

BAB IV

HASIL DAN ANALISIS PENELITIAN

4.1 Analisis Perancangan Sistem

Perancangan sistem ini didasarkan pada evaluasi sistem sebelumnya atau sistem yang sedang beroperasi, sebagaimana diuraikan dalam Bab III. Dengan mempertimbangkan masalah-masalah yang terungkap dalam penelitian ini, peneliti berupaya untuk menguraikan secara terperinci bagaimana mengembangkan sistem dengan menyertakan fitur *Payment Gateway*. Dikembangkannya sistem ini bertujuan untuk meningkatkan efisiensi dan kemudahan dalam proses pembayaran uji kompetensi, sekaligus memperluas pilihan pembayaran bagi mahasiswa serta mempercepat akses layanan. Selain itu, sistem ini juga dimaksudkan untuk memfasilitasi pemantauan transaksi oleh unit LSP dan bagian keuangan, sehingga memperbaiki proses pencatatan dan rekonsiliasi transaksi, serta mempercepat proses administrasi dengan menghilangkan proses verifikasi pembayaran secara manual.

Berikut adalah perbandingan antara sistem sekarang dengan sistem yang akan dikembangkan.

Tabel 4.1 Perbandingan Sistem

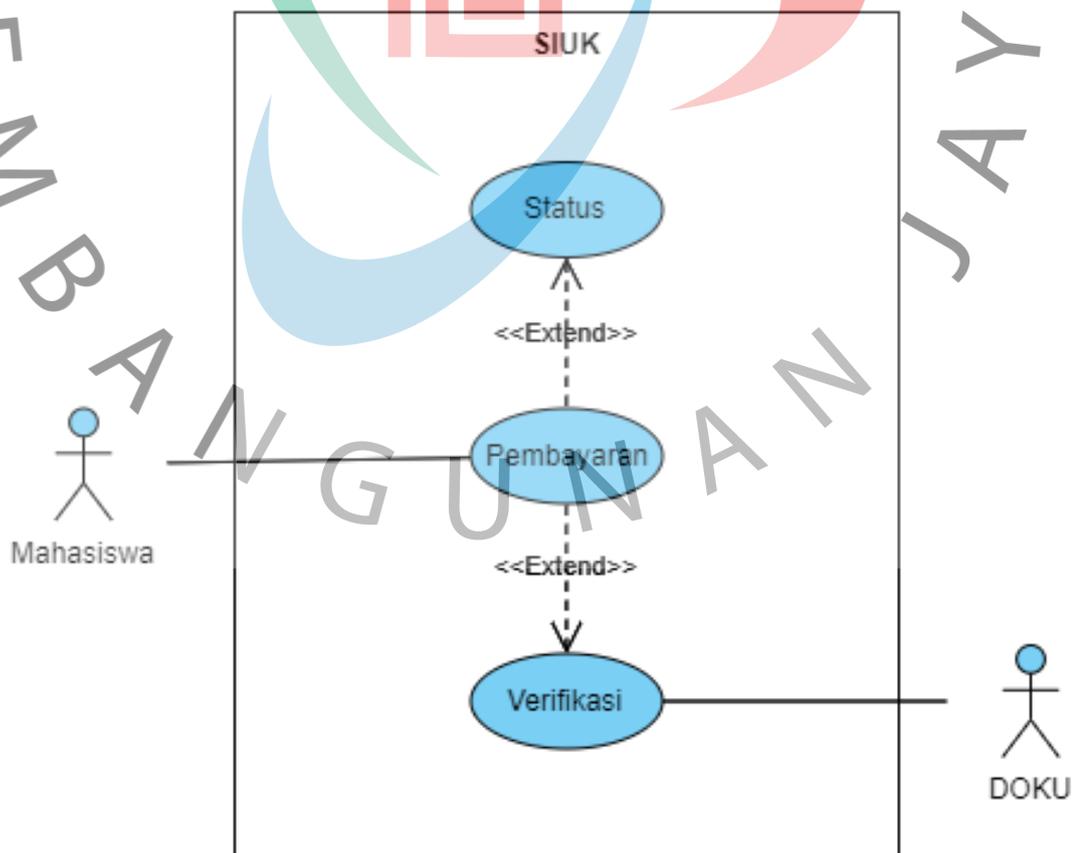
Aspek	Sistem Lama	Sistem Baru
Proses Pembayaran	Manual dan memerlukan verifikasi.	Otomatis dengan integrasi <i>Payment Gateway</i> .
Keamanan Data	Pembayaran yang manual memiliki keamanan yang rentan dan memungkinkan terjadinya <i>fraud</i> .	Implementasi enkripsi data yang kuat dan otentifikasi ganda.
Monitoring Transaksi	Manual dan rentan kesalahan.	Dilakukan secara <i>real-time</i> dalam dashboard DOKU atau dashboard sistem oleh unit LSP dan Keuangan.
Rekonsiliasi Transaksi	Dilakukan secara manual memungkinkan kesalahan.	Rekonsiliasi transaksi otomatis dengan pencatatan yang akurat.

Proses Administrasi	Memerlukan verifikasi yang memakan waktu.	Mempercepat proses administrasi karena tidak perlu verifikasi pembayaran secara manual.
Skalabilitas	Tidak dapat menangani pertumbuhan peserta dengan baik.	Dapat ditingkatkan kapasitas dengan mudah sesuai kebutuhan pertumbuhan peserta.
Ketersediaan Fitur	Terbatas	Menyediakan fitur yang lebih lengkap dan responsif

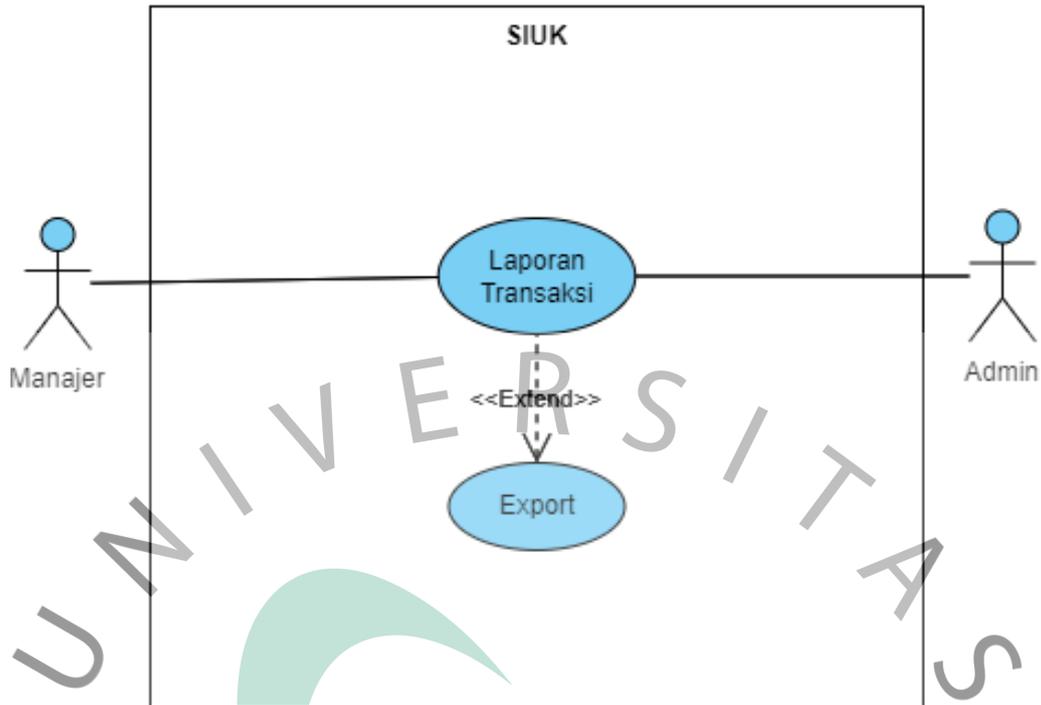
4.2 Perancangan Diagram Sistem Usulan

Untuk memastikan bahwa pengembangan sistem dapat divisualisasikan dengan jelas dan terstruktur maka dari itu peneliti mencoba menggunakan beberapa model diagram yang seperti diagram *use-case*, spesifikasi *use-case*, *activity diagram*, *sequence diagram* dan *class diagram*.

4.2.1 Use-Case Diagram



Gambar 4.1 Diagram Use Case Pembayaran



Gambar 4.2 Diagram Use Case Laporan Transaksi

4.2.2 Spesifikasi Use-Case

Tabel 4.2 Spesifikasi Diagram Use Case

Use Case Name: Pembayaran	ID: SIUK - 01	Priority: High
Description: Mahasiswa melakukan pembayaran pada SIUK kemudian menu pembayaran terhubung langsung dengan <i>payment gateway</i> DOKU yang menyediakan berbagai metode pembayaran. Mahasiswa diberikan kesempatan selama 30 menit untuk menyelesaikan pembayaran, apabila dalam waktu tersebut pembayaran belum diselesaikan maka pembayaran tersebut dianggap gagal dan mahasiswa dapat melakukan pembayaran kembali.		
Actor: <ul style="list-style-type: none"> - Mahasiswa - DOKU 		
Trigger: <p>DOKU: apabila mahasiswa telah melakukan pembayaran (uang telah dikirim ke lembaga pengelola keuangan).</p> <p>SIUK : apabila mahasiswa telah melakukan pembayaran dan berhasil, maka akan langsung terverifikasi dalam sistem dan akan menerima notifikasi di sistem dan email.</p>		

Pre-conditions:

1. Mahasiswa telah berhasil registrasi akun pada SIUK.
2. Mahasiswa masuk menggunakan kombinasi email dan *password* yang telah terdaftar sebelumnya.
3. Mahasiswa berhasil melakukan verifikasi akun.
4. Mahasiswa berhasil melakukan verifikasi dokumen administratif.
5. Mahasiswa melakukan pembayaran dengan metode yang dipilih.

Post-conditions:

1. Mahasiswa menerima tagihan yang harus dibayarkan.
2. Mahasiswa memilih metode pembayaran yang akan dilakukan.
3. Mahasiswa berhasil melakukan pembayaran melalui sistem SIUK sesuai metode yang dipilih (verifikasi berhasil dilakukan oleh sistem).
4. Mahasiswa menerima notifikasi status pembayaran.

	Mahasiswa	System	DOKU
Normal Case:	1. Memilih metode pembayaran		
		2. Bayar sesuai dengan metode yang dipilih	
			3. Verifikasi Pembayaran
		5. Update Status	
		6. Kirim Notifikasi	
	7. Menerima Notifikasi		

Exceptions:

1. Apabila mahasiswa belum terdaftar maka akan diarahkan terlebih dahulu untuk melakukan registrasi.
2. Apabila akun mahasiswa belum terverifikasi, maka mahasiswa akan diarahkan untuk melengkapi data verifikasi.
3. Apabila verifikasi dokumen mahasiswa belum lulus, maka mahasiswa akan diarahkan untuk melengkapi dokumen administratif.

Tabel 4.3 Spesifikasi Diagram Use Case

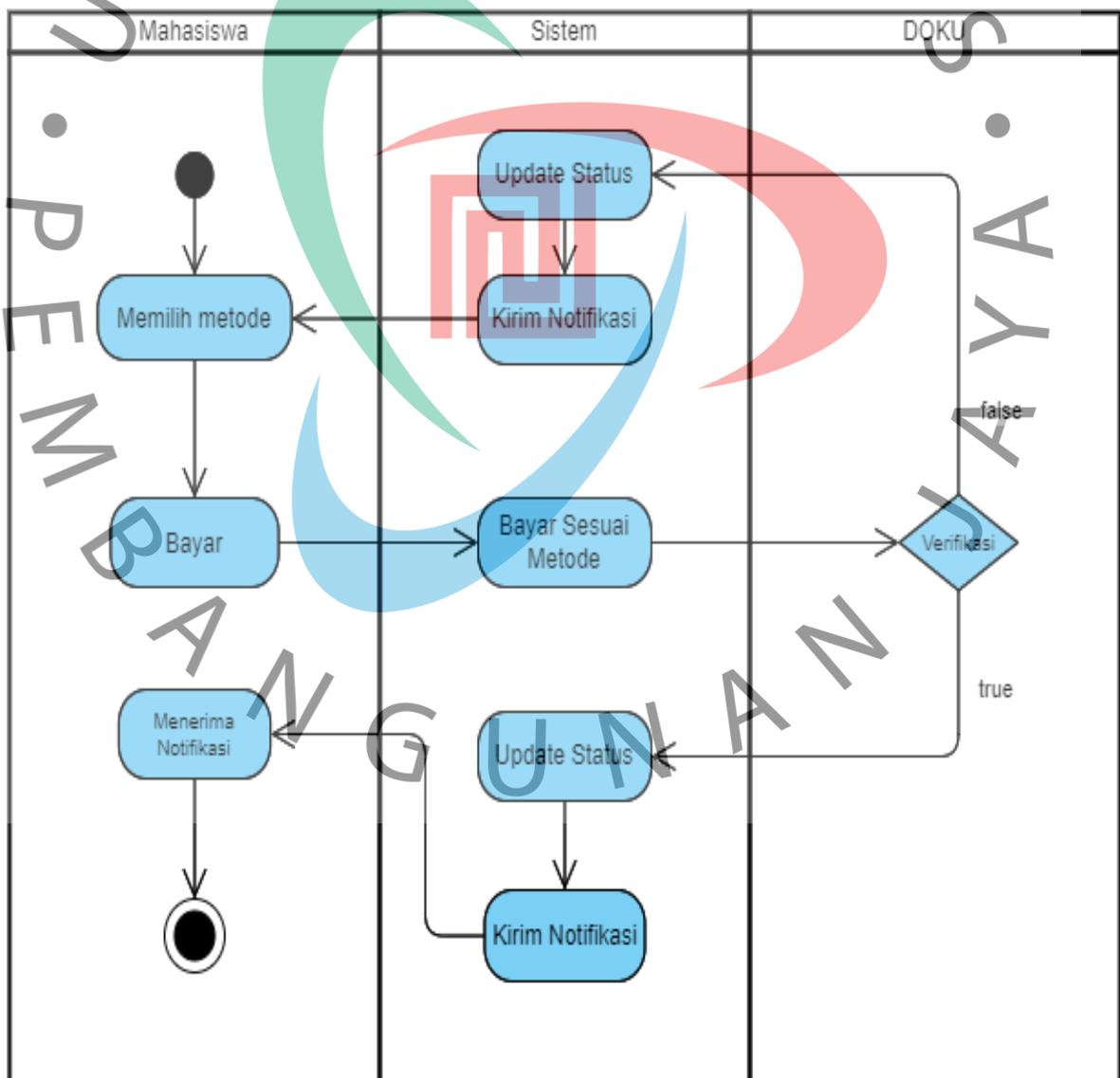
Use Case Name: Laporan Transaksi	ID: SIUK - 02	Priority: Medium
Description: Use-case ini menjelaskan proses pembuatan laporan transaksi. Laporan ini menyediakan ringkasan transaksi selama periode tertentu, termasuk rincian seperti ID transaksi, tanggal, jumlah, dan jenis transaksi.		
Actor: <ul style="list-style-type: none"> - Admin - Manajer - Sistem 		
Trigger:		
Pre-conditions: <ul style="list-style-type: none"> - Pengguna telah berhasil login ke dalam sistem. - Pengguna memiliki akses ke modul laporan transaksi. 		
Post-conditions: <ol style="list-style-type: none"> 1. Laporan transaksi telah berhasil dihasilkan dan ditampilkan kepada pengguna sesuai dengan rentang waktu dan filter yang ditentukan. 2. Data yang ditampilkan dalam laporan transaksi mencerminkan informasi yang akurat dan terbaru dari database. 3. Jika fitur ekspor diaktifkan, laporan transaksi dapat diunduh dalam format excel. 4. Sistem mencatat aktivitas pembuatan laporan transaksi dalam log audit untuk tujuan pemantauan dan keamanan, termasuk waktu akses dan pengguna yang menghasilkan laporan. 		
Normal Case:	Admin dan Manajer	System
	1. Mengakses Sistem	
	2. Mengakses menu Laporan Transaksi	
		3. Menampilkan data transaksi
	2. Export data transaksi	
	3. Export data transaksi dengan bentuk excel	

	4. Filter data transaksi	
		5. Menampilkan data transaksi sesuai dengan filter

Exceptions:

1. Apabila user tidak memiliki hak akses atau izin yang diperlukan, maka akan dialihkan ke halaman login dan menghapus *session*.
2. Jika pengguna memasukkan rentang tanggal yang tidak valid atau tidak masuk akal, sistem akan menampilkan pesan kesalahan "Rentang tanggal tidak valid. Mohon masukkan rentang tanggal yang benar."

