

**IMPLEMENTASI SCRUM DALAM PERANCANGAN
FRONTEND APLIKASI M-BANKING
DI BANK ABC JAWA BARAT**

SKRIPSI



Disusun Oleh :

Awal Ariansyah

NIM : 195520019

**STMIK KOMPUTAMA
MAJENANG**

**PROGRAM STUDI TEKNIK INFORMATIKA
STMIK KOMPUTAMA MAJENANG**

2023

**IMPLEMENTASI SCRUM DALAM PERANCANGAN
FRONTEND APLIKASI M-BANKING
DI BANK ABC JAWA BARAT**

SEMBARU

SKRIPSI

Untuk memenuhi sebagian persyaratan mencapai derajat Sarjana S1
pada Program Studi Teknik Informatika

Disusun Oleh :
Awal Ariansyah
NIM : 195520019

**STMIK KOMPUTAMA
MAJENANG**

**PROGRAM STUDI TEKNIK INFORMATIKA
STMIK KOMPUTAMA MAJENANG**

2023

LEMBAR PENGESAHAN

SKRIPSI

**IMPLEMENTASI SCRUM DALAM PERANCANGAN
FRONTEND APLIKASI M-BANKING
DI BANK ABC JAWA BARAT**

Disusun Oleh :
Awal Ariansyah
NIM : 195520019

Telah disetujui Oleh Dosen Pembimbing Skripsi
Pada tanggal 31 Juli 2023

Pembimbing



Nur Muniroh, S.T., M.Kom.
NIDN. : 0611119202


Skripsi ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana Komputer (S.Kom)
Tanggal 31 Juli 2023

Wakil Ketua I
STMIK Komputama Majenang



Much Solehudin, M.Pd.
NIDN. 0622089201

Ketua Program Studi
Teknik Informatika



Nur Muniroh, S.T., M.Kom.
NIDN. 0611119202

PENGESAHAN PENGUJI

SKRIPSI

**IMPLEMENTASI SCRUM DALAM PERANCANGAN
FRONTEND APLIKASI M-BANKING
DI BANK ABC JAWA BARAT**

**Disusun Oleh :
Awal Ariansyah
NIM. : 195520019**

Telah dipertahankan di depan dewan Penguji
Pada tanggal 31 Juli 2023

Penguji 1



Beny Riswanto, M.Kom.

NIDN. 0619029001

Penguji 2



Nur Muniroh, S.T., M.Kom.

NIDN. 0611119202

Skripsi ini telah disahkan sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana Komputer (S. Kom)
Tanggal : 31 Juli 2023

**Mengetahui
Wakil Ketua I Bidang Akademik**



Much Solehudin, M.Pd.

NIDN. 0622089201

PENGESAHAN PEMBIMBING INSTITUSI

SKRIPSI



**IMPLEMENTASI SCRUM DALAM PERANCANGAN
FRONTEND APLIKASI M-BANKING
DI BANK ABC JAWA BARAT**

Disusun Oleh :

Awal Ariansyah

NIM : 195520019

Cilacap, 30 Juni 2023

Mengetahui

Kepala/Pimpinan Institusi

Menyetujui

Pembimbing Institusi

Dian R Bastian
Project Owner Bank ABC

Ferry Mohammad Ferdian
Project Manager Bank ABC

**STMIK KOMPUTAMA
MAJENANG**





**PROGRAM STUDI TEKNIK INFORMATIKA
STMIK KOMPUTAMA MAJENANG**

**LEMBAR PERNYATAAN KEASLIAN
SKRIPSI**

Saya yang bertanda tangan di bawah ini :

Nama : Awal Ariansyah

NIM : 195520019

Program Studi : Teknik Informatika

Menyatakan bahwa Skripsi yang berjudul / Tema* :

**Implementasi *Scrum* Dalam Perancangan *Frontend* Aplikasi M-Banking Di
Bank ABC Jawa Barat**

Benar-benar merupakan hasil karya pribadi dan bukan merupakan hasil karya orang lain atau pihak mana pun, serta **BUKAN PLAGIAT**. Seluruh sumber yang dijadikan rujukan dan dikutip dalam laporan Skripsi ini telah saya nyatakan dengan benar. Apabila dikemudian hari pernyataan saya ini tidak benar, maka saya bersedia menanggung semua akibat atau sanksi yang berlaku.

Cilacap, 21 Juli 2023



Awal Ariansyah

NIM. 195520019

KATA PENGANTAR

Kata Pertama-tama, penulis menyampaikan puji syukur kepada Tuhan Yang Maha Esa atas limpahan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul “Implementasi *Scrum* Dalam Perancangan *Frontend* Aplikasi M-Banking Di Bank ABC Jawa Barat”.

Ucapan terima kasih penulis sampaikan kepada pihak yang membantu penyelesaian skripsi ini baik secara langsung maupun tidak langsung, kepada:

1. Bapak Dr. KH Fathul Aminudin Aziz, M.M., selaku ketua yayasan El - Bayan Majenang.
2. Bapak Ahmad Irfangi, M.M., selaku ketua STMIK Komputama Majenang.
3. Ibu Nur Muniroh, S.T., M.Kom., selaku dosen pembimbing dan Kaprodi Teknik Informatika STMIK Komputama Majenang yang telah memberikan masukan dalam penyelesaian skripsi ini.
4. Orang tua beserta keluarga besar Wangsa Witana yang telah mendukung dan mendoakan selama proses penyelesaian skripsi ini.
5. Rekan-rekan mahasiswa STMIK Komputama Majenang angkatan 2019 yang telah berjuang bersama-sama.
6. Ibnu, Danu, Ivan, Omar, Vitri, Lana dan teman-teman daring lainnya yang senantiasa berbagi humor, ilmu dan motivasi.

Segala bentuk saran dan kritik sangat diterima penulis untuk perbaikan yang lebih baik lagi di masa depan.

Cilacap, 31 Juli 2023



Awal Ariansyah
NIM. : 195520019

ABSTRACT

Banking is an important sector in people's lives. The increase in digital transaction frequency by internet users and digitalization has forced the banking sector to adapt by developing m-banking services. Bank ABC is a Regional Owned Enterprise (BUMD) bank in one of the districts in West Java with about 160,000 customers who do not have m-banking services. A m-banking application frontend is needed to integrate business processes from Bank ABC's internal system backend that is already running. The research starts from problem identification, literature study, data collection, designing with scrum implementation, and testing. The scrum method is used in designing which is suitable for rapid development with three main actors: the product owner, the scrum master, and the developer. The scrum stages are the product backlog, sprint planning, sprint backlog, sprint, daily scrum, sprint review, increment, and sprint retrospective. Data collection was conducted using interview techniques to produce primary data, as well as observation and literature studies to produce secondary data. Scrum implementation starts with sprint planning, executing four sprints within four weeks, and testing. Testing is carried out using heuristic evaluation to measure the usability of the application and resulting in Errors aspect for the displayed error messages with consistency language. The research resulted in an initial m-banking application stage that digitalize the business process as well as integrated with currently running system of Bank ABC West Java.

Keywords: Banking, Scrum, Android, iOS, Frontend, Mobile Banking.

STMIK KOMPUTAMA
MAJENANG

ABSTRAK

Perbankan merupakan sektor penting dalam kehidupan masyarakat. Meningkatnya frekuensi transaksi digital pengguna internet dan digitalisasi membuat perbankan harus beradaptasi dengan mengembangkan layanan m-banking. Bank ABC merupakan sebuah Bank Badan Usaha Milik Daerah (BUMD) di salah satu kabupaten di Jawa Barat dengan jumlah nasabah sekitar 160.000 nasabah yang belum memiliki layanan m-banking. Sebuah *frontend* aplikasi m-banking diperlukan untuk mengintegrasikan proses bisnis dari *backend* sistem internal Bank ABC yang sudah berjalan. Penelitian dimulai dari identifikasi masalah, studi literatur, pengumpulan data, perancangan dengan implementasi *scrum*, dan pengujian. Metode *scrum* digunakan dalam pengembangannya yang cocok untuk pengembangan cepat dengan tiga aktor utama yaitu *product owner*, *scrum master* dan *developer*. Tahapan-tahapan *scrum* yaitu *product backlog*, *sprint planning*, *sprint backlog*, *sprint*, *daily scrum*, *sprint review*, *increment*, dan *sprint retrospective*. Pengumpulan data dilakukan menggunakan teknik wawancara untuk menghasilkan data primer, serta observasi dan studi literatur untuk menghasilkan data sekunder. Implementasi *scrum* dimulai dari *sprint planning*, eksekusi *sprint* sebanyak 4 kali dalam 4 minggu, serta pengujian. Pengujian dilakukan menggunakan evaluasi heuristik untuk mengukur aspek kegunaan dari aplikasi dan menghasilkan rekomendasi pada aspek *Errors* untuk konsistensi bahasa pesan kesalahan yang ditampilkan. Penelitian menghasilkan tahap awal aplikasi m-banking yang mendigitalisasi proses bisnis serta terintegrasi dengan sistem yang berjalan di Bank ABC Jawa Barat.

Kata kunci: Perbankan, *Scrum*, Android, iOS, *Frontend*, *Mobile Banking*

STMIK KOMPUTAMA
MAJENANG

DAFTAR ISI

LEMBAR JUDUL.....	ii
LEMBAR PENGESAHAN.....	iii
PENGESAHAN PENGUJI.....	iv
PENGESAHAN PEMBIMBING INSTITUSI.....	v
LEMBAR PERNYATAAN KEASLIAN.....	vi
KATA PENGANTAR.....	vii
ABSTRACT.....	viii
ABSTRAK.....	ix
DAFTAR ISI.....	x
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xiv
DAFTAR LAMPIRAN.....	xvi
BAB I PENDAHULUAN.....	1
A. Latar Belakang Masalah.....	1
B. Rumusan Masalah.....	3
C. Batasan Masalah.....	4
D. Tujuan Penelitian.....	5
E. Manfaat Penelitian.....	5
F. Sistematika Penulisan.....	6
BAB II TINJAUAN PUSTAKA.....	8
A. Landasan Teori.....	8
B. Tinjauan Terdahulu.....	25
BAB III METODOLOGI PENELITIAN.....	29
A. Tempat dan Waktu Penelitian.....	29
B. Bahan dan Alat Penelitian.....	29
C. Metode Pengumpulan Data.....	30
D. Konsep Penelitian.....	30
BAB IV HASIL DAN PEMBAHASAN.....	41
A. Pembentukan <i>Product Backlog</i>	41
B. <i>Sprint</i> I.....	51
C. <i>Sprint</i> II.....	56
D. <i>Sprint</i> III.....	68
E. <i>Sprint</i> IV.....	82
F. Pengujian.....	93
BAB V PENUTUP.....	95
A. Simpulan.....	95
B. Saran.....	95
DAFTAR PUSTAKA.....	97
LAMPIRAN.....	99

DAFTAR TABEL

Tabel 2.1 Perbandingan <i>User Story</i> dan <i>Use Case</i>	17
Tabel 2.2 Aspek <i>Usability</i> Dalam Metode Heuristik	24
Tabel 2.3 Skala <i>Severity Ratings</i>	24
Tabel 2.4 Tinjauan Terdahulu	26
Tabel 3.1 Kebutuhan Desain	33
Tabel 3.2 Kebutuhan Informasi Tambahan	34
Tabel 3.3 Daftar <i>Endpoint API Server</i>	35
Tabel 3.4 <i>Scrum Team</i>	36
Tabel 3.5 Struktur <i>Product Backlog</i>	37
Tabel 3.6 Struktur <i>Sprint Backlog</i>	37
Tabel 3.7 Daftar Kuesioner Evaluasi Heuristik	39
Tabel 4.1 <i>User Story Login</i>	42
Tabel 4.2 <i>User Story</i> Pendaftaran	42
Tabel 4.3 <i>User Story</i> Informasi Rekening.....	43
Tabel 4.4 <i>User Story</i> Informasi Produk	43
Tabel 4.5 <i>User Story</i> Pembukaan Rekening.....	44
Tabel 4.6 <i>User Story</i> Mutasi Rekening	45
Tabel 4.7 <i>User Story</i> Transfer	46
Tabel 4.8 <i>User Story</i> Penarikan Dana	47
Tabel 4.9 <i>User Story</i> Pembayaran Angsuran Kredit.....	48
Tabel 4.10 <i>User Story</i> Informasi Lainnya	48
Tabel 4.11 <i>User Story</i> Pengaturan PIN Transaksi.....	49
Tabel 4.12 <i>Product Backlog</i>	50
Tabel 4.13 Jadwal <i>Sprint I</i>	51
Tabel 4.14 <i>Sprint Backlog Sprint I</i>	52
Tabel 4.15 Interaksi Layar <i>Onboard</i>	54
Tabel 4.16 Interaksi Layar <i>Login</i>	54
Tabel 4.17 Integrasi API Layar <i>Sprint I</i>	55
Tabel 4.18 Skema Validasi API Layar <i>Login</i>	55
Tabel 4.19 Jadwal <i>Sprint II</i>	56

Tabel 4.20 <i>Sprint Backlog Sprint II</i>	56
Tabel 4.21 Interaksi <i>Header</i> dan <i>Bottom Tab Menu</i>	60
Tabel 4.22 Interaksi Layar Beranda	60
Tabel 4.23 Interaksi Layar Mutasi	61
Tabel 4.24 Interaksi Layar Transaksi	61
Tabel 4.25 Interaksi Layar Informasi	61
Tabel 4.26 Integrasi API Layar-Layar Akar	62
Tabel 4.27 Interaksi Layar-Layar Informasi	63
Tabel 4.28 Integrasi API Layar-Layar Informasi Rekening	63
Tabel 4.29 Interaksi Layar Informasi Mutasi	65
Tabel 4.30 Integrasi API Layar Daftar Mutasi.....	66
Tabel 4.31 Integrasi API Layar Informasi Produk.....	67
Tabel 4.32 Jadwal <i>Sprint III</i>	68
Tabel 4.33 <i>Sprint Backlog Sprint III</i>	69
Tabel 4.34 Interaksi Layar Transfer Antar Rekening	72
Tabel 4.35 Interaksi Layar Transfer Antar Bank	72
Tabel 4.36 Interaksi Layar Tarik Tunai	72
Tabel 4.37 Interaksi Layar Bayar Kredit.....	72
Tabel 4.38 Interaksi Layar Riwayat Aktifitas.....	73
Tabel 4.39 Interaksi Layar Nominal	75
Tabel 4.40 Interaksi Layar Validasi Rekening	76
Tabel 4.41 Interaksi Layar Cek Tagihan	76
Tabel 4.42 Interaksi Layar Konfirmasi	76
Tabel 4.43 Integrasi API Layar-Layar Transaksi.....	76
Tabel 4.44 Skema Validasi Layar-Layar Transaksi.....	77
Tabel 4.45 Interaksi Layar Simulasi Kredit	79
Tabel 4.46 Interaksi Layar Buka Rekening.....	79
Tabel 4.47 Interaksi Layar Buka Rekening.....	80
Tabel 4.48 Integrasi API Sub-Layar Buka Rekening.....	80
Tabel 4.49 Skema Validasi Sub-Layar Buka Rekening	81
Tabel 4.50 Jadwal <i>Sprint IV</i>	82
Tabel 4.51 <i>Sprint Backlog Sprint IV</i>	82

Tabel 4.52 Interaksi Layar Bukti Transaksi	84
Tabel 4.53 Interaksi Layar Proses Tarik Tunai	84
Tabel 4.54 Interaksi Layar Pendaftaran	86
Tabel 4.55 Integrasi API Fitur Pendaftaran	86
Tabel 4.56 Skema Validasi Fitur Pendaftaran.....	87
Tabel 4.57 Interaksi Layar-Layar Informasi Umum	88
Tabel 4.58 Integrasi API Layar-Layar Informasi Umum.....	89
Tabel 4.59 Interaksi Layar Pengaturan	92
Tabel 4.60 Data Hasil Kuesioner	93
Tabel 4.61 Perhitungan Evaluasi Heuristik.....	94
Tabel 4.62 Nilai <i>Severity Ratings</i> Keseluruhan	94

**STMIK KOMPUTAMA
MAJENANG**

DAFTAR GAMBAR

Gambar 1.1 Grafik Transaksi Perbankan Berdasarkan Frekuensi	2
Gambar 2.1 Struktur Organisasi Bank ABC	9
Gambar 2.2 Metode <i>Scrum</i>	19
Gambar 3.1 Konsep Penelitian	31
Gambar 3.2 Antarmuka Aplikasi Paypal, OVO dan Dana.....	32
Gambar 3.3 Antarmuka Aplikasi BRImo dan BNI m-banking.....	32
Gambar 3.4 Rancangan Tata Letak Utama Aplikasi.....	33
Gambar 4.1 Peta Alur Navigasi Layar	53
Gambar 4.2 Peta Navigasi Layar <i>Login</i>	53
Gambar 4.3 Antarmuka Layar <i>Splash</i> , <i>Onboard</i> dan <i>Login</i>	54
Gambar 4.4 Antarmuka Layar <i>Loading</i>	58
Gambar 4.5 Peta Navigasi Layar-Layar Akar	58
Gambar 4.6 Peta Navigasi Layar Beranda	58
Gambar 4.7 Peta Navigasi Layar Mutasi	59
Gambar 4.8 Peta Navigasi Layar Transaksi	59
Gambar 4.9 Peta Navigasi Layar Informasi	59
Gambar 4.10 Antarmuka Layar-Layar Akar	60
Gambar 4.11 Peta Navigasi Layar Informasi Rekening	62
Gambar 4.12 Antarmuka Layar-Layar Informasi Rekening	63
Gambar 4.13 Peta Navigasi Layar Mutasi	64
Gambar 4.14 Antarmuka Layar Mutasi dan Daftar Mutasi Tabungan	65
Gambar 4.15 Peta Navigasi Layar Produk Perbankan	66
Gambar 4.16 Antarmuka Layar Produk Tabungan	67
Gambar 4.17 Perbaikan Antarmuka Layar Beranda dan Produk.....	70
Gambar 4.18 Peta Navigasi Jenis-Jenis Transaksi	71
Gambar 4.19 Antarmuka Layar-Layar Jenis Transaksi	71
Gambar 4.20 Antarmuka Layar Riwayat Aktifitas	73
Gambar 4.21 Peta Navigasi Sub-Layar Jenis Transaksi	74
Gambar 4.22 Antarmuka Layar Nominal dan Validasi Rekening	74
Gambar 4.23 Antarmuka Layar Cek Tagihan dan Proses Tarik Tunai	75

Gambar 4.24 Antarmuka Layar Konfirmasi dan Bukti Transaksi	75
Gambar 4.25 Peta Navigasi Layar Pembukaan Rekening.....	78
Gambar 4.26 Antarmuka Layar Buka Rekening dan Simulasi Kredit	78
Gambar 4.27 Antarmuka Sub-Layar Buka Rekening	79
Gambar 4.28 Perbaikan Layar Bukti Transaksi & Proses Tarik Tunai.....	83
Gambar 4.29 Layar Pendaftaran Untuk Nasabah.....	85
Gambar 4.30 Layar Pendaftaran Untuk Non-Nasabah.....	85
Gambar 4.31 Peta Navigasi Fitur Informasi Umum	88
Gambar 4.32 Antarmuka Layar-Layar Informasi Umum	88
Gambar 4.33 Validasi Kata Sandi Pada Layar <i>Login</i>	89
Gambar 4.34 <i>Signature</i> SHA256 Untuk Transaksi Antar Bank	90
Gambar 4.35 Pengaturan PIN Transaksi	90
Gambar 4.36 Sesi <i>Login</i> Berakhir	90
Gambar 4.37 Identifikasi Perangkat	91
Gambar 4.38 Antarmuka Layar Pengaturan	92

**STMIK KOMPUTAMA
MAJENANG**

DAFTAR LAMPIRAN

Lampiran 1 Foto Pelaksanaan	99
Lampiran 2 Implementasi Kode	101
Lampiran 3 Kuesioner Evaluasi Heuristik	105
Lampiran 4 Hasil Kuesioner Evaluasi Heuristik	106
Lampiran 5 Data Respon API	107
Lampiran 6 Kebutuhan Desain	110
Lampiran 7 Lembar Wawancara	111
Lampiran 8 Surat Tugas <i>Project Manager</i>	114
Lampiran 9 <i>Daily Scrum</i>	115
Lampiran 10 <i>Sprint Review</i>	118

**STMIK KOMPUTAMA
MAJENANG**

BAB I

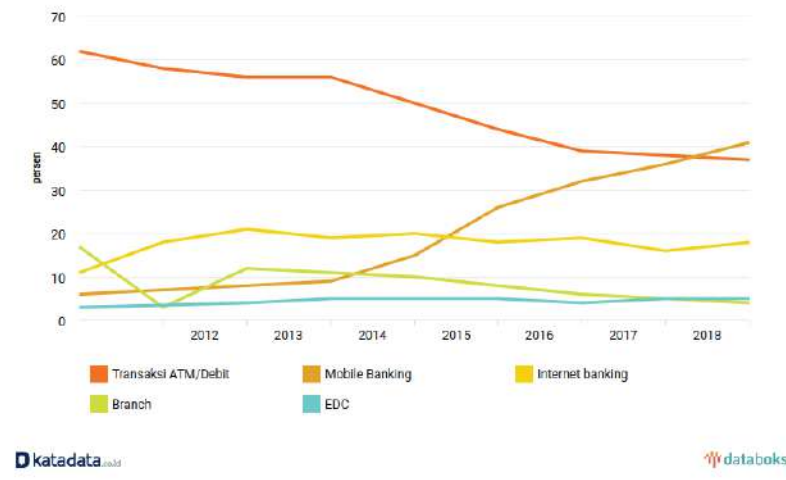
PENDAHULUAN

A. Latar Belakang Masalah

Perbankan adalah sektor keuangan yang menyediakan layanan keuangan, seperti pembiayaan, kredit, deposito, dan transfer uang kepada masyarakat. Perbankan telah menjadi layanan yang tidak bisa dipisahkan dari ekonomi masyarakat sehari-hari. Dalam era digital seperti sekarang ini, perbankan telah beradaptasi dengan menyediakan layanan keuangan secara *online* melalui layanan m-banking.

Mobile banking atau m-banking adalah layanan keuangan yang dapat diakses melalui perangkat *mobile* seperti *smartphone* atau tablet. Layanan ini memungkinkan pengguna untuk melakukan transaksi keuangan, seperti transfer uang, pembukaan rekening, pembayaran tagihan, dan pembelian produk secara *online*. M-banking menawarkan kemudahan bagi penggunanya karena dapat diakses kapan saja dan di mana saja, selama terdapat koneksi internet.

Berdasarkan laporan Katadata tahun 2019, penggunaan m-banking di Indonesia telah mengalami peningkatan sebagai transaksi perbankan dengan frekuensi terbanyak (2010-2018) menggeser peran kantor cabang dan ATM Bank. Hal ini menunjukkan bahwa layanan ini sangat dibutuhkan oleh masyarakat dan menjadi salah satu layanan yang penting bagi perbankan untuk menyediakannya.



Gambar 1.1 Grafik Transaksi Perbankan Berdasarkan Frekuensi

Sumber: (Widowati, 2019)

Selain dari peningkatan penggunaan, pemerintah telah mendorong percepatan digitalisasi di berbagai sektor termasuk perbankan. Bank ABC yang merupakan Badan Usaha Milik Daerah (BUMD) yang dimiliki oleh Pemda di salah satu Kabupaten di Pemprov Jawa Barat, saat ini belum mempunyai layanan *Mobile Banking* untuk melayani nasabahnya secara digital. Namun, Bank ABC sudah memiliki *Backend* sistem internal yang digunakan untuk proses bisnisnya. Pada tahun 2022 dengan alasan berkembangnya bisnis *micro-finance*, ketatnya persaingan, dan percepatan digitalisasi, tim internal divisi IT Bank ABC merasa diperlukannya perancangan dan pembuatan aplikasi m-banking untuk melayani nasabahnya yang kian bertambah.

Skripsi berjudul “Implementasi *scrum* dalam perancangan *frontend* aplikasi m-banking di Bank ABC Jawa Barat” ditujukan untuk memberikan solusi dari permasalahan yang ada pada Bank ABC Jawa Barat berupa tahap

awal *frontend* aplikasi m-banking. Penulis menggunakan *scrum* dalam perancangannya karena fleksibilitas kebutuhan dan diperlukannya interaksi intensif dengan tim pengembang internal atau pemilik produk terutama dalam integrasi dengan sistem (*backend*) yang sudah berjalan. Perancangan mengambil fokus pada permintaan fitur utama yang meliputi akses informasi rekening, akses transaksi rekening, melakukan transaksi rekening serta pembukaan rekening dengan mendukung platform Android dan iOS. Penulis menggunakan evaluasi heuristik untuk menguji kegunaan tampilan dikarenakan pengujian seperti keamanan merupakan hal sensitif yang dilakukan oleh pihak internal Bank ABC.

B. Rumusan Masalah

Rumusan permasalahan yang ingin diselesaikan oleh penulis berdasarkan latar belakang permasalahan yang ada dapat dirumuskan sebagai berikut:

1. Bagaimana implementasi *scrum* dalam perancangan aplikasi m-banking Bank ABC untuk dapat digitalisasi proses bisnis utama Bank ABC?
2. Apakah implementasi *scrum* dalam perancangan aplikasi m-banking Bank ABC dapat menghasilkan aplikasi m-banking yang terintegrasi dengan sistem (*backend*) yang ada dan berjalan di platform Android dan iOS?
3. Bagaimana hasil evaluasi kegunaan aplikasi m-banking Bank ABC melalui kuesioner evaluasi heuristik?

C. Batasan Masalah

1. Analisis kebutuhan sistem berupa kebutuhan desain, informasi tambahan, data API, dan *user story* disediakan oleh pihak internal Bank ABC.
2. Fitur-fitur aplikasi didasarkan pada pengumpulan data berupa wawancara dengan *Project Manager* dan ketersediaan data *backend* (API) yang sudah berjalan di Bank ABC.
3. Rancangan desain hanya menggunakan prinsip konsistensi UX melalui referensi beberapa aplikasi keuangan digital populer dan eksplorasi desain.
4. Penerapan proses perancangan menggunakan metode *scrum* dilakukan dengan membagi tahapan alur kerja ke dalam waktu satu bulan perancangan dan pengembangan.
5. Implementasi perancangan aplikasi m-banking Bank ABC menggunakan teknologi React Native dan bahasa pemrograman Javascript.
6. Uji coba pencarian *bug* aplikasi dan uji coba keamanan aplikasi dilakukan dan dipegang sepenuhnya oleh tim internal Bank ABC.
7. Evaluasi dan rekomendasi pengembangan dilakukan dengan metode evaluasi heuristik terhadap pihak internal Bank ABC Jawa Barat.
8. Perancangan hanya meliputi *frontend* aplikasi (tampilan), sementara struktur basis data dan relasi data dirahasiakan dan dikelola secara menyeluruh oleh tim internal Bank ABC.
9. Protokol fitur keamanan aplikasi ditetapkan oleh tim internal Bank ABC untuk menjaga *confidentiality*, *integrity* dan *availability* sistem informasi internal yang sudah berjalan.

D. Tujuan Penelitian

Berkaitan dengan masalah-masalah yang telah dirumuskan, perancangan *frontend* aplikasi m-banking Bank ABC memiliki tujuan:

1. Untuk menghasilkan tahap awal aplikasi m-banking yang mampu melakukan digitalisasi proses bisnis utama Bank ABC.
2. Untuk menghasilkan tahap awal aplikasi m-banking yang mampu terintegrasi dengan sistem (*backend*) yang sudah ada.
3. Untuk menghasilkan rekomendasi pengembangan dan perbaikan tahap awal aplikasi melalui evaluasi heuristik.

E. Manfaat Penelitian

Perancangan *frontend* aplikasi m-banking Bank ABC memiliki manfaat teoritis dan manfaat aplikatif. Manfaat teoritis dari implementasi *scrum* perancangan *frontend* aplikasi m-banking Bank ABC adalah sebagai berikut:

1. Memberikan kontribusi pemahaman penggunaan metode *scrum* dalam perancangan *frontend* aplikasi m-banking.
2. Menjadi sumber referensi penelitian sejenis di masa yang akan datang.
3. Menambah wawasan tentang implementasi *scrum* perancangan *frontend* aplikasi m-banking bagi para akademisi dan praktisi.

Sedangkan manfaat aplikatif implementasi *scrum* perancangan *frontend* aplikasi m-banking Bank ABC adalah sebagai berikut:

1. Memberikan solusi terkait layanan m-banking di Bank ABC Jawa Barat yang sesuai dengan kebutuhan nasabah.

2. Menjadi acuan, rekomendasi perbaikan dan pengembangan lebih lanjut dalam peningkatan kualitas aplikasi m-banking di Bank ABC.
3. Dapat diadopsi oleh bank-bank lain dengan proses bisnis serupa yang ingin melakukan implementasi *scrum* dalam perancangan *frontend* aplikasi m-banking.

