



APLIKASI *PAYMENT GATEWAY AGGREGATOR*

Oleh :

**MOHAMAD FEBRIAN MOSII
4519215002**

**SKRIPSI
PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK UNIVERSITAS PANCASILA
JAKARTA
2023**



PAYMENT GATEWAY AGGREGATOR APPLICATION

Oleh :

**MOHAMAD FEBRIAN MOSII
4519215002**

**ESSAY
INFORMATICS ENGINEERING STUDY PROGRAM
ENGINEERING FACULTY OF PANCASILA UNIVERSITY
JAKARTA
2023**

LEMBAR PERSETUJUAN

Skripsi ini dibuat untuk
memenuhi syarat guna mencapai gelar

SARJANA TEKNIK

pada Program Studi Teknik Informatika Fakultas Teknik
Universitas Pancasila Jakarta dengan judul:

APLIKASI *PAYMENT GATEWAY AGGREGATOR*

dinyatakan telah disetujui dan diterima:

Menyetujui:

Gregorius Hendita Artha Kusuma,S.Si.,M.Cs
Pembimbing

Mengetahui:

Dr. Ionia Veritawati, S.Si., MT
Ketua Program Studi Teknik Informatika

LEMBAR KEASLIAN

Dengan ini saya yang bertanda tangan di bawah ini :

Nama : Mohamad Febrian Mosii

No. Pokok : 4519215002

Fakultas : Teknik

Program Studi : Teknik Informatika

Peminatan : *Software Engineering*

Menyatakan :

Bahwa skripsi ini dibuat dan diselesaikan secara mandiri dan bukan hasil saduran karya orang lain serta hanya menggunakan acuan dari literatur yang ada.

Jika terbukti tidak sesuai dengan yang tersebut di atas, maka saya bersedia menerima sanksi yang ada.

Jakarta, 2 Juni 2023

Mohamad Febrian Mosii

KATA PENGANTAR

Assalamu'alaikum Wr.Wb.

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya, yang telah memungkinkan penulis menyelesaikan skripsi ini dengan baik.

Skripsi ini merupakan bagian penting dari perjalanan akademik penulis sebagai mahasiswa Program Studi Teknik Informatika Universitas Pancasila, dalam rangka meraih gelar sarjana. Skripsi ini disusun berdasarkan penelitian dan observasi yang penulis lakukan di CV Himalaya Teknologi, khususnya di produk Gasmaj, serta dengan merujuk pada jurnal-jurnal ilmiah terkait yang relevan dengan judul skripsi ini.

Dengan selesainya laporan skripsi ini tidak terlepas dari bantuan banyak pihak yang telah memberikan masukan-masukan kepada penulis. Untuk itu penulis mengucapkan banyak terima kasih kepada:

1. Ibu Dr. Ionia Veritawati S.Si., MT., selaku Ketua Program Studi Teknik Informatika Fakultas Teknik Universitas Pancasila.
2. Bapak Amir Murtako, S.Kom., M.Kom, selaku Koordinator Skripsi Program Studi Teknik Informatika Fakultas Teknik Universitas Pancasila.
3. Bapak Gregorius Hendita Artha Kusuma,S.Si.,M.Cs, selaku pembimbing skripsi.
4. Bapak Muhammad Azhar Rasyad, selaku staff perwakilan dari CV Himalaya Teknologi.

5. Kedua orang tua, yang telah memberikan dukungan penuh untuk menempuh pendidikan di Universitas Pancasila.
6. Istri dan Anggota keluarga lainnya, yang juga memberi dukungan dan saran untuk penelitian yang saya kerjakan.
7. Seluruh pihak yang telah terlibat dalam kelancaran skripsi ini dan belum disebutkan di atas, penulis ucapkan terima kasih.

Penulis dengan rendah hati menyadari bahwa dalam laporan ini masih terdapat banyak kekurangan, baik dalam hal materi maupun teknik penyajiannya. Penulis mengakui bahwa hal ini disebabkan oleh keterbatasan pengetahuan dan pengalaman yang dimiliki.

Penulis menerima kritik dan saran dengan lapang dada, dan akan berupaya semaksimal mungkin untuk memperbaiki kekurangan-kekurangan yang ada. Dengan adanya kontribusi dan bimbingan dari pihak-pihak yang terlibat, penulis yakin bahwa laporan ini dapat menjadi lebih baik dan lebih berarti.

Terima kasih atas perhatian dan dukungannya. Kritik dan saran yang diberikan sangat penulis harapkan untuk kemajuan laporan ini.

Wassallamu 'alaikum, Wr. Wb.

Jakarta, 2 Juni 2023

Mohamad Febrian Mosii

DAFTAR ISI

LEMBAR PERSETUJUAN.....	i
LEMBAR KEASLIAN	ii
KATA PENGANTAR	iii
DAFTAR ISI.....	v
DAFTAR GAMBAR	xii
DAFTAR TABEL.....	xvi
DAFTAR LAMPIRAN.....	xvii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Tujuan dan Manfaat Penelitian	5
1.3.1 Tujuan Penelitian	5
1.3.2 Manfaat Penelitian	5
1.4 Batasan Masalah.....	7
1.5 Metodologi Penelitian	8
1.5.1 Tahap Pengumpulan Data	8
1.5.2 Tahapan Pembuatan Sistem	8
1.6 Sistematika Penulisan.....	10
1.7 Kerangka Pemikiran.....	12

BAB II LANDASAN TEORI	14
2.1 Pembeli.....	14
2.2 <i>Merchant</i>	14
2.3 Transaksi Online	16
2.4 Pembayaran <i>Online</i>	17
2.5 Integrasi Sistem.....	19
2.6 <i>Payment Gateway</i>	20
2.7 <i>Payment Gateway Aggregator</i>	24
2.8 Hubungan <i>Payment Gateway</i> dan <i>Payment Gateway Aggregator</i>	25
2.9 Penelitian Terdahulu	27
BAB III ANALISA SISTEM SEDANG BERJALAN	28
3.1 Gambaran Umum Perusahaan.....	28
3.1.1 Sejarah Perusahaan.....	28
3.1.2 Struktur Organisasi Perusahaan	28
3.1.3 Uraian Pekerjaan	29
3.1.4 Bidang dan Kegiatan Perusahaan.....	30
3.2 Sistem yang Sedang Berjalan.....	31
3.2.1 Deskripsi Sistem yang Sedang Berjalan	31
3.2.2 <i>Workflow</i> Sistem yang Sedang Berjalan	32
3.3 Analisis Sistem.....	34

3.3.1	Analisis Permasalahan	34
3.3.2	Analisis Kebutuhan	34
3.3.3	Solusi Permasalahan.....	34
	BAB IV PERANCANGAN SISTEM.....	35
4.1	Pendahuluan	35
4.2	Arsitektur Perangkat Lunak	37
4.3	Pemodelan Sistem / UML	40
4.3.1	Use Case Diagram.....	40
4.3.2	<i>Activity Diagram</i>	43
4.3.3	<i>Sequence Diagram</i>	53
4.3.4	Class Diagram	64
4.3.5	Entity Relationship Diagram.....	66
4.4	Perancangan <i>Output</i>	67
4.4.1	Halaman Pilih Metode Pembayaran.....	67
4.4.2	Halaman Konfirmasi Pembayaran	70
4.4.3	Halaman Pembayaran.....	71
4.5	Perancangan <i>Input</i>	74
4.5.1	Input Data <i>Login</i>	74
4.5.2	Input Data Admin.....	75
4.5.3	Input Data Metode Pembayaran.....	76
4.5.4	Input Data Tipe Metode Pembayaran	77
4.5.5	Input Data <i>Payment Gateway</i> Provider.....	77

4.5.6	Input Data Akun <i>Payment Gateway</i>	78
4.5.7	Input Data Merchant	79
4.6	Percanganan Menu	81
BAB V IMPLEMENTASI.....		84
5.1	Spesifikasi Perangkat Keras dan Perangkat Lunak.....	84
5.1.1	Spesifikasi Perangkat Keras.....	84
5.1.2	Spesifikasi Perangkat Lunak.....	85
5.2	Implementasi Basis Data.....	86
5.2.1	Membuat Tabel lulapay_merchant_merchants	86
5.2.2	Membuat Tabel lulapay_paymentgateway_accounts	86
5.2.3	Membuat Tabel lulapay_merchant_payment_accounts.....	87
5.2.4	Membuat Tabel lulapay_transaction_payment_method_types.....	87
5.2.5	Membuat Tabel lulapay_transaction_payment_methods.....	87
5.2.6	Membuat Tabel lulapay_merchant_payment_methods	88
5.2.7	Membuat Tabel lulapay_paymentgateway_providers	88
5.2.8	Membuat Tabel lulapay_transaction_customers.....	88
5.2.9	Membuat Tabel lulapay_transaction_statuses.....	89
5.2.10	Membuat Tabel lulapay_transaction_transactions.....	89
5.2.11	Membuat Tabel lulapay_transaction_transaction_details	89
5.2.12	Membuat Tabel lulapay_transaction_transaction_logs.....	90
5.2.13	Membuat Relasi Tabel	90
5.3	Implementasi Proses.....	91

5.3.1	Implementasi Login	91
5.3.2	Implementasi API <i>Create Transaction</i>	92
5.3.3	Implementasi API Check Status.....	95
5.3.4	Implementasi API Untuk <i>Handle Notification Callback Midtrans</i>	97
5.3.5	Implementasi API Untuk <i>Handle Notification Callback Brankas</i>	
	101	
5.3.6	Percobaan Menggunakan Postman	103
5.3.7	Implementasi API Untuk <i>Handle Notification Callback Stripe..</i>	104
5.3.8	Implementasi <i>Scheduler Expiry Transaction</i>	106
	Implementasi Output.....	107
5.3.9	Halaman Pilih Metode Pembayaran.....	107
5.3.10	Halaman Konfirmasi Pembayaran	110
5.3.11	Halaman Pembayaran <i>Payment Gateway Midtrans</i>	111
5.3.12	Halaman Pembayaran <i>Payment Gateway Brankas</i>	115
5.3.13	Halaman Pembayaran <i>Payment Gateway Stripe</i>	120
5.4	Implementasi Input.....	124
5.4.1	Input Data <i>Backend Users</i>	124
5.4.2	Input Data Metode Pembayaran.....	124
5.4.3	Input Data Tipe Metode Pembayaran	125
5.4.4	Input Data <i>Payment Gateway Provider</i>	125
5.4.5	Input Data Akun <i>Payment Gateway Provider</i>	126
5.5	Implementasi Menu.....	127
5.5.1	Dashboard	127

5.5.2	Implementasi Menu <i>Payment Gateway Providers</i>	128
5.5.3	Implementasi Menu Akun <i>Payment Gateway Providers</i>	129
5.5.4	Implementasi Menu <i>Merchants</i>	129
5.5.5	Implementasi Menu <i>Transactions</i>	130
5.5.6	Implementasi Menu <i>Payment Methods</i>	131
5.5.7	Implementasi Menu <i>Payment Method Types</i>	131
5.5.8	Implementasi Email Notifikasi	132
5.6	Evaluasi Kepuasan Pengguna	137
5.6.1	Kuesioner Pengelola.....	137
5.6.2	Hasil Perhitungan Kuesioner.....	138
	BAB VI KESIMPULAN DAN SARAN	140
6.1	Kesimpulan	140
6.2	Saran.....	141
	DAFTAR PUSTAKA	142

DAFTAR GAMBAR

Gambar 1.1 Kerangka pemikiran	12
Gambar 3.1 Struktur Organisasi CV Himalaya Teknologi	28
Gambar 3.2 Workflow sistem yang berjalan saat ini	32
Gambar 4.1 Diagram Asitektur Perangkat Lunak.....	38
Gambar 4.2 Use Case Diagram Admin.....	41
Gambar 4.3 Use Case Diagram Merchant.....	42
Gambar 4.4 Use Case Customer / Pelanggan.....	42
Gambar 4.5 Activity Diagram Admin Login	43
Gambar 4.6 Activity Diagram Logout	44
Gambar 4.7 Melihat Dashboard	44
Gambar 4.8 Activity Diagram Mengelola Data Provider Payment Gateway	45
Gambar 4.9 Activity Diagram Mengelola Data Akun Payment Gateway	46
Gambar 4.10 Acivity Diagram Mengelola Data Transaksi.....	47
Gambar 4.11 Activity Diagram Mengelola Data Merchant.....	48
Gambar 4.12 Acitivity Diagram Mengelola Data Metode Pembayaran	49
Gambar 4.13 Activity Diagram Mengelola Data Tipe Metode Pembayaran.....	50
Gambar 4.14 Activity Diagram Mengelola Data Admin	51
Gambar 4.15 Activity Diagram Melakukan Transaksi	52
Gambar 4.16 Sequence Diagram Login	53
Gambar 4.17 Sequence Diagram Logout	54
Gambar 4.18 Sequence Diagram Melihat Dashboard.....	54
Gambar 4.19 Sequence Diagram Mengelola Data Akun Payment Gateway.....	55

Gambar 4.20 Sequence Diagram Mengelola Data Transaksi	56
Gambar 4.21 Sequence Diagram Mengelola Data Merchant.....	57
Gambar 4.22 Sequence Diagram Mengelola Data Metode Pembayaran.....	58
Gambar 4.23- Sequence Diagram Mengelola Data Payment Gateway Provider..	59
Gambar 4.24 Sequence Diagram Mengelola Data Tipe Metode Pembayaran.....	60
Gambar 4.25 Sequence Diagram Mengelola Data Admin.....	62
Gambar 4.26 Sequence Diagram Melakukan Transaksi.....	63
Gambar 4.27 Class Diagram	65
Gambar 4.28 Entity Relationship Diagra	66
Gambar 4.29 Output Halaman Pilih Metode Pembayaran.....	67
Gambar 4.30 Output Info Pelanggan	68
Gambar 4.31 Output Accordion Item.....	68
Gambar 4.32 Output Kategori / Tipe Metode Pembayaran	69
Gambar 4.33 Output Daftar Metode Pembayaran.....	69
Gambar 4.34 Output Halaman Konfirmasi Pembayaran	70
Gambar 4.35 Output Halaman Pembayaran.....	71
Gambar 4.36 Logo Metode Pembayaran	72
Gambar 4.37 Batas Waktu Pembayaran.....	72
Gambar 4.38 Output Informasi Pembayaran	73
Gambar 4.39 Output Informasi dan Tata Cara Pembayaran	73
Gambar 4.40 Rancangan formulir login.....	74
Gambar 4.41 Rancangan Input Data Admin	75
Gambar 4.42 Rancangan Input Data Metode Pembayaran	76

Gambar 4.43 Rancangan Input Data Tipe Metode Pembayaran.....	77
Gambar 4.44 Rancangan Input Data Payment Gateway Provider	77
Gambar 4.45 Rancangan Input Data Akun Payment Gateway	78
Gambar 4.46 Rancangan Input Data Merchant.....	79
Gambar 5.1 Implementasi Login.....	91
Gambar 5.2 Curl API Create Transaction	93
Gambar 5.3 Postman API Create Transaction	94
Gambar 5.4 Response API Create Transaction.....	94
Gambar 5.5 Curl API Check Status	96
Gambar 5.6 Postman API Check Status.....	96
Gambar 5.7 Response API Check Status	97
Gambar 5.8 Curl API Handle Notification Callback Midtrans.....	99
Gambar 5.9 Postman API Handle Notification Callback Midtrans	99
Gambar 5.10 Response API Handle Notification Callback Midtrans.....	100
Gambar 5.11 Referensi kode status Brankas.....	102
Gambar 5.12 Curl API Handle Notification Callback Brankas	103
Gambar 5.13 Postman API Handle Notification Callback Brankas	103
Gambar 5.14 Response API Handle Notification Callback Brankas	104
Gambar 5.15 Hasil Implementasi Handle Notification Callback Stripe	105
Gambar 5.16 Implementasi Scheduler Expiry Transaction	106
Gambar 5.17 Implementasi Halaman Pilih Metode Pembayaran	107
Gambar 5.18 Implementasi Info Pelanggan.....	108
Gambar 5.19 Implementasi Accordion Item.....	109

Gambar 5.20 Implementasi Kategori / Tipe Metode Pembayaran.....	109
Gambar 5.21 Implementasi Halaman Konfirmasi Pembayaran.....	110
Gambar 5.22 Implementasi Halaman Pembayaran	111
Gambar 5.23 Implementasi Logo Metode Pembayaran.....	112
Gambar 5.24 Batas Waktu Pembayaran.....	112
Gambar 5.25 Implementasi Informasi Pembayaran.....	113
Gambar 5.26 Implementasi Informasi dan Tata Cara Pembayaran	114
Gambar 5.27 Implementasi Halaman Pembayaran Brankas	115
Gambar 5.28 Implementasi Halaman Konfirmasi Brankas	116
Gambar 5.29 Implementasi Halaman Konfirmasi APPLI 2 Brankas	117
Gambar 5.30 Implementasi Halaman Konfirmasi APPLI 1 Brankas	117
Gambar 5.31 Halaman Informasi Pembayaran Success Brankas	118
Gambar 5.32 Halaman Informasi Pembayaran Aplikasi.....	119
Gambar 5.33 Halaman Pembayaran.....	119
Gambar 5.34 Implementasi Halaman Pembayaran Stripe	120
Gambar 5.35 Implementasi Halaman Konfirmasi Stripe	121
Gambar 5.36 Implementasi Halaman Pembayara Stripe	122
Gambar 5.37 Halaman Informasi Pembayaran Aplikasi.....	123
Gambar 5.38 Halaman Pembayaran.....	123
Gambar 5.39 Implementasi Input Data Backend Users	124
Gambar 5.40 Implementasi Input Data Metode Pembayaran	124
Gambar 5.41 Implementasi Input Data Tipe Metode Pembayaran.....	125
Gambar 5.42 Implementasi Input Data Payment Gateway Provider	125

Gambar 5.43 Implementasi Input Data Akun Payment Gateway	126
Gambar 5.44 Implementasi Menu Dashboard	127
Gambar 5.45 Implementasi Menu Payment Gateway Providers	128
Gambar 5.46 Implementasi Menu Payment Gateway Providers	129
Gambar 5.47 Implementasi Menu Merchant	129
Gambar 5.48 Implementasi Menu Transactions	130
Gambar 5.49 Implementasi Menu Payment Methods.....	131
Gambar 5.50 Implementasi Menu Payment Method Types.....	131
Gambar 5.51 Email Notifikasi Pending	133
Gambar 5.52 Email Notifikasi Paid	134
Gambar 5.53 Email Notifikasi Expired.....	135
Gambar 5.54 Email Notifikasi Failed	136

DAFTAR TABEL

Tabel 2.1 Penelitian terdahulu.....	27
Tabel 5.1 Kuesioner untuk pengelola merchant.....	137
Tabel 5.2 Kuesioner untuk calon pelanggan	138
Tabel 5.3 Hasil Perhitungan Kuesioner	138

DAFTAR LAMPIRAN

Lampiran 1: Perhitungan Evaluasi Kepuasan Pelanggan 145

Mohamad Febrian Mosii
Aplikasi Payment Gateway Aggregator
Dibawah bimbingan Gregorius Hendita Artha Kusuma,S.Si.,M.Cs. 144 hal + 4
tabel / 103 gambar / 20 pustaka tahun 2019 s.d 2023.

ABSTRAK

Gasmaj, sebuah sistem informasi pengelolaan masjid yang dikembangkan oleh CV Himalaya Teknologi, menghadapi tantangan dalam mengelola pembayaran secara manual. Tanpa sistem pembayaran terintegrasi. Keterbatasan dalam opsi metode pembayaran yang ditawarkan dapat mempengaruhi preferensi pembayaran pelanggan.

Untuk mengatasi tantangan ini, dilakukan penelitian untuk membangun sebuah “Aplikasi *Payment Gateway Aggregator*”, dengan metode pengembangan *waterfall* dan perancangan berorientasi objek menggunakan UML (*Unified Modeling Language*). Subjek pada penelitian ini adalah pengelolaan pembayaran transaksi yang dilakukan oleh pelanggan.

Sumber data didapatkan langsung dari pemilik usaha yang mampu menceritakan kembali peristiwa yang dialaminya, dan bersedia dijadikan subjek penelitian yang mendapatkan kesimpulan bahwa dibutuhkan *aplikasi payment gateway aggregator* untuk pengelolaan pembayaran transaksi secara online.

Dari hasil penelitian yang dilaksanakan, penulis mengimplementasikan hasil penelitian tersebut yang diharapkan akan mempermudah proses pengelolaan transaksi dan meningkatkan kepuasan pelanggan dalam melakukan transaksi secara cepat dan efektif.

Kata Kunci: Aggregator, sistem pembayaran terintegrasi, payment gateway aggregator

Mohamad Febrian Mosii

Payment Gateway Aggregator Application

Under the guidance of Gregorius Hendita Artha Kusuma, S.Si., M.Cs. 144 pages

+ 4 tables / 103 pictures / 20 references from 2019 to 2023.

ABSTRACT

Gasmaj, an mosque management information system developed by CV Himalaya Teknologi, faces challenges in manually handling payments. Without an integrated payment system, the limited options for payment methods offered can affect customer payment preferences.

To address this challenge, research is conducted to develop a "Payment Gateway Aggregator Application" using the waterfall development method and object-oriented design using UML (Unified Modeling Language). The subject of this research is the management of transaction payments made by customers.

Data is obtained directly from business owners who are capable of recounting their experiences and willing to be subjects of the research, leading to the conclusion that an application for a payment gateway aggregator is needed for managing online transaction payments.

Based on the conducted research, the author implements the research findings, aiming to streamline the transaction management process and enhance customer satisfaction by enabling quick and effective transactions.

Kata Kunci: Aggregator, integrated payment system, payment gateway

BAB I

PENDAHULUAN

1.1 Latar Belakang

CV Himalaya Teknologi, memiliki sebuah produk bernama Gasmaj, sebuah aplikasi pengelolaan masjid yang menargetkan puluhan hingga ratusan masjid yang terintegrasi. Adapun layanan Gasmaj salah satunya adalah menyediakan sebuah fitur pembayaran zakat, infaq, dan qurban secara *online*.

Saat ini aplikasi masih dalam tahapan pengembangan dan pemilik usaha masih belum terpikirkan untuk menggunakan sistem *payment gateway* untuk menerima dan mengelola pembayaran, mereka ingin mengelola pembayaran secara manual dengan menggunakan WhatsApp, cara ini tentu sangat kurang efisien jika sudah memiliki banyak pelanggan.

Jika tidak memiliki *channel* pembayaran menggunakan *payment gateway* dapat menjadi tantangan yang signifikan karena *payment gateway* memainkan peran penting dalam memfasilitasi transaksi *online* dengan menyediakan metode pembayaran yang aman dan efisien. Tanpa *payment gateway*, Gasmaj akan kehilangan kemampuan untuk menerima pembayaran secara langsung melalui berbagai metode pembayaran populer seperti kartu kredit, kartu debit, transfer bank, atau dompet digital [1].

Tanpa *payment gateway*, Gasmaj mungkin terbatas dalam memenuhi preferensi pembayaran pelanggan. Pelanggan *modern* cenderung mencari kemudahan dan keamanan dalam bertransaksi *online*. Dengan tidak adanya

payment gateway yang terintegrasi dengan Gasmaj, Gasmaj mungkin kehilangan peluang untuk menjangkau dan memenuhi kebutuhan pelanggan yang menginginkan opsi pembayaran yang mudah atau beragam [3].

Selain itu, Gasmaj juga harus menghadapi tantangan dalam mengelola proses pembayaran secara manual. Ini termasuk mengumpulkan informasi pembayaran dari pelanggan, memverifikasi pembayaran, dan melakukan aksi secara manual. Mengelola aspek pembayaran ini secara manual dapat memakan waktu dan sumber daya yang berharga, mengganggu fokus Anda pada pengembangan bisnis dan pengalaman pelanggan secara keseluruhan [4], [5].

Selain keterbatasan operasional, tidak memiliki *channel* pembayaran yang terintegrasi juga dapat mempengaruhi kepercayaan pelanggan terhadap Gasmaj. Dalam era digital ini, keamanan pembayaran menjadi perhatian utama bagi pelanggan. Dengan tidak adanya sistem pembayaran yang terintegrasi, pelanggan mungkin merasa kurang percaya untuk memasukkan informasi pembayaran mereka secara manual, meningkatkan risiko kehilangan pelanggan [5].

Setiap *payment gateway* biasanya memiliki beberapa opsi pembayaran seperti kartu kredit, transfer bank, atau dompet digital. Namun, jika hanya bergantung pada satu *payment gateway*, tetap ada batasan ketersediaan metode pembayaran, sebagai contoh *payment gateway* “Midtrans” menyediakan pembayaran dompet digital Gopay, namun tidak memiliki metode pembayaran OVO, dan sebaliknya, untuk *payment gateway* “Brankas” memiliki metode pembayaran OVO tapi tidak memiliki metode pembayaran Gopay. Hal ini dapat mengurangi fleksibilitas dalam memenuhi preferensi pembayaran pelanggan.

Gasmaj akan menghadapi tantangan dalam mengelola integrasi *payment gateway* secara terpisah untuk setiap metode pembayaran yang disediakan. Setiap *payment gateway* memiliki persyaratan teknis yang berbeda, dan mengelola masing-masing *payment gateway* secara terpisah membutuhkan waktu dan upaya yang tidak sedikit. Selain itu, jika Gasmaj ingin menambahkan atau mengubah metode pembayaran yang ada, Gasmaj akan menghadapi kompleksitas tambahan dalam mengelola integrasi tersebut.

Tanpa *payment gateway aggregator*, Gasmaj juga akan menghadapi keterbatasan dalam membandingkan tarif dan biaya antara *payment gateway*. *Payment gateway aggregator* adalah sebuah platform atau layanan yang mengintegrasikan banyak *payment gateway* atau metode pembayaran ke dalam satu antarmuka terpusat. *Payment gateway aggregator* ini bertindak sebagai perantara antara *merchant* dan berbagai *payment gateway* yang berbeda, sehingga pemilik usaha tidak perlu mengelola integrasi secara terpisah. Dengan menggunakan *payment gateway aggregator*, *merchant* dapat menerima berbagai metode pembayaran seperti kartu kredit, transfer bank, dompet digital, atau metode pembayaran lainnya yang ditawarkan oleh berbagai *payment gateway* yang terhubung.

Dalam rangka membangun dan mengembangkan layanan berbayar *online*, peneliti menyarankan kepada pihak CV Himalaya Teknologi untuk mempertimbangkan Gasmaj melakukan integrasi dengan aplikasi *payment gateway aggregator*.

Secara keseluruhan, aplikasi dapat memberikan solusi yang efisien dan terintegrasi, memungkinkan pemilik usaha untuk mengelola dan memperluas pilihan metode pembayaran dengan lebih mudah, mengoptimalkan pengeluaran, dan meningkatkan kepuasan pelanggan.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah diuraikan di atas, maka rumusan masalah pada penelitian ini adalah:

1. Bagaimana penggunaan aplikasi *payment gateway aggregator* dapat meningkatkan efisiensi operasional dalam pengelolaan pembayaran di aplikasi Gasmaj CV Himalaya Teknologi?
2. Bagaimana penggunaan *payment gateway aggregator* dapat memperluas pilihan metode pembayaran dan memenuhi preferensi pembayaran pelanggan?
3. Bagaimana penggunaan *payment gateway aggregator* dapat mengurangi kompleksitas dan waktu yang diperlukan dalam mengelola integrasi *payment gateway* secara terpisah untuk setiap metode pembayaran dalam Gasmaj?

1.3 Tujuan dan Manfaat Penelitian

Berdasarkan pemaparan yang sudah dijelaskan di atas, berikut adalah tujuan dan manfaat dari penelitian ini.

1.3.1 Tujuan Penelitian

Berikut adalah beberapa tujuan penelitian berdasarkan masalah-masalah yang telah diidentifikasi:

1. Menganalisis dampak penggunaan sistem *payment gateway aggregator* terhadap efisiensi operasional perusahaan.
2. Menjelahi cara penggunaan *payment gateway aggregator* dapat memperluas pilihan metode pembayaran dan memenuhi preferensi pembayaran pelanggan.
3. Mengevaluasi bagaimana penggunaan *payment gateway aggregator* dapat mengurangi kompleksitas dan waktu yang diperlukan dalam mengelola integrasi *payment gateway* secara terpisah untuk setiap metode pembayaran.

1.3.2 Manfaat Penelitian

Berikut adalah beberapa manfaat penelitian dari masalah yang telah diidentifikasi di atas:

1.3.2.1 Manfaat Akademis

1. Meningkatkan pemahaman tentang implementasi sistem *payment gateway aggregator* dalam pengelolaan pembayaran aplikasi berbasis bisnis.
2. Memberikan perspektif baru dalam memahami peran *payment gateway aggregator* sebagai solusi efisien dalam pengelolaan pembayaran *online*.
3. Menyediakan pemahaman tentang kepentingan keamanan dan kepercayaan pelanggan dalam sistem pembayaran *online*.

4. Menyelidiki perbedaan antara menggunakan *payment gateway aggregator* dan tidak menggunakan *payment gateway aggregator* dalam hal integrasi *payment gateway* dan manfaatnya.
5. Menyediakan kerangka konseptual yang dapat digunakan sebagai referensi dalam penelitian lanjutan tentang penggunaan *payment gateway aggregator* dalam pengelolaan pembayaran.

1.3.2.2 Manfaat Praktis

Penelitian ini juga memiliki kegunaan bagi pihak-pihak yang mengembangkan atau menggunakan hasil dari penelitian ini, yaitu:

1.3.2.2.1 Bagi Pengembang

1. Hasil dari penelitian dapat digunakan sebagai bahan masukan CV Himalaya Teknologi.
2. Meningkatkan wawasan pengembang terkait implementasi sistem *payment gateway aggregator* dalam pengelolaan pembayaran aplikasi berbasis bisnis.
3. Hasil dari program dan aplikasi yang dibuat akan dapat dikembangkan kembali, sehingga dapat menjadi peluang bisnis bagi pengembang.

1.3.2.2.2 Bagi Perusahaan

1. Dengan menggunakan *payment gateway aggregator*, CV Himalaya Teknologi dapat mengoptimalkan pengeluaran dan meminimalkan biaya yang terkait dengan integrasi *payment gateway*.
2. Dengan adanya pilihan metode pembayaran yang lebih luas melalui *payment gateway aggregator*, CV Himalaya Teknologi dapat memenuhi preferensi pembayaran pelanggan.

3. Dengan menggunakan *payment gateway aggregator*, CV Himalaya Teknologi dapat menawarkan layanan pembayaran yang lebih efisien dan beragam kepada pelanggan.
4. Dengan kemampuan untuk membandingkan tarif dan biaya antara *payment gateway* yang terhubung melalui *payment gateway aggregator*, CV Himalaya Teknologi dapat mengoptimalkan pengeluaran dan memaksimalkan keuntungan bisnisnya.

