengguna, tombol hijau untuk menambah wallet baru, dan pada tiap wallet terdapat tombol menu untuk melakukan pengaturan pada wallet masing-masing. Wallet dapat ditambah lebih dari satu. Wallet juga dapat dihapus, namun tidak bisa menghapus apabila hanya terdapat satu wallet.



Gambar 2.2  
Tampilan aliran dana

Gambar 2.2 menampilkan layar daftar transaksi pada MoneyLover. Layar ini akan menampilkan sejarah transaksi yang dilakukan pada sebuah wallet. Daftar transaksi dapat disaring menurut jenisnya (pendapatan atau pengeluaran), berdasarkan kategori, atau berdasarkan tanggal. Pendapatan dan pengeluaran akan dibedakan menurut warna dari jumlah uangnya sehingga mempermudah pengguna untuk mengukur seberapa banyak pemasukan dan pengeluaran pengguna.



Gambar 2.3  
Tampilan tambah transaksi

Pada gambar 2.3 dijelaskan tampilan form untuk menambahkan transaksi pada sebuah wallet. Transaksi dapat berupa pemasukan atau pengeluaran. Transaksi juga dapat diberi kategori sesuai kebutuhan. Tanggal transaksi dapat disesuaikan dalam kasus apabila pengguna lupa menambahkan pemasukan atau pengeluaran pada hari sebelum hari ini. Pengguna juga dapat memberikan teks catatan pada setiap transaksi yang ditambahkan.

Gambar 2.4  
Tampilan kategori (foto kiri) dan tambah kategori (foto kanan)

Pada gambar 2.4, ditampilkan layar untuk mengisi kategori transaksi berdasarkan pilihan yang tersedia. Pilihan kategori dapat ditambah sendiri sesuai kebutuhan dengan isian pada foto kanan. Kategori baru dapat diberikan nama, jenis (pendapatan atau pengeluaran) dan wallet mana yang memiliki kategori butan tersebut. Setelah semua field sudah diisi, maka transaksi dapat ditambahkan.

Berikut ini adalah dokumentasi dari aplikasi “MoneyLover”. Dokumentasi yang kami ambil adalah report-report yang dapat dihasilkan MoneyLover berdasarkan data keuangan yang tercatat.Report dalam kasus ini merupakan sebuah laporan keuangan yang dimiliki. Sejumlah report yang dihasilkan di antaranya adalah report pengeluaran (*expense*), report aliran dana (*cash flow*), dan report perbandingan pemasukan dan pengeluaran.



Gambar 2.5  
Report pengeluaran

Gambar 2.5 menampilkan bentuk report pertama yang penting, yaitu laporan pengeluaran. Dalam report ini ditampikan kategori-kategori pengeluaran yang pernah dilakukan beserta nominal pengeluaran tiap kategori. Report ini juga menampilkan sebuah pie chart (diagram lingkaran) yang menampilkan persentase pengeluaran tiap kategori. Report ini juga menampilkan pengeluaran keseluruhan pengguna.



Gambar 2.6  
Report aliran dana

Gambar 2.6 menampilkan report kedua yang menyajikan data seluruh pengeluaran beserta pemasukan. Rentang waktu yang diberikan adalah 1 bulan. Pemasukan diberi warna biru agar berbeda dengan pengeluaran yang berwarna merah. Total pengeluaran juga ditampilkan pada bagian khusus.

Gambar 2.7  
Report perbandingan pemasukan dan pengeluaran dalam  
bentuk summary (foto kiri) dan detail (foto kanan)

Gambar 2.7 menunjukkan report perbandingan pemasukan dan pengeluaran. Gambar kiri menunjukkan bentuk summary dalam format “*pie chart*”, sementara gambar kanan menunjukkan detail dalam format daftar. Kedua tersebut dapat dipakai secara bergantian.Tersedia juga sebuah grafik balok di atas untuk memudahkan pengguna membandingkan pemasukan dan pengeluaran.

Berdasarkan analisis yang telah dilakukan, maka dapat dilakukan prototyping. Kami menyusun sebuah prototype/purwarupa aplikasi kami dalam bentuk purwarupa mockup yang statis. Mockup ini akan menjelaskan sejumlah tampilan yang akan dimiliki OurMoney. Mockup ini thanyalah acuan dan belum tentu sama dengan hasil akhir kita



Gambar 2.8  
Layar Splash dan Layar Home OurMoney

Gambar 2.8 menyajikan gambar mockup aplikasi OurMoney pada tampilan sprite dan home. Layar sprite adalah layar pertama yang muncul ketika pengguna membuka OurMoney. Layar sprite menunjukkan logo aplikasi untuk beberapa saat. Kemudian user dibawa ke layar home di mana terdapat total saldo user dan daftar transaksi hari ini.



Gambar 2.9  
Layar - Layar Transaksi OurMoney

Berikutnya pada gambar 2.9 dijelaskan tampilan ketiga layar transaksi dari OurMoney. Pada prototype yang kita design, tampilan awal ketika hendak menambah transaksi adalah daftar transaksi. Bila transaksi tersebut diklik maka pengguna dapat mengubah nominal dan kategori transaksi sesuai kebutuhan sesuai dengan layar yang di kiri. Kedua itu akan mempengaruhi dompet yang dipilih dengan menambah atau mengurangi isinya.

Selain itu, pengguna juga dapat menambah transaksi baru dengan menambah nominal dan kategorinya. Layar tersebut akan memiliki tampilan seperti layar di tengah gambar 2.9. Lalu bila nominal transaksi berupa pengeluaran dan melebihi saldo wallet, maka diberi peringatan dan tidak bisa menambah pengeluaran sesuai dengan layer kanan gambar 2.9. Peringatan tersebut akan berwarna merah dan di-bold sehingga susah dilewatkan oleh para pengguna OurMoney.

Terakhir, pengguna dapat melakukan edit transaksi sewaktu-waktu jika dibutuhkan. Pengguna dapat mengakses tampilan edit transaksi melalui tampilan daftar transaksi pengguna dengan menekan lama transaksi yang ingin diubah. Tampilan edit transaksi bisa dilihat di bagian kiri gambar 2.9. Dan saat tombol Edit Transaksi ditekan maka akan diubah transaksi yang dipilih.



Gambar 2.10  
Layar Report Summary OurMoney

Gambar 2.10 menjelaskan interface laporan OurMoney yang menampilkan statistik transaksi berdasarkan rentang waktu yang ditentukan. Rentang waktu transaksi yang ditampilkan dapat dipilih oleh user melalui sebuah combo box pada bagian atas layar laporan. Daftar pemasukan dan pengeluaran akan ditampilkan secara terpisah untuk memudahkan user menganalisis transaksi yang telah dimasukkan. Total saldo dari seluruh transaksi juga ditampilkan sebagai ringkasan.



Gambar 2.11  
Layar Report Detail Wallet

Selain report summary, OurMoney akan menyediakan report detail. Gambar 2.11 menunjukkan tampilan report detail di mana report dibuat berdasarkan data dari satu wallet saja.Bila ingin mengubah walletnya maka bisa dipilih di combobox wallet. Selain itu juga dapat dilihat berdasarkan rentang waktu yang tersedia. Nanti total dari seluruh transaksi akan ditampilkan dibagian bawah halaman.



Gambar 2.12  
Layar Penambahan Wallet dan Transfer OurMoney

Gambar 2.12 menjelaskan interface wallet OurMoney. Pengguna OurMoney bisa memiliki lebih dari satu wallet. Tampilan awal wallet akan menampilkan seluruh wallet yang dimiliki beserta total uang dari setiap wallet. Ketika akan menambahkan wallet baru, aplikasi akan meminta nama yang bersifat wajib dan jumlah isi wallet yang opsional. Jika jumlah isi wallet tidak diisi angka, maka secara default akan memberi nominal 0 pada wallet yang ditambahkan. Terdapat pula fitur transaksi antar wallet di mana wallet yang satu dapat memberikan dana pada wallet lainnya. Sebelum transfer dana, pengguna dapat memilih wallet asal dana dan wallet tujuan beserta berapa banyak yang ingin ditransferkan.



Gambar 2.13  
Tampilan Target Menabung OurMoney

Pada gambar 2.13 dijelaskan bagian target menabung dari OurMoney. Pengguna dapat memasukkan jumlah target tabungan sesuai keinginan, dan tenggat waktu tabungan tersebut. Setelah ditambahkan, maka OurMoney akan menghitung seluruh pemasukan dan pengeluaran setelah penambahan target. Jika totalnya sudah mencapai target, maka sisa target bernilai nol. Bila pengguna berusaha bertransaksi tapi target sudah dibawah jumlah transaksi, maka akan diberi warning saat berusaha bertransaksi. Pengguna juga dapat mengubah jumlah target menabung dan tenggat waktu menabung. Ketika tenggat waktu tabungan sudah tiba, maka target menabung akan direset.



Gambar 2.14  
Layar Tambah Kategori dan Edit Kategori

Gambar 2.14 menampilkan layar pengelolaan kategori OurMoney. Transaksi OurMoney dapat diberi kategori sebagai informasi tambahan. Kategori buatan juga dapat ditambahkan oleh pengguna sesuai kebutuhan. Pengguna dapat memasukkan nama kategori beserta jenis transaksinya, yaitu pemasukan atau pengeluaran. Kategori juga dapat diubah datanya jika diperlukan.

Bila kategori mau diubah, maka saat terbuka layer baru, akan diisikan di isian nama kategori yang akan diubah dan apakah kategori tersebut pengeluaran atau pemasukan, untuk kemudahan pengguna OurMoney.



Gambar 2.15  
Layar Profil OurMoney

Bagian terakhir pada prototype adalah bagian profil pengguna. Profil pengguna memuat serangkaian tombol untuk menjalankan fitur-fitur yang tersedia pada OurMoney. Di antaranya adalah fitur pengaturan, pengelolaan wallet, pengelolaan kategori, ekspor data, dan informasi tentang OurMoney. Gambar 2.15 menampilkan keseluruhan tampilan profil OurMoney.

