

**PERANCANGAN *PLATFORM* SIMPAN PINJAM DAN INVESTASI
LEMBAGA KEUANGAN MIKRO SYARIAH DENGAN METODE *RAPID*
APPLICATION DEVELOPMENT
(STUDI KASUS BAITUL MAAL WA TAMWIL FEB UNPAD)**

SKRIPSI

Diajukan untuk menempuh ujian sarjana
pada Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Padjadjaran

**FIDRIYANTO RIZKILLAH
NPM 140810160043**



**UNIVERSITAS PADJADJARAN
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
PROGRAM STUDI TEKNIK INFORMATIKA
JATINANGOR
2019**

ABSTRAK

Baitul Maal wa Tamwil adalah lembaga keuangan mikro yang dioperasikan dengan prinsip syari'ah. Kurangnya fasilitas berbasis teknologi informasi yang sebenarnya dapat menunjang pengalaman nasabah merupakan faktor-faktor yang menjadi problematika operasional BMT di Indonesia. Keharusan nasabah untuk datang ke sekretariat BMT ketika ingin melakukan registrasi atau transaksi merupakan salah satu hal yang menjadi sorotan problematika operasional BMT. Sebuah aplikasi ponsel merupakan solusi yang dapat diberikan sehingga para nasabah dari BMT dapat melakukan berbagai transaksi tanpa harus datang ke sekretariat BMT. Aplikasi dirancang menggunakan framework android native Java dan Laravel untuk API, karena kedua framework ini memiliki dokumentasi yang jelas serta komunitas yang luas sehingga mempermudah developer untuk membangun aplikasi. Metode pengembangan yang digunakan adalah *Rapid Application Development*, karena metode ini dapat mengurangi kebutuhan sumber daya serta dapat membantu pengembangan aplikasi dengan waktu yang singkat. Pengujian usabilitas dengan skala likert dilakukan kepada nasabah BMT FEB Unpad atau calo pengguna dengan hasil **<hasil>**. Dari hasil tersebut dapat disimpulkan bahwa sistem tersebut berjalan dengan baik dan memberikan manfaat baik untuk nasabah BMT.

Kata Kunci : BMT (Baitul Maal wa Tamwil), Java, Android, PHP, Laravel.

ABSTRACT

*Baitul Maal wa Tamwil is a microfinance institution operated with sharia principles. The lack of an IT-based facility that actually can support the customer's experience are factors that have become operational problems of BMT in Indonesia. The requirement for customers to come to the BMT secretariat every time they want to do any transaction or registration is one of the highlights of BMT's operational problems. A mobile application can be a solution for the customer so they can do any transaction without the must to come to the BMT secretariat. The application developed by using the android native Java framework and Laravel for its API because of these two frameworks has great documentation and a large community that can help the developer while building the application. The development methodology used is Rapid Application Development because this method can reduce the resource requirements and can help develop the application in a short time. Usability testing with a Likert scale is carried out to BMT FEB Unpad customers or touts of users with the results **<results>**. From these results, it can be concluded that the system runs well and provides good benefits for BMT customers.*

Keywords : BMT (Baitul Maal wa Tamwil), Java, Android, PHP, Laravel.

KATA PENGANTAR

Bismillahirrahmanirrahim. Dengan menyebut nama Allah yang Maha Pengasih dan Maha Penyayang

Segala puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa yang senantiasa selalu memberikan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul “**PERANCANGAN *PLATFORM* SIMPAN PINJAM DAN INVESTASI LEMBAGA KEUANGAN MIKRO SYARIAH DENGAN METODE *RAPID APPLICATION DEVELOPMENT* (STUDI KASUS BAITUL MAAL WA TAMWIL FEB UNPAD)**”, sebagai salah satu syarat dalam menempuh sarjana pada Program Studi Strata Satu (S1) Teknik Informatika Departemen Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Padjadjaran.

Dalam proses perjalanan menyusun skripsi ini, penulis mendapatkan banyak sekali bantuan dan bimbingan dari berbagai pihak. Penulis mengucapkan terima kasih yang sebanyak-banyaknya kepada orang tua, kakak dan keluarga besar tercinta atas dukungan doa, motivasi, serta bantuan moral dan materil yang diberikan. Ucapan terima kasih yang besar juga diucapkan kepada Bapak Deni Setiana, S.Si., M.CS selaku dosen wali sekaligus dosen pembimbing pendamping dan Bapak Dr. Asep Sholahuddin, MT. selaku dosen pembimbing utama penulis yang telah meluangkan waktunya untuk memberikan arahan, masukan, motivasi serta tekanan yang sangat bermanfaat bagi penulis.

Tidak lupa penulis juga mengucapkan terima kasih kepada:

1. Dr. Setiawan Hadi, M.Sc.CS selaku Kepala Departemen Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Padjadjaran.
2. Dr. Juli Rejito, M.Kom selaku Ketua Program Studi S-1 Teknik Informatika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Padjadjaran.
3. Segenap dosen yang akan menjadi dosen penguji sidang sarjana penulis

4. Seluruh staff pengajar dan tata usaha Departemen Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Padjadjaran yang juga telah membantu penulis dalam menyelesaikan proses penulisan skripsi.
5. Teman saya Khansa Fathiyah yang sudah menjadi narasumber dari objek penelitian dari skripsi ini.
6. Teman-teman saya Baby Cattleya, Achun Armando dan Bariq Mbani yang sudah mengajari saya banyak hal guna perancangan program pada skripsi ini.
7. Teman-teman saya Dzakia Rayhana, Vega Savera Yuana, Reynaldi Noer Rizki dan Afifah Kho'eriah yang telah memberikan dukungan moral serta motivasi kepada saya dalam pengerjaan skripsi ini.
8. Segenap keluarga JAMET CLUB yang senantiasa menemani, memberikan dukungan baik secara moral, ilmu dan tekanan yang bermanfaat bagi penulis di dalam suka-duka kehidupan perkuliahan.
9. Teman-teman seperjuangan Cyber 2016 yang telah (Teknik Informatika FMIPA Unpad angkatan 2016) yang bersama-sama menempuh perkuliahan sejak pertama kali menapakkan kaki di gedung PPBS D.
10. Teman-teman terdekat penulis yang tidak bisa dituliskan satu per satu yang telah memberikan dukungan, doa dan semangat yang sangat berarti bagi penulis.

Semoga Tuhan membalas kebaikan mereka dengan melimpahkan mereka segala berkat dan rahmat-Nya. Semoga skripsi ini dapat memberikan ilmu dan motivasi yang bermanfaat bagi pembaca.

Jatinangor, Februari 2020

Penulis

DAFTAR ISI

ABSTRAK	i
ABSTRACT	ii
KATA PENGANTAR	iii
DAFTAR ISI	v
DAFTAR GAMBAR	ix
DAFTAR LAMPIRAN	xi
BAB I	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	3
1.3 Batasan Masalah	3
1.4 Maksud dan Tujuan Penelitian	4
1.5 Manfaat Penelitian	4
1.6 Metodologi Penelitian	4
1.7 Sistematika Penelitian	5
BAB II	7
2.1 Deskripsi Organisasi (Baitul Maal wa Tamwil FEB Unpad)	7
2.2 UML	8
2.3 Pengembangan Aplikasi Berbasis Android	11
2.3.1 Android	11
2.3.2 Java	12
2.3.3 XML	12
2.3.4 Clean Architecture	12
2.3.5 Model View Presenter	13

2.4	API	13
2.5	Laravel	13
2.6	MySQL.....	14
2.7	Metode <i>Rapid Application Development</i>	14
BAB III.....		16
3.1	Desain Sistematisa Penelitian	16
3.2	Studi Literatur.....	17
3.3	Pengumpulan Data.....	17
3.4	Analisis Kebutuhan.....	17
3.4.1	Analisis Kebutuhan Pengguna.....	17
3.4.2	Analisis Kebutuhan Perangkat Lunak.....	18
3.4.3	Analisis Kebutuhan Perangkat Keras	18
3.4.4	Analisis Kebutuhan Fungsional.....	19
3.5	Perencanaan Proses dan GUI	20
3.5.1	Proses RAD (<i>Rapid Application Development</i>)	20
3.5.2	Perancangan Proses	21
3.5.3	Perencanaan <i>Database</i>	26
3.5.4	Struktur Tabel.....	27
3.5.5	Perancangan GUI.....	34
3.6	Perancangan Pengujian Sistem.....	40
3.6.1	Pengujian Oleh Nasabah	41
BAB IV		43
4.1	Instalasi dan Konfigurasi Laravel.....	43
4.2	Konfigurasi dan Migrasi Database	44
4.2.1	Migrasi Tabel User	45

4.2.2	Migrasi Tabel Nasabah Smart Saving	45
4.2.3	Migrasi Tabel Nasabah Custom Saving	46
4.2.4	Migrasi Tabel Pengadaan	46
4.2.5	Migrasi Tabel Qardh	47
4.2.6	Migrasi Tabel Transaksi Smart	48
4.2.7	Migrasi Tabel Transaksi Custom	48
4.2.8	Migrasi Tabel Angsur Pengadaan	48
4.2.9	Migrasi Tabel Angsur Qardh	49
4.3	Penulisan Kode Pemrograman untuk Model	49
4.3.1	Model User	49
4.3.2	Model Smart Saving	50
4.3.3	Model Custom Saving	50
4.3.4	Model Transaksi Smart	51
4.3.5	Model Transaksi Custom	51
4.3.6	Model Pengadaan	51
4.3.7	Model Angsur Pengadaan	52
4.3.8	Model Angsur Qardh	52
4.4	Implementasi Controller	53
4.5	Implementasi Route	61
4.6	Implementasi Tampilan	61
4.6.1	Tampilan Halaman Login	62
4.6.2	Tampilan halaman register	62
4.6.3	Tampilan halaman utama	63
4.6.4	Tampilan halaman produk	63
4.6.5	Tampilan halaman riwayat	64

4.6.6	Tampilan halaman jenis transaksi	65
4.6.7	Tampilan halaman transaksi.....	65
4.6.8	Tampilan halaman profil.....	66
4.6.9	Tampilan halaman FAQ.....	66
4.6.10	Tampilan halaman about.....	67
4.7	Pengujian Aplikasi.....	67
4.7.1	Pengujian Usabilitas Kepada Nasabah	67
BAB V		72
5.1	Simpulan	72
5.2	Saran	72
DAFTAR PUSTAKA		73

DAFTAR GAMBAR

Gambar 2.1 The Clean Architecture (Martin, 2012)	13
Gambar 2.2 Tahapan RAD dengan system prototyping	14
Gambar 3.1 Sistematika Penelitian	16
Gambar 3.2 Metode RAD	21
Gambar 3.3 Use Case Diagram	22
Gambar 3.4 Activity Diagram halaman utama	23
Gambar 3.5 Activity Diagram produk	23
Gambar 3.6 Activity Diagram Form Nasabah.....	24
Gambar 3.7 Activity Diagram transaksi	24
Gambar 3.8 Skema Perancangan Database	26
Gambar 3.9 Halaman Register dan Login.....	35
Gambar 3.11 Halaman Utama	36
Gambar 3.12 Halaman Profil.....	36
Gambar 3.13 Halaman Produk	37
Gambar 3.14 Halaman Form Nasabah	38
Gambar 3.15 Halaman Transaksi	39
Gambar 3.16 Halaman FAQ.....	39
Gambar 3.17 Halaman About.....	40
Gambar 4.1 Tampilan Utama Laravel	44
Gambar 4.2 Tampilan halaman login.....	62
Gambar 4.3 Tampilan halaman register	63
Gambar 4.4 Tampilan halaman utama	63
Gambar 4.5 Tampilan halaman produk.....	64
Gambar 4.6 Tampilan halaman riwayat	64
Gambar 4.7 Tampilan halaman jenis transaksi.....	65
Gambar 4.8 Tampilan halaman transaksi.....	65
Gambar 4.9 Tampilan halaman profil	66
Gambar 4.10 Tampilan Halaman FAQ	66
Gambar 4.11 Tampilan Halaman About	67

DAFTAR TABEL

Tabel 2.1 Use Case Diagram	9
Tabel 2.2 Tabel Class Diagram	10
Tabel 2.3 Tabel Activity Diagram	11
Tabel 3.1 Kebutuhan Funsgional Pengguna	19
Tabel 3.2 Atribut user	27
Tabel 3.3 Atribut nasabah smart	28
Tabel 3.4 Atribut nasabah custom	29
Tabel 3.5 Atribut qardh	30
Tabel 3.6 Atribut pengadaan	31
Tabel 3.7 Atribut transaksi smart	32
Tabel 3.8 Atribut transaksi_custom	33
Tabel 3.9 Atribut angsur_pengadaan	33
Tabel 3.10 Atribut angsur qardh	34
Tabel 3.11 Penilaian Skala Likert	41
Tabel 4.1 Hasil Penilaian Kuisisioner Pengguna	68
Tabel 4.2 Kriteria Nilai	69
Tabel 4.3 Hasil Pengujian	70
Tabel 4.4 Tabel Kesimpulan Pengujian	71

DAFTAR LAMPIRAN

BAB I

PENDAHULUAN

Pada bab ini berisi tentang latar belakang penelitian, identifikasi masalah, batasan masalah, maksud dan tujuan penelitian, manfaat penelitian, metodologi penelitian, dan juga sistematika penulisan.

1.1 Latar Belakang

Teknologi merupakan salah satu hasil dari temuan manusia yang dikembangkan oleh ilmu pengetahuan dari zaman dahulu hingga kini. Teknologi serta perkembangannya yang sangat pesat dipastikan memiliki peran yang sangat penting dalam menunjang seluruh kegiatan manusia. Kegiatan operasional dalam suatu perusahaan atau organisasi seperti menyimpan data dan bertukar informasi merupakan contoh dari kegiatan manusia yang selalu didukung oleh teknologi. Kegiatan-kegiatan ini dapat dipermudah dengan menggunakan teknologi yang tersedia dalam bentuk komputer atau *gadget*.

BMT adalah lembaga keuangan syariah yang beroperasi menggunakan gabungan konsep “Baitul Maal” dan Baitul Tamwil” dengan target operasionalnya fokus kepada sektor Usaha Kecil Menengah (UKM). Konsep Baitul maal berarti BMT berperan sebagai lembaga sosial keagamaan. Sedangkan pada konsep Baitul Tamwil, BMT mempunyai peranan sebagai lembaga bisnis maupun lembaga keuangan yang bertujuan untuk mencari keuntungan. Namun jika kita lihat prakteknya di lapangan, BMT lebih cenderung berperan sebagai lembaga keuangan syariah yang menawarkan produk simpan pinjam ke masyarakat yang berlandaskan pada prinsip-prinsip syariah. BMT mulai berkembang dengan pesat sebagai alternatif lembaga keuangan bagi mereka yang mempunyai pendapatan rendah (Rusby, Hamzah, & Hamzah, 2018).

Pesatnya perkembangan BMT merupakan penanda bahwa BMT mendapatkan penghargaan yang tinggi dari masyarakat. Namun karena disikapi dengan kurang bijaksana, tidak semua BMT yang sudah berdiri dapat bertahan

karena mengalami kebangkrutan. Kurangnya fasilitas berbasis teknologi informasi yang sebenarnya dapat menunjang pengalaman nasabah merupakan faktor-faktor yang menjadi problematika operasional BMT di Indonesia. Bahkan teknologi informasi merupakan salah satu faktor penentu utama keberhasilan BMT (Sa'roni & Hosen, 2012).

Dilihat dari peningkatan transaksi setiap bulannya, BMT FEB Unpad dapat dikatakan cukup mendapatkan perhatian yang tinggi dari para nasabah-nya. Apabila hal ini tidak dibersamai dengan peningkatan serta perkembangan dari segala sumber daya yang terlibat, maka perhatian yang sudah didapat ini dapat menjadi *boomerang* bagi keberlangsungan BMT FEB Unpad dalam jangka waktu yang cukup panjang kedepannya. Kasus yang sudah terjadi di lapangan seperti bantroknya jadwal jaga *teller* untuk menjaga sekretariat dari yang satu hari memiliki *shift* pagi dan siang lalu menjadi satu *shift* hingga tidak ada yang menjaga sekretariat BMT. Hal ini pasti akan berpengaruh besar terhadap nasabah yang ingin melakukan transaksi namun tidak ada *teller* yang menjaga di sekretariat BMT. Tidak hanya terjadi pada *teller*, hal-hal yang bisa mengancam keberlangsungan BMT juga terjadi pada nasabah seperti malas-nya nasabah yang diharuskan mendatangi sekretariat untuk melakukan segala macam transaksi. Bila dibiarkan, permasalahan ini dapat menyebabkan risiko besar yang dapat mengancam kelangsungan proses bisnis BMT FEB Unpad.

Dengan adanya aplikasi BMT FEB Unpad, diharapkan dapat meningkatkan pengalaman pengguna atau nasabah yang baik serta dapat membantu *teller* dalam melakukan pekerjaannya. Aplikasi ini dapat menjadi solusi dari sebagian besar permasalahan-permasalahan yang terjadi baik pada sisi nasabah maupun *teller* seperti, memudahkan cara calon nasabah dalam melakukan registrasi, memudahkan nasabah dalam melakukan segala transaksi, serta mengatasi jadwal jaga yang bantrok dari *teller* untuk menjaga ruang sekretaria. Aplikasi ini akan dibangun pada *platform* android sehingga para nasabah dan calon nasabah dapat melakukan transaksi dan registrasi hanya dalam genggam tangan.

Dengan adanya latar belakang tersebut peneliti tertarik untuk menyusun skripsi atau tugas akhir ini dengan judul “**PERANCANGAN PLATFORM SIMPAN PINJAM DAN INVESTASI LEMBAGA KEUANGAN MIKRO SYARIAH DENGAN METODE RAPID APPLICATION DEVELOPMENT (STUDI KASUS BAITUL MAAL WA TAMWIL FEB UNPAD)**”. Diharapkan dengan adanya penelitian ini, dapat membantu mengurangi tugas yang harus dikerjakan oleh *teller* BMT secara bersamaan dan memudahkan nasabah dalam melakukan transaksi. Tidak perlu lagi datang ke ruang sekre setiap ingin melakukan transaksi.

