

Pengembangan Aplikasi Mobile Manajemen Keuangan Dengan Metode Scrum (Studi Kasus Mahasiswa FILKOM UB)

Januari Arka¹, Adam Hendra Brata², Komang Candra Brata³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹january.arka@gmail.com, ²adam@ub.ac.id, ³k.candra.brata@ub.ac.id

Abstrak

FM App adalah aplikasi manajemen keuangan untuk memecahkan masalah keuangan bagi mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya (FILKOM UB). FM App dapat digunakan dimana saja dan kapan saja karena dikembangkan dengan Android *mobile platform* yang sebagian besar merupakan jenis *smartphone* yang digunakan mahasiswa dan Firebase untuk menyimpan data secara *real time*. 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, biaya tak terduga, dan perencanaan pembelian. 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. FM App memiliki nilai akurasi rekomendasi 100%, sehingga dapat disimpulkan bahwa aplikasi dapat merekomendasikan status keuangan dengan 100% benar setelah pengeluaran tak terduga dibuat. Disamping itu, FM App memiliki nilai 79 pada *usability*, sehingga dapat dikategorikan bahwa aplikasi memiliki tingkat kemudahan penggunaan yang baik.

Kata kunci: manajemen keuangan, Android, Firebase, Naive 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 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 real time. FM application 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. 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. The FM application has a 100% recommendation accuracy value, so it can be concluded that the application can recommend financial status with 100% correct after unexpected expenses are made. Besides that, FM App has a value of 79 on usability, so it can be categorized that the application has a good level of ease of use.

Keywords: financial management, Android, Firebase, Naïve Bayes, Scrum

1. PENDAHULUAN

Manajemen keuangan merupakan suatu kegiatan yang didalamnya melakukan perencanaan, analisis, dan pengendalian keuangan untuk mencapai suatu tujuan dalam memaksimalkan penggunaan uangnya (Husnan,

2014). 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, 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 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, 2012). Hal ini juga didasari dengan hasil survei yang menyatakan bahwa 96,7% responden menginginkan sistem yang mudah untuk digunakan.

Dalam pengembangan 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).

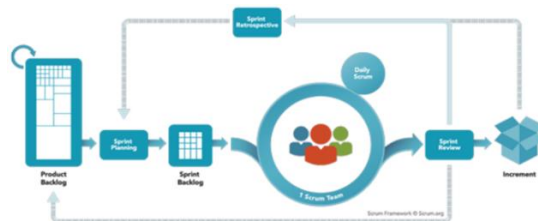
Berdasarkan penjelasan latar belakang diatas, maka dapat dirumuskan menjadi beberapa rumusan masalah. Pertama, bagaimana cara memastikan pengguna ikut berperan langsung dalam perancangan sistem dan mempercepat pengembangan sistem. Kedua, bagaimana hasil rekayasa kebutuhan dalam pengembangan aplikasi *mobile* manajemen keuangan. Ketiga, bagaimana cara mengatasi pengeluaran yang melewati batas harian. Keempat, bagaimana cara mengatasi pengeluaran tidak terduga yang sebelumnya tidak dianggarkan. Kelima, bagaimana cara mengatasi pembelian suatu barang. Keenam, bagaimana tingkat kemudahan penggunaan pada aplikasi *mobile* manajemen keuangan mahasiswa FILKOM Universitas Brawijaya.

Selain itu, penelitian ini juga memiliki beberapa tujuan. Pertama, memastikan pengguna ikut berperan langsung dalam perancangan sistem dan mempercepat pengembangan sistem. Kedua, menjadikan aplikasi *mobile* manajemen keuangan sebagai solusi untuk permasalahan manajemen keuangan mahasiswa FILKOM UB. Ketiga, memberikan

rekomendasi status keuangan mahasiswa FILKOM UB ketika pengeluaran tidak terduga telah terjadi. Keempat, memudahkan pengguna aplikasi *mobile* manajemen keuangan oleh mahasiswa FILKOM UB..

Manfaat dilakukannya penelitian ini, yaitu mahasiswa FILKOM UB diharapkan dapat menjadikan aplikasi *mobile* manajemen keuangan sebagai solusi permasalahan keuangannya dan dapat mengatasi permasalahan pengeluaran tidak terduga dengan kemampuan keuangan yang dimilikinya sekarang. Selain itu, mahasiswa FILKOM UB diharapkan dapat mengatasi permasalahan pembelian suatu barang dengan kemampuan keuangan yang dimilikinya sekarang dan dapat mencatat keuangan pribadinya dimanapun dan kapanpun.

