

**PENGEMBANGAN APLIKASI *MOBILE* MANAJEMEN
KEUANGAN DENGAN METODE SCRUM
(STUDI KASUS MAHASISWA FILKOM UB)**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Januari Arka
NIM: 145150207111003



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018**

PENGESAHAN

PENGEMBANGAN APLIKASI *MOBILE* MANAJEMEN KEUANGAN DENGAN METODE SCRUM (STUDI KASUS MAHASISWA FILKOM UB)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer

Disusun Oleh :
Januari Arka
NIM: 145150207111003

Skripsi ini telah diuji dan dinyatakan lulus pada
17 Oktober 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Adam Hendra Brata, S.Kom., M.T., M.Sc.
NIK: 2016079001051001

Komang Candra Brata, S.Kom., M.T., M.Sc.
NIK: 2016078907111001

Mengetahui
Ketua Jurusan **Teknik Informatika**

Tri Astoto Kurniawan, S.T., M.T., Ph.D.
NIP: 197105182003121001



PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 17 Oktober 2018



Januari Arka

NIM: 145150207111003

KATA PENGANTAR

Puji dan syukur penulis panjatkan atas kehadiran Allah SWT karena atas segala rahmat dan karunia – Nya, penulis dapat menyelesaikan skripsi yang digunakan sebagai syarat untuk memperoleh gelar Sarjana Komputer pada Program Studi Teknik Informatika pada Fakultas Ilmu Komputer Universitas Brawijaya. Skripsi yang dibuat berjudul “PENGEMBANGAN APLIKASI *MOBILE* MANAJEMEN KEUANGAN DENGAN METODE SCRUM (STUDI KASUS MAHASISWA FILKOM UB)” dengan tujuan mempermudah mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya dalam mengelola keuangan pribadinya.

Untuk kesempatan ini, penulis juga menyampaikan banyak terimakasih kepada pihak – pihak yang telah membantu penulis selama mengerjakan skripsi ini, diantaranya:

1. Allah SWT yang telah memberi kemudahan dalam proses pengerjaan skripsi ini.
2. Kepada kedua orang tua penulis, yaitu Bapak Ujang Suryana dan Ibu Mujiati beserta keluarga besar yang selalu memberikan saran, pengertian, motivasi, dan do’a.
3. Bapak Adam Hendra Brata, S.Kom, M.T., M.Sc., selaku dosen pembimbing I yang telah memberikan banyak waktu kepada penulis selama proses pengerjaan skripsi, baik dalam masukan, saran, motivasi, hingga ilmu yang tentunya sangat bermanfaat bagi penulis.
4. Bapak Komang Candra Brata, S.Kom., M.T., M.Sc., selaku dosen pembimbing II yang telah memberikan banyak waktu kepada penuli selama proses pengerjaan skripsi, baik dalam masukan, saran, motivasi, hingga ilmu yang tentunya sangat bermanfaat bagi penulis.
5. Seluruh Dosen Fakultas Ilmu Komputer Universitas Brawijaya yang telah memberikan banyak ilmu kepada penulis selama masa perkuliahan berlangsung.
6. Bapak Hermawan Dwi Putra, S.H., yang telah banyak memberikan motivasi dan semangat kepada penulis dalam mengerjakan skripsi.
7. Seluruh staff di Fakultas Ilmu Komputer yang telah membantu dalam proses akademik maupun non – akademik selama proses pengerjaan skripsi.
8. Aryun Nadaa Aniesiyah, yang telah banyak memberikan motivasi, saran, serta masukan kepada penulis dalam mengerjakan skripsi.
9. Ahmad Kamil Almahsyhur, yang telah banyak memberikan masukan, saran, dan mengajarkan tentang Firebase kepada penulis.

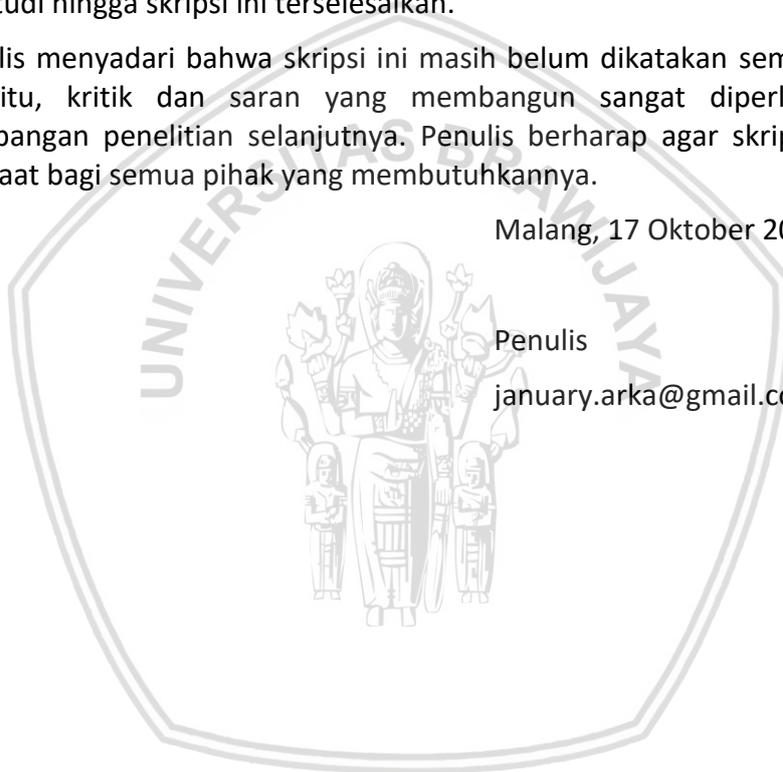
10. Aryun Nadaa Aniesiyah, Rizky Novriansyah, Gani Kharisma Wardana, Abel Filemon Haganta Kaban, dan Iqbal Agtitar Rofiy yang telah memberikan waktu dan kesediaanya dalam mengikuti pengujian *usability* yang penulis lakukan.
11. Rekan – rekan Advokesma EMIF 2016/2017 dan 2017/2018 yang telah memberikan bantuan dan semangat kepada penulis.
12. Rekan – rekan Gazebo Army yang telah memberikan bantuan dan semangat kepada penulis.
13. Teman – teman mahasiswa Fakultas Ilmu Komputer program studi Teknik Informatika angkatan 2014 yang telah memberikan bantuan selama masa studi hingga skripsi ini terselesaikan.

Penulis menyadari bahwa skripsi ini masih belum dikatakan sempurna. Oleh karena itu, kritik dan saran yang membangun sangat diperlukan untuk pengembangan penelitian selanjutnya. Penulis berharap agar skripsi ini dapat bermanfaat bagi semua pihak yang membutuhkannya.

Malang, 17 Oktober 2018

Penulis

january.arka@gmail.com



ABSTRAK

FM App adalah aplikasi manajemen keuangan untuk memecahkan masalah keuangan bagi mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya (FILKOM UB). FM App memiliki beberapa fitur untuk menyelesaikan masalah keuangan mahasiswa FILKOM UB, seperti pengeluaran harian yang melebihi batas harian, pengeluaran yang telah digunakan tidak dapat dilacak, pengeluaran tak terduga, dan perencanaan pembelian. Selain itu, FM App dapat digunakan dimana saja dan kapan saja karena dikembangkan dengan Android *mobile platform* yang merupakan mayoritas jenis *smartphone* yang digunakan mahasiswa dan Firebase untuk menyimpan data secara *realtime*.

FM App menggunakan algoritma Naïve Bayes untuk merekomendasikan status keuangan mahasiswa ketika pengeluaran tak terduga dibuat. Selain itu, pengembangan ini membutuhkan metode yang memiliki pendekatan langsung ke pengguna, membutuhkan umpan balik yang berkelanjutan, meningkatkan produktivitas pengembang, dan mempercepat pengembangan sistem. Oleh karena itu, metode Scrum telah digunakan. Pengembangan ini memiliki dua *sprint*, yaitu *sprint 1* untuk membuat fitur dasar FM App dan *sprint 2* untuk membuat fitur lanjutan pada FM App.

FM App memiliki nilai akurasi rekomendasi 100% pada pengujian algoritma Naïve Bayes, sehingga dapat disimpulkan bahwa aplikasi dapat merekomendasikan status keuangan dengan 100% benar setelah pengeluaran tak terduga dibuat. FM App juga memiliki nilai 79 berdasarkan pengujian *usability*, sehingga dapat dikategorikan sebagai aplikasi dengan tingkat kemudahan penggunaan yang baik. Selain itu, pengujian unit telah dilakukan menggunakan *white box testing* dengan teknik *basis path testing* dan pengujian validasi dengan *black box testing*. Pengujian tersebut memiliki nilai keberhasilan 100%, sehingga dapat disimpulkan bahwa pengerjaan aplikasi telah sesuai dengan perancangan yang telah dibuat.

Kata kunci: manajemen keuangan, Android, Firebase, Naïve Bayes, Scrum

ABSTRACT

FM App is a financial management application to solve financial problems for students of the Faculty of Computer Science Brawijaya University (FILKOM UB). FM App has several features to solve financial problems of FILKOM UB students, such as daily expenses that exceed daily limits, expenditures that have been used cannot be tracked, unexpected expenses, and purchase planning. In addition, FM App can be used anywhere and anytime because it is developed with an Android mobile platform which is mostly a type of smartphone used by students and Firebase to store data in realtime.

FM App uses the Naïve Bayes algorithm to recommend students' financial status when unexpected expenses are made. In addition, this development requires a method that has a direct approach to users, requires continuous feedback, increases developer productivity, and accelerates system development. Therefore, the Scrum method has been used. This development has two sprints, namely Sprint 1 to make the basic features of FM App and sprint 2 to make the advanced features of FM App.

FM App has a 100% recommendation accuracy value on the Naïve Bayes algorithm testing, so it can be concluded that the application can recommend financial status with 100% correct after unexpected expenses are made. FM App also has a value of 79 based on usability testing, so it can be categorized as an application with a good level of ease of use. In addition, unit testing has been done using white box testing with basis path testing techniques and validation testing with black box testing. The test has a 100% success value, so it can be concluded that the application work in accordance with the design that has been made.

Keyword: Financial Management, Android, Firebase, Naïve Bayes, Scrum

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xv
DAFTAR LAMPIRAN	xviii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	3
1.3 Tujuan	3
1.4 Manfaat.....	3
1.5 Batasan masalah	4
1.6 Sistematika pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN	6
2.1 Kajian Pustaka	6
2.2 Manajemen Keuangan	7
2.3 Naïve Bayes.....	7
2.3.1 Pengujian Akurasi.....	9
2.4 Firebase.....	9
2.5 Unified Modelling Language (UML)	10
2.5.1 <i>Use Case Diagram</i>	10
2.5.2 <i>Sequence Diagram</i>	11
2.5.3 <i>Class Diagram</i>	13
2.6 <i>Agile Development Method</i>	15
2.6.1 Scrum	15
2.7 Pengujian Validasi	16



2.8 Pengujian Unit.....	16
2.9 Pengujian <i>Usability</i>	17
2.9.1 System Usability Scale.....	17
BAB 3 METODOLOGI	20
3.1 Observasi Masalah	21
3.2 Landasan Kepustakaan	21
3.3 Rekayasa Kebutuhan.....	21
3.3.1 <i>Product Backlog</i>	21
3.3.2 <i>Sprint Planning</i>	22
3.4 Perancangan dan Implementasi	22
3.4.1 <i>Development</i>	22
3.5 Pengujian	23
3.5.1 Pengujian Unit.....	23
3.5.2 Pengujian Algoritma Naïve Bayes	23
3.5.3 <i>Sprint Review</i>	24
3.5.4 Pengujian <i>Usability</i>	24
3.6 Kesimpulan dan Saran	24
BAB 4 REKAYASA KEBUTUHAN.....	25
4.1 Gambaran Umum Sistem.....	25
4.2 Identifikasi Aktor.....	26
4.3 Daftar Kebutuhan	26
4.3.1 <i>Product Backlog</i>	26
4.3.2 Kebutuhan Fungsional.....	31
4.3.3 Kebutuhan Non Fungsional.....	32
4.4 <i>Use Case Diagram</i>	33
4.5 <i>Use Case Scenario</i>	34
4.5.1 <i>Use Case Scenario Sign In</i>	34
4.5.2 <i>Use Case Scenario Sign Up</i>	35
4.5.3 <i>Use Case Scenario</i> Melihat Keuangan Saya.....	36
4.5.4 <i>Use Case Scenario</i> Menambah Pengeluaran	37
4.5.5 <i>Use Case Scenario</i> Menambah Pemasukan	38
4.5.6 <i>Use Case Scenario</i> Melihat Riwayat Keuangan	39



4.5.7 <i>Use Case Scenario</i> Melihat Perencanaan Keuangan.....	39
4.5.8 <i>Use Case Scenario</i> Membuat Perencanaan Anggaran.....	40
4.5.9 <i>Use Case Scenario</i> Membuat Batas Harian.....	41
4.5.10 <i>Use Case Scenario</i> Melihat Perencanaan Pembelian.....	42
4.5.11 <i>Use Case Scenario</i> Melihat Profil.....	44
4.6 <i>Sprint Planning</i>	44
4.6.1 <i>Sprint Goal</i>	44
4.6.2 <i>Sprint Backlog</i>	44
4.7 Analisis Data.....	50
4.7.1 <i>Sprint 1</i>	50
4.7.2 <i>Sprint 2</i>	51
BAB 5 PERANCANGAN DAN IMPLEMENTASI.....	52
5.1 Perancangan.....	52
5.1.1 Perancangan Arsitektural.....	52
5.1.2 Perancangan Komponen.....	64
5.1.3 Perancangan Basis Data.....	67
5.1.4 Perancangan Antarmuka.....	82
5.2 Implementasi.....	103
5.2.1 Spesifikasi Sistem.....	103
5.2.2 Batasan Implementasi.....	103
5.2.3 Implementasi Kode Program.....	104
5.2.4 Implementasi Basis Data.....	114
5.2.5 Implementasi Antarmuka.....	115
5.3 <i>Daily Scrum</i>	122
5.3.1 <i>Sprint 1</i>	123
5.3.2 <i>Sprint 2</i>	123
BAB 6 PENGUJIAN.....	124
6.1 Pengujian Unit.....	124
6.1.1 <i>Sprint 1</i>	124
6.1.2 <i>Sprint 2</i>	136
6.2 Pengujian Algoritma Naïve Bayes.....	140
6.2.2 Kasus Uji Pengujian Algoritma Naïve Bayes.....	144



6.3 Pengujian Validasi	171
6.3.1 <i>Sprint</i> 1	171
6.3.2 <i>Sprint</i> 2	189
6.4 Pengujian <i>Usability</i>	207
6.5 Analisis Hasil Pengujian.....	208
6.5.1 Analisis Hasil Pengujian Unit	208
6.5.2 Analisis Hasil Pengujian Algoritma Naïve Bayes	208
6.5.3 Analisis Hasil Pengujian Validasi	209
6.5.4 Analisis Hasil Pengujian <i>Usability</i>	210
BAB 7 Penutup	211
7.1 Kesimpulan.....	211
7.2 Saran	212
DAFTAR PUSTAKA.....	213
LAMPIRAN A HASIL KUESIONER UNTUK LATAR BELAKANG.....	215
LAMPIRAN B HASIL KUESIONER UNTUK REKOMENDASI KETIKA PENGELUARAN TIDAK TERDUGA DILAKUKAN	219
LAMPIRAN C KUESIONER <i>USABILITY TESTING</i>	231



DAFTAR TABEL

Tabel 2. 1 Elemen <i>Use Case Diagram</i>	10
Tabel 2. 2 Elemen <i>Sequence Diagram</i>	11
Tabel 2. 3 Lembar Kuesioner <i>System Usability Scale (SUS)</i>	17
Tabel 4. 1 Gambaran Umum FM App	25
Tabel 4. 2 Identifikasi Aktor	26
Tabel 4. 3 Daftar <i>Product Backlog</i>	27
Tabel 4. 4 Kebutuhan Fungsional	32
Tabel 4. 5 Kebutuhan Non Fungsional	33
Tabel 4. 6 Informasi Nama <i>Use Case</i> Berdasarkan Kode Fungsi	33
Tabel 4. 7 <i>Use Case Scenario Sign In</i>	34
Tabel 4. 8 <i>Use Case Scenario Sign Up</i>	35
Tabel 4. 9 <i>Use Case Scenario</i> Melihat Keuangan Saya	36
Tabel 4. 10 <i>Use Case Scenario</i> Menambah Pengeluaran.....	37
Tabel 4. 11 <i>Use Case Scenario</i> Menambah Pemasukan.....	38
Tabel 4. 12 <i>Use Case Scenario</i> Melihat Riwayat Keuangan	39
Tabel 4. 13 <i>Use Case Scenario</i> Melihat Perencanaan Keuangan	39
Tabel 4. 14 <i>Use Case Scenario</i> Membuat Perencanaan Anggaran	40
Tabel 4. 15 <i>Use Case Scenario</i> Membuat Batas Harian	41
Tabel 4. 16 <i>Use Case Scenario</i> Melihat Perencanaan Pembelian	42
Tabel 4. 17 <i>Use Case Scenario</i> Melihat Profil.....	44
Tabel 4. 18 Deskripsi Pekerjaan pada <i>Sprint 1</i>	45
Tabel 4. 19 Deskripsi Pekerjaan pada <i>Sprint 2</i>	48
Tabel 4. 20 Data Pada <i>Sprint 1</i>	50
Tabel 4. 21 Data Pada <i>Sprint 2</i>	51
Tabel 5. 1 Algoritma <i>Sign In</i>	64
Tabel 5. 2 Algoritma Tambah Pengeluaran Tidak Dlanggarkan.....	65
Tabel 5. 3 Algoritma Membuat Batas Harian.....	66
Tabel 5. 4 Algoritma Melihat Halaman Keterangan Kategori Anggaran.....	66
Tabel 5. 5 Perancangan Data FM App.....	67
Tabel 5. 6 Deskripsi Perancangan Basis Data <i>Sprint 1</i>	81

Tabel 5. 7 Deskripsi Perancangan Basis Data <i>Sprint 2</i>	82
Tabel 5. 8 Penjelasan <i>Screenflow</i> Umum FM App	94
Tabel 5. 9 Spesifikasi Sistem.....	103
Tabel 5. 10 Kode Program <code>SignIn.java</code>	104
Tabel 5. 11 Kode Program <code>TambahPengeluaran.java</code>	105
Tabel 5. 12 Kode Program <code>BudgetPlanning.java</code>	111
Tabel 6. 1 Algoritma <i>Sign In</i> pada klas <code>SignIn</code>	125
Tabel 6. 2 Kasus Uji Algoritma <i>Sign In</i>	126
Tabel 6. 3 Algoritma Data Uji pada klas <code>TambahPengeluaran</code>	127
Tabel 6. 4 Kasus Uji Algoritma Data Uji.....	134
Tabel 6. 5 Algoritma <code>onClick</code> pada klas <code>BudgetPlanning</code>	136
Tabel 6. 6 Penentuan Status Keuangan Setelah Pengeluaran Tidak Terduga Terjadi	140
Tabel 6. 7 Pengelompokan Anggaran Pada Kategori.....	141
Tabel 6. 8 Konversi Anggaran Yang Diambil Menjadi Data <i>Training</i>	142
Tabel 6. 9 Data <i>Training</i>	144
Tabel 6. 10 Kondisi Aplikasi Sebelum Pengujian.....	145
Tabel 6. 11 Kasus Uji Pengujian Algoritma Naïve Bayes	145
Tabel 6. 12 Kasus Uji 1 Algoritma Naïve Bayes	146
Tabel 6. 13 Kasus Uji 2 Algoritma Naïve Bayes	148
Tabel 6. 14 Kasus Uji 3 Algoritma Naïve Bayes	151
Tabel 6. 15 Kasus Uji 4 Algoritma Naïve Bayes	153
Tabel 6. 16 Kasus Uji 5 Algoritma Naïve Bayes	156
Tabel 6. 17 Kasus Uji 6 Algoritma Naïve Bayes	158
Tabel 6. 18 Kasus Uji 7 Algoritma Naïve Bayes	160
Tabel 6. 19 Kasus Uji 8 Algoritma Naïve Bayes	163
Tabel 6. 20 <i>Resume</i> Pengujian Algoritma Naïve Bayes	166
Tabel 6. 21 Daftar Pengujian Fungsionalitas Pada <i>Sprint 1</i>	171
Tabel 6. 22 Kasus Uji <i>Sign Up</i>	171
Tabel 6. 23 Kasus Uji <i>Sign Up (Alternative Flow)</i>	172
Tabel 6. 24 Kasus Uji <i>Sign In</i>	172
Tabel 6. 25 Kasus Uji <i>Sign In (Alternative Flow 1)</i>	172

Tabel 6. 26 Kasus Uji <i>Sign In</i> (<i>Alternative Flow 2</i>)	173
Tabel 6. 27 Kasus Uji <i>Sign In</i> (<i>Alternative Flow 3</i>)	173
Tabel 6. 28 Kasus Uji Melihat Keuangan Saya.....	173
Tabel 6. 29 Kasus Uji Menambah Pengeluaran.....	173
Tabel 6. 30 Kasus Uji Menambah Pengeluaran (<i>Alternative Flow 1</i>).....	174
Tabel 6. 31 Kasus Uji Menambah Pengeluaran (<i>Alternative Flow 2</i>).....	174
Tabel 6. 32 Kasus Uji Menambah Pengeluaran (<i>Alternative Flow 3</i>).....	175
Tabel 6. 33 Kasus Uji Menambah Pengeluaran (<i>Alternative Flow 4</i>).....	175
Tabel 6. 34 Kasus Uji Menambah Pemasukan	175
Tabel 6. 35 Kasus Uji Melihat Riwayat Pemasukan.....	176
Tabel 6. 36 <i>Resume</i> Akhir Pengujian Validasi Fungsionalitas	177
Tabel 6. 37 Daftar Pengujian Fungsionalitas Pada <i>Sprint 2</i>	189
Tabel 6. 38 Kasus Uji Melihat Perencanaan Keuangan.....	189
Tabel 6. 39 Kasus Uji Membuat Perencanaan Anggaran	189
Tabel 6. 40 Kasus Uji Membuat Perencanaan Anggaran (<i>Alternative Flow 1</i>) ...	190
Tabel 6. 41 Kasus Uji Membuat Perencanaan Anggaran (<i>Alternative Flow 2</i>) ...	190
Tabel 6. 42 Kasus Uji Membuat Batas Harian	190
Tabel 6. 43 Kasus Uji Membuat Batas Harian (<i>Alternative Flow 1</i>)	191
Tabel 6. 44 Kasus Uji Membuat Perencanaan Pembelian	191
Tabel 6. 45 Kasus Uji Membuat Perencanaan Pembelian (<i>Alternative Flow 1</i>)..	192
Tabel 6. 46 Kasus Uji Melihat Profil	192
Tabel 6. 47 Kasus Uji Melihat Profil (<i>Alternative Flow 1</i>).....	192
Tabel 6. 48 Kasus Uji Melihat Profil (<i>Alternative Flow 2</i>).....	193
Tabel 6. 49 <i>Resume</i> Akhir Pengujian Validasi Fungsionalitas <i>Sprint 2</i>	194
Tabel 6. 50 Nilai Hasil Kuesioner Pengujian SUS.....	207
Tabel 6. 51 Nilai Hasil Kuesioner Pengujian SUS Setelah di Konversi	207

DAFTAR GAMBAR

Gambar 2. 1 Firebase Google.....	9
Gambar 2. 2 <i>Class Diagram</i>	13
Gambar 2. 3 <i>Inheritance (Generalization)</i>	14
Gambar 2. 4 <i>Association</i>	14
Gambar 2. 5 <i>Aggregation</i>	14
Gambar 2. 6 <i>Composition</i>	14
Gambar 2. 7 <i>Dependency</i>	15
Gambar 2. 8 <i>Scrum Sprint Life Cycle</i>	16
Gambar 2. 9 Contoh <i>Flow Graph</i>	17
Gambar 2. 10 Nilai <i>System Usability Scale</i>	19
Gambar 3. 1 Diagram Alir Metodologi Penelitian.....	20
Gambar 4. 1 <i>Use Case Diagram</i> FM App.....	34
Gambar 5. 1 Arsitektur FM App	52
Gambar 5. 2 <i>Class Diagram</i> Umum pada FM App	54
Gambar 5. 3 <i>Package Controller Sprint 1</i>	56
Gambar 5. 4 <i>Package Model Sprint 1</i>	57
Gambar 5. 5 <i>Package Controller Sprint 2</i>	58
Gambar 5. 6 <i>Package Model Sprint 2</i>	59
Gambar 5. 7 <i>Sequence Diagram Sign In</i>	60
Gambar 5. 8 <i>Sequence Diagram Sign Up</i>	61
Gambar 5. 9 <i>Sequence Diagram</i> Menambah Pengeluaran.....	62
Gambar 5. 10 <i>Sequence Diagram</i> Membuat Batas Harian	63
Gambar 5. 11 <i>Sequence Diagram</i> Membuat Perencanaan Anggaran	64
Gambar 5. 12 <i>Wireframe</i> Halaman Utama	83
Gambar 5. 13 <i>Wireframe Sign Up</i>	83
Gambar 5. 14 <i>Wireframe Sign In</i>	84
Gambar 5. 15 <i>Wireframe Forgot Password</i>	84
Gambar 5. 16 <i>Wireframe</i> Keuangan Saya	85
Gambar 5. 17 <i>Wireframe</i> Tambah Pengeluaran	86
Gambar 5. 18 <i>Wireframe</i> Tambah Pemasukan.....	86

Gambar 5. 19 <i>Wireframe</i> Riwayat Keuangan.....	87
Gambar 5. 20 <i>Wireframe</i> Perencanaan Keuangan	87
Gambar 5. 21 <i>Wireframe</i> Perencanaan Anggaran.....	88
Gambar 5. 22 <i>Wireframe</i> Keterangan Kategori Anggaran.....	89
Gambar 5. 23 <i>Wireframe</i> Batas Harian.....	89
Gambar 5. 24 <i>Wireframe</i> Perencanaan Pembelian	90
Gambar 5. 25 <i>Wireframe</i> Berdasarkan Jumlah Tabungan.....	90
Gambar 5. 26 <i>Wireframe</i> Berdasarkan Tanggal Pembelian.....	91
Gambar 5. 27 <i>Wireframe</i> Profil.....	91
Gambar 5. 28 <i>Wireframe Change Password</i>	92
Gambar 5. 29 <i>Screenflow</i> Umum FM App.....	93
Gambar 5. 30 <i>Screenflow Sign Up</i>	94
Gambar 5. 31 <i>Screenflow Sign In</i>	95
Gambar 5. 32 <i>Screenflow Forgot Password</i>	95
Gambar 5. 33 <i>Screenflow</i> Keuangan Saya.....	96
Gambar 5. 34 <i>Screenflow</i> Menambah Pengeluaran	96
Gambar 5. 35 <i>Screenflow</i> Menambah Pemasukan.....	97
Gambar 5. 36 <i>Screenflow</i> Riwayat Keuangan	97
Gambar 5. 37 <i>Screenflow</i> Perencanaan Keuangan	98
Gambar 5. 38 <i>Screenflow</i> Perencanaan Anggaran.....	98
Gambar 5. 39 <i>Screenflow</i> Keterangan Kategori Anggaran.....	99
Gambar 5. 40 <i>Screenflow</i> Batas Harian.....	99
Gambar 5. 41 <i>Screenflow</i> Perencanaan Pembelian.....	100
Gambar 5. 42 <i>Screenflow</i> Berdasarkan Jumlah Tabungan	100
Gambar 5. 43 <i>Screenflow</i> Berdasarkan Tanggal Pembelian	101
Gambar 5. 44 <i>Screenflow</i> Profil	101
Gambar 5. 45 <i>Screenflow Change Password</i>	102
Gambar 5. 46 <i>Screenflow Sign Out</i>	102
Gambar 5. 47 Implementasi Basis Data FM App	114
Gambar 5. 48 Tampilan Halaman Utama.....	115
Gambar 5. 49 Tampilan <i>Sign Up</i>	115
Gambar 5. 50 Tampilan <i>Sign In</i>	116



Gambar 5. 51 Tampilan <i>Forgot Password</i>	116
Gambar 5. 52 Tampilan Keuangan Saya	117
Gambar 5. 53 Tampilan Menambah Pengeluaran	117
Gambar 5. 54 Tampilan Menambah Pemasukan.....	118
Gambar 5. 55 Tampilan Riwayat Keuangan	118
Gambar 5. 56 Tampilan Perencanaan Keuangan.....	119
Gambar 5. 57 Tampilan Perencanaan Anggaran	119
Gambar 5. 58 Tampilan Keterangan Kategori Anggaran	120
Gambar 5. 59 Tampilan Batas Harian	120
Gambar 5. 60 Tampilan Perencana Pembelian.....	121
Gambar 5. 61 Tampilan Berdasarkan Jumlah Tabungan	121
Gambar 5. 62 Tampilan Berdasarkan Tanggal Pembelian	122
Gambar 5. 63 Tampilan Profil	122
Gambar 6. 1 <i>Flow Graph</i> Algoritma <i>Sign In</i>	126
Gambar 6. 2 <i>Flow Graph</i> Algoritma Data Uji	133
Gambar 6. 3 <i>Flow Graph</i> Algoritma <i>onClick</i>	139



DAFTAR LAMPIRAN

LAMPIRAN A HASIL KUESIONER UNTUK LATAR BELAKANG	215
LAMPIRAN B HASIL KUESIONER UNTUK REKOMENDASI KETIKA PENGELUARAN TIDAK TERDUGA DILAKUKAN	219
LAMPIRAN C KUESIONER <i>USABILITY TESTING</i>	231



BAB 1 PENDAHULUAN

1.1 Latar belakang

Manajemen keuangan merupakan suatu kegiatan yang didalamnya melakukan perencanaan, analisis, dan pengendalian keuangan untuk mencapai suatu tujuan dalam memaksimalkan penggunaan uangnya (Husnan, 2014). Hal ini dapat dikatakan bahwa manajemen keuangan berperan penting dalam memaksimalkan penggunaan uang. Dengan kata lain, manajemen kebutuhan perlu dilakukan agar dapat mensejahterakan masa depan dan dapat mengatasi permasalahan yang tidak diinginkan. Akan tetapi, pada tingkat mahasiswa manajemen keuangan masih belum optimal untuk dilakukan.

Sebagai contoh ialah mahasiswa Fakultas Ilmu Komputer (FILKOM) Universitas Brawijaya. Dari survei yang telah dilakukan dan dilampirkan pada lampiran A, dari 30 responden mahasiswa FILKOM menyatakan bahwa seluruh responden pernah mengeluarkan uangnya melewati batas yang telah ditentukan setiap harinya, padahal 60% menyatakan membatasi pengeluaran setiap harinya. Selain itu, 83,3% responden juga pernah tidak menyadari uangnya digunakan untuk apa saja. Disamping itu, 93,3% responden pernah mengeluarkan uang tak terduga yang sebelumnya tidak pernah direncanakan dengan rincian 50% menyatakan sering melakukannya dan 43,3% pernah melakukannya, padahal hal tersebut dapat merugikan pengeluaran untuk kedepannya. Survei menyatakan 66,7% responden mendapatkan uang setiap bulannya yang seharusnya dapat dikelola dengan baik agar kesalahan penggunaan uang tidak terjadi. Selain itu, keinginan mahasiswa FILKOM untuk membeli suatu barangpun menjadi suatu permasalahan yang tidak bisa mereka selesaikan. Sebanyak 80% responden menyatakan mereka menginginkan untuk membeli sesuatu, namun tidak tahu harus menabung berapa setiap harinya.

Untuk mengatasi permasalahan tersebut, penelitian yang berjudul "*Aplikasi Sistem Informasi Keuangan Mahasiswa Pada Telepon Pintar*" memberikan solusi berupa aplikasi sistem informasi keuangan berbasis website (Febriani & Fauzi, 2015). Dari hasil pengujian dapat disimpulkan bahwa mahasiswa dapat mengetahui informasi keuangannya dengan memasukan pengeluaran dan pemasukan pada aplikasi, melihat catatan pengeluaran dan pemasukan yang sudah ada di aplikasi, serta dapat melihat rincian keuangan berdasarkan harian maupun periode.

Penelitian lainnya berjudul "*Aplikasi Pengelola Keuangan Pribadi Berbasis Android*" (Susanto, Noertjahyana, & Setiawan, 2016) yang memberikan solusi dengan membuat aplikasi untuk mengelola keuangan berbasis Android. Dari hasil aplikasi yang dibuat terdapat fitur – fitur pada aplikasi tersebut, yaitu transaksi untuk mencatat pemasukan atau pengeluaran, pencatatan hutang piutang, *reminder* pengingat pembayaran tagihan hutang piutang, dapat memberi laporan dalam bentuk grafik, dapat membatasi anggaran pengeluaran, dapat membuat

tabungan sendiri, melihat kurs secara *realtime*, dan memberi *reminder* untuk membayar tagihan kartu kredit sebelum jatuh tempo.

Berdasarkan pemaparan penelitian yang sudah pernah dilakukan tersebut, dapat disimpulkan bahwa untuk penelitian pertama masih belum ada yang dapat mengatasi permasalahan yang ada pada mahasiswa FILKOM, yaitu permasalahan pengeluaran melewati batas setiap harinya karena hanya melakukan pencatatan tanpa ada yang fitur yang mengingatkan pengguna agar tidak melewati batas pengeluaran yang sudah ditentukan. Selanjutnya ialah penelitian kedua. Pada penelitian kedua juga masih belum dapat mengatasi permasalahan pada mahasiswa FILKOM. Pada permasalahan pernah mengeluarkan uangnya melewati batas yang telah ditentukan setiap harinya, penelitian ini memiliki fitur batas pengeluaran, namun untuk batas pengeluaran setiap kategori anggaran saja. Selain itu, tidak ada fitur untuk mengingatkan pengguna jika pengeluaran sudah mendekati batas yang sudah ditentukan.

Untuk menyelesaikan permasalahan tersebut, dibutuhkan suatu sistem yang dapat mengingatkan pengguna agar tidak mengeluarkan uangnya ketika mendekati batas yang telah ditentukan setiap harinya, memiliki perencanaan keuangan untuk mengatasi permasalahan pembelian suatu produk atau barang, dan dapat merekomendasikan status keuangannya ketika pengeluaran tidak terduga dilakukan. Dalam memberikan rekomendasi dibutuhkan suatu metode klasifikasi dari suatu data keuangan yang ada untuk mengetahui bagaimana status keuangan setelah terjadi pengeluaran tidak terduga. Naïve Bayes dipilih karena merupakan metode klasifikasi yang memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya (Saleh, 2015). Selain itu, sistem perlu dibuat dengan *mobile* karena seluruh responden menyatakan pengguna aktif *smartphone* dan jenis *smartphone* yang banyak digunakan pada responden, yaitu 86,7% merupakan pengguna *smartphone* berbasis Android. Perhatian pada kemudahan dalam penggunaan sistem juga diperlukan agar pengguna dapat menerima sistem yang ditawarkan (Nielsen, Usability 101: Introduction to Usability, 2012). Hal ini juga didasari dengan hasil survei yang menyatakan bahwa 96,7% responden menginginkan sistem yang mudah untuk digunakan.

Untuk membangun sistem tersebut diperlukan suatu siklus pengembangan sistem yang dapat mendefinisikan setiap langkah dalam proses pengembangan perangkat lunak agar pengembangan sistem ini dapat dibuat dengan baik dan jelas serta bisa sesuai dengan pengguna. Siklus tersebut adalah *System Development Life Cycle* (SDLC). SDLC merupakan proses yang terdiri dari rencana rinci yang menjelaskan cara mengembangkan, memelihara, mengganti, dan mengubah atau meningkatkan perangkat lunak tertentu (Tutorialspoin, 2018). SDLC memiliki banyak metodologi, namun untuk permasalahan ini dibutuhkan suatu metodologi yang dapat memastikan pengguna ikut berperan langsung dalam perancangan sistem agar mendapatkan kebutuhan yang sesuai dengan permasalahan yang ada pada lingkungan pengguna. Selain itu, metodologi yang dipilih mampu mempercepat pengembangan sistem. Oleh karena itu, Scrum dipilih. Scrum menggunakan pendekatan berulang yang iteratif yang membutuhkan *feedback*

secara berlanjut kepada pengguna dan pengembangan sistem juga dapat diselesaikan dengan cepat karena adanya *daily scrum* yang memungkinkan produktifitas dari pengembang meningkat (Schwaber & Sutherland, The Definitive Guide to Scrum: The Rules of the Game, 2016).

Oleh karena itu, penelitian ini dilakukan dengan harapan dapat menyelesaikan permasalahan tentang manajemen keuangan yang ada pada mahasiswa FILKOM Universitas Brawijaya sehingga dapat memaksimalkan dalam menggunakan uangnya. Selain itu, besar harapan agar hasil dari penelitian ini, yaitu aplikasi *mobile* untuk manajemen keuangan agar dapat digunakan oleh seluruh mahasiswa FILKOM Universitas Brawijaya.

1.2 Rumusan masalah

Dengan uraian latar belakang tersebut, maka dapat dirumuskan beberapa rumusan masalah terkait, yaitu:

1. Bagaimana cara memastikan pengguna ikut berperan langsung dalam perancangan sistem dan mempercepat pengembangan sistem?
2. Bagaimana hasil rekayasa kebutuhan dalam pengembangan aplikasi *mobile* manajemen keuangan?
3. Bagaimana cara mengatasi pengeluaran yang melewati batas harian?
4. Bagaimana cara mengatasi pengeluaran tidak terduga yang sebelumnya tidak dianggarkan?
5. Bagaimana cara merencanakan pembelian suatu barang?
6. Bagaimana tingkat kemudahan penggunaan pada aplikasi *mobile* manajemen keuangan mahasiswa Fakultas Ilmu Komputer (FILKOM) Universitas Brawijaya?

1.3 Tujuan

Tujuan yang ingin dicapai dalam penelitian ini ialah:

1. Memastikan pengguna ikut berperan langsung dalam perancangan sistem dan mempercepat pengembangan sistem.
2. Menjadikan aplikasi *mobile* manajemen keuangan sebagai solusi untuk permasalahan manajemen keuangan mahasiswa Fakultas Ilmu Komputer (FILKOM) Universitas Brawijaya.
3. Memberikan rekomendasi status keuangan mahasiswa Fakultas Ilmu Komputer (FILKOM) Universitas Brawijaya ketika pengeluaran tidak terduga telah terjadi.
4. Memudahkan penggunaan aplikasi *mobile* manajemen keuangan oleh mahasiswa Fakultas Ilmu Komputer (FILKOM) Universitas Brawijaya.

1.4 Manfaat

Dengan adanya penelitian ini, maka manfaat yang didapat adalah:

1. Mahasiswa Fakultas Ilmu Komputer (FILKOM) Universitas Brawijaya diharapkan dapat menjadikan aplikasi *mobile* manajemen keuangan sebagai solusi permasalahan keuangannya.

2. Mahasiswa Fakultas Ilmu Komputer (FILKOM) Universitas Brawijaya diharapkan dapat mengatasi permasalahan pengeluaran tidak terduga dengan kemampuan keuangan yang dimilikinya sekarang.
3. Mahasiswa Fakultas Ilmu Komputer (FILKOM) Universitas Brawijaya diharapkan dapat mengatasi permasalahan pembelian suatu barang dengan kemampuan keuangan yang dimilikinya sekarang.
4. Mahasiswa Fakultas Ilmu Komputer (FILKOM) Universitas Brawijaya diharapkan dapat mencatat keuangan pribadinya dimanapun dan kapanpun.

1.5 Batasan masalah

Agar penelitian terfokus dan tidak terjadi pelebaran topik permasalahan, maka dibatasi dalam hal sebagai berikut:

1. Penelitian ini berfokus pada pengeluaran, pemasukan, dan perencanaan keuangan pribadi.
2. Ruang lingkup penelitian ini dilakukan di Fakultas Ilmu Komputer (FILKOM) Universitas Brawijaya.
3. Aplikasi dibuat pada perangkat *mobile* Android.
4. Aplikasi hanya mampu berjalan pada *mobile* dengan sistem operasi Android 4.0.3 keatas.

1.6 Sistematika pembahasan

Sistematika penulisan penelitian ini dijelaskan sebagai berikut:

- BAB I. Pendahuluan
Berisi latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan pada penelitian pengembangan aplikasi *mobile* untuk manajemen keuangan.
- BAB II. Landasan Kepustakaan
Berisi tentang kajian teori dan dasar teori sebagai rujukan untuk mengerjakan penelitian dan merupakan inti dari dasar teori yang digunakan dalam penelitian.
- BAB III. Metodologi Penelitian
Berisi tentang alur tahapan perancangan aplikasi yang terdiri dari studi pustaka, perancangan dan implementasi, pengujian, penarikan kesimpulan dan saran.
- BAB IV. Rekayasa Kebutuhan
Berisi tentang analisis kebutuhan dari perangkat lunak yang akan dibuat serta digambarkan perancangan sistem yang akan dibangun sebelum dilakukan pengkodean kedalam suatu bahasa pemrograman.

- BAB V. Perancangan dan Implementasi
Berisi tentang perancangan dan implementasi aplikasi yang telah dibuat. Pada pembuatan aplikasi *mobile* untuk manajemen keuangan menggunakan bahasa pemrograman *java*.
- BAB VI. Pengujian
Berisi tentang pengujian terhadap aplikasi *mobile* untuk manajemen keuangan yang telah diimplementasikan serta melakukan analisa terhadap hasil pengujian sehingga diperoleh suatu kesimpulan.
- BAB VII. Penutup
Berisi tentang kesimpulan dan saran pada penelitian yang telah dilakukan.



BAB 2 LANDASAN KEPUSTAKAAN

Bab Landasan Kepustakaan berisi uraian dan pembahasan dasar teori sebagai rujukan untuk mengerjakan penelitian dan merupakan inti dari dasar teori yang digunakan dalam penelitian. Teori – teori dasar yang akan dibahas pada bab ini adalah Kajian Pustaka, Manajemen Keuangan, Naïve Bayes, Firebase, *Unified Modelling Language (UML)*, dan *Agile Software Development Method*, Pengujian Validasi, Pengujian Unit, Pengujian Usability.

2.1 Kajian Pustaka

Penelitian mengenai manajemen keuangan sudah pernah dilakukan oleh beberapa orang. Penelitian pertama adalah “Aplikasi Sistem Informasi Keuangan Mahasiswa Pada Telepon Pintar” (Febriani & Fauzi, 2015). Penelitian ini dilakukan berdasarkan keluhan yang sering terjadi dikalangan mahasiswa mengenai pengeluaran kebutuhannya. Pengontrolan keuangan yang kurang baik seperti pemasukan ataupun pengeluaran keuangan mahasiswa berdampak pada kesadaran mahasiswa terhadap besarnya biaya untuk kebutuhan perkuliahan itu kurang baik. Untuk menunjang permasalahan tersebut, sistem bernama SIKEMAH berbasis *website* dibuat. Dari hasil pengujian dapat disimpulkan bahwa dapat diketahuinya keuangan mahasiswa dan terkontrol dengan efektif, pencatatan pemasukan serta pengeluaran keuangan mahasiswa dapat dilakukan dengan mudah, melihat rincian keuangan berdasarkan harian maupun periode dengan mudah dapat diketahui.

Penelitian lainnya adalah “Aplikasi Pengelola Keuangan Pribadi Berbasis Android” (Susanto, Noertjahyana, & Setiawan, 2016). Penelitian ini dilakukan berdasarkan pengelolaan keuangan yang tidak mudah. Hal ini diakibatkan oleh tingginya sikap konsumtif pada mahasiswa serta generasi muda lainnya. Pengendalian serta perencanaan keuangan juga belum dapat dibuat sehingga tujuan individu belum tercapai. Dengan adanya permasalahan tersebut, maka dibuat aplikasi berbasis android yang bertujuan untuk memudahkan pengelolaan keuangan pada mahasiswa dan generasi muda lainnya. Harapannya, *user* dengan mudah belajar mengelola keuangannya dengan *gadget* yang dimilikinya dengan mudah. Perbedaan dengan penelitian sebelumnya, *user* dapat melihat laporan anggaran yang melebihi batas atau sesuai dengan anggaran yang sudah ditetapkan.

Penelitian lainnya adalah “Implementasi Metode Klasifikasi Naïve Bayes Dalam Memprediksi Besarnya Penggunaan Listrik Rumah Tangga” (Saleh, 2015). Penelitian ini menggunakan metode Naïve Bayes dalam memprediksi besarnya penggunaan listrik rumah tangga. Pada penelitian ini dilakukan 60 pengujian pada hasil prediksinya dan memperoleh hasil persentase sebesar 78,3333% yang berarti terdapat 47 kasus uji yang dapat diklasifikasikan dengan benar dari 60 kasus uji.

Penelitian lainnya adalah “Penerapan Metode Scrum dan Virtual Private Network Dalam Perancangan Sistem Ordersales” (Wijaya, 2018). Pada penelitian ini dalam pengembangannya menggunakan metode Scrum. Dengan mempertimbangkan kerangka kerja yang difokuskan pada pada tujuan utama

yang dibagi menjadi tujuan – tujuan kecil sehingga mudah dicapai dan definisi aplikasi yang tidak dapat digambarkan secara menyeluruh diawal sehingga ada kemungkinan perubahan pada *end user*, maka hal tersebut menjadi menjadi alasan kenapa Scrum digunakan pada penelitian tersebut.

Dari penelitian yang sudah dipaparkan tersebut, penelitian ini lebih dipusatkan pada *user* yang akan melakukan pengelolaan keuangannya. Aplikasi yang akan dibuat nantinya akan memiliki fitur yang berbeda yaitu, suatu *reminder* batas minimal yang dikeluarkan setiap harinya, perencanaan pembelian suatu barang dengan menentukan harga dari produk serta jumlah nabung setiap harinya, serta *reminder* untuk mengisi catatan pemasukan atau pengeluaran tiap harinya. Selain itu, dengan menggunakan klasifikasi data yang ada, maka dalam memberikan prediksi status keuangan ketika pengeluaran tidak terduga terjadi akan menggunakan metode Naïve Bayes. Disamping itu, untuk mempercepat pengembangannya, maka metode Scrum akan digunakan. Dengan metode Scrum, maka kerangka kerja dengan tujuan utama akan dibagi menjadi tujuan – tujuan kecil sehingga tujuan utama mudah dicapai.

2.2 Manajemen Keuangan

Manajemen Keuangan merupakan ilmu dan seni dalam pengelolaan keuangan (Gitman, 2002). Bentuk aplikasi dari manajemen keuangan ialah perencanaan serta pengendalian keuangan pada suatu individu atau keluarga.

Keuntungan akan didapatkan ketika kita memiliki perencanaan keuangan yang baik (Finansialku, 2015). Berikut ini merupakan keuntungan memiliki perencanaan keuangan, yaitu

1. Keuangan akan dikelola sesuai dengan panduan yang kita punya.
2. Tujuan – tujuan keuangan akan tercapai.
3. Kebiasaan tidak baik dalam mengelola keuangan serta utang konsumtif akan terhindar.
4. Lebih bijak dalam mengelola risiko.
5. Dalam memaksimalkan keuangan, kita lebih bebas dan tahu bagaimana caranya.
6. Kebebasan keuangan (*financial freedom*) lebih memungkinkan untuk dicapai.

2.3 Naïve Bayes

Naïve Bayes adalah metode klasifikasi yang berasal dari penerapan teorema Bayes dengan asumsi *independence* (saling bebas). Asumsi ini disebut *class conditional independence* yang memudahkan perhitungan – perhitungan ini dianggap “naive”. Naive merupakan asumsi bahwa kemunculan pada suatu *term* kata dalam suatu kalimat tidak dipengaruhi kemungkinan kata – kata lain dalam kalimat. Pada kenyataannya, kemungkinan kata dalam kalimat sangat dipengaruhi kemungkinan keberadaan kata – kata dalam kalimat (Destuardi & Sumpeno, 2009). Dengan menggunakan metode Naïve Bayes, penentuan estimasi parameter yang diperlukan hanya membutuhkan data pelatihan yang sedikit (Saleh, 2015). Berikut persamaan 2.1 yang merupakan persamaan metode Naïve Bayes (Saleh, 2015):

$$P(C|X) = \frac{P(X|C).P(C)}{P(X)} \quad (2.1)$$

Keterangan dari persamaan (2.1), yaitu:

X = Data dengan kelas yang belum diketahui

C = Hipotesis data merupakan kelas spesifik

$P(C|X)$ = Probabilitas hipotesis C berdasar kondisi X (probabilitas *posterior*)

$P(C)$ = Probabilitas hipotesis C (probabilitas *prior*)

$P(X|C)$ = Probabilitas X berdasarkan kondisi pada hipotesis C (probabilitas *likelihood*)

$P(X)$ = Probabilitas X (probabilitas *evidence*)

Persamaan 2.1 menjelaskan bahwa peluang masuknya sampel karakteristik tertentu dalam kelas C (*posterior*) adalah peluang munculnya kelas C (sebelum masuknya sampel tersebut, seringkali disebut *prior*) dikalikan dengan peluang kemunculan karakteristik – karakteristik sampel pada kelas C (*likelihood*) dan dibagi dengan peluang kemunculan karakteristik sampel secara *global* (*evidence*). Nilai *evidence* akan bernilai selalu tetap untuk setiap kelas pada satu sampel. Nilai *posterior* akan dibandingkan dengan nilai – nilai *posterior* pada kelas lain untuk menentukan pada kelas apa suatu sampel akan diklasifikasikan.

$$P(C) = \frac{s_i}{s} \quad (2.2)$$

Keterangan dari persamaan (2.2), yaitu:

$P(C)$ = Probabilitas hipotesis C (probabilitas *prior*)

s_i = Jumlah dari data *training* dari kategori C

s = Jumlah total data *training*

Persamaan 2.2 menjelaskan perhitungan untuk mencari nilai probabilitas *prior* (Nurani, Susanto, & Proboyekti, 2011). Dalam perhitungannya, s_i merupakan jumlah dari data *training* dari kategori C dan s merupakan jumlah total data *training*. Nilai dari s_i akan dibagi dengan nilai dari s sehingga nilai probabilitas *prior* ($P(C)$) dapat ditentukan.

$$P(X|C) = \prod_{k=1}^n P(x_k|C) \quad (2.3)$$

Keterangan dari persamaan (2.3), yaitu:

$P(X|C)$ = Probabilitas X berdasarkan kondisi pada hipotesis C (*likelihood*)

$P(x_k|C)$ = Nilai – nilai atribut dalam sampel X berdasarkan kondisi pada hipotesis C

Persamaan 2.3 menjelaskan perhitungan untuk mencari nilai *likelihood* (Nurani, Susanto, & Proboyekti, 2011). Nilai x merupakan nilai atribut dalam sampel X dan probabilitas $P(x|C)$, $P(x_1|C)$, ..., $P(x_n|C)$ dapat diperkirakan pada data *training*.

2.3.1 Pengujian Akurasi

Akurasi merupakan nilai hasil pengukuran kuantitas untuk nilai kebenaran (Andriyanto, Santoso, & Suprpto, 2018). Pengujian ini dilakukan untuk mengukur nilai akurasi dari algoritma Naïve Bayes terhadap rekomendasi yang akan diberikan. Nilai akurasi didapatkan dari hasil jumlah kasus uji yang benar dibagi dengan jumlah total prediksi dan dikalikan dengan 100%. Untuk lebih jelasnya dapat dilihat pada persamaan 2.4 (Andriyanto, Santoso, & Suprpto, 2018).

$$\text{Akurasi} = \frac{\text{Jumlah Prediksi Benar}}{\text{Jumlah Total Prediksi}} \times 100\% \quad (2.4)$$

2.4 Firebase

Firebase Realtime Database merupakan database yang di-host di cloud (Firebase, 2018). Data ini akan disimpan dalam bentuk JSON dan secara realtime klien yang terhubung akan tersinkronkan.

Firebase memiliki tiga kemampuan utama yang free (Firebase, 2018), yaitu

1. *Realtime*
Dengan menggunakan sinkronisasi data, setiap data yang berubah, maka *update* akan diterima oleh semua perangkat yang terhubung.
2. *Offline*
Dapat diakses dengan *offline* dan tetap responsif. Hal tersebut terjadi karena penyimpanan data dilakukan pada disk oleh SDK Firebase Realtime Database.
3. Dapat diakses dari perangkat klien.
Tidak diperlukan server aplikasi karena database ini dapat diakses secara langsung oleh perangkat seluler atau browser web.

Selain itu, pada pengembangan aplikasi penelitian ini akan menggunakan beberapa fitur yang disediakan oleh Firebase (Firebase, 2018), yaitu

1. *Firebase Authentication*
Firebase *authentication* digunakan untuk mengautentikasi pengguna sebelum masuk kedalam sistem. Autentikasi pengguna dilakukan dengan menggunakan *email* dan *password* yang sudah didaftarkan sebelumnya oleh pengguna kedalam sistem.
2. *Firebase Realtime Database*
Firebase *realtime database* digunakan untuk menyimpan dan menyinkronkan data pengguna pada aplikasi. Keunggulan menggunakan fitur ini adalah sinkronisasi data yang dapat dilakukan setiap kali data berubah dan dapat diakses dari perangkat lain, dan tidak memerlukan server.



Gambar 2. 1 Firebase Google

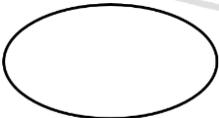
2.5 Unified Modelling Language (UML)

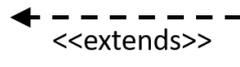
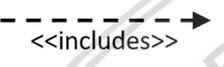
Unified Modelling Language (UML) merupakan gambaran untuk menjelaskan dan merancang suatu sistem perangkat lunak, terutama pada pendekatan berorientasi *object* (Fowler, 2003). Dalam beberapa jenis diagram yang ada pada UML, pengembangan sistem aplikasi *mobile* manajemen keuangan ini menggunakan tiga diagram yaitu, *class diagram*, *sequence diagram*, dan *use case diagram*.

2.5.1 Use Case Diagram

Use case diagram merupakan gambaran fungsionalitas dan fitur pada perangkat lunak dari sudut pandang aktor. Interaksi aktor dengan sistem merupakan gambaran yang dijelaskan pada *use case diagram* (Pressman, 2010).

Tabel 2. 1 Elemen *Use Case Diagram*

No	Elemen	Nama Elemen	Deskripsi
1		Actor	Interaksi orang atau sistem dengan sistem lainnya untuk mendapatkan nilai dari sistem. Jadi, <i>actor</i> merupakan <i>role</i> yang diperankan pengguna saat berinteraksi dengan sistem.
2		Use Case	<i>Use case</i> merupakan pemberian manfaat dalam beberapa hal bagi seorang aktor yang dilakukan oleh sistem. <i>Use case</i> juga diberi nama dengan frase kata kerja yang deskriptif.
3		Generalization	Menjelaskan hubungan yang bersifat umum dan bersifat lebih spesifik antara <i>use case</i>

			atau <i>actor</i> dengan <i>use case</i> atau <i>actor</i> lainnya.
4		<i>Association</i>	Menjelaskan hubungan komunikasi dua arah antara <i>use case</i> dengan <i>actor</i> .
5		<i>Extend</i>	Menjelaskan hanya kondisi keadaan tertentu akan dieksekusi. Anak panah menuju ke <i>base use case</i>
6		<i>Include</i>	Menjelaskan hubungan fungsionalitas antara sebuah <i>use case</i> dengan <i>use case</i> lainnya.

(Dennis, Wixom, & Roth, 2012)

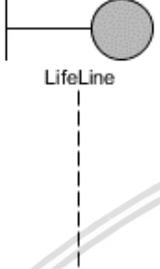
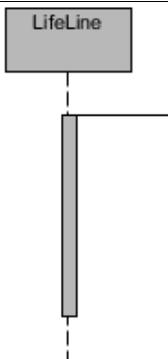
2.5.2 Sequence Diagram

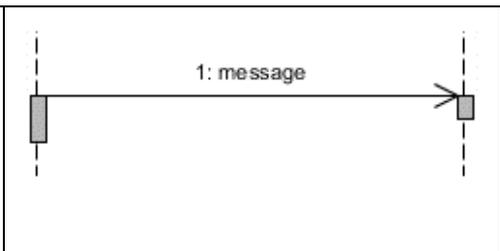
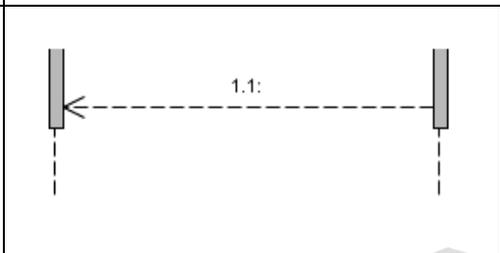
Sequence diagram adalah penjelasan mengenai interaksi antar objek (Wiley & Sons, 2005). Selama perancangan subsistem, diagram ini digunakan. *Sequence diagram* lebih menggambarkan fokus waktu dan urutan interaksi secara visual dengan menggunakan sumbu vertikal diagram untuk mewakili waktu pesan apa yang dikirim dan kapan pada suatu objek (Visual Paradigm, 2018).

Beberapa elemen pada *sequence diagram* dijelaskan pada Tabel 2.2.

Tabel 2. 2 Elemen Sequence Diagram

No	Elemen	Nama Elemen	Deskripsi
1		<i>Actor</i>	Merupakan gambaran yang berinteraksi dengan sistem. <i>Actor</i> tidak selamanya adalah orang, namun

			sistem juga bisa dikatakan sebagai <i>actor</i> .
2		<p><i>Life Line</i> <<Boundary>></p>	Merupakan gambaran interaksi antara <i>actor</i> dengan sistem.
3		<p><i>Life Line</i> <<Control>></p>	Merupakan gambaran penghubung antara <i>boundary</i> dengan tabel.
4		<p><i>Life Line</i> <<Entity>></p>	Merupakan gambaran informasi yang harus disimpan oleh sistem.
5		<p><i>Activations</i></p>	Sebuah persegi panjang yang mendklarasikan periode sebuah elemen melakukan operasi.

6		<p><i>Call Message</i></p>	<p>Sejenis pesan yang merupakan seruan pengoperasian pada suatu <i>life line</i>.</p>
7		<p><i>Return Message</i></p>	<p>Sejenis jawaban pesan untuk pemberi pesan yang sesuai.</p>

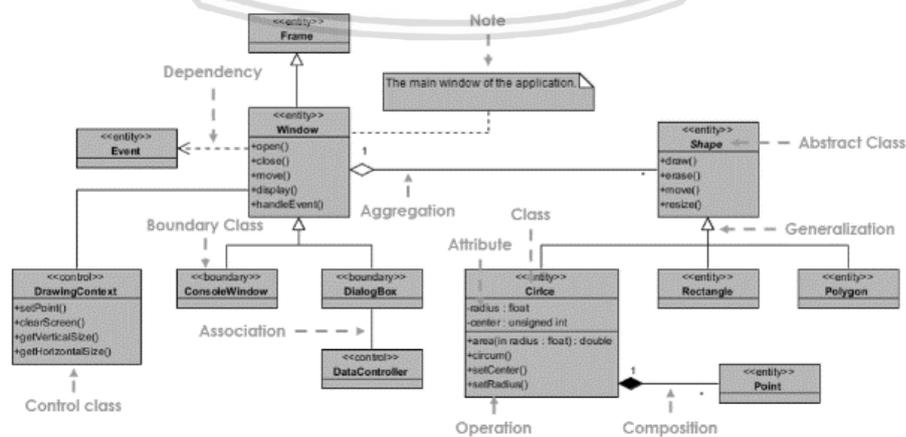
(Visual Paradigm, 2018)

2.5.3 Class Diagram

Class diagram merupakan model struktur statis pada sebuah sistem yang menjelaskan elemen – elemen sistem dan hubungan diantara keduanya (Seidl & dkk, 2015). Dengan adanya *class diagram*, pengembang dapat terbantu elemen – elemen mana saja yang berhubungan sehingga jika melakukan perubahan dapat mempermudah pelaksanaannya. Penjelasan mengenai dasar tentang *class diagram* dapat dilihat pada gambar 2.2.

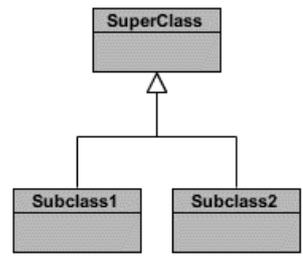
Class diagram juga dapat memiliki satu atau lebih hubungan dengan klas lainnya. Untuk lebih jelasnya mengenai hubungan antar klas dapat dilihat sebagai berikut,

1. *Inheritance (Generalization)* merupakan hubungan antar klas yang merepresentasikan hubungan “*is – a*”. Hubungan antar klas dilambangkan dengan garis panah kosong yang mengarah dari anak ke klas induk. Pada gambar 2.3 merupakan contoh dari *Inheritance*, dimana SubClass1 dan SubClass2 adalah spesialisasi *Super Class*.



(Visual Paradigm, 2018)

Gambar 2. 2 Class Diagram



(Visual Paradigm, 2018)

Gambar 2.3 Inheritance (Generalization)

2. *Association* merupakan hubungan antar klas yang umum dan dilambangkan dengan garis lurus yang menghubungkan antara dua klas. Untuk lebih jelasnya dapat dilihat pada gambar 2.4.



(Visual Paradigm, 2018)

Gambar 2.4 Association

3. *Aggregation* merupakan hubungan antar klas yang merepresentasikan hubungan "part of". Hubungan antar klas dilambangkan garis dengan ujung garis berbentuk *diamond* yang tidak terisi yang terhubung ke klas komposit. Pada gambar 2.5 merupakan contoh dari *Aggregation*, dimana Class2 adalah bagian dari Class1.



(Visual Paradigm, 2018)

Gambar 2.5 Aggregation

4. *Composition* merupakan hubungan antar klas yang saling berkaitan, dimana antar klas tidak bisa berdiri sendiri dan harus merupakan bagian dari klas lain. Hubungan antar klas dilambangkan dengan garis berbentuk *diamond* yang terisi/solid yang terhubung dengan klas komposit. Untuk lebih jelasnya dapat dilihat pada gambar 2.6.

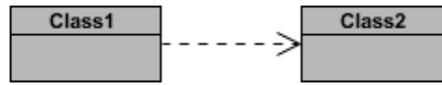


(Visual Paradigm, 2018)

Gambar 2.6 Composition

5. *Dependency* merupakan hubungan antar klas yang jika terdapat perubahan pada definisi satu dapat menyebabkan perubhana pada yang lain, namun tidak sebaliknya. Hubungan antar klas dilambangkan dengan garis putus –

putus dengan panah terbuka. Pada gambar 2.7 merupakan contoh *Dependency*, dimana Class1 tergantung (*depends*) pada Class2.



