

**SKPL-XXXX**

## **SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK**

untuk:

Sri Widowati,M.T

Dipersiapkan oleh:

1301190459 - Andi Muh.Syahrul Ulum


1301190337 - Bagus Seno P.

1301194098 - Nur Laili Ramadhani

1301194263 - Fauzaan Rakan Tama

Program Studi S1 Informatika – Fakultas Informatika Universitas  
Telkom

Jalan Telekomunikasi Terusan Buah Batu, Bandung Indonesia

 <b>UNIVERSITAS Telkom</b>	<b>Program Studi S1 Informatika - Fakultas Informatika</b>	<b>Nomor Dokumen</b>		<b>Halaman</b>
		<b>SKPL-xxx</b>		<b>13</b>
		<b>Revisi</b>	<i>&lt;nomor revisi&gt;</i>	<i>Tgl: &lt;isi tanggal&gt;</i>

--	--	--	--

<i>Prodi S1 Informatika - Universitas Telkom</i>	<i>SKPL-xxx</i>	<i>Halaman 2 dari 13</i>
<b><i>Dokumen ini dan informasi yang ada di dalamnya adalah milik Prodi S1 Informatika-Universitas Telkom dan bersifat rahasia. Dilarang untuk mereproduksi dokumen ini tanpa diketahui oleh Program Studi S1 Informatika, Universitas Telkom</i></b>		

## Daftar Perubahan

Revisi	Deskripsi
<b>A</b>	
<b>B</b>	
<b>C</b>	
<b>D</b>	
<b>E</b>	
<b>F</b>	
<b>G</b>	

INDEX	-	A	B	C	D	E	F	G
TGL								
Ditulis oleh								
Diperiksa oleh								
Disetujui oleh								

## Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

# Daftar Isi

Daftar Perubahan.....	1
Daftar Halaman Perubahan.....	2
Daftar Isi .....	3
1. Pendahuluan.....	6
1.1 Tujuan Penulisan Dokumen .....	6
1.2 Konvensi Dokumen .....	6
1.3 Cakupan Produk .....	6
1.4 Referensi .....	6
2. Overall Description.....	7
2.1 Perspektif Produk.....	7
2.2 Kelas dan Karakteristik Pengguna .....	8
2.3 Arsitektur Perangkat Lunak.....	8
2.4 Batasan Perancangan dan Implementasi.....	8
2.5 Dokumentasi Pengguna .....	9
3. Requirements Antarmuka Eksternal.....	10
3.1 Antarmuka Pengguna .....	10
3.2 Antarmuka Perangkat Keras .....	11
3.3 Antarmuka Perangkat Lunak.....	11
3.4 Antarmuka Komunikasi .....	12
4. Fitur Sistem (Use Cases) .....	14
4.1 Use Case 1 .....	14
4.1.1 Nama Use Case: Transaksi .....	14
4.1.2 Tujuan: customer mengirim uang ke rekening lain .....	15
4.1.3 Input: 1. Masukkan nomer rekening yang ingin di transfer .....	15
2. Masukkan jumlah uang yang ingin dikirim .....	15
4.1.4 Output: berhasil mentrasfer sejumlah uang yang dengan nomor rekening yang dituju	15
4.1.5 Skenario Utama: .....	15
4.1.6 Prakondisi: .....	15
4.1.7 Langkah-langkah: .....	15
4.1.8 Pascakondisi.....	15

4.1.9	Skenario eksepsional 1 .....	15
4.2	Use Case 2 .....	16
4.2.1	Nama Use Case: Deposit.....	16
4.2.2	Tujuan: customer mendeposit sejumlah uang dengan jangka waktu.....	16
4.2.3	Input:.....	16
1.	Customer menginput formulir aplikasi pembukaan rekening. ....	16
4.2.4	Output: .....	16
1.	Jika Berhasil akan dikabari lewat email. ....	16
4.2.5	Skenario Utama: .....	16
4.2.6	Prakondisi: .....	16
4.2.7	Langkah-langkah: .....	16
4.2.8	Pascakondisi.....	17
4.2.9	Skenario eksepsional 1 .....	17
4.3	Use Case 3 .....	17
4.3.1	Nama Use Case: Cek Saldo .....	17
4.3.2	Tujuan: Customer mendapat insight jumlah saldo miliknya. ....	17
4.3.3	Input:.....	17
4.3.4	Output: .....	18
4.3.5	Skenario Utama: .....	18
4.3.6	Prakondisi: .....	18
4.3.7	Langkah-langkah .....	18
4.3.8	Pascakondisi.....	18
4.3.9	Skenario eksepsional 1 .....	18
4.4	Use Case 4 .....	18
4.4.1	Nama Use Case: Payment Merchant .....	18
4.4.2	Tujuan: customer bertransaksi dengan merchant menggunakan barkode.....	18
4.4.3	Input:.....	18
4.4.4	Output: .....	18
4.4.5	Skenario Utama: .....	18
4.4.6	Prakondisi: .....	19
4.4.7	Langkah-langkah .....	19
4.4.8	Pascakondisi.....	19
4.4.9	Skenario eksepsional 1 .....	19
4.5	Use Case 5 .....	19