Agar penelitian terfokus dan tidak terjadi pelebaran topik permasalahan, maka dibatasi dalam beberapa hal. Hal tersebut diantaranya yaitu penelitian ini berfokus pada pengeluaran, pemasukan, dan perencanaan keuangan pribadi. Ruang lingkup penelitian ini dilakukan di FILKOM UB. Selain itu, aplikasi dibuat pada perangkat *mobile* Android dan aplikasi hanya mampu berjalan pada *mobile* dengan sistem operasi Android 4.0.3 keatas.



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

Gambar 1. *Scrum Sprint Life Cycle*

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. Oleh karena itu, dalam proses scrum, rencana proyek terus diperiksa dan disesuaikan berdasarkan kenyataan empiris proyek (Szalvay, 2004). Terdapat beberapa tahapan yang dilakukan selama proses Scrum dilakukan, yaitu *product backlog*, *sprint planning*, dan *sprint* (Schwaber & Sutherland, The Definitive Guide to Scrum: The Rules of the

Game, 2016). Untuk lebih jelasnya dapat dilihat pada gambar 1.

Naïve Bayes adalah metode klasifikasi yang berasal dari penerapan teorema Bayes dengan asumsi *independence* (saling bebas). *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 1.1 yang merupakan persamaan metode Naïve Bayes (Saleh, 2015).

$$P(C|X) = \frac{P(X|C).P(C)}{P(X)} \quad (1.1)$$

Keterangan dari persamaan (1.1), yaitu:

X : Data dengan kelas yang belum diketahui

C : Hipotesis data merupakan kelas spesifik

$P(C|X)$: Probabilitas hipotesis C berdasarkan 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*)

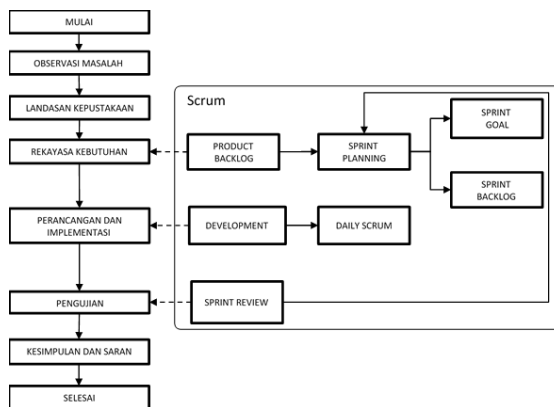
Akurasi merupakan nilai hasil pengukuran kuantitas untuk nilai kebenaran (Andriyanto, Santoso, & Suprpto, 2018). Pengukuran ini dilakukan untuk mengetahui nilai akurasi yang dihasilkan terhadap rekomendasi yang diberikan oleh algoritma Naïve Bayes. Nilai akurasi didapatkan dari hasil jumlah kasus uji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

$$Akurasi = \frac{Jumlah\ Prediksi\ Benar}{Jumlah\ Total\ Prediksi} \times 100\% \quad (1.2)$$

2. METODOLOGI PENELITIAN

Dalam pengembangannya, penelitian ini menggunakan metode Scrum. Scrum dipilih karena dari kebutuhan yang sudah dijabarkan pada pendahuluan membutuhkan pendekatan berulang yang iteratif yang membutuhkan

feedback secara berlanjut kepada pengguna dan dalam pengembangan sistem juga dibutuhkan penyelesaian yang cepat karena adanya *daily scrum* yang memungkinkan produktifitas dari pengembang meningkat. Untuk lebih jelasnya mengenai Scrum, maka dapat dilihat pada gambar 2.



Gambar 2. Diagram Alir Metodologi Penelitian

Pada gambar 2 dapat dijelaskan bahwa 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 implementasi 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 dikejakan. Jika masih ada, maka akan dilakukan *sprint* lagi dan jika tidak ada, maka dilakukan penarikan kesimpulan dan saran terkait penelitian ini.

3. REKAYASA KEBUTUHAN

3.1 Gambaran Umum Aplikasi

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.

3.2 Identifikasi Aktor

Identifikasi Aktor bertujuan untuk memberikan penjelasan siapa saja yang berperan pada FM App. Pada tabel 1 akan dijelaskan aktor yang terkait dan deskripsi dari aktor tersebut.

Tabel 1. 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.

3.3 Daftar Kebutuhan

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 Specification*, 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 2 berikut.