F. Sistematika Penulisan

Garis besar keseluruhan penelitian dapat digambarkan melalui sistematika penulisan berikut:

1. Bab I Pendahuluan

Berisi tentang latar belakang permasalahan yang diambil penulis dalam judul “Implementasi *scrum* dalam perancangan *frontend* aplikasi m-banking Bank ABC Jawa Barat” beserta rumusan-rumusan masalah, batasan masalah, tujuan penelitian, manfaat dan sistematika penulisan.

2. Bab II Tinjauan Pustaka

Memuat pustaka berupa landasan teori dan penelitian terdahulu yang dijadikan sebagai referensi penelitian yang mendasari perancangan. Penulis membandingkan penelitian terdahulu dengan penelitian yang dilakukan untuk melihat perbedaan atau pengembangan dari penelitian sebelumnya.

3. Bab III Metodologi Penelitian

Pada bab ini menjelaskan secara spesifik terkait metodologi penelitian yang dilakukan, meliputi tempat dan waktu, bahan dan alat, pengumpulan data, dan konsep bagaimana penelitian dilakukan. Konsep

penelitian memuat hal teknis dan spesifik seperti metode pengembangan, analisis kebutuhan, dan pengujian yang dilakukan dalam penelitian.

4. Bab IV Hasil Penelitian

Hasil dari konsep penelitian pada Bab III dijabarkan kembali oleh penulis serta diimplementasikannya landasan teori pada Bab II untuk dapat mencapai tujuan dari rumusan-rumusan permasalahan pada Bab I.

5. Bab V Penutup

Dalam Bab V penulis mempresentasikan kesimpulan dari keseluruhan penelitian dengan mengaitkan teori yang diambil serta sesuai dengan tujuan penelitian. Penulis juga memberikan saran untuk penelitian selanjutnya yang berkaitan dengan topik penelitian yang dilakukan.

6. Daftar Pustaka

Memuat daftar sumber pustaka berupa artikel ilmiah pada jurnal, buku, artikel situs web maupun skripsi yang dikutip dan dipakai dalam penulisan penelitian di skripsi ini menggunakan gaya pengutipan APA.

**STMIK KOMPUTAMA
MAJENANG**

BAB II

TINJAUAN PUSTAKA

A. Landasan Teori

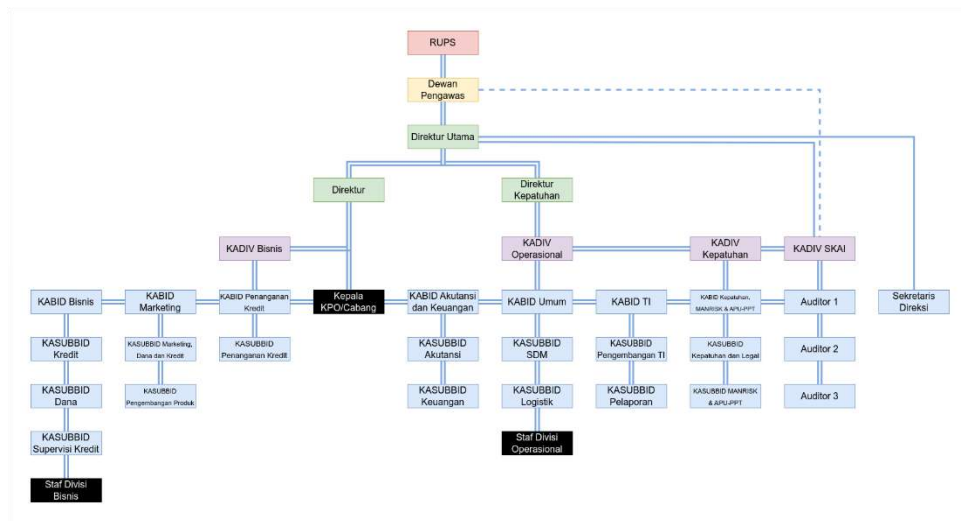
1. *Mobile Banking*

Mobile Banking atau m-banking adalah sebuah bentuk transaksi perbankan yang dilakukan melalui media *handphone/smartphone* atau tablet baik dalam bentuk aplikasi m-banking atau aplikasi bawaan operator seluler. Manfaat utama dari m-banking adalah praktis dan kemudahan dalam bertransaksi baik finansial maupun non finansial. Contoh transaksi finansial adalah transfer antar rekening/antar bank, pembayaran tagihan, pembelian pulsa dan lainnya. Sementara transaksi non-finansial adalah informasi saldo, informasi mutasi rekening dan lainnya (OJK, 2017).

2. Bank ABC Jawa Barat

Bank ABC merupakan Badan Usaha Milik Daerah (BUMD) yang dimiliki oleh salah satu Pemda Kabupaten dan Pemprov Jawa Barat dengan kegiatan usaha perbankan yang menjadi solusi keuangan masyarakat terutama bagi pelaku Usaha Kecil, Mikro, dan Menengah (UMKM). Bank ABC memiliki tiga produk utama yaitu kredit, tabungan dan deposito.

Bank ABC memiliki jumlah nasabah sekitar 160.000 dengan 7 kantor cabang dan 4 kantor kas dalam satu kabupaten. Struktur organisasi dari Bank ABC adalah sebagai berikut:



Gambar 2.1 Struktur Organisasi Bank ABC

3. Desain *Frontend*

a. Pengertian *Frontend*, UI/UX dan Prototipe

Frontend berkaitan dengan perancangan aplikasi adalah bagian dari sebuah aplikasi atau situs web yang dapat dilihat dan diakses oleh pengguna. Terdiri dari bagian yang membuat tampilan aplikasi atau situs web dan menangani interaksi pengguna dengan aplikasi tersebut. Sementara itu, *backend* adalah bagian dari aplikasi atau situs web yang tidak terlihat oleh pengguna dan berfungsi untuk menangani proses di balik layar yang memungkinkan aplikasi atau situs web bekerja dengan benar.

Contoh elemen *frontend* dalam sebuah aplikasi adalah tata letak, grafis, audio video, konten teks, alur penggunaan, preferensi pengguna dan elemen antarmuka pengguna seperti tombol, navigasi, menu dan sebagainya (Christensson, Frontend Definition, 2020).

Antarmuka pengguna atau *User Interface* (UI) adalah tampilan yang digunakan seseorang untuk mengontrol sebuah aplikasi perangkat lunak atau perangkat keras. Sedangkan pengalaman pengguna atau *User Experience* (UX) adalah pengalaman yang dialami seseorang saat menggunakan produk atau layanan. Dalam dunia teknologi, UX biasanya mengacu pada perangkat keras atau program perangkat lunak. UX dibagi menjadi dua kategori yaitu UX positif dan UX negatif. UX yang positif adalah sederhana, intuitif, dan menyenangkan. Sedangkan UX yang negatif adalah rumit, membingungkan, dan menjengkelkan (Christensson, User Experience Definition, 2019).

Prototype atau prototipe dalam kaitannya dengan perancangan aplikasi adalah suatu cara untuk mengembangkan sebuah aplikasi dengan membuat rancangan, sampel, atau model untuk menguji konsep atau proses kerja dari aplikasi tersebut. Prototipe bukan merupakan aplikasi akhir yang akan disebarluaskan. Prototipe dibuat untuk keperluan tahap awal pengembangan aplikasi dan untuk mengetahui apakah fitur dan fungsi dalam program bekerja sesuai dengan yang direncanakan. Dengan demikian, pengembang aplikasi dapat mengetahui kekurangan dan kesalahan lebih awal sebelum menambahkan fitur lain ke dalam aplikasi dan merilis aplikasi tersebut (Setiawan, 2021).

b. Metode Desain

Referensi desain dipakai sebagai panduan dalam membuat tampilan antarmuka sesuai salah satu prinsip UX yaitu prinsip konsistensi. Prinsip konsistensi menyatakan bahwa ketika pengguna mengunjungi situs tertentu, mereka berharap antarmuka pengguna yang sama seperti saat mengunjungi situs lain. Prinsip ini membantu pengguna untuk mengerti bagaimana menggunakan situs secara efisien, karena mereka tidak perlu meluangkan waktu untuk mempelajari cara kerja situs yang berbeda. Prinsip konsistensi juga memiliki tujuan lain, seperti memastikan bahwa pengguna dapat memprediksi bagaimana antarmuka pengguna akan berfungsi (Nikolov, 2017).

c. Figma

Figma merupakan perangkat lunak yang memungkinkan pengguna untuk membuat dan membagikan desain grafis, seperti *wireframe*, *mockup*, dan *prototype* untuk aplikasi web dan *mobile*.

Figma menawarkan fitur kolaborasi yang memungkinkan tim desain untuk bekerja bersama secara *real-time*, baik secara *online* maupun *offline*. Pengguna juga dapat membagikan desain mereka dengan orang lain melalui tautan yang dapat diakses melalui peramban web, tanpa harus mengunduh perangkat lunak apa pun sehingga sangat cocok untuk digunakan untuk pekerjaan tim (Putra, Ajie, & Safitri, 2021).

4. Implementasi Frontend

a. Javascript

Javascript adalah bahasa pemrograman yang bisa digunakan untuk menambahkan fitur yang lebih canggih pada halaman web. Misalnya, ketika sebuah halaman web tidak hanya menampilkan informasi statis yang bisa kita lihat tetapi juga menampilkan pembaruan konten yang tepat waktu, peta interaktif, grafik 2D/3D. Javascript memungkinkan untuk membuat konten yang terus diperbarui secara dinamis, mengontrol multimedia, menganimasikan gambar, dan hal dinamis lainnya (MDN, 2022).

b. Node.js

Node.js merupakan lingkungan pengembangan yang memungkinkan menjalankan kode Javascript di luar peramban web menggunakan *engine* Javascript V8 dari Google. Node.js dapat mengeksekusi kode dengan cepat dan tanpa memerlukan peramban di mana hal ini memungkinkan pengembang untuk membuat aplikasi sisi server menggunakan bahasa yang sama dengan yang digunakan untuk membuat aplikasi sisi klien. Selain itu, Node.js juga memiliki lingkungan *runtime* yang membantu pengembangan aplikasi yang mudah dikelola dan dapat menangani banyak permintaan secara bersamaan dengan cara yang efisien (Orsini, 2013).

c. React Native

React Native adalah sebuah *framework* yang memungkinkan pengembangan aplikasi *mobile* menggunakan Javascript yang dapat berjalan dengan performa mendekati aplikasi *native* untuk iOS dan Android. React Native merupakan pustaka turunan dari React, sebuah pustaka Javascript dari Facebook untuk membuat antarmuka pengguna, tetapi ditujukan untuk platform *mobile* bukan untuk peramban web. Sehingga memungkinkan pengembang web untuk dapat membuat aplikasi *mobile* yang terlihat dan terasa seperti aplikasi *native* menggunakan Javascript tanpa perlu belajar bahasa baru. Selain itu, React Native memudahkan untuk mengembangkan aplikasi untuk kedua platform Android dan iOS secara bersamaan karena sebagian besar kode yang ditulis dapat dibagi antar kedua platform tersebut (Eisenman, 2015).

d. Android

Android adalah sebuah sistem operasi perangkat *mobile* yang dimiliki Google yang dibuat berdasarkan kernel Linux. Android saat ini merupakan sistem operasi *smartphone* yang paling banyak digunakan di mana Android menyediakan lingkungan pengembangan platform yang gratis atau terbuka bagi para pengembang untuk membuat aplikasi mereka sendiri (Android, n.d.).

e. IOS

iOS (dulunya dikenal sebagai iPhone OS) adalah sistem operasi *mobile* yang dikembangkan dan dipasarkan oleh Apple untuk perangkat iPhone, iPod Touch, dan iPad. Sistem operasi iOS merupakan salah satu sistem operasi *mobile* terpopuler di dunia selain Android, terutama di pasar *smartphone*. iOS menawarkan antarmuka pengguna yang intuitif dan mudah digunakan serta banyak fitur yang memungkinkan pengguna untuk mengelola perangkat mereka dengan mudah (Apple, iOS 16 - Apple, n.d.).

5. IDE

Integrated Development Environment (IDE) dalam ruang lingkup perangkat lunak merupakan sebuah aplikasi yang digunakan pengembang yang terintegrasi dengan alat bantu lainnya dalam satu program meliputi teks editor, *debugger*, dan *compiler* serta manajemen versi untuk mempermudah pengembangan perangkat lunak (Christensson, IDE Definition, 2015).

a. Visual Studio Code

Visual Studio Code (VSCode) adalah sebuah teks editor sekaligus IDE yang ringan dan kuat dalam mendukung pengembangan perangkat lunak dengan berbagai dukungan bahasa seperti Javascript, Typescript, dan Node.js, dan memiliki ekosistem yang kaya akan ekstensi untuk bahasa dan *runtime*

lainnya seperti C++, Python, PHP, Go, dan lainnya. VSCode berjalan di desktop dan tersedia untuk Windows, macOS, dan Linux (Microsoft, n.d.).

b. Android Studio

Android Studio adalah perangkat lunak atau IDE yang digunakan untuk membuat aplikasi Android, yang dapat diunduh secara gratis. Diumumkan oleh Google pada tahun 2013, Android Studio menggantikan Eclipse sebagai alat bantu resmi untuk pengembangan aplikasi Android (Continue, 2020).

c. XCode

XCode adalah IDE keluaran Apple untuk macOS, yang digunakan untuk mengembangkan perangkat lunak untuk macOS, iOS, iPadOS, watchOS, dan tvOS. Pertama kali dirilis pada akhir tahun 2003 dan dapat diunduh melalui Mac App Store secara gratis (Apple, Xcode 14 Overview, n.d.).

6. Integrasi

a. Git

Git adalah sebuah sistem yang dirancang untuk mengelola versi suatu proyek (*Version Management System*), baik proyek kecil maupun proyek besar dengan cepat dan efisien. Git memiliki ukuran yang kecil serta performa yang cepat yang lebih baik daripada alat SCM lain seperti Subversion, CVS, Perforce, dan ClearCase. Kemampuan Git lainnya adalah untuk membuat

local branch dengan mudah, *staging* area yang nyaman, dan beberapa pilihan *workflow*. Git tersedia secara gratis dan terbuka untuk semua orang (Git, n.d.).

b. GitHub

GitHub adalah sebuah aplikasi berbasis web untuk membantu mengelola dan menyimpan proyek-proyek yang menggunakan Git. Selain itu, GitHub memungkinkan untuk dilakukannya kerja sama dengan orang lain dalam mengerjakan suatu proyek dan terus melakukan *update* secara berkala, termasuk melacak dan menyimpan perubahan yang terjadi dalam proyek tersebut (Prasetiawan, 2021).

c. API

API singkatan dari *Application Programming Interface* atau antarmuka pemrograman aplikasi. API adalah sekumpulan perintah, fungsi, protokol, dan objek yang dapat digunakan oleh *programmer* untuk membuat perangkat lunak atau berinteraksi dengan sistem eksternal. API memberikan standar untuk melakukan interaksi sehingga tidak diperlukannya penulisan ulang implementasi kode (Christensson, API Definition, 2016).

d. JSON

JSON (*Javascript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diproses oleh komputer. Format ini

didasarkan pada bagian dari bahasa pemrograman Javascript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON adalah format teks yang tidak tergantung pada bahasa pemrograman apa pun karena menggunakan gaya bahasa yang umum digunakan oleh *programmer* keluarga C, termasuk C, C++, Javascript, Perl, Python, dan lainnya. Karena sifat-sifat tersebut, JSON menjadi format yang ideal untuk pertukaran data (Continue, 2020).

7. *User Story*

Dalam artikel riset Daniella (2020), *user story* didefinisikan sebagai deskripsi sederhana dari keinginan atau kebutuhan pengguna. *User story* adalah metode mudah dalam merekam kebutuhan pengguna, mengidentifikasi kompleksitas sistem dan memecahnya ke dalam kata-kata yang mudah dipahami. *User story* berbeda dengan *use case*, dengan perbandingan sebagai berikut (Varga, 2020):

Tabel 2.1 Perbandingan *User Story* dan *Use Case*

No	<i>User Story</i>	<i>Use Case</i>
1	Terfokus pada <i>user</i> , menyorot permasalahan <i>user</i> setiap harinya	Terfokus pada hal teknis, untuk tim pengembang saja
2	Interaksi digambarkan secara umum	Secara spesifik menggambar <i>user</i> dengan sistem apa saja
3	Pendek dan menyisakan detail untuk diskusi dan pengembangan	Mendetail, spesifik semua langkah yang harus diikuti pengembang

User story memiliki kelebihan seperti pemecahan masalah atau tujuan besar menjadi tujuan-tujuan kecil yang dapat diraih, dapat

dimengerti semua kalangan, secara jelas mendefinisikan tujuan dan mengedepankan pengguna. *User story* dapat dituliskan dengan format sebagai berikut:

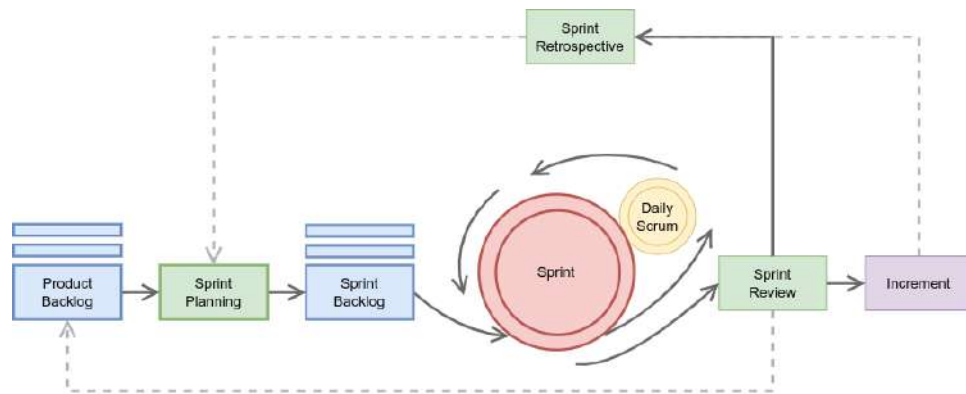
“Sebagai seorang [pengguna], saya ingin [tujuan] sehingga saya dapat [manfaat]”.

Naufaldi Satriya (2021) dalam penelitiannya melakukan implementasi *user story* dengan tambahan representasi alur dengan bagian-bagian sebagai berikut:

- a. *Actor* atau pelaku yang melakukan kegiatan dalam *Story*.
- b. *Objective* adalah tujuan dari *Story* yang dibuat untuk *Actor*.
- c. *Pre-Condition* adalah kondisi awal sebelum *Story* dilakukan.
- d. *Main Flow* adalah tahapan utama *Actor* untuk mencapai *Objective*.
- e. *Alternative* adalah tahapan alternatif jika *Pre-Condition* tidak terpenuhi.
- f. *Post Condition* adalah kondisi akhir setelah *Objective* tercapai.

8. Metode Scrum

Scrum adalah kerangka kerja ringan yang membantu individu, tim dan organisasi membuat nilai melalui solusi adaptif untuk masalah kompleks yang dikembangkan pada tahun 1993 oleh Ken Schwaber dan Jeff Sutherland. *Scrum* berfokus pada pengembangan produk yang cepat dan terus-menerus. Schwaber dan Sutherland mengembangkan *scrum* dengan mengacu pada praktik terbaik dari proyek-proyek *software* yang mereka ikuti serta literatur yang terdapat di bidang manajemen proyek.



Gambar 2.2 Metode *Scrum*

Pada tahun 2010, Ken Schwaber dan Jeff Sutherland membuat *Scrum Guide* sebagai panduan dalam penggunaan *scrum* untuk seluruh orang di dunia. Berikut adalah beberapa terminologi dalam metodologi *scrum* berdasarkan pembaruan tahun 2020 (Schwaber & Sutherland, 2020):

a. *Scrum Theory*

Scrum merupakan metodologi pengembangan produk yang didasarkan pada pengalaman dan pemikiran yang efisien dengan tujuan untuk mengoptimalkan kemampuan prediksi dan mengontrol risiko. *Scrum* menggunakan pendekatan iterasi dan inkremental dengan melakukan inspeksi dan adaptasi secara teratur dalam suatu periode yang disebut *sprint*. *Scrum* memiliki keunggulan yaitu transparansi, inspeksi dan adaptasi.

b. *Scrum Team*

Scrum team atau tim *scrum* terdiri dari orang-orang yang bertanggung jawab untuk membuat produk yang bernilai. Tim

scrum bertanggung jawab atas semua kegiatan yang terkait dengan produk, mulai dari kerja sama dengan *stakeholders*, verifikasi, pemeliharaan, operasi, eksperimen, penelitian dan pengembangan, serta segala sesuatu yang mungkin dibutuhkan. Scrum menentukan tiga tanggung jawab spesifik di dalam tim *scrum*: *Developer*, *Product Owner*, dan *Scrum Master*. Tidak ada kriteria khusus untuk anggota *scrum team* dalam pembentukannya. Berikut tanggung jawab masing-masing anggota *scrum team*:

1) *Developer*

Orang-orang yang berada dalam tim *scrum* yang berkomitmen untuk menciptakan segala aspek *increment* yang dapat digunakan setiap *sprint* dengan mengeksekusi daftar item yang ada.

2) *Product Owner*

Product owner bertanggung jawab untuk memaksimalkan nilai produk yang dihasilkan dari pekerjaan tim *scrum*.

3) *Scrum Master*

Scrum Master bertanggung jawab atas efektivitas tim *scrum*. *Scrum master* memastikan dan meninjau tim *scrum* untuk meningkatkan praktik-praktik mereka, di dalam kerangka *scrum*.

c. *Scrum Events*

Scrum events merupakan bagian-bagian dari *scrum* yang digunakan untuk menciptakan keteraturan dan meminimalkan kebutuhan pertemuan yang tidak perlu. Terdapat beberapa *event* di dalam *scrum* di antaranya adalah:

1) *The Sprint*

Sprint merupakan komponen utama dalam *scrum* di mana masing-masing *sprint* ditetapkan dengan jangka waktu tetap untuk menjaga konsistensi dan berjalan inkremental berdasarkan *sprint* sebelumnya. Aturan ketika *sprint* berjalan adalah:

- a) Perubahan tidak boleh mengubah tujuan *sprint*.
- b) Kualitas tidak menurun.
- c) *Product backlog* diperbaiki sesuai kebutuhan.
- d) Jangkauan dapat dijelaskan dan dinegosiasikan kepada *product owner* sejauh apa yang telah dipelajari dalam *sprint*.

2) *Sprint Planning*

Sprint planning adalah *event* untuk mempersiapkan *sprint* apa saja yang diperlukan dan *product backlog* apa yang akan dimasukkan ke dalam *sprint* tersebut. *Sprint planning* dibatasi maksimal 8 jam untuk 1 bulan *sprint*, untuk *sprint* yang singkat, waktu dapat lebih pendek.

3) *Daily Scrum*

Daily scrum adalah *event* 15 menit untuk *developer* setiap harinya untuk menginspeksi kemajuan terhadap *sprint goal* dan adaptasi *sprint backlog* sesuai kemajuan.

4) *Sprint Review*

Sprint review meninjau hasil dari *sprint* dan menentukan adaptasi lanjutan yang harus dilakukan untuk *sprint* selanjutnya. *Sprint review* dilakukan di akhir *sprint* sebelum *sprint retrospective* dan waktu maksimalnya adalah 4 jam untuk 1 bulan *sprint*, dapat lebih singkat untuk *sprint* pendek.

5) *Sprint Retrospective*

Tujuan *sprint retrospective* adalah merencanakan cara untuk menambah kualitas dan efektivitas dari *sprint*. *Event* ini menyimpulkan hasil dari *sprint* dengan maksimal 3 jam untuk 1 bulan *Sprint*, dapat lebih singkat untuk *sprint* pendek.

d. *Scrum Artifacts*

Scrum artifacts merupakan struktur-struktur yang digunakan untuk mengelola dan mengikuti progres kerja dalam proyek. *Scrum artifacts* terdiri dari tiga struktur utama yaitu *Product Backlog*, *Sprint Backlog*, dan *Increment*.

1) *Product Backlog*

Product backlog adalah daftar semua fitur, perbaikan, dan kebutuhan yang diinginkan dalam produk yang sedang dikembangkan. *Product backlog* dikelola oleh *product owner*, yang bertanggung jawab untuk menentukan prioritas dan nilai dari setiap item di dalamnya.

2) *Sprint Backlog*

Sprint backlog adalah daftar pekerjaan yang akan dilakukan dalam *sprint* saat ini. *Sprint backlog* merupakan sub dari *product backlog* yang dipilih oleh tim *scrum* untuk dikerjakan dalam *sprint* tersebut.

3) *Increment*

Increment adalah hasil dari setiap *sprint*, yaitu produk yang selesai dan dapat digunakan. *Increment* harus merupakan tambahan yang bernilai bagi produk yang telah ada sebelumnya, dan harus memenuhi kriteria yang ditetapkan oleh tim *scrum*.

9. Evaluasi Heuristik

Evaluasi heuristik adalah sebuah metode yang dibuat oleh Nielsen dan Molich untuk menilai kualitas tampilan *User Interface* (UI) dengan berfokus pada aspek kegunaan (*usability*). Keunggulan metode heuristik adalah murah, intuitif, tanpa perencanaan khusus serta dapat dilakukan di tahap awal pengembangan (Ariansyah & Bambang Sumantri, 2020).

Pengukuran *usability* dengan evaluasi heuristik dapat dilakukan dengan minimal 3 responden, serta terdapat 5 aspek yang digunakan dalam pengukuran *usability* di antaranya yaitu:

Tabel 2.2 Aspek *Usability* Dalam Metode Heuristik

No	Aspek	Keterangan
1	<i>Learnability</i>	Kemudahan pengguna dalam menjalankan aksi saat pertama kali melihat desain tersebut.
2	<i>Efficiency</i>	Kecepatan pengguna dalam menjalankan aksi setelah mempelajarinya.
3	<i>Memorability</i>	Kemudahan pengguna dalam mengingat aksi setelah sekian lama tidak menggunakan desain tersebut.
4	<i>Errors</i>	Seberapa mudah pengguna menyelesaikan kesalahan atau seberapa banyak pengguna membuat kesalahan.
5	<i>Satisfaction</i>	Seberapa menyenangkan pengguna dalam memakai desain tersebut.

Masalah-masalah yang ditemukan dari setiap aspeknya dinilai sesuai dengan tingkat permasalahannya atau dalam tingkatan *severity ratings*. *Severity ratings* didefinisikan oleh Nielsen sebagai serangkaian kombinasi dengan 5 tingkatan yang terdiri dari frekuensi masalah yang terjadi, dampak dari masalah dan kegigihan dari masalah.

Tabel 2.3 Skala *Severity Ratings*

Skala	Keterangan
0	Tidak ada masalah dan tidak perlu perbaikan.
1	Masalah yang tidak mengganggu dan perbaikan jika ada waktu.
2	Masalah tingkat kecil, perbaikan dengan prioritas rendah.
3	Masalah tingkat menengah, perbaikan dengan prioritas tinggi.
4	Masalah besar, harus diperbaiki secepatnya sebelum rilis ke publik.

B. Tinjauan Terdahulu

Penelitian berjudul “Implementasi *Scrum* Dalam Perancangan *Frontend* Aplikasi M-Banking Di Bank ABC Jawa Barat” mengambil beberapa referensi terdahulu sebagai bahan acuan dalam meningkatkan penelitian-penelitian yang sudah dilakukan sebelumnya. Penelitian-penelitian terdahulu yang dirujuk merupakan penelitian yang memiliki kesamaan terhadap penelitian yang dilakukan. Persamaan penelitian dapat berupa penggunaan metodologi *scrum*, perancangan aplikasi *mobile*, aplikasi m-banking, atau perancangan *frontend*.