(Visual Paradigm, 2018)

Gambar 2. 7 Dependency

2.6 Agile Development Method

Agile development method merupakan salah satu dari *software development life cycle*. *Agile* berfokus pada pengembangan *incremental* dan *iterative* dimana kebutuhan, perancangan, implementasi, dan pengujian dilakukan selama proyek berlangsung (Szalvay, 2004).

2.6.1 Scrum

Scrum merupakan salah satu model dari *agile development method*. Dengan menggunakan scrum, kita dikenalkan dengan konsep proses kontrol untuk mengelola perubahan proyek perangkat lunak. Scrum menjelaskan bahwa proses pendefinisian langsung tidak dapat digunakan secara efektif dalam mengelola proyek perangkat lunak yang kompleks dan dinamis. Dari percobaan yang telah dilakukan masa lalu menyatakan bahwa tidak mungkin ada suatu perpustakaan yang memiliki pengetahuan lengkap untuk menangani setiap situasi yang mungkin bisa muncul pada proyek perangkat lunak. Oleh karena itu, dalam proses scrum, rencana proyek terus diperiksa dan disesuaikan berdasarkan kenyataan empiris proyek (Szalvay, 2004).

Pada Scrum terdapat tahapan yang dilakukan selama proses melakukan pengembangan (Schwaber & Sutherland, *The Definitive Guide to Scrum: The Rules of the Game*, 2016), yaitu

1. *Product Backlog*

Product backlog merupakan daftar urut yang memungkinkan dibutuhkan di dalam produk dan juga merupakan sumber utama dari daftar kebutuhan mengenai semua hal yang dilakukan terhadap produk. *Produk backlog* menjabarkan semua fitur, fungsi, kebutuhan, penyempurnaan, dan perbaikan terhadap produk dirilis mendatang. *Product backlog* akan dibagi lagi pengerjaannya menjadi beberapa *product backlog item*. *Product backlog item* memiliki atribut deskripsi serta urutan pengerjaan.

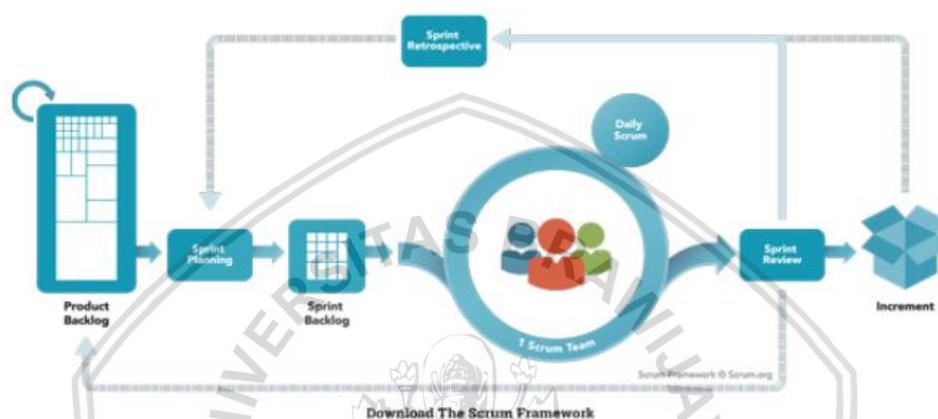
2. *Sprint Planning*

Sprint planning merupakan tahapan perencanaan daftar pekerjaan yang akan dilaksanakan di dalam *sprint*. *Sprint planning* terdiri dari dua bagian yaitu, *sprint goal* dan *sprint backlog*. *Sprint goal* merupakan sekumpulan tujuan yang akan dicapai dalam satu *sprint* sepanjang pengimplmentasian *product backlog*. *Sprint backlog* merupakan sekumpulan *product backlog item* yang telah dipilih untuk dikerjakan di *sprint* untuk mencapai *sprint goal*.



3. *Sprint*

Sprint merupakan tahapan putaran atau iterasi yang dilakukan secara berulang – ulang hingga mencapai waktu yang telah ditentukan atau sistem telah selesai. Didalam satu *sprint* terdapat *daily scrum*, *development*, dan *sprint review*. *Daily Scrum* merupakan kegiatan yang dilakukan untuk evaluasi apa saja yang telah dikerjakan, hambatan yang ada, dan kesesuaian target penyelesaian. *Development* merupakan proses pengembangan yang bertujuan untuk menghasilkan perancangan dan implementasi produk. *Sprint review* merupakan evaluasi akhir dari *sprint* yang telah selesai dilakukan dan menghasilkan *product backlog item* untuk *sprint* berikutnya.



(Schwaber & Sutherland, The Definitive Guide to Scrum: The Rules of the Game, 2016)

Gambar 2. 8 Scrum Sprint Life Cycle

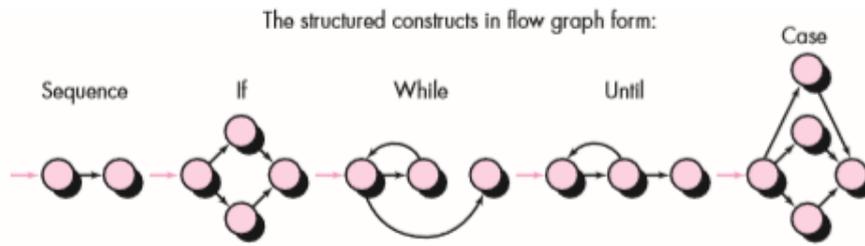
2.7 Pengujian Validasi

Pengujian validasi akan dilakukan dengan teknik *black – box testing*. *Black – box testing* merupakan suatu pengujian yang berfokus pada kebutuhan perangkat lunak (Pressman, 2010). *Black – box testing* memungkinkan suatu pengujian dilakukan dengan menjalankan seluruh kebutuhan fungsional yang ada pada suatu program. Dengan melakukan *black – box testing*, diharapkan dapat ditemukannya kesalahan dalam suatu fungsi yang salah atau belum diimplementasikan, kesalahan antarmuka, kesalahan dalam struktur data atau akses *database* eksternal, kesalahan kinerja, hingga kesalahan pada inisialisasi maupun terminasi.

2.8 Pengujian Unit

Pengujian Unit akan dilakukan dengan teknik *white – box testing*. *White – box testing* merupakan suatu pengujian yang didasarkan pada pengetahuan komponen dan struktur program (Pressman, 2010). Pencarian jalur independen pada kode program menjadi dasar pada pengeujian *white – box testing*. Dengan membuat perancangan *flow graph*, kita dapat membuat jalur independe dari kode program. Diagram yang menggambarkan alur program itulah yang disebut dengan *flow graph*. *Flow graph* terdiri dari *node* (simpul) yang direpresentasikan dalam

bulatan berisi data yang diberi angka dan angka tersebut merupakan urutan alur program serta *edge* yaitu penghubung antar *node*.



(Pressman, 2010)

Gambar 2. 9 Contoh Flow Graph

2.9 Pengujian Usability

Pengujian *Usability* dilakukan untuk mengetahui tingkat kemudahan penggunaan terhadap sistem yang telah diimplementasikan dan diuji secara langsung kepada sejumlah responden. Setelah sistem siap digunakan, pengujian ini dilakukan. Dalam pengujian *usability* akan digunakan metode *System Usability Scale* (SUS).

2.9.1 System Usability Scale

System Usability Scale (SUS) merupakan suatu pengujian dengan menyediakan alat ukur yang “*quick and dirty*” dan dapat diandalkan yang didasari dari sisi pengguna (Sauro, 2011). Metode SUS menggunakan 10 pertanyaan berbentuk kuisioner dengan 5 opsi jawaban untuk setiap pertanyaannya dan untuk lebih jelasnya dapat dilihat pada tabel 2.3 (Pudjoatmodjo & Wijaya, 2016). Lima opsi ini terdiri dari angka 1 hingga 5, dimana semakin kecil nilainya menyatakan Sangat Tidak Setuju dan jika nilainya semakin besar, maka menyatakan Sangat Setuju.

Tabel 2. 3 Lembar Kuesioner System Usability Scale (SUS)

No	Pertanyaan	Sangat Tidak Setuju					Sangat Setuju
		1	2	3	4	5	
1	Saya pikir bahwa saya akan ingin lebih sering menggunakan aplikasi ini						
2	Saya menemukan bahwa aplikasi ini, tidak harus dibuat serumit ini						
3	Saya pikir aplikasi ini mudah untuk digunakan						

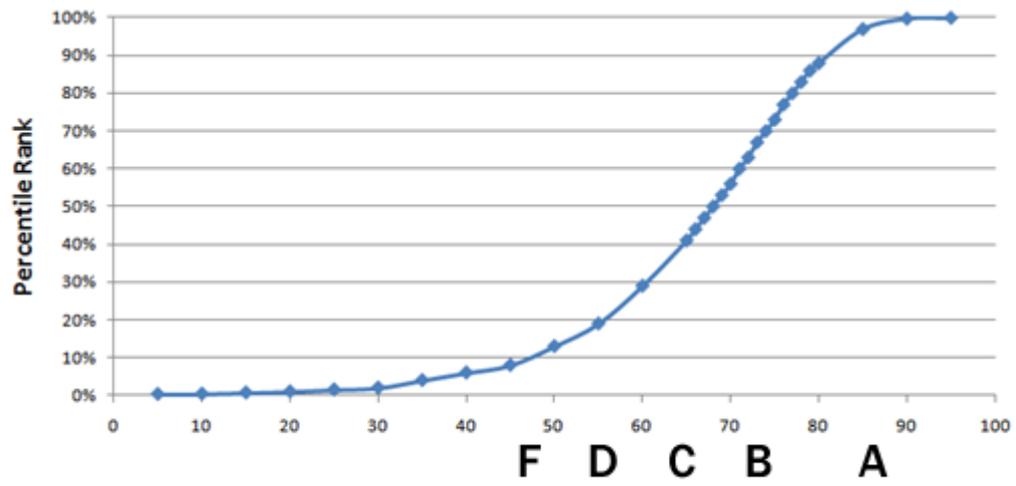
4	Saya pikir bahwa saya akan membutuhkan bantuan dari orang teknis untuk dapat menggunakan aplikasi ini					
5	Saya menemukan berbagai fungsi di aplikasi ini diintegrasikan dengan baik					
6	Saya pikir ada terlalu banyak ketidaksesuaian dalam aplikasi ini					
7	Saya bayangkan bahwa kebanyakan orang akan mudah untuk mempelajari aplikasi ini dengan sangat cepat					
8	Saya menemukan, aplikasi ini sangat rumit untuk digunakan					
9	Saya merasa sangat percaya diri untuk menggunakan aplikasi ini					
10	Saya perlu belajar banyak hal sebelum saya bisa memulai menggunakan aplikasi ini					

Untuk mendapatkan nilai SUS dibutuhkan beberapa perhitungan (Sauro, 2011), yaitu:

1. Untuk item ganjil : kurangi satu dari respons pengguna.
2. Untuk butir bernomor genap : kurangi tanggapan pengguna dari 5.
3. Ini mengukur semua nilai dari 0 hingga 4 (dengan empat merupakan respons yang paling positif).
4. Tambahkan tanggapan yang dikonversi untuk setiap pengguna dan kalikan total dengan 2,5. Ini mengubah rentang nilai yang mungkin dari 0 hingga 100 bukannya dari 0 hingga 40.

Setelah hasil perhitungan didapat, untuk mendapatkan nilai minimal SUS yang dapat diterima pengguna adalah 68 (Sauro, 2011). Selain itu, untuk mendapatkan hasil terbaik pengujian SUS dilakukan oleh 5 orang (Nielsen, Articles: Why You Only Need to Test with 5 Users, 2000). Oleh karena itu, pengujian SUS pada sistem ini akan dilakukan oleh 5 orang.

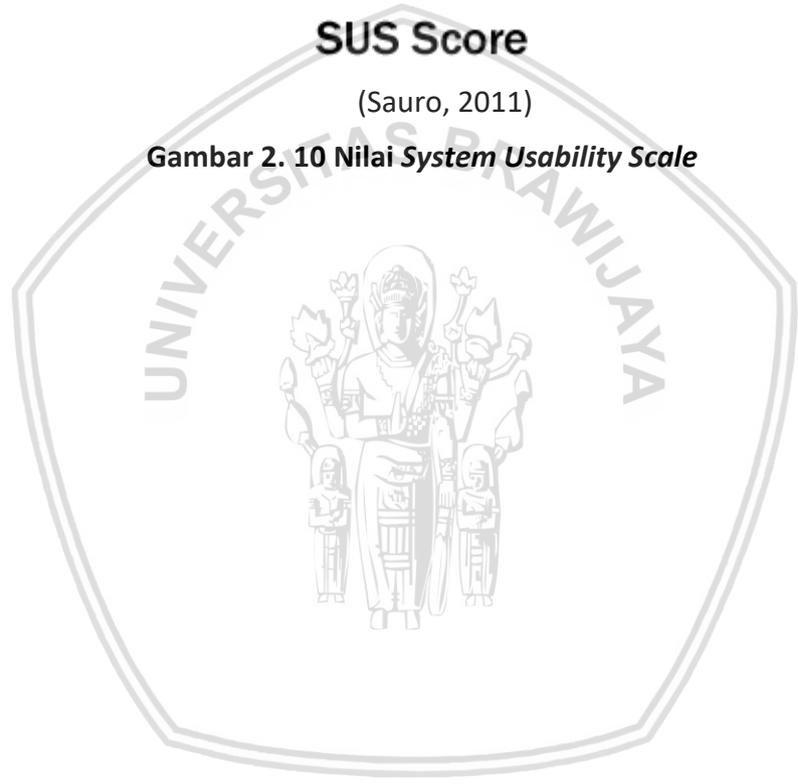




SUS Score

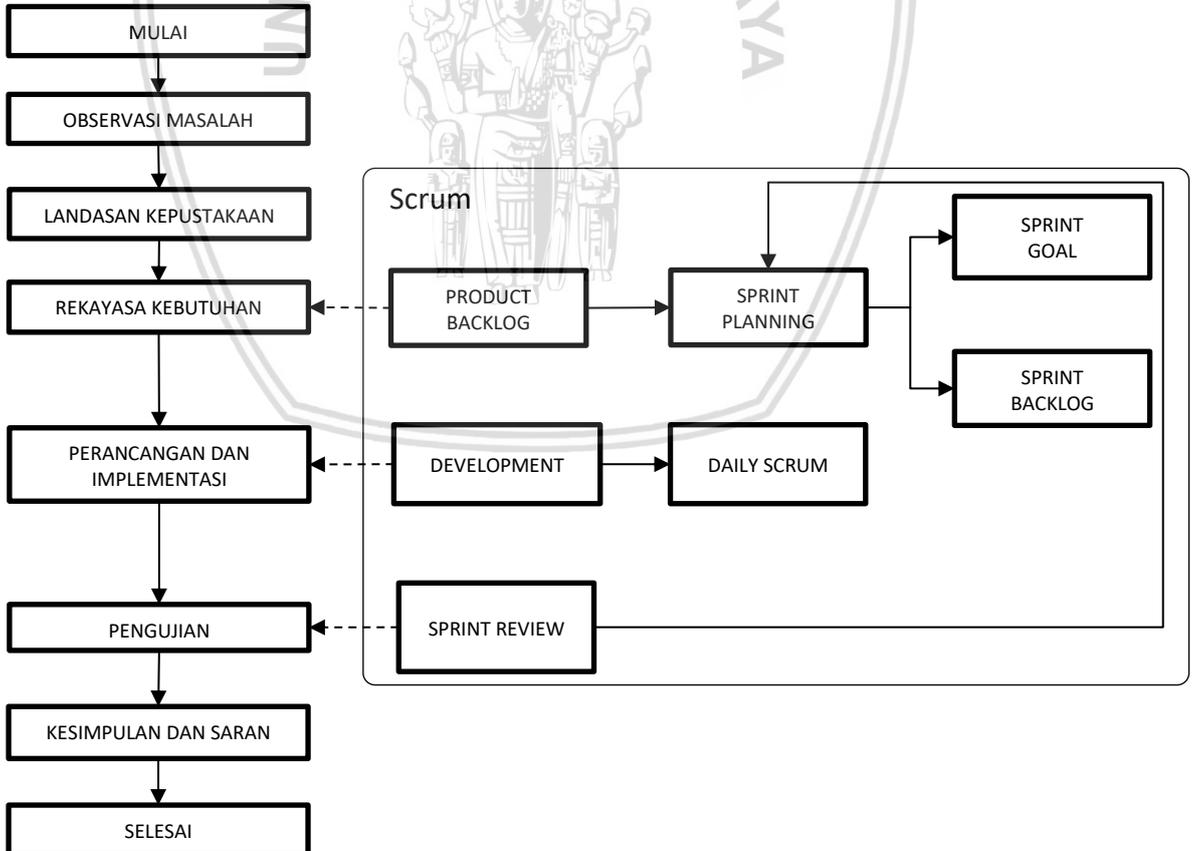
(Sauro, 2011)

Gambar 2. 10 Nilai *System Usability Scale*



BAB 3 METODOLOGI

Pada bab ini akan membahas langkah – langkah yang diperlukan untuk melakukan pengembangan aplikasi *mobile* manajemen keuangan dengan metode Scrum. Langkah pertama yang dilakukan adalah menentukan landasan kepustakaan agar penelitian ini tetap pada tujuannya. Setelah menentukan landasan kepustakaan, maka dilakukan rekayasa kebutuhan. Pada tahap rekayasa kebutuhan terdapat *product backlog* yang menjadi daftar kebutuhan sistem yang akan direncanakan pengerjaannya pada *sprint planning* dan pada *sprint planning* akan dibagi lagi menjadi *sprint goal* dan *sprint backlog*. Setelah itu, akan dilakukan *sprint* yang terdiri dari perancangan dan implemmentasi dan pengujian. Pada perancangan dan implementasi terdapat fase *development* yang didalamnya terdapat *daily scrum*, yaitu menjelaskan apa saja yang sudah dilakukan pada satu kali *sprint*. Setelah perancangan dan implementasi dilakukan, tahap selanjutnya adalah pengujian. Pada tahap pengujian, didalamnya terdapat *sprint review*, yaitu melakukan pengujian setiap satu kali *sprint* dan evaluasi dari hasil satu *sprint*. Setelah *sprint review*, maka dilakukan pengecekan kembali apakah masih ada *product backlog* yang belum dikejerjakan. Jika masih ada, maka akan dilakukan *sprint* lagi dan jika tidak ada, maka dilakukan penarikan kesimpulan dan saran terkait penelitian ini.



Gambar 3. 1 Diagram Alir Metodologi Penelitian



3.1 Observasi Masalah

Observasi masalah merupakan tahapan yang dilakukan sebelum dilakukannya penelitian. Tahapan ini dilakukan untuk menggali permasalahan lebih dalam lagi yang nantinya akan diangkat sebagai topik permasalahan. Observasi masalah ini dilakukan dengan melihat aplikasi manajemen pada *playstore*, wawancara, dan penyebaran kuesioner pada mahasiswa FILKOM UB.

3.2 Landasan Kepustakaan

Landasan Kepustakaan merupakan kumpulan beberapa sumber informasi untuk menunjang penelitian yang akan dilakukan. Sumber – sumber landasan kepastakaan didapat dari buku, jurnal, *e-books*, maupun penelitian orang lain. Rincian dari landasan kepastakaan yang berkaitan dengan penelitian ini, ialah:

1. Kajian Pustaka
2. Manajemen Keuangan
3. Naïve Bayes
4. Firebase
5. *Unified Modelling Language* (UML)
 - a. *Use Case Diagram*
 - b. *Sequence Diagram*
 - c. *Class Diagram*
6. *Agile Development Method*
 - a. Scrum
7. Pengujian Validasi
8. Pengujian Unit
9. Pengujian *Usability*
 - a. *System Usability Scale (SUS)*

3.3 Rekayasa Kebutuhan

Rekayasa kebutuhan merupakan tahapan yang dilakukan untuk menggali kebutuhan apa saja yang dibutuhkan untuk pengembangan sistem. Dalam menggali kebutuhan ini maka dilakukan penyebaran lembar kuesioner ke mahasiswa FILKOM UB. Hasil dari lembar kuesioner tersebut maka akan jadi daftar kebutuhan yang pada penelitian ini akan menjadi *product backlog*. Setelah mendapatkan kebutuhan, maka dibuat *use case diagram* dan *use case scenario* untuk menjelaskan interaksi aktor dengan sistem yang akan dibuat. Karena penelitian ini menggunakan metode Scrum, maka pada tahap ini akan dibagi lagi dalam tahapan *sprint planning* yang didalamnya terdapat dua tahapan lagi, yaitu *sprint backlog* dan *sprint goal*.

3.3.1 Product Backlog

Pada tahap ini, dari daftar kebutuhan yang didapat akan dijadikan menjadi *product backlog*. Dari daftar kebutuhan yang didapat akan diurutkan berdasarkan prioritas pengerjaanya pada *product backlog*. *Product backlog* yang sudah diurutkan, maka akan diperkecil lagi menjadi beberapa sub bagian yang disebut

sebagai *product backlog item*. Kumpulan *product backlog item* itulah yang nantinya akan dikerjakan pada setiap *sprint*.

3.3.2 Sprint Planning

Sprint planning merupakan kegiatan yang dilakukan untuk merumuskan tugas – tugas apa saja yang akan dikerjakan pada penelitian ini dalam satu *sprint*.

3.3.2.1 Sprint Goal

Sprint goal adalah tujuan yang ditetapkan untuk penelitian ini dalam satu *sprint* yang akan dilakukan. *Sprint goal* ini dibuat agar dalam pengerjaan satu *sprint* lebih terarah.

3.3.2.2 Sprint Backlog

Sprint Backlog adalah sebagian dari *product backlog* yang telah disepakati untuk dikerjakan dalam satu *sprint* beserta daftar cara pengerjaannya. *Sprint backlog* dapat berupa *to-do list* yang sederhana.

3.4 Perancangan dan Implementasi

Perancangan dan Implementasi adalah tahapan yang dilakukan untuk menggambarkan perancangan sistem dan melakukan implementasi kode berdasarkan hasil dari pemodelan *use case diagram* dan *use case scenario*. Penggambaran perancangan sistem akan dibuat dengan *Unified Modelling Language* (UML). UML yang digunakan pada perancangan sistem adalah *class diagram* dan *sequence diagram*. *Class diagram* digunakan untuk menggambarkan hubungan antar kelas pada sistem yang akan dikembangkan, sedangkan *sequence diagram* digunakan untuk menjelaskan alur proses interaksi objek berdasarkan urutan waktu. Karena penelitian ini menggunakan metode Scrum, perancangan dan implementasi masuk kedalam fase *development* yang didalamnya terdapat *daily scrum*.

3.4.1 Development

Development merupakan tahapan yang bertujuan untuk menghasilkan perancangan dan implementasi sistem. Pada tahapan perancangan dan implementasi akan dibagi menjadi beberapa tahap, yaitu perancangan arsitektural, perancangan basis data, perancangan antarmuka, implementasi arsitektural, implementasi antarmuka, dan implementasi *database*. Untuk lebih jelasnya akan dijelaskan sebagai berikut,

3.4.1.1 Perancangan Arsitektural

Perancangan arsitektural adalah perancangan sistem dengan membuat class diagram dan sequence diagram.

3.4.1.2 Perancangan Komponen

Perancangan komponen adalah perancangan sistem dengan membuat gambaran rincian dari sub – *system* komponen perangkat lunak. Perencanaan komponen ini terdiri dari rincian algoritma pada suatu komponen.

3.4.1.3 Perancangan Basis Data

Perancangan basis data adalah perancangan sistem dengan membuat skema JSON.

3.4.1.4 Perancangan Antarmuka

Perancangan antarmuka adalah perancangan sistem dengan membuat *screenflow diagram* dan *wireframe*.

3.4.1.5 Implementasi Arsitektural

Implementasi arsitektural merupakan implementasi sistem dengan menggunakan bahasa Java dengan *tools* Android Studio sebagai *back – end*.

3.4.1.6 Implementasi Basis Data

Implementasi basis data merupakan implementasi *database* sistem dengan menggunakan Firebase.

3.4.1.7 Implementasi Antarmuka

Implementasi antarmuka merupakan implementasi untuk membuat *interface* aplikasi FM App. Pembuatan tampilan sistem akan menggunakan bahasa XML.

3.4.1.8 Daily Scrum

Daily Scrum merupakan kegiatan yang dilakukan untuk evaluasi apa saja yang telah dikerjakan, hambatan yang ada, dan kesesuaian target penyelesaian.

3.5 Pengujian

Setelah implementasi selesai dilakukan, pengujian terhadap sistem yang telah dibuat dan pengecekan kesesuaian fitur dengan daftar kebutuhan yang sudah didapat pada rekayasa kebutuhan dilakukan pada tahap ini. Karena penelitian ini menggunakan metode Scrum, maka setelah satu *sprint* selesai dikerjakan akan dilakukan *sprint review*. Pada metode Scrum, pengujian termasuk pada fase *sprint review*. Untuk lebih jelasnya dapat dilihat pada sub bab berikut.

3.5.1 Pengujian Unit

Pengujian unit menggunakan metode *white box testing*. *White box testing* merupakan proses pengujian untuk mengetahui algoritma dengan kompleksitas tertinggi yang digunakan pada saat implementasi masih berada pada kategori *low risk program* atau tidak. Pengujian ini akan dilakukan beberapa tahap sebagai berikut,

- (a) Pembuatan *Flow Graph*
- (b) Pencarian nilai kompleksitas (*VG*)
- (c) Analisis hasil uji

3.5.2 Pengujian Algoritma Naïve Bayes

Pengujian algoritma Naïve Bayes akan dilakukan dengan mencoba kemungkinan yang akan terjadi pada sistem. Pengujian digunakan untuk

menentukan hasil rekomendasi dari aplikasi apakah dapat diterima atau tidak terhadap pengguna. Untuk menentukan besarnya tingkat diterimanya rekomendasi, maka akan dilakukan perhitungan nilai akurasi pada hasil rekomendasi tersebut.

3.5.3 *Sprint Review*

Sprint review merupakan kegiatan pengecekan apakah implementasi sudah sesuai atau tidak dengan *item – item* yang ada pada *product backlog*. Hal ini dilakukan setelah satu kali pengerjaan *sprint*. *Sprint review* ini akan melakukan proses validasi *Product Backlog* pada sistem yang telah dibuat.

3.5.3.1 Pengujian Validasi

Pengujian validasi menggunakan metode *black box testing*. *Black box testing* merupakan proses pengujian untuk mengetahui sistem yang telah dibuat sudah memenuhi kebutuhan atau tidak. Pengujian ini akan dilakukan beberapa tahap sebagai berikut,

- (a) Pengecekan hasil dari implementasi ke daftar kebutuhan
- (b) Analisis hasil uji

3.5.4 Pengujian *Usability*

Pengujian *usability* menggunakan metode *System Usability Scale (SUS)*. *System Usability Scale (SUS)* merupakan suatu pengujian dengan menyediakan alat ukur dan dapat diandalkan yang didasari dari sisi pengguna. Pengujian ini akan dilakukan dengan menggunakan 10 pertanyaan berbentuk kuisisioner dengan 5 opsi jawaban untuk setiap pertanyaannya. Lima opsi ini terdiri dari angka 1 hingga 5, dimana semakin kecil nilainya menyatakan sangat tidak setuju dan jika nilainya semakin besar, maka menyatakan sangat setuju. Pengujian ini akan dilakukan oleh 5 orang pengguna.

3.6 Kesimpulan dan Saran

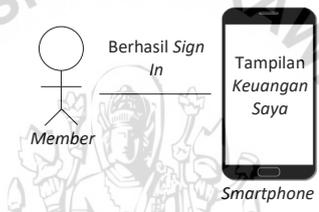
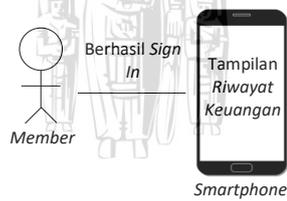
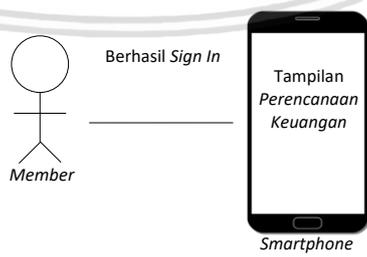
Setelah *sprint* sudah selesai dilakukan, maka akan dilakukan tahap kesimpulan dan saran. Kesimpulan pada penelitian ini didapat dari seluruh kegiatan dalam *sprint* hingga *sprint* selesai dilakukan dan menjawab pertanyaan pada rumusan masalah. Saran diberikan untuk pengembangan selanjutnya pada penelitian ini, sehingga dapat menjadi bahan pertimbangan serta penyempurnaan penelitian yang dilakukan.

BAB 4 REKAYASA KEBUTUHAN

4.1 Gambaran Umum Sistem

Financial Management App (FM App) merupakan aplikasi manajemen keuangan pribadi yang ditujukan untuk mahasiswa FILKOM Universitas Brawijaya. FM App bertujuan sebagai media pencatat dan pengaturan keuangan pribadi. FM App lebih ditujukan kepada pengguna yang membutuhkan untuk menghemat pengeluaran harian dan merencanakan suatu pembelian barang. FM App dapat diakses dimana saja dengan *handphone* Android yang terkoneksi dengan internet. Untuk lebih jelasnya mengenai gambaran umum pada FM App dapat dilihat pada tabel 4.1 berikut.

Tabel 4. 1 Gambaran Umum FM App

Kasus	Solusi
Mencatat pemasukan, mencatat pengeluaran, mengalami pengeluaran tidak dianggarkan/tidak terduga	
Melihat riwayat keuangan	
Perencanaan anggaran keuangan, membatasi pengeluaran, pembelian suatu barang dengan kemampuan uang yang dimiliki	

Pada tabel 4.1 menjelaskan mengenai gambaran umum dari FM App. Ketika pengguna menginginkan untuk mencatat pemasukan, mencatat pengeluaran, mengatasi pengeluaran yang tidak dianggarkan atau tidak terduga, maka dapat mengatasinya dengan menjadi *member* pada FM App dan memasuki halaman keuangan saya. Selain itu, jika pengguna menginginkan melihat riwayat keuangannya, maka dapat mengatasinya dengan menjadi mem pada FM App dan

memasukan halaman Riwayat Keuangan. Selain itu juga, apabila pengguna menginginkan merencanakan anggaran keuangan, membuat batas pengeluaran harian, dan pembelian terhadap sesuatu barang, maka dapat mengatasinya dengan menjadi member pada FM App dan membuka halaman perencanaan keuangan.

4.2 Identifikasi Aktor

Identifikasi Aktor bertujuan untuk memberikan penjelasan siapa saja yang berperan pada FM App. Pada tabel 4.2 akan dijelaskan aktor yang terkait dan deskripsi dari aktor tersebut.

Tabel 4. 2 Identifikasi Aktor

Aktor	Deskripsi
Pengguna	Pengguna merupakan seluruh orang yang menggunakan aplikasi FM App. Pengguna dapat mengakses Halaman Utama yang berisi <i>Sign In</i> dan <i>Sign Up</i> .
Member	Member merupakan pengguna yang sudah terdaftar pada <i>database</i> pengguna aplikasi FM App. Member dapat mengakses Halaman Keuangan Saya, Menambah Pengeluaran, Menambah Pemasukan, Halaman Riwayat Keuangan, Halaman Perencanaan Keuangan, Membuat Perencanaan Anggaran, Membuat Batas Harian, Membuat Perencanaan Pembelian dan Halaman Profil.

4.3 Daftar Kebutuhan

Daftar kebutuhan merupakan penjelasan mengenai kebutuhan apa saja yang dibutuhkan pada aplikasi. Pada tahap rekayasa kebutuhan menghasilkan beberapa kebutuhan yang dibutuhkan untuk pengembangan aplikasi. Untuk penjabaran mengenai daftar kebutuhan akan dibagi menjadi tiga, yaitu *product backlog*, kebutuhan fungsional dan kebutuhan non – fungsional.

4.3.1 Product Backlog

Dari definisi kebutuhan yang didapat, maka dikelompokan menjadi 11 *product backlog* dan setiap *product backlog* memiliki beberapa *product backlog item*. Setiap *product backlog* akan diberikan kode PB_X dan *product backlog item* akan diberikan kode PBI_PBX_Y. PB merupakan *product backlog*, X merupakan nomor dari *product backlog*, PBI merupakan *product backlog item*, dan Y merupakan nomor dari *product backlog item*. Untuk lebih jelasnya dapat dilihat pada tabel 4.3 berikut.

Tabel 4. 3 Daftar *Product Backlog*

No	Kode <i>Product Backlog</i>	<i>Product Backlog Item</i>
1	PB_01	<ol style="list-style-type: none"> 1. Sistem harus mampu menampilkan halaman <i>sign up</i> untuk pengguna (PBI_PB1_01) 2. Sistem harus mampu menyediakan kolom untuk memasukan nama, <i>email</i>, dan <i>password</i> (PBI_PB1_02) 3. Sistem harus mampu menampilkan peringatan ketika kolom nama, <i>email</i>, dan <i>password</i> kosong (PBI_PB1_03) 4. Sistem harus mampu menyediakan tombol <i>sign up</i> untuk mendaftarkan pengguna sebagai <i>member</i> (PBI_PB1_04) 5. Sistem harus mampu menyimpan data pengguna yang sudah terdaftar ke dalam <i>database</i> (PBI_PB1_05)
2	PB_02	<ol style="list-style-type: none"> 1. Sistem harus mampu menampilkan halaman <i>sign in</i> untuk pengguna (PBI_PB2_01) 2. Sistem harus mampu menyediakan kolom untuk memasukan <i>email</i> dan <i>password</i> (PBI_PB2_02) 3. Sistem harus mampu menampilkan peringatan ketika kolom <i>email</i> dan <i>password</i> kosong (PBI_PB2_03) 4. Sistem harus mampu menyediakan tombol Sign In untuk autentifikasi <i>email</i> dan <i>password</i> pengguna (PBI_PB2_04) 5. Sistem harus mampu menyediakan fungsi <i>forgot password</i> apabila pengguna lupa <i>password</i> dari <i>email</i> yang sudah terdaftar (PBI_PB2_05) 6. Sistem harus mampu menyediakan <i>pop up</i> untuk <i>forgot password</i> yang menyediakan kolom untuk memasukan <i>email</i> dan tombol <i>submit</i> untuk mengirimkan perubahan <i>password</i> ke <i>email</i> yang terdaftar (PBI_PB2_06)
3	PB_03	<ol style="list-style-type: none"> 1. Sistem harus mampu menampilkan halaman keuangan saya untuk <i>member</i> (PBI_PB3_01) 2. Sistem harus mampu menampilkan tanggal untuk memberitahu <i>member</i> sekarang tanggal berapa (PBI_PB3_02) 3. Sistem harus mampu menampilkan total uang yang dimiliki <i>member</i> sekarang (PBI_PB3_03)

		<ol style="list-style-type: none"> 4. Sistem harus mampu menyediakan fungsi untuk menampilkan total uang yang dimiliki <i>member</i> sekarang (PBI_PB3_04) 5. Sistem harus mampu menampilkan diagram berdasarkan kategori anggaran untuk mengetahui pengeluaran tiap kategori anggaran yang sudah dilakukan (PBI_PB3_05) 6. Sistem harus mampu menyediakan fungsi untuk menampilkan diagram berdasarkan kategori anggaran untuk mengetahui pengeluaran tiap kategori anggaran yang sudah dilakukan (PBI_PB3_06) 7. Sistem harus mampu menampilkan sisa dari batas harian untuk mengetahui sisa pengeluaran yang dapat dikeluarkan berapa (PBI_PB3_07) 8. Sistem harus mampu menyediakan fungsi untuk menampilkan sisa dari batas harian untuk mengetahui sisa pengeluaran yang dapat dikeluarkan berapa (PBI_PB3_08)
4	PB_04	<ol style="list-style-type: none"> 1. Sistem harus mampu menampilkan halaman menambah pengeluaran untuk <i>member</i> (PBI_PB4_01) 2. Sistem harus mampu menampilkan tanggal untuk memberitahu <i>member</i> sekarang tanggal berapa (PBI_PB4_02) 3. Sistem harus mampu menyediakan kategori pilihan anggaran untuk pengeluaran yang akan dilakukan (PBI_PB4_03) 4. Sistem harus mampu menyediakan kolom untuk memasukan jumlah pengeluaran dan <i>note</i> terhadap pengeluaran yang akan dikeluarkan (PBI_PB4_04) 5. Sistem harus mampu menyediakan tombol <i>submit</i> untuk menyimpan data pengeluaran ke <i>database</i> (PBI_PB4_05) 6. Sistem harus mampu menyediakan halaman perencanaan alokasi dana tidak dianggarkan yang terdiri dari perencanaan alokasi dana pada semua kategori dan status keuangan setelah perencanaan alokasi dana digunakan ketika kategori tidak dianggarkan dipilih (PBI_PB4_06) 7. Sistem harus mampu menyediakan fungsi untuk menyimpan data pengeluaran ke <i>database</i> (PBI_PB4_07) 8. Sistem harus mampu meng-<i>update</i> diagram pengeluaran berdasarkan kategori dari pengeluaran yang telah dicatat pada halaman keuangan saya (PBI_PB4_08)



5	PB_05	<ol style="list-style-type: none"> 1. Sistem harus mampu menampilkan halaman menambah pemasukan untuk <i>member</i> (PBI_PB5_01) 2. Sistem harus mampu menampilkan tanggal untuk memberitahu <i>member</i> sekarang tanggal berapa (PBI_PB5_02) 3. Sistem harus mampu menyediakan kolom untuk memasukan jumlah pengeluaran dan <i>note</i> terhadap pengeluaran yang akan dikeluarkan (PBI_PB5_03) 4. Sistem harus mampu menyediakan tombol <i>submit</i> untuk menyimpan data pemasukan ke <i>database</i> (PBI_PB5_04) 5. Sistem harus mampu menyediakan fungsi untuk menyimpan data pemasukan ke <i>database</i> (PBI_PB5_05) 6. Sistem harus mampu meng-<i>update</i> total uang pada halaman keuangan saya dengan menambahkan jumlah pemasukan dengan total uang (PBI_PB5_06)
6	PB_06	<ol style="list-style-type: none"> 1. Sistem harus mampu menampilkan halaman riwayat keuangan untuk <i>member</i> (PBI_PB6_01) 2. Sistem harus mampu menampilkan tanggal untuk memberitahu <i>member</i> sekarang tanggal berapa (PBI_PB6_02) 3. Sistem harus mampu menyediakan tampilan daftar pengeluaran dan pemasukan (PBI_PB6_03) 4. Sistem harus mampu menampilkan daftar pengeluaran atau pemasukan berdasarkan tanggal dari <i>database</i> (PBI_PB6_04)
7	PB_07	<ol style="list-style-type: none"> 1. Sistem harus mampu menampilkan halaman perencanaan keuangan untuk <i>member</i> (PBI_PB7_01) 2. Sistem harus mampu menyediakan pilihan tombol perencanaan keuangan yang terdiri dari perencanaan anggaran untuk membuka halaman perencanaan anggaran, batas harian untuk membuka halaman batas harian, dan perencanaan pembelian untuk membuka halaman perencanaan pembelian (PBI_PB7_02)
8	PB_08	<ol style="list-style-type: none"> 1. Sistem harus mampu menampilkan halaman perencanaan anggaran untuk <i>member</i> (PBI_PB8_01) 2. Sistem harus mampu menyediakan kolom untuk memasukan anggaran gadget, harian, hobi dan hiburan, kesehatan, liburan, pendidikan, transportasi, dan zakat atau amal (PBI_PB8_02)

		<ol style="list-style-type: none"> 3. Sistem harus mampu menyediakan tombol untuk membuka halaman keterangan anggaran (PBI_PB8_03) 4. Sistem harus mampu menyediakan halaman keterangan kategori anggaran untuk melihat keterangan dari setiap kategori pada anggaran (PBI_PB8_04) 5. Sistem harus mampu menyediakan tombol <i>submit</i> untuk menyimpan data perencanaan anggaran ke <i>database</i> (PBI_PB8_05) 6. Sistem harus mampu menyediakan fungsi untuk menyimpan data perencanaan anggaran ke <i>database</i> (PBI_PB8_06) 7. Sistem harus mampu meng-<i>update</i> nilai maksimum diagram pengeluaran berdasarkan kategori pada halaman keuangan saya (PBI_PB8_07)
9	PB_09	<ol style="list-style-type: none"> 1. Sistem harus mampu menampilkan halaman batas harian untuk <i>member</i> (PBI_PB9_01) 2. Sistem harus mampu menyediakan kolom untuk menginputkan jumlah batas harian (PBI_PB9_02) 3. Sistem harus mampu menyediakan pilihan <i>show reminder</i> yang bernilai <i>on</i> atau <i>off</i> (PBI_PB9_03) 4. Sistem harus mampu menyediakan tombol <i>submit</i> untuk menyimpan data batasan harian ke <i>database</i> (PBI_PB9_04) 5. Sistem harus mampu menyediakan fungsi untuk menyimpan data batasan harian ke <i>database</i> (PBI_PB9_05) 6. Sistem harus mampu memberikan notifikasi pengingat terkait sisa batas harian apabila <i>show reminder</i> bernilai <i>on</i> pada batas harian yang sudah dilakukan (PBI_PB9_06) 7. Sistem harus mampu menampilkan sisa dari batas harian pada halaman keuangan saya yang telah disimpan pada <i>database</i> (PBI_PB9_07)
10	PB_10	<ol style="list-style-type: none"> 1. Sistem harus mampu menampilkan halaman perencanaan pembelian untuk <i>member</i> (PBI_PB10_01) 2. Sistem harus mampu menyediakan tombol berdasarkan jumlah tabungan untuk membuka halaman berdasarkan jumlah tabungan dan tombol berdasarkan tanggal pembelian untuk membuka halaman berdasarkan tanggal pembelian pada halaman perencanaan pembelian (PBI_PB10_02) 3. Sistem harus mampu menampilkan halaman berdasarkan jumlah tabungan apabila tombol berdasarkan jumlah

		<p>tabungan ditekan yang terdiri dari kolom untuk menginputkan harga produk dan jumlah nabung per hari, pilihan <i>show reminder</i> yang bernilai <i>on</i> atau <i>off</i>, serta tombol <i>submit</i> untuk menyimpan data berdasarkan jumlah tabungan ke <i>database</i> (PBI_PB10_03)</p> <p>4. Sistem harus mampu menyediakan fungsi untuk menyimpan data berdasarkan jumlah tabungan ke <i>database</i> (PBI_PB10_04)</p> <p>5. Sistem harus mampu menampilkan halaman berdasarkan tanggal pembelian apabila tombol berdasarkan tanggal pembelian ditekan yang terdiri dari kolom untuk menginputkan harga produk dan tanggal pembelian, pilihan <i>show reminder</i> yang bernilai <i>on</i> atau <i>off</i>, serta tombol <i>submit</i> untuk menyimpan data berdasarkan tanggal pembelian ke <i>database</i> (PBI_PB10_05)</p> <p>6. Sistem harus mampu menyediakan fungsi untuk menyimpan data berdasarkan tanggal pembelian ke <i>database</i> (PBI_PB10_06)</p> <p>7. Sistem harus mampu memberikan notifikasi pengingat terkait nabung yang harus dilakukan apabila <i>show reminder</i> bernilai <i>on</i> pada perencanaan pembelian berdasarkan Jumlah Tabungan atau berdasarkan tanggal pembelian yang sudah dilakukan (PBI_PB10_07)</p>
11	PB_11	<p>1. Sistem harus mampu menampilkan halaman profil untuk <i>member</i> (PBI_PB11_01)</p> <p>2. Sistem harus mampu menampilkan nama pengguna yang sudah <i>sign in</i> ke dalam aplikasi (PBI_PB11_02)</p> <p>3. Sistem harus mampu menampilkan <i>email</i> pengguna yang sudah <i>sign in</i> ke dalam aplikasi (PBI_PB11_03)</p> <p>4. Sistem harus mampu menyediakan tombol <i>change password</i> untuk mengubah <i>password</i> pengguna (PBI_PB11_04)</p> <p>5. Sistem harus mampu menyediakan fungsi untuk <i>change password</i> (PBI_PB11_05)</p> <p>6. Sistem harus mampu menyediakan tombol <i>sign out</i> untuk keluar dari aplikasi sebagai <i>member</i> (PBI_PB11_06)</p>

4.3.2 Kebutuhan Fungsional

Dari hasil penyebaran lembar kuesioner maka didapat 11 kebutuhan. Setiap kebutuhan fungsional akan diberikan kode SRS_FM_F_X dan setiap *product backlog* akan diberikan kode PB_Y. SRS merupakan singkatan dari *System Requirement Spesification*, FM merupakan singkatan dari *Financial Management*

App, F merupakan kebutuhan fungsional, X merupakan nomor dari definisi kebutuhan, PB merupakan *product backlog*, dan Y merupakan nomor dari *product backlog*. Untuk lebih jelasnya dapat dilihat pada tabel 4.4 berikut.

Tabel 4. 4 Kebutuhan Fungsional

No	Kode Fungsional	Definisi Kebutuhan	Kode Product Backlog
1	SRS_FM_F_01	Pengguna dapat <i>sign up</i> untuk mendaftarkan diri sebagai <i>member</i>	PB_01
2	SRS_FM_F_02	Pengguna dapat <i>sign in</i> untuk masuk ke dalam sistem	PB_02
3	SRS_FM_F_03	<i>Member</i> dapat melihat informasi keuangannya	PB_03
4	SRS_FM_F_04	<i>Member</i> dapat mencatat pengeluaran keuangan ke dalam sistem	PB_04
5	SRS_FM_F_05	<i>Member</i> dapat mencatat pemasukan keuangan ke dalam sistem	PB_05
6	SRS_FM_F_06	<i>Member</i> dapat melihat riwayat keuangan berdasarkan tanggal pada sistem	PB_06
7	SRS_FM_F_07	<i>Member</i> dapat memilih daftar perencanaan keuangan pada sistem	PB_07
8	SRS_FM_F_08	<i>Member</i> dapat membuat perencanaan anggaran keuangan pada sistem	PB_08
9	SRS_FM_F_09	<i>Member</i> dapat membuat batas harian keuangan pada sistem	PB_09
10	SRS_FM_F_10	<i>Member</i> dapat membuat perencanaan pembelian pada sistem	PB_10
11	SRS_FM_F_11	<i>Member</i> dapat melihat informasi pengguna	PB_11

4.3.3 Kebutuhan Non Fungsional

Didapat satu kebutuhan non fungsional. Kebutuhan non fungsional akan diberikan kode SRS_FM_NF_X. SRS merupakan singkatan dari *System Requirement Specification*, FM merupakan singkatan dari Financial Management App, NF merupakan kebutuhan non fungsional, dan X merupakan nomor dari definisi kebutuhan. Untuk lebih jelasnya dapat dilihat pada tabel 4.5 berikut.

Tabel 4. 5 Kebutuhan Non Fungsional

No	Kode Fungsi	Jenis	Deskripsi
1	SRS_FM_NF_01	<i>Usability</i>	Sistem memiliki tampilan <i>user friendly</i> sehingga mudah digunakan

4.4 Use Case Diagram

Pada tahap ini akan membahas gambar diagram yang menjelaskan interaksi antara pengguna dengan sistem aplikasi. Untuk lebih jelasnya dapat dilihat pada gambar 4.6 berikut.

Tabel 4. 6 Informasi Nama Use Case Berdasarkan Kode Fungsi

No	Kode Fungsi	Use Case
1	SRS_FM_F_01	<i>Sign Up</i>
2	SRS_FM_F_02	<i>Sign In</i>
3	SRS_FM_F_03	Melihat Keuangan Saya
4	SRS_FM_F_04	Menambah Pengeluaran
5	SRS_FM_F_05	Menambah Pemasukan
6	SRS_FM_F_06	Melihat Riwayat Keuangan
7	SRS_FM_F_07	Melihat Perencanaan Keuangan
8	SRS_FM_F_08	Membuat Perencanaan Anggaran
9	SRS_FM_F_09	Membuat Batas Harian
10	SRS_FM_F_10	Membuat Perencanaan Pembelian
11	SRS_FM_F_11	Melihat Profil

Pada gambar 4.1 merupakan gambar *use case diagram* FM App. Pada gambar 4.1 menjelaskan interaksi antara pengguna dan *member* terhadap aplikasi. Interaksi antara pengguna dengan aplikasi adalah pengguna dapat mendaftar sebagai *member* aplikasi dengan *Sign Up* serta masuk kedalam sistem aplikasi dengan *Sign In*, sedangkan interaksi antara *member* dengan aplikasi adalah *member* dapat melihat Keuangan Saya yang *extend* dengan Menambah Pengeluaran dan Menambah Pemasukan, melihat Riwayat Keuangan, melihat Perencanaan Keuangan yang *extend* dengan Membuat Perencanaan Anggaran, Membuat Batas Harian, Membuat Perencanaan Pembelian, dan yang terakhir dapat Melihat Profil. Selain itu, *member* merupakan generalisasi aktor dari pengguna.

	1. <i>Actor</i> mengisikan <i>email</i> dan <i>password</i> pada <i>form Sign In</i> (<i>email</i> dan <i>password</i> harus diisi)	
	2. <i>Actor</i> menekan tombol <i>Sign In</i> pada halaman <i>Sign In</i>	
		3. <i>System</i> memroses validasi data <i>email</i> dan <i>password</i> yang dimasukan oleh <i>actor</i>
<i>Alternative Flow</i>	<p>1. Jika <i>email</i> dan <i>password</i> salah, maka <i>actor</i> akan diarahkan ke halaman <i>Sign In</i> kembali oleh <i>system</i> dan akan ada pemberitahuan “<i>Wrong Email or Password</i>”</p> <p>2. Jika <i>email</i> dan <i>password</i> tidak diisi, maka akan ada <i>attention error notification</i> pada kolom <i>email</i> dan <i>password</i> yang bertuliskan “<i>Required</i>”</p> <p>3. Jika pengguna lupa <i>password</i>, maka pengguna dapat menekan tulisan “<i>Here</i>” pada <i>forgot password</i>. Lalu pengguna dapat memasukan <i>email</i> yang sudah terdaftar</p>	
<i>Post – Condition</i>	<i>Actor</i> berubah menjadi <i>member</i> dan <i>System</i> <i>direct</i> ke halaman <i>Keuangan Saya</i>	

4.5.2 Use Case Scenario Sign Up

Tabel 4. 8 Use Case Scenario Sign Up

<i>Flow of Events</i> untuk <i>Sign Up</i>		
<i>Objective</i>	<i>Use Case</i> ini berfungsi untuk mendaftarkan sebagai <i>member</i>	
<i>Actor</i>	Pengguna	
<i>Pre – Condition</i>	<i>Actor</i> belum memiliki <i>email</i> dan <i>password</i> yang terdaftar pada <i>system</i> serta berada pada halama <i>Sign Up</i>	
<i>Main Flow</i>	<i>Actor</i>	<i>System</i>



	1. <i>Actor</i> mengisi <i>sign up form</i> yang terdiri dari <i>Name</i> , <i>Email</i> , dan <i>Password</i>	
	2. <i>Actor</i> menekan tombol <i>Sign Up</i> pada halaman <i>Sign Up</i>	
		3. <i>System</i> memroses dan memasukan data <i>actor</i> ke <i>database</i>
<i>Alternative Flow</i>	1. Apabila ada data yang belum dimasukan maka akan muncul <i>attention error notification</i> pada setiap kolom yang tidak dimasukan	
<i>Post – Condition</i>	<i>System</i> men- <i>direct</i> ke halaman Keuangan Saya	

4.5.3 Use Case Scenario Melihat Keuangan Saya

Tabel 4. 9 Use Case Scenario Melihat Keuangan Saya

<i>Flow of Events</i> untuk Melihat Keuangan Saya		
<i>Objective</i>	<i>Use Case</i> ini berfungsi untuk melihat informasi keuangan <i>member</i>	
<i>Actor</i>	<i>Member</i>	
<i>Pre – Condition</i>	Sudah <i>Sign In</i> sebagai <i>member</i>	
<i>Main Flow</i>	<i>Actor</i>	<i>System</i>
	1. Tombol Keuangan Saya ditekan pada <i>Bottom Navigation Bar</i>	2. <i>System</i> menampilkan halaman Keuangan Saya
<i>Alternative Flow</i>	-	
<i>Post – Condition</i>	Actor dapat melihat informasinya yang terdiri dari total uang pemasukan, diagram pengeluaran berdasarkan kategori anggaran, sisa batas harian, serta dapat menambahkan pemasukan atau pengeluaran	



4.5.4 Use Case Scenario Menambah Pengeluaran

Tabel 4. 10 Use Case Scenario Menambah Pengeluaran

<i>Flow of Events</i> untuk Menambah Pengeluaran		
<i>Objective</i>	<i>Use Case</i> ini berfungsi untuk mencatat pengeluaran yang sudah dilakukan	
<i>Actor</i>	<i>Member</i>	
<i>Pre – Condition</i>	Sudah <i>Sign In</i> sebagai <i>member</i> dan tombol Keuangan Saya ditekan	
<i>Main Flow</i>	<i>Actor</i>	<i>System</i>
	1. Tombol “+Pengeluaran ” ditekan	
		2. <i>System</i> menampilkan halaman pengeluaran
	3. <i>Actor</i> memilih kategori pengeluaran, menginputkan jumlah pengeluaran, dan menuliskan note pengeluaran	
	4. Tombol submit ditekan	
		5. <i>System</i> menyimpan data pengeluaran ke <i>database</i>
<i>Alternative Flow</i>	1. Jika <i>Actor</i> memilih kategori Tidak Dianggarkan dan menekan tombol submit, maka halaman Perencanaan Alokasi Dana Tidak Dianggarkan ditampilkan 2. Jika <i>Actor</i> menyetujui dari rekomendasi rencana alokasi dana tidak dianggarkan, maka batas harian tidak berkurang 3. Jika <i>Actor</i> memasukan jumlah pengeluaran tidak dianggarkan melebihi kemampuan keuangan pribadi, maka akan ditampilkan <i>attention error notification</i> pada kolom jumlah	

	<p>pengeluaran berupa “Max pengeluaran tidak dianggarkan = Rp...”</p> <p>4. Jika Actor memasukan pengeluaran dengan kategori tidak dianggarkan dan sudah melewati batas maksimal tidak dianggarkan, maka akan ditampilkan peringatan “Anda tidak dapat mengeluarkan dana Tidak Dianggarkan!”</p>
<i>Post – Condition</i>	<p>Actor dapat melihat diagram pengeluaran yang telah dilakukan berdasarkan kategori pada halaman Keuangan Saya dan Batas Harian berkurang</p>

4.5.5 Use Case Scenario Menambah Pemasukan

Tabel 4. 11 Use Case Scenario Menambah Pemasukan

Flow of Events untuk Menambah Pemasukan													
<i>Objective</i>	Use Case ini berfungsi untuk mencatat pemasukan yang sudah dilakukan												
<i>Actor</i>	Member												
<i>Pre – Condition</i>	Sudah <i>Sign In</i> sebagai <i>member</i> dan tombol Keuangan Saya ditekan												
<i>Main Flow</i>	<table border="1"> <thead> <tr> <th>Actor</th> <th>System</th> </tr> </thead> <tbody> <tr> <td>1. Tombol “+Pemasukan ” ditekan</td> <td></td> </tr> <tr> <td></td> <td>2. <i>System</i> menampilkan halaman pemasukan</td> </tr> <tr> <td>3. <i>Actor</i> menginputkan jumlah pengeluaran, dan menuliskan note pengeluaran</td> <td></td> </tr> <tr> <td>4. Tombol submit ditekan</td> <td></td> </tr> <tr> <td></td> <td>5. <i>System</i> menyimpan data pemasukan ke <i>database</i></td> </tr> </tbody> </table>	Actor	System	1. Tombol “+Pemasukan ” ditekan			2. <i>System</i> menampilkan halaman pemasukan	3. <i>Actor</i> menginputkan jumlah pengeluaran, dan menuliskan note pengeluaran		4. Tombol submit ditekan			5. <i>System</i> menyimpan data pemasukan ke <i>database</i>
	Actor	System											
	1. Tombol “+Pemasukan ” ditekan												
		2. <i>System</i> menampilkan halaman pemasukan											
	3. <i>Actor</i> menginputkan jumlah pengeluaran, dan menuliskan note pengeluaran												
4. Tombol submit ditekan													
	5. <i>System</i> menyimpan data pemasukan ke <i>database</i>												
<i>Alternative Flow</i>	-												



<i>Post – Condition</i>	Total Uang pada halaman Keuangan Saya akan bertambah sesuai dengan pemasukan yang telah dimasukan
-------------------------	---

4.5.6 Use Case Scenario Melihat Riwayat Keuangan

Tabel 4. 12 Use Case Scenario Melihat Riwayat Keuangan

<i>Flow of Events</i> untuk Melihat Riwayat Keuangan		
<i>Objective</i>	<i>Use Case</i> ini berfungsi untuk melihat riwayat keuangan berdasarkan tanggal yang terdiri dari informasi pengeluaran dan atau pemasukan yang sudah pernah dilakukan	
<i>Actor</i>	<i>Member</i>	
<i>Pre – Condition</i>	Sudah <i>Sign In</i> sebagai <i>member</i>	
<i>Main Flow</i>	<i>Actor</i>	<i>System</i>
	1. Tombol Riwayat Keuangan ditekan pada <i>Bottom Navigation Bar</i>	2. <i>System</i> menampilkan halaman Riwayat Keuangan
<i>Alternative Flow</i>	-	
<i>Post – Condition</i>	Actor dapat melihat riwayat keuangan berdasarkan tanggal yang terdiri dari informasi pemasukan dan pengeluaran yang pernah dilakuakn	

4.5.7 Use Case Scenario Melihat Perencanaan Keuangan

Tabel 4. 13 Use Case Scenario Melihat Perencanaan Keuangan

<i>Flow of Events</i> untuk Melihat Perencanaan Keuangan		
<i>Objective</i>	<i>Use Case</i> ini berfungsi untuk merencanakan penggunaan keuangan yang dimiliki	
<i>Actor</i>	<i>Member</i>	
<i>Pre – Condition</i>	Sudah <i>Sign In</i> sebagai <i>member</i>	
<i>Main Flow</i>	<i>Actor</i>	<i>System</i>
	1. Tombol Perencanaan	

	Keuangan ditekan pada <i>Bottom Navigation Bar</i>	
		2. <i>System</i> menampilkan halaman Perencanaan Keuangan
<i>Alternative Flow</i>	-	
<i>Post – Condition</i>	Actor dapat memilih menampilkan halaman Perencanaan Anggaran, Batas Harian, dan Perencanaan Pembelian	

4.5.8 Use Case Scenario Membuat Perencanaan Anggaran

Tabel 4. 14 Use Case Scenario Membuat Perencanaan Anggaran

<i>Flow of Events</i> untuk Melihat Perencanaan Anggaran		
<i>Objective</i>	Use Case ini berfungsi untuk merencanakan anggaran keuangan yang akan digunakan	
<i>Actor</i>	Member	
<i>Pre – Condition</i>	Sudah <i>Sign In</i> sebagai member dan tombol Perencanaan Keuangan ditekan	
<i>Main Flow</i>	<i>Actor</i>	<i>System</i>
	1. Tombol Perencanaan Anggaran ditekan	
		2. <i>System</i> menampilkan halaman Perencanaan Anggaran
	3. Actor menginputkan jumlah anggaran setiap kategori	
	4. Actor menekan tombol submit	
		5. <i>System</i> menyimpan data perencanaan anggaran ke <i>database</i>

<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika <i>Actor</i> ingin mengetahui keterangan kategori anggaran, maka dapat menekan tulisan "<i>here</i>" disebelah tulisan "Keterangan kategori anggaran?" 2. Jika <i>Actor</i> memasukan alokasi dana anggaran melebihi total keuangan yang dimiliki, maka akan ditampilkan "Total anggaran melebihi total keuangan Anda! Tambah Pemasukan!"
<i>Post – Condition</i>	<i>Actor</i> dapat mengalokasikan keuangannya berdasarkan kategori yang tersedia dan <i>actor</i> dapat melihat diagram anggaran berdasarkan kategori yang tersedia pada halaman Keuangan Saya

4.5.9 Use Case Scenario Membuat Batas Harian

Tabel 4. 15 Use Case Scenario Membuat Batas Harian

<i>Flow of Events</i> untuk Membuat Batas Harian		
<i>Objective</i>	<i>Use Case</i> ini berfungsi untuk membuat batas pengeluaran harian dan pengingat ketika pengeluaran sudah mendekati batas	
<i>Actor</i>	<i>Member</i>	
<i>Pre – Condition</i>	Sudah <i>Sign In</i> sebagai <i>member</i> dan tombol Perencanaan Keuangan ditekan	
<i>Main Flow</i>	<i>Actor</i>	<i>System</i>
	<ol style="list-style-type: none"> 1. Tombol Batas Harian ditekan pada halaman perencanaan keuangan 	
		<ol style="list-style-type: none"> 2. <i>System</i> menampilkan halaman Batas Harian
	<ol style="list-style-type: none"> 3. <i>Actor</i> menginputkan jumlah uang yang menjadi batas pengeluaran setiap harinya dan menekan <i>toggle</i> pada <i>show reminder</i> 	



	jika menginginkan untuk mengaktifkan atau tidak mengaktifkan <i>reminder notification</i>	
	4. Tombol submit ditekan	
		5. <i>System</i> menyimpan data Batas Harian pada <i>database</i>
<i>Alternative Flow</i>	1. Jika jumlah batasan harian yang dimasukkan melebihi dari kemampuan uang yang dimiliki, akan ditampilkan peringatan "Maksimum batas harian ..."	
<i>Post – Condition</i>	<i>Actor</i> dapat melihat Sisa Batas Harian pada halaman Keuangan Saya dan akan mendapatkan notifikasi jika <i>toggle</i> pada <i>show reminder</i> bernilai on	

4.5.10 Use Case Scenario Melihat Perencanaan Pembelian

Tabel 4. 16 Use Case Scenario Melihat Perencanaan Pembelian

<i>Flow of Events</i> untuk Membuat Perencanaan Pembelian		
<i>Objective</i>	<i>Use Case</i> ini berfungsi untuk merencanakan pembelian suatu barang	
<i>Actor</i>	<i>Member</i>	
<i>Pre – Condition</i>	Sudah <i>Sign In</i> sebagai <i>member</i> dan tombol Perencanaan Keuangan ditekan	
<i>Main Flow</i>	<i>Actor</i>	<i>System</i>
	1. Tombol Perencanaan Pembelian ditekan	
		2. <i>System</i> menampilkan halaman Perencanaan Keuangan
	3. <i>Actor</i> memilih Berdasarkan Jumlah Tabungan	



	4. Tombol Berdasarkan Jumlah Tabungan ditekan	
		5. <i>System</i> menampilkan halaman Berdasarkan Jumlah Tabungan
	6. <i>Actor</i> menginputkan harga produk, jumlah nabung perhari, menekan <i>toggle</i> pada <i>show reminder</i> jika menginginkan untuk mengaktifkan atau tidak mengaktifkan <i>reminder notification</i> , dan menekan tombol submit	
		7. <i>System</i> menyimpan data Berdasarkan Jumlah Tabungan ke <i>database</i>
<i>Alternative Flow</i>	1. <i>Actor</i> memilih Berdasarkan Tanggal Pembelian, <i>Actor</i> menginputkan harga produk, tanggal pembelian, menekan <i>toggle</i> pada <i>show reminder</i> jika menginginkan untuk mengaktifkan atau tidak mengaktifkan <i>reminder notification</i> , dan menekan tombol submit, maka actor dapat melihat progress uang yang terkumpul untuk membeli barang pada halaman Perencanaan Pembelian dan akan mendapatkan notifikasi jika <i>toggle</i> pada <i>show reminder</i> bernilai on	
<i>Post – Condition</i>	<i>Actor</i> dapat melihat progress uang yang terkumpul untuk membeli barang pada halaman Perencanaan Pembelian dan akan mendapatkan notifikasi jika <i>toggle</i> pada <i>show reminder</i> bernilai on	



4.5.11 Use Case Scenario Melihat Profil

Tabel 4. 17 Use Case Scenario Melihat Profil

Flow of Events untuk Melihat Profil		
Objective	Use Case ini berfungsi untuk melihat data <i>member</i> , mengubah <i>password</i> , dan <i>Sign Out</i>	
Actor	Member	
Pre – Condition	Sudah <i>Sign In</i> sebagai <i>member</i>	
Main Flow	Actor	System
	1. Tombol Profil ditekan pada <i>Bottom Navigation Bar</i>	
		2. System menampilkan halaman Profil
Alternative Flow	1. Jika tombol <i>Change Password</i> ditekan, maka akan menampilkan halaman untuk merubah <i>password</i> dan tombol <i>submit</i> untuk menyimpan dan <i>actor</i> otomatis keluar dari sistem 2. Jika tombol <i>Sign Out</i> ditekan, maka <i>actor</i> akan keluar dari sistem	
Post – Condition	Actor dapat melihat informasi pengguna sistem	

4.6 Sprint Planning

Tahap *sprint planning* pada penelitian ini akan menjadi dua bagian, yaitu *sprint goal* dan *sprint backlog*. Pada *sprint planning* ini akan dibuat dua *sprint*. Untuk lebih jelasnya dapat dilihat pada sub bab berikutnya.