1.4 Batasan Masalah

Batasan masalah pada penelitian ini adalah:

1. Masalah hanya berfokus pada produk Gasmaj yang akan dikembangkan CV Himalaya Teknologi.
2. Untuk analisa dan perancangan sistem hanya berfokus pada aplikasi *payment gateway aggregator*, sehingga untuk aplikasi Gasmaj tidak akan dibahas secara detail.
3. Pada penelitian ini, *payment gateway* yang akan pertama diintegrasikan adalah Midtrans, Brankas, dan Stripe
4. *Payment gateway aggregator* yang dikembangkan hanya sebagai pihak ketiga pembayaran.
5. Untuk fitur pembayaran, implementasi penelitian dilakukan menggunakan *environment testing* atau *sandbox*.

1.5 Metodologi Penelitian

Dalam upaya penyusunan penulisan ini, diperlukan pengolahan data yang akurat. Oleh karena itu, penulis telah menerapkan beberapa metode pengumpulan data guna memperoleh informasi yang relevan. Beberapa teknik pengumpulan dan pengembangan sistem yang telah digunakan dalam penelitian ini mencakup:

1.5.1 Tahap Pengumpulan Data

Tahapan pengumpulan data akan menggunakan teknik pengumpulan dari studi literatur, observasi, dan wawancara. Studi literatur digunakan untuk mendapatkan teori dan informasi yang terkait dengan penelitian ini. Wawancara digunakan untuk mendapatkan informasi yang lebih mendalam dari responden.

1.5.2 Tahapan Pembuatan Sistem

Berikut adalah tahapan-tahapan dalam penyajian data yang akan diimplementasi pada penelitian ini:

1.5.2.1 Analisis Kebutuhan Pengguna

Melakukan wawancara dengan calon pengguna untuk memahami kebutuhan dan harapan mereka terhadap pengelolaan pembayaran *payment gateway aggregator*. Hal ini dapat dilakukan dengan menggunakan pendekatan kualitatif seperti wawancara mendalam.

1.5.2.2 Analisis Platform *Payment Gateway* yang Tersedia

Melakukan analisis terhadap *platform payment gateway* yang tersedia dan menentukan *platform* yang paling sesuai dengan kebutuhan pengguna. Dalam analisis ini, dapat digunakan metode-metode kuantitatif seperti pemilihan dan penilaian kriteria berdasarkan bobot yang telah ditentukan.

1.5.2.3 Desain Antarmuka Aplikasi

Merancang antarmuka aplikasi yang mudah digunakan dan sesuai dengan kebutuhan pengguna. Pada tahap ini, dapat digunakan metode desain *user experience* (UX) untuk memastikan antarmuka aplikasi yang dihasilkan dapat memberikan pengalaman pengguna yang baik.

1.5.2.4 Implementasi Algoritma Pengelolaan Data Transaksi

Implementasi algoritma yang sesuai untuk meningkatkan efisiensi pengelolaan data transaksi.

1.5.2.5 Uji Coba Aplikasi

Melakukan uji coba aplikasi untuk memastikan kualitas dan efektivitas aplikasi. Uji coba dapat dilakukan dengan menggunakan metode *black box*, *black box testing* adalah pengujian yang dilakukan untuk mengamati hasil input dasn output dari perangkat lunak tanpa mengetahui struktur kode dari perangkat lunak. Pengujian ini dilakukan di akhir pembuatan perangkat lunak untuk mengetahui apakah perangkat lunak dapat berfungsi dengan baik.

1.5.2.6 Evaluasi dan Pengembangan

Melakukan evaluasi terhadap hasil implementasi dan pengembangan aplikasi secara terus-menerus untuk memastikan aplikasi selalu dapat memenuhi kebutuhan pengguna dan memberikan pengalaman pengguna yang baik. Evaluasi dapat dilakukan dengan menggunakan metode *feedback* dari pengguna atau *analytics* data dari penggunaan aplikasi.

1.6 Sistematika Penulisan

Penulisan skripsi ini tersusun dari 5 (lima) bab dengan sistematika penulisan sebagai berikut :

BAB I PENDAHULUAN

Pada bab ini, penulis membahas tentang Latar Belakang Masalah, Rumusan Masalah, Tujuan dan Manfaat Penelitian, Batasan Masalah, Metodologi Penelitian, Sistematika Penulisan, dan Kerangka Pemikiran.

BAB II LANDASAN TEORI

Pada bab ini, akan dijelaskan teori-teori yang menjadi dasar dalam implementasi *payment gateway aggregator* guna meningkatkan efisiensi operasional, memperluas pilihan metode pembayaran, dan mengoptimalkan pengeluaran CV Himalaya Teknologi..

BAB III ANALISA SISTEM

Pada bab ini, akan dilakukan analisis terperinci mengenai sistem *payment gateway aggregator* yang dapat membantu CV Himalaya Teknologi dalam mengatasi tantangan operasional, meningkatkan kepuasan pelanggan, dan memaksimalkan efisiensi pengeluaran.

BAB IV PERANCANGAN SISTEM

Pada bab ini, akan dijelaskan tentang desain arsitektur sistem, integrasi dengan *payment gateway*, pengelolaan metode pembayaran, fitur keamanan, dan antarmuka pengguna.

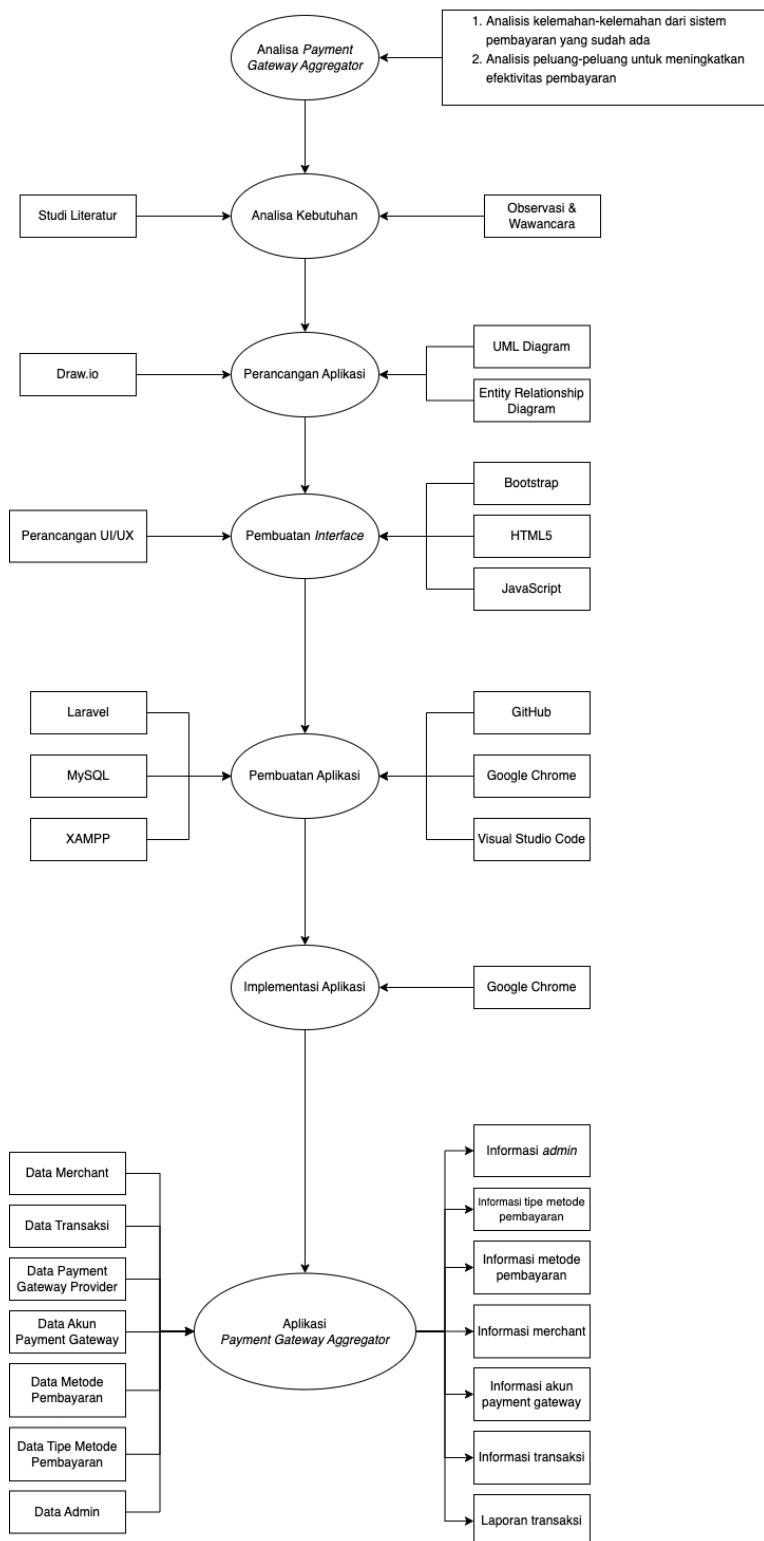
BAB V IMPLEMENTASI

Pada bab ini, akan dijelaskan langkah-langkah yang dilakukan dalam mengimplementasikan aplikasi, termasuk konfigurasi teknis, integrasi dengan *payment gateway* dan uji coba sistem.

BAB VI KESIMPULAN DAN SARAN

Pada bab ini, akan dilakukan analisis terperinci mengenai sistem *payment gateway aggregator* yang dapat membantu CV Himalaya Teknologi dalam mengatasi tantangan operasional, meningkatkan kepuasan pelanggan, dan memaksimalkan efisiensi pengeluaran.

1.7 Kerangka Pemikiran



Gambar 1.1 Kerangka pemikiran

Membangun aplikasi *payment gateway aggregator* untuk produk dilakukan dengan cara melakukan analisa dari sistem pembayaran yang sudah dan berjalan saat ini, yang selanjutnya akan dilanjutkan dengan melakukan analisa peluang-peluang untuk meningkatkan efektivitas pembayaran.

Analisa kebutuhan dilakukan dengan mencari referensi-referensi dari penelitian sebelumnya dan juga melakukan observasi serta wawancara kepada subyek penelitian, yakni pemilik usaha sebagai rujukan rancangan aplikasi yang akan dibuat menggunakan *tools* Draw.io untuk menciptakan UML (*Unified Modeling Language*) dan juga *Entity Relationship Diagram*.

Pada tahap proses pembuatan aplikasi, penulis menggunakan beberapa teknologi antaralain adalah Laravel, MySQL, XAMPP, Github, dan Visual Studio Code, yang selanjutnya dilakukan implementasi untuk melakukan uji coba terhadap aplikasi yang sudah dibuat menggunakan *browser* Google Chrome.

Hasil akhir dari penelitian ini yakni “Aplikasi *Payment Gateway Aggregator*”, dengan masukan data *merchant*, data transaksi, data *payment gateway provider*, data akun *payment gateway*, data metode pembayaran, data tipe metode pembayaran, dan data admin, dengan hasil keluaran informasi admin, informasi metode pembayaran, informasi tipe metode pembayaran, informasi status transaksi, informasi *merchant*, informasi akun *payment gateway*, informasi transaksi, dan laporan transaksi.

BAB II

LANDASAN TEORI

2.1 Pembeli

Pembeli adalah individu atau entitas yang melakukan pembelian atau transaksi untuk memperoleh barang atau jasa dari penjual. Dalam konteks bisnis, pembeli adalah orang atau organisasi yang melakukan pembelian sebagai bagian dari kegiatan komersial. Mereka memainkan peran penting dalam proses ekonomi dengan mempengaruhi permintaan dan penawaran barang dan jasa. pembeli merupakan langkah mencakup perilaku seseorang, beberapa orang dan masyarakat yang terus berubah-ubah [7].

Perilaku pembeli juga dapat dipengaruhi oleh faktor sosial, psikologis, budaya, dan demografis. Sebagai contoh, motivasi pembelian dapat berkaitan dengan kebutuhan dasar, status sosial, pengaruh kelompok, atau preferensi pribadi. Berdasarkan beberapa pengertian dapat diartikan bahwa perilaku pembeli merupakan tindakan perorangan atau kelompok untuk mendapatkan, memakai dan mengolah produk [8].

2.2 *Merchant*

Merchant adalah istilah yang mengacu pada individu, bisnis, atau organisasi yang menjual barang atau jasa kepada pembeli. Mereka adalah pihak yang menyediakan produk atau layanan yang dapat dibeli oleh konsumen atau pembeli bisnis [9].

Merchant bisa beroperasi dalam berbagai bentuk, termasuk toko fisik, toko *online*, pasar elektronik, atau *platform e-commerce*. Mereka dapat menjual berbagai jenis produk atau layanan, mulai dari barang konsumsi sehari-hari hingga barang-barang khusus atau layanan yang spesifik [3].

Merchant bertanggung jawab untuk menyediakan barang atau jasa yang diinginkan oleh pembeli, menjaga kualitas produk atau layanan yang mereka tawarkan, menentukan harga, dan menangani transaksi penjualan. Mereka juga dapat bertanggung jawab untuk memberikan layanan pelanggan, mengelola inventaris, dan mengelola aspek lain dari operasi bisnis mereka [9].

Dalam era digital dan perdagangan elektronik, *merchant* juga dapat mengintegrasikan sistem pembayaran elektronik, menggunakan platform *e-commerce*, atau menjalin kemitraan dengan penyedia layanan pembayaran *online* untuk memudahkan proses transaksi dan pembayaran [3].

Merchant berperan penting dalam ekosistem perdagangan dan memainkan peran yang krusial dalam memenuhi kebutuhan pembeli, memperluas pasar, dan mendorong pertumbuhan ekonomi [3].

Merchant dapat dibagi menjadi beberapa kategori berdasarkan jenis bisnis atau platform yang mereka gunakan. Berikut adalah beberapa kategori umum untuk *merchant*:

1. *Merchant Fisik*

Merchant fisik adalah bisnis yang memiliki toko atau gerai fisik di lokasi tertentu [10]. Mereka biasanya menjual barang langsung kepada pelanggan di toko mereka. Contoh *merchant* fisik termasuk toko ritel, restoran, supermarket, atau toko pakaian.

2. *Merchant online*:

Merchant *online* adalah bisnis yang menjual produk atau jasa secara eksklusif melalui platform *online* [3]. Mereka tidak memiliki toko fisik dan transaksi dilakukan secara elektronik. Contoh *merchant* *online* termasuk toko *online*, *marketplace*, atau penyedia layanan digital.

2.3 Transaksi Online

Transaksi *online* merujuk pada proses pembelian, penjualan, atau pertukaran produk atau layanan yang dilakukan secara elektronik melalui internet [9]. Dalam transaksi *online*, pelanggan dan penjual berinteraksi melalui platform digital untuk memproses pembayaran dan pengiriman barang atau layanan.

Transaksi *online* dapat melibatkan berbagai jenis aktivitas, seperti pembelian produk di toko *online*, pemesanan tiket pesawat atau kereta api, pembayaran tagihan *online*, transfer uang antar rekening bank, dan banyak lagi. Pelanggan dapat mengakses platform *e-commerce* atau aplikasi perbankan melalui perangkat komputer, ponsel pintar, atau tablet untuk melakukan transaksi *online* [9].

Proses transaksi *online* umumnya melibatkan langkah-langkah berikut [3]:

1. Pencarian dan penelusuran, pelanggan mencari produk atau layanan yang mereka inginkan melalui situs web atau aplikasi *e-commerce*.

2. Pemilihan, pelanggan memilih produk atau layanan yang diinginkan dan menambahkannya ke keranjang belanja mereka.
3. Pembayaran, pelanggan memilih metode pembayaran, seperti kartu kredit, transfer bank, atau dompet digital, dan memasukkan informasi yang diperlukan untuk memproses pembayaran.
4. Konfirmasi, setelah pembayaran selesai, pelanggan menerima konfirmasi pembelian dan rincian pesanan melalui email atau notifikasi lainnya.
5. Pengiriman, penjual mengatur pengiriman produk ke alamat yang ditentukan oleh pelanggan atau memberikan akses digital untuk layanan yang dipesan.
6. Evaluasi dan umpan balik, Setelah menerima produk atau layanan, pelanggan dapat memberikan umpan balik atau ulasan mengenai pengalaman transaksi mereka.

2.4 Pembayaran *Online*

Pembayaran *online* mengacu pada proses pembayaran yang dilakukan melalui internet menggunakan metode elektronik. Dalam pembayaran *online*, transaksi keuangan dapat dilakukan melalui berbagai platform, seperti situs web, aplikasi seluler, atau perangkat elektronik lainnya yang terhubung ke internet [14].

Metode pembayaran *online* dapat melibatkan penggunaan kartu kredit atau debit, transfer bank, dompet digital, atau layanan pembayaran *online* seperti PayPal, Alipay, atau Stripe. Pengguna biasanya perlu memasukkan informasi pembayaran mereka, seperti nomor kartu kredit atau rekening bank, untuk mengotorisasi dan melaksanakan transaksi [15].

Keuntungan pembayaran *online* termasuk kemudahan, kecepatan, dan kenyamanan. Pengguna dapat melakukan pembayaran kapan saja dan di mana saja asalkan mereka memiliki akses internet. Ini juga mengurangi kebutuhan untuk membawa uang tunai atau cek fisik, serta memungkinkan transaksi dengan pedagang atau individu di seluruh dunia [14].

Pembayaran *online* mulai menjadi tren yang signifikan seiring dengan berkembangnya internet dan teknologi digital. Meskipun sulit untuk menentukan titik awal pasti dari tren ini, beberapa tonggak penting dalam sejarah pembayaran *online* adalah [15]:

1990-an: Pada akhir 1990-an, dengan munculnya World Wide Web dan penyebaran internet yang semakin luas, beberapa perusahaan mulai menawarkan layanan pembayaran *online* yang mendasar seperti penggunaan kartu kredit melalui situs web [5].

1. Awal 2000-an: Seiring dengan pertumbuhan *e-commerce*, terjadi peningkatan signifikan dalam jumlah transaksi pembayaran *online*. Perusahaan-perusahaan seperti PayPal mulai menawarkan solusi pembayaran yang lebih aman dan nyaman.
2. Pertengahan hingga akhir 2000-an: Perkembangan teknologi dan platform *e-commerce* semakin memfasilitasi pembayaran *online*. Kemunculan dompet digital dan platform pembayaran seperti Alipay di Tiongkok, atau Apple Pay dan Google Wallet, memberikan opsi yang lebih luas bagi konsumen untuk melakukan pembayaran *online*.

3. 2010-an: Pembayaran *online* semakin berkembang dengan adopsi yang lebih luas di berbagai sektor, termasuk ritel, perhotelan, transportasi, dan layanan digital. Munculnya perusahaan *fintech* juga ikut mempercepat pertumbuhan pembayaran *online* dengan inovasi seperti dompet digital, transfer uang *peer-to-peer*, dan teknologi pembayaran baru.

Sejak saat itu, pembayaran *online* terus berkembang dan semakin populer di seluruh dunia. *Trend* ini diperkuat oleh perubahan perilaku konsumen yang semakin mengandalkan transaksi *online*, kenyamanan yang ditawarkan oleh pembayaran *online*, dan kemajuan teknologi yang terus berlanjut [5].

2.5 Integrasi Sistem

Integrasi sistem adalah proses menggabungkan atau menyatukan beberapa sistem yang berbeda menjadi satu kesatuan yang terintegrasi. Hal ini bertujuan untuk meningkatkan efisiensi, produktivitas, dan kualitas dalam operasi bisnis atau organisasi [16].

Integrasi sistem melibatkan berbagai aspek, termasuk perangkat lunak, perangkat keras, dan infrastruktur teknologi informasi. Beberapa contoh integrasi sistem yang umum meliputi [16]:

1. Integrasi Basis Data

Menggabungkan beberapa basis data yang terpisah menjadi satu kesatuan yang terpusat. Hal ini memungkinkan akses yang lebih mudah dan efisien terhadap informasi yang diperlukan.

2. Integrasi Aplikasi

Menghubungkan atau menyatukan beberapa aplikasi perangkat lunak yang berbeda sehingga mereka dapat berinteraksi dan berbagi data antara satu sama lain. Misalnya, mengintegrasikan sistem manajemen keuangan dengan sistem manajemen persediaan untuk mengoptimalkan proses pengadaan dan pembayaran.

3. Integrasi Sistem *Enterprise Resource Planning* (ERP)

Mengintegrasikan semua fungsi bisnis inti seperti manufaktur, keuangan, sumber daya manusia, dan logistik menjadi satu sistem terpadu. Ini memungkinkan informasi yang saling terkait dan pemrosesan yang efisien di seluruh organisasi.

4. Integrasi Aplikasi Web

Menghubungkan aplikasi web yang berbeda agar dapat saling berinteraksi dan berbagi data secara *real-time*. Contohnya adalah mengintegrasikan sistem *e-commerce* dengan sistem manajemen persediaan dan sistem pembayaran.

5. Integrasi *Cloud*

Menggabungkan sistem yang berbasis *cloud* dari berbagai penyedia layanan *cloud* untuk memungkinkan akses dan kolaborasi yang mudah di seluruh platform [5].

2.6 *Payment Gateway*

Payment gateway adalah sebuah sistem yang memfasilitasi dan mengelola proses pembayaran secara elektronik antara penjual (*merchant*) dan pembeli (*customer*) dalam transaksi *online*. *Payment gateway* bertindak sebagai perantara antara berbagai metode pembayaran seperti kartu kredit, kartu debit, transfer bank, dompet digital, dan lain-lain [5].

Fungsi utama *payment gateway* adalah memproses transaksi pembayaran dengan aman dan efisien. Ketika pelanggan melakukan pembelian di situs web atau aplikasi, informasi pembayaran mereka akan dikirim melalui *payment gateway* untuk verifikasi dan *otentikasi*. Setelah itu, *payment gateway* akan melakukan pemrosesan transaksi dengan pihak bank atau penyedia layanan pembayaran terkait. Jika transaksi diterima, pembayaran akan berhasil dan penjual akan menerima konfirmasi pembayaran [15].

Payment gateway juga menyediakan fitur keamanan untuk melindungi data pembayaran pelanggan. Informasi sensitif seperti nomor kartu kredit atau data rekening bank dienkripsi agar tidak dapat diakses oleh pihak yang tidak berwenang. *Payment gateway* juga dapat mengimplementasikan protokol keamanan seperti *Secure Sockets Layer* (SSL) untuk menjaga keamanan transmisi data [5].

Dengan adanya *payment gateway*, transaksi *online* menjadi lebih mudah, aman, dan efisien. *Payment gateway* memungkinkan penjual untuk menerima pembayaran dari berbagai metode pembayaran yang berbeda, sehingga meningkatkan fleksibilitas dan kemudahan bagi pelanggan dalam bertransaksi secara elektronik [14].

Adapun berikut beberapa contoh *payment gateway* yang tersedia antara lain:

1. Midtrans [17]

Midtrans adalah salah satu perusahaan *payment gateway* yang terkemuka di Indonesia. Mereka menyediakan solusi pembayaran *online* yang aman dan terpercaya untuk bisnis *e-commerce* dan layanan digital. Midtrans memungkinkan

penjual untuk menerima pembayaran melalui berbagai metode, seperti kartu kredit, kartu debit, transfer bank, dan dompet digital.

Sebagai *payment gateway*, Midtrans memiliki fitur-fitur yang memudahkan proses pembayaran, seperti integrasi yang mudah dengan platform *e-commerce*, kemampuan untuk mengelola transaksi, pemantauan *real-time*, serta sistem keamanan yang canggih untuk melindungi data pembayaran pelanggan. Midtrans juga menyediakan *dashboard* atau antarmuka yang intuitif bagi penjual untuk melihat dan mengelola riwayat transaksi serta laporan keuangan.

Selain itu, Midtrans juga telah terintegrasi dengan berbagai penyedia layanan finansial, seperti bank-bank lokal di Indonesia dan dompet digital populer, seperti OVO dan Gopay. Hal ini memberikan fleksibilitas kepada penjual dalam menerima pembayaran dari berbagai metode yang digunakan oleh pelanggan.

Dengan menggunakan Midtrans sebagai *payment gateway*, penjual dapat meningkatkan kemudahan, keamanan, dan efisiensi dalam proses pembayaran *online*. Midtrans telah membangun reputasi yang kuat dalam industri pembayaran digital di Indonesia dan menjadi pilihan yang populer bagi banyak bisnis dalam mengelola transaksi pembayaran mereka.

2. Stripe [18]

Stripe adalah sebuah perusahaan *fintech* yang menyediakan layanan *payment gateway* yang populer di tingkat global. Mereka memungkinkan penjual untuk menerima pembayaran secara *online* melalui berbagai metode, termasuk kartu kredit, kartu debit, dan dompet digital.

Salah satu keunggulan utama Stripe adalah kemudahan penggunaannya. Mereka menyediakan integrasi yang mudah dengan platform *e-commerce* dan API yang lengkap untuk pengembang. Hal ini memungkinkan penjual untuk dengan cepat mengintegrasikan Stripe ke dalam toko *online* mereka dan mulai menerima pembayaran dengan sedikit kerumitan teknis.

Stripe juga menonjol dalam hal keamanan dan kepatuhan. Mereka memiliki standar keamanan yang tinggi untuk melindungi data pembayaran pelanggan dan menerapkan langkah-langkah untuk memastikan kepatuhan terhadap peraturan keuangan, seperti Standar Keamanan Data Kartu Pembayaran (PCI DSS).

Selain itu, Stripe menawarkan berbagai fitur tambahan yang berguna, seperti kemampuan untuk mengatur penagihan berulang (*recurring billing*), pembayaran dalam mata uang asing, serta kemampuan untuk melakukan analisis dan pelaporan transaksi secara terperinci.

Dengan menggunakan Stripe sebagai *payment gateway*, penjual dapat menghadirkan pengalaman pembayaran yang lancar, cepat, dan aman bagi pelanggan mereka. Stripe telah menjadi salah satu pilihan yang populer di kalangan bisnis *e-commerce* dan startup teknologi, memberikan solusi pembayaran yang handal dan inovatif di pasar global.

2.7 *Payment Gateway Aggregator*

Payment gateway aggregator adalah sebuah platform atau layanan yang mengintegrasikan berbagai *payment gateway* atau metode pembayaran ke dalam satu antarmuka terpusat [19]. *Payment gateway aggregator* bertindak sebagai perantara antara penjual (*merchant*) dan berbagai *payment gateway* yang berbeda, sehingga penjual tidak perlu mengelola integrasi secara terpisah dengan setiap *payment gateway*.

Fungsi utama *payment gateway aggregator* adalah menyediakan solusi yang terintegrasi untuk menerima pembayaran dari berbagai metode pembayaran yang ditawarkan oleh *payment gateway* yang terhubung. Dengan menggunakan *payment gateway aggregator*, penjual dapat menerima berbagai jenis pembayaran seperti kartu kredit, kartu debit, transfer bank, dompet digital, dan lain-lain, tanpa perlu melakukan integrasi secara terpisah dengan setiap *payment gateway* [6].

Keuntungan menggunakan *payment gateway aggregator* adalah meningkatnya fleksibilitas dalam memenuhi preferensi pembayaran pelanggan. Penjual dapat menawarkan beragam opsi pembayaran kepada pelanggan mereka, tanpa harus terbatas pada satu jenis *payment gateway* atau metode pembayaran tertentu. Hal ini memungkinkan penjual untuk menjangkau lebih banyak pelanggan dan meningkatkan pengalaman berbelanja *online* [19].

Selain itu, *payment gateway aggregator* juga memudahkan penjual dalam mengelola proses pembayaran secara efisien. Penjual hanya perlu mengintegrasikan sistem dengan *payment gateway aggregator*, sehingga mengurangi kompleksitas administratif dan teknis dalam mengelola integrasi dengan setiap *payment gateway*.

secara terpisah. *Payment gateway aggregator* juga dapat menyediakan laporan transaksi yang terpusat, memudahkan penjual dalam melacak dan mengelola pembayaran [20].

Dengan adanya *payment gateway aggregator*, penjual dapat memanfaatkan keuntungan dari berbagai metode pembayaran yang ditawarkan oleh berbagai *payment gateway*, meningkatkan fleksibilitas, efisiensi, dan kepuasan pelanggan dalam proses pembayaran secara *online* [20].

2.8 Hubungan *Payment Gateway* dan *Payment Gateway Aggregator*

Payment gateway aggregator merupakan entitas atau platform yang menggabungkan atau mengagregasi berbagai *payment gateway* atau penyedia layanan pembayaran dalam satu antarmuka tunggal. *Payment gateway aggregator* bertindak sebagai perantara antara bisnis atau organisasi dengan berbagai penyedia *payment gateway* yang berbeda [1].

Dalam konteks ini, *payment gateway aggregator* bertanggung jawab untuk mengintegrasikan dan menyediakan akses ke berbagai *payment gateway* kepada bisnis. Hal ini memungkinkan bisnis untuk menerima pembayaran dari pelanggan melalui berbagai metode pembayaran yang didukung oleh masing-masing *payment gateway* yang terhubung [4].

Dengan menggunakan *payment gateway aggregator*, bisnis tidak perlu melakukan integrasi langsung dengan setiap penyedia *payment gateway* secara terpisah. Sebaliknya, mereka dapat mengintegrasikan *payment gateway aggregator* ke dalam sistem mereka, dan *payment gateway aggregator* akan menangani proses integrasi dan koneksi dengan berbagai *payment gateway* yang terhubung [4].

Dengan menggunakan *payment gateway aggregator*, bisnis dapat mengelola berbagai metode pembayaran, seperti kartu kredit, debit, transfer bank, dan metode pembayaran elektronik lainnya, melalui satu antarmuka tunggal. Hal ini memberikan kemudahan dalam administrasi, pelaporan, dan manajemen transaksi pembayaran bagi bisnis [1].

Jadi, hubungan antara *payment gateway* dan *payment gateway aggregator* adalah bahwa *payment gateway aggregator* menggunakan berbagai *payment gateway* sebagai penyedia layanan pembayaran untuk memfasilitasi proses pembayaran bagi bisnis atau organisasi yang menggunakan platform mereka.

2.9 Penelitian Terdahulu

Berikut adalah tabel mengenai penelitian-penelitian terdahulu yang dijadikan referensi pada penelitian ini.

Tabel 2.1 Penelitian terdahulu

No	Tahun	Nama Penulis	Judul	Masalah	Metode	Hasil
1	2022	Fajrin M, Kautsar I, Aji S	<i>Design and Build Payment Gateways in Sharia-Based E-Commerce (Case Study: Murabahah Ijabqabul.Id Contract)</i>	<i>E-commerce</i> di Indonesia meningkat 5 s.d 10 kali selama pandemi	<i>Agile Development</i>	Penelitian ini menghasilkan sistem <i>payment gateway aggregator</i> yang digunakan untuk e-commerce ijabqabul.id.
2	2019	Puspitasari T, Maulina D	Implementasi <i>Payment Gateway</i> Menggunakan Midtrans Pada Marketplace Travnesia.Com	Pembayaran dilakukan secara manual sehingga kurang efisien	Perancangan: UML, Implementasi, Evaluasi	Digitalisasi memudahkan pembayaran baik dari sisi pengguna dan juga admin Travnesia.com.
3	2021	Sausi J, Mtebe J, Mbelwa J	Evaluating user satisfaction with the <i>e-payment gateway system</i> in Tanzania	Kurang efektif dari sisi administratif, adanya celah untuk admin melakukan korupsi, terbatasnya opsi pembayaran	Perancangan: UML, Implementasi, Evaluasi	Transaksi menjadi lebih efektif, kepercayaan pengguna semakin baik, tertutupnya celah korupsi dikarenakan data dicatat otomatis oleh sistem, banyak pilihan opsi pembayaran
4	2021	M. Hassan, Z. Shukur, and M. Hasan	<i>An Efficient Secure Electronic Payment System for E-Commerce</i>	Kurang efisien dalam pengelolaan transaksi	<i>Study design, Sample size</i>	Pengelolaan transaksi menjadi lebih efisiesn dan aman.