STMIK KOMPUTAMA
MAJENANG

Tabel 2.4 Tinjauan Terdahulu

No	Judul, Nama Penulis, dan Tahun Publikasi	Kesimpulan	Saran atau Kelemahan	Perbandingan
1	<i>The Implementation of Scrum Framework for Developing Fundraising Mobile Applications</i> (Sasmito & Fauzan, 2020).	Demonstrasi implementasi <i>scrum</i> pada perancangan aplikasi penggalangan dana bahwa menggunakan <i>scrum</i> lebih cepat dilakukan hanya selama 35 hari, data terintegrasi dengan baik, dan permasalahan diselesaikan dengan cepat.	Pengujian yang dipakai berupa pengujian <i>usability</i> atau daya guna menggunakan metode SUS (<i>System Usability Scale</i>) dengan kelemahannya adalah tidak terdapat informasi akurat mengenai apa kelemahan yang dimiliki produk.	Implementasi <i>scrum</i> aplikasi <i>frontend</i> m-banking Bank ABC dieksekusi lebih cepat lagi selama 28 hari dan pengujian <i>usability</i> menggunakan metode evaluasi heuristik yang lebih memberi tahu detail aspek kelemahan apa yang ada.
2	Implementasi Metode Scrum Dalam Pengembangan Perangkat Lunak Deliv (Rafif, 2021).	Pembuatan aplikasi Deliv dalam waktu 12 minggu dan 6 kali <i>sprint</i> menunjukkan <i>scrum</i> dapat secara efektif menghasilkan tahap awal atau MVP (<i>Minimum Viable Product</i>).	Pengujian menggunakan metode UEQ (<i>User Experience Qestionnaire</i>) kepada 10 responden yang lebih memakan waktu dengan 6 aspek yaitu daya tarik, kejelasan, efisiensi, ketepatan, stimulasi, dan kebaruan.	Penelitian ini dilaksanakan selama 4 minggu dengan 4 kali <i>sprint</i> . Pengujian evaluasi heuristik lebih cepat dilakukan dengan 3 responden terhadap 5 aspek yaitu <i>learnability</i> , <i>efficiency</i> , <i>memorability</i> , <i>errors</i> serta <i>satisfaction</i> .

No	Judul, Nama Penulis, dan Tahun Publikasi	Kesimpulan	Saran atau Kelemahan	Perbandingan
3	Pengembangan <i>Front-end</i> Pada Aplikasi M-banking Agen46 Dengan Teknologi Flutter (Anastasia, 2022).	Dihasilkan <i>frontend</i> aplikasi m-banking Agen46 dengan teknologi Flutter dengan metode agile <i>scrum</i> yang fokus pada fitur <i>top up e-money</i> dan tiket KAI.	Pengujian hanya fungsionalitas serta disarankan mempelajari teknologi lain seperti Reactjs, Golang, Docker serta penentuan <i>stack</i> teknologi saat pembentukan <i>backlog</i> .	<i>Frontend</i> aplikasi m-banking Bank ABC menggunakan <i>stack</i> teknologi React Native dengan ekosistem lebih luas, evaluasi heuristik, serta fitur m-banking yang lebih banyak.
4	Model RAD Dalam Perancangan <i>Front End</i> Aplikasi <i>Payment Point Online Bank</i> (PPOB) Berbasis Android (Wasesha & Frieyadi, 2020).	Dihasilkan aplikasi PPOB berbasis Android <i>frontend</i> dan <i>backend</i> yang terintegrasi lewat JSON dengan fitur pembayaran berbagai jenis tagihan.	Pengujian hanya sebatas fungsional dan model RAD dalam perancangannya tidak cocok untuk skala besar yang membutuhkan kolaborasi dan transparansi.	Integrasi antara <i>frontend</i> dan sistem <i>backend</i> Bank ABC juga menggunakan JSON, namun menggunakan <i>scrum</i> yang unggul dalam transparansi, inspeksi dan adaptasi.
5	Perancangan <i>Desain User Interface</i> (UI) Aplikasi Pencari Kost (Reynaldi, 2019).	Desain UI aplikasi pencari kost dalam bentuk prototipe yang dirancang dengan prinsip perancangan UI <i>user familiarity, consistency, minimal surprise, recoverability, user guidance, dan user diversity</i> .	Penelitian memberikan saran perancangan UI sesuai dengan pengalaman pengguna, berdasarkan prinsip UI yang ada, serta berpikiran maju untuk membuat desain yang siap beradaptasi dengan perubahan masa depan.	Aplikasi <i>frontend</i> m-banking Bank ABC hanya dirancang dengan prinsip UI yaitu <i>consistency</i> dengan referensi aplikasi m-banking lainnya serta desain kompatibel untuk penambahan fitur di masa depan seperti fitur PPOB.

Setelah melakukan peninjauan terhadap penelitian-penelitian terdahulu, didapatkan beberapa perbedaan dan kelebihan dengan penelitian yang akan dilakukan peneliti dengan memfokuskan pada pengembangan *frontend* tanpa melakukan desain data atau arsitektur sistem. Di samping itu, pada penelitian terdahulu tidak menggunakan metode *user story* dalam pembangunannya dengan keunggulan yang telah peneliti sebutkan pada landasan teori. Pada penelitian yang akan dilakukan peneliti, penelitian akan memadukan metode *user story*, kerangka kerja *scrum* serta evaluasi heuristik di mana belum dilakukan sebelumnya untuk menghasilkan sebuah *frontend* aplikasi yang difokuskan pada kegunaan penggunaanya.

Penggunaan teknologi *React Native* juga dipilih dengan keunggulan ekosistem yang luas di dukung oleh Facebook, *multi-platform* Android dan iOS sehingga menghemat biaya pengembangan dengan satu basis kode, serta kemudahan pengembangan dan dokumentasi yang lengkap di mana pada penelitian terdahulu belum ada yang menggunakannya.

STMIK KOMPUTAMA
MAJENANG

BAB III

METODOLOGI PENELITIAN

A. Tempat dan Waktu Penelitian

Penelitian dilaksanakan di Bank ABC Jawa Barat dengan dilakukan secara *remote* di mana setiap pertemuan dan komunikasi dilaksanakan melalui jarak jauh. Waktu penelitian dilakukan dalam jangka waktu 37 hari dimulai tanggal 23 Februari 2023 hingga 31 Maret 2023 dengan tahapan perancangan selama 28 hari.

B. Bahan dan Alat Penelitian

Penelitian ditunjang oleh beberapa bahan alat sebagai berikut:

1. Kebutuhan perangkat lunak

- a. Visual Studio Code
- b. Android Studio
- c. XCode
- d. Nodejs
- e. Brave Web Browser

f. Figma

g. Github

2. Kebutuhan perangkat keras

- a. Laptop Ideapad L340 Ryzen 3 RAM 8GB
- b. *Smartphone* Android Samsung Galaxy A10s
- c. Konektor USB

C. Metode Pengumpulan Data

Data yang digunakan dalam penelitian ini melibatkan variabel dependen dan variabel independen. Variabel dependen adalah implementasi *scrum* dalam perancangan *frontend* aplikasi m-banking, sedangkan variabel independen terdiri dari daftar fitur-fitur aplikasi secara spesifik, kebutuhan-kebutuhan desain, sistem keamanan aplikasi, dan daftar *scrum team*.

1. Data Primer

Data primer diambil berdasarkan wawancara yang dilakukan peneliti terhadap perwakilan Bank ABC yang menangani proyek perancangan m-banking. Data primer terdiri dari daftar fitur-fitur aplikasi secara spesifik, kebutuhan-kebutuhan desain, sistem keamanan aplikasi dan daftar *scrum team*.

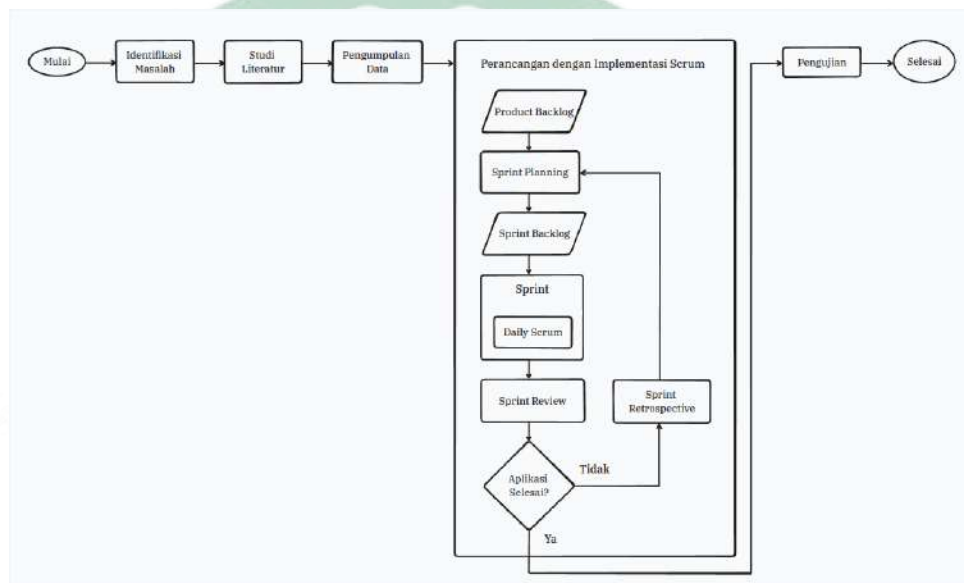
2. Data Sekunder

Data sekunder diambil melalui observasi terhadap situs web Bank ABC dengan melakukan pencatatan informasi seperti slogan, struktur organisasi, profil, serta informasi alamat dan kontak dari Bank ABC. Studi literatur juga dilakukan dengan membaca jurnal, skripsi, artikel yang berkaitan dengan implementasi *scrum* dalam perancangan *frontend* aplikasi m-banking.

D. Konsep Penelitian

Konsep penelitian “Implementasi *Scrum* Dalam Perancangan *Frontend* Aplikasi M-Banking Di Bank ABC Jawa Barat” dimulai dengan identifikasi masalah melalui wawancara peneliti dengan narasumber dan

studi literatur mengenai permasalahan yang dirumuskan untuk memperoleh metodologi yang digunakan yaitu *scrum*. Selanjutnya dilakukan analisis kebutuhan dan desain sistem, mengimplementasikan tahapan-tahapan *scrum* dan melakukan pengujian dengan evaluasi heuristik.



Gambar 3.1 Konsep Penelitian

1. Identifikasi Masalah

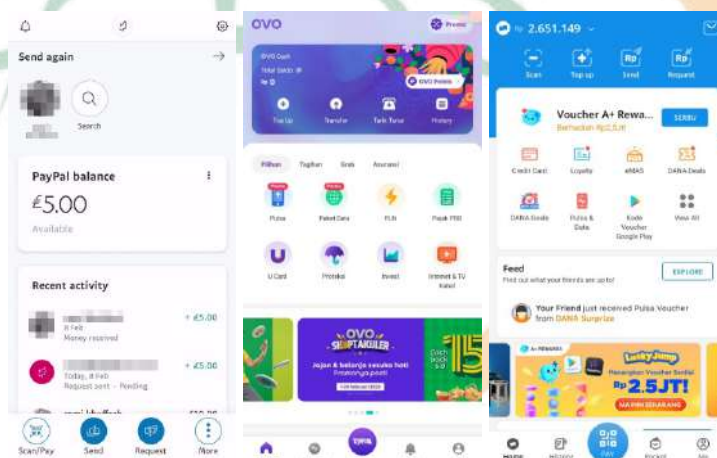
Tahapan identifikasi masalah, peneliti mengidentifikasi masalah yang ada pada Bank ABC di Jawa Barat melalui wawancara dan observasi. Pada tahapan ini, penelitian merumuskan latar belakang diperlukannya perancangan *frontend* aplikasi m-banking Bank ABC yang telah disebutkan pada bagian latar belakang dan rumusan permasalahan.

2. Studi Literatur

Studi literatur terkait metodologi *scrum* dilakukan dengan mempelajari buku petunjuk *scrum* berjudul *Scrum Guide* yang ditulis

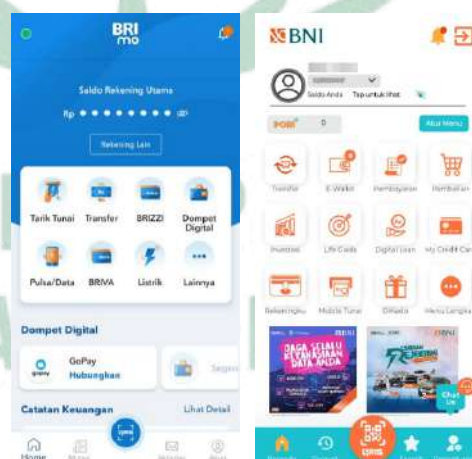
langsung oleh Ken Schwaber dan Jeff Sutherland. Sementara untuk menghasilkan konsep tata letak desain antarmuka adalah dengan membandingkan referensi desain pada aplikasi-aplikasi yang serupa. Beberapa desain aplikasi *fintech*/m-banking dijadikan referensi desain aplikasi m-banking Bank ABC antara lain:

a. Aplikasi dompet digital



Gambar 3.2 Antarmuka Aplikasi Paypal, OVO dan Dana

b. Aplikasi m-banking



Gambar 3.3 Antarmuka Aplikasi BRI Mo dan BNI m-banking

Terdapat tiga rancangan tata letak utama aplikasi m-banking Bank ABC berdasarkan referensi desain dan tata letak lainnya akan mengembangkan tata letak utama yang ada.



Gambar 3.4 Rancangan Tata Letak Utama Aplikasi

3. Pengumpulan Data

Data kebutuhan desain untuk warna primer dan logo diambil dari situs web Bank ABC sementara jenis huruf, warna selain primer, ikon dan selainnya didasarkan pada asumsi peneliti. Kebutuhan informasi tambahan, serta data API diberikan oleh pihak Bank ABC baik melalui undangan email ke dalam situs Postman dan Google Spreadsheet maupun secara langsung lewat media komunikasi.

a. Kebutuhan desain dan informasi tambahan

Tabel 3.1 Kebutuhan Desain

No	Kebutuhan	Keterangan
1	Jenis huruf	Oxygen
2	Palet warna	1. Primer: #2181C7 2. Sukses: #28A328 3. Informasi: #0075E2 4. Peringatan: #FF970F

No	Kebutuhan	Keterangan
		5. Bahaya: #FF3838 6. Latar belakang: #FFFFFF 7. Aksent: #F6F7F6 8. Teks: #656565
3	Logo	Disediakan
4	Ilustrasi	Undraw
5	Ikon	Material Icons
6	Tema	Terang

Tabel 3.2 Kebutuhan Informasi Tambahan

No	Kebutuhan	Keterangan
1	Kode layanan	Tabungan (1), Deposito (2), Kredit (3)
2	Pendaftaran (mempunyai rekening)	1. NIK KTP 2. Email 3. Kata sandi
3	Pendaftaran (tidak mempunyai rekening)	1. NIK KTP 2. Nama lengkap 3. Tempat, tanggal lahir 4. Nomor telepon 5. Alamat 6. Produk tabungan yang dipilih 7. Tujuan pembukaan tabungan 8. Sumber dana tabungan 9. Kantor cabang pilihan 10. Email 11. Kata sandi
4	Buka rekening tabungan	1. Jenis produk tabungan yang dipilih 2. Rekening sumber 3. Kantor cabang yang dipilih 4. Nama yang didaftarkan 5. Tujuan Pembukaan 6. Sumber dana
5	Buka rekening deposito	1. Jenis produk tabungan yang dipilih 2. Rekening sumber 3. Kantor cabang yang dipilih 4. Nama yang didaftarkan 5. Tujuan Pembukaan 6. Sumber dana

No	Kebutuhan	Keterangan
		7. Jangka waktu (bulan) 8. Jumlah penempatan (rupiah)
6	Buka rekening kredit	Dialihkan ke situs web Bank ABC
7	Rumus simulasi produk kredit	$\frac{pokok}{jangka\ waktu} + \frac{pokok * \left(\frac{bunga}{100}\right)}{jangka\ waktu}$
8	Fitur keamanan	1. Kata sandi yang kuat 2. Validasi masukan 3. PIN transaksi 4. Penggunaan <i>Signature</i> dalam <i>POST Request</i> dengan enkripsi SHA256 5. Identifikasi perangkat

b. Kebutuhan Data API

Sistem yang berjalan pada Bank ABC menyediakan API yang akan dipakai dalam aplikasi sebagai berikut:

Tabel 3.3 Daftar *Endpoint API Server*

No	Endpoint	Keterangan
1	<i>/get-token</i>	Mendapatkan akses <i>token</i> aplikasi
2	<i>/login</i>	Proses <i>login</i> nasabah ke dalam aplikasi
3	<i>/register</i>	Pendaftaran akun nasabah baik yang sudah mempunyai rekening maupun belum
4	<i>/getpromo</i>	Mendapatkan promo terbaru
5	<i>/getproduk</i>	Menampilkan produk tabungan, kredit dan deposito
6	<i>/getkantor</i>	Menampilkan daftar kantor cabang tersedia
7	<i>/cektrx</i>	Mendapatkan mutasi transaksi
8	<i>/myinfo</i>	Mendapatkan informasi dasar akun
9	<i>/getrek</i>	Menampilkan informasi detail rekening
10	<i>/getjstrf</i>	Mendapatkan daftar metode transfer antar bank
11	<i>/getbank</i>	Mendapatkan daftar bank yang didukung
12	<i>/getlistrx</i>	Mendapatkan daftar rekening tersedia
13	<i>/regtab</i>	Pembukaan rekening tabungan

No	Endpoint	Keterangan
14	<i>/regdep</i>	Pembukaan rekening deposito
15	<i>/reqbill</i>	Permintaan penarikan dana dengan <i>QR Code</i>
16	<i>/cekrektrf</i>	Mengecek rekening tujuan transfer yang berbeda bank
17	<i>/posttrf</i>	Melakukan transaksi transfer antar rekening dan antar bank
18	<i>/cekangs</i>	Mendapatkan daftar angsuran kredit (jika tersedia)
19	<i>/postang krd</i>	Melakukan pembayaran angsuran kredit (jika tersedia)

4. Perancangan Aplikasi

a. *Product Backlog*

Product baclog disusun sebelum dilaksanakannya pembentukan *sprint* yang didiskusikan oleh tim *scrum* melalui media komunikasi yang disepakati. Tim *scrum* yang dibentuk adalah sebagai berikut:

Tabel 3.4 *Scrum Team*

No	Nama	Peran
1	Fery Mohammad Ferdian	<i>Product Owner</i>
2	Dede Yusuf	<i>Scrum Master</i>
3	Awal Ariansyah	<i>Developer</i>

Dengan sifat pengerjaan secara *remote*, maka disepakati media komunikasi yang dipakai adalah sebagai berikut:

- 1) Grup *Whatsapp* sebagai media komunikasi utama.
- 2) *Gmeet* dan *Zoom* sebagai media *video conference*.
- 3) *GitHub* sebagai media *monitoring* implementasi kode.
- 4) *Figma* sebagai media *monitoring* desain dan prototipe.

Struktur dalam penyusunan rancangan *product backlog* adalah sebagai berikut:

Tabel 3.5 Struktur *Product Backlog*

Prioritas	<i>User Story</i>	<i>Backlog Item</i>
1	Narasi sebagai pengguna untuk mendemonstrasikan alur penggunaan sistem	<i>Item</i> atau nama fitur atau fungsi secara umum dari produk/aplikasi

b. *Sprint Planning*

Pada tahapan ini direncanakan pekerjaan berdasarkan *product backlog* dan prioritas yang dibuat oleh tim *scrum*. Durasi *sprint* yang disepakati adalah dengan total waktu 4 minggu atau satu bulan dengan 4 kali *sprint* yang dimulai pada tanggal 01 Maret 2023 - 28 Maret 2023.

c. *Sprint Backlog*

Sprint backlog merupakan hasil dari tahapan *sprint planning* yang berisikan daftar *product backlog* dengan pemecahan pekerjaan beserta *role* dalam penyelesaiannya. Struktur rancangan *sprint backlog* adalah sebagai berikut:

Tabel 3.6 Struktur *Sprint Backlog*

<i>Item</i>	<i>Task</i>	<i>Est.</i>
<i>Item</i> dari <i>product backlog</i> berupa fungsi atau fitur secara umum	Daftar pekerjaan untuk menyelesaikan <i>item product backlog</i>	Estimasi penyelesaian <i>task</i> (dalam hari)

d. *Sprint*

Tahap *sprint* merupakan tahapan eksekusi dari pekerjaan yang terdapat pada *sprint backlog*. *Sprint* dilakukan dengan waktu terbatas dengan konsisten melakukan *daily scrum* setiap harinya.

e. *Daily Scrum*

Developer setiap hari selama 1-15 menit sebelum atau sesudah dalam melaksanakan *task sprint* mendiskusikan beberapa hal kepada *scrum master* seperti:

- 1) Apa yang dikerjakan kemarin
- 2) Apa yang dikerjakan hari ini
- 3) Permasalahan yang dialami

f. *Sprint Review*

Peninjauan *sprint* dilakukan di akhir *sprint* dengan diskusi antara *developer*, *scrum master* dan *product owner* untuk mendapatkan umpan balik dengan melihat sejauh mana dan fitur apa yang telah diselesaikan *developer*.

g. *Sprint Retrospective*

Pada tahapan ini, tim *scrum* berkumpul untuk mengevaluasi proses dari *sprint* yang baru saja selesai. Tujuannya adalah untuk menemukan cara untuk memperbaiki proses dan meningkatkan kinerja tim di *sprint* berikutnya. Pertemuan ini biasanya diadakan pada akhir *sprint*, setelah *sprint review*. Perbedaan antara *sprint*

review dan *sprint retrospective* adalah fokus evaluasi yaitu fokus pada hasil dan fokus pada proses.

5. Pengujian

Peneliti menguji aspek dari aplikasi yaitu pengujian kegunaan aplikasi. Aplikasi m-banking Bank ABC diuji dengan menggunakan evaluasi heuristik dengan membagi 5 aspek *usability* (*Learnability, Efficiency, Memorability, Errors, Satisfaction*) menjadi 4 pertanyaan untuk masing-masing aspeknya dengan menggunakan skala penilaian *severity ratings* (0-4).

Tabel 3.7 Daftar Kuesioner Evaluasi Heuristik

No	Aspek	Pertanyaan	Parameter (0-4)
1	<i>Learnability</i>	Cara penggunaan	Sangat paham - sangat sulit
		Adaptasi	Sangat adaptif - sangat sulit
		Penemuan fitur	Sangat mudah - sangat sulit
		Panduan deskriptif	Sangat membantu - sangat kurang membantu
2	<i>Efficiency</i>	Kecepatan akses menu	Sangat cepat - sangat lambat
		Penemuan informasi	Sangat mudah - sangat sulit
		Penggunaan waktu	Sangat hemat - sangat boros
		Langkah menuju aksi	Sangat sedikit - sangat banyak
3	<i>Memorability</i>	Penggunaan setelah jeda	Sangat mudah ingat - Sangat mudah lupa
		Penemuan fitur yang sering dipakai setelah jeda	Sangat mudah ditemukan - sangat sulit ditemukan

No	Aspek	Pertanyaan	Parameter (0-4)
		Peninjauan informasi yang dimasukkan	Sangat informatif - sangat kurang informatif
		Lokasi menu	Sangat mudah diingat - sangat sulit diingat
4	<i>Errors</i>	Frekuensi kesalahan penggunaan fitur	Sangat jarang - sangat sering
		Perbaikan aksi yang telah dilakukan	Sangat mudah - sangat sulit
		Kejelasan pesan kesalahan	Sangat jelas - sangat kurang jelas
		Frekuensi memulai ulang aplikasi karena kesalahan	Sangat jarang - sangat sering
5	<i>Satisfaction</i>	Kinerja aplikasi	Sangat puas - sangat kurang puas
		Penyelesaian aksi dalam fitur	Sangat puas - sangat kurang puas
		Kesenangan dalam penggunaan	Sangat menyenangkan - sangat menyedihkan
		Kenyamanan penggunaan	Sangat nyaman - sangat kurang nyaman

STMIK KOMPUTAMA
MAJENANG

BAB IV

HASIL DAN PEMBAHASAN

A. Pembentukan *Product Backlog*

Berdasarkan wawancara saat identifikasi permasalahan, terdapat data layanan utama yang disampaikan *Project Manager* yang menyatakan bahwa “Layanan di tahap awal, layanan aplikasi *mobile* yang memudahkan mereka mendapatkan informasi. Nasabah bisa melihat saldo, bisa melihat transaksi di bank tersebut, bisa melakukan pemindahan dana baik ke rekening lain maupun ke simpanan berjangka (*deposito*). Tambahan ini juga diharapkan bisa punya fitur untuk melakukan pembukaan rekening, baik untuk nasabah yang transisi maupun nasabah baru”.

Peneliti menggunakan data tersebut dan juga data respon API untuk kemudian diubah ke dalam *user story* dengan berkaca pada praktik aplikasi bank-bank pada umumnya.

1. *User Story*

User story dibuat berdasarkan data wawancara diatas dan khususnya data respon API untuk memahami data apa saja yang tersedia untuk keperluan integrasi dengan sistem *backend* yang sudah berjalan.

a. *Login*

User story: Sebagai seorang nasabah, saya ingin melakukan *login* sehingga saya dapat mengakses dan menggunakan layanan dan produk perbankan melalui aplikasi m-banking Bank ABC.

Tabel 4.1 *User Story Login*

No	Alur	Keterangan
1	<i>Actor</i>	Nasabah
2	<i>Objective</i>	Nasabah melakukan proses <i>login</i>
3	<i>Pre-Condition</i>	Nasabah sudah terdaftar di sistem
4	<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Nasabah membuka aplikasi 2. Nasabah memasukkan email, kata sandi dan menekan tombol masuk 3. Nasabah dialihkan ke layar beranda
5	<i>Alternative</i>	Nasabah yang belum terdaftar dapat menekan tombol daftar
6	<i>Post Condition</i>	Nasabah masuk ke layar beranda dengan kondisi <i>login</i>

b. Pendaftaran

User story: Sebagai seorang nasabah, saya ingin melakukan pendaftaran pengguna aplikasi sehingga saya dapat mendapatkan akses ke layanan dan produk perbankan yang tersedia melalui aplikasi m-banking Bank ABC.

Tabel 4.2 *User Story Pendaftaran*

No	Alur	Keterangan
1	<i>Actor</i>	Nasabah
2	<i>Objective</i>	Nasabah melakukan pendaftaran akun
3	<i>Pre-Condition</i>	-
4	<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Nasabah menekan tombol untuk pendaftaran 2. Nasabah memilih apakah sudah memiliki rekening atau belum 3. Nasabah memasukkan data yang diperlukan untuk pendaftaran 4. Nasabah menekan tombol daftar 5. Nasabah dialihkan ke layar <i>login</i>
5	<i>Alternative</i>	-
6	<i>Post Condition</i>	Nasabah kembali ke layar <i>login</i>

c. Informasi Rekening

User story: Sebagai seorang nasabah, saya ingin melihat informasi rekening (tabungan / deposito / kredit) sehingga saya dapat mengetahui informasi rekening yang saya miliki, seperti saldo, nama rekening, nomor rekening, limit kredit, bunga, tanggal jatuh tempo dan lainnya.

Tabel 4.3 *User Story* Informasi Rekening

No	Alur	Keterangan
1	<i>Actor</i>	Nasabah
2	<i>Objective</i>	Nasabah melihat informasi rekening (tabungan / deposito / kredit)
3	<i>Pre-Condition</i>	Nasabah sudah berada di layar beranda
4	<i>Main Flow</i>	1. Nasabah menekan ikon menu informasi (tabungan / deposito / kredit) 2. Nasabah melihat informasi rekening
5	<i>Alternative</i>	-
6	<i>Post Condition</i>	Nasabah berada di layar detail informasi rekening (tabungan / deposito / kredit)

d. Informasi Produk

User story: Sebagai seorang nasabah, saya ingin melihat informasi produk (tabungan / deposito / kredit) sehingga saya dapat membandingkan dan memutuskan produk keuangan yang sesuai dengan kebutuhan dan tujuan keuangan saya.

Tabel 4.4 *User Story* Informasi Produk

No	Alur	Keterangan
1	<i>Actor</i>	Nasabah
2	<i>Objective</i>	Nasabah melihat informasi produk (tabungan / deposito / kredit)

No	Alur	Keterangan
3	<i>Pre-Condition</i>	Nasabah sudah berada di layar beranda
4	<i>Main Flow</i>	1. Nasabah menekan ikon menu produk (tabungan / deposito / kredit) 2. Nasabah mencari dan melihat produk yang diinginkan
5	<i>Alternative</i>	-
6	<i>Post Condition</i>	Nasabah berada pada layar detail produk (tabungan / deposito / kredit)

e. Pembukaan Rekening

User story: Sebagai seorang nasabah, saya ingin membuka rekening (tabungan / deposito / kredit) sehingga saya dapat mendapatkan layanan dan produk perbankan yang sesuai dengan kebutuhan dan tujuan keuangan saya.