4.6.1 Sprint Goal

Sprint Goal akan dibagi menjadi dua bagian yaitu, *sprint goal* pada *sprint 1* dan *sprint 2*. Untuk *sprint goal* pada *sprint 1* adalah membuat fitur dasar FM App, sedangkan untuk *sprint 2* adalah membuat fitur lanjutan pada FM App.

4.6.2 Sprint Backlog

Sprint backlog merupakan daftar pekerjaan yang akan dilakukan pada setiap *sprint*. *Sprint backlog* akan dibagi menjadi dua bagian, yaitu *sprint backlog* berdasarkan *sprint 1* dan *sprint 2*. Untuk lebih jelasnya dapat dilihat pada sub bab berikutnya.

4.6.2.1 Sprint 1

Tabel 4. 18 Deskripsi Pekerjaan pada *Sprint 1*

SPRINT 1		
Task Description: Membuat Fitur Dasar FM App		
No	Kode Product Backlog	Product Backlog Item
1	PB_01	<ol style="list-style-type: none"> 1. Sistem harus mampu menampilkan halaman <i>sign up</i> untuk pengguna (PBI_PB1_01) 2. Sistem harus mampu menyediakan kolom untuk menginputkan nama, <i>email</i>, dan <i>password</i> (PBI_PB1_02) 3. Sistem harus mampu menampilkan peringatan ketika kolom nama, <i>email</i>, dan <i>password</i> kosong (PBI_PB1_03) 4. Sistem harus mampu menyediakan tombol <i>sign up</i> untuk mendaftarkan pengguna sebagai <i>member</i> (PBI_PB1_04) 5. Sistem harus mampu menyimpan data pengguna yang sudah terdaftar ke dalam <i>database</i> (PBI_PB1_05)
2	PB_02	<ol style="list-style-type: none"> 1. Sistem harus mampu menampilkan halaman <i>sign in</i> untuk pengguna (PBI_PB2_01) 2. Sistem harus mampu menyediakan kolom untuk menginputkan <i>email</i> dan <i>password</i> (PBI_PB2_02) 3. Sistem harus mampu menampilkan peringatan ketika kolom <i>email</i> dan <i>password</i> kosong (PBI_PB2_03) 4. Sistem harus mampu menyediakan tombol Sign In untuk autentikasi <i>email</i> dan <i>password</i> pengguna (PBI_PB2_04) 5. Sistem harus mampu menyediakan fungsi <i>forgot password</i> apabila pengguna lupa <i>password</i> dari <i>email</i> yang sudah terdaftar (PBI_PB2_05) 6. Sistem harus mampu menyediakan <i>pop up</i> untuk <i>forgot password</i> yang menyediakan kolom untuk menginputkan <i>email</i> dan tombol <i>submit</i> untuk mengirimkan perubahan <i>password</i> ke <i>email</i> yang terdaftar (PBI_PB2_06)
3	PB_03	<ol style="list-style-type: none"> 1. Sistem harus mampu menampilkan halaman keuangan saya untuk <i>member</i> (PBI_PB3_01) 2. Sistem harus mampu menampilkan tanggal untuk memberitahu <i>member</i> sekarang tanggal berapa (PBI_PB3_02)

		<ol style="list-style-type: none"> 3. Sistem harus mampu menampilkan total uang yang dimiliki <i>member</i> sekarang (PBI_PB3_03) 4. Sistem harus mampu menyediakan fungsi untuk menampilkan total uang yang dimiliki <i>member</i> sekarang (PBI_PB3_04) 5. Sistem harus mampu menampilkan diagram berdasarkan kategori anggaran untuk mengetahui pengeluaran tiap kategori anggaran yang sudah dilakukan (PBI_PB3_05) 6. Sistem harus mampu menyediakan fungsi untuk menampilkan diagram berdasarkan kategori anggaran untuk mengetahui pengeluaran tiap kategori anggaran yang sudah dilakukan (PBI_PB3_06) 7. Sistem harus mampu menampilkan sisa dari batas harian untuk mengetahui sisa pengeluaran yang dapat dikeluarkan berapa (PBI_PB3_07) 8. Sistem harus mampu menyediakan fungsi untuk menampilkan sisa dari batas harian untuk mengetahui sisa pengeluaran yang dapat dikeluarkan berapa (PBI_PB3_08)
4	PB_04	<ol style="list-style-type: none"> 1. Sistem harus mampu menampilkan halaman keuangan saya untuk <i>member</i> (PBI_PB3_01) 2. Sistem harus mampu menampilkan tanggal untuk memberitahu <i>member</i> sekarang tanggal berapa (PBI_PB3_02) 3. Sistem harus mampu menampilkan total uang yang dimiliki <i>member</i> sekarang (PBI_PB3_03) 4. Sistem harus mampu menyediakan fungsi untuk menampilkan total uang yang dimiliki <i>member</i> sekarang (PBI_PB3_04) 5. Sistem harus mampu menampilkan diagram berdasarkan kategori anggaran untuk mengetahui pengeluaran tiap kategori anggaran yang sudah dilakukan (PBI_PB3_05) 6. Sistem harus mampu menyediakan fungsi untuk menampilkan diagram berdasarkan kategori anggaran untuk mengetahui pengeluaran tiap kategori anggaran yang sudah dilakukan (PBI_PB3_06) 7. Sistem harus mampu menampilkan sisa dari batas harian untuk mengetahui sisa pengeluaran yang dapat dikeluarkan berapa (PBI_PB3_07)



		8. Sistem harus mampu menyediakan fungsi untuk menampilkan sisa dari batas harian untuk mengetahui sisa pengeluaran yang dapat dikeluarkan berapa (PBI_PB3_08)
5	PB_05	<ol style="list-style-type: none"> 1. Sistem harus mampu menampilkan halaman menambah pemasukan untuk <i>member</i> (PBI_PB5_01) 2. Sistem harus mampu menampilkan tanggal untuk memberitahu <i>member</i> sekarang tanggal berapa (PBI_PB5_02) 3. Sistem harus mampu menyediakan kolom untuk menginputkan jumlah pengeluaran dan <i>note</i> terhadap pengeluaran yang akan dikeluarkan (PBI_PB5_03) 4. Sistem harus mampu menyediakan tombol <i>submit</i> untuk menyimpan data pemasukan ke <i>database</i> (PBI_PB5_04) 5. Sistem harus mampu menyediakan fungsi untuk menyimpan data pemasukan ke <i>database</i> (PBI_PB5_05) 6. Sistem harus mampu meng-<i>update</i> total uang pada halaman keuangan saya dengan menambahkan jumlah pemasukan dengan total uang (PBI_PB5_06)
6	PB_06	<ol style="list-style-type: none"> 1. Sistem harus mampu menampilkan halaman riwayat keuangan untuk <i>member</i> (PBI_PB6_01) 2. Sistem harus mampu menampilkan tanggal untuk memberitahu <i>member</i> sekarang tanggal berapa (PBI_PB6_02) 3. Sistem harus mampu menyediakan tampilan daftar pengeluaran dan pemasukan (PBI_PB6_03) 4. Sistem harus mampu menampilkan daftar pengeluaran atau pemasukan berdasarkan tanggal dari <i>database</i> (PBI_PB6_04)

Pada *sprint* 1 akan dikerjakan 6 *use case* dari 11 *use case* yang ada. *Use case* ini menunjang fitur dasar pada FM App, dimana jika 6 *use case* ini tidak dikerjakan terlebih dahulu, maka 5 *use case* selanjutnya tidak dapat dikerjakan karena memiliki keterkaitan.

4.6.2.2 Sprint 2

Tabel 4. 19 Deskripsi Pekerjaan pada Sprint 2

SPRINT 2		
Task Description: Membuat Fitur Lanjutan FM App		
No	Kode Product Backlog	Product Backlog Item
1	PB_07	<ol style="list-style-type: none"> 1. Sistem harus mampu menampilkan halaman perencanaan keuangan untuk <i>member</i> (PBI_PB7_01) 2. Sistem harus mampu menyediakan pilihan tombol perencanaan keuangan yang terdiri dari perencanaan anggaran untuk membuka halaman perencanaan anggaran, batas harian untuk membuka halaman batas harian, dan perencanaan pembelian untuk membuka halaman perencanaan pembelian (PBI_PB7_02)
2	PB_08	<ol style="list-style-type: none"> 1. Sistem harus mampu menampilkan halaman perencanaan anggaran untuk <i>member</i> (PBI_PB8_01) 2. Sistem harus mampu menyediakan kolom untuk menginputkan anggaran gadget, harian, hobi dan hiburan, kesehatan, liburan, pendidikan, transportasi, dan zakat atau amal (PBI_PB8_02) 3. Sistem harus mampu menyediakan tombol untuk membuka halaman keterangan anggaran (PBI_PB8_03) 4. Sistem harus mampu menyediakan halaman keterangan kategori anggaran untuk melihat keterangan dari setiap kategori pada anggaran (PBI_PB8_04) 5. Sistem harus mampu menyediakan tombol <i>submit</i> untuk menyimpan data perencanaan anggaran ke <i>database</i> (PBI_PB8_05) 6. Sistem harus mampu menyediakan fungsi untuk menyimpan data perencanaan anggaran ke <i>database</i> (PBI_PB8_06) 7. Sistem harus mampu meng-<i>update</i> nilai maksimum diagram pengeluaran berdasarkan kategori pada halaman keuangan saya (PBI_PB8_07)
3	PB_09	<ol style="list-style-type: none"> 1. Sistem harus mampu menampilkan halaman batas harian untuk <i>member</i> (PBI_PB9_01) 2. Sistem harus mampu menyediakan kolom untuk menginputkan jumlah batas harian (PBI_PB9_02)

		<ol style="list-style-type: none"> 3. Sistem harus mampu menyediakan pilihan <i>show reminder</i> yang bernilai <i>on</i> atau <i>off</i> (PBI_PB9_03) 4. Sistem harus mampu menyediakan tombol <i>submit</i> untuk menyimpan data batasan harian ke <i>database</i> (PBI_PB9_04) 5. Sistem harus mampu menyediakan fungsi untuk menyimpan data batasan harian ke <i>database</i> (PBI_PB9_05) 6. Sistem harus mampu memberikan notifikasi pengingat terkait sisa batas harian apabila <i>show reminder</i> bernilai <i>on</i> pada batas harian yang sudah dilakukan (PBI_PB9_06) 7. Sistem harus mampu menampilkan sisa dari batas harian pada halaman keuangan saya yang telah disimpan pada <i>database</i> (PBI_PB9_07)
4	PB_10	<ol style="list-style-type: none"> 1. Sistem harus mampu menampilkan halaman perencanaan pembelian untuk <i>member</i> (PBI_PB10_01) 2. Sistem harus mampu menyediakan tombol berdasarkan jumlah tabungan untuk membuka halaman berdasarkan jumlah tabungan dan tombol berdasarkan tanggal pembelian untuk membuka halaman berdasarkan tanggal pembelian pada halaman perencanaan pembelian (PBI_PB10_02) 3. Sistem harus mampu menampilkan halaman berdasarkan jumlah tabungan apabila tombol berdasarkan jumlah tabungan ditekan yang terdiri dari kolom untuk menginputkan harga produk dan jumlah nabung per hari, pilihan <i>show reminder</i> yang bernilai <i>on</i> atau <i>off</i>, serta tombol <i>submit</i> untuk menyimpan data berdasarkan jumlah tabungan ke <i>database</i> (PBI_PB10_03) 4. Sistem harus mampu menyediakan fungsi untuk menyimpan data berdasarkan jumlah tabungan ke <i>database</i> (PBI_PB10_04) 5. Sistem harus mampu menampilkan halaman berdasarkan tanggal pembelian apabila tombol berdasarkan tanggal pembelian ditekan yang terdiri dari kolom untuk menginputkan harga produk dan tanggal pembelian, pilihan <i>show reminder</i> yang bernilai <i>on</i> atau <i>off</i>, serta tombol <i>submit</i> untuk menyimpan data berdasarkan tanggal pembelian ke <i>database</i> (PBI_PB10_05) 6. Sistem harus mampu menyediakan fungsi untuk menyimpan data berdasarkan tanggal pembelian ke <i>database</i> (PBI_PB10_06)



		7. Sistem harus mampu memberikan notifikasi pengingat terkait nabung yang harus dilakukan apabila <i>show reminder</i> bernilai <i>on</i> pada perencanaan pembelian berdasarkan Jumlah Tabungan atau berdasarkan tanggal pembelian yang sudah dilakukan (PBI_PB10_07)
5	PB_11	<ol style="list-style-type: none"> 1. Sistem harus mampu menampilkan halaman profil untuk <i>member</i> (PBI_PB11_01) 2. Sistem harus mampu menampilkan nama pengguna yang sudah <i>sign in</i> ke dalam sistem (PBI_PB11_02) 3. Sistem harus mampu menampilkan <i>email</i> pengguna yang sudah <i>sign in</i> ke dalam sistem (PBI_PB11_03) 4. Sistem harus mampu menyediakan tombol <i>change password</i> untuk mengubah <i>password</i> pengguna (PBI_PB11_04) 5. Sistem harus mampu menyediakan fungsi untuk <i>change password</i> (PBI_PB11_05) 6. Sistem harus mampu menyediakan tombol <i>sign out</i> untuk keluar dari sistem sebagai <i>member</i> (PBI_PB11_06)

Pada *sprint 2* akan dikerjakan 5 sisa *use case* dari 11 *use case* yang ada. *Use case* ini menunjang fitur lanjutan pada FM App.

4.7 Analisis Data

Pada analisis data akan dijelaskan data – data apa saja yang akan digunakan untuk memenuhi kebutuhan sistem. Pada analisis data akan dibagi menjadi dua bagian, yaitu analisis data pada *sprint 1* dan *sprint 2*. Analisis data ini juga akan digunakan sebagai dasar perancangan dan implementasi sistem.

4.7.1 Sprint 1

Pada *sprint 1* dibutuhkan beberapa data selama proses *sprint* berlangsung. Tabel 4.20 merupakan daftar data yang dibutuhkan selama proses *sprint 1* berlangsung. Untuk lebih jelasnya dapat dilihat pada tabel 4.20.

Tabel 4. 20 Data Pada Sprint 1

Data	Deskripsi
<i>Users</i>	Digunakan sebagai menyimpan data – data <i>member</i> FM App.
<i>ID User</i>	Digunakan untuk menyimpan data <i>member</i> berdasarkan <i>id user</i> yang terdaftar.



Pemasukan	Digunakan untuk menyimpan data pemasukan setiap <i>member</i> .
Pengeluaran	Digunakan untuk menyimpan data pengeluaran setiap <i>member</i> .
Total Pengeluaran	Digunakan untuk menyimpan data total pengeluaran setiap <i>member</i> .
Total Uang	Digunakan untuk menyimpan data total uang yang dimiliki setiap <i>member</i> .
Rencana Alokasi Dana	Digunakan untuk menyimpan data rencana alokasi dana ketika melakukan pengeluaran tidak dianggarkan/tidak terduga.
Data Uji	Digunakan untuk menyimpan nilai kondisi keuangan setelah rencana alokasi anggaran diterapkan yang sebagai data uji untuk algoritma Naïve Bayes.

4.7.2 Sprint 2

Pada *sprint 2* dibutuhkan beberapa data selama proses *sprint* berlangsung. Tabel 4.21 merupakan daftar data yang dibutuhkan selama proses *sprint 2* berlangsung. Untuk lebih jelasnya dapat dilihat pada tabel 4.21.

Tabel 4. 21 Data Pada *Sprint 2*

Data	Deskripsi
Anggaran	Digunakan untuk menyimpan data perencanaan anggaran yang telah dibuat.
Batas Harian	Digunakan untuk menyimpan data batas harian yang telah dibuat
Pembelian Berdasarkan Jumlah Tabungan	Digunakan untuk menyimpan data pembelian berdasarkan jumlah tabungan
Pembelian Berdasarkan Tanggal Pembelian	Digunakan untuk menyimpan data pembelian berdasarkan tanggal pembelian

BAB 5 PERANCANGAN DAN IMPLEMENTASI

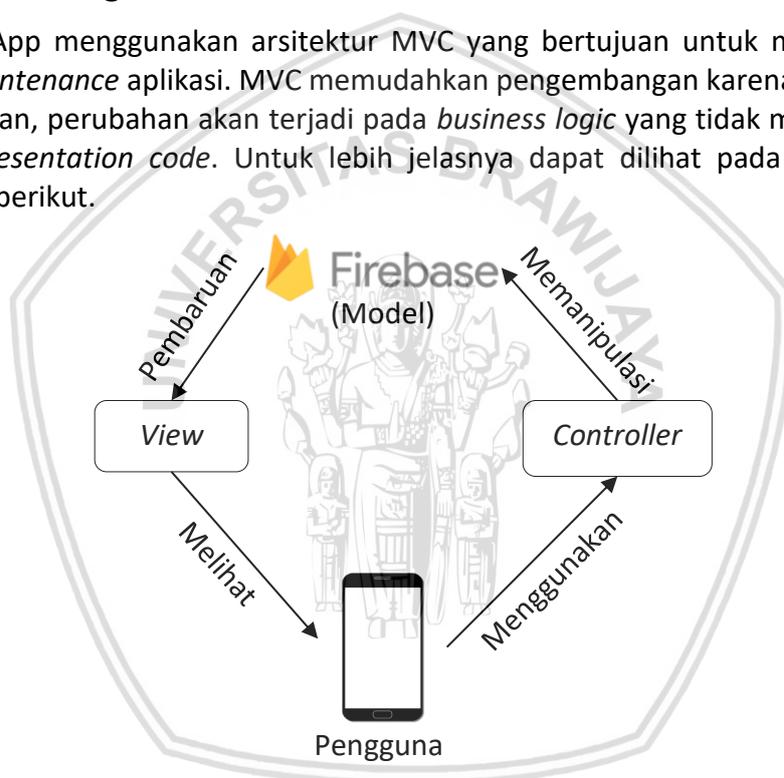
Bab ini akan menjelaskan mengenai tahapan yang dikerjakan setelah rekayasa kebutuhan didapat. Bab ini dibagi menjadi dua bagian, yaitu perancangan dan implementasi.

5.1 Perancangan

Pada sub bab ini akan dijelaskan mengenai perancangan dari aplikasi FM App yang terdiri dari perancangan arsitektural, perancangan komponen, perancangan basis data, dan perancangan antarmuka.

5.1.1 Perancangan Arsitektural

FM App menggunakan arsitektur MVC yang bertujuan untuk memudahkan saat *maintenance* aplikasi. MVC memudahkan pengembangan karena apabila ada perubahan, perubahan akan terjadi pada *business logic* yang tidak memengaruhi pada *presentation code*. Untuk lebih jelasnya dapat dilihat pada gambar 5.1 sebagai berikut.



Gambar 5. 1 Arsitektur FM App

Berikut penjelasan mengenai masing – masing entitas pada gambar 5.1

1. Pengguna

Pengguna merupakan aktor yang berinteraksi dengan aplikasi. Pengguna dapat menggunakan semua fitur yang ada pada aplikasi. Pengguna dapat melihat seluruh tampilan dari *view* yang ditampilkan oleh aplikasi dan dapat berinteraksi dengan *view* tersebut dengan menggunakan *controller* yang menjadikan interaksi pengguna menjadi suatu proses

2. Model

Model merupakan bagian untuk memanipulasi data dari aplikasi. *Model* pada aplikasi ini adalah firebase. Hasil proses dari *controller* akan dimanipulasi ke dalam *model* dan *model* akan memperbarui nilai *view*

3. View

View merupakan bagian representasi visual dari aplikasi. *View* dapat berubah apabila ada perubahan dari *model*. Selain itu, *view* yang menjadi perantara antara pengguna dengan sistem dari aplikasi.

4. Controller

Controller merupakan bagian dari perantara antara *view* dengan *model*. *Controller* yang akan meneruskan interaksi pengguna terhadap *view* ke *model*. *Controller* juga memanipulasi nilai ke *model*, sehingga dapat memperbarui *view*.

5.1.1.1 Class Diagram

(a) Perancangan Umum

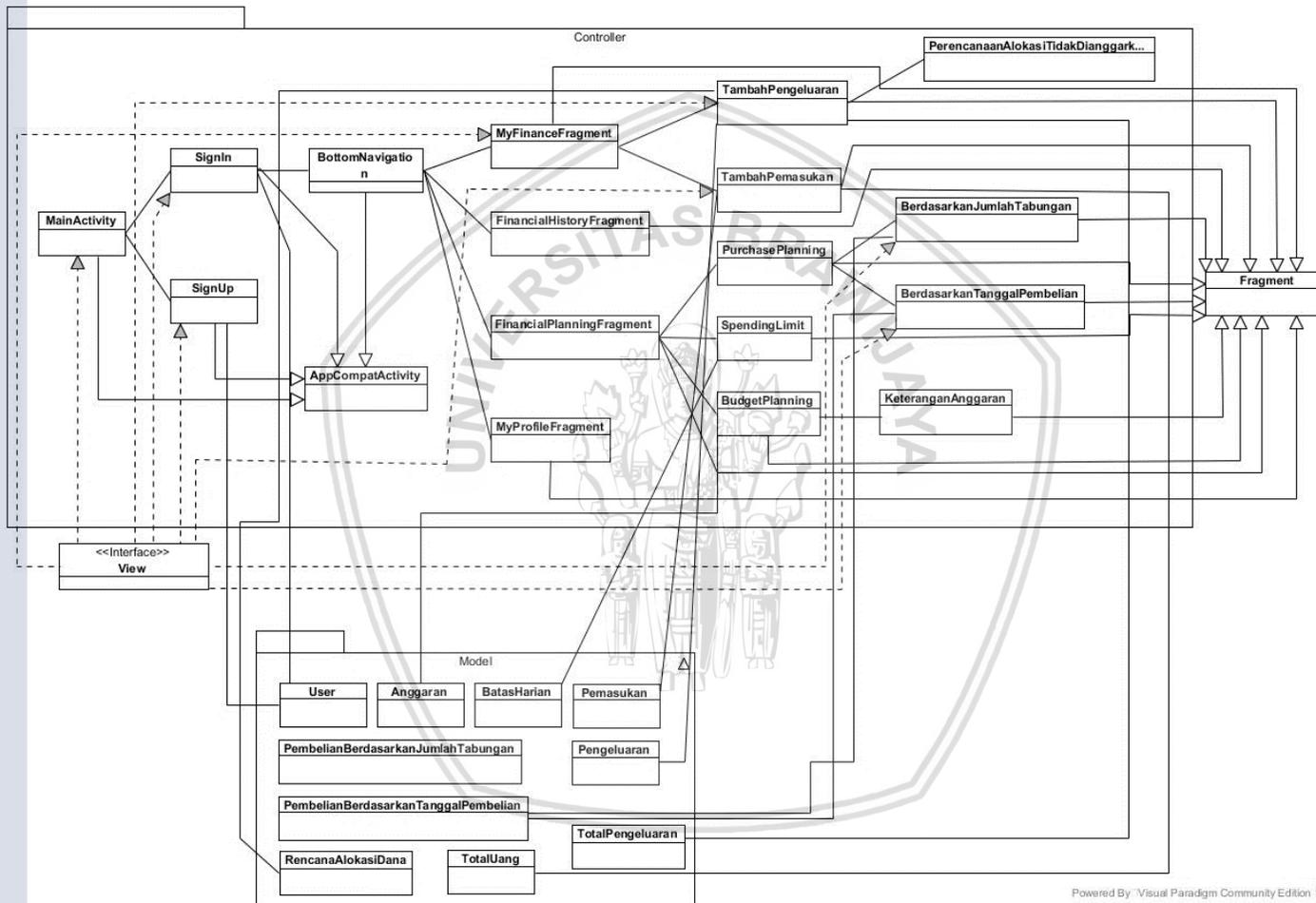
Perancangan umum merupakan rancangan umum klas diagram yang terdiri dari *package* klas yang berisi nama klas dan hubungannya.

(i) Sprint 1

Pada perancangan umum klas diagram, terdapat 9 klas pada *package controller* dan 7 klas pada *package model* yang dibuat pada *sprint 1*. Klas tersebut dirancang untuk menjalankan fitur dasar FM App yang merupakan *sprint goal* dari *sprint 1*. Untuk lebih jelasnya terkait perancangan umum klas diagram dapat dilihat pada gambar 5.2.

(ii) Sprint 2

Pada perancangan umum klas diagram, terdapat 8 klas pada *package controller* dan 4 klas pada *package model* yang dibuat pada *sprint 1*. Klas tersebut dirancang untuk menjalankan fitur lanjutan FM App yang merupakan *sprint goal* dari *sprint 2*. Untuk lebih jelasnya terkait perancangan umum klas diagram dapat dilihat pada gambar 5.2.



Powered By Visual Paradigm Community Edition

Gambar 5. 2 Class Diagram Umum pada FM App

(b) Perancangan *Detail*

Perancangan *detail* merupakan rancangan *detail* klas diagram yang terdiri dari *package* klas yang berisi klas dengan atribut dan *method*.

(i) *Sprint 1*

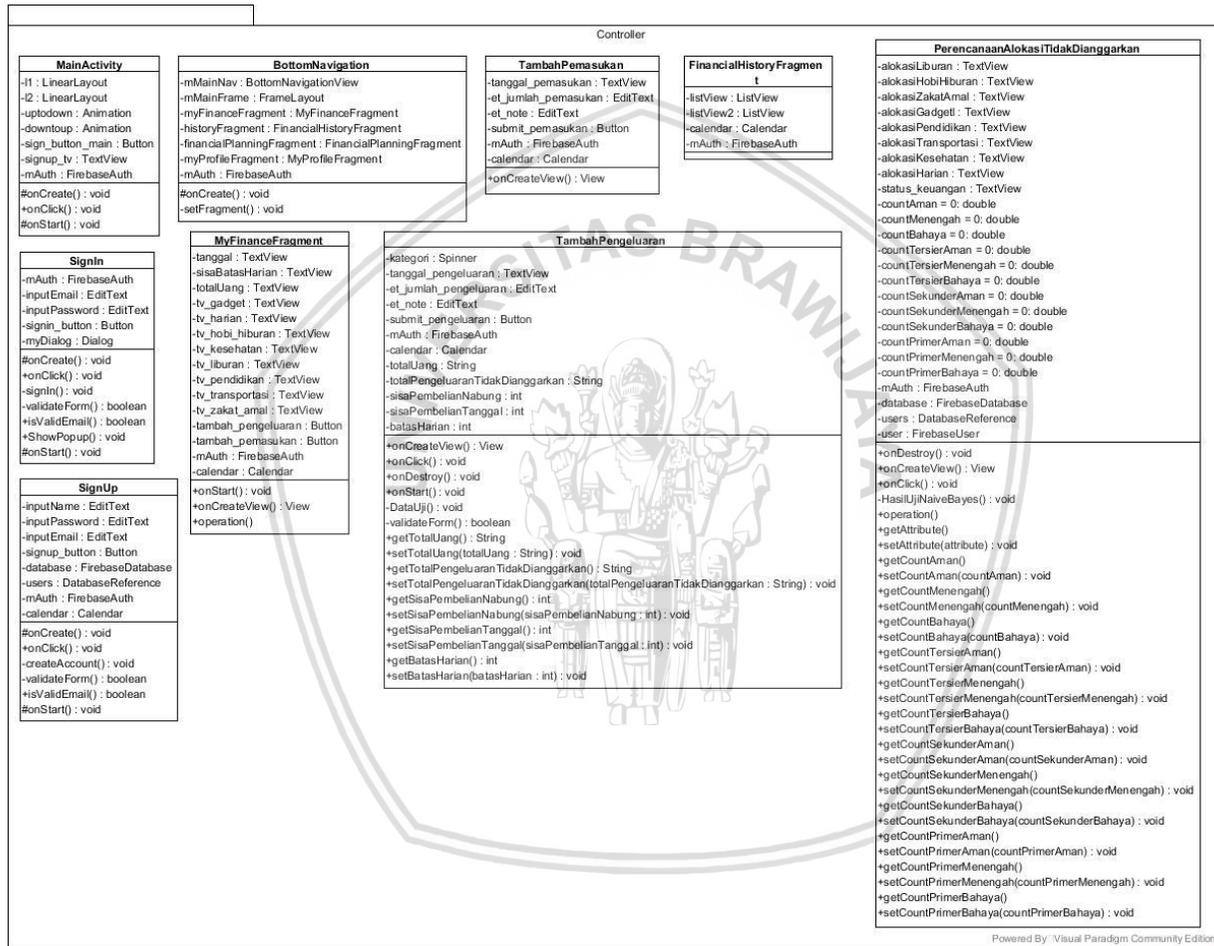
Pada *sprint 1*, perancangan *detail* yang akan dijelaskan adalah *package* yang ada pada perancangan umum klas diagram. *Package* tersebut terdiri dari *package controller* dan *model*. Fungsi utama dari masing – masing klas pada *package* tersebut akan dijelaskan pada perancangan *detail* ini.

1. *Package Controller*

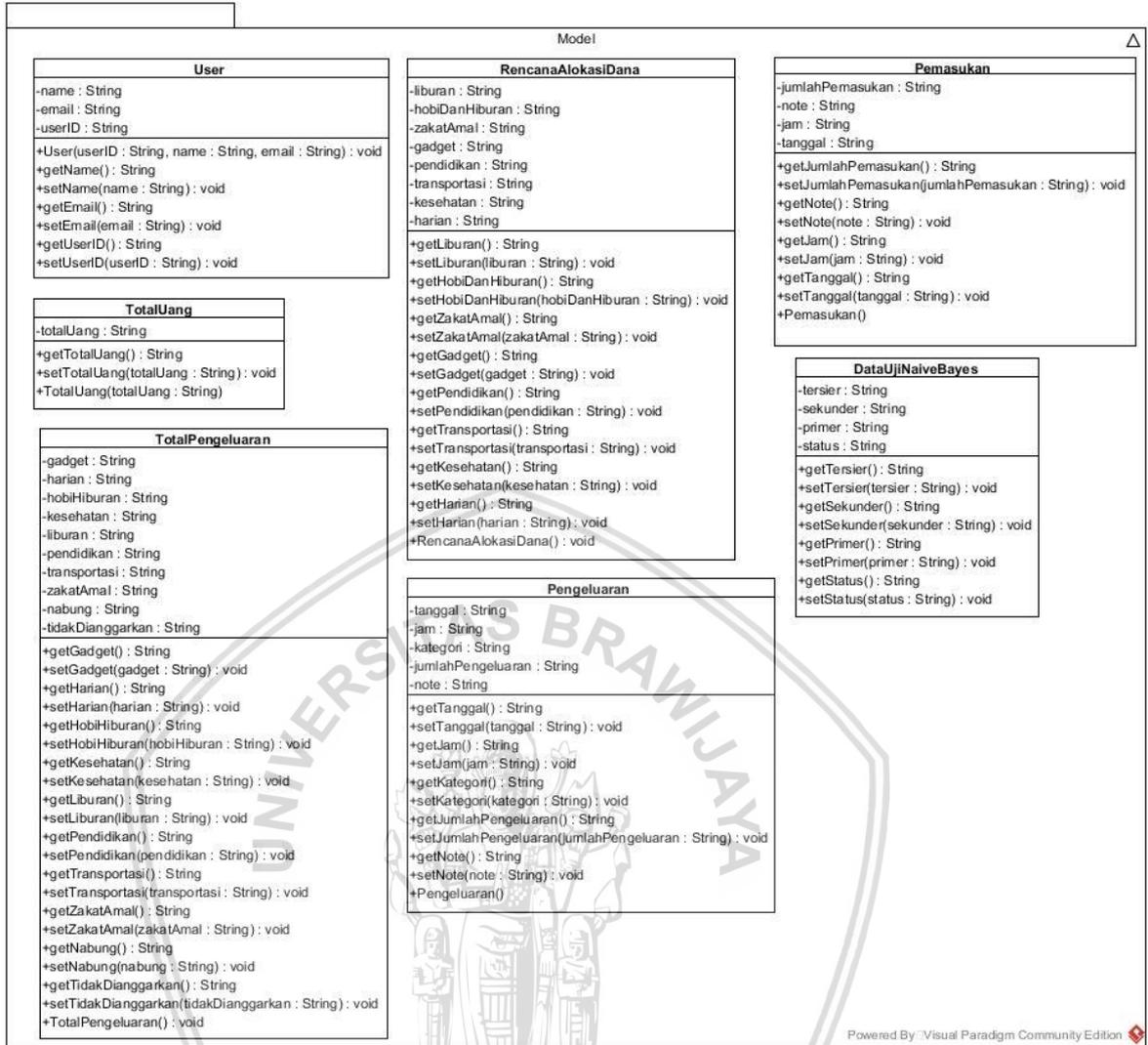
Pada *package controller* terdapat 9 klas, yaitu klas `MainActivity` untuk menampilkan halaman utama, klas `SignIn` untuk menampilkan halaman *sign in* dan masuk kedalam *system*, klas `SignUp` untuk menampilkan halaman sign up dan daftar sebagai member, klas `BottomNavigation` untuk pengaturan tombol navigasi tampilan, klas `MyFinanceFragment` untuk menampilkan halaman Keuangan Saya, klas `TambahPemasukan` untuk menampilkan halaman Menambah Pemasukan, klas `TambahPengeluaran` untuk menampilkan halaman Menambah Pengeluaran, klas `PerencanaanAlokasiTidakDianggarkan` untuk menampilkan halaman Perencanaan Alokasi Tidak Dianggarkan, klas `FinancialHistoryFragment` untuk menampilkan halaman Riwayat Keuangan. Untuk lebih jelasnya dapat dilihat pada gambar 5.3.

2. *Package Model*

Pada *package model* terdapat 7 klas, yaitu klas `User` digunakan untuk menyimpan data pengguna ke firebase, klas `TotalUang` digunakan untuk menyimpan data total uang yang dimiliki *member*, klas `TotalPengeluaran` untuk menyimpan total pengeluaran *member* berdasarkan kategori, klas `RencanaAlokasiDana` untuk menyimpan rencana alokasi dana jika pengeluaran tidak dianggarkan/tidak terduga dilakukan, klas `Pengeluaran` untuk menyimpan data ketika *member* melakukan pencatatan pengeluaran, klas `Pemasukan` untuk menyimpan data ketika *member* melakukan pencatatan pemasukan, dan klas `DataUjiNaiveBayes` untuk menyimpan data status keuangan ketika pengeluaran tidak dianggarkan diterapkan. Untuk lebih jelasnya dapat melihat gambar 5.4.



Gambar 5. 3 Package Controller Sprint 1



Gambar 5. 4 Package Model Sprint 1

(ii) Sprint 2

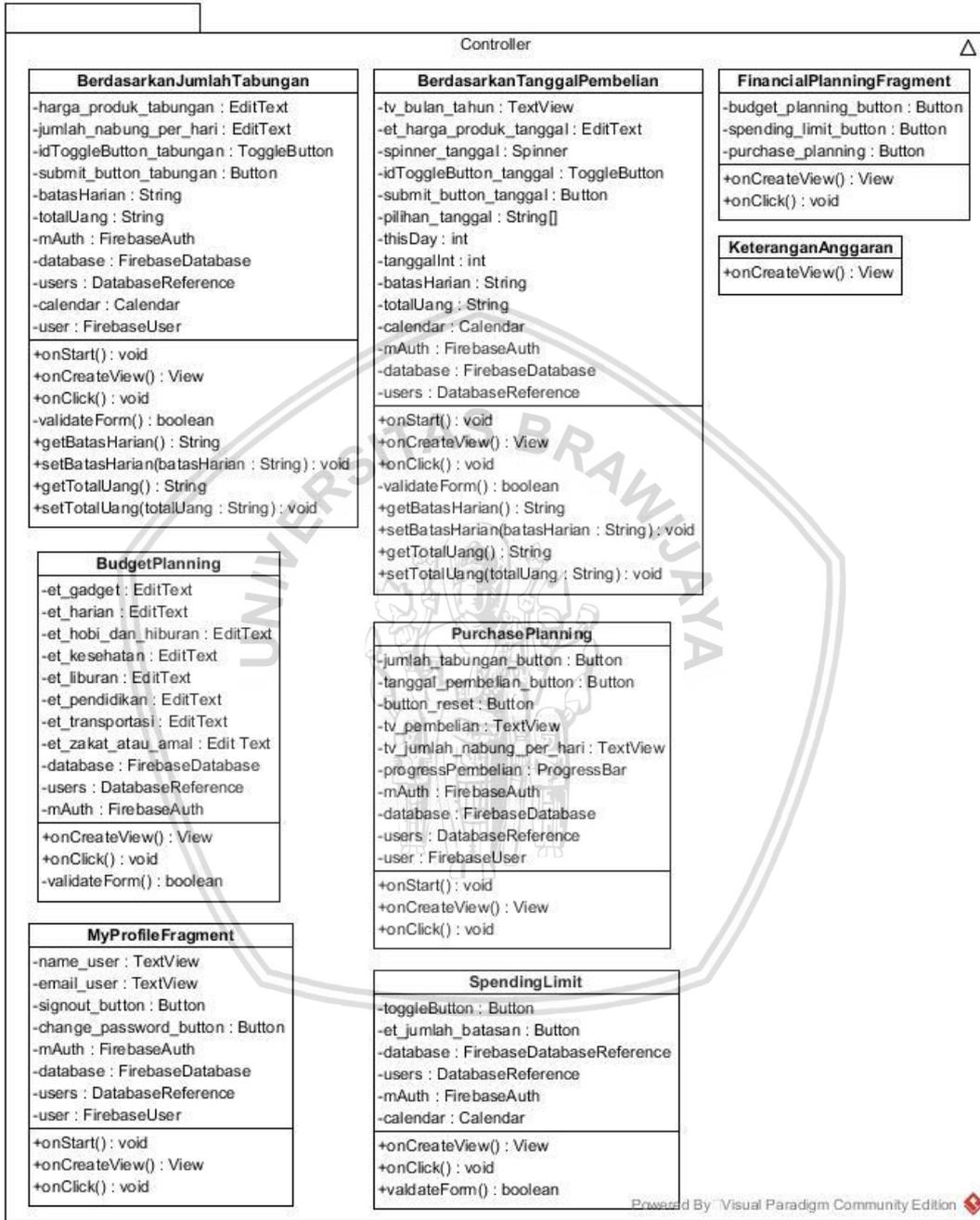
Pada *sprint 2*, perancangan *detail* yang akan dijelaskan adalah *package* yang ada pada perancangan umum klas diagram. *Package* tersebut terdiri dari *package controller* dan *model*. Fungsi utama dari masing – masing klas pada *package* tersebut akan dijelaskan pada perancangan *detail* ini.

1. Package Controller

Pada *package controller* terdapat 8 klas, yaitu klas *FinancialPalnningFragment* untuk menyediakan pilihan perencanaan kelola keuangan, klas *BudgetPlanning* untuk mengalokasikan dana anggaran perbulan pada kategori yang tersedia, klas *KeteranganAnggaran* untuk menyediakan penjelasan terhadap *member* terakit kategori alokasi anggaran, klas *SpendingLimit* untuk menentukan batas pengeluaran harian, klas *PurchasePlanning* untuk menyediakan pilihan perencanaan pembelian dengan keuangan yang dimiliki, klas *BerdasarkanJumlahTabungan* untuk membuat



perencanaan pembelian berdasarkan jumlah nabung setiap harinya, klas BerdasarkanTanggalPembelian untuk membuat perencanaan pembelian berdasarkan tanggal pembelian, klas MyProfileFragment untuk melihat data profil *member* yang sudah *sign in*. Untuk lebih jelasnya dapat melihat gambar 5.5.



Gambar 5. 5 Package Controller Sprint 2

2. Package Model

Pada *package model* terdapat 4 klas, yaitu klas Anggaran digunakan untuk menyimpan data alokasi anggaran setelah alokasi dana anggaran telah dilakukan, klas BatasHarian digunakan untuk menyimpan data batas harian setelah *member* mengatur batas harian, klas



PembelianBerdasarkanJumlahTabungan digunakan untuk menyimpan data setelah perencanaan pembelian berdasarkan jumlah tabungan dilakukan, klas PembelianBerdasarkanTanggalPembelian digunakan untuk menyimpan data setelah perencanaan pembelian berdasarkan tanggal pembelian dilakukan. Untuk lebih jelasnya dapat melihat gambar 5.6.



Gambar 5. 6 Package Model Sprint 2

5.1.1.2 Sequence Diagram

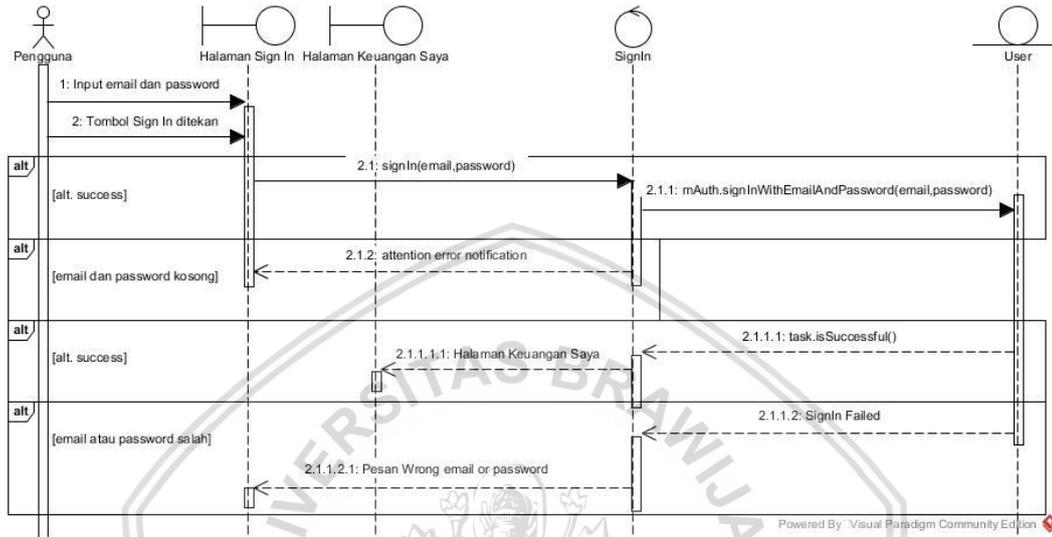
Pada sequence diagram akan dibagi menjadi dua bagian, yaitu sequence diagram pada sprint 1 dan sprint 2. Sequence diagram yang ditampilkan merupakan sequence diagram dari FM App. Sequence diagram yang dituliskan

pada bagian ini tidak semuanya, namun beberapa *sequence diagram* yang dianggap penting dalam aplikasi ini.

(a) Sprint 1

(i) Sequence Diagram Sign In

Sequence diagram ini digunakan untuk melakukan *sign in*.

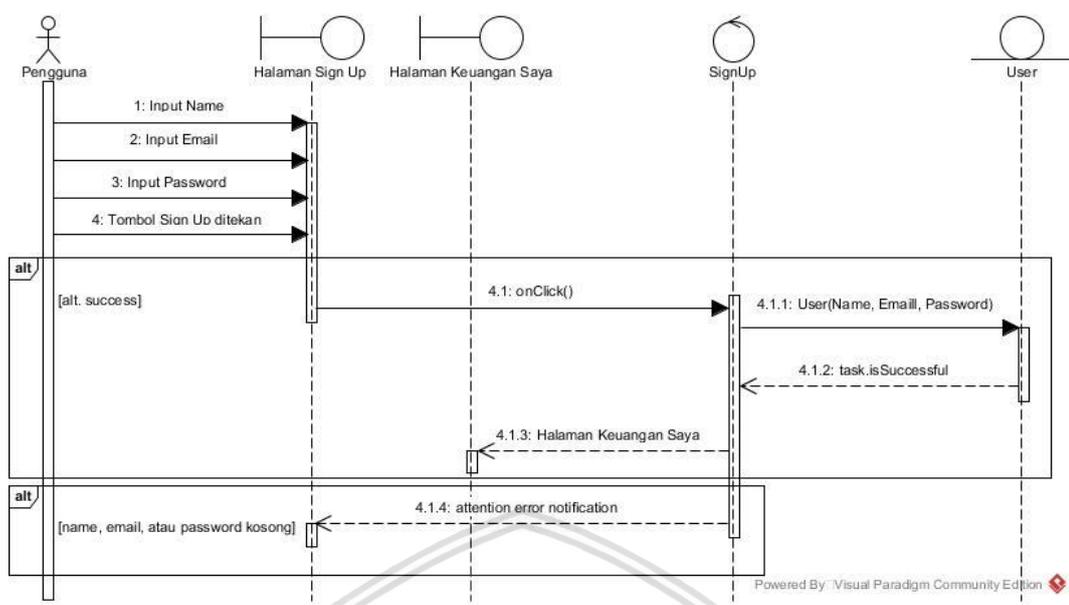


Gambar 5. 7 Sequence Diagram Sign In

Gambar 5.7 menjelaskan bahwa proses *sign in* berawal dari pengguna menginputkan *email* dan *password* serta menekan tombol *sign in* pada halaman Sign In, apabila saat menekan tombol *sign in* *email* dan *password* kosong, maka ditampilkan *attention error notification* dihalaman Sign In. Kemudian *system* memrosesnya dengan memanggil *method* `signIn(email, password)`. Setelah itu, diteruskan ke *database* untuk dilakukan pengecekan *email* dan *password* dengan `mAuth.signInWithEmailAndPassword(email, password)`. Jika proses pengecekan berhasil, maka akan diteruskan ke halaman Keuangan Saya, sedangkan jika gagal, akan ditampilkan pesan *wrong email or password*.

(ii) Sequence Diagram Sign Up

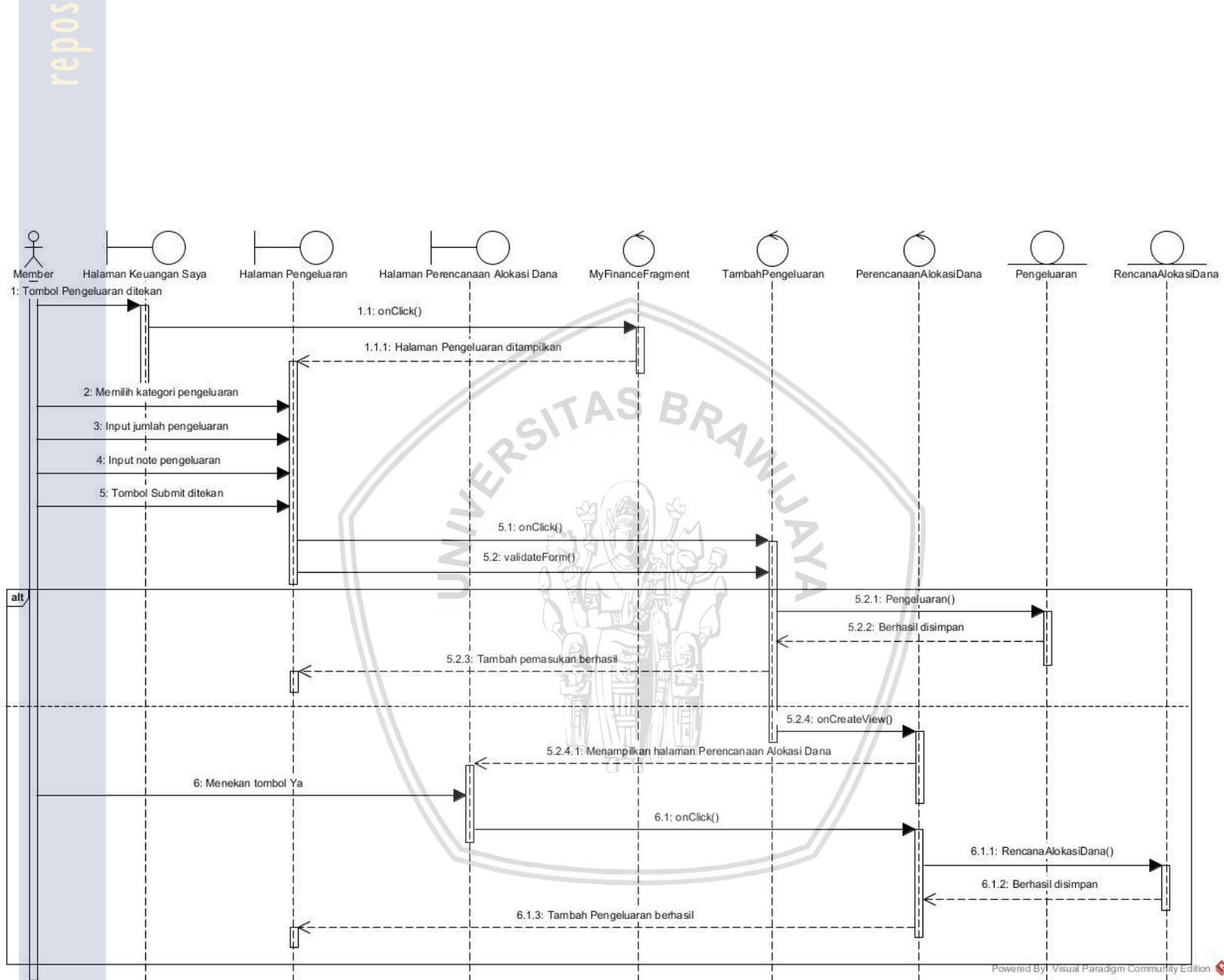
Gambar 5.8 menjelaskan bahwa proses *sign up* berawal dari pengguna menginputkan *name*, *email*, *password* serta menekan tombol *sign up* pada halaman Sign Up, apabila saat menekan tombol *sign up* *name*, *email*, atau *password* kosong, maka ditampilkan *attention error notification* dihalaman Sign Up. Kemudian *system* memrosesnya dengan memanggil *method* `onClick()`. Setelah itu, diteruskan ke *database* untuk dilakukan pendaftaran *member* dengan *name*, *email* dan *password* dengan `User(Name, Email, Password)`. Jika proses pendaftaran berhasil, maka akan diteruskan ke halaman Keuangan Saya.



Gambar 5. 8 Sequence Diagram Sign Up

(iii) Sequence Diagram Menambah Pengeluaran

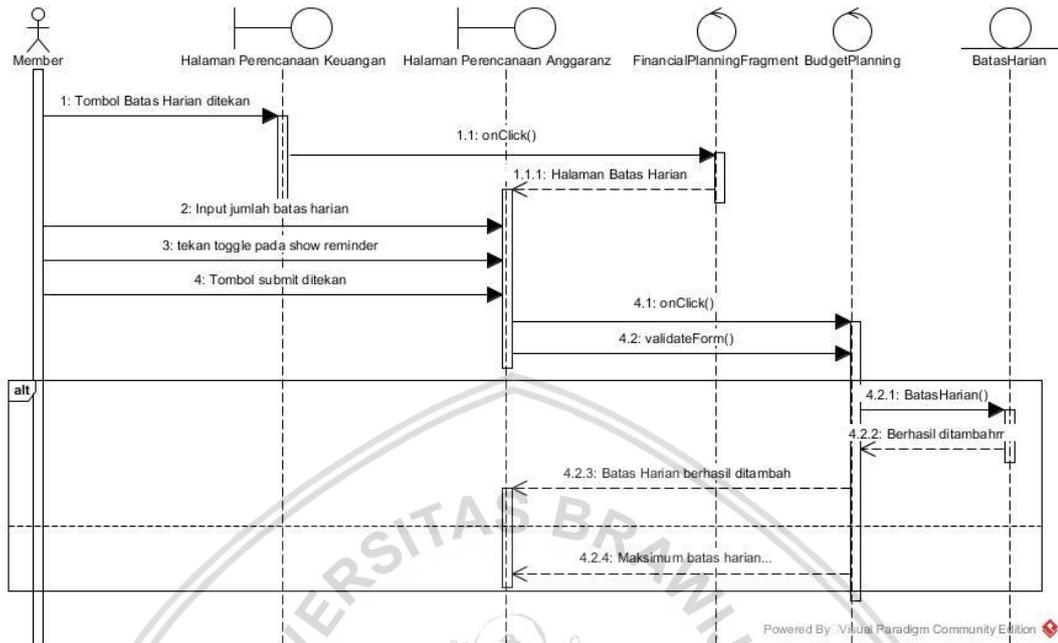
Gambar 5.9 menjelaskan tentang proses menambah pengeluaran. Langkah awal yang dilakukan adalah menekan tombol pengeluaran pada halaman keuangan saya. Lalu, method onClick() pada controller MyFinanceFragment memrosesnya. Setelah diproses oleh controller MyFinanceFragment, maka akan ditampilkan halaman pengeluaran. Pada halaman pengeluaran, member memilih kategori pengeluaran, memasukan jumlah pengeluaran, memasukan note pengeluaran, dan menekan tombol submit. Setelah tombol submit ditekan, method onClick() dan validateForm() pada controller TambahPengeluaran memrosesnya. Jika nilai dari validateForm() benar, maka akan data pengeluaran akan disimpan dengan memroses method Pengeluaran() pada model Pengeluaran(), dan member akan ditampilkan pesan "Tambah pengeluaran berhasil". Akan tetapi, jika pada pemilihan kategori dipilih kategori Tidak Dianggarkan maka, akan dialihkan ke halaman perencanaan alokasi dana dengan mengakses onCreateView() pada controller PerencanaanAlokasiDana. Jika member menekan tombol Ya pada halaman perencanaan alokasi dana, maka method onClick() pada controller PerencanaanAlokasiDana diproses. Lalu, method RencanaAlokasiDana() diproses pada model RencanaAlokasiDana. Setelah itu, halaman pengeluaran dan pesan "Tambah Pengeluaran berhasil!" ditampilkan



Gambar 5. 9 Sequence Diagram Menambah Pengeluaran

(b) Sprint 2

(i) Sequence Diagram Membuat Batas Harian



Gambar 5. 10 Sequence Diagram Membuat Batas Harian

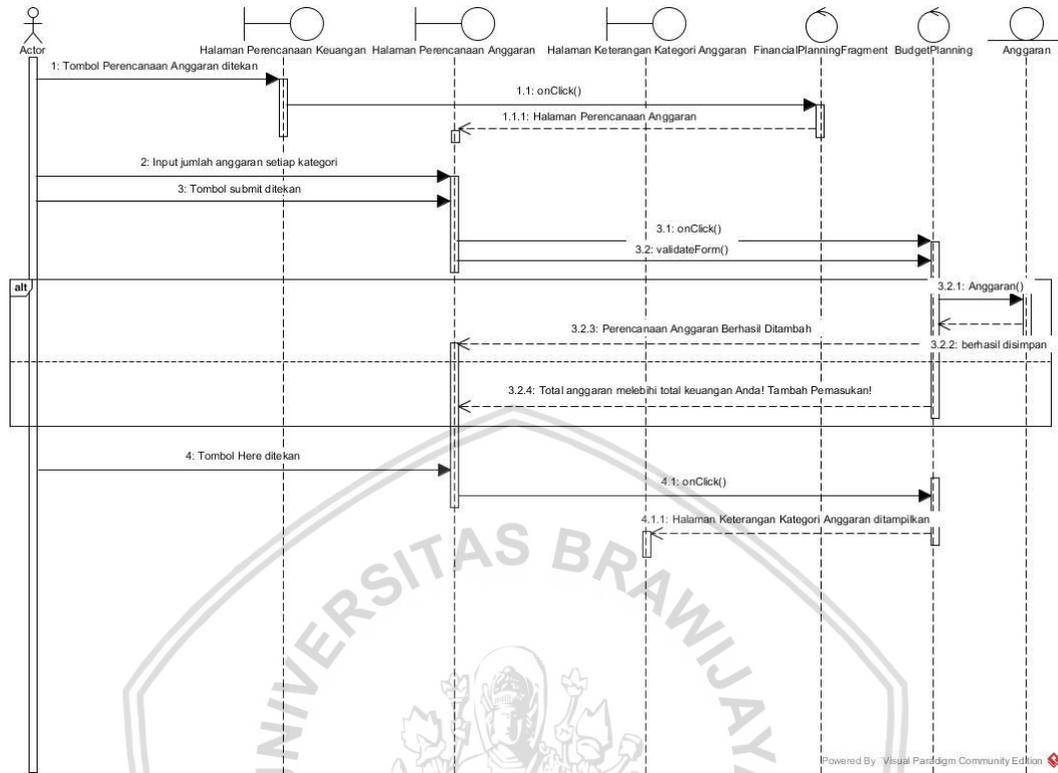
Gambar 5.10 menjelaskan bahwa proses membuat Batas Harian berawal dari *member* menekan tombol batas harian pada halaman Perencanaan Keuangan. Kemudian system memrosesnya dengan memanggil *method* `onClick()`. Setelah itu, halaman Batas Harian ditampilkan. *Member* memasukkan jumlah batas harian, menekan tombol *toggle* pada *show reminder*, serta tombol *submit* ditekan. *Method* `onClick()` dan *method* `validateForm()` pada *controller* `BudgetPlanning` memrosesnya. Jika nilai pada `validateForm()` benar, maka akan diteruskan ke *model* `BatasHarian` dengan memroses *method* `BatasHarian()` dan menampilkan pesan “Batas Harian berhasil ditambah”. Akan tetapi, jika `validateForm()` bernilai salah, maka akan ditampilkan peringatan “.Maksimum batas harian...”.

(ii) Sequence Diagram Membuat Perencanaan Anggaran

Gambar 5.11 menjelaskan bahwa proses membuat perencanaan Anggaran berawal dari tombol perencanaan anggaran ditekan pada halam perencanaan keuangan, lalu *method* `onClick()` pada *Controller* `FinancialPlanningFragment` memrosesnya. Setelah diproses, maka ditampilkan halaman perencanaan anggaran. Aktor menginputkan jumlah anggaran setiap kategori dan tombol *submit* ditekan. *Method* `onClick()` dan `validateForm()` meneruskan interaksi aktor tersebut. Jika tervalidasi dengan benar, maka akan diteruskan ke *Model* `Anggaran()`, dan aktor akan ditampilkan pesan Perencanaan Anggaran berhasil ditambah. Akan tetapi, jika tidak tervalidasi dengan benar, maka akan ditampilkan “Total anggaran melebihi total keuangan Anda! Tambah Pemasukan!”. Selain itu, jika tombol *here* ditekan pada halaman



perencanaan anggaran, maka akan ditampilkan halaman keterangan kategori anggaran.



Gambar 5. 11 Sequence Diagram Membuat Perencanaan Anggaran

5.1.2 Perancangan Komponen

Pada tahap ini, akan dijelaskan tentang gambaran rincian dari sub – system komponen perangkat lunak. Perencanaan komponen ini terdiri dari struktur data dan rincian algoritma pada suatu komponen. Pada perencanaan komponen ini akan dibagi menjadi dua, yaitu perencanaan komponen pada *sprint 1* dan *sprint 2*.

5.1.2.1 Sprint 1

(a) Perancangan Komponen Sign In

Nama Klas : SignIn

Nama Method: signIn()

Deskripsi : *method* ini dijalankan ketika pengguna masuk kedalam *system* sebagai *member* dan ingin membuka halaman Batas Harian. Untuk lebih jelasnya dapat dilihat pada tabel 5.1.

Tabel 5. 1 Algoritma Sign In

Pengguna menekan tombol Sign In pada halaman Utama Halaman Sign In ditampilkan Pengguna memasukan email Pengguna memasukan password
--

```

Tombol Sign In ditekan
    Sistem memvalidasi masukan pada email dan password
    Jika validasi benar
        Email dan password dicek pada database
        Jika email dan password ada pada database
            Sign in success ditampilkan
            Halaman keuangan saya ditampilkan
        Jika email dan password tidak ada pada database
            Wrong email or password ditampilkan
    
```

(b) Perancangan Komponen Tambah Pengeluaran Tidak Dianggarkan

Nama Klas : TambahPengeluaran

Nama Method : onClick()

Deskripsi : *method* ini dijalankan ketika pengguna masuk kedalam *system* sebagai *member* dan menambah pengeluaran. Untuk lebih jelasnya dapat dilihat pada tabel 5.2.