BAB III

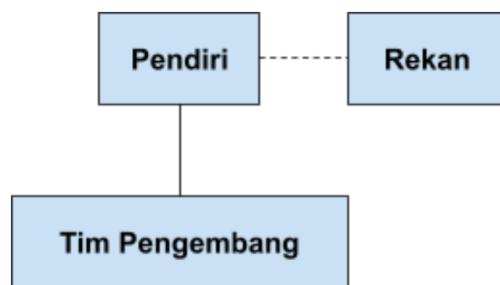
ANALISA SISTEM SEDANG BERJALAN

3.1 Gambaran Umum Perusahaan

3.1.1 Sejarah Perusahaan

CV Himalaya Teknologi didirikan sekitar tahun 2000an yang diawali dengan jasa *service hardware* meliputi komputer, monitor, printer, laptop, hingga *hardware* komputer lainnya. Perusahaan tersebut dibangun oleh Bapak Hendriyatna dan dibantu oleh berbagai pihak mulai dari temannya, toko-toko komputer, perusahaan lain yang saling bekerja sama hingga sekarang. Pada tahun sekitar 2016 perusahaan tersebut mulai bergerak dibidang *software engineering* seiring dengan berkembangnya teknologi seperti *IT consultant*, *website development*, *digital marketing*, dan sebagainya.

3.1.2 Struktur Organisasi Perusahaan



Gambar 3.1 Struktur Organisasi CV Himalaya Teknologi

3.1.3 Uraian Pekerjaan

Berikut adalah uraian pekerjaan dari masing-masing peran di atas:

1. Pendiri

Secara umum, peran Bapak Hendrieatna sebagai pendiri CV Himalaya Teknologi adalah sebagai penggerak utama dalam mendirikan dan mengembangkan perusahaan. Berikut adalah beberapa peran penting yang dimainkan oleh Bapak Hendrieatna:

- a. Pengambil Keputusan
- b. Koordinator dan Kolaborator
- c. Membangun Jaringan dan Kemitraan
- d. Bertanggung Jawab atas Pekerjaan yang Diberikan
- e. Membawa Inovasi dan Pengembangan

2. Tim Pengembang

Tim pengembang dalam CV Himalaya Teknologi merupakan kelompok profesional yang bertanggung jawab dalam merancang, mengembangkan, dan mengimplementasikan solusi perangkat lunak untuk berbagai proyek perusahaan. Berikut adalah peran dan tanggung jawab umum dari tim pengembang:

- a. Analisis Kebutuhan
- b. Perancangan Sistem
- c. Pengembangan Perangkat Lunak
- d. Pengujian dan QA (Quality Assurance)
- e. Kolaborasi Tim

- f. Pemeliharaan dan Pembaruan
- 3. Rekan
 - Dalam konteks CV Himalaya Teknologi, istilah "rekan" dapat merujuk kepada berbagai pihak atau individu yang memberikan dukungan, kerja sama, atau kontribusi dalam berbagai bentuk kepada perusahaan. Berikut adalah beberapa kemungkinan interpretasi dari "rekan" dalam konteks tersebut:
 - a. Teman dan Mitra Bisnis
 - b. Toko-toko Komputer
 - i. Perusahaan Lain yang Bekerja Sama

3.1.4 Bidang dan Kegiatan Perusahaan

CV Himalaya Teknologi beroperasi dalam berbagai bidang pekerjaan yang berkaitan dengan teknologi komputer. Perusahaan menawarkan berbagai layanan dan kegiatan yang meliputi:

1. Perbaikan Perangkat Keras (Hardware)

CV Himalaya Teknologi menyediakan jasa perbaikan perangkat keras komputer seperti komputer, printer, dan laptop. Tim mereka terampil dalam mendiagnosis masalah, mengganti komponen yang rusak, dan memperbaiki kerusakan perangkat keras untuk memulihkan fungsionalitasnya.

2. Instalasi dan Pemeliharaan Perangkat Keras

Perusahaan ini juga menawarkan layanan instalasi perangkat keras, termasuk pemasangan dan konfigurasi komputer, printer, dan perangkat keras lainnya. Selain itu, mereka juga menyediakan layanan pemeliharaan rutin untuk memastikan performa optimal dan keandalan perangkat keras pelanggan.

3. Pembelian Komponen Suku Cadang

CV Himalaya Teknologi membantu pelanggan dalam pembelian komponen suku cadang untuk perangkat keras komputer. Mereka dapat memberikan rekomendasi, membantu dalam pemilihan suku cadang yang sesuai, dan memastikan ketersediaan produk yang dibutuhkan.

4. Konsultasi Teknologi

Perusahaan ini menyediakan layanan konsultasi teknologi kepada pelanggan. Tim mereka memiliki pengetahuan yang luas tentang perkembangan teknologi terkini dan dapat memberikan saran ahli mengenai pemilihan perangkat keras yang tepat, pembaruan teknologi, dan strategi implementasi IT yang efektif.

3.2 Sistem yang Sedang Berjalan

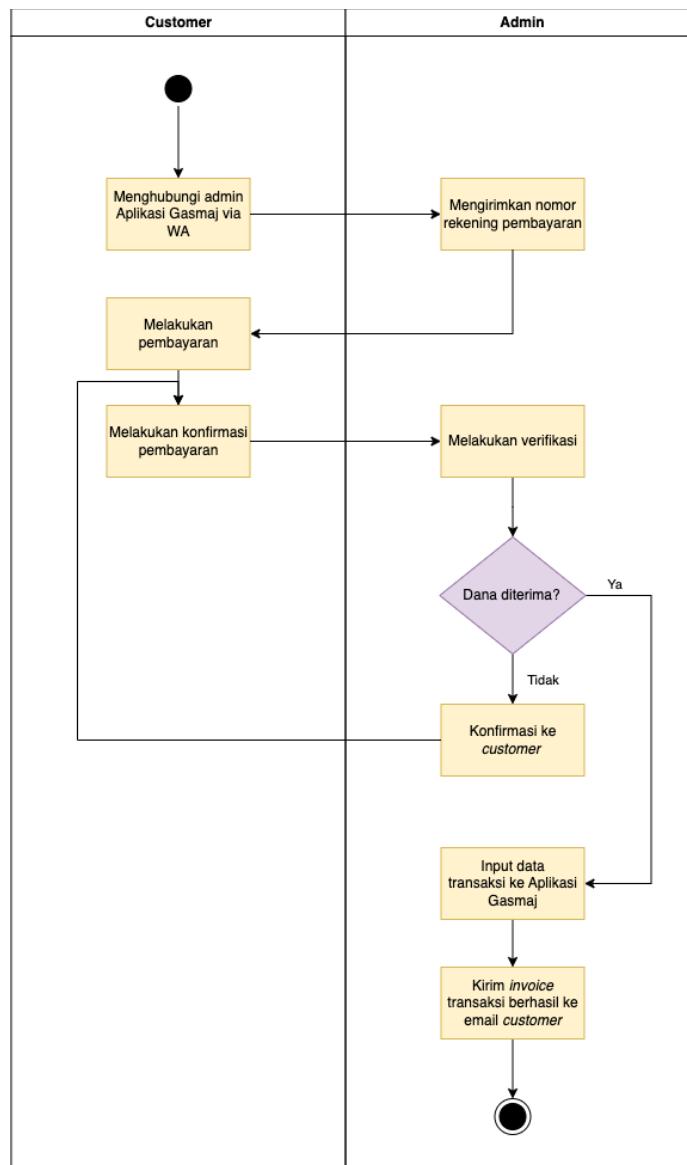
Pada tahun 2023 selain melakukan jasa *service computer*, CV Himalaya Teknologi memiliki proyek situs web Gasmaj, yakni aplikasi untuk manajemen masjid-masjid yang terdiri dari fitur inventaris, akuntansi, produk islam, acara-acara, hingga beberapa jasa yang diterapkan. Proyek inilah yang akan peneliti integrasikan dengan aplikasi *payment gateway* dalam *scope* pembayaran online.

3.2.1 Deskripsi Sistem yang Sedang Berjalan

Alur dalam proyek Gasmaj berawal dari permasalahan di sebuah mushola yang di mana sistem manajemennya masih dilakukan secara manual dan kadang terjadi kesalahan karena *human error* mengingat keterbatasan yang ada. Berdasarkan kasus tersebut, CV Himalaya Teknologi melakukan analisa kebutuhan apa saja yang diperlukan agar dapat mempermudah mushola tersebut menjadi lebih baik dan bahkan dapat diimplementasikan ke mushola lain hingga masjid.

Selain itu, mushola atau masjid yang akan terintegrasi dengan Gasmaj, juga bisa menerima zakat dan infaq, atau pembelian produk islam lainnya seperti qurban. Saat ini semua pembayaran dilakukan secara *offline*, sehingga dianggap kurang efisien.

3.2.2 Workflow Sistem yang Sedang Berjalan



Gambar 3.2 Workflow sistem yang berjalan saat ini

Workflow sistem yang sedang berjalan pada CV Himalaya Teknologi adalah sebagai berikut:

1. Untuk bertransaksi pada Aplikasi Gasmaj, pengguna harus menghubungi *admin* Aplikasi Gasmaj.
2. Pihak gasmaj akan mengirimkan nomor rekening pembayaran yang tersedia.
3. *Customer* melakukan pembayaran pada nomor rekening yang telah diterima.
4. *Customer* melakukan konfirmasi pembayaran ke *admin* masjid.
5. *Admin* akan melakukan verifikasi data pembayaran terlebih dahulu.
6. Jika transaksi dinyatakan sesuai dan dana sudah diterima, *admin* akan memasukkan data transaksi ke dalam catatan.
7. *Terakhir, admin akan mengirimkan invoice* transaksi berhasil ke *customer*.

3.3 Analisis Sistem

3.3.1 Analisis Permasalahan

Dalam analisis permasalahan, terdapat beberapa kendala yang dihadapi oleh CV Himalaya Teknologi, yaitu:

1. Pengelolaan proses pembayaran secara manual memakan waktu dan sumber daya.
2. Opsi metode pembayaran sedikit, sehingga mengurangi fleksibilitas dalam memenuhi preferensi pembayaran pelanggan.

3.3.2 Analisis Kebutuhan

Dalam analisis kebutuhan, CV Himalaya Teknologi perlu mempertimbangkan hal-hal berikut:

1. Mengintegrasikan sistem pembayaran yang terintegrasi, untuk mengelola transaksi pembayaran.
2. Menyediakan variasi metode pembayaran yang dapat memenuhi preferensi pelanggan.

3.3.3 Solusi Permasalahan

Solusi menggunakan Aplikasi *Payment Gateway Aggregator* dapat membantu CV Himalaya Teknologi untuk:

1. Mengelola proses pembayaran secara otomatis, menghindari keterlambatan dan kesalahan manusia dalam verifikasi dan konfirmasi pembayaran.
2. Menerima pembayaran melalui berbagai metode pembayaran seperti kartu kredit, kartu debit, transfer bank, dan dompet digital.

BAB IV

PERANCANGAN SISTEM

4.1 Pendahuluan

Pada bagian ini, akan dijelaskan mengenai tujuan dan ruang lingkup sistem perangkat lunak yang terkait dengan penggunaan *payment gateway aggregator* dalam konteks aplikasi Gasmaj. Berikut ini adalah penjelasan mengenai tujuan dan ruang lingkup sistem perangkat lunak yang akan dirancang:

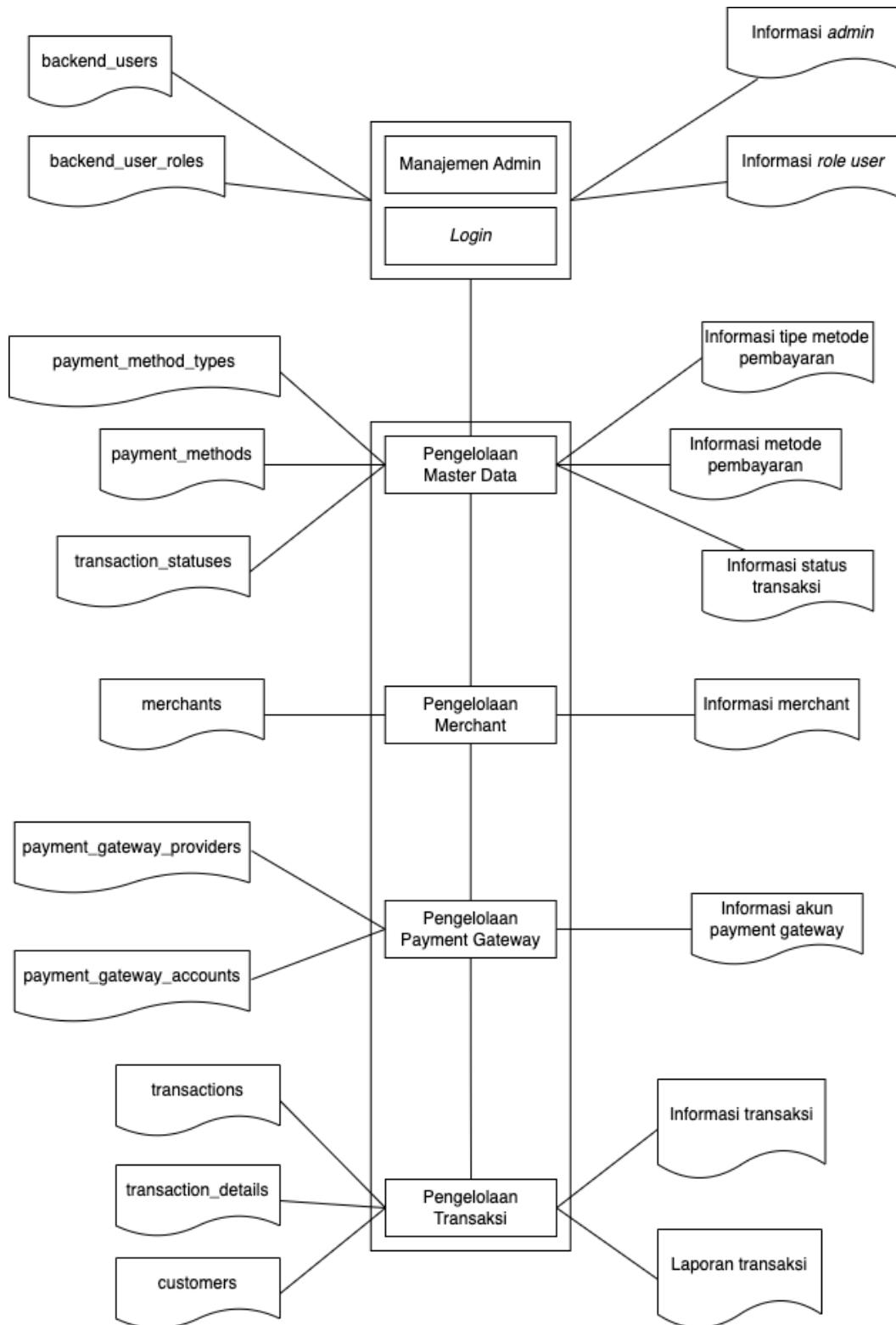
1. Tujuan Sistem Perangkat Lunak
 - a. Meningkatkan efisiensi pembayaran. Sistem perangkat lunak ini bertujuan untuk menggantikan metode konfirmasi pembayaran manual yang dilakukan melalui WhatsApp dengan integrasi *payment gateway aggregator*. Dengan menggunakan *payment gateway aggregator*, diharapkan proses pembayaran menjadi lebih efisien, cepat, dan otomatis.
 - b. Memperluas pilihan metode pembayaran. Sistem perangkat lunak ini akan menyediakan berbagai opsi metode pembayaran yang dapat dipilih oleh pengguna, seperti kartu kredit, transfer bank, dompet digital, atau metode pembayaran lainnya yang disediakan oleh *payment gateway aggregator* yang terintegrasi. Hal ini bertujuan untuk memenuhi preferensi pembayaran pelanggan dengan memberikan pilihan yang lebih fleksibel.

- c. Meningkatkan keamanan transaksi. Dengan menggunakan *payment gateway aggregator* yang terpercaya dan memiliki standar keamanan yang tinggi, sistem perangkat lunak ini akan meningkatkan keamanan transaksi pembayaran yang dilakukan oleh pengguna. Informasi pembayaran akan dienkripsi dan disimpan secara aman, mengurangi risiko kehilangan atau penyalahgunaan data pembayaran.
 - d. Meningkatkan pengalaman pengguna. Sistem perangkat lunak ini akan dirancang dengan antarmuka pengguna yang intuitif dan responsif, sehingga pengguna dapat dengan mudah melakukan pembayaran melalui berbagai metode yang disediakan oleh *payment gateway aggregator*. Hal ini akan meningkatkan pengalaman pengguna dalam menggunakan aplikasi Gasmaj dan melakukan transaksi pembayaran.
2. Ruang Lingkup Sistem Perangkat Lunak

Ruang lingkup sistem perangkat lunak yang mencakup *payment gateway aggregator* dalam aplikasi Gasmaj meliputi integrasi dengan *payment gateway aggregator*. Sistem perangkat lunak ini akan dirancang untuk melakukan integrasi dengan *payment gateway aggregator* yang dipilih. Integrasi ini akan memungkinkan aplikasi Gasmaj untuk menerima pembayaran melalui berbagai metode pembayaran yang disediakan oleh *payment gateway aggregator* tersebut.

4.2 Arsitektur Perangkat Lunak

Arsitektur perangkat lunak adalah representasi struktur sistem perangkat lunak yang akan dibangun, termasuk komponen-komponen yang ada, hubungan antara komponen, serta cara komponen-komponen tersebut berinteraksi. Dalam konteks sistem perangkat lunak yang mencakup *payment gateway aggregator* dalam aplikasi , berikut adalah arsitektur perangkat lunak yang direkomendasikan



Gambar 4.1 Diagram Asitektur Perangkat Lunak

Dari gambar arsitektur perangkat lunak untuk aplikasi ini, data yang diproses yaitu:

1. **Input**, yaitu data-data yang di masukan dalam aplikasi berupa data:

- *backend_users*
- *backend_user_roles*
- *payment_method_types*
- *payment_methods*
- *transaction_statuses*
- *merchants*
- *payment_gateway_providers*
- *payment_gateway_accounts*
- *transactions*
- *transaction_details*

2. **Proses**, yaitu proses dari data yang telah *diinput*, meliputi:

- *Management User* dan *Login*, untuk mengelola data user.
- *Merchant*, untuk mengelola data *merchant*, data ini akan dijadikan sebagai referensi autentikasi ke aplikasi pada saat melakukan proses transaksi.
- *Payment Gateway*, untuk mengelola data berkaitan dengan kredensial *payment gateway*. Data ini akan dijadikan sebagai referensi autentikasi ke *payment gateway* terkait untuk melakukan proses transaksi.

3. **Output**, yaitu data yang ditampilkan setelah diproses yang meliputi:

- Informasi *user*
- Informasi *role user*
- Informasi tipe metode pembayaran
- Informasi metode pembayaran
- Informasi status transaksi
- Informasi *merchant*
- Informasi *payment gateway*
- Informasi akun *payment gateway*
- Informasi transaksi
- Laporan transaksi

4.3 Pemodelan Sistem / UML

4.3.1 Use Case Diagram

Pada Aplikasi *payment gateway aggregator* yang akan dikembangkan terdapat 3 jenis pengguna aplikasi, masing-masing aktor memiliki perannya masing-masing, berikut penjelasan dari masing-masing aktor:

a. Pengguna Admin / *User Admin*

- Memiliki hak akses untuk membuka halaman *dashboard*.
- Memiliki hak akses untuk mengelola data transaksi.
- Memiliki hak akses untuk mengelola data akun *payment gateway* yang terhubung.
- Memiliki hak akses untuk mengelola data atau profil *merchant*.

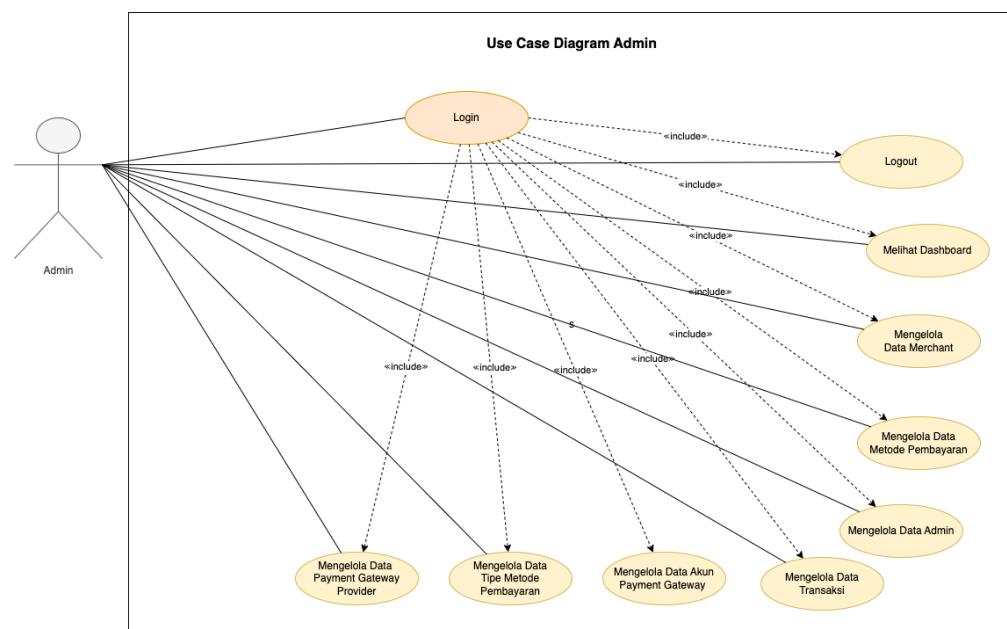
b. Pengguna Merchant / User Merchant

- Memiliki hak akses untuk membuka halaman *dashboard*.
- Memiliki hak akses untuk mengelola data transaksi.
- Memiliki hak akses untuk mengelola data akun *payment gateway* yang terhubung.
- Memiliki hak akses untuk mengelola data atau profil *merchant*.

c. Pelanggan / Customer

- Dapat memilih metode pembayaran yang diinginkan
- Dapat melakukan pembayaran terhadap transaksi yang dilakukan.

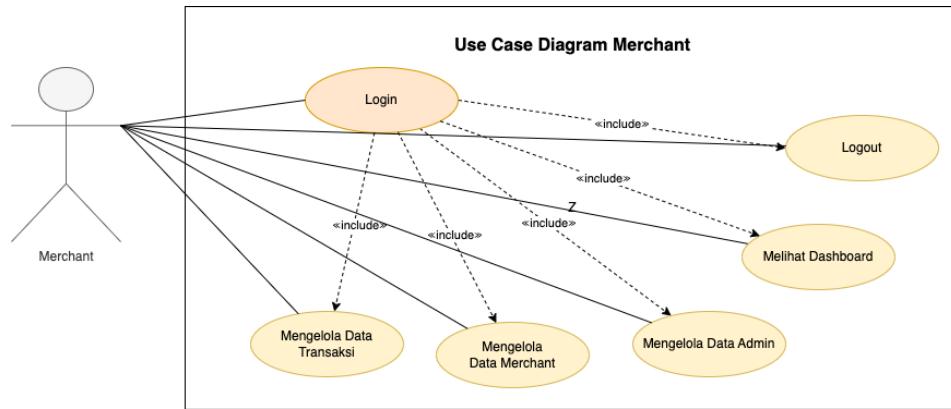
4.3.1.1 Use Case Diagram Admin



Gambar 4.2 Use Case Diagram Admin

Gambar di atas merupakan *use case diagram* pengguna Admin, pengguna Admin memiliki hak akses tertinggi di aplikasi.

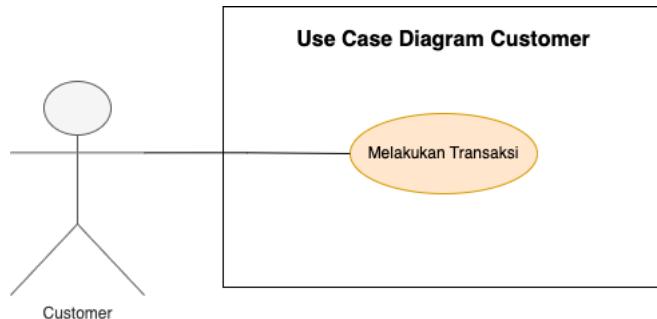
4.3.1.2 Use Case Diagram Merchant



Gambar 4.3 Use Case Diagram Merchant

Gambar di atas merupakan *use case diagram* pengguna *merchant*, pengguna *merchant* merupakan pengguna dengan hak akses yang lebih terbatas dari pengguna *Admin*.

4.3.1.3 Use Case Diagram Customer



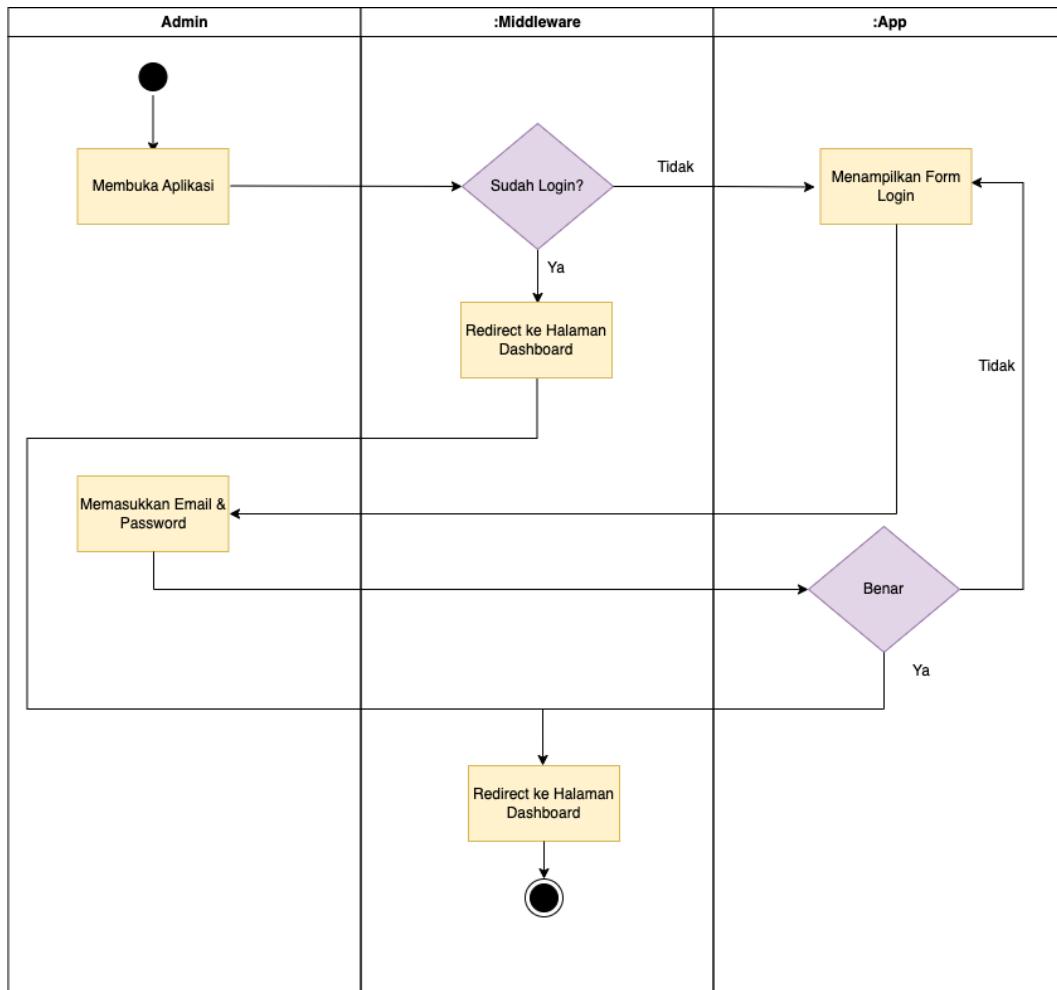
Gambar 4.4 Use Case Customer / Pelanggan

Gambar di atas merupakan *use case diagram* pelanggan atau *customer*, pelanggan tidak diwajibkan *login* ke aplikasi karena aplikasi *payment gateway aggregator* hanya sebagai pihak ketiga untuk melakukan pembayaran.

4.3.2 Activity Diagram

Berikut adalah *activity diagram* yang akan dikembangkan pada “Aplikasi Payment Gateway Aggregator”:

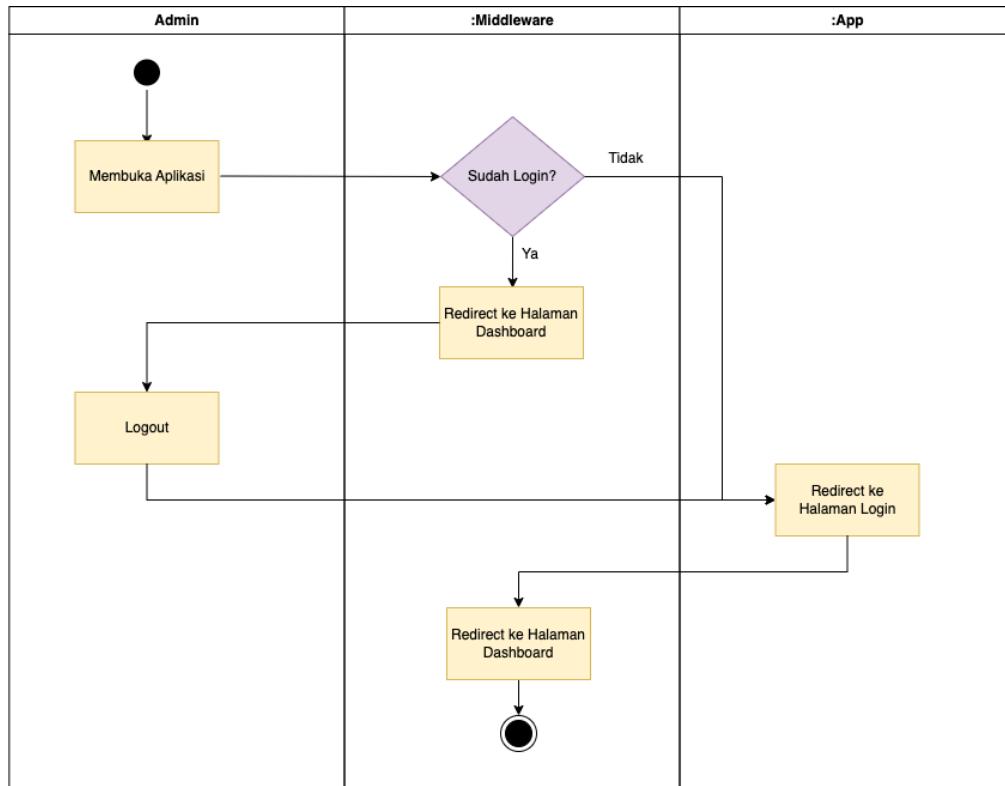
4.3.2.1 Login



Gambar 4.5 Activity Diagram Admin Login

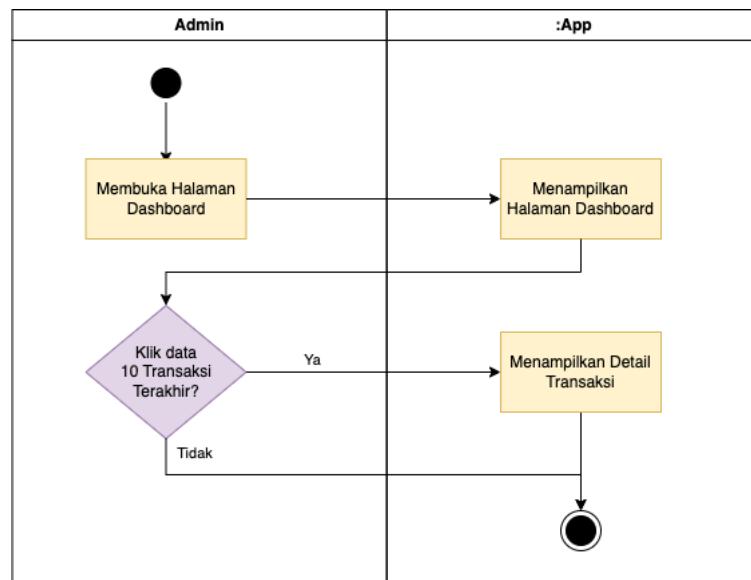
Gambar di atas adalah alur aktivitas yang dilakukan oleh pengguna Admin untuk melakukan *login* ke dalam aplikasi internal *payment gateway aggregator*.