Tabel 2. Daftar Kebutuhan Fungsional

No	Kode Fungsi	Definisi Kebutuhan	Kode <i>Product Backlog</i>
1	SRS_FM_F_01	Pengguna dapat <i>sign up</i> untuk	PB_01

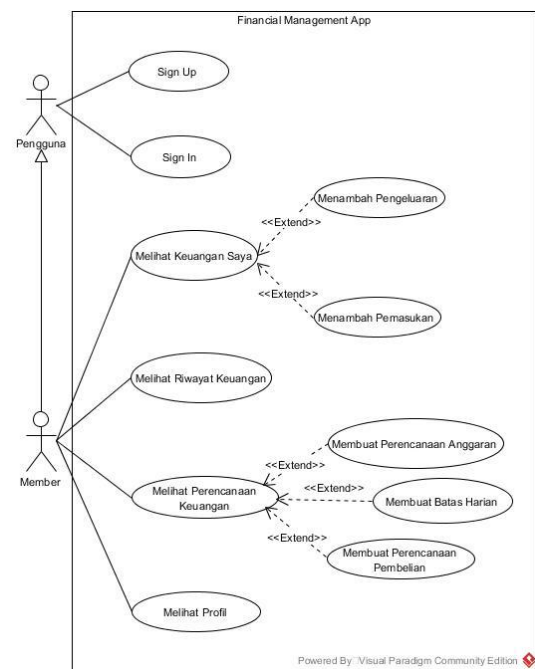
		mendaftarkan diri sebagai member	
2	SRS_FM_F_02	Pengguna dapat <i>sign in</i> untuk masuk kedalam sistem	PB_02
3	SRS_FM_F_03	Member dapat melihat informasi keuangannya	PB_03
4	SRS_FM_F_04	Member dapat mencatat pengeluaran keuangan ke dalam sistem	PB_04
5	SRS_FM_F_05	Member dapat mencatat pemasukan keuangan ke dalam sistem	PB_05
6	SRS_FM_F_06	Member dapat melihat riwayat keuangan berdasarkan tanggal pada sistem	PB_06
7	SRS_FM_F_07	Member dapat memilih daftar perencanaan keuangan pada sistem	PB_07
8	SRS_FM_F_08	Member dapat membuat perencanaan anggaran keuangan pada sistem	PB_08
9	SRS_FM_F_09	Member dapat membuat batas harian keuangan pada sistem	PB_09
10	SRS_FM_F_10	Member dapat membuat perencanaan pembelian pada sistem	PB_10
11	SRS_FM_F_11	Member dapat melihat informasi pengguna	PB_11

Selain itu, didapat juga 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 3 berikut

Tabel 3. Kebutuhan Non Fungsional

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

3.4 Use Case Diagram



Gambar 3. Use Case Diagram FM App

Pada gambar 3 merupakan gambar *use case diagram* FM App. Gambar 3 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.

4. PERANCANGAN DAN IMPLEMENTASI

4.1 Perancangan

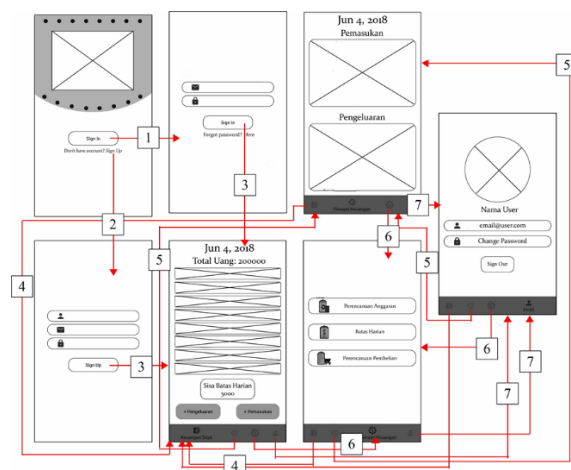
4.1.1 Perancangan Arsitektural

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*.

4.1.2 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*.

4.1.2.1 Screenflow



Gambar 4. Screenflow Umum FM App

Screenflow umum ini menjelaskan tampilan utama yang dapat diakses pada FM App. Pada screenflow ini terdapat 7 bagian. Untuk lebih jelasnya dapat dilihat pada gambar 4.

Untuk menjelaskan screenflow pada gambar 4, maka dapat dilihat pada tabel 4.