Tabel 5. 2 Algoritma Tambah Pengeluaran Tidak Dianggarkan

```

Pengguna masuk ke system sebagai member
Member menambah pengeluaran
Member memilih kategori tidak dianggarkan
Member memasukan jumlah pengeluaran
Member memasukan note
Member menekan tombol submit
    Jika validasi masukan benar
        Jika nilai sisa batas harian null
            Merencanakan alokasi dana pada kategori lainnya
            Merencanakan status keuangan setelah perencanaan alokasi dana digunakan
            Menampilkan halaman perencanaan alokasi tidak dianggarkan
                Rekomendasi alokasi dana ditampilkan
                Status keuangan setelah perencanaan alokasi dana digunkana ditampilkan
                Rekomendasi alokasi dana ditampilkan digunakan
                Tambah pengeluaran berhasil
        Jika nilai sisa batas harian tidak null
            Merencanakan alokasi dana pada kategori lainnya
            Merencanakan status keuangan setelah perencanaan alokasi dana digunakan
            Menampilkan halaman perencanaan alokasi tidak dianggarkan
    
```

```

Rekomendasi alokasi dana ditampilkan
Status keuangan setelah perencanaan alokasi
dana digunakan ditampilkan
Rekomendasi alokasi dana ditampilkan
digunakan
Tambah pengeluaran berhasil
Jikat validasi masukan salah
Informasi Error ditampilkan

```

5.1.2.2 Sprint 2

(a) Perancangan Komponen Membuat Batas Harian

Nama Klas : SpendingLimit

Nama *Method*: onClick()

Deskripsi : *method* ini dijalankan ketika pengguna masuk kedalam *system* sebagai *member* dan ingin membuat batasan pengeluaran harian. Untuk lebih jelasnya dapat dilihat pada tabel 5.3.

Tabel 5. 3 Algoritma Membuat Batas Harian

```

Member memasukan jumlah batas harian
Member menekan toggle show reminder
Member menekan tombol submit
    Jika validasi masukan bernilai benar
        Data masukan batas harian disimpan di database
    Jika validasi masukan bernilai salah
        Member tidak memasukan jumlah batasn
        Member menekan tombol submit

```

(b) Perancangan Komponen Melihat Halaman Keterangan Kategori Anggaran

Nama Klas : BudgetPlanning

Nama *Method*: onClick()

Deskripsi : *method* ini dijalankan ketika pengguna masuk kedalam *system* sebagai *member* dan membuka halaman Keterangan Kategori Anggaran. Untuk lebih jelasnya dapat dilihat pada tabel 5.2.

Tabel 5. 4 Algoritma Melihat Halaman Keterangan Kategori Anggaran

```

Pengguna masuk ke system sebagai member
Tombol icon perencanaan keuangan ditekan
    Halaman Perencanaan Keuangan ditampilkan
Tombol Perencanaan Anggaran ditekan
    Halaman Perencanaan Anggaran ditampilkan
Tombol keterangan kategori anggaran ditekan
    Halaman Keterangan Kategori Anggaran ditampilkan

```

5.1.3 Perancangan Basis Data

Pada perancangan ini menjelaskan tentang perancangan pertukaran antar data firebase *realtime database*. Pertukaran data ini akan berbentuk skema JSON yang nantinya akan diimplementasikan ke dalam struktur data firebase. Untuk lebih jelasnya dapat dilihat pada tabel 5.5.

Tabel 5. 5 Perancangan Data FM App

1	{
2	"\$id": "http://example.com/example.json",
3	"type": "object",
4	"definitions": {},
5	"\$schema": "http://json-schema.org/draft-07/schema#",
6	"properties": {
7	"Users": {
8	"\$id": "/properties/Users",
9	"type": "object",
10	"properties": {
11	"9E58P5Pms0YhLhfIPo512frprEq2": {
12	"\$id":
13	"/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2",
14	"type": "object",
15	"properties": {
16	"Anggaran": {
17	"\$id":
18	"/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
19	ies/Anggaran",
20	"type": "object",
21	"properties": {
22	"gadget": {
23	"\$id":
24	"/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
25	ies/Anggaran/properties/gadget",
26	"type": "string",
27	"title": "The Gadget Schema ",
28	"default": "",
29	"examples": [
30	"0"
31]
32	},
33	"harian": {
34	"\$id":
35	"/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
36	ies/Anggaran/properties/harian",
37	"type": "string",
38	"title": "The Harian Schema ",
39	"default": "",
40	"examples": [
41	"0"
42]
43	},
44	"hobiDanHiburan": {
45	"\$id":
46	"/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
47	ies/Anggaran/properties/hobiDanHiburan",
48	"type": "string",
49	"title": "The Hobidanhiburan Schema ",
50	"default": "",
51	"examples": [
52	"0"
53]
54	},
55	"kesehatan": {



```

56         "$id":
57         "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
58         ies/Anggaran/properties/kesehatan",
59         "type": "string",
60         "title": "The Kesehatan Schema ",
61         "default": "",
62         "examples": [
63             "0"
64         ]
65     },
66     "liburan": {
67         "$id":
68         "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
69         ies/Anggaran/properties/liburan",
70         "type": "string",
71         "title": "The Liburan Schema ",
72         "default": "",
73         "examples": [
74             "0"
75         ]
76     },
77     "pendidikan": {
78         "$id":
79         "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
80         ies/Anggaran/properties/pendidikan",
81         "type": "string",
82         "title": "The Pendidikan Schema ",
83         "default": "",
84         "examples": [
85             "0"
86         ]
87     },
88     "transportasi": {
89         "$id":
90         "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
91         ies/Anggaran/properties/transportasi",
92         "type": "string",
93         "title": "The Transportasi Schema ",
94         "default": "",
95         "examples": [
96             "0"
97         ]
98     },
99     "zakatAtauAmal": {
100         "$id":
101         "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
102         ies/Anggaran/properties/zakatAtauAmal",
103         "type": "string",
104         "title": "The Zakatatauamal Schema ",
105         "default": "",
106         "examples": [
107             "0"
108         ]
109     }
110 }
111 },
112     "Batas Harian": {
113         "$id":
114         "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
115         ies/Batas Harian",
116         "type": "object",
117         "properties": {
118             "jumlahBatasan": {

```

```

119         "$id":
120         "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
121         ies/Batas Harian/properties/jumlahBatasan",
122         "type": "string",
123         "title": "The Jumlahbatasan Schema ",
124         "default": "",
125         "examples": [
126         "0"
127         ]
128     },
129     "showReminder": {
130         "$id":
131         "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
132         ies/Batas Harian/properties/showReminder",
133         "type": "boolean",
134         "title": "The Showreminder Schema ",
135         "default": false,
136         "examples": [
137         false
138         ]
139     },
140     "sisabatasHarian": {
141         "$id":
142         "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
143         ies/Batas Harian/properties/sisabatasHarian",
144         "type": "string",
145         "title": "The Sisabatasharian Schema ",
146         "default": "",
147         "examples": [
148         "0"
149         ]
150     },
151     "tanggal": {
152         "$id":
153         "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
154         ies/Batas Harian/properties/tanggal",
155         "type": "string",
156         "title": "The Tanggal Schema ",
157         "default": "",
158         "examples": [
159         "Jul 19, 2018"
160         ]
161     }
162 }
163 },
164 "Data Set": {
165     "$id":
166     "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
167     ies/Data Set",
168     "type": "object",
169     "properties": {
170         "Feature": {
171             "$id":
172             "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
173             ies/Data Set/properties/Feature",
174             "type": "object",
175             "properties": {
176                 "Primer": {
177                     "$id":
178                     "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
179                     ies/Data Set/properties/Feature/properties/Primer",
180                     "type": "array",
181                     "items": {

```

```

182         "$id":
183         "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
184         ies/Data Set/properties/Feature/properties/Primer/items",
185         "type": "null",
186         "title": "The 0th Schema ",
187         "default": null,
188         "examples": [
189             null,
190             "Tidak",
191             "Tidak",
192             "Tidak",
193             "Ya",
194             "Tidak",
195             "Ya",
196             "Tidak",
197             "Ya"
198         ]
199     }
200 },
201     "Sekunder": {
202         "$id":
203         "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
204         ies/Data Set/properties/Feature/properties/Sekunder",
205         "type": "array",
206         "items": {
207             "$id":
208             "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
209             ies/Data Set/properties/Feature/properties/Sekunder/items",
210             "type": "null",
211             "title": "The 0th Schema ",
212             "default": null,
213             "examples": [
214                 null,
215                 "Tidak",
216                 "Tidak",
217                 "Tidak",
218                 "Ya",
219                 "Ya",
220                 "Ya",
221                 "Ya",
222                 "Ya"
223             ]
224         }
225     },
226     "Status": {
227         "$id":
228         "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
229         ies/Data Set/properties/Feature/properties/Status",
230         "type": "array",
231         "items": {
232             "$id":
233             "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
234             ies/Data Set/properties/Feature/properties/Status/items",
235             "type": "null",
236             "title": "The 0th Schema ",
237             "default": null,
238             "examples": [
239                 null,
240                 "Aman",
241                 "Aman",
242                 "Aman",
243                 "Bahaya",
244                 "Menengah",
245                 "Bahaya",
246                 "Menengah",

```

```

247         "Bahaya"
248     ]
249 }
250 },
251 "Tersier": {
252     "$id":
253     "/properties/Users/properties/9E58P5PMs0YhLhfIPo512frprEq2/propert
254     ies/Data Set/properties/Feature/properties/Tersier",
255     "type": "array",
256     "items": {
257         "$id":
258         "/properties/Users/properties/9E58P5PMs0YhLhfIPo512frprEq2/propert
259         ies/Data Set/properties/Feature/properties/Tersier/items",
260         "type": "null",
261         "title": "The 0th Schema ",
262         "default": null,
263         "examples": [
264             null,
265             "Ya",
266             "Ya",
267             "Ya",
268             "Ya",
269             "Ya",
270             "Ya",
271             "Ya",
272             "Ya"
273         ]
274     }
275 }
276 }
277 }
278 }
279 },
280 "Data Uji": {
281     "$id":
282     "/properties/Users/properties/9E58P5PMs0YhLhfIPo512frprEq2/propert
283     ies/Data Uji",
284     "type": "object",
285     "properties": {
286         "primer": {
287             "$id":
288             "/properties/Users/properties/9E58P5PMs0YhLhfIPo512frprEq2/propert
289             ies/Data Uji/properties/primer",
290             "type": "string",
291             "title": "The Primer Schema ",
292             "default": "",
293             "examples": [
294                 "Tidak"
295             ]
296         },
297         "sekunder": {
298             "$id":
299             "/properties/Users/properties/9E58P5PMs0YhLhfIPo512frprEq2/propert
300             ies/Data Uji/properties/sekunder",
301             "type": "string",
302             "title": "The Sekunder Schema ",
303             "default": "",
304             "examples": [
305                 "Tidak"
306             ]
307         },
308         "status": {
309             "$id":
310             "/properties/Users/properties/9E58P5PMs0YhLhfIPo512frprEq2/propert
311             ies/Data Uji/properties/status",

```

```

312         "type": "string",
313         "title": "The Status Schema ",
314         "default": "",
315         "examples": [
316             "Aman"
317         ]
318     },
319     "tersier": {
320         "$id":
321         "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
322         ies/Data Uji/properties/tersier",
323         "type": "string",
324         "title": "The Tersier Schema ",
325         "default": "",
326         "examples": [
327             "Ya"
328         ]
329     }
330 }
331 },
332 "Hasil Status": {
333     "$id":
334     "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
335     ies/Hasil Status",
336     "type": "object",
337     "properties": {
338         "status": {
339             "$id":
340             "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
341             ies/Hasil Status/properties/status",
342             "type": "string",
343             "title": "The Status Schema ",
344             "default": "",
345             "examples": [
346                 "-"
347             ]
348         }
349     }
350 },
351 "Pemasukan": {
352     "$id":
353     "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
354     ies/Pemasukan",
355     "type": "object",
356     "properties": {
357         "Jul 19, 2018": {
358             "$id":
359             "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
360             ies/Pemasukan/properties/Jul 19, 2018",
361             "type": "object",
362             "properties": {
363                 "-LHljRu7wYF48o28XFuq": {
364                     "$id":
365                     "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
366                     ies/Pemasukan/properties/Jul 19, 2018/properties/-
367                     LHljRu7wYF48o28XFuq",
368                     "type": "object",
369                     "properties": {
370                         "jam": {
371                             "$id":
372                             "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
373                             ies/Pemasukan/properties/Jul 19, 2018/properties/-
374                             LHljRu7wYF48o28XFuq/properties/jam",
375                             "type": "string",
376                             "title": "The Jam Schema ",

```

```

377         "default": "",
378         "examples": [
379             "15:52:43"
380         ]
381     },
382     "jumlahPemasukan": {
383         "$id":
384         "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
385         ies/Pemasukan/properties/Jul 19, 2018/properties/-
386         LHljRu7wYF48o28XFuq/properties/jumlahPemasukan",
387         "type": "string",
388         "title": "The Jumlahpemasukan Schema ",
389         "default": "",
390         "examples": [
391             "0"
392         ]
393     },
394     "note": {
395         "$id":
396         "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
397         ies/Pemasukan/properties/Jul 19, 2018/properties/-
398         LHljRu7wYF48o28XFuq/properties/note",
399         "type": "string",
400         "title": "The Note Schema ",
401         "default": "",
402         "examples": [
403             "0"
404         ]
405     },
406     "tanggal": {
407         "$id":
408         "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
409         ies/Pemasukan/properties/Jul 19, 2018/properties/-
410         LHljRu7wYF48o28XFuq/properties/tanggal",
411         "type": "string",
412         "title": "The Tanggal Schema ",
413         "default": "",
414         "examples": [
415             "Jul 19, 2018"
416         ]
417     }
418 }
419 }
420 }
421 }
422 }
423 },
424 "Pembelian Berdasarkan Jumlah Tabungan": {
425     "$id":
426     "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
427     ies/Pembelian Berdasarkan Jumlah Tabungan",
428     "type": "object",
429     "properties": {
430         "hargaProduk": {
431             "$id":
432             "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
433             ies/Pembelian Berdasarkan Jumlah Tabungan/properties/hargaProduk",
434             "type": "string",
435             "title": "The Hargaproduk Schema ",
436             "default": "",
437             "examples": [
438                 "0"
439             ]
440         },
441         "nabungPerHari": {

```

```

442         "$id":
443         "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
444         ies/Pembelian Berdasarkan Jumlah
445         Tabungan/properties/nabungPerHari",
446         "type": "string",
447         "title": "The Nabungperhari Schema ",
448         "default": "",
449         "examples": [
450         "0"
451         ]
452     },
453     "showReminder": {
454         "$id":
455         "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
456         ies/Pembelian Berdasarkan Jumlah
457         Tabungan/properties/showReminder",
458         "type": "boolean",
459         "title": "The Showreminder Schema ",
460         "default": false,
461         "examples": [
462         false
463         ]
464     },
465     "status": {
466         "$id":
467         "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
468         ies/Pembelian Berdasarkan Jumlah Tabungan/properties/status",
469         "type": "string",
470         "title": "The Status Schema ",
471         "default": "",
472         "examples": [
473         "tidak aktif"
474         ]
475     }
476 },
477 "Pembelian Berdasarkan Tanggal Pembelian": {
478     "$id":
479     "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
480     ies/Pembelian Berdasarkan Tanggal Pembelian",
481     "type": "object",
482     "properties": {
483         "hargaProduk": {
484             "$id":
485             "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
486             ies/Pembelian Berdasarkan
487             Tanggal
488             Pembelian/properties/hargaProduk",
489             "type": "string",
490             "title": "The Hargaproduk Schema ",
491             "default": "",
492             "examples": [
493             "0"
494             ]
495         },
496         "nabung": {
497             "$id":
498             "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
499             ies/Pembelian Berdasarkan Tanggal Pembelian/properties/nabung",
500             "type": "string",
501             "title": "The Nabung Schema ",
502             "default": "",
503             "examples": [
504             "0"
505             ]
506         },

```

```

507         "showReminder": {
508             "$id":
509             "/properties/Users/properties/9E58P5PMs0YhLhfIPo512frprEq2/propert
510             ies/Pembelian Berdasarkan Tanggal
511             Pembelian/properties/showReminder",
512             "type": "boolean",
513             "title": "The Showreminder Schema ",
514             "default": false,
515             "examples": [
516                 false
517             ]
518         },
519         "status": {
520             "$id":
521             "/properties/Users/properties/9E58P5PMs0YhLhfIPo512frprEq2/propert
522             ies/Pembelian Berdasarkan Tanggal Pembelian/properties/status",
523             "type": "string",
524             "title": "The Status Schema ",
525             "default": "",
526             "examples": [
527                 "tidak aktif"
528             ]
529         },
530         "tanggal": {
531             "$id":
532             "/properties/Users/properties/9E58P5PMs0YhLhfIPo512frprEq2/propert
533             ies/Pembelian Berdasarkan Tanggal Pembelian/properties/tanggal",
534             "type": "string",
535             "title": "The Tanggal Schema ",
536             "default": "",
537             "examples": [
538                 "0"
539             ]
540         }
541     },
542     "Pengeluaran": {
543         "$id":
544         "/properties/Users/properties/9E58P5PMs0YhLhfIPo512frprEq2/propert
545         ies/Pengeluaran",
546         "type": "object",
547         "properties": {
548             "Jul 19, 2018": {
549                 "$id":
550                 "/properties/Users/properties/9E58P5PMs0YhLhfIPo512frprEq2/propert
551                 ies/Pengeluaran/properties/Jul 19, 2018",
552                 "type": "object",
553                 "properties": {
554                     "-LHljRuB-hmI5SFyW35a": {
555                         "$id":
556                         "/properties/Users/properties/9E58P5PMs0YhLhfIPo512frprEq2/propert
557                         ies/Pengeluaran/properties/Jul 19, 2018/properties/-LHljRuB-
558                         hmI5SFyW35a",
559                         "type": "object",
560                         "properties": {
561                             "jam": {
562                                 "$id":
563                                 "/properties/Users/properties/9E58P5PMs0YhLhfIPo512frprEq2/propert
564                                 ies/Pengeluaran/properties/Jul 19, 2018/properties/-LHljRuB-
565                                 hmI5SFyW35a/properties/jam",
566                                 "type": "string",
567                                 "title": "The Jam Schema ",
568                                 "default": "",
569                                 "examples": [
570                                     "15:52:43"

```

<p>572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634</p>	<pre>] }, "jumlahPengeluaran": { "\$id": "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert ies/Pengeluaran/properties/Jul 19, 2018/properties/-LHljRuB- hmI5SFyW35a/properties/jumlahPengeluaran", "type": "string", "title": "The Jumlahpengeluaran Schema ", "default": "", "examples": ["0"] }, "kategori": { "\$id": "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert ies/Pengeluaran/properties/Jul 19, 2018/properties/-LHljRuB- hmI5SFyW35a/properties/kategori", "type": "string", "title": "The Kategori Schema ", "default": "", "examples": ["0"] }, "note": { "\$id": "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert ies/Pengeluaran/properties/Jul 19, 2018/properties/-LHljRuB- hmI5SFyW35a/properties/note", "type": "string", "title": "The Note Schema ", "default": "", "examples": ["0"] }, "tanggal": { "\$id": "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert ies/Pengeluaran/properties/Jul 19, 2018/properties/-LHljRuB- hmI5SFyW35a/properties/tanggal", "type": "string", "title": "The Tanggal Schema ", "default": "", "examples": ["Jul 19, 2018"] } } } } } }, "Rencana Alokasi Dana": { "\$id": "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert ies/Rencana Alokasi Dana", "type": "object", "properties": { "gadget": { </pre>
--	--



```

635         "$id":
636         "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
637         ies/Rencana Alokasi Dana/properties/gadget",
638         "type": "string",
639         "title": "The Gadget Schema ",
640         "default": "",
641         "examples": [
642             "0"
643         ]
644     },
645     "harian": {
646         "$id":
647         "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
648         ies/Rencana Alokasi Dana/properties/harian",
649         "type": "string",
650         "title": "The Harian Schema ",
651         "default": "",
652         "examples": [
653             "0"
654         ]
655     },
656     "hobiDanHiburan": {
657         "$id":
658         "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
659         ies/Rencana Alokasi Dana/properties/hobiDanHiburan",
660         "type": "string",
661         "title": "The Hobidanhiburan Schema ",
662         "default": "",
663         "examples": [
664             "0"
665         ]
666     },
667     "kesehatan": {
668         "$id":
669         "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
670         ies/Rencana Alokasi Dana/properties/kesehatan",
671         "type": "string",
672         "title": "The Kesehatan Schema ",
673         "default": "",
674         "examples": [
675             "0"
676         ]
677     },
678     "liburan": {
679         "$id":
680         "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
681         ies/Rencana Alokasi Dana/properties/liburan",
682         "type": "string",
683         "title": "The Liburan Schema ",
684         "default": "",
685         "examples": [
686             "0"
687         ]
688     },
689     "note": {
690         "$id":
691         "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
692         ies/Rencana Alokasi Dana/properties/note",
693         "type": "string",
694         "title": "The Note Schema ",
695         "default": "",
696         "examples": [
697             ""
698         ]
699     },

```

700	"pendidikan": {
701	"\$id":
702	"/properties/Users/properties/9E58P5PMs0YhLhfIPo512frprEq2/propert
703	ies/Rencana Alokasi Dana/properties/pendidikan",
704	"type": "string",
705	"title": "The Pendidikan Schema ",
706	"default": "",
707	"examples": [
708	"0"
709]
710	},
711	"transportasi": {
712	"\$id":
713	"/properties/Users/properties/9E58P5PMs0YhLhfIPo512frprEq2/propert
714	ies/Rencana Alokasi Dana/properties/transportasi",
715	"type": "string",
716	"title": "The Transportasi Schema ",
717	"default": "",
718	"examples": [
719	"0"
720]
721	},
722	"zakatAmal": {
723	"\$id":
724	"/properties/Users/properties/9E58P5PMs0YhLhfIPo512frprEq2/propert
725	ies/Rencana Alokasi Dana/properties/zakatAmal",
726	"type": "string",
727	"title": "The Zakatamal Schema ",
728	"default": "",
729	"examples": [
730	"0"
731]
732	}
733	}
734	},
735	"Total Pengeluaran": {
736	"\$id":
737	"/properties/Users/properties/9E58P5PMs0YhLhfIPo512frprEq2/propert
738	ies/Total Pengeluaran",
739	"type": "object",
740	"properties": {
741	"gadget": {
742	"\$id":
743	"/properties/Users/properties/9E58P5PMs0YhLhfIPo512frprEq2/propert
744	ies/Total Pengeluaran/properties/gadget",
745	"type": "string",
746	"title": "The Gadget Schema ",
747	"default": "",
748	"examples": [
749	"0"
750]
751	},
752	"harian": {
753	"\$id":
754	"/properties/Users/properties/9E58P5PMs0YhLhfIPo512frprEq2/propert
755	ies/Total Pengeluaran/properties/harian",
756	"type": "string",
757	"title": "The Harian Schema ",
758	"default": "",
759	"examples": [
760	"0"
761]
762	},
763	"hobiHiburan": {



```

764         "$id":
765         "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
766         ies/Total Pengeluaran/properties/hobiHiburan",
767         "type": "string",
768         "title": "The Hobihiburan Schema ",
769         "default": "",
770         "examples": [
771             "0"
772         ]
773     },
774     "kesehatan": {
775         "$id":
776         "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
777         ies/Total Pengeluaran/properties/kesehatan",
778         "type": "string",
779         "title": "The Kesehatan Schema ",
780         "default": "",
781         "examples": [
782             "0"
783         ]
784     },
785     "liburan": {
786         "$id":
787         "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
788         ies/Total Pengeluaran/properties/liburan",
789         "type": "string",
790         "title": "The Liburan Schema ",
791         "default": "",
792         "examples": [
793             "0"
794         ]
795     },
796     "nabung": {
797         "$id":
798         "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
799         ies/Total Pengeluaran/properties/nabung",
800         "type": "string",
801         "title": "The Nabung Schema ",
802         "default": "",
803         "examples": [
804             "0"
805         ]
806     },
807     "pendidikan": {
808         "$id":
809         "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
810         ies/Total Pengeluaran/properties/pendidikan",
811         "type": "string",
812         "title": "The Pendidikan Schema ",
813         "default": "",
814         "examples": [
815             "0"
816         ]
817     },
818     "tidakDianggarkan": {
819         "$id":
820         "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
821         ies/Total Pengeluaran/properties/tidakDianggarkan",
822         "type": "string",
823         "title": "The Tidakdianggarkan Schema ",
824         "default": "",
825         "examples": [
826             "0"
827         ]
828     },

```

```

829         "transportasi": {
830             "$id":
831             "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
832             ies/Total Pengeluaran/properties/transportasi",
833             "type": "string",
834             "title": "The Transportasi Schema ",
835             "default": "",
836             "examples": [
837                 "0"
838             ]
839         },
840         "zakatAmal": {
841             "$id":
842             "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
843             ies/Total Pengeluaran/properties/zakatAmal",
844             "type": "string",
845             "title": "The Zakatamal Schema ",
846             "default": "",
847             "examples": [
848                 "0"
849             ]
850         }
851     },
852     },
853     "Total Uang": {
854         "$id":
855         "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
856         ies/Total Uang",
857         "type": "object",
858         "properties": {
859             "totalUang": {
860                 "$id":
861                 "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
862                 ies/Total Uang/properties/totalUang",
863                 "type": "string",
864                 "title": "The Totaluang Schema ",
865                 "default": "",
866                 "examples": [
867                     "0"
868                 ]
869             }
870         }
871     },
872     "email": {
873         "$id":
874         "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
875         ies/email",
876         "type": "string",
877         "title": "The Email Schema ",
878         "default": "",
879         "examples": [
880             "january.arka@gmail.com"
881         ]
882     },
883     "name": {
884         "$id":
885         "/properties/Users/properties/9E58P5Pms0YhLhfIPo512frprEq2/propert
886         ies/name",
887         "type": "string",
888         "title": "The Name Schema ",
889         "default": "",
890         "examples": [
891             "Januari Arka"
892         ]
893     }

```

894	}
895	}
896	}
897	}
898	}
899	}

Pada tabel 5.5 diatas, perancangan data setiap pengguna yang terdaftar masuk kedalam satu data Users yang dibagi menjadi Anggaran, Batas Harian, Data Set, Data Uji, Hasil Status, Pemasukan, Pembelian Berdasarkan Jumlah Tabungan, Pembelian Berdasarkan Tanggal Pembelian, Pengeluaran, Rencana Alokasi Dana, Total Pengeluaran, Total Uang, email dan name. Pada keseluruhan data tersebut dibuat berdasarkan *sprint* yang berbeda. Untuk lebih jelasnya dapat dilihat pada sub bab berikut.

5.1.3.1 Sprint 1

Pada perancangan basis data *sprint 1*, di rancang dengan pembuatan *parent* Users yang memiliki *child* berupa *push id user*. Setiap *push id user* memiliki *child* Data Uji, Pemasukan, Pengeluaran, Rencana Alokasi Dana, Total Pengeluaran, dan Total Uang yang digunakan untuk menunjang fitur dasar yang menjadi *sprint goal* pada *sprint 1*. Untuk lebih jelasnya dapat dilihat pada tabel 5.6.

Tabel 5. 6 Deskripsi Perancangan Basis Data Sprint 1

Parent	Child	Child	Deskripsi
Users	Push ID User	Data Uji	<i>Child</i> Data Uji digunakan untuk menyimpan nilai dari data uji ketika pengeluaran dengan kategori Tidak Dianggarkan dilakukan.
		Pemasukan	<i>Child</i> Pemasukan digunakan untuk menyimpan nilai dari menambah pemasukan yang dilakukan.
		Pengeluaran	<i>Child</i> Pengeluaran digunakan untuk menyimpan nilai dari menambah pengeluaran.
		Rencana Alokasi Dana	<i>Child</i> Pengeluaran digunakan untuk menyimpan nilai rencana alokasi dana tidak dianggarkan ketika menambah pengeluaran dengan kategori Tidak Dianggarkan dilakukan.
		Total Pengeluaran	<i>Child</i> Total Pengeluaran digunakan untuk menyimpan nilai dari total pengeluaran tiap kategori anggaran.



		Total Uang	<i>Child</i> Total Uang digunakan untuk menyimpan nilai total uang dari pemasukan yang dilakukan.
--	--	------------	---

5.1.3.2 Sprint 2

Pada perancangan basis data *sprint 2*, di rancang dengan menambahkan *child* pada *parent* Users yang memiliki *child* berupa *push id user*. Setiap *push id user* memiliki *child* Anggaran, Batas Harian, Pembelian Berdasarkan Jumlah Tabungan, dan Pembelian Berdasarkan Tanggal Pembelian yang digunakan untuk menunjang fitur lanjutan yang menjadi *sprint goal* pada *sprint 2*. Untuk lebih jelasnya dapat dilihat pada tabel 5.7.

Tabel 5. 7 Deskripsi Perancangan Basis Data Sprint 2

Parent	Child	Child	Deskripsi
Users	Push ID User	Anggaran	<i>Child</i> Anggaran digunakan untuk menyimpan nilai dari perencanaan anggaran yang telah dilakukan.
		Batas Harian	<i>Child</i> Batas Harian digunakan untuk menyimpan nilai dari batas harian yang telah dibuat.
		Pembelian Berdasarkan Jumlah Tabungan	<i>Child</i> Pembelian Berdasarkan Jumlah Tabungan digunakan untuk menyimpan nilai dari perencanaan pembelian berdasarkan jumlah tabungan yang telah dilakukan.
		Pembelian Berdasarkan Tanggal Pembelian	<i>Child</i> Pembelian Berdasarkan Tanggal Pembelian digunakan untuk menyimpan nilai dari perencanaan pembelian berdasarkan tanggal pembelian yang telah dilakukan.

5.1.4 Perancangan Antarmuka

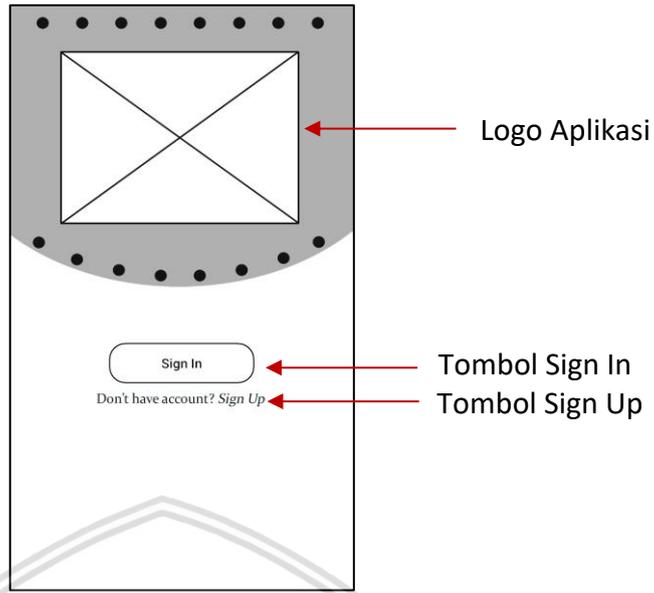
Perancangan ini akan membuat *wireframe* dan *screenflow diagram*. Pada *screenflow diagram* akan dibagi menjadi 2, yaitu *screenflow diagram* umum dan *screenflow diagram detail*.

5.1.4.1 Wireframe

(a) Halaman Utama

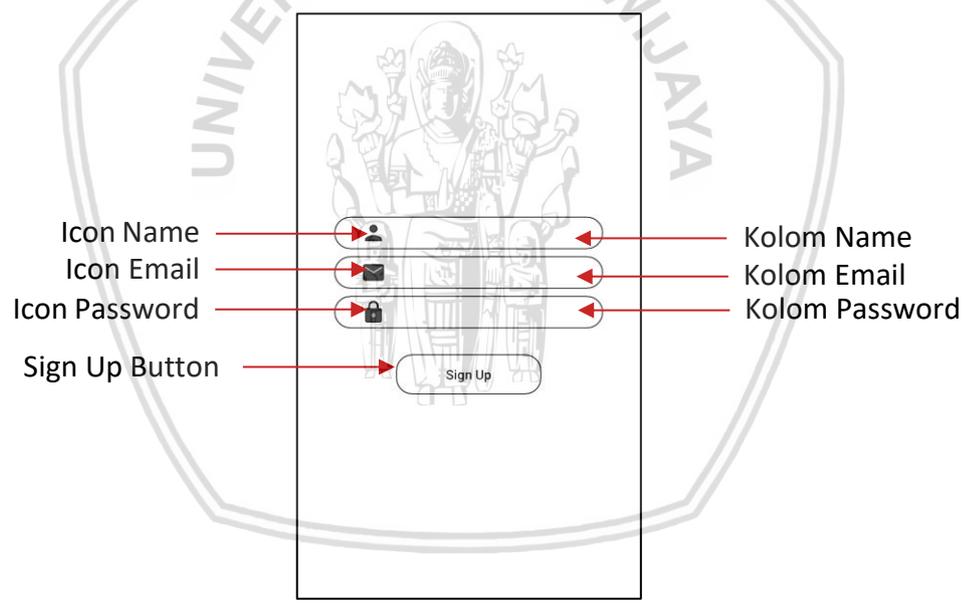
Pada gambar 5.12 merupakan halaman utama FM App. Pada bagian atas berbentuk kotak merupakan logo dari aplikasi FM App. Lalu, dibagian tengah terdapat tombol Sign In untuk masuk kedalam sistem sebagai *member* dan tombol Sign Up untuk mendaftar sebagai *member* kedalam sistem.





Gambar 5.12 Wireframe Halaman Utama

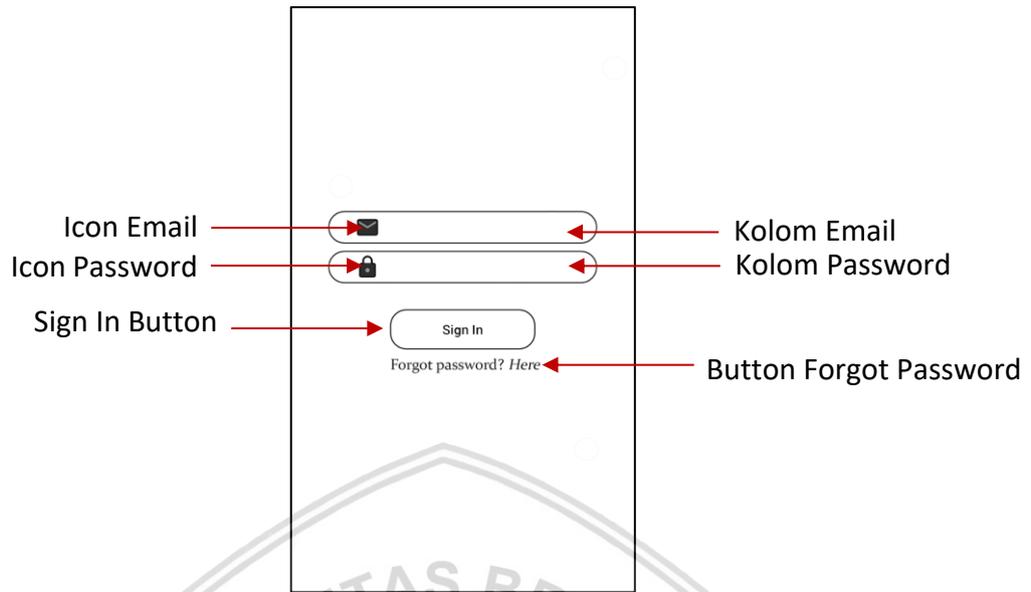
(b) Sign Up



Gambar 5.13 Wireframe Sign Up

Pada gambar 5.13 merupakan halaman *Sign Up*. Pada bagian tengah terdapat tiga kolom penginputan, yaitu kolom penginputan *name*, *email*, dan *password*. Dibawahnya kolom penginputan merupakan tombol *Sign Up* sebagai *trigger* mendaftarkan *member* baru.

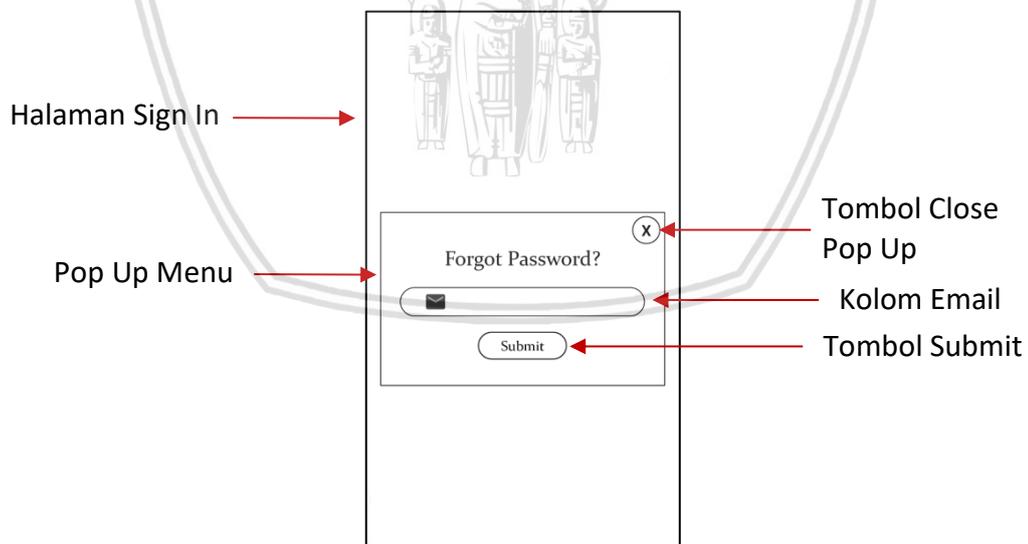
(c) *Sign In*



Gambar 5. 14 *Wireframe Sign In*

Pada gambar 5.14 merupakan halaman *Sign In*. Pada bagian tengah terdapat dua kolom penginputan dan satu tombol. Dua kolom penginputan tersebut merupakan kolom penginputan *email* dan *password* serta tombol tersebut merupakan tombol *sign in* sebagai *trigger* autentikasi data *member*.

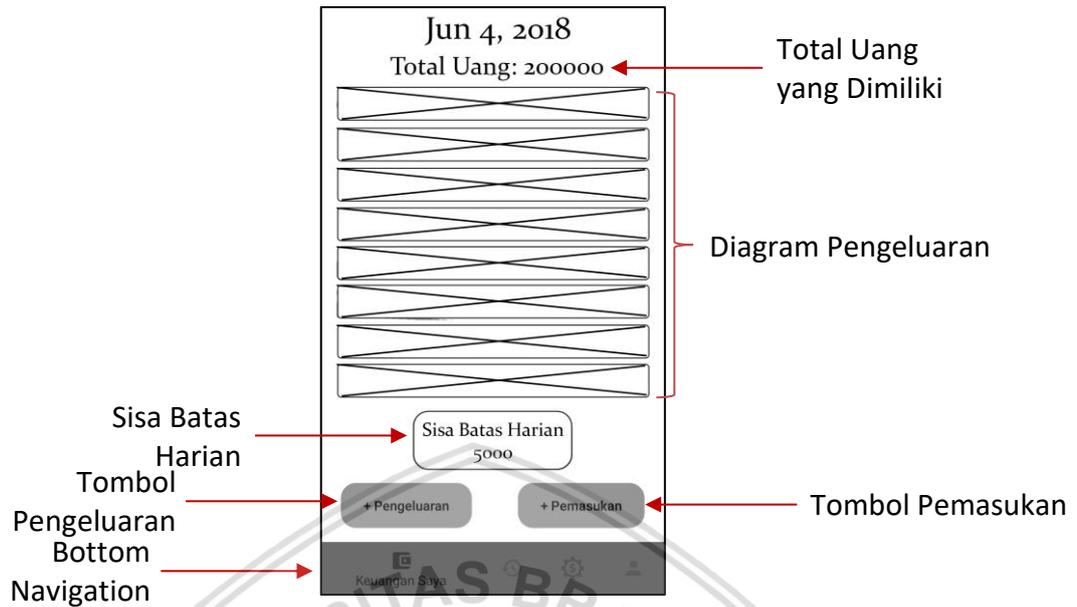
(d) *Forgot Password*



Gambar 5. 15 *Wireframe Forgot Password*

Pada gambar 5.15 merupakan halaman *Forgot Password*. Halaman ini berbentuk pop up menu pada halaman *Sign In*. Pada pojok kanan atas terdapat tombol *close* untuk menutup *pop up*. Pada bagian tengah terdapat kolom penginputan *email* dan tombol *submit* untuk perubahan *password* dengan mengirimkan intruksi perubahan *password* ke *email* yang telah dimasukan.

(e) Keuangan Saya

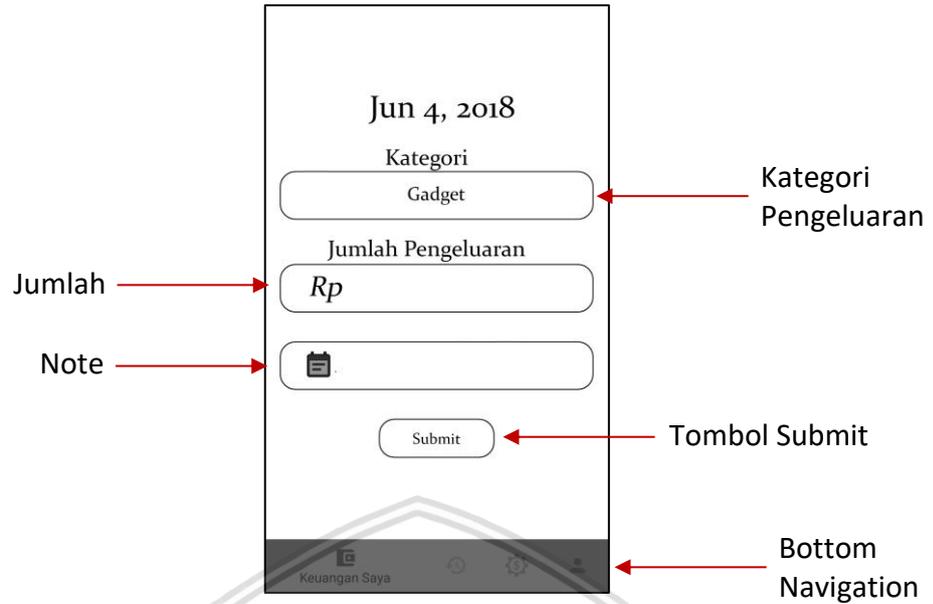


Gambar 5. 16 Wireframe Keuangan Saya

Pada gambar 5.16 merupakan halaman Keuangan Saya. Halaman ini merupakan halaman inti. Pada bagian atas terdapat tanggal dan total uang yang dimiliki member. Dibawahnya terdapat diagram pengeluaran berdasarkan kategori anggaran. Dibawah diagram pengeluaran terdapat informasi sisa batas harian pengeluaran. Dibawahnya informasi sisa batas harian terdapat dua tombol, yaitu tombol pengeluaran dan tombol pemasukan yang berfungsi untuk memasukkan catatan pengeluaran atau pemasukan. Pada bagian paling bawah terdapat tombol *bottom navigation* yang berfungsi sebagai *trigger* membuka halaman inti.

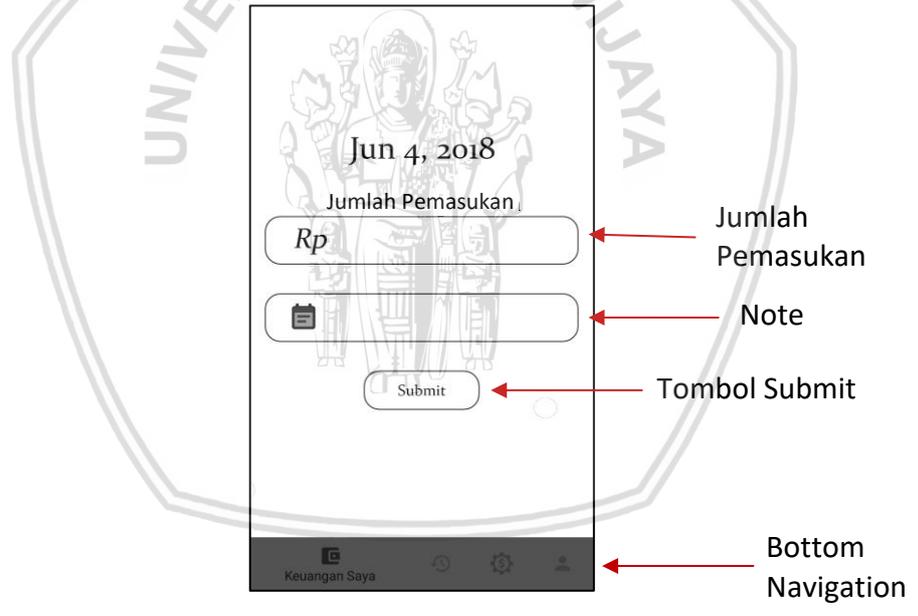
(f) Tambah Pengeluaran

Pada gambar 5.19 merupakan halaman Tambah Pengeluaran. Pada halaman ini terdapat tanggal, kolom penginputan kategori, jumlah pengeluaran, dan *note*. Selain itu, terdapat tombol *submit* yang berfungsi sebagai *trigger* menyimpan data pengeluaran ke *database*. Pada bagian paling bawah terdapat tombol *bottom navigation* yang berfungsi sebagai *trigger* membuka halaman inti.



Gambar 5. 17 *Wrifame* Tambah Pengeluaran

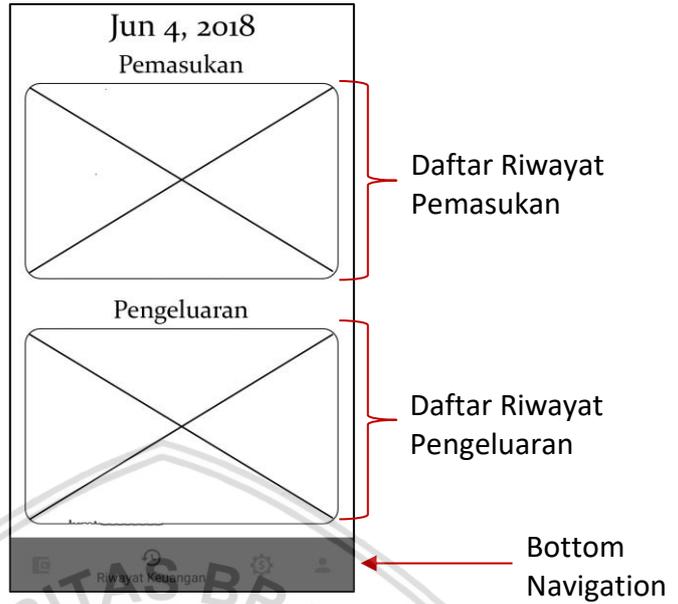
(g) Tambah Pemasukan



Gambar 5. 18 *Wireframe* Tambah Pemasukan

Pada gambar 5.18 merupakan halaman Tambah Pemasukan. Pada halaman ini terdapat dua kolom penginputan, yaitu kolom penginputan jumlah pemasukan dan *note*. Selain itu, terdapat tombol *submit* sebagai *trigger* menyimpan data pemasukan ke *database* dan tombol *bottom navigation* yang berfungsi sebagai *trigger* membuka halaman inti.

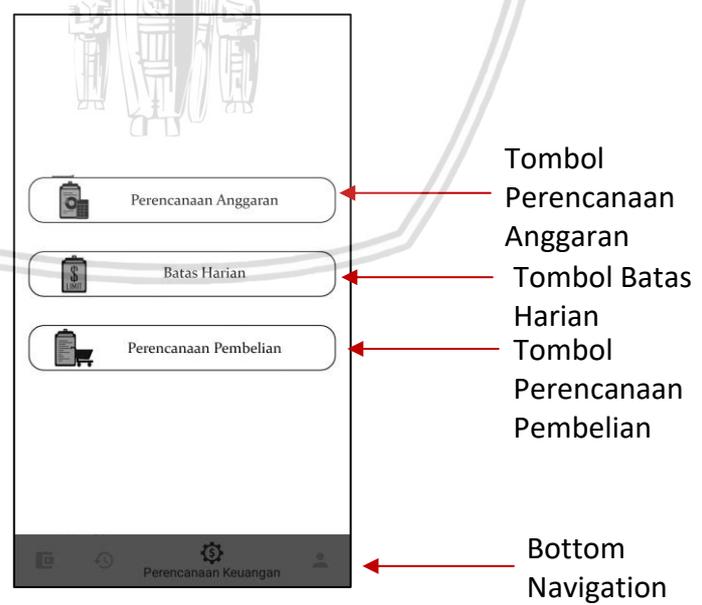
(h) Riwayat Keuangan



Gambar 5. 19 Wireframe Riwayat Keuangan

Pada gambar 5.19 merupakan halaman Riwayat Keuangan. Halaman ini merupakan halaman inti. Pada halaman ini terdapat kolom informasi, yaitu daftar kolom informasi daftar riwayat pemasukan pada bagian atas dan daftar riwayat pengeluaran pada bagian bawah. Selain itu, terdapat tombol *bottom navigation* yang berfungsi sebagai *trigger* membuka halaman inti.

(i) Perencanaan Keuangan



Gambar 5. 20 Wireframe Perencanaan Keuangan

Pada gambar 5.20 merupakan halaman Perencanaan Keuangan. Halaman ini merupakan halaman inti. Halaman ini terdapat tiga tombol pada bagian tengah, yaitu tombol perencanaan anggaran, batas harian, dan perencanaan pembelian.

Selain itu, terdapat tombol *bottom navigation* yang berfungsi sebagai *trigger* membuka halaman inti.

(j) Perencanaan Anggaran



Gambar 5. 21 Wireframe Perencanaan Anggaran

Pada gambar 5.21 merupakan halaman Perencanaan Anggaran. Halaman ini terdapat delapan kolom penginputan daftar alokasi perencanaan anggaran pengeuaran berdasarkan kategori, tombol untuk mengakses keterangan kategori anggaran, tombol *submit* berfungsi sebagai *trigger* menyimpan data anggaran ke *database*, dan tombol *bottom navigation* yang berfungsi sebagai *trigger* membuka halaman inti.

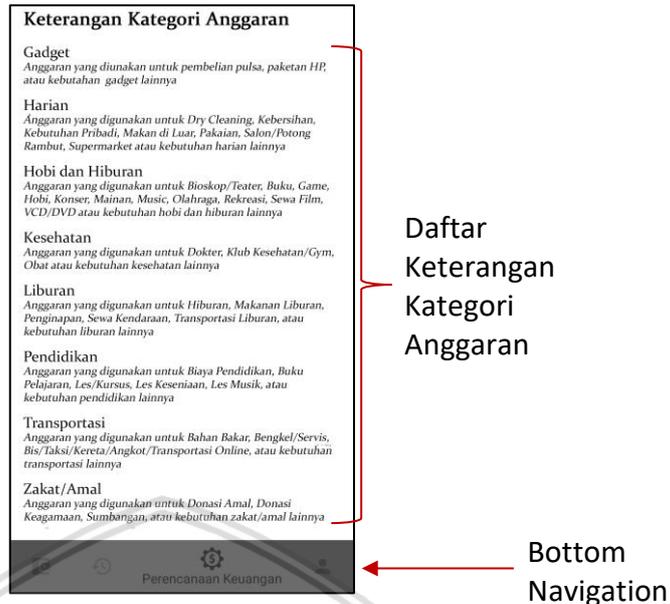
(k) Keterangan Kategori Anggaran

Pada gambar 5.22 merupakan halaman Keterangan Kategori Anggaran. Pada halaman ini terdapat daftar informasi keterangan kategori anggaran dan tombol *bottom navigation* yang berfungsi sebagai *trigger* membuka halaman inti.

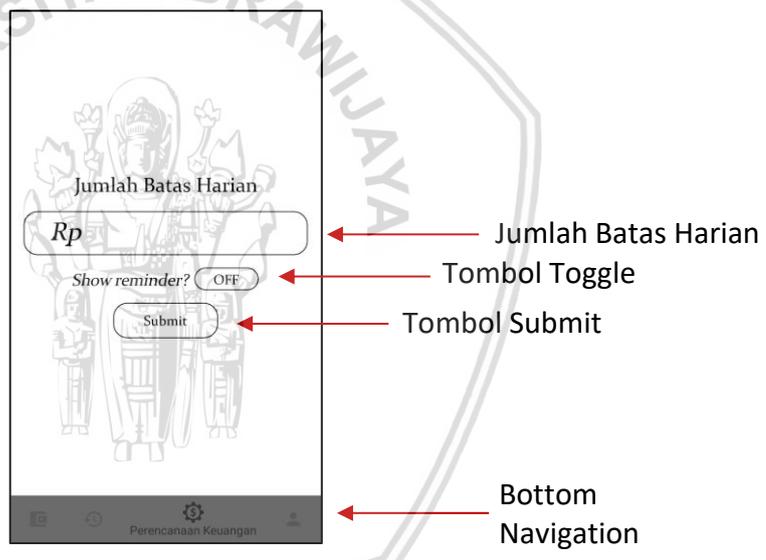
(l) Batas Harian

Pada gambar 5.23 merupakan halaman Batas Harian. Pada bagian ini terdapat kolom penginputan baatas harian, tombol *toggle* untuk mengatur apakah *member* mendapat *reminder notification* atau tidak, dan tombol *submit* sebagai *trigger* untuk menyimpan data batas harian ke *database*. Selain itu, terdapat tombol *bottom navigation* yang berfungsi sebagai *trigger* membuka halaman inti.





Gambar 5. 22 Wireframe Keterangan Kategori Anggaran

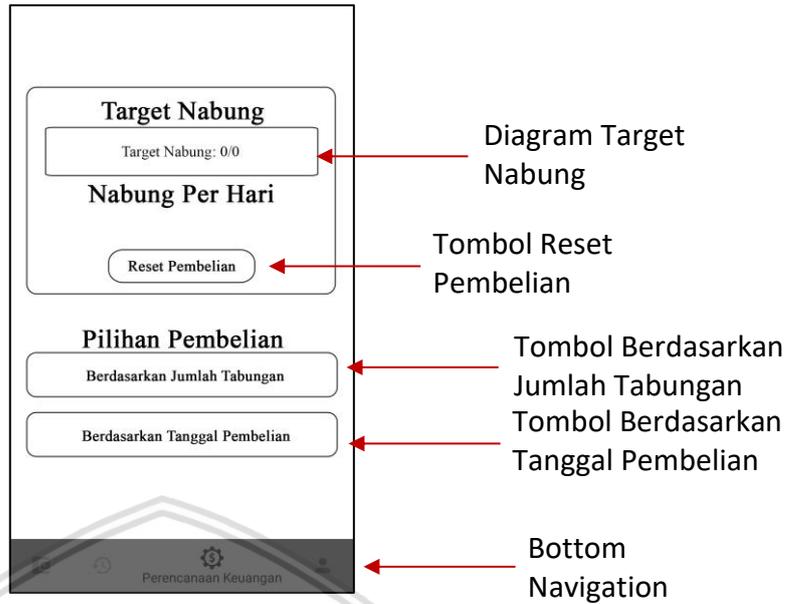


Gambar 5. 23 Wireframe Batas Harian

(m)Perencanaan Pembelian

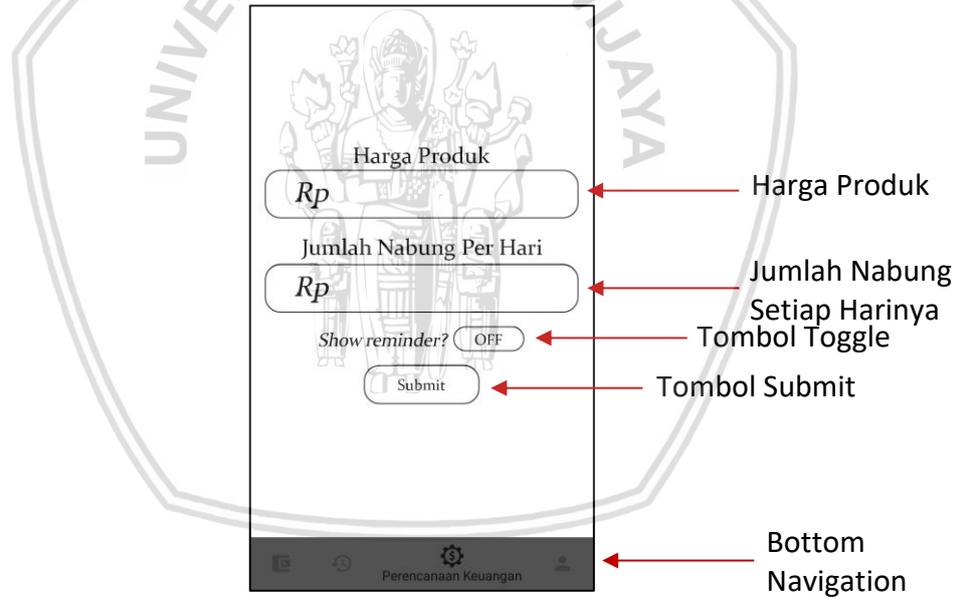
Pada gambar 5.24 merupakan halaman Perencanaan Pembelian. Halaman ini merupakan halaman inti. Pada halaman ini terdapat keterangan target menabung dan jumlah nabung perhari. Selain itu terdapat tiga tombol pada bagian tengah, yaitu tombol reset pembelian, tombol berdasarkan jumlah tabungan dan berdasarkan tanggal pembelian. Selain itu, terdapat tombol *bottom navigation* yang berfungsi sebagai *trigger* membuka halaman inti.





Gambar 5. 24 Wireframe Perencanaan Pembelian

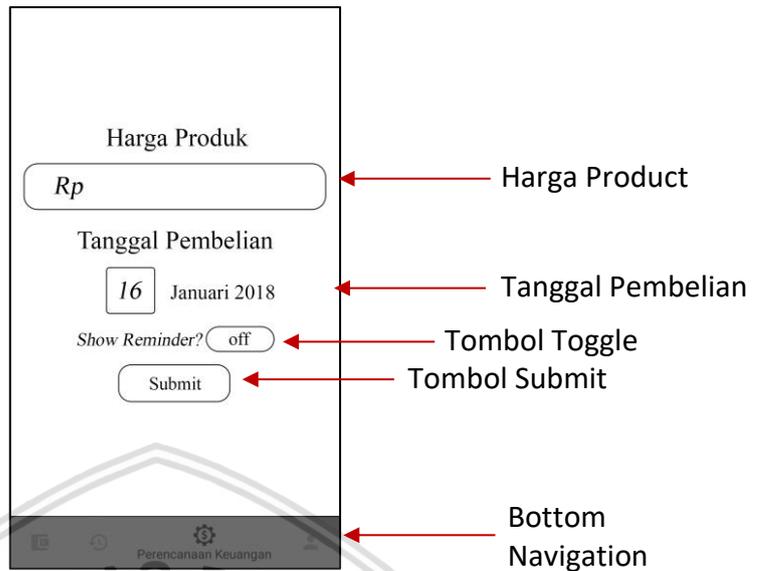
(n) Berdasarkan Jumlah Tabungan



Gambar 5. 25 Wireframe Berdasarkan Jumlah Tabungan

Pada gambar 5.25 merupakan halaman Berdasarkan Jumlah Tabungan. Pada halaman ini terdapat dua kolom penginputan, yaitu kolom penginputan harga produk dan jumlah nabung setiap harinya. Selain itu, terdapat juga tiga tombol, yaitu tombol *toggle* untuk mengatur apakah *member* mendapat *reminder notification* atau tidak, tombol *submit* sebagai *trigger* untuk menyimpan data perencanaan pembelian berdasarkan jumlah tabungan ke *database*, dan tombol *bottom navigation* yang berfungsi sebagai *trigger* membuka halaman inti.

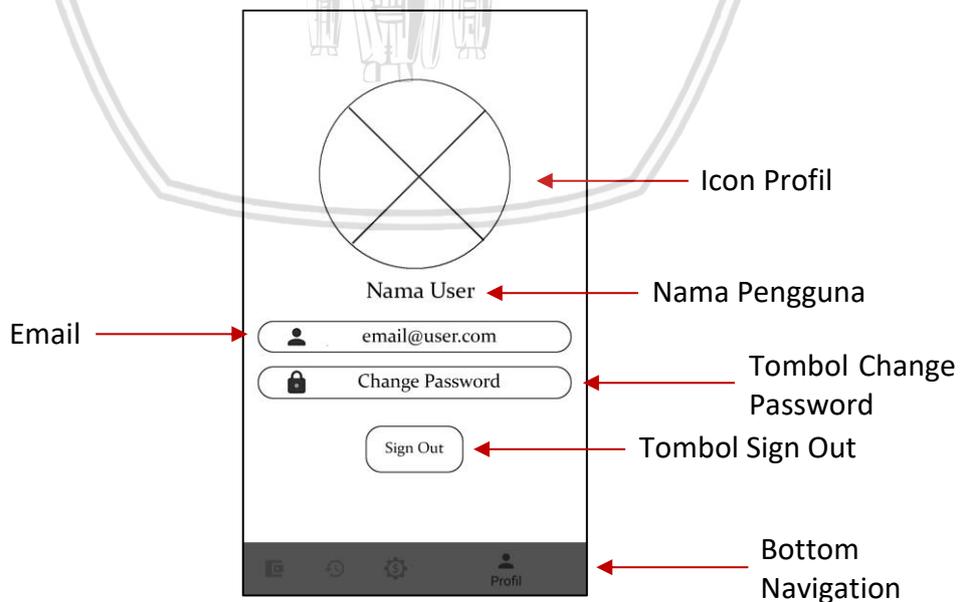
(o) Berdasarkan Tanggal Pembelian



Gambar 5. 26 Wireframe Berdasarkan Tanggal Pembelian

Pada gambar 5.26 merupakan halaman Berdasarkan Tanggal Pembelian. Pada halaman ini terdapat dua kolom penginputan, yaitu kolom penginputan harga produk dan tanggal pembelian. Selain itu, terdapat juga tiga tombol, yaitu tombol *toggle* untuk mengatur apakah *member* mendapat *reminder notification* atau tidak, tombol *submit* sebagai *trigger* untuk menyimpan data perencanaan pembelian berdasarkan tanggal pembelian ke *database*, dan tombol *bottom navigation* yang berfungsi sebagai *trigger* membuka halaman inti.