Tabel 4.5 *User Story* Pembukaan Rekening

No	Alur	Keterangan
1	<i>Actor</i>	Nasabah
2	<i>Objective</i>	Nasabah melakukan pembukaan rekening (tabungan / deposito)
3	<i>Pre-Condition</i>	Nasabah berada di layar beranda
4	<i>Main Flow</i>	1. Nasabah menekan ikon menu buka rekening 2. Nasabah memilih jenis pembukaan rekening (tabungan/deposito/kredit) 3. Nasabah menyetujui syarat & ketentuan dengan menekan tombol lanjutkan 4. Nasabah mengisi data yang diperlukan untuk pembukaan rekening 5. Nasabah menekan tombol buka rekening 6. Nasabah dialihkan ke layar beranda
5	<i>Alternative</i>	-

No	Alur	Keterangan
6	<i>Post Condition</i>	Nasabah berada di layar beranda dengan notifikasi pembukaan rekening berhasil

f. Mutasi Rekening

User story: Sebagai seorang nasabah, saya ingin melihat mutasi rekening (tabungan / deposito / kredit) sehingga saya dapat memantau dan memahami riwayat transaksi pada rekening (tabungan / deposito / kredit) saya.

Tabel 4.6 *User Story* Mutasi Rekening

No	Alur	Keterangan
1	<i>Actor</i>	Nasabah
2	<i>Objective</i>	Nasabah melihat mutasi rekening (tabungan / deposito / kredit)
3	<i>Pre-Condition</i>	Nasabah berada pada layar mutasi
4	<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Nasabah memilih jenis mutasi rekening (tabungan / deposito / kredit) 2. Nasabah memilih rekening dan mengisikan tanggal awal, tanggal akhir 3. Nasabah menekan tombol cek mutasi dan melihat mutasi yang tersedia
5	<i>Alternative</i>	<ol style="list-style-type: none"> 1. Nasabah kembali ke menu beranda 2. Nasabah menekan menu mutasi
6	<i>Post Condition</i>	Nasabah melihat detail informasi mutasi rekening yang dipilih sesuai jangka waktu yang dimasukkan

g. Transfer

User story: Sebagai seorang nasabah, saya ingin melakukan transfer (antar rekening / antar bank) sehingga saya dapat

mengirimkan uang ke (rekening lain / rekening bank lain) dengan mudah dan cepat melalui aplikasi.

Tabel 4.7 *User Story* Transfer

No	Alur	Keterangan
1	<i>Actor</i>	Nasabah
2	<i>Objective</i>	Nasabah melakukan transfer uang antar rekening (sesama bank/bank lain)
3	<i>Pre-Condition</i>	<ol style="list-style-type: none"> 1. Nasabah berada pada layar transaksi 2. Nasabah memilih jenis transfer antar rekening
4	<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Nasabah memilih rekening sumber, rekening tujuan, dan keterangan transaksi 2. Nasabah dialihkan ke layar nominal untuk mengisi nominal transaksi 3. Nasabah dialihkan ke layar konfirmasi untuk meninjau ulang informasi yang dimasukkan 4. Nasabah memasukkan PIN transaksi untuk konfirmasi 5. Nasabah menekan tombol kirim dan dialihkan ke layar bukti transaksi
5	<i>Alternative</i>	<ol style="list-style-type: none"> 1. Nasabah berada pada layar beranda dan menekan menu transaksi 2. Nasabah memilih jenis transfer antar bank 3. Nasabah memilih metode transfer antar bank 4. Nasabah dialihkan ke layar detail transfer antar bank untuk memilih bank tujuan, keterangan, dan nomor rekening tujuan 5. Nasabah dialihkan ke layar nominal untuk memasukkan nominal transfer 6. Nasabah dialihkan ke layar konfirmasi untuk meninjau ulang informasi yang dimasukkan

No	Alur	Keterangan
		7. Nasabah memasukkan PIN transaksi untuk konfirmasi
		8. Nasabah menekan tombol kirim dan dialihkan ke layar bukti transaksi
6	<i>Post Condition</i>	Nasabah berada pada layar bukti transaksi dengan opsi kembali ke layar beranda atau menyimpan dan bagikan bukti transaksi berupa berkas format PDF.

h. Penarikan Dana

User story: Sebagai seorang nasabah, saya ingin melakukan penarikan dana sehingga saya dapat mengambil uang dari rekening tabungan atau deposito saya dengan mudah dan cepat.

Tabel 4.8 *User Story* Penarikan Dana

No	Alur	Keterangan
1	<i>Actor</i>	Nasabah
2	<i>Objective</i>	Nasabah melakukan penarikan dana menggunakan <i>QR Code</i>
3	<i>Pre-Condition</i>	Nasabah berada pada layar transaksi
4	<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Nasabah memilih menu tarik tunai 2. Nasabah memilih rekening sumber dan nominal penarikan 3. Nasabah menekan tombol lanjutkan 4. Nasabah dialihkan ke layar permintaan tarik tunai dengan informasi <i>QR Code</i> beserta kode OTP, dan waktu kadaluwarsa
5	<i>Alternative</i>	Pada layar beranda nasabah memilih menu tarik tunai
6	<i>Post Condition</i>	Nasabah berada pada layar permintaan tarik tunai

i. Pembayaran Angsuran Kredit

User story: Sebagai seorang nasabah, saya ingin membayar angsuran kredit sehingga saya dapat membayar angsuran kredit saya tepat waktu dan menghindari denda atau bunga tambahan.

Tabel 4.9 *User Story* Pembayaran Angsuran Kredit

No	Alur	Keterangan
1	<i>Actor</i>	Nasabah
2	<i>Objective</i>	Nasabah melakukan pembayaran tagihan kredit
3	<i>Pre-Condition</i>	Nasabah berada pada layar transaksi
4	<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Nasabah memilih menu bayar kredit 2. Nasabah melihat dan memilih daftar tagihan yang tersedia 3. Nasabah dialihkan ke menu pembayaran tagihan kredit 4. Nasabah memilih rekening sumber untuk melakukan pembayaran dan dialihkan ke layar transaksi
5	<i>Alternative</i>	-
6	<i>Post Condition</i>	Nasabah berada di layar transaksi

j. Informasi Lainnya

User story: Sebagai seorang nasabah, saya ingin melihat informasi (akun / kantor cabang / bantuan) sehingga saya dapat memperoleh informasi tentang akun saya, kantor cabang terdekat, dan bantuan yang tersedia.

Tabel 4.10 *User Story* Informasi Lainnya

No	Alur	Keterangan
1	<i>Actor</i>	Nasabah
2	<i>Objective</i>	Nasabah melihat informasi lainnya (akun / kantor cabang / bantuan)
3	<i>Pre-Condition</i>	Nasabah berada di layar informasi

No	Alur	Keterangan
4	Main Flow	1. Nasabah memilih jenis informasi lainnya (akun / kantor cabang / bantuan) yang ingin dilihat 2. Nasabah melihat informasi lainnya
5	Alternative	-
6	Post Condition	Nasabah berada pada layar detail informasi lainnya

k. Pengaturan

User story: Sebagai seorang nasabah, saya ingin mengubah pengaturan seperti PIN transaksi sehingga saya dapat mengubah PIN transaksi dan memastikan keamanan akun m-banking saya.

Tabel 4.11 *User Story* Pengaturan PIN Transaksi

No	Alur	Keterangan
1	Actor	Nasabah
2	Objective	Nasabah mengubah PIN transaksi
3	Pre-Condition	Nasabah di layar informasi
4	Main Flow	1. Nasabah memilih menu pengaturan 2. Nasabah memilih menu ubah PIN transaksi di antara pengaturan yang tersedia 3. Nasabah memasukkan PIN lama dan PIN baru 4. Nasabah menekan tombol ubah PIN transaksi
5	Alternative	-
6	Post Condition	Nasabah berada pada layar pengaturan dengan PIN transaksi baru

2. Product Backlog

Product backlog disusun sebagai landasan dalam proses *sprint* dan merupakan hasil dari *user story*. *Product backlog* disusun menggunakan struktur *product backlog* serta berdasarkan hasil dari *user story*.

Tabel 4.12 *Product Backlog*

Prioritas	User Story	Backlog Item
1	-	Peta alur navigasi layar
2	Sebagai seorang nasabah, saya ingin melakukan <i>login</i>	Fitur <i>login</i> nasabah
3	Sebagai seorang nasabah, saya ingin melihat informasi rekening (tabungan / deposito / kredit)	Fitur menampilkan informasi rekening nasabah
4	Sebagai seorang nasabah, saya ingin melihat mutasi rekening (tabungan / deposito / kredit)	Fitur menampilkan informasi transaksi/mutasi rekening
5	Sebagai seorang nasabah, saya ingin melihat informasi produk (tabungan / deposito / kredit)	Fitur menampilkan informasi produk perbankan
6	Sebagai seorang nasabah, saya ingin melakukan transfer (antar rekening / antar bank)	Fitur transaksi antar rekening, antar bank, tarik tunai dan pembayaran kredit
	Sebagai seorang nasabah, saya ingin melakukan penarikan dana	
	Sebagai seorang nasabah, saya ingin membayar angsuran kredit	
7	Sebagai seorang nasabah, saya ingin membuka rekening (tabungan / deposito / kredit)	Fitur pembukaan rekening
8	Sebagai seorang nasabah, saya ingin melakukan pendaftaran aplikasi	Fitur pendaftaran nasabah sebagai pengguna aplikasi

Prioritas	User Story	Backlog Item
9	Sebagai seorang nasabah, saya ingin melihat informasi (akun / kantor cabang / bantuan)	Fitur melihat informasi umum
10	Sebagai seorang nasabah, saya ingin mengubah pengaturan seperti PIN transaksi	Fitur pengaturan aplikasi

B. *Sprint I*

Sprint I mengambil 2 item *backlog* yaitu peta alur navigasi antar layar aplikasi dan fitur *login* nasabah. Pada pembuatan peta navigasi layar untuk memetakan bagaimana nasabah berpindah layar serta fitur *login* di mana merupakan fitur dengan prioritas tinggi agar nasabah dapat menggunakan aplikasi. Ketentuan jadwal *sprint I* adalah sebagai berikut:

Tabel 4.13 Jadwal *Sprint I*

Periode	Jumlah hari	Jam kerja/hari
01 Maret 2023 - 07 Maret 2023	7 hari	± 10 jam

1. *Sprint Planning*

Pemetaan alur navigasi layar direncanakan secara sederhana menggunakan bagan sederhana dan panah alur berdasarkan referensi aplikasi m-banking bank-bank lain. Fitur *login* memiliki estimasi waktu lebih banyak karena pertimbangan dalam mengatur lingkungan pengembangan proyek. *Product backlog* yang dieksekusi pada *sprint I* berjumlah 2 item *backlog*.

Tabel 4.14 *Sprint Backlog Sprint I*

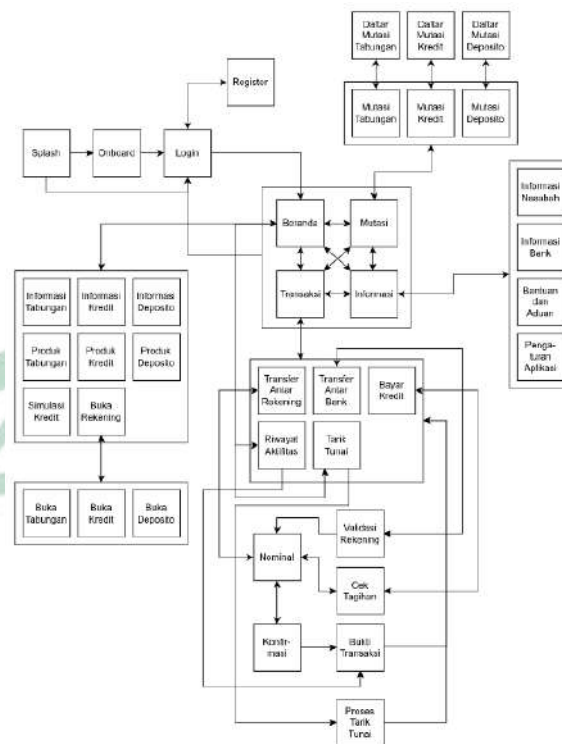
<i>Item</i>	<i>Task</i>	<i>Est. (hari)</i>
Peta alur navigasi layar	<ol style="list-style-type: none"> 1. Menyiapkan <i>resources</i> sistem 2. Menyusun peta layar aplikasi 3. Melakukan <i>review</i> alur 	1
Fitur <i>login</i> nasabah	<ol style="list-style-type: none"> 1. Persiapan proyek dan lingkungan pengembangan 2. Membuat layar <i>splash</i> 3. Membuat layar <i>onboard</i> 4. Membuat layar <i>login</i> 5. Menerapkan validasi dan integrasi API <i>backend</i> 	6

2. *The Sprint*

a. Peta alur navigasi antar layar aplikasi

Pemetaan navigasi layar berdasarkan data yang disajikan oleh API yang tersedia, *user story* dan observasi terhadap referensi desain. Hasil peta navigasi adalah sebagai berikut dengan panah sebagai penunjuk arah alur navigasi:

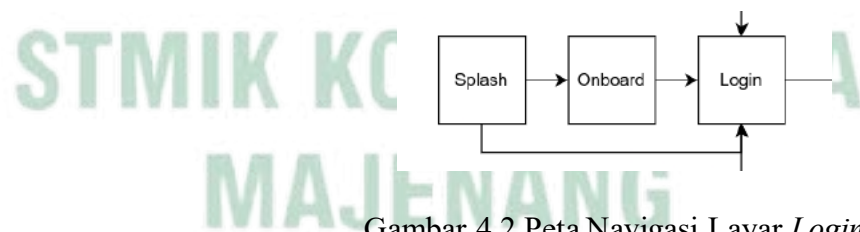
STMIK KOMPUTAMA
MAJENANG



Gambar 4.1 Peta Alur Navigasi Layar

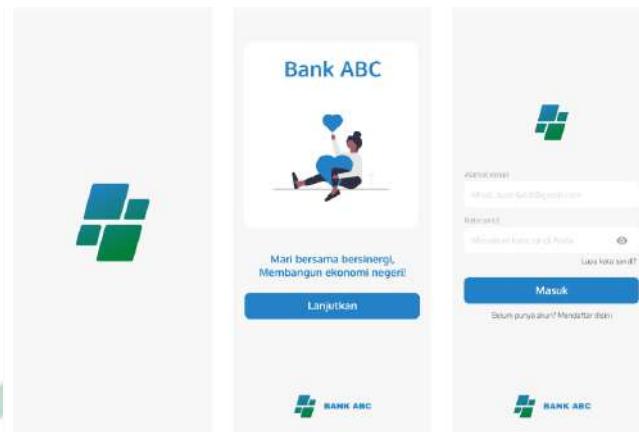
b. Fitur *login* nasabah

Persiapan proyek dan lingkungan pengembangan dilakukan dengan inisiasi Node.js, React Native dan IDE Visual Studio Code. Persiapan desain dilakukan dengan melakukan pengumpulan aset ke dalam Figma dari data kebutuhan yang telah dikumpulkan.



Gambar 4.2 Peta Navigasi Layar *Login*

Layar *login* pada peta navigasi layar terhubung dengan 2 layar lainnya yaitu layar *splash* dan *onboard*. Hasil dari layar *splash*, *onboard*, dan *login* adalah sebagai berikut:



Gambar 4.3 Antarmuka Layar *Splash*, *Onboard* dan *Login*

Ketiga layar terhubung secara berurutan dalam pemetaan alur navigasi dengan layar *onboard* yang hanya muncul satu kali setelah tombol “Lanjutkan” ditekan. Tabel interaksi antara pengguna dengan layar adalah sebagai berikut:

Tabel 4.15 Interaksi Layar *Onboard*

No	Elemen	Label	Aksi
1	Button	Lanjutkan	Ke layar <i>login</i> dan hapus layar <i>onboard</i>

Tabel 4.16 Interaksi Layar *Login*

No	Elemen	Label	Aksi
1	Input box	Alamat email	Memasukkan alamat email
		Kata sandi	Memasukkan kata sandi
2	Icon	Mata	Toggle teks masukan kata sandi
3	Text	Lupa kata sandi?	Popup lupa kata sandi
		Belum punya akun? mendaftar di sini	Ke layar <i>register</i>
4	Button	Masuk	1. Ke layar <i>home</i> 2. Popup <i>login</i> gagal

Integrasi API:

Tabel 4.17 Integrasi API Layar *Sprint I*

No	Layar	API <i>Endpoint</i>	Keterangan
1	<i>Login</i>	<i>/get-token</i>	Mengambil <i>token</i> akses aplikasi
		<i>/login</i>	Mengirimkan masukan email dan kata sandi untuk proses <i>login</i>

Skema validasi masukan:

Tabel 4.18 Skema Validasi API Layar *Login*

No	<i>Input</i> (Wajib diisi*)	Aturan
1	Email*	1. Terdapat simbol '@' 2. Terdapat alamat domain
2	Kata sandi*	1. Minimal 8 karakter 2. Minimal 1 huruf kapital, huruf non-kapital, angka, serta simbol

3. *Sprint Review*

Sprint review pada *sprint I* dilaksanakan pada tanggal 07 Maret 2023 di mana *developer* mempresentasikan hasil dari *sprint I* kepada *product owner* untuk mendapatkan peninjauan terhadap hasil dari *sprint I*.

Hasil dari *sprint* atau *increment* pada *sprint I* berupa:

- Peta alur navigasi layar aplikasi.
- Layar *splash*, *onboard* dan *login*.
- Fitur, validasi dan integrasi API proses *login*.

Detail *sprint review* terdapat pada dokumen lampiran.

4. *Sprint Retrospective*

Sprint retrospective membahas proses dari *sprint* yang dilakukan yang berupa kendala atau hal apa yang dapat ditingkatkan untuk *sprint* selanjutnya. Hasil dari *sprint retrospective* pada *sprint* I membahas hal-hal dalam:

- a. Peningkatan komunikasi dalam grup Whatsapp.
- b. Penambahan dan perbaikan yang diusulkan *product owner* segera diselesaikan sebelum memulai *sprint* berikutnya.

C. *Sprint II*

Sprint II mengambil 3 *item backlog* meliputi fitur menampilkan informasi rekening, informasi transaksi/mutasi, serta informasi produk perbankan. Ketentuan jadwal *sprint II* adalah sebagai berikut:

Tabel 4.19 Jadwal *Sprint II*

Periode	Jumlah hari	Jam kerja/hari
08 Maret 2023 - 14 Maret 2023	7 hari	± 10 jam

1. *Sprint Planning*

Hasil *sprint review* pada *sprint* I yaitu penambahan layar *loading* dan layar-layar akar dieksekusi ke dalam 1 *item backlog* tambahan sehingga total *item backlog* yang dieksekusi pada *sprint II* adalah 4 *item*.

Tabel 4.20 *Sprint Backlog Sprint II*

Item	Task	Est. (hari)
Penambahan layar <i>loading</i> dan layar-layar akar (<i>item backlog</i> tambahan)	1. Membuat layar <i>loading</i> 2. Membuat layar beranda 3. Membuat layar mutasi 4. Membuat layar transaksi 5. Membuat layar informasi	3

<i>Item</i>	<i>Task</i>	<i>Est. (hari)</i>
	6. Mengintegrasikan layar beranda dengan API <i>backend</i>	
Fitur menampilkan informasi rekening	1. Membuat layar informasi rekening tabungan 2. Membuat layar informasi rekening kredit 3. Membuat layar informasi rekening deposito 4. Mengintegrasikan layar dengan API <i>backend</i>	1
Fitur menampilkan informasi transaksi	1. Membuat layar mutasi rekening tabungan 2. Membuat layar mutasi rekening kredit 3. Membuat layar mutasi rekening deposito 4. Mengintegrasikan layar dengan API <i>backend</i>	2
Fitur menampilkan informasi produk perbankan	1. Membuat layar produk perbankan tabungan 2. Membuat layar produk perbankan kredit 3. Membuat layar produk perbankan deposito 4. Mengintegrasikan layar dengan API <i>backend</i>	1

2. The Sprint

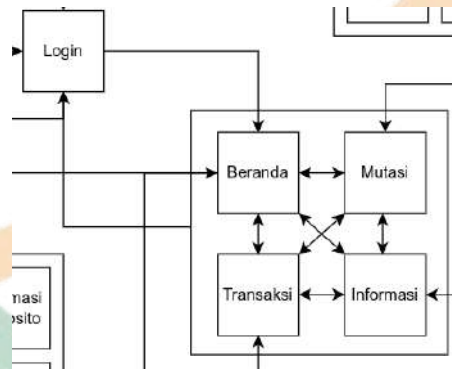
a. Penambahan layar *loading* dan layar-layar akar

Layar *loading* merupakan layar yang akan dipakai di setiap transisi yang membutuhkan waktu perpindahannya, oleh karena itu layar *loading* diharapkan tidak terlalu kompleks dan tidak terlalu monoton serta tidak terdapat interaksi pengguna di dalamnya.

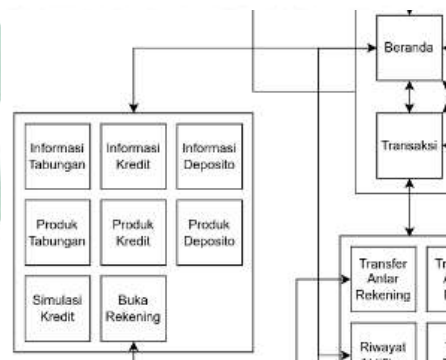


Gambar 4.4 Antarmuka Layar *Loading*

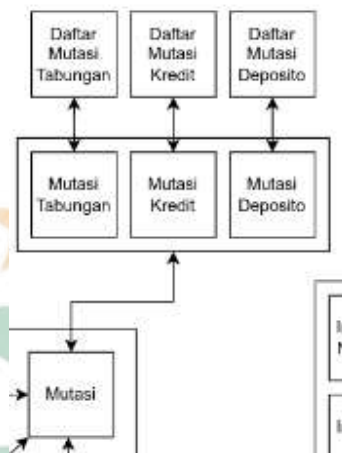
Layar-layar akar mengacu pada layar yang menghubungkan layar *login* dengan layar sub menu lainnya. Jika melihat pada peta navigasi alur layar, layar-layar akar adalah sebagai berikut:



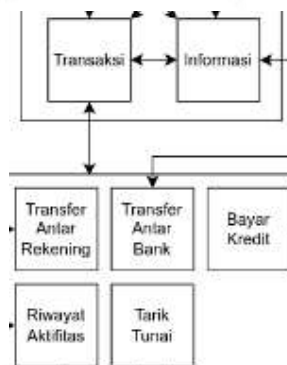
Gambar 4.5 Peta Navigasi Layar-Layar Akar



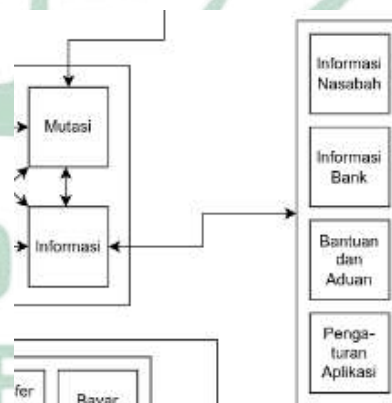
Gambar 4.6 Peta Navigasi Layar Beranda



Gambar 4.7 Peta Navigasi Layar Mutasi

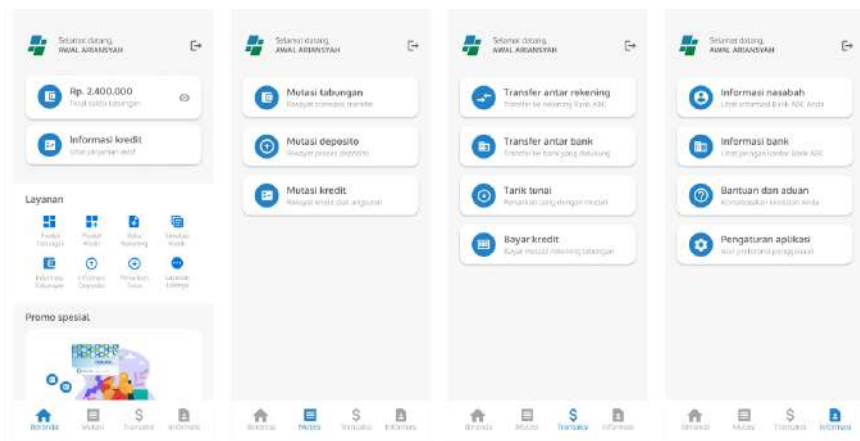


Gambar 4.8 Peta Navigasi Layar Transaksi



Gambar 4.9 Peta Navigasi Layar Informasi

Hasil dari layar-layar akar yang terdiri dari layar beranda, layar mutasi, layar transaksi, serta layar informasi berdasarkan desain dari peta navigasi layar adalah sebagai berikut:



Gambar 4.10 Antarmuka Layar-Layar Akar

Layar beranda memiliki kompleksitas lebih dan konten yang lebih banyak dikarenakan layar yang pertama kali diakses pengguna setelah *login*, hal ini berkaca pula pada referensi pada aplikasi-aplikasi m-banking pada umumnya. Sementara layar mutasi, transaksi, dan informasi memiliki tata letak yang sama. Keempat layar tersebut saling terhubung melalui perantara menu bawah atau *bottom tab menu* serta memiliki elemen *header* yang sama. Tabel-tabel interaksi layar adalah sebagai berikut:

Tabel 4.21 Interaksi *Header* dan *Bottom Tab Menu*

No	Elemen	Label	Aksi
1	Icon	Keluar pintu	Keluar ke layar <i>login</i>
2	Icon	Beranda	Ke layar beranda
3	Icon	Mutasi	Ke layar mutasi
4	Icon	Transaksi	Ke layar transaksi
5	Icon	Informasi	Ke layar informasi

Tabel 4.22 Interaksi Layar Beranda

No	Elemen	Label	Aksi
1	Card	Total saldo tabungan	Ke layar informasi tabungan

No	Elemen	Label	Aksi
2	Icon	Mata	Toggle saldo tabungan
3	Card	Informasi kredit	Ke layar informasi kredit
4	Icon	Produk tabungan	Ke layar informasi produk tabungan
5	Icon	Produk kredit	Ke layar informasi produk kredit
6	Icon	Buka rekening	Ke layar buka rekening
7	Icon	Simulasi kredit	Ke layar simulasi kredit
8	Icon	Informasi tabungan	Ke layar informasi tabungan
9	Icon	Informasi deposito	Ke layar informasi deposito
10	Icon	Penarikan tunai	Ke layar tarik tunai
11	Icon	Layanan lainnya	Menampilkan <i>pop up</i> layanan lainnya
12	Card	Promo spesial	Ke web informasi promo

Tabel 4.23 Interaksi Layar Mutasi

No	Elemen	Label	Aksi
1	Card	Mutasi tabungan	Ke layar mutasi tabungan
2	Card	Mutasi deposito	Ke layar mutasi deposito
3	Card	Mutasi kredit	Ke layar mutasi kredit

Tabel 4.24 Interaksi Layar Transaksi

No	Elemen	Label	Aksi
1	Card	Transfer antar rekening	Ke layar transaksi antar rekening
2	Card	Transfer antar bank	Ke layar transaksi antar bank
3	Card	Tarik tunai	Ke layar tarik tunai
4	Card	Bayar kredit	Ke layar angsuran kredit

Tabel 4.25 Interaksi Layar Informasi

No	Elemen	Label	Aksi
1	Card	Informasi nasabah	Ke layar informasi nasabah
2	Card	Informasi bank	Ke layar informasi bank
3	Card	Bantuan dan aduan	Ke layar bantuan

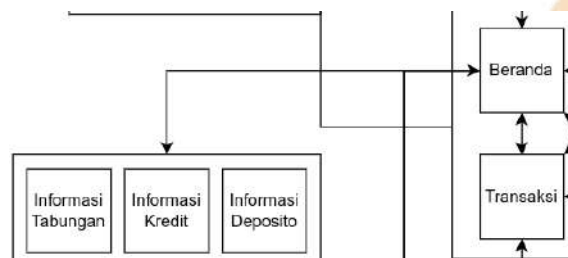
No	Elemen	Label	Aksi
4	Card	Pengaturan aplikasi	Ke layar pengaturan

Tabel 4.26 Integrasi API Layar-Layar Akar

No	Layar	API <i>Endpoint</i>	Keterangan
1	Beranda	<i>/getpromo</i>	Menampilkan promo tersedia

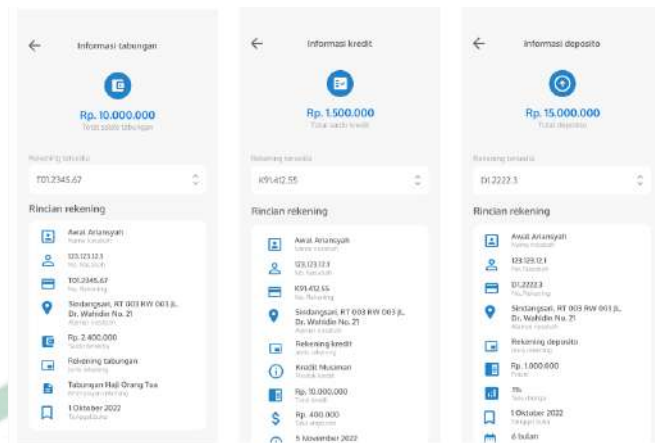
Skema validasi masukan tidak diperlukan karena tidak ada layar yang memerlukan masukan dari pengguna.

b. Fitur menampilkan informasi rekening



Gambar 4.11 Peta Navigasi Layar Informasi Rekening

Layar informasi rekening terdiri dari 3 layar yaitu layar informasi tabungan, layar informasi kredit dan layar informasi deposito. Ketiga layar tersebut secara independen dapat kembali ke layar-layar akar dengan implementasi *header* yang mempunyai tombol kembali. Hasil dari layar informasi rekening adalah sebagai berikut:



Gambar 4.12 Antarmuka Layar-Layar Informasi Rekening

Ketiga layar memiliki tata letak dan elemen-elemen yang sama sehingga memiliki tabel interaksi yang sama. Tabel interaksi layar-layar informasi rekening adalah sebagai berikut:

Tabel 4.27 Interaksi Layar-Layar Informasi

No	Elemen	Label	Aksi
1	<i>Dropdown</i>	Rekening tersedia	Memilih daftar rekening tersedia
2	<i>Icon</i>	Arah kiri	Kembali ke layar akar sebelumnya

Meskipun tidak ada masukan dari pengguna secara eksplisit sehingga tidak ada skema validasi masukan, namun masing-masing layar masih harus menampilkan data detail dari rekening tergantung pada nomor rekening yang dipilih. Integrasi API *backend* yang dipakai sebagai berikut:

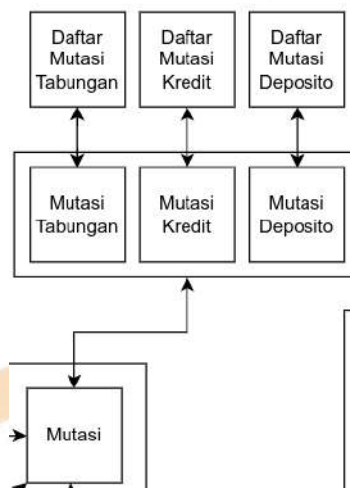
Tabel 4.28 Integrasi API Layar-Layar Informasi Rekening

No	Layar	API <i>Endpoint</i>	Keterangan
1	Informasi tabungan	<i>/getlistrx</i>	Mengambil detail rekening tabungan (<i>app_id=1</i>)

No	Layar	API <i>Endpoint</i>	Keterangan
2	Informasi deposito	<i>/getlistrx</i>	Mengambil detail rekening deposito (<i>app_id=2</i>)
3	Informasi kredit	<i>/getlistrx</i>	Mengambil detail rekening kredit (<i>app_id=3</i>)

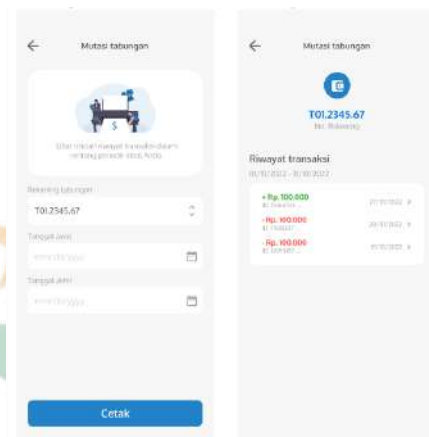
c. Fitur menampilkan informasi transaksi

Layar pada informasi transaksi atau lebih dikenal dengan mutasi terdiri beberapa layar di antaranya yaitu layar mutasi tabungan, layar mutasi kredit, dan layar mutasi deposito.