1. Kebutuhan Aplikasi

MoneyLover adalah sebuah aplikasi pengelolaan keuangan yang didesain oleh profesional. Fungsi-fungsi yang terdapat pada MoneyLover adalah fungsi-fungsi yang telah dipikirkan dan didesain sedemikian rupa untuk membantu pengguna mengelola keuangan. Untuk mencapai hasil proyek yang baik, kami melakukan pengkajian pada subyek MoneyLover, dan pengkajian tersebut mencakup eksperimen terhadap fitur-fitur yang disediakan oleh MoneyLover. Kami merangkum fitur-fitur tersebut dan menyusun rencana untuk melakukan implementasi pada proyek yang dibuat.

* + - **Sistem Wallet**

Pengguna umumnya membagi-bagi keuangannya menjadi beberapa tabungan.

* + - **Report Overview**

Pengguna dapat membuat report/laporan keuangan berdasarkan keseluruhan data-data dalam bentuk pie chart dan bar chart.

* + - **Report Wallet**

Pengguna dapat melihat report/laporan transaksi untuk setiap wallet secara rinci.

* + - **Transaksi**

Pengguna dapat menambahkan transaksi dengan terlebih dahulu mendefinisikan jenis, kategori, dan nominal transaksi. Transaksi dibedakan menjadi dua jenis, yaitu pemasukan dan pengeluaran. Kategori transaksi dapat berisi bebas. Nominal transaksi juga dapat diisi sesuai kebutuhan pengguna.

* + - **Save Data dan Export**

Semua data yang telah dimasukkan pengguna dapat disimpan dalam ponsel Android secara luring (luar jaringan) atau offline. Data yang tersimpan akan dimuat kembali setiap kali pengguna membuka aplikasi. Data yang tersimpan juga dapat diexport ke ponsel baru jika pengguna mengganti ponselnya. Data harus dapat disimpan dengan format yang aman dan mudah dipindahkan ke device baru.

* + - **Budget Management/Target Menabung**

Kembali kepada tujuan utama aplikasi manajemen keuangan, fitur yang tidak kalah penting adalah pembuatan target menabung. Pengguna dapat menambahkan target menabung dengan mengisi nominal target serta tenggat waktu target. Target menabung direset ketika sudah melewati tenggat waktu. Bila target menabung akan dilebihi maka akan diperingati.

1. Sistem yang akan Dibuat

Dari kebutuhan aplikasi yang telah ditentukan, Ourmoney akan memiliki sistem wallet, report, transaksi dan tranfer data. Seluruh fitur tersebut dapat diakses melalui navigasi yang ada di bagian bawah layar OurMoney. OurMoney akan memiliki sistem wallet yang bisa lebih dari 1 sebagai tempat penyimpanan uang user. User dapat menambah wallet dan melakukan perubahan pada wallet miliknya bila dibutuhkan.

Kita juga akan membuat report secara overview yang meliputi semua wallet yang dimiliki, dan juga user dapat melihat report transaksi masing- masing wallet yang tersedia. Lalu setelah terdapat wallet – wallet tersebut, user dapat menambah transaksi entah dalam bentuk pemasukan maupun pengeluaran. Kedua bentuk pemasukan tersebut masing - masing menambah jumlah uang di wallet dan mengurangi jumlah uang di wallet. Selain itu dalam setiap transaksi memiliki kategori untuk memperjelas untuk apa uang yang dimasukkan tersebut dipakai bila pengeluaran atau dalam kasus apa anda mendapat uang bila pemasukan.

Setelah dimasukkan jumlah uang, dipilih pengeluaran/pemasukan ,wallet yang akan dipakai untuk transaksi dan kategorinya, maka transaksi akan ditambah, wallet yang dipilih akan berubah jumlah uangnya dan transaksi yang ada dapat dilihat di dalam report. Selanjutnya bila user mendadak memutuskan untuk menggunakan ponsel Android baru, pengguna dapat mentransfer data antar dua android agar data-data transaksi dan report tidak hilang dan tidak perlu memasukkan seluruh data dari awal.

Terakhir adalah fitur budget dimana user akan memasukkan jumlah uang dan tanggal berakhirnya budget. Jadi selama belum sampai masa berakhirnya budget, aplikasi akan mencatat setiap transaksi yang terjadi dan mempengaruhi budgetnya sesuai transaksi yang terjadi. Bila terdapat banyak pengeluaran dan transaksi selanjutnya akan melebih budget, akan dikeluarkan peringatan untuk mengingatkan user bila transaksi tersebut akan melebihi budget. Setelah masa batas selesai maka peringatan tidak akan muncul lagi.



Gambar 2.16  
Diagram Use Case

Gambar 2.16 adalah use case diagram dari aplikasi yang akan dibuat. Ruang lingkup aplikasi meliputi semua fitur-fitur aplikasi seperti yang sudah dijelaskan yang dapat digunakan oleh pengguna. Sistem OurMoney melibatkan 1 aktor, yaitu pengguna aplikasi. Terdapat sejumlah use case yang merupakan use case extend dari use case lainnya.

|  |  |
| --- | --- |
| Tabel 2.2  Tabel Contoh Aktifitas Mahasiswa | |
| No | Nama |
| 1 | Badan Eksekutif Mahasiswa |
| 2 | Himpunan Mahasiswa |
| 3 | Unit Kegiatan Mahasiswa |
| Tabel 2.2  (Lanjutan) | |
| No | Nama |
| 4 | Unit Kegiatan Kerohanian |
| 5 | Berbagai Macam Kepanitiaan |

BAB III

SISTEM DESAIN

OurMoney merupakan aplikasi seluler yang berjalan pada platform Android (sistem operasi Android) yang bertujuan untuk membantu pengelolaan keuangan. Kami merancang OurMoney menggunakan perangkat lunak Android Studio yang dijalankan pada sistem operasi Windows 10. Android Studio menggunakan Java sebagai bahasa pemrograman. Desain OurMoney akan dijabarkan dalam bidang desain interface, desain database, dan desain procedural.

1. Desain Interface (Interface mock up final)

Desain final antarmuka OurMoney tidak berbeda jauh dari prototype yang telah dibuat. Perbedaan yang sangat jelas akan terlihat pada bahasa desain yang digunakan. Prototype kami menampilkan interface OurMoney dengan grafik tampilan yang sangat sederhana dan tidak memiliki tema. Pada desain final OurMoney, kami menerapkan flat design dari material design yang terlihat sederhana dan mudah digunakan namun profesional. Tata letak dari komponen OurMoney dibuat semirip mungkin dengan prototype.



Gambar 3.1  
Tampilan navigation bar OurMoney

OurMoney memiliki sebuah navigation bar atau biasa disebut bilah navigasi sebagai alat bagi user untuk berpindah layar dari satu fitur ke fitur lain. Navigation bar dari OurMoney kami buat dengan material design sehingga desain yang ditampilkan terlihat sederhana namun memiliki kejelasan bahasa dan fungsi. Pada navigation bar terdapat 4 buah tombol navigasi yang terdiri dari: Home, Menabung, Laporan, dan Profil. Tampilan akhir dari navigation bar yang digunakan OurMoney terdapat pada gambar 3.1 di atas.



Gambar 3.2  
Tampilan Input Nama OurMoney

Sebelum anda bisa menggunakan OurMoney,anda harus mengisi nama anda dulu. Ini bisa dipakai sebagai pengidentifikasi pengguna aplikasi agar tidak dipusingkan dengan aplikasi milik orang lain. Setelah nama diisi, maka anda akan diarahkan menuju home page.

\

Gambar 3.3  
Tampilan HomePage OurMoney

Tampilan home OurMoney akan menampilkan data-data keuangan secara singkat dan mudah ditangkap pengguna. Data-data yang ditampilkan antara lain nama pengguna, total uang secara keseluruhan, dan daftar transaksi terbaru (akhir-akhir ini). Tiap transaksi pada daftar transaksi tersebut dapat diklik agar muncul sebuah tampilan detail transaksi.



Gambar 3.4  
Tampilan Tambah Transaksi OurMoney

Setelah anda memencet tombol tambah transaksi, anda akan memasuki halaman Tambah Transaksi (Gambar 3.4). Di halaman ini anda bisa memilih dua tombol pengeluaran dan pemasukan untuk memilih jenis transaksi anda. Setelah memilih jenis transaksi anda anda bisa mengisi jumlah pengeluaran/pemasukan anda, kategori transaksi anda, wallet yang dipakai untuk transaksi anda, keterangan transaksi anda dan yang terakhir adalah tanggal transaksi dimana defaultnya adalah hari tersebut. Setelah semuanya diisi maka bila tombol simpan tranksi dipencet maka transaksi akan bisa dilihat di homepage.



Gambar 3.5  
Tampilan Detail Transaksi OurMoney

Bila salah satu transaksi di home dipencet, maka aplikasi akan diarahkan ke halaman detail transaksi(Gambar 3.5). Di sini ditampilkan detail dari transaksi bersama dengan tombol edit dan hapus transaksi. Bilamana tombol hapus transaksi dipencet transaki akan menanyakan anda apabila anda yakin ingin menghapusnya, bila iya maka anda akan kembali ke halaman home dengan transaksi yang sebelumnya terhapus.

Sementara bila anda memilih untuk mengedit transaksi, anda akan diarahkan ke Halaman Tambah Transaksi(Gambar 3.4).Setelah anda mengubah seluruh isian yang tersedia, and dapat memencet simpan transaksi transaksi yang anda pilih sebelumnya telah merubah. Anda bisa mengubah seluruh bagian dari transaksi termasuk tanggal transaksi itu terlaksana.



Gambar 3.6  
Tampilan Menabung OurMoney

Tampilan menabung OurMoney mencakup data-data target menabung pengguna. Tampilan menabung akan menampilkan data angka target menabung, tanggal tenggat waktu menabung, dan jumlah uang yang terkumpul. Bila Tidak Ada target maka Target akan berkata bila target tidak aktif, sementara bila ada target maka akan diisi dengan target yang telah anda tentukan.Di bawah adalah deadline target anda dan yang di paling bawah adalah perkembangan anda menuju target tersebut.



Gambar 3.7  
Tampilan Target Baru OurMoney

Bila anda memilih untuk membuat targer baru maka anda akan dialihkan ke halaman ini. Di halaman ini anda bisa mengisi target anda dan tanggat waktu target anda. Setelah mengisi target dan masa tenggat waktu, anda bisa memencet tombol tambah target untuk selesai membuat target baru.