4.5.1	Nama Use Case: Pembayaran.....	19
4.5.2	Tujuan: Customer membayar tagihan .....	19
4.5.3	Input:.....	19
4.5.4	Output:.....	20
4.5.5	Skenario Utama: .....	20
4.5.6	Prakondisi: .....	20
4.5.7	Langkah-langkah: .....	20
4.5.8	Pascakondisi.....	20
4.5.9	Skenario eksepsional 1 .....	20
4.6	Use Case 6 .....	20
4.6.1	Nama Use Case: Pembelian .....	20
4.6.2	Tujuan: Customer Membeli atau topup saldo .....	21
4.6.3	Input:.....	21
4.6.4	Output:.....	21
4.6.5	Skenario Utama: .....	21
4.6.6	Prakondisi: .....	21
4.6.7	Langkah-langkah .....	21
4.6.8	Pascakondisi.....	21
4.6.9	Skenario eksepsional 1 .....	21
5.	Requirements Nonfungsional Lainnya.....	22
5.1	Atribut Kualitas Perangkat Lunak .....	22
	Lampiran A : Daftar Kata-Kata Sukar .....	23
	Lampiran B : Analysis Models .....	23

# **1. Pendahuluan**

## **1.1 Tujuan Penulisan Dokumen**

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) merupakan spesifikasi kebutuhan perangkat lunak yang akan dikembangkan. Dokumen ini digunakan oleh pengembang perangkat lunak sebagai acuan teknis pengembangan perangkat lunak pada tahap selanjutnya.

## **1.2 Konvensi Dokumen**

Perangkat lunak yang akan dikembangkan adalah perangkat lunak aplikasi “Smart Banking”, yaitu perangkat lunak yang digunakan oleh seluruh mahasiswa Universitas Telkom baik yang sudah umur 17 tahun keatas maupun 17 tahun kebawah. untuk melakukan transaksi dengan menggunakan mobile phone pribadi tanpa harus pergi ke ATM. Smart Banking dapat melakukan hal-hal berikut :

1. User dapat melakukan transfer ke rekening lain.
2. User dapat melakukan top-up seperti top-up gopay, ovo, pulsa
3. User dapat melihat saldo rekening melalui mobile phone.
4. User dapat membayar tagihan seperti tagihan listrik, internet, air,dll.

## **1.3 Cakupan Produk**

Smart Banking adalah aplikasi/perangkat lunak yang diperuntukkan untuk seluruh mahasiswa Telkom University. Smart Banking dibuat untuk memudahkan mahasiswa Universitas Telkom khususnya yang belum memiliki sebuah rekening. Sebelum menggunakan Smart Banking, user harus melakukan pendaftaran di business center yang ada di Universitas Telkom lalu user akan mendapatkan user ID dan password yang akan digunakan untuk log in ke aplikasi Smart Banking lalu menggunakan layanan-layanan yang telah disediakan di Smart Banking.

## **1.4 Referensi**

1. Muchammad Husni, Erry Yudianto. Perangkat Lunak untuk Layanan Mobile Banking Berbasis Teknologi Wireless Application Protocol (WAP). 2006.
2. Andy Pramurjadi, Aang Nahrowi, Arifa Desfamita. VMS. 2011.