1.2 Identifikasi Masalah

Berdasarkan pada uraian latar belakang masalah, maka dapat dirumuskan masalah pada penelitian ini, yaitu :

1. Bagaimana cara membantu memudahkan nasabah BMT FEB Unpad dalam melakukan transaksi tanpa harus datang ke sekretariat?
2. Bagaimana cara mengatasi permasalahan kosongnya ruang sekretariat ketika jadwal jaga *teller* ada yang bentrok?

1.3 Batasan Masalah

Dari permasalahan yang telah disampaikan di atas, akan dilakukan pembatasan masalah pada penelitian ini sebagai berikut:

1. Metode penelitian ini menggunakan Metode *Rapid Application Development*
2. Data yang digunakan merupakan data operasional Baitul Maal wa Tamwil FEB Unpad.
3. Aplikasi yang dirancang berbasis *android* untuk nasabah..
4. Perangkat lunak yang digunakan untuk membangun aplikasi ini adalah Android Studio dengan bahasa Java, API menggunakan *framework* Laravel, dan *database management system* MySQL.

1.4 Maksud dan Tujuan Penelitian

Maksud dari penelitian ini adalah untuk membangun sebuah aplikasi *android* yang diharapkan bisa memudahkan nasabah BMT FEB Unpad dalam melakukan transaksi yang sebelumnya harus dilakukan secara langsung di sekretariat BMT FEB Unpad.

Tujuan yang ingin dicapai oleh penulis dari penelitian ini adalah:

1. Membuat aplikasi BMT FEB Unpad berbasis *android*.
2. Untuk mengembangkan hasil dari pembuatan aplikasi BMT FEB Unpad berbasis *android* menjadi Tugas Akhir.
3. Untuk perancangan dan bangun pada tugas akhir “PERANCANGAN *PLATFORM* SIMPAN PINJAM DAN INVESTASI LEMBAGA KEUANGAN MIKRO SYARIAH DENGAN METODE *RAPID APPLICATION DEVELOPMENT* (STUDI KASUS BAITUL MAAL WA TAMWIL FEB UNPAD)”.
4. Membangun suatu sistem agar nasabah dapat melakukan transaksi tanpa harus datang ke Sekretariat BMT FEB Unpad.

1.5 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah:

1. Membantu pihak yang bersangkutan yaitu nasabah BMT FEB Unpad dalam melakukan transaksi.
2. Mengatasi permasalahan jadwal jaga yang bentrok dari *teller* BMT FEB Unpad sehingga tidak mengganggu tahapan yang akan dilakukan ketika nasabah ingin bertransaksi.

1.6 Metodologi Penelitian

Metode yang digunakan dalam melakukan penelitian adalah sebagai berikut:

1. Tahap studi literatur dilakukan dengan mencari dan mempelajari informasi yang berhubungan dengan skripsi ini. Informasi didapatkan dari berbagai sumber seperti internet, buku, paper, jurnal atau skripsi-skripsi sebelumnya.
2. Pengumpulan data dari Baitul Maal wa Tamwil FEB Unpad dengan langsung mendatangi sekre terkait. .Mengumpulkan data-data yang diperlukan untuk penelitian dari Baitul Maal wa Tamwil FEB Unpad.
3. Analisis kebutuhan pengguna, analisis kebutuhan perangkat lunak dan analisis kebutuhan perangkat keras.
4. Merancang serta membangun aplikasi yang menjadi media implementasi dari penelitian.
5. Melakukan uji coba terhadap aplikasi yang telah dibangun.
6. Menuliskan hasil dari penelitian tersebut dalam bentuk sebuah skripsi.

1.7 Sistematika Penelitian

Untuk memberi gambaran yang jelas tentang penelitian ini, maka disusunlah sistematika penulisan yang berisi materi yang akan dibahas pada setiap bab. Sistematika dalam penulisan tugas akhir ini adalah sebagai berikut:

BAB I PENDAHULUAN

Pada bab ini dijelaskan tentang latar belakang diangkatnya topik penulisan skripsi, pokok permasalahan berupa identifikasi dan batasan masalah, tujuan dan manfaat yang diharapkan dari penulisan skripsi, metodologi yang digunakan serta sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Pada bab ini dijelaskan seluruh landasan teori yang berhubungan dengan penelitian, yaitu tentang profil dan latar belakang organisasi, metode yang digunakan, serta penjelasan teoritis mengenai bahasa pemrograman dan *framework* yang digunakan dalam proses pengimplementasian aplikasi yang dibangun, serta teori lainnya guna memahami permasalahan yang dibahas.

BAB III ANALISIS DAN PERANCANGAN

Pada bab ini dijelaskan tentang metode pengembangan aplikasi yang digunakan meliputi analisis kebutuhan sistem, perancangan aplikasi, *data flow diagram* yang ada pada sistem, model perancangan data dan rancangan antarmuka pengguna.

BAB IV IMPLEMENTASI DAN EVALUASI SISTEM

Pada bab ini dijelaskan tentang implementasi aplikasi yang telah dibangun, tampilan aplikasi dan pengujian dari aplikasi, serta hasil pengujian aplikasi terhadap responden.

BAB V KESIMPULAN DAN SARAN

Bab ini merupakan penutup yang berisi kesimpulan dan saran dari penelitian yang sudah dilakukan.

BAB II

TINJAUAN PUSTAKA

Bab ini berisi tentang seluruh landasan teori yang berhubungan dengan penelitian, yaitu tentang profil dan latar belakang organisasi, metode yang digunakan, serta penjelasan teoritis mengenai bahasa pemrograman dan *framework* yang digunakan dalam proses pengimplementasian aplikasi yang dibangun, serta teori lainnya guna memahami permasalahan yang dibahas.

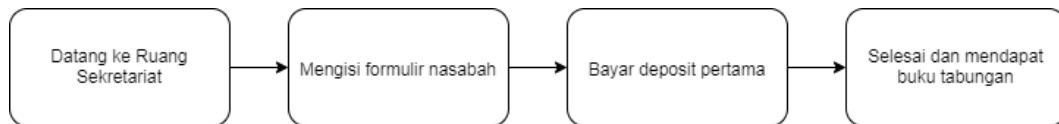
2.1 Deskripsi Organisasi (Baitul Maal wa Tamwil FEB Unpad)

BMT FEB Unpad adalah lembaga keuangan mikro syariah yang melayani mahasiswa dan civitas akademika FEB Unpad. BMT FEB Unpad berperan sebagai lembaga sosial keagamaan yang mempunyai fungsi untuk menerima sedekah dan menyalurkan kepada yang berhak menerima dana tersebut, serta sebagai lembaga bisnis maupun lembaga keuangan yang bertujuan untuk mencari keuntungan seperti menawarkan produk simpan pinjam ke masyarakat. Di BMT FEB Unpad memiliki berbagai macam transaksi yang dapat dilakukan yaitu simpanan, pengadaan dan donasi. Simpanan adalah proses menyimpan tabungan yang dilakukan oleh nasabah. Dalam transaksi simpanan terdapat dua jenis pelayanan, yaitu tabungan biasa dan tabungan investasi. Dimana tabungan investasi dapat mengalami keuntungan atau kerugian, tabungan ini juga hanya dapat diambil pada waktu yang sudah ditentukan pada rapat umum nasabah. Pengadaan adalah transaksi pinjaman yang dapat dilakukan oleh nasabah. Layanan pengadaan ini dapat diberikan kepada nasabah dalam bentuk uang maupun barang yang akan dikembalikan oleh nasabah secara berangsur.

2.1.1 Custom Saving dan Smart Saving

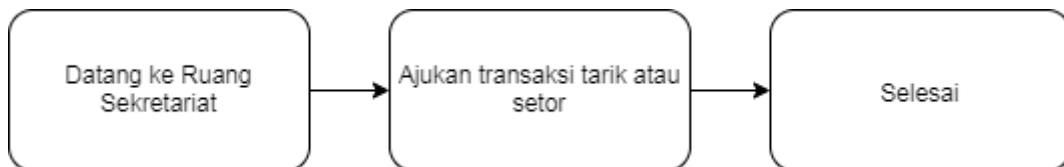
Produk yang paling sering digunakan oleh nasabah BMT adalah *Custom Saving* dan *Smart Saving* dimana nasabah dapat menggunakan produk ini sebagai tabungan mereka. Selama ini dalam pendaftaran atau melakukan transaksi, nasabah

diharuskan untuk datang ke ruang sekretariat. *Flow* nasabah yang sekarang untuk membuka tabungan dapat dilihat pada Gambar 2.1.



Gambar 2.1 Flow buka tabungan

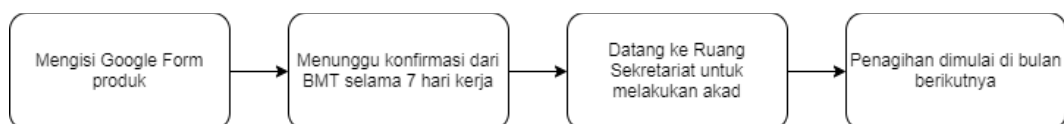
Sama halnya dengan pembukaan tabungan, dalam melakukan transaksi setor atau tarik-pun nasabah juga diharuskan untuk datang ke ruang sekretariat. *Flow* nasabah dalam melakukan transaksi dapat dilihat pada Gambar 2.2.



Gambar 2.2 Flow transaksi

2.1.2 Qardh dan Pengadaan

Produk lainnya dari BMT FEB Unpad adalah Qardh dan Pengadaan. Produk qardh memperbolehkan nasabah untuk mengajukan peminjaman uang yang nantinya akan dikembalikan dengan cicilan tanpa adanya biaya tambahan. Sedangkan dengan produk pengadaan, nasabah dapat mengajukan pengadaan barang yang diinginkan nasabah lalu dapat dibayar dengan cicilan tanpa biaya tambahan. *Flow* nasabah yang sekarang untuk mengajukan produk dapat dilihat pada Gambar 2.3.



Gambar 2.3 Flow Qardh dan Pengadaan

2.2 UML






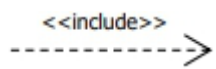
Unified Modeling Language (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun

perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem (Gata & Gata, 2013).

a. *Use Case Diagram*

Menggambarakan *actor*, *use case* dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk aktor. Sebuah *use case* digambarkan sebagai elips horizontal dalam suatu diagram UML *use case* (Ropianto, 2016). Perhatikan Tabel 2.1 untuk penjelasan dari simbol-simbol *use case diagram*.

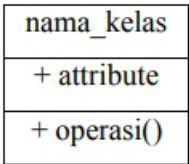






Tabel 2.1 *Use Case Diagram*

No	Symbol	Keterangan
1	<p><i>Use Case</i></p> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor
2	<p><i>Actor</i></p> 	Orang atau sistem lain yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri. Biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3	<p><i>Association</i></p> 	Komunikasi antara aktor dengan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.
4	<p><i>Extend</i></p> 	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu.
5	<p><i>Generalization</i></p> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antar dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6	<p><i>Include</i></p> 	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini.

b. *Class Diagram*

Menggambarkan struktur statis dari kelas dalam sistem, serta menggambarkan atribut, operasi dan hubungan antara kelas. Selama tahap desain, *class diagram* berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat. Perhatikan Tabel 2.2 untuk penjelasan dari simbol-simbol *class diagram*.


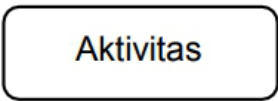
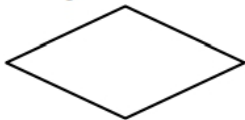


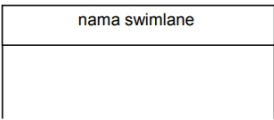
Tabel 2.2 Tabel *Class Diagram*

No	Symbol	Keterangan
1	<p>Kelas</p> 	Kelas pada struktur sistem.
2	<p>Antarmuka / <i>interface</i></p>  <p>nama_interface</p>	Sama dengan konsep interface dalam pemrograman berorientasi objek.
3	<p>Asosiasi / <i>association</i></p> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity.
4	<p>Asosiasi berarah / <i>directed association</i></p> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity.
5	<p>Generalisasi</p> 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus).
6	<p>Kebergantungan / <i>depedency</i></p> 	Relasi antarkelas dengan makna kebergantungan antarkelas.
7	<p>Agregasi / <i>aggregation</i></p> 	Relasi antarkelas dengan makna semuabagian (whole-part).

c. *Activity Diagram*

Menggambarkan aktifitas-aktifitas, objek, state, transisi state dan event. Dengan kata lain, kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas. Perhatikan Tabel 2.3 untuk penjelasan dari simbol-simbol *activity diagram*.

Tabel 2.3 Tabel *Activity Diagram*

No	Symbol	Keterangan
1	Status Awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3	Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4	Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5	Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6	Swimlane 	Swimlane memisahkan organisasi bisnis yang bertanggungjawab terhadap aktivitas yang terjadi.

2.3 Pengembangan Aplikasi Berbasis Android

Pada bagian ini akan dijelaskan mengenai komponen-komponen yang digunakan untuk membangun aplikasi berbasis *android* seperti bahasa dan arsitektur yang digunakan.

2.3.1 Android

Android adalah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android SDK menyediakan *tools* dan *Application Programming Interface* yang diperlukan untuk mengembangkan aplikasi pada *platform android* menggunakan bahasa

pemrograman Java. Android merupakan sistem operasi berbasis Linux, sehingga layanan sistemnya dikontrol oleh Linux (Holla & Katti, 2012).

2.3.2 Java

Java adalah bahasa pemrograman yang dapat membuat seluruh bentuk aplikasi, desktop, web, mobile dan lainnya sebagaimana dibuat dengan menggunakan bahasa pemrograman konvensional yang lain. Bahasa pemrograman Java ini berorientasi objek dan dapat dijalankan pada berbagai platform sistem operasi. Perkembangan java tidak hanya terfokus pada suatu sistem operasi tetapi dikembangkan untuk berbagai sistem operasi dan bersifat *open source* (Enjelina & Sagala, 2016).

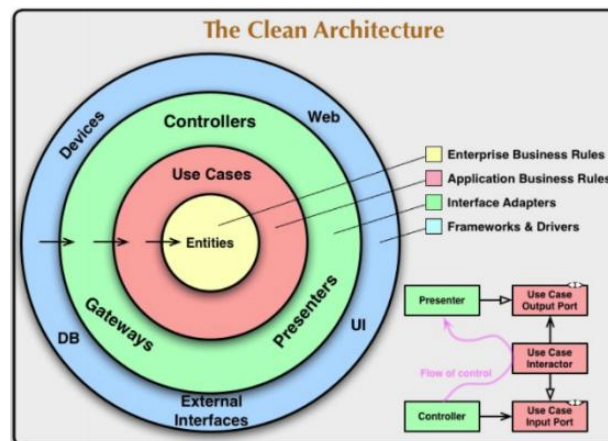
2.3.3 XML

XML adalah singkatan dari eXtensible Markup Language. Bahasa markup adalah sekumpulan aturan-aturan yang mendefinisikan suatu sintaks yang digunakan untuk menjelaskan, dan mendeskripsikan teks atau data dalam sebuah dokumen melalui penggunaan tag. XML didesain untuk mampu menyimpan data secara ringkas dan mudah diatur.

XML adalah sebuah bahasa markup yang digunakan untuk mengolah metadata (informasi tentang data) yang menggambarkan struktur dan maksud/tujuan data yang terdapat dalam dokumen XML, namun bukan menggambarkan format tampilan data tersebut. XML adalah sebuah standar sederhana yang digunakan untuk mendeskripsikan data teks dengan cara *self-describing* (Hidayah, 2017).

2.3.4 Clean Architecture

Clean Architecture menghasilkan suatu sistem yang memungkinkan *business logic* dapat diuji secara independen dari *framework*, *User Interface* atau komponen lainnya (Duy, 2017). Gambar 2.1 memperlihatkan bagaimana setiap *layer* dan *data flow* terpisah pada *clean architecture*.



Gambar 2.4 *The Clean Architecture* (Martin, 2012)

2.3.5 Model View Presenter

MVP *pattern* memiliki konsep yang kurang lebih sama dengan MVC, tetapi dengan paradigma yang lebih modern yang membuat pemisahan komponen lebih baik. Seperti pada Gambar 2.1, *View* disini benar-benar pasif sehingga tidak bisa menerima data apapun dari model sampai *presenter* memanggilnya untuk melakukan sesuatu. *Presenter* berperan sebagai mediator antara model dan *view*, mendapatkan data dari model lalu memberitahu *view* untuk memproses data untuk ditampilkan ke pengguna (Duy, 2017).

2.5 API

Secara umum API merupakan keseluruhan fungsional dalam suatu modul *software* yang dapat diakses oleh orang yang membutuhkan dengan cara yang telah ditentukan layanan. Representasi terfokus dari fungsi yang dideklarasikan dalam API dimaksudkan untuk menyediakan rangkaian layanan yang spesifik untuk target tertentu (Rama & Avinash Kak, 2013).

2.6 Laravel

Laravel adalah sebuah MVC web development framework yang didesain untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan dan perbaikan serta meningkatkan produktifitas pekerjaan dengan sintak yang bersih dan fungsional yang dapat mengurangi banyak waktu untuk

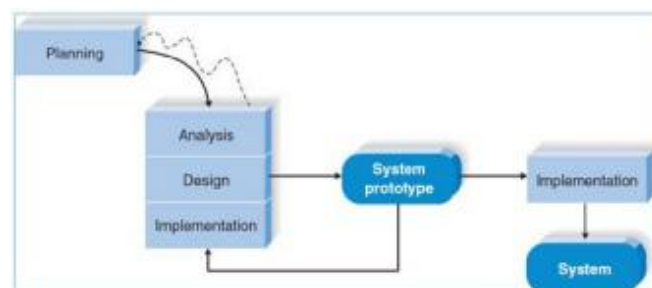
implementasi (Widodo & Purnomo, 2016). Laravel merupakan framework PHP yang menekankan pada kesederhanaan dan fleksibilitas pada desainnya (Naista, 2016).