Gambar 3.8  
Tampilan Laporan OurMoney

Tampilan laporan OurMoney menyajikan laporan keuangan pengguna. Laporan berbentuk grafik pie chart sebanyak 2 buah yang masing-masing akan menampilkan data pengeluaran dan pemasukan. Rentang waktu dari data yang ditampilkan dapat diatur melalui sebuah spinnerber sama dengan wallet yang dipilih. Memampukan pengguna untuk menampilkan transaksi secara keseluruhan,atau bisa memilih diantara bulan ini, hingga 3 bulan yang lalu dengan wallet pilihan.

Seluruh pengeluaran dan pemasukan akan ditampilkan di bawah chart, pemasukan dan pengeluaran. Mereka digabungkan berdasakan kategori, jadi bila ada 2 atau lebih transaksi dengan kategori makanan, maka jumlahnya akan digabung menjadi 1 dan dimasukkan sebagai makanan di sini. Dan terakhir setelah semua kategori telah dicetak maka untung atau rugi secara keseluruhan akan dicatat di bagian paling bawah halaman ini.



Gambar 3.9

Tampilan detail laporan OurMoney

Tampilan detail laporan OurMoney memberikan pengguna detail informasi mengenai jenis transaksi yang dilakukan pada laporan. Terdapat jumlah transaksi yang dilakukan pada bagian atas halaman. Juga pada bagian bawahnya terdapat total dari jumlah transaksi dan transaksi apa saja yang dilakukan oleh pengguna. Halaman ini dapat di akses dengan cara menekan sub-report dari pemasukan atau pengeluaran pada halaman report aplikasi OurMoney.



Gambar 3.10  
Tampilan profil OurMoney

Tampilan profil OurMoney akan berfungsi sebagai tampilan profil dan pengaturan sekaligus. Terdapat nama pengguna aplikasi pada bagian atas beserta saldo total, terdapat juga pilihan untuk mengelola wallet dan kategori serta pilihan untuk mengimpor/expor data. Terakhir juga terdapat pilihan untuk mengubah nama anda bila anda ingin mengubah nama tampilan anda.

OurMoney memiliki layar aktifitas dengan beragam fungsi. Layar aktifitas akan membuka suatu tampilan baru di luar tampilan utama. Aktifitas yang bisa dibuka antara lain adalah aktifitas penambahan transaksi, pengubahan (edit) transaksi, pengelolaan wallet, pengelolaan jenis kategori, penambahan target menabung dan ekspor impor data.



Gambar 3.11  
Tampilan Halaman Wallet OurMoney

Ini halaman wallet anda. Halaman ini berisi seluruh wallet anda. Di kanan atas ada simbol yang dapat dipencet agar bisa menambah wallet baru, atau anda bisa mengklik salah satu walletnya agar bisa mengedit wallet yang dipilih.



Gambar 3.12  
Tampilan Tambah Wallet OurMoney

Bila anda menglik simbol + di kanan atas, maka anda akan dialihkan ke halaman ini untuk membuat wallet baru. Isi nama wallet dan saldo awal wallet tersebut lali klik simpan wallet maka wallet baru anda siap dipakai. Setelah itu anda akan diarahkan ke halaman wallet dengan wallet baru anda.



Gambar 3.13  
Tampilan Edit Wallet OurMoney

Tapi bila anda menglik wallet di halaman sebelumnya(Gambar 3.6), maka anda akan masuk ke halaman ini . Anda cukup mengisi nama wallet anda. Setelah itu anda klik simpan wallet dan wallet anda akan berubah.Perubahan anda akan terlihat setelah anda memencet tombol simpan wallet.



Gambar 3.14  
Tampilan Halaman Kategori OurMoney

Ini adalah tampilah kategori OurMoney, disini anda bisa melihat seluruh kategori anda. Anda bisa memisahkan berdasarkan pemasukan maupun pengeluaran. Dan seperti di halaman wallet, anda bisa menekan simbol + di kanan atas untuk menambah kategori dan menekan nama kategori untuk mengeditnya.

Gambar 3.15  
Tampilan Halaman Tambah/Edit Kategori OurMoney

Ini adalah kalaman untuk menambah atau mengedit kategori anda. Anda hnya perlu memasukkan nama kategori, dan pilih jenis dari kategori tersebut. Setelah itu anda bisa menekan tombol simpan kategori dan kategori anda seharusnya kelihatan di halaman Kategori.



Gambar 3.16  
Tampilan Halaman Ekspor/Impor Data OurMoney

Terakhir kita memiliki halaman expor/impor data. Halaman ini berguna bila anda mau melanjutkan perkembangan anda di telepon genggam lainnya atau mau menghapus seluruh isi telepon anda, anda bisa memakai halaman ini. Ekspor data dapat ditekan untuk mengeluarkan file yang bisa dibaca saat impor data ditekan. Dan juga ditampilkan lokasi file anda disimpan, sehingga anda dapat mengambil file itu, menempatkan file tersebut ke hp lain dengan lokasi yang sama lalu klik impor data untuk mengambil data dari file tersebut.

1. Desain Database



Gambar 3.17  
Conceptual Data Model



Gambar 3.18  
Physical Data Model

Database lokal terdiri atas 4 tabel utama, yaitu:

* Wallet : Wallet dapat dibilang seperti tempat menampung uang anda. Wallet akan berisi nama wallet anda dan jumlah uang yang dimiliki di dalam wallet anda dengan nama balance. Wallet dapat berjumlah lebih dari 1 buah. Meskipun sederhana, Wallet adalah bagian dari fitur utama dari aplikasi OurMoney.
* Category : Category adalah sebuah sebuah penanda yang dipasangkan ke transaksi anda sebagai keterangan apa yang akan dilakukan dengan pengeluaran / pendapatan tersebut. Dalam sebuah category terdapat category\_name yang berisi nama kategori tersebut dan category\_type. Mereka menentukan apakah kategori tersebut dapat dipasang ke pengeluaran seperti makanan, minuman, biaya listrik dll sementara pemasukan bisa seperti Gaji, Bonus, Kado dst.
* Transaction : Transaction adalah seluruh transaksi yang pernah dilakukan oleh user OurMoney. Transaction merupakan fitur utama dari aplikasi OurMoney dikarenakan transaksi inilah yang menggabungkan seluruh table yang lain. Di dalam transaction terdapat :
  + wallet\_id untuk menarik wallet apa yang akan dipengaruhi
  + category\_id untuk menarik kategori dari transaksi ini
  + timestamp untuk menentukan kapan transaksi ini dilaksanakan
  + note adalah keterangan tambahan yang bisa dimasukkan ke transaction tetapi tidak bersifat wajib.
  + Value adalah jumlah uang yang dipakai/didapatkan dalam transaksi ini.
* RTrans adalah tabel untuk report yang akan digenerate di setiap awal hari, minggu dan bulan untuk melihat report user OurMoney. Isi dari RTrans adalah:
  + Wallet\_id : Menarik wallet yang reportnya akan ditampilkan
  + Report\_date : Tanggal report diciptakan
  + Starting\_balance : Jumlah uang di awal hari/minggu/bulan
  + Ending\_balance : Jumlah uang di saat report sudah difinalkan
  + Total\_income : Total pemasukan selama report masih aktif
  + Total\_expense : Total pengeluaran selama report masih aktif
  + Saving\_target : Jumlah target berhemat yang telah ditentukan user

1. Desain Procedural

OurMoney tersusun atas beberapa fitur-fitur yang penting dalam pengelolaan keuangan pribadi. Fitur-fitur yang telah disebutkan dapat digambarkan dalam pseudocode sebagai sebuah prosedur.

Algoritma 3.1 Algoritma Menampilkan Report

1. SIAPKAN ARRAYLIST KATEGORI
2. SIAPKAN ARRAYLIST JUMLAH (SEUKURAN DENGAN ARRAYLIST KATEGORI)
3. SIAPKAN ARRAYLIST WALLET
4. SIAPKAN ARRAYLIST TRANSAKSI
5. SIAPKAN ARRAYLIST TRANSAKSIFILTERED
6. IF WALLET DIPILIH “SELURUH WALLET” THEN
7. ISI TRANSAKSIFILTERED DENGAN SEMUA ISI TRANSAKSI
8. ELSE
9. ISI TRANSAKSIFILTERED DENGAN TRANSAKSI DI WALLET DIPILIH
10. ENDIF
11. IF WAKTU DIPILIH “KESELURUHAN” THEN
12. FOR i = 0 KE UKURAN ARRAYLIST TRANSAKSI - 1
13. FOR j = 0 KE UKURAN ARRAYLIST KATEGORI -1
14. IF KATEGORI TRANSAKSI i = ARRAYLIST KATEGORI j
15. TAMBAHKAN JUMLAH KE J SEBANYAK TRANSAKSIFILTERED KE-i
16. BREAK LOOP
17. END IF
18. ELSE
19. PENGULANGAN LOOP SEBELUMNYA DENGAN TAMBAHAN PENGECEKAN BULAN DI IF-NYA
20. END IF
21. MENGISI PIECHART DENGAN DATA YANG TELAH DIGUNAKAN
22. TOTAL = 0
23. FOR I = 0 TO UKURAN KATEGORI
24. IF KATEGORI KE I PENGELUARAN
25. TOTAL = TOTAL – JUMLAH KE I
26. ELSE
27. TOTAL = TOTAL + JUMLAH KE I
28. END IF
29. IF TOTAL>0 ATAU TOTAL = 0
30. TAMPILKAN UNTUNG KESELURUHAN : TOTAL
31. ELSE
32. TTOTAL = TOTAL \* -1
33. TAMPILKAN RUGI KESELURUHAN : TOTAL
34. END IF

Algoritma 3.1, yang telah ditampilkan, akan digunakan untuk menampilkan laporan keuangan di OurMoney. Dimulai dengan menyiapkan arraylist yang akan dipakai yaitu arraylist Kategori untuk semua kategori yang tersedia, Jumlah yang menyimpan total dari isi kategori dengan index yang sama, lalu terakhir adalah arraylist transaksi yang berisi semua transaksi. Lalu dilanjutkan ke proses kodenya.