## 2. Overall Description

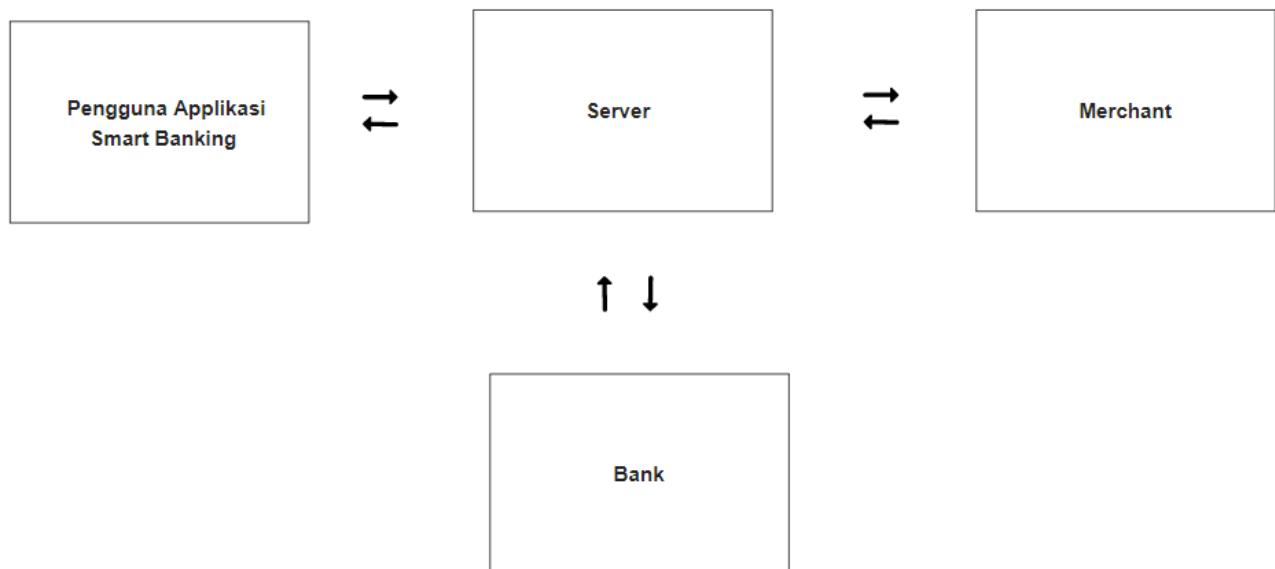
### 2.1 Perspektif Produk

Smart Banking adalah suatu aplikasi mobile banking yang akan di kembangkan di Telkom Univesity yang bertujuan mempermudah melakukan transaksi keuangan di manapun dan kapan saja ,aplikasi ini diperuntukan bagi mahasiswa telkom, dosen, maupun karyawan pegawai telkom University. Adapun Fitur Utama dari aplikasi ini seperti layanan informasi yaitu menampilkan saldo nasabah, mutasi rekening, maupun suku bunga. kemudian fitur layanan transaksi yaitu melakukan transfer sejumlah uang ke sesama smartbanking, antarbank, maupun virtual account billing. Adapun fitur pembayaran tagihan yaitu nasabah dapat mebayar tagihan seperti listrik,air, maupun layanan internet. Fitur layanan pembelian yaitu nasabah dapat melakukan topup seperti aplikasi yang menggunakan cashless, contohnya topup ovo, gopay, shoope pay, dll. Nasabah yang ingin menggunakan aplikasi ini harus melakukan registrasi dan melakukan pengisiin saldo Smart Banking minimal Rp. 250.000, Kemudian nasabah akan dibuatkan akun Smart Banking.

Fungsi Produk ;

1. Fungsi login
2. Fungsi melihat informasi saldo
3. Fungsi transfer
4. Fungsi deposito
5. Fungsi top-up untuk e-wallet
6. Fungsi pembayaran tagihan
7. Fungsi pembayaran pendidikan dll.
8. Fungsi melihat riwayat transaksi
9. Fungsi logout

Sistem cara kerja SmartBanking



Gambar 1. Sitem kerja Smart Banking.

## 2.2 Kelas dan Karakteristik Pengguna

Kelas	Hak Akses ke Aplikasi
Mahasiswa Universitas Telkom	Hak akses mahasiswa terhadap aplikasi Smart Banking adalah melakukan login, melakukan transaksi, melakukan top-up untuk e-wallet, melihat jumlah saldo rekening, melakukan pembayaran tagihan, pembayaran pendidikan seperti membayar uang UKT dan melakukan logout.
Dosen dan Pegawai Universitas Telkom	Hak akses dosen dan pegawai terhadap aplikasi Smart Banking hampir sama dengan hak akses mahasiswa tetapi Smart Banking lebih mengutamakan mahasiswa.
Admin	Hak akses admin terhadap aplikasi Smart Banking adalah mengelola akun mahasiswa dan mengelola akun dosen/pegawai Universitas Telkom

## 2.3 Arsitektur Perangkat Lunak

### 2.3.1 Software

Software Smart Banking terdiri atas *backend* dan *frontend*. Backend yang digunakan adalah bahasa pemrograman Node.JS dengan menggunakan framework ExpressJs. Lalu, frontend yang digunakan adalah *react native*.

### 2.3.2 Operation

Software server akan di-deploy di VPS (Compute Engine) di Google Cloud Platform (GCP). Adapun spesifikasi VPS yang digunakan adalah sebagai berikut:

- 4 vCPU
- 16GB RAM
- Linux Ubuntu 18.04 LTS.

Adapun aplikasi berupa aplikasi *mobile* yang dibentuk dari framework React Native akan di-deploy di PlayStore dan AppStore. Semua user bisa mendapatkan aplikasi dari kedua platform marketplace tersebut.