(p) Profil

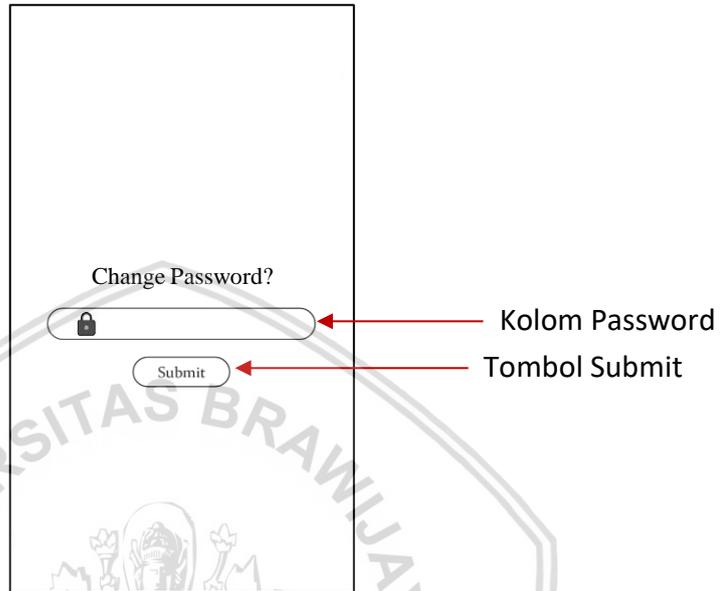


Gambar 5. 27 Wireframe Profil

Pada gambar 5.27 merupakan halaman Profil. Pada halaman ini terdapat gambar *icon* profil pada bagian atas, informasi nama pengguna, informasi email

pengguna. Selain itu terdapat tiga tombol, yaitu tombol *change password* untuk mengubah *password*, tombol *sign out* untuk keluar dari sistem sebagai *member*, dan tombol *bottom navigation* yang berfungsi sebagai *trigger* membuka halaman inti.

(q) Change Password



Gambar 5. 28 Wireframe Change Password

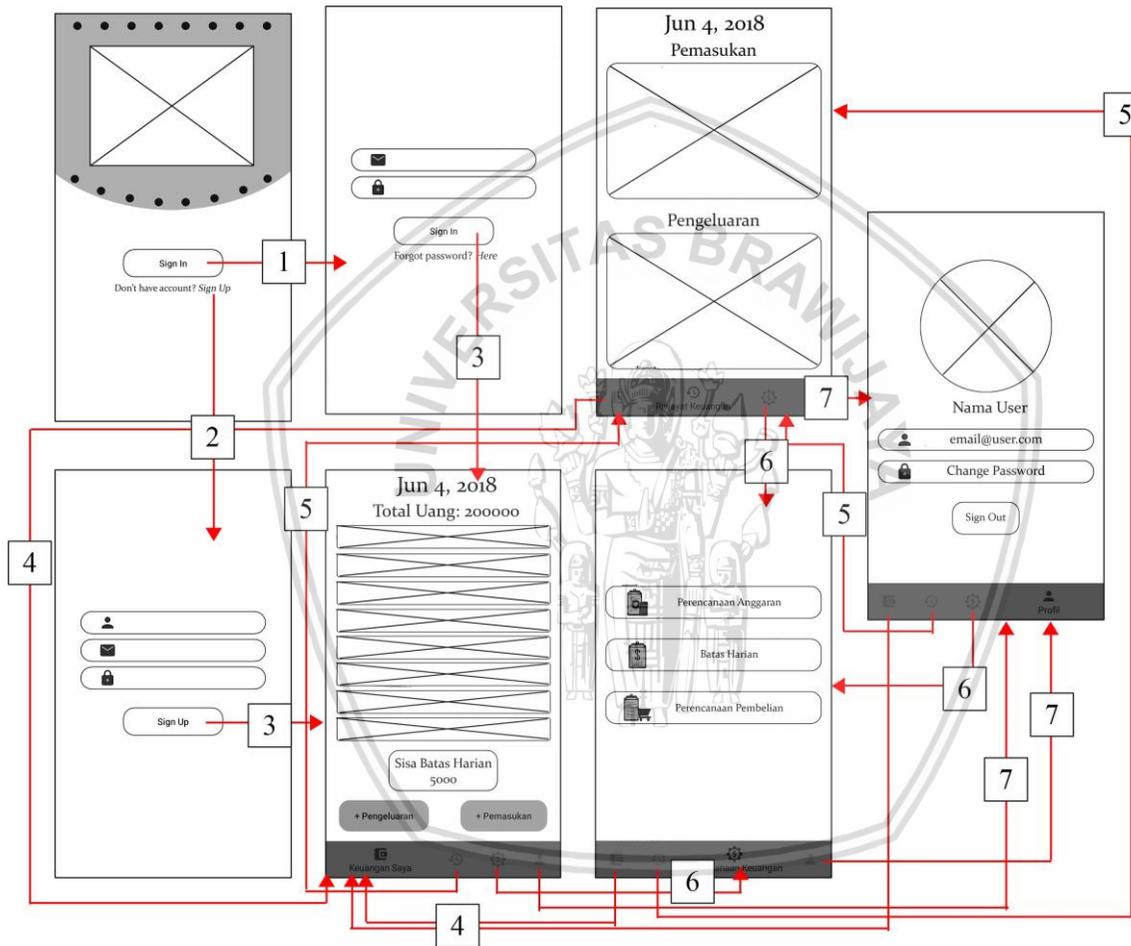
Pada gambar 5.28 merupakan halaman *Change Password*. Pada halaman ini terdapat kolom masukan *password* dan tombol *submit* sebagai *trigger* menyimpan data perubahan *password* ke *database*.

5.1.4.2 Screenflow Diagram

Screenflow Diagram merupakan penjelasan alur untuk mengerjakan suatu fungsi. *Screenflow Diagram* akan dibagi menjadi dua bagian, yaitu *screen flow* umum dan *screen flow detail*. Untuk lebih jelasnya dapat dilihat pada sub bab berikut.

(a) Screenflow Umum

Screenflow umum ini menjelaskan tampilan utama yang dapat diakses pada FM App. Pada *screenflow* ini terdapat 7 bagian. Untuk menjelaskan *screenflow* umum pada gambar 5.29, maka dapat dilihat pada tabel 5.8.



Gambar 5. 29 Screenflow Umum FM App

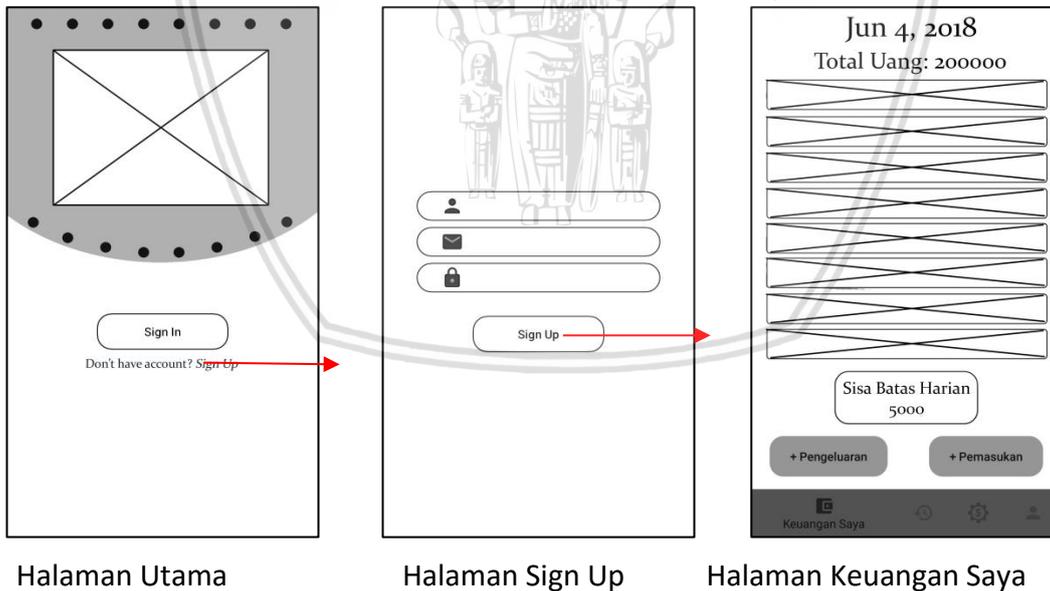
Tabel 5. 8 Penjelasan *Screenflow* Umum FM App

Nomor Flow	Deskripsi
1	Merupakan cara untuk menampilkan halaman <i>sign in</i>
2	Merupakan cara untuk menampilkan halaman <i>sign up</i>
3	Merupakan tampilan yang ditampilkan ketika tombol <i>sign in</i> dan <i>sign up</i> ditekan dan berhasil masuk ke sistem FM App
4	Merupakan cara untuk menampilkan halaman keuangan saya
5	Merupakan cara untuk menampilkan halaman riwayat keuangan
6	Merupakan cara untuk menampilkan halaman perencanaan keuangan
7	Merupakan cara untuk menampilkan halaman profil

(b) Screenflow Detail

Screenflow detail ini menjelaskan apa saja yang dapat dilakukan pada setiap tampilan utama pada FM App. Pada *screenflow* ini terdapat 17 bagian. Untuk lebih jelasnya dapat dilihat pada sub bab berikutnya.

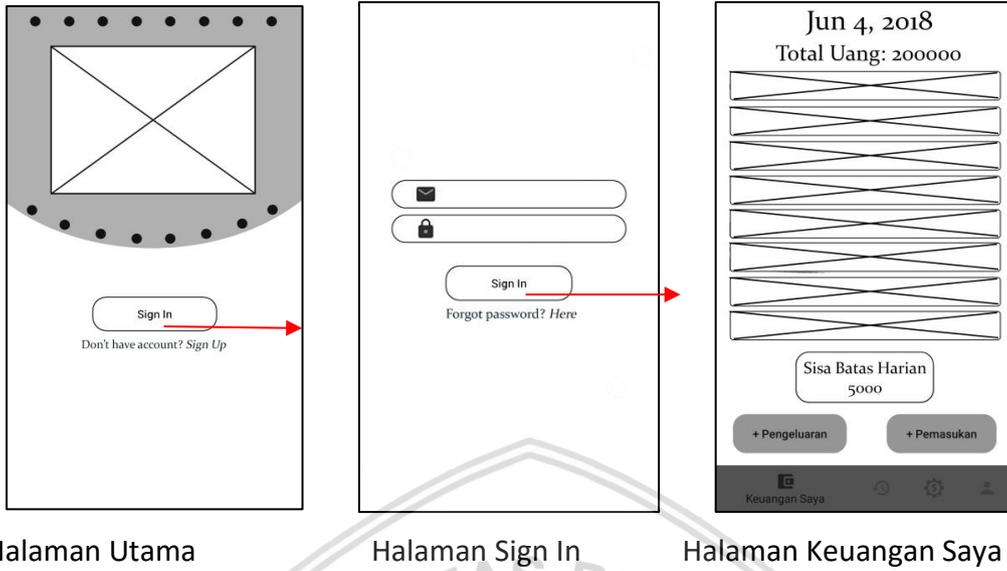
(i) Screenflow Sign Up



Gambar 5. 30 Screenflow Sign Up

Pada gambar 5.30 merupakan *screenflow* untuk *sign up*. Tahap awal adalah membuka aplikasi dengan ditampilkannya halaman utama. Pada halaman utama, *text Sign Up* ditekan yang menjadi *trigger* untuk membuka halaman *Sign Up*. Setelah *sign up form* sudah diisi dan tombol *sign up* ditekan, maka akan langsung masuk kedalam halaman Keuangan Saya sebagai *member*.

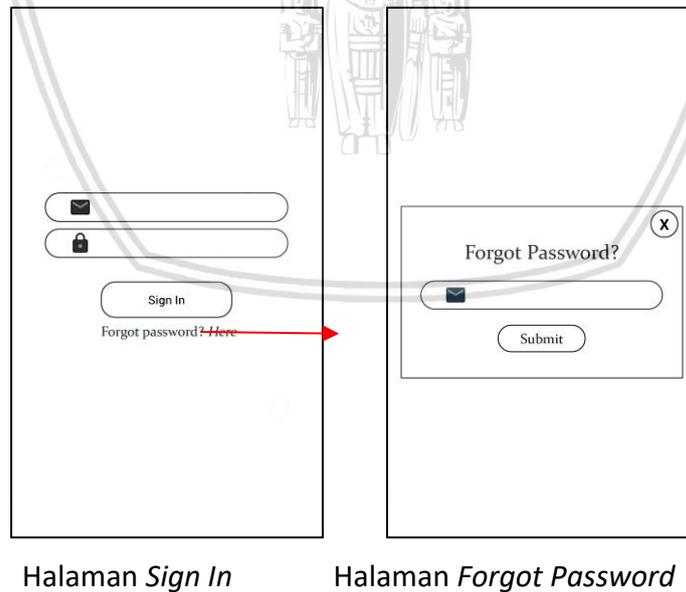
(ii) Screenflow Sign In



Gambar 5. 31 Screenflow Sign In

Pada gambar 5.31 merupakan *screenflow* untuk *sign in*. Tahap awal adalah membuka aplikasi dengan ditampilkannya halaman utama. Pada halaman utama, tombol *Sign In* ditekan yang menjadi *trigger* untuk membuka halaman *Sign In*. Setelah *sign in form* sudah diisi dan tombol *sign in* ditekan, maka akan langsung masuk kedalam halaman Keuangan Saya sebagai *member*.

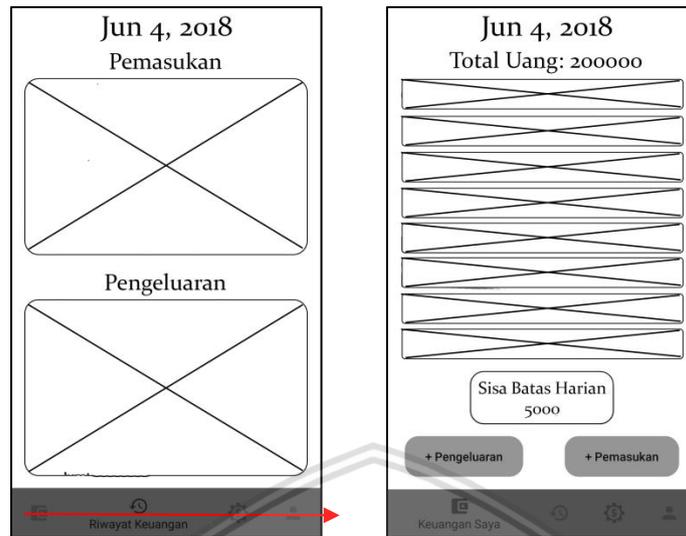
(iii) Screenflow Forgot Password



Gambar 5. 32 Screenflow Forgot Password

Pada gambar 5.32 merupakan *screenflow* untuk *forgot password*. Pada halaman *Sign In*, *text Here* ditekan yang menjadi *trigger* untuk membuka halaman *Forgot Password*.

(iv) Screenflow Keuangan Saya

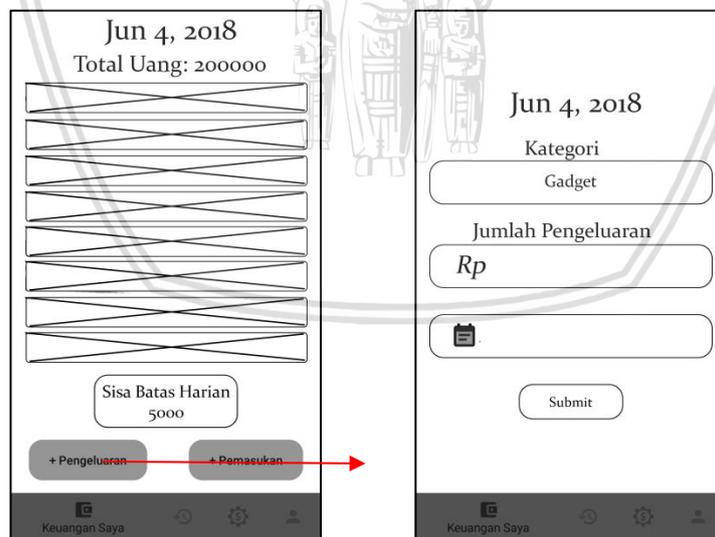


Halaman Riwayat Keuangan Halaman Keuangan Saya

Gambar 5. 33 Screenflow Keuangan Saya

Pada gambar 5.33 merupakan *screenflow* untuk melihat Halaman Keuangan Saya. Pada setiap halaman inti akan terdapat tombol *Bottom Navigation*. Tombol icon Keuangan Saya yang terletak pada pojok kiri bawah ditekan, lalu halaman Keuangan Saya akan ditampilkan.

(v) Screenflow Menambah Pengeluaran



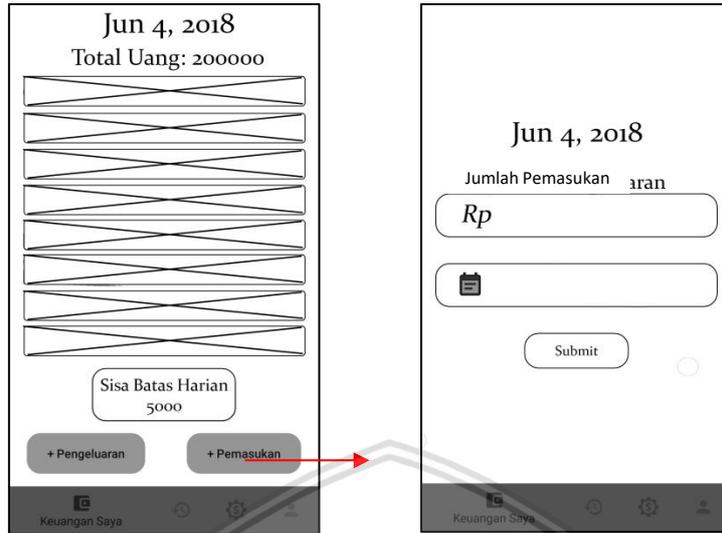
Halaman Keuangan Saya

Halaman Pengeluaran

Gambar 5. 34 Screenflow Menambah Pengeluaran

Pada gambar 5.34 merupakan *screenflow* untuk menambah Pengeluaran. Pada halaman Keuangan Saya, tombol tambah pengeluaran ditekan. Lalu, halaman Pengeluaran akan ditampilkan.

(vi) **Screenflow Menambah Pemasukan**

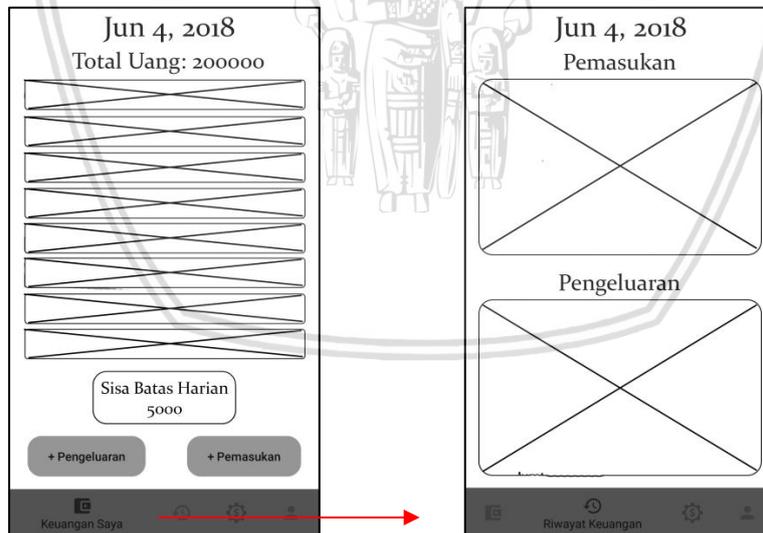


Halaman Keuangan Saya Halaman Pemasukan

Gambar 5. 35 Screenflow Menambah Pemasukan

Pada gambar 5.35 merupakan *screenflow* untuk menambah Pemasukan. Pada halaman Keuangan Saya, tombol tambah pemasukan ditekan. Lalu, halaman Pemasukan akan ditampilkan.

(vii) **Screenflow Riwayat Keuangan**

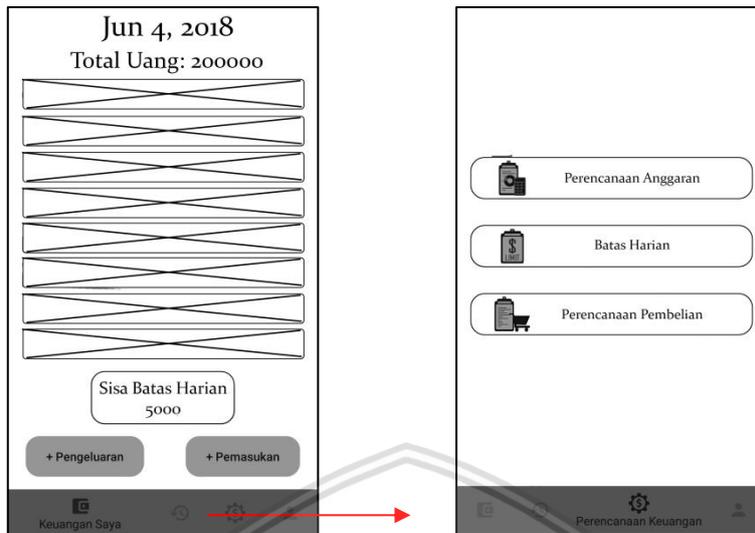


Halaman Keuangan Saya Halaman Riwayat Keuangan

Gambar 5. 36 Screenflow Riwayat Keuangan

Pada gambar 5.36 merupakan *screenflow* untuk melihat Halaman Riwayat Keuangan. Pada setiap halaman inti akan terdapat tombol *Bottom Navigation*. Tombol *icon* Riwayat Keuangan yang terletak pada nomor dua dari pojok kiri bawah ditekan, lalu halaman Riwayat Keuangan akan ditampilkan.

(viii) **Screenflow** Perencanaan Keuangan

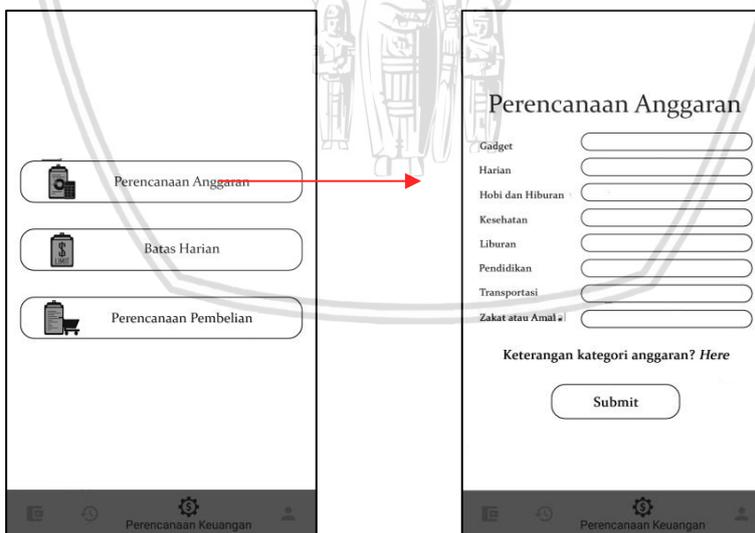


Halaman Keuangan Saya Halaman Perencanaan Keuangan

Gambar 5. 37 Screenflow Perencanaan Keuangan

Pada gambar 5.37 merupakan *screenflow* untuk melihat Halaman Perencanaan Keuangan. Pada setiap halaman inti akan terdapat tombol *Bottom Navigation*. Tombol *icon* Perencanaan Keuangan yang terletak pada nomor dua dari pojok kanan bawah ditekan, lalu halaman Perencanaan Keuangan akan ditampilkan.

(ix) **Screenflow** Perencanaan Anggaran



Halaman Perencanaan Keuangan Halaman Perencanaan Anggaran

Gambar 5. 38 Screenflow Perencanaan Anggaran

Pada gambar 5.38 merupakan *screenflow* untuk melihat Halaman Perencanaan Anggaran. Tombol Perencanaan Anggaran pada halaman Perencanaan Keuangan ditekan, lalu halaman Perencanaan Anggaran akan ditampilkan.

(x) **Screenflow** Keterangan Kategori Anggaran



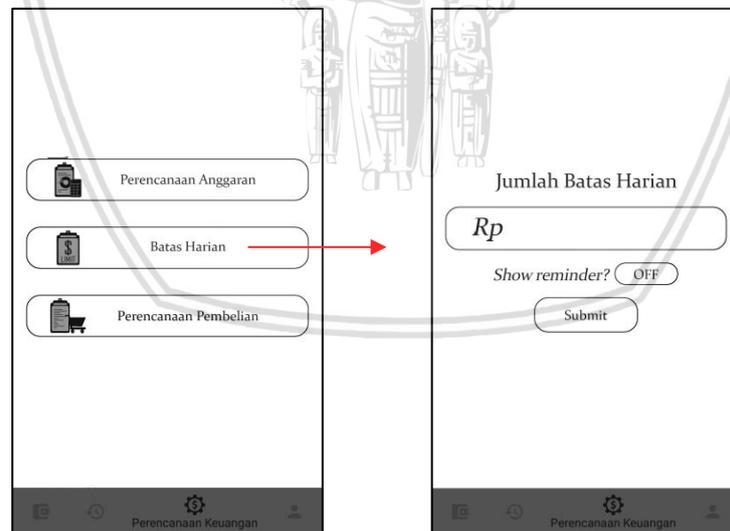
Halaman Perencanaan Anggaran

Halaman Keterangan Kategori Anggaran

Gambar 5. 39 Screenflow Keterangan Kategori Anggaran

Pada gambar 5.39 merupakan *screenflow* untuk melihat Halaman Keterangan Kategori Anggaran. *Text Here* pada halaman Perencanaan Anggaran ditekan, lalu halaman Keterangan Kategori Anggaran akan ditampilkan.

(xi) **Screenflow** Batas Harian



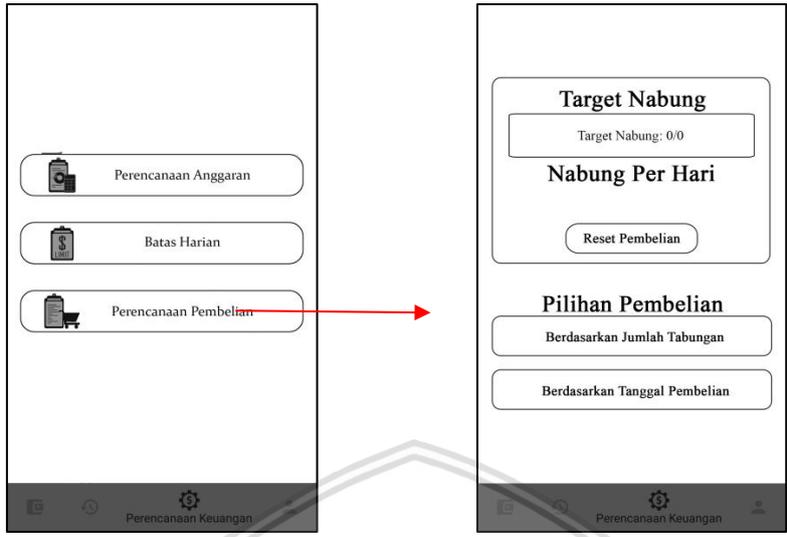
Halaman Perencanaan Keuangan

Halaman Batas Harian

Gambar 5. 40 Screenflow Batas Harian

Pada gambar 5.40 merupakan *screenflow* untuk melihat Halaman Batas Harian. Tombol Batas Harian pada halaman Perencanaan Keuangan ditekan, lalu halaman Keterangan Batas Harian akan ditampilkan.

(xii) **Screenflow Perencanaan Pembelian**

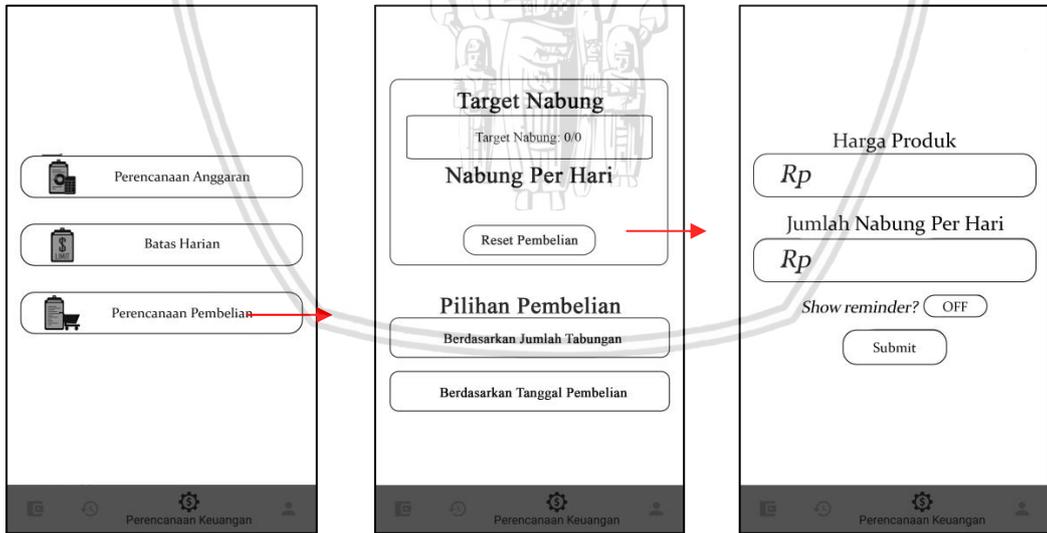


Halaman Perencanaan Keuangan Halaman Perencanaan Pembelian

Gambar 5. 41 Screenflow Perencanaan Pembelian

Pada gambar 5.41 merupakan *screenflow* untuk melihat Halaman Perencanaan Pembelian. Tombol Perencanaan Pembelian pada halaman Perencanaan Keuangan ditekan, lalu halaman Perencanaan Keuangan akan ditampilkan.

(xiii) **Screenflow Berdasarkan Jumlah Tabungan**



Halaman Perencanaan Anggaran

Halaman Perencanaan Pembelian

Halaman Berdasarkan Jumlah Tabungan

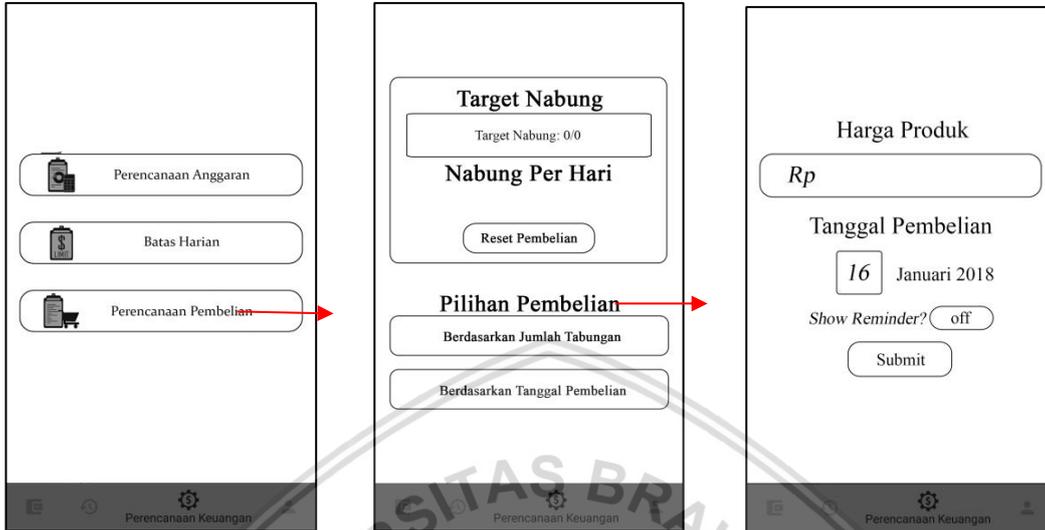
Gambar 5. 42 Screenflow Berdasarkan Jumlah Tabungan

Pada gambar 5.42 merupakan *screenflow* untuk melihat Halaman Berdasarkan Jumlah Tabungan. Tombol Perencanaan Pembelian pada halaman Perencanaan Keuangan ditekan, lalu halaman Perencanaan Keuangan akan ditampilkan. Pada



halaman Perencanaan Pembelian, tombol Berdasarsakan Jumlah Tabungan ditekan, maka akan menampilkan halaman Berdasarkan Jumlah Tabungan.

(xiv) Screenflow Berdasarkan Tanggal Pembelian



Halaman Perencanaan Anggaran

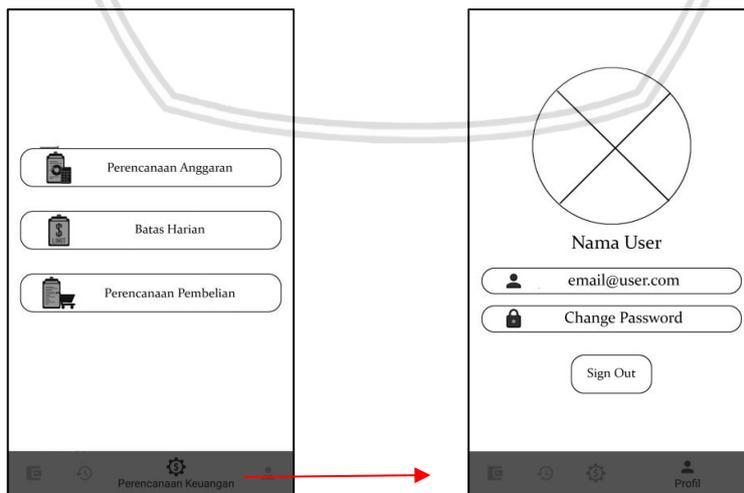
Halaman Perencanaan Pembelian

Halaman Berdasarkan Tanggal Pembelian

Gambar 5. 43 Screenflow Berdasarkan Tanggal Pembelian

Pada gambar 5.43 merupakan *screenflow* untuk melihat Halaman Berdasarkan Tanggal Pembelian. Tombol Perencanaan Pembelian pada halaman Perencanaan Keuangan ditekan, lalu halaman Perencanaan Keuangan akan ditampilkan. Pada halaman Perencanaan Pembelian, tombol Berdasarsakan Tanggal Pembelian ditekan, maka akan menampilkan halaman Berdasarkan Tanggal Pembelian.

(xv) Screenflow Profil



Halaman Perencanaan Keuangan

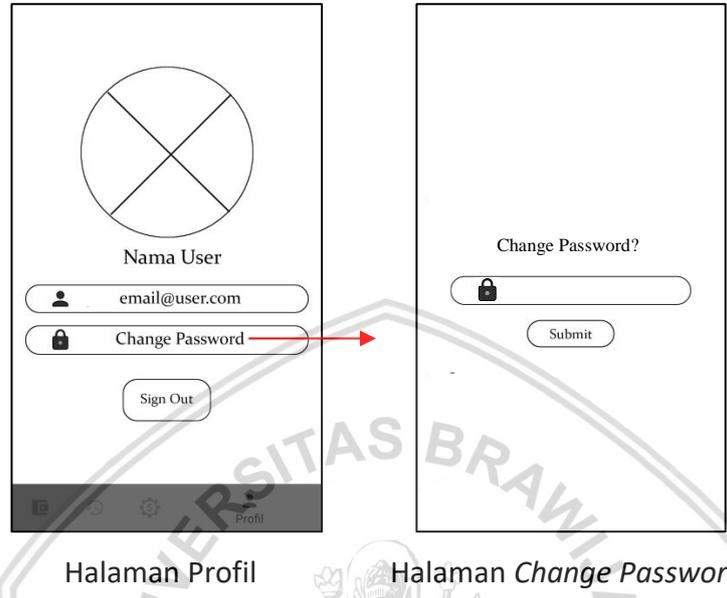
Halaman Profil

Gambar 5. 44 Screenflow Profil



Pada gambar 5.44 merupakan *screenflow* untuk melihat halaman Profil. Pada setiap halaman inti akan terdapat tombol *Bottom Navigation*. Tombol *icon* Profil yang terletak pada pojok kanan bawah ditekan, lalu halaman Profil akan ditampilkan.

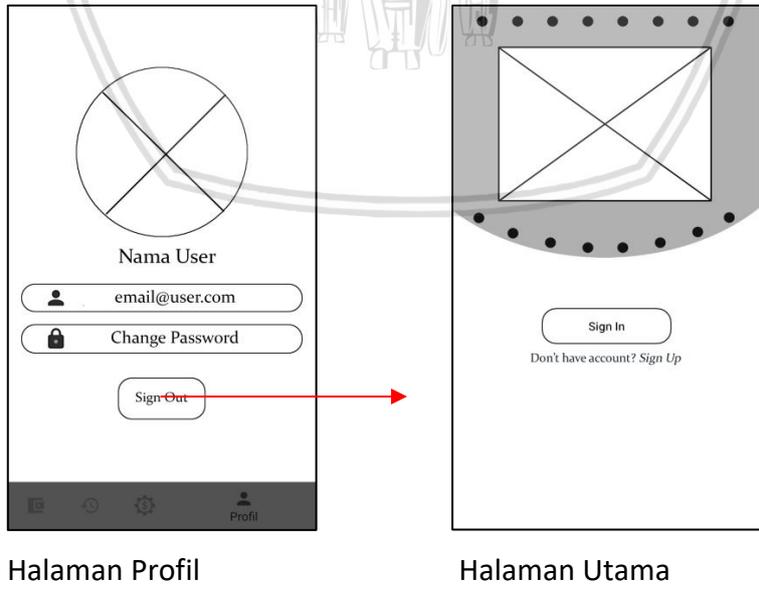
(xvi) Screenflow Change Password



Gambar 5. 45 Screenflow Change Password

Pada gambar 5.45 merupakan *screenflow* untuk melihat halaman *Change Password*. Tombol *Change Password* yang terletak pada halaman Profil ditekan, maka halaman *Change Password* akan ditampilkan.

(xvii) Screenflow Sign Out



Gambar 5. 46 Screenflow Sign Out

Pada gambar 5.46 merupakan *screenflow* untuk keluar dari sistem sebagai member. Tombol *Sign Out* yang terletak pada halaman Profil ditekan, maka akan keluar sistem dan kembali pada halaman Utama.

5.2 Implementasi

Pada sub bab ini akan dijelaskan mengenai implementasi dari aplikasi FM App yang terdiri dari spesifikasi sistem, batasan implementasi, implementasi kode program, implementasi basis data, dan implementasi antarmuka.

5.2.1 Spesifikasi Sistem

Dalam pengembangan aplikasi FM App, pengembangan aplikasi dilakukan dengan menggunakan perangkat keras dan perangkat lunak. Pada perangkat keras terdiri dari perangkat keras untuk pengembangan kode program dan perangkat keras untuk menjalankan hasil kode program, sedangkan perangkat lunak digunakan dalam pembuatan kode program. Untuk lebih jelasnya dapat dilihat pada tabel 5.9 berikut.

Tabel 5. 9 Spesifikasi Sistem

Jenis Perangkat	Fungsi	Nama Komponen	Spesifikasi
Perangkat Keras	Untuk pengembangan kode program	Nama Model	Acer Aspire E5-475G
		Prosesor	Intel Core i5-7200U
		RAM	12GB
		Harddisk	1TB
		Sistem Operasi	Windows 10
	Untuk menjalankan hasil kode program	Nama Model	Redmi Note 2
		RAM	2 GB
		Harddisk	16 GB
		Sistem Operasi	Android
		Android Version	5.0.2 (Lollipop)
Perangkat Lunak	Untuk pengembangan kode program	Nama	Android Studio
		Versi	3.0

5.2.2 Batasan Implementasi

Berikut merupakan batasan implementasi FM App:

1. FM App dikembangkan dengan menggunakan bahasa pemrograman Java untuk *back end* kode program, XML untuk pembuatan tampilan, Firebase *Database* untuk *database*, dan Firebase Auth untuk autentikasi data *member*.
2. FM App memiliki fitur dasar, yaitu melihat informasi keuangan *member*, melihat informasi riwayat keuangan, membuat perencanaan keuangan, dan melihat informasi profil.
3. Data yang digunakan dalam pengembangan FM App diambil dari lembar kuesioner yang didapat dari mahasiswa FILKOM UB.

5.2.3 Implementasi Kode Program

Proses implementasi FM App menggunakan bahasa pemrograman Java untuk *back end* kode program dan XML untuk pembuatan tampilan. Pengembangan ini menggunakan konsep MVC. Dalam implementasi kode program akan dibagi menjadi dua bagian, yaitu *sprint* 1 dan *sprint* 2. Implementasi kode program yang dituliskan pada penelitian ini hanya beberapa kode program yang dianggap penting dalam pengembangan FM App.

5.2.3.1 Sprint 1

Pada implementasi kode program pada *sprint* 1 terdapat implementasi kode program *Sign In* dan Menambah Pengeluaran. Untuk lebih jelasnya dapat dilihat pada sub bab berikut.

(a) Implementasi Kode Program *Sign In*

Berikut tabel 5.10 yang berisi kode program pada method `signIn()` pada klas `SignIn` yang diimplementasikan menggunakan bahasa pemrograman Java.

Tabel 5. 10 Kode Program `SignIn.java`

No	Kode Program
1	<code>private void signIn(String email, String password) {</code>
2	<code> if (!validateForm()) {</code>
3	<code> return;</code>
4	<code> }</code>
5	<code> mAuth.signInWithEmailAndPassword(email, password)</code>
6	<code> .addOnCompleteListener(this, new</code>
7	<code>OnCompleteListener<AuthResult>() {</code>
8	<code> @Override</code>
9	<code> public void onComplete(@NonNull Task<AuthResult></code>
10	<code>task) {</code>
11	<code> if (task.isSuccessful()) {</code>
12	<code> // Sign in success, update UI with the</code>
13	<code>signed-in user's information</code>
14	<code> // Log.d(TAG, "signInWithEmail:success");</code>
15	<code> Toast.makeText(SignIn.this, "Sign in</code>
16	<code>success", Toast.LENGTH_SHORT).show();</code>
17	<code> FirebaseUser user =</code>
18	<code>mAuth.getCurrentUser();</code>
19	<code> Intent sign_in_page = new</code>
20	<code>Intent(getApplicationContext(), BottomNavigation.class);</code>
21	<code> startActivity(sign_in_page);</code>
22	<code> finish();</code>
23	<code> //updateUI(user);</code>



```

24         } else {
25             // If sign in fails, display a message to
26 the user.
27             //Log.w(TAG, "signInWithEmail:failure",
28 task.getException());
29             Toast.makeText(SignIn.this, "Wrong email
30 or password", Toast.LENGTH_SHORT).show();
31             //updateUI(null);
32         }
33
34             // [START_EXCLUDE]
35             if (!task.isSuccessful()) {
36
37 //mStatusTextView.setText(R.string.auth_failed);
38             }
39         }
40     });
41 }

```

Berikut penjelasan dari tabel kode program `SignIn.java` yang terdapat pada Tabel 5.14.

1. Baris 1 : deklarasi *method* `signIn()` yang berparameter `String email` dan `String password`.
2. Baris 2 – 4 : Memvalidasi masukan nilai `email` dan `password`.
3. Baris 5 – 7 : mengecek nilai `email` dan `password` pada *database*.
4. Baris 8 – 23 : pengecekan apabila nilai `email` dan `password` terdaftar pada *database*, maka “Sign in success” ditampilkan dan diteruskan pada kelas `BottomNavigation` dengan *Intent*.
5. Baris 24 – 41 : pengecekan apabila nilai `email` dan `password` tidak terdaftar pada *database*, maka ditampilkan “Wrong email or password”.

(b) Implementasi Kode Program Menambah Pengeluaran

Berikut tabel 5.11 yang berisi kode program *method* `DataUji()` pada kelas `TambahPengeluaran` yang diimplementasikan menggunakan bahasa pemrograman Java.

Tabel 5. 11 Kode Program `TambahPengeluaran.java`

No	Kode Program
1	<code>private void DataUji(int jumlahPengeluaranInt) {</code>
2	<code> //set jumlah pengeluaran tidak dianggarkan</code>
3	<code> final double jumlahPengeluaran = (double)</code>
4	<code>jumlahPengeluaranInt;</code>
5	<code> final FirebaseUser currentUser = mAuth.getCurrentUser();</code>
6	<code> users.addListenerForSingleValueEvent(new ValueEventListener()</code>
7	<code>{</code>
8	<code> @Override</code>
9	<code> public void onDataChange(DataSnapshot dataSnapshot) {</code>
10	<code> //liburan (tersier)</code>
11	<code> String anggaranLiburan =</code>
12	<code>dataSnapshot.child(currentUser.getId()).child("Anggaran").child("</code>
13	<code>liburan").getValue(String.class);</code>
14	<code> String totalLiburan =</code>
15	<code>dataSnapshot.child(currentUser.getId()).child("Total</code>
16	<code>Pengeluaran").child("liburan").getValue(String.class);</code>
17	<code> double anggaranLiburanD =</code>



```

18 Double.parseDouble(anggaranLiburan);
19     double totalLiburanD =
20 Double.parseDouble(totalLiburan);
21     double sisaAnggaranLiburan = anggaranLiburanD -
22 totalLiburanD - jumlahPengeluaran;
23     if (sisaAnggaranLiburan >= 0) {
24         //masuk ke perencanaan alokasi dana liburan
25         String alokasiLiburan =
26 String.valueOf(jumlahPengeluaran);
27         final RencanaAlokasiDana alokasiDana = new
28 RencanaAlokasiDana(alokasiLiburan, "0", "0", "0", "0", "0", "0",
29 "0", et_note.getText().toString());
30         users.child(currentUser.getId()).child("Rencana
31 Alokasi Dana").setValue(alokasiDana);
32         final DataUjiNaiveBayes dataUjiNaiveBayes = new
33 DataUjiNaiveBayes("Ya", "Tidak", "Tidak");
34         users.child(currentUser.getId()).child("Data
35 Uji").setValue(dataUjiNaiveBayes);
36
37     } else {
38         //masuk ke perencanaan alokasi dana liburan
39         double alokasiLiburanD = jumlahPengeluaran +
40 sisaAnggaranLiburan; // ditambah karena nilai sisa anggaran
41 liburan (-)
42         String alokasiLiburan =
43 String.valueOf(alokasiLiburanD); // alokasi liburan
44
45         //hobi dan hiburan (tersier)
46         String anggaranHobiHiburan =
47 dataSnapshot.child(currentUser.getId()).child("Anggaran").child("
48 hobiDanHiburan").getValue(String.class);
49         String totalHobiHiburan =
50 dataSnapshot.child(currentUser.getId()).child("Total
51 Pengeluaran").child("hobiHiburan").getValue(String.class);
52         double anggaranHobiHiburanD =
53 Double.parseDouble(anggaranHobiHiburan);
54         double totalHobiHiburanD =
55 Double.parseDouble(totalHobiHiburan);
56         double sisaAnggaranHobiHiburan =
57 anggaranHobiHiburanD - totalHobiHiburanD + sisaAnggaranLiburan;
58         if (sisaAnggaranHobiHiburan >= 0) {
59             double sisaAnggaranLiburanBaru = -
60 sisaAnggaranLiburan;
61             String alokasiHobiHiburan =
62 String.valueOf(sisaAnggaranLiburanBaru);
63             final RencanaAlokasiDana alokasiDana = new
64 RencanaAlokasiDana(alokasiLiburan, alokasiHobiHiburan, "0", "0",
65 "0", "0", "0", "0", et_note.getText().toString());
66
67         users.child(currentUser.getId()).child("Rencana Alokasi
68 Dana").setValue(alokasiDana);
69         final DataUjiNaiveBayes dataUjiNaiveBayes =
70 new DataUjiNaiveBayes("Ya", "Tidak", "Tidak");
71         users.child(currentUser.getId()).child("Data
72 Uji").setValue(dataUjiNaiveBayes);
73     } else {
74         //masuk ke perencanaan alokasi dana Hobi dan
75 Hiburan
76         double alokasiHobiHiburanD = -
77 sisaAnggaranLiburan + sisaAnggaranHobiHiburan; //ditambah karena
78 nilai sisa anggaran hobi dan hiburan (-)
79         String alokasiHobiHiburan =
80 String.valueOf(alokasiHobiHiburanD);
81
82         //zakat/amal (sekunder)

```

```

83         String anggaranZakatAmal =
84 dataSnapshot.child(currentUser.getUid()).child("Anggaran").child("
85 zakatAtauAmal").getValue(String.class);
86         String totalZakatAmal =
87 dataSnapshot.child(currentUser.getUid()).child("Total
88 Pengeluaran").child("zakatAmal").getValue(String.class);
89         double anggaranZakatAmalD =
90 Double.parseDouble(anggaranZakatAmal);
91         double totalZakatAmalD =
92 Double.parseDouble(totalZakatAmal);
93         double sisaAnggaranZakatAmal =
94 anggaranZakatAmalD - totalZakatAmalD + sisaAnggaranHobiHiburan;
95         if (sisaAnggaranZakatAmal >= 0) {
96             double sisaAnggaranHobiHiburanBaru = -
97 sisaAnggaranHobiHiburan;
98             String alokasiZakatAmal =
99 String.valueOf(sisaAnggaranHobiHiburanBaru);
100            final RencanaAlokasiDana alokasiDana = new
101 RencanaAlokasiDana(alokasiLiburan, alokasiHobiHiburan,
102 alokasiZakatAmal, "0", "0", "0", "0", "0",
103 et_note.getText().toString());
104
105 users.child(currentUser.getUid()).child("Rencana Alokasi
106 Dana").setValue(alokasiDana);
107            final DataUjiNaiveBayes dataUjiNaiveBayes
108 = new DataUjiNaiveBayes("Ya", "Ya", "Tidak");
109
110 users.child(currentUser.getUid()).child("Data
111 Uji").setValue(dataUjiNaiveBayes);
112        } else {
113            //masukan ke perencanaan alokasi dana
114 zakat amal
115            double alokasiZakatAmalD = -
116 sisaAnggaranHobiHiburan + sisaAnggaranZakatAmal; //ditambah karena
117 nilai sisa anggaran zakat/amal (-)
118            String alokasiZakatAmal =
119 String.valueOf(alokasiZakatAmalD);
120
121            //gadget (sekunder)
122            String anggaranGadget =
123 dataSnapshot.child(currentUser.getUid()).child("Anggaran").child("
124 gadget").getValue(String.class);
125            String totalGadget =
126 dataSnapshot.child(currentUser.getUid()).child("Total
127 Pengeluaran").child("gadget").getValue(String.class);
128            double anggaranGadgetD =
129 Double.parseDouble(anggaranGadget);
130            double totalGadgetD =
131 Double.parseDouble(totalGadget);
132            double sisaAnggaranGadget =
134 anggaranGadgetD - totalGadgetD + sisaAnggaranZakatAmal;
135            if (sisaAnggaranGadget >= 0) {
136                double sisaAnggaranZakatAmalBaru = -
137 sisaAnggaranZakatAmal;
138                String alokasiGadget =
139 String.valueOf(sisaAnggaranZakatAmalBaru);
140                final RencanaAlokasiDana alokasiDana =
141 new RencanaAlokasiDana(alokasiLiburan, alokasiHobiHiburan,
142 alokasiZakatAmal, alokasiGadget, "0", "0", "0", "0",
143 et_note.getText().toString());
144
145 users.child(currentUser.getUid()).child("Rencana Alokasi
146 Dana").setValue(alokasiDana);
147            final DataUjiNaiveBayes
148 dataUjiNaiveBayes = new DataUjiNaiveBayes("Ya", "Ya", "Tidak");

```

```

149
150 users.child(currentUser.getUid()).child("Data
151 Uji").setValue(dataUjiNaiveBayes);
152     } else {
153         //masukan ke perencanaan alokasi dana
154 gadget
155         double alokasiGadgetD = -
156 sisaAnggaranZakatAmal + sisaAnggaranGadget; //ditambah karena
157 nilai sisa anggaran gadget (-)
158         String alokasiGadget =
159 String.valueOf(alokasiGadgetD);
160
161         ////pendidikan (sekunder)
162         String anggaranPendidikan =
163 dataSnapshot.child(currentUser.getUid()).child("Anggaran").child("
164 pendidikan").getValue(String.class);
165         String totalPendidikan =
166 dataSnapshot.child(currentUser.getUid()).child("Total
167 Pengeluaran").child("pendidikan").getValue(String.class);
168         double anggaranPendidikanD =
169 Double.parseDouble(anggaranPendidikan);
170         double totalPendidikanD =
171 Double.parseDouble(totalPendidikan);
172         double sisaAnggaranPendidikan =
173 anggaranPendidikanD + sisaAnggaranGadget;
174         if (sisaAnggaranPendidikan >= 0) {
175             double sisaAnggaranGadgetBaru = -
176 sisaAnggaranGadget;
177             String alokasiPendidikan =
178 String.valueOf(sisaAnggaranGadgetBaru);
179             final RencanaAlokasiDana
180 alokasiDana = new RencanaAlokasiDana(alokasiLiburan,
181 alokasiHobiHiburan, alokasiZakatAmal, alokasiGadget,
182 alokasiPendidikan, "0", "0", "0", et_note.getText().toString());
183
184 users.child(currentUser.getUid()).child("Rencana Alokasi
185 Dana").setValue(alokasiDana);
186             final DataUjiNaiveBayes
187 dataUjiNaiveBayes = new DataUjiNaiveBayes("Ya", "Ya", "Tidak");
188
189 users.child(currentUser.getUid()).child("Data
190 Uji").setValue(dataUjiNaiveBayes);
191         } else {
192             //masukan ke perencanaan alokasi
193 dana pendidikan
194             double alokasiPendidikanD = -
195 sisaAnggaranGadget + sisaAnggaranPendidikan; //ditambah karena
196 nilai sisa anggaran pendidikan (-)
197             String alokasiPendidikan =
198 String.valueOf(alokasiPendidikanD);
199
200             //transportasi (primer)
201             String anggaranTransportasi =
202 dataSnapshot.child(currentUser.getUid()).child("Anggaran").child("
203 transportasi").getValue(String.class);
204             String totalTransportasi =
205 dataSnapshot.child(currentUser.getUid()).child("Total
206 Pengeluaran").child("transportasi").getValue(String.class);
207             double anggaranTransportasiD =
208 Double.parseDouble(anggaranTransportasi);
209             double totalTransportasiD =
210 Double.parseDouble(totalTransportasi);
211             double sisaAnggaranTransportasi =
212 anggaranTransportasiD - totalTransportasiD +
213 sisaAnggaranPendidikan;

```

```

214         if (sisaAnggaranTransportasi >= 0)
215     {
216         double
217     sisaAnggaranPendidikanBaru = -sisaAnggaranPendidikan;
218         String alokasiTransportasi =
219     String.valueOf(sisaAnggaranPendidikanBaru);
220         final RencanaAlokasiDana
221     alokasiDana = new RencanaAlokasiDana(alokasiLiburan,
222     alokasiHobiHiburan, alokasiZakatAmal, alokasiGadget,
223     alokasiPendidikan, alokasiTransportasi, "0", "0",
224     et_note.getText().toString());
225
226     users.child(currentUser.getId()).child("Rencana Alokasi
227     Dana").setValue(alokasiDana);
228         final DataUjiNaiveBayes
229     dataUjiNaiveBayes = new DataUjiNaiveBayes("Ya", "Ya", "Ya");
230
231     users.child(currentUser.getId()).child("Data
232     Uji").setValue(dataUjiNaiveBayes);
233     } else {
234         //masukan ke perencanaan
235     alokasi dana transportasi
236         double alokasiTransportasiD =
237     -sisaAnggaranPendidikan + sisaAnggaranTransportasi; //ditambah
238     karena nilai sisa anggaran transportasi (-)
239         String alokasiTransportasi =
240     String.valueOf(alokasiTransportasiD);
241
242         //kesehatan (primer)
243         String anggaranKesehatan =
244     dataSnapshot.child(currentUser.getId()).child("Anggaran").child("
245     kesehatan").getValue(String.class);
246         String totalKesehatan =
247     dataSnapshot.child(currentUser.getId()).child("Total
248     Pengeluaran").child("kesehatan").getValue(String.class);
249         double anggaranKesehatanD =
250     Double.parseDouble(anggaranKesehatan);
251         double totalKesehatanD =
252     Double.parseDouble(totalKesehatan);
253         double sisaAnggaranKesehatan =
254     anggaranKesehatanD - totalKesehatanD + sisaAnggaranTransportasi;
255         if (sisaAnggaranKesehatan >=
256     0) {
257             double
258         sisaAnggaranTransportasiBaru = -sisaAnggaranTransportasi;
259             String alokasiKesehatan =
260         String.valueOf(sisaAnggaranTransportasiBaru);
261             final RencanaAlokasiDana
262         alokasiDana = new RencanaAlokasiDana(alokasiLiburan,
263         alokasiHobiHiburan, alokasiZakatAmal, alokasiGadget,
264         alokasiPendidikan, alokasiTransportasi, alokasiKesehatan, "0",
265         et_note.getText().toString());
266
267         users.child(currentUser.getId()).child("Rencana Alokasi
268         Dana").setValue(alokasiDana);
269             final DataUjiNaiveBayes
270         dataUjiNaiveBayes = new DataUjiNaiveBayes("Ya", "Ya", "Ya");
271
272         users.child(currentUser.getId()).child("Data
273         Uji").setValue(dataUjiNaiveBayes);
274         } else {
275             //masukan ke perencanaan
276         alokasi dana kesehatan
277             double alokasiKesehatanD =
278         -sisaAnggaranTransportasi + sisaAnggaranKesehatan; //ditambah

```

```

279 karena nilai sisa anggaran Kesehatan (-)
280                                     String alokasiKesehatan =
281 String.valueOf(alokasiKesehatanD);
282
283                                     //harian (primer)
284                                     String anggaranHarian =
285 dataSnapshot.child(currentUser.getUId()).child("Anggaran").child("
286 harian").getValue(String.class);
287                                     String totalHarian =
288 dataSnapshot.child(currentUser.getUId()).child("Total
289 Pengeluaran").child("harian").getValue(String.class);
290                                     double anggaranHarianD =
291 Double.parseDouble(anggaranHarian);
292                                     double totalHarianD =
293 Double.parseDouble(totalHarian);
294                                     double sisaAnggaranHarian
295 = anggaranHarianD - totalHarianD + sisaAnggaranKesehatan;
296                                     if (sisaAnggaranHarian >=
297 0) {
298                                     double
299 sisaAnggaranKesehatanBaru = -sisaAnggaranKesehatan;
300                                     String alokasiHarian =
301 String.valueOf(sisaAnggaranKesehatanBaru);
302                                     final
303 RencanaAlokasiDana alokasiDana = new
304 RencanaAlokasiDana(alokasiLiburan, alokasiHobiHiburan,
305 alokasiZakatAmal, alokasiGadget, alokasiPendidikan,
306 alokasiTransportasi, alokasiKesehatan, alokasiHarian,
307 et_note.getText().toString());
308
309 users.child(currentUser.getUId()).child("Rencana Alokasi
310 Dana").setValue(alokasiDana);
311                                     final
312 DataUjiNaiveBayes dataUjiNaiveBayes = new DataUjiNaiveBayes("Ya",
313 "Ya", "Ya");
314
315 users.child(currentUser.getUId()).child("Data
316 Uji").setValue(dataUjiNaiveBayes);
317                                     } else {
318                                     //masuk ke
319 perencanaan alokasi dana Harian
320                                     double alokasiHarianD
321 = -sisaAnggaranKesehatan + sisaAnggaranHarian; //ditambah karena
322 nilai sisa anggaran Kesehatan (-)
323                                     String alokasiHarian =
324 String.valueOf(alokasiHarianD);
325                                     final
326 RencanaAlokasiDana alokasiDana = new
327 RencanaAlokasiDana(alokasiLiburan, alokasiHobiHiburan,
328 alokasiZakatAmal, alokasiGadget, alokasiPendidikan,
329 alokasiTransportasi, alokasiKesehatan, alokasiHarian,
330 et_note.getText().toString());
331
332 users.child(currentUser.getUId()).child("Rencana Alokasi
333 Dana").setValue(alokasiDana);
334                                     final
335 DataUjiNaiveBayes dataUjiNaiveBayes = new DataUjiNaiveBayes("Ya",
336 "Ya", "Ya");
337
338 users.child(currentUser.getUId()).child("Data
339 Uji").setValue(dataUjiNaiveBayes);
340
341                                     }
342                                     }
343                                     }

```

```

344         }
345     }
346 }
347 }
348 }
349 }
350
351     @Override
352     public void onCancelled(DatabaseError databaseError) {
353
354     }
355 });
356 }
    
```

Berikut penjelasan dari tabe kode program `TambahPengeluaran.java` yang terdapat pada Tabel 5.11.

1. Baris 1 : deklarasi *method* `DataUji()` dengan parameter `int jumlahPengeluaranInt`
2. Baris 2 – 4 : inialisasi nilai jumlah pengelaura tidak dianggarkan
3. Baris 5 – 9 : mengambil data ke firebase dari pengguna yang aktif pada aplikasi.
4. Baris 10 – 37 : merencanakan alokasi dana pada kategori liburan dan menginisialisasi nilai data uji naive bayes.
5. Baris 38 – 73 : merencanakan alokasi dana pada kategori hobi dan hiburan dan menginisialisasi nilai data uji naive bayes.
6. Baris 74 – 112 : merencanakan alokasi dana pada kategori zakat/amal dan menginisialisasi nilai data uji naive bayes.
7. Baris 113 – 152 : merencanakan alokasi dana pada kategori gadget dan menginisialisasi nilai data uji naive bayes.
8. Baris 153 – 191 : merencanakan alokasi dana pada kategori pendidikan dan menginisialisasi nilai data uji naive bayes.
9. Baris 192 – 233 : merencanakan alokasi dana pada kategori transportasi dan menginisialisasi nilai data uji naive bayes.
10. Baris 234 – 274 : merencanakan alokasi dana pada kategori kesehatan dan menginisialisasi nilai data uji naive bayes.
11. Baris 275 – 317 : merencanakan alokasi dana pada kategori harian dan menginisialisasi nilai data uji naive bayes.
12. Baris 318 – 356 : merencanakan alokasi dana pada kategori harian dan menginisialisasi nilai data uji naive bayes.

5.2.3.2 Sprint 2

(a) Implementasi Kode Program Membuat Perencanaan Anggaran

Berikut tabel 5.12 yang berisi kode program `onClick()` pada klas `BudgetPlanning` yang diimplementasikan menggunakan bahasa pemrograman Java.