Dimulai denga menyaring data sesuai dengan wallet yang dipilih di baris 6-10m agar data yang ditampilkan sesua dengan wallet yang dipilih Lalu pada line ke – 11 hingga 17, setiap transaksi akan diperiksa kategorinya lalu jumlah dari transaksinya dimasukkan di index arraylist jumlah yang sesuai. Proses ini berulang terus hingga seluruh transaksi sudah dipakai. Sementara line 18 hingga 20 melakukan loop yang mirip dengan satu perbedaan, yaitu dengan menambah pemeriksaan bulan di dalam ifnya dengan tanggal transaksi.

Setelah itu bagian terakhir dari pseudocode ini dipakai untuk menunjukkan rugi atau untung yang dimiliki. Dengan memulai total dengan 0 lalu mengecek seluruh jumlah ,kategorinya dan mengecek apakah kategori itu pengeluaran, kita bisa mendapatkan keseluruhan total dari seluruh data kita. Total itu lalu dipakai untuk ditampilkan di layer utama dalam bentuk untung maupun rugi

Algoritma 3.2 Algoritma Add/Update Wallet/Kategori

1. SIAPKAN KATEGORI LAMA YANG INGIN DIEDIT JIKA ADA (KATEG)
2. SIAPKAN WALLET LAMA YANG INGIN DIEDIT JIKA ADA (WALL)
3. NAMA = NAMA YANG DIINPUTKAN;
4. IF NAMA TIDAK DIISI MAKA
5. CETAK PESAN "Mohon diisi semuanya";
6. ELSE
7. IF YANG DIINPUTKAN ADALAH KATEGORI MAKA
8. IF KATEG TIDAK KOSONG THEN
9. ATUR PENGISIAN KATEGORI MENJADI UPDATE KATEGORI
10. ENDIF
11. LAKUKAN TAMBAH KATEGORI;
12. ELSE
13. IF SALDO AWAL TIDAK ADA ISI DAN INI BUKAN EDIT THEN
14. CETAK PESAN "Mohon diisi semuanya";
15. ELSE
16. IF WALL TIDAK KOSONG THEN
17. ATUR PENGISIAN WALLET MENJADI UPDATE WALLET
18. ENDIF
19. LAKUKAN TAMBAH WALLET;
20. ENDIF
21. ENDIF
22. ENDIF

Algoritma 3.2 dipakai untuk menambah atau mengubah wallet atau kategori yang dipilih. Pertama kita menyiapkan seluruh variable yang diperlukan, seperti kateg atau wall yang dipakai untuk menandakan apabila sesi sekarang merupakan sebuah sesi editan atau tidak. Bila hedua variable tersebut tidak ada, maka bisa ditentukan bila sesi sekarang ini adalah sesi penambahan kategori atau wallet.

Di awal bila input nama dikosongkan, maka akan keluarkan pesan untuk isi semua lalu program selesai, tapi bila terisi maka akan ada pengecekan. Bila sekarang ini sedang mengolah kategori maka akan dijalankan kode baris nomor 8 hingga 11 sementara bila mengolah wallet akan dijalankan line 13 hingga 18. Ini terjadi dikarenakan Pemrosesan wallet dan kategori tidak terlalu panjang sehingga dapat digabung menjadi 1 untuk menghemat tempat dan sumber daya.

Di bagian pemrosesan kategori akan dilihat bila variable simpanan kategori kita bernama kateg kita kosong atau tidak. kateg menandakan bila sistem harus mengubah atau menambah kategori baru. Bila kateg kosong maka sistem akan diberitahu sekarang waktunya menambah kategori dan sebaliknya, itulah gunanya baris 8 dan 9, untuk mengubah penambahan kategori yang akan terjadi di baris 11 menjadi pengubahan kategori.Metode Pemrosesan wallet serupa dengan pemrosesan kategori dengan 1 perbedaan di baris 13 dan 14 dimana ada 1 inputan lagi yang harus ada agar bisa penambahan wallet.

Algoritma 3.3 Algoritma Ekspor Data

1. SIAPKAN LOKASI FILE;
2. SIAPKAN DAFTAR WALLET;
3. SIAPKAN DAFTAR KATEGORI;
4. SIAPKAN TARGET MENABUNG;
5. SIAPKAN DAFTAR TRANSAKSI;
6. TRY
7. BUKA FILE OUTPUT STREAM DARI LOKASI FILE;
8. BUKA OBJECT OUTPUT STREAM;
9. WRITEOBJECT DAFTAR WALLET;
10. WRITEOBJECT DAFTAR KATEGORI;
11. WRITEOBJECT TARGET MENABUNG;
12. WRITEOBJECT DAFTAR TRANSAKSI;
13. TUTUP OBJECT OUTPUT STREAM;
14. TUTUP FILE OUTPUT STREAM;
15. CETAK PESAN "Save data success!";
16. CATCH FILE TIDAK DITEMUKAN
17. CETAK PESAN "Save data fail!";
18. CATCH INPUT OUTPUT ERROR
19. CETAK PESAN "Save data fail!";
20. ENDTRY

Berikutnya adalah algoritma ekspor data. Pada algoritma 3.3 ditampilkan pseudocode algoritma untuk membuat file save data pada directory yang telah ditentukan. Baris pertama hingga ke-5 merupakan sequence persiapan di mana variabel lokasi dan object yang disimpan disiapkan. Baris ke-6 hingga ke-15 adalah try untuk melakukan ekspor data. Untuk melakukan ekspor data digunakan try-catch karena dalam proses ekspor yang melibatkan output stream dapat terjadi error yang bisa ditangkap (catch). Pada ujung program, baris ke-16 hingga ke-20 merupakan penutup algoritma di mana catch error diletakkan. Catch error berfungsi menangkap kesalahan yang terjadi dalam proses penulisan pada output stream yang terbuka.

Algoritma 3.4 Algoritma Impor Data

1. SIAPKAN STRING LOKASI FILE;
2. TRY
3. BUKA FILE INPUT STREAM DARI STRING LOKASI FILE;
4. BUKA OBJECT INPUT STREAM;
5. DAFTARWALLET = (ARRAYLIST<WALLET>)BACA OBJECT;
6. DAFTARCATEGORY = (ARRAYLIST<CATEGORY>)BACA OBJECT;
7. TARGETMENABUNG = (TARGET)BACA OBJECT;
8. DAFTARTRANSAKSI = (ARRAYLIST<TRANSAKSI>)BACA OBJECT;
9. SIMPAN 4 VARIABEL KE DATABASE LOKAL;
10. TUTUP OBJECT INPUT STREAM;
11. TUTUP FILE INPUT STREAM;
12. CATCH FILE TIDAK DITEMUKAN
13. CETAK PESAN ERROR;
14. CATCH INPUT OUTPUT ERROR
15. CETAK PESAN ERROR;
16. CATCH CLASS TIDAK DITEMUKAN
17. CETAK PESAN ERROR;
18. ENDTRY

Selanjutnya adalah algoritma untuk impor data yang ditampilkan di algoritma 3.4 di atas. Baris yang pertama berisi persiapan lokasi file untuk dimuat. Selanjutnya pada baris ke-2 hingga ke-11 berisi try untuk melakukan impor data. File input stream dibuka dan kemudian dari stream tersebut object-object dari file akan dibaca dan dimasukkan ke variabel. Baris ke-12 hingga ke-18 berisi catch error untuk menangkap kesalahan yang terjadi pada proses pembacaan input stream.

Algoritma 3.5 Algoritma Tambah/Edit Transaksi

1. SIAPKAN VARIABEL BOOLEAN EDIT ATAU BUKAN (ISEDIT)
2. SIAPKAN VARIABEL TRANSAKSI (TRANSAKSI)
3. SIAPKAN VARIABEL WALLET TRANSAKSI (WALLET)
4. IF TEXTBOX JUMLAH TRANSAKSI KOSONG THEN
5. WRITE LINE ERROR;
6. ELSE
7. IF TEXTBOX JUMLAH TRANSAKSI > 0 THEN
8. IF ISEDIT = true THEN
9. IF TRANSAKSI BARU ADALAH PENGELUARAN THEN
10. AMOUNT WALLET DIKURANGI JUMLAH TRANSAKSI;
11. ELSE
12. AMOUNT WALLET DITAMBAH JUMLAH TRANSAKSI;
13. ENDIF
14. SIMPAN TRANSAKSI YANG DIUPDATE KE DATABASE LOKAL;
15. ELSE
16. IF TRANSAKSI BARU ADALAH PENGELUARAN THEN
17. IF TRANSAKSI MEMBUAT TOTAL PENDAPATAN NEGATIF THEN
18. SHOW MESSAGE ("Nominal melebihi budget. Apakah ingin melanjutkan??");
19. IF ANSWER = YES THEN
20. TAMBAH TRANSAKSI BARU KE DATABASE LOKAL;
21. ENDIF
22. ELSE
23. TAMBAH TRANSAKSI BARU KE DATABASE LOKAL;
24. ENDIF
25. ELSE
26. TAMBAH TRANSAKSI BARU KE DATABASE LOKAL;
27. ENDIF
28. ENDIF
29. ELSE
30. WRITE LINE ERROR;
31. ENDIF
32. ENDIF

Terakhir adalah algoritma untuk menambah maupun mengedit transaksi.dimulai dari menyiapkan data yang diperlukan di baris 1 hingga 3, lalu di baris 4 dicek bila input transaksi kosong maka akan ditampilkan pesan bila jumlah harus diisi. Bila sudah diisi maka akan di lanjut ke tahab selanjutnya.

Di baris 8 dicek apakah proses yang akan dilaksanskan sebuah perubahan transaksi, bila iya maka baris 9 – 13 akan dijalankan dimana diperiksa bila transaksi pengeluaran atau tudak, dan jumlah wallet yang dipilih akan diatur sesuai dengan jenisnya dan masukkan datanya ke perangkat.

Bila proses sekarang bukanlah sebuah edit, maka baris 16 – 27 yang dilaksanakan. Dicek apakah transaksi adalah sebuah pengeluaran atau tidak. Bila transaksi adalah sebuah pengeluaran dan saldo wallet tidak mencukupi (baris 17) maka akan dikeluarkan peringatan bila jumlah transaksi melebihi budget wallet. Bila pengguna menjawab iya (baris 19) maka transaksi baru tersebut masuk kedalam data perangkat, sementara bila tidak proses akan berhenti. Lalu bila tranasksi merupakan sebuah pemasukkan maka transaksi tsb akan masuk langsung tanpa pengecekan.