4.3.2.2 Logout



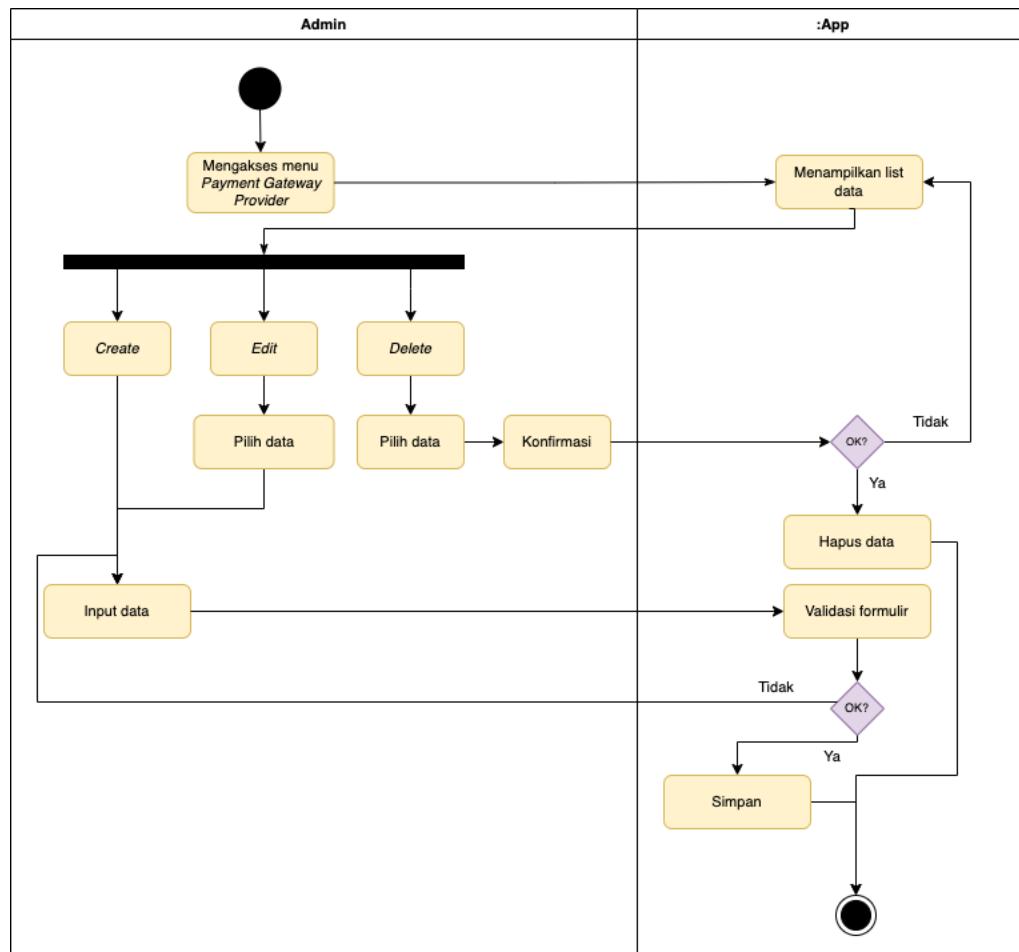
Gambar 4.6 Activity Diagram Logout

4.3.2.3 Melihat Dashboard



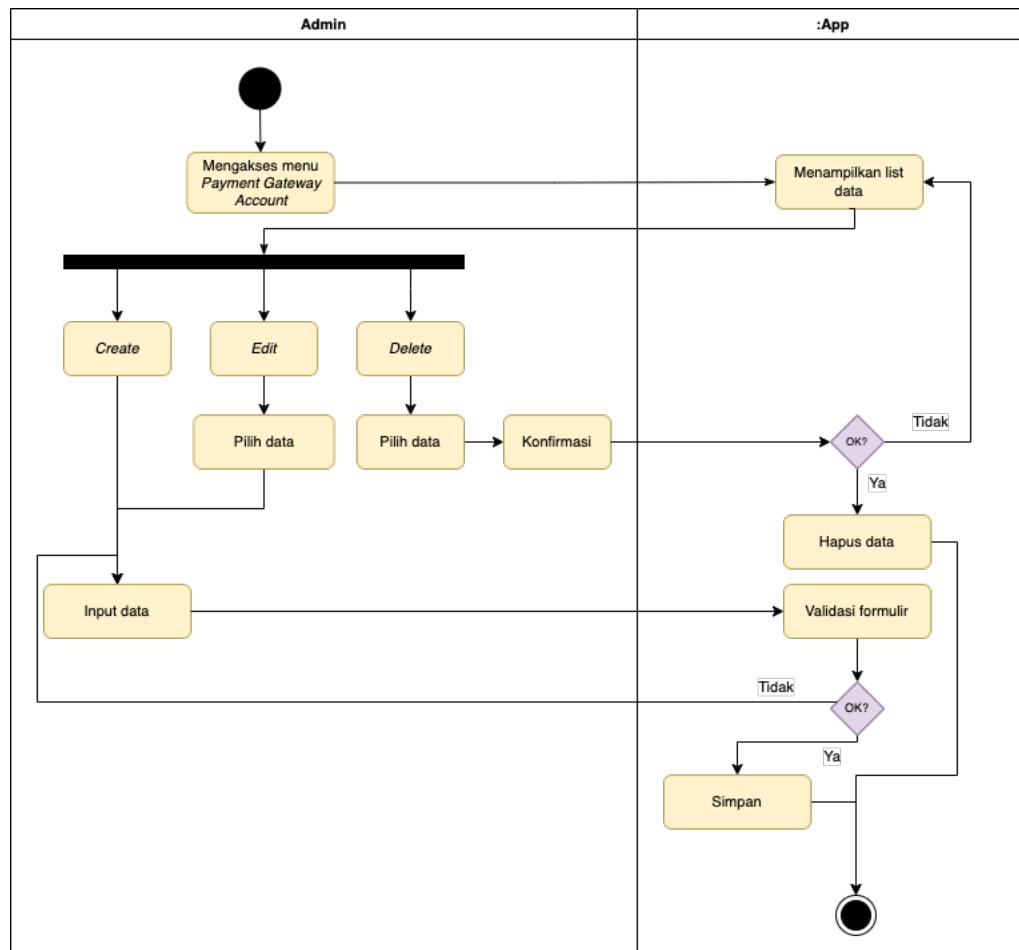
Gambar 4.7 Melihat Dashboard

4.3.2.4 Mengelola Data *Payment Gateway Provider*



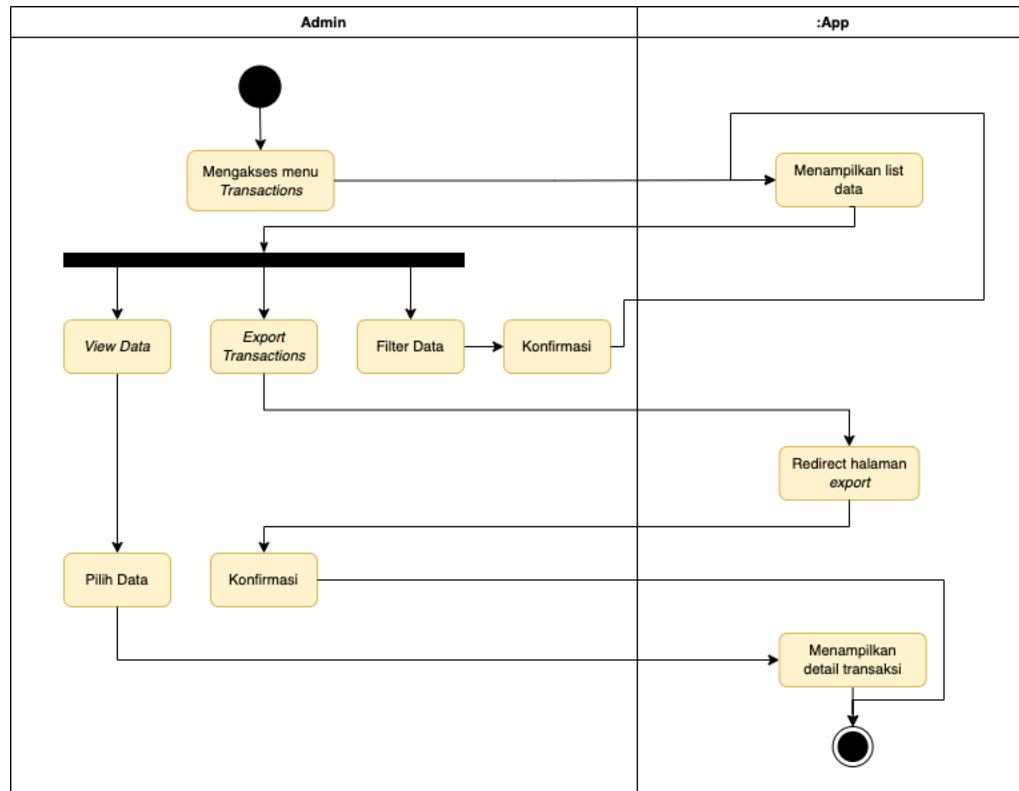
Gambar 4.8 Activity Diagram Mengelola Data Provider *Payment Gateway*

4.3.2.5 Mengelola Data Akun *Payment Gateway*



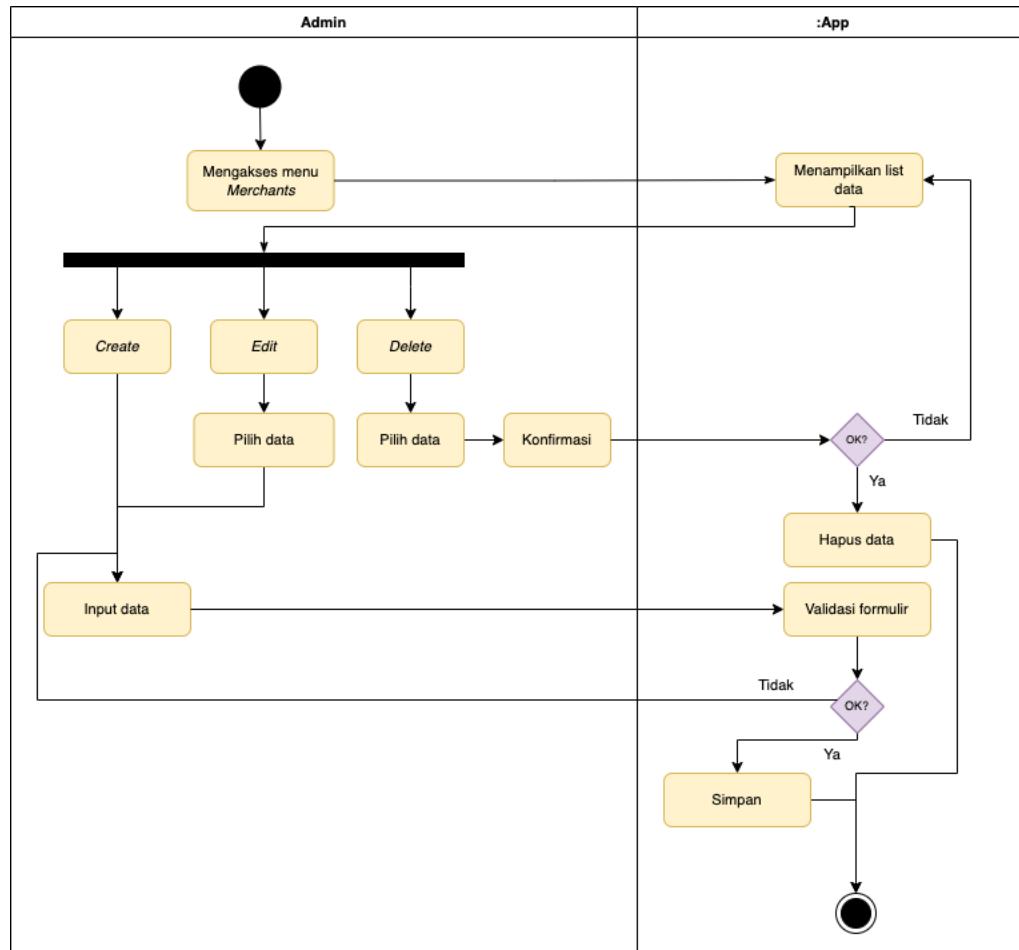
Gambar 4.9 Activity Diagram Mengelola Data Akun *Payment Gateway*

4.3.2.6 Mengelola Data Transaksi



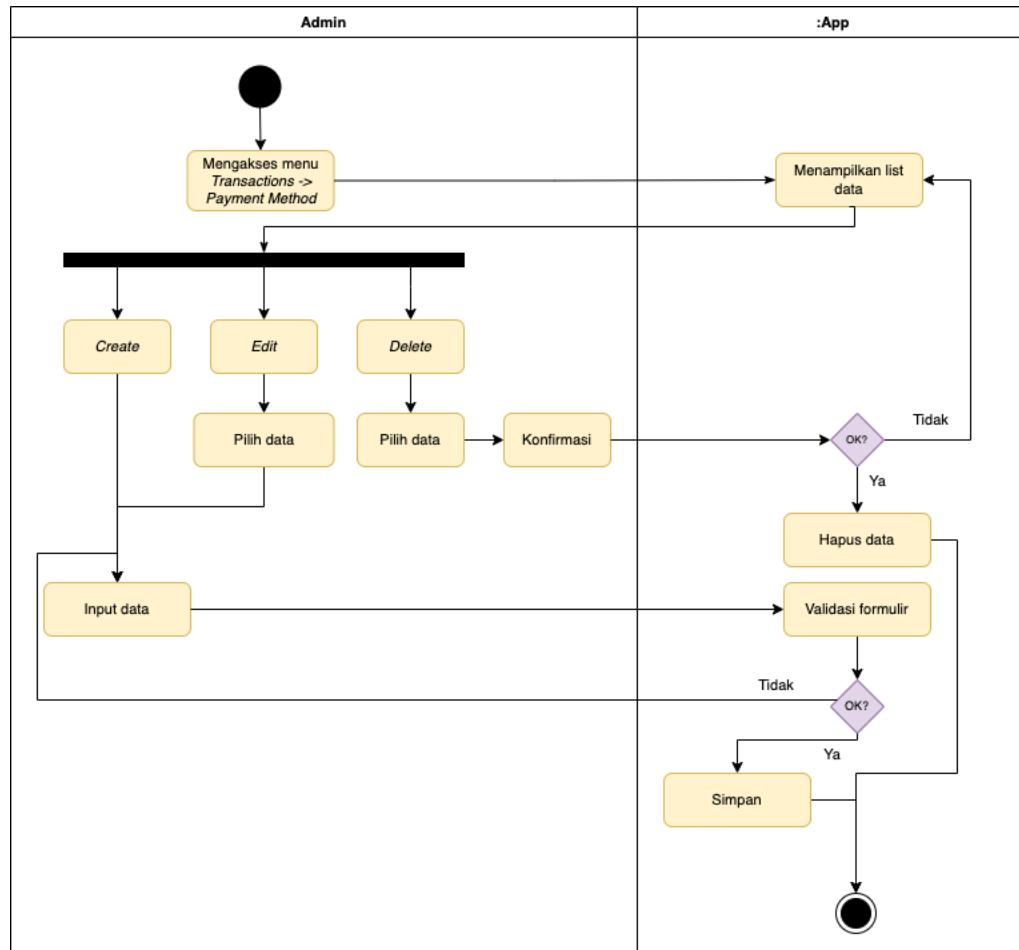
Gambar 4.10 Activity Diagram Mengelola Data Transaksi

4.3.2.7 Mengelola Data Merchant



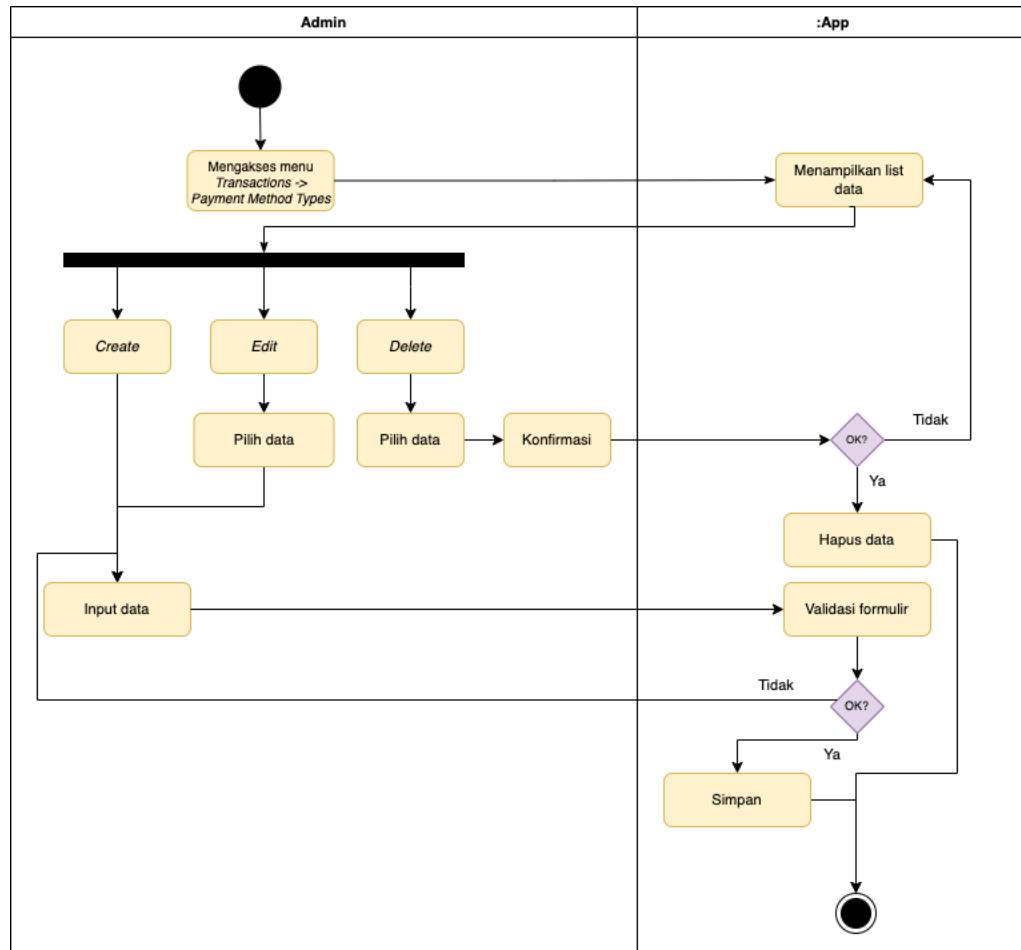
Gambar 4.11 Activity Diagram Mengelola Data Merchant

4.3.2.8 Mengelola Data Metode Pembayaran



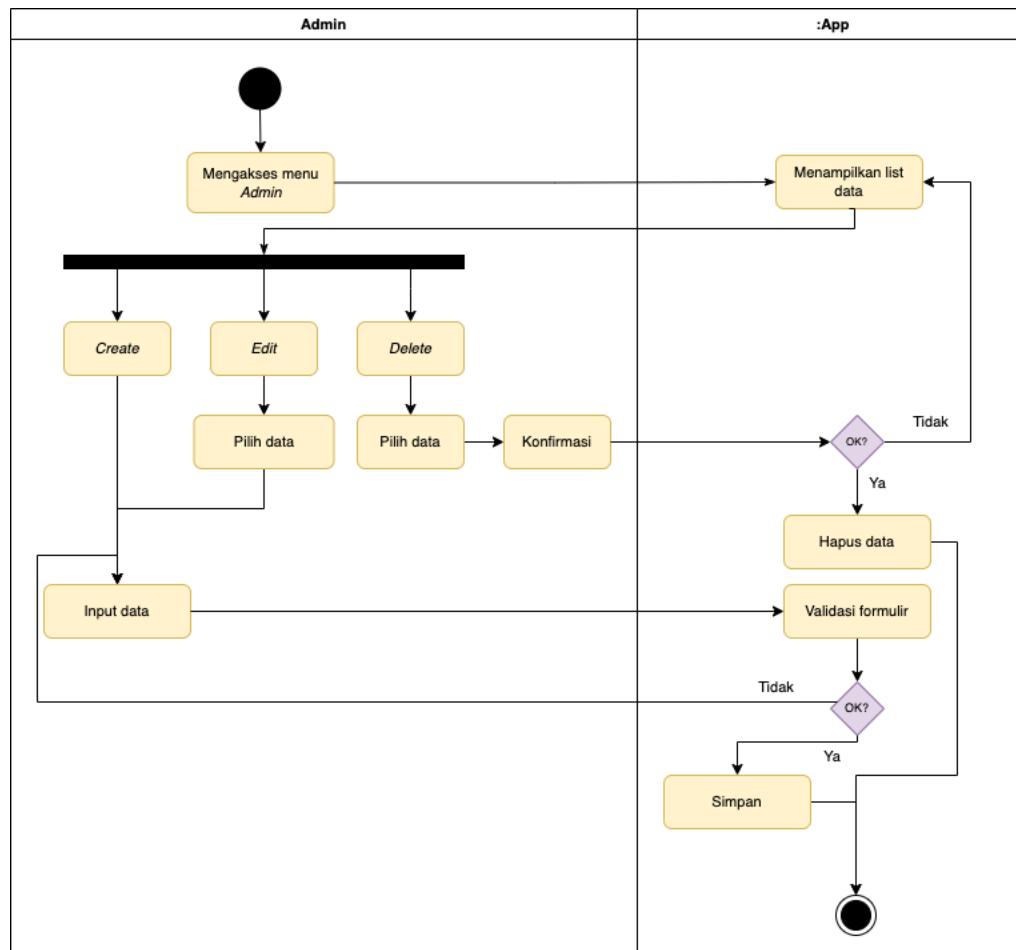
Gambar 4.12 Acitivity Diagram Mengelola Data Metode Pembayaran

4.3.2.9 Mengelola Data Tipe Metode Pembayaran



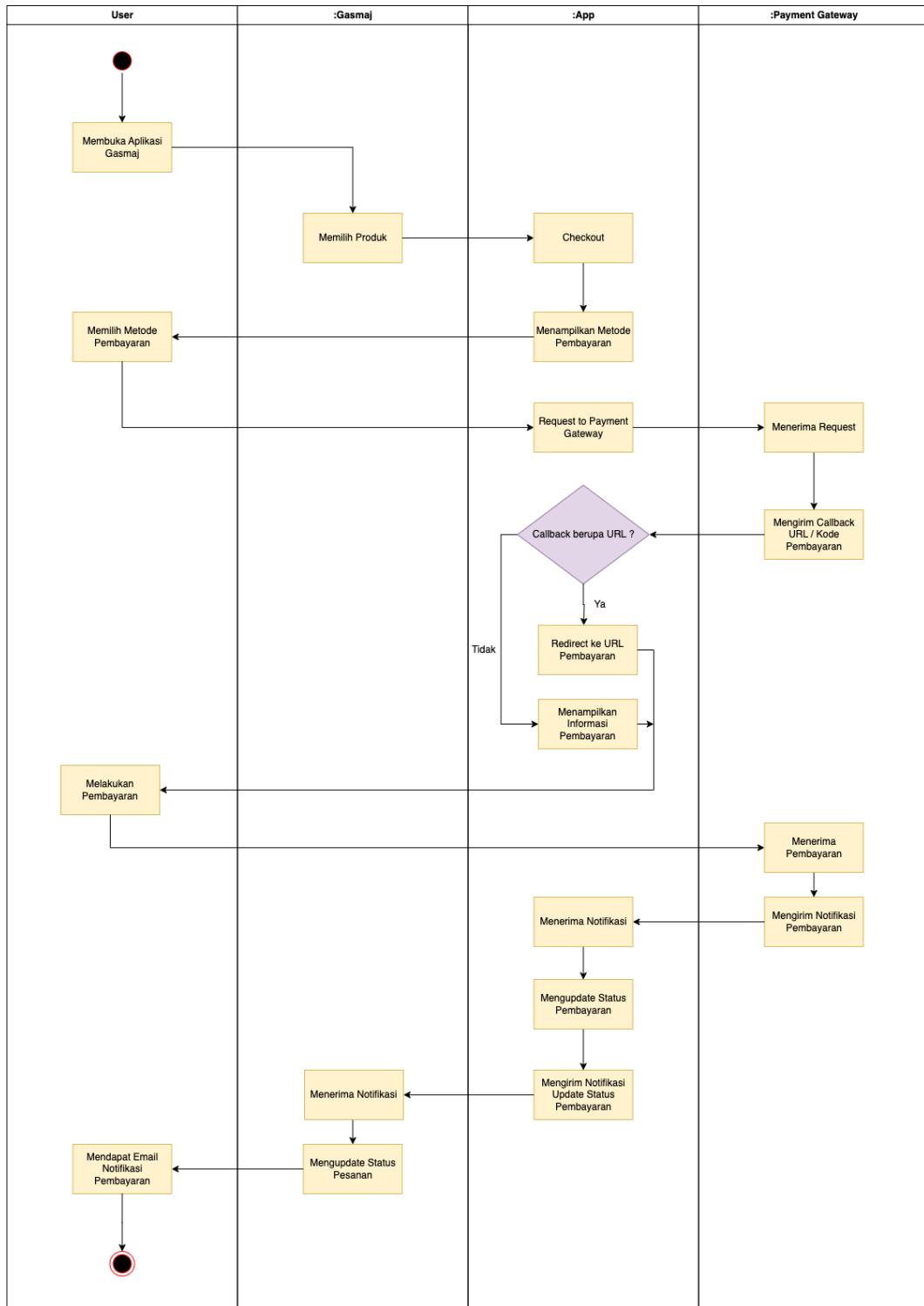
Gambar 4.13 Activity Diagram Mengelola Data Tipe Metode Pembayaran

4.3.2.10 Mengelola Data Admin



Gambar 4.14 Activity Diagram Mengelola Data Admin

4.3.2.11 Melakukan Transaksi

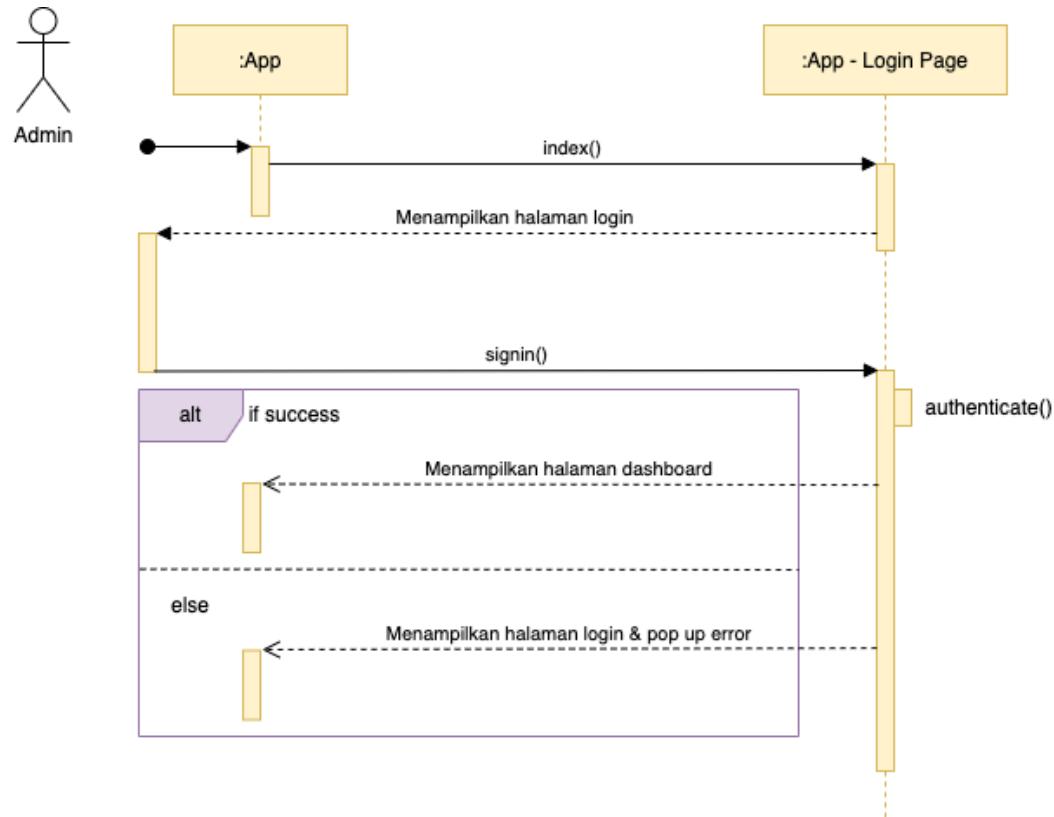


Gambar 4.15 Activity Diagram Melakukan Transaksi

4.3.3 Sequence Diagram

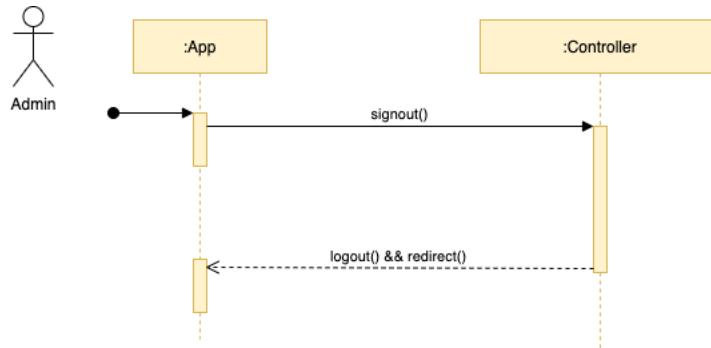
Berikut adalah *sequence diagram* yang akan dikembangkan pada Aplikasi *Payment Gateway Aggregator*:

4.3.3.1 Login



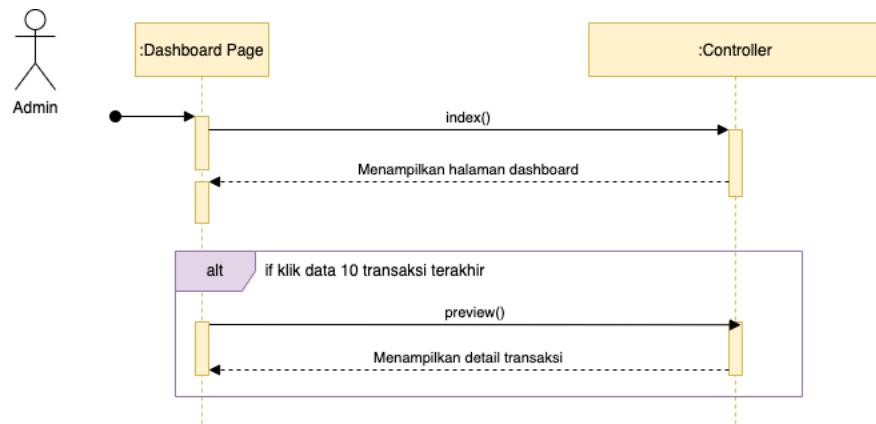
Gambar 4.16 Sequence Diagram Login

4.3.3.2 Logout



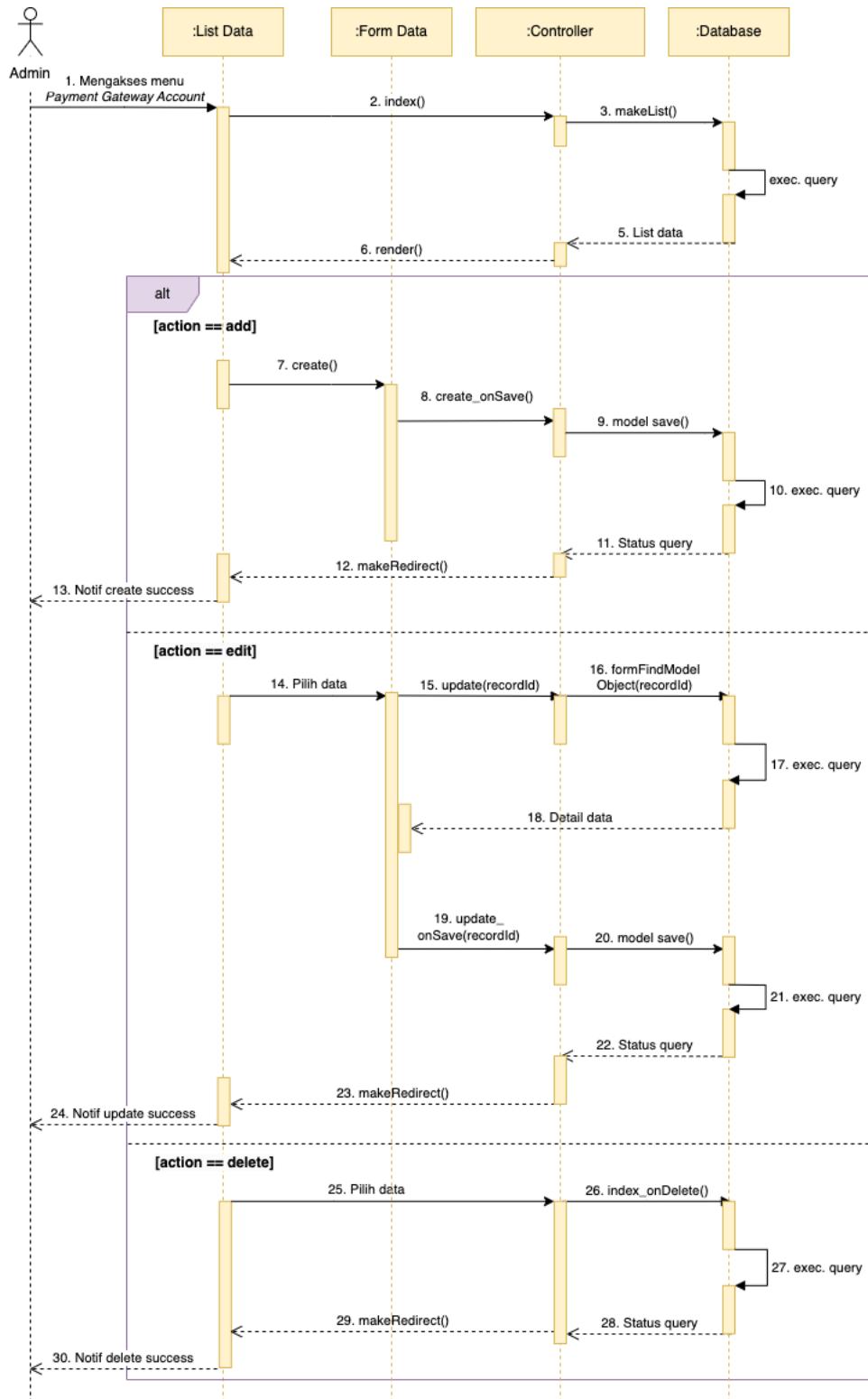
Gambar 4.17 Sequence Diagram Logout

4.3.3.3 Melihat Dashboard



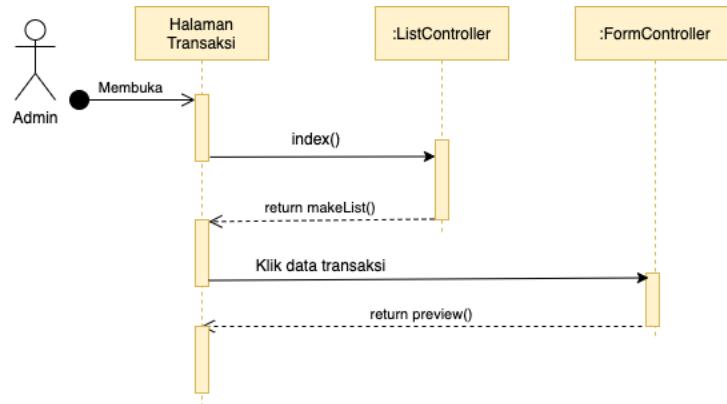
Gambar 4.18 Sequence Diagram Melihat Dashboard

4.3.3.4 Mengelola Data Akun *Payment Gateway*



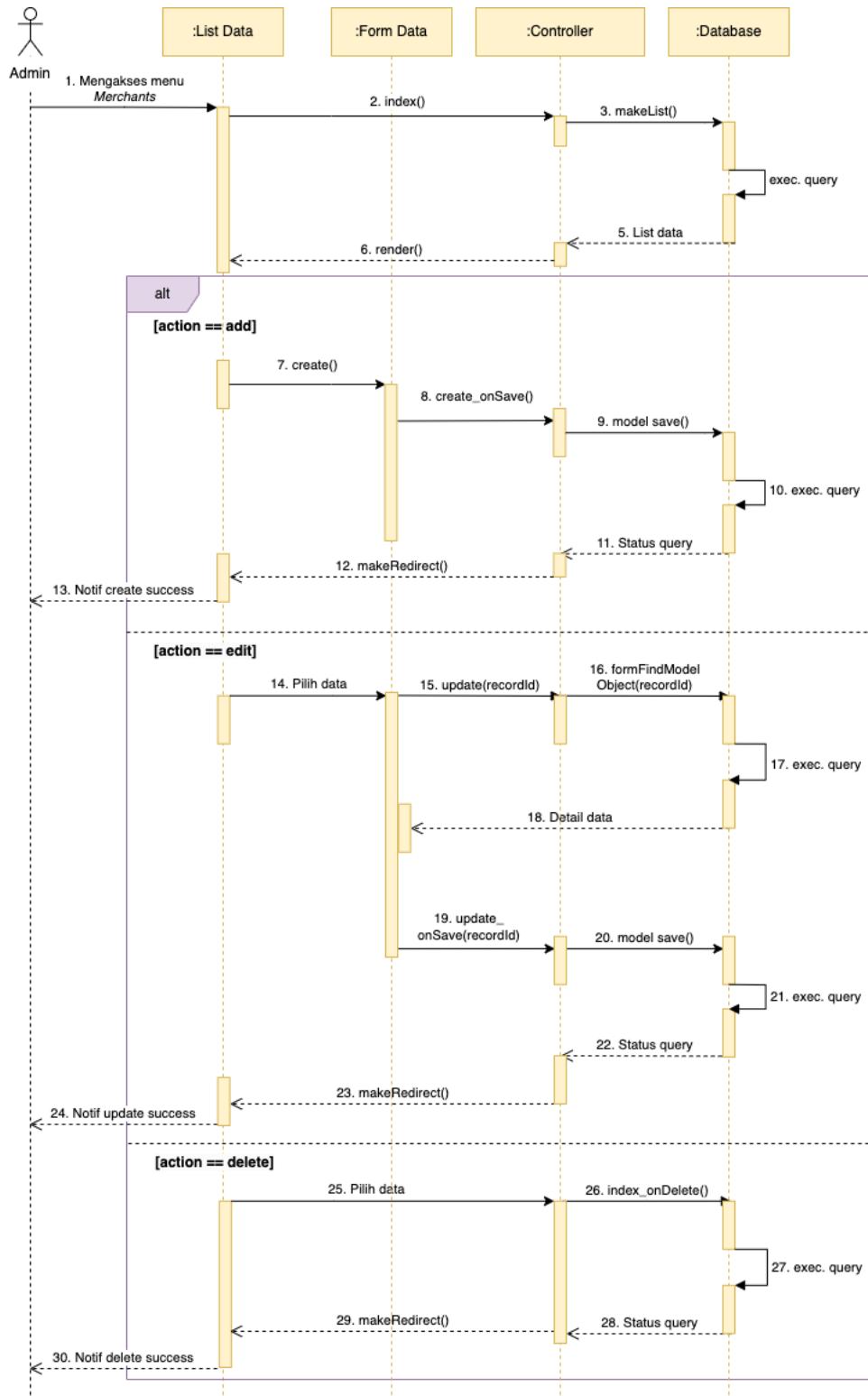
Gambar 4.19 Sequence Diagram Mengelola Data Akun *Payment Gateway*

4.3.3.5 Mengelola Data Transaksi



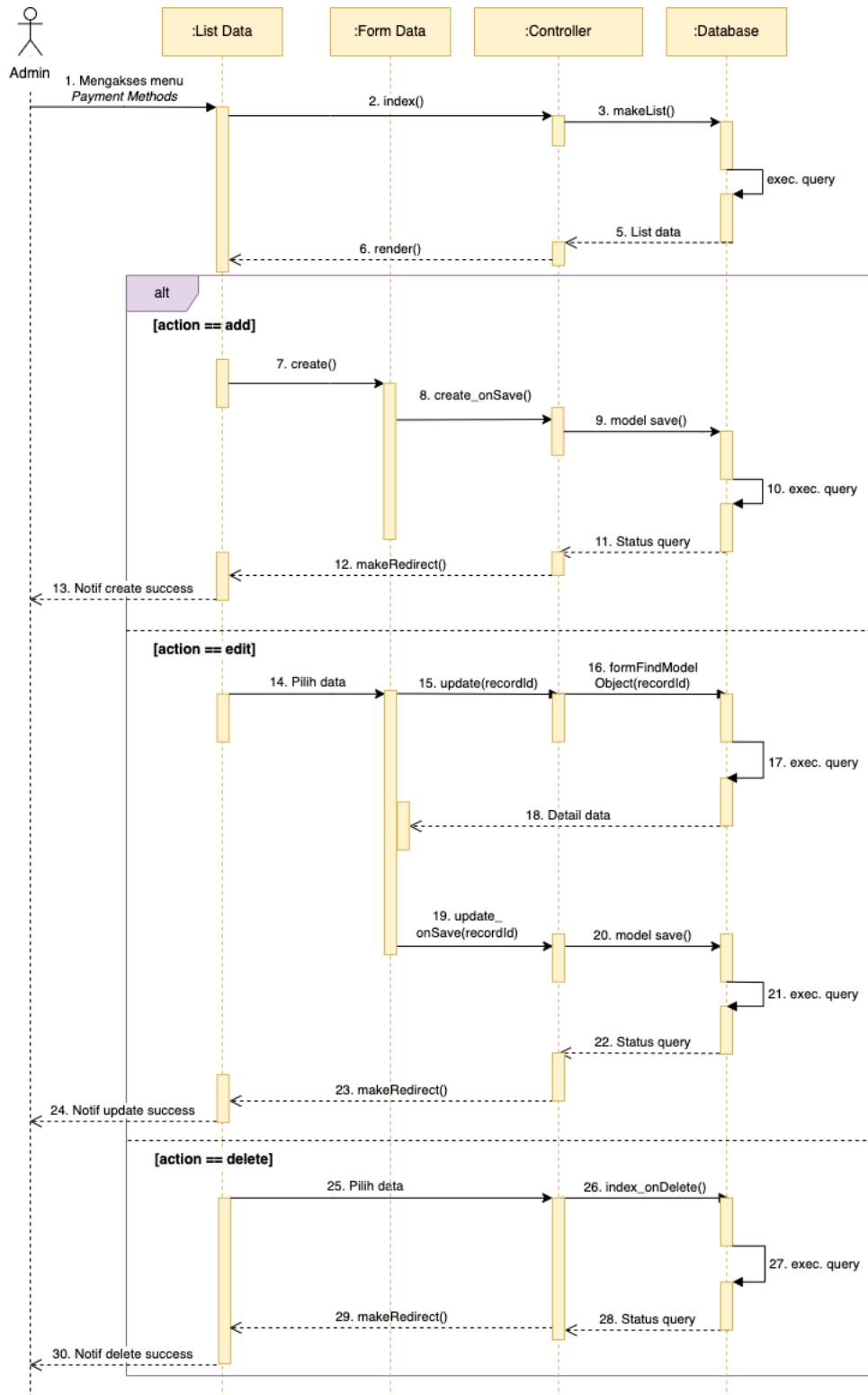
Gambar 4.20 Sequence Diagram Mengelola Data Transaksi

4.3.3.6 Mengelola Data Merchant



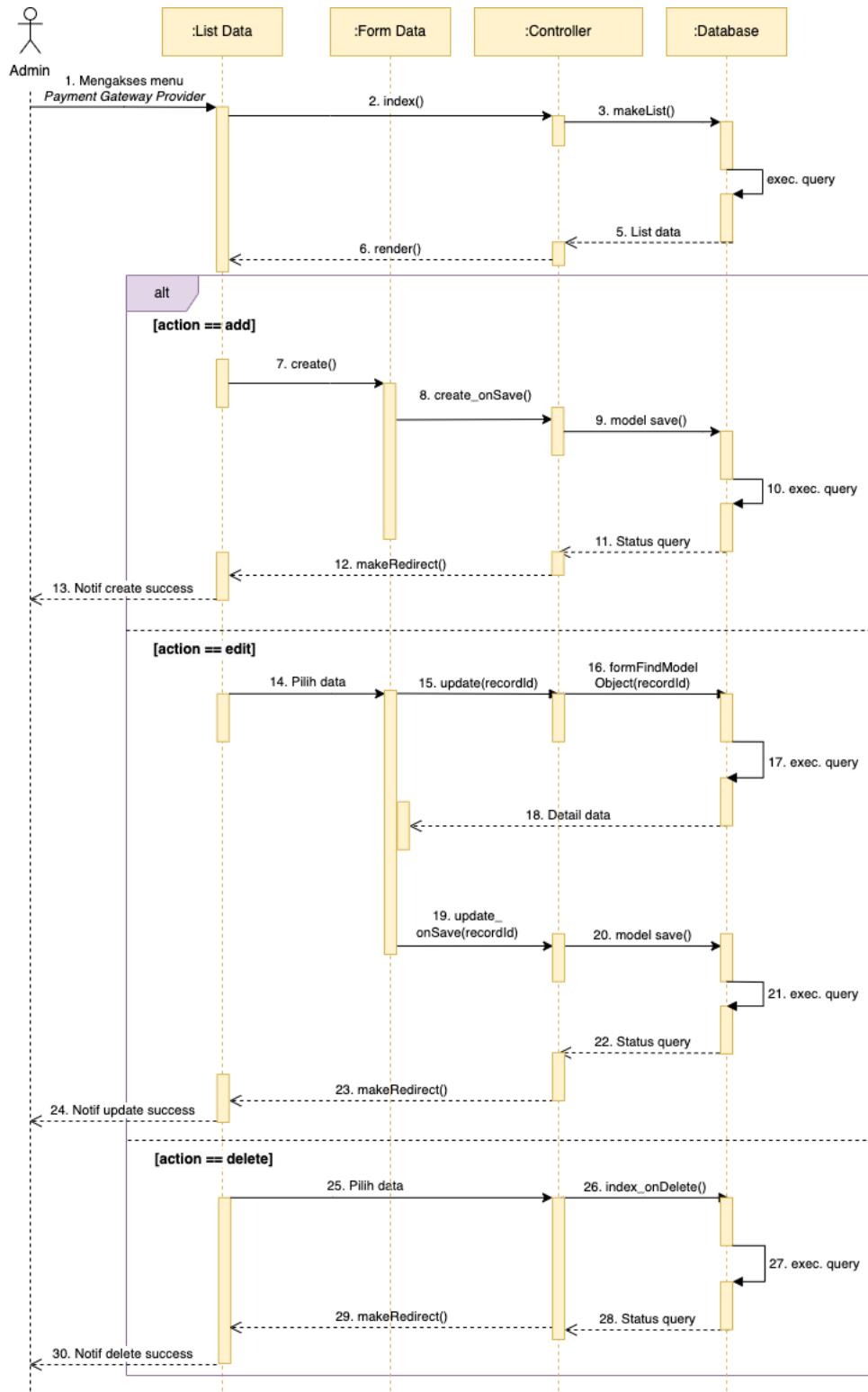
Gambar 4.21 Sequence Diagram Mengelola Data Merchant

4.3.3.7 Mengelola Data Metode Pembayaran



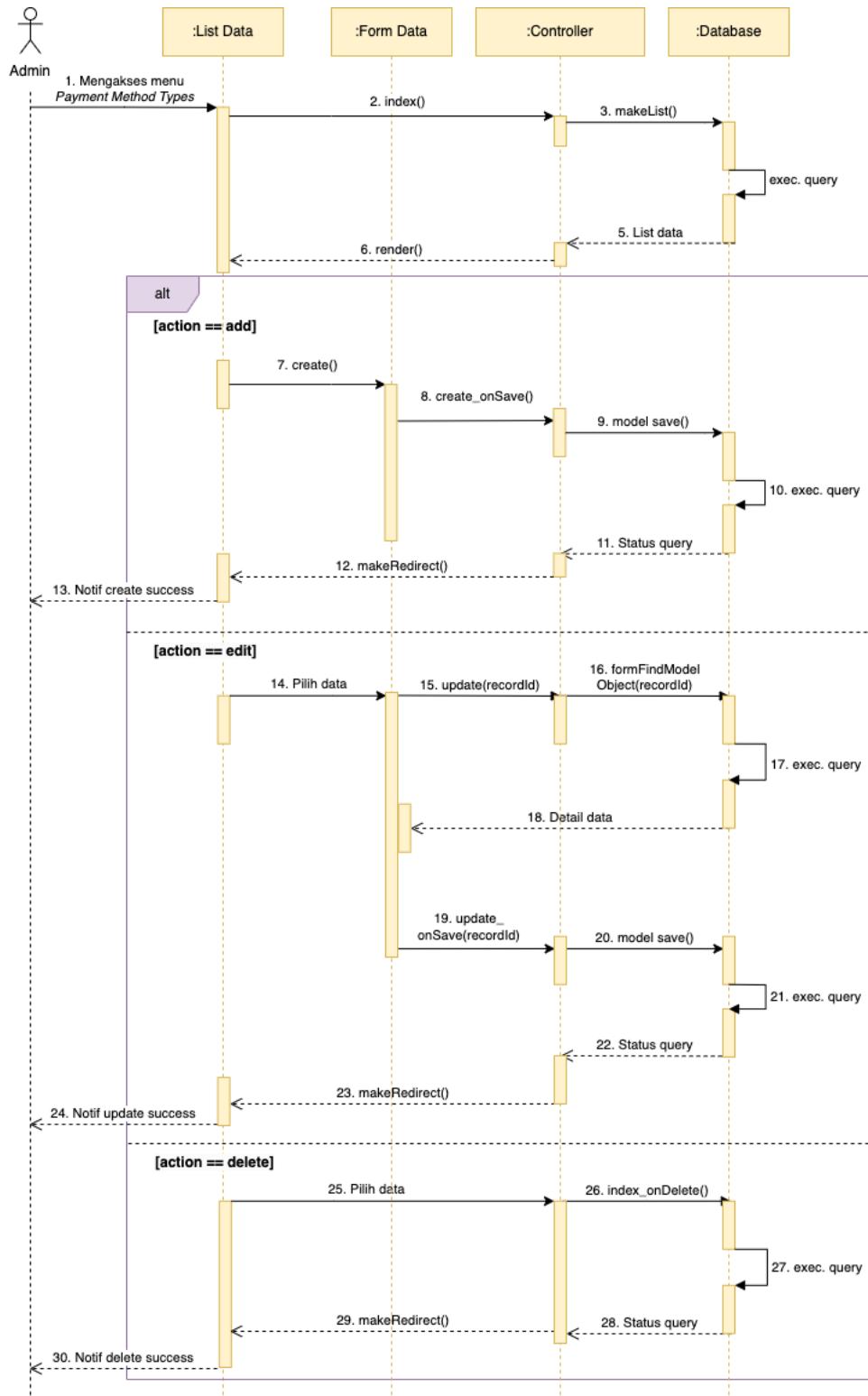
Gambar 4.22 Sequence Diagram Mengelola Data Metode Pembayaran

4.3.3.8 Mengelola Data *Payment Gateway Provider*



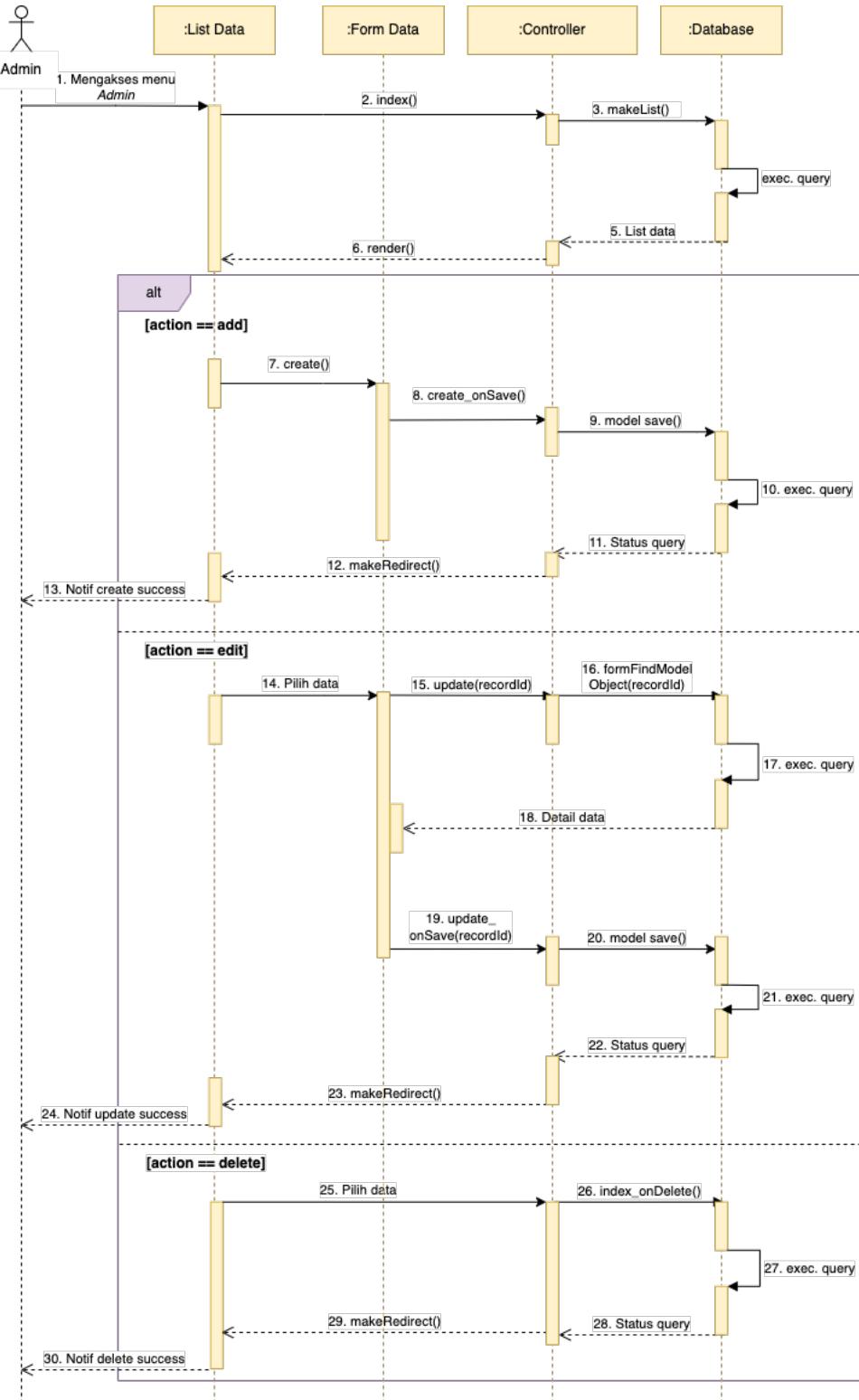
Gambar 4.23- Sequence Diagram Mengelola Data *Payment Gateway Provider*

4.3.3.9 Mengelola Data Tipe Metode Pembayaran



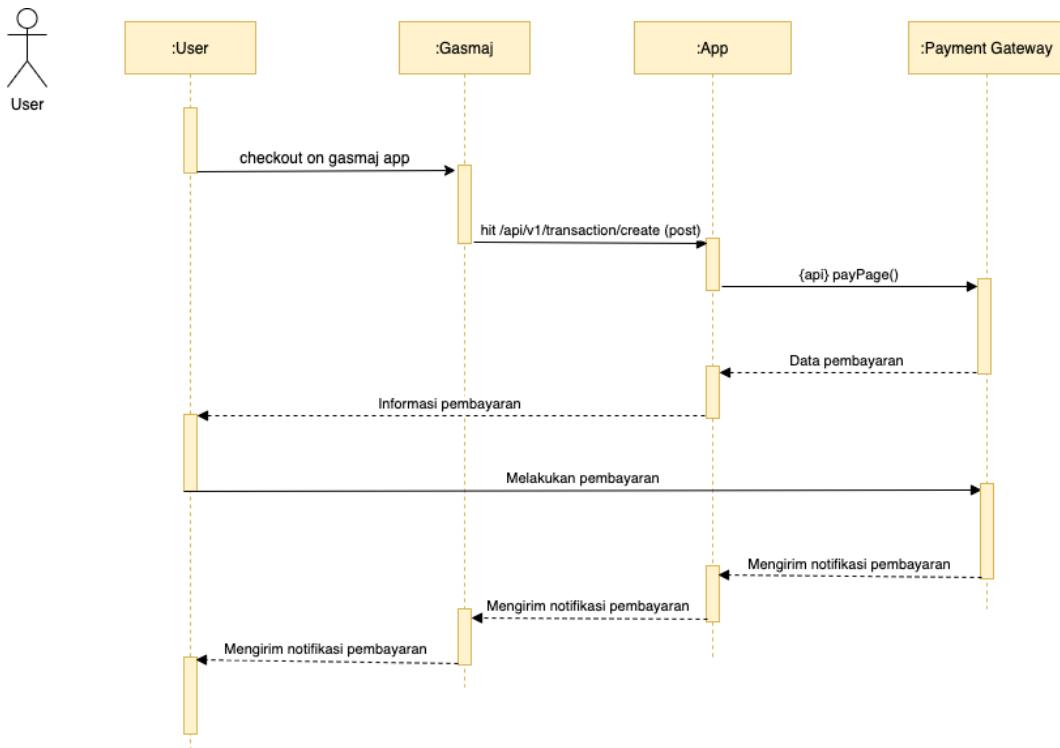
Gambar 4.24 Sequence Diagram Mengelola Data Tipe Metode Pembayaran

4.3.3.10 Mengelola Data Admin



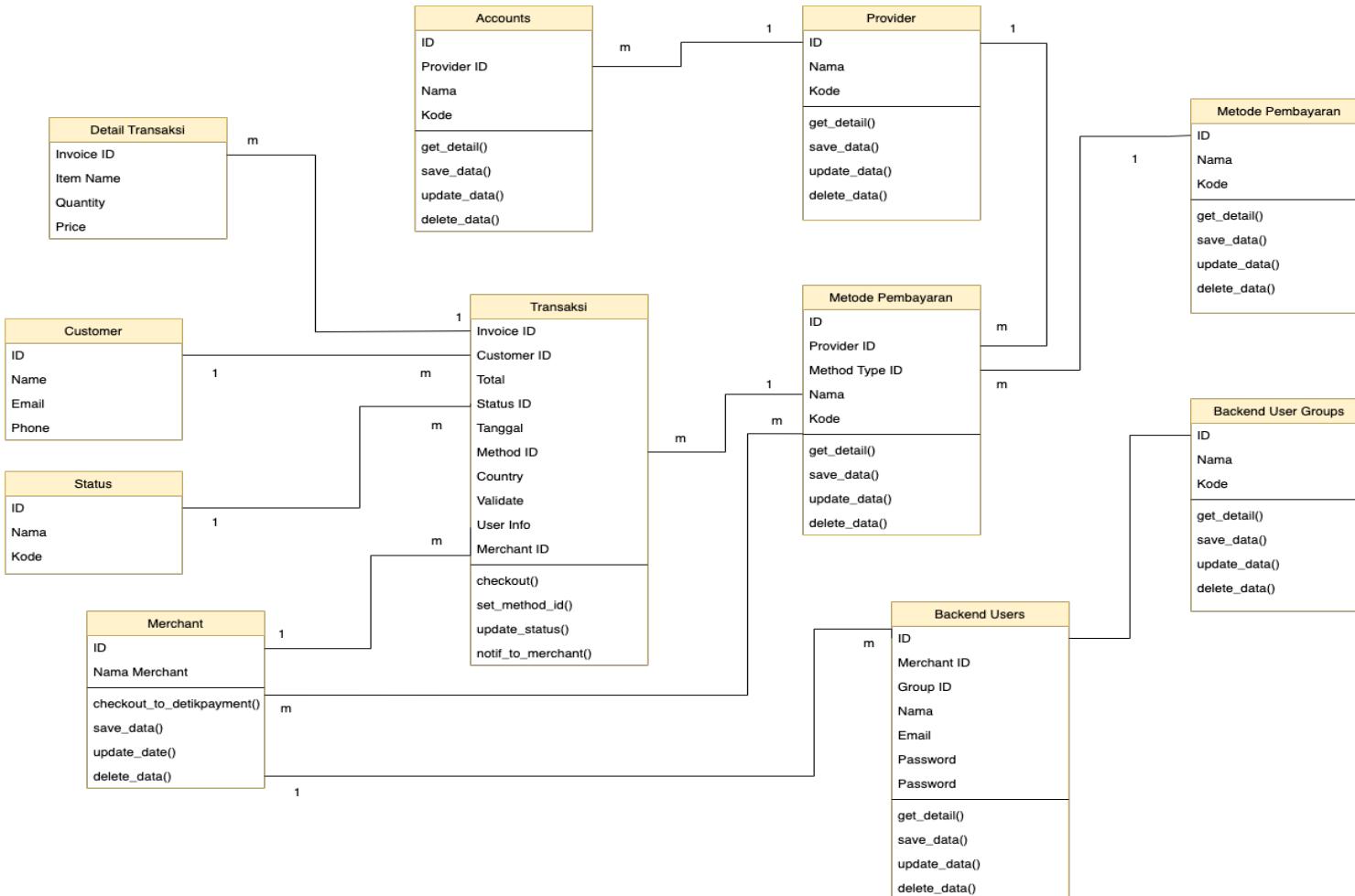
Gambar 4.25 Sequence Diagram Mengelola Data Admin