Tabel 5. 12 Kode Program `BudgetPlanning.java`

No	Kode Program
1	<code>public void onClick(View view) {</code>
2	<code>switch (view.getId()) {</code>

```

3         case R.id.ganti_halaman:
4             //pindah halaman ke tambah pemasukan
5
6         fragmentManager().beginTransaction().replace(R.id.main_frame,
7         new KeteranganAnggaran()).addToBackStack(null).commit();
8             break;
9         case R.id.submit_button_budget:
10            if (!validateForm()) {
11                return;
12            }
13
14            final FirebaseUser currentUser =
15            mAuth.getCurrentUser();
16            users.addListenerForSingleValueEvent(new
17            ValueEventListener() {
18                @Override
19                public void onDataChange(DataSnapshot
20                dataSnapshot) {
21                    FirebaseUser user;
22                    String userID;
23                    String gadget =
24                    et_gadget.getText().toString();
25                    String harian =
26                    et_harian.getText().toString();
27                    String hobiHiburan =
28                    et_hobi_dan_hiburan.getText().toString();
29                    String kesehatan =
30                    et_kesehatan.getText().toString();
31                    String liburan =
32                    et_liburan.getText().toString();
33                    String pendidikan =
34                    et_pendidikan.getText().toString();
35                    String transportasi =
36                    et_transportasi.getText().toString();
37                    String zakatAmal =
38                    et_zakat_atau_amal.getText().toString();
39
40                    double gadgetDouble =
41                    Double.parseDouble(gadget);
42                    double harianDouble =
43                    Double.parseDouble(harian);
44                    double hobiHiburanDouble =
45                    Double.parseDouble(hobiHiburan);
46                    double kesehatanDouble =
47                    Double.parseDouble(kesehatan);
48                    double liburanDouble =
49                    Double.parseDouble(liburan);
50                    double pendidikanDouble =
51                    Double.parseDouble(pendidikan);
52                    double transportasiDouble =
53                    Double.parseDouble(transportasi);
54                    double zakatAmalDouble =
55                    Double.parseDouble(zakatAmal);
56
57                    double total = gadgetDouble + harianDouble +
58                    hobiHiburanDouble + kesehatanDouble + liburanDouble
59                    + pendidikanDouble +
60                    transportasiDouble + zakatAmalDouble;
61                    System.out.println("total anggaran = "
62                    +total);
63
64                    //manggil total uang user
65                    String totalUang =
66                    dataSnapshot.child(currentUser.getId()).child("Total
67                    Uang").child("totalUang").getValue(String.class);

```

```

68         double totalUangDouble =
69 Double.parseDouble(totalUang);
70         double cek = totalUangDouble - total;
71         System.out.println("total cek = "+cek);
72         if(cek >= 0){
73             //untuk menyimpan data anggaran ke model
74     class Anggaran
75             final Anggaran anggaranUser = new
76 Anggaran(gadget, harian, hobiHiburan, kesehatan, liburan,
77 pendidikan, transportasi, zakatAmal);
78
79             //untuk menulis data yang sudah di
80 register ke firebase
81             user = mAuth.getCurrentUser();
82             userID = user.getUid();
83
84             //untuk menulis data yang sudah di
85 register ke firebase
86
87 users.child(userID).child("Anggaran").setValue(anggaranUser);
88
89             //untuk info berhasil di upload
90 Toast.makeText(getActivity(), "Perencanaan
91 anggaran berhasil ditambah!", Toast.LENGTH_SHORT).show();
92
93             et_gadget.setText("");
94             et_harian.setText("");
95             et_hobi_dan_hiburan.setText("");
96             et_kesehatan.setText("");
97             et_liburan.setText("");
98             et_pendidikan.setText("");
99             et_transportasi.setText("");
100            et_zakat_atau_amal.setText("");
101        } else {
102            //untuk info tidak berhasil di upload
103 Toast.makeText(getActivity(), "Total
104 anggaran melebihi total uang Anda! Tambah pemasukan Anda!",
105 Toast.LENGTH_LONG).show();
106            et_gadget.setText("");
107            et_harian.setText("");
108            et_hobi_dan_hiburan.setText("");
109            et_kesehatan.setText("");
110            et_liburan.setText("");
111            et_pendidikan.setText("");
112            et_transportasi.setText("");
113            et_zakat_atau_amal.setText("");
114        }
115    }
116
117    @Override
118    public void onCancelled(DatabaseError
119 databaseError) {
120
121    }
122    });
123    break;
124    }
125 }

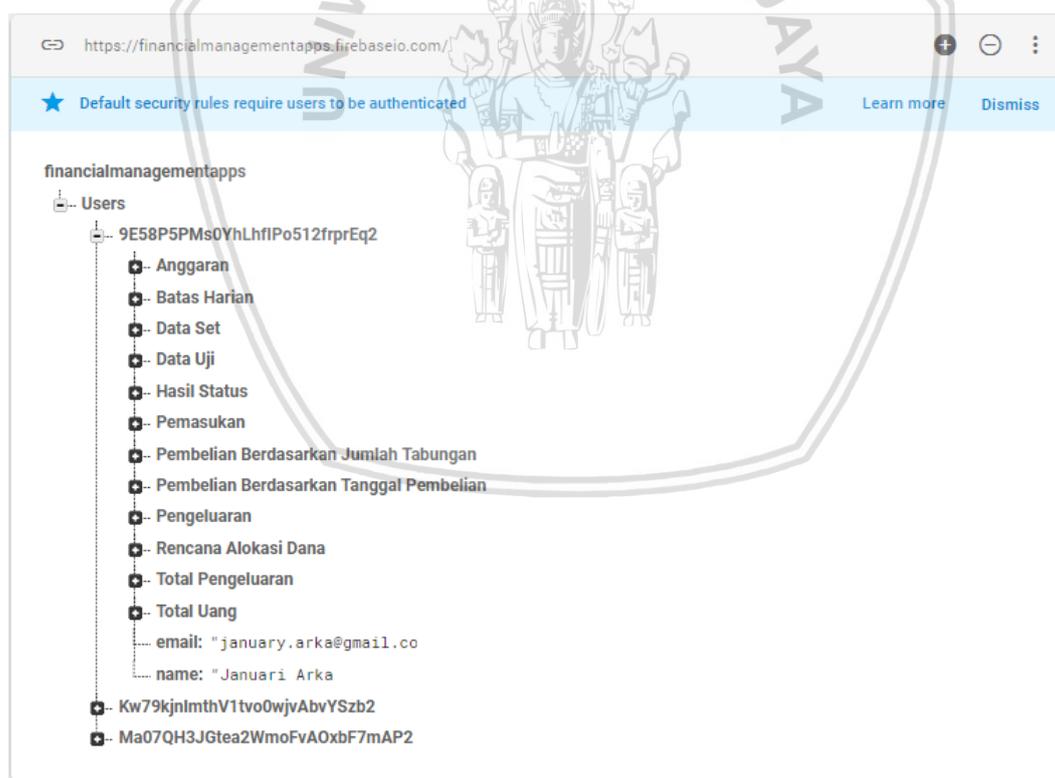
```

Berikut penjelasan dari tabe kode program BudgetPlanning.java yang terdapat pada Tabel 5.12.

1. Baris 1 : deklarasi *method* onClick() .
2. Baris 2 : deklarasi kondisi berdasarkan view.getID() .

3. Baris 3 – 8 : jika tombol here diklik, maka keterangan kategori anggaran akan ditampilkan.
4. Baris 9 : deklarasi tombol *submit* ditekan.
5. Baris 10 – 12 : kondisi pengecekan pengisian *form* pada halaman perencanaan anggaran
6. Baris 13 – 20 : jika nilai `validateForm()` adalah `true`, maka pengambilan data pengguna aktif pada aplikasi pada `firebase`.
7. Baris 21 – 56 : inialisasi nilai kategori anggaran berdasarkan masukan pengguna.
8. Baris 57 – 71 : deklarasi total perencanaan anggaran dan keuangan yang dimiliki
9. Baris 72 – 101 : deklarasi kondisi jika total keuangan yang dimiliki lebih besar dari total perencanaan anggaran, maka data perencanaan anggaran akan disimpan pada *database*.
10. Baris 102 – 125 : deklarasi kondisi jika total keuangan yang dimiliki lebih kecil dari total perencanaan anggaran, maka data perencanaan anggaran tidak disimpan pada *database*.

5.2.4 Implementasi Basis Data



Gambar 5. 47 Implementasi Basis Data FM App

Pada gambar 5.47 menjelaskan tentang implementasi basis data yang dibutuhkan pada FM App. *Database* pada `firebase` tersebut digunakan untuk tempat menyimpan data interaksi yang dilakukan pengguna atau *member* pada FM App. Pada *sprint 1*, *parent* `Users`, *child* `push ID user`, *child* `Data Uji`, *child* `Pemasukan`, *child*

Pengeluaran, *child* Rencana Alokasi Dana, *child* Total Pengeluaran, dan *child* Total Uang dibuat. Pada *sprint 2*, dilakukan penambahan pada implementasi basis data yang sudah dibuat pada *sprint 1*, yaitu *child* Anggaran, *child* Batas Harian, *child* Pembelian Berdasarkan Jumlah Tabungan, dan *child* Pembelian Berdasarkan Tanggal Pembelian yang merupakan *child* dari *push ID user* dan *parent* Users.

5.2.5 Implementasi Antarmuka

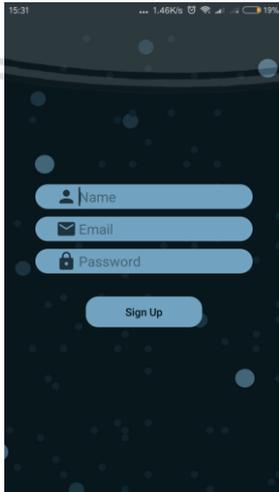
5.2.5.1 Tampilan Halaman Utama

Pada gambar 5.48 merupakan gambar tampilan utama dari aplikasi FM App. Pada halaman ini pengguna dapat mendaftarkan diri sebagai *member* dengan menekan tombol *Sign Up* atau masuk ke dalam sistem sebagai *member* dengan menekan tombol *Sign In*.



Gambar 5. 48 Tampilan Halaman Utama

5.2.5.2 Tampilan Sign Up

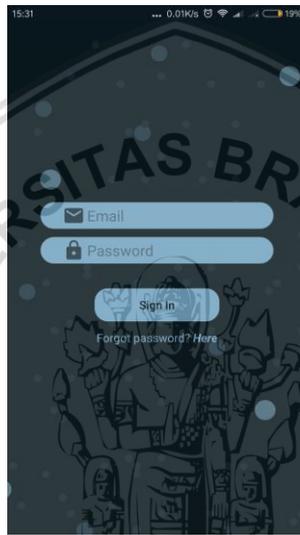


Gambar 5. 49 Tampilan Sign Up

Pada gambar 5.49 merupakan gambaran dari halaman *sign up*. Pada halaman ini disediakan kolom masukan nama, *email*, dan *password* untuk mendaftarkan diri sebagai *member* ke dalam sistem.

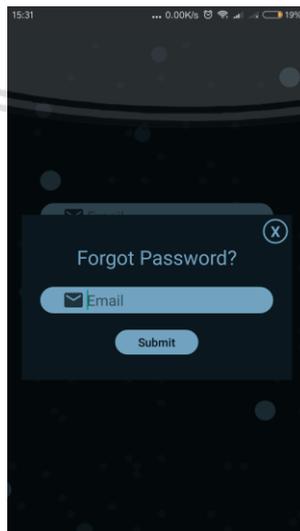
5.2.5.3 Tampilan *Sign In*

Pada gambar 5.50 merupakan gambar halaman *sign in*. Pada halaman ini terdapat kolom masukan *email* dan kolom masukan *password*. Kolom masukan *email* digunakan untuk memasukan *email* pengguna yang sudah terdaftar pada sistem dan kolom masukan *password* digunakan untuk memasukan *password* pengguna yang sudah terdaftar pada sistem. Selain itu, terdapat dua tombol, yaitu tombol *sign in* dan *here*. Tombol *sign in* digunakan untuk autentikasi data masukan *email* dan *password* ke *database* dan tombol *here* digunakan untuk lupa *password*.



Gambar 5. 50 Tampilan *Sign In*

5.2.5.4 Tampilan *Forgot Password*



Gambar 5. 51 Tampilan *Forgot Password*



Pada gambar 5.51 merupakan gambar tampilan *pop up menu forgot password*. Pada *pop up menu* ini terdapat kolom masukan *email* yang digunakan untuk memasukan *email* pengguna yang terdaftar pada sistem. Selain itu, terdapat tombol *submit* untuk memroses *forgot password* tersebut.

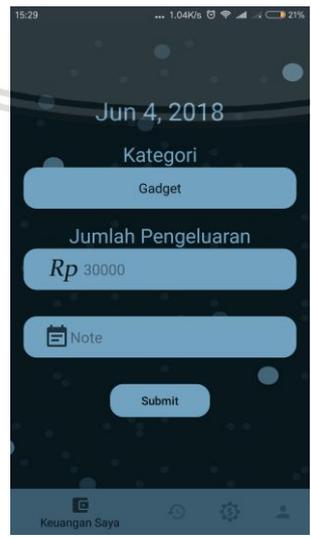
5.2.5.5 Tampilan Keuangan Saya

Pada gambar 5.52 merupakan gambar halaman keuangan saya. Pada halaman ini terdapat informasi mengenai keuangan kita, dimulai dari total uang yang dimiliki, diagram anggaran tiap kategori, hingga sisa batas harian. Selain itu, pada halaman ini juga terdapat tiga tombol, yaitu tombol reset diagram per bulan untuk reset diagram tiap bulan, tombol tambah pengeluaran untuk menambah pengeluaran, dan tombol tambah pemasukan untuk menambah pemasukan.



Gambar 5. 52 Tampilan Keuangan Saya

5.2.5.6 Tampilan Menambah Pengeluaran



Gambar 5. 53 Tampilan Menambah Pengeluaran

Pada gambar 5.53 merupakan gambar halaman menambah pengeluaran. Pada halaman ini terdapat pilihan kategori pengeluaran, kolom masukan jumlah pengeluaran, kolom masukan *note*, dan tombol *submit* untuk memroses data pengeluaran.

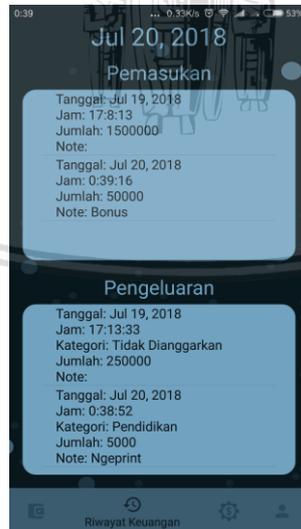
5.2.5.7 Tampilan Menambah Pemasukan

Pada gambar 5.54 merupakan gambar halaman menambah pemasukan. Pada halaman ini terdapat kolom masukan jumlah pemasukan, kolom masukan *note*, dan tombol *submit* untuk memroses data pemasukan.



Gambar 5. 54 Tampilan Menambah Pemasukan

5.2.5.8 Tampilan Riwayat Keuangan



Gambar 5. 55 Tampilan Riwayat Keuangan

Pada gambar 5.55 merupakan gambar halaman riwayat keuangan. Pada halaman ini akan diinformasikan seluruh pemasukan dan pengeluaran yang pernah kita lakukan.

5.2.5.9 Tampilan Perencanaan Keuangan

Pada gambar 5.56 merupakan gambar halaman perencanaan keuangan. Pada halaman ini terdapat tiga pilihan tombol, yaitu tombol perencanaan anggaran untuk membuat perencanaan anggaran, tombol batas harian untuk membuat batas harian, dan tombol perencanaan pembelian untuk membuat perencanaan pembelian.



Gambar 5. 56 Tampilan Perencanaan Keuangan

5.2.5.10 Tampilan Perencanaan Anggaran



Gambar 5. 57 Tampilan Perencanaan Anggaran

Pada gambar 5.57 merupakan gambar halaman perencanaan anggaran. Pada halaman ini terdapat 8 kolom masukan berdasarkan kategori anggaran. Selain itu, pada halaman ini juga terdapat dua tombol, yaitu tombol *here* untuk menampilkan halaman keterangan kategori anggaran dan tombol *submit* untuk memroses data perencanaan anggaran.

5.2.5.11 Tampilan Keterangan Kategori Anggaran

Pada gambar 5.58 merupakan gambar halaman keterangan kategori anggaran. Pada halaman ini menampilkan informasi tiap kategori anggaran yang ada pada halaman perencanaan anggaran.



Gambar 5. 58 Tampilan Keterangan Kategori Anggaran

5.2.5.12 Tampilan Batas Harian



Gambar 5. 59 Tampilan Batas Harian

Pada gambar 5.59 merupakan gambar halaman batas harian. Pada halaman ini terdapat kolom masukan jumlah batas harian, tombol *toggle reminder*, dan tombol *submit* untuk memroses data batas harian.

5.2.5.13 Tampilan Perencanaan Pembelian

Pada gambar 5.60 merupakan gambar halaman perencanaan pembelian. Pada halaman ini terdapat diagram target nabung, jumlah nabung per harinya, dan tombol *reset* pembelian untuk membatalkan perencanaan pembelian. Selain itu, terdapat

pilihan tombol pembelian, yaitu tombol berdasarkan jumlah tabungan dan tombol berdasarkan tanggal pembelian.



Gambar 5. 60 Tampilan Perencanaan Pembelian

5.2.5.14 Tampilan Berdasarkan Jumlah Tabungan



Gambar 5. 61 Tampilan Berdasarkan Jumlah Tabungan

Pada gambar 5.61 merupakan gambar halaman pembelian berdasarkan jumlah tabungan. Pada halaman ini terdapat kolom masukan harga produk untuk harga produk yang ingin dibeli, kolom masukan nabung per hari untuk jumlah nabung yang *member* inginkan per harinya, tombol *toggle reminder* untuk mengaktifkan *reminder* jika tabungan sudah siap, dan tombol *submit* untuk memroses data pembelian berdasarkan jumlah tabungan.

5.2.5.15 Tampilan Berdasarkan Tanggal Pembelian

Pada gambar 5.62 merupakan gambar halaman pembelian berdasarkan tanggal pembelian. Pada halaman ini terdapat kolom masukan harga produk untuk menentukan

harga produk yang ingin dibeli, pilihan tanggal pembelian, tombol *toggle reminder*, dan tombol *submit* untuk memroses data perencanaan pembelian berdasarkan tanggal.



Gambar 5. 62 Tampilan Berdasarkan Tanggal Pembelian

5.2.5.16 Tampilan Profil



Gambar 5. 63 Tampilan Profil

Pada gambar 5.63 merupakan gambar halaman profil. Pada halaman ini terdapat nama *member*, *email* yang digunakan *member*, tombol *change password* untuk mengubah *password*, dan tombol *sign out* untuk keluar dari sistem serta sebagai *member*.

5.3 Daily Scrum

Pada tahap ini merupakan evaluasi dari setiap kegiatan pada satu sprint. Evaluasi ini merupakan evaluasi terhadap apa saja yang telah dikerjakan, hambatan yang ada, dan kesaian target penyelesaian. Untuk lebih jelasnya akan dibagi menjadi dua bagian, yaitu pada *sprint 1* dan *sprint 2*.



5.3.1 Sprint 1

Pada tahap *sprint* 1 terdapat permasalahan teknis terkait implementasi, yaitu *session member* tidak tersimpan, sehingga ketika kita tekan kembali pada *handphone*, sistem akan mengembalikan kita pada halaman utama, sedangkan kondisinya sudah *sign in*. Solusi dari permasalahan tersebut adalah menambahkan kondisi pengecekan *session* pada *method onStart* pada kelas *SignIn*. Disamping itu, keseluruhan *sprint backlog* pada *sprint* 1 dapat selesai dilakukan seluruhnya. Dengan selesainya fitur dasar pada *sprint* 1, maka pada *sprint* 2 dapat dilakukan implementasi fitur lanjutan pada FM app.

5.3.2 Sprint 2

Pada tahap *sprint* 2 dikerjakan pekerjaan lanjutan dari *sprint* 1, yaitu pembuatan fitur lanjutan. Dalam proses implementasinya terdapat permasalahan teknis, yaitu saat membuat batas harian data terus menerus *update* ke *database*, sehingga setiap pengeluaran dilakukan batas harian tidak berkurang. Solusinya dari permasalahan ini adalah mengganti *addValueEventListener* menjadi *AddForSingleEventListener* pada kelas *SpendingLimit*. Disamping itu, keseluruhan *sprint backlog* pada *sprint* 2 dapat selesai dilakukan seluruhnya.



BAB 6 PENGUJIAN

Pada bab ini akan dijelaskan mengenai pengujian dari hasil implementasi yang sudah dilakukan pada bab sebelumnya. Pengujian ini dilakukan berdasarkan *sprint* yang telah dilakukan, yaitu pengujian *sprint 1* dan *sprint 2*. Pengujian pada *sprint 1* adalah pengujian terhadap fungsi dasar dari FM App, sedangkan pengujian pada *sprint 2* adalah pengujian terhadap fungsi lanjutan dari FM App. Setelah pengujian selesai, maka dilakukan analisis hasil yang bertujuan untuk menyimpulkan hasil pengujian yang telah dilakukan.

6.1 Pengujian Unit

Pengujian unit menggunakan metode *white box testing* dengan *basis path testing*. Pengujian diawali dengan pembuatan *flowgraph* berdasarkan algoritma pada setiap fungsi. Setelah *flowgraph* sudah dibuat, maka pengujian *cyclomatic complexity* dilakukan untuk mendapatkan jumlah *path*. Setelah itu, pembuatan jalur *independent path* dan membuat *test case* pada jalur *independent path* tersebut. Untuk jumlah *cyclomatic complexity* diperoleh dengan tiga persamaan berikut:

$$V(G) = E - N + 2 \quad (6.1)$$

$$V(G) = P + 1 \quad (6.2)$$

$$V(G) = R \quad (6.3)$$

Keterangan rumus diatas adalah sebagai berikut,

$V(G)$: jumlah *cyclomatic complexity*

E : sisi atau *edge* (garis penghubung antar *node*)

N : jumlah simpul (*node*)

P : *predicate node* pada grafik alir

R : jumlah *region* pada *flow graph*

Tujuan dilakukannya pengujian adalah untuk memastikan beberapa algoritma yang memiliki prioritas tinggi telah mengimplementasikan sesuai dengan yang diharapkan. Pengujian ini akan dibagi menjadi dua bagian, yaitu berdasarkan *sprint 1* dan *sprint 2*.

6.1.1 Sprint 1

Pengujian unit pada *sprint 1* dilakukan pada algoritma *Sign In* dan algoritma Data Uji.

6.1.1.1 Pengujian algoritma *Sign In*

Pada tabel 6.1 menjelaskan bagian kode yang menjadi *node* pada pengujian algoritma *sign in* pada klas `SignIn` dengan *white box testing*.

Tabel 6. 1 Algoritma Sign In pada kelas SignIn

```

private void signIn(String email, String password) { 1
    if (!validateForm()) { 2
        return; } 3
    mAuth.signInWithEmailAndPassword(email, password)
        .addOnCompleteListener(this, new 4
    OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task 5
    {
        if (task.isSuccessful()) { 6
            // Sign in success, update UI with the signed-
            in user's information
            // Log.d(TAG, "signInWithEmail:success");
            Toast.makeText(SignIn.this, "Sign in success",
            Toast.LENGTH_SHORT).show();
            FirebaseUser user = mAuth.getCurrentUser();
            Intent sign_in_page = new 7
            Intent(getApplicationContext(), BottomNavigation.class);
            startActivity(sign_in_page);
            finish();
            //updateUI(user);
        } else {
            // If sign in fails, display a message to the
            user.
            //Log.w(TAG, "signInWithEmail:failure",
            task.getException());
            Toast.makeText(SignIn.this, "Wrong email or 8
            password", Toast.LENGTH_SHORT).show();
            //updateUI(null);
        }
    }); } 9
}
    
```

(a) Basis Path Testing

(i) Flow Graph

Pada gambar 6.1 menjelaskan flow graph algoritma *sign in*.

(ii) Cyclomatic Complexity

$$V(G) = E - N + 2 \tag{6.4}$$

$$V(G) = 10 - 9 + 2 \tag{6.5}$$

$$V(G) = 3 \tag{6.6}$$

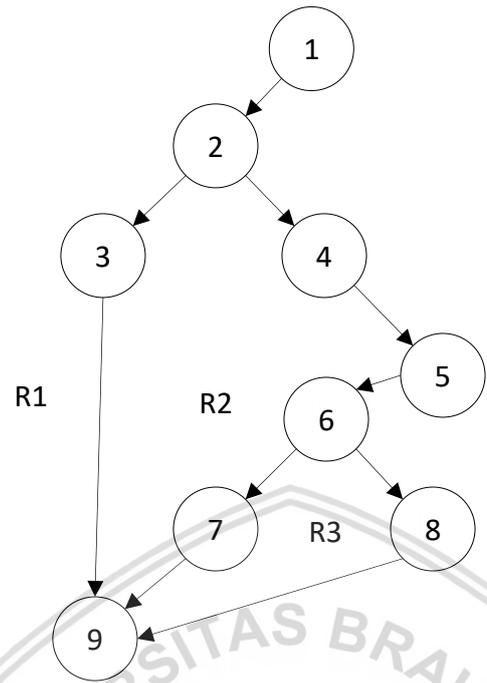
(iii) Independent Path

Jalur 1 = 1 - 2 - 3

Jalur 2 = 1 - 2 - 4 - 5 - 6 - 7 - 9

Jalur 3 = 1 - 2 - 4 - 5 - 6 - 8 - 9





Gambar 6. 1 Flow Graph Algoritma Sign In

(iv) Kasus Uji Algoritma Sign In

Hasil dari kasus uji algoritma sign in dapat dilihat pada tabel 6.2 dibawah ini.

Tabel 6. 2 Kasus Uji Algoritma Sign In

Jalur	Kasus Uji	Prosedur Uji	Expected Result	Test Result
1	Membuka halaman sign in	Mengosongkan kolom masukan email dan password, lalu menekan tombol sign in	Peringatan ditampilkan pada kolom masukan email dan password yang bertuliskan "Required"	Peringatan ditampilkan pada kolom masukan email dan password yang bertuliskan "Required"
2	Membuka halaman sign in	Memasukan email dan password terdaftar pada kolom masukan email dan password serta menekan tombol submit	Masuk kedalam sistem sebagai member dan menampilkan halaman keuangan saya	Masuk kedalam sistem sebagai member dan menampilkan halaman keuangan saya



3	Membuka halaman <i>sign in</i>	Memasukan email terdaftar pada kolom masukan email, memasukan password yang salah pada kolom masukan password, dan menekan tombol submit	Menampilkan pesan " <i>Wrong email or password</i> "	Menampilkan pesan " <i>Wrong email or password</i> "
---	--------------------------------	--	--	--

6.1.1.2 Pengujian Algoritma Data Uji

Pada tabel 6.3 menjelaskan bagian kode yang menjadi node pada pengujian algoritma data uji pada klas `TambahPengeluaran` dengan *white box testing*.

Tabel 6. 3 Algoritma Data Uji pada klas `TambahPengeluaran`

```

private void DataUji(int jumlahPengeluaranInt) {
    //set jumlah pengeluaran tidak dianggarkan
    final double jumlahPengeluaran = (double) jumlahPengeluaranInt;
    final FirebaseUser currentUser = mAuth.getCurrentUser();
    users.addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot) {
            //liburan (tersier)
            String anggaranLiburan =
dataSnapshot.child(currentUser.getId()).child("Anggaran").child("liburan").
getValue(String.class);
            String totalLiburan =
dataSnapshot.child(currentUser.getId()).child("Total
Pengeluaran").child("liburan").getValue(String.class);
            double anggaranLiburanD = Double.parseDouble(anggaranLiburan);
            double totalLiburanD = Double.parseDouble(totalLiburan);
            double sisaAnggaranLiburan = anggaranLiburanD - totalLiburanD -
jumlahPengeluaran;
            if (sisaAnggaranLiburan >= 0) {
                //masukan ke perencanaan alokasi dana liburan
                String alokasiLiburan = String.valueOf(jumlahPengeluaran);
                final RencanaAlokasiDana alokasiDana = new
RencanaAlokasiDana(alokasiLiburan, "0", "0", "0", "0", "0", "0",
et_note.getText().toString());
                users.child(currentUser.getId()).child("Rencana Alokasi
Dana").setValue(alokasiDana);
                final DataUjiNaiveBayes dataUjiNaiveBayes = new
DataUjiNaiveBayes("Ya", "Tidak", "Tidak");
                users.child(currentUser.getId()).child("Data
Uji").setValue(dataUjiNaiveBayes);
            } else {
                //masukan ke perencanaan alokasi dana liburan
                double alokasiLiburanD = jumlahPengeluaran +
sisaAnggaranLiburan; // ditambah karena nilai sisa anggaran liburan (
                String alokasiLiburan = String.valueOf(alokasiLiburanD);
                alokasiLiburan
    }
}
    }
}
    
```

```

        //hobi dan hiburan (tersier)
        String anggaranHobiHiburan =
dataSnapshot.child(currentUser.getId()).child("Anggaran").child("hobiDanHib
uran").getValue(String.class);
        String totalHobiHiburan =
dataSnapshot.child(currentUser.getId()).child("Total
Pengeluaran").child("hobiHiburan").getValue(String.class);
        double anggaranHobiHiburanD =
Double.parseDouble(anggaranHobiHiburan);
        double totalHobiHiburanD =
Double.parseDouble(totalHobiHiburan);
        double sisaAnggaranHobiHiburan = anggaranHobiHiburanD -
totalHobiHiburanD + sisaAnggaranLiburan;
        if (sisaAnggaranHobiHiburan >= 0) {
            double sisaAnggaranLiburanBaru = -sisaAnggaranLiburan;
            String alokasiHobiHiburan =
String.valueOf(sisaAnggaranLiburanBaru);
            final RencanaAlokasiDana alokasiDana = new
RencanaAlokasiDana(alokasiLiburan, alokasiHobiHiburan, "0", "0", "0", "0", "0",
"0", "0", et_note.getText().toString());
            users.child(currentUser.getId()).child("Rencana Alokasi
Dana").setValue(alokasiDana);
            final DataUjiNaiveBayes dataUjiNaiveBayes = new
DataUjiNaiveBayes("Ya", "Tidak", "Tidak");
            users.child(currentUser.getId()).child("Data
Uji").setValue(dataUjiNaiveBayes);
        } else {
            //masuk ke perencanaan alokasi dana Hobi dan Hiburan
            double alokasiHobiHiburanD = -sisaAnggaranLiburan +
sisaAnggaranHobiHiburan; //ditambah karena nilai sisa anggaran hobi dan
hiburan (-)
            String alokasiHobiHiburan =
String.valueOf(alokasiHobiHiburanD);

            //zakat/amal (sekunder)
            String anggaranZakatAmal =
dataSnapshot.child(currentUser.getId()).child("Anggaran").child("zakatAtau
amal").getValue(String.class);
            String totalZakatAmal =
dataSnapshot.child(currentUser.getId()).child("Total
Pengeluaran").child("zakatAmal").getValue(String.class);
            double anggaranZakatAmalD =
Double.parseDouble(anggaranZakatAmal);
            double totalZakatAmalD =
Double.parseDouble(totalZakatAmal);
            double sisaAnggaranZakatAmal = anggaranZakatAmalD -
totalZakatAmalD + sisaAnggaranHobiHiburan;
            if (sisaAnggaranZakatAmal >= 0) {
                double sisaAnggaranHobiHiburanBaru = -
sisaAnggaranHobiHiburan;
                String alokasiZakatAmal =
String.valueOf(sisaAnggaranHobiHiburanBaru);
                final RencanaAlokasiDana alokasiDana = new
RencanaAlokasiDana(alokasiLiburan, alokasiHobiHiburan, alokasiZakatAmal,
"0", "0", "0", "0", "0", "0", et_note.getText().toString());
                users.child(currentUser.getId()).child("Rencana
Alokasi Dana").setValue(alokasiDana);
                final DataUjiNaiveBayes dataUjiNaiveBayes = new
DataUjiNaiveBayes("Ya", "Ya", "Tidak");
                users.child(currentUser.getId()).child("Data
Uji").setValue(dataUjiNaiveBayes);
            } else {
                //masuk ke perencanaan alokasi dana zakat amal
                double alokasiZakatAmalD = -sisaAnggaranHobiHibu
+ sisaAnggaranZakatAmal; //ditambah karena nilai sisa anggaran zakat/amal (-)

```

```

)
        String alokasiZakatAmal =
String.valueOf(alokasiZakatAmalID);

        //gadget (sekunder)
        String anggaranGadget =
dataSnapshot.child(currentUser.getId()).child("Anggaran").child("gadget").g
etValue(String.class);
        String totalGadget =
dataSnapshot.child(currentUser.getId()).child("Total
Pengeluaran").child("gadget").getValue(String.class);
        double anggaranGadgetD =
Double.parseDouble(anggaranGadget);
        double totalGadgetD =
Double.parseDouble(totalGadget);
        double sisaAnggaranGadget = anggaranGadgetD -
totalGadgetD + sisaAnggaranZakatAmal;
        if (sisaAnggaranGadget >= 0) {
            double sisaAnggaranZakatAmalBaru = -
sisaAnggaranZakatAmal;
            String alokasiGadget =
String.valueOf(sisaAnggaranZakatAmalBaru);
            final RencanaAlokasiDana alokasiDana = new
RencanaAlokasiDana(alokasiLiburan, alokasiHobiHiburan, alokasiZakatAmal,
alokasiGadget, "0", "0", "0", "0", et_note.getText().toString());
            users.child(currentUser.getId()).child("Rencana
Alokasi Dana").setValue(alokasiDana);
            final DataUjiNaiveBayes dataUjiNaiveBayes = new
DataUjiNaiveBayes("Ya", "Ya", "Tidak");
            users.child(currentUser.getId()).child("Data
Uji").setValue(dataUjiNaiveBayes);
        } else {
            //masukan ke perencanaan alokasi dana gadget
            double alokasiGadgetD = -sisaAnggaranZakatAmal +
sisaAnggaranGadget; //ditambah karena nilai sisa anggaran gadget (-)
            String alokasiGadget =
String.valueOf(alokasiGadgetD);

            ////pendidikan (sekunder)
            String anggaranPendidikan =
dataSnapshot.child(currentUser.getId()).child("Anggaran").child("pendid
").getValue(String.class);
            String totalPendidikan =
dataSnapshot.child(currentUser.getId()).child("Total
Pengeluaran").child("pendidikan").getValue(String.class);
            double anggaranPendidikanD =
Double.parseDouble(anggaranPendidikan);
            double totalPendidikanD =
Double.parseDouble(totalPendidikan);
            double sisaAnggaranPendidikan =
anggaranPendidikanD - totalPendidikanD + sisaAnggaranGadget;
            if (sisaAnggaranPendidikan >= 0) {
                double sisaAnggaranGadgetBaru = -
sisaAnggaranGadget;
                String alokasiPendidikan =
String.valueOf(sisaAnggaranGadgetBaru);
                final RencanaAlokasiDana alokasiDana = new
RencanaAlokasiDana(alokasiLiburan, alokasiHobiHiburan, alokasiZakatAmal,
alokasiGadget, alokasiPendidikan, "0", "0", "0",
et_note.getText().toString());
                users.child(currentUser.getId()).child("Rencana Alokasi
Dana").setValue(alokasiDana);
                final DataUjiNaiveBayes dataUjiNaiveBayes =
new DataUjiNaiveBayes("Ya", "Ya", "Tidak");

```

9

10

11

12

```

users.child(currentUser.getUId()).child("Data
Uji").setValue(dataUjiNaiveBayes);
        } else {
            //masuk ke perencanaan alokasi dana
            pendidikan
                double alokasiPendidikanD = -
            sisaAnggaranGadget + sisaAnggaranPendidikan; //ditambah karena nilai sisa
            anggaran pendidikan (-)
                String alokasiPendidikan =
            String.valueOf(alokasiPendidikanD);

                //transportasi (primer)
                String anggaranTransportasi =
            dataSnapshot.child(currentUser.getUId()).child("Anggaran").child("transportasi").getValue(String.class);
                String totalTransportasi =
            dataSnapshot.child(currentUser.getUId()).child("Total
Pengeluaran").child("transportasi").getValue(String.class);
                double anggaranTransportasiD =
            Double.parseDouble(anggaranTransportasi);
                double totalTransportasiD =
            Double.parseDouble(totalTransportasi);
                double sisaAnggaranTransportasi =
            anggaranTransportasiD - totalTransportasiD + sisaAnggaranPendidikan;
                if (sisaAnggaranTransportasi >= 0) {
                    double sisaAnggaranPendidikanBaru = -
            sisaAnggaranPendidikan;
                    String alokasiTransportasi =
            String.valueOf(sisaAnggaranPendidikanBaru);
                    final RencanaAlokasiDana alokasiDana =
            new RencanaAlokasiDana(alokasiLiburan, alokasiHobiHiburan, alokasiZakatAmal,
            alokasiGadget, alokasiPendidikan, alokasiTransportasi, "0", "0",
            et_note.getText().toString());

            users.child(currentUser.getUId()).child("Rencana Alokasi
            Dana").setValue(alokasiDana);
                final DataUjiNaiveBayes
            dataUjiNaiveBayes = new DataUjiNaiveBayes("Ya", "Ya", "Ya");

            users.child(currentUser.getUId()).child("Data
            Uji").setValue(dataUjiNaiveBayes);
        } else {
            //masuk ke perencanaan alokasi dana
            transportasi
                double alokasiTransportasiD = -
            sisaAnggaranPendidikan + sisaAnggaranTransportasi; //ditambah karena nilai
            sisa anggaran transportasi (-)
                String alokasiTransportasi =
            String.valueOf(alokasiTransportasiD);

                //kesehatan (primer)
                String anggaranKesehatan =
            dataSnapshot.child(currentUser.getUId()).child("Anggaran").child("kesehatan")
            ).getValue(String.class);
                String totalKesehatan =
            dataSnapshot.child(currentUser.getUId()).child("Total
Pengeluaran").child("kesehatan").getValue(String.class);
                double anggaranKesehatanD =
            Double.parseDouble(anggaranKesehatan);
                double totalKesehatanD =
            Double.parseDouble(totalKesehatan);
                double sisaAnggaranKesehatan =
            anggaranKesehatanD - totalKesehatanD + sisaAnggaranTransportasi;
                if (sisaAnggaranKesehatan >= 0) {

```

13

14

15

16



```

        = -sisaAnggaranTransportasi;
        String.valueOf(sisaAnggaranTransportasiBaru);
        final RencanaAlokasiDana alokasiDana
    = new RencanaAlokasiDana(alokasiLiburan, alokasiHobiHiburan,
    alokasiZakatAmal, alokasiGadget, alokasiPendidikan, alokasiTransportasi,
    alokasiKesehatan, "0", et_note.getText().toString());

    users.child(currentUser.getId()).child("Rencana Alokasi
    Dana").setValue(alokasiDana);

    dataUjiNaiveBayes = new DataUjiNaiveBayes("Ya", "Ya", "Ya");

    users.child(currentUser.getId()).child("Data
    Uji").setValue(dataUjiNaiveBayes);

    } else {
        //masukan ke perencanaan alokasi
        dana kesehatan
        double alokasiKesehatanD = -
        sisaAnggaranTransportasi + sisaAnggaranKesehatan; //ditambah karena nilai
        sisa anggaran Kesehatan (-)
        String.valueOf(alokasiKesehatanD);

        //harian (primer)
        String anggaranHarian =
    dataSnapshot.child(currentUser.getId()).child("Anggaran").child("harian").g
    etValue(String.class);
        String totalHarian =
    dataSnapshot.child(currentUser.getId()).child("Total
    Pengeluaran").child("harian").getValue(String.class);
        double anggaranHarianD =
    Double.parseDouble(anggaranHarian);
        double totalHarianD =
    Double.parseDouble(totalHarian);
        double sisaAnggaranHarian =
    anggaranHarianD - totalHarianD + sisaAnggaranKesehatan;
        if (sisaAnggaranHarian >= 0) {
            double sisaAnggaranKesehatanBaru
        = -sisaAnggaranKesehatan;
            String.valueOf(sisaAnggaranKesehatanBaru);

            final RencanaAlokasiDana
        alokasiDana = new RencanaAlokasiDana(alokasiLiburan, alokasiHobiHiburan,
        alokasiZakatAmal, alokasiGadget, alokasiPendidikan, alokasiTransportasi,
        alokasiKesehatan, alokasiHarian, et_note.getText().toString());

        users.child(currentUser.getId()).child("Rencana Alokasi
        Dana").setValue(alokasiDana);

        dataUjiNaiveBayes = new DataUjiNaiveBayes("Ya", "Ya", "Ya");

        users.child(currentUser.getId()).child("Data
        Uji").setValue(dataUjiNaiveBayes);

        } else {
            //masukan ke perencanaan alokasi
            dana Harian
            double alokasiHarianD = -
            sisaAnggaranKesehatan + sisaAnggaranHarian; //ditambah karena nilai sis
            anggaran Kesehatan (-)
            String.valueOf(alokasiHarianD);

            final RencanaAlokasiDana
        alokasiDana = new RencanaAlokasiDana(alokasiLiburan, alokasiHobiHiburan,
    
```

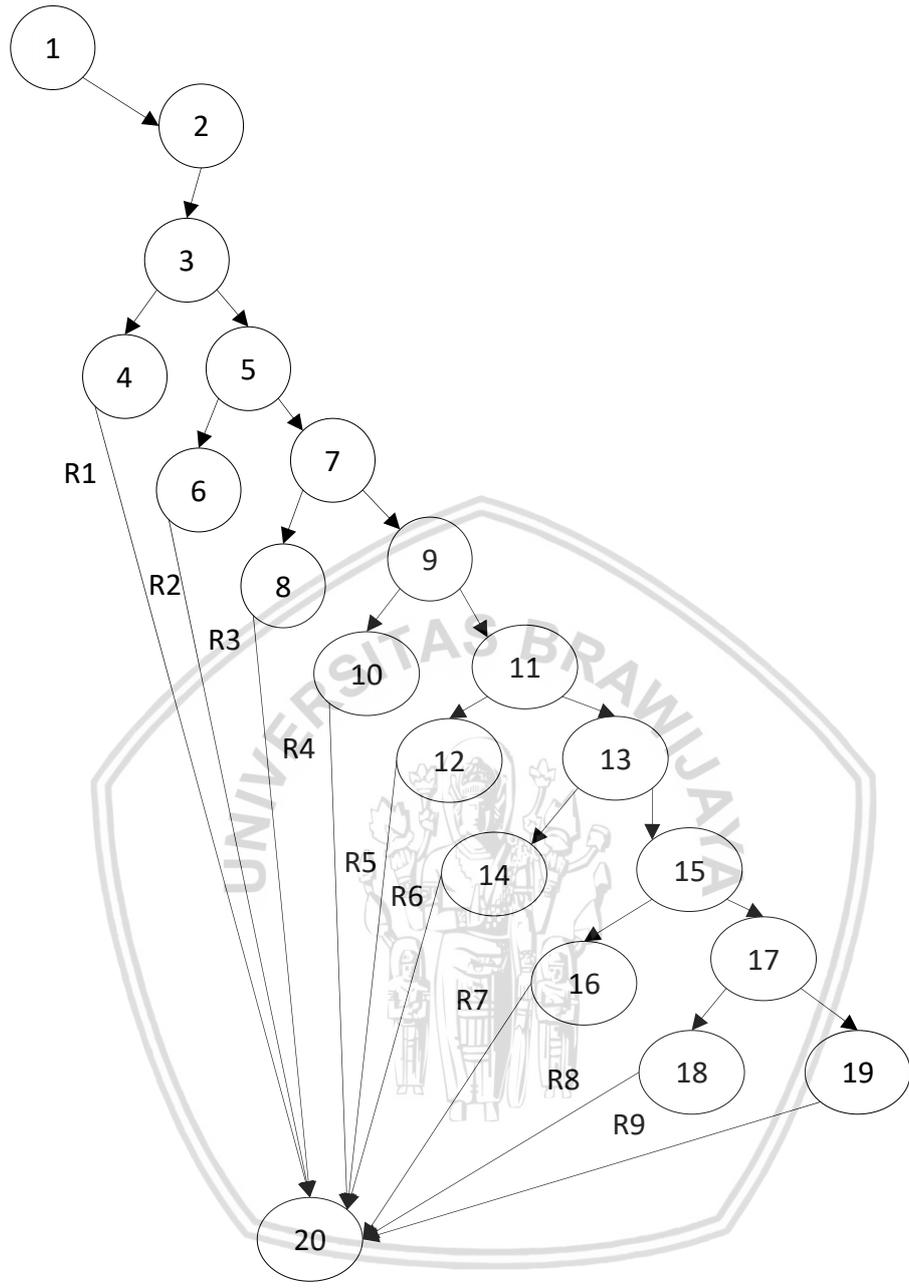
16

17

18

19





Gambar 6. 2 Flow Graph Algoritma Data Uji

Pada gambar 6.2 menjelaskan *flow graph* algoritma data uji.

(iii) Independent Path

- Jalur 1 = 1 – 2 – 3 – 4 – 20
- Jalur 2 = 1 – 2 – 3 – 5 – 6 – 20
- Jalur 3 = 1 – 2 – 3 – 5 – 7 – 8 – 20
- Jalur 4 = 1 – 2 – 3 – 5 – 7 – 9 – 10 – 20
- Jalur 5 = 1 – 2 – 3 – 5 – 7 – 9 – 11 – 12 – 20

Jalur 6 = 1 – 2 – 3 – 5 – 7 – 9 – 11 – 13 – 14 – 20

Jalur 7 = 1 – 2 – 3 – 5 – 7 – 9 – 11 – 13 – 15 – 16 – 20

Jalur 8 = 1 – 2 – 3 – 5 – 7 – 9 – 11 – 13 – 15 – 17 – 18 – 20

Jalur 9 = 1 – 2 – 3 – 5 – 7 – 9 – 11 – 13 – 15 – 17 – 19 – 20

(iv) Kasus Uji

Tabel 6. 4 Kasus Uji Algoritma Data Uji

Jalur	Kasus Uji	Prosedur Uji	Expected Result	Test Result
1	Menambah pengeluaran dengan memilih kategori tidak dianggarkan	Mengeluarkan pengeluaran dengan kategori tidak dianggarkan sehingga mendapatkan rencana alokasi merekomendasikan kategori liburan saja	Nilai data uji pada <i>database</i> , yaitu tersier = Ya, sekunder = Tidak, primer = Tidak	Nilai data uji pada <i>database</i> , yaitu tersier = Ya, sekunder = Tidak, primer = Tidak
2	Menambah pengeluaran dengan memilih kategori tidak dianggarkan	Mengeluarkan pengeluaran dengan kategori tidak dianggarkan sehingga mendapatkan rencana alokasi merekomendasikan kategori liburan serta hobi dan hiburan saja	Nilai data uji pada <i>database</i> , yaitu tersier = Ya, sekunder = Tidak, primer = Tidak	Nilai data uji pada <i>database</i> , yaitu tersier = Ya, sekunder = Tidak, primer = Tidak
3	Menambah pengeluaran dengan memilih kategori tidak dianggarkan	Mengeluarkan pengeluaran dengan kategori tidak dianggarkan sehingga mendapatkan rencana alokasi merekomendasikan kategori liburan, hobi dan hiburan, dan zakat/amal saja	Nilai data uji pada <i>database</i> , yaitu tersier = Ya, sekunder = Ya, primer = Tidak	Nilai data uji pada <i>database</i> , yaitu tersier = Ya, sekunder = Ya, primer = Tidak



4	Menambah pengeluaran dengan memilih kategori tidak dianggarkan	Mengeluarkan pengeluaran dengan kategori tidak dianggarkan sehingga mendapatkan rencana alokasi merekomendasikan kategori liburan, hobi dan hiburan, zakat/amal, dan gadget saja	Nilai data uji pada <i>database</i> , yaitu tersier = Ya, sekunder = Ya, primer = Tidak	Nilai data uji pada <i>database</i> , yaitu tersier = Ya, sekunder = Ya, primer = Tidak
5	Menambah pengeluaran dengan memilih kategori tidak dianggarkan	Mengeluarkan pengeluaran dengan kategori tidak dianggarkan sehingga mendapatkan rencana alokasi merekomendasikan kategori liburan, hobi dan hiburan, zakat/amal, gadget, dan pendidikan saja	Nilai data uji pada <i>database</i> , yaitu tersier = Ya, sekunder = Ya, primer = Tidak	Nilai data uji pada <i>database</i> , yaitu tersier = Ya, sekunder = Ya, primer = Tidak
6	Menambah pengeluaran dengan memilih kategori tidak dianggarkan	Mengeluarkan pengeluaran dengan kategori tidak dianggarkan sehingga mendapatkan rencana alokasi merekomendasikan kategori liburan, hobi dan hiburan, zakat/amal, gadget, pendidikan, dan transportasi saja	Nilai data uji pada <i>database</i> , yaitu tersier = Ya, sekunder = Ya, primer = Ya	Nilai data uji pada <i>database</i> , yaitu tersier = Ya, sekunder = Ya, primer = Ya
7	Menambah pengeluaran dengan memilih kategori	Mengeluarkan pengeluaran dengan kategori tidak dianggarkan sehingga mendapatkan	Nilai data uji pada <i>database</i> , yaitu tersier = Ya, sekunder = Ya, primer = Ya	Nilai data uji pada <i>database</i> , yaitu tersier = Ya, sekunder = Ya, primer = Ya



	tidak dianggarkan	rencana alokasi merekomendasikan kategori liburan, hobi dan hiburan, zakat/amal, gadget, pendidikan, transportasi, dan kesehatan saja		
8	Menambah pengeluaran dengan memilih kategori tidak dianggarkan	Mengeluarkan pengeluaran dengan kategori tidak dianggarkan sehingga mendapatkan rencana alokasi merekomendasikan seluruh kategori	Nilai data uji pada <i>database</i> , yaitu tersier = Ya, sekunder = Ya, primer = Ya	Nilai data uji pada <i>database</i> , yaitu tersier = Ya, sekunder = Ya, primer = Ya
9	Menambah pengeluaran dengan memilih kategori tidak dianggarkan	Mengeluarkan pengeluaran dengan kategori tidak dianggarkan sehingga mendapatkan rencana alokasi merekomendasikan seluruh kategori dan menghabiskan maksimal anggaran	Nilai data uji pada <i>database</i> , yaitu tersier = Ya, sekunder = Ya, primer = Ya	Nilai data uji pada <i>database</i> , yaitu tersier = Ya, sekunder = Ya, primer = Ya

6.1.2 Sprint 2

Pengujian unit pada *sprint 1* dilakukan pada algoritma *onClick*.

6.1.2.1 Pengujian Algoritma onClick

Pada tabel 6.3 menjelaskan bagian kode yang menjadi node pada pengujian algoritma *onClick* pada klas *BudgetPlanning* dengan *white box testing*.

Tabel 6. 5 Algoritma onClick pada klas BudgetPlanning

```

public void onClick(View view) {
    switch (view.getId()) {
        case R.id.ganti_halaman:
            //pindah halaman ke tambah pemasukan
            getFragmentManager().beginTransaction().replace(R.id.main_frame, new KeteranganAnggaran()).addToBackStack(null).commit();
            break;
        case R.id.submit_button_budget:
            if (!validateForm()) {
    
```

4

```

        return;
    }

    final FirebaseAuth currentUser = FirebaseAuth.getInstance().getCurrentUser();
    users.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot) {
            FirebaseUser user;
            String userID;
            String gadget = et_gadget.getText().toString();
            String harian = et_harian.getText().toString();
            String hobiHiburan =
            et_hobi_dan_hiburan.getText().toString();
            String kesehatan =
            et_kesehatan.getText().toString();
            String liburan = et_liburan.getText().toString();
            String pendidikan =
            et_pendidikan.getText().toString();
            String transportasi =
            et_transportasi.getText().toString();
            String zakatAmal =
            et_zakat_atau_amal.getText().toString();

            double gadgetDouble = Double.parseDouble(gadget);
            double harianDouble = Double.parseDouble(harian);
            double hobiHiburanDouble =
            Double.parseDouble(hobiHiburan);
            double kesehatanDouble =
            Double.parseDouble(kesehatan);
            double liburanDouble = Double.parseDouble(liburan);
            double pendidikanDouble =
            Double.parseDouble(pendidikan);
            double transportasiDouble =
            Double.parseDouble(transportasi);
            double zakatAmalDouble =
            Double.parseDouble(zakatAmal);

            double total = gadgetDouble + harianDouble +
            hobiHiburanDouble + kesehatanDouble + liburanDouble
            + pendidikanDouble + transportasiDouble +
            zakatAmalDouble;
            System.out.println("total anggaran = " +total);

            //manggil total uang user
            String totalUang =
            dataSnapshot.child(currentUser.getUid()).child("Total
            Uang").child("totalUang").getValue(String.class);
            double totalUangDouble =
            Double.parseDouble(totalUang);
            double cek = totalUangDouble - total;
            System.out.println("total cek = "+cek);

            if(cek >= 0){
                //untuk menyimpan data anggaran ke model class
                Anggaran
                final Anggaran anggaranUser = new
                Anggaran(gadget, harian, hobiHiburan, kesehatan, liburan, pendidikan,
                transportasi, zakatAmal);

                //untuk menulis data yang sudah di register ke
                firebase
                user = FirebaseAuth.getInstance().getCurrentUser();
                userID = user.getUid();
            }
        }
    });
}

```

5

6

```

//untuk menulis data yang sudah di register ke
firebase
users.child(userID).child("Anggaran").setValue(anggaranUser);

//untuk info berhasil di upload
Toast.makeText(getActivity(), "Perencanaan
anggaran berhasil ditambah!", Toast.LENGTH_SHORT).show();

et_gadget.setText("");
et_harian.setText("");
et_hobi_dan_hiburan.setText("");
et_kesehatan.setText("");
et_liburan.setText("");
et_pendidikan.setText("");
et_transportasi.setText("");
et_zakat_atau_amal.setText("");
} else {
//untuk info tidak berhasil di upload
Toast.makeText(getActivity(), "Total anggaran
melebihi total uang Anda! Tambah pemasukan Anda!",
Toast.LENGTH_LONG).show();
et_gadget.setText("");
et_harian.setText("");
et_hobi_dan_hiburan.setText("");
et_kesehatan.setText("");
et_liburan.setText("");
et_pendidikan.setText("");
et_transportasi.setText("");
et_zakat_atau_amal.setText("");
}
}
@Override
public void onCancelled(DatabaseError databaseError) {
}
});
break;
}
}

```

6

7

8

(a) Basis Path Testing

(i) Flow Graph

Pada gambar 6.3 menjelaskan *flow graph* algoritma onClick.

(ii) Cyclomatic Complexity

$$V(G) = E - N + 2 \tag{6.10}$$

$$V(G) = 10 - 8 + 2 \tag{6.11}$$

$$V(G) = 4 \tag{6.12}$$

(iii) Independent Path

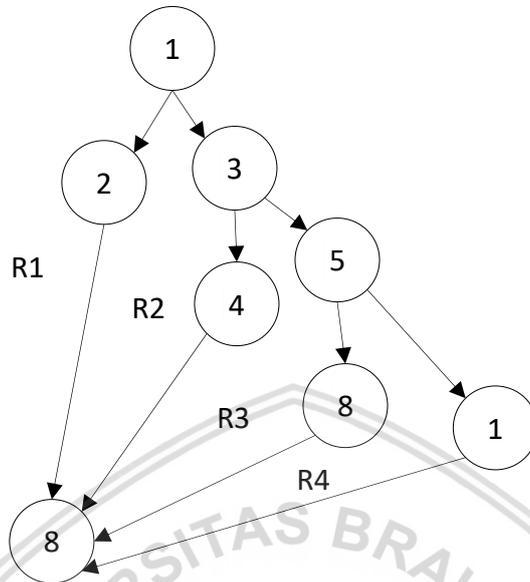
Jalur 1 = 1 - 2 - 8

Jalur 2 = 1 - 2 - 3 - 4 - 8



Jalur 3 = 1 – 2 – 5 – 6 – 8

Jalur 4 = 1 – 2 – 5 – 7 – 8



Gambar 6. 3 Flow Graph Algoritma onClick

(iv) Kasus Uji Algoritma OnClick

Jalur	Kasus Uji	Prosedur Uji	Expected Result	Test Result
1	Membuka halaman perencanaan anggaran	Menekan tombol here pada halaman perencanaan anggaran	Menampilkan halaman keterangan kategori anggaran	Menampilkan halaman keterangan kategori anggaran
2	Membuka halaman perencanaan anggaran	Mengosongkan form perencanaan anggaran dan menekan tombol submit	Warning error ditampilkan pada setiap kolom masukan kategori anggaran dan bertuliskan "Required"	Warning error ditampilkan pada setiap kolom masukan kategori anggaran dan bertuliskan "Required"
3	Membuka halaman perencanaan anggaran	memasukan perencanaan anggaran pada setiap kategori dan tidak melebihi total uang yang sudah	Menampilkan pesan "Perencanaan Anggaran berhasil ditambah!"	Menampilkan pesan "Perencanaan Anggaran berhasil ditambah!"



		dimasukan pada FM App		
4	Membuka halaman perencanaan anggaran	Memasukan perencanaan anggaran pada setiap kategori dengan total melebihi total uang yang sudah dimasukan pada FM App	Menampilkan pesan "Total anggaran melebihi total uang Anda! Tambah pemasukan Anda!"	Menampilkan pesan "Total anggaran melebihi total uang Anda! Tambah pemasukan Anda!"

6.2 Pengujian Algoritma Naïve Bayes

Pengujian algoritma Naïve Bayes dilakukan untuk menguji hasil rekomendasi apakah sesuai dengan harapan pengguna atau tidak ketika pengeluaran tidak dianggarkan dikeluarkan. Sebelum dilakukan perhitungan, maka data *training* harus dibuat terlebih dahulu. Untuk mendapatkan data *training*, terdapat beberapa tahapan yang dilakukan, yaitu penentuan status keuangan setelah pengeluaran tidak dianggarkan atau tidak terduga terjadi, penentuan urutan pengambilan dana ketika pengeluaran tidak dianggarkan atau tidak terduga dilakukan, pengelompokan anggaran berdasarkan kategori pengeluaran, dan terakhir menyimpulkan data *training*. Tahap pertama yaitu, penentuan status keuangan setelah pengeluaran tidak dianggarkan atau tidak terduga terjadi. Berdasarkan lampiran B bagian B1, terdapat 8 pengambilan anggaran keuangan untuk menutupi pengeluaran tidak terduga atau tidak dianggarkan yang telah dilakukan oleh 40 responden dan menentukan status keuangannya ketika pengeluaran tersebut telah dilakukan. Untuk lebih jelasnya dapat dilihat pada tabel 6.6.

Tabel 6. 6 Penentuan Status Keuangan Setelah Pengeluaran Tidak Terduga Terjadi

No	Anggaran Yang Diambil Untuk Menutupi Pengeluaran Tidak Terduga	Status
1	Liburan	Aman
2	Hobi & Hiburan	Aman
3	Liburan dan Hobi & Hiburan	Aman
4	Harian	Bahaya
5	Liburan, Hobi & Hiburan, Gadget, dan Zakat	Menengah
6	Harian dan Pendidikan	Bahaya
7	Liburan, Hobi & Hiburan, dan Transportasi	Menengah
8	Harian, Pendidikan, dan Kesehatan	Bahaya

Setelah penentuan status keuangan ketika pengeluaran tidak terduga dilakukan, maka tahap selanjutnya adalah menentukan urutan pengambilan anggaran untuk menutupi pengeluaran tidak terduga. Pengambilan anggaran ini dilakukan secara urut dan jika anggaran belum habis, maka anggaran berikutnya tidak boleh diambil. Berdasarkan lampiran B bagian B2 yang telah dilakukan oleh 47 responden, urutan pengambilan dana anggaran untuk menutupi pengeluaran tidak dianggarkan adalah sebagai berikut:

1. Pengambilan dana dari anggaran Liburan
2. Pengambilan dana dari anggaran Hobi & Hiburan
3. Pengambilan dana dari anggaran Zakat/Amal
4. Pengambilan dana dari anggaran Gadget
5. Pengambilan dana dari anggaran Pendidikan
6. Pengambilan dana dari anggaran Transportasi
7. Pengambilan dana dari anggaran Kesehatan
8. Pengambilan dana dari anggaran Harian

Setelah penentuan status keuangan ketika pengeluaran tidak terduga terjadi dan urutan pengambilan anggaran sudah ditentukan, maka tahap selanjutnya adalah pengelompokan anggaran kedalam kategori pengeluaran. Berdasarkan lampiran B bagian B3 yang telah dilakukan oleh 70 responden, pengelompokan anggaran dibagi menjadi 3, yaitu tersier, sekunder, dan primer. Untuk lebih jelasnya dapat dilihat pada tabel 6.7.

Tabel 6. 7 Pengelompokan Anggaran Pada Kategori

No	Jenis Anggaran	Tersier		Sekunder		Primer	
		Ya	Tidak	Ya	Tidak	Ya	Tidak
1	Liburan	√	X	X	X	X	X
2	Hobi & Hiburan	√	X	X	X	X	X
3	Zakat/Amal	X	X	√	X	X	X
4	Gadget	X	X	√	X	X	X
5	Pendidikan	X	X	√	X	X	X
6	Transportasi	X	X	X	X	√	X
7	Kesehatan	X	X	X	X	√	X
8	Harian	X	X	X	X	√	X

Berdasarkan tabel 6.7, tanda centang (√) merupakan tanda yang digunakan apabila jenis anggaran termasuk pada kategori tersebut dan tanda silang (X) merupakan tanda yang digunakan apabila jenis anggaran tidak termasuk pada kategori tersebut. Disamping itu, dapat diketahui bahwa nilai dari klas Tersier akan bernilai Ya apabila

pengambilan dana diambil pada anggaran liburan dan atau hobi & hiburan. Untuk nilai dari klas Sekunder akan bernilai Ya apabila pengambilan dana diambil pada anggaran zakat/amal dan atau *gadget* dan atau pendidikan. Untuk nilai dari klas Primer akan bernilai Ya apabila pengambilan dana diambil pada anggaran transportasi dan atau kesehatan dan atau harian. Selain dari kemungkinan tersebut, maka akan bernilai Tidak pada setiap klasnya.

Setelah penentuan status keuangan setelah pengeluaran tidak terduga terjadi, penentuan urutan pengambilan dana ketika pengeluaran tidak dianggarkan atau tidak terduga dilakukan, dan pengelompokan anggaran berdasarkan kategori pengeluaran telah dilakukan, maka tahap selanjutnya adalah mengkonversi status keuangan setelah pengeluaran tidak terduga terjadi yang ada pada tabel 6.7 menjadi data *training*. Untuk lebih jelasnya dapat dilihat pada tabel 6.8.