2.7 MySQL

MySQL adalah suatu sistem basis data relation atau *Relational Database Management System* (RDBMS) yang mampu bekerja secara cepat dan mudah digunakan MySQL juga merupakan program pengakses *database* yang bersifat jaringan, sehingga dapat digunakan untuk aplikasi *multi-user* dan *multi-thread* (Kristanto, 2010).

2.8 Metode *Rapid Application Development*

Rapid Application Development (RAD) adalah model pengembangan perangkat lunak yang terdiri dari sekumpulan metode yang telah dikembangkan untuk mengatasi kelemahan dari pengembangan model tradisional. Pertama kali dikenalkan oleh James Martin pada tahun 1990-an, ia percaya bahwa model ini lebih fleksibel dan adaptif pada kebutuhan *user* dan dapat mengembangkan sistem dengan cepat, berkualitas baik dengan biaya minimum. Pada implementasinya, RAD menekankan pada proses perencanaan yang pendek dengan berfokus pada proses yang terdiri dari pengembangan, pengujian dan *feedback*. Model *Rapid Application Development* terdiri atas empat tahapan utama yaitu, perencanaan kebutuhan, desain pengguna, konstruksi dan implementasi. Terdapat tiga cara dalam tahapan implementasi, dimana cara-cara tersebut adalah pengembangan iteratif, *system prototyping* dan *throwaway prototyping* (Delima, Santoso, & Purwadi, 2017). Gambar 2.2 merupakan flow dari metode *system prototyping*.



Gambar 2.5 Tahapan RAD dengan system prototyping

Penulis akan menggunakan metode RAD dengan tahapan yang pertama melakukan perencanaan. Perencanaan meliputi estimasi waktu pengerjaan, tujuan dan manfaat yang akan dicapai dari solusi dalam bentuk aplikasi untuk menyelesaikan masalah dari *stakeholder*.

Tahapan berikutnya adalah analisis. Analisis yang pertama dilakukan adalah analisis kebutuhan pengguna, pada analisis ini akan dijabarkan hal-hal yang dibutuhkan calon pengguna untuk menggunakan aplikasi. Dilanjutkan dengan analisis terhadap perangkat lunak dan keras yang dibutuhkan untuk mengembangkan aplikasi. Analisis berikutnya adalah analisis kebutuhan sistem atau fitur-fitur yang harus dihadirkan dalam aplikasi yang dikembangkan. Dari analisis kebutuhan pengguna yang sudah dilakukan, pengembang akan melakukan desain *flow* aplikasi pengguna yang akan digambarkan dalam bentuk *activity diagram*. Lalu dari analisis fitur dan *flow* aplikasi yang sudah dibuat, pengembang akan mendesain skema database berdasarkan dari dua hal yang sudah dimiliki tersebut. Ketika desain *flow* dan database sudah selesai, baru-lah pengembang dapat melakukan desain *low-fidelity* dari *user interface* aplikasi yang akan dikembangkan.

Setelah ini pengembangan akan memasuki siklus *prototype*. Pada siklus ini pengembang akan melakukan *develop-demonstrate-refine*, dimana *prototype* yang sudah dibuat pengembang akan didemonstrasikan kepada *stakeholder*, dan *feedback* yang didapat akan digunakan sebagai dasar pengembangan *prototype* pada siklus selanjutnya. Tahapan ini akan berlanjut hingga pengembang, pengguna dan *stakeholder* setuju bahwa sistem telah mencakup seluruh kebutuhan dan fungsionalitas yang dibutuhkan. Ketika *prototype* yang dikembangkan sudah mencapai tahapan final, akan dilakukan testing terhadap aplikasi yang dikembangkan. Pengembang dalam tahapan ini menggunakan *usability testing* sebagai instrument pengujian aplikasinya. Bila hasil *testing* yang dilakukan sudah dirasa baik, barulan aplikasi di-*release*.

BAB III

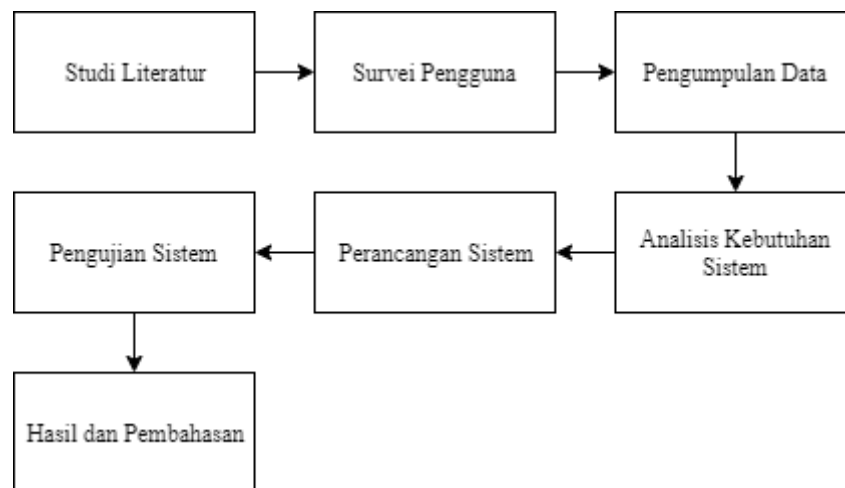
ANALISIS PERANCANGAN SISTEM

Bab ini menguraikan secara rinci mengenai metode penelitian yang digunakan yaitu analisis kebutuhan, proses perancangan, pengujian perangkat lunak, serta alat dan bahan yang dibutuhkan.

3.1 Desain Sistemika Penelitian

Dalam melakukan penelitian diperlukan acuan dasar pelaksanaan agar penelitian ini dapat berjalan sesuai dengan apa yang diharapkan, sehingga dapat diimplementasikan dan memberikan hasil yang baik. Bagan pada Gambar 3.1 merupakan desain penelitian yang menggambarkan setiap tahap dari kegiatan penelitian yang akan dilakukan.

Penelitian ini dilakukan secara sistematis melalui beberapa tahap yaitu studi literatur, melakukan survei, pengumpulan data, analisis kebutuhan sistem perancangan sistem, pengujian sistem, serta hasil dan pembahasan.



Gambar 3.1 Sistemika Penelitian

Pada Gambar 3.1, dilakukan yang diimplementasikan melalui aplikasi berbasis android dan API menggunakan framework Laravel. Kemudian, dilakukan survei yang ditujukan kepada nasabah BMT FEB Unpad terkait hal-hal apa saja

yang diharapkan ada di dalam aplikasi yang dibangun. Selanjutnya, dilakukan analisis terhadap kebutuhan sistem. Untuk tahap perancangan sistem, menggunakan framework android *native* Java, database MySQL dan framework Laravel. Pengujian sistem akan dilakukan oleh pengurus dan nasabah BMT FEB Unpad. Tahap selanjutnya, dibuat laporan terkait hasil dan pembahasan dalam bentuk skripsi.

3.2 Studi Literatur

Studi literatur adalah kegiatan mencari teori, sumber-sumber, dan referensi yang diperlukan untuk membantu penelitian. Pada penelitian ini studi literatur yang dilakukan adalah mencari jurnal, artikel dan teori yang relevan dengan Baitul Maal wa Tamwil, UML, Android, Java, XML, *Clean Architecture*, MVP, CMS, Laravel, PHP, MySQL.

3.3 Pengumpulan Data

Data yang dikumpulkan bersumber dari BMT FEB Unpad yang didapat langsung dari pengurus BMT FEB Unpad. Data yang dikumpulkan berupa jawaban dari kuesioner yang disebar ke calon pengguna tentang harapan serta masukan untuk rancangan aplikasi yang akan dirancang.

3.4 Analisis Kebutuhan

Analisis kebutuhan pada penelitian ini dibagi menjadi empat bagian, yaitu analisis kebutuhan pengguna, analisis kebutuhan perangkat lunak (*software*), analisis kebutuhan perangkat keras (*hardware*), serta analisis kebutuhan fungsional.

3.4.1 Analisis Kebutuhan Pengguna

Analisis kebutuhan pengguna bertujuan untuk mengetahui siapa saja pengguna yang dapat menggunakan sistem ini. Berikut ini adalah pengguna yang dapat menggunakan sistem ini:

1. Pengguna memiliki kemampuan untuk mengoperasikan *smartphone android* (untuk nasabah).

2. Pengguna memiliki kemampuan dan sudah terbiasa menggunakan PC (Personal Computer).

3.4.2 Analisis Kebutuhan Perangkat Lunak

Perangkat lunak yang digunakan untuk mengembangkan aplikasi adalah sebagai berikut:

- a. Android Studio 3.5, yaitu aplikasi yang digunakan untuk melakukan *coding* aplikasi BMT di android.
- b. Visual Studio Code, yaitu aplikasi yang digunakan untuk melakukan *coding* API.
- c. *Framework* Laravel 6.5.2, yaitu *framework* yang digunakan untuk melakukan *coding* API.
- d. Composer 1.8.2, yaitu sebuah *dependency manager* untuk PHP yang digunakan untuk menambah *library website* secara otomatis.
- e. DBeaver, yaitu sebuah *SQL client* aplikasi sebagai aplikasi untuk mengakses database.

3.4.3 Analisis Kebutuhan Perangkat Keras

Perangkat keras yang digunakan untuk mengembangkan aplikasi adalah satu buah laptop dengan spesifikasi sebagai berikut:

- a. Sistem Operasi : Windows 7 Ultimate 64 Bit
- b. Sistem Manufaktur : Lenovo G40-45
- c. *Processor* : AMD A8-6410 APU @2.00 GHz
- d. RAM : 8.00 GB
- e. Hard Disk Drive : 500 GB
- f. *Graphic Processor* : AMD Radeon R5 Graphics

Sedangkan perangkat keras yang dibutuhkan untuk menjalankan aplikasi adalah satu buah *smartphone* android dengan spesifikasi sebagai berikut:

- a. Sistem Operasi : 5.0 (Lollipop) atau API Level 21
- b. Memiliki kamera dan internet yang berfungsi dengan baik

3.4.4 Analisis Kebutuhan Fungsional

Adapun spesifikasi kebutuhan fungsional dari pembangunan sistem ini dijabarkan ke dalam kebutuhan atau fitur serta keterangan sebagaimana dituliskan pada Tabel 3.1 di bawah ini.

Tabel 3.1 Kebutuhan Fungsional Pengguna

Kode Requirement	Kebutuhan	Keterangan
Req-01	Login pengguna	Perangkat lunak melakukan login
Req-02	Register pengguna	Perangkat lunak melakukan registrasi
Req-03	Profil pengguna	Perangkat lunak menampilkan data profil pengguna
Req-04	Pengguna edit data profil	Perangkat lunak dapat mengubah data profil
Req-05	Logout pengguna	Perangkat lunak melakukan logout
Req-06	Pengguna melihat produk simpan/pinjam	Perangkat lunak menampilkan daftar produk
Req-07	Pengguna mendapat penjelasan produk simpan/pinjam	Perangkat lunak menampilkan penjelasan produk
Req-08	Pengguna mendaftar untuk produk simpan/pinjam	Perangkat lunak dapat menambahkan data untuk produk yang dibutuhkan

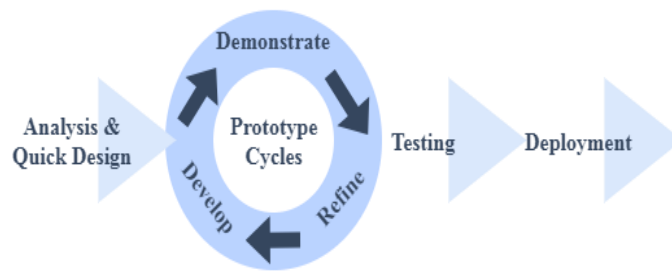
Req-09	Pengguna mengajukan simpan/pinjam	Perangkat lunak dapat melakukan transaksi simpan/pinjam
Req-10	Pengguna melihat riwayat simpan/pinjam	Perangkat lunak dapat menampilkan riwayat transaksi pengguna
Req-11	Pengguna mengkonfirmasi transaksi yang diajukan	Perangkat lunak dapat digunakan untuk mengkonfirmasi transaksi yang diajukan
Req-12	Pengguna melihat FAQ	Perangkat lunak menampilkan daftar FAQ
Req-13	Pengguna edit data FAQ	Perangkat lunak dapat mengubah data FAQ
Req-14	Pengguna melihat tentang organisasi	Perangkat lunak menampilkan tentang organisasi

3.5 Perencanaan Proses dan GUI

Pada perencanaan penelitian yang diteliti oleh penulis, akan dipaparkan di bawah ini Proses RAD (*Rapid Application Development*), Perancangan Proses, Struktur Database, dan Perancangan GUI (*Graphical User Interface*).

3.5.1 Proses RAD (*Rapid Application Development*)

Metode yang digunakan pada penelitian ini adalah metode *Rapid Application Development*. Metode yang fleksibel dan adaptif pada kebutuhan *user* dan dapat mengembangkan sistem dengan cepat, berkualitas baik dengan biaya minimum. Pada implementasinya, RAD menekankan pada proses perencanaan yang pendek. Pada Gambar 3.2, merupakan metode RAD yang digunakan oleh peneliti untuk perancangan aplikasi.



Gambar 3.2 Metode RAD

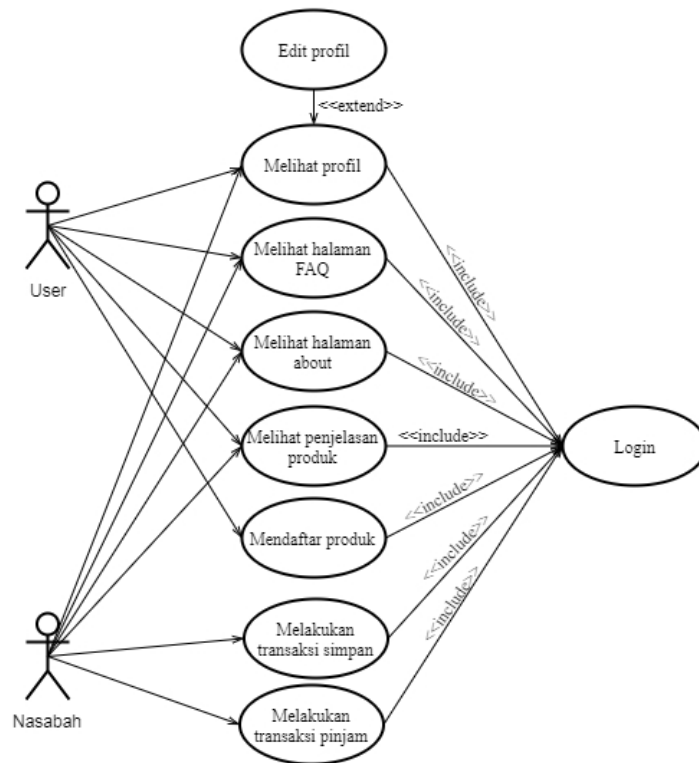
Karena metode ini cocok untuk aplikasi yang belum ada dan dikembangkan oleh hanya satu orang, serta ditargetkan untuk skala pengguna yang tidak terlalu besar. Metode ini dapat memperkecil sumber daya yang dipakai baik waktu dan tenaga dikarenakan hanya ada satu kali *testing* dan *deployment*. Jadi segala proses analisis dan perancangan hanya dilakukan satu kali di awal, dan hanya ada perbaikan-perbaikan minor di siklus pengembangannya.

3.5.2 Perancangan Proses

Rancangan proses merupakan gambaran perencanaan dan pembangunan aplikasi yang menjelaskan cara kerja dari aplikasi tersebut. Tujuan dari perancangan ini adalah memberikan penjelasan kepada user mengenai cara kerja aplikasi yang akan dibangun. Perancangan sistem dimodelkan menggunakan metode RAD.

c. *Use Case Diagram*

Use case diagram menggambarkan secara ringkas sebuah interaksi yang dapat dilakukan oleh satu atau lebih user dengan sistem yang akan dibuat. Use case diagram tidak akan menjelaskan secara detail tentang penggunaan use case, namun hanya memberikan gambaran singkat hubungan antara user dengan sistem. Sebuah use case harus mendeskripsikan sebuah pekerjaan dimana pekerjaan tersebut akan memberikan nilai yang bermanfaat bagi aktor. *Use case diagram* dari aplikasi yang akan dikembangkan dapat dilihat pada Gambar 3.3.



Gambar 3.3 Use Case Diagram

Berdasarkan Gambar 3.3 dapat dijelaskan sebagai berikut:

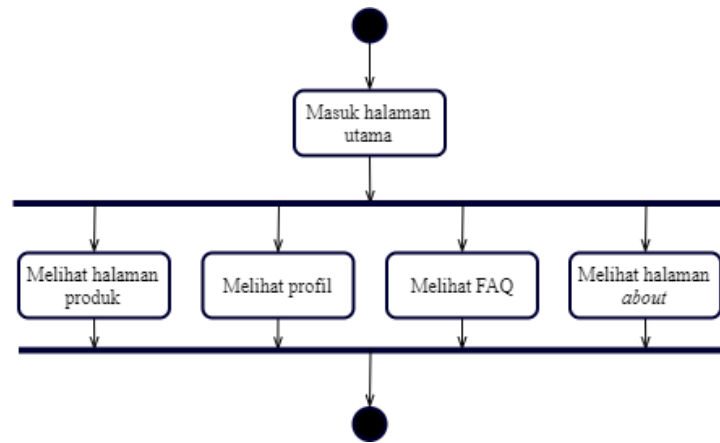
Pengguna dalam sistem ini dibedakan menjadi dua yaitu *user* dan *full-access user*.