Gambar 4.13 Peta Navigasi Layar Mutasi

Layar informasi mutasi merupakan turunan dari layar akar mutasi. Dalam layar informasi mutasi, memiliki satu anak layar turunan yaitu layar daftar mutasi yang berisi mutasi dari rekening yang dipilih. Hasil dari implementasi layar mutasi adalah sebagai berikut:



Gambar 4.14 Antarmuka Layar Mutasi dan Daftar Mutasi Tabungan

Implementasi layar mutasi dan daftar mutasi untuk dua kategori lainnya (kredit dan deposito) memiliki implementasi yang sama, bahkan dapat dikatakan hanya membutuhkan satu implementasi seperti pada gambar. Namun, *product owner* menyarankan untuk tetap membuatnya menjadi 3 menu dan layar yang berbeda untuk menghindari kebingungan pengguna. Daftar interaksi yang ada pada layar adalah sebagai berikut:

Tabel 4.29 Interaksi Layar Informasi Mutasi

No	Elemen	Label	Aksi
1	Icon	Arah kiri	Kembali ke layar sebelum
2	Dropdown	Rekening tersedia	Memilih daftar rekening tersedia
3	Input box	Tanggal awal	Memilih tanggal awal
4	Input box	Tanggal akhir	Memilih tanggal akhir
5	Button	Cetak	Ke layar daftar mutasi sesuai masukan pengguna

Dalam layar daftar mutasi, hanya terdapat interaksi kembali pada menu sebelumnya. Masukan dari pengguna pada layar

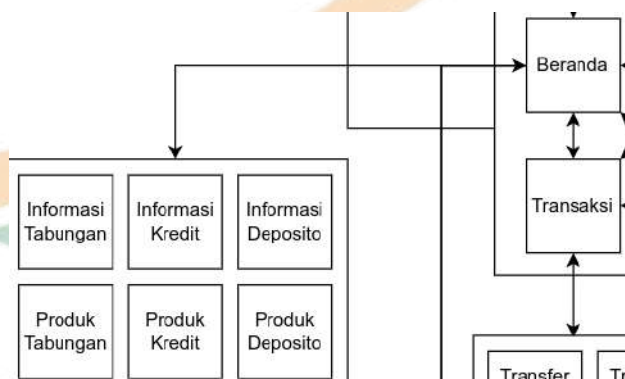
informasi mutasi hanya berupa pemilihan rekening dan tanggal pada kalender, sehingga tidak terdapat skema validasi. Namun untuk mengambil informasi data mutasi, perlu integrasi terhadap *API backend* yang disediakan.

Tabel 4.30 Integrasi API Layer Daftar Mutasi

No	Layar	API <i>Endpoint</i>	Keterangan
1	Daftar mutasi	<i>/cektrx</i>	Menampilkan daftar mutasi sesuai masukan,

d. Fitur menampilkan informasi produk perbankan

Fitur menampilkan informasi produk perbankan memiliki tiga layar berdasarkan peta navigasi, yaitu layar produk tabungan, layar produk kredit, serta layar produk deposito.



Gambar 4.15 Peta Navigasi Layar Produk Perbankan

Implementasi dari fitur untuk menampilkan informasi produk perbankan adalah sebagai berikut:



Gambar 4.16 Antarmuka Layar Produk Tabungan

Produk kredit dan produk deposito memiliki antarmuka yang sama dengan perbedaan di isi produk. Interaksi yang ada hanya pada *icon* panah kiri untuk kembali ke layar sebelumnya. Integrasi API *backend* yang digunakan untuk mendapatkan data produk adalah sebagai berikut:

Tabel 4.31 Integrasi API Layar Informasi Produk

No	Layar	API <i>Endpoint</i>	Keterangan
1	Produk tabungan	<i>/getproduk</i>	Menampilkan daftar produk tabungan (<i>app_id</i> =1)
2	Produk deposito	<i>/getproduk</i>	Menampilkan daftar produk deposito (<i>app_id</i> =2)
3	Produk kredit	<i>/getproduk</i>	Menampilkan daftar produk kredit (<i>app_id</i> =3)

3. *Sprint Review*

Sprint II memuat perubahan *product backlog* sebagai perbaikan dari *Sprint I*. *Sprint review* dari *Sprint II* yang dilaksanakan pada tanggal 14 Maret 2023 memuat beberapa evaluasi dari hasil *sprint* seperti pengubahan *icon*, tampilan menu maupun karakteristik layar seperti pada

layar informasi mutasi. Hasil *Increment* dari *sprint II* adalah sebagai berikut:

- a. Layar *loading* dan layar-layar akar meliputi layar beranda, mutasi, transaksi, serta informasi.
- b. Layar informasi rekening meliputi layar informasi tabungan, deposito, dan kredit.
- c. Layar informasi mutasi dan daftar mutasi dari rekening tabungan, deposito, dan kredit.
- d. Layar produk-produk perbankan dari bank ABC meliputi produk tabungan, deposito, dan kredit.

4. *Sprint Retrospective*

Pembahasan proses *sprint* atau *sprint retrospective* dari *sprint II* menyimpulkan untuk tetap konsisten dalam pelaksanaan eksekusi *sprint* pada *sprint* selanjutnya, yaitu *sprint III*.

D. *Sprint III*

Melanjutkan *sprint* sebelumnya, *sprint III* memprioritaskan evaluasi pada *increment*. *Increment* yang ditambahkan dari *sprint II* adalah layar beranda dan layar produk untuk diperbaiki berdasarkan *sprint review II*. Ketentuan jadwal *sprint III* adalah sebagai berikut:

Tabel 4.32 Jadwal *Sprint III*

Periode	Jumlah hari	Jam kerja/hari
15 Maret 2023 - 21 Maret 2023	7 hari	± 14 jam

1. *Sprint Planning*

Dengan tetap melihat pada prioritas dari *product backlog* selain dari *item backlog* tambahan, maka *item backlog* yang dieksekusi selanjutnya adalah fitur berbagai jenis transaksi dan fitur pembukaan rekening. Fitur transaksi sangat rumit sehingga diperlukan waktu lebih banyak sementara melihat dari data API *backend*, pembukaan rekening berupa rekening tabungan dan deposito tidak memerlukan banyak masukan serta pembukaan rekening kredit hanya dialihkan ke situs web untuk melakukan pendaftaran di dalam situs web Bank ABC.

Product backlog yang dieksekusi adalah 2 *item backlog* dan 1 *item backlog* tambahan sehingga total menjadi 3 *item backlog* pada *sprint* III. Total kumulatif seluruh *item backlog* pada *product backlog* dimulai dari *sprint* I hingga *sprint* III berjumlah 9 *item backlog*.

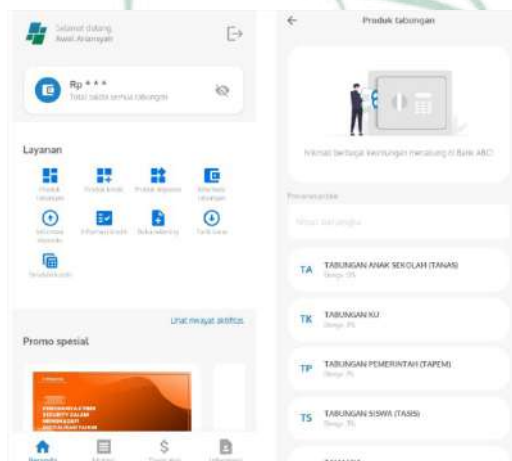
Tabel 4.33 *Sprint Backlog Sprint* III

<i>Item</i>	<i>Task</i>	<i>Est. (hari)</i>
Perbaikan layar (<i>item backlog</i> tambahan)	1. Memperbaiki layar beranda 2. Memperbaiki layar produk	0,5
Fitur transaksi antar rekening, antar bank, tarik tunai dan pembayaran kredit	1. Membuat layar transaksi antar rekening 2. Membuat layar transaksi antar bank 3. Membuat layar tarik tunai 4. Membuat layar bayar kredit 5. Membuat layar-layar sub-transaksi	5,5
Fitur pembukaan rekening	1. Membuat layar buka rekening 2. Membuat layar simulasi kredit	1

2. *The Sprint*

a. Perbaikan layar

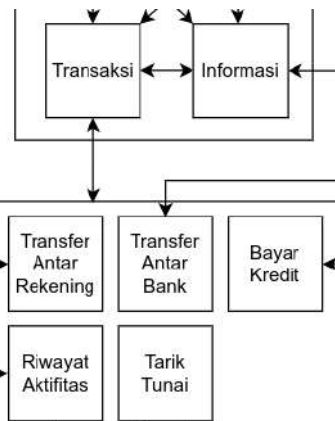
Perbaikan layar meliputi perbaikan layar beranda dan layar produk, keduanya hanya mempermasalahkan beberapa pemilihan tampilan dan bukan permasalahan pada integrasi API maupun skema validasi atau fungsionalitas. Hasil dari perbaikan layar beranda dan layar produk adalah sebagai berikut:



Gambar 4.17 Perbaikan Antarmuka Layar Beranda dan Produk

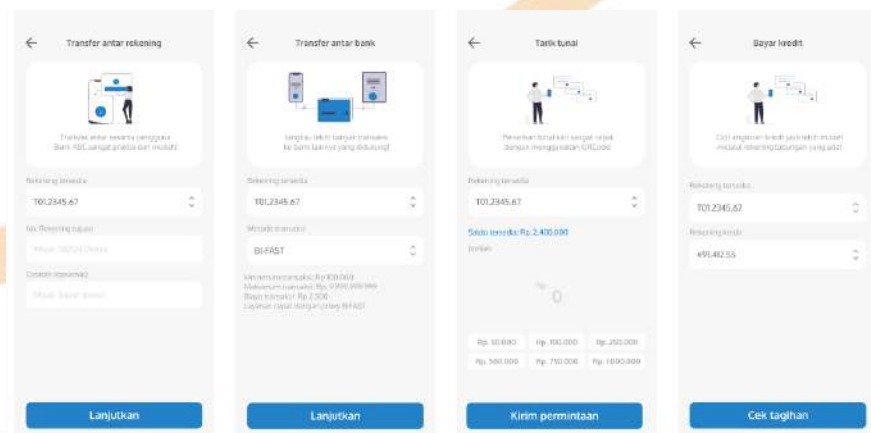
b. Fitur transaksi antar rekening, antar bank, tarik tunai dan pembayaran kredit

Terdapat 4 jenis transaksi yang terdapat pada layar transaksi, hal ini tercermin pula pada peta navigasi layar berikut:



Gambar 4.18 Peta Navigasi Jenis-Jenis Transaksi

Berikut adalah hasil dari implementasi layar transfer antar rekening, transfer antar bank, tarik tunai, bayar kredit, serta tambahan layar riwayat aktivitas untuk melihat riwayat aktivitas transaksi pada akun.



Gambar 4.19 Antarmuka Layar-Layar Jenis Transaksi

Selain dari *header* dan *dropdown* untuk memilih rekening yang tersedia, layar-layar tersebut memiliki interaksi elemen sebagai berikut:

Tabel 4.34 Interaksi Layar Transfer Antar Rekening

No	Elemen	Label	Aksi
1	<i>Input box</i>	No. Rekening tujuan	Memasukkan nomor rekening tujuan transfer
2	<i>Input box</i>	Catatan	Memasukkan catatan transaksi
3	<i>Button</i>	Lanjutkan	Ke layar nominal

Tabel 4.35 Interaksi Layar Transfer Antar Bank

No	Elemen	Label	Aksi
1	<i>Dropdown</i>	Metode transaksi	Memilih metode transaksi yang tersedia
2	<i>Button</i>	Lanjutkan	Ke layar validasi rekening

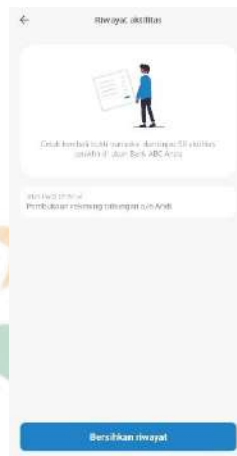
Tabel 4.36 Interaksi Layar Tarik Tunai

No	Elemen	Label	Aksi
1	<i>Card</i>	Nominal uang Rp 50.000 - Rp 1.000.000	Memilih jumlah uang yang akan ditarik tunai
2	<i>Button</i>	Kirim permintaan	Ke layar proses penarikan tunai

Tabel 4.37 Interaksi Layar Bayar Kredit

No	Elemen	Label	Aksi
1	<i>Dropdown</i>	Rekening kredit	Memilih rekening kredit tersedia untuk mengecek tagihan
2	<i>Button</i>	Cek tagihan	Ke layar cek tagihan

Selanjutnya untuk hasil dari layar riwayat aktifitas adalah sebagai berikut:



Gambar 4.20 Antarmuka Layar Riwayat Aktivitas

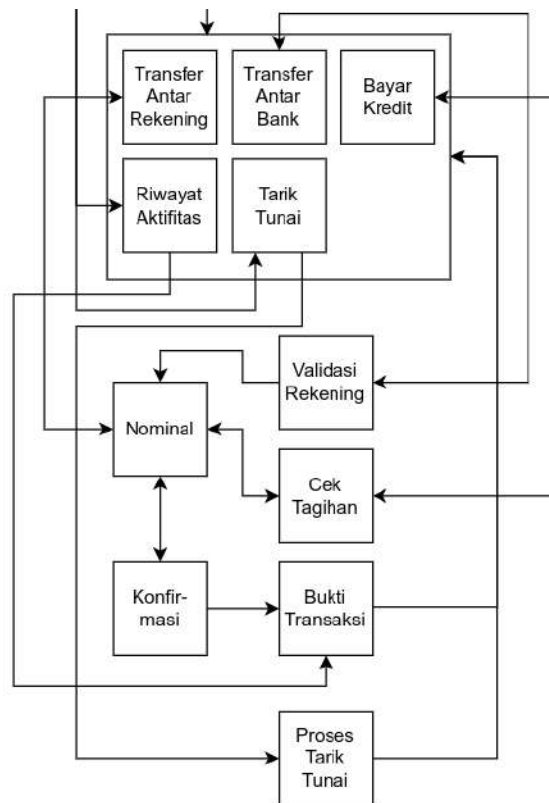
Layar riwayat aktivitas menampilkan 50 histori aksi terakhir berupa transfer dana, tarik tunai, angsuran, dan buka rekening.

Tabel 4.38 Interaksi Layar Riwayat Aktivitas

No	Elemen	Label	Aksi
1	Card	Bukti transaksi	Ke layar bukti transaksi
2	Button	Bersihkan riwayat	Membersihkan riwayat aktivitas di perangkat

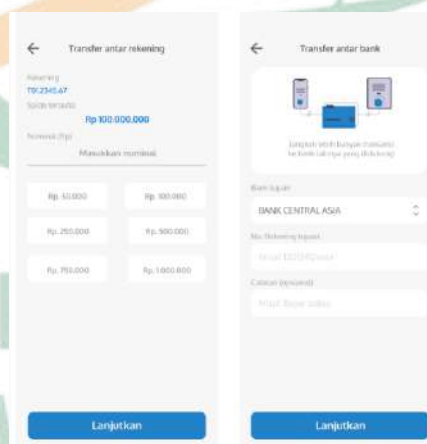
Layar-layar tersebut memiliki sub-layar sesuai dengan alur pada peta navigasi seperti:

- 1) Layar nominal.
- 2) Layar validasi rekening.
- 3) Layar cek tagihan.
- 4) Layar bukti transaksi.
- 5) Layar konfirmasi.
- 6) Layar proses tarik tunai.

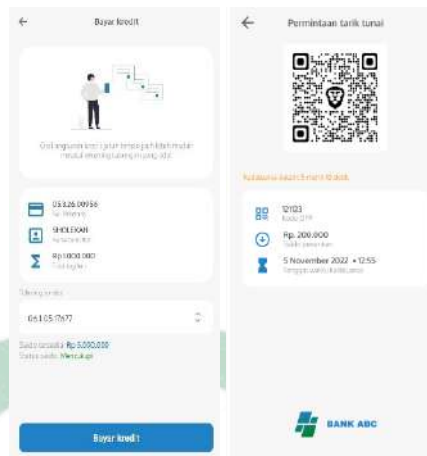


Gambar 4.21 Peta Navigasi Sub-Layar Jenis Transaksi

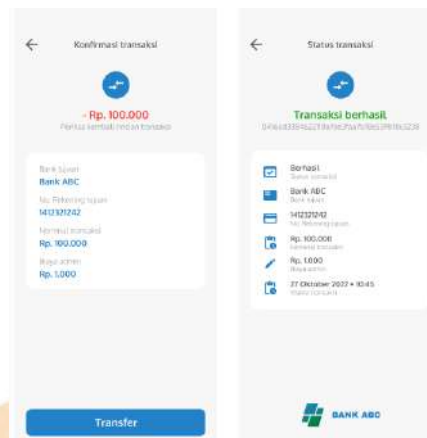
Hasil dari sub-layar jenis transaksi adalah sebagai berikut:



Gambar 4.22 Antarmuka Layar Nominal dan Validasi Rekening



Gambar 4.23 Antarmuka Layar Cek Tagihan dan Proses Tarik Tunai



Gambar 4.24 Antarmuka Layar Konfirmasi dan Bukti Transaksi

Interaksi yang dihasilkan antara layar-layar tersebut dengan pengguna selain dari interaksi dengan elemen *header* adalah sebagai berikut:

Tabel 4.39 Interaksi Layar Nominal

No	Elemen	Label	Aksi
1	<i>Card</i>	Nominal uang Rp. 50.000 - Rp 1.000.000	Memasukkan nominal dengan nilai terdefinisi
2	<i>Input box</i>	Masukkan nominal	Memasukkan nominal secara manual
3	<i>Button</i>	Lanjutkan	Ke layar konfirmasi

Tabel 4.40 Interaksi Layar Validasi Rekening

No	Elemen	Label	Aksi
1	<i>Dropdown</i>	Bank tujuan	Memilih bank tujuan transaksi yang didukung
2	<i>Input box</i>	No. Rekening tujuan	Memasukkan nomor rekening tujuan transaksi
3	<i>Input box</i>	Catatan	Memasukkan catatan transaksi
4	<i>Button</i>	Lanjutkan	Ke layar nominal

Tabel 4.41 Interaksi Layar Cek Tagihan

No	Elemen	Label	Aksi
1	<i>Dropdown</i>	Rekening sumber	Memilih rekening tabungan untuk membayar angsuran
2	<i>Button</i>	Bayar kredit	Ke layar nominal

Tabel 4.42 Interaksi Layar Konfirmasi

No	Elemen	Label	Aksi
1	<i>Button</i>	Transfer	Melakukan transaksi dan ke layar bukti transaksi

Tidak ada interaksi khusus pada layar bukti transaksi dan proses tarik tunai selain *header* untuk kembali ke layar akar dari transaksi. Integrasi API *backend* yang dipakai pada layar-layar jenis transaksi dan sub-layarnya adalah sebagai berikut:

Tabel 4.43 Integrasi API Layar-Layar Transaksi

No	Layar	API <i>Endpoint</i>	Keterangan
1	Antar rekening	<i>/myinfo</i>	Mengambil informasi daftar rekening tabungan terbaru
2	Antar bank	<i>/getjnsrjf</i>	Mengambil metode transaksi tersedia
3	Tarik tunai	<i>/reqbill</i>	Permintaan tarik tunai
		<i>/getlistrx</i>	Melihat saldo tabungan terbaru

No	Layar	API Endpoint	Keterangan
4	Bayar kredit	/myinfo	Mengambil informasi daftar rekening tabungan dan kredit terbaru
5	Nominal	/getrek	Mengambil saldo rekening yang dipilih
6	Validasi rekening	/getbank	Mengambil daftar bank yang didukung
		/cekrektrf	Mengecek nomor rekening bank lain yang dimasukkan
7	Cek tagihan	/cekangs	Mengecek daftar angsuran yang belum dibayarkan
		/getlistrx	Mengambil daftar rekening kredit yang tersedia
8	Konfirmasi	/postangkrd	Membayar angsuran kredit
		/reqbill	Mengirimkan transaksi antar rekening
		/posttrf	Mengirimkan transaksi antar bank

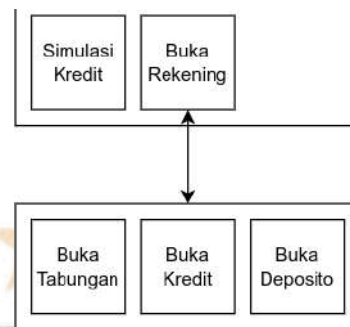
Meskipun transaksi merupakan fitur krusial dalam aplikasi perbankan, namun ternyata validasi masukan yang digunakan sangat sedikit karena nilai yang perlu dimasukkan hanya sebatas nomor transaksi dan catatan serta nominal. Skema validasi yang digunakan adalah sebagai berikut:

Tabel 4.44 Skema Validasi Layar-Layar Transaksi

No	Input (Wajib diisi*)	Aturan Lainnya
1	Nominal*	1. Bilangan bulat positif 2. Tergantung pada respon API
2	Catatan	1. Maksimal 50 karakter
3	Rekening tujuan*	1. Bilangan positif

c. Fitur pembukaan rekening

Fitur pembukaan rekening memiliki peta navigasi layar sebagai berikut:

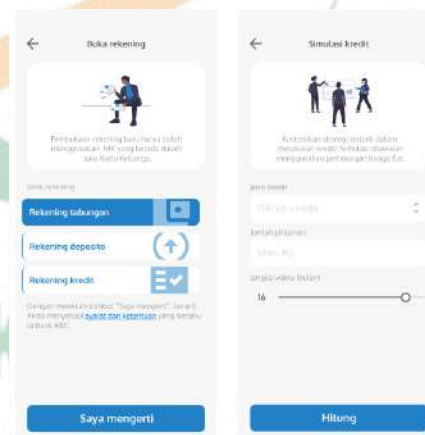


Gambar 4.25 Peta Navigasi Layar Pembukaan Rekening

Terdapat layar independen yaitu layar simulasi kredit yang berfungsi untuk melakukan perhitungan jenis produk kredit yang dipilih. Rumus yang digunakan adalah sebagai berikut:

$$\frac{\text{pokok}}{\text{jangka waktu}} + \frac{\text{pokok} * \left(\frac{\text{bunga}}{100}\right)}{\text{jangka waktu}}$$

Hasil dari implementasi layar buka rekening dan simulasi kredit adalah sebagai berikut:



Gambar 4.26 Antarmuka Layar Buka Rekening dan Simulasi Kredit

Daftar interaksi elemen yang ada pada kedua layar tersebut selain *header* adalah sebagai berikut:

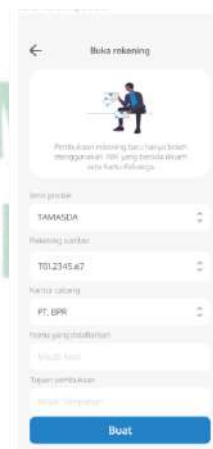
Tabel 4.45 Interaksi Layar Simulasi Kredit

No	Elemen	Label	Aksi
1	<i>Dropdown</i>	Jenis kredit	Memilih jenis produk kredit
2	<i>Input box</i>	Jumlah pinjaman	Memasukkan nominal uang yang akan dipinjam
3	<i>Slider</i>	Jangka waktu (bulan)	Menentukan jangka waktu pinjaman
4	<i>Button</i>	Hitung	Mengkalkulasi setoran yang harus dibayarkan per bulannya

Tabel 4.46 Interaksi Layar Buka Rekening

No	Elemen	Label	Aksi
1	<i>Card</i>	Rekening tabungan	Ke layar buka rekening tabungan
2	<i>Card</i>	Rekening deposito	Ke layar buka rekening deposito
3	<i>Card</i>	Rekening kredit	Dialihkan ke situs web buka rekening kredit

Keduanya tidak memerlukan skema validasi maupun integrasi API karena hanya berfungsi sebagai kalkulator dan jembatan ke sub-layar. Hasil dari sub-layar tersebut adalah sebagai berikut:



Gambar 4.27 Antarmuka Sub-Layar Buka Rekening

Daftar interaksi dari sub-layar buka rekening hanya berlaku untuk layar buka rekening tabungan dan rekening deposito, karena pembukaan rekening kredit diarahkan langsung ke web Bank ABC. Daftar interaksi tersebut adalah sebagai berikut:

Tabel 4.47 Interaksi Layar Buka Rekening

No	Elemen	Label	Aksi
1	<i>Dropdown</i>	Jenis produk	Memilih jenis produk
2	<i>Dropdown</i>	Rekening sumber	Memilih rekening sumber yang dijadikan pokok
3	<i>Dropdown</i>	Kantor cabang	Memilih kantor cabang terdekat untuk keperluan lainnya

Interaksi masukan lainnya dapat dilihat melalui skema validasi yang ada. Integrasi API *backend* dan skema validasi digunakan untuk mengirimkan data masukan pengguna adalah sebagai berikut:

Tabel 4.48 Integrasi API Sub-Layar Buka Rekening

No	Layar	API <i>Endpoint</i>	Keterangan
1	Buka rekening	<i>/getkantor</i>	Mengambil daftar kantor cabang tersedia
		<i>/getlistrx</i>	Melihat info saldo rekening
2	Buka rekening tabungan	<i>/getproduk</i>	Mengambil produk-produk tabungan (<i>app id=1</i>)
		<i>/regtab</i>	Melakukan pembukaan tabungan
3	Buka rekening deposito	<i>/getproduk</i>	Mengambil produk-produk tabungan (<i>app id=2</i>)
		<i>/regdep</i>	Melakukan pembukaan deposito

Tabel 4.49 Skema Validasi Sub-Layar Buka Rekening

No	Input (Wajib diisi*)	Aturan
1	Nama yang didaftarkan*	1. Huruf 2. Panjang 2-75 karakter
2	Tujuan pembukaan*	1. Huruf 2. Panjang 4-50 karakter
3	Sumber dana*	
4	Jumlah penempatan*	1. Minimal Rp 10.000 2. Bilangan bulat positif

3. *Sprint Review*

Sprint III merupakan *sprint* yang paling kompleks jika dibandingkan dengan *sprint* lainnya. Peninjauan *Sprint* III menghasilkan *product backlog* baru yang dieksekusi di *sprint* terakhir atau *sprint* IV.