4.2.3 Activity Diagram



Gambar 4.3 Activity Diagram Pembayaran

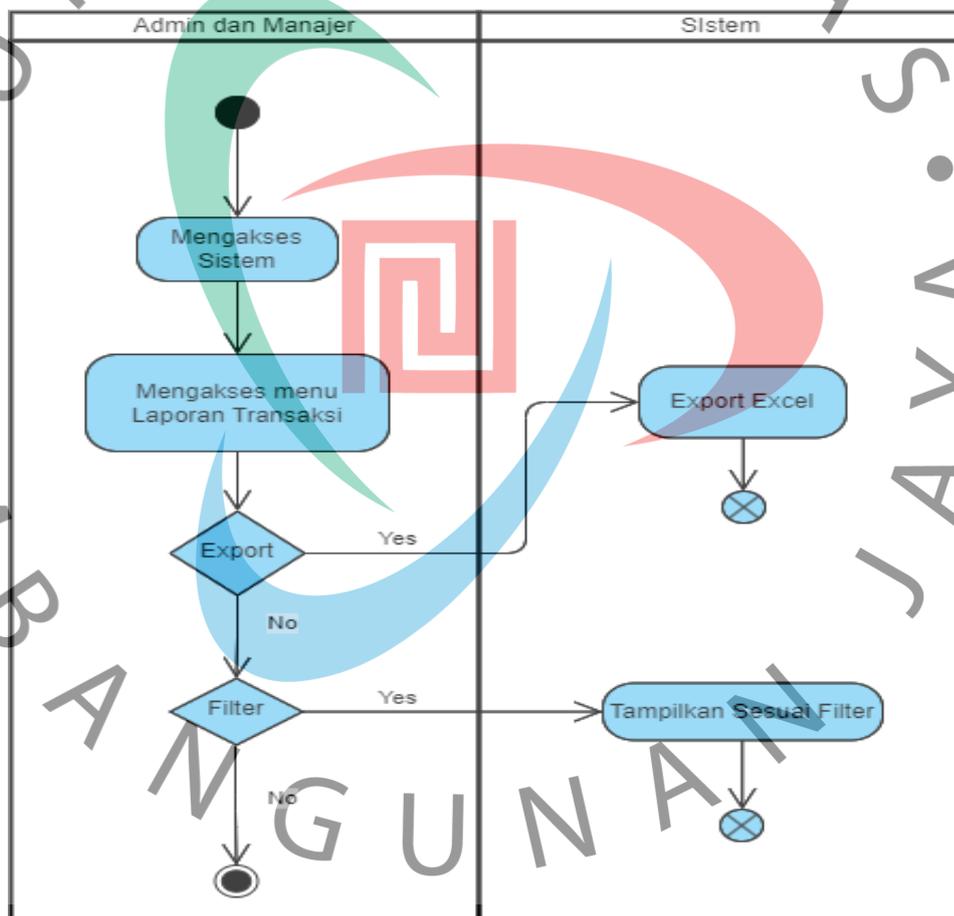
Pada Gambar diatas, dijelaskan tentang serangkaian aktivitas yang terjadi dalam proses pembayaran yang terintegrasi dalam sistem usulan yang akan dikembangkan. Proses ini dimulai ketika pengguna atau peserta telah menyelesaikan semua langkah-langkah untuk melengkapi bukti kelengkapan yang menjadi syarat untuk mengikuti ujian. Setelah semua dokumen dan persyaratan terpenuhi, pengguna dapat melanjutkan ke tahap pembayaran dengan mengklik tombol bayar yang tersedia di antarmuka pengguna. Setelah tombol bayar diklik, sistem akan menampilkan *pop-up* pembayaran yang dirancang untuk memudahkan pengguna dalam memilih metode pembayaran yang diinginkan. Pengguna dapat memilih berbagai metode pembayaran yang telah disediakan oleh sistem, seperti transfer bank atau *e-wallet*. Setiap metode pembayaran akan menampilkan instruksi yang jelas untuk memastikan pengguna dapat melakukan pembayaran dengan benar dan sesuai dengan tagihan yang telah ditentukan.

Setelah pengguna memilih metode pembayaran dan memasukkan informasi yang diperlukan, sistem akan memproses pembayaran tersebut. Proses ini melibatkan penggunaan API (*Application Programming Interface*) untuk melakukan verifikasi pembayaran. SIUK akan mengirimkan permintaan (*request*) ke API milik DOKU, yang merupakan penyedia layanan pembayaran yang terpercaya. Permintaan ini akan mencakup detail transaksi yang diperlukan untuk memverifikasi pembayaran. API DOKU akan menerima permintaan tersebut dan melakukan berbagai pemeriksaan serta validasi untuk memastikan bahwa pembayaran dapat diproses dengan benar. Setelah pemeriksaan selesai, API DOKU akan mengirimkan *response* kembali ke sistem SIUK. *Response* ini akan berisi informasi tentang status transaksi, apakah pembayaran berhasil atau gagal, beserta detail lainnya yang relevan.

Sistem SIUK kemudian akan membaca *response* yang diterima dari API DOKU. Jika transaksi berhasil, sistem akan mengupdate status pembayaran pada *database* untuk mencatat bahwa pembayaran telah dilakukan. Pembaruan status ini penting untuk memastikan bahwa data pengguna selalu akurat dan terkini. Selain itu, sistem akan memberikan pesan notifikasi kepada pengguna yang berisi hasil transaksi tersebut. Notifikasi ini dapat berupa email, pesan teks, atau notifikasi langsung di dalam sistem, yang menginformasikan pengguna tentang keberhasilan atau kegagalan pembayaran. Proses ini tidak hanya memastikan bahwa pembayaran dilakukan dengan aman dan efisien, tetapi juga memberikan pengalaman yang baik

dan transparan bagi pengguna. Dengan integrasi API DOKU, sistem SIUK dapat menangani berbagai metode pembayaran dan memastikan bahwa setiap transaksi diverifikasi dengan cepat dan akurat. Selain itu, pengguna dapat dengan mudah melacak status pembayaran mereka dan menerima konfirmasi segera setelah transaksi selesai.

Dengan demikian, Gambar 4.3 menggambarkan keseluruhan proses pembayaran yang dirancang untuk memberikan kemudahan, keamanan, dan keandalan bagi pengguna dalam menyelesaikan pembayaran untuk mengikuti ujian atau layanan lainnya yang disediakan oleh sistem SIUK. Integrasi ini merupakan bagian penting dari upaya untuk meningkatkan pengalaman pengguna dan memastikan bahwa semua proses administrasi berjalan dengan lancar dan efisien.



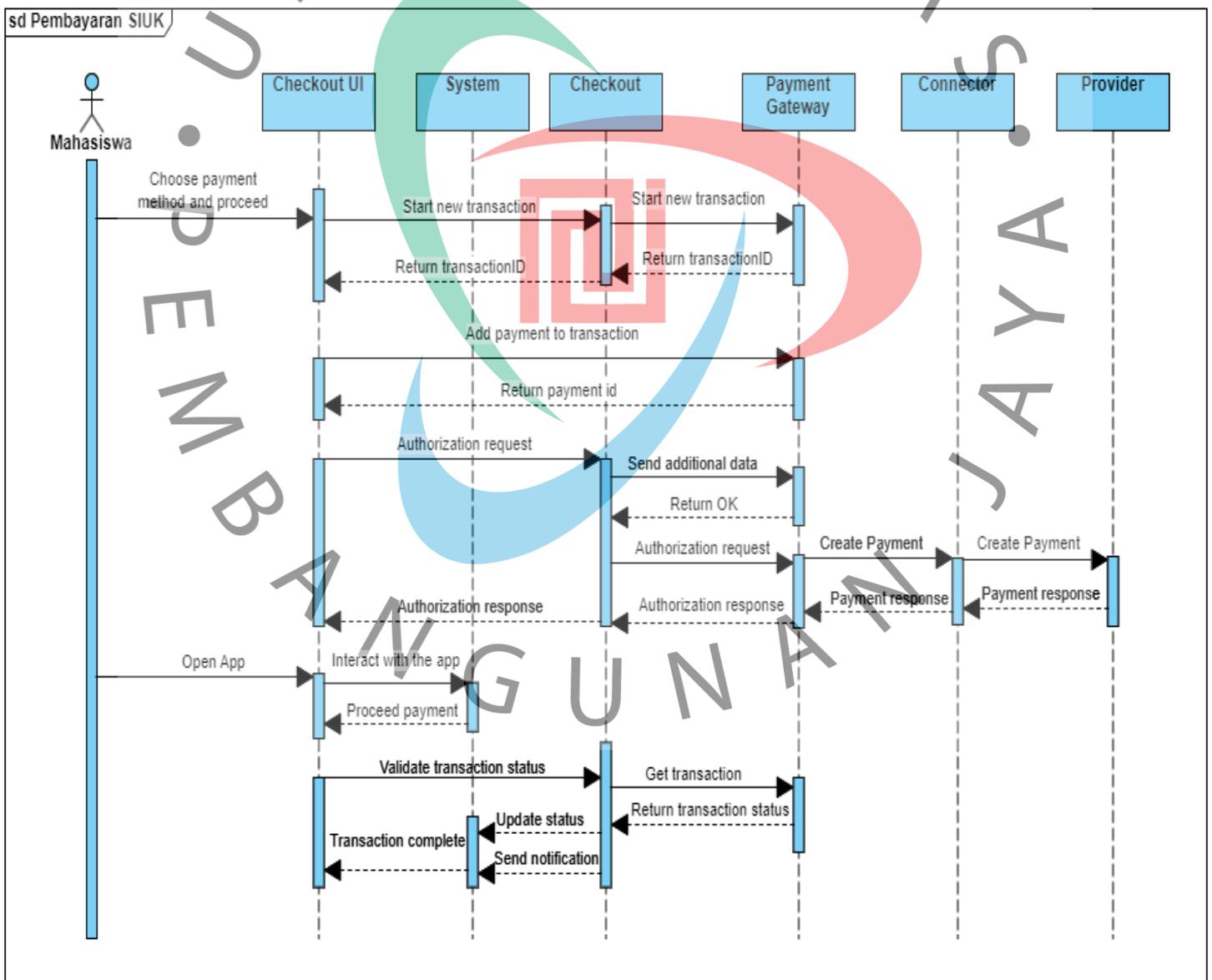
Gambar 4.4 Activity Diagram Laporan Transaksi

Gambar di atas menjelaskan tentang serangkaian aktivitas laporan transaksi. Laporan transaksi dapat dilakukan oleh admin dan manajer, yang memiliki akses eksklusif ke fitur ini. Laporan transaksi hanya dapat diakses oleh admin dan manajer. Selain itu, halaman laporan transaksi juga dilengkapi dengan opsi untuk

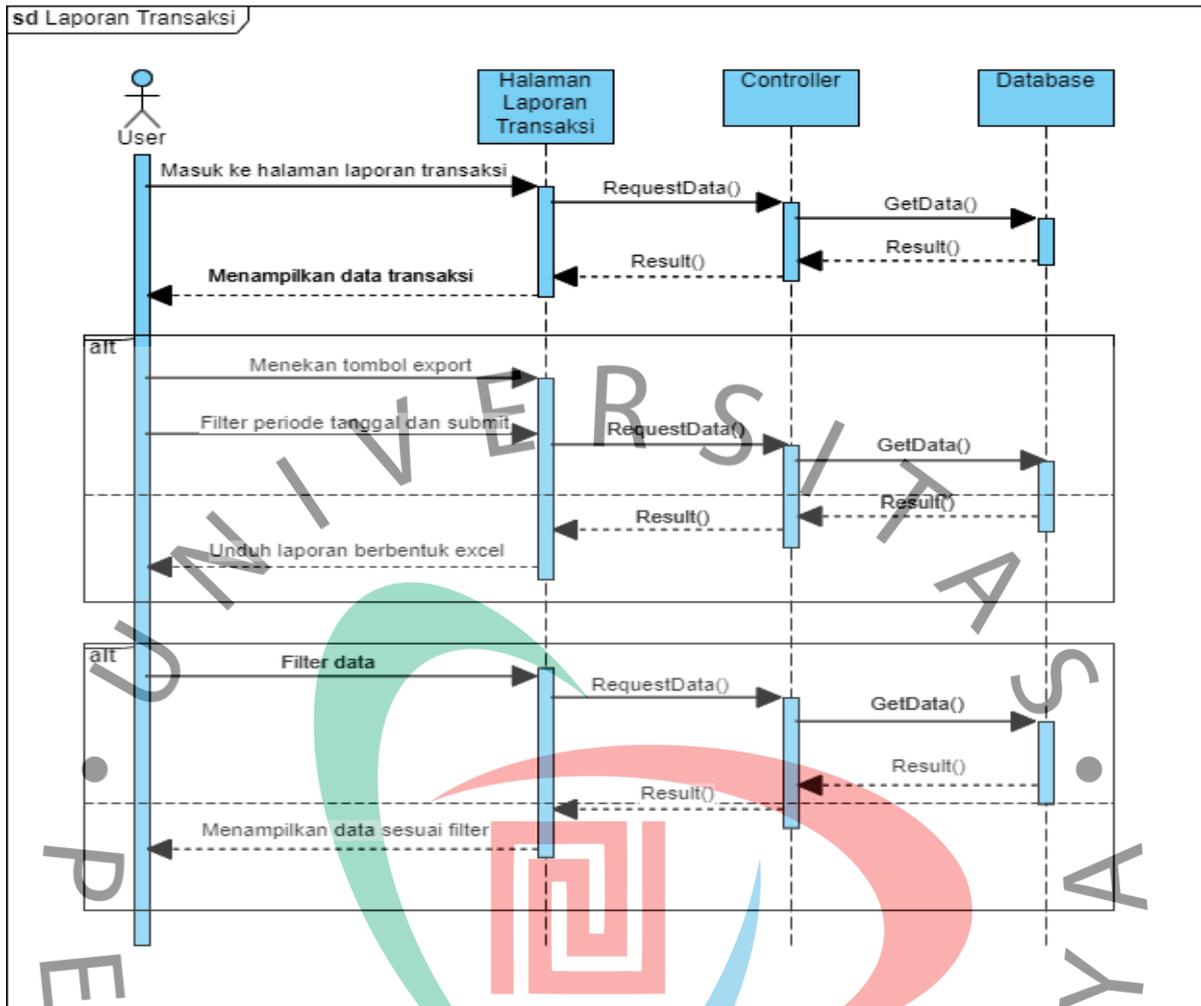
mengekspor data. Admin dan manajer dapat mengekspor laporan dalam format Excel. Fitur ini dirancang untuk memberikan fleksibilitas dan kemudahan bagi admin dan manajer dalam mengelola dan memantau transaksi.