BAB IV

IMPLEMENTASI

Penggunaan aplikasi OurMoney cukup mudah. OurMoney adalah aplikasi pengelolaan keuangan yang dikemas dalam bentuk aplikasi seluler. Karena itu penggunaannya relatif fleksibel dan praktis. OurMoney dapat digunakan di mana saja.

Pada saat pengguna menggunakan OurMoney untuk pertama kalinya, aplikasi OurMoney akan meminta nama pengguna. Nama yang diinputkan bebas sesuai kebutuhan pengguna. Oleh karena OurMoney berjalan secara luring(luar jaringan), maka pemberian nama tidak perlu ketentuan yang rumit. Nama yang telah diinputkan akan ditampilkan dalam beberapa tampilan dan dapat diubah lagi pada halaman profil jika dibutuhkan.

1. Program Pembuatan Report

­­Tujuan utama dari sebuah aplikasi pengelolaan keuangan tidak lain adalah mencatat data keuangan. Dari catatan data tersebut dapat menyimpulkan kondisi keuangan pengguna. Agar pengguna aplikasi dapat membaca kondisi keuangannya dengan nyaman dan mudah, maka dibentuklah fitur report. Fitur ini akan menyajikan data keuangan dari seluruh input transaksi pengguna dalam bentuk daftar dan grafik (chart).

**Segmen Program 4.1** **Class GetTransactionAndCategories**

1. class GetTransactionAndCategories {
2. private final WeakReference<Context> weakContext;
3. private final WeakReference <GetTransactionAndCategoriesCallback> weakCallback;
4. public GetTransactionAndCategories( Context context, GetTransactionAndCategoriesCallback callback){
5. this.weakContext = new WeakReference<>(context);
6. this.weakCallback = new WeakReference<>(callback);
7. }
8. void execute(){
9. ExecutorService executorService = Executors.*newSingleThreadExecutor*();
10. Handler handler = new Handler(Looper.*getMainLooper*());
11. executorService.execute(() -> {
12. Context context = weakContext.get();
13. AppDatabase appDatabase = AppDatabase.*getAppDatabase*(context);
14. List<Category> cat = appDatabase.appDao().getallCategories();
15. List<TransactionWithRelation> tra = appDatabase.appDao().getAllTransactions();
16. handler.post(()->{
17. weakCallback.get().postExecute(cat,tra);
18. });
19. });
20. }
21. interface GetTransactionAndCategoriesCallback{
22. void postExecute(List<Category> c, List<TransactionWithRelation> t);
23. }
24. }

Segmen program 4.1 menampilkan sebuah class bernama GetTransactionAndCategories yang berfungsi untuk mendapatkan data dari database lokal. Class memiliki 2 property yaitu weakContext dan weakCallback. Baris ke-4 hingga ke-7 merupakan constructor dari class. Baris ke-8 hingga ke-20 adalah method untuk mengeksekusi pengambilan data dari database lokal. Hasil pembacaan daftar kategori akan disimpan pada variabel List of Category bernama “cat”. Hasil pembacaan daftar transaksi dengan relasi akan disimpan pada List of TransactionWithRelation bernama "tra”. Terakhir, pada baris ke-21 hingga ke-23 memuat sebuah interface bagi class agar dapat melakukan callback.

**Segmen Program 4.2 Method setdata**

1. void setdata(int indextime,int indexwallet) {
2. ArrayList<TransactionWithRelation> filteredtrans = new ArrayList<>();
3. if (indexwallet == 0) {
4. filteredtrans.addAll(trans);
5. }else{
6. for (TransactionWithRelation tr: trans) {
7. if(tr.transaction.getWallet\_id() == wallets.get(indexwallet).getWallet\_id()){
8. filteredtrans.add(tr);
9. }
10. }
11. }
12. int tempmonth, total;
13. total = 0;
14. entries = new ArrayList<>();
15. entries2 = new ArrayList<>();
16. ArrayList<Float> jumlah = new ArrayList<>();
17. ArrayList<Integer> colors1 = new ArrayList<>();
18. ArrayList<Integer> colors2 = new ArrayList<>();
19. for (int i = 0; i < cats.size(); i++) {
20. jumlah.add((float) 0);
21. }
22. if (indextime == 0) {
23. for (int i = 0; i < trans.size(); i++) {
24. for (int j = 0; j < cats.size(); j++) {
25. if (trans.get(i).category.getCategoryId() == cats.get(j).getCategoryId()) {
26. jumlah.set(j, Math.*abs*(trans.get(i).transaction.getTransaction\_amount()) + jumlah.get(j) + indextime);
27. break;
28. }
29. }
30. }
31. } else {
32. for (int i = 0; i < trans.size(); i++) {
33. for (int j = 0; j < cats.size(); j++) {
34. tempmonth = currmonth - indextime + 1;
35. if (tempmonth < 0) tempmonth += 12;
36. tempmonth = Integer.*parseInt*((String) DateFormat.*format*("MM", trans.get(i).transaction.getTransaction\_date())) - tempmonth;
37. if (trans.get(i).category.getCategoryId() == cats.get(j).getCategoryId() && tempmonth == 0) {
38. jumlah.set(j, Math.*abs*(trans.get(i).transaction.getTransaction\_amount()) + jumlah.get(j));
39. break;
40. }
41. }
42. }
43. }
44. ReportAdapter adapter1 = new ReportAdapter(jumlah, cats, false);
45. binding.rvmasuk.setLayoutManager(new LinearLayoutManager(getActivity()));
46. binding.rvmasuk.setAdapter(adapter1);
47. ReportAdapter adapter2 = new ReportAdapter(jumlah, cats, true);
48. binding.rvkeluar.setLayoutManager(new LinearLayoutManager(getActivity()));
49. binding.rvkeluar.setAdapter(adapter2);
50. for (int i = 0; i < cats.size(); i++) {
51. if (jumlah.get(i) > 0) {
52. if (cats.get(i).isPengeluaran()) {
53. entries.add(new PieEntry(jumlah.get(i), cats.get(i).getCategoryName()));
54. total -= jumlah.get(i);
55. colors1.add(Color.*rgb*(220 + colors1.size() \* 5, 20 + colors1.size() \* 30, 20 + colors1.size() \* 35));
56. } else {
57. entries2.add(new PieEntry(jumlah.get(i), cats.get(i).getCategoryName()));
58. total += jumlah.get(i);
59. colors2.add(Color.*rgb*(0 + colors2.size() \* 30, 0 + colors2.size() \* 30, 255));
60. }
61. }
62. }
63. if(total>=0){
64. String displayBalance = "Rp. " + String.*format*("%,.0f", Float.*parseFloat*(total+""));
65. binding.tvtotalreport.setText("Untung keseluruhan : "+displayBalance);
66. }else{
67. total = total \* -1;
68. String displayBalance = "Rp. " + String.*format*("%,.0f", Float.*parseFloat*(total+""));
69. binding.tvtotalreport.setText("Rugi keseluruhan : "+displayBalance);
70. }
71. PieDataSet dataSet = new PieDataSet(entries, "");
72. PieDataSet dataSet2 = new PieDataSet(entries2, "");
73. dataSet.setColors(colors1);
74. dataSet2.setColors(colors2);
75. dataSet.setValueFormatter(new PercentFormatter(binding.pie1));
76. dataSet2.setValueFormatter(new PercentFormatter(binding.pie2));
77. PieData data = new PieData(dataSet);
78. data.setValueTextColor(Color.*WHITE*);
79. PieData data2 = new PieData(dataSet2);
80. data2.setValueTextColor(Color.*WHITE*);
81. binding.pie1.getDescription().setEnabled(false);
82. binding.pie2.getDescription().setEnabled(false);
83. binding.pie1.setUsePercentValues(true);
84. binding.pie2.setUsePercentValues(true);
85. binding.pie1.getLegend().setEnabled(false);
86. binding.pie2.getLegend().setEnabled(false);
87. binding.pie1.setData(data);
88. binding.pie1.highlightValues(null);
89. binding.pie1.invalidate();
90. binding.pie2.setData(data2);
91. binding.pie2.highlightValues(null);
92. binding.pie2.invalidate();
93. }

Pada segmen program 4.2 ditampilkan sebuah method bernama setdata. Method ini berfungsi menampilkan data report pada interface yang tersedia. Method memiliki parameter integer bernama “indextime” dan “indexwallet” yang berfungsi menentukan rentang waktu report dan wallet sumber yang ditampilkan. Bila index tersebut 0 berarti pengguna memilih untuk melihat seluruh transaksi yang tersedia sementara bila indexnya bukan 0 berarti pengguna memilih transaksi di kurun waktu tertentu atau/dan dari wallet tertentu.

Baris 2 hingga baris 11 merupakan penyaringan data yang akan ditampilkan berdasarkan wallet yang dipilih. Lalu setelah itu di baris 22 akan dilihat index yang dipilih oleh pengguna, bila indexnya adalah 0 atau keseluruhan maka akan lanjut ke baris 23 sementara bila tidak maka akan dipindah ke baris 32. Kode di baris 23 sampai 30 melihat semua transaksi yang ada, dilihat dengan kategori yang ada, membandingkan kategori setiap transaksi dan memasukkan ke arraylist jumlah yang sesuai. Baris 32 hingga 42 juga sama, tetapi dengan pengecekan tambahan waktu transaksi.

Selanjutnya baris 52 hingga 55 menyiapkan daftar semua pengeluaran sementara baris 57 hingga 59 menyiapkan semua pemasukan. Setelah itu baris 52 hingga 62 melihat semua arraylist jumlah dan kategorinya lalu membuat entri data untuk bisa ditampilkan di piechart dan sekalian menyiapkan warna dari tampilannya.Lalu terakhir di baris 53 hingga 60 menunjukkan jumlah untung atau rugi dari semua transaksi sementara baris 73 – 92 menyiapkan tampilan untuk piechart.