4.3.3.11 Melakukan Transaksi



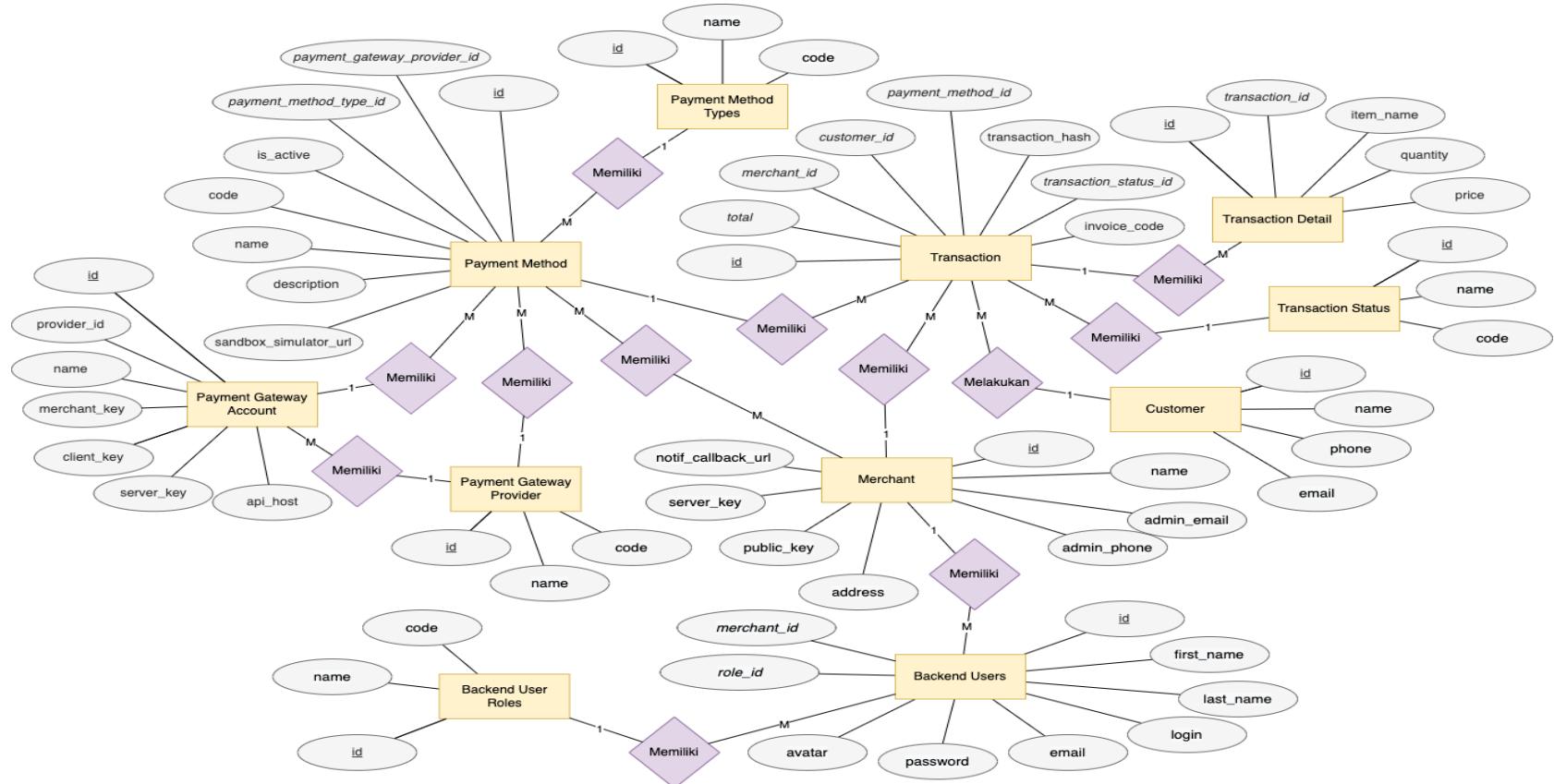
Gambar 4.26 Sequence Diagram Melakukan Transaksi

4.3.4 Class Diagram



Gambar 4.27 Class Diagram

4.3.5 Entity Relationship Diagram



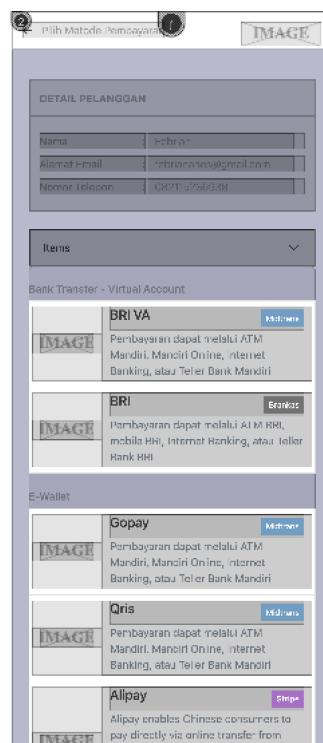
Gambar 4.28 Entity Relationship Diagram

4.4 Perancangan Output

Rancangan *output* akan lebih berfokus ke halaman depan aplikasi. Berikut adalah rancangan output dari *Aplikasi Payment Gateway Aggregator* yang akan dikembangkan:

4.4.1 Halaman Pilih Metode Pembayaran

Pada halaman ini, pengguna akan ditampilkan produk yang akan dibayarkan dan pengguna juga akan ditampilkan metode pembayaran yang disediakan oleh *merchant* (Gasmaj).



Gambar 4.29 Output Halaman Pilih Metode Pembayaran

Dari *output* di atas, terdapat beberapa komponen dalam halaman tersebut, komponen utama yang ada pada halaman tersebut adalah sebagai berikut:

a. Detail Pelanggan



Gambar 4.30 Output Info Pelanggan

Menampilkan data detail pelanggan yang terdiri dari Nama, Alamat Email dan Nomor Telepon.

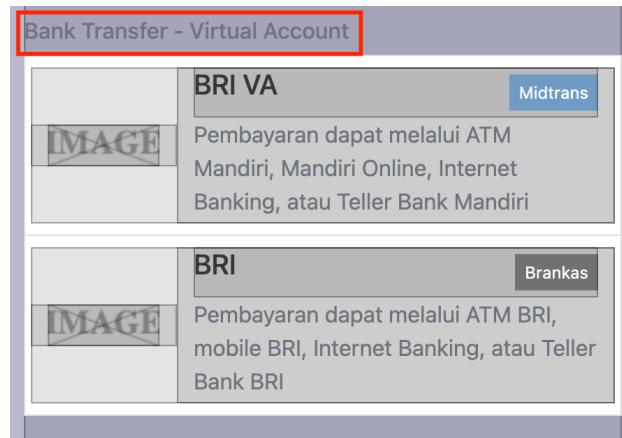
b. *Items*

Items			
#	Item Name	Qty	Price
1	Tshirt	1	Rp. 40,000
Total			Rp. 40,000

Gambar 4.31 Output Accordion Item

Menampilkan detail pembelian produk pelanggan yang akan dibayarkan. *Items* berupa *accordion* yang dapat diklik oleh *customer*.

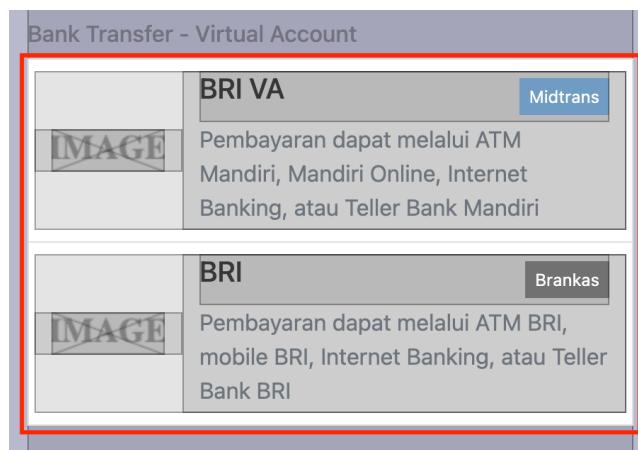
c. Kategori / Tipe Metode Pembayaran



Gambar 4.32 Output Kategori / Tipe Metode Pembayaran

Merupakan *header section* dari setiap metode pembayaran.

d. Metode Pembayaran



Gambar 4.33 Output Daftar Metode Pembayaran

Merupakan metode pembayaran yang dapat dipilih oleh pengguna, metode pembayaran terdiri dari dua komponen yaitu judul, dan juga deskripsi metode pembayaran.

4.4.2 Halaman Konfirmasi Pembayaran

Pada halaman ini, pengguna akan ditampilkan halaman konfirmasi pemilihan metode pembayaran yang dipilih oleh pengguna. Halaman ini diperlukan untuk memvalidasi apakah pengguna sudah yakin dengan metode pembayaran yang dipilih olehnya:



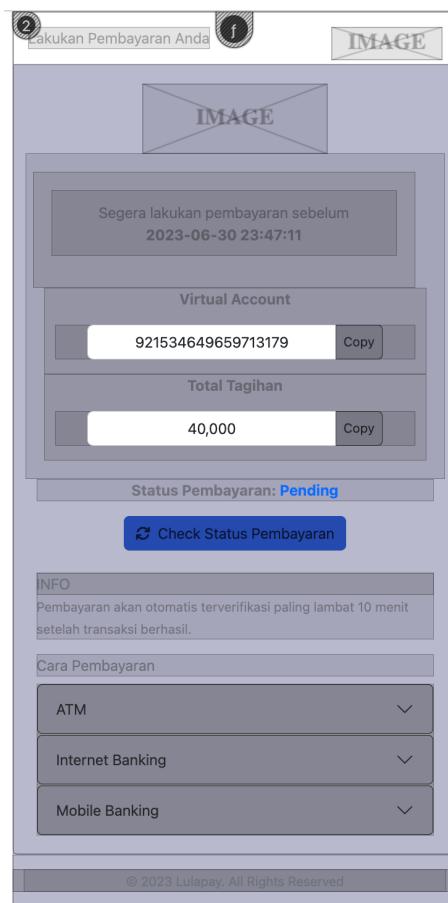
Gambar 4.34 Output Halaman Konfirmasi Pembayaran

Dari *output* di atas, terdapat beberapa komponen dalam halaman tersebut, komponen utama yang ada pada halaman tersebut adalah sebagai berikut:

- a. *Icon arrow left*, untuk kembali ke menu “Pilih Metode Pembayaran”
- b. *Image 1*, logo *payment gateway aggregator*.
- c. *Image 2*, logo metode pembayaran yang telah dipilih.
- d. Deskripsi Konfirmasi pembayaran.
- e. Tombol “Bayar Tagihan” untuk melakukan konfirmasi.

4.4.3 Halaman Pembayaran

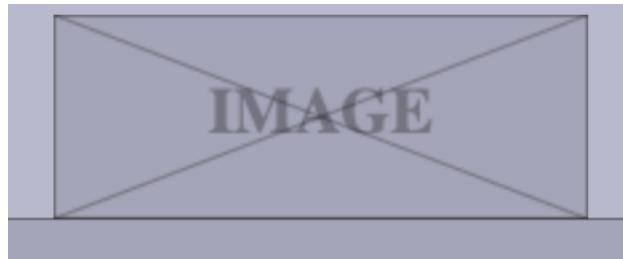
Pada Halaman ini, akan menampilkan informasi dan instruksi pembayaran, perlu dicatat bahwa tidak semua *provider payment gateway* memberikan informasi pembayaran dalam bentuk data, ada beberapa *provider* dan metode pembayaran yang perlu membuka halaman dari *payment gateway* tersebut, sehingga rancangan ini bersifat relatif, namun jika semua data bisa didapatkan seperti *virtual account* Midtrans, berikut adalah gambar rancangannya pada aplikasi *payment gateway aggregator*:



Gambar 4.35 Output Halaman Pembayaran

Dari *output* pada halaman sebelumnya, terdapat beberapa komponen dalam halaman tersebut, komponen utama yang ada pada halaman tersebut adalah sebagai berikut:

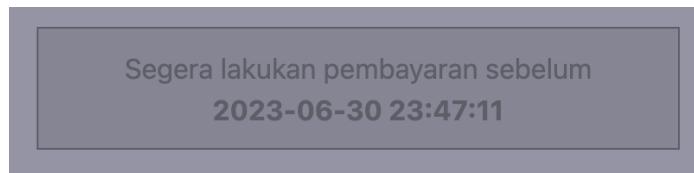
- a. Logo Metode Pembayaran



Gambar 4.36 Logo Metode Pembayaran

Menampilkan logo metode pembayaran yang telah dipilih sebelumnya oleh pengguna di “Halaman Pilih Metode Pembayaran”.

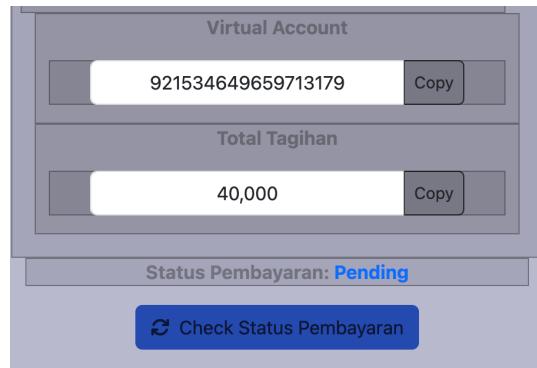
- b. Batas Waktu Pembayaran



Gambar 4.37 Batas Waktu Pembayaran

Pada bagian ini, pengguna dapat mengetahui batas waktu pembayaran atau batas waktu akhir pelunasan dari produk yang akan dibayarkan.

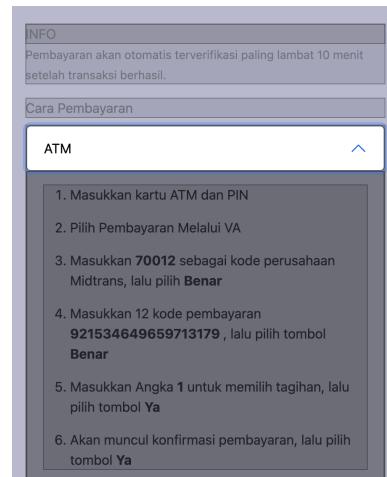
c. Informasi Pembayaran



Gambar 4.38 Output Informasi Pembayaran

Pada bagian ini, pengguna akan ditampilkan *output* berupa informasi nomor *virtual account* dan Total Tagihan yang bisa dibayarkan. Selain itu, pengguna juga dapat melihat status dari pembayaran yang sedang dilakukan beserta tombol untuk melakukan *refresh* untuk mendapatkan data status terbaru dari *payment gateway*.

d. Informasi dan Tata Cara Pembayaran

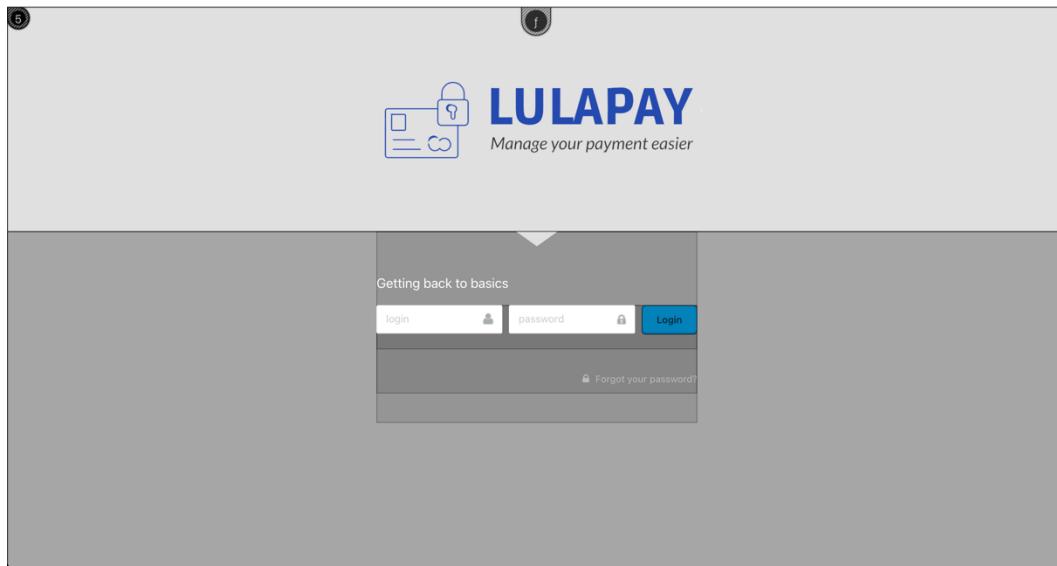


Gambar 4.39 Output Informasi dan Tata Cara Pembayaran

Menampilkan informasi penting berkaitan dengan transaksi dan menampilkan tata cara pembayaran dalam bentuk *accordion*.

4.5 Perancangan Input

4.5.1 Input Data Login



Gambar 4.40 Rancangan formulir login

Formulir *Input* di atas merupakan formulir *login* untuk masuk ke dalam aplikasi internal, yang terdiri dari:

- a. *Login*, merupakan *email* atau *username* login pengguna.
- b. *Password*, merupakan kata sandi yang digunakan untuk masuk ke dalam aplikasi.

4.5.2 Input Data Admin

The screenshot shows a user interface for creating a new administrator account. At the top, there are navigation buttons for 'Administrators' and 'New Administrator'. The main area is titled 'Account' and contains the following fields:

- Login ***: A text input field.
- Email ***: A text input field.
- Send invitation by email**: A checkbox with a descriptive subtitle: "Sends a welcome message containing login and password information."
- First Name**: A text input field.
- Last Name**: A text input field.
- Password ***: A text input field.
- Confirm Password ***: A text input field.
- Role**: A section describing roles and their permissions. It includes two radio button options:
 - Super Admin**: Described as a "Site administrator with access to developer tools."
 - Merchant**: Described as "User's merchant"

At the bottom of the form are three buttons: 'Create', 'Create and close', and 'or Cancel'.

Gambar 4.41 Rancangan Input Data Admin

Formulir *Input* di atas merupakan formulir data login admin atau pengelola aplikasi untuk masuk ke dalam aplikasi internal, yang terdiri dari:

- c. *Login*, merupakan *username* login pengguna.
- d. *Email*, merupakan email pengguna.
- e. *First Name & Last Name*, merupakan nama pengguna.
- f. *Password*, merupakan kata sandi yang digunakan untuk masuk ke dalam aplikasi.
- g. *Confirm Password*, untuk melakukan validasi agar *password* yang dimasukkan sesuai.
- h. *Role*, merupakan peran pengguna yang ingin didaftarkan.
- i. *Avatar*, merupakan *avatar* atau foto profil pengguna.

4.5.3 Input Data Metode Pembayaran

The screenshot shows a software interface for creating a new payment method. At the top, there's a breadcrumb navigation: 'Payment Methods > Create Payment Method'. Below this, there are several input fields and dropdown menus:

- Provider:** A dropdown menu with placeholder text 'Click the button to find a record'.
- Method Type:** A dropdown menu with placeholder text 'Click the button to find a record'.
- Name:** An input field.
- Code:** An input field.
- Description:** A large text area.
- Sandbox Simulator URL (if any):** An input field.
- Is Active?**: A checkbox.
- Logo:** A file upload input field with a small preview thumbnail showing an upward arrow icon.

At the bottom of the form are three buttons: 'Create', 'Create and Close', and 'Cancel'.

Gambar 4.42 Rancangan Input Data Metode Pembayaran

Formulir *Input* di atas merupakan formulir data metode pembayaran yang nantinya dapat digunakan oleh *merchant* sebagai pilihan metode pembayaran, yang terdiri dari:

- a. *Provider*, merupakan *payment gateway* dari metode pembayaran tersebut
- b. *Method Type*, merupakan jenis metode pembayaran
- c. *Name*, nama metode pembayaran
- d. *Code*, kode metode pembayaran
- e. *Description*, deskripsi metode pembayaran
- f. *Sandbox Simulator URL (if any)*, URL *sandbox* atau *environment testing* yang disediakan oleh *payment gateway* untuk melakukan percobaan pembayaran.
- g. *Is active*, untuk menonaktifkan atau mengaktifkan metode pembayaran, jika diaktifkan, akan tampil pada halaman depan.

- h. *Logo*, logo dari metode pembayaran yang akan ditampilkan pada halaman pembayaran.

4.5.4 Input Data Tipe Metode Pembayaran

Gambar 4.43 Rancangan Input Data Tipe Metode Pembayaran

Formulir *Input* di atas merupakan formulir data tipe dari metode pembayaran, yang terdiri dari:

- Name*, merupakan nama tipe metode pembayaran.
- Code*, metode pembayaran

4.5.5 Input Data *Payment Gateway Provider*

Gambar 4.44 Rancangan Input Data Payment Gateway Provider

Formulir *Input* di atas merupakan formulir data *payment gateway* yang tersedia pada aplikasi, yang terdiri dari:

- Name*, merupakan nama *provider payment gateway*.
- Code*, merupakan kode *provider payment gateway*.

4.5.6 Input Data Akun *Payment Gateway*

The screenshot shows a software application window titled 'Accounts' with a sub-tab 'Create Account'. The main area is labeled 'Provider' and contains a note: 'Click the button to find a record'. Below this are several input fields: 'Name' (with a red asterisk), 'API Host (Base URL)' (with a red asterisk), 'Merchant Key', 'Client Key', and 'Server Key'. At the bottom of the form are three buttons: 'Create', 'Create and Close', and 'Cancel'.

Gambar 4.45 Rancangan Input Data Akun Payment Gateway

Formulir *Input* di atas merupakan formulir data akun atau kredensial *payment gateway*, yang nantinya akan digunakan untuk proses pembayaran melalui metode pembayaran yang dipilih oleh pelanggan, yang terdiri dari:

- a. *Provider*, merupakan *provider payment gateway*.
- b. *Name*, merupakan nama akun.
- c. *API Host (Base URL)*, merupakan *base url* dari API *payment gateway* yang akan digunakan.
- d. *Merchant Key*, merupakan kode *merchant* atau informasi terkait nama dari *akun payment gateway*.
- e. *Client Key*, merupakan akses kunci publik yang digunakan sebagai autentikasi saat untuk terhubung dengan *payment gateway*.
- f. *Server Key*, merupakan akses kunci rahasia yang digunakan sebagai autentikasi saat untuk terhubung dengan *payment gateway*.

4.5.7 Input Data Merchant

The screenshot shows the 'Edit Merchant' interface. At the top, there's a breadcrumb navigation: 'Merchants > Edit Merchant'. The main area contains several input fields and sections:

- Name:** GasMaj
- Address:** Test
- Email:** febrianaries@gmail.com
- Phone:** 082115236938
- Notification Callback URL:** https://example.com/notif
- Public Key (auto generated):** 0edf10ab-5fe1-47cd-bfae-7985b6bd4ce5
- Server Key (auto generated):** 302cdb73-95a2-4b47-bf54-82718bd0e109
- Logo:** An empty placeholder with an upload icon.
- Payment Methods:** A section titled 'Select available payment method for this merchant' with two buttons: 'Select all' and 'Select none'. Below this, a list of payment methods is shown:
 - Bank Transfer - Virtual Account**
 - Midtrans - Mandiri Bill
 - Midtrans - BNI VA
 - Midtrans - BRI VA
 - Midtrans - Permata VA
 - Brankas - BCA
 - Brankas - BRI
 - Midtrans - BCA
 - Brankas - Brankas

At the bottom, there are three buttons: 'Save', 'Save and Close', and 'Cancel'.

Gambar 4.46 Rancangan Input Data Merchant

Formulir *Input* di atas merupakan formulir data *merchant* atau yang dapat terhubung dengan aplikasi *payment gateway aggregator*:

- Name*, merupakan nama *merchant* yang akan terhubung.
- Address*, merupakan alamat *merchant*.
- Email*, merupakan alamat *email* dari *admin merchant*.
- Phone*, merupakan nomor telepon dari *admin merchant*.

- e. *Notification Callback URL*, merupakan endpoint API aplikasi *merchant* untuk menerima informasi perubahan status pada setiap transaksi dari *payment gateway aggregator*.
- f. *Public Key*, merupakan akses kunci publik yang digunakan sebagai autentikasi saat untuk terhubung dengan aplikasi.
- g. *Server Key*, merupakan akses kunci rahasia yang digunakan sebagai autentikasi saat untuk terhubung dengan aplikasi.
- h. *Logo*, merupakan logo merchant.
- i. *Payment Methods*, merupakan data metode pembayaran yang ingin dapat dipilih sebagai opsi pembayaran di halaman pembayaran.

4.6 Percanganan Menu

Berikut adalah rancangan menu yang akan dikembangkan pada Aplikasi *Payment Gateway Aggregator*.

1. Dashboard

Menu "*Dashboard*" adalah tampilan utama atau laman beranda pada aplikasi. Ini memberikan ringkasan atau ikhtisar informasi penting tentang aktivitas transaksi, kinerja keuangan, statistik, dan metrik lainnya.

2. Payment Gateways

a. Providers

Menu "*Payment Gateways*" menyediakan daftar atau pengaturan untuk mengelola berbagai penyedia *payment gateway* yang terhubung ke aplikasi. Ini mencakup informasi tentang penyedia *payment gateway* yang tersedia, seperti Midtrans, Brankas, dan Stripe.

b. Accounts

Menu "*Payment Gateways Accounts*" digunakan untuk mengatur akun atau koneksi spesifik untuk setiap penyedia *payment gateway*. Di sini pengguna dapat melakukan konfigurasi data akun seperti kunci API, autentikasi *token*, dan pengaturan lainnya yang diperlukan untuk menghubungkan aplikasi dengan *payment gateway*.

3. *Merchants*

Menu "*Merchants*" adalah bagian yang berkaitan dengan manajemen dan pengaturan *merchant* Gasmaj yang menggunakan aplikasi. Di sini, pengguna dapat membuat, mengedit, atau menghapus profil *merchant*, dan mengatur preferensi pembayaran.

4. *Transactions*

a. *Transactions*

Menu "*Transactions*" menyediakan daftar atau riwayat lengkap transaksi pembayaran yang telah dilakukan melalui aplikasi. Pengguna dapat melihat detail transaksi, seperti tanggal, jumlah, metode pembayaran, status, dan informasi pelanggan terkait.

b. *Payment Methods*

Menu "*Payment Methods*" berhubungan dengan manajemen dan pengaturan metode pembayaran yang tersedia dalam aplikasi. Ini mencakup daftar metode pembayaran yang diterima, seperti kartu kredit, transfer bank, atau dompet digital, serta pengaturan terkait, seperti biaya transaksi atau validasi.

c. *Payment Method Types*

Menu "*Payment Method Types*" adalah tempat untuk mengelola jenis-jenis metode pembayaran yang digunakan dalam aplikasi. Ini mencakup kategori atau jenis metode pembayaran.

5. Admin

- a. Menu "Admin" adalah menu tempat mengelola data admin yang dapat *login* atau masuk ke dalam aplikasi internal *payment gateway aggregator*, dalam menu ini, admin dapat menambah pengguna admin lainnya atau pengguna admin merchant untuk melihat dan mengelola data transaksi.

BAB V

IMPLEMENTASI

5.1 Spesifikasi Perangkat Keras dan Perangkat Lunak

Ketika melakukan penelitian dan pengembangan aplikasi seperti yang dijelaskan sebelumnya, ada beberapa aspek yang perlu dipertimbangkan terkait perangkat keras dan perangkat lunak yang digunakan.

5.1.1 Spesifikasi Perangkat Keras

Dalam penelitian ini, digunakan satu laptop, satu *smartphone* dan satu *Virtual Private Server* (VPS) dengan spesifikasi perangkat keras sebagai berikut:

1. Spesifikasi Laptop:

- a. *Processors* : Intel Core i5
- b. *Memory (RAM)* : 8 GB
- c. *Storage* : 256 GB

2. Spesifikasi *Smartphone*:

- a. Merk Laptop : iPhone 11
- b. *Processors* : A13 Bionic
- c. *Memory (RAM)* : 4 GB
- d. *Storage* : 64 GB

3. Spesifikasi *Virtual Private Server*:

- a. CPU : 1 Core
- b. *Memory (RAM)* : 2GB
- c. *Storage* : 20 GB

5.1.2 Spesifikasi Perangkat Lunak

Dalam penelitian ini, digunakanlah perangkat lunak atau software dengan spesifikasi sebagai berikut:

1. Perangkat Lunak Pengembangan:
 - a. Sistem Operasi macOS Monterey 12.6 – Laptop
 - b. Visual Studio Code 1.79.2 (*Universal*)
 - c. Git 2.38.0 & Git Desktop 3.2.6
 - d. MPP 7.4.29
 - e. Laravel 5.5.50
 - f. Postman 10.15.6
2. Perangkat Lunak *Testing*:
 - a. Sistem Operasi macOS Monterey 12.6 – Laptop
 - b. Sistem Operasi iOS v16.5.1- *Smartphone*
 - c. Google Chrome Desktop & Mobile
3. Perangkat Lunak *Server*:
 - a. Sistem Operasi Linux Ubuntu 20.04
 - b. *Webserver* Nginx
 - c. PHP 7.4
 - d. MySQL Server

5.2 Implementasi Basis Data

5.2.1 Membuat Tabel lulapay_merchant_merchants

Script di bawah ini digunakan untuk membuat tabel daftar *merchant* yang didaftarkan ke dalam aplikasi.

```

1. CREATE TABLE `lulapay_merchant_merchants` (
2.   `id` int(10) UNSIGNED NOT NULL,
3.   `name` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
4.   `address` text COLLATE utf8mb4_unicode_ci DEFAULT NULL,
5.   `admin_email` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
6.   `admin_phone` varchar(14) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
7.   `public_key` varchar(64) COLLATE utf8mb4_unicode_ci NOT NULL,
8.   `server_key` varchar(64) COLLATE utf8mb4_unicode_ci NOT NULL,
9.   `notif_callback_url` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
10.  `created_at` timestamp NULL DEFAULT NULL,
11.  `updated_at` timestamp NULL DEFAULT NULL,
12.  `deleted_at` timestamp NULL DEFAULT NULL,
13.  `disabled_at` datetime DEFAULT NULL
14. ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

5.2.2 Membuat Tabel lulapay_paymentgateway_accounts

Script di bawah ini digunakan untuk membuat tabel yang menampung data kredensial akun *payment gateway* yang dapat digunakan untuk pembayaran. Kredensial bisa berupa sandi penting untuk bisa terhubung ke *payment gateway*.