Adapun hasil atau *increment* dari *sprint* III adalah sebagai berikut:

- a. Layar dari jenis-jenis transaksi dan pembukaan rekening
- b. Validasi dan integrasi API fitur-fitur transaksi seperti transaksi antar rekening, antar bank, penarikan tunai serta pembayaran kredit
- c. Layar riwayat aktifitas dari transaksi
- d. Validasi dan integrasi API pembukaan tabungan dan deposito

4. *Sprint Retrospective*

Peninjauan proses *sprint* dari *sprint* III mengkhawatirkan peningkatan jam rata-rata harian *developer* yang semula 10 jam pada *sprint* I dan *sprint* II menjadi 14 jam pada *sprint* III. Tim *scrum* menyepakati untuk proses *sprint* berikutnya pengurangan menjadi 8 jam per hari dalam eksekusi *sprint* harian.

E. *Sprint IV*

Sprint IV merupakan *sprint* terakhir di mana dalam *sprint* ini seluruh sisa *product backlog* dieksekusi. Dalam *sprint* ini hanya menyisakan 4 *item backlog* yaitu fitur pendaftaran aplikasi, informasi umum, keamanan, serta pengaturan. Penambahan *product backlog* dari *sprint* sebelumnya berupa increment perbaikan layar bukti transaksi dan layar tarik tunai menjadi 1 *item backlog tambahan*. Ketentuan jadwal dari *sprint IV* adalah sebagai berikut:

Tabel 4.50 Jadwal *Sprint IV*

Periode	Jumlah hari	Jam kerja/hari
22 Maret 2023 - 28 Maret 2023	7 hari	± 8 jam

1. *Sprint Planning*

Evaluasi dari *sprint* sebelumnya tidak terlalu banyak sehingga mendapatkan alokasi waktu pengerjaan yang relatif cepat. Fitur pendaftaran aplikasi memerlukan perhatian khusus karena membutuhkan banyak skema validasi masukan dari pengguna nantinya. Fitur informasi umum, fitur keamanan, dan pengaturan tidak memerlukan masukan dari pengguna sehingga tidak memerlukan integrasi API *backend*. Total *item backlog* yang dieksekusi pada *sprint IV* berjumlah 5 *item backlog*.

Tabel 4.51 *Sprint Backlog Sprint IV*

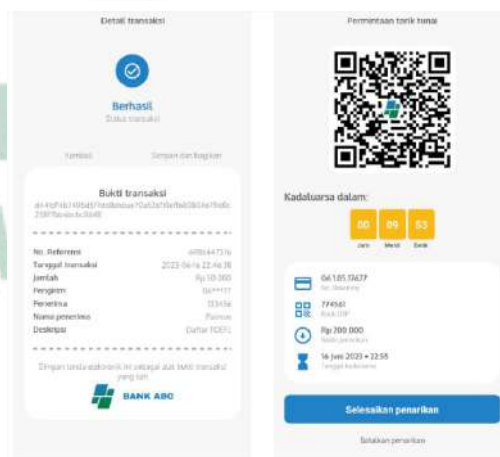
Item	Task	Est. (hari)
Perbaikan layar (<i>item backlog tambahan</i>)	1. Perbaikan layar bukti transaksi 2. Perbaikan layar tarik tunai	1
Fitur pendaftaran nasabah sebagai pengguna aplikasi	1. Membuat layar daftar jika sudah menjadi nasabah 2. Membuat layar daftar jika belum menjadi nasabah	4

<i>Item</i>	<i>Task</i>	<i>Est. (hari)</i>
Fitur melihat informasi umum	<ol style="list-style-type: none"> 1. Membuat layar informasi nasabah 2. Membuat layar informasi bank ABC 3. Membuat layar bantuan dan aduan 	1
Fitur keamanan	<ol style="list-style-type: none"> 1. Membuat fitur keamanan sesuai tabel kebutuhan 	0,5
Fitur pengaturan aplikasi	<ol style="list-style-type: none"> 1. Membuat layar pengaturan 	0,5

2. The Sprint

a. Perbaikan layar

Perbaikan layar bukti transaksi dilakukan dengan menambah tombol simpan ke dalam berkas *pdf* dan tombol bagikan untuk membagikan hasil ke sosial media seperti Whatsapp serta tampilan yang menyerupai nota. Sementara layar proses penarikan tunai diberi dua tombol untuk menyelesaikan atau membatalkan transaksi, pemberian logo QR serta pembesaran teks untuk hitung mundur. Hasil perbaikan layar adalah sebagai berikut:



Gambar 4.28 Perbaikan Layar Bukti Transaksi & Proses Tarik Tunai

Ada pun interaksi yang ada pada layar tersebut adalah sebagai berikut:

Tabel 4.52 Interaksi Layar Bukti Transaksi

No	Elemen	Label	Aksi
1	Button	Kembali	Kembali ke layar akar transaksi
2	Button	Simpan dan bagikan	Menyimpan bukti transaksi ke dalam <i>pdf</i> dan membuka opsi bagikan

Tabel 4.53 Interaksi Layar Proses Tarik Tunai

No	Elemen	Label	Aksi
1	Button	Selesaikan penarikan	Menyelesaikan penarikan dan kembali ke layar akar transaksi
2	Button	Batalkan penarikan	Membatalkan penarikan dan kembali ke layar akar transaksi

b. Fitur pendaftaran nasabah sebagai pengguna aplikasi

Pendaftaran nasabah sebagai pengguna aplikasi memiliki dua jenis yaitu apabila sudah menjadi nasabah atau belum menjadi nasabah di Bank ABC. Pengimplementasian masukan untuk pendaftaran sesuai dengan kebutuhan informasi tambahan pada

Tabel 3.2. Hasil dari fitur pendaftaran berupa layar pendaftaran sebagai berikut:

← Registrasi

Agar bisa Anda sudah mempunyai nomor.
BANK ABC selamatkanmu!

Sudah **Kembali**

Nama :

Muhammad Rizki Nur Hafidza

Nama email :

Muhammad.rizki@gmail.com

No. HP :

0812345678901234567890

Kartu Kredit bank sendi :

Kartu Kredit Bank ABC

Daftar

BANK ABC

Gambar 4.29 Layar Pendaftaran Untuk Nasabah

The image displays three sequential screenshots of the BPR Dana Wira Buana mobile application interface, showing the login and registration process.

Screenshot 1 (Left): The app is titled "BPR Dana Wira Buana". It features a logo consisting of four green squares. Below the logo, it says "Apakah Anda sudah memiliki rekening Bank ABC? (wajib diisi)". There are two buttons: "Daftar" (disabled) and "Login" (active).

Screenshot 2 (Middle): The app is titled "BPR Dana Wira Buana". It features a logo consisting of four green squares. Below the logo, it says "Apakah Anda sudah memiliki rekening Bank ABC? (wajib diisi)". There are two buttons: "Daftar" (active) and "Login" (disabled).

Screenshot 3 (Right): The app is titled "BPR Dana Wira Buana". It features a logo consisting of four green squares. Below the logo, it says "Apakah Anda sudah memiliki rekening Bank ABC? (wajib diisi)". There are two buttons: "Daftar" (active) and "Login" (disabled).

Gambar 4.30 Layar Pendaftaran Untuk Non-Nasabah

Layar pendaftaran untuk pengguna yang sudah menjadi nasabah hanya satu layar, sementara pengguna yang belum merupakan nasabah akan mengisi 3 tahap pendaftaran yaitu pengisian informasi pribadi, informasi rekening, dan informasi akun. Interaksi yang terdapat pada layar pendaftaran adalah sebagai berikut:

Tabel 4.54 Interaksi Layar Pendaftaran

No	Elemen	Label	Aksi
1	<i>Switch</i>	Sudah - Belum	Berpindah layar jenis pendaftaran
2	<i>Button</i>	Daftar	Melakukan pendaftaran dan diarahkan ke layar <i>login</i>
3	<i>Button</i>	Lanjutkan	Ke tahapan berikutnya
4	<i>Button</i>	Kembali	Ke tahapan sebelumnya
5	<i>Dropdown</i>	Produk tabungan	Memilih daftar produk tabungan
6	<i>Dropdown</i>	Kantor terdekat	Memilih kantor cabang terdekat
7	<i>Input box</i>	NIK, Nama, Tempat Lahir, Tanggal Lahir, No. Telepon, Alamat, Tujuan, Sumber dana, Alamat email, Kata sandi, Konfirmasi kata sandi	Memasukkan masukan sesuai dengan masing-masing label

Sementara untuk integrasi API dan skema validasi yang digunakan adalah sebagai berikut:

Tabel 4.55 Integrasi API Fitur Pendaftaran

No	Layar	API <i>Endpoint</i>	Keterangan
1	Daftar sebagai nasabah	<i>/get-token</i>	Mengambil <i>token</i> akses aplikasi
		<i>/register</i>	Mengirimkan data untuk pendaftaran sebagai nasabah
2	Daftar sebagai non-nasabah	<i>/get-token</i>	Mengambil <i>token</i> akses aplikasi
		<i>/register</i>	Mengirimkan data untuk pendaftaran sebagai non-nasabah
		<i>/getkantor</i>	Mengambil daftar kantor
		<i>/getproduk</i>	Mengambil daftar produk tabungan (<i>app_id=1</i>)

Tabel 4.56 Skema Validasi Fitur Pendaftaran

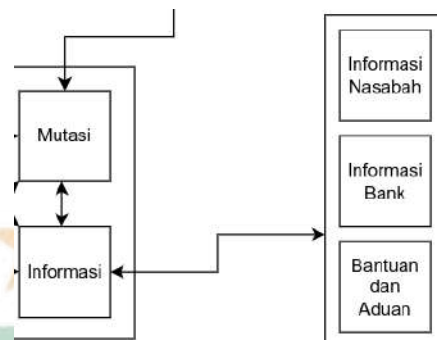
No	Input (wajib diisi*)	Aturan
1	Email*	1. Terdapat simbol '@' 2. Terdapat alamat domain
2	Kata sandi*	1. Minimal 8 karakter
3	Konfirmasi kata sandi*	2. Minimal 1 huruf kapital 3. Minimal 1 huruf non-kapital 4. Minimal 1 angka & 1 simbol
4	NIK *	1. Bilangan bulat positif 2. Minimal 16 karakter 3. Maksimal 16 karakter
5	Nama*	1. Huruf 2. Panjang 2-75 karakter
6	Tempat lahir*	1. Huruf 2. Panjang 2-30 karakter
7	Tanggal lahir*	1. Tidak melebihi tanggal sekarang
8	Nomor telepon*	1. Bilangan bulat positif 2. Panjang 9-15 karakter
9	Alamat*	1. Huruf 2. Panjang 3-175 karakter
10	Tujuan*	1. Huruf
11	Sumber dana*	2. Panjang 4-150 karakter

c. Fitur melihat informasi umum

Informasi umum terdiri dari informasi akun nasabah, informasi bank atau kantor-kantor cabang yang tersedia, serta informasi kontak untuk keperluan bantuan maupun aduan.

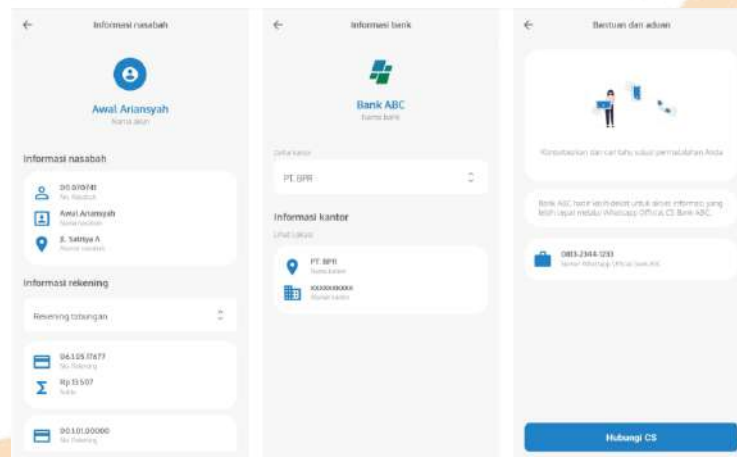
Berdasarkan peta navigasi, alur dari fitur informasi adalah sebagai berikut:

STMIK KOMPUTAMA
MAJENANG



Gambar 4.31 Peta Navigasi Fitur Informasi Umum

Hasil dari fitur untuk menampilkan informasi umum adalah ketiga layar sebagai berikut:



Gambar 4.32 Antarmuka Layar-Layar Informasi Umum

Layar informasi nasabah dan bank mengambil data dari API sementara layar bantuan dan aduan merupakan layar dengan informasi statis. Daftar interaksi yang berada pada layar-layar tersebut kecuali *header* adalah sebagai berikut:

Tabel 4.57 Interaksi Layar-Layar Informasi Umum

No	Elemen	Label	Aksi
1	<i>Dropdown</i>	Rekening tabungan	Memilih tipe rekening
2	<i>Dropdown</i>	Daftar kantor	Memilih informasi kantor
3	<i>Button</i>	Hubungi CS	Dialihkan ke Whatsapp

Tidak terdapat skema validasi namun terdapat API yang digunakan untuk mendapatkan data nasabah dan bank.

Tabel 4.58 Integrasi API Layar-Layar Informasi Umum

No	Layar	API <i>Endpoint</i>	Keterangan
1	Informasi nasabah	<i>/getlistrx</i>	Mengambil informasi daftar rekening sesuai jenis rekening
2	Informasi bank	<i>/getkantor</i>	Mengambil daftar kantor

d. Fitur keamanan

Fitur keamanan sesuai dengan data kebutuhan yang diberikan telah diimplementasikan di antaranya adalah sebagai berikut:

1) Validasi masukan

Setiap masukan pengguna yang akan dikirimkan ke server di validasi seperti yang tercantum dalam tabel skema validasi layar, termasuk validasi kata sandi yang kuat.

```
export const loginSchema = yup.object({
  email: yup.string().email(errors.email).required(errors.req('Email')),
  password: yup
    .string()
    .required(errors.req('Kata sandi'))
    .min(8, errors.password.matches)
    .minLowercase(1, errors.password.oneChar('huruf kecil'))
    .minUppercase(1, errors.password.oneChar('huruf kapital'))
    .minNumbers(1, errors.password.oneChar('angka'))
    .minSymbols(1, errors.password.oneChar('simbol spesial')),
});
```

Gambar 4.33 Validasi Kata Sandi Pada Layar *Login*

2) *Signature* dalam POST request

Signature atau tanda identifikasi unik dimasukkan ke dalam POST request menggunakan enkripsi SHA256.

```
export const transferBankSignature = req => {
  const time = getRequestDateTime();
  const str = JSON.stringify(req).replace(/\s/g, '');

  const hash_message = `${req.token}:${time}:${str}`;
  const signature = Base64.stringify(HmacSHA256(hash_message, HARD_TOKEN));

  return {time, signature};
};
```

Gambar 4.34 *Signature* SHA256 Untuk Transaksi Antar Bank

3) PIN transaksi

PIN transaksi digunakan untuk konfirmasi setiap akan melakukan transaksi.



Gambar 4.35 Pengaturan PIN Transaksi

4) Pewaktuan sesi

Sesi *login* akan kadaluarsa dalam jangka waktu 5 menit sesuai dengan masa aktif *token* akses yang diberikan

API backend.



Gambar 4.36 Sesi *Login* Berakhir

5) Identifikasi perangkat

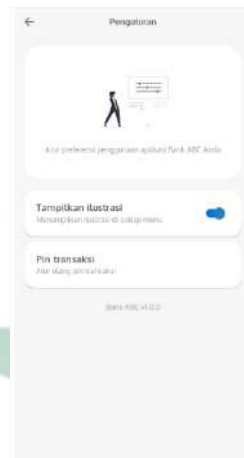
Identifikasi perangkat digunakan sehingga hanya diperbolehkan satu perangkat untuk satu akun.



Gambar 4.37 Identifikasi Perangkat

e. Fitur pengaturan aplikasi

Aplikasi berusaha mementingkan penggunaan yang sederhana agar tidak membingungkan pengguna yang awam. Terdapat dua opsi pengaturan yang tersedia yaitu, sembunyikan ilustrasi dan atur ulang PIN. Sembunyikan ilustrasi dipilih untuk mengakomodasi tampilan dengan layar kecil atau sebisa mungkin mengurangi beban dalam memuat layar. Pengaturan ulang PIN sendiri adalah fitur esensial untuk penggantian PIN. Tidak ada pengaturan tema seperti tema gelap, karena aspek keterbacaan lebih baik pada tema terang. Hasil dari layar pengaturan aplikasi adalah sebagai berikut:



Gambar 4.38 Antarmuka Layar Pengaturan

Tabel 4.59 Interaksi Layar Pengaturan

No	Elemen	Label	Aksi
1	Card	Tampilkan ilustrasi	<i>Toggle</i> tampilkan/sembunyikan ilustrasi
2	Card	PIN transaksi	Mengatur ulang PIN transaksi

3. *Sprint Review*

Peninjauan *sprint* terakhir melakukan demonstrasi final aplikasi kepada *product owner*. Hasil dari *sprint* IV diuji oleh tim dari Bank ABC menandai berakhirnya tahapan awal pengembangan dari aplikasi dan kelayakan produk minimum serta lembar perbaikan untuk tahapan pengembangan aplikasi di masa mendatang.

4. *Sprint Retrospective*

Sprint retrospective pada *sprint* IV merupakan penutup dari rangkaian *sprint* yang telah dilakukan. *Product owner*, *scrum master*, serta *developer* saling berterima kasih atas kelancaran *sprint* yang telah dilakukan hingga tercapainya produk minimum dari aplikasi.

F. Pengujian

Selain pengujian tertutup dilakukan oleh tim Bank ABC yang menangani proyek digitalisasi Bank ABC, terdapat pengujian yang dilakukan oleh peneliti secara independen yaitu evaluasi heuristik untuk mengukur aspek kegunaan dari aplikasi m-banking Bank ABC. Evaluasi heuristik dapat dilakukan dengan responden tiga orang, peneliti melakukan evaluasi heuristik terhadap tim *scrum* yang terdiri dari *developer*, *scrum master*, dan *product owner*. Evaluasi dengan responden A, B, C serta skala *severity ratings* (0-4) menghasilkan data sebagai berikut:

Tabel 4.60 Data Hasil Kuesioner

No	Pertanyaan	A	B	C
1	Cara penggunaan	1	0	0
2	Adaptasi	1	1	0
3	Penemuan fitur	1	0	0
4	Panduan deskriptif	2	1	1
5	Kecepatan akses menu	1	1	0
6	Penemuan informasi	1	0	1
7	Penggunaan waktu	1	1	0
8	Langkah menuju aksi	1	1	0
9	Penggunaan setelah jeda	1	1	0
10	Penemuan fitur yang dipakai setelah jeda	1	0	0
11	Peninjauan informasi yang dimasukkan	1	0	1
12	Lokasi menu	1	0	0
13	Frekuensi kesalahan penggunaan fitur	1	1	0
14	Perbaikan aksi yang dilakukan	1	1	0
15	Kejelasan pesan kesalahan	1	1	3
16	Frekuensi memulai ulang aplikasi karena kesalahan	1	0	1
17	Kinerja aplikasi	1	0	1
18	Penyelesaian aksi dalam fitur	1	0	0
19	Kesenangan penggunaan	1	0	0
20	Kenyamanan penggunaan	1	0	0

Hasil akhir dari pengujian evaluasi heuristik dengan dikelompokkan berdasarkan aspek dan pengambilan nilai rata-rata adalah sebagai berikut:

Tabel 4.61 Perhitungan Evaluasi Heuristik

No	Aspek	A	B	C	Total	Severity ratings
1	<i>Learnability</i>	1.25	0.5	0.25	2	0.67
2	<i>Efficiency</i>	1	0.75	0.25	2	0.67
3	<i>Memorability</i>	1	0.25	0.25	1.5	0.5
4	<i>Errors</i>	1	0.75	1	2.75	0.91
5	<i>Satisfaction</i>	1	0	0.25	1.25	0.42

Nilai responden A, B, C merupakan nilai total pertanyaan tiap aspek dibagi dengan jumlah pertanyaan yang ada pada aspek tersebut yaitu 4 pertanyaan. Nilai *severity ratings* didapat dengan membagi total nilai responden per aspek dengan jumlah responden. Hasil akhir dari perhitungan evaluasi heuristik adalah sebagai berikut:

Tabel 4.62 Nilai *Severity Ratings* Keseluruhan

No	Aspek	Severity ratings	Pembulatan
1	<i>Learnability</i>	0.67	1
2	<i>Efficiency</i>	0.67	1
3	<i>Memorability</i>	0.5	1
4	<i>Errors</i>	0.91	1
5	<i>Satisfaction</i>	0.42	0
Rata-rata		0.63	1

Hasil dari pengujian evaluasi heuristik aplikasi m-banking Bank ABC menghasilkan nilai keseluruhan *severity ratings* pada skala 1 yang berarti masih terdapat masalah yang mengganggu dan dapat dilakukan perbaikan jika terdapat waktu. Rekomendasi perbaikan dengan aspek *severity ratings* tertinggi yaitu pada aspek *errors* dapat dilakukan dengan meninjau pesan-pesan kesalahan seperti pesan kesalahan dari respon API *backend* yang masih menggunakan bahasa inggris agar dimengerti pengguna.

BAB V

PENUTUP

A. Simpulan

1. Dihasilkan tahap awal aplikasi m-banking Bank ABC untuk memenuhi digitalisasi proses bisnis pada Bank ABC meliputi fitur *login* dan pendaftaran, menampilkan informasi produk, pembukaan rekening, mutasi rekening, transaksi antar rekening, antar bank, penarikan dana, angsuran kredit, pengaturan, serta menampilkan informasi umum lainnya.
2. Dihasilkan aplikasi m-banking yang terintegrasi dengan sistem API *backend* yang telah ada sesuai dengan tabel kebutuhan diikuti dengan skema validasi serta fitur-fitur keamanan seperti validasi masukan, penggunaan *signature*, PIN transaksi, identifikasi perangkat, serta pewaktuan sesi.
3. Dihasilkan rekomendasi pengembangan dan perbaikan melalui evaluasi heuristik pada aspek *errors* dengan meninjau pesan-pesan kesalahan seperti pesan kesalahan yang sebagian masih berbahasa Inggris yang bersumber dari respon API *backend*.

B. Saran

1. Pada implementasi *scrum* dengan hanya 1 orang *developer*, *sprint* dieksekusi secara linier karena tidak ada *developer* lain yang mengeksekusi *product backlog* lain. Penelitian dengan *developer* lebih

dari 1 sangat direkomendasikan karena dapat dijalankan secara *paralel* mengeksekusi *product backlog* berbeda secara bersamaan dan menambah efisiensi waktu pengerjaan.

2. Penggunaan *user story* digunakan karena hanya untuk kebutuhan alur *frontend*, penelitian berikutnya dapat mengembangkan tidak hanya *frontend* namun juga *backend* sehingga dapat dipadukan dengan metode lain seperti *use case diagram* maupun *entity relationship diagram*.



STMIK KOMPUTAMA
MAJENANG

DAFTAR PUSTAKA

- Anastasia, F. D. (2022). Pengembangan Front-end Pada Aplikasi M-banking Agen46 Dengan Teknologi Flutter. *DScape UII*.
- Android. (n.d.). *Android - Secure & Reliable Mobile Operating System*. Retrieved from Android: <https://www.android.com/>
- Apple. (n.d.). *iOS 16 - Apple*. Retrieved from Apple Inc: <https://www.apple.com/ios/ios-16/>
- Apple. (n.d.). *Xcode 14 Overview*. Retrieved from Apple Developer: <https://developer.apple.com/xcode/>
- Ariansyah, A., & Bambang Sumantri, R. B. (2020, December 20). Evaluasi Tampilan Antarmuka Web SMK Komputama Majenang Menggunakan Metode Heuris. *Jurnal Teknologi Dan Bisnis*, 2(2), 64-75. doi:<https://doi.org/10.37087/jtb.v2i2.22>
- Christensson, P. (20019, March 31). *User Interface Definition*. Retrieved from Techterms: https://techterms.com/definition/user_interface
- Christensson, P. (2015, July 08). *IDE Definition*. Retrieved from Techterms: <https://techterms.com/definition/ide>
- Christensson, P. (2016, June 20). *API Definition*. Retrieved from Techterms: <https://techterms.com/definition/api>
- Christensson, P. (2019, May 04). *User Experience Definition*. Retrieved from Techterms: https://techterms.com/definition/user_experience
- Christensson, P. (2020, April 18). *Frontend Definition*. Retrieved from Techterms: <https://techterms.com/definition/frontend>
- Continue, R. S. (2020). Pembangunan Aplikasi Pembelajaran Mengenal Warna, Huruf Dan Angka Bagi Anak Tunagrahita Di Skh Pgri Rangkasbitung Berbasis Android. *Doctoral dissertation, Universitas Komputer Indonesia*.
- Eisenman, B. (2015). Chapter 1. What is React Native? In B. Eisenman, *Learning React Native*. O'Reilly Media, Inc.
- Git. (n.d.). *Git*. Retrieved from Git: <https://git-scm.com/>
- MDN. (2022, December 14). *Javascript | MDN*. Retrieved from Mozilla Developer Network: <https://developer.mozilla.org/en-US/docs/Web/javascript>
- Microsoft. (n.d.). *Visual Studio Code - Code Editing. Redefined*. Retrieved from Visual Studio Code: <https://code.visualstudio.com/>
- Nikolov, A. (2017, April 08). *Design principle: Consistency. The most known and most fragile design*. Retrieved from Medium: <https://uxdesign.cc/design-principle-consistency-6b0cf7e7339f>
- OJK. (2017). *Brosur Seri Literasi Mudah Dan Aman Dengan Internet Banking Dan Mobile Banking*. Retrieved from OJK: <https://sikapiuangmu.ojk.go.id/FrontEnd/CMS/NewDetailMateri/417>

- Orsini, L. (2013, November 07). *What You Need To Know About Node.js*. Retrieved from Readwrite: <https://readwrite.com/what-you-need-to-know-about-nodejs/>
- Prasetiawan, R. (2021, July 14). Membuat Website Sederhana. doi:<https://doi.org/10.31219/osf.io/mxfjw>
- Putra, Z. F., Ajie, H., & Safitri, I. A. (2021). Designing A User Interface and User Experience from Piring Makanku Application by Using Figma Application for Teens. *International Journal of Information System & Technology*, 308-315.
- Rafif, N. S. (2021). Implementasi Metode Scrum Dalam Pengembangan Perangkat Lunak Deliv. *Etheses Universitas Islam Negeri Maulana Malik Ibrahim*.
- Reynaldi, A. (2019). Perancangan Desain User Interface (UI) Aplikasi Pencari Kost. *Doctoral dissertation, Universitas Negeri Makassar*.
- Sasmito, G. W., & Fauzan, A. (2020). The Implementation of Scrum Framework for Developing Fundraising Mobile Applications. *International Journal of Advanced and Technology*, 29(1), 1654-1663.
- Schwaber, K., & Sutherland, J. (2020, November). *Scrum Guide*. Retrieved from Scrum Guides: <https://scrumguides.org/scrum-guide.html>
- Setiawan, R. (2021, August 11). *Apa itu Prototype? Kenapa itu Penting?* Retrieved from Dicoding: <https://www.dicoding.com/blog/apa-itu-prototype-kenapa-itu-penting/>
- Varga, D. (2020, August 06). *User Story vs Use Case: Everything You Need to Know*. Retrieved from Digital Natives: <https://www.digitalnatives.hu/blog/user-story-vs-use-case/>
- Wasesha, D. A., & Frieyadie. (2020, February). Model RAD Dalam Perancangan Front End Aplikasi Payment Point Online Bank (PPOB) Berbasis Android. *JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer)*, 5(2), 237-244.
- Widowati, H. (2019, August 08). *Transaksi Digital Menggeser Peran Kantor Cabang dan ATM Bank*. Retrieved from Katadata: <https://databoks.katadata.co.id/datapublish/2019/08/08/transaksi-digital-menggeser-peran-kantor-cabang-dan-atm-bank>