1. Pengguna dengan level *user* hanya dapat melihat profil, mengubah data profil, melihat halaman FAQ, melihat halaman about, melihat penjelasan produk serta melakukan registrasi produk
2. Nasabah dapat melakukan semua hal yang dapat dilakukan user kecuali mendaftar pada produk yang sama. Nasabah juga dapat melakukan transaksi simpan ataupun transaksi pinjam.

b. *Activity Diagram*

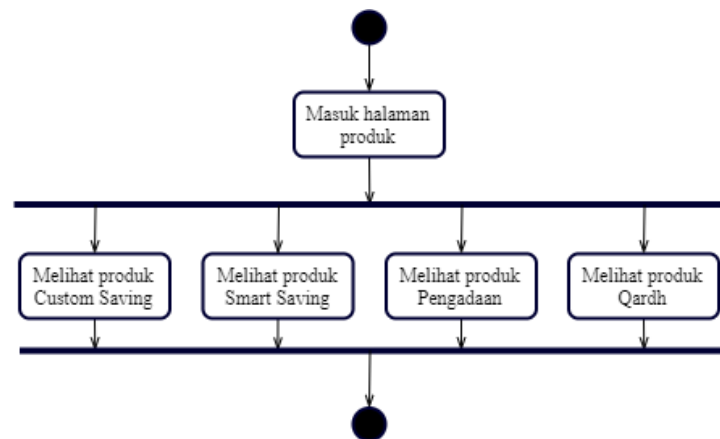
Activity Diagram adalah diagram yang menggambarkan *workflow* (alur kerja) atau aktivitas dari proses bisnis. Yang perlu diperhatikan adalah *activity diagram* menggambarkan aktivitas sistem bukan apa yang dilakukan aktor melainkan aktivitas yang dapat dilakukan oleh sistem. Ketika pengguna masuk

halaman utama, halaman produk, melakukan transaksi, melihat daftar FAQ, melihat halaman about, melihat halaman profil dan mengubah data profil



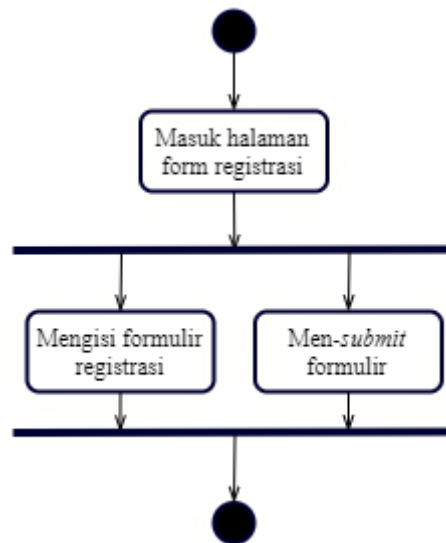
Gambar 3.4 *Activity Diagram* halaman utama

Gambar 3.4 menjelaskan diagram aktivitas ketika pengguna masuk pada halaman utama sistem. Ketika pengguna masuk pada halaman utama, pengguna dapat melakukan navigasi ke halaman-halaman produk, profil, FAQ dan *about*.



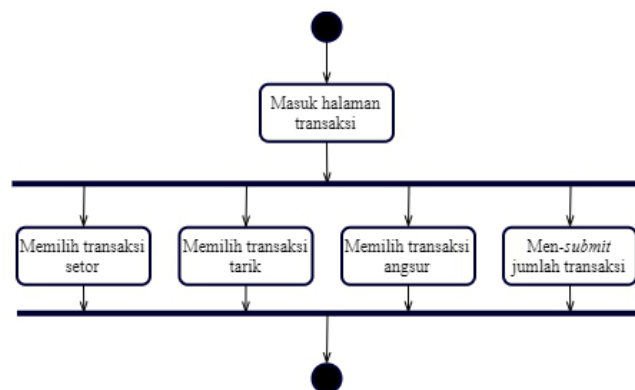
Gambar 3.5 *Activity Diagram* produk

Kemudian pada Gambar 3.5 menjelaskan diagram aktivitas untuk halaman produk. Halaman produk berfungsi untuk menavigasikan pengguna ke halaman-halaman produk yang tersedia. Apabila pengguna yang memilih produk belum terdaftar sebagai nasabah, maka pengguna akan diarahkan untuk mengisi form pendaftaran terlebih dahulu.



Gambar 3.6 *Activity Diagram* Form Nasabah

Kemudian pada Gambar 3.6 menjelaskan diagram aktivitas untuk halaman formulir registrasi. Halaman formulir akan diisi dengan data-data calon nasabah yang akan dikirim ke database BMT. Data-data ini bersifat personal guna memvalidasi keaslian nasabah.



Gambar 3.7 *Activity Diagram* transaksi

Kemudian pada Gambar 3.7 menjelaskan diagram aktivitas untuk halaman transaksi. Halaman ini nasabah akan memilih jenis transaksi yang akan dilakukan (tarik atau setor), lalu nasabah diharuskan memilih nominal dan metode transaksi yang akan dilakukan.

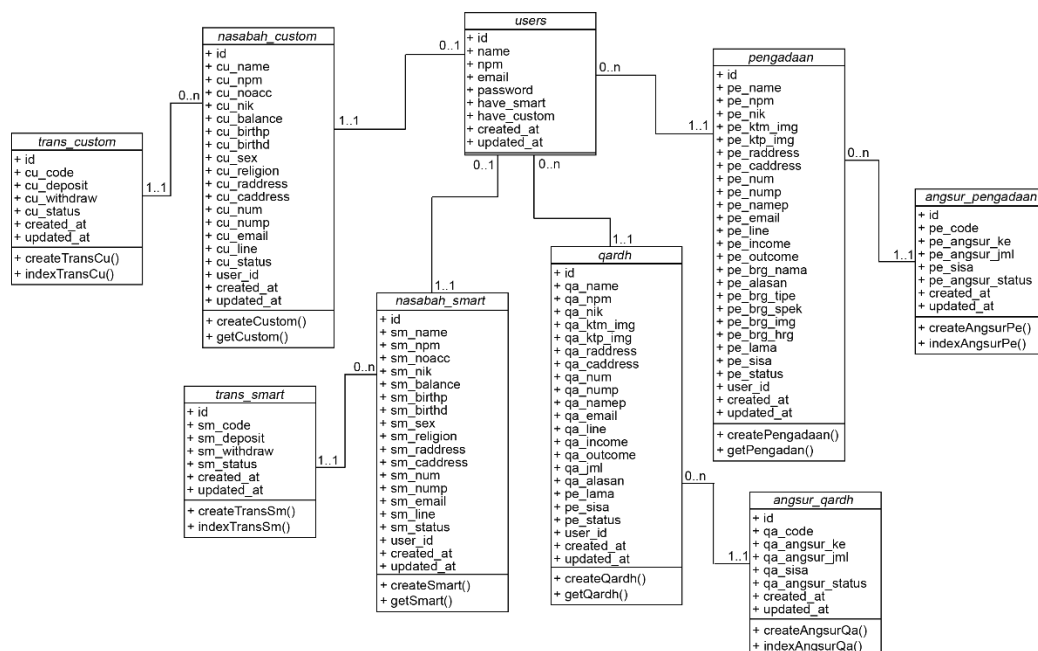
c. *Class Diagram*

Gambar di bawah ini menjelaskan hubungan antar *class* yang digunakan untuk merancang sistem ini. Terdapat 9 class yang digunakan diantaranya class *users*, *nasabah_custom*, *nasabah_smart*, *qardh*, *pengadaan*, *transaksi_custom*, *transaksi_smart*, *angsur_qardh* dan *angsur_pengadaan*. Hubungan antar setiap class juga dijelaskan pada gambar dibawah dengan menggunakan notasi kardinalitas atau *multiplicity* seperti pada Tabel 3.2.

Tabel 3.2 Multiplicity

Multiplicity	Arti
n (default)	Banyak
0..0	Nol
0..1	Nol atau satu
0..n	Nol tau banyak
1..1	Tepat satu
1..n	Satu atau banyak

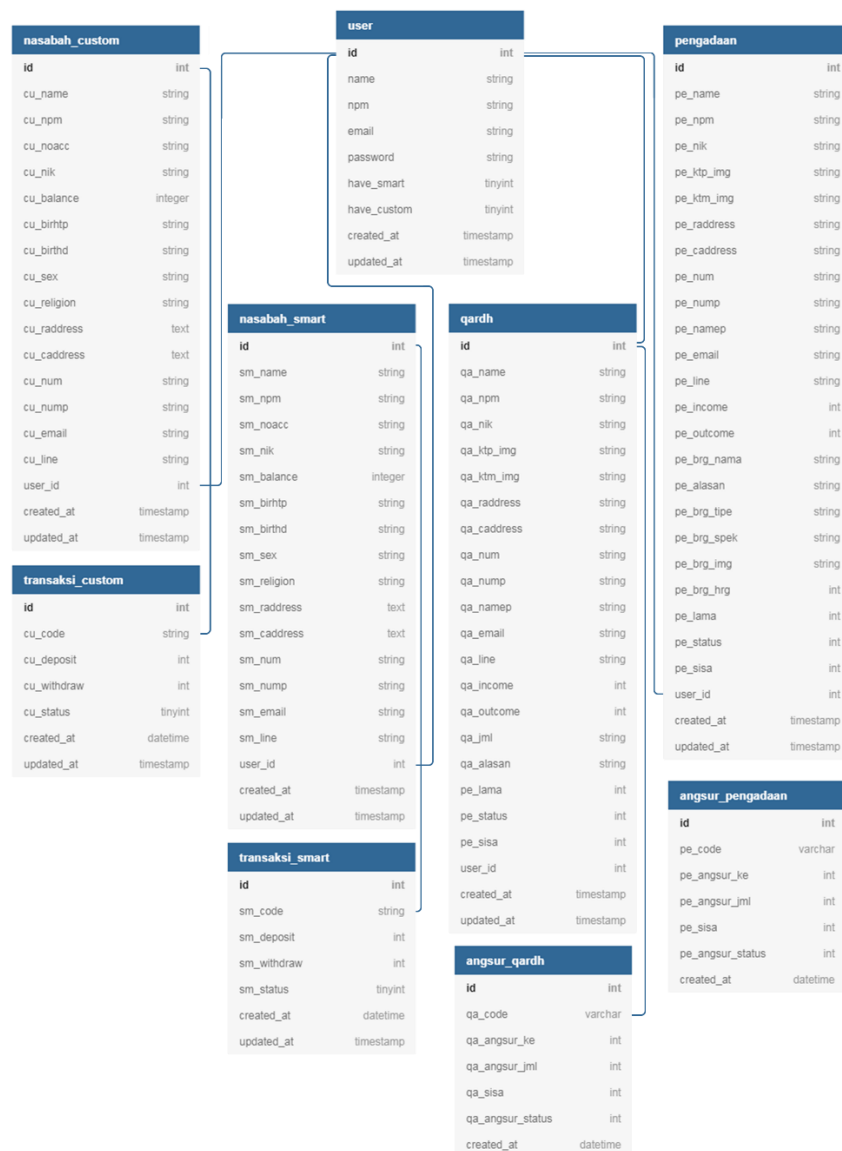
Gambaran dari hubungan antar setiap class beserta atributnya dapat dilihat pada Gambar 3.8.



Gambar 3.8 Class Diagram

3.5.3 Perencanaan Database

Pada Gambar 3.8 Skema Perancangan Database, merupakan skema database dari aplikasi yang akan dikembangkan. Perancangan database ini dirancang oleh pengembang sesuai dengan data nasabah yang didapatkan dari objek penelitian, yaitu Baitul Maal wa Tamwil FEB Unpad. Setiap kolom pada tabel dapat dilengkapi oleh user yang sudah terdaftar. Database yang akan dirancang terdiri dari 9 (sembilan) tabel yang terdiri dari tabel users, tabel nasabah_smart, tabel nasabah_custom, tabel pengadaan, tabel qardh, tabel transaksi_smart, tabel transaksi_custom, tabel angsur_qardh dan tabel angsur_pengadaan.



Gambar 3.9 Skema Perancangan Database

3.5.4 Struktur Tabel

Pada implementasi database yang terdiri dari beberapa tabel akan dibahas pada struktur tabel di bawah ini. Untuk menjelaskan tipe relasi pada tabel dan *query* tabel apa saja yang ada pada database sesuai dengan skema perancangan database yang sudah dibahas sebelumnya.

a. Tabel user

Tabel user berisi nama, npm, email, dan password yang terdaftar di dalam basis data. Field beserta tipe data dari tabel user dapat dilihat pada tabel 3.3.

Tabel 3.3 Atribut *user*

No	Nama Field	Tipe Data	Keterangan	Null
1	id	Int	Primary Key, auto-increment	Tidak
2	name	String	Menyimpan nama	Tidak
3	npm	String	Menyimpan NPM	Tidak
4	email	String	Menyimpan email	Tidak
5	password	String	Menyimpan password	Tidak
6	have_smart	Int	Menyimpan status kepemilikan user terhadap smart saving	Tidak
7	have_custom	Int	Menyimpan status kepemilikan user terhadap custom saving	Tidak
8	created_at	Timestamp	Tanggal pembuatan user	Tidak
9	updated_at	Timestamp	Tanggal terakhir diubah	Tidak

b. Tabel nasabah_smart

Tabel nasabah_smart berisi informasi yang dibutuhkan untuk membuka rekening smart saving. Atribut dari tabel ini bersifat personal agar yang nantinya akan dikonfirmasi langsung oleh pengurus BMT FEB Unpad. Field beserta tipe data pada tabel nasabah_smart dapat dilihat pada Tabel 3.4.

Tabel 3.4 Atribut nasabah smart

No	Nama Field	Tipe Data	Keterangan	Null
1	id	String	Primary Key, auto-increment	Tidak
2	sm_name	String	Menyimpan nama	Tidak
3	sm_npm	String	Menyimpan NPM	Tidak
4	sm_noacc	String	Menyimpan nomor rekening smart	Ya
5	sm_nik	String	Menyimpan nik	Tidak
6	sm_balance	Int	Menyimpan saldo	Tidak
7	sm_birthp	String	Menyimpan alamat asal	Tidak
8	sm_birthd	String	Menyimpan tanggal lahir	Tidak
9	sm_gender	String	Menyimpan jenis kelamin	Tidak
10	sm_religion	String	Menyimpan agama	Tidak
11	sm_raddress	String	Menyimpan alamat asal	Tidak
12	sm_caddress	String	Meyimpan alamat sekarang atau kos	Tidak
13	sm_phone	String	Menyimpan nomor telepon pribadi	Tidak
14	sm_phonep	String	Menyimpan nomor telepon orang tua	Tidak
15	sm_email	String	Menyimpan email	Tidak
16	sm_line	String	Menyimpan ID Line	Tidak
17	sm_status	Integer	Menyimpan status rekening	Tidak
18	user_id	Integer	Mengambil id dari tabel user terkait	Tidak
19	created_at	Timestamp	Tanggal pembuatan rekening	Tidak
20	updated_at	Timestamp	Tanggal terakhir diubah	Tidak

c. Tabel nasabah_custom

Tabel nasabah_custom berisi informasi yang dibutuhkan untuk membuka rekening custom saving. Atribut dari tabel ini bersifat personal agar yang nantinya akan dikonfirmasi langsung oleh pengurus BMT FEB Unpad. Field beserta tipe data dari tabel nasabah_custom dapat dilihat pada Tabel 3.5.

Tabel 3.5 Atribut nasabah custom

No	Nama Field	Tipe Data	Keterangan	Null
1	id	String	Primary Key, auto-increment	Tidak
2	cu_name	String	Menyimpan nama	Tidak
3	cu_npm	String	Menyimpan NPM	Tidak
4	cu_noacc	String	Menyimpan nomor rekening	Ya
5	cu_nik	String	Menyimpan nik	Tidak
6	cu_balance	Int	Menyimpan saldo	Tidak
7	cu_birthp	String	Menyimpan alamat asal	Tidak
8	cu_birtd	String	Menyimpan tanggal lahir	Tidak
9	cu_gender	String	Menyimpan jenis kelamin	Tidak
10	cu_religion	String	Menyimpan agama	Tidak
11	cu_raddress	String	Menyimpan alamat asal	Tidak
12	cu_caddress	String	Meyimpan alamat sekarang atau kos	Tidak
13	cu_phone	String	Menyimpan nomor telepon pribadi	Tidak
14	cu_phonep	String	Menyimpan nomor telepon orang tua	Tidak
15	cu_email	String	Menyimpan email	Tidak
16	cu_line	String	Menyimpan ID Line	Tidak
17	cu_status	Integer	Menyimpan status dari rekening	Tidak
18	user_id	Integer	Mengambil id dari tabel user terkait	Tidak
19	created_at	Timestamp	Tanggal pembuatan rekening	Tidak
20	updated_at	Timestamp	Tanggal terakhir diubah	Tidak

d. Tabel qardh

Tabel qardh berisi informasi yang dibutuhkan untuk mengajukan peminjaman jenis qardh. Atribut dari tabel ini bersifat personal agar yang nantinya akan dikonfirmasi langsung oleh pengurus BMT FEB Unpad. Field beserta tipe data pada tabel ini dapat dilihat pada tabel 3.6.