## 2.4 Batasan Perancangan dan Implementasi

1. Admin memiliki hak akses yang lebih tinggi dari *user* biasa. Namun, harus tetap ada protokol yang menjaga keamanan sistem dari admin itu sendiri.

2. Keseluruhan sistem menggunakan bahasa *JavaScript*.
3. Desain antarmuka harus sesuai standar.
4. Dokumentasi lengkap.
5. Adanya sistem *backup* untuk semua data.

## 2.5 Dokumentasi Pengguna

### 2.5.1 Manual Pengguna

1. Login
  - a. Buka aplikasi Smart Banking.
  - b. Masukkan email dan PIN pada sesuai kolomnya.
  - c. Tekan tombol Login.
2. Pengecekan Saldo
  - a. Lakukan Login seperti pada langkah 1.
  - b. Perhatikan jumlah saldo pada tampilan 'Saldo Anda'.
3. Pembayaran *merchant*
  - a. Lakukan Login seperti pada langkah 1.
  - b. Tekan 'Pembayaran'.
  - c. Pilih merchant.
  - d. Ikuti petunjuk sesuai merchant.
4. Pembelian dari *merchant*
  - a. Lakukan Login seperti pada langkah 1.
  - b. Tekan 'Pembelian'.
  - c. Pilih merchant.
  - d. Ikuti petunjuk sesuai merchant.
5. Transfer
  - a. Lakukan Login seperti pada langkah 1.
  - b. Tekan 'Transfer'.
  - c. Pilih tujuan bank, apakah bank yang sama atau berbeda.
  - d. Masukkan nomor rekening tujuan.
  - e. Masukkan jumlah saldo yang akan ditransfer.
  - f. Masukkan kode password transaksi.
  - g. Tekan 'Transfer'.
6. Pengecekan data rekening.
  - a. Lakukan Login seperti pada langkah 1.
  - b. Tekan 'Rekeningku'.
  - c. Perhatikan data yang ditampilkan.

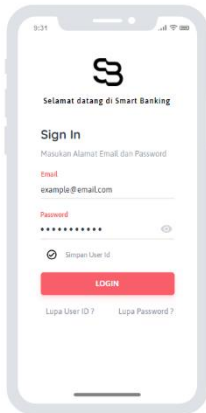
### 2.5.2 Protokol Pelaporan Teknis

Apabila terjadi masalah pada teknis, harap segera laporkan dengan mengirim email pada [laporan@smartbanking.com](mailto:laporan@smartbanking.com). Apabila urgensi terlalu tinggi, maka harap segera telepon ke +620873812738.

### 3. Requirements Antarmuka Eksternal

#### 3.1 Antarmuka Pengguna

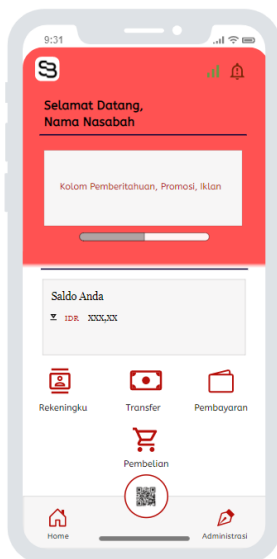
##### 1. Login Pengguna



Gambar 2. Login Aplikasi Smart Banking.

Tampilan pertama jika memasuki aplikasi Smart Bankking, Pengguna harus memasukan akun yang sudah di buat, untuk melanjutkan penggunaan Smart Banking.

##### 2. Setelah Login



Gambar 3. Halaman Home Aplikasi Smart Banking.

Tampilan setelah pengguna login, pengguna dapat melihat jumlah saldonya, dapat melihat detail rekeningnya, dapat Transfer uang, Pembayaran, maupun Pembelian.

## 3.2 Antarmuka Perangkat Keras

Perangkat yang dapat digunakan untuk Smart Banking adalah mobile phone baik dari platform Android maupun iOS. Hardware yang akan digunakan adalah tentunya hardware pendukung jaringan internet. Adapun protokol komunikasi jaringan internet tersebut menggunakan protokol komunikasi HTTPS (Hyper Text Transfer Protocol Secure).