Tabel 6. 8 Konversi Anggaran Yang Diambil Menjadi Data *Training*

No	Jenis Anggaran Yang Diambil	Urutan Pengambilan	Tersier	Sekunder	Primer	Status
1	Liburan	1. Liburan	Ya	Tidak	Tidak	Aman
2	Hobi & Hiburan	1. Liburan 2. Hobi & Hiburan	Ya	Tidak	Tidak	Aman
3	Liburan dan Hobi & Hiburan	1. Liburan 2. Hobi&Hiburan	Ya	Tidak	Tidak	Aman
4	Harian	1. Liburan 2. Hobi & Hiburan 3. Zakat/Amal 4. Gadget 5. Transportasi 6. Pendidikan 7. Kesehatan 8. Harian	Ya	Ya	Ya	Bahaya
5	Liburan, Hobi & Hiburan, Gadget, dan Zakat	1. Liburan 2. Hobi & Hiburan 3. Zakat/Amal 4. Gadget	Ya	Ya	Tidak	Menengah

6	Harian dan Pendidikan	<ol style="list-style-type: none"> 1. Liburan 2. Hobi & Hiburan 3. Zakat/Amal 4. Gadget 5. Transportasi 6. Pendidikan 7. Kesehatan 8. Harian 	Ya	Ya	Ya	Bahaya
7	Liburan, Hobi & Hiburan, dan Transportasi	<ol style="list-style-type: none"> 1. Liburan 2. Hobi & Hiburan 3. Zakat/Amal 4. Gadget 5. Transportasi 	Ya	Ya	Tidak	Menengah
8	Harian, Pendidikan, dan Kesehatan	<ol style="list-style-type: none"> 1. Liburan 2. Hobi & Hiburan 3. Zakat/Amal 4. Gadget 5. Transportasi 6. Pendidikan 7. Kesehatan 8. Harian 	Ya	Ya	Ya	Bahaya

Berdasarkan tabel 6.8, pada kasus 1, ketika jenis anggaran yang diambil adalah Liburan, maka urutan pengambilannya adalah Liburan, sehingga memiliki nilai kategori tersier = “Ya”, sekunder = “Tidak”, primer = “Tidak”, dan status = “Aman”. Pada kasus 2, ketika jenis anggaran yang diambil adalah Hobi & Hiburan, maka urutan pengambilannya adalah pertama Liburan dan kedua Hobi & Hiburan, sehingga memiliki nilai kategori tersier = “Ya”, sekunder = “Tidak”, primer = “Tidak”, dan status = “Aman”. Pada kasus 3, ketika jenis anggaran yang diambil adalah Liburan dan Hobi & Hiburan, maka urutan pengambilannya adalah pertama Liburan dan kedua Hobi & Hiburan, sehingga memiliki nilai kategori tersier = “Ya”, sekunder = “Tidak”, primer = “Tidak”, dan status = “Aman”. Pada kasus 4, ketika jenis anggaran yang diambil adalah Harian, maka urutan pengambilannya adalah pertama Liburan, kedua Hobi & Hiburan, ketiga Zakat/Amal, keempat Gadget, kelima Transportasi, keenam Pendidikan, ketujuh Kesehatan, dan



kedelapan Harian, sehingga memiliki nilai kategori tersier = “Ya”, sekunder = “Ya”, primer = “Ya”, dan status = “Bahaya”. Pada kasus 5, ketika jenis anggaran yang diambil adalah Liburan, Hobi & Hiburan, Gadget, dan Zakat, maka urutan pengambilannya adalah pertama Liburan, kedua Hobi & Hiburan, ketiga Zakat/Amal, dan keempat Gadget, sehingga memiliki nilai kategori tersier = “Ya”, sekunder = “Ya”, primer = “Tidak”, dan status = “Menengah”. Pada kasus 6, ketika jenis anggaran yang diambil adalah Harian dan Pendidikan, maka urutan pengambilannya adalah pertama Liburan, kedua Hobi & Hiburan, ketiga Zakat/Amal, keempat Gadget, kelima Transportasi, keenam Pendidikan, ketujuh Kesehatan, dan kedelapan Harian, sehingga memiliki nilai kategori tersier = “Ya”, sekunder = “Ya”, primer = “Ya” dan status = “Bahaya”. Pada kasus 7, ketika jenis anggaran yang diambil adalah Liburan, Hobi & Hiburan, dan Transportasi, maka urutan pengambilannya adalah pertama Liburan, kedua Hobi & Hiburan, ketiga Zakat/Amal, keempat Gadget, dan kelima Transportasi, sehingga memiliki nilai kategori tersier = “Ya”, sekunder = “Ya”, primer = “Tidak” dan status = “Menengah”. Pada kasus 8, ketika jenis anggaran yang diambil adalah Harian, Pendidikan, dan Kesehatan, maka urutan pengambilannya adalah pertama Liburan, kedua Hobi & Hiburan, ketiga Zakat/Amal, keempat Gadget, kelima Transportasi, keenam Pendidikan, ketujuh Kesehatan, dan kedelapan Harian, sehingga memiliki nilai kategori tersier = “Ya”, sekunder = “Ya”, primer = “Ya” dan status = “Bahaya”.

Setelah konversi anggaran yang diambil menjadi data *training* yang ada di tabel 6.8 dilakukan dan berdasarkan lampiran B bagian B4 serta B5, maka dapat disimpulkan menjadi data *training*. Untuk lebih jelasnya mengenai data *training* dapat dilihat pada tabel 6.9.

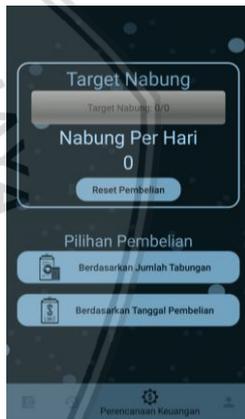
Tabel 6. 9 Data Training

No	Tersier	Sekunder	Primer	Status
1	Ya	Tidak	Tidak	Aman
2	Ya	Tidak	Tidak	Aman
3	Ya	Tidak	Tidak	Aman
4	Ya	Ya	Ya	Bahaya
5	Ya	Ya	Tidak	Menengah
6	Ya	Ya	Ya	Bahaya
7	Ya	Ya	Tidak	Menengah
8	Ya	Ya	Ya	Bahaya

6.2.2 Kasus Uji Pengujian Algoritma Naïve Bayes

Pada tabel 6.10 merupakan tabel kondisi aplikasi yang ditetapkan sebelum pengujian algoritma Naïve Bayes dilakukan.

Tabel 6. 10 Kondisi Aplikasi Sebelum Pengujian

No	Nama	Kondisi	Screenshot
1	Total Uang	1,600,000	
2	Perencanaan Anggaran	Gadget: 0/50,000 Harian: 0/900,000 Hobi & Hiburan: 0/100,000 Kesehatan: 0/100,000 Liburan: 0/100,000 Pendidikan: 0/100,000 Transportasi: 0/150,000 Zakat/Amal: 0/100,000	
3	Batas Harian	30,000	
4	Pembelian Berdasarkan Jumlah Nabung	Tidak Diaktifkan	
5	Pembelian Berdasarkan Tanggal Pembelian	Tidak Diaktifkan	

Berdasarkan tabel 6.10, rencana alokasi dana anggaran dari pengeluaran tidak dianggarkan dipengaruhi oleh total uang yang dimiliki saat itu, kondisi anggaran pada tiap kategori, batas pengeluaran harian, dan perencanaan pembelian suatu barang baik berdasarkan jumlah nabung atau tanggal pembelian. Selanjutnya, setelah kondisi sebelum kasus uji ditentukan, maka akan dilakukan 8 kasus uji untuk mengetahui status keuangan selanjutnya ketika perencanaan alokasi anggaran pengeluaran tidak dianggarkan diterapkan. Untuk lebih jelasnya mengenai kasus uji pengujian ini dapat dilihat pada tabel 6.11.

Tabel 6. 11 Kasus Uji Pengujian Algoritma Naïve Bayes

Nama	Jumlah Pengeluaran Tidak Dianggarkan	Tujuan Pengujian
Kasus Uji 1	50,000	Untuk mengetahui status keuangan selanjutnya ketika perencanaan alokasi anggaran
Kasus Uji 2	200,000	
Kasus Uji 3	250,000	



Kasus Uji 4	325,000	pengeluaran tidak dianggarkan diterapkan
Kasus Uji 5	380,000	
Kasus Uji 6	525,000	
Kasus Uji 7	675,000	
Kasus Uji 8	1,200,000	

6.2.2.1 Kasus Uji 1

Tabel 6.12 merupakan tabel yang menjelaskan kasus uji pertama yang akan dilakukan. Tujuan dilakukan pengujian ini adalah untuk mengetahui status keuangan selanjutnya ketika perencanaan alokasi anggaran pengeluaran tidak dianggarkan diterapkan. Pada kasus uji pertama akan dilakukan pemasukan pengeluaran dengan kategori tidak dianggarkan sebesar 50,000 pada aplikasi. Berdasarkan hasil yang diharapkan pengguna, dana yang diambil hanya anggaran Liburan dan hasil yang diharapkan ketika pengeluaran tidak dianggarkan diterapkan adalah Aman. Berdasarkan dana yang diambil, maka data *testing* akan bernilai tersier = "ya", sekunder = "tidak", primer = "tidak".

Tabel 6. 12 Kasus Uji 1 Algoritma Naïve Bayes

Nama	Kasus Uji 1
Tujuan Pengujian	Untuk mengetahui status keuangan selanjutnya ketika perencanaan alokasi anggaran pengeluaran tidak dianggarkan diterapkan
Jumlah Pengeluaran Tidak Dianggarkan	50,000
Dana Anggaran Yang Diambil	Liburan: 50,000/100,000 Hobi & Hiburan: 0/100,000 Zakat/Amal: 0/100,000 Gadget: 0/50,000 Pendidikan: 0/100,000 Transportasi: 0/150,000 Kesehatan: 0/100,000 Harian: 0/900,000
Data Testing/Sampel	Tersier = Ya
	Sekunder = Tidak
	Primer = Tidak
Hasil Yang Diharapkan	Status = Aman



Berdasarkan tabel 6.12, tahap berikutnya akan menjelaskan mengenai perhitungan algoritma Naïve Bayes yang dihitung secara manual untuk menentukan hasil yang diharapkan. Dalam perhitungan Naïve Bayes, tahap pertama yang dilakukan adalah menghitung nilai dari probabilitas *prior* ($P(C)$). Setelah nilai *prior* didapat, tahap kedua adalah mencari nilai dari probabilitas *likelihood* ($P(X|C)$). Setelah nilai probabilitas dari *prior* dan *likelihood* didapat, maka tahap selanjutnya mencari nilai dari probabilitas *posterior* ($P(C|X)$) untuk mengetahui hasil yang diharapkan. Untuk lebih jelasnya dapat dilihat pada sub bab berikut.

(a) Probabilitas *Prior* ($P(C)$)

Pada perhitungan mencari nilai probabilitas *prior*, perhitungan akan dilakukan dengan pencarian probabilitas berdasarkan data *training* yang dapat dilihat pada tabel 6.9. Pencarian probabilitas *prior* dilakukan dengan membandingkan antara jumlah dari data *training* dari kategori C dengan jumlah total data *training*.

$$P(\text{Status} = \text{"Aman"}) = 3/8$$

$$P(\text{Status} = \text{"Menengah"}) = 2/8$$

$$P(\text{Status} = \text{"Bahaya"}) = 3/8$$

(b) Probabilitas *Likelihood* ($P(X|C)$)

Pada perhitungan mencari nilai probabilitas *Likelihood*, perhitungan akan dilakukan dengan pencarian probabilitas berdasarkan data *training* yang dapat dilihat pada tabel 6.9. pencarian probabilitas *likelihood* dilakukan dengan membandingkan nilai – nilai atribut dalam sampel X berdasarkan kondisi pada hipotesis C.

$$P(\text{Tersier} = \text{"Ya"} | \text{Status} = \text{"Aman"}) = 3/3$$

$$P(\text{Tersier} = \text{"Ya"} | \text{Status} = \text{"Menengah"}) = 2/2$$

$$P(\text{Tersier} = \text{"Ya"} | \text{Status} = \text{"Bahaya"}) = 3/3$$

$$P(\text{Sekunder} = \text{"Tidak"} | \text{Status} = \text{"Aman"}) = 3/3$$

$$P(\text{Sekunder} = \text{"Tidak"} | \text{Status} = \text{"Menengah"}) = 0/2$$

$$P(\text{Sekunder} = \text{"Tidak"} | \text{Status} = \text{"Bahaya"}) = 0/3$$

$$P(\text{Primer} = \text{"Tidak"} | \text{Status} = \text{"Aman"}) = 3/3$$

$$P(\text{Primer} = \text{"Tidak"} | \text{Status} = \text{"Menengah"}) = 2/2$$

$$P(\text{Primer} = \text{"Tidak"} | \text{Status} = \text{"Bahaya"}) = 0/3$$

$$P(X | \text{Status} = \text{"Aman"})$$

$$= P(\text{Tersier} = \text{"Ya"}, \text{Sekunder} = \text{"Tidak"}, \text{Primer} = \text{"Tidak"} | \text{Status} = \text{"Aman"})$$

$$= 3/3 \times 3/3 \times 3/3 = 1$$

$$\begin{aligned}
 &P(X|Status = \text{"Menengah"}) \\
 &= P(Tersier = \text{"Ya"}, Sekunder = \text{"Tidak"}, Primer = \text{"Tidak"} | Status = \text{"Menengah"}) \\
 &= \frac{2}{2} \times \frac{0}{2} \times \frac{2}{2} = 0 \\
 &P(X|Status = \text{"Bahaya"}) \\
 &= P(Tersier = \text{"Ya"}, Sekunder = \text{"Tidak"}, Primer = \text{"Tidak"} | Status = \text{"Bahaya"}) \\
 &= \frac{3}{3} \times \frac{0}{3} \times \frac{0}{3} = 0
 \end{aligned}$$

(c) Probabilitas Posterior (P(C|X))

Setelah nilai dari probabilitas *prior* dan probabilitas *likelihood* didapat, maka probabilitas *posterior* dapat dihitung. Dalam perhitungannya, nilai *posterior* didapat dengan mengalikan nilai probabilitas *likelihood* dan probabilitas *prior*. Setelah dikalikan setiap nilai probabilitas *likelihood* dan probabilitas *prior*, maka selanjutnya dicari nilai yang paling besar dan nilai yang paling besar merupakan hasil akhir perhitungan tersebut.

$$P(X|Status = \text{"Aman"}) \times P(Status = \text{"Aman"}) = 1 \times \frac{3}{8} = 0,375$$

$$P(X|Status = \text{"Menengah"}) \times P(Status = \text{"Menengah"}) = 0 \times \frac{2}{8} = 0$$

$$P(X|Status = \text{"Bahaya"}) \times P(Status = \text{"Bahaya"}) = 0 \times \frac{3}{8} = 0$$

Berdasarkan hasil tersebut, nilai dengan status aman adalah 0,375, status menengah adalah 0, dan status bahaya adalah 0. Hal ini menyatakan, nilai dari status aman paling besar, sehingga status yang diharapkan adalah **aman**.

6.2.2.2 Kasus Uji 2

Tabel 6.13 merupakan tabel yang menjelaskan kasus uji kedua yang akan dilakukan. Tujuan dilakukan pengujian ini adalah untuk mengetahui status keuangan selanjutnya ketika perencanaan alokasi anggaran pengeluaran tidak dianggarkan diterapkan. Pada kasus uji kedua akan dilakukan pemasukan pengeluaran dengan kategori tidak dianggarkan sebesar 200,000 pada aplikasi. Berdasarkan hasil yang diharapkan pengguna, dana yang diambil hanya anggaran Liburan dan Hobi & Hiburan. Selain itu, hasil yang diharapkan ketika pengeluaran tidak dianggarkan diterapkan adalah Aman. Berdasarkan dana yang diambil, maka data *testing* akan bernilai tersier = "ya", sekunder = "tidak", primer = "tidak".

Tabel 6. 13 Kasus Uji 2 Algoritma Naïve Bayes

Nama	Kasus Uji 2
Tujuan Pengujian	Untuk mengetahui status keuangan selanjutnya ketika perencanaan alokasi anggaran pengeluaran tidak dianggarkan diterapkan
Jumlah Pengeluaran Tidak Dianggarkan	200,000



Dana Anggaran Yang Diambil	Liburan: 100,000/100,000 Hobi & Hiburan: 100,000/100,000 Zakat/Amal: 0/100,000 Gadget: 0/50,000 Pendidikan: 0/100,000 Transportasi: 0/150,000 Kesehatan: 0/100,000 Harian: 0/900,000
Data Testing/Sampel	Tersier = Ya
	Sekunder = Tidak
	Primer = Tidak
Hasil Yang Diharapkan	Status = Aman

Berdasarkan tabel 6.13, tahap berikutnya akan menjelaskan mengenai perhitungan algoritma Naïve Bayes yang dihitung secara manual untuk menentukan hasil yang diharapkan. Dalam perhitungan Naïve Bayes, tahap pertama yang dilakukan adalah menghitung nilai dari probabilitas *prior* ($P(C)$). Setelah nilai *prior* didapat, tahap kedua adalah mencari nilai dari probabilitas *likelihood* ($P(X|C)$). Setelah nilai probabilitas dari *prior* dan *likelihood* didapat, maka tahap selanjutnya mencari nilai dari probabilitas *posterior* ($P(C|X)$) untuk mengetahui hasil yang diharapkan. Untuk lebih jelasnya dapat dilihat pada sub bab berikut.

(a) Probabilitas Prior ($P(C)$)

Pada perhitungan mencari nilai probabilitas *prior*, perhitungan akan dilakukan dengan pencarian probabilitas berdasarkan data *training* yang dapat dilihat pada tabel 6.9. Pencarian probabilitas *prior* dilakukan dengan membandingkan antara jumlah dari data *training* dari kategori C dengan jumlah total data *training*.

$$P(\text{Status} = \text{"Aman"}) = 3/8$$

$$P(\text{Status} = \text{"Menengah"}) = 2/8$$

$$P(\text{Status} = \text{"Bahaya"}) = 3/8$$

(b) Probabilitas Likelihood ($P(X|C)$)

Pada perhitungan mencari nilai probabilitas *Likelihood*, perhitungan akan dilakukan dengan pencarian probabilitas berdasarkan data *training* yang dapat dilihat pada tabel 6.9. Pencarian probabilitas *likelihood* dilakukan dengan membandingkan nilai – nilai atribut dalam sampel X berdasarkan kondisi pada hipotesis C.

$$P(\text{Tersier} = \text{"Ya"} | \text{Status} = \text{"Aman"}) = 3/3$$



$$P(\text{Tersier} = \text{"Ya"} | \text{Status} = \text{"Menengah"}) = 2/2$$

$$P(\text{Tersier} = \text{"Ya"} | \text{Status} = \text{"Bahaya"}) = 3/3$$

$$P(\text{Sekunder} = \text{"Tidak"} | \text{Status} = \text{"Aman"}) = 3/3$$

$$P(\text{Sekunder} = \text{"Tidak"} | \text{Status} = \text{"Menengah"}) = 0/2$$

$$P(\text{Sekunder} = \text{"Tidak"} | \text{Status} = \text{"Bahaya"}) = 0/3$$

$$P(\text{Primer} = \text{"Tidak"} | \text{Status} = \text{"Aman"}) = 3/3$$

$$P(\text{Primer} = \text{"Tidak"} | \text{Status} = \text{"Menengah"}) = 2/2$$

$$P(\text{Primer} = \text{"Tidak"} | \text{Status} = \text{"Bahaya"}) = 0/3$$

$$P(X | \text{Status} = \text{"Aman"})$$

$$= P(\text{Tersier} = \text{"Ya"}, \text{Sekunder} = \text{"Tidak"}, \text{Primer} = \text{"Tidak"} | \text{Status} = \text{"Aman"})$$

$$= 3/3 \times 3/3 \times 3/3 = 1$$

$$P(X | \text{Status} = \text{"Menengah"})$$

$$= P(\text{Tersier} = \text{"Ya"}, \text{Sekunder} = \text{"Tidak"}, \text{Primer} = \text{"Tidak"} | \text{Status} = \text{"Menengah"})$$

$$= 2/2 \times 0/2 \times 2/2 = 0$$

$$P(X | \text{Status} = \text{"Bahaya"})$$

$$= P(\text{Tersier} = \text{"Ya"}, \text{Sekunder} = \text{"Tidak"}, \text{Primer} = \text{"Tidak"} | \text{Status} = \text{"Bahaya"})$$

$$= 3/3 \times 0/3 \times 0/3 = 0$$

(c) Probabilitas Posterior ($P(C|X)$)

Setelah nilai dari probabilitas *prior* dan probabilitas *likelihood* didapat, maka probabilitas *posterior* dapat dihitung. Dalam perhitungannya, nilai *posterior* didapat dengan mengalikan nilai probabilitas *likelihood* dan probabilitas *prior*. Setelah dikalikan setiap nilai probabilitas *likelihood* dan probabilitas *prior*, maka selanjutnya dicari nilai yang paling besar dan nilai yang paling besar merupakan hasil akhir perhitungan tersebut.

$$P(X | \text{Status} = \text{"Aman"}) \times P(\text{Status} = \text{"Aman"}) = 1 \times 3/8 = 0,375$$

$$P(X | \text{Status} = \text{"Menengah"}) \times P(\text{Status} = \text{"Menengah"}) = 0 \times 2/8 = 0$$

$$P(X | \text{Status} = \text{"Bahaya"}) \times P(\text{Status} = \text{"Bahaya"}) = 0 \times 3/8 = 0$$

Berdasarkan hasil tersebut, nilai dengan status aman adalah 0,375, status menengah adalah 0, dan status bahaya adalah 0. Hal ini menyatakan, nilai dari status aman paling besar, sehingga status yang diharapkan adalah **aman**.

6.2.2.3 Kasus Uji 3

Tabel 6.14 merupakan tabel yang menjelaskan kasus uji kedua yang akan dilakukan. Tujuan dilakukan pengujian ini adalah untuk mengetahui status keuangan selanjutnya



ketika perencanaan alokasi anggaran pengeluaran tidak dianggarkan diterapkan. Pada kasus uji kedua akan dilakukan pemasukan pengeluaran dengan kategori tidak dianggarkan sebesar 250,000 pada aplikasi. Berdasarkan hasil yang diharapkan pengguna, dana yang diambil hanya anggaran Liburan, Hobi & Hiburan, dan Zakat/Amal. Selain itu, hasil yang diharapkan ketika pengeluaran tidak dianggarkan diterapkan adalah Menengah. Berdasarkan dana yang diambil, maka data *testing* akan bernilai tersier = “ya”, sekunder = “ya”, primer = “tidak”.

Tabel 6. 14 Kasus Uji 3 Algoritma Naïve Bayes

Nama	Kasus Uji 3
Tujuan Pengujian	Untuk mengetahui status keuangan selanjutnya ketika perencanaan alokasi anggaran pengeluaran tidak dianggarkan diterapkan
Jumlah Pengeluaran Tidak Dianggarkan	250,000
Dana Anggaran Yang Diambil	Liburan: 100,000/100,000 Hobi & Hiburan: 100,000/100,000 Zakat/Amal: 50,000/100,000 Gadget: 0/50,000 Pendidikan: 0/100,000 Transportasi: 0/150,000 Kesehatan: 0/100,000 Harian: 0/900,000
Data Testing/Sampel	Tersier = Ya
	Sekunder = Ya
	Primer = Tidak
Hasil Yang Diharapkan	Status = Menengah

Berdasarkan tabel 6.14, tahap berikutnya akan menjelaskan mengenai perhitungan algoritma Naïve Bayes yang dihitung secara manual untuk menentukan hasil yang diharapkan. Dalam perhitungan Naïve Bayes, tahap pertama yang dilakukan adalah menghitung nilai dari probabilitas *prior* ($P(C)$). Setelah nilai *prior* didapat, tahap kedua adalah mencari nilai dari probabilitas *likelihood* ($P(X|C)$). Setelah nilai probabilitas dari *prior* dan *likelihood* didapat, maka tahap selanjutnya mencari nilai dari probabilitas *posterior* ($P(C|X)$) untuk mengetahui hasil yang diharapkan. Untuk lebih jelasnya dapat dilihat pada sub bab berikut.



(a) Probabilitas Prior ($P(C)$)

Pada perhitungan mencari nilai probabilitas *prior*, perhitungan akan dilakukan dengan pencarian probabilitas berdasarkan data *training* yang dapat dilihat pada tabel 6.9. Pencarian probabilitas *prior* dilakukan dengan membandingkan antara jumlah dari data *training* dari kategori C dengan jumlah total data *training*.

$$P(\text{Status} = \text{"Aman"}) = 3/8$$

$$P(\text{Status} = \text{"Menengah"}) = 2/8$$

$$P(\text{Status} = \text{"Bahaya"}) = 3/8$$

(b) Probabilitas Likelihood ($P(X|C)$)

Pada perhitungan mencari nilai probabilitas *Likelihood*, perhitungan akan dilakukan dengan pencarian probabilitas berdasarkan data *training* yang dapat dilihat pada tabel 6.9. Pencarian probabilitas *likelihood* dilakukan dengan membandingkan nilai – nilai atribut dalam sampel X berdasarkan kondisi pada hipotesis C.

$$P(\text{Tersier} = \text{"Ya"} | \text{Status} = \text{"Aman"}) = 3/3$$

$$P(\text{Tersier} = \text{"Ya"} | \text{Status} = \text{"Menengah"}) = 2/2$$

$$P(\text{Tersier} = \text{"Ya"} | \text{Status} = \text{"Bahaya"}) = 3/3$$

$$P(\text{Sekunder} = \text{"Ya"} | \text{Status} = \text{"Aman"}) = 0/3$$

$$P(\text{Sekunder} = \text{"Ya"} | \text{Status} = \text{"Menengah"}) = 2/2$$

$$P(\text{Sekunder} = \text{"Ya"} | \text{Status} = \text{"Bahaya"}) = 3/3$$

$$P(\text{Primer} = \text{"Tidak"} | \text{Status} = \text{"Aman"}) = 3/3$$

$$P(\text{Primer} = \text{"Tidak"} | \text{Status} = \text{"Menengah"}) = 2/2$$

$$P(\text{Primer} = \text{"Tidak"} | \text{Status} = \text{"Bahaya"}) = 0/3$$

$$P(X | \text{Status} = \text{"Aman"})$$

$$= P(\text{Tersier} = \text{"Ya"}, \text{Sekunder} = \text{"Ya"}, \text{Primer} = \text{"Tidak"} | \text{Status} = \text{"Aman"})$$

$$= 3/3 \times 0/3 \times 3/3 = 0$$

$$P(X | \text{Status} = \text{"Menengah"})$$

$$= P(\text{Tersier} = \text{"Ya"}, \text{Sekunder} = \text{"Ya"}, \text{Primer} = \text{"Tidak"} | \text{Status} = \text{"Menengah"})$$

$$= 2/2 \times 2/2 \times 2/2 = 1$$

$$P(X | \text{Status} = \text{"Bahaya"})$$

$$= P(\text{Tersier} = \text{"Ya"}, \text{Sekunder} = \text{"Ya"}, \text{Primer} = \text{"Tidak"} | \text{Status} = \text{"Bahaya"})$$

$$= 3/3 \times 3/3 \times 0/3 = 0$$

(c) Probabilitas Posterior ($P(C|X)$)

Setelah nilai dari probabilitas *prior* dan probabilitas *likelihood* didapat, maka probabilitas *posterior* dapat dihitung. Dalam perhitungannya, nilai *posterior* didapat dengan mengalikan nilai probabilitas *likelihood* dan probabilitas *prior*. Setelah dikalikan setiap nilai probabilitas *likelihood* dan probabilitas *prior*, maka selanjutnya dicari nilai yang paling besar dan nilai yang paling besar merupakan hasil akhir perhitungan tersebut.

$$P(X|Status = "Aman") \times P(Status = "Aman") = 0 \times \frac{3}{8} = 0$$

$$P(X|Status = "Menengah") \times P(Status = "Menengah") = 1 \times \frac{2}{8} = 0,25$$

$$P(X|Status = "Bahaya") \times P(Status = "Bahaya") = 0 \times \frac{3}{8} = 0$$

Berdasarkan hasil tersebut, nilai dengan status aman adalah 0, status menengah adalah 0,25, dan status bahaya adalah 0. Hal ini menyatakan, nilai dari status menengah paling besar, sehingga status yang diharapkan adalah **menengah**.

6.2.2.4 Kasus Uji 4

Tabel 6.15 merupakan tabel yang menjelaskan kasus uji kedua yang akan dilakukan. Tujuan dilakukan pengujian ini adalah untuk mengetahui status keuangan selanjutnya ketika perencanaan alokasi anggaran pengeluaran tidak dianggarkan diterapkan. Pada kasus uji kedua akan dilakukan pemasukan pengeluaran dengan kategori tidak dianggarkan sebesar 325,000 pada aplikasi. Berdasarkan hasil yang diharapkan pengguna, dana yang diambil hanya anggaran Liburan, Hobi & Hiburan, Zakat/Amal, dan Gadget. Selain itu, hasil yang diharapkan ketika pengeluaran tidak dianggarkan diterapkan adalah Menengah. Berdasarkan dana yang diambil, maka data *testing* akan bernilai tersier = "ya", sekunder = "ya", primer = "tidak".

Tabel 6. 15 Kasus Uji 4 Algoritma Naïve Bayes

Nama	Kasus Uji 4
Tujuan Pengujian	Untuk mengetahui status keuangan selanjutnya ketika perencanaan alokasi anggaran pengeluaran tidak dianggarkan diterapkan
Jumlah Pengeluaran Tidak Dianggarkan	325,000
Dana Anggaran Yang Diambil	Liburan: 100,000/100,000 Hobi & Hiburan: 100,000/100,000 Zakat/Amal: 100,000/100,000 Gadget: 25,000/50,000 Pendidikan: 0/100,000 Transportasi: 0/150,000



	Kesehatan: 0/100,000 Harian: 0/900,000
Data Testing/Sampel	Tersier = Ya
	Sekunder = Ya
	Primer = Tidak
Hasil Yang Diharapkan	Status = Menengah

Berdasarkan tabel 6.15, tahap berikutnya akan menjelaskan mengenai perhitungan algoritma Naïve Bayes yang dihitung secara manual untuk menentukan hasil yang diharapkan. Dalam perhitungan Naïve Bayes, tahap pertama yang dilakukan adalah menghitung nilai dari probabilitas *prior* ($P(C)$). Setelah nilai *prior* didapat, tahap kedua adalah mencari nilai dari probabilitas *likelihood* ($P(X|C)$). Setelah nilai probabilitas dari *prior* dan *likelihood* didapat, maka tahap selanjutnya mencari nilai dari probabilitas *posterior* ($P(C|X)$) untuk mengetahui hasil yang diharapkan. Untuk lebih jelasnya dapat dilihat pada sub bab berikut.

(a) Probabilitas Prior ($P(C)$)

Pada perhitungan mencari nilai probabilitas *prior*, perhitungan akan dilakukan dengan pencarian probabilitas berdasarkan data *training* yang dapat dilihat pada tabel 6.9. Pencarian probabilitas *prior* dilakukan dengan membandingkan antara jumlah dari data *training* dari kategori C dengan jumlah total data *training*.

$$P(\text{Status} = \text{"Aman"}) = 3/8$$

$$P(\text{Status} = \text{"Menengah"}) = 2/8$$

$$P(\text{Status} = \text{"Bahaya"}) = 3/8$$

(b) Probabilitas Likelihood ($P(X|C)$)

Pada perhitungan mencari nilai probabilitas *Likelihood*, perhitungan akan dilakukan dengan pencarian probabilitas berdasarkan data *training* yang dapat dilihat pada tabel 6.9. Pencarian probabilitas *likelihood* dilakukan dengan membandingkan nilai – nilai atribut dalam sampel X berdasarkan kondisi pada hipotesis C.

$$P(\text{Tersier} = \text{"Ya"} | \text{Status} = \text{"Aman"}) = 3/3$$

$$P(\text{Tersier} = \text{"Ya"} | \text{Status} = \text{"Menengah"}) = 2/2$$

$$P(\text{Tersier} = \text{"Ya"} | \text{Status} = \text{"Bahaya"}) = 3/3$$

$$P(\text{Sekunder} = \text{"Ya"} | \text{Status} = \text{"Aman"}) = 0/3$$

$$P(\text{Sekunder} = \text{"Ya"} | \text{Status} = \text{"Menengah"}) = 2/2$$

$$P(\text{Sekunder} = \text{"Ya"} | \text{Status} = \text{"Bahaya"}) = 3/3$$



$$P(\text{Primer} = \text{"Tidak"} | \text{Status} = \text{"Aman"}) = 3/3$$

$$P(\text{Primer} = \text{"Tidak"} | \text{Status} = \text{"Menengah"}) = 2/2$$

$$P(\text{Primer} = \text{"Tidak"} | \text{Status} = \text{"Bahaya"}) = 0/3$$

$$P(X | \text{Status} = \text{"Aman"})$$

$$= P(\text{Tersier} = \text{"Ya"}, \text{Sekunder} = \text{"Ya"}, \text{Primer} = \text{"Tidak"} | \text{Status} = \text{"Aman"})$$

$$= 3/3 \times 0/3 \times 3/3 = 0$$

$$P(X | \text{Status} = \text{"Menengah"})$$

$$= P(\text{Tersier} = \text{"Ya"}, \text{Sekunder} = \text{"Ya"}, \text{Primer} = \text{"Tidak"} | \text{Status} = \text{"Menengah"})$$

$$= 2/2 \times 2/2 \times 2/2 = 1$$

$$P(X | \text{Status} = \text{"Bahaya"})$$

$$= P(\text{Tersier} = \text{"Ya"}, \text{Sekunder} = \text{"Ya"}, \text{Primer} = \text{"Tidak"} | \text{Status} = \text{"Bahaya"})$$

$$= 3/3 \times 3/3 \times 0/3 = 0$$

(c) Probabilitas Posterior ($P(C|X)$)

Setelah nilai dari probabilitas *prior* dan probabilitas *likelihood* didapat, maka probabilitas *posterior* dapat dihitung. Dalam perhitungannya, nilai *posterior* didapat dengan mengalikan nilai probabilitas *likelihood* dan probabilitas *prior*. Setelah dikalikan setiap nilai probabilitas *likelihood* dan probabilitas *prior*, maka selanjutnya dicari nilai yang paling besar dan nilai yang paling besar merupakan hasil akhir perhitungan tersebut.

$$P(X | \text{Status} = \text{"Aman"}) \times P(\text{Status} = \text{"Aman"}) = 0 \times 3/8 = 0$$

$$P(X | \text{Status} = \text{"Menengah"}) \times P(\text{Status} = \text{"Menengah"}) = 1 \times 2/8 = 0,25$$

$$P(X | \text{Status} = \text{"Bahaya"}) \times P(\text{Status} = \text{"Bahaya"}) = 0 \times 3/8 = 0$$

Berdasarkan hasil tersebut, nilai dengan status aman adalah 0, status menengah adalah 0,25, dan status bahaya adalah 0. Hal ini menyatakan, nilai dari status menengah paling besar, sehingga status yang diharapkan adalah **menengah**.

6.2.2.5 Kasus Uji 5

Tabel 6.16 merupakan tabel yang menjelaskan kasus uji kedua yang akan dilakukan. Tujuan dilakukan pengujian ini adalah untuk mengetahui status keuangan selanjutnya ketika perencanaan alokasi anggaran pengeluaran tidak dianggarkan diterapkan. Pada kasus uji kedua akan dilakukan pemasukan pengeluaran dengan kategori tidak dianggarkan sebesar 380,000 pada aplikasi. Berdasarkan hasil yang diharapkan pengguna, dana yang diambil hanya anggaran Liburan, Hobi & Hiburan, Zakat/Amal, Gadget, dan Pendidikan. Selain itu, hasil yang diharapkan ketika pengeluaran tidak dianggarkan diterapkan adalah Menengah. Berdasarkan dana yang diambil, maka data *testing* akan bernilai tersier = "ya", sekunder = "ya", primer = "tidak".

Tabel 6. 16 Kasus Uji 5 Algoritma Naïve Bayes

Nama	Kasus Uji 5
Tujuan Pengujian	Untuk mengetahui status keuangan selanjutnya ketika perencanaan alokasi anggaran pengeluaran tidak dianggarkan diterapkan
Jumlah Pengeluaran Tidak Dianggarkan	380,000
Dana Anggaran Yang Diambil	Liburan: 100,000/100,000 Hobi & Hiburan: 100,000/100,000 Zakat/Amal: 100,000/100,000 Gadget: 50,000/50,000 Pendidikan: 30,000/100,000 Transportasi: 0/150,000 Kesehatan: 0/100,000 Harian: 0/900,000
Data Testing/Sampel	Tersier = Ya
	Sekunder = Ya
	Primer = Tidak
Hasil Yang Diharapkan	Status = Menengah

Berdasarkan tabel 6.16, tahap berikutnya akan menjelaskan mengenai perhitungan algoritma Naïve Bayes yang dihitung secara manual untuk menentukan hasil yang diharapkan. Dalam perhitungan Naïve Bayes, tahap pertama yang dilakukan adalah menghitung nilai dari probabilitas *prior* ($P(C)$). Setelah nilai *prior* didapat, tahap kedua adalah mencari nilai dari probabilitas *likelihood* ($P(X|C)$). Setelah nilai probabilitas dari *prior* dan *likelihood* didapat, maka tahap selanjutnya mencari nilai dari probabilitas *posterior* ($P(C|X)$) untuk mengetahui hasil yang diharapkan. Untuk lebih jelasnya dapat dilihat pada sub bab berikut.

(a) Probabilitas *Prior* ($P(C)$)

Pada perhitungan mencari nilai probabilitas *prior*, perhitungan akan dilakukan dengan pencarian probabilitas berdasarkan data *training* yang dapat dilihat pada tabel 6.9. Pencarian probabilitas *prior* dilakukan dengan membandingkan antara jumlah dari data *training* dari kategori C dengan jumlah total data *training*.

$$P(\text{Status} = \text{"Aman"}) = \frac{3}{8}$$

$$P(\text{Status} = \text{"Menengah"}) = \frac{2}{8}$$

$$P(\text{Status} = \text{"Bahaya"}) = 3/8$$

(b) Probabilitas Likelihood ($P(X|C)$)

Pada perhitungan mencari nilai probabilitas *Likelihood*, perhitungan akan dilakukan dengan pencarian probabilitas berdasarkan data *training* yang dapat dilihat pada tabel 6.9. Pencarian probabilitas *likelihood* dilakukan dengan membandingkan nilai – nilai atribut dalam sampel X berdasarkan kondisi pada hipotesis C.

$$P(\text{Tersier} = \text{"Ya"} | \text{Status} = \text{"Aman"}) = 3/3$$

$$P(\text{Tersier} = \text{"Ya"} | \text{Status} = \text{"Menengah"}) = 2/2$$

$$P(\text{Tersier} = \text{"Ya"} | \text{Status} = \text{"Bahaya"}) = 3/3$$

$$P(\text{Sekunder} = \text{"Ya"} | \text{Status} = \text{"Aman"}) = 0/3$$

$$P(\text{Sekunder} = \text{"Ya"} | \text{Status} = \text{"Menengah"}) = 2/2$$

$$P(\text{Sekunder} = \text{"Ya"} | \text{Status} = \text{"Bahaya"}) = 3/3$$

$$P(\text{Primer} = \text{"Tidak"} | \text{Status} = \text{"Aman"}) = 3/3$$

$$P(\text{Primer} = \text{"Tidak"} | \text{Status} = \text{"Menengah"}) = 2/2$$

$$P(\text{Primer} = \text{"Tidak"} | \text{Status} = \text{"Bahaya"}) = 0/3$$

$$P(X | \text{Status} = \text{"Aman"})$$

$$= P(\text{Tersier} = \text{"Ya"}, \text{Sekunder} = \text{"Ya"}, \text{Primer} = \text{"Tidak"} | \text{Status} = \text{"Aman"})$$

$$= 3/3 \times 0/3 \times 3/3 = 0$$

$$P(X | \text{Status} = \text{"Menengah"})$$

$$= P(\text{Tersier} = \text{"Ya"}, \text{Sekunder} = \text{"Ya"}, \text{Primer} = \text{"Tidak"} | \text{Status} = \text{"Menengah"})$$

$$= 2/2 \times 2/2 \times 2/2 = 1$$

$$P(X | \text{Status} = \text{"Bahaya"})$$

$$= P(\text{Tersier} = \text{"Ya"}, \text{Sekunder} = \text{"Ya"}, \text{Primer} = \text{"Tidak"} | \text{Status} = \text{"Bahaya"})$$

$$= 3/3 \times 3/3 \times 0/3 = 0$$

(c) Probabilitas Posterior ($P(C|X)$)

Setelah nilai dari probabilitas *prior* dan probabilitas *likelihood* didapat, maka probabilitas *posterior* dapat dihitung. Dalam perhitungannya, nilai *posterior* didapat dengan mengalikan nilai probabilitas *likelihood* dan probabilitas *prior*. Setelah dikalikan setiap nilai probabilitas *likelihood* dan probabilitas *prior*, maka selanjutnya dicari nilai yang paling besar dan nilai yang paling besar merupakan hasil akhir perhitungan tersebut.

$$P(X | \text{Status} = \text{"Aman"}) \times P(\text{Status} = \text{"Aman"}) = 0 \times 3/8 = 0$$

$$P(X|Status = \text{"Menengah"}) \times P(Status = \text{"Menengah"}) = 1 \times \frac{2}{8} = 0,25$$

$$P(X|Status = \text{"Bahaya"}) \times P(Status = \text{"Bahaya"}) = 0 \times \frac{3}{8} = 0$$

Berdasarkan hasil tersebut, nilai dengan status aman adalah 0, status menengah adalah 0,25, dan status bahaya adalah 0. Hal ini menyatakan, nilai dari status menengah paling besar, sehingga status yang diharapkan adalah **menengah**.

6.2.2.6 Kasus Uji 6

Tabel 6.17 merupakan tabel yang menjelaskan kasus uji kedua yang akan dilakukan. Tujuan dilakukan pengujian ini adalah untuk mengetahui status keuangan selanjutnya ketika perencanaan alokasi anggaran pengeluaran tidak dianggarkan diterapkan. Pada kasus uji kedua akan dilakukan pemasukan pengeluaran dengan kategori tidak dianggarkan sebesar 525,000 pada aplikasi. Berdasarkan hasil yang diharapkan pengguna, dana yang diambil hanya anggaran Liburan, Hobi & Hiburan, Zakat/Amal, Gadget, Pendidikan, dan Transportasi. Selain itu, hasil yang diharapkan ketika pengeluaran tidak dianggarkan diterapkan adalah Bahaya. Berdasarkan dana yang diambil, maka data *testing* akan bernilai tersier = "ya", sekunder = "ya", primer = "ya".

Tabel 6. 17 Kasus Uji 6 Algoritma Naïve Bayes

Nama	Kasus Uji 6
Tujuan Pengujian	Untuk mengetahui status keuangan selanjutnya ketika perencanaan alokasi anggaran pengeluaran tidak dianggarkan diterapkan
Jumlah Pengeluaran Tidak Dianggarkan	525,000
Dana Anggaran Yang Diambil	Liburan: 100,000/100,000 Hobi & Hiburan: 100,000/100,000 Zakat/Amal: 100,000/100,000 Gadget: 50,000/50,000 Pendidikan: 100,000/100,000 Transportasi: 75,000/150,000 Kesehatan: 0/100,000 Harian: 0/900,000
Data Testing/Sampel	Tersier = Ya
	Sekunder = Ya
	Primer = Ya
Hasil Yang Diharapkan	Status = Bahaya

Berdasarkan tabel 6.17, tahap berikutnya akan menjelaskan mengenai perhitungan algoritma Naïve Bayes yang dihitung secara manual untuk menentukan hasil yang diharapkan. Dalam perhitungan Naïve Bayes, tahap pertama yang dilakukan adalah menghitung nilai dari probabilitas *prior* ($P(C)$). Setelah nilai *prior* didapat, tahap kedua adalah mencari nilai dari probabilitas *likelihood* ($P(X|C)$). Setelah nilai probabilitas dari *prior* dan *likelihood* didapat, maka tahap selanjutnya mencari nilai dari probabilitas *posterior* ($P(C|X)$) untuk mengetahui hasil yang diharapkan. Untuk lebih jelasnya dapat dilihat pada sub bab berikut.

(a) Probabilitas *Prior* ($P(C)$)

Pada perhitungan mencari nilai probabilitas *prior*, perhitungan akan dilakukan dengan pencarian probabilitas berdasarkan data *training* yang dapat dilihat pada tabel 6.9. Pencarian probabilitas *prior* dilakukan dengan membandingkan antara jumlah dari data *training* dari kategori C dengan jumlah total data *training*.

$$P(\text{Status} = \text{"Aman"}) = 3/8$$

$$P(\text{Status} = \text{"Menengah"}) = 2/8$$

$$P(\text{Status} = \text{"Bahaya"}) = 3/8$$

(b) Probabilitas *Likelihood* ($P(X|C)$)

Pada perhitungan mencari nilai probabilitas *Likelihood*, perhitungan akan dilakukan dengan pencarian probabilitas berdasarkan data *training* yang dapat dilihat pada tabel 6.9. Pencarian probabilitas *likelihood* dilakukan dengan membandingkan nilai – nilai atribut dalam sampel X berdasarkan kondisi pada hipotesis C.

$$P(\text{Tersier} = \text{"Ya"} | \text{Status} = \text{"Aman"}) = 3/3$$

$$P(\text{Tersier} = \text{"Ya"} | \text{Status} = \text{"Menengah"}) = 2/2$$

$$P(\text{Tersier} = \text{"Ya"} | \text{Status} = \text{"Bahaya"}) = 3/3$$

$$P(\text{Sekunder} = \text{"Ya"} | \text{Status} = \text{"Aman"}) = 0/3$$

$$P(\text{Sekunder} = \text{"Ya"} | \text{Status} = \text{"Menengah"}) = 2/2$$

$$P(\text{Sekunder} = \text{"Ya"} | \text{Status} = \text{"Bahaya"}) = 3/3$$

$$P(\text{Primer} = \text{"Ya"} | \text{Status} = \text{"Aman"}) = 0/3$$

$$P(\text{Primer} = \text{"Ya"} | \text{Status} = \text{"Menengah"}) = 0/2$$

$$P(\text{Primer} = \text{"Ya"} | \text{Status} = \text{"Bahaya"}) = 3/3$$

$$P(X | \text{Status} = \text{"Aman"})$$

$$= P(\text{Tersier} = \text{"Ya"}, \text{Sekunder} = \text{"Ya"}, \text{Primer} = \text{"Ya"} | \text{Status} = \text{"Aman"})$$

$$= 3/3 \times 0/3 \times 0/3 = 0$$



$$\begin{aligned}
 &P(X|Status = \text{"Menengah"}) \\
 &= P(Tersier = \text{"Ya"}, Sekunder = \text{"Ya"}, Primer = \text{"Ya"}|Status = \text{"Menengah"}) \\
 &= \frac{2}{2} \times \frac{2}{2} \times \frac{0}{2} = 0
 \end{aligned}$$

$$\begin{aligned}
 &P(X|Status = \text{"Bahaya"}) \\
 &= P(Tersier = \text{"Ya"}, Sekunder = \text{"Ya"}, Primer = \text{"Ya"}|Status = \text{"Bahaya"}) \\
 &= \frac{3}{3} \times \frac{3}{3} \times \frac{3}{3} = 1
 \end{aligned}$$

(c) Probabilitas Posterior ($P(C|X)$)

Setelah nilai dari probabilitas *prior* dan probabilitas *likelihood* didapat, maka probabilitas *posterior* dapat dihitung. Dalam perhitungannya, nilai *posterior* didapat dengan mengalikan nilai probabilitas *likelihood* dan probabilitas *prior*. Setelah dikalikan setiap nilai probabilitas *likelihood* dan probabilitas *prior*, maka selanjutnya dicari nilai yang paling besar dan nilai yang paling besar merupakan hasil akhir perhitungan tersebut.

$$P(X|Status = \text{"Aman"}) \times P(Status = \text{"Aman"}) = 0 \times \frac{3}{8} = 0$$

$$P(X|Status = \text{"Menengah"}) \times P(Status = \text{"Menengah"}) = 0 \times \frac{2}{8} = 0$$

$$P(X|Status = \text{"Bahaya"}) \times P(Status = \text{"Bahaya"}) = 1 \times \frac{3}{8} = 0,375$$

Berdasarkan hasil tersebut, nilai dengan status aman adalah 0, status menengah adalah 0, dan status bahaya adalah 0,375. Hal ini menyatakan, nilai dari status bahaya paling besar, sehingga status yang diharapkan adalah **bahaya**.

6.2.2.7 Kasus Uji 7

Tabel 6.18 merupakan tabel yang menjelaskan kasus uji kedua yang akan dilakukan. Tujuan dilakukan pengujian ini adalah untuk mengetahui status keuangan selanjutnya ketika perencanaan alokasi anggaran pengeluaran tidak dianggarkan diterapkan. Pada kasus uji kedua akan dilakukan pemasukan pengeluaran dengan kategori tidak dianggarkan sebesar 675,000 pada aplikasi. Berdasarkan hasil yang diharapkan pengguna, dana yang diambil hanya anggaran Liburan, Hobi & Hiburan, Zakat/Amal, Gadget, Pendidikan, Transportasi, dan Kesehatan. Selain itu, hasil yang diharapkan ketika pengeluaran tidak dianggarkan diterapkan adalah Bahaya. Berdasarkan dana yang diambil, maka data *testing* akan bernilai tersier = "ya", sekunder = "ya", primer = "ya".

Tabel 6. 18 Kasus Uji 7 Algoritma Naïve Bayes

Nama	Kasus Uji 7
Tujuan Pengujian	Untuk mengetahui status keuangan selanjutnya ketika perencanaan alokasi anggaran pengeluaran tidak dianggarkan diterapkan
Jumlah Pengeluaran Tidak Dianggarkan	675,000

Dana Anggaran Yang Diambil	Liburan: 100,000/100,000 Hobi & Hiburan: 100,000/100,000 Zakat/Amal: 100,000/100,000 Gadget: 50,000/50,000 Pendidikan: 100,000/100,000 Transportasi: 150,000/150,000 Kesehatan: 75,000/100,000 Harian: 0/900,000
Data Testing/Sampel	Tersier = Ya
	Sekunder = Ya
	Primer = Ya
Hasil Yang Diharapkan	Status = Bahaya

Berdasarkan tabel 6.18, tahap berikutnya akan menjelaskan mengenai perhitungan algoritma Naïve Bayes yang dihitung secara manual untuk menentukan hasil yang diharapkan. Dalam perhitungan Naïve Bayes, tahap pertama yang dilakukan adalah menghitung nilai dari probabilitas *prior* ($P(C)$). Setelah nilai *prior* didapat, tahap kedua adalah mencari nilai dari probabilitas *likelihood* ($P(X|C)$). Setelah nilai probabilitas dari *prior* dan *likelihood* didapat, maka tahap selanjutnya mencari nilai dari probabilitas *posterior* ($P(C|X)$) untuk mengetahui hasil yang diharapkan. Untuk lebih jelasnya dapat dilihat pada sub bab berikut.

(a) Probabilitas Prior ($P(C)$)

Pada perhitungan mencari nilai probabilitas *prior*, perhitungan akan dilakukan dengan pencarian probabilitas berdasarkan data *training* yang dapat dilihat pada tabel 6.9. Pencarian probabilitas *prior* dilakukan dengan membandingkan antara jumlah dari data *training* dari kategori C dengan jumlah total data *training*.

$$P(\text{Status} = \text{“Aman”}) = 3/8$$

$$P(\text{Status} = \text{“Menengah”}) = 2/8$$

$$P(\text{Status} = \text{“Bahaya”}) = 3/8$$

(b) Probabilitas Likelihood ($P(X|C)$)

Pada perhitungan mencari nilai probabilitas *Likelihood*, perhitungan akan dilakukan dengan pencarian probabilitas berdasarkan data *training* yang dapat dilihat pada tabel 6.9. Pencarian probabilitas *likelihood* dilakukan dengan membandingkan nilai – nilai atribut dalam sampel X berdasarkan kondisi pada hipotesis C.

$$P(\text{Tersier} = \text{“Ya”} | \text{Status} = \text{“Aman”}) = 3/3$$



$$P(\text{Tersier} = \text{"Ya"} | \text{Status} = \text{"Menengah"}) = 2/2$$

$$P(\text{Tersier} = \text{"Ya"} | \text{Status} = \text{"Bahaya"}) = 3/3$$

$$P(\text{Sekunder} = \text{"Ya"} | \text{Status} = \text{"Aman"}) = 0/3$$

$$P(\text{Sekunder} = \text{"Ya"} | \text{Status} = \text{"Menengah"}) = 2/2$$

$$P(\text{Sekunder} = \text{"Ya"} | \text{Status} = \text{"Bahaya"}) = 3/3$$

$$P(\text{Primer} = \text{"Ya"} | \text{Status} = \text{"Aman"}) = 0/3$$

$$P(\text{Primer} = \text{"Ya"} | \text{Status} = \text{"Menengah"}) = 0/2$$

$$P(\text{Primer} = \text{"Ya"} | \text{Status} = \text{"Bahaya"}) = 3/3$$

$$P(X | \text{Status} = \text{"Aman"})$$

$$= P(\text{Tersier} = \text{"Ya"}, \text{Sekunder} = \text{"Ya"}, \text{Primer} = \text{"Ya"} | \text{Status} = \text{"Aman"})$$

$$= 3/3 \times 0/3 \times 0/3 = 0$$

$$P(X | \text{Status} = \text{"Menengah"})$$

$$= P(\text{Tersier} = \text{"Ya"}, \text{Sekunder} = \text{"Ya"}, \text{Primer} = \text{"Ya"} | \text{Status} = \text{"Menengah"})$$

$$= 2/2 \times 2/2 \times 0/2 = 0$$

$$P(X | \text{Status} = \text{"Bahaya"})$$

$$= P(\text{Tersier} = \text{"Ya"}, \text{Sekunder} = \text{"Ya"}, \text{Primer} = \text{"Ya"} | \text{Status} = \text{"Bahaya"})$$

$$= 3/3 \times 3/3 \times 3/3 = 1$$

(c) Probabilitas Posterior ($P(C|X)$)

Setelah nilai dari probabilitas *prior* dan probabilitas *likelihood* didapat, maka probabilitas *posterior* dapat dihitung. Dalam perhitungannya, nilai *posterior* didapat dengan mengalikan nilai probabilitas *likelihood* dan probabilitas *prior*. Setelah dikalikan setiap nilai probabilitas *likelihood* dan probabilitas *prior*, maka selanjutnya dicari nilai yang paling besar dan nilai yang paling besar merupakan hasil akhir perhitungan tersebut.

$$P(X | \text{Status} = \text{"Aman"}) \times P(\text{Status} = \text{"Aman"}) = 0 \times 3/8 = 0$$

$$P(X | \text{Status} = \text{"Menengah"}) \times P(\text{Status} = \text{"Menengah"}) = 0 \times 2/8 = 0$$

$$P(X | \text{Status} = \text{"Bahaya"}) \times P(\text{Status} = \text{"Bahaya"}) = 1 \times 3/8 = 0,375$$

Berdasarkan hasil tersebut, nilai dengan status aman adalah 0, status menengah adalah 0, dan status bahaya adalah 0,375. Hal ini menyatakan, nilai dari status bahaya paling besar, sehingga status yang diharapkan adalah **bahaya**.

6.2.2.8 Kasus Uji 8

Tabel 6.19 merupakan tabel yang menjelaskan kasus uji kedua yang akan dilakukan. Tujuan dilakukan pengujian ini adalah untuk mengetahui status keuangan selanjutnya



ketika perencanaan alokasi anggaran pengeluaran tidak dianggarkan diterapkan. Pada kasus uji kedua akan dilakukan pemasukan pengeluaran dengan kategori tidak dianggarkan sebesar 1,200,000 pada aplikasi. Berdasarkan hasil yang diharapkan pengguna, dana yang diambil hanya anggaran Liburan, Hobi & Hiburan, Zakat/Amal, Gadget, Pendidikan, Transportasi, dan Kesehatan. Selain itu, hasil yang diharapkan ketika pengeluaran tidak dianggarkan diterapkan adalah Bahaya. Berdasarkan dana yang diambil, maka data *testing* akan bernilai tersier = “ya”, sekunder = “ya”, primer = “ya”.

Tabel 6. 19 Kasus Uji 8 Algoritma Naïve Bayes

Nama	Kasus Uji 8
Tujuan Pengujian	Untuk mengetahui status keuangan selanjutnya ketika perencanaan alokasi anggaran pengeluaran tidak dianggarkan diterapkan
Jumlah Pengeluaran Tidak Dianggarkan	1,200,000
Dana Anggaran Yang Diambil	Liburan: 100,000/100,000 Hobi & Hiburan: 100,000/100,000 Zakat/Amal: 100,000/100,000 Gadget: 50,000/50,000 Pendidikan: 100,000/100,000 Transportasi: 150,000/150,000 Kesehatan: 100,000/100,000 Harian: 500,000/900,000
Data Testing/Sampel	Tersier = Ya
	Sekunder = Ya
	Primer = Ya
Hasil Yang Diharapkan	Status = Bahaya

Berdasarkan tabel 6.19, tahap berikutnya akan menjelaskan mengenai perhitungan algoritma Naïve Bayes yang dihitung secara manual untuk menentukan hasil yang diharapkan. Dalam perhitungan Naïve Bayes, tahap pertama yang dilakukan adalah menghitung nilai dari probabilitas *prior* ($P(C)$). Setelah nilai *prior* didapat, tahap kedua adalah mencari nilai dari probabilitas *likelihood* ($P(X|C)$). Setelah nilai probabilitas dari *prior* dan *likelihood* didapat, maka tahap selanjutnya mencari nilai dari probabilitas *posterior* ($P(C|X)$) untuk mengetahui hasil yang diharapkan. Untuk lebih jelasnya dapat dilihat pada sub bab berikut.



(a) Probabilitas Prior ($P(C)$)

Pada perhitungan mencari nilai probabilitas *prior*, perhitungan akan dilakukan dengan pencarian probabilitas berdasarkan data *training* yang dapat dilihat pada tabel 6.9. Pencarian probabilitas *prior* dilakukan dengan membandingkan antara jumlah dari data *training* dari kategori C dengan jumlah total data *training*.

$$P(\text{Status} = \text{"Aman"}) = 3/8$$

$$P(\text{Status} = \text{"Menengah"}) = 2/8$$

$$P(\text{Status} = \text{"Bahaya"}) = 3/8$$

(b) Probabilitas Likelihood ($P(X|C)$)

Pada perhitungan mencari nilai probabilitas *Likelihood*, perhitungan akan dilakukan dengan pencarian probabilitas berdasarkan data *training* yang dapat dilihat pada tabel 6.9. Pencarian probabilitas *likelihood* dilakukan dengan membandingkan nilai – nilai atribut dalam sampel X berdasarkan kondisi pada hipotesis C.

$$P(\text{Tersier} = \text{"Ya"} | \text{Status} = \text{"Aman"}) = 3/3$$

$$P(\text{Tersier} = \text{"Ya"} | \text{Status} = \text{"Menengah"}) = 2/2$$

$$P(\text{Tersier} = \text{"Ya"} | \text{Status} = \text{"Bahaya"}) = 3/3$$

$$P(\text{Sekunder} = \text{"Ya"} | \text{Status} = \text{"Aman"}) = 0/3$$

$$P(\text{Sekunder} = \text{"Ya"} | \text{Status} = \text{"Menengah"}) = 2/2$$

$$P(\text{Sekunder} = \text{"Ya"} | \text{Status} = \text{"Bahaya"}) = 3/3$$

$$P(\text{Primer} = \text{"Ya"} | \text{Status} = \text{"Aman"}) = 0/3$$

$$P(\text{Primer} = \text{"Ya"} | \text{Status} = \text{"Menengah"}) = 0/2$$

$$P(\text{Primer} = \text{"Ya"} | \text{Status} = \text{"Bahaya"}) = 3/3$$

$$P(X | \text{Status} = \text{"Aman"})$$

$$= P(\text{Tersier} = \text{"Ya"}, \text{Sekunder} = \text{"Ya"}, \text{Primer} = \text{"Ya"} | \text{Status} = \text{"Aman"})$$

$$= 3/3 \times 0/3 \times 0/3 = 0$$

$$P(X | \text{Status} = \text{"Menengah"})$$

$$= P(\text{Tersier} = \text{"Ya"}, \text{Sekunder} = \text{"Ya"}, \text{Primer} = \text{"Ya"} | \text{Status} = \text{"Menengah"})$$

$$= 2/2 \times 2/2 \times 0/2 = 0$$

$$P(X | \text{Status} = \text{"Bahaya"})$$

$$= P(\text{Tersier} = \text{"Ya"}, \text{Sekunder} = \text{"Ya"}, \text{Primer} = \text{"Ya"} | \text{Status} = \text{"Bahaya"})$$

$$= 3/3 \times 3/3 \times 3/3 = 1$$

(c) Probabilitas Posterior ($P(C|X)$)

Setelah nilai dari probabilitas *prior* dan probabilitas *likelihood* didapat, maka probabilitas *posterior* dapat dihitung. Dalam perhitungannya, nilai *posterior* didapat dengan mengalikan nilai probabilitas *likelihood* dan probabilitas *prior*. Setelah dikalikan setiap nilai probabilitas *likelihood* dan probabilitas *prior*, maka selanjutnya dicari nilai yang paling besar dan nilai yang paling besar merupakan hasil akhir perhitungan tersebut.

$$P(X|\text{Status} = \text{"Aman"}) \times P(\text{Status} = \text{"Aman"}) = 0 \times \frac{3}{8} = 0$$

$$P(X|\text{Status} = \text{"Menengah"}) \times P(\text{Status} = \text{"Menengah"}) = 0 \times \frac{2}{8} = 0$$

$$P(X|\text{Status} = \text{"Bahaya"}) \times P(\text{Status} = \text{"Bahaya"}) = 1 \times \frac{3}{8} = 0,375$$

Berdasarkan hasil tersebut, nilai dengan status aman adalah 0, status menengah adalah 0, dan status bahaya adalah 0,375. Hal ini menyatakan, nilai dari status bahaya paling besar, sehingga status yang diharapkan adalah **bahaya**.