Tabel 3.6 Atribut qardh

No	Nama Field	Type Data	Keterangan	Null
1	id	String	Primary Key, auto-increment	Tidak
2	qa_name	String	Menyimpan nama	Tidak
3	qa_npm	String	Menyimpan NPM	Tidak
4	qa_nik	String	Menyimpan nik	Tidak
5	qa_ktm_img	String	Menyimpan link dari foto KTM	Tidak
6	qa_ktp_img	String	Menyimpan link dari foto KTP	Tidak
7	qa_raddress	String	Menyimpan alamat asal	Tidak
8	qa_caddress	String	Meyimpan alamat sekarang atau kos	Tidak
9	qa_phone	String	Menyimpan nomor telepon pribadi	Tidak
10	qa_phonep	String	Menyimpan nomor telepon orang tua	Tidak
11	qa_namep	String	Menyimpan nama orang tua	Tidak
12	qa_email	String	Menyimpan email	Tidak
13	qa_line	String	Menyimpan ID Line	Tidak
14	qa_outcome	Integer	Menyimpan pengeluaran per bulan	Tidak
15	qa_income	Integer	Menyimpan pemasukan per bulan	Tidak
16	qa_jml	Integer	Menyimpan jumlah uang yang dipinjam	Tidak
17	qa_alasan	String	Menyimpan alasan pengajuan peminjaman	Tidak
18	qa_lama	Integer	Menyimpan lama peminjaman akan dicicil	Tidak
19	qa_sisa	Integer	Menyimpan sisa angsuran	Tidak
20	qa_status	Integer	Menyimpan status pengajuan	Tidak
21	user_id	Integer	Mengambil id dari user terkait	Tidak
22	created_at	Timestamp	Tanggal pengajuan qardh	Tidak
23	updated_at	Timestamp	Tanggal terakhir diubah	Tidak

e. Tabel pengadaan

Tabel pengadaan berisi informasi yang dibutuhkan untuk mengajukan peminjaman jenis pengadaan. Atribut dari tabel ini bersifat personal agar yang nantinya akan dikonfirmasi langsung oleh pengurus BMT FEB Unpad Field beserta tipe data pada tabel ini dapat dilihat pada tabel 3.7

Tabel 3.7 Atribut pengadaan

No	Nama Field	Tipe Data	Keterangan	Null
1	id	String	Primary Key, auto-increment	Tidak
2	pe_name	String	Menyimpan nama	Tidak
3	pe_npm	String	Menyimpan NPM	Tidak
4	pe_nik	String	Menyimpan nik	Tidak
5	pe_ktm_img	String	Menyimpan link dari foto KTM	Tidak
6	pe_ktp_img	String	Menyimpan link dari foto KTP	Tidak
7	pe_raddress	String	Menyimpan alamat asal	Tidak
8	pe_caddress	String	Meyimpan alamat sekarang atau kos	Tidak
9	pe_phone	String	Menyimpan nomor telepon pribadi	Tidak
10	pe_phonep	String	Menyimpan nomor telepon orang tua	Tidak
11	pe_namep	String	Menyimpan nama orang tua	Tidak
12	pe_email	String	Menyimpan email	Tidak
13	pe_line	String	Menyimpan ID Line	Tidak
14	pe_outcome	Integer	Menyimpan pengeluaran per bulan	Tidak
15	pe_income	Integer	Menyimpan pemasukan per bulan	Tidak
16	pe_brg_nama	String	Menyimpan nama barang	Tidak
17	pe_alasan	String	Menyimpan alasan pengajuan peminjaman	Tidak
18	pe_brg_tipe	String	Menyimpan tipe barang	Tidak
19	pe_brg_spek	String	Menyimpan spek barang	Tidak

20	pe_brg_img	String	Menyimpan link dari foto barang	Tidak
21	pe_brg_hrg	Integer	Menyimpan harga dari barang	Tidak
22	pe_lama	Integer	Menyimpan lama peminjaman akan dicicil	Tidak
23	pe_sisa	Integer	Menyimpan sisa angsuran	Tidak
24	pe_status	Integer	Menyimpan status pengajuan	Tidak
25	user_id	Integer	Mengambil id dari user terkait	Tidak
26	created_at	Timestamp	Tanggal pengajuan pengadaan	Tidak
27	updated_at	Timestamp	Tanggal terakhir diubah	Tidak

f. Tabel transaksi_smart

Tabel transaksi_smart berisi nominal yang diajukan setiap melakukan transaksi pada smart saving. Transaksi yang dilakukan akan dikonfirmasi terlebih dahulu oleh *teller*. Field beserta tipe data pada tabel ini dapat dilihat pada Tabel 3.8.

Tabel 3.8 Atribut transaksi smart

No	Nama Field	Tipe Data	Keterangan	Null
1	id	Integer	Primary Key, auto-increment	Tidak
2	sm_code	Integer	Foreign Key dari nasabah_smart.id	Tidak
3	sm_deposit	Integer	Menyimpan nominal setoran yang terjadi	Tidak
4	sm_withdraw	Integer	Menyimpan nominal penarikan yang terjadi	Tidak
6	sm_trans_status	Integer	Menyimpan status transaksi	Tidak
7	created_at	Timestamp	Tanggal pembuatan transaksi	Tidak
8	updated_at	Timestamp	Tanggal terakhir diubah	Tidak

g. Tabel transaksi_custom

Tabel transaksi_custom berisi nominal yang diajukan setiap melakukan transaksi pada custom saving. Transaksi yang dilakukan akan dikonfirmasi terlebih dahulu oleh *teller*. Field beserta tipe data pada tabel ini dapat dilihat pada Tabel 3.9.

Tabel 3.9 Atribut transaksi_custom

No	Nama Field	Tipe Data	Keterangan	Null
1	id	Integer	Primary Key, auto-increment	Tidak
2	cu_code	Integer	Foreign Key dari nasabah_custom.id	Tidak
3	cu_deposit	Integer	Menyimpan nominal setoran yang terjadi	Tidak
4	cu_withdraw	Integer	Menyimpan nominal penarikan yang terjadi	Tidak
6	cu_trans_status	Integer	Menyimpan status transaksi	Tidak
7	created_at	Timestamp	Tanggal pembuatan transaksi	Tidak
8	updated_at	Timestamp	Tanggal terakhir diubah	Tidak

h. Tabel angsur_pengadaan

Tabel angsur_pengadaan berisi nominal angsuran yang akan dibayarkan untuk melunasi pengadaan yang sudah dilakukan. Transaksi yang dilakukan akan dikonfirmasi terlebih dahulu oleh *teller*. Field beserta tipe data pada tabel ini dapat dilihat pada tabel 3.10.

Tabel 3.10 Atribut angsur_pengadaan

No	Nama Field	Tipe Data	Keterangan	Null
1	id	Integer	Primary Key, menyimpan nomor pinjam	Tidak
2	pe_code	Integer	Foreign Key dari pengadaan.id	Tidak
3	pe_angsur_ke	Integer	Menyimpan angsuran ke berapa yang diajukan	Tidak
4	pe_angsur_jml	Integer	Menyimpan nominal angsuran yang dibayarkan	Tidak

5	pe_sisa	Integer	Menyimpan jumlah angsuran yang tersisa	Tidak
6	pe_angsur_status	Integer	Menyimpan status dari angsuran	Tidak
7	created_at	Timestamp	Tanggal pembuatan angsuran	Tidak
8	updated_at	Timestamp	Tanggal terakhir diubah	Tidak

i. Tabel angsur qardh

Tabel angsur_qardh berisi nominal angsuran yang akan dibayarkan untuk melunasi qardh yang sudah dilakukan. Transaksi yang dilakukan akan dikonfirmasi terlebih dahulu oleh *teller*. Field beserta tipe data pada tabel ini dapat dilihat pada tabel 3.11.

Tabel 3.11 Atribut angsur qardh

No	Nama Field	Tipe Data	Keterangan	Null
1	id	Integer	Primary Key, menyimpan nomor pinjam	Tidak
2	qa_code	Integer	Foreign Key dari qardh.id	Tidak
3	qa_angsur_ke	Integer	Menyimpan angsuran ke berapa yang diajukan	Tidak
4	qa_angsur_jml	Integer	Menyimpan nominal angsuran yang dibayarkan	Tidak
5	qa_sisa	Integer	Menyimpan jumlah angsuran yang tersisa	Tidak
6	pe_angsur_status	Integer	Menyimpan status dari angsuran	Tidak
7	created_at	Timestamp	Tanggal pembuatan angsuran	Tidak
8	updated_at	Timestamp	Tanggal terakhir diubah	Tidak

3.5.5 Perancangan GUI

Desain *user interface* dalam aplikasi ini akan dibuat sesederhana mungkin agar memudahkan *user* dalam menjalankan aplikasi ini dan meningkatkan pengalaman pengguna-nya, serta agar aplikasi yang dikembangkan tidak terlalu memakan memori yang besar. Berikut ini adalah rancangan desain *user interface* dari aplikasi yang akan dibuat:

a. Tampilan Halaman Login dan Register

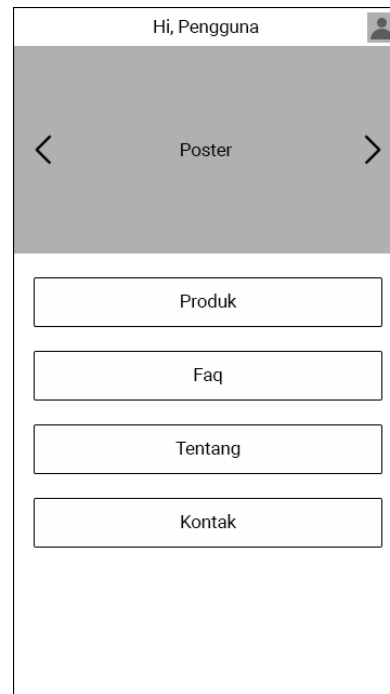
The image displays two side-by-side wireframe diagrams for a user authentication interface. Both diagrams feature a gray square placeholder labeled 'LOGO' at the top center. The left diagram represents the 'Register' page, containing four stacked input fields labeled 'Nama', 'NPM', 'Email', and 'Password', followed by a 'Register' button and a 'Login' link below it. The right diagram represents the 'Login' page, containing two stacked input fields labeled 'Email' and 'Password', followed by a 'Login' button and a 'Register' link below it.

Gambar 3.10 Halaman Register dan Login

Rancangan desain *user interface* untuk halaman login dapat dilihat pada Gambar 3.9, sedangkan rancangan desain *user interface* untuk halaman register dapat dilihat pada Gambar 3.8.

Ketika pertama membuka aplikasi, akan ditampilkahn *splashscreen* terlebih dahulu sebelum diarahkan ke halaman autentikasi yang terdiri dari halaman login dan register. Pada halaman login, user dapat menavigasi halaman ke halaman register. User dan nasabah memiliki tampilan halaman awal yang sama. User yang sudah memiliki akun dapat melakukan login dengan memasukkan email dan password yang didaftarkan pada saat register. Sedangkan user yang belum memiliki akun diharuskan untuk register terlebih dahulu agar dapat login untuk masuk ke halaman utama. Data yang harus diisi oleh user untuk melakukan register adalah nama, NPM, email dan password. Selanjutnya data ini akan disimpan di dalam database dari aplikasi.

b. Tampilan Halaman Utama



Gambar 3.11 Halaman Utama

Halaman ini (Gambar 3.11) akan terbuka ketika user sudah melakukan login. Di halaman ini terdapat 4 (tiga) button utama yang masing-masing akan mengarahkan user ke halaman dari button yang diklik, yaitu Produk, FAQ, Tentang dan Kontak. Di pojok kanan atas juga terdapat button yang akan mengarahkan user ke halaman *profile*.

c. Tampilan Halaman Profil



Gambar 3.12 Halaman Profil

Rancangan desain user interface untuk halaman profil dapat dilihat pada Gambar 3.12. Pada halaman ini, user dapat melihat data profil mereka yang sudah didaftarkan. User juga dapat melakukan pengubahan pada data profil mereka. Akan tetapi dikarenakan harus valid-nya data yang didaftarkan oleh user, maka data yang dapat diubah oleh user menjadi terbatas. Pengguna hanya dapat mengubah beberapa data yang sudah didaftarkan seperti email. Pada halaman ini user juga dapat melakukan logout atau *me-revoke* autentikasi yang sudah disimpan pada aplikasi saat user melakukan login.

d. Tampilan Halaman Produk



Gambar 3.13 Halaman Produk

Rancangan desain user interface untuk halaman produk dapat dilihat pada Gambar 3.13. Pada halaman ini, user dapat melakukan navigasi menuju halaman produk yang tersedia. Ada 4 produk yang ada yaitu, Custom Saving, Smart Saving, Pengadaan dan Qardh. Pada halaman ini juga user dapat melihat saldo yang dimiliki pada rekening produk apabila user yang login sudah memiliki nasabah pada rekening tersebut.

e. Tampilan Halaman Pendaftaran Nasabah

The image shows a mobile application interface for a customer registration form. At the top, there is a back arrow and the title 'Form Nasabah'. Below the title, there are 11 input fields, each labeled 'Field'. These fields are arranged vertically. At the bottom of the form, there is a 'Submit' button.

Gambar 3.14 Halaman Form Nasabah

Rancangan halaman pendaftaran nasabah dapat dilihat di gambar 3.14. Pada halaman form nasabah untuk *smart* dan *custom saving*, user dapat membuka rekening untuk mendaftarkan diri sebagai nasabah untuk produk yang dipilih. Untuk melakukan pendaftaran ini, user diharuskan melengkapi beberapa data yang bersifat pribadi seperti agama, tempat tanggal lahir dan kontak pribadi serta kontak orang tua. Formulir yang harus dilengkapi oleh user berbeda-beda sesuai pada produk yang dipilih oleh user. Pada produk Qardh dan Pengadaan user harus memasukkan foto atau *scan* dari KTP dan KTM. Hal ini diperlukan agar *teller* dapat memvalidasi data dari user yang akan mengajukan Qardh atau pengadaan. Pada kedua produk ini-pun user diharuskan memasukkan penghasilan dan pengeluaran yang dimiliki user dalam per-bulannya. Ada perbedaan lagi yaitu pada Qardh, user harus memasukkan nominal uang yang akan dilakukan pengajuan peminjaman, sedangkan pada Pengadaan user diharuskan untuk memasukkan nama, spek, tipe, foto dari barang pengadaan yang akan diajukan oleh user.

f. Tampilan Halaman Transaksi

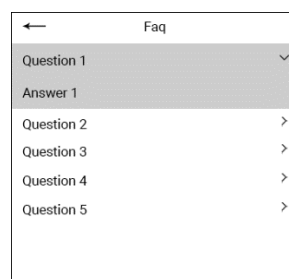


Produk	
Saldo Tabungan Rp 125.000	
Nomor Anggota 317126-2017-031	
24 Oktober 2019	
Penarikan	- 25.000
24 Oktober 2019	
Penarikan	- 25.000
24 Oktober 2019	
Penarikan	- 25.000
24 Oktober 2019	
Penarikan	- 25.000

Gambar 3.15 Halaman Transaksi

Rancangan halaman transaksi dapat dilihat di gambar 3.15. Di halaman ini, nasabah dapat melakukan transaksi yang kemudian akan dikonfirmasi oleh administrator. Nasabah juga dapat melihat transaksi-transaksi yang telah dilakukan di waktu yang lalu. Riwayat yang ditampilkan kepada nasabah berupa tanggal transaksi, jenis transaksi serta nominal transaksi. Di halaman ini juga ditampilkan nomor anggota serta saldo tabungan yang dimiliki oleh nasabah. Halaman ini hanya dapat ditampilkan kepada nasabah yang sudah terdaftar di database.

g. Tampilan Halaman FAQ

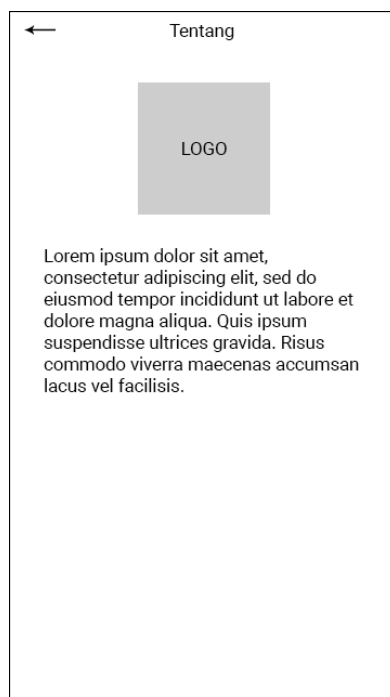


Faq	
Question 1	▼
Answer 1	
Question 2	>
Question 3	>
Question 4	>
Question 5	>

Gambar 3.16 Halaman FAQ

Rancangan halaman FAQ dapat dilihat pada Gambar 3.16. Pada halaman ini user dapat melihat *Frequently Asked Question* yang sudah dibuat oleh pengurus BMT FEB Unpad.

h. Tampilan Halaman *About*



Gambar 3.17 Halaman About

Rancangan halaman about dapat dilihat pada Gambar 3.17. Pada halaman ini user dapat membaca deskripsi serta sejarah singkat dari Baitul Maal wa Tamwil FEB Unpad.

3.6 Perancangan Pengujian Sistem

Tahapan yang akan dilakukan setelah implementasi pada aplikasi selesai adalah pengujian sistem Aplikasi Baitul Maal wa Tamwil FEB Unpad. Pengujian akan dilakukan terhadap para calon pengguna atau nasabah. Pengujian dilakukan dengan cara memberikan kuesioner kepada para calon pengguna atau nasabah sebagai pengujian usability sistem terhadap sistem yang telah dikembangkan. Kuesioner yang diberikan terdiri dari ... pertanyaan. Usability testing ini menggunakan skema penilaian 5 titik respon dari skala likert untuk mengukur hasil

respon dari setiap pernyataan yang disajikan dalam kuisioner yang ditujukan kepada para calon pengguna. Berikut ini isi dari 5 titik respon penilaian tersebut pada tabel 3.12.

Tabel 3.12 Penilaian Skala Likert

Pilihan Titik Respon	Nilai
A. Sangat Tidak Setuju	1 poin
B. Tidak Setuju	2 poin
C. Cukup	3 poin
D. Setuju	4 poin
E. Sangat Setuju	5 poin

3.6.1 Pengujian Oleh Nasabah

Untuk melaksanakan *usability testing* ini, dibutuhkan komposisi pertanyaan yang dapat mewakili aspek-aspek *usability testing*. Pertanyaan ini pun akan direspon dengan penguji dengan pilihan respon pada tabel 3.12. Dengan menggunakan skema penilaian skala likert, berikut ini adalah daftar pertanyaan beserta aspeknya yang akan digunakan untuk *usability testing* oleh nasabah pada tabel 3.13.

Tabel 3.13 Rancangan Komposisi Pernyataan *Usability Testing*

Aspek <i>Usability</i>	No	Pernyataan
<i>Usefulness</i>	1.1	Fitur pendaftaran <i>smart saving</i> atau <i>custom saving</i> memudahkan calon nasabah dalam melakukan pendaftaran menjadi nasabah baru.
	1.2	Fitur <i>qardh</i> atau pengadaan memudahkan nasabah untuk mengajukan <i>qardh</i> atau pengadaan.
	1.3	Fitur transaksi memudahkan nasabah dalam melakukan transaksi
	1.4	Fitur <i>history</i> transaksi memudahkan nasabah dalam melihat riwayat transaksi yang sudah dilakukan.
	1.5	Fitur FAQ menjawab kebingungan user dalam melakukan transaksi.