STMIK KOMPUTAMA
MAJENANG

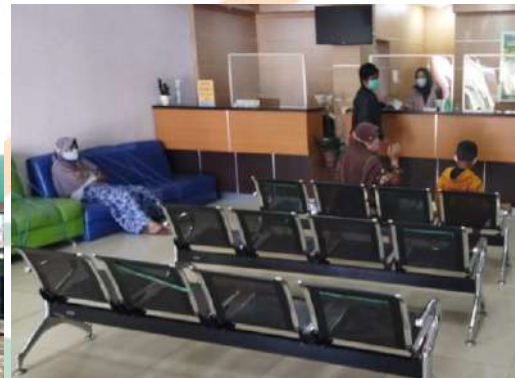
LAMPIRAN

Lampiran 1 Foto Pelaksanaan

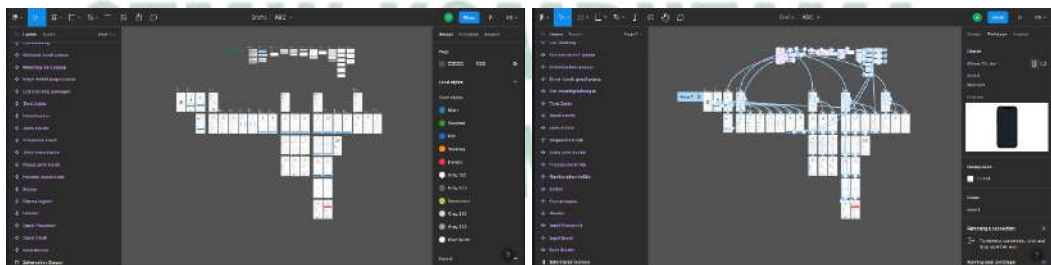
Proses Wawancara



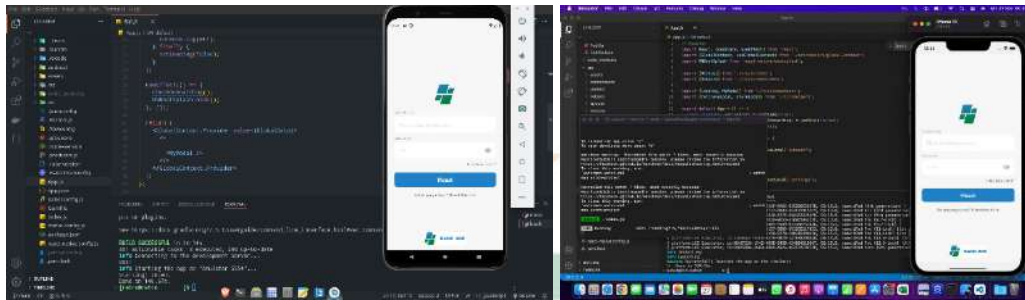
Lokasi Penelitian



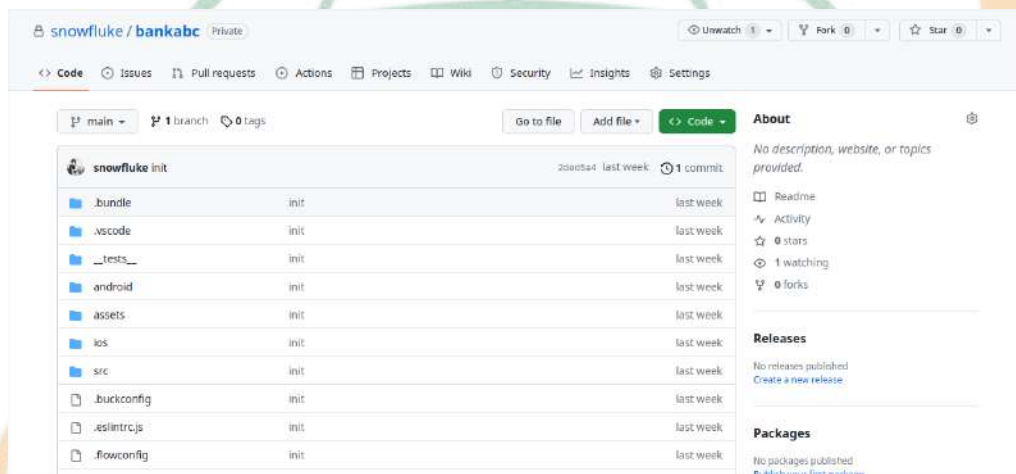
Proses Desain Antarmuka dan Prototipe di Figma



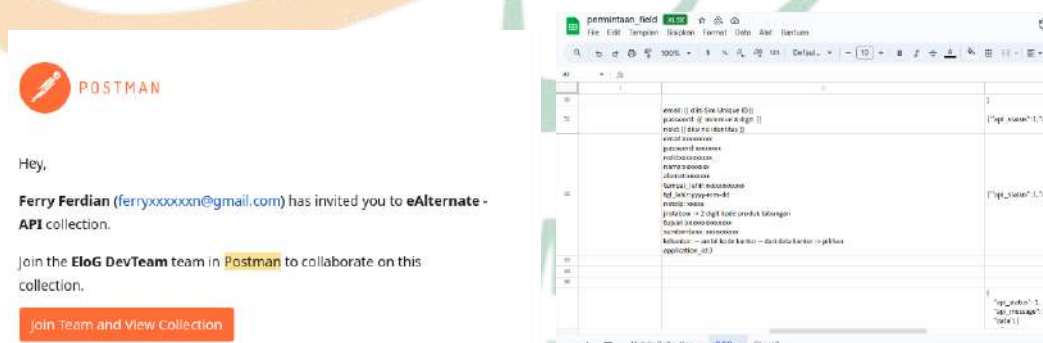
Pengujian Dengan VSCode di Platform Android dan iOS



Repositori GitHub



Undangan Data API pada Postman dan Dokumentasi API di Google Spreadsheet



Lampiran 2 Implementasi Kode

Fitur Login

```
export const Login = ({route, navigation}) => {
  const [modal, user, access_token, token, settings] =
    useContext(GlobalContext);
  const [error, setError] = useState();
  const [loading, setLoading] = useState(false);

  const [creds, setCreds] = useState({
    email: '',
    password: '',
  });

  const setEmail = email => setCreds(prev => ({...prev, email}));
  const setPassword = password => setCreds(prev => ({...prev, password}));

  const handleSubmit = async () => {
    try {
      setLoading(true);
      setError('');
      await loginSchema.validate(
        {
          email: creds.email,
          password: creds.password,
        },
        {abortEarly: false},
      );

      try {
        const app = await getAccessToken();
        if (!app.api_status) throw app.api_message;

        const res = await login({
          email: creds.email,
          password: creds.password,
          access_token: app.data.access_token,
          dummy: settings.get().mockup,
        });

        if (!res.api_status) throw res.api_message;
        user.set(res.detail);
        token.set(res.token);
        access_token.set(app.data.access_token);
        navigation.dispatch(StackActions.replace('TabNavigator'));
      } catch (error) {
        sendError(error, modal);
      }
    } catch (err) {
      const errs = formError(err);
      setError(errs);
    } finally {
      setLoading(false);
    }
  };

  const clearCredentials = () => {
    access_token.set('');
    user.set(null);
    token.set('');
  };

  const forgotPassword = () =>
    displayText(
      'Silakan datang langsung ke kantor cabang Bank ABC terdekat dan'
    );
};
```

Fitur Pendaftaran

```
export const Register = ({route, navigation}) => {
  const [settings] = useContext(GlobalContext);
  const MODE_LIST = ['Sudah', 'Belum'];
  const [mode, setMode] = useState('0');

  const [step, setStep] = useState('0');

  const [error1, setError1] = useState();
  const [error2, setError2] = useState();

  const [registerRek, setRegisterRek] = useState({
    noid: 0,
    email: '',
    password: '',
  });

  const [fullRegisterRek, setFullRegisterRek] = useState({});
  const [step1, setStep1] = useState({});
  const [step2, setStep2] = useState({});
  const [step3, setStep3] = useState({});
  const dummy = settings.get().mockup;

  const RenderScreen = () => {
    <ScrollableScreen watermark={false}>
      <Container>
        <Img src={require('../assets/logo/logo.png')} style={styl}
        <Spacing height={10} />
        <P style={style.textCenter}>
          Apakah Anda sudah menjadi nasabah Bank ABC sebelumnya?
          (dummy && ' (DEMO)')
        </P>
        <Spacing height={10} />
        <Switch list={MODE_LIST} selectedId={mode} switcher={setMode} />
        <Spacing />
        {mode === '0' ? (
          <HavingRek
            state={registerRek}
            setter={setRegisterRek}
            error={error1}
            setError={setError1}
            navigation={navigation}
          />
        ) : (
          <NotHavingRek
            state={fullRegisterRek}
            setter={setFullRegisterRek}
            step1={
              {
                state: step1,
                setter: setStep1,
              }
            }
            step2={
              {
                state: step2,
                setter: setStep2,
              }
            }
            step3={
              {
                state: step3,
                setter: setStep3,
              }
            }
          />
        )}
      </Container>
    </ScrollableScreen>
  };
};
```

Fitur Menampilkan Informasi Produk

```
export const Products = ({navigation, route}) => {
  const [token, access_token, redirect, settings] = useContext(GlobalContext);
  const [loading, setLoading] = useState();
  const [query, setQuery] = useState('');

  const [refreshing, setRefreshing] = useState();
  const [data, setData] = useState([]);

  const (key) = route.params;
  const (img, description, app_id, icon) = SCREEN_LIST[key];
  const dummy = settings.get().mockup;

  const fetchData = async () => {
    try {
      setLoading(true);
      const res = await fetchProduct({
        token: token.get(),
        access_token: access_token.get(),
        app_id,
        dummy,
      });

      if (!res.api_status) throw res.api_message;

      setData(res.detail);
    } catch (error) {
      fetchError(error, 'data produk', () => redirect.set(true));
    } finally {
      setLoading(false);
    }
  };

  const renderProduct = data => {
    data
    .filter(el => el.keterangan.toLowerCase().includes(query.toLowerCase()))
    .sort((a, b) => (a.keterangan > b.keterangan ? 1 : -1))
    .map((item, id) => (
      <Wrapper key={id} style={style.mb}>
        <DetailsCard
          volume={item.keterangan}
          label={`Bunga: ${item.bunga}%`}
          // icon={icon}
        />
      </Wrapper>
    ));

    useEffect(() => {
      fetchData();
    }, []);

    const onRefresh = useCallback(() => {
      fetchData().then(() => setRefreshing(false));
    }, []);

    const renderScreen = () => {
      <
        {data.length ? (
          <ProductLayout img={img} description={description}>
            <ScrollView
              style={style.scrollView}
              stickyHeaderIndices={[0]}
              refreshControl={
                <RefreshControl onRefresh={onRefresh} refreshing={refreshing} />
              }
            </ScrollView>
          </ProductLayout>
        ) : (
          <EmptyState />
        )}
      </
    };
  };
};
```


Fitur Transaksi Antar Bank

```
export const TransferBank = ({navigation, route}) => {
  const img = require('.../assets/illustrations/undraw_online_transactions');
  const {redirect, user, token, access_token, settings} =
    useContext(GlobalContext);
  const USER_DATA = user.get();

  const [loading, setLoading] = useState(false);

  const [tabungans, setTabungans] = useState({
    keys: ['Gagal dimuat'],
  });
  const [tabunganId, setTabunganId] = useState('0');

  const [methods, setMethods] = useState({
    data: [],
    keys: ['Gagal dimuat'],
  });
  const [methodId, setMethodId] = useState('0');
  const dummy = settings.get().mockup;

  const fetchData = async () => {
    try {
      setLoading(true);

      const tabunganKeys = extractListReks(USER_DATA['tabungan']);
      setTabungans({keys: tabunganKeys});

      const allMethods = await fetchMethodList({
        token: token.get(),
        access_token: access_token.get(),
      });
    } catch (error) {
      // ...
    }
  };

  const RenderScreen = () => (
    <BannerButtonLayout>
      <img={img}/>
      <description={
        'Jangkau lebih banyak transaksi ke bank lainnya yang didukung!'
      }/>
      <button={
        {
          title: 'Lanjutkan',
          onPress: handleSubmit,
        }
      }/>
      <InputDropdown>
        <list={banks.keys}/>
        <selectedId={bankId}/>
        <onChange={setBankId}/>
        <label="Bank tujuan"/>
      </InputDropdown>
      <InputText>
        <label={('No. Rekening tujuan')}/>
        <placeholder="Misal: 123.123.xxxxx"/>
        <maxLength={30}/>
        <val={inputs.acc_no_to}/>
        <keyboardType="numeric"/>
        <state={setAccNoTo}/>
        <error={error7.acc_no_to7.message}/>
      </InputText>
      <InputText>
        <label={('Catatan (optional')}/>
        <placeholder="Misal: 123.123.xxxxx"/>
        <maxLength={30}/>
        <val={inputs.description}/>
        <state={setDescrption}/>
      </InputText>
    </BannerButtonLayout>
  );
};
```

Fitur Penarikan Dana

```
export const Withdraw = ({navigation, route}) => {
  const img = require('.../assets/illustrations/undraw_mobile_payments_re_7');
  const {redirect, token, access_token, model, user, settings} =
    useContext(GlobalContext);
  const id = user.get()?.nasabah?.no_nasabah;

  const [loading, setLoading] = useState(true);
  const [error, setError] = useState(false);
  const dummy = settings.get().mockup;

  const [tabungans, setTabungans] = useState({
    data: [],
    keys: ['Gagal dimuat'],
  });
  const [tabunganId, setTabunganId] = useState('0');
  const [inputs, setInputs] = useState({
    description: '',
    trx_amount: {
      display: '',
      value: 0,
    },
  });

  const fetchData = async () => {
    try {
      setLoading(true);
      setError(false);

      const allTabungan = await fetchInfoRekening({
        token: token.get(),
        access_token: access_token.get(),
      });
    } catch (error) {
      // ...
    }
  };

  const RenderScreen = () => (
    <ScrollableScreen>
      <Container>
        <View style={style.qrContainer}>
          <View style={style.qrInner}>
            <QRCode>
              <size={200}/>
              <value={data.signature}/>
              <logo={require('.../assets/logo/mini-logo.png')}/>
            </QRCode>
          </View>
        </View>
        <Spacing height={30}/>
        <SectionTitle title={('Kadaluarsa dalam:')} />
        <CountDown>
          <until={wd.seconds_left}/>
          <digitTxtStyle={color: Color.theme.background}/>
          <onFinish={() => withdrawDone('Waktu penarikan tunai telah habis')}/>
          <timeToShow={['H', 'M', 'S']}/>
          <timeLabels={({h: 'Jam', m: 'Menit', s: 'Detik'})}/>
          <size={20}/>
        </CountDown>
        <Spacing />
        <ScrollView style={style.scrollViewDetails}>
          <Wrapper>
            <DetailsCard>
              <label={('No. Rekening')}/>
              <icon={Icons.creditCard}/>
              <value={wd.acc_no_from}/>
            </DetailsCard>
          </Wrapper>
        </ScrollView>
      </Container>
    </ScrollableScreen>
  );
};
```

Fitur Angsuran Kredit

```
export const PayCredit = ({navigation, route}) => {
  const img = require('.../assets/illustrations/undraw_mobile_payments_re_7');
  const {redirect, token, access_token, user, settings} =
    useContext(GlobalContext);
  const USER_DATA = user.get();

  const [loading, setLoading] = useState(false);
  const [refreshing, setRefreshing] = useState(false);

  const [kredits, setKredits] = useState({
    keys: ['Gagal dimuat'],
  });
  const [kreditId, setKreditId] = useState('0');
  const dummy = settings.get().mockup;

  const fetchData = async () => {
    try {
      setLoading(true);
      const info = await fetchInfo({
        token: token.get(),
        access_token: access_token.get(),
        dummy,
      });
    } catch (error) {
      // ...
    }
  };

  const RenderScreen = () => (
    <kredits.keys.length ? (
      <BannerButtonLayout>
        <img={img}/>
        <description={
          'Cicil angsuran kredit jatuh tempo jauh lebih mudah melalui reken'
        }/>
        <button={
          {
            title: 'Cek tagihan',
            onPress: handleSubmit,
          }
        }/>
        <refreshControl={
          <RefreshControl onRefresh={onRefresh} refreshing={refreshing} />
        }/>
        <InputDropdown>
          <list={kredits.keys}/>
          <selectedId={kreditId}/>
          <onChange={setKreditId}/>
          <label="Rekening kredit"/>
        </InputDropdown>
      </BannerButtonLayout>
    ) : (
      <Container>
        <NotFound text="Belum terdapat rekening kredit saat ini" />
      </Container>
    )
  );

  return <loading ? <Loading /> : RenderScreen() />;
};
```

Fitur Menampilkan Informasi Umum

```
export const Information = ({route, navigation}) => {
  const renderInformationMenu = () => {
    menu.map((item, id) => {
      <Card
        key={id}
        title={item.title}
        description={item.description}
        leftIcon={item.leftIcon}
        onPress={() => item.onPress(navigation)}
      />
    )
  }

  return <ScrollableScreen>{renderInformationMenu()}</ScrollableScreen>;
};

const menu = [
  {
    title: 'Informasi nasabah',
    description: 'Lihat informasi Bank ABC Anda',
    leftIcon: Icons.profile,
    onPress: navigation => navigation.navigate('Informasi nasabah'),
  },
  {
    title: 'Informasi bank',
    description: 'Lihat jaringan kantor Bank ABC',
    leftIcon: Icons.bank,
    onPress: navigation => navigation.navigate('Informasi bank'),
  },
];

<View>
  <InputDropdown
    lists={banks.keys}
    selectedId={bankId}
    onChange={setBankId}
    label="Daftar kantor"
  />
</View>
<Spacing />
<SectionTitle title='Informasi kantor' />

{banks.data ? (
  <TouchableOpacity
    onPress={() => Linking.openURL(banks.data[bankId].link)}
    <P>Lihat Lokasi</P>
  </TouchableOpacity>
  <Spacing height={10} />
  <ScrollView
    style={style.scrollViewDetails}
    refreshControl={
      <RefreshControl onRefresh={onRefresh} refreshing={refreshing} />
    }
    <DetailsContent data={banks.data[bankId].data} />
  </ScrollView>
) : (
  <NotFound text="Informasi kantor belum tersedia" />
)}
</DetailsLayout>
```

Fitur Pengaturan

```
export const Settings = ({navigation, route}) => {
  const img = require('../assets/illustrations/undraw_Adjustments_re_gvct');
  const (settings, modal, user) = useContext(GlobalContext);
  const id = user.get().nasabah?.no_nasabah;

  const onToggle = async (key, value) => {
    let oldSettings = settings.get();
    oldSettings[key] = value;

    settings.set({...oldSettings});
    await storeLocal('settings', oldSettings);
  };

  const checkPin = async () => {
    try {
      if (!id) return;
      const pin = await retrieveEncrypted('pin@${id}');
      if (!pin) return modal.openModal(() => <NoPIN />);

      modal.openModal(() => <ResetPIN oldPin={pin} />);
    } catch (error) {
      displayToast('Terjadi kesalahan dalam memuat pengaturan');
    }
  };

  return (
    <ProductLayout
      img={img}
      description='Atur preferensi penggunaan aplikasi Bank ABC Anda'
    >
      <ScrollView style={style.scrollView}>
        <SwitchSetting
          type='illustration'
          value={settings.get().illustration}
          title='Tampilkan ilustrasi'
          onToggle={onToggle}
          description='Menampilkan ilustrasi di setiap menu'
        />
        <SettingCard
          title='Pin transaksi'
          description='Atur ulang pin transaksi'
          onPress={checkPin}
        />
        <SwitchSetting
          type='mockup'
          value={settings.get().mockup}
          title='Mockup API'
          onToggle={onToggle}
          description='Gunakan data untuk demonstrasi'
        />
        <Spacing />
        <P style={style.textCenter}>
          {APP_NAME} {APP_VERSION}
        </P>
      </ScrollView>
    </ProductLayout>
  );
};
```

STMIK KOMPUTAMA
MAJENANG

Lampiran 3 Kuesioner Evaluasi Heuristik

Kuesioner pengujian aplikasi m-banking

Bank ABC

Pengisian cara saya Awal Alamsyah seorang mahasiswa STMIK Komputera
Meningkatkan menggunakan kuesioner metode heuristik yang akan digunakan sebagai
bahan evaluasi dan penelitian dengan tujuan implementasi Sistem Dalam Peningkatan
Fremend Aplikasi M-Banking Di Bank ABC Jawa Barat

Estimasi pengerjaan: 35 menit

awalsyah07@gmail.com (Tidak ditagih) [Cari lain](#)

* [Angka](#)

Nama lengkap *

Jawaban teks

Jenis kelamin *

☒ Laki-laki
 ☐ Perempuan
 ☐ Memilih tidak menjawab

Berikutnya

Kesimpulan formulir

Learnability (Kemampuan belajar)

Cara penggunaan *

0 1 2 3 4

0 (sangat buruk) ☐ ☐ ☐ ☐ ☐ 4 (sangat baik)

Adaptasi *

0 1 2 3 4

0 (sangat adaptif) ☐ ☐ ☐ ☐ ☐ 4 (sangat sulit)

Penemuan fitur *

0 1 2 3 4

0 (sangat mudah) ☐ ☐ ☐ ☐ ☐ 4 (sangat sulit)

Panduan deskriptif *

0 1 2 3 4

0 (sangat membantu) ☐ ☐ ☐ ☐ ☐ 4 (sangat kurang membantu)

Kembali

Berikutnya

Kesimpulan formulir

Efficiency (Efisiensi)

Kecepatan akses menu *

0 1 2 3 4

0 (sangat cepat) ☐ ☐ ☐ ☐ ☐ 4 (sangat lambat)

Penemuan informasi *

0 1 2 3 4

0 (sangat mudah) ☐ ☐ ☐ ☐ ☐ 4 (sangat sulit)

Penggunaan waktu *

0 1 2 3 4

0 (sangat efisien) ☐ ☐ ☐ ☐ ☐ 4 (sangat tidak efisien)

Langkah menuju aksi *

0 1 2 3 4

0 (sangat sedikit) ☐ ☐ ☐ ☐ ☐ 4 (sangat banyak)

Kembali

Berikutnya

Kesimpulan formulir

Memorability (Bersifat ingatan)

Penggunaan aplikasi setelah jeda *

0 1 2 3 4

0 (sangat mudah ingati) ☐ ☐ ☐ ☐ ☐ 4 (sangat mudah lupa)

Penemuan fitur yang sering digunakan setelah jeda *

0 1 2 3 4

0 (sangat mudah ditemukan) ☐ ☐ ☐ ☐ ☐ 4 (sangat sulit ditemukan)

Penemuan informasi yang dimasukkan *

0 1 2 3 4

0 (sangat informatif) ☐ ☐ ☐ ☐ ☐ 4 (sangat kurang informatif)

Lokasi menu *

0 1 2 3 4

0 (sangat mudah diingat) ☐ ☐ ☐ ☐ ☐ 4 (sangat sulit diingat)

Kembali

Berikutnya

Kesimpulan formulir

Errors (Kesalahan)

Frekuensi kesalahan penggunaan *

0 1 2 3 4

0 (sangat jarang) ☐ ☐ ☐ ☐ ☐ 4 (sangat sering)

Perbaikan aksi yang telah dilakukan *

0 1 2 3 4

0 (sangat mudah) ☐ ☐ ☐ ☐ ☐ 4 (sangat sulit)

Kategori pesan kesalahan *

0 1 2 3 4

0 (sangat jelas) ☐ ☐ ☐ ☐ ☐ 4 (sangat kurang jelas)

Frekuensi memulai ulang aplikasi karena kesalahan *

0 1 2 3 4

0 (sangat jarang) ☐ ☐ ☐ ☐ ☐ 4 (sangat sering)

Kembali

Berikutnya

Kesimpulan formulir

Satisfaction (Kepuasan)

Kinerja aplikasi *

0 1 2 3 4

0 (sangat puas) ☐ ☐ ☐ ☐ ☐ 4 (sangat kurang puas)

Penyelesaian aksi *

0 1 2 3 4

0 (sangat puas) ☐ ☐ ☐ ☐ ☐ 4 (sangat kurang puas)

Kesenangan penggunaan *

0 1 2 3 4

0 (sangat menyenangkan) ☐ ☐ ☐ ☐ ☐ 4 (sangat menyebalkan)

Kenyamanan penggunaan *

0 1 2 3 4

0 (sangat nyaman) ☐ ☐ ☐ ☐ ☐ 4 (sangat kurang nyaman)

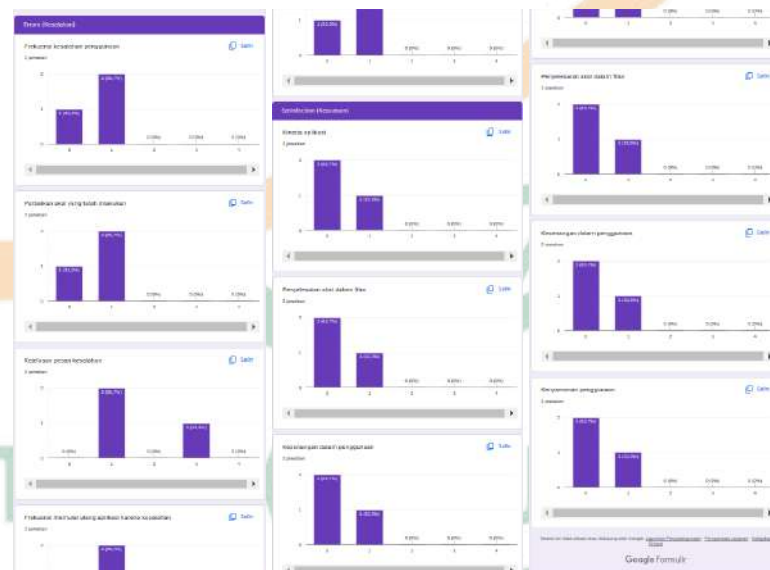
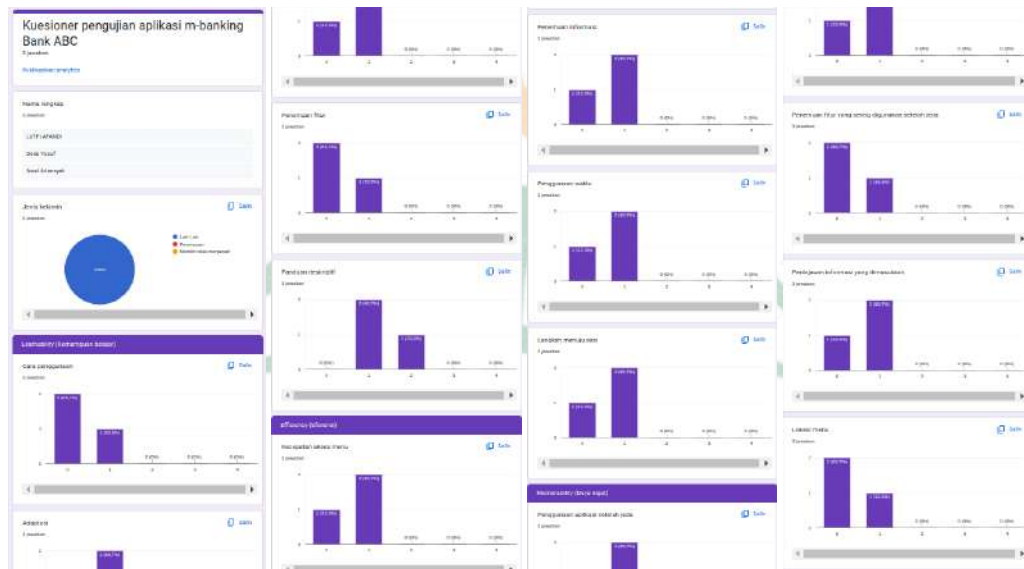
Kembali