Adapun untuk hardware server:

- Intel® Xeon® Processor E7-8890 v4
- 128 GB RAM
- 100 TB SSD

## 3.3 Antarmuka Perangkat Lunak

Data yang Dikomunikasikan

- Login
  - Email
  - Password
- Transfer
  - Token login
  - No Rek tujuan
  - Nominal
- Payment
  - Merchant Payment ID
  - Token login
- Purchase
  - Merchant Product ID
  - Token login
- Mengambil data user
  - Token login

Spesifikasi Perangkat Lunak

- Server
  - Framework: Expressjs (Node.JS)
  - Sistem Operasi: Linux Ubuntu 18.04 LTS
  - Protokol jaringan: HTTPS
- Client
  - Framework: React Native
  - Deployment: PlayStore & AppStore

Protokol API

- /api/login
  - Accept body:
    - username
    - password

- Responses:
  - 204: success
  - 401: failed
- /api/transfer
  - Accept body:
    - nominal
    - targetNumber
    - token
  - Responses
    - 204: success
    - 401: Unauthorized
    - 404: targetNumber not found
    - 500: Server error
- /api/merchantPayment
  - Accept body:
    - merchantProductId
    - token
  - Responses
    - 204: success
    - 401: Unauthorized
    - 404: targetNumber not found
    - 500: Server error
- /api/merchantPurchase
  - Accept body:
    - merchantProductId
    - token
  - Responses
    - 204: success
    - 401: Unauthorized
    - 404: targetNumber not found
    - 500: Server error
- /api/user
  - Accept body:
    - token
  - Responses
    - 200: success with data
    - 401: Unauthorized
    - 404: user not found
    - 500: Server error

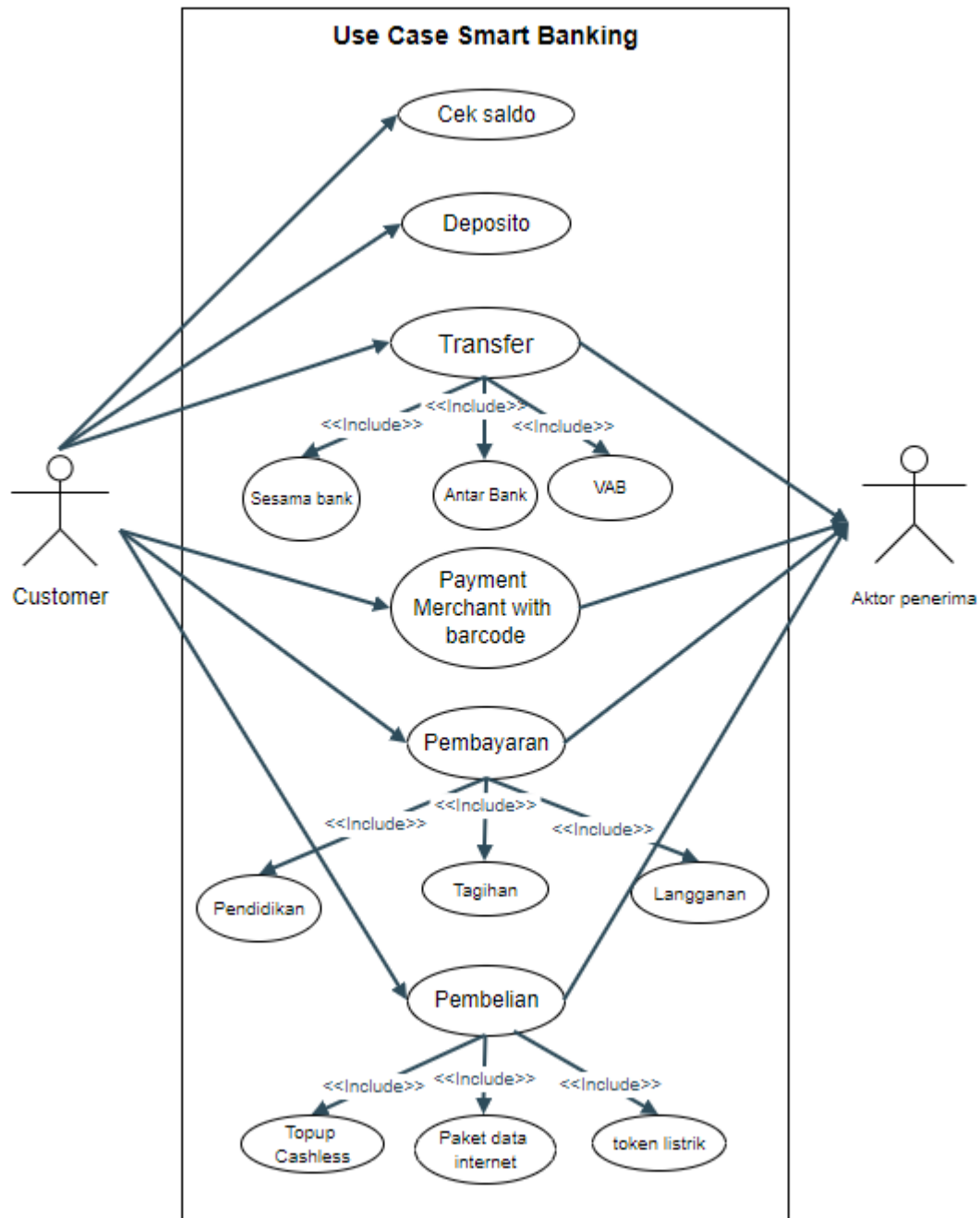
### 3.4 Antarmuka Komunikasi

Komunikasi apabila pengguna ingin bertanya atau ada masalah bisa menghubungi Customer Service, Standar protokol komunikasi yang berkaitan dengan Smart Banking antara lain :

1. Email server
2. Pengiriman data menggunakan HTTPS sebagai standar, HTTP is not allowed. Data dikirim dengan terenkripsi.