Dengan kemampuan untuk memfilter periode dan mengekspor laporan, mereka dapat memastikan bahwa semua transaksi tercatat dengan baik dan dapat diakses dengan mudah untuk keperluan analisis dan pelaporan. Dengan demikian, Gambar di atas tidak hanya menggambarkan aktivitas laporan transaksi, tetapi juga menekankan pentingnya akses filter periode, dan kemampuan ekspor untuk mendukung manajemen keuangan yang efektif dan efisien dalam sistem.

4.2.4 Sequence Diagram

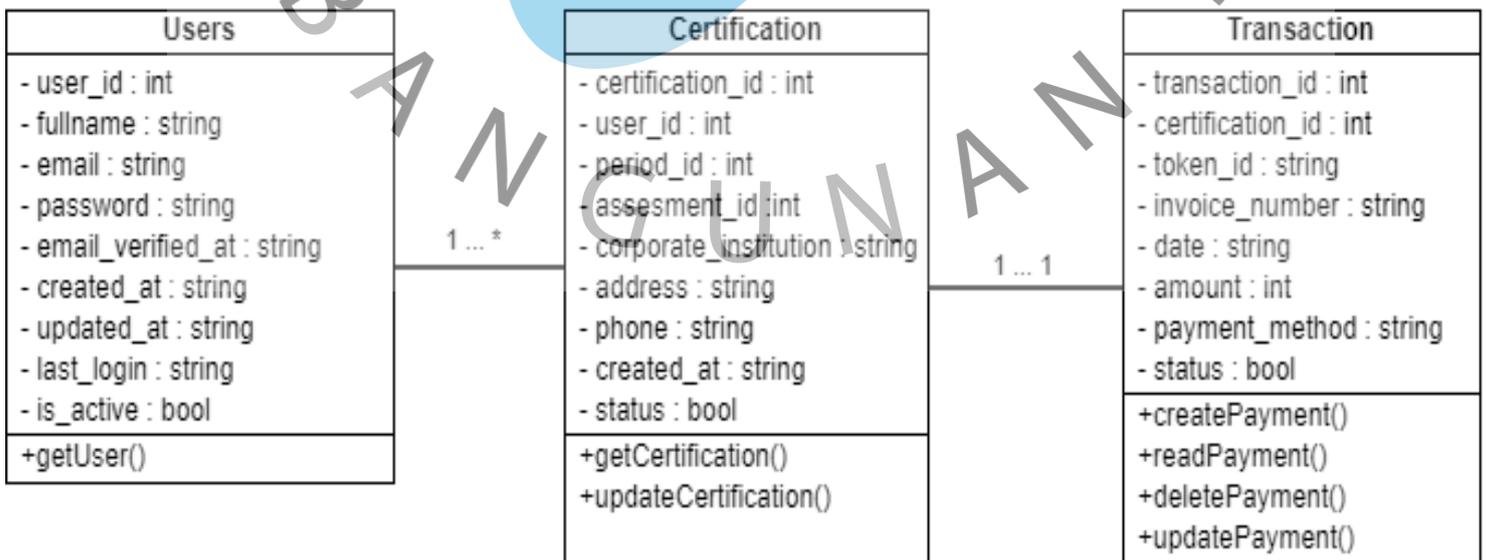


Gambar 4.5 Sequence Diagram Pembayaran



Gambar 4.6 Sequence Diagram Laporan Transaksi

4.2.5 Class Diagram



Gambar 4.7 Class Diagram Pembayaran SIUK

4.2.6 Spesifikasi Tabel

Berdasarkan dari *class diagram* yang telah digambarkan sebelumnya, pada bagian spesifikasi tabel ini bertujuan untuk merinci setiap *attribute* yang ada dalam tabel. Berikut adalah rincian dari masing-masing tabel beserta tipe datanya:

1. Tabel **users**

Nama : **users**
 Deskripsi : tabel yang digunakan untuk login ke dalam SIUK
 Primary Key : *user_id*
 Foreign Key :-

Tabel 4.4 Spesifikasi Tabel Pengguna

Nama Field	Type	Length	Keterangan
<i>user_id</i>	<i>int</i>	11	Primary Key
<i>fullname</i>	<i>varchar</i>	64	Nama lengkap mahasiswa
<i>email</i>	<i>varchar</i>	128	Email mahasiswa
<i>password</i>	<i>varchar</i>	255	Password
<i>email_verified_at</i>	<i>timestamp</i>	-	Tanggal email diverifikasi
<i>created_at</i>	<i>timestamp</i>	-	Tanggal dibuatnya data
<i>updated_at</i>	<i>timestamp</i>	-	Tanggal diubahnya data
<i>last_login</i>	<i>timestamp</i>	-	Terakhir login
<i>is_active</i>	<i>enum (Y,N)</i>		Y = Aktif, N = Tidak Aktif

2. Tabel **certification**

Nama : **certification**
 Deskripsi : Tabel untuk pendaftaran ujian sertifikasi
 Primary Key : *certification_id*
 Foreign Key : *user_id, period_id, assesment_id*

Tabel 4.5 Spesifikasi Tabel Sertifikasi

Nama Field	Type	Length	Keterangan
<i>certification_id</i>	<i>int</i>	11	Primary Key

user_id	<i>int</i>	11	Foreign Key
period_id	<i>int</i>	11	Foreign Key
assesment_id	<i>int</i>	11	Foreign Key
corporate_institution	<i>varchar</i>	255	Lembaga Perusahaan
address	<i>tinytext</i>	-	Alamat perusahaan
phone	<i>char</i>	15	No Telp Perusahaan
created_at	<i>timestamp</i>	-	Tanggal dibuatnya data
status	<i>enum</i>	-	Status kegiatan sertifikasi

3. Tabel **transaction**

Nama : **transaction**

Deskripsi : Tabel untuk menyimpan data transaksi

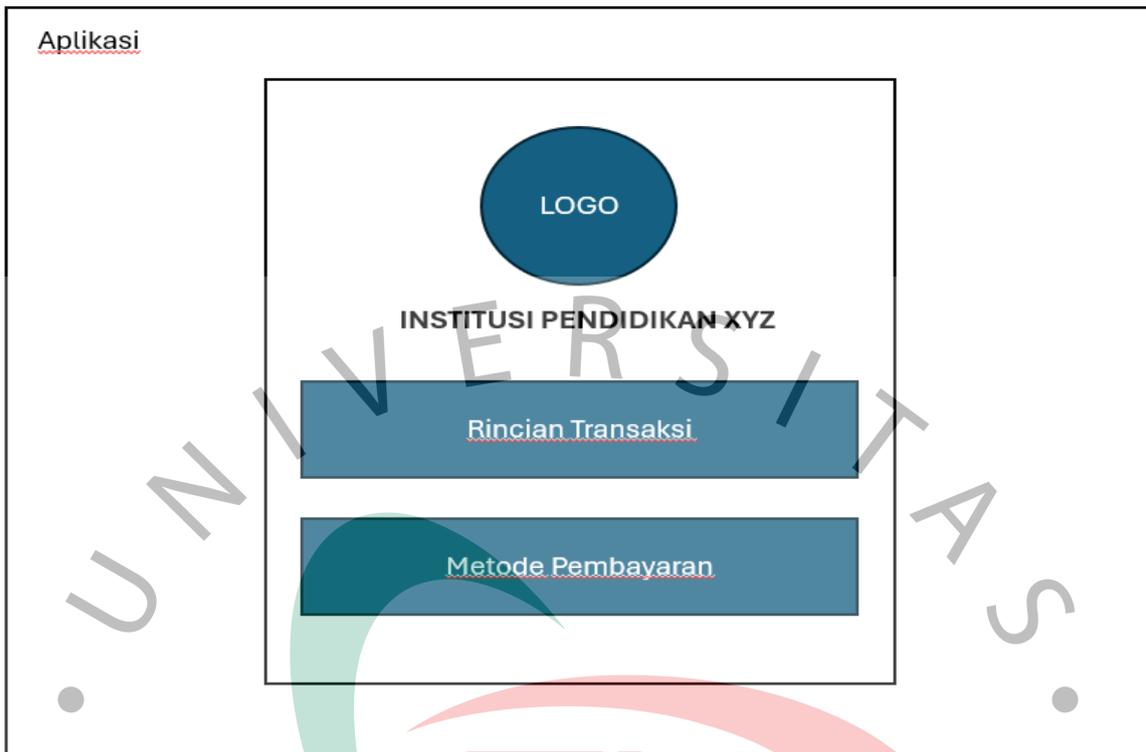
Primary Key : transaction_id

Foreign Key : certification_id

Tabel 4.6 Spesifikasi Tabel Transaksi

Nama Field	Type	Length	Keterangan
transaction_id	<i>int</i>	11	Primary Key
certification_id	<i>int</i>	11	Foreign Key
token_id	<i>varchar</i>	128	Token yang didapatkan dari <i>request</i> API
invoice_number	<i>varchar</i>	64	Nomor Invoice
date	<i>datetime</i>	-	Tanggal transaksi
amount	<i>int</i>	11	Total harga
payment_method	<i>varchar</i>	128	Metode pembayaran
status	<i>enum</i> (<i>pending</i> , <i>success</i> , <i>failed</i>)	-	Status Pembayaran