<i>Ease of Use</i>	2.1	Anda dapat mengakses aplikasi ini dengan mudah.
	2.2	Sistem navigasi antar halaman mudah dimengerti dan dilakukan.
	2.3	Fitur register dan login dapat digunakan dengan mudah.
	2.4	Fitur pendaftaran <i>smart saving</i> atau <i>custom saving</i> dapat digunakan dengan mudah.
	2.5	Fitur transaksi setor atau tarik dapat digunakan dengan mudah.
	2.6	Fitur pengajuan qardh dapat digunakan dengan mudah.
	2.7	Fitur pengajuan pengadaan dapat digunakan dengan mudah.
	2.8	Fitur FAQ dapat digunakan dengan mudah.
	2.9	Fitur <i>logout</i> dapat digunakan dengan mudah.
<i>Ease of Learning</i>	3.1	Tampilan menu sistem mudah dikenali oleh anda.
	3.2	Penyajian sub-menu jelas dan mudah dipahami.
	3.3	Penyajian informasi data transaksi jelas dan mudah dimengerti.
	3.4	Penyajian data FAQ jelas dan mudah dipahami.
<i>Satisfaction</i>	4.1	Tampilan warna sistem enak dilihat dan tidak membosankan.
	4.2	Anda akan menggunakan sistem ini untuk kedua kali dan seterusnya.

BAB IV

IMPLEMENTASI DAN PEMBAHASAN

Setelah proses analisis kebutuhan dan perancangan sistem selesai dibuat, berikutnya dilakukan implementasi sistem dalam bentuk pengkodean pada aplikasi yang akan dikembangkan. Desain dari sistem yang telah dibuat selanjutnya diterjemahkan ke dalam bahasa mesin (pengkodean) agar dapat dieksekusi oleh komputer. Implementasi dari sistem akan dibagi menjadi dua, yaitu implementasi pada API dan aplikasi android.

4.1 Instalasi dan Konfigurasi Laravel

Implementasi pengkodean untuk API menggunakan *framework* Laravel dengan basis data MySQL.

Sebelum melakukan instalasi Laravel, pastikan Composer sudah ter-*install* pada computer. Composer adalah alat manajemen dependency pada untuk bahasa pemrograman PHP. Composer memungkinkan untuk meng-*install* atau meng-*update* library pada project secara otomatis sehingga dapat mempermudah pengerjaan proyeknya. Untuk menginstal Laravel ketikkan potongan kode berikut pada *command prompt* :

```
composer global require "laravel/installer"
```

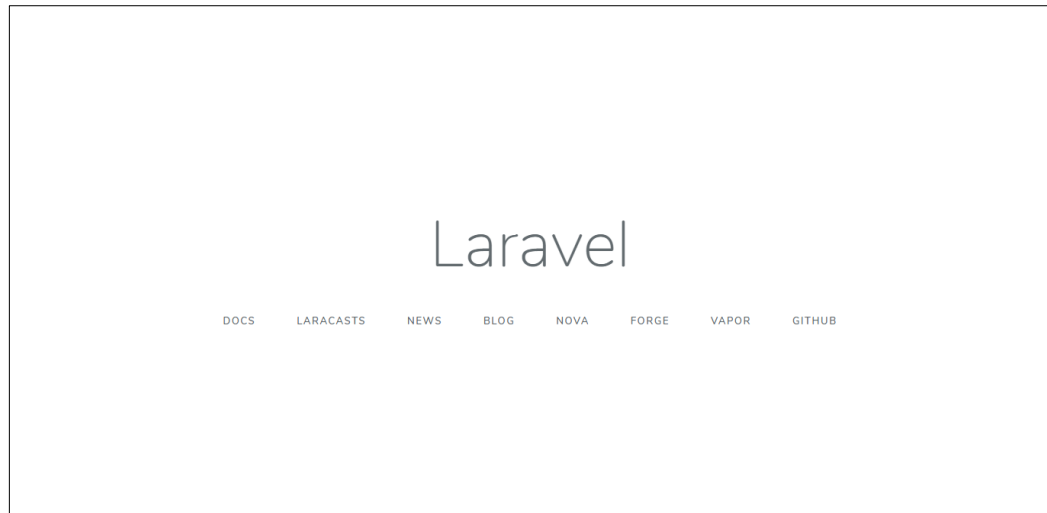
Kemudian buat sebuah direktori baru pada sistem untuk proyek Laravel baru. Setelah itu, pindah ke direktori baru yang telah dibuat dan ketik perintah berikut untuk menginstal Laravel:

```
composer create-project laravel/laravel bmt-admin --prefer-dist
```

Setelah membuat proyek Laravel baru, buka cmd pada lokasi direktori yang telah dibuat. Setelah itu gunakan perintah “php artisan serve” pada cmd untuk menjalankan Laravel pada *web browser*:

```
C:\xampp\htdocs\bmt-admin> php artisan serve
```

Jika Laravel berhasil ter-*install*, maka akan muncul tampilan seperti Gambar 4.1



Gambar 4.1 Tampilan Utama Laravel

4.2 Konfigurasi dan Migrasi Database

Langkah selanjutnya adalah melakukan pengaturan database yang akan digunakan, pengaturan ini dapat dilakukan melalui file `.env` yang terdapat pada direktori Laravel tersebut. Pengaturan yang dilakukan meliputi url aplikasi, jenis database, *host* dan *port* database, nama database, serta username dan password database. Database yang digunakan adalah bmt yang memiliki 9 (sembilan) tabel yang terdiri dari tabel *users*, tabel *nasabah_smart*, tabel *nasabah_custom*, tabel *pengadaan*, tabel *qardh*, tabel *transaksi_smart*, tabel *transaksi_custom*, tabel *angsur_qardh* dan tabel *angsur_pengadaan*.

```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:Qzgs5UGIvcFotlMXwkjjiFUmboTXb3WXEMX17wOhpk0=
APP_DEBUG=true
APP_URL=http://localhost

LOG_CHANNEL=stack

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=bmt
DB_USERNAME=root
DB_PASSWORD=
```


4.2.1 Migrasi Tabel User

Tabel users terdiri dari atribut *id*, *full_name*, *npm*, *email*, *password*, *created_at*, *updated_at*. Berikut adalah potongan kode program untuk migrasi tabel users:

```
Schema::create('users', function (Blueprint $table) {
    $table->bigIncrements('id');
    $table->string('name');
    $table->string('npm')->unique();
    $table->string('email')->unique();
    $table->string('password');
    $table->tinyInteger('have_smart')->default('0');
    $table->tinyInteger('have_custom')->default('0');
    $table->timestamps();
});
}
```

4.2.2 Migrasi Tabel Nasabah Smart Saving

Tabel smart saving terdiri dari atribut *id*, *sm_nama*, *sm_npm*, *sm_noacc*, *sm_nik*, *sm_balance*, *sm_birthp*, *sm_birtd*, *sm_gender*, *sm_religion*, *sm_raddress*, *sm_caddress*, *sm_phone*, *sm_phonep*, *sm_line*, *sm_status*, *user_id*, *created_at*, *updated_at*. Berikut adalah potongan kode program untuk migrasi tabel smart saving:

```
Schema::create('nasabah_smart', function (Blueprint $table) {
    $table->bigIncrements('id');
    $table->string('sm_name');
    $table->string('sm_npm')->unique();
    $table->string('sm_noacc')->unique()->nullable();
    $table->string('sm_nik')->unique();
    $table->integer('sm_balance')->default('0');
    $table->string('sm_birthp');
    $table->string('sm_birtd');
    $table->string('sm_gender');
    $table->string('sm_religion');
    $table->string('sm_raddress');
    $table->string('sm_caddress');
    $table->string('sm_phone');
    $table->string('sm_phonep');
    $table->string('sm_email');
    $table->string('sm_line');
    $table->integer('sm_status')->default('0');
    $table->unsignedBigInteger('user_id');
    $table->timestamps();

    $table->foreign('user_id')->references('id')->on('users');
});
```

4.2.3 Migrasi Tabel Nasabah Custom Saving

Tabel custom saving terdiri dari atribut *id*, *cu_nama*, *cu_npm*, *cu_noacc*, *cu_nik*, *cu_balance*, *cu_birthp*, *cu_birtd*, *cu_gender*, *cu_religion*, *cu_raddress*, *cu_caddress*, *cu_phone*, *cu_phonep*, *cu_line*, *cu_status*, *user_id*, *created_at*, *updated_at*. Berikut adalah potongan kode program untuk migrasi tabel custom saving:

```
Schema::create('nasabah_custom', function (Blueprint $table) {
    $table->bigIncrements('id');
    $table->string('cu_name');
    $table->string('cu_npm')->unique();
    $table->string('cu_noacc')->unique()->nullable();
    $table->string('cu_nik')->unique();
    $table->integer('cu_balance')->default('0');
    $table->string('cu_birthp');
    $table->string('cu_birtd');
    $table->string('cu_gender');
    $table->string('cu_religion');
    $table->string('cu_raddress');
    $table->string('cu_caddress');
    $table->string('cu_phone');
    $table->string('cu_phonep');
    $table->string('cu_email');
    $table->string('cu_line');
    $table->integer('cu_status')->default('0');
    $table->unsignedBigInteger('user_id');
    $table->timestamps();

    $table->foreign('user_id')->references('id')->on('users');
```

4.2.4 Migrasi Tabel Pengadaan

Tabel pengadaan terdiri dari atribut *id*, *pe_name*, *pe_npm*, *pe_nik*, *pe_ktm_img*, *pe_ktp_img*, *pe_raddress*, *pe_caddress*, *pe_phone*, *pe_phonep*, *pe_namep*, *pe_email*, *pe_line*, *pe_outcome*, *pe_income*, *pe_brg_nama*, *pe_alasan*, *pe_brg_tipe*, *pe_brg_spek*, *pe_brg_img*, *pe_brg_hrg*, *pe_lama*, *pe_sisa*, *pe_status*, *user_id*, *created_at*, *updated_at*. Berikut adalah potongan kode program untuk migrasi tabel pengadaan:

```
Schema::create('pengadaan', function (Blueprint $table) {
    $table->bigIncrements('id');
    $table->string('pe_name');
    $table->string('pe_npm')->unique();
    $table->string('pe_nik')->unique();
    $table->string('pe_ktm_img');
    $table->string('pe_ktp_img');
    $table->string('pe_raddress');
    $table->string('pe_caddress');
```

```

$table->string('pe_phone');
$table->string('pe_phonep');
$table->string('pe_namep');
$table->string('pe_email');
$table->string('pe_line');
$table->integer('pe_outcome');
$table->integer('pe_income');
$table->string('pe_brg_nama');
$table->string('pe_alasan');
$table->string('pe_brg_tipe');
$table->string('pe_brg_spek');
$table->string('pe_brg_img');
$table->integer('pe_brg_hrg')->default('0');
$table->integer('pe_lama');
$table->integer('pe_sisa')->default('0');
$table->integer('pe_status')->default('0');
$table->unsignedBigInteger('user_id');
$table->timestamps();

```

4.2.5 Migrasi Tabel Qardh

Tabel qardh terdiri dari atribut *id*, *qa_name*, *qa_npm*, *qa_nik*, *qa_ktm_img*, *qa_ktp_img*, *qa_raddress*, *qa_caddress*, *qa_phone*, *qa_phonep*, *qa_namep*, *qa_email*, *qa_line*, *qa_outcome*, *qa_income*, *qa_jml*, *qa_alasan*, *qa_lama*, *qa_sisa*, *qa_status*, *user_id*, *created_at*, *updated_at*. Berikut adalah potongan kode program untuk migrasi tabel qardh:

```

Schema::create('qardh', function (Blueprint $table) {
    $table->bigIncrements('id');
    $table->string('qa_name');
    $table->string('qa_npm')->unique();
    $table->string('qa_nik')->unique();
    $table->string('qa_ktm_img');
    $table->string('qa_ktp_img');
    $table->string('qa_raddress');
    $table->string('qa_caddress');
    $table->string('qa_phone');
    $table->string('qa_phonep');
    $table->string('qa_namep');
    $table->string('qa_email');
    $table->string('qa_line');
    $table->integer('qa_outcome');
    $table->integer('qa_income');
    $table->integer('qa_jml');
    $table->string('qa_alasan');
    $table->integer('qa_lama');
    $table->integer('qa_sisa');
    $table->integer('qa_status')->default('0');
    $table->unsignedBigInteger('user_id');
    $table->timestamps();

    $table->foreign('user_id')->references('id')->on('users');
});

```

4.2.6 Migrasi Tabel Transaksi Smart

Tabel *transaksi_smart* terdiri dari atribut *id*, *sm_code*, *sm_deposit*, *sm_withdraw*, *sm_trans_status*, *created_at*, *updated_at*. Berikut adalah potongan kode program untuk migrasi tabel *transaksi_smart*:

```
Schema::create('transaksi_smart', function (Blueprint $table) {
    $table->bigIncrements('id');
    $table->unsignedBigInteger('sm_code');
    $table->integer('sm_deposit');
    $table->integer('sm_withdraw');
    $table->tinyInteger('sm_trans_status')->default('0');
    $table->timestamps();

    $table->foreign('sm_code')->references('id')-
    >on('nasabah_smart');
});
```

4.2.7 Migrasi Tabel Transaksi Custom

Tabel *transaksi_smart* terdiri dari atribut *id*, *qa_code*, *qa_deposit*, *qa_withdraw*, *qa_trans_status*, *created_at*, *updated_at*. Berikut adalah potongan kode program untuk migrasi tabel *transaksi_custom*:

```
Schema::create('transaksi_custom', function (Blueprint $table) {
    $table->bigIncrements('id');
    $table->unsignedBigInteger('cu_code');
    $table->integer('cu_deposit');
    $table->integer('cu_withdraw');
    $table->tinyInteger('cu_trans_status')->default('0');
    $table->timestamps();

    $table->foreign('cu_code')->references('id')-
    >on('nasabah_custom');
});
```

4.2.8 Migrasi Tabel Angsur Pengadaan

Tabel *angsur_pengadaan* terdiri dari atribut *id*, *pe_code*, *pe_angsur_ke*, *pe_angsur_jml*, *pe_sisa*, *pe_angsur_status*, *created_at*, *updated_at*. Berikut adalah potongan kode program untuk migrasi tabel *angsur_pengadaan*:

```
Schema::create('angsur_pengadaan', function (Blueprint $table) {
    $table->bigIncrements('id');
    $table->unsignedBigInteger('pe_code');
    $table->integer('pe_angsur_ke');
    $table->integer('pe_angsur_jml');
    $table->unsignedInteger('pe_sisa');
    $table->tinyInteger('pe_angsur_status')->default('0');
    $table->timestamps();
});
```

```

        $table->foreign('pe_code')->references('id')->on('pengadaan');
    });

```

4.2.9 Migrasi Tabel Angsur Qardh

Tabel angsur pengadaan terdiri dari atribut *id*, *qa_code*, *qa_angsur_ke*, *qa_angsur_jml*, *qa_sisa*, *qa_angsur_status*, *created_at*, *updated_at*. Berikut adalah potongan kode program untuk migrasi tabel angsur_qardh:

```

Schema::create('angsur_qardh', function (Blueprint $table) {
    $table->bigIncrements('id');
    $table->unsignedBigInteger('qa_code');
    $table->integer('qa_angsur_ke');
    $table->integer('qa_angsur_jml');
    $table->unsignedInteger('qa_sisa');
    $table->tinyInteger('qa_angsur_status')->default('0');
    $table->timestamps();

    $table->foreign('qa_code')->references('id')->on('qardh');
});

```

4.3 Penulisan Kode Pemrograman untuk Model

Pada Model, pemrogram dapat menuliskan kode pemrograman untuk melakukan fungsi logika aplikasi terutama yang berhubungan dengan koneksi database.