6.2.2.9 Pengujian Akurasi

Berdasarkan tabel 6.20, terdapat 8 kasus uji yang sesuai dengan hasil yang diharapkan dari 8 kasus uji. Dibawah ini merupakan perhitungan nilai akurasi pada aplikasi mengenai status keuangan yang diberikan ketika pengeluaran tidak dianggarkan diterapkan.

$$\text{Akurasi (\%)} = \frac{8}{8} \times 100 = 100\%$$

Berdasarkan dari perhitungan akurasi tersebut, dapat disimpulkan bahwa akurasi pada aplikasi ini bernilai 100%.

6.2.2.10 Resume Pengujian Algoritma Naïve Bayes

Tabel 6. 20 Resume Pengujian Algoritma Naïve Bayes

No	Nama	Jumlah Pengeluaran Tidak Dianggarkan	Data Testing	Expected Result	Result	Screenshot
1	Kasus Uji 1	50,000	Tersier = "Ya"	Status = Aman	Status = Aman	
			Sekunder = "Tidak"			
2	Kasus Uji 2	200,000	Tersier = "Ya"	Status = Aman	Status = Aman	
			Sekunder = "Tidak"			

			Primer = "Tidak"			
3	Kasus Uji 3	250,000	Tersier = "Ya"	Status = Menengah	Status = Menengah	
			Sekunder = "Ya"			
4	Kasus Uji 4	325,000	Tersier = "Ya"	Status = Menengah	Status = Menengah	
			Sekunder = "Ya"			

			Primer = "Tidak"			
5	Kasus Uji 5	380,000	Tersier = "Ya"	Status = Menengah	Status = Menengah	
			Sekunder = "Ya"			
6	Kasus Uji 6	525,000	Tersier = "Ya"	Status = Bahaya	Status = Bahaya	
			Sekunder = "Ya"			

			Primer = "Ya"			
7	Kasus Uji 7	675,000	Tersier = "Ya"	Status = Bahaya	Status = Bahaya	
			Sekunder = "Ya"			
8	Kasus Uji 8	1,200,000	Tersier = "Ya"	Status = Bahaya	Status = Bahaya	
			Sekunder = "Ya"			

			Primer = "Ya"			
--	--	--	---------------	--	--	---



6.3 Pengujian Validasi

Pengujian validasi dilakukan untuk memastikan bahwa kebutuhan fungsional dari sistem berjalan sesuai dengan yang diharapkan. Pengujian validasi dilakukan pada seluruh kebutuhan fungsional pada *sprint 1* dan *sprint 2*. Pengujian ini dilakukan dengan metode *black box testing*.

6.3.1 Sprint 1

Tabel 6.21 menjelaskan tentang pengujian fungsionalitas pada FM App yang telah diimplementasikan.

Tabel 6. 21 Daftar Pengujian Fungsionalitas Pada Sprint 1

No	Kode Fungsi	Nama Fungsi	Tujuan Pengujian
1	SRS_FM_F_01	<i>Sign Up</i>	Mengetahui apakah pengguna dapat melakukan <i>sign up</i> ke dalam sistem
2	SRS_FM_F_02	<i>Sign In</i>	Mengetahui apakah pengguna dapat melakukan <i>sign in</i> ke dalam sistem
3	SRS_FM_F_03	Melihat Keuangan Saya	Mengetahui apakah <i>member</i> dapat melihat informasi keuangan pribadinya ke dalam sistem
4	SRS_FM_F_04	Menambah Pengeluaran	Mengetahui apakah <i>member</i> dapat mencatat pengeluaran keuangan ke dalam sistem
5	SRS_FM_F_05	Menambah Pemasukan	Mengetahui apakah <i>member</i> dapat mencatat pemasukan keuangan ke dalam sistem
6	SRS_FM_F_06	Melihat Riwayat Keuangan	Mengetahui apakah <i>member</i> dapat melihat riwayat keuangan berdasarkan tanggal ke dalam sistem

(i) Kasus Uji *Sign Up*

Tabel 6. 22 Kasus Uji *Sign Up*

Nama Kasus Uji	<i>Sign Up</i>
Tujuan Pengujian	Untuk menguji kinerja sistem dalam proses <i>sign up</i> pengguna ke dalam sistem
Prosedur Uji	<ol style="list-style-type: none"> 1. Pengguna memilih tombol <i>here</i> pada halaman utama 2. Pengguna mengisi <i>Sign Up form</i> pada halaman <i>Sign Up</i> 3. Pengguna menekan tombol <i>sign up</i> pada halaman <i>Sign Up</i>



Hasil Yang Diharapkan	Sistem dapat menampilkan halaman Sign Up ketika tombol <i>here</i> ditekan dan pendaftaran <i>member</i> baru berhasil dilakukan setelah tombol <i>sign up</i> ditekan
-----------------------	--

Tabel 6. 23 Kasus Uji Sign Up (Alternative Flow)

Nama Kasus Uji	<i>Sign Up</i>
Tujuan Pengujian	Untuk menguji kinerja sistem dalam proses <i>sign up</i> pengguna ke dalam sistem
Prosedur Uji	1. Pengguna tidak mengisi beberapa kolom pada <i>Sign Up form</i> pada halaman <i>Sign Up</i>
Hasil Yang Diharapkan	Sistem akan menampilkan <i>attention error notification</i> pada setiap kolom yang tidak dimasukkan

(ii) Kasus Uji Sign In

Tabel 6. 24 Kasus Uji Sign In

Nama Kasus Uji	<i>Sign In</i>
Tujuan Pengujian	Untuk menguji kinerja sistem dalam proses <i>sign in</i> pengguna ke dalam sistem
Prosedur Uji	<ol style="list-style-type: none"> 1. Pengguna menekan tombol <i>sign in</i> pada halaman utama 2. Pengguna mengisi <i>Sign In form</i> pada halaman <i>Sign In</i> 3. Pengguna menekan tombol <i>sign in</i> pada halaman <i>Sign In</i>
Hasil Yang Diharapkan	Sistem dapat menampilkan halaman Keuangan Saya dan pengguna berhasil masuk kedalam sistem

Tabel 6. 25 Kasus Uji Sign In (Alternative Flow 1)

Nama Kasus Uji	<i>Sign In</i>
Tujuan Pengujian	Untuk menguji kinerja sistem dalam proses <i>sign in</i> pengguna ke dalam sistem
Prosedur Uji	1. Pengguna memasukkan <i>email</i> dan <i>password</i> yang salah pada <i>Sign In form</i>
Hasil Yang Diharapkan	Sistem dapat menampilkan <i>pop up notification</i> "Wrong Email or Password" pada halaman <i>Sign In</i>



Tabel 6. 26 Kasus Uji Sign In (Alternative Flow 2)

Nama Kasus Uji	<i>Sign In</i>
Tujuan Pengujian	Untuk menguji kinerja sistem dalam proses <i>sign in</i> pengguna ke dalam sistem
Prosedur Uji	1. Pengguna tidak memasukan <i>email</i> dan <i>password</i> pada <i>Sign In form</i>
Hasil Yang Diharapkan	Sistem akan menampilkan <i>attention error notification</i> pada kolom <i>email</i> dan <i>password</i> yang bertuliskan "Required"

Tabel 6. 27 Kasus Uji Sign In (Alternative Flow 3)

Nama Kasus Uji	<i>Sign In</i>
Tujuan Pengujian	Untuk menguji kinerja sistem dalam proses <i>sign in</i> pengguna ke dalam sistem
Prosedur Uji	1. Pengguna lupa <i>password</i> 2. Pengguna menekan tulisan " <i>Here</i> " pada <i>forgot password</i> 3. Pengguna memasukan <i>email</i> yang sudah terdaftar pada <i>forgot password form</i>
Hasil Yang Diharapkan	Sistem akan mengirimkan <i>email</i> perubahan <i>password</i> ke email yang dituliskan pada <i>forgot password form</i>

(iii) Kasus Uji Melihat Keuangan Saya

Tabel 6. 28 Kasus Uji Melihat Keuangan Saya

Nama Kasus Uji	Melihat Keuangan Saya
Tujuan Pengujian	Untuk menguji kinerja sistem menampilkan informasi keuangan <i>member</i>
Prosedur Uji	1. Pengguna berhasil <i>sign in</i> 2. <i>Member</i> menekan tombol <i>icon</i> Keuangan Saya pada <i>bottom navigation</i>
Hasil Yang Diharapkan	Sistem dapat menampilkan informasi keuangannya yang terdiri dari total uang pemasukan, diagram pengeluaran berdasarkan kategori anggaran, dan sisa batas harian

(iv) Kasus Uji Menambah Pengeluaran

Tabel 6. 29 Kasus Uji Menambah Pengeluaran

Nama Kasus Uji	Menambah Pengeluaran
----------------	----------------------



Tujuan Pengujian	Untuk menguji kinerja sistem ketika <i>member</i> mencatat pengeluaran ke dalam sistem
Prosedur Uji	<ol style="list-style-type: none"> 1. Pengguna berhasil <i>sign in</i> 2. <i>Member</i> menekan tombol <i>icon</i> Keuangan Saya pada <i>bottom navigation</i> 3. <i>Member</i> menekan tombol "+Pengeluaran" pada halaman Keuangan Saya 4. <i>Member</i> memilih kategori pengeluaran 5. <i>Member</i> memasukan jumlah pengeluaran 6. <i>Member</i> memasukan note 7. <i>Member</i> menekan tombol submit
Hasil Yang Diharapkan	Sistem dapat mencatat pengeluaran ke dalam sistem yang dapat dilihat pada halaman riwayat keuangan, diagram pada halaman keuangan saya diperbarui, batas harian berkurang jika sebelumnya sudah diterapkan

Tabel 6. 30 Kasus Uji Menambah Pengeluaran (Alternative Flow 1)

Nama Kasus Uji	Menambah Pengeluaran
Tujuan Pengujian	Untuk menguji kinerja sistem ketika <i>member</i> mencatat pengeluaran ke dalam sistem
Prosedur Uji	<ol style="list-style-type: none"> 1. <i>Member</i> memilih kategori Tidak Dianggarkan pada halaman tambah pengeluaran 2. <i>Member</i> memasukan jumlah pengeluaran 3. <i>Member</i> memasukan note 4. <i>Member</i> menekan tombol submit
Hasil Yang Diharapkan	Sistem menampilkan halaman perencanaan alokasi dana tidak dianggarkan

Tabel 6. 31 Kasus Uji Menambah Pengeluaran (Alternative Flow 2)

Nama Kasus Uji	Menambah Pengeluaran
Tujuan Pengujian	Untuk menguji kinerja sistem ketika <i>member</i> mencatat pengeluaran ke dalam sistem
Prosedur Uji	<ol style="list-style-type: none"> 1. <i>Member</i> menekan tombol Ya pada halaman perencanaan alokasi dana tidak dianggarkan
Hasil Yang Diharapkan	Sistem memperbarui diagram pada halaman keuangan saya dan batas harian tidak berkurang



Tabel 6. 32 Kasus Uji Menambah Pengeluaran (Alternative Flow 3)

Nama Kasus Uji	Menambah Pengeluaran
Tujuan Pengujian	Untuk menguji kinerja sistem ketika <i>member</i> mencatat pengeluaran ke dalam sistem
Prosedur Uji	1. <i>Member</i> memilih kategori Tidak Dianggarkan 2. <i>Member</i> memasukan jumlah pengeluaran melebihi batas kemampuan keuangan pribadi
Hasil Yang Diharapkan	Sistem menampilkan <i>attention notification error</i> pada kolom jumlah pengeluaran berupa "Max pengeluaran tidak dianggarkan = Rp..."

Tabel 6. 33 Kasus Uji Menambah Pengeluaran (Alternative Flow 4)

Nama Kasus Uji	Menambah Pengeluaran
Tujuan Pengujian	Untuk menguji kinerja sistem ketika <i>member</i> mencatat pengeluaran ke dalam sistem
Prosedur Uji	1. <i>Member</i> memasukan jumlah pengeluaran tidak dianggarkan melebihi batas kemampuan keuangan pribadi
Hasil Yang Diharapkan	Sistem menampilkan <i>pop up notification</i> berupa "Anda tidak dapat mengeluarkan dana Tidak Dianggarkan!"

(v) Kasus Uji Menambah Pemasukan**Tabel 6. 34 Kasus Uji Menambah Pemasukan**

Nama Kasus Uji	Menambah Pemasukan
Tujuan Pengujian	Untuk menguji kinerja sistem ketika <i>member</i> mencatat pemasukan ke dalam sistem
Prosedur Uji	1. <i>Member</i> berhasil <i>sign in</i> 2. <i>Member</i> membuka halaman Keuangan Saya 3. <i>Member</i> menekan tombol tambah pemasukan pada halaman Keuangan Saya 4. <i>Member</i> memasukan jumlah pemasukan 5. <i>Member</i> memasukan note 6. <i>Member</i> menekan tombol submit
Hasil Yang Diharapkan	Sistem dapat menambah total uang pada halaman keuangan saya sesuai dengan jumlah pemasukan yang telah dimasukan

(vi) Kasus Uji Melihat Riwayat Keuangan

Tabel 6. 35 Kasus Uji Melihat Riwayat Pemasukan

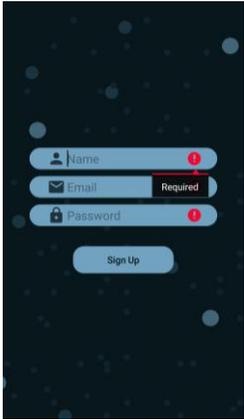
Nama Kasus Uji	Melihat Riwayat Keuangan
Tujuan Pengujian	Untuk menguji kinerja sistem menampilkan Riwayat Keuangan
Prosedur Uji	1. Pengguna berhasil <i>sign in</i> 2. Pengguna menekan tombol <i>icon</i> Riwayat Keuangan pada <i>bottom navigation</i>
Hasil Yang Diharapkan	Sistem dapat menampilkan Riwayat Keuangan yang terdiri dari pemasukan dan pengeluaran

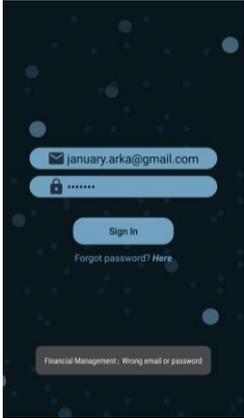
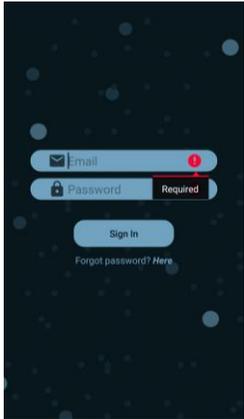


(vii) Resume Pengujian Validasi Fungsionalitas Sprint 1

Tabel 6. 36 Resume Akhir Pengujian Validasi Fungsionalitas

No	Test Name	Test Case	Expected Result	Result	Screenshot	Status
1	Sign Up	<ol style="list-style-type: none"> Pengguna memilih tombol <i>here</i> pada halaman utama Pengguna mengisi <i>Sign Up form</i> pada halaman <i>Sign Up</i> Pengguna menekan tombol <i>sign up</i> pada halaman <i>Sign Up</i> 	Sistem dapat menampilkan halaman Sign Up ketika tombol <i>here</i> ditekan dan pendaftaran <i>member</i> baru berhasil dilakukan setelah tombol <i>sign up</i> ditekan	Sistem dapat menampilkan halaman Sign Up ketika tombol <i>here</i> ditekan dan pendaftaran <i>member</i> baru berhasil dilakukan setelah tombol <i>sign up</i> ditekan		Valid

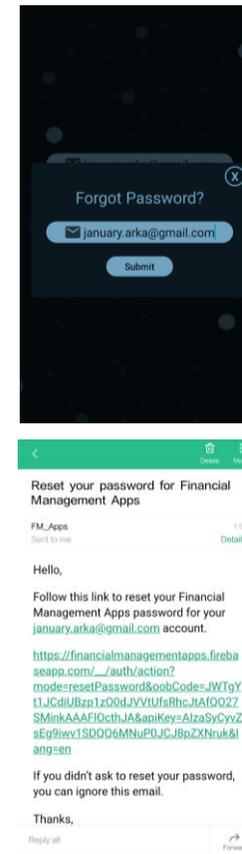
		<ol style="list-style-type: none"> Pengguna tidak mengisi beberapa kolom pada <i>Sign Up form</i> pada halaman <i>Sign Up</i> 	<p>Sistem akan menampilkan <i>attention error notification</i> pada setiap kolom yang tidak dimasukan</p>	<p>Sistem akan menampilkan <i>attention error notification</i> pada setiap kolom yang tidak dimasukan</p>		<p>Valid</p>
2	<i>Sign In</i>	<ol style="list-style-type: none"> Pengguna menekan tombol <i>sign in</i> pada halaman utama Pengguna mengisi <i>Sign In form</i> pada halaman <i>Sign In</i> Pengguna menekan tombol <i>sign in</i> pada halaman <i>Sign In</i> 	<p>Sistem dapat menampilkan halaman Keuangan Saya dan pengguna berhasil masuk kedalam sistem</p>	<p>Sistem dapat menampilkan halaman Keuangan Saya dan pengguna berhasil masuk kedalam sistem</p>		<p>Valid</p>

		<p>1. Pengguna memasukan <i>email</i> dan <i>password</i> yang salah pada <i>Sign In form</i></p>	<p>Sistem dapat menampilkan <i>pop up message</i> “Wrong Email or Password” pada halaman <i>Sign In</i></p>	<p>Sistem dapat menampilkan <i>pop up message</i> “Wrong Email or Password” pada halaman <i>Sign In</i></p>		<p>Valid</p>
		<p>1. Pengguna tidak memasukan <i>email</i> dan <i>password</i> pada <i>Sign In form</i></p>	<p>Sistem akan menampilkan <i>attention error notification</i> pada kolom <i>email</i> dan <i>password</i> yang bertuliskan “Required”</p>	<p>Sistem akan menampilkan <i>attention error notification</i> pada kolom <i>email</i> dan <i>password</i> yang bertuliskan “Required”</p>		<p>Valid</p>

1. Pengguna lupa *password*
2. Pengguna menekan tulisan "*Here*" pada *forgot password*
3. Pengguna memasukan *email* yang sudah terdaftar pada *forgot password form*

Sistem akan mengirimkan *email* perubahan *password* ke email yang dituliskan pada *forgot password form*

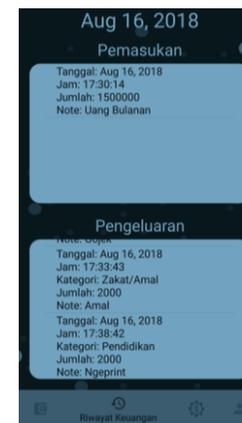
Sistem akan mengirimkan *email* perubahan *password* ke email yang dituliskan pada *forgot password form*



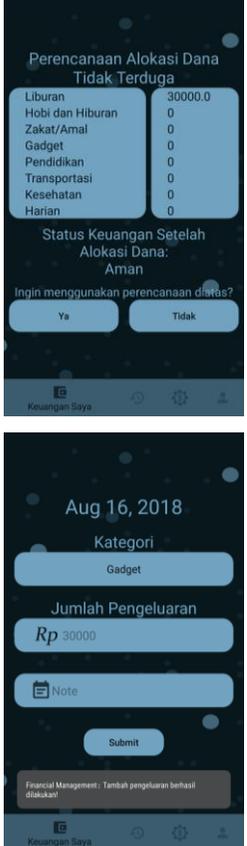
Valid

<p>3</p>	<p>Melihat Keuangan Saya</p>	<ol style="list-style-type: none"> 1. Pengguna berhasil <i>sign in</i> 2. <i>Member</i> menekan tombol <i>icon</i> Keuangan Saya pada <i>bottom navigation</i> 	<p>Sistem dapat menampilkan informasi keuangannya yang terdiri dari total uang pemasukan, diagram pengeluaran berdasarkan kategori anggaran, dan sisa batas harian</p>	<p>Sistem dapat menampilkan informasi keuangannya yang terdiri dari total uang pemasukan, diagram pengeluaran berdasarkan kategori anggaran, dan sisa batas harian</p>		<p>Valid</p>
<p>4</p>	<p>Menambah Pengeluaran</p>	<ol style="list-style-type: none"> 1. Pengguna berhasil <i>sign in</i> 2. <i>Member</i> menekan tombol <i>icon</i> Keuangan Saya pada <i>bottom navigation</i> 3. <i>Member</i> menekan tombol "+Pengeluaran" pada halaman Keuangan Saya 4. <i>Member</i> memilih kategori pengeluaran 5. <i>Member</i> memasukan jumlah pengeluaran 6. <i>Member</i> memasukan note 	<p>Sistem dapat mencatat pengeluaran ke dalam sistem yang dapat dilihat pada halaman riwayat keuangan, diagram pada halaman keuangan saya diperbarui, batas harian berkurang jika sebelumnya sudah diterapkan</p>	<p>Sistem dapat mencatat pengeluaran ke dalam sistem yang dapat dilihat pada halaman riwayat keuangan, diagram pada halaman keuangan saya diperbarui, batas harian berkurang jika sebelumnya sudah diterapkan</p>		<p>Valid</p>

7. *Member* menekan tombol submit

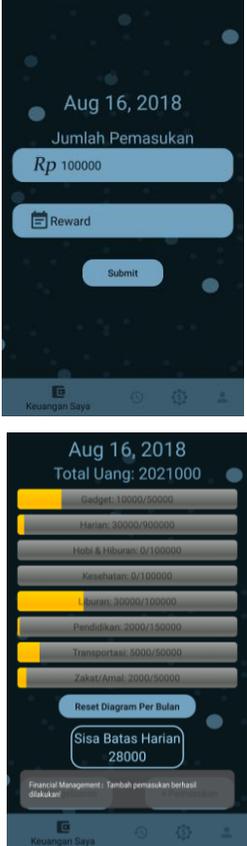


		<ol style="list-style-type: none"> 1. <i>Member</i> memilih kategori Tidak Dianggarkan pada halaman tambah pengeluaran 2. <i>Member</i> memasukan jumlah pengeluaran 3. <i>Member</i> memasukan note 4. <i>Member</i> menekan tombol submit 	<p>Sistem menampilkan halaman perencanaan alokasi dana tidak dianggarkan</p>	<p>Sistem menampilkan halaman perencanaan alokasi dana tidak dianggarkan</p>		<p><i>Valid</i></p>
--	--	---	--	--	--	---------------------

		<p>1. <i>Member</i> menekan tombol Ya pada halaman perencanaan alokasi dana tidak dianggarkan</p>	<p>Sistem memperbarui diagram pada halaman keuangan saya dan batas harian tidak berkurang</p>	<p>Sistem memperbarui diagram pada halaman keuangan saya dan batas harian tidak berkurang</p>		<p><i>Valid</i></p>
--	--	---	---	---	--	---------------------

						
		<ol style="list-style-type: none"> 1. <i>Member</i> memilih kategori Tidak Dianggarkan 2. <i>Member</i> memasukan jumlah pengeluaran melebihi batas kemampuan keuangan pribadi 	<p>Sistem menampilkan <i>attention notification error</i> pada kolom jumlah pengeluaran berupa “Max pengeluaran tidak dianggarkan = Rp...”</p>	<p>Sistem menampilkan <i>attention notification error</i> pada kolom jumlah pengeluaran berupa “Max pengeluaran tidak dianggarkan = Rp...”</p>		<p><i>Valid</i></p>

						
		<p>1. <i>Member</i> memasukan jumlah pengeluaran tidak dianggarkan melebihi batas kemampuan keuangan pribadi</p>	<p>Sistem menampilkan <i>pop up</i> peringatan berupa “Anda tidak dapat mengeluarkan dana Tidak Dianggarkan!”</p>	<p>Sistem menampilkan <i>pop up</i> peringatan berupa “Anda tidak dapat mengeluarkan dana Tidak Dianggarkan!”</p>		<p><i>Valid</i></p>

<p>5</p>	<p>Menambah Pemasukan</p>	<ol style="list-style-type: none"> 1. <i>Member</i> berhasil <i>sign in</i> 2. <i>Member</i> membuka halaman Keuangan Saya 3. <i>Member</i> menekan tombol tambah pemasukan pada halaman Keuangan Saya 4. <i>Member</i> memasukan jumlah pemasukan 5. <i>Member</i> memasukan note 6. <i>Member</i> menekan tombol submit 	<p>Sistem dapat menambah total uang pada halaman keuangan saya sesuai dengan jumlah pemasukan yang telah dimasukan</p>	<p>Sistem dapat menambah total uang pada halaman keuangan saya sesuai dengan jumlah pemasukan yang telah dimasukan</p>		<p>Valid</p>
----------	---------------------------	---	--	--	--	--------------

6	Melihat Riwayat Keuangan	<ol style="list-style-type: none"> 1. Pengguna berhasil <i>sign in</i> 2. Pengguna menekan tombol <i>icon</i> Riwayat Keuangan pada <i>bottom navigation</i> 	Sistem dapat menampilkan Riwayat Keuangan yang terdiri dari pemasukan dan pengeluaran	Sistem dapat menampilkan Riwayat Keuangan yang terdiri dari pemasukan dan pengeluaran		Valid
---	--------------------------	--	---	---	---	-------

6.3.2 Sprint 2

Tabel 6.37 menjelaskan tentang pengujian fungsionalitas pada FM App yang telah diimplementasikan.

Tabel 6. 37 Daftar Pengujian Fungsionalitas Pada *Sprint 2*

No	Kode Fungsi	Nama Fungsi	Tujuan Pengujian
1	SRS_FM_F_07	Melihat Perencanaan Keuangan	Mengetahui apakah <i>member</i> dapat memilih perencanaan keuangan pada sistem
2	SRS_FM_F_08	Membuat Perencanaan Anggaran	Mengetahui apakah <i>member</i> dapat membuat perencanaan anggaran keuangan pada sistem
3	SRS_FM_F_09	Membuat Batas Harian	Mengetahui apakah <i>member</i> dapat membuat batas harian keuangan pada sistem
4	SRS_FM_F_10	Membuat Perencanaan Pembelian	Mengetahui apakah <i>member</i> dapat membuat perencanaan pembelian pada sistem
5	SRS_FM_F_11	Melihat Profil	Mengetahui apakah <i>member</i> dapat melihat informasi pengguna pada sistem

(i) Kasus Uji Melihat Perencanaan Keuangan

Tabel 6. 38 Kasus Uji Melihat Perencanaan Keuangan

Nama Kasus Uji	Halaman Perencanaan Keuangan
Tujuan Pengujian	Untuk menguji kinerja sistem menampilkan pilihan perencanaan keuangan kepada <i>member</i>
Prosedur Uji	1. Pengguna berhasil <i>sign in</i> 2. <i>Member</i> menekan tombol <i>icon</i> Perencanaan Keuangan pada <i>bottom navigation</i>
Hasil Yang Diharapkan	Sistem dapat menampilkan pilihan perencanaan keuangan kepada <i>member</i>

(ii) Kasus Uji Membuat Perencanaan Anggaran

Tabel 6. 39 Kasus Uji Membuat Perencanaan Anggaran

Nama Kasus Uji	Membuat Perencanaan Anggaran
Tujuan Pengujian	Untuk menguji kinerja ketika <i>member</i> membuat perencanaan anggaran keuangan
Prosedur Uji	1. Pengguna berhasil <i>sign in</i>

	<ol style="list-style-type: none"> 2. <i>Member</i> menekan tombol <i>icon</i> Perencanaan Keuangan pada <i>bottom navigation</i> 3. <i>Member</i> menekan tombol perencanaan anggaran pada halaman Perencanaan Keuangan 4. <i>Member</i> memasukkan jumlah anggaran setiap kategori yang tersedia 5. <i>Member</i> menekan tombol submit
Hasil Yang Diharapkan	Sistem dapat mengalokasikan keuangannya berdasarkan kategori yang tersedia dan nilai maksimum diagram setiap kategori pada halaman Keuangan Saya diperbarui

Tabel 6. 40 Kasus Uji Membuat Perencanaan Anggaran (Alternative Flow 1)

Nama Kasus Uji	Membuat Perencanaan Anggaran
Tujuan Pengujian	Untuk menguji kinerja ketika <i>member</i> membuat perencanaan anggaran keuangan
Prosedur Uji	6. <i>Member</i> menekan tulisan " <i>here</i> " pada halaman Perencanaan Anggaran
Hasil Yang Diharapkan	Sistem dapat menampilkan keterangan kategori anggaran

Tabel 6. 41 Kasus Uji Membuat Perencanaan Anggaran (Alternative Flow 2)

Nama Kasus Uji	Membuat Perencanaan Anggaran
Tujuan Pengujian	Untuk menguji kinerja ketika <i>member</i> membuat perencanaan anggaran keuangan
Prosedur Uji	7. <i>Member</i> memasukkan jumlah alokasi dana pada setiap kategori memiliki total melebihi jumlah total uang yang telah dimasukan pada sistem
Hasil Yang Diharapkan	Sistem dapat menampilkan <i>pop up notification</i> berupa "Total anggaran melebihi total keuangan Anda! Tambah Pemasukan!"

(iii) Kasus Uji Membuat Batas Harian

Tabel 6. 42 Kasus Uji Membuat Batas Harian

Nama Kasus Uji	Membuat Batas Harian
Tujuan Pengujian	Untuk menguji kinerja sistem ketika <i>member</i> membuat batas pengeluaran harian dan pengingat ketika pengeluaran sudah mendekati batas
Prosedur Uji	1. Pengguna berhasil <i>sign in</i>



	<ol style="list-style-type: none"> 2. <i>Member</i> menekan tombol <i>icon</i> Perencanaan Keuangan pada <i>bottom navigation</i> 3. <i>Member</i> menekan tombol batas harian pada halaman Perencanaan Keuangan 4. <i>Member</i> memasukan jumlah batas harian 5. <i>Member</i> mengaktifkan <i>reminder</i> 6. <i>Member</i> menekan tombol <i>submit</i>
Hasil Yang Diharapkan	Sistem dapat menampilkan sisa batas pengeluaran harian pada halaman keuangan saya, <i>member</i> akan mendapatkan reminder batas harian ketika sudah mendekati batas, dan pada halaman keuangan saya selalu diperbarui setiap ada pengeluaran kecuali, nabung dan dana tidak dianggarkan

Tabel 6. 43 Kasus Uji Membuat Batas Harian (Alternative Flow 1)

Nama Kasus Uji	Membuat Batas Harian
Tujuan Pengujian	Untuk menguji kinerja sistem ketika <i>member</i> membuat batas pengeluaran harian dan pengingat ketika pengeluaran sudah mendekati batas
Prosedur Uji	<ol style="list-style-type: none"> 1. <i>Member</i> memasukan jumlah batas harian 2. <i>Member</i> menekan tombol <i>submit</i>
Hasil Yang Diharapkan	Sistem dapat menampilkan <i>attention error notification</i> berupa "Maksimum batas harian ..."

(iv) Kasus Uji Membuat Perencanaan Pembelian

Tabel 6. 44 Kasus Uji Membuat Perencanaan Pembelian

Nama Kasus Uji	Membuat Perencanaan Pembelian
Tujuan Pengujian	Untuk menguji kinerja sistem ketika <i>member</i> membuat perencanaan pembelian pada sistem
Prosedur Uji	<ol style="list-style-type: none"> 1. Pengguna berhasil <i>sign in</i> 2. <i>Member</i> menekan tombol <i>icon</i> Perencanaan Keuangan pada <i>bottom navigation</i> 3. <i>Member</i> menekan tombol perencanaan pembelian pada halaman Perencanaan Keuangan 4. <i>Member</i> menekan tombol berdasarkan jumlah tabungan pada halaman Perencanaan Pembelian 5. <i>Member</i> memasukan harga produk, jumlah nabung perhari, menekan <i>toggle</i> pada <i>show reminder</i>, dan tombol <i>submit</i> ditekan
Hasil Yang Diharapkan	Sistem dapat menampilkan progress uang yang terkumpul untuk membeli barang pada halaman



	Perencanaan Pembelian dan akan mendapatkan notifikasi jika <i>toggle</i> pada <i>show reminder</i> bernilai on
--	--

Tabel 6. 45 Kasus Uji Membuat Perencanaan Pembelian (Alternative Flow 1)

Nama Kasus Uji	Membuat Perencanaan Pembelian
Tujuan Pengujian	Untuk menguji kinerja sistem ketika <i>member</i> membuat perencanaan pembelian pada sistem
Prosedur Uji	<ol style="list-style-type: none"> 1. Pengguna berhasil <i>sign in</i> 2. <i>Member</i> menekan tombol <i>icon</i> Perencanaan Keuangan pada <i>bottom navigation</i> 3. <i>Member</i> menekan tombol perencanaan pembelian pada halaman Perencanaan Keuangan 4. <i>Member</i> menekan tombol berdasarkan tanggal pembelian pada halaman Perencanaan Pembelian 5. <i>Member</i> memasukkan harga produk, tanggal pembelian, menekan <i>toggle</i> pada <i>show reminder</i>, dan tombol <i>submit</i> ditekan
Hasil Yang Diharapkan	Sistem dapat menampilkan progress uang yang terkumpul untuk membeli barang pada halaman Perencanaan Pembelian dan akan mendapatkan notifikasi jika <i>toggle</i> pada <i>show reminder</i> bernilai on

(v) Kasus Uji Melihat Profil

Tabel 6. 46 Kasus Uji Melihat Profil

Nama Kasus Uji	Melihat Profil
Tujuan Pengujian	Untuk menguji kinerja sistem menampilkan data profil <i>member</i>
Prosedur Uji	<ol style="list-style-type: none"> 1. Pengguna berhasil <i>sign in</i> 2. <i>Member</i> menekan tombol <i>icon</i> profil pada <i>bottom navigation</i>
Hasil Yang Diharapkan	Sistem dapat menampilkan data profil <i>member</i>

Tabel 6. 47 Kasus Uji Melihat Profil (Alternative Flow 1)

Nama Kasus Uji	Melihat Profil
Tujuan Pengujian	Untuk menguji kinerja sistem menampilkan data profil <i>member</i>
Prosedur Uji	<ol style="list-style-type: none"> 1. <i>Member</i> menekan tombol <i>change password</i> pada halaman Profil

Hasil Yang Diharapkan	Sistem dapat menampilkan halaman untuk merubah <i>password</i> dan jika <i>password</i> baru dimasukan dan tombol submit ditekan, maka akan otomatis keluar dari sistem
-----------------------	---

Tabel 6. 48 Kasus Uji Melihat Profil (Alternative Flow 2)

Nama Kasus Uji	Melihat Profil
Tujuan Pengujian	Untuk menguji kinerja sistem menampilkan data profil <i>member</i>
Prosedur Uji	1. <i>Member</i> menekan tombol <i>sign out</i> pada halaman Profil
Hasil Yang Diharapkan	Sistem dapat mengakhiri <i>session</i> sebagai <i>member</i> dan halaman Utama ditampilkan



(vi) Resume Akhir Pengujian Validasi Fungsionalitas *Sprint 2*

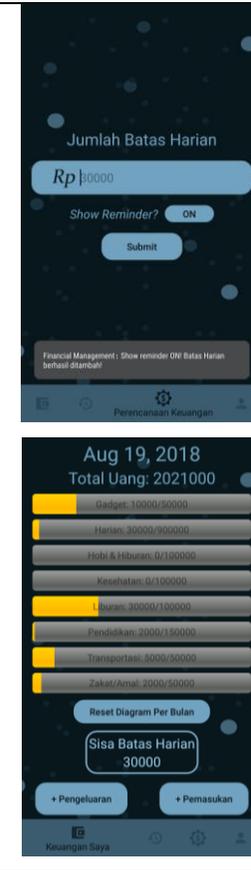
Tabel 6. 49 Resume Akhir Pengujian Validasi Fungsionalitas *Sprint 2*

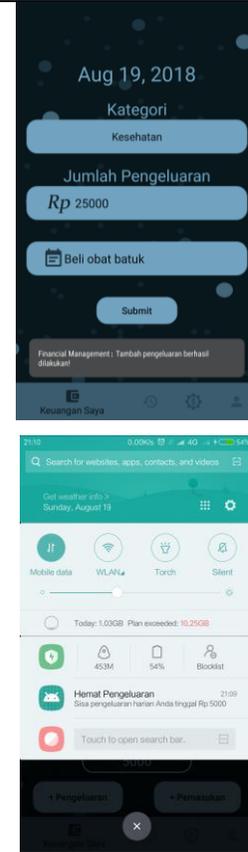
No	Test Name	Test Case	Expected Result	Result	Screenshot	Status
1	Melihat Perencanaan Keuangan	<ol style="list-style-type: none"> Pengguna berhasil <i>sign in</i> <i>Member</i> menekan tombol <i>icon</i> Perencanaan Keuangan pada <i>bottom navigation</i> 	Sistem dapat menampilkan pilihan perencanaan keuangan kepada <i>member</i>	Sistem dapat menampilkan pilihan perencanaan keuangan kepada <i>member</i>		Valid
2	Membuat Perencanaan Anggaran	<ol style="list-style-type: none"> Pengguna berhasil <i>sign in</i> <i>Member</i> menekan tombol <i>icon</i> Perencanaan Keuangan pada <i>bottom navigation</i> <i>Member</i> menekan tombol perencanaan anggaran pada 	Sistem dapat mengalokasikan keuangannya berdasarkan kategori yang tersedia dan nilai maksimum diagram setiap kategori pada halaman Keuangan Saya diperbarui	Sistem dapat mengalokasikan keuangannya berdasarkan kategori yang tersedia dan nilai maksimum diagram setiap kategori pada halaman Keuangan Saya diperbarui		Valid

		<p>halaman Perencanaan Keuangan</p> <p>4. <i>Member</i> memasukan jumlah anggaran setiap kategori yang tersedia</p> <p>5. <i>Member</i> menekan tombol submit</p>				
		<p>1. <i>Member</i> menekan tulisan "here" pada halaman Perencanaan Anggaran</p>	<p>Sistem dapat menampilkan keterangan kategori anggaran</p>	<p>Sistem dapat menampilkan keterangan kategori anggaran</p>		<p>Valid</p>

		<ol style="list-style-type: none"> 1. <i>Member</i> memasukan jumlah alokasi dana pada setiap kategori memiliki total melebihi jumlah total uang yang telah dimasukan pada sistem 	<p>Sistem dapat menampilkan <i>pop up notification</i> berupa “Total anggaran melebihi total keuangan Anda! Tambah Pemasukan!”</p>	<p>Sistem dapat menampilkan <i>pop up notification</i> berupa “Total anggaran melebihi total keuangan Anda! Tambah Pemasukan!”</p>		Valid
3	Membuat Batas Harian	<ol style="list-style-type: none"> 1. Pengguna berhasil <i>sign in</i> 2. <i>Member</i> menekan tombol <i>icon</i> Perencanaan Keuangan pada <i>bottom navigation</i> 3. <i>Member</i> menekan tombol batas harian pada halaman Perencanaan Keuangan 4. <i>Member</i> memasukan jumlah batas harian 	<p>Sistem dapat menampilkan sisa batas pengeluaran harian pada halaman keuangan saya, <i>member</i> akan mendapatkan reminder batas harian ketika sudah mendekati batas, dan pada halaman keuangan saya selalu diperbarui setiap ada pengeluaran kecuali, nabung dan dana tidak dianggarkan</p>	<p>Sistem dapat menampilkan sisa batas pengeluaran harian pada halaman keuangan saya, <i>member</i> akan mendapatkan reminder batas harian ketika sudah mendekati batas, dan pada halaman keuangan saya selalu diperbarui setiap ada pengeluaran kecuali, nabung dan dana tidak dianggarkan</p>		Valid

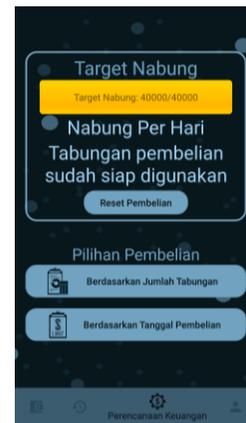
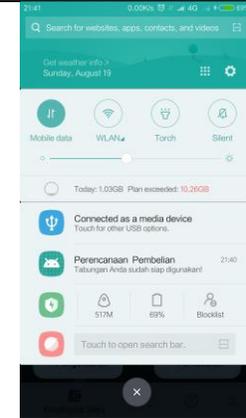
5. *Member* mengaktifkan *reminder*
6. *Member* menekan tombol *submit*





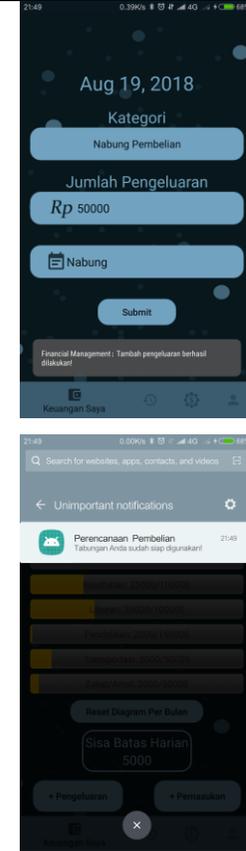
		<ol style="list-style-type: none"> 1. <i>Member</i> memasukan jumlah batas harian 2. <i>Member</i> menekan tombol <i>submit</i> 	<p>Sistem dapat menampilkan <i>attention error notification</i> berupa “Maksimum batas harian ...”</p>	<p>Sistem dapat menampilkan <i>attention error notification</i> berupa “Maksimum batas harian ...”</p>		<p><i>Valid</i></p>
4	Membuat Perencanaan Pembelian	<ol style="list-style-type: none"> 1. Pengguna berhasil <i>sign in</i> 2. <i>Member</i> menekan tombol <i>icon</i> Perencanaan Keuangan pada <i>bottom navigation</i> 3. <i>Member</i> menekan tombol perencanaan pembelian pada halaman Perencanaan Keuangan 4. <i>Member</i> menekan tombol berdasarkan jumlah 	<p>Sistem dapat menampilkan progress uang yang terkumpul untuk membeli barang pada halaman Perencanaan Pembelian dan akan mendapatkan notifikasi jika <i>toggle</i> pada <i>show reminder</i> bernilai on</p>	<p>Sistem dapat menampilkan progress uang yang terkumpul untuk membeli barang pada halaman Perencanaan Pembelian dan akan mendapatkan notifikasi jika <i>toggle</i> pada <i>show reminder</i> bernilai on</p>		<p><i>Valid</i></p>

		<p>tabungan pada halaman Perencanaan Pembelian</p> <p>5. <i>Member</i> memasukan harga produk, jumlah nabung perhari, menekan <i>toogle</i> pada <i>show reminder</i>, dan tombol <i>submit</i> ditekan</p>			
--	--	---	---	--	--



		<ol style="list-style-type: none"> 1. Pengguna berhasil <i>sign in</i> 2. <i>Member</i> menekan tombol <i>icon</i> Perencanaan Keuangan pada <i>bottom navigation</i> 3. <i>Member</i> menekan tombol perencanaan pembelian pada halaman Perencanaan Keuangan 4. <i>Member</i> menekan tombol berdasarkan tanggal pembelian pada halaman Perencanaan Pembelian 5. <i>Member</i> memasukkan harga produk, tanggal pembelian, menekan <i>toogle</i> pada <i>show reminder</i>, dan 	<p>Sistem dapat menampilkan progress uang yang terkumpul untuk membeli barang pada halaman Perencanaan Pembelian dan akan mendapatkan notifikasi jika <i>toggle</i> pada <i>show reminder</i> bernilai on</p>	<p>Sistem dapat menampilkan progress uang yang terkumpul untuk membeli barang pada halaman Perencanaan Pembelian dan akan mendapatkan notifikasi jika <i>toggle</i> pada <i>show reminder</i> bernilai on</p>		<p>Valid</p>
--	--	---	---	---	--	--------------

tombol *submit* ditekan



						
5	Melihat Profil	<ol style="list-style-type: none"> 1. Pengguna berhasil <i>sign in</i> 2. <i>Member</i> menekan tombol <i>icon</i> profil pada <i>bottom navigation</i> 	Sistem dapat menampilkan data profil <i>member</i>	Sistem dapat menampilkan data profil <i>member</i>		Valid

		<p>1. <i>Member</i> menekan tombol <i>change password</i> pada halaman Profil</p>	<p>Sistem dapat menampilkan halaman untuk merubah <i>password</i> dan jika <i>password</i> baru dimasukan dan tombol submit ditekan, maka akan otomatis keluar dari sistem</p>	<p>Sistem dapat menampilkan halaman untuk merubah <i>password</i> dan jika <i>password</i> baru dimasukan dan tombol submit ditekan, maka akan otomatis keluar dari sistem</p>		<p>Valid</p>
--	--	---	--	--	--	--------------

		<p>1. <i>Member</i> menekan tombol <i>sign out</i> pada halaman Profil</p>	<p>Sistem dapat mengakhiri <i>session</i> sebagai <i>member</i> dan halaman Utama ditampilkan</p>	<p>Sistem dapat mengakhiri <i>session</i> sebagai <i>member</i> dan halaman Utama ditampilkan</p>		<p><i>Valid</i></p>
--	--	--	---	---	---	---------------------



6.4 Pengujian *Usability*

Pengujian *usability* dilakukan untuk menguji tingkat kemudahan responden dalam menggunakan aplikasi FM App. Metode yang digunakan pada pengujian *usability* adalah *System Usability Scale* (SUS). Berdasarkan lampiran C, pengujian ini meminta responden untuk menggunakan aplikasi dengan menintruksikan mencatat pemasukan keuangan, membuat perencanaan keuangan, membuat batas pengeluaran, mencatat pengeluaran keuangan, melihat riwayat keuangan, membuat perencanaan pembelian, dan melihat halaman profil, lalu menjawab 10 pertanyaan kuesioner yang terdiri dari pertanyaan positif pada nomor pertanyaan ganjil dan pertanyaan negatif pada nomor pertanyaan genap. Pertanyaan memiliki pilihan jawaban 1 – 5, dimana dalam peniliannya dilakukan dengan pengurangan 1 nilai dari respon yang diberikan responden pada pertanyaan ganjil dan 5 dikurangi dengan respon yang diberikan responden pada pertanyaan genap. Nilai SUS akan didapat dengan mengalikan total nilai dengan 2,5. Nilai akhir SUS akan berada pada kisaran 0 – 100.

Pada tabel 6.50 merupakan daftar hasil kuesioner yang telah diajukan kepada responden. Hasil kuesioner ini digunakan sebagai sample pengguna sistem dalam menentukan tingkat kemudahan penggunaan FM App.

Tabel 6. 50 Nilai Hasil Kuesioner Pengujian SUS

No	Responden	Nilai Pertanyaan ke-									
		1	2	3	4	5	6	7	8	9	10
1	R1	5	2	5	3	5	2	4	2	5	2
2	R2	4	2	3	3	5	2	4	2	4	3
3	R3	5	1	4	4	4	2	5	2	5	3
4	R4	5	1	4	3	5	2	4	1	4	2
5	R5	4	3	5	2	4	2	5	1	4	1

Pada tabel 6.51 merupakan daftar nilai kuesioner yang sudah dikonversi kedalam perhitungan *Usability*.

Tabel 6. 51 Nilai Hasil Kuesioner Pengujian SUS Setelah di Konversi

No	Responden	Nilai Pertanyaan ke-										Total Skor
		1	2	3	4	5	6	7	8	9	10	
1	R1	4	3	4	2	4	3	3	3	4	3	33
2	R2	3	3	2	2	4	3	3	3	3	2	28
3	R3	4	4	3	1	3	3	4	3	4	2	31
4	R4	4	4	3	2	4	3	3	4	3	3	33
5	R5	3	2	4	3	3	3	4	4	3	4	33

Total Akhir Skor	158
-------------------------	-----

Berdasarkan tabel 6.51, maka dapat dilanjutkan dengan perhitungan berikut:

$$Usability = \frac{\sum Total\ Skor}{\sum Responden} \times 2,5 \quad (6.13)$$

$$Usability = \frac{158}{5} \times 2,5 = 79 \quad (6.14)$$

6.5 Analisis Hasil Pengujian

Setelah semua pengujian telah selesai dilakukan, maka hasil dari seluruh pengujian dianalisis untuk mendapatkan kesimpulan. Karena pada penelitian ini menggunakan Scrum, analisis hasil pengujian ini masuk ke dalam *sprint review*.

6.5.1 Analisis Hasil Pengujian Unit

6.5.1.1 Sprint 1

Dari semua pengujian yang telah dilakukan pada *sprint 1*, yaitu pengujian terhadap *method* `signIn()` pada klas `SignIn` dan `DataUji()` pada klas `TambahPengeluaran`, dihasilkan beberapa *basis path* yang merupakan semua kemungkinan yang dapat terjadi pada *method* tersebut. Dari semua kemungkinan tersebut, dilakukan pengujian dan mendapatkan hasil yang *valid*, sehingga semua pengujian sudah sesuai dengan kebutuhan yang telah ditentukan pada *sprint 1*.

6.5.1.2 Sprint 2

Dari pengujian yang telah dilakukan pada *sprint 2*, yaitu pengujian terhadap *method* `onClick()` pada klas `BudgetPlanning`, dihasilkan beberapa *basis path* yang merupakan semua kemungkinan yang dapat terjadi pada *method* tersebut. Dari semua kemungkinan tersebut, dilakukan pengujian dan mendapatkan hasil yang *valid*, sehingga semua pengujian sudah sesuai dengan kebutuhan yang telah ditentukan pada *sprint 2*.

6.5.2 Analisis Hasil Pengujian Algoritma Naïve Bayes

Pengujian terhadap algoritma Naïve Bayes dilakukan dengan 8 kasus uji. Pengujian ini dilakukan untuk menguji hasil rekomendasi apakah sesuai dengan harapan pengguna atau tidak ketika pengeluaran tidak dianggarkan dikeluarkan. Dalam perhitungan Naïve Bayes, tahap pertama yang dilakukan adalah menghitung nilai dari probabilitas *prior* ($P(C)$). Setelah nilai *prior* didapat, tahap kedua adalah mencari nilai dari probabilitas *likelihood* ($P(X|C)$). Setelah nilai probabilitas dari *prior* dan *likelihood* didapat, maka tahap selanjutnya mencari nilai dari probabilitas *posterior* ($P(C|X)$) untuk mengetahui hasil yang diharapkan. Dari kedelapan kasus uji tersebut, memiliki kesesuaian dengan harapan yang diinginkan. Dengan kata lain, algoritma Naïve Bayes yang diterapkan pada aplikasi FM App memiliki tingkat akurasi yang tinggi, yaitu 100%.

6.5.3 Analisis Hasil Pengujian Validasi

6.5.3.1 Sprint 1

Pada pengujian fungsional menggunakan *black box testing*. Proses pada pengujian aplikasi ini untuk mengetahui apakah ada kesalahan atau ada fungsi yang tidak berjalan sesuai harapan. Berikut ini analisis hasil pengujian masing – masing fungsi yang sudah ditentukan:

1. Pengujian *Sign Up* agar sistem dapat menampilkan halaman Sign Up ketika tombol *here* ditekan dan pendaftaran *member* baru berhasil dilakukan setelah tombol *submit* ditekan.
2. Pengujian *Sign In* agar sistem dapat menampilkan halaman Keuangan Saya dan pengguna berhasil masuk kedalam aplikasi.
3. Pengujian melihat keuangan agar sistem dapat menampilkan halaman Keuangan Saya ketika pengguna berhasil sign in dan saat tombol icon Keuangan Saya ditekan. Selain itu, halaman tambah pengeluaran ditampilkan ketika tombol tambah pengeluaran ditekan dan halaman tambah pemasukan ditampilkan ketika tombol tambah pemasukan ditekan.
4. Pengujian menambah pengeluaran agar sistem dapat menampilkan halaman Pengeluaran dan data pengeluaran tersimpan pada *database*.
5. Pengujian menambah pemasukan agar sistem dapat menampilkan halaman Pemasukan dan data pemasukan tersimpan pada database.
6. Pengujian melihat riwayat keuangan agar sistem dapat menampilkan halaman Riwayat Keuangan.

Berdasarkan analisis hal tersebut didapatkan kesimpulan bahwa implementasi dari fungsional sistem FM App memberikan hasil yang sesuai dengan kebutuhan yang ada pada *sprint 1*.

6.5.3.2 Sprint 2

Pada pengujian fungsional menggunakan *black box testing*. Proses pada pengujian aplikasi ini untuk mengetahui apakah ada kesalahan atau ada fungsi yang tidak berjalan sesuai harapan. Berikut ini analisis hasil pengujian masing – masing fungsi yang sudah ditentukan:

1. Pengujian melihat perencanaan keuangan agar sistem dapat menampilkan halaman Perencanaan Keuangan.
2. Pengujian membuat perencanaan anggaran agar sistem dapat menampilkan halaman Perencanaan Anggaran dan halaman Keterangan Kategori Anggaran ketika tombol *here* pada halaman Perencanaan Anggaran ditekan.
3. Pengujian membuat batas harian agar sistem dapat menampilkan halaman Batas Harian.
4. Pengujian membuat perencanaan pembelian agar sistem dapat menampilkan halaman Perencanaan Pembelian, menampilkan halaman Berdasarkan Jumlah Tabungan ketika tombol berdasarkan Jumlah Tabungan ditekan, dan menampilkan halaman Berdasarkan Tanggal Pembelian ketika tombol berdasarkan tanggal pembelian ditekan.

5. Pengujian melihat profil agar sistem dapat menampilkan halaman Profil.

Berdasarkan analisis hal tersebut didapatkan kesimpulan bahwa implementasi dan fungsional sistem FM App memberikan hasil yang sesuai dengan kebutuhan yang ada pada *sprint* 2.

6.5.4 Analisis Hasil Pengujian *Usability*

Berdasarkan hasil pengujian yang telah dilakukan, maka dapat disimpulkan bahwa responden dapat menggunakan aplikasi dengan baik. Skor pengujian *usability* pada FM App memiliki nilai rata – rata adalah 79 dan berdasarkan penilai dari Sauro pada gambar 2.10 menyatakan nilai B untuk hasil pengujian *usability*. Hal ini menunjukkan bahwa aplikasi dapat digolongkan sebagai aplikasi yang mudah digunakan. Akan tetapi, dalam pengujian ini terdapat hal yang memungkinkan untuk dievaluasi, yaitu masih terdapat 1 orang yang memberikan nilai 3 pada pernyataan “Saya menemukan bahwa aplikasi ini, tidak harus dibuat serumit ini”, 1 orang memberikan nilai 3 pada pernyataan “Saya pikir aplikasi ini mudah untuk digunakan, 3 orang memberikan nilai 3 pada pernyataan “Saya pikir bahwa saya akan membutuhkan bantuan dari orang teknis untuk dapat menggunakan aplikasi ini”, dan 2 orang pada pernyataan “Saya perlu belajar banyak hal sebelum saya bisa memulai menggunakan aplikasi ini”.



BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan hasil rekayasa kebutuhan, perancangan dan implementasi, serta pengujian yang telah dilakukan sebelumnya, maka dapat diambil kesimpulan sebagai berikut:

1. Dengan menerapkan metode Scrum dalam pengembangan aplikasi *mobile* manajemen keuangan, maka pengguna dapat ikut berperan langsung dalam perancangannya. Selain itu, untuk mempercepat pengembangan sistem akan dilakukan *sprint planning*. Pada penerapan *sprint planning* akan dilakukan perencanaan daftar pekerjaan untuk mencapai tujuan utama, yaitu dengan membuat *sprint planning* dan *sprint goal*. Pada *sprint goal*, tujuan utama akan dibagi menjadi tujuan – tujuan kecil, sedangkan *sprint backlog* merupakan daftar pekerjaan yang akan dikerjakan untuk mencapai *sprint goal*. Setelah *sprint planning* dilakukan, maka dilakukan tahap perancangan dan implementasi yaitu, *development* dan *daily scrum*. Setelah itu, dilakukan tahap pengujian, yaitu *sprint review*. Disamping itu, pada setiap pengerjaan dalam satu *sprint*, pengguna ikut berperan langsung untuk memberikan *feedback* aplikasi.
2. Rekayasa kebutuhan didapat dari penyebaran kuesioner kepada mahasiswa FILKOM Universitas Brawijaya. Dari hasil kuesioner tersebut didapat 11 kebutuhan fungsional dan 1 kebutuhan non – fungsional untuk menunjang aplikasi *mobile* manajemen keuangan.
3. Fitur *reminder* pada perencanaan batas harian aplikasi akan memberikan notifikasi ketika pengeluaran sudah mendekati nilai minimum batas harian yang telah ditentukan, sehingga pengeluaran melewati batas dapat diatasi.
4. Dengan menggunakan metode Naïve Bayes dalam memberikan rekomendasi status keuangan setelah pengeluaran tidak terduga dilakukan, maka status keuangan pengguna dapat diketahui aman atau tidak ketika pengeluaran tidak terduga dilakukan. Metode ini memiliki tingkat akurasi rekomendasi sebesar 100% setelah diuji.
5. Dengan adanya fitur perencanaan pembelian berdasarkan jumlah menabung atau penentuan tanggal pembelian pada aplikasi, menjadikan solusi untuk membeli suatu barang.
6. Pada aplikasi *mobile* manajemen keuangan ini memiliki tingkat *usability* sebesar 79. Hal ini dapat dikatakan bahwa aplikasi ini dapat digunakan dengan baik atau dapat diterima bagi pengguna karena sudah melebihi nilai minimal *usability*, yaitu 68.

7.2 Saran

Terdapat beberapa saran untuk pengembangan sistem selanjutnya, yaitu:

1. Berdasarkan hasil pengujian *usability*, nilai *usability* yang didapat adalah 79. Dengan penyesuaian *user experience* dalam pembuatan aplikasi yang didasari oleh permasalahan kerumitan penggunaan aplikasi, kemudahan penggunaan aplikasi, kemungkinan bantuan dari orang teknis dalam penggunaan aplikasi, dan perlunya pembelajaran sebelum menggunakan aplikasi, maka memungkinkan terjadinya peningkatan pada nilai *usability*.
2. Berdasarkan data pengujian algoritma Naïve Bayes, pada pengujian menggunakan data *training* sebanyak 8 data yang diuji dengan 8 kasus uji dan memiliki nilai akurasi sebesar 100%. Untuk menguji nilai akurasi apakah dapat konsisten pada 100% atau tidak, maka memungkinkan untuk menambahkan data *training* yang lebih banyak lagi.



DAFTAR PUSTAKA

- Andriyanto, I., Santoso, E., & Suprpto. (2018). Pemodelan Sistem Pakar Untuk Menentukan Penyakit Diabetes Mellitus Menggunakan Metode Naive Bayes Studi Kasus: Puskesmas Poncokusumo Malang. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 880-887.
- Bangor, A., Kortum, P., & Miller, J. (2009). Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. *Journal Of Usability Studies*, 114-123.
- Dennis, A., Wixom, B., & Roth, R. (2012). *System Analysis and Design 5th ed*. New Jersey: John Wiley & Sons, Inc.
- Destuardi, & Sumpeno, S. (2009). Klasifikasi Emosi Untuk Teks Bahasa Indonesia Menggunakan Metode Naive Bayes. *Seminar Nasional Pascasarjana*, 3.
- Febriani, & Fauzi, A. (2015). Aplikasi Sistem Informasi Keuangan Mahasiswa Pada Telepon Pintar. 1-6.
- Finansialku. (2015). *Perencanaan Keuangan Untuk Usia 20an (Kelola Keuangan dengan Benar sejak Muda)*. Finansialku.
- Firebase. (2018, September 11). *Firestore Authentication*. Diambil kembali dari Firebase: <https://firebase.google.com/docs/auth/?hl=ID>
- Firebase. (2018, Maret 23). *Firestore Realtime Database*. Diambil kembali dari Firebase: https://firebase.google.com/docs/database/?hl=ID#key_capabilities
- Fowler, M. (2003). *UML Distilled: A Brief Guide to the Standard Object Modeling Language 3rd Edition*. Addison - Wesley Professional.
- Gitman, L. J. (2002). *Principles of Managerial Finance (Tenth Edition)*. Addison Wesley.
- Husnan, S. (2014). *Manajemen Keuangan (2nd ed.)*. Tangerang Selatan: Universitas Terbuka.
- Nielsen, J. (2000, Maret 19). *Articles: Why You Only Need to Test with 5 Users*. Diambil kembali dari Nielsen Norman Group: <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>
- Nielsen, J. (2012, Januari 4). *Usability 101: Introduction to Usability*. Diambil kembali dari Nielsen Norman Group: <https://www.nngroup.com/articles/usability-101-introduction-to-usability/>
- Nurani, A., Susanto, B., & Proboyekti, U. (2011). Implementasi Naive Bayes Classifier Pada Program Bantu Penentuan Buku Referensi Matakuliah. *Jurnal Informatika*.
- Pressman, R. (2010). *Software Engineering : A Practitioner's Approach 7th Edition*. New York: McGraw - Hill.
- Pudjoatmodjo, B., & Wijaya, R. (2016). Tes Kegunaan (Usability Testing) Pada Aplikasi Kepegawaian Dengan Menggunakan System Usability Scale (Studi Kasus: Dinas Pertanian Kabupaten Bandung). *Seminar Nasional Teknologi Informasi dan Multimedia 2016*.

- Saleh, A. (2015). Implementasi Metode Klasifikasi Naïve Bayes Dalam Memprediksi Besarnya Penggunaan Listrik Rumah Tangga. 207-217.
- Sauro, J. (2011, February 2). *MEASURING USABILITY WITH THE SYSTEM USABILITY SCALE (SUS)*. Diambil kembali dari <https://measuringu.com/sus/>
- Schwaber, K., & Sutherland, J. (2016). *The Definitive Guide to Scrum: The Rules of the Game*. 1-11.
- Seidl, M., & dkk. (2015). *UML @ Classroom: An Introduction to Object - Oriented Modeling*. Switzerland.
- Susanto, A., Noertjahyana, A., & Setiawan, A. (2016). *Aplikasi Pengelola Keuangan Pribai Berbasis Android*. 1-4.
- Szalvay, V. (2004). *An Introduction to Agile Software Development*. Danube Technologies, Inc.
- Tutorialspoin. (2018, Maret 22). *SDLC - Overview*. Diambil kembali dari https://www.tutorialspoint.com/sdlc/sdlc_overview.htm
- Visual Paradigm. (2018, Januari 19). *What is Sequence Diagram?* Diambil kembali dari Visual Paradigm: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>
- Wijaya, T. (2018). Penerapan Metode Scrum Dan Virtual Private Network Dalam Perancangan Sistem Ordersales. 115-125.
- Wiley, J., & Sons. (2005). *Object - Oriented Analysis & Design: Understanding System Development with UML 2.0*. England: British Library Cataloguing.