```

1. CREATE TABLE `lulapay_paymentgateway_accounts` (
2.   `id` int(10) UNSIGNED NOT NULL,
3.   `provider_id` int(10) UNSIGNED NOT NULL,
4.   `name` varchar(50) COLLATE utf8mb4_unicode_ci NOT NULL,
5.   `merchant_key` varchar(25) COLLATE utf8mb4_unicode_ci NOT NULL,
6.   `client_key` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
7.   `server_key` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
8.   `created_at` timestamp NULL DEFAULT NULL,
9.   `updated_at` timestamp NULL DEFAULT NULL,
10.  `deleted_at` timestamp NULL DEFAULT NULL,
11.  `api_host` varchar(100) COLLATE utf8mb4_unicode_ci DEFAULT NULL
12. ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

5.2.3 Membuat Tabel lulapay_merchant_payment_accounts

Script di bawah ini digunakan untuk membuat tabel yang menampung relasi antara akun *payment gateway* dengan *merchant*.

```
1. CREATE TABLE `lulapay_merchant_payment_accounts` (
2.   `merchant_id` int(10) UNSIGNED NOT NULL,
3.   `payment_account_id` int(10) UNSIGNED NOT NULL
4. ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

5.2.4 Membuat Tabel lulapay_transaction_payment_method_types

Script di bawah ini digunakan untuk membuat tabel daftar jenis metode pembayaran yang didaftarkan ke dalam aplikasi.

```
1. CREATE TABLE `lulapay_transaction_payment_method_types` (
2.   `id` int(10) UNSIGNED NOT NULL,
3.   `name` varchar(50) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
4.   `code` varchar(50) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
5.   `created_at` timestamp NULL DEFAULT NULL,
6.   `updated_at` timestamp NULL DEFAULT NULL,
7.   `deleted_at` timestamp NULL DEFAULT NULL
8. ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

5.2.5 Membuat Tabel lulapay_transaction_payment_methods

Script di bawah ini digunakan untuk membuat tabel daftar metode pembayaran yang didaftarkan ke dalam aplikasi.

```
1. CREATE TABLE `lulapay_transaction_payment_methods` (
2.   `id` int(10) UNSIGNED NOT NULL,
3.   `payment_gateway_provider_id` int(11) UNSIGNED DEFAULT NULL,
4.   `name` varchar(50) COLLATE utf8mb4_unicode_ci NOT NULL,
5.   `code` varchar(30) COLLATE utf8mb4_unicode_ci NOT NULL,
6.   `created_at` timestamp NULL DEFAULT NULL,
7.   `updated_at` timestamp NULL DEFAULT NULL,
8.   `deleted_at` timestamp NULL DEFAULT NULL,
9.   `payment_method_type_id` int(10) UNSIGNED NOT NULL,
10.  `description` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
11.  `tata_cara` text COLLATE utf8mb4_unicode_ci DEFAULT NULL,
12.  `sandbox_simulator_url` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
13.  `is_active` tinyint(1) DEFAULT 1
14. ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

5.2.6 Membuat Tabel lulapay_merchant_payment_methods

Script di bawah ini digunakan untuk membuat tabel yang menampung relasi antara akun jenis pembayaran dan *merchant*.

```
1. CREATE TABLE `lulapay_merchant_payment_methods` (
2.   `merchant_id` int(10) UNSIGNED NOT NULL,
3.   `payment_method_id` int(10) UNSIGNED NOT NULL
4. ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

5.2.7 Membuat Tabel lulapay_paymentgateway_providers

Script di bawah ini digunakan untuk membuat tabel daftar *payment gateway* yang didaftarkan ke dalam aplikasi.

```
1. CREATE TABLE `lulapay_paymentgateway_providers` (
2.   `id` int(11) UNSIGNED NOT NULL,
3.   `name` varchar(12) COLLATE utf8mb4_unicode_ci NOT NULL,
4.   `code` varchar(12) COLLATE utf8mb4_unicode_ci NOT NULL,
5.   `created_at` timestamp NULL DEFAULT NULL,
6.   `updated_at` timestamp NULL DEFAULT NULL,
7.   `deleted_at` timestamp NULL DEFAULT NULL
8. ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

5.2.8 Membuat Tabel lulapay_transaction_customers

Script di bawah ini digunakan untuk membuat tabel daftar pelanggan yang melakukan pembayaran menggunakan aplikasi.

```
1. CREATE TABLE `lulapay_transaction_customers` (
2.   `id` int(10) UNSIGNED NOT NULL,
3.   `name` varchar(25) COLLATE utf8mb4_unicode_ci NOT NULL,
4.   `phone` varchar(14) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
5.   `email` varchar(55) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
6.   `created_at` timestamp NULL DEFAULT NULL,
7.   `updated_at` timestamp NULL DEFAULT NULL,
8.   `deleted_at` timestamp NULL DEFAULT NULL
9. ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
10.
```

5.2.9 Membuat Tabel lulapay_transaction_statuses

Script di bawah ini digunakan untuk membuat tabel daftar metode pembayaran yang didaftarkan ke dalam aplikasi.

```
1. CREATE TABLE `lulapay_transaction_statuses` (
2.   `id` int(10) UNSIGNED NOT NULL,
3.   `name` varchar(12) COLLATE utf8mb4_unicode_ci NOT NULL,
4.   `code` varchar(12) COLLATE utf8mb4_unicode_ci NOT NULL,
5.   `created_at` timestamp NULL DEFAULT NULL,
6.   `updated_at` timestamp NULL DEFAULT NULL,
7.   `deleted_at` timestamp NULL DEFAULT NULL
8. ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

5.2.10 Membuat Tabel lulapay_transaction_transactions

Script di bawah ini digunakan untuk membuat tabel daftar transaksi yang dilakukan oleh *merchant* dan *pelanggan*.

```
1. CREATE TABLE `lulapay_transaction_transactions` (
2.   `id` int(10) UNSIGNED NOT NULL,
3.   `merchant_id` int(10) UNSIGNED NOT NULL,
4.   `payment_method_id` int(11) DEFAULT NULL,
5.   `invoice_code` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
6.   `transaction_hash` varchar(64) COLLATE utf8mb4_unicode_ci NOT NULL,
7.   `transaction_status_id` int(10) UNSIGNED NOT NULL,
8.   `total` decimal(10,0) NOT NULL,
9.   `created_at` timestamp NULL DEFAULT NULL,
10.  `updated_at` timestamp NULL DEFAULT NULL,
11.  `deleted_at` timestamp NULL DEFAULT NULL,
12.  `customer_id` int(11) UNSIGNED DEFAULT NULL,
13.  `expired_time` datetime DEFAULT NULL
14. ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

5.2.11 Membuat Tabel lulapay_transaction_transaction_details

Script di bawah ini digunakan untuk membuat tabel detail transaksi yang dilakukan, menampung informasi rincian produk dan harga.

```
1. CREATE TABLE `lulapay_transaction_transaction_details` (
2.   `id` int(10) UNSIGNED NOT NULL,
3.   `transaction_id` int(10) UNSIGNED NOT NULL,
4.   `item_name` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
5.   `quantity` int(10) UNSIGNED NOT NULL,
6.   `price` decimal(10,0) DEFAULT NULL,
7.   `created_at` timestamp NULL DEFAULT NULL,
8.   `updated_at` timestamp NULL DEFAULT NULL,
9.   `deleted_at` timestamp NULL DEFAULT NULL
10. ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

5.2.12 Membuat Tabel lulapay_transaction_transaction_logs

Script di bawah ini digunakan untuk membuat tabel yang menampung pencatatan perubahan transaksi, setiap kali terjadi perubahan status, akan dicatat ke dalam tabel ini.

```

1. CREATE TABLE `lulapay_transaction_transaction_logs` (
2.   `id` int(10) UNSIGNED NOT NULL,
3.   `transaction_id` int(10) UNSIGNED NOT NULL,
4.   `type` varchar(191) COLLATE utf8mb4_unicode_ci NOT NULL,
5.   `data` text COLLATE utf8mb4_unicode_ci DEFAULT NULL,
6.   `created_at` timestamp NULL DEFAULT NULL,
7.   `updated_at` timestamp NULL DEFAULT NULL,
8.   `deleted_at` timestamp NULL DEFAULT NULL,
9.   `transaction_status_id` int(10) UNSIGNED NOT NULL
10. ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

```

5.2.13 Membuat Relasi Tabel

Script di bawah ini digunakan untuk menambahkan relasi ke masing-masing tabel yang sudah diimplementasi di atas.

```

1. -- Constraints for table `lulapay_merchant_payment_accounts`
2. ALTER TABLE `lulapay_merchant_payment_accounts`
3.   ADD CONSTRAINT `lulapay_merchant_payment_accounts_ibfk_1` FOREIGN KEY
(`merchant_id`) REFERENCES `lulapay_merchant_merchants` (`id`),
4.   ADD CONSTRAINT `lulapay_merchant_payment_accounts_ibfk_2` FOREIGN KEY
(`payment_account_id`) REFERENCES `lulapay_paymentgateway_accounts` (`id`);
5.
6. -- Constraints for table `lulapay_merchant_payment_methods`
7. ALTER TABLE `lulapay_merchant_payment_methods`
8.   ADD CONSTRAINT `lulapay_merchant_payment_methods_ibfk_1` FOREIGN KEY
(`merchant_id`) REFERENCES `lulapay_merchant_merchants` (`id`),
9.   ADD CONSTRAINT `lulapay_merchant_payment_methods_ibfk_2` FOREIGN KEY
(`payment_method_id`) REFERENCES `lulapay_transaction_payment_methods` (`id`);
10.
11. -- Constraints for table `lulapay_paymentgateway_accounts`
12. ALTER TABLE `lulapay_paymentgateway_accounts`
13.   ADD CONSTRAINT `lulapay_paymentgateway_accounts_ibfk_1` FOREIGN KEY
(`provider_id`) REFERENCES `lulapay_paymentgateway_providers` (`id`);
14.
15. -- Constraints for table `lulapay_transaction_payment_methods`
16. ALTER TABLE `lulapay_transaction_payment_methods`
17.   ADD CONSTRAINT `lulapay_transaction_payment_methods_ibfk_1` FOREIGN KEY
(`payment_gateway_provider_id`) REFERENCES `lulapay_paymentgateway_providers` (`id`),
18.   ADD CONSTRAINT `lulapay_transaction_payment_methods_ibfk_2` FOREIGN KEY
(`payment_method_type_id`) REFERENCES `lulapay_transaction_payment_method_types` (`id`);
19. -- Constraints for table `lulapay_transaction_transactions`

```

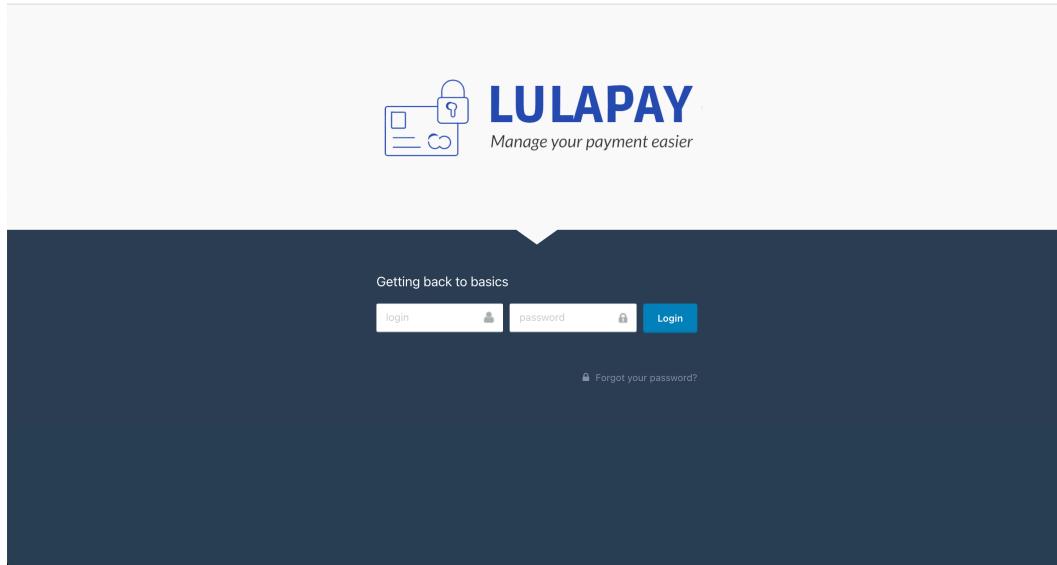
```

20. ALTER TABLE `lulapay_transaction_transactions`
21.   ADD CONSTRAINT `lulapay_transaction_transactions_ibfk_1` FOREIGN KEY
(`merchant_id`) REFERENCES `lulapay_merchant_merchants` (`id`),
22.   ADD CONSTRAINT `lulapay_transaction_transactions_ibfk_2` FOREIGN KEY
(`transaction_status_id`) REFERENCES `lulapay_transaction_statuses` (`id`),
23.   ADD CONSTRAINT `lulapay_transaction_transactions_ibfk_3` FOREIGN KEY
(`customer_id`) REFERENCES `lulapay_transaction_customers` (`id`);
24.
25. -- Constraints for table `lulapay_transaction_transaction_details`
26. ALTER TABLE `lulapay_transaction_transaction_details`
27.   ADD CONSTRAINT `lulapay_transaction_transaction_details_ibfk_1` FOREIGN KEY
(`transaction_id`) REFERENCES `lulapay_transaction_transactions` (`id`);
28.
29. -- Constraints for table `lulapay_transaction_transaction_logs`
30. ALTER TABLE `lulapay_transaction_transaction_logs`
31.   ADD CONSTRAINT `lulapay_transaction_transaction_logs_ibfk_1` FOREIGN KEY
(`transaction_id`) REFERENCES `lulapay_transaction_transactions` (`id`),
32.   ADD CONSTRAINT `lulapay_transaction_transaction_logs_ibfk_2` FOREIGN KEY
(`transaction_status_id`) REFERENCES `lulapay_transaction_statuses` (`id`);

```

5.3 Implementasi Proses

5.3.1 Implementasi Login



Gambar 5.1 Implementasi Login

Gambar di atas adalah formulir *login* pada aplikasi internal *payment gateway aggregator*. Untuk masuk ke dalam aplikasi internal, menggunakan *email/username* dan *password*.

5.3.2 Implementasi API *Create Transaction*

Untuk aplikasi *merchant* dapat terintegrasi dengan aplikasi, disediakan satu *endpoint* API untuk membuat transaksi baru. Dengan detail sebagai berikut:

- *Host*: <http://lulapay.my.id/>
- *Endpoint*: /api/v1/transaction/create
- *Method*: POST
- *Headers*:
 - o *Content-Type*: application/json
 - o *Accept*: application/json
 - o *Public-Key*: 0edf10ab-5fe1-47cd-bfae-7985b6bd4ce5
 - o *Server-Key*: 302cdb73-95a2-4b47-bf54-82718bd0e109

Public-Key dan *Server-Key* merupakan autentikasi yang digunakan untuk dapat terhubung dengan aplikasi, autentikasi ini didapatkan pada Menu *Merchant*.

- *Body (raw)*:

```
{
  "invoice_code": "INV-J20230526000060",
  "name": "Febrian",
  "email": "febrianaries@gmail.com",
  "phone": "082115236938",
  "total_charged": 12000,
  "items": [
    {
      "item_name": "Item 1",
      "price": 2000,
      "quantity": 1
    }
  ]
}
```

```

},
{
  "item_name": "Item 2",
  "price": 5000,
  "quantity": 2
}
]
}

```

5.3.2.1 Percobaan Menggunakan CURL

```

febrrian@febrrians-MacBook-Pro ~ % curl --location 'https://lulapay.my.id/api/v1/transaction/create' \
--header 'Content-Type: application/json' \
--header 'Accept: application/json' \
--header 'Public-Key: 0edf10ab-5fe1-47cd-bfae-7985b6bd4ce5' \
--header 'Server-Key: 302cdb73-95a2-4b47-bf54-82718bd0e109' \
--data-raw '{
  "invoice_code": "INV-J20230526000060",
  "name": "Febrrian",
  "email": "febrrianaries@gmail.com",
  "phone": "082115236938",
  "total_charged": 12000,
  "items": [
    {
      "item_name": "Item 1",
      "price": 2000,
      "quantity": 1
    },
    {
      "item_name": "Item 2",
      "price": 5000,
      "quantity": 2
    }
  ]
}', {
  "error": false,
  "message": "Transaction Created Successfully",
  "redirect_url": "https://lulapay.my.id/transaction/46a80487-6305-4f31-8554-69aa84fb1d8b/cart"
}

```

Gambar 5.2 Curl API Create Transaction

5.3.2.2 Percobaan Menggunakan Postman

The screenshot shows the Postman interface for a POST request to `((BASE_URL))/api/v1/transaction/create`. The request body is a JSON object representing a transaction with an invoice code, name, email, phone, total charged, and an item. The response status is 201 Created, and the response body is a JSON object indicating success, a message, and a redirect URL.

```

1 {
2   "invoice_code": "INV-J20230526000061",
3   "name": "Febrian",
4   "email": "febrianaries@gmail.com",
5   "phone": "082115236938",
6   "total_charged": 12000,
7   "items": [
8     {
9       "item_name": "Item 1",
10      "price": 2000,
11      "quantity": 1
12    }
13  ]

```

```

1 {
2   "error": false,
3   "message": "Transaction Created Successfully",
4   "redirect_url": "https://lulapay.my.id/transaction/
f867b985-28e1-4ecf-ae48-8420ce03e088/cart"
5

```

Gambar 5.3 Postman API Create Transaction

5.3.2.3 Hasil / Response API

Dari dua cara di atas, masing-masing menghasilkan hasil seperti berikut:

```

1 {
2   "error": false,
3   "message": "Transaction Created Successfully",
4   "redirect_url": "https://lulapay.my.id/transaction/
f867b985-28e1-4ecf-ae48-8420ce03e088/cart"
5

```

Gambar 5.4 Response API Create Transaction

Berikut penjelasan dari masing-masing kolom di atas:

- ***error***: Merupakan *flag* atau penanda bahwa API di atas *error* atau tidak, terdiri dari *false* atau *true*.
- ***message***: Merupakan informasi dari Response API.
- ***redirect_url***: Merupakan url yang nantinya digunakan oleh pengguna untuk melakukan pembayaran secara *online*.

5.3.3 Implementasi API Check Status

API ini digunakan untuk aplikasi *merchant* melakukan cek status pada aplikasi. Dengan detail sebagai berikut:

- *Host*: <http://lulipay.my.id/>
- *Endpoint*: /api/v1/transaction/status
- *Method*: GET
- *Headers*:
 - o *Content-Type*: application/json
 - o *Accept*: application/json
 - o *Public-Key*: 0edf10ab-5fe1-47cd-bfae-7985b6bd4ce5
 - o *Server-Key*: 302cdb73-95a2-4b47-bf54-82718bd0e109

Public-Key dan *Server-Key* merupakan autentikasi yang digunakan untuk dapat terhubung dengan aplikasi, autentikasi ini didapatkan pada Menu *Merchant*.

- *Query Params*:
invoice_code : {{INVOICE_ID}}

5.3.3.1 Percobaan Menggunakan CURL

```
febrrian@Febrians-MacBook-Pro ~ % curl --location 'https://lulapay.my.id/api/v1/transaction/status?invoice_code=INV-J20230526000060' \
--header 'Content-Type: application/json' \
--header 'Accept: application/json' \
--header 'Public-Key: 0edf10ab-5fe1-47cd-bfae-7985b6bd4ce5' \
--header 'Server-Key: 302cdb73-95a2-4b47-bf54-8271bd0e189' \
--header 'Cookie: october_session=eyJpdiI6ImhQV1zRFR2XC90bE41MFV0RmxGNct3PT0iLCJ2YWx1ZSI6IlZKSFAYs3NEdFdWnjFcL0xCee5yS3ZQcxZN0UdkSEUxMituNUVqtT3Njd0t0CdgrNVremVmVzRKRTVoVTQUTzVMDnP0vzVm92Y1wvbkdxsVwvOvpreBZUJU3yTEh5UVhYnpPOUF4dUzwU12aG5YeWMxeHjp0lx2FNMXdcL3VOQiisIm1hYYI6IjViYzM40TQ5Yjk2MWZjNmQzYjhMCiyZmUwNDVmZjVmZDg1ZjRKYjhmyWU0ZGYxZGV10WVkoGI5njRizjY5NzIif0%3D%3D'
{"error":false,"message":"Transaction found","data":{"transaction_id":"46a80487-6305-4f31-8554-69aa84fb1d8b","invoice_code":"INV-J20230526000060","transaction_status":"Pending"}}

febrrian@Febrians-MacBook-Pro ~ %
```

Gambar 5.5 Curl API Check Status

5.3.3.2 Percobaan Menggunakan Postman

The screenshot shows the Postman interface for a GET request. The URL is `({BASE_URL})/api/v1/transaction/status?invoice_code=INV-J20230526000060`. The 'Params' tab is selected, showing a table with one row for 'invoice_code' set to 'INV-J20230526000060'. The 'Body' tab shows a JSON response with the following structure:

```

1
2   "error": false,
3   "message": "Transaction found",
4   "data": {
5     "transaction_id": "46a80487-6305-4f31-8554-69aa84fb1d8b",
6     "invoice_code": "INV-J20230526000060",
7     "transaction_status": "Pending"
8   }
9

```

Gambar 5.6 Postman API Check Status

5.3.3.3 Hasil / Response API

Dari dua cara di atas, masing-masing menghasilkan hasil seperti berikut:

```
{
  "error": false,
  "message": "Transaction found",
  "data": {
    "transaction_id": "46a80487-6305-4f31-8554-69aa84fb1d8b",
    "invoice_code": "INV-J20230526000060",
    "transaction_status": "Pending"
  }
}
```

Gambar 5.7 Response API Check Status

Berikut penjelasan dari masing-masing kolom di atas:

- **error**: Merupakan *flag* atau penanda bahwa API di atas error atau tidak, terdiri dari false atau true.
- **message**: Merupakan informasi dari *Response API*.
- **data**: Menampilkan data transaksi yang dipanggil.

5.3.4 Implementasi API Untuk *Handle Notification Callback Midtrans*

API ini digunakan untuk aplikasi menerima notifikasi perubahan status yang terjadi pada setiap transaksi untuk kembali diproses datanya oleh aplikasi:

- *Host*: <http://lulipay.my.id/>
- *Endpoint*: /api/v1/transaction/notif/midtrans
- *Method*: POST

- *Body (raw)*:

```
{
  "transaction_time": "2023-05-30 00:55:22",
  "transaction_status": "paid",
  "transaction_id": "f8ab5a41-6a20-4948-bb67-9463c59d7103",
  "status_message": "midtrans payment notification",
  "status_code": "201",
  "signature_key": "3a0ab06f64f5f243b48b7e581b7933ef54262053348c9c518d37b6e30fb0599f7d292583677a83ba1f7e51bf7925cc673461c8a2810eb8b821986295ec2c3b83",
  "payment_type": "gopay",
  "order_id": "INV-J20230526000024|1685382922",
  "merchant_id": "G135392153",
  "gross_amount": "12000.00",
  "fraud_status": "accept",
  "expiry_time": "2023-05-30 01:10:22",
  "currency": "IDR"
}
```

Payload body di atas merupakan contoh notifikasi dari Midtrans yang dikirim ke aplikasi, dalam contoh, telah terjadi perubahan status dengan nomor invoice INV-J20230526000024 menjadi paid atau sukses. Sehingga proses selanjutnya, aplikasi harus melakukan perubahan data untuk mengubah status dengan *nomor invoice* tersebut menjadi berhasil.

5.3.4.1 Percobaan Menggunakan CURL

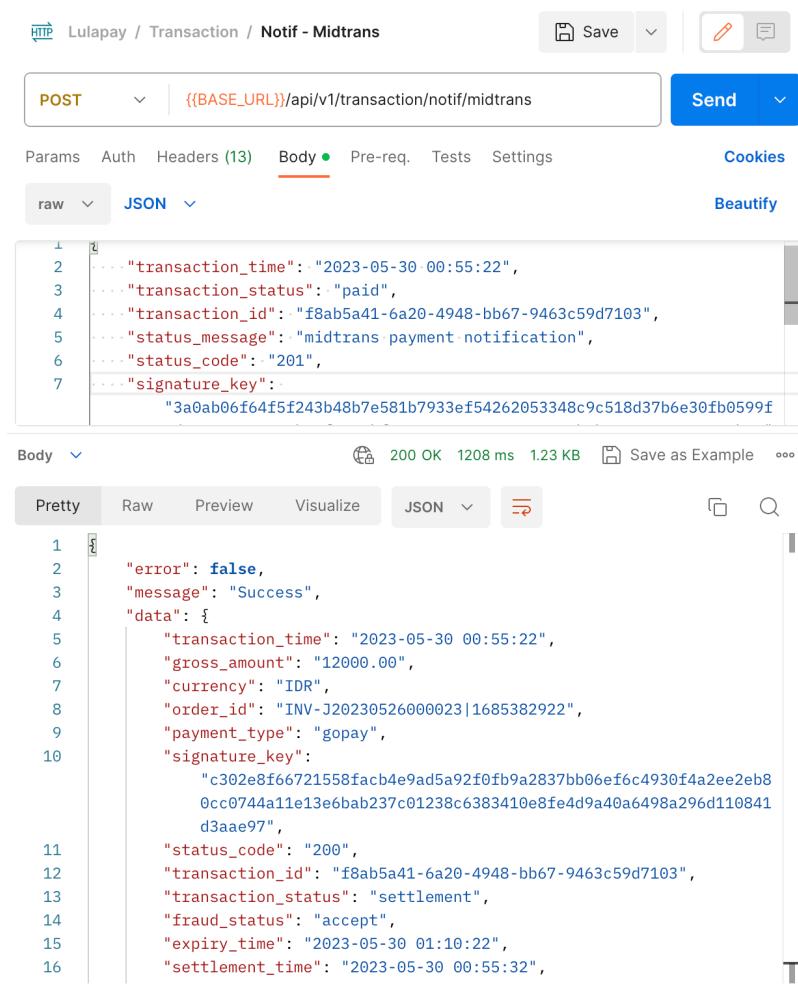
```
febriani@Febriani-MacBook-Pro ~ % curl --location 'https://lulipay.my.id/api/v1/transaction/notif/midtrans' \
--header 'Content-Type: application/json' \
--header 'Accept: application/json' \
--header 'Public-Key: 0edf1ab-5fe1-47cd-bfae-798b5b6bd4ce5' \
--header 'Server-Key: 382cd573-95e2-4b47-bf54-82718bd8e109' \
--data '{
    "transaction_time": "2023-05-30 00:55:22",
    "transaction_status": "paid",
    "transaction_id": "f8ab5a41-6a20-4948-bb67-9463c59d7103",
    "status_message": "midtrans payment notification",
    "status_code": "201",
    "signature_key": "3a0ab06f64f5f243b48b7e581b7933e54262053348c9c518d37b6e30fb0599f7d292583677a83be1f7e51bf7925cc673461c8a2810eb8b821986295ec2c3b83",
    "payment_type": "gopay",
    "order_id": "INV-J20230526000023|1685382922",
    "merchant_id": "0135392153",
    "gross_amount": "12000.00",
    "fraud_status": "accept",
    "expiry_time": "2023-05-30 01:10:22",
    "currency": "IDR"
}'

{"error":false,"message":"Success","data":{"transaction_time":"2023-05-30 00:55:22","gross_amount":"12000.00","currency":"IDR","order_id":"INV-J20230526000023|1685382922","merchant_id":"0135392153","payment_type":"gopay","signature_key":"3a0ab06f64f5f243b48b7e581b7933e54262053348c9c518d37b6e30fb0599f7d292583677a83be1f7e51bf7925cc673461c8a2810eb8b821986295ec2c3b83","status_message":"midtrans payment notification","status_code":201,"transaction_id":"f8ab5a41-6a20-4948-bb67-9463c59d7103","transaction_status":"settlement","settlement_time":"2023-05-30 00:55:32","status_message":"Success, transaction is found","merchant_id":"0135392153"}}

febriani@Febriani-MacBook-Pro ~ % top
```

Gambar 5.8 Curl API Handle Notification Callback Midtrans

5.3.4.2 Percobaan Menggunakan Postman



Gambar 5.9 Postman API Handle Notification Callback Midtrans

5.3.4.3 Hasil / Response API

Dari dua cara di atas, masing-masing menghasilkan hasil seperti berikut:

Body

Pretty Raw Preview Visualize JSON ↪

```
1
2     "error": false,
3     "message": "Success",
4     "data": {
5         "transaction_time": "2023-05-30 00:55:22",
6         "gross_amount": "12000.00",
7         "currency": "IDR",
8         "order_id": "INV-J20230526000023|1685382922",
9         "payment_type": "gopay",
10        "signature_key": "c302e8f66721558facb4e9ad5a92f0fb9a2837bb06ef6c4930f4a2ee2eb80cc0744a11e13e6bab237c01238c6383410e8fe4d9a40a6498a296d110841d3aae97",
11        "status_code": "200",
12        "transaction_id": "f8ab5a41-6a20-4948-bb67-9463c59d7103",
13        "transaction_status": "settlement",
14        "fraud_status": "accept",
15        "expiry_time": "2023-05-30 01:10:22",
16        "settlement_time": "2023-05-30 00:55:32",
17        "status_message": "Success, transaction is found",
18        "merchant_id": "G135392153"
19    }
20 }
```

Gambar 5.10 Response API Handle Notification Callback Midtrans

Message “Success” menandakan notifikasi telah berhasil diterima dan status transaksi sudah dirubah sesuai data yang diterima dari Midtrans. Sehingga nomor *invoice* tersebut kini statusnya sudah berubah menjadi “*Paid*” pada aplikasi.