3. Token terenkripsi dengan teknologi JWT.

## 4. Fitur Sistem (Use Cases)



Gambar 4. Diagram Use Case

### 4.1 Use Case 1

#### 4.1.1 Nama Use Case: Transaksi



**4.1.2 Tujuan:** customer mengirim uang ke rekening lain

**4.1.3 Input:** 1. Masukkan nomer rekening yang ingin di transfer  
2. Masukkan jumlah uang yang ingin dikirim

**4.1.4 Output:** berhasil mentrasfer sejumlah uang yang dengan nomor rekening yang dituju

**4.1.5 Skenario Utama:**

Use case dimulai dari customer login menggunakan username dan password yang sudah dibuat. kemudian customer memilih transaksi di halaman home. Lalu customer memilih sesama bank, antar bank, atau Virtual Account Billing.

**4.1.6 Prakondisi:**

1. Customer harus sudah sign in
2. Aplikasi terhubung dengan jaringan
3. Customer memiliki uang

**4.1.7 Langkah-langkah:**

Actor Action	System Action
1. Customer Login menggunakan akun yang sudah dibuat.	2. Sistem menampilkan halaman home
3. Customer memilih Transfer	4. Sistem menampilkan halaman transfer
5. Customer memilih antara sesama bank, antarbank, atau Virtual Account Billing	6. Sistem menampilkan form yang akan diinput oleh customer
7. Customer meninputkan nomer rekening bank yang dituju	
8. Customer meninputkan jumlah uang yang ingin di transfer	9. Sistem menampilkan halaman konfirmasi
10. Customer meninputkan password transaksi	11. Sistem menampilkan halaman berhasil

**4.1.8 Pascakondisi**

1. uang customer berpindah ke rekening lain yang dituju

**4.1.9 Skenario eksepsional 1**

1. Step 1 : Customer salah memasuki username atau password. Sistem mengirim kesalahan.
2. Step 7 : customer salah meninputkan nomer rekening yang di tuju, Tampilkan kesalahan
3. Step 8 : customer memasukan lebihh dari jumlah uang yang dimiliki.Tampilkan kesalahan

4. Step 9 : customer salah memasuki password transaksi. Tampilkan kesalahan

## 4.2 Use Case 2

### 4.2.1 Nama Use Case: Deposit

**4.2.2 Tujuan:** customer menodeposit sejumlah uang dengan jangka waktu.

### 4.2.3 Input:

1. Customer menginput formulir aplikasi pembukaan rekening.
2. Customer menginput KTP atau identitas lainnya.
3. Melakukan setoran untuk pembukaan rekening sebesar Rp10.000.000,- untuk mata uang Rupiah.

### 4.2.4 Output:

1. Jika Berhasil akan dikabari lewat email.
2. Jika sudah lebih dari jangka waktu, pihak Smart Banking akan mentransfer bunga yang di dapat.

### 4.2.5 Skenario Utama:

Use case dimulai dari customer login menggunakan username dan password yang sudah dibuat. kemudian customer memilih halaman administrasi, customer memilih pembuatan rekening deposito.

### 4.2.6 Prakondisi:

1. Customer harus sudah sign in.
2. Aplikasi terhubung dengan jaringan.
3. Customer memiliki uang.

### 4.2.7 Langkah-langkah:

Actor Action	System Action
1. Customer login dengan akun yang sudah dibuat	2. Sistem menampilkan halaman home
3. Customer memilih halaman administrasi	4. Sistem menampilkan beberapa pilihan
5. Customer memilih Smart Deposito	6. Sistem menampilkan syarat dan sistem kerja deposito

7. Customer menyetujui	8. Sistem menampilkan halaman formulir yang wajib diisi bagi customer, guna membuat rekening deposito
9. Customer menginputkan formulir apa saja yang di butuhkan	10. Sistem menampilkan halaman formulir sedah di proses
	11. Sitem mengirim email tanda verifikasi pembuatan rekening deposito
12. Customer memverifikasi	12. Sistem memasuki aplikasi smartbank dan menampilkan halaman berhasil di verifikasi
	13. Sistem menampilkan halaman jumlah uang yang di depositkan, minimal setoran awal sebesar 10.000.000 mata uang Rupiah.
14. Customer memasukan jumlah uang yang ingin di depositkan	15. Sistem menampilkan halaman berhasil di depositkan.