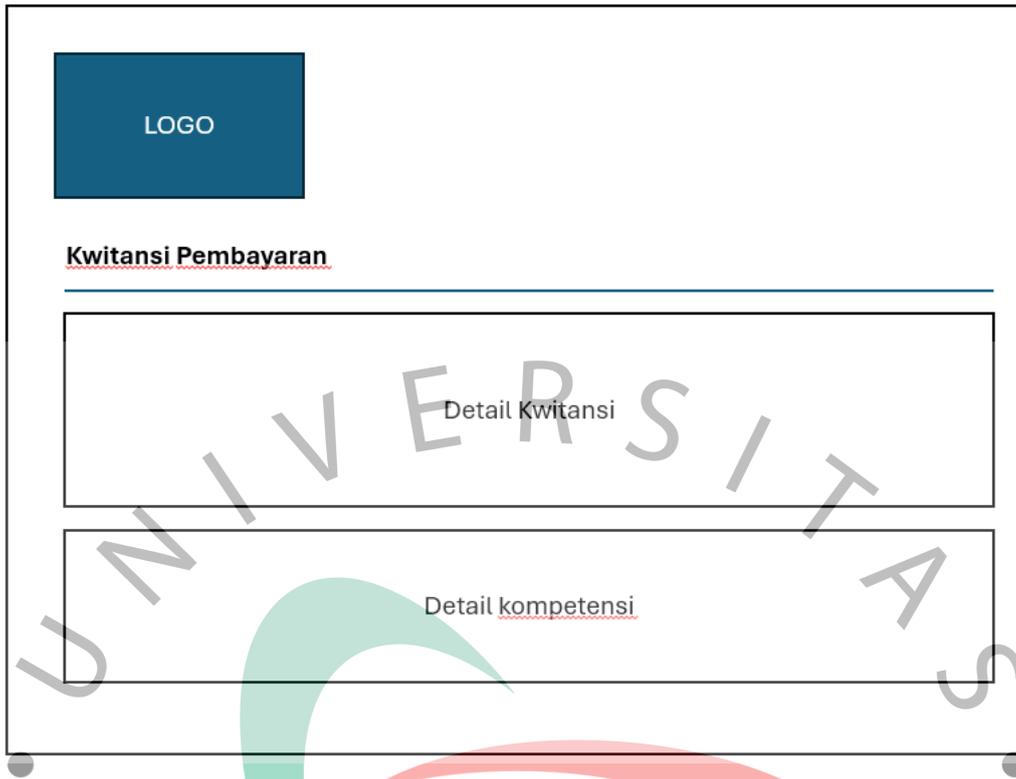
4.3 Perancangan Antarmuka Pengguna



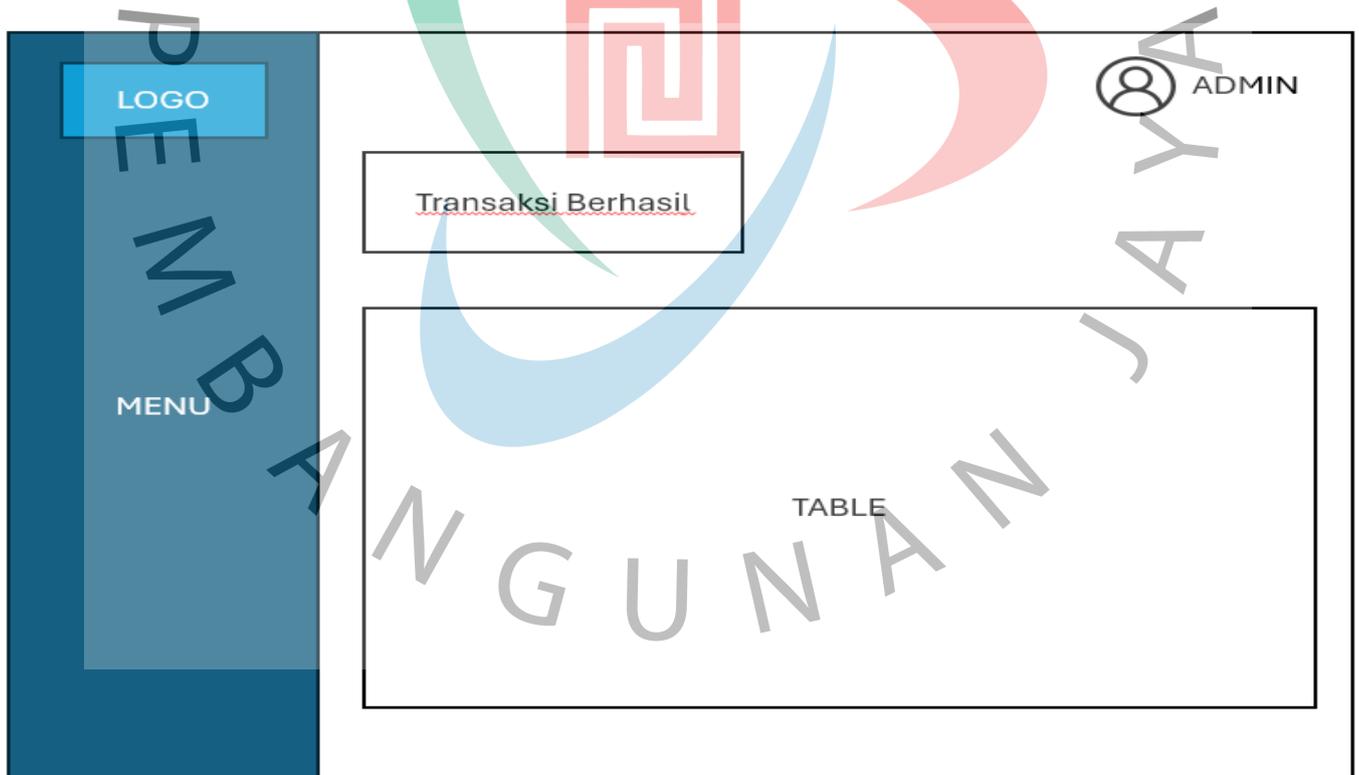
Gambar 4.8 Mockup Modal Checkout



Gambar 4.9 Mockup Modal Bukti Pembayaran



Gambar 4.10 Mockup Contoh Kwitansi Pembayaran



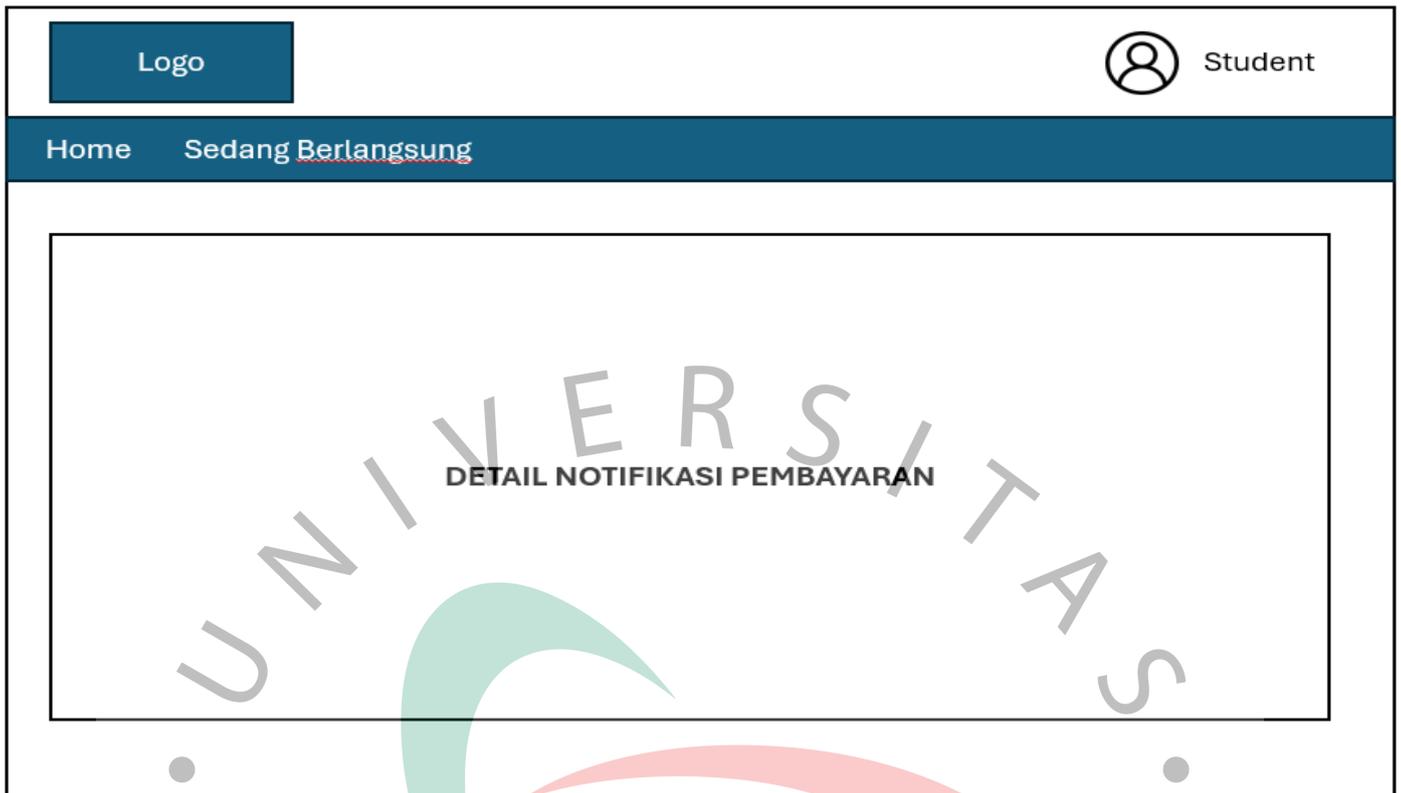
Gambar 4.11 Mockup Laporan Transaksi



Gambar 4.12 Mockup Modal Export Laporan Transaksi



Gambar 4.13 Mockup Detail Transaksi

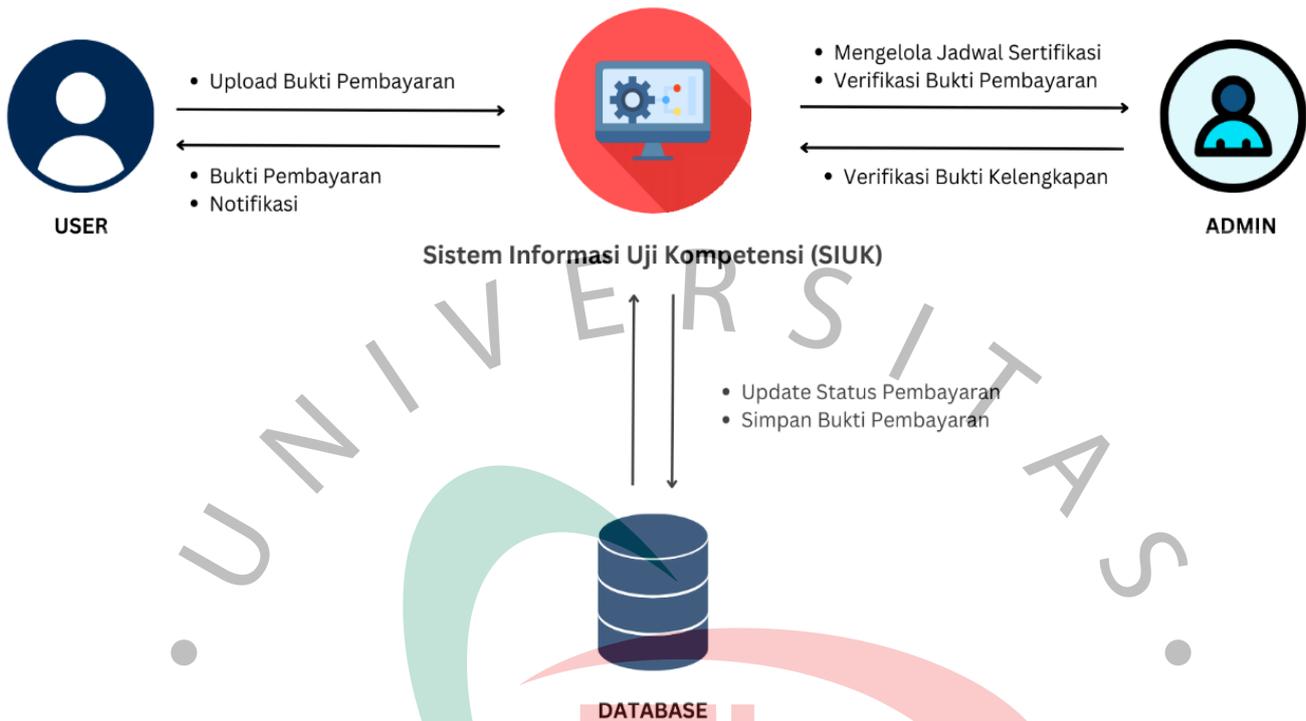


Gambar 4.14 Mockup Detail Notifikasi Pembayaran

4.4 Perancangan Implementasi

Pada tahapan ini, akan diuraikan dengan detail bagaimana rancangan yang telah disiapkan sebelumnya diimplementasikan. Peneliti memanfaatkan beberapa perangkat lunak (*software*) untuk mendukung dan membantu dalam menyelesaikan permasalahan yang terdapat dalam penelitian ini. Hasil analisis dan desain diterapkan menggunakan PHP dengan dukungan *framework* Codeigniter 3 melalui aplikasi kode *editor* Visual Studio Code. Sebagai sistem basis data, dipilih DBMS MySQL, sementara pengujian sistem dilakukan dengan metode *Black Box*. Selain itu, untuk memastikan kehandalan sistem, digunakan pula Postman sebagai alat untuk menguji dan memvalidasi fungsi API yang terintegrasi dalam aplikasi. Implementasi ini mengikuti desain sistem yang telah disusun sebelumnya, yang mengintegrasikan teknologi-teknologi tersebut untuk mencapai tujuan pengembangan SIUK dengan menggunakan *Payment Gateway* menggunakan DOKU. Setiap langkah implementasi dilakukan dengan teliti untuk memastikan bahwa sistem dapat beroperasi secara efisien, aman dari risiko keamanan, serta memberikan pengalaman pengguna yang optimal dalam hal pembayaran ujian sertifikasi BNSP.

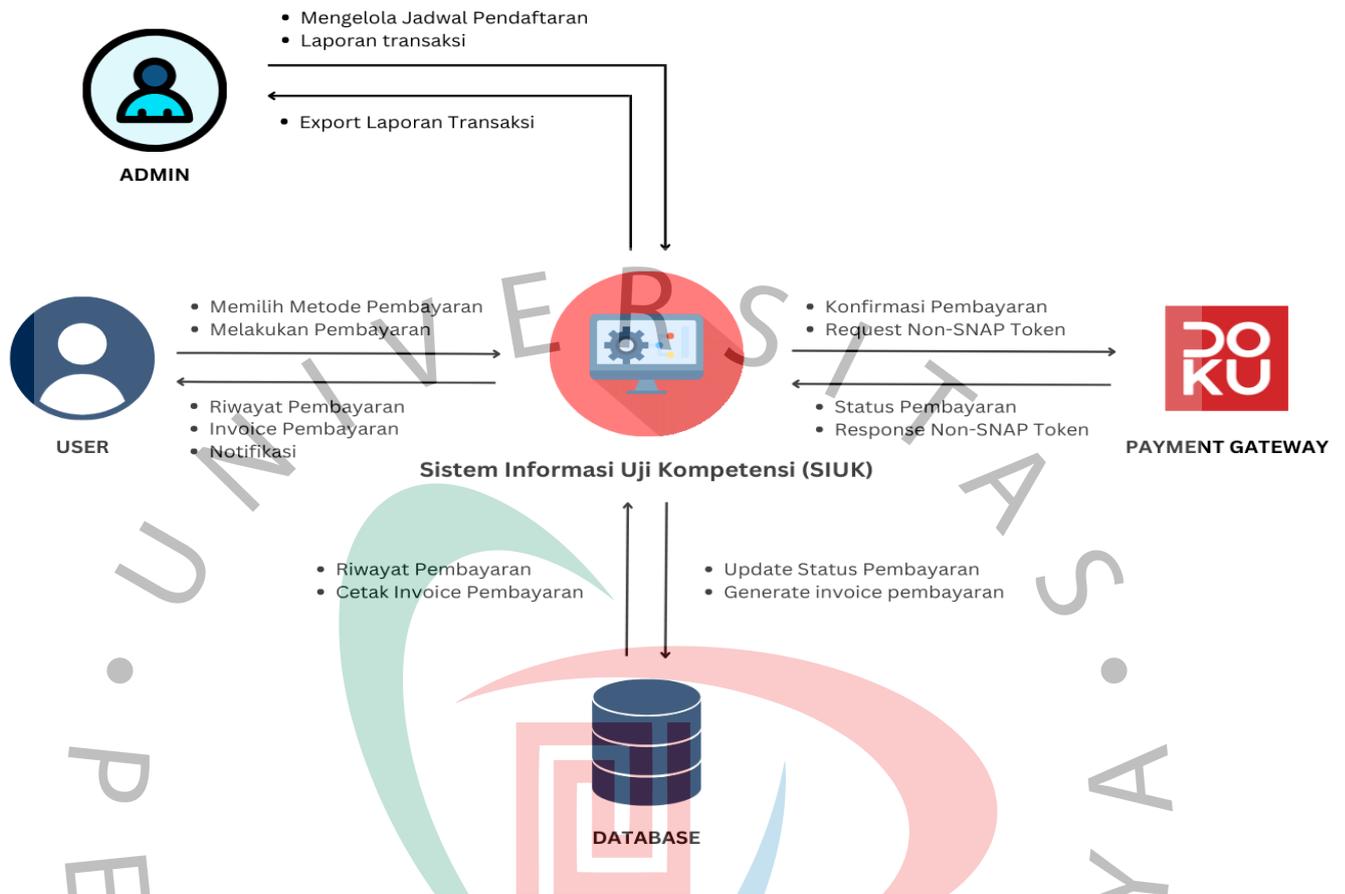
4.4.1 Arsitektur Sistem Berjalan



Gambar 4.15 Arsitektur Sistem Berjalan

Gambar di atas menunjukkan gambaran tentang arsitektur sistem yang sedang berjalan, di mana metode pembayaran dilakukan dengan cara upload bukti pembayaran melalui sistem, kemudian admin memverifikasi bukti pembayaran tersebut. Tentu saja, hal ini menyebabkan proses administrasi menjadi lambat karena memerlukan campur tangan manual dari admin untuk melakukan verifikasi setiap kali ada transaksi pembayaran. Selain itu, metode ini juga berpotensi menimbulkan masalah keamanan dan kebijakan, karena ada kemungkinan adanya manipulasi atau pemalsuan bukti pembayaran oleh pengguna yang tidak bertanggung jawab. Selain itu, proses verifikasi manual oleh admin juga dapat menyebabkan kesalahan atau ketidakkonsistenan dalam penanganan transaksi, sehingga dapat menimbulkan ketidakpuasan pelanggan dan kerugian bagi pihak *merchant*. Oleh karena itu, dibutuhkan solusi yang lebih efisien, aman, dan terintegrasi untuk menangani proses pembayaran, misalnya dengan memanfaatkan sistem *Payment Gateway* yang dapat melakukan verifikasi dan otomasi transaksi secara *real-time*.

4.4.2 Arsitektur Sistem Usulan



Gambar 4.16 Arsitektur Sistem Usulan

Gambar diatas menjelaskan tentang arsitektur sistem yang akan dikembangkan, sistem yang dikembangkan adalah ketika peserta telah berhasil mendaftarkan diri kedalam sistem kemudian memilih kompetensi dan melakukan pembayaran dengan *Payment Gateway*. Sistem pembayaran yang dikembangkan nantinya akan menggunakan metode transfer bank serta pembayaran di minimarket seperti Alfamart, Alfamidi, dan Lawson, sehingga memberikan fleksibilitas dan kemudahan bagi peserta dalam melakukan transaksi.

Payment Gateway yang digunakan dalam pengembangan sistem adalah DOKU. DOKU dipilih karena menyediakan berbagai layanan API yang dapat diintegrasikan dengan sistem untuk memproses pembayaran secara efisien dan aman. Dengan API yang disediakan oleh DOKU, sistem dapat mengotomatiskan proses verifikasi pembayaran, mengirim notifikasi pembayaran sukses kepada peserta, dan mencatat transaksi secara *real-time* dalam *database*.