Tabel 4. Penjelasan Screenflow 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

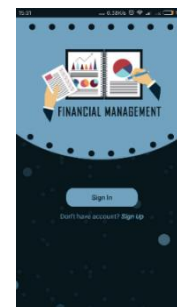
4	Merupakan cara untuk menampilkan halaman keuangan saya
5	Merupakan cara untuk menampilkan halaman riwayat keuangan
6	Merupakan cara untuk menampilkan halaman perencanaan keuangan
7	Merupakan cara untuk menampilkan halaman profil

4.2 Implementasi

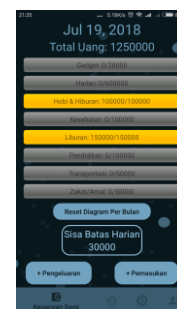
4.2.1 Implementasi Kode Program

Proses implementasi FM App menggunakan bahasa pemrograman Java untuk *back end* code 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.

4.2.2 Implementasi Antarmuka



Gambar 5. Tampilan Halaman Utama



Gambar 6. Tampilan Keuangan Saya

Terdapat 16 antarmuka pada FM App. Ke enam belas antarmuka tersebut adalah tampilan utama, halaman *sign up*, halaman *sign in*, tampilan *pop up menu forgot password*, halaman keuangan saya, halaman menambah pengeluaran, halaman menambah pemasukan, halaman riwayat keuangan, halaman

perencanaan keuangan, halaman perencanaan anggaran, halaman keterangan kategori anggaran, halaman batas harian, halaman perencanaan pembelian, halaman pembelian berdasarkan jumlah tabungan, halaman pembelian berdasarkan tanggal pembelian, halaman profil. Pada gambar 5 dan 6 merupakan salah satu contoh tampilan yang sudah diimplementasikan.

5. Pengujian

5.1 Pengujian Algoritma Naïve Bayes

Tujuan dilakukan pengujian ini adalah untuk mengetahui status keuangan selanjutnya ketika perencanaan alokasi anggaran pengeluaran tidak dianggarkan diterapkan. Pengujian dilakukan sebanyak 8 pengujian, namun pada jurnal ini hanya dijelaskan satu kasus uji, yaitu kasus uji pertama. 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". Untuk perhitungannya dapat dilihat pada sub bab berikut.

5.1.1 Probabilitas Prior (P(C))

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

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

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

5.1.2 Probabilitas Likelihood (P(X|C))

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

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

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

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

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

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

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

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

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

$$\begin{aligned} P(X|\text{Status} = \text{"Aman"}) \\ &= P\left(\begin{array}{l} \text{Tersier} = \text{"Ya"}, \\ \text{Sekunder} = \text{"Tidak"}, \\ \text{Primer} = \text{"Tidak"} \end{array} \middle| \begin{array}{l} \text{Status} = \\ \text{"Aman"} \end{array}\right) \\ &= \frac{3}{3} \times \frac{3}{3} \times \frac{3}{3} = 1 \end{aligned}$$

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

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

5.1.3 Probabilitas Posterior (P(C|X))

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

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

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

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**.

5.1.4 Pengujian Akurasi

Berdasarkan hasil pengujian algoritma Naïve Bayes, maka akan dihitung akurasi dari hasil rekomendasi tersebut. Perhitungan ini didasari dari 8 kasus uji yang seluruhnya sesuai dengan hasil yang diharapkan. 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%.

5.2 Pengujian Usability

Pengujian *Usability* dilakukan untuk mengetahui tingkat kemudahan penggunaan terhadap sistem yang telah diimplementasikan dan diuji secara langsung kepada sejumlah responden. Dalam pengujian *usability* akan digunakan metode *System Usability Scale* (SUS). *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 5 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 (Pudjoatmodjo & Wijaya, 2016). Untuk 10 pertanyaan tersebut adalah sebagai berikut:

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

Untuk lebih jelasnya mengenai pengujian *usability* dapat dilihat pada tabel 5 dan 6.

Tabel 5. 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

Tabel 6. 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
												8

Berdasarkan tabel 5 dan 6, maka dapat dihitung nilai *usability* sebagai berikut:

$$Usability = \frac{\sum Total Skor}{\sum Responden} \times 2,5$$

$$Usability = \frac{158}{5} \times 2,5 = 79$$

6. KESIMPULAN

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.

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.

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. Selain itu, 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. Adanya fitur perencanaan pembelian berdasarkan jumlah menabung atau penentuan tanggal pembelian pada aplikasi, juga menjadikan solusi untuk membeli suatu barang.

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.

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.

Destuardi, & Sumpeno, S. (2009). Klasifikasi Emosi Untuk Teks Bahasa Indonesia Menggunakan Metode Naive Bayes. *Seminar Nasional Pascasarjana*, 3.

Husnan, S. (2014). *Manajemen Keuangan* (2nd ed.). Tangerang Selatan: Universitas Terbuka.

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/>

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.

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