#### **4.2.8 Pascakondisi**

1. uang customer tersimpan dalam deposito dan tidak bisa diambil sampai jangka waktu yang telah ditentukan.

#### **4.2.9 Skenario eksepsional 1**

1. Step 1 : Customer salah memasuki username atau password. Sistem mengirim kesalahan.
2. Step 9 : Customer salah mengisi formulir yang di butuhkan. Sitem mengirim email formulir tidak di terima.
3. Step 14 : Customer mengisi jumlah uang yang melebihi tabungannya. Sistem mengirim kesalahan

### **4.3 Use Case 3**

#### **4.3.1 Nama Use Case: Cek Saldo**

**4.3.2 Tujuan:** Customer mendapat insight jumlah saldo miliknya.

**4.3.3 Input:**

1. Customer Login memasukan Username dan password.

#### **4.3.4 Output:**

1. Sistem melihatkan jumlah saldo rekening customer.

#### **4.3.5 Skenario Utama:**

Customer membuka aplikasi Smart Banking, Customer Login menggunakan akun yang sudah di buat, Customer memilih halaman Saldo rekening.

#### **4.3.6 Prakondisi:**

1. Customer harus sudah sign in.
2. Aplikasi terhubung dengan jaringan.

#### **4.3.7 Langkah-langkah**

Actor Action	System Action
1. Customer Login menggunakan akun yang sudah di buat	2. Sistem menampilkan halaman home, yang tertera saldo rekening pemilik.
3. Customer mengklik jumlah saldo, jika ingin melihat detail saldo miliknya	4. Sitem menampilkann halaman saldo rekening

#### **4.3.8 Pascakondisi**

1. Uang customer ditampilkan di layar

#### **4.3.9 Skenario eksepsional 1**

1. Step 1 : Customer salah memasuki username atau password. Sistem mengirim kesalahan.

### **4.4 Use Case 4**

#### **4.4.1 Nama Use Case: Payment Merchant**

**4.4.2 Tujuan:** customer bertransaksi dengan merchant menggunakan barcode

#### **4.4.3 Input:**

1. Customer menginput kode barcode.
2. Customer menginput jumlah uang yang ingin di transfer ke merchant.

#### **4.4.4 Output:**

1. Customer berhasil membayar ke merchant.

#### **4.4.5 Skenario Utama:**

Use case berjalan dimulai dari customer login menggunakan akun yang sudah di buat. Lalu customer memilih kode barcode yang ada di halaman home. Customer memfoto kode barcode dari merchant yang ingin di bayar.

#### **4.4.6 Prakondisi:**

- Customer harus sudah sign in
- Aplikasi terhubung dengan jaringan
- Customer memiliki uang

#### **4.4.7 Langkah-langkah**

Actor Action	System Action
1. Customer Login menggunakan akun yang sudah di buat	2. Sistem menampilkan halaman home
3. Customer memilih lambang kode barkode yang ada dibawah tengah	4. Sistem menampilkan halaman scan kode barkode
5. Customer mescan kode barkode dari mechant yang ingin di bayar	6. SIstem menampilkan halam jumlah uang yang hendak di bar customer
7. Customer men inpukan jumlah uang yang ingin di bayar	8. Sistem menampilkan halaman konfirmasi pembayaran
9. Customer menyetujui dan memasukan password transaksi	10. Sistem menampilkan halaman berhasil melakukan pembayaran

#### **4.4.8 Pascakondisi**

1. Uang customer berpindah ke rekening merchant
2. Customer mendapat pelayanan dari merchant (pembelian/jasa)

#### **4.4.9Skenario eksepsional 1**

1. Step 1 : Customer salah memasuki username atau password. Sistem mengirim kesalahan.
2. Step 7 : Customer mengisi jumlah uang yang melebihi tabungannya. Sistem mengirim kesalahan.

### **4.5 Use Case 5**

#### **4.5.1 Nama Use Case: Pembayaran**

#### **4.5.2 Tujuan:** Customer membayar tagihan

#### **4.5.3 Input:**

1. Customer menginput kode Tagihan

#### **4.5.4 Output:**

1. Tagihan berhasil dibayar

#### **4.5.5 Skenario Utama:**

Use case dimulai dari customer login menggunakan username dan password yang sudah dibuat. kemudian customer memilih Pembayaran yang tertera di halaman home. Lalu customer dapat memilih tagihan apa saja yang bisa di bayar.