4.3.1 Model User

Pada model User menggunakan tabel users yang memiliki atribut *full_name*, *email*, *npm*, *password*, *have_smart*, *have_custom*. Tabel users memiliki relasi dengan seluruh tabel rekening, maka dari itu digunakan perintah *Eloquent Relationship* pada tabel rekening. Kode program model User sebagai berikut:

```

class User extends Authenticatable
{
    use Notifiable, HasApiTokens;
    protected $fillable = [
        'name', 'npm', 'email', 'password'
    ];
    protected $hidden = [
        'password',
    ];
    public function smart(){
        return $this->hasOne(SmartSaving::class);
    }
    public function custom(){

```

```

        return $this->hasOne(CustomSaving::class);
    }
    public function pengadaan(){
        return $this->hasMany(Pengadaan::class);
    }
    public function qardh(){
        return $this->hasOne(Qardh::class);
    }
}

```

4.3.2 Model Smart Saving

Pada model Smart Saving menggunakan tabel nasabah_smart dan memiliki relasi dengan tabel users dan transaksi_smart, maka dari itu digunakan perintah *Eloquent Relationship* pada tabel-tabel tersebut. Kode program model SmartSaving sebagai berikut:

```

class SmartSaving extends Model
{
    protected $guarded = [];
    protected $table = 'nasabah_smart';

    public function user(){
        return $this->belongsTo(User::class);
    }
    public function transaksi(){
        return $this->hasMany(TransaksiSmartSaving::class);
    }
}

```

4.3.3 Model Custom Saving

Pada model Custom Saving menggunakan tabel nasabah_custom dan memiliki relasi dengan tabel users dan transaksi_custom, maka dari itu digunakan perintah *Eloquent Relationship* pada tabel-tabel tersebut. Kode program model Custom Saving sebagai berikut:

```

class CustomSaving extends Model
{
    protected $guarded = [];
    protected $table = 'nasabah_custom';
    public function user(){
        return $this->belongsTo(User::class);
    }
    public function transaksi(){
        return $this->hasMany(TransaksiCustomSaving::class);
    }
}

```

4.3.4 Model Transaksi Smart

Pada model TransaksiSmart menggunakan tabel transaksi_smart dan memiliki relasi dengan tabel nasabah_smart, maka dari itu digunakan perintah *Eloquent Relationship* pada nasabah_smart. Kode program model TransaksiSmart sebagai berikut:

```
class TransaksiSmartSaving extends Model
{
    protected $guarded =[];
    protected $table = 'transaksi_smart';

    public function tabungan()
    {
        return $this->belongsTo(SmartSaving::class);
    }
}
```

4.3.5 Model Transaksi Custom

Pada model TransaksiCustom menggunakan tabel transaksi_custom dan memiliki relasi dengan tabel nasabah_custom maka dari itu digunakan perintah *Eloquent Relationship* pada tabel-tabel tersebut. Kode program model TransaksiCustom sebagai berikut:

```
class TransaksiCustomSaving extends Model
{
    protected $guarded =[];
    protected $table = 'transaksi_custom';

    public function tabungan()
    {
        return $this->belongsTo(CustomSaving::class);
    }
}
```

4.3.6 Model Pengadaan

Pada model Pengadaan menggunakan tabel pengadaan dan memiliki relasi dengan tabel users dan angsur_pengadaan, maka dari itu digunakan perintah *Eloquent Relationship* pada tabel pinjam. Kode program model AngsurPinjam sebagai berikut:

```
class Pengadaan extends Model
{
    protected $guarded =[];
    protected $table = 'pengadaan';
}
```

```

    public function user()
    {
        return $this->belongsTo(User::class);
    }

    public function angsur()
    {
        return $this->hasMany(AngsurPengadaan::class);
    }
}

```

4.3.7 Model Angsur Pengadaan

Pada model AngsurPengadaan menggunakan tabel pengadaan dan memiliki relasi dengan tabel users dan angsur_pengadaan, maka dari itu digunakan perintah *Eloquent Relationship* pada tabel pinjam. Kode program model AngsurPinjam sebagai berikut:

```

class AngsurPengadaan extends Model
{
    protected $guarded = [];
    protected $table = 'angsur_pengadaan';
    public function pengadaan()
    {
        return $this->belongsTo(Pengadaan::class);
    }
}

```

4.3.8 Model Angsur Qardh

Pada model AngsurQardh menggunakan tabel qardh dan juga memiliki relasi dengan tabel users dan qardh, maka dari itu model ini menggunakan perintah *Eloquent Relationship* pada tabel qardh. Kode program model AngsurQardh sebagai berikut:

```

class AngsurQardh extends Model
{
    protected $guarded = [];
    protected $table = 'angsur_qardh';

    public function qardh()
    {
        return $this->belongsTo(Qardh::class);
    }
}

```


4.4 Implementasi Controller

Controller adalah penghubung antara view dan model. Pada controller dijalankan validasi inputan dan juga pengolahan data model ke dalam format json untuk ditampilkan view atau aplikasi. Berkas controller inilah yang dikerjakan oleh pemrogram dengan meng-extend dari class controller yang ada dan disimpan pada direktori ini. Potongan kode program pada AuthController untuk melakukan *register*, *login* dan *logout* adalah sebagai berikut:

```
public function register(Request $request)
{
    $validatedData = $request->validate([
        'name' => 'required',
        'email' => 'required|email|unique:users',
        'npm' => 'required|unique:users',
        'password' => 'required|min:6'
    ]);
    $validatedData['password'] = bcrypt($request->password);
    $user = User::create($validatedData);
    $accessToken = $user->createToken('authToken')->accessToken;
    return (new UserResource($user))->additional([
        'access_token' => $accessToken,
    ]);
}

public function login(Request $request)
{
    $loginData = $request->validate([
        'email' => 'required|email',
        'password' => 'required|min:6'
    ]);
    if (!auth()->attempt($loginData)) {
        return response(['error' => 'Your credential does not match']);
    }
}
```

```

    $currentUser = auth()->user();
    $accessToken = auth()->user()->createToken('authToken')->accessToken;

    return (new UserResource($currentUser))->additional([
        'access_token' => $accessToken,
    ]);

    return response()->json([
        'error' => 'Your credential does not match',
    ], 401);
}

public function logout (Request $request) {
    $accessToken = $request->user()->token();
    $accessToken->revoke();
    $response = 'You have been succesfully logged out!';
    return response($response, 200);
}
```

Potongan kode program pada NasabahController untuk melakukan registrasi nasabah dan menampilkan data nasabah adalah sebagai berikut:

```
public function smart(Request $request)
{
    $this->validate($request, [
        'sm_name' => 'required',
        'sm_npm' => 'required',
        'sm_nik' => 'required',
        'sm_birthp' => 'required',
        'sm_birthd' => 'required',
        'sm_gender' => 'required',
        'sm_religion' => 'required',
        'sm_raddress' => 'required',
        'sm_caddress' => 'required',
        'sm_phone' => 'required',
        'sm_phonep' => 'required',
        'sm_email' => 'required',
        'sm_line' => 'required',
    ]);

    $saving = SmartSaving::create([
        'sm_name' => $request->sm_name,
        'sm_npm' => $request->sm_npm,
        'sm_nik' => $request->sm_nik,
        'sm_birthp' => $request->sm_birthp,
        'sm_birthd' => $request->sm_birthd,
        'sm_gender' => $request->sm_gender,
        'sm_religion' => $request->sm_religion,
        'sm_raddress' => $request->sm_raddress,
        'sm_caddress' => $request->sm_caddress,
        'sm_phone' => $request->sm_phone,
        'sm_phonep' => $request->sm_phonep,
        'sm_email' => $request->sm_email,
        'sm_line' => $request->sm_line,
        'user_id' => auth()->id(),
    ]);

    DB::table('users')
        ->where('id', auth()->id())
        ->update(['have_smart' => 1]);

    return new SmartSavingResource($saving);
}

public function showSmart($user_id)
{
    $smart = SmartSaving::where('user_id', $user_id)->first();

    return new SmartSavingResource($smart);
}

public function custom(Request $request)
{
    $this->validate($request, [
        'cu_name' => 'required',
        'cu_npm' => 'required',
        'cu_nik' => 'required',
        'cu_birthp' => 'required',
    ];
```

```

        'cu_birthd' => 'required',
        'cu_gender' => 'required',
        'cu_religion' => 'required',
        'cu_raddress' => 'required',
        'cu_caddress' => 'required',
        'cu_phone' => 'required',
        'cu_phonep' => 'required',
        'cu_email' => 'required',
        'cu_line' => 'required',
    ));

    $saving = CustomSaving::create([
        'cu_name' => $request->cu_name,
        'cu_npm' => $request->cu_npm,
        'cu_nik' => $request->cu_nik,
        'cu_birthp' => $request->cu_birthp,
        'cu_birthd' => $request->cu_birthd,
        'cu_gender' => $request->cu_gender,
        'cu_religion' => $request->cu_religion,
        'cu_raddress' => $request->cu_raddress,
        'cu_caddress' => $request->cu_caddress,
        'cu_phone' => $request->cu_phone,
        'cu_phonep' => $request->cu_phonep,
        'cu_email' => $request->cu_email,
        'cu_line' => $request->cu_line,
        'user_id' => auth()->id(),
    ]);

    DB::table('users')
        ->where('id', auth()->id())
        ->update(['have_custom' => 1]);

    return new CustomSavingResource($saving);
}

public function showCustom($user_id)
{
    $custom = CustomSaving::where('user_id', $user_id)->first();

    return new CustomSavingResource($custom);
}

```

Potongan kode program pada TransaksiController untuk melakukan transaksi pada rekening tabungan. Berikut adalah potongan kodenya:

```

public function smart(Request $request)
{
    $this->validate($request, [
        'sm_deposit' => 'integer',
        'sm_withdraw' => 'integer',
    ]);

    $nasabah = DB::table('nasabah_smart')->where('user_id', auth()->id())->first();

    $transaksi = TransaksiSmartSaving::create([
        'sm_code' => $nasabah->id,
        'sm_deposit' => $request->sm_deposit,

```

```

        'sm_withdraw' => $request->sm_withdraw,
    ]]);

    DB::table('nasabah_smart')
        ->where('user_id', auth()->id())
        ->update(['sm_balance' => $nasabah->sm_balance + $request-
>sm_deposit - $request->sm_withdraw]);

    return new TransaksiSmartSavingResource($transaksi);
}

public function indexSmart()
{
    $nasabahSmart = DB::table('nasabah_smart')->where('user_id', auth()-
>id()->first());
    $transaksiSmart = DB::table('transaksi_smart')-
>where('sm_code', $nasabahSmart->id)->get();

    return TransaksiSmartSavingResource::collection($transaksiSmart);
}

public function custom(Request $request)
{
    $this->validate($request, [
        'cu_deposit' => 'integer',
        'cu_withdraw' => 'integer',
    ]);

    $nasabah = DB::table('nasabah_custom')->where('user_id', auth()-
>id()->first());

    $transaksi = TransaksiCustomSaving::create([
        'cu_code' => $nasabah->id,
        'cu_deposit' => $request->cu_deposit,
        'cu_withdraw' => $request->cu_withdraw,
    ]);

    DB::table('nasabah_custom')
        ->where('user_id', auth()->id())
        ->update(['cu_balance' => $nasabah->cu_balance + $request-
>cu_deposit - $request->cu_withdraw]);

    return new TransaksiCustomSavingResource($transaksi);
}

public function indexCustom()
{
    $nasabahCustom = DB::table('nasabah_custom')-
>where('user_id', auth()->id()->first());
    $transaksiCustom = DB::table('transaksi_custom')-
>where('cu_code', $nasabahCustom->id)->get();

    return TransaksiCustomSavingResource::collection($transaksiCustom);
}

```

Potongan kode program pada PinjamController untuk melakukan pengajuan peminjaman baik dalam bentuk pengadaan maupun qardh adalah sebagai berikut:

```

public function pengadaan(Request $request)
{
    $this->validate($request, [
        'pe_name' => 'required',
        'pe_npm' => 'required',
        'pe_nik' => 'required',
        'pe_ktm_img' => 'required',
        'pe_ktp_img' => 'required',
        'pe_raddress' => 'required',
        'pe_caddress' => 'required',
        'pe_phone' => 'required',
        'pe_phonep' => 'required',
        'pe_namep' => 'required',
        'pe_email' => 'required',
        'pe_line' => 'required',
        'pe_outcome' => 'required',
        'pe_income' => 'required',
        'pe_brg_nama' => 'required',
        'pe_alasan' => 'required',
        'pe_brg_tipe' => 'required',
        'pe_brg_spek' => 'required',
        'pe_brg_img' => 'required',
        'pe_lama' => 'required',
    ]);

    $user = DB::table('users')->where('id', auth()->id())->first();

    if ($request->hasFile('pe_ktm_img') && $request-
    >hasFile('pe_ktp_img') && $request->hasFile('pe_brg_img')) {

        $current = Carbon::now()->format('YmdHs');

        $fileKtm = $request->file('pe_ktm_img');
        $extensionKtm = $fileKtm->getClientOriginalExtension();
        $fileNameKtm = $user->id . $current . "_ktm." . $extensionKtm;
        $fileKtm->move('image/ktm/', $fileNameKtm);
        $ktmUrl = url('image/ktm/' . $fileNameKtm);

        $fileKtp = $request->file('pe_ktp_img');
        $extensionKtp = $fileKtp->getClientOriginalExtension();
        $fileNameKtp = $user->id . $current . "_ktp." . $extensionKtp;
        $fileKtp->move('image/ktp/', $fileNameKtp);
        $ktpUrl = url('image/ktp/' . $fileNameKtp);

        $fileBarang = $request->file('pe_brg_img');
        $extensionBarang = $fileBarang->getClientOriginalExtension();
        $fileNameBarang = time() . $current . "_" . $user-
        >id . "_barang." . $extensionBarang;
        $fileBarang->move('image/barang/', $fileNameBarang);
        $barangUrl = url('image/barang/' . $fileNameBarang);

        $pengadaan = Pengadaan::create([
            'pe_name' => $request->pe_name,
            'pe_npm' => $request->pe_npm,
            'pe_nik' => $request->pe_nik,
            'pe_ktm_img' => $ktmUrl,
            'pe_ktp_img' => $ktpUrl,
            'pe_brg_img' => $barangUrl,
            'pe_lama' => $request->pe_lama,
        ]);
    }
}

```

```

        'pe_raddress' => $request->pe_raddress,
        'pe_caddress' => $request->pe_caddress,
        'pe_phone' => $request->pe_phone,
        'pe_phonep' => $request->pe_phonep,
        'pe_namep' => $request->pe_namep,
        'pe_email' => $request->pe_email,
        'pe_line' => $request->pe_line,
        'pe_outcome' => $request->pe_outcome,
        'pe_income' => $request->pe_income,
        'pe_brg_nama' => $request->pe_brg_nama,
        'pe_alasan' => $request->pe_alasan,
        'pe_brg_tipe' => $request->pe_brg_tipe,
        'pe_brg_spek' => $request->pe_brg_spek,
        'pe_brg_img' => $barangUrl,
        'pe_lama' => $request->pe_lama,
        'user_id' => auth()->id(),
    ]);

    return new PengadaanResource($pengadaan);
} else {
    return response()->json([
        'message' => "Masukkan gambar"
    ]);
}
}

public function showPengadaan($user_id)
{
    $pengadaan = Pengadaan::where('user_id', $user_id)->first();

    return new PengadaanResource($pengadaan);
}

public function qardh(Request $request)
{
    $this->validate($request, [
        'qa_name' => 'required',
        'qa_npm' => 'required',
        'qa_nik' => 'required',
        'qa_ktm_img' => 'required',
        'qa_ktp_img' => 'required',
        'qa_raddress' => 'required',
        'qa_caddress' => 'required',
        'qa_phone' => 'required',
        'qa_phonep' => 'required',
        'qa_namep' => 'required',
        'qa_email' => 'required',
        'qa_line' => 'required',
        'qa_outcome' => 'required',
        'qa_income' => 'required',
        'qa_jml' => 'required',
        'qa_alasan' => 'required',
        'qa_lama' => 'required',
    ]);

    $user = DB::table('users')->where('id', auth()->id())->first();

```

```

        if ($request->hasFile('qa_ktm_img') && $request->hasFile('qa_ktp_img')) {

            $fileKtm = $request->file('qa_ktm_img');
            $extensionKtm = $fileKtm->getClientOriginalExtension();
            $fileNameKtm = $user->id . "_ktm." . $extensionKtm;
            $fileKtm->move('image/ktm/', $fileNameKtm);
            $ktmUrl = url('image/ktm/' . $fileNameKtm);

            $fileKtp = $request->file('qa_ktp_img');
            $extensionKtp = $fileKtp->getClientOriginalExtension();
            $fileNameKtp = $user->id . "_ktp." . $extensionKtp;
            $fileKtp->move('image/ktp/', $fileNameKtp);
            $ktpUrl = url('image/ktp/' . $fileNameKtp);

            $qardh = Qardh::create([
                'qa_name' => $request->qa_name,
                'qa_npm' => $request->qa_npm,
                'qa_nik' => $request->qa_nik,
                'qa_ktm_img' => $ktmUrl,
                'qa_ktp_img' => $ktpUrl,
                'qa_raddress' => $request->qa_raddress,
                'qa_caddress' => $request->qa_caddress,
                'qa_phone' => $request->qa_phone,
                'qa_phonep' => $request->qa_phonep,
                'qa_namep' => $request->qa_namep,
                'qa_email' => $request->qa_email,
                'qa_line' => $request->qa_line,
                'qa_outcome' => $request->qa_outcome,
                'qa_income' => $request->qa_income,
                'qa_alasan' => $request->qa_alasan,
                'qa_lama' => $request->qa_lama,
                'qa_jml' => $request->qa_jml,
                'qa_sisa' => $request->qa_jml,
                'user_id' => auth()->id(),
            ]);

            return new QardhResource($qardh);
        } else {
            return response()->json([
                'message' => "Masukkan gambar"
            ]);
        }
    }

    public function showQardh($user_id)
    {
        $qardh = Qardh::where('user_id', $user_id)->first();

        return new QardhResource($qardh);
    }

```

Potongan kode program pada AngsuranController untuk membayar angsuran dari peminjaman yang dimiliki adalah sebagai berikut:

```

public function pengadaan(Request $request)
{
    $this->validate($request, [

```

```

        'pe_angsur_jml' => 'required|integer',
    ]));

    $pengadaan = DB::table('pengadaan')->where('user_id', auth()->id())->first();

    $angsur = DB::table('angsur_pengadaan')->where('pe_code', $pengadaan->id)->latest()->first();

    if ($angsur != null) {
        $angsurBaru = AngsurPengadaan::create([
            'pe_code' => $pengadaan->id,
            'pe_angsur_jml' => $request->pe_angsur_jml,
            'pe_angsur_ke' => $angsur->pe_angsur_ke + 1,
            'pe_sisa' => $pengadaan->pe_sisa - $request->pe_angsur_jml
        ]);
    } else {
        $angsurBaru = AngsurPengadaan::create([
            'pe_code' => $pengadaan->id,
            'pe_angsur_jml' => $request->pe_angsur_jml,
            'pe_angsur_ke' => 1,
            'pe_sisa' => $pengadaan->pe_sisa - $request->pe_angsur_jml
        ]);
    }

    DB::table('pengadaan')
        ->where('id', $pengadaan->id)
        ->update(['pe_sisa' => $pengadaan->pe_sisa - $request->pe_angsur_jml]);

    return new AngsurPengadaanResource($angsurBaru);
}

public function indexPengadaan()
{
    $nasabahPengadaan = DB::table('pengadaan')->where('user_id', auth()->id())->first();
    $angsurPengadaan = DB::table('angsur_pengadaan')->where('pe_code', $nasabahPengadaan->id)->get();

    return AngsurPengadaanResource::collection($angsurPengadaan);
}

public function qardh(Request $request)
{
    $this->validate($request, [
        'qa_angsur_jml' => 'required|integer',
    ]);

    $qardh = DB::table('qardh')->where('user_id', auth()->id())->first();

    $angsur = DB::table('angsur_qardh')->where('qa_code', $qardh->id)->latest()->first();

    if ($angsur != null) {
        $angsurBaru = AngsurQardh::create([
            'qa_code' => $qardh->id,
            'qa_angsur_jml' => $request->qa_angsur_jml,
            'qa_angsur_ke' => $angsur->qa_angsur_ke + 1,
            'qa_sisa' => $qardh->qa_sisa - $request->qa_angsur_jml
        ]);
    }
}

```



```

        'qa_sisa' => $qardh->qa_sisa - $request->qa_angsur_jml
    ]]);
} else {
    $angsurBaru = AngsurQardh::create([
        'qa_code' => $qardh->id,
        'qa_angsur_jml' => $request->qa_angsur_jml,
        'qa_angsur_ke' => 1,
        'qa_sisa' => $qardh->qa_sisa - $request->qa_angsur_jml
    ]]);

    DB::table('qardh')
        ->where('id', $qardh->id)
        ->update(['qa_sisa' => $qardh->qa_sisa - $request->qa_angsur_jml]);

    return new AngsurQardhResource($angsurBaru);
}

public function indexQardh()
{
    $nasabahQardh = DB::table('qardh')->where('user_id', auth()->id())->first();
    $angsurQardh = DB::table('angsur_qardh')->where('qa_code', $nasabahQardh->id)-
>get();

    return AngsurQardhResource::collection($angsurQardh);
}

```

4.5 Implementasi Route

Route merupakan bagian yang akan menangani *request* dari aplikasi lalu diteruskan ke *endpoint* terkait. Route akan mengarahkan sistem untuk memanggil *resource* tertentu. Singkatnya, *route* bekerja untuk menerima lalu meneruskan *request*. Berikut adalah sedikit potongan dari *endpoint* yang ada pada route.

```

. . .
Route::post('/register', 'Api\AuthController@register');
Route::post('/login', 'Api\AuthController@login');
Route::middleware('auth:api')->group(function () {
    Route::get('/logout', 'Api\AuthController@logout')->name('logout');
});
. . .