Segmen Program 4.3 Program utama membuat report

1. ArrayList<PieEntry> entries;
2. ArrayList<PieEntry> entries2;
3. FragmentReportBinding binding;
4. ArrayList<TransactionWithRelation> trans, filteredtrans;
5. ArrayList<Category> cats;
6. int currmonth,pilihanwallet,pilihantime;
7. @Override
8. public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {
9. super.onViewCreated(view, savedInstanceState);
10. new GetTransactionAndCategories(getContext(), new GetTransactionAndCategories.GetTransactionAndCategoriesCallback() {
11. @Override
12. public void postExecute(List<Category> c, List<TransactionWithRelation> t) {
13. cats = new ArrayList<>();
14. cats.addAll(c);
15. trans = new ArrayList<>();
16. trans.addAll(t);
17. wallets = new ArrayList<>();
18. Wallet wtemp = new Wallet("Seluruh Wallet",0);
19. wtemp.setWallet\_id(-1);
20. wallets.add(wtemp);
21. wallets.addAll(w);
22. ArrayAdapter<Wallet> spinnerArrayAdapter = new ArrayAdapter<Wallet>(getContext(), android.R.layout.simple\_spinner\_item,wallets);
23. spinnerArrayAdapter.setDropDownViewResource( android.R.layout.simple\_spinner\_dropdown\_item);
24. binding.spnPilihwallet.setAdapter( spinnerArrayAdapter);
25. entries = new ArrayList<>();
26. entries2 = new ArrayList<>();
27. setdata(0,0);
28. binding.spnPilihrentang. setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
29. @Override
30. public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l) {
31. Pilihantime = i;
32. setdata(I,pilihanwallet);
33. }
34. @Override
35. public void onNothingSelected(AdapterView<?> adapterView) {
36. }
37. });
38. binding.spnPilihwallet. setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
39. @Override
40. public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l) {
41. Pilihanwallet = i;
42. setdata(pilihantime,i);
43. }
44. @Override
45. public void onNothingSelected(AdapterView<?> adapterView) {
46. }
47. });
48. }
49. }).execute();
50. currmonth = Integer.*parseInt*((String) DateFormat.*format*("MM", new Date()));}

Kemudian terdapat program utama yang menggunakan class GetCategoryAndTransaction dan method setdata. Pada baris pertama hingga baris ke-6 memuat seluruh variabel yang digunakan dalam satu fragment secara keseluruhan. Baris ke-7 hingga baris yang terakhir merupakan method override onViewCreated untuk sebuah fragment dalam sebuah project Android Studio. Method ini mutlak dijalankan seriap kali fragment ini ditampilkan pada layar aplikasi.

Dalam method onViewCreated terdapat dua bagian utama. Bagian pertama adalah baris ke-10 hingga baris ke-49 yang berisi pemanggilan class GetCategoryAndTransaction. Class tersebut memiliki interface method bernama "postExecute" yang dipanggil di baris ke-12 hingga baris ke-48. Pada baris ke-13 hingga baris ke-16 dilakukan pengambilan data category dan transaction dari database lokal. Dilanjutkan dengan baris 17 hingga 21 mempersiapkan arraylist dompet yang akan ditampilkan dengan tambahan wallet Bernama “Seluruh Wallet” yang menunjukkan bila pengguna ingin melihat report penuhnya,

Baris ke-25 dan ke-26 membuat instance arraylist baru. Baris ke-27 memanggil method setdata dengan parameter 0 yang berarti menampilkan seluruh transaksi. Baris ke-30 hingga ke-37 adalah method yang membuat sebuah spinner pada interface melakukan perubahan rentang waktu pada data yang ditampilkan dalam report. Dapat terlihat bahwa method setdata dipanggil setiap item dari spinner dipilih, yaitu pada baris ke-30 hingga ke-33, hal yang seupa juga terjadi di baris ke-38 hingga ke-49 dimana daripada memilih rentang waktu, yang dipilih adalah wallet-nya.

1. Program Penyedia Data Transaksi Home

OurMoney menampilkan data pada home dengan mengambil data beserta relasinya dari database lokal. Kemudian data-data tersebut diolah agar dapat ditampilkan dan dimengerti dengan mudah oleh pengguna.

Segmen program 4.4 Class GetTransactionAsync

1. public class getTransactionAsync {
2. private final WeakReference<Context> weakContext;
3. private final WeakReference<getTransactionCallback> weakCallback;
4. private List<TransactionWithRelation> transactionList;
5. public getTransactionAsync(Context context, getTransactionCallback callback) {
6. this.weakContext = new WeakReference<>(context);
7. this.weakCallback = new WeakReference<>(callback);
8. }
9. public void execute() {
10. ExecutorService executorService = Executors.newSingleThreadExecutor();
11. Handler handler = new Handler(Looper.getMainLooper());
12. weakCallback.get().preExecute();
13. executorService.execute(() -> {
14. Context context = weakContext.get();
15. AppDatabase appDatabase = AppDatabase.getAppDatabase(context);
16. transactionList = appDatabase.appDao().getAllTransactions();
17. handler.post(() -> {
18. weakCallback.get().postExecute(transactionList);
19. });
20. });
21. }
22. public interface getTransactionCallback {
23. void preExecute();
24. void postExecute(List<TransactionWithRelation> transactionList);
25. }
26. }

Potongan program diatas merupakan sebuah class asynchronous yang berguna untuk mengambil dan mengembalikan data dari database lokal. Data yang dikembalikan kemudian akan diproses pada potongan program 4.5 yang berisi procedure SetupRV().

Segmen program 4.5 Procedure SetupRV untuk menampilkan data pada home

1. binding.rvHome.setLayoutManager(new LinearLayoutManager(getContext()));
2. binding.rvHome.setHasFixedSize(true);
3. transactions.addAll(transactionList);
4. int totalIncome=0, totalExpense=0;
5. for (int i=0; i<transactions.size(); i++){
6. if (transactions.get(i).category.isPengeluaran()){
7. totalExpense += transactions.get(i).transaction.getTransaction\_amount();
8. }
9. else{
10. totalIncome += transactions.get(i).transaction.getTransaction\_amount();
11. }
12. }
13. String displayIncome = String.format("%,.0f", Float.parseFloat(totalIncome+""));
14. String displayExpense = String.format("%,.0f", Float.parseFloat(totalExpense+""));
15. binding.lblIncome.setText("Rp " + displayIncome);
16. binding.lblExpense.setText("Rp " + displayExpense);
17. adapter = new TransactionAdapter(transactions);
18. adapter.setOnItemClickCallback(new TransactionAdapter.OnItemClickCallback() {
19. @Override
20. public void onItemClicked(TransactionWithRelation transaction) {
21. Intent moveData = new Intent(getActivity(), DetailActivity.class);
22. moveData.putExtra("transaction", transaction);
23. resultLauncher.launch(moveData);
24. }
25. });
26. binding.rvHome.setAdapter(adapter);

Potongan program diatas berguna untuk mengolah data yang dikembalikan oleh potongan program 4.4. pada baris ke 4, data yang dikembalikan dalam bentuk List dimasukkan kedalam ArrayList agar dapat diproses oleh adapter pada baris 20-29. Pada baris 6-14 dihitung total pemasukan dan pengeluaran dari data transaksi agar dapat ditampilkan pada overview halaman home. Lalu pada 15-16 data tersebut diformat desimalnya agar lebih mudah dibaca dan kemudian pada baris 17-18 nilai yang tadi sudah diformat dimasukkan kedalam TextView untuk ditampilkan pada halaman home.

1. Program Ekspor dan Impor Data

OurMoney memiliki fitur ekspor dan impor data yang berguna agar pengguna tidak kehilangan kemajuan menabung mereka. Eskspor berguna untuk membuat file dari seluruh data yang dimiliki sementara Impor berguna untuk membaca file yang tersedia. Bila ingin melanjukan kemajuan pengguna di perangkat lain bisa diambil file di lokasi yang tercetak dan memindahkannya ke perangkat lain.

Segmen Program 4.6 Procedure doWriteSerialize untuk membuat save file

1. String pathToOurMoneyFolder = getExternalFilesDir(null).getAbsolutePath();
2. String fileName = pathToOurMoneyFolder + File.separator +"saveourmoney.txt";
3. FileOutputStream fos = null;
4. try {
5. fos = new FileOutputStream(fileName);
6. ObjectOutputStream os = new ObjectOutputStream(fos);
7. os.writeObject(currentTarget);
8. os.writeObject(daftarwallet);
9. os.writeObject(daftarkategori);
10. os.writeObject(daftartrans);
11. os.close();
12. fos.close();
13. Toast.makeText(getApplicationContext(), "Berhasil", Toast.LENGTH\_SHORT).show();
14. System.out.println("Save data success!");
15. } catch (FileNotFoundException e) {
16. e.printStackTrace();
17. System.out.println("Save data fail!");
18. } catch (IOException e) {
19. e.printStackTrace();
20. System.out.println("Save data fail!");
21. }

Potongan kode diatas berguna untuk membuat file dari data yang ada. Dimulai dari baris 1 – 3, disiapkan 3 variabel yaitu secara berurutan lokasi file akan dibuat, nama file dan isi dari file tersebut yang masih kosong. Setelah ketiga hal tersebut telah siap maka akan dilaksanakan try catch agar bila terjadi kesalahan program tidak mati mendadak.

Di baris 5 didefinisikan fileoutputstream baru agar bisa dipakai, lalu didefinisikan di baris selanjutnya sebuat objectoutputstream yang dipakai untuk mengisi data primitif ke dalam fileoutputsteam agar bisa dicetak. Setelah kedua objek itu dibuat maka line 7 hingga 10 program menulis seluruh data yang penting disimpan dalam urut target,wallet,kategori dan transaksi pengguna. Setelah itu di line 11 dan 12 objectoutputstream dan fileoutputstream ditutup untuk menandakan bila file sudah dicetak.

Bila semuanya berjalan lancer, maka kode di line 13 yang berguna untuk mencetak “Berhasil” akan dijalankan. Tapi bila ada kesalahan nanti akan dicek error yang terjadi dengan baris 15 atau 18, dimana secara berurutan mengecek bila ada error dengan file dan ada error dengan penulisan ke dalam filenya. Bila ada error akan dicetak di compiler errornya dari baris 16 dan 19 lalu akan dicetak juga bila save gagal di compiler.