5.3.5 Implementasi API Untuk *Handle Notification Callback Brankas*

API ini digunakan untuk aplikasi menerima notifikasi perubahan status yang terjadi pada setiap transaksi untuk kembali diproses datanya oleh aplikasi:

- *Host*: <http://lulipay.my.id/>
- *Endpoint*: /api/v1/transaction/notif/brankas
- *Method*: POST
- *Body (raw)*:

```
{
  "transaction_id": "c1ffb76f-476c-46e0-ab77-683a71213e36",
  "status": 2,
  "reference_id": "b1d1e8a3-2cb5-44ff-be0d-9f12011b1758"
}
```

Payload body di atas merupakan contoh notifikasi dari Brankas yang dikirim ke aplikasi, dalam contoh, telah terjadi perubahan status dengan *reference_id* b1d1e8a3-2cb5-44ff-be0d-9f12011b1758 menjadi 2, *reference_id* merupakan ID transaksi yang telah dienkripsi dan disimpan ke dalam *database* aplikasi *payment gateway aggregator*. Seperti yang dilihat statusnya bukan dalam bentuk informasi namun dalam bentuk data angka, berikut adalah tabel referensi kode angka status dari dokumentasi resmi Brankas:

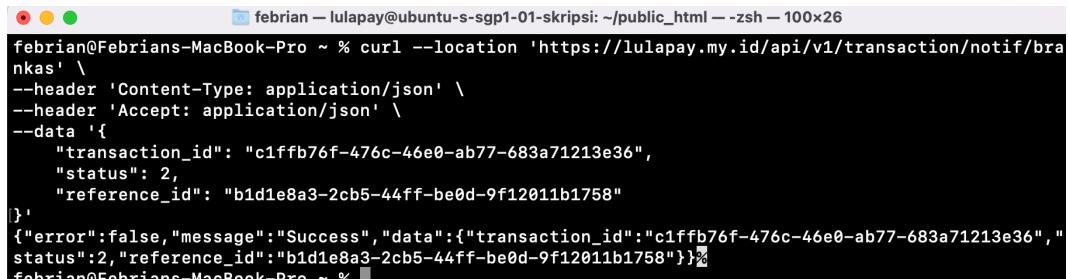
Transaction Status Guide

STATUS	VALUE	DESCRIPTION
0	PENDING	Pending operation.
1	AWAITING_TFA	Awaiting TFA authentication.
2	SUCCESS	Successful operation.
3	ERROR	Error during operation.
4	LOGIN_ERROR	Login Error.
5	INVOICE_CREATED	Checkout invoice created.
6	AWAITING_LOGIN	Login in progress.
7	AWAITING_LOGIN_TFA	Awaiting login TFA Authentication.
8	AWAITING_ACCOUNT_SELECT	Awaiting account selection.
9	AWAITING_TRANSFER_TFA	Awaiting transfer TFA Authentication.
10	IN_PROGRESS	Transfer is in progress.
11	EXPIRED	Transaction Expired.
12	FLAGGED	Transaction Flagged - receiver needs to do manual verification and settlement.
13	CANCELLED	Transaction Cancelled - sender cancelled the transaction.
14	DENIED	Transaction Denied - sender denied the consent.
15	FAILED	Transaction Failed - receiver was not able to receive the funds.

Gambar 5.11 Referensi kode status Brankas

Dapat disimpulkan bahwa status = 2 adalah SUCCESS, Sehingga proses selanjutnya, aplikasi harus melakukan perubahan data untuk mengubah status dengan nomor *invoice* tersebut menjadi berhasil.

5.3.5.1 Percobaan Menggunakan CURL

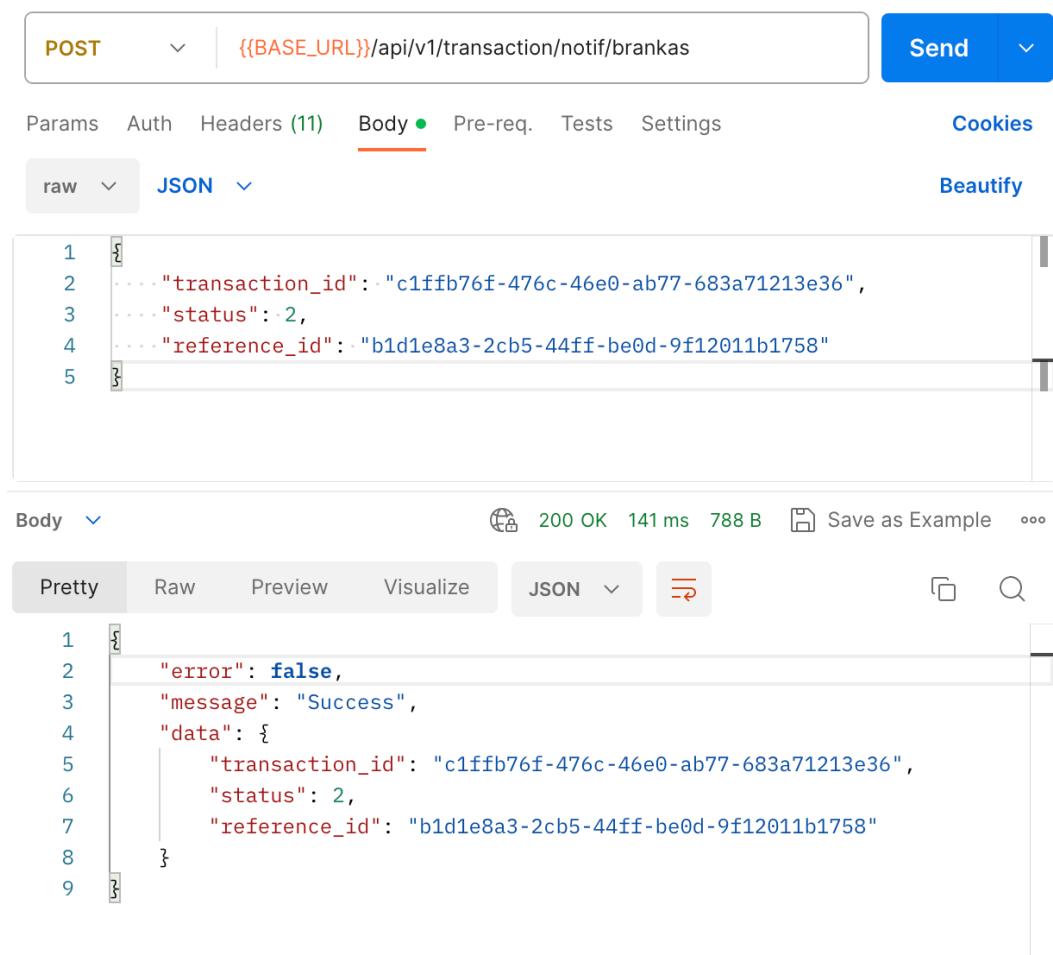


```
febrrian@Febrians-MacBook-Pro ~ % curl --location 'https://lulapay.my.id/api/v1/transaction/notif/brankas' \
--header 'Content-Type: application/json' \
--header 'Accept: application/json' \
--data '{
    "transaction_id": "c1ffb76f-476c-46e0-ab77-683a71213e36",
    "status": 2,
    "reference_id": "b1d1e8a3-2cb5-44ff-be0d-9f12011b1758"
}'
{"error":false,"message":"Success","data":{"transaction_id":"c1ffb76f-476c-46e0-ab77-683a71213e36","status":2,"reference_id":"b1d1e8a3-2cb5-44ff-be0d-9f12011b1758"}}

febrrian@Febrians-MacBook-Pro ~ %
```

Gambar 5.12 Curl API Handle Notification Callback Brankas

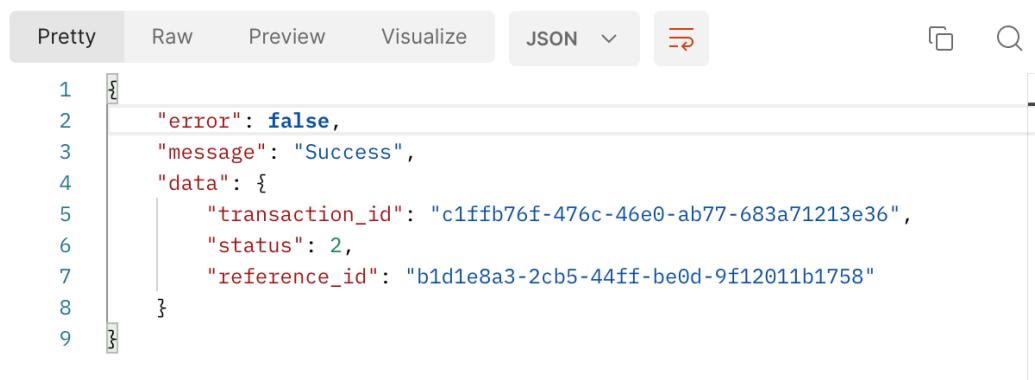
5.3.6 Percobaan Menggunakan Postman



Gambar 5.13 Postman API Handle Notification Callback Brankas

5.3.6.1 Hasil / Response API

Dari dua cara di atas, masing-masing menghasilkan hasil seperti berikut:



```

1 {
2   "error": false,
3   "message": "Success",
4   "data": {
5     "transaction_id": "c1ffb76f-476c-46e0-ab77-683a71213e36",
6     "status": 2,
7     "reference_id": "b1d1e8a3-2cb5-44ff-be0d-9f12011b1758"
8   }
9 }

```

Gambar 5.14 Response API Handle Notification Callback Brankas

Message “Success” menandakan notifikasi telah berhasil diterima dan status transaksi sudah diubah sesuai data yang diterima dari Midtrans. Sehingga nomor invoice tersebut kini statusnya sudah berubah menjadi “Paid” pada Aplikasi *payment gateway aggregator*.

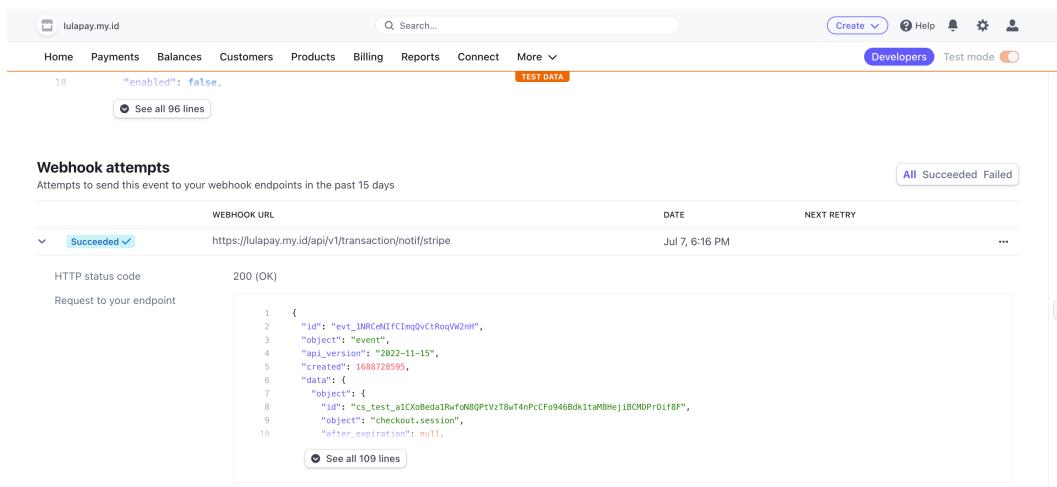
5.3.7 Implementasi API Untuk *Handle Notification Callback Stripe*

API ini digunakan untuk aplikasi menerima notifikasi perubahan status yang terjadi pada setiap transaksi untuk kembali diproses datanya oleh aplikasi:

- *Host*: <http://lulapay.my.id/>
- *Endpoint*: /api/v1/transaction/notif/stripe
- *Method*: POST

5.3.7.1 Hasil Implementasi *Handle Notification Callback Stripe*

Untuk pengujian notifikasi Stripe sedikit berbeda, karena menggunakan transaksi menggunakan *token signature* yang sangat ketat, sehingga tidak bisa diujicoba secara manual, penulis hanya menyediakan *endpoint* POST dan melakukan implementasi menggunakan SDK dari *website resmi Stripe*. Setelah dilakukan implementasi SDK pada Aplikasi *Payment Gateway Aggregator*, berikut adalah hasil uji coba yang bisa dilihat dari *dashboard Stripe*:



The screenshot shows the Stripe dashboard under the 'Webhook attempts' section. It lists a single successful attempt:

WEBHOOK URL	DATE	NEXT RETRY
https://lulapay.my.id/api/v1/transaction/notif/stripe	Jul 7, 6:16 PM	...

Details of the attempt:

- HTTP status code: 200 (OK)
- Request to your endpoint: https://lulapay.my.id/api/v1/transaction/notif/stripe
- Date: Jul 7, 6:16 PM
- Next retry: ...

The JSON response body is partially visible:

```

1   {
2     "id": "evt_1NRCnHFCImqQvCtRqyW2mH",
3     "object": "event",
4     "api_version": "2022-11-15",
5     "created": 168872859,
6     "data": {
7       "object": {
8         "id": "cs_test_a1CXoBeda1RwfNBQPtvT8wT4nPccF0946Bdk1taM8Hej1BCHDPr0if8F",
9         "object": "checkout.session",
10        "after_expiration": null
11      }
12    }
13  }
  
```

Gambar 5.15 Hasil Implementasi *Handle Notification Callback Stripe*

Hasil “*Succeeded*” pada gambar di atas membuktikan bahwa *Aplikasi Payment Gateway Aggregator* telah berhasil menerima dengan baik. Sesaat setelah notifikasi diterima oleh aplikasi, akan ada proses perubahan data pada data transaksi di aplikasi.

5.3.8 Implementasi *Scheduler Expiry Transaction*

Scheduler expiry transaction merupakan perintah yang dijalankan setiap satu menit sekali untuk melakukan pengecekan terhadap transaksi yang tidak dilakukan pembayaran sampai menyentuh batas waktu pembayaran yang telah ditentukan.

Untuk implementasi ini, penulis mengembangkan satu buah *script* untuk dijalankan pada *command line*, dengan perintah sebagai berikut:

```
lulapay@ubuntu-s-sgp1-01-skrripsi:~/public_html$ sudo /usr/bin/php7.4 artisan transaction:set_expired_transaction
Running expiry transaction scheduler...
27/27 [██████████] 100%
Expiry transaction scheduler has been run successfully...
```

Gambar 5.16 Implementasi *Scheduler Expiry Transaction*

Terlihat pada gambar di atas perintah yang dijalankan adalah :

- sudo /usr/bin/php7.4 artisan transaction:set_expired_transaction

Dari hasil yang dijalankan menunjukkan bahwa ada 27 transaksi yang belum dilakukan pembayaran hingga menyentuh batas waktu pembayaran, sehingga 27 transaksi itu akan diubah statusnya dari *Pending* menjadi *Expired*.

Untuk membuat *script* tersebut berjalan setiap satu menit sekali, penulis melakukan pengaturan pada Cron Job. Cron job merupakan salah satu *command line* yang bekerja untuk mengeksekusi tugas terjadwal secara otomatis, pengaturan Cron Job nya adalah seperti berikut:

- * * * * * sudo /usr/bin/php7.4 artisan transaction:set_expired_transaction

Pada pengaturan di atas, “* * * * *” berarti *every minute, every hour, every day, every month, every year* atau dengan kata lain, akan dijalankan setiap menit.

Implementasi Output

5.3.9 Halaman Pilih Metode Pembayaran

LULAPAY
Manage your payment easier

INFO PELANGGAN

Nama	:	Febrian
Alamat Email	:	febrianaries@gmail.com
Nomor Telepon	:	082115236938

Items

#	Item Name	Qty	Price
1	Tshirt	3	Rp. 40,000
Total			Rp. 120,000

Bank Transfer - Virtual Account

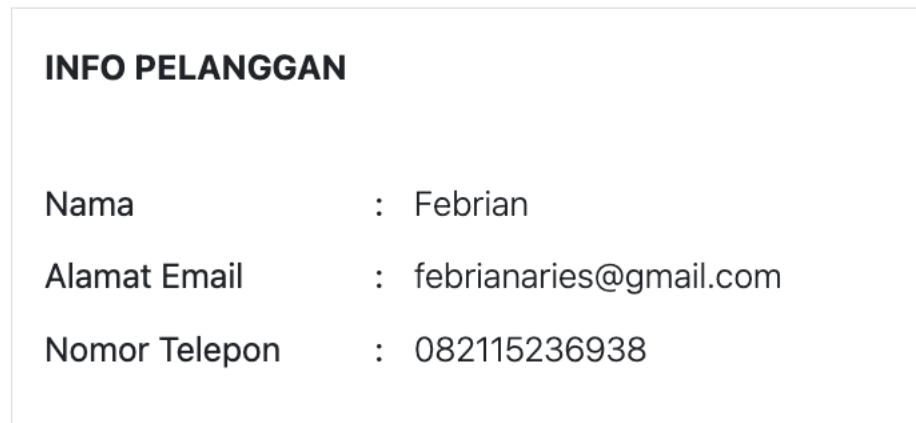
Mandiri Bill	Midtrans
mandiri	Pembayaran dapat melalui ATM Mandiri, Mandiri Online, Internet Banking, atau Teller Bank Mandiri
BNI VA	Midtrans
BNI	Pembayaran dapat melalui ATM BNI, Internet Banking, atau Teller Bank BNI
BRI VA	Midtrans

Gambar 5.17 Implementasi Halaman Pilih Metode Pembayaran

URL: <https://lulapay.my.id/transaction/7f303ff6-b252-4d51-80fc-0657379205a9/cart>

Dari *output* di atas, terdapat beberapa komponen dalam halaman tersebut, komponen utama yang ada pada halaman tersebut adalah sebagai berikut:

- a. Detail Pelanggan



Gambar 5.18 Implementasi Info Pelanggan

Menampilkan data detail pelanggan yang terdiri dari Nama, Alamat Email dan Nomor Telepon. Data ini akan sangat aman karena URL untuk halaman ini bersifat sangat unik dan akan dilakukan enkripsi.

Dengan menerapkan enkripsi di atas, data pengguna akan cenderung aman karena hanya pengguna tersebut yang akan mendapatkan URL tersebut.

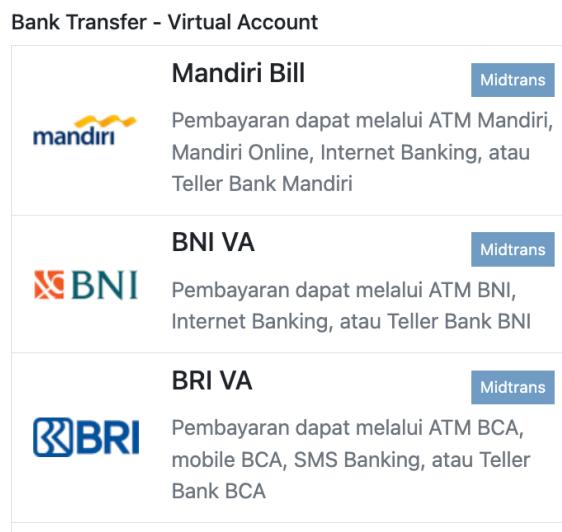
b. *Items*

Items			
#	Item Name	Qty	Price
1	Tshirt	3	Rp. 40,000
Total			Rp. 120,000

Gambar 5.19 Implementasi Accordion Item

Menampilkan detail pembelian produk pelanggan yang akan dibayarkan. *Items* berupa *accordion* yang dapat diklik oleh user.

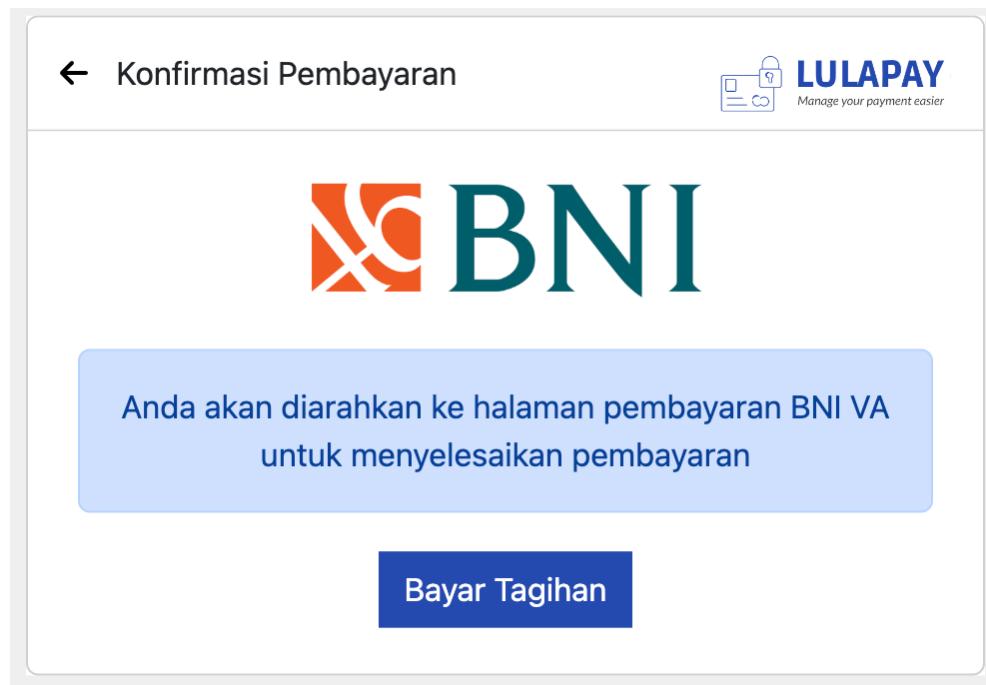
c. Metode Pembayaran



Gambar 5.20 Implementasi Kategori / Tipe Metode Pembayaran

Merupakan metode pembayaran yang dapat dipilih oleh pengguna, metode pembayaran terdiri dari tiga komponen yaitu tipe metode pembayaran, judul, dan juga deskripsi metode pembayaran.

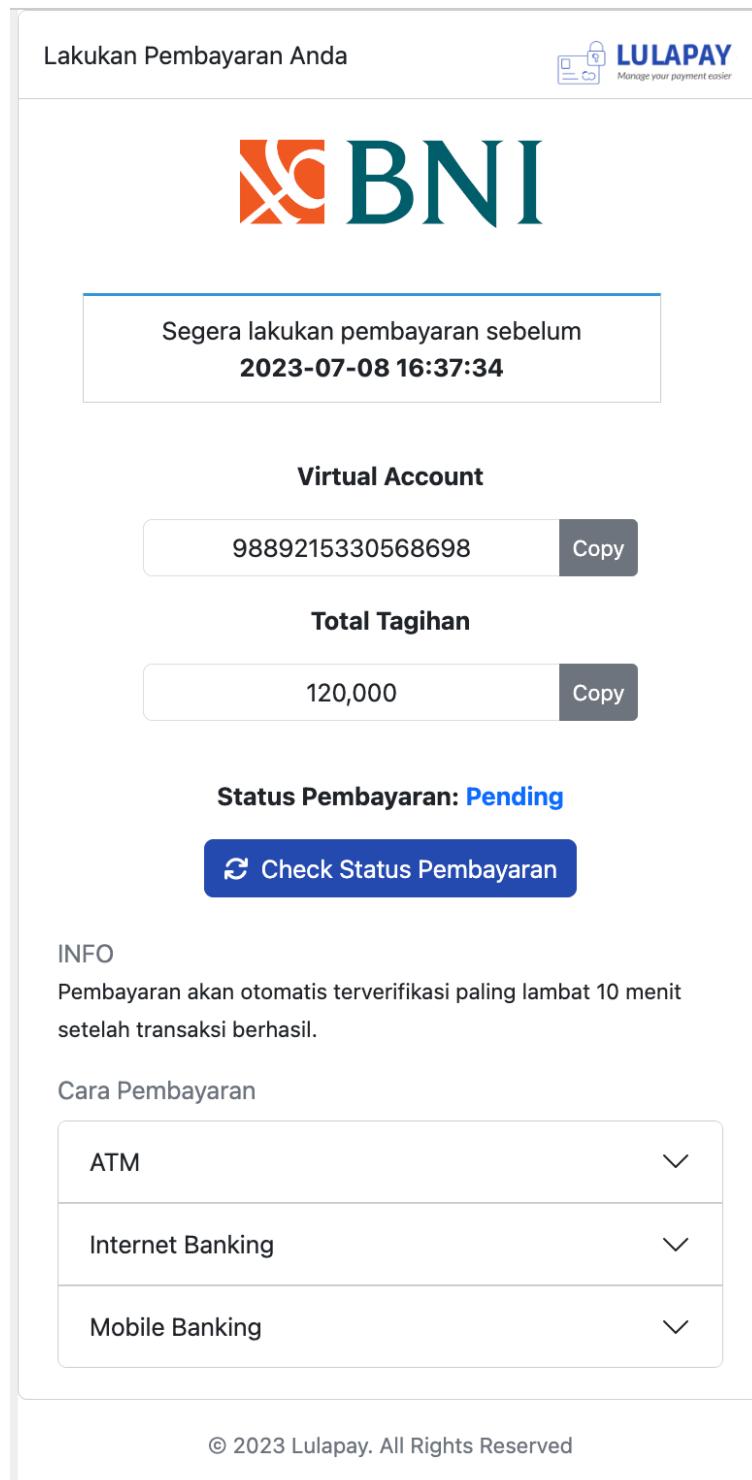
5.3.10 Halaman Konfirmasi Pembayaran



Gambar 5.21 Implementasi Halaman Konfirmasi Pembayaran

Halaman ini ditampilkan agar pengguna dapat melakukan konfirmasi terlebih dahulu sebelum memilih metode pembayaran, sehingga jika tidak sesuai, pengguna dapat kembali ke halaman sebelumnya.

5.3.11 Halaman Pembayaran *Payment Gateway* Midtrans



Gambar 5.22 Implementasi Halaman Pembayaran

Dari *output* pada halaman sebelumnya, terdapat beberapa komponen dalam halaman tersebut, komponen utama yang ada pada halaman tersebut adalah sebagai berikut:

- a. Logo Metode Pembayaran



Gambar 5.23 Implementasi Logo Metode Pembayaran

Menampilkan logo metode pembayaran yang telah dipilih sebelumnya oleh pengguna di “Halaman Pilih Metode Pembayaran”.

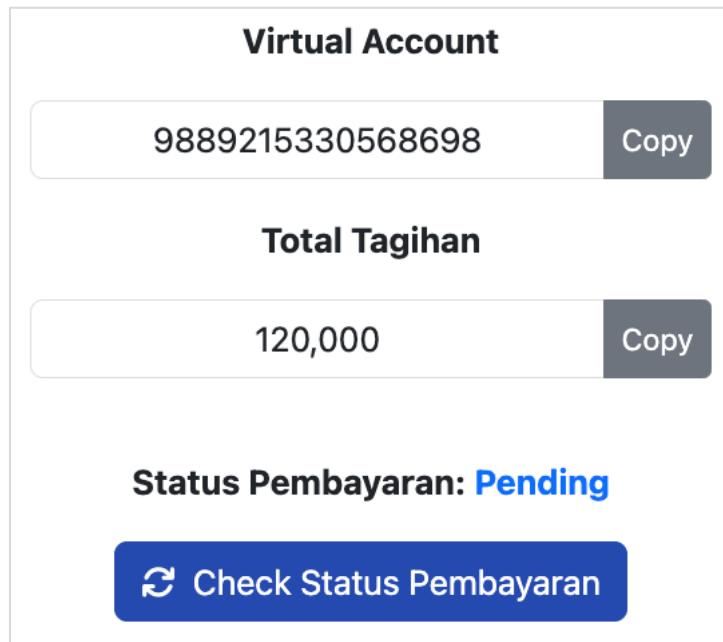
- b. Batas Waktu Pembayaran

Segera lakukan pembayaran sebelum
2023-07-08 16:37:34

Gambar 5.24 Batas Waktu Pembayaran

Pada bagian ini, pengguna dapat mengetahui batas waktu pembayaran atau batas waktu akhir pelunasan dari produk yang akan dibayarkan.

c. Informasi Pembayaran



Gambar 5.25 Implementasi Informasi Pembayaran

Pada bagian ini, pengguna akan ditampilkan *output* berupa informasi nomor *virtual account* dan Total Tagihan yang bisa dibayarkan. Selain itu, pengguna juga dapat melihat status dari pembayaran yang sedang dilakukan beserta tombol untuk melakukan *refresh* untuk mendapatkan data status terbaru dari *payment gateway*.

d. Informasi dan Tata Cara Pembayaran

INFO

Pembayaran akan otomatis terverifikasi paling lambat 10 menit setelah transaksi berhasil.

Cara Pembayaran

ATM

1. Masukkan kartu ATM dan PIN

2. Pilih Pembayaran Melalui VA

3. Masukkan **70012** sebagai kode perusahaan Midtrans, lalu pilih **Benar**

4. Masukkan 12 kode pembayaran
9889215330568698, lalu pilih tombol **Benar**

5. Masukkan Angka **1** untuk memilih tagihan, lalu pilih tombol **Ya**

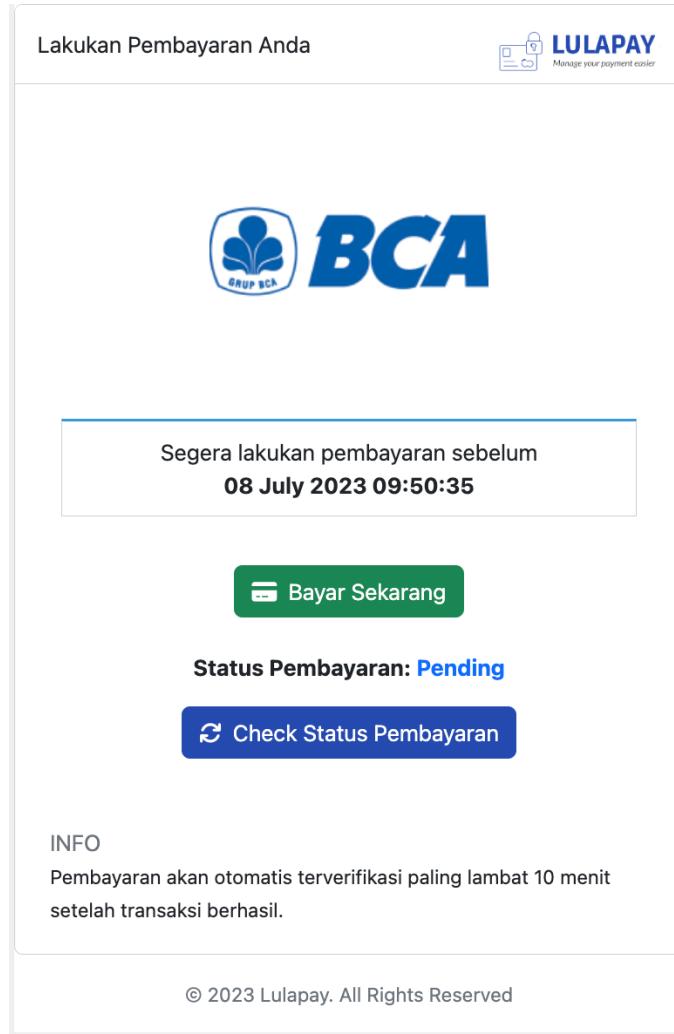
6. Akan muncul konfirmasi pembayaran, lalu pilih tombol **Ya**

Gambar 5.26 Implementasi Informasi dan Tata Cara Pembayaran

Menampilkan informasi penting berkaitan dengan transaksi dan menampilkan tata cara pembayaran dalam bentuk accordion.

5.3.12 Halaman Pembayaran *Payment Gateway* Brankas