Penggunaan DOKU juga membawa manfaat dari sisi keamanan dan keandalan transaksi, karena DOKU telah memiliki sertifikasi keamanan yang memenuhi standar industri. Integrasi ini akan memastikan bahwa setiap transaksi diproses dengan enkripsi yang kuat, sehingga data sensitif peserta terlindungi dengan baik. Selain itu, sistem ini akan mencakup fitur untuk memantau dan mencatat semua aktivitas pembayaran, memberikan laporan transaksi yang rinci, dan memungkinkan tim administrasi untuk melacak status setiap pembayaran dengan mudah. Dengan demikian, pengembangan sistem ini diharapkan dapat memberikan pengalaman yang lebih baik bagi peserta dan memastikan kelancaran dalam proses pendaftaran dan pembayaran kompetensi.

4.4.3 Integrasi *Payment Gateway* DOKU

A. DOKU

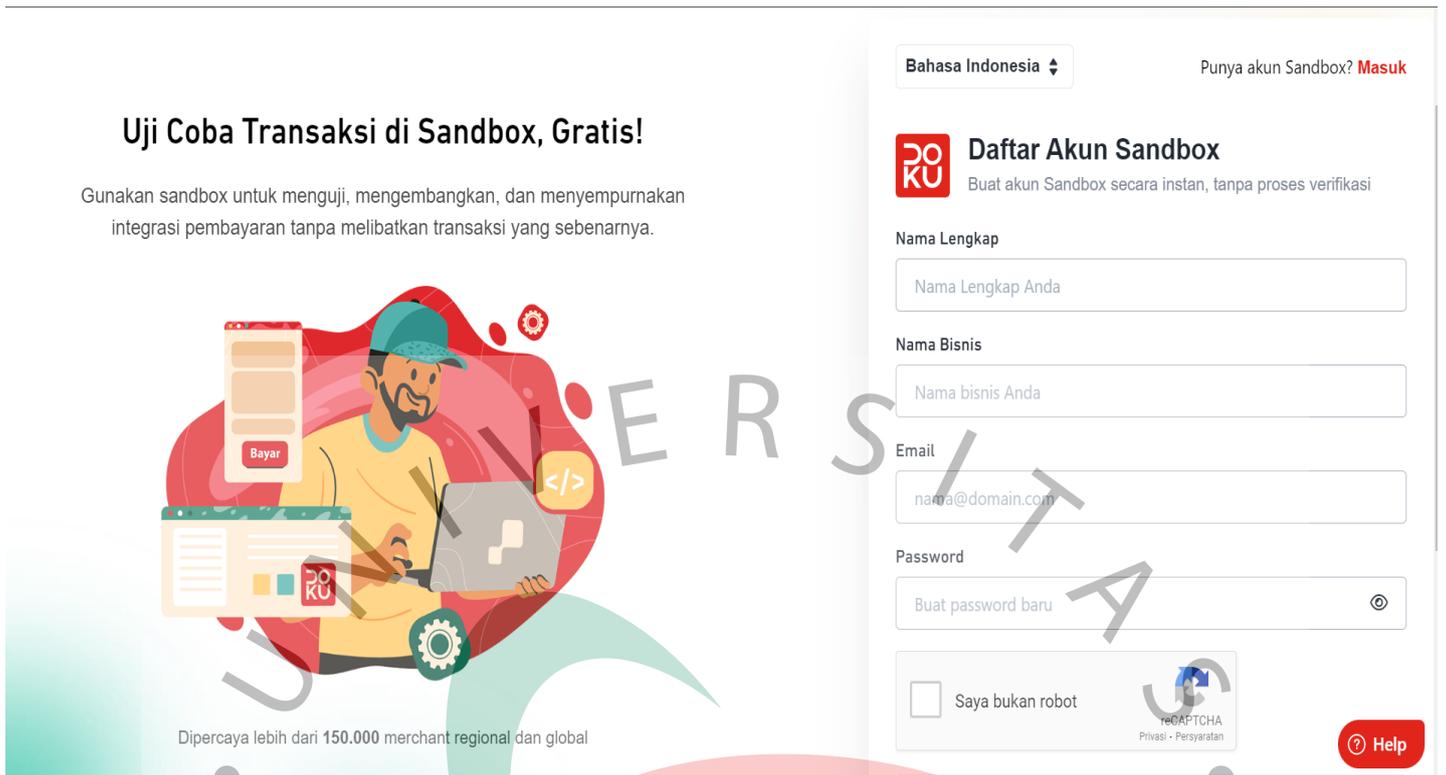
DOKU merupakan perusahaan *fintech* di bidang pembayaran yang telah terbukti memberikan layanan berkualitas tinggi dan memiliki lisensi resmi. Perusahaan ini berkomitmen untuk menyediakan solusi pembayaran yang efisien dan andal, sehingga mendapatkan kepercayaan dari berbagai kalangan pelanggan dan mitra bisnis.



Gambar 4.17 Logo DOKU

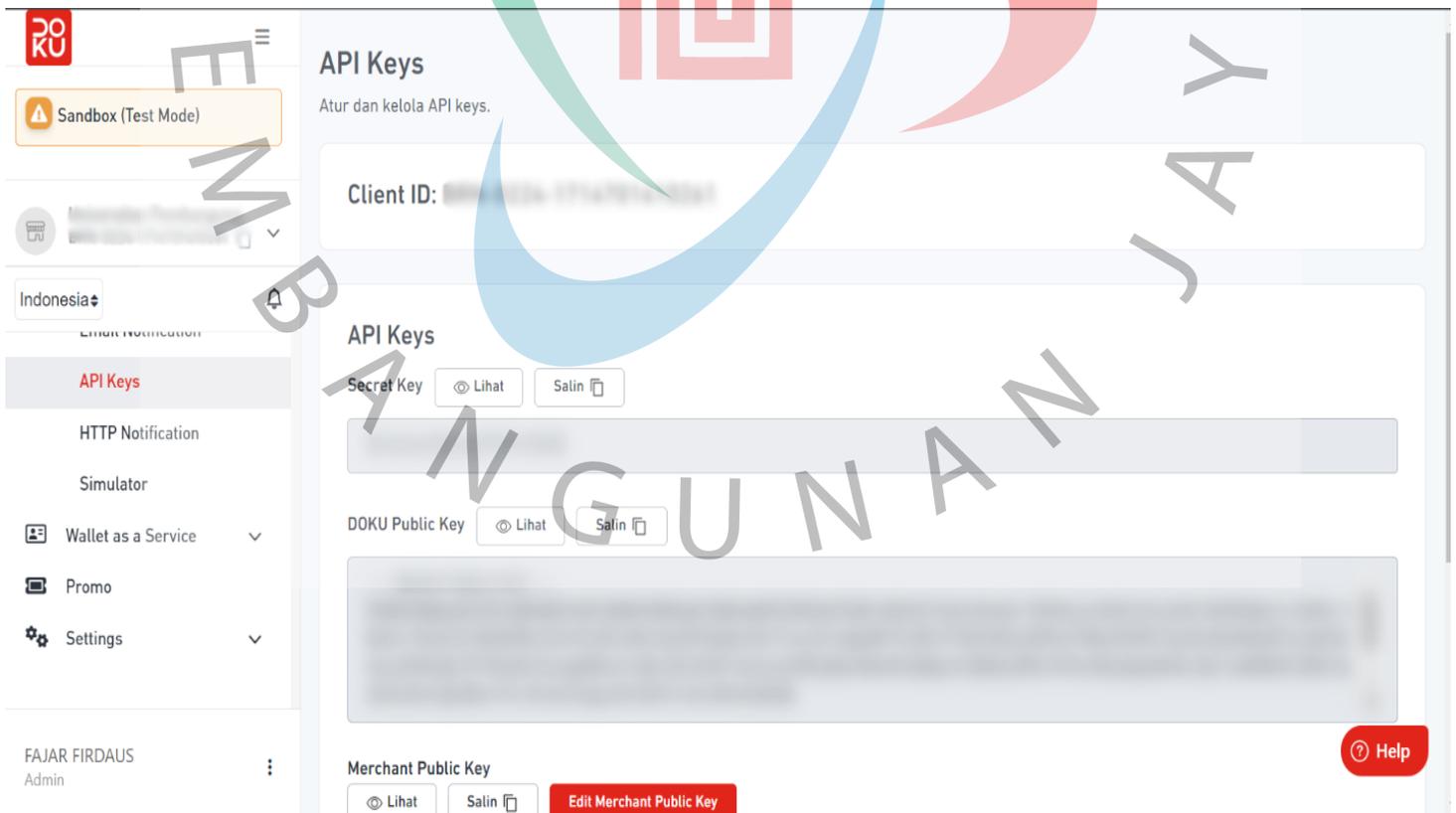
B. Pendaftaran DOKU

Sebelum mengintegrasikan sistem ke *Payment Gateway*, terlebih dahulu harus mendaftar sebagai *Merchant* guna mendapatkan *Client-ID* dan *API Keys*. Pendaftaran ini penting untuk mendapatkan kredensial yang diperlukan agar sistem dapat berkomunikasi dengan *Payment Gateway* secara aman dan efektif. Berikut adalah halaman pendaftaran sebagai *Merchant* pada DOKU:



Gambar 4.18 Halaman Pendaftaran DOKU

C. API Keys



Gambar 4.19 Halaman DOKU API Key

D. Accepting Payments

Dalam *Payment Gateway* DOKU terdapat beberapa cara untuk menerima pembayaran dengan mudah dengan berbagai pilihan, berikut adalah beberapa pilihan yang telah disediakan oleh DOKU:

- ***Built-in Payment Page (Checkout)***

Built-in Payment Page (Checkout) adalah halaman pembayaran yang disediakan dan dikelola oleh DOKU, sehingga *merchant* tidak perlu membuat atau menyediakan halaman pembayaran sendiri. DOKU telah mengurus semua aspek tampilan dan keamanan halaman pembayaran tersebut.

- ***Customize Payment Page (Direct API)***

Customize Payment Page (Direct API) adalah solusi pembayaran yang memungkinkan *merchant* untuk sepenuhnya mengendalikan dan menyesuaikan tampilan serta pengalaman halaman pembayaran sesuai kebutuhan mereka.

Dalam penelitian ini peneliti menggunakan metode *Built-in Payment Page (Checkout)* sehingga peneliti tidak perlu menyiapkan halaman pembayaran karena sudah disediakan oleh DOKU itu sendiri.

E. Direct API

Direct API memungkinkan *merchant* untuk langsung terhubung dengan sistem pembayaran DOKU melalui API. Dengan menggunakan *Direct API*, *merchant* dapat sepenuhnya mengendalikan dan menyesuaikan proses pembayaran di situs web atau aplikasi *merchant* tanpa harus mengarahkan pelanggan ke halaman pembayaran yang disediakan oleh DOKU. Berikut adalah *channel* pembayaran yang tersedia:

- *Virtual Account*
- *Online to Offline (O2O)*
- *Credit Card*
- *E-Money*
- *Direct Debit*

4.4.4 *Generate Signature and Validate Signature*

Signature atau tanda tangan merupakan salah satu aspek keamanan yang ada dalam sistem *Payment Gateway*. *Signature* ini dijadikan sebagai parameter keamanan yang perlu dibuat dan diterapkan oleh *merchant* atau pedagang yang dimana nantinya digunakan untuk memverifikasi keaslian permintaan atau *Request API* yang dilakukan oleh *merchant*. Berikut adalah cara bagaimana *Signature* itu bekerja:

1. Ketika *merchant* melakukan permintaan ke API DOKU
 - i. *Merchant* membuat *signature* dan meletakkannya pada *request header*: *Merchant* harus membuat *signature* pada *request header* yang nantinya akan dicek oleh DOKU untuk memverifikasi keasliannya.
 - ii. *Merchant* memvalidasi *Signature* dari *response header*: DOKU harus membuat *signature* pada *response header* kemudian *merchant* akan memverifikasi keasliannya.
2. Ketika DOKU berhasil memverifikasi keaslian *signature* dari *merchant*, *merchant* juga harus memverifikasi *signature* yang dikirimkan oleh DOKU pada *response header* untuk memastikan keaslian bahwa *request* berasal dari DOKU.

4.4.5 *Signature components from Request Header*

Sebelum mengajukan permintaan ke API DOKU, ada beberapa komponen penting yang perlu disiapkan untuk membuat *signature*. Komponen-komponen ini mencakup *Merchant ID*, *Merchant Key*, dan Data Transaksi yang akan dikirim melalui API. *Merchant ID* adalah identitas unik yang diberikan oleh DOKU kepada setiap *merchant* yang telah terdaftar dan bekerjasama. *Merchant Key* merupakan kunci rahasia yang juga diberikan oleh DOKU kepada *merchant* dan digunakan untuk enkripsi serta autentikasi. Sementara itu, Data Transaksi mencakup informasi-informasi penting terkait transaksi yang akan dilakukan, seperti jumlah pembayaran, waktu transaksi, dan detail lainnya yang diperlukan. Semua komponen ini harus dikumpulkan dan disiapkan dengan baik oleh *merchant* sebelum membuat *signature* yang akan digunakan untuk mengamankan komunikasi dan memverifikasi permintaan ke API DOKU. Berikut ini adalah penjelasan komponen yang ada dalam *signature*:

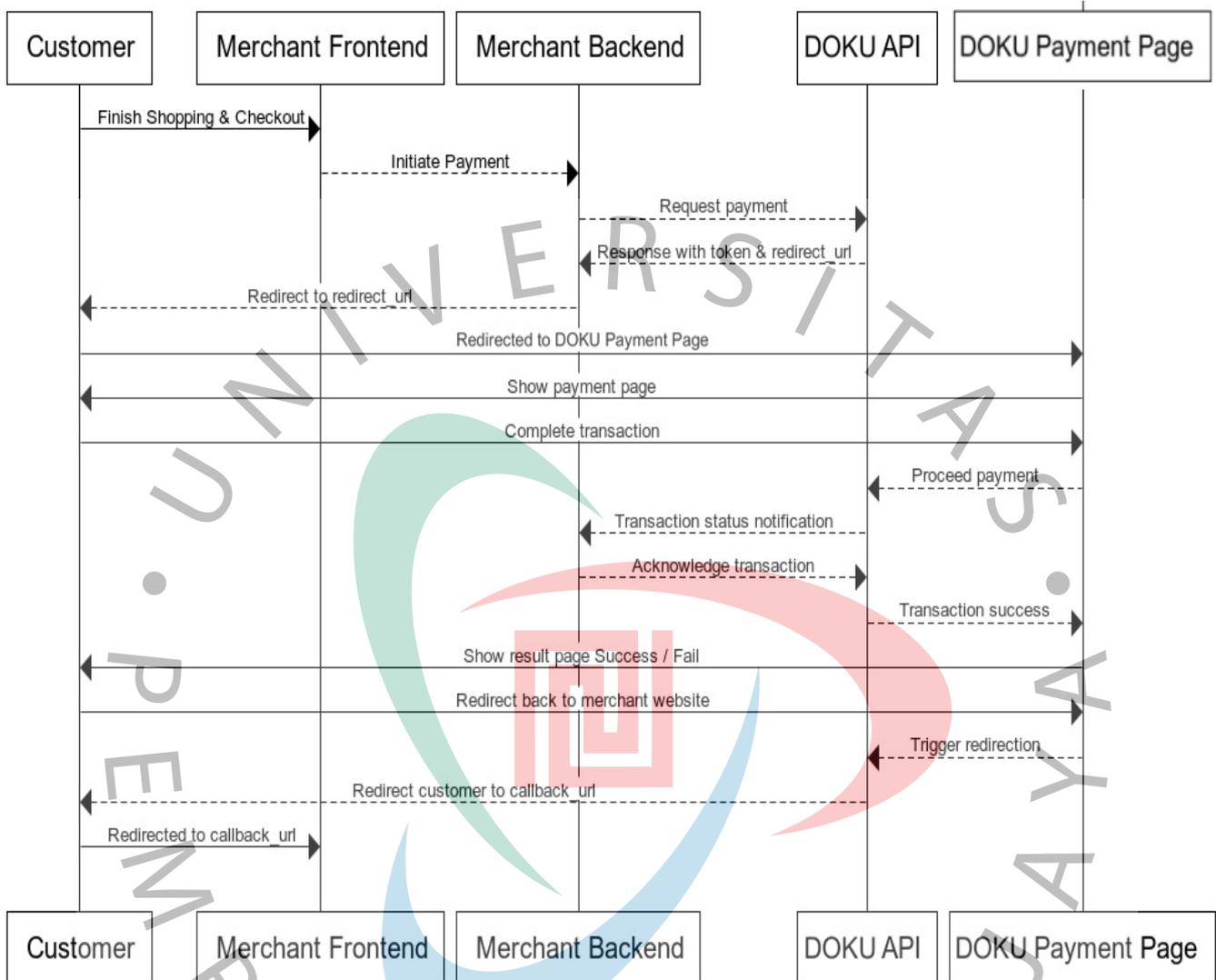
Tabel 4.7 Komponen Signature DOKU API

Nama	Deskripsi
<i>Client-ID</i>	Kode identitas unik yang diperoleh dari DOKU. <i>Client-ID</i> ini digunakan untuk mengidentifikasi <i>merchant</i> yang melakukan permintaan.
<i>Request-ID</i>	Kode yang harus unik yang <i>merchant</i> harus siapkan. Contoh penggunaan <i>request-ID</i> dapat digunakan untuk membuat kode <i>invoice</i> . Kode ini berfungsi sebagai identitas unik dari setiap permintaan yang dilakukan.
<i>Request-Timestamp</i>	Waktu yang digunakan untuk melakukan permintaan adalah waktu saat ini (WIB) dikurangi 7 jam dengan format ISO8601 UTC+0. <i>Timestamp</i> ini memastikan bahwa permintaan dibuat dalam jangka waktu yang valid.
<i>Request-Target</i>	URL API yang akan dieksekusi. <i>Request-Target</i> menentukan endpoint API yang ingin diakses oleh <i>merchant</i> .
<i>Digest</i>	<i>Value</i> yang dienkrpsi dengan base64, dimana isi JSON yang di-hash menggunakan SHA-256. Komponen ini khusus digunakan pada metode <i>POST</i> untuk memastikan integritas data yang dikirimkan.

Untuk membuat *signature*, *merchant* harus menggabungkan komponen-komponen di atas dengan format tertentu dan kemudian melakukan hash menggunakan algoritma SHA-256. Hasil hash ini kemudian dienkrpsi dengan base64 untuk menghasilkan *signature* yang aman. *Signature* ini akan digunakan sebagai bagian dari *header* permintaan ke API DOKU untuk memastikan bahwa permintaan tersebut sah dan tidak dimodifikasi.

4.4.6 Checkout Workflow

DOKU Checkout Flow (Redirect)

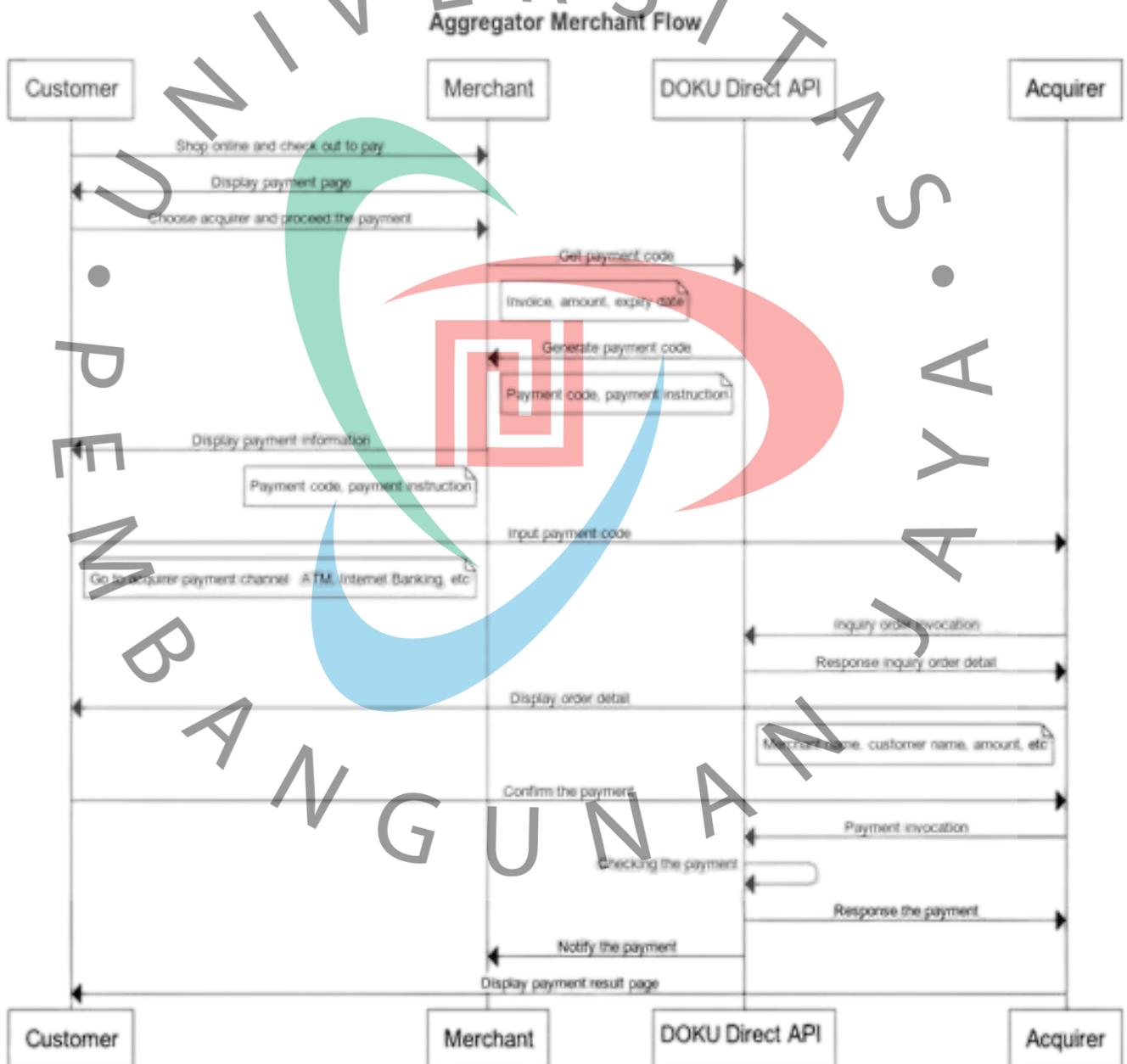


Gambar 4.20 DOKU Checkout Workflow

Gambar di atas menjelaskan cara kerja proses *checkout*. Proses *checkout* dimulai dengan peserta menekan tombol *checkout*, kemudian sistem *backend* mengirimkan permintaan ke API DOKU dan menerima *respons* berupa token yang dilengkapi dengan URL pengalihan. Setelah itu, peserta akan diarahkan ke halaman pembayaran DOKU untuk menyelesaikan transaksi. Setelah pembayaran selesai, *Payment Gateway* akan memverifikasi transaksi dan mengirimkan *respons* kembali ke sistem. Sistem kemudian akan memperbarui status pembayaran dan memberikan notifikasi kepada peserta bahwa pembayaran telah berhasil.

4.4.7 Non-SNAP Virtual Account Workflow

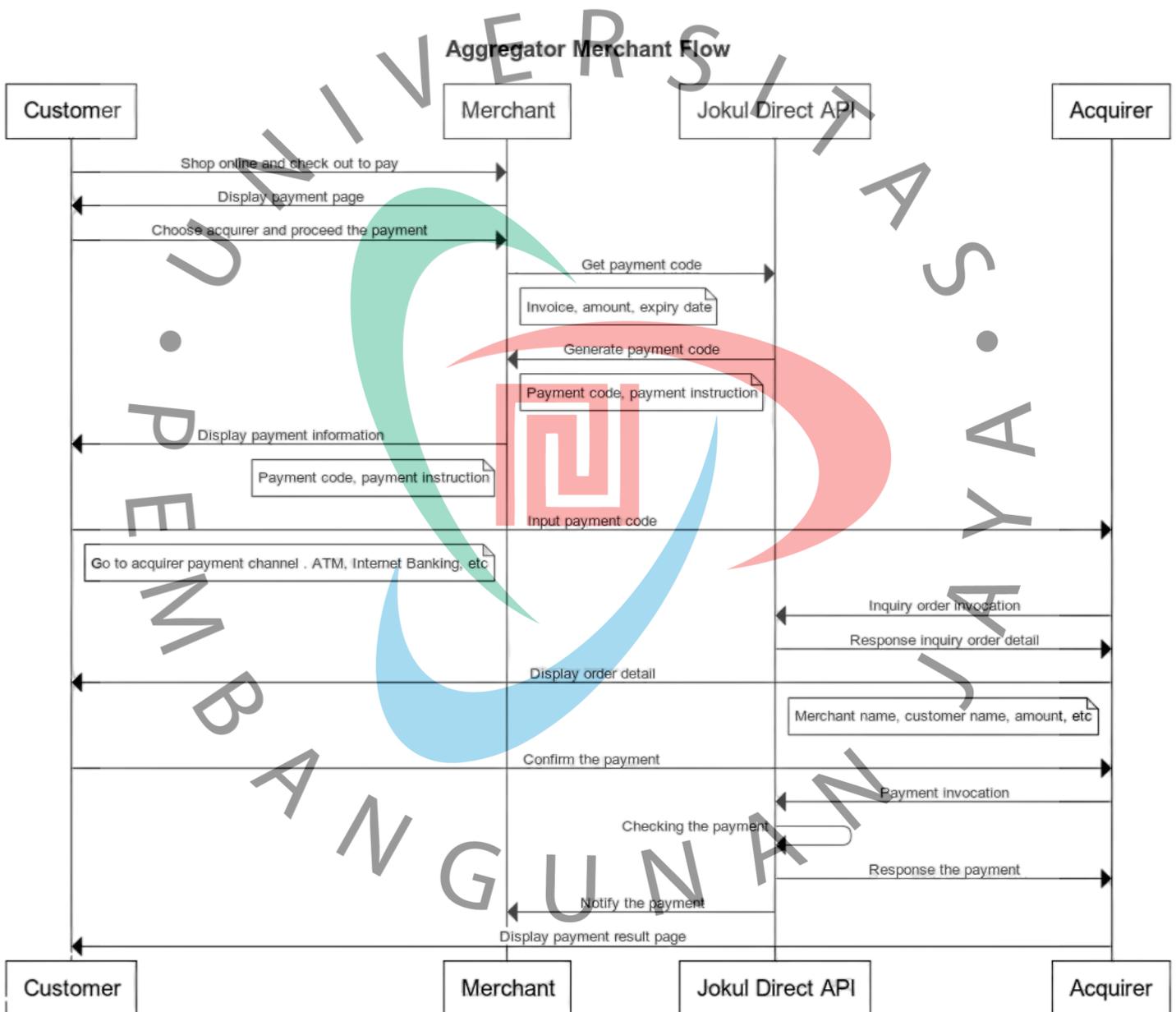
Non-SNAP Virtual Account adalah jenis layanan yang disediakan oleh berbagai bank untuk memfasilitasi pembayaran melalui *virtual account* tanpa menggunakan platform SNAP (Standar Nasional Open API Pembayaran). Layanan ini memungkinkan *merchant* atau pengguna untuk menerima pembayaran dari pelanggan dengan menggunakan nomor *virtual account* yang unik untuk setiap transaksi atau pelanggan. Berikut adalah gambaran mengenai cara kerja *Non-SNAP* pada *Payment Gateway* DOKU:



Gambar 4.21 DOKU Generate Virtual Account

4.4.8 Online to Offline (O2O) Workflow

O2O merupakan layanan gabungan antara platform dengan metode pembayaran *offline*. Metode ini memungkinkan pelanggan atau peserta untuk berinteraksi secara *online* dan melakukan pembayaran secara fisik pada minimarket seperti Alfa Group (Alfamart, Alfamidi dan Lawson). Berikut adalah Gambaran cara kerja O2O pada *Payment Gateway* DOKU:



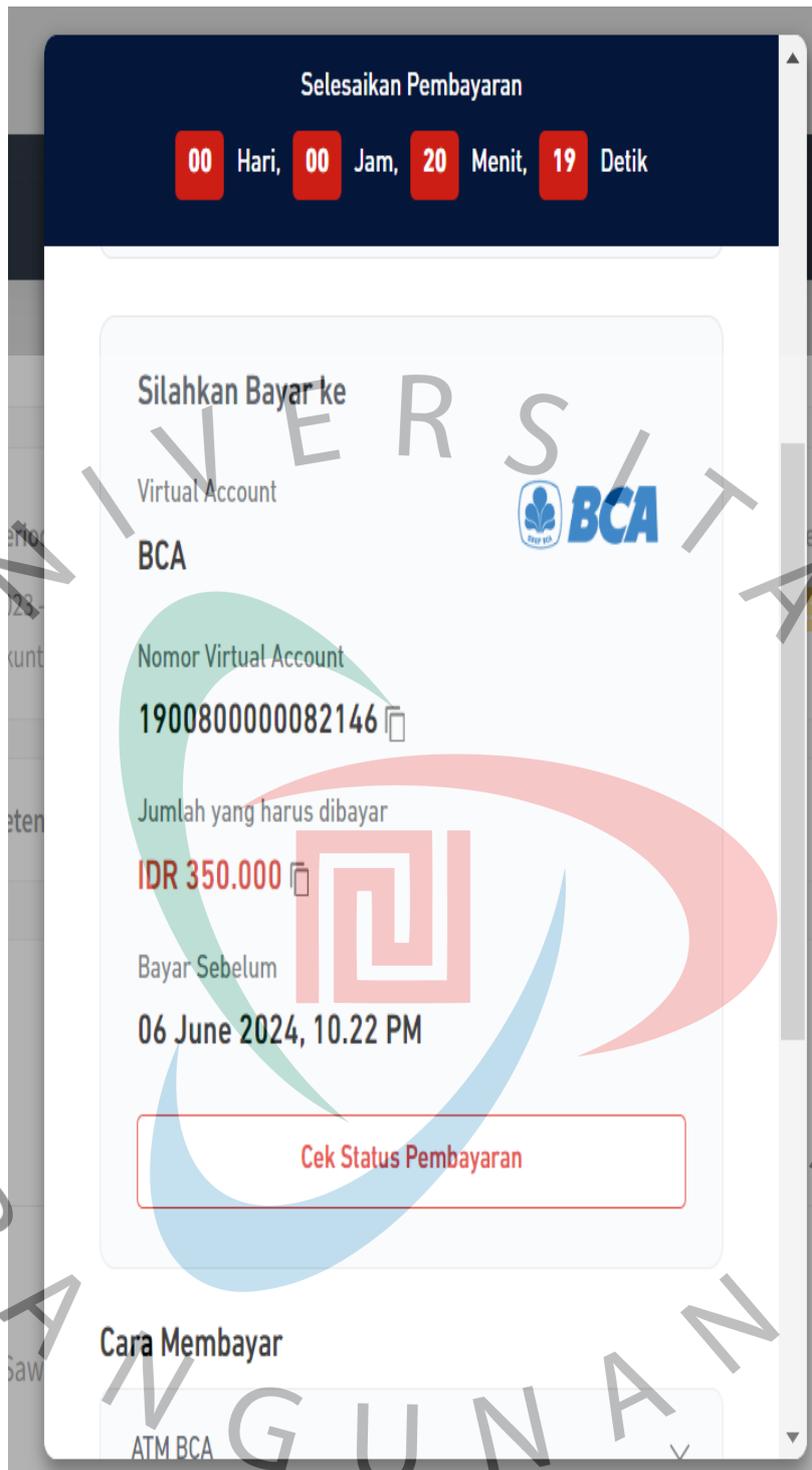
Gambar 4.22 DOKU O2O Direct Api

4.4.9 Implementasi Antarmuka

Berdasarkan perancangan antarmuka yang telah dijelaskan sebelumnya, pada tahap ini peneliti mencoba untuk mengimplementasikannya dengan kode program yang telah dibuat, berikut adalah implementasi antarmuka yang dibuat:



Gambar 4.23 Implementasi Checkout Pembayaran



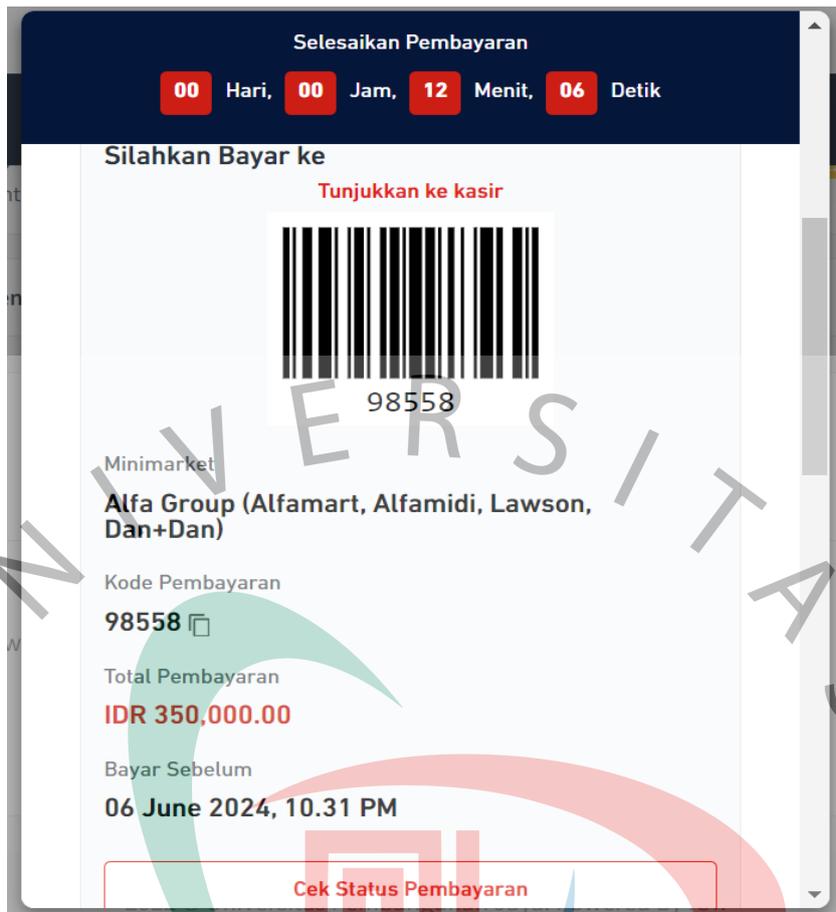
Gambar 4.24 Implementasi Pembayaran Virtual Account Bank BCA



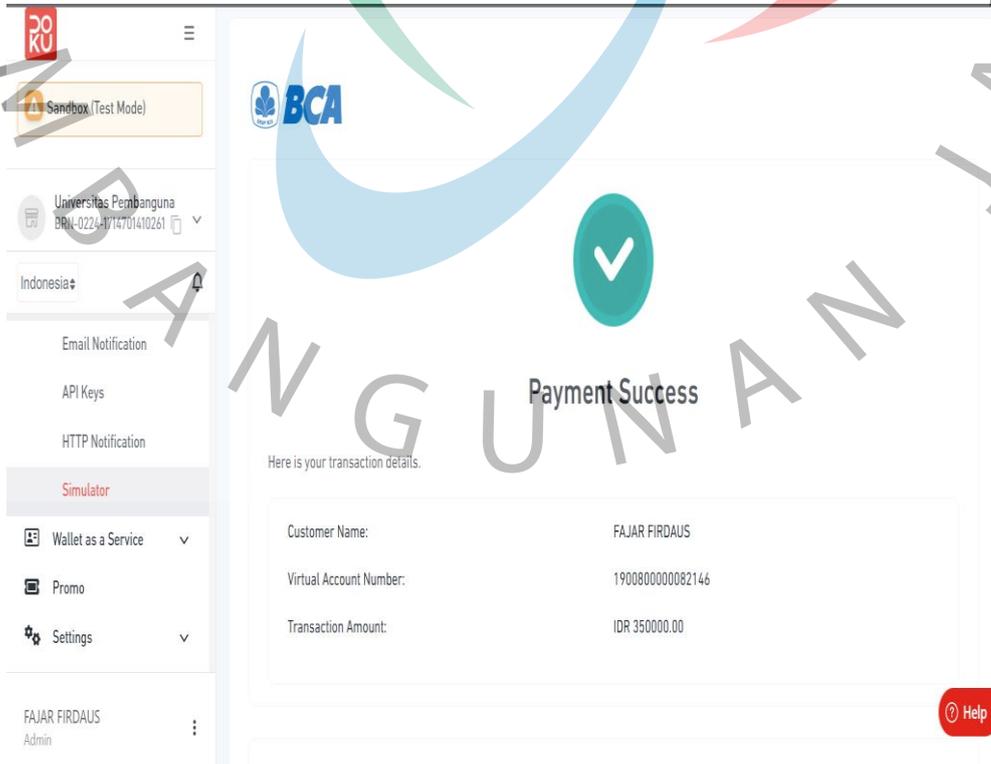
Gambar 4.25 Implementasi Pembayaran Virtual Account Bank Mandiri



Gambar 4.26 Implementasi Pembayaran Virtual Account Bank BRI



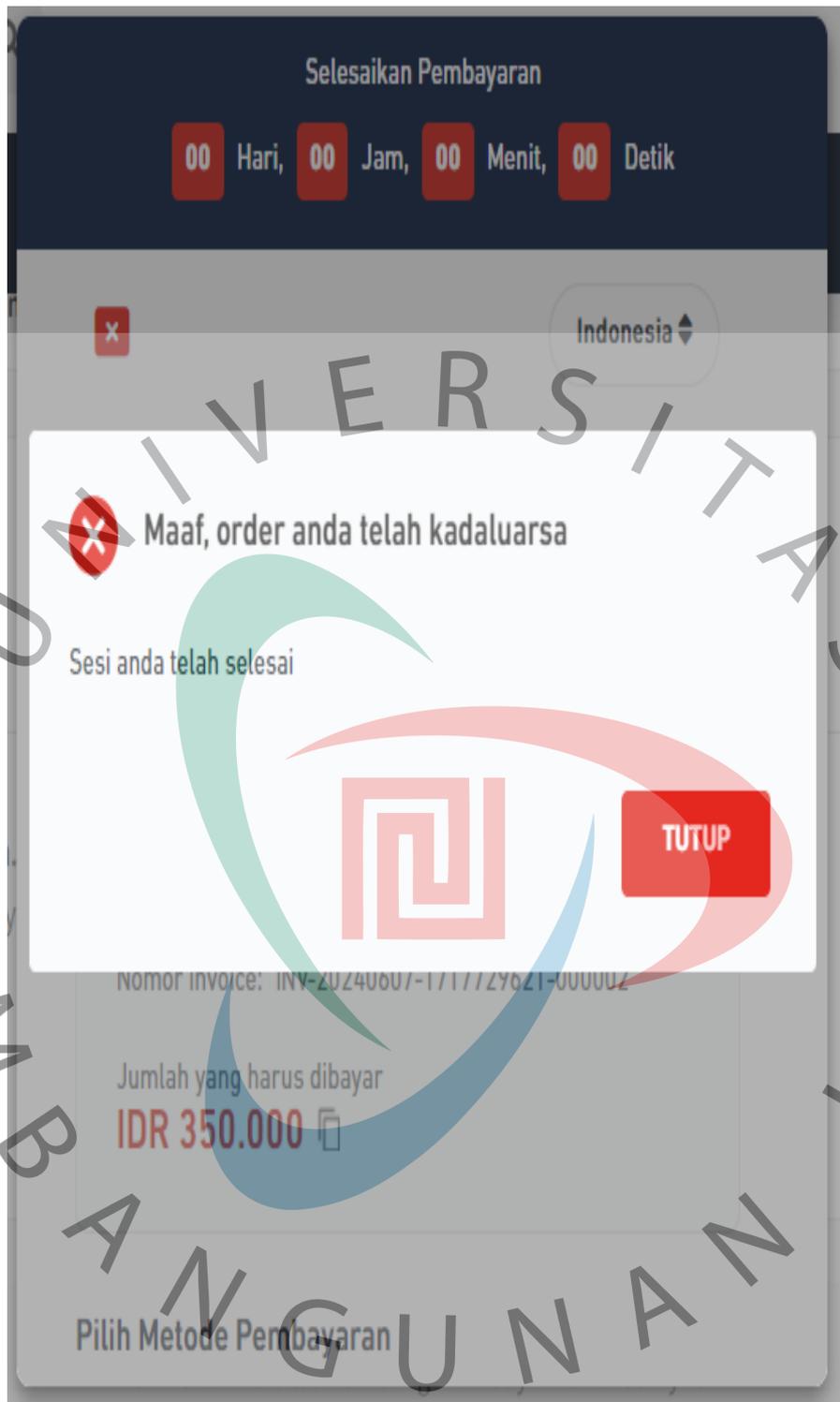
Gambar 4.27 Implementasi Metode Pembayaran O2O (Online to Offline)



Gambar 4.28 Implementasi Simulator Pembayaran Virtual Account BCA DOKU Sandbox



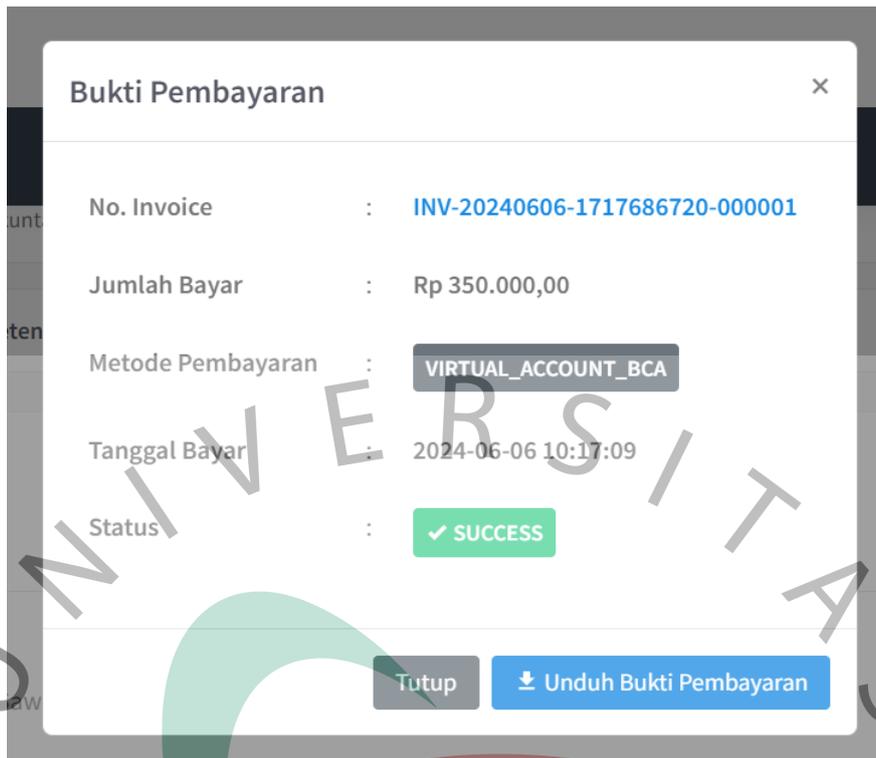
Gambar 4.29 Hasil Implementasi Pembayaran Success Virtual Account BCA



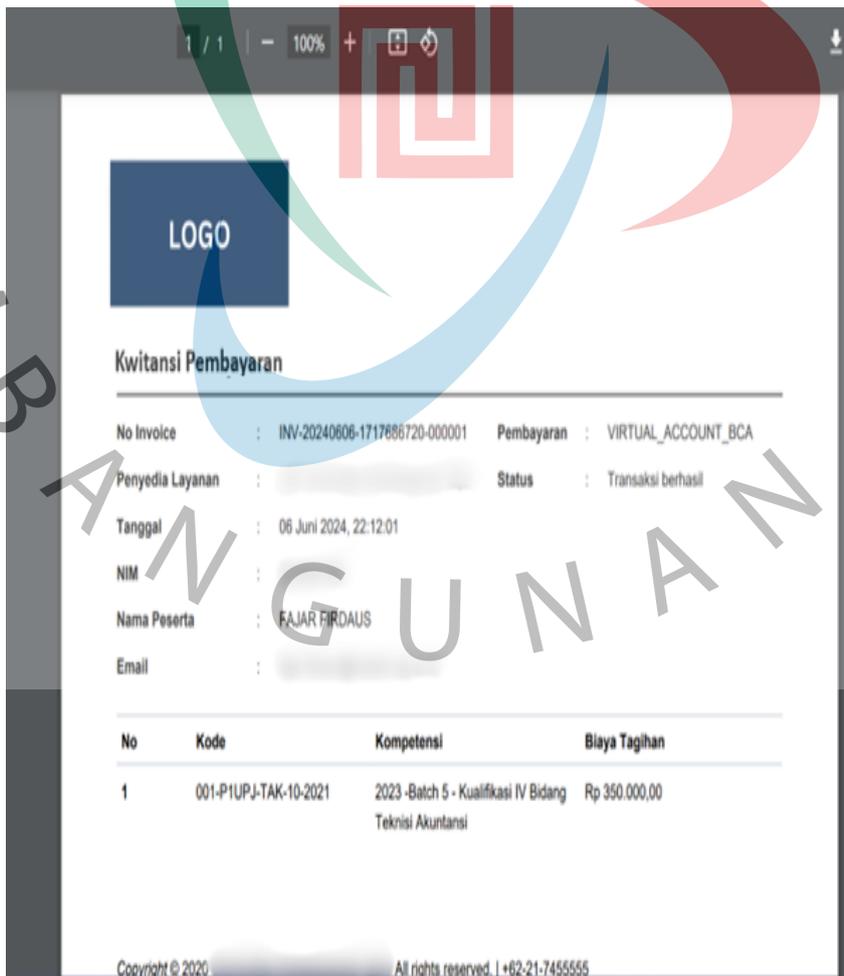
Gambar 4.30 Implementasi Modal Pembayaran Expired