#### **4.5.6 Prakondisi:**

1. Customer harus sudah sign in
2. Aplikasi terhubung dengan jaringan
3. Customer memiliki uang

#### **4.5.7 Langkah-langkah:**

Actor Action	System Action
1. Customer Login menggunakan akun yang sudah di buat	2. Sistem menampilkan halaman home
3. Customer memilih Pembayaran yang tertera di halaman home	4. Sistem menampilkan halaman pembayaran, dan menampilkan beberapa pilihan pembayaran
5. Customer memilih pembayaran yang ingin dibayarkan seperti tagihan listrik, air, Langganan internet dll	6. Sistem menampilkan halaman form yang akan di input customer
7. Customer men inputkan kode pembayaran	8. Sistem menampilkan halaman konfirmasi pembayaran
9. Customer menyetujui dan memasukan password transaksi	10. Sistem menampilkan halaman berhasil melakukan pembayaran

#### **4.5.8 Pascakondisi**

1. uang customer berkurang karena pembayaran
2. Customer berhasil membayar tangihan

#### **4.5.9 Skenario eksepsional 1**

1. Step 1 : Customer salah memasuki username atau password. Sistem mengirim kesalahan.
2. Step 7 : Customer salah meninputkan kode pembayaran. Sistem mengirim kesalahan
3. Step 9 : Customer salah memasuki password transaksi. Sistem mengirim kesalahan.

## **4.6 Use Case 6**

### **4.6.1 Nama Use Case: Pembelian**

**4.6.2 Tujuan:** Customer Membeli atau topup saldo

**4.6.3 Input:**

1. Customer menginput kode atau nomor tujuan yang hendak melakukan pembelian

**4.6.4 Output:**

1. Berhasil membeli sesuatu yang diinginkan customer

**4.6.5 Skenario Utama:**

Use case dimulai dari customer login menggunakan username dan password yang sudah dibuat. kemudian customer memilih Pembelian yang tertera di halaman home. Lalu customer dapat memilih untuk pembelian yang di butuhkan seperti topup gopay, paket data, link aja dll.

**4.6.6 Prakondisi:**

1. Customer harus sudah sign in
2. Aplikasi terhubung dengan jaringan
3. Customer memiliki uang

**4.6.7 Langkah-langkah**

Actor Action	System Action
1. Customer Login menggunakan akun yang sudah di buat	2. Sistem menampilkan halaman home
3. Customer memilih Pembelian yang tertera di halaman home	4. Sistem menampilkan halaman pembelian, dan menampilkan beberapa pilihan pembelian
5. Customer memilih Pembelian yang ingin dibeli atau topup seperti topup gopay, paket data, link aja dll.	6. Sistem menampilkan halaman form yang akan di input customer
7. Customer men inputkan kode pembelian	8. Sistem menampilkan halaman konfirmasi pembayaran
9. Customer menyetujui dan memasukan password transaksi	10. Sistem menampilkan halaman berhasil melakukan Pembelian

**4.6.8 Pascakondisi**

1. uang customer berkurang karena pembelian paket data internet
2. Customer berhasil membeli sesuatu yang hendak dibeli

**4.6.9 Skenario eksepsional 1**

1. Step 1 : Customer salah memasuki username atau password. Sistem mengirim kesalahan.
2. Step 7 : Customer salah meninputkan kode pembelian. Sistem mengirim kesalahan.
3. Step 9 : Customer salah memasuki password transaksi. Sistem mengirim kesalahan.

## 5. Requirements Nonfungsional Lainnya

### 5.1 Atribut Kualitas Perangkat Lunak

Kriteria Kualitas	Tuntutan kualitas
Ketersediaan	Aplikasi ini tersedia untuk mahasiswa, dosen, pegawai di Universitas Telkom
Kehandalan	Aplikasi ini dapat diakses setiap saat dan harus memiliki koneksi internet
Maintainability	Sistem ini telah dibuat sedemikian rupa agar dapat memberikan kehandalan yang baik bagi penggunaanya. Masalah yang sekiranya mungkin terjadi adalah crash.
Keamanan	Keamanan user dapat terjaga karena telah memiliki username dan password



## Lampiran A: Daftar Kata-Kata Sukar

SKPL	SKPL merupakan dokumen yang menjelaskan spesifikasi dari perangkat lunak yang akan dikembangkan
Crash	Crash adalah suatu keadaan dimana sebuah komputer atau program, baik aplikasi atau bagian dari suatu sistem operasi berhenti berfungsi.
insight	interpretasi tren dalam perilaku manusia yang bertujuan untuk meningkatkan efektivitas suatu produk atau layanan bagi konsumen, serta meningkatkan penjualan untuk keuntungan finansial dari mereka yang menyediakan produk atau layanan.
barkode	menyimpan informasi mengenai suatu produk atau barang-barang yang dilabeli dengan barcode. Biasanya barcode menyimpan informasi berupa tanggal kadaluarsa, kode produksi dan nomor identitas.

## Lampiran B : Analysis Models