Kirim

Kesimpulan formulir

MAJENANG

Lampiran 4 Hasil Kuesioner Evaluasi Heuristik



MAJENANG

Lampiran 5 Data Respon API

No	Endpoint	Data Respon (JSON)
1	/get-token	{ "api_status": 1, "application_id": 3, "app_code": "ABC", "api_message": "success", "data": { "access_token": "AD237SD2", "expiry": 1672016800 } }
2	/login	{ "api_status": 1, "id_aplikasi": 3, "api_message": "success", "token": "5o0nhcywd", "detail": { "nasabah": { "nama": "Awal Ariansyah", "no_nasabah": "12345 " }, "tabungan": { "jml": 1, "saldo": "1307918" }, "deposito": { "jml": 1, "saldo": "10000000" }, "kredit": { "jml": 4, "saldo": "15221673" } } }
3	/register	{ "api_status": 1, "api_message": "success", "data": { "id": 12345 } }
4	/getpromo	{ "api_status": 1, "api_message": "success", "data": [{ "id": 2, "judul_promo": "Promo Ramadan", "deskripsi": "Pembukaan rekening tabungan mahasiswa bonus cashback", "link": "https://scholar. google.com/", "image_promo": "https://images.google.com", "expired_at": "2023-02-16 00:00:00" }, ...] }
5	/getproduk	{ "api_status": 1, "api_message": "success", "detail": [{ "kode": "98", "keterangan": "Tabungan Berjangka", "bunga": "0" }, ...], "disclaimer": "Info" }
6	/getkantor	{ "api_status": 1, "api_message": "Success", "detail": [{ "kode": "01", "nama_kantor": "ABC", "alamat": "xxx", "long": "10.34", "lat": "-6.134" }, ...] }
7	/cektrx	{ "api_status": 1, "api_message": "Success", "detail": [{ "tgl_transaksi": "2022-10-02", "no_kwitansi": 123, "keterangan": "Pembukaan Rek Baru a.n Ani", "d_k": "D", "jumlah_transaksi": "100", "saldo": "300" }, ...] }

No	Endpoint	Data Respon (JSON)
8	/myinfo	{ "api_status": 1, "id_aplikasi": 3, "api_message": "success", "detail": { "nasabah": { "nama": "Awal Ariansyah", "alamat": "Jl. Sudirman", "no_nasabah": "12345 " }, "tabungan": { "no_nasabah": "11.234 ", "jml": 6, "saldo": "1990", "no_rekening": "11.3444" }, "deposito": { "no_nasabah": "11.234 ", "jml": 1, "pokok": "10000", "no_rekening": "22.11 " }, "kredit": null } }
9	/getrek	{ "api_status": 1, "api_message": "success", "detail": [{ "nama": "Awal Ariansyah", "alamat_nasabah": "Jl. Sudirman", "no_nasabah": "12345 ", "no_rekening": "11.2345", "saldo_efektif": "9980000" }] }
10	/getjnsrf	{ "api_status": 1, "api_message": "success", "data": [{ "id": 4, "keterangan": "BI-FAST", "min_trx": 10000, "max_trx": 200000, "deskripsi": "Tersedia Transfer ke bank BIFAST real-time (24/7)", "biaya": 2500 }, ...] }
11	/getbank	{ "api_status": 1, "api_message": "success", "detail": [{ "id": 1, "bank_code": "400309", "bank_name": "BANGKOK BANK" }, ...] }
12	/getlistrx	{ "api_status": 1, "id_aplikasi": 3, "api_message": "success", "detail": [{ "keterangan": "TAMASDA", "no_nasabah": "12345", "no_rekening": "11.2345 ", "nama": "Awal Ariansyah", "alamat_nasabah": "Jl. Sudirman", "saldo_efektif": "9753500", "saldo_min": "0", "saldo_blokir": "0", "saldo_blok_krd": "0"}, ...] }
13	/regtab	{ "api_status": 1, "application_id": 3, "api_message": "success", "detail": { "status": "1", "no_rekening": "11.3456 ", "nama": "Aria", "saldo_efektif": "20000", "messages": "Pendaftaran rekening Sukses dan rekening telah aktif" } }
14	/regdep	{ "api_status": 1, "application_id": 3, "api_message": "success", "detail": { "status": "1", "no_rekening":

No	Endpoint	Data Respon (JSON)
		"22.2341 ", "nama": "Aria", "pokok": null, "tgl_awal": "2022-10-01", "tgl_akhir": "2023-01-01", "jkw": "3", "messages": "Pendaftaran rekening Sukses dan rekening telah aktif" } }
15	/reqbill	{ "api_status": 1, "id_aplikasi": 3, "api_message": "success", "detail": { "acc_no_from": "11.234 ", "trx_amount": 300000, "otp": "647810", "signature": "c0e6ffea095e7e3ea6e3af0...", "expired_date": "2022-12-24 08:27:40" } } }
16	/cekrektrf	{ "api_status": 1, "id_aplikasi": 3, "api_message": "success", "detail": { "AcctInqRs": { "MsgRsHdr": { "ResponseTimestamp": "2022-11-09T11:12:04.850+07:00", "CustRefID": "4eea8f0f81dcc15c055...", "StatusCode": "00", "StatusDesc": "Success" }, "InqInfo": { "AccountNumber": "701075331", "AccountName": "Ani" } } } }
17	/posttrf	{ "api_status": 1, "id_aplikasi": 3, "api_message": "success", "detail": { "XferAddRs": { "MsgRsHdr": { "ResponseTimestamp": "2022-11-16T13:28:12.437+07:00", "CustRefID": "783d81643dc83d8e5d5d", "StatusCode": "00", "StatusDesc": "Success" }, "TrxReffNo": "0867221116" } } }
18	/cekangs	{ "api_status": 1, "id_aplikasi": 3, "api_message": "success", "detail": { "no_rekening": "33.14", "nama": "Umi", "tot_tagihan": "100", "kolektibilitas": 4, "jmlangs": 8 } }
19	/postangkrd	{ "api_status": 1, "application_id": 3, "api_message": "success", "detail": { "id": 240, "tgl_transaksi": "2022-11-17 18:10:02", "acc_no_to": "33.1234", "acc_name_to": "Umi", "trx_amount": 10000, "kwit_internal": 5333 } }

Lampiran 6 Kebutuhan Desain

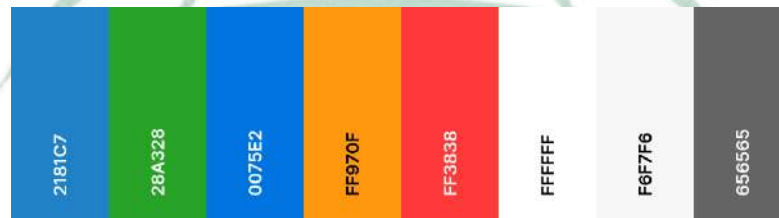
Jenis Huruf

Oxygen
Designed by Vernon Adams

Menimbang bahwa mengabaikan dan memandang rendah hak-hak asasi

Select preview text: Asia Indonesian

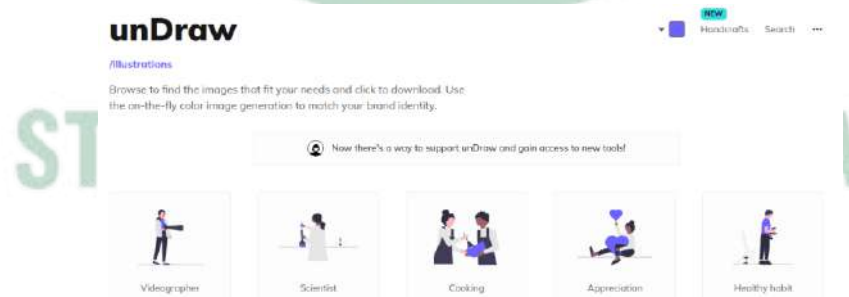
Palet Warna



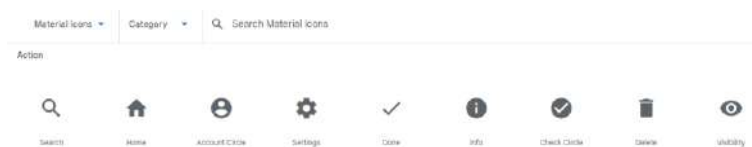
Logo



Ilustrasi



Ikon



Lampiran 7 Lembar Wawancara

**LEMBAR WAWANCARA
DENGAN PROJECT MANAGER BANK ABC**

No.	Pertanyaan dan jawaban	
1	Q	Seperti apakah gambaran singkat dari BPR Bank ABC di wilayah Jawa Barat ini?
	A	Bank ABC adalah sebuah BPR (Bank Perkreditan Rakyat) di wilayah Jawa Barat di mana model bisnis utamanya adalah layanan keuangan seperti layaknya Bank pada umumnya.
2	Q	Apa alasan Bank ABC memutuskan untuk membangun aplikasi m-banking?
	A	Gempuran teknologi di bisnis <i>microfinance</i> , menjamurnya bisnis <i>fintech</i> yang memberikan kemudahan masyarakat serta dorongan OJK dalam digitalisasi. Bank ABC memanfaatkan kesempatan tersebut dengan keinginan untuk membangun aplikasi yang mempermudah nasabahnya mulai dari membuka rekening hingga transaksi keuangan.
3	Q	Bagaimana regulasi mengenai penggunaan teknologi di bank BPR?
	A	Hingga 2021, masih banyak bank BPR yang belum tersentuh teknologi (masih konvensional) bukan karena <i>ketidak-sanggupan</i> tetapi juga regulasi penggunaan teknologi masih hanya ada di level-level BPR dengan level modal tertentu. Dengan munculnya disrupsi teknologi dan permintaan digitalisasi, mempermudah izin sehingga BPR-BPR mulai membangun teknologi informasi di lingkungan BPR-nya. Mereka menambah pola bisnis konvensional juga menggaet juga pasar digital.
4	Q	Berapa jumlah nasabah bank ABC saat ini?
	A	Kalau dilihat dari statistik di BPR yang sekarang, mengambil data nasabah secara umum (bukan pemisahan nasabah kredit, nasabah tabungan). Ada sekitar 160 ribu nasabah dengan 80% rekening memiliki fasilitas pinjaman - yang otomatis akan memiliki rekening tabungan.
5	Q	Apa target pasar Bank ABC baik sebelum maupun sekarang dengan terjadinya disrupsi teknologi?
	A	Target BPR dulunya adalah pasar-pasar ekonomi mikro di pasar, usaha-usaha kecil, pedagang-pedagang kecil, UMKM, daerah-daerah yang tidak terjangkau bank umum. Sekarang pun pasarnya masih sama, dengan disrupsi teknologi, masuknya <i>pinjol</i> , masuknya <i>fintech</i> , mau tidak mau bisnis mereka secara langsung maupun tidak

No.	Pertanyaan dan jawaban	
		langsung membandingkan kemudahan pemberian fasilitas finansial dari <i>fintech</i> dan bank.
6	Q	Sistem seperti apakah yang sudah berjalan di Bank ABC saat ini?
	A	Secara pencatatan, mereka sudah memakai teknologi. Ada beberapa aplikasi-aplikasi yang berjalan untuk melakukan pencatatan penagihan secara <i>online</i> . Namun belum memiliki aplikasi yang sifatnya <i>mobile</i> untuk menangani nasabah yang ingin melakukan pengecekan informasi, pembukaan rekening maupun transaksi. Dikatakan konvensional, iya namun tidak secara umum karena beberapa bagian sudah ada sistem informasinya.
7	Q	Apa layanan utama yang diharapkan dapat diberikan oleh aplikasi m-banking Bank ABC?
	A	Layanan di tahap awal, layanan aplikasi <i>mobile</i> yang memudahkan mereka mendapatkan informasi. Nasabah bisa melihat saldo, bisa melihat transaksi di bank tersebut, bisa melakukan pemindahan dana baik ke rekening lain maupun ke simpanan berjangka (<i>deposito</i>). Tambahan ini juga diharapkan bisa punya fitur untuk melakukan pembukaan rekening, baik untuk nasabah yang transisi maupun nasabah baru.
8	Q	Menurut Peraturan Bank Indonesia No. 19/6/PBI/2016 tentang Penyelenggaraan Usaha Bank Perkreditan Rakyat yang tidak membolehkan BPR bertransaksi dengan bank lain. Apakah transaksi antar bank tidak disediakan untuk layanan m-banking Bank ABC?
	A	Betul transfer antar bank ini sebenarnya BPR tidak boleh melakukannya. Sehingga BPR menggaet bank lain sebagai <i>anchor</i> bank, bank yang menaungi, bank yang menjembatani transaksi ke bank lain. Itu yang menjembatani BPR ingin punya aplikasi, <i>mobile apps</i> untuk nasabahnya. Itu fitur-fitur utama, fitur awal yang diharapkan ada di sistem. Lalu setelah ini berjalan cukup baik, lalu kemudian secara regulasi terpenuhi secara perijinan sudah terpenuhi maka akan dikembangkan fitur-fitur pembayaran lain secara umum seperti <i>payment point</i> setelah regulasi berjalan.
9	Q	Sebagai bentuk persona, apakah tim internal Bank ABC sudah terbiasa menggunakan layanan m-banking pada aplikasi lain?
	A	Secara internal BPR, mereka menggunakan layanan bank umum, sudah umum dengan layanan aplikasi <i>mobile banking</i> .
10	Q	Antarmuka seperti apakah yang ingin difokuskan pada aplikasi m-banking Bank ABC ini? Seperti warna, bahasa dan sebagainya?
	A	Hal yang perlu diperhatikan dalam membangun aplikasi <i>mobile</i> adalah <i>user friendly</i> , pasar kita adalah pasar-pasar di mana literasi keuangan, literasi digitalnya masih

No.	Pertanyaan dan jawaban	
		rendah. Sehingga harapannya, dari fitur dari kemudahan penggunaan ini kita buat sesederhana mungkin tetapi juga mempunyai manfaat yang besar. Dari sisi UI, khususnya untuk warna, kita dari BPR sudah punya tema warna. Jadi harapannya aplikasi nanti menggunakan warna-warna yang sesuai dengan temanya BPR. Tetapi tidak mengganggu sisi kenyamanan pengguna. Literasi finansial, literasi digital harus menjadi pertimbangan warna, ukuran huruf, diatur sedemikian rupa. Sehingga penggunaannya tidak hanya “Kaum Milenial” namun juga untuk kalangan yang sebelumnya. Untuk bahasa, dikarenakan untuk pasar lokal, bahasa yang digunakan adalah Bahasa Indonesia.
11	Q	Aplikasi m-banking sangat sensitif soal keamanan, fitur keamanan seperti apa yang akan diterapkan pada aplikasi m-banking Bank ABC?
	A	Dari sisi keamanan, tentunya punya regulasi. Latar belakang regulasi di dunia perbankan, kita punya regulasi yang <i>mempersyaratkan</i> seluruh transaksi harus diamankan. Meskipun dilihat dari infrastruktur, BPR masih belum memiliki infrastruktur yang memadai, tetapi sekurang-kurangnya dari sisi aplikasi sudah mengadopsi praktik terbaik keamanan. Paling minimum misalkan, kata sandi. Kita gunakan kata sandi yang panjangnya sudah diatur sedemikian rupa, kemudian proses enkripsi pengiriman data sekurang-kurangnya kita menggunakan SSL dengan SSL versi terbaru, ada juga fitur untuk PIN. Selain fitur <i>login</i> , juga ada fitur PIN untuk transaksi, <i>hard-token</i> dan <i>session timeout</i> .
12	Q	Kesepakatan kerangka kerja pembuatan aplikasi m-banking Bank ABC:
	A	<ol style="list-style-type: none"> 1. Perancangan menggunakan kerangka kerja <i>scrum</i> dalam jangka waktu 28 hari 2. Fery Mohammad Ferdian sebagai <i>product owner</i> 3. Dede Yusuf sebagai <i>scrum master</i> 4. Awal Ariansyah sebagai <i>developer</i> 5. Komunikasi menggunakan Whatsapp dan aplikasi konferensi video 6. Manajemen proyek menggunakan Github 7. Dukungan platform adalah Android dan iOS

Mengetahui

Cilacap, 28 Februari 2023

Narasumber


Dede Yusuf

Pengawas Eksternal


Fery Mohammad Ferdian

Project Manager Bank ABC

Lampiran 8 Surat Tugas *Project Manager*

SURAT TUGAS
No. 006/ALT-RnD/XII/2022

Yang bertandatangan dibawah ini :

Nama : Dian Razak Bastian, S.E. MM,
Jabatan : Direktur Utama

Memberikan tugas kepada saudara:

Nama : Ferry Moh Ferdian, S.Kom.,M.Kom.
No KTP :
Alamat :
Jabatan : Direktur Operasional
Waktu Pekerjaan : Desember 2022 s/d Selesai

Sebagai Project Manager pada pelaksanaan project implementasi Digitalisasi Bank sesuai Perjanjian Kerjasama Anugerah Lestari dengan Bank pada perusahaan, dan diberikan kewenangan sesuai dengan tugas dan tanggung jawab dan kewenangan yang mengikat.

Demikian Surat Tugas ini dibuat dengan sebenarnya, untuk digunakan sebagaimana mestinya.

Dikeluarkan di : Bandung
Pada Tanggal : 13 November 2022

Hormat kami,
PT. _____



ALTERNATE
Dian Razak Bastian
Direktur Utama

**STMIK KOMPUTAMA
MAJENANG**

Lampiran 9 *Daily Scrum*

Tanggal	Apa yang dikerjakan kemarin?	Apa yang dikerjakan hari ini?	Permasalahan yang dialami?
01 Maret 2023	-	Mengamati aplikasi m-banking lain dan menyusun peta navigasi layar	Membutuhkan akun m-banking tertentu untuk melihat isi aplikasi
02 Maret 2023	Menyelesaikan peta navigasi layar	Melakukan inisiasi proyek minimal dan pengumpulan aset	-
03 Maret 2023	Inisiasi proyek di Figma dan React Native	<i>Prototyping</i> layar <i>splash</i> , <i>onboard</i> , dan <i>login</i>	-
04 Maret 2023	Menyelesaikan <i>prototyping</i> layar <i>splash</i> , <i>onboard</i> , dan <i>login</i>	<i>Coding</i> antarmuka layar <i>splash</i> dan <i>onboard</i>	Perubahan <i>bundle id</i> dari <i>product owner</i> terhadap proyek React Native yang sudah diinisiasi
05 Maret 2023	Menyelesaikan <i>coding</i> layar <i>splash</i> dan <i>onboard</i> serta pergantian <i>bundle id</i>	Melakukan <i>coding</i> layar <i>login</i>	<i>Watermark</i> yang menutupi layar ketika <i>keyboard</i> muncul
06 Maret 2023	Menyelesaikan <i>coding</i> layar <i>login</i> dan mengakali <i>watermark</i>	Menambahkan logika validasi dan integrasi API proses <i>login</i>	Kesulitan validasi kata sandi dengan regex
07 Maret 2023	Mengintegrasikan proses <i>login</i> dan validasi kata sandi dengan <i>library yup-password</i>	<i>Refactor</i> , membuat <i>utility function</i> , dan abstraksi kode ke dalam komponen-komponen <i>reusable</i>	-
08 Maret 2023	Optimasi kode dan <i>sprint review</i>	<i>Prototyping</i> layar-layar akar dan layar <i>loading</i>	-
09 Maret 2023	Menyelesaikan <i>prototyping</i> layar-layar akar dan layar <i>loading</i>	<i>Coding</i> layar <i>loading</i> dan layar-layar akar	-
10 Maret 2023	<i>Coding</i> layar <i>loading</i> dan layar-layar akar	Melanjutkan <i>coding</i> dan integrasi layar beranda dengan API <i>backend</i>	-
11 Maret 2023	Menyelesaikan <i>coding</i> dan integrasi layar beranda	<i>Prototyping</i> dan <i>coding</i> layar informasi rekening serta integrasinya	-
12 Maret 2023	Menyelesaikan fitur informasi rekening	<i>Prototyping</i> dan <i>coding</i> layar-layar mutasi	Bingung menentukan desain mutasi karena tidak ada API untuk melihat detail dari mutasi

Tanggal	Apa yang dikerjakan kemarin?	Apa yang dikerjakan hari ini?	Permasalahan yang dialami?
13 Maret 2023	Percobaan menyelesaikan layar-layar mutasi	Perbaikan layar-layar mutasi sesuai arahan <i>product owner</i> (hanya menampilkan daftar mutasi)	-
14 Maret 2023	Menyelesaikan layar-layar mutasi	<i>Prototyping, coding</i> , dan integrasi API untuk layar informasi produk bank	-
15 Maret 2023	Penyelesaian layar produk bank	Perbaikan layar beranda dan layar informasi produk serta <i>prototyping</i> layar-layar transaksi	-
16 Maret 2023	Perbaikan layar dan <i>prototyping</i> layar-layar transaksi (antar rekening, antar bank, tarik tunai, bayar kredit dan riwayat aktifitas)	<i>Coding</i> dan integrasi API layar transaksi antar rekening, mendesain dan <i>prototyping</i> layar nominal	Kesalahan pada API <i>backend</i> tidak dapat menemukan rekening untuk uji coba
17 Maret 2023	Penyelesaian layar transaksi antar rekening dan <i>prototyping</i> layar nominal	<i>Coding</i> dan integrasi API layar nominal serta <i>prototyping</i> layar konfirmasi, bukti transaksi, dan riwayat aktifitas	-
18 Maret 2023	Penyelesaian layar nominal dan <i>prototyping</i> layar konfirmasi, bukti transaksi, dan riwayat aktifitas	<i>Coding</i> dan integrasi API layar konfirmasi, bukti transaksi, dan riwayat aktifitas serta <i>prototyping</i> semua sub-layar transaksi lainnya	Kelelahan dan kurang enak badan
19 Maret 2023	Penyelesaian layar dari fitur transaksi antar-rekening serta <i>prototyping</i> sub-layar transaksi	<i>Coding</i> dan integrasi API layar dan sub-layar transaksi (antar bank, validasi rekening, cek tagihan, tarik tunai, proses tarik tunai)	
20 Maret 2023	Penyelesaian layar dan sub-layar transaksi	-	Istirahat
21 Maret 2023	-	<i>Prototyping, coding</i> , dan integrasi fitur pembukaan rekening tabungan dan kredit	-

Tanggal	Apa yang dikerjakan kemarin?	Apa yang dikerjakan hari ini?	Permasalahan yang dialami?
22 Maret 2023	Penyelesaian fitur pembukaan rekening	Perbaikan layar bukti transaksi dan proses tarik tunai	Tombol simpan dan bagikan jika dipisah harus melakukan simpan ulang
23 Maret 2023	Penyelesaian perbaikan layar bukti transaksi dan proses tarik tunai	<i>Prototyping</i> dan <i>coding</i> layar pendaftaran aplikasi untuk nasabah	-
24 Maret 2023	Penyelesaian <i>coding</i> fitur pendaftaran untuk nasabah	<i>Coding</i> layar pendaftaran aplikasi untuk non-nasabah	Strategi menangani masukan yang terlalu banyak
25 Maret 2023	Percobaan penyelesaian <i>coding</i> fitur pendaftaran untuk non-nasabah	Melanjutkan <i>coding</i> layar pendaftaran untuk non-nasabah	-
26 Maret 2023	Penyelesaian <i>coding</i> pendaftaran non-nasabah	Pembuatan skema validasi masukan serta integrasi API <i>backend</i>	-
27 Maret 2023	Penyelesaian fitur pendaftaran aplikasi	<i>Prototyping</i> , <i>coding</i> dan integrasi API layar informasi nasabah, informasi bank serta layar bantuan	-
28 Maret 2023	Penyelesaian fitur untuk menampilkan informasi lainnya	<i>Prototyping</i> , <i>coding</i> layar pengaturan serta implementasi fitur keamanan	Pencarian <i>library</i> enkripsi yang lebih lengkap karena beberapa <i>signature</i> berbeda-beda

Cilacap, 20 Juni 2023



Awal Ariansyah
Developer



Dede Yusuf
Scrum Master

Lampiran 10 *Sprint Review*

Sprint : I
Tanggal : 07 Maret 2023

No	Deskripsi	Product Owner	Keterangan
1	Peta alur navigasi layar yang dibuat telah mencakup kebutuhan pengguna berdasarkan API yang ada	Setuju	Sesuai
2	Tampilan dan fungsi layar <i>splash</i> sudah sesuai dan berfungsi dengan baik	Setuju	Sesuai dan berfungsi
3	Tampilan dan fungsi layar <i>onboard</i> sudah sesuai dan berfungsi dengan baik	Setuju	Sesuai dan berfungsi
4	Fitur <i>login</i> sudah sesuai dan berfungsi dengan baik	Setuju	Sesuai dan berfungsi
5	Diperlukan <i>product backlog</i> tambahan	Setuju	1. Layar <i>loading</i> untuk transisi 2. Layar-layar akar seperti beranda untuk layar ke menu-menu lain

Sprint : II
Tanggal : 14 Maret 2023

No	Deskripsi	Product Owner	Keterangan
1	Antarmuka layar <i>loading</i> sudah sesuai termasuk animasi yang dipilih	Setuju	Sesuai
2	Layar beranda sudah sesuai dan berfungsi dengan baik	Tidak setuju	Tampilan kurang sesuai
3	Fitur mutasi sudah sesuai dan berfungsi	Setuju	Sesuai dan berfungsi
4	Antarmuka layar akar transaksi sudah sesuai	Setuju	Benar, tambahkan lagi <i>card</i> riwayat aktifitas
5	Antarmuka layar informasi sudah sesuai	Setuju	Sesuai
6	Fitur melihat informasi produk perbankan sudah sesuai dan berfungsi	Tidak setuju	Sudah berfungsi namun tampilan belum sesuai.
7	Diperlukan <i>product backlog</i> tambahan	Setuju	1. Perbaiki layar beranda 2. Perbaiki layar informasi produk

Sprint : III
Tanggal : 21 Maret 2023

No	Deskripsi	Product Owner	Keterangan
1	Perbaikan layar beranda dan layar informasi produk sudah sesuai dan berfungsi dengan baik	Setuju	Sesuai dan berfungsi
2	Fitur transaksi antar rekening sudah sesuai dan berfungsi dengan baik	Setuju	Sesuai dan berfungsi
3	Fitur transaksi antar bank sudah sesuai dan berfungsi dengan baik	Setuju	Sesuai dan berfungsi
4	Fitur tarik tunai sudah sesuai dan berfungsi dengan baik	Tidak Setuju	Tampilan kurang sesuai
5	Fitur bayar kredit sudah sesuai dan berfungsi dengan baik	Setuju	Sesuai dan berfungsi
6	Antarmuka layar nominal, konfirmasi, dan bukti transaksi sudah sesuai	Tidak Setuju	Nominal sesuai, konfirmasi sesuai, bukti transaksi kurang sesuai
7	Antarmuka layar riwayat aktifitas sudah sesuai dan berfungsi	Setuju	Sesuai dan berfungsi
8	Diperlukan <i>product backlog</i> tambahan	Setuju	1. Perbaikan layar bukti transaksi dengan simpan pdf dan <i>share</i> 2. Perbaikan layar tarik tunai tambahan tombol aksi, perjas tampilan durasi, logo pada QR 3. Fitur keamanan

Sprint : IV
Tanggal : 28 Maret 2023

No	Deskripsi	Product Owner	Keterangan
1	Perbaikan layar bukti transaksi dan layar proses penarikan tunai sudah sesuai dan berfungsi dengan baik	Setuju	Sesuai dan berfungsi
2	Fitur pendaftaran sebagai pengguna aplikasi untuk nasabah sudah sesuai dan berfungsi dengan baik	Setuju	Sesuai dan berfungsi
3	Fitur pendaftaran sebagai pengguna aplikasi untuk non-nasabah sudah sesuai dan berfungsi dengan baik	Setuju	Sesuai dan berfungsi
4	Fitur melihat informasi umum (nasabah, bank, bantuan) sudah sesuai dan berfungsi dengan baik	Setuju	Sesuai dan berfungsi

No	Deskripsi	Product Owner	Keterangan
5	Fitur pengaturan aplikasi sudah sesuai dan berfungsi dengan baik	Setuju	Sesuai dan berfungsi
6	Fitur keamanan sudah sesuai dan berfungsi dengan baik	Setuju	Sesuai dan berfungsi
7	Diperlukan <i>product backlog</i> tambahan	Tidak Setuju	Sejauh ini sudah sesuai dan siap untuk di presentasikan kepada <i>stakeholder</i>

Cilacap, 20 Juni 2023

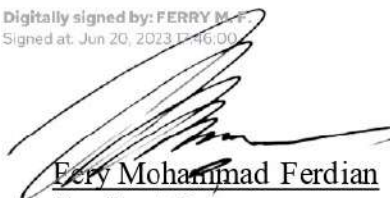


Awal Ariansyah
Developer



Dede Yusuf
Scrum Master

Digitally signed by: FERRY M.F.
Signed at: Jun 20, 2023 13:46:00



Fery Mohammad Ferdian
Product Owner