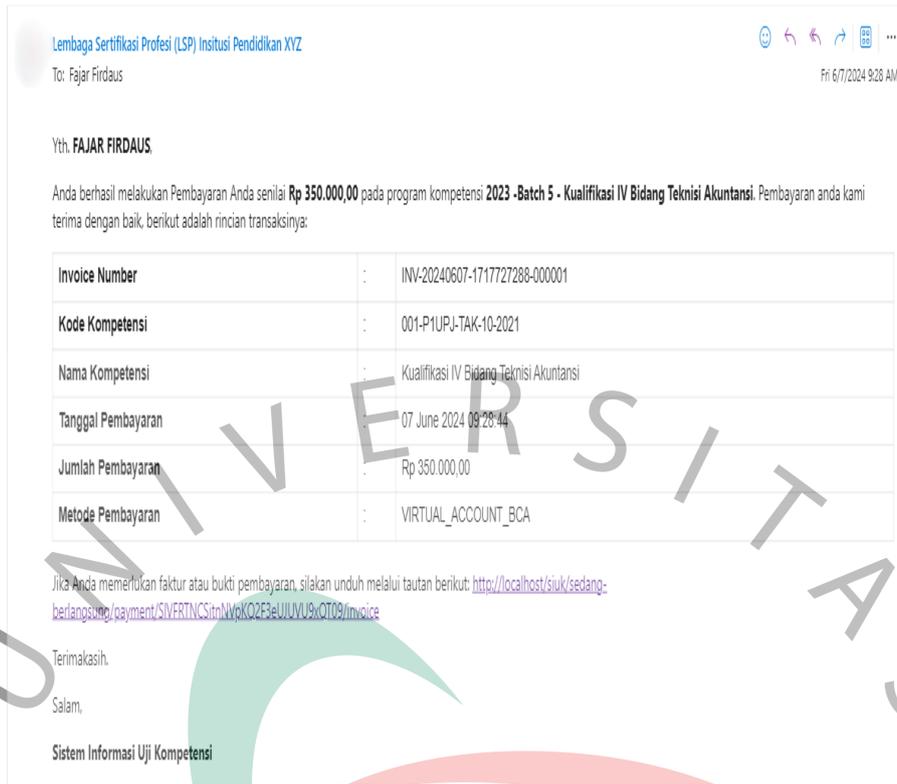
Gambar 4.31 Implementasi Pembayaran Expired



Gambar 4.32 Implementasi Bukti Pembayaran

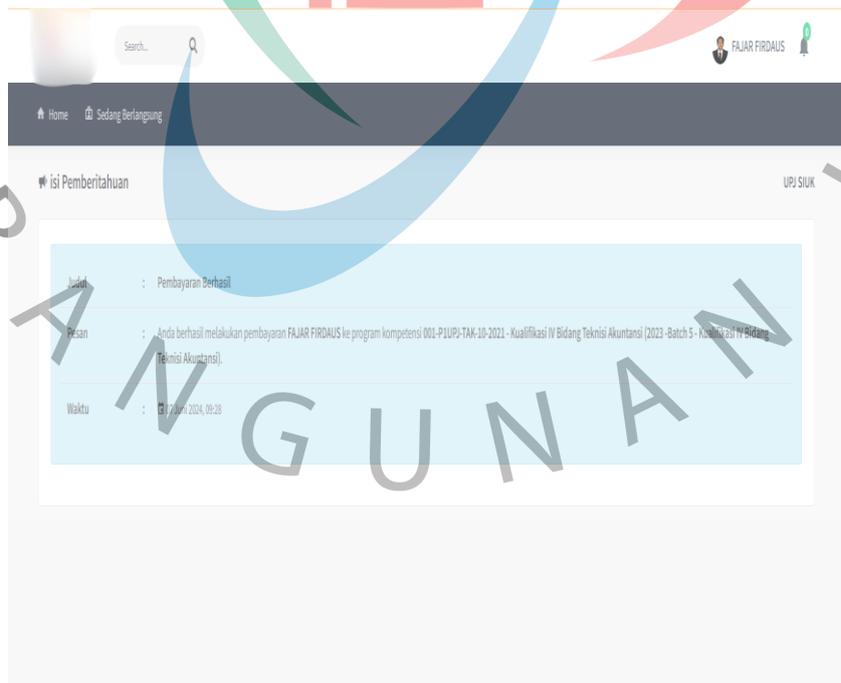


Gambar 4.33 Implementasi Cetak Kwitansi Pembayaran

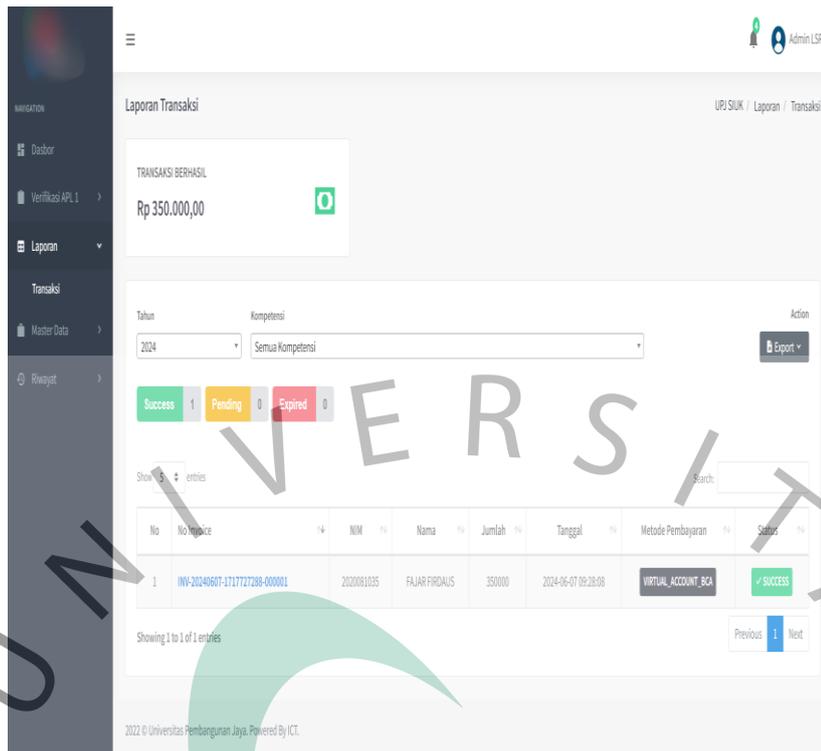


Mohon untuk tidak membalas email otomatis ini.
<http://localhost/siuk/>

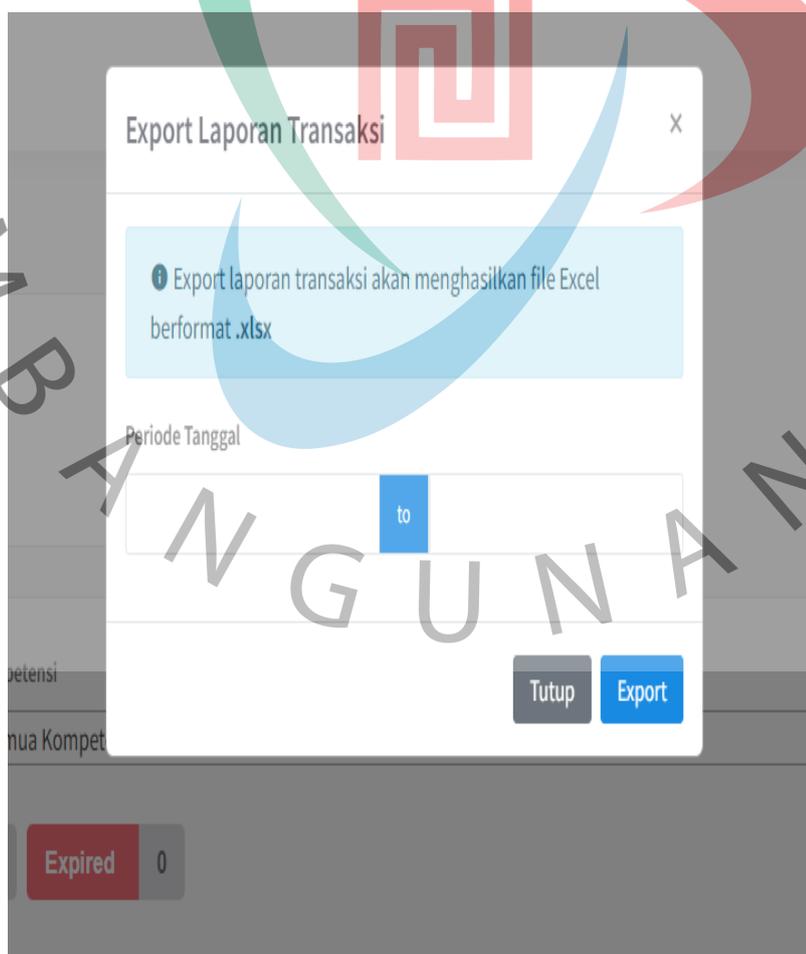
Gambar 4.34 Implementasi Notifikasi Email Pembayaran



Gambar 4.35 Implementasi Notifikasi Pembayaran pada Aplikasi



Gambar 4.36 Implementasi Halaman Laporan Transaksi



Gambar 4.37 Implementasi Modal Export Laporan Transaksi

4.5 Pengujian Sistem

Bagian ini menjelaskan pengujian sistem yang sudah direncanakan sebelumnya, pengujian ini bertujuan agar dapat dijadikan sebagai pedoman utama untuk pengembangan selanjutnya. Pengujian yang menyeluruh dapat memastikan bahwa semua fitur yang dikembangkan dan fungsi sistem bekerja sesuai dengan spesifikasi yang ditetapkan.

Selain itu, pengujian ini bermaksud untuk mengidentifikasi dan memperbaiki bug atau masalah yang mungkin timbul selama penggunaan. Dengan demikian, sistem dapat mencapai tingkat keandalan dan performa yang optimal sebelum diterapkan secara penuh di lingkungan produksi. Hasil dari pengujian ini akan memberikan wawasan bermanfaat yang mendukung proses peningkatan kualitas dan penyesuaian lebih lanjut untuk memenuhi kebutuhan pengguna secara lebih efektif.

Tabel 4.8 Tabel Skenario Pengujian

No.	Test ID	Test Condition	Scenario	Result
1	VIRTUAL_ACCO UNT_BCA_PAY	Positif	Klik tombol bayar dan memilih metode transfer bank BCA dan melakukan pembayaran	Pembayaran berhasil dan menerima notifikasi
2	VIRTUAL_ACCO UNT_BCA_PAY	Negatif	Klik tombol bayar dan memilih metode transfer bank BCA dan tidak melakukan pembayaran selama 30 menit	Pembayaran gagal
3	VIRTUAL_ACCO UNT_MANDIRI_ PAY	Positif	Klik tombol bayar dan memilih metode transfer bank Mandiri dan melakukan pembayaran	Pembayaran berhasil dan menerima notifikasi
4	VIRTUAL_ACCO UNT_MANDIRI_ PAY	Negatif	Klik tombol bayar dan memilih metode transfer bank Mandiri dan tidak melakukan pembayaran selama 30 menit	Pembayaran gagal

5	VIRTUAL_ACCO UNT_BRI_PAY	Positif	Klik tombol bayar dan memilih metode transfer bank BRI dan melakukan pembayaran	Pembayaran berhasil dan menerima notifikasi
6	VIRTUAL_ACCO UNT_BRI_PAY	Negatif	Klik tombol bayar dan memilih metode transfer bank BRI dan tidak melakukan pembayaran selama 30 menit	Pembayaran gagal
7	VIRTUAL_ACCO UNT_PERMATA _PAY	Positif	Klik tombol bayar dan memilih metode pembayaran transfer bank Permata dan melakukan pembayaran	Pembayaran berhasil dan menerima notifikasi
8	VIRTUAL_ACCO UNT_PERMATA _PAY	Negatif	Klik tombol bayar dan memilih pembayaran transfer bank Permata dan tidak melakukan pembayaran selama 30 menit	Pembayaran gagal
9	VIRTUAL_ACCO UNT_CIMB_PAY	Positif	Klik tombol bayar dan memilih metode pembayaran transfer bank Cimb Niaga dan melakukan pembayaran	Pembayaran berhasil dan menerima notifikasi
10	VIRTUAL_ACCO UNT_CIMB_PAY	Negatif	Klik tombol bayar dan memilih metode pembayaran transfer bank Cimb Niaga dan tidak melakukan pembayaran selama 30 menit	Pembayaran gagal
11	ONLINE_TO_OF FLINE_ALFA_PA	Positif	Klik tombol bayar dan memilih metode O2O	Pembayaran berhasil dan menerima

	Y		(<i>Online to Offline</i>) dan melakukan pembayaran	notifikasi
12	ONLINE_TO_OF FLINE_ALFA_PA Y	Negatif	Klik tombol bayar dan memilih metode O2O (<i>Online to Offline</i>) dan tidak melakukan pembayaran selama 30 menit	Pembayaran gagal
13	HISTORY_PAYM ENT	Positif	History pembayaran dapat dilihat ketika selesai melakukan pembayaran dengan sukses	Detail pembayaran dapat dilihat
14	RECEIPT_PAYM ENT	Positif	Kwitansi Pembayaran dapat diunduh ketika selesai melakukan pembayaran dengan sukses	Kwitansi Pembayaran dapat ditampilkan dalam bentuk pdf dan dapat diunduh
15	TRANSACTION_ REPORT	Positif	Admin atau manajer telah berhasil login kemudian mengakses halaman laporan transaksi	Sistem menampilkan laporan transaksi
16	TRANSACTION_ REPORT_FILTER	Positif	Admin atau manajer mengakses halaman laporan transaksi dan filter data	Sistem menampilkan data sesuai filter
17	EXPORT_TRANS ACTION_REPOR T	Positif	Admin atau manajer mengakses halaman laporan transaksi dan export laporan	Laporan transaksi dapat diexport dalam bentuk excel sesuai dengan periode yang dipilih

4.6 Perencanaan Implementasi

Berikut ini merupakan penjelasan tentang jadwal kegiatan yang dilakukan dalam penelitian ini yang terdiri dari perencanaan, analisis, desain dan implementasi.

Tabel 4.9 Gantt Chart jadwal Implementasi Pengembangan Sistem

Kegiatan	Februari				Maret				April				Mei				Juni			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Perencanaan (<i>Planning</i>)																				
Pembuatan Jadwal Pengembangan																				
Analisis (<i>Analysis</i>)																				
Mengidentifikasi data beserta dokumen yang sedang berjalan																				
Identifikasi Kebutuhan Pengguna																				
Analisis sistem yang sedang berjalan																				
Desain (<i>Design</i>)																				
Perancangan diagram UML																				
Membuat perancangan antarmuka																				
Perancangan database																				
Implementasi (<i>Implementation</i>)																				
Memahami kebutuhan integrasi <i>Payment Gateway</i> DOKU																				
Pembuatan kode program																				
Pengujian Sistem																				