```

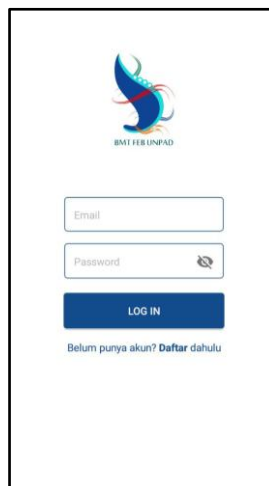
4.6 Implementasi Tampilan

Implementasi perancangan tampilan antar muka (interface) pada sistem adalah peranan yang penting, karena pada tahap ini dijelaskan bagaimana cara pengguna menggunakan sistem yang telah dibuat. Implementasi pengkodean dilakukan setelah proses analisis kebutuhan dan desain sistem selesai dibuat. Desain

sistem yang sudah ada selanjutnya diterjemahkan ke dalam bahasa mesin (pengkodean) agar dapat dieksekusi oleh komputer. Implementasi pengkodean untuk aplikasi ini menggunakan *framework android native* dengan bahasa pemrograman Java dan XML, serta menggunakan MySQL/PostgreSQL untuk basis data-nya.

4.6.1 Tampilan Halaman Login

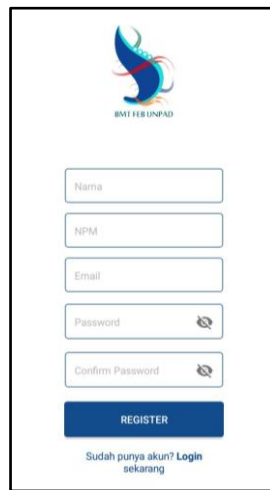
Halaman ini ditujukan untuk pengguna yang telah memiliki akun, user perlu memasukkan email dan kata sandi dari akun masing-masing untuk melakukan login. Halaman ini merupakan halaman yang pertama ditampilkan setelah splashscreen ketika membuka aplikasi. Implementasi tampilan untuk halaman login dapat dilihat pada Gambar 4.2.



Gambar 4.2 Tampilan halaman login

4.6.2 Tampilan halaman register

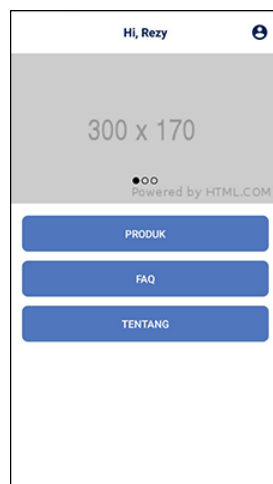
Halaman ini ditujukan untuk calon pengguna yang belum memiliki akun. Calon pengguna perlu memasukkan nama, npm email dan kata sandi untuk melakukan registrasi. Halaman ini dapat diakses melalui halaman login dengan cara menekan tulisan “Daftar sekarang”. Implementasi tampilan untuk halaman register dapat dilihat pada Gambar 4.3.



Gambar 4.3 Tampilan halaman register

4.6.3 Tampilan halaman utama

Implementasi tampilan untuk halaman utama dapat dilihat pada Gambar 4.4. Halaman ini merupakan halaman pertama yang akan ditampilkan setelah user melakukan login. Pada halaman ini user dapat melakukan navigasi ke fitur-fitur yang tersedia pada aplikasi.

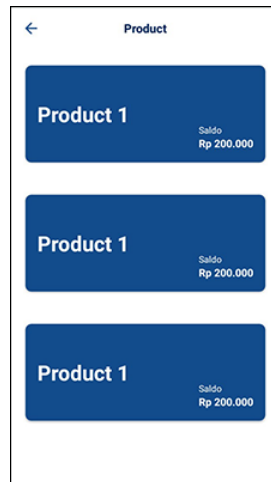


Gambar 4.4 Tampilan halaman utama

4.6.4 Tampilan halaman produk

Pada halaman ini, user dapat melihat produk yang terdapat pada aplikasi. Apabila user sudah terdaftar pada salah satu produk, maka user bisa

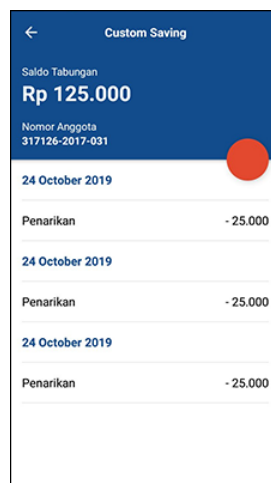
melihat saldo yang ia miliki pada produk tersebut. Tampilan pada halaman produk dapat dilihat pada Gambar 4.5.



Gambar 4.5 Tampilan halaman produk

4.6.5 Tampilan halaman riwayat

Pada halaman ini user dapat melihat riwayat transaksi yang pernah dilakukan dan dapat melakukan transaksi baru. Implementasi tampilan untuk halaman riwayat transaksi dapat dilihat pada Gambar 4.6.



Gambar 4.6 Tampilan halaman riwayat

4.6.6 Tampilan halaman jenis transaksi

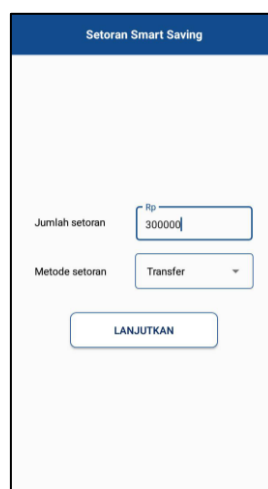
Pada halaman ini user akan diminta untuk memilih jenis transaksi dari produk yang akan dilakukan. Implementasi tampilan untuk halaman riwayat transaksi dapat dilihat pada Gambar 4.7.



Gambar 4.7 Tampilan halaman jenis transaksi

4.6.7 Tampilan halaman transaksi

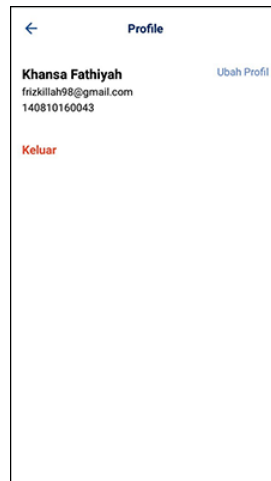
Pada halaman ini user akan diminta untuk memasukkan nominal dan metode dari transaksi yang akan dilakukan. Implementasi tampilan untuk halaman transaksi dapat dilihat pada Gambar 4.8.



Gambar 4.8 Tampilan halaman transaksi

4.6.8 Tampilan halaman profil

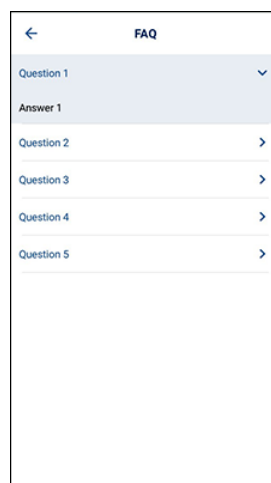
Pada halaman ini user dapat melihat data dari profil mereka, di halaman ini juga user dapat melakukan navigasi ke halaman edit profil serta melakukan logout. Implementasi tampilan untuk halaman profil dapat dilihat pada Gambar 4.9.



Gambar 4.9 Tampilan halaman profil

4.6.9 Tampilan halaman FAQ

Pada halaman ini user dapat melihat halaman FAQ yang berisi pertanyaan dan jawaban jika menemukan kesulitan dalam melakukan transaksi. Implementasi tampilan untuk halaman FAQ dapat dilihat pada Gambar 4.10.



Gambar 4.10 Tampilan Halaman FAQ

4.6.10 Tampilan halaman about

Pada halaman ini user dapat melihat halaman about yang berisi sejarah singkat dan latar belakang dari BMT FEB Unpad. Implementasi tampilan untuk halaman about dapat dilihat pada Gambar 4.11.



Gambar 4.11 Tampilan Halaman About

4.7 Pengujian Aplikasi

Setelah aplikasi selesai dibuat, maka penguji melakukan tiga pengujian, yaitu pengujian usabilitas terhadap calon pengguna atau nasabah dengan menggunakan kuisioner agar mendapat tanggapan setelah fitur FAQ tersebut diperbaharui.

4.7.1 Pengujian Usabilitas Kepada Nasabah

Pada proses pengujian ini, penguji akan memberikan link untuk mengunduh aplikasi di ponsel *android* yang dimiliki sebelum nasabah dapat menilai aspek-aspek dari aplikasi sesuai dari aspek pengujian usabilitas yang sudah dibuat. Hasil dari pengujian ini diharapkan dapat menjawab atau menjadi solusi permasalahan yang dicantumkan penguji pada identifikasi masalah. Pengujian ini dilakukan kepada 30 nasabah BMT FEB Unpad. Hasil dari kuesioner pengujian dapat dilihat pada Tabel 4.1.

Tabel 4.1 Hasil Penilaian Kuisisioner Pengguna

Aspek <i>Usability</i>	No	Pernyataan	Nilai				
			1	2	3	4	5
<i>Usefulness</i>	1.1	Fitur pendaftaran <i>smart saving</i> atau <i>custom saving</i> memudahkan calon nasabah dalam melakukan pendaftaran menjadi nasabah baru.					
	1.2	Fitur qardh atau pengadaan memudahkan nasabah untuk mengajukan qardh atau pengadaan.					
	1.3	Fitur transaksi memudahkan nasabah dalam melakukan transaksi					
	1.4	Fitur <i>history</i> transaksi memudahkan nasabah dalam melihat riwayat transaksi yang sudah dilakukan.					
	1.5	Fitur FAQ menjawab kebingungan user dalam melakukan transaksi.					
<i>Ease of Use</i>	2.1	Anda dapat mengakses aplikasi ini dengan mudah.					
	2.2	Sistem navigasi antar halaman mudah dimengerti dan dilakukan.					
	2.3	Fitur register dan login dapat digunakan dengan mudah.					
	2.4	Fitur pendaftaran <i>smart saving</i> atau <i>custom saving</i> dapat digunakan dengan mudah.					
	2.5	Fitur transaksi setor atau tarik dapat digunakan dengan mudah.					

	2.6	Fitur pengajuan qardh dapat digunakan dengan mudah.					
	2.7	Fitur pengajuan pengadaan dapat digunakan dengan mudah.					
	2.8	Fitur FAQ dapat digunakan dengan mudah.					
	2.9	Fitur <i>logout</i> dapat digunakan dengan mudah.					
<i>Ease of Learning</i>	3.1	Tampilan menu sistem mudah dikenali oleh anda.					
	3.2	Penyajian sub-menu jelas dan mudah dipahami.					
	3.3	Penyajian informasi data transaksi jelas dan mudah dimengerti.					
	3.4	Penyajian data FAQ jelas dan mudah dipahami.					
<i>Satisfaction</i>	4.1	Tampilan warna sistem enak dilihat dan tidak membosankan.					
	4.2	Anda akan menggunakan sistem ini untuk kedua kali dan seterusnya.					

Dari kuisioner tersebut, terdapat lima tingkatan jawaban kuisioner yang digunakan. Untuk hasil penilaian dari kuesioner tersebut dapat kita hitung dengan menentukan interval dan kriteria nilai terlebih dahulu dengan menggunakan persamaan di bawah:

$$Interval = \frac{100}{5} = 20$$

Sehingga didapatkan kriteria nilai untuk pengujian ini dapat dilihat pada Tabel 4.2:

Tabel 4.2 Kriteria Nilai

Jawaban	Keterangan
0 – 19.99	Sangat buruk
20 – 39.99	Buruk
40 – 59.99	Cukup
60 – 79.99	Baik
80 – 100.0	Sangat Baik

Aspek <i>usabilitiy</i>	No	Nilai					Total	Persentase (%)
		1	2	3	4	5		
<i>Usefulness</i>	1.1							
	1.2							
	1.3							
	1.4							
	1.5							
	Total Persentase							
<i>Ease of Use</i>	2.1							
	2.2							
	2.3							
	2.4							
	2.5							
	2.6							
	2.7							
	2.8							
	2.9							
	Total Persentase							

<i>Ease of Learning</i>	3.1							
	3.2							
	3.3							
	3.4							
	Total Persentase							
<i>Satisfaction</i>	4.1							
	4.2							
	Total Persentase							

Dari hasil pengujian yang ditampilkan pada tabel di atas, maka dapat disimpulkan hasil kuisioner melalui tabel kesimpulan seperti pada Tabel 4.4:

Tabel 4.4 Tabel Kesimpulan Pengujian

No	Aspek	Total Persentase (%)
1	<i>Usefulness</i>	
2	<i>Ease of Use</i>	
3	<i>Ease of Learning</i>	
4	<i>Satisfaction</i>	
Total Keseluruhan		

Dari hasil *usability testing* di atas serta sesuai dengan nilai interval skala likert yaitu **xx,xx%** maka aplikasi Baitul Maal wa Tamwil FEB Unpad ini dalam interval sangat baik, yang artinya responden sudah memahami penggunaan aplikasi secara fungsional serta mendapatkan kepuasan yang baik terutama dalam mengatasi permasalahan nasabah.

BAB V

SIMPULAN DAN SARAN

5.1 Simpulan

5.2 Saran

DAFTAR PUSTAKA

- Codepath. (2019). *Extensive Open-Source Guides for Android Developers*. Retrieved from https://github.com/codepath/android_guides
- Delima, R., Santoso, H., & Purwadi, J. (2017). Development of Dutatani Website Using Rapid Application Development. *IJITEE (International Journal of Information Technology and Electrical Engineering)*, 1, 36-44.
- Duy, T. B. (2017). Reactive Programming and Clean Architecture in Android Development.
- Fakhriza, M., & Siambaton, M. (2016). Aplikasi Content Management System (Cms) Pada Joomla Untuk Membuat Web Service. *InfoTekJar (Jurnal Nasional Informatika dan Teknologi Jaringan)*, 11-13.
- Gata, W., & Gata, G. (2013). *Sukses Membangun Aplikasi Penjualan dengan Java*. Jakarta: Elex Media Komputindo.
- Hidayah, D. (2017). Simulasi Aplikasi Kalkulator Resistor Berbasis Android.
- Holla, S., & Katti, M. M. (2012). Android Based Mobile Application Development and Its Security. *International Journal of Computer Trends and Technology*, 3(3), 486-490.
- Kristanto, A. (2010). *Kupas Tuntas PHP & MYSQL*. Klaten: Cable Book.
- Martin, R. C. (2012, August 13). The Clean Architecture [online]. Retrieved from <https://8thlight.com/blog/uncle-bob/2012/08/13/the-clean-architecture.html>
- Naista, D. (2016). *Bikin Framework PHP Sendiri dengan Teknik OOP dan MVC*. Jakarta: Lokomedia.
- Otwell, Taylor; Laravel LLC. (2019). *Laravel Documentation*. Available at: <https://laravel.com/docs/6.x>
- Rama, G. M., & Avinash Kak. (2013). *Software – Practice and Experience*. New Jersey: Wiley Online Library.

- Ropianto, M. (2016, Oktober). Pemahaman Penggunaan Unified Modelling Language. *JT-IBSI*.
- Rusby, Z., Hamzah, Z., & Hamzah. (2018, January). Analisa Permasalahan Baitul Maal Wat Tamwil (BMT) melalui Pendekatan Analytical Network Process (ANP). *Al-Hikmah: Jurnal Agama dan Ilmu Pengetahuan, [S.l.], n. 1., 13*, 18-29.
- Sa'roni, L., & Hosen, M. (2012, August). Determinant Factors of the Successful of Baitul Maal Wat Tamwil (BMT). *International Journal of Academic Research in Economics and Management Sciences, 1*, 36-55.
- Square, I. (2013). *Retrofit, Type-safe HTTP client for Android and Java*. Retrieved from <https://github.com/square/retrofit>
- Widodo, B., & Purnomo, H. (2016). Perancangan Aplikasi Pencarian Layanan Kesehatan Berbasis HTML 5 Geolocation. *Jurnal Sistem Komputer, 6*, 44-51.