Segmen Program 4.7 Procedure doReadSerialize untuk membaca save file

1. String pathToOurMoneyFolder = getExternalFilesDir(null).getAbsolutePath();
2. String fileName = pathToOurMoneyFolder + File.separator +"saveourmoney.txt";
3. FileInputStream fis = null;
4. try {
5. fis = new FileInputStream(fileName);
6. ObjectInputStream is = new ObjectInputStream(fis);
7. currentTarget = (SavingTarget) is.readObject();
8. daftarwallet = (ArrayList<Wallet>) is.readObject();
9. daftarkategori = (ArrayList<Category>) is.readObject();
10. daftartrans = (ArrayList<MoneyTransaction>) is.readObject();
11. System.out.println("Wallets : "+daftarwallet.size());
12. System.out.println("Categories : "+daftarkategori.size());
13. System.out.println("Current target : "+currentTarget.toString());
14. System.out.println("Trans : "+daftartrans.toString());
15. new SaveDatabaseAsync(daftarwallet, daftarkategori, daftartrans, currentTarget, getApplicationContext(), new SaveDatabaseAsync.SaveDatabaseCallback() {
16. @Override
17. public void postExecute(String message) {
18. Toast.makeText(getApplicationContext(),message, Toast.LENGTH\_SHORT).show();
19. }
20. }).execute();
21. is.close();
22. fis.close();
23. }catch (FileNotFoundException e) {
24. e.printStackTrace();
25. } catch (IOException e) {
26. e.printStackTrace();
27. } catch (ClassNotFoundException e) {
28. e.printStackTrace();
29. }

Sementara potongan kode di atas digunakan untuk membaca file untuk dimasukkan ke dalam perangkat anda. Baris 1 dan dua secara beruntun dipakai untuk menyiapkan lokasi dari file dan nama dari file tersebut, sementara baris 3 menyiapkan FileInputStream yang dipakai untuk mendapat file yang ditentukan.

Di baris 5 dan 6 didefinisikan FileInputStream sebelumnya dan dibuat ObjectInputStream agar bisa membaca isi filenya. Setelah itu selesai baris 6 hingga 10 dipakai untuk mengambil is file satu demi satu dan dimasukkan ke dalam perangkat. File yang diambil secara berurutan adalah target, wallet, kategori dan trans.

Setelah itu di baris 11 hingga 14 dipakai untuk memastikan apabila file sudah masuk. Bila tidak ada error maka di baris 15 sampai 20 akan dilaksanakan proses untuk mengubah data yang sekarang ada dengan menghapus semua data di perangkat lalu menggantikannya dengan data yang barusan didapat. Setelah semua itu selesai maka akan ditup FileInputStream dan ObjectInputStream yang ada agar tidak memakan memori. Dengan itu baris 23 hingga 29 dipakai untuk mencetak error di console bila ada kesalahan.

BAB V

UJI COBA

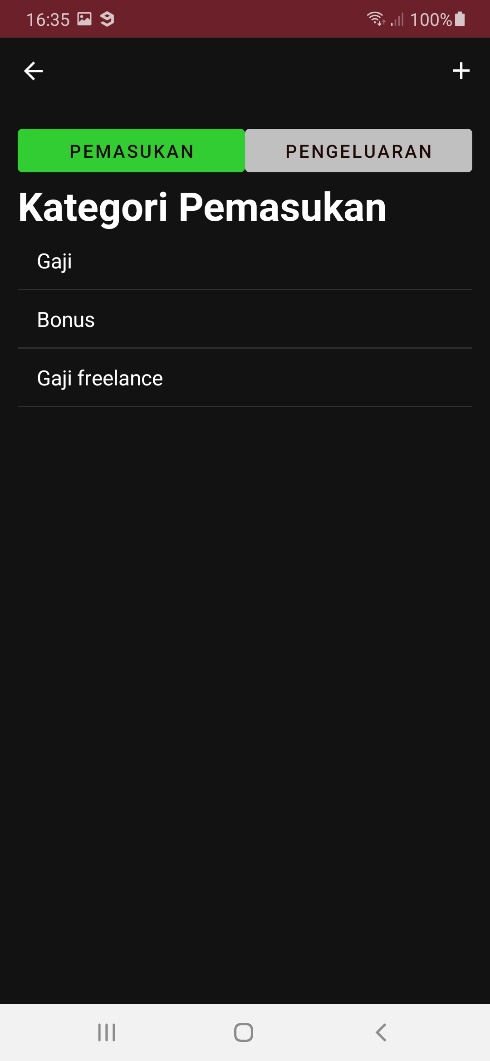
OurMoney dibuat dengan usaha agar aplikasi yang dihasilkan sangat matang dan dapat berjalan dengan baik sesuai fungsinya. Untuk memastikan kualitas baik, maka dilakukan serangkaian pengujian dalam berbagai bentuk pada aplikasi tersebut. Pengujian aplikasi OurMoney dilakukan menggunakan program emulator Android dan telepon genggam Android asli.

1. Uji Coba Mengubah Isi File Save Data

OurMoney memiliki sistem ekspor dan impor data menggunakan sebuah file save data. File tersebut memiliki nama “saveourmoney” dengan format .txt dan berisi data penyimpanan OurMoney. Pada pengujian ini akan dilakukan pengubahan konten yang dimuat file save data.

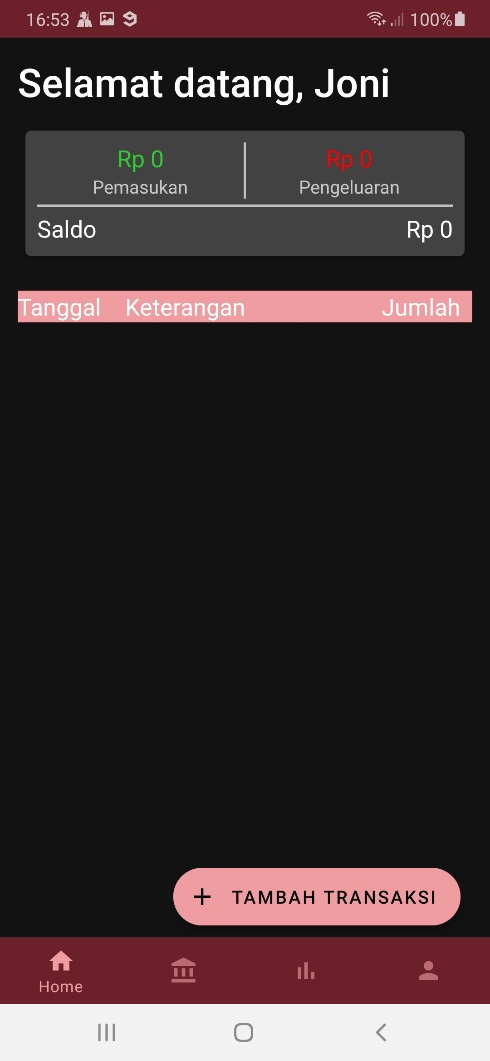
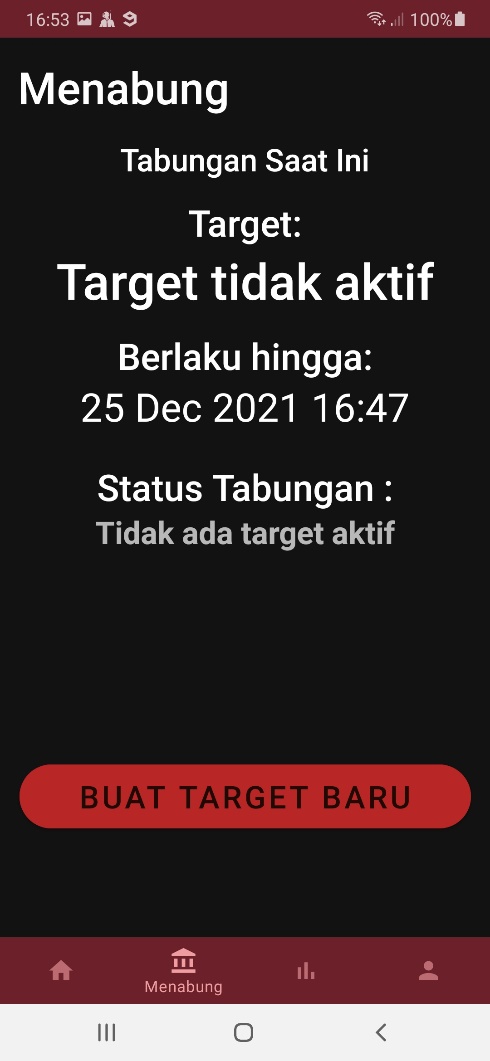
 

Gambar 5.1  
Tampilan transaksi dan target menabung yang diekspor

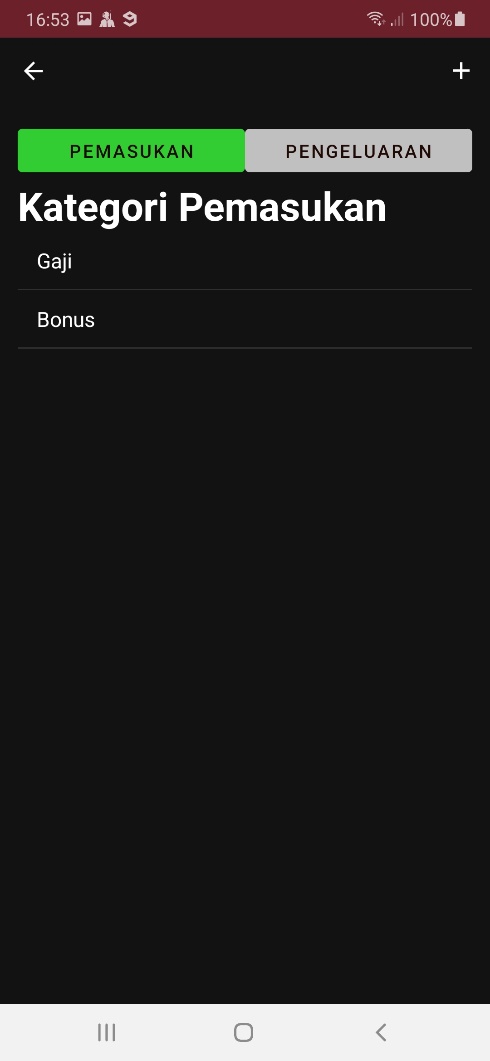
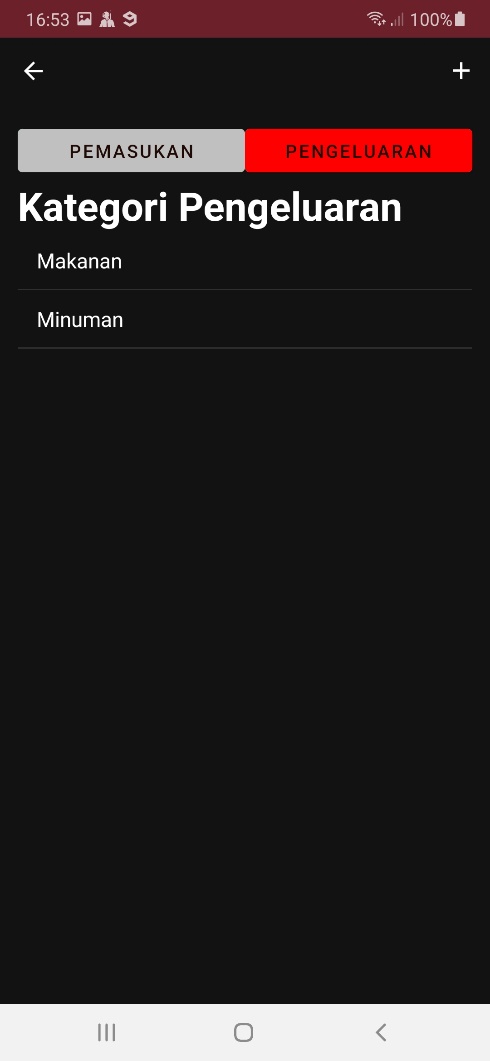
 

Gambar 5.2  
Daftar kategori sebelum diekspor

Untuk skenario percobaan ini, akan disiapkan 3 buah transaksi dengan nominal berbeda, 2 buah kategori tambahan dengan nama berbeda dan jenis yang berbeda, dan sebuah target menabung dengan nominal 500,000 (lima ratus ribu) dan berakhir pada 14 Januari 2022. Gambar 5.1 dan gambar 5.2 menunjukkan bagaimana data-data tersebut telah dimasukkan ke aplikasi. Kemudian lakukan ekspor file pada layar ekspor impor agar file save data terisi. Isi file yang diekspor akan dihapus pada bagian tertentu secara acak. Langkah yang terakhir adalah melakukan uninstall aplikasi agar data lokal benar-benar hilang dan melakukan impor data dari file save data yang isinya sudah berubah.

Gambar 5.3  
Daftar transaksi dan target menabung setelah diimpor

Gambar 5.4  
Daftar kategori setelah diimpor

Uji coba berhasil dengan cukup baik. File save data yang telah diubah tidak dapat dimuat aplikasi sama sekali. Aplikasi dapat mengetahui file tersebut tidak dapat dibaca sehingga tidak ada transaksi, kategori, atau target yang terbaca secara tidak lengkap. Dapat dibuktikan dengan hasil tangkapan layar (screenshot) pada gambar 5.3 dan 5.4 di atas. Dapat disimpulkan bahwa sistem impor data dalam OurMoney tidak bisa dilakukan dengan file apapun selain file save data yang asli.

1. Uji Coba Mengubah Sebuah Transaksi Ketika Target Menabung Aktif

Pada pengujian ini dilakukan perubahan pada transaksi yang ada sebelum dan sesudah target dibuat. Target ini dipengaruhi oleh transaksi yang ditambahkan oleh pengguna. Oleh karena itu fitur transaksi dan target menabung berelasi dan dapat mempengaruhi satu sama lain. Perlu dilakukan pengujian pada relasi tersebut agar kesalahan yang ada dapat diperbaiki sehingga aplikasi dapat digunakan dengan baik.

Pada awalnya disiapkan beberapa data transaksi sebelum target ditambahkan. Diambil sebuah transaksi yang akan diubah nilainya, nilai awal transaksi adalah 50,000 (lima puluh ribu) yang berupa pemasukan. Lalu buat sebuah target menabung sebesar 100,000 (seratus ribu). Kemudian ditambahkan lagi beberapa data transaksi pengeluaran sebesar 5,000 (lima ribu) dan 15,000 (lima belas ribu) setelah target ditambahkan. Setelah itu dilakukan perubahan pada nominal dari data transaksi pemasukan yang tadi ditambahkan, nilainya akan diubah menjadi 80,000 (delapan puluh ribu).

Hasil uji coba adalah progress menabung menjadi 60,000 (enam puluh ribu). Padahal seharusnya hasil progress menabung menjadi 10,000 (sepuluh ribu) karena transaksi yang sudah dibuat sebelum target menabung seharusnya tidak mempengaruhi target sebesar nominal barunya, melainkan selisih dari nominal lama dan nominal barunya. Dapat disimpulkan bahwa perhitungan pada target menabung masih kurang akurat. Hal ini disebabkan oleh kesalahan perhitungan ketika melakukan update transaksi. Dimana progress menabung langsung dijumlahkan dengan nominal yang baru dan tidak dijumlahkan dulu selisihnya dengan nominal yang lama.

1. Uji Coba Melihat Report

Pada pengujian ini dilakukan percobaan pertama dalam melihat report. Seluruh komponen dari fragment report telah ditempatkan sesuai tempatnya dan fragment report akan dibuka, tapi ditemukan masalah, yaitu PieChart yang ditempatkan tidak ditampilkan di halaman laporan tersebut.

PieChart adalah sebuah visualisasi grafik yang menunjukkan seluruh data yang ingin ditunjukkan. Dimana seharusnya di telepon genggam sudah bisa dilihat grafik yang berisi seluruh data transaksi yang dipilih, tapi tidak tampil apa apa dan setelah diperiksa, penyebab utama dari kesalahan ini adalah PieChart dan LinearLayout yang menampung keddua PieChart memiliki property tinggi wrap\_content.

Dengan property wrap\_content itu seperti memberitahu PieChart dan LinearLayout bila mereka bisa membesar, tapi bila tidak harus mereka tidak perlu membersar. Dan karena itu, kedua bagian tsb merasa tidak perlu membesar sehingga mereka berada di ukuran awal mereka yaitu sekecil butiran debu di layer anda. Jadi cara kita memperbaikinya adalah memaksa PieChart untuk membesar agar mereka bisa kelihatan di layer dengan ukuran tertentu. Setelah pengaturan itu diubah, Piechart bisa tampil dan berjalan dengan lancar

BAB VI

PENUTUP

1. Kesimpulan

Proyek OurMoney merupakan proyek aplikasi pengelolaan keuangan yang memuat berbagai fitur. Karena itu pengerjaan proyek ini memakan waktu yang cukup banyak. Dari sepanjang waktu yang telah digunakan untuk proses pengerjaan proyek ini, kami menyimpulkan beberapa hal penting dari proyek ini. Seluruh kesimpulan dirangkum pada bagian berikut.

1. Aplikasi OurMoney memiliki sejumlah kegunaan yang memenuhi kebutuhan pengguna untuk mengelola keuangan. Di antaranya adalah sistem wallet ganda, pencatatan data transaksi pada wallet, pengelompokan jenis transaksi berdasarkan tipe (pemasukan atau pengeluaran) dan kategori, pembuatan target menabung, dan pembuatan laporan keuangan.
2. Aplikasi OurMoney berhasil dirancang sebagai aplikasi seluler praktis yang berjalan pada platform dengan jumlah pengguna terbanyak, yaitu Android. Dengan demikian, OurMoney memberikan kemudahan dan kesederhanaan dalam penggunaannya.
3. Aplikasi OurMoney memiliki sebuah kekurangan. Berjalannya OurMoney secara luar jaringan berarti aplikasi tersebut tidak menampung data pada sebuah server. Maka dari itu pemindahan data harus dilakukan secara luar jaringan juga, yaitu melalui sebuah save file.
4. Saran

Berdasarkan testing yang dilakukan pada aplikasi ini, diajukan saran-saran terhadap aplikasi sebagai berikut :

1. Tampilan pada aplikasi perlu ditingkatkan terutama pada tampilan home dan laporan aplikasi. Tampilan pada halaman tersebut masih terasa kaku dan kurang menarik. Misalnya pemilihan warna dan komponen pada halaman tersebut perlu difokuskan. Karena tampilan aplikasi merupakan salah satu daya tarik penting agar pengguna nyaman dalam menggunakan aplikasi.
2. Pada halaman home aplikasi, sebaiknya diberikan fitur pencarian atau pemilihan jangka waktu dari transaksi yang ingin ditampilkan. Agar pengguna dapat lebih mudah dalam mencari dan mengatur histori transaksi. Penambahan fitur ini juga berguna untuk menurunkan waktu pemuatan data pada home karena tidak perlu memuat semua data sekaligus dari database kedalam aplikasi.
3. Fitur impor dan ekspor data sebaiknya diberikan fitur untuk memilih jalur data akan disimpan atau dimuat dari mana. Fitur ini dapat lebih memudahkan pengguna dalam mengekspor dan mengimpor data. Pengguna tidak perlu lagi secara manual memindahkan data dari folder aplikasi. Atau untuk kedepannya bisa diimplementasikan fitur cloud save agar pengguna tidak perlu mengirim data dari satu perangkat ke perangkat lainnya jika ingin berpindah perangkat.

DAFTAR PUSTAKA

Anonymous, 2011. *The C++ Resource Network.* [Online]   
Available at: http://www.cplusplus.com  
[Accessed 21 November 2012].

Busby, J., Parrish, Z. & Wilson, J., 2010. *Mastering Unreal Technology Volume 1: Introduction to Level Design with Unreal Engine 3.* 1 ed. Indiana: Sams Publishing.

Anonymous, 2021. *Java*. [Online]

Available at: https://id.wikipedia.org/wiki/Java

[Accessed 20 November 2021]

Anonymous, 2021. *Android (sistem operasi)*. [Online]

Available at: https://id.wikipedia.org/wiki/Android\_(sistem\_operasi)

[Accessed 22 November 2021]

Anonymous, 2021. *SQL*. [Online]

Available at: https://id.wikipedia.org/wiki/SQL

[Accessed 29 Desember 2021]

Anonymous, 2021. *Android Studio*. [Online]

Available at: https://id.wikipedia.org/wiki/Android\_Studio

[Accessed 29 Desember 2021]

Anonymous, 2020. *Aplikasi Seluler*. [Online]

Available at: https://id.wikipedia.org/wiki/Aplikasi\_seluler

[Accessed 29 Desember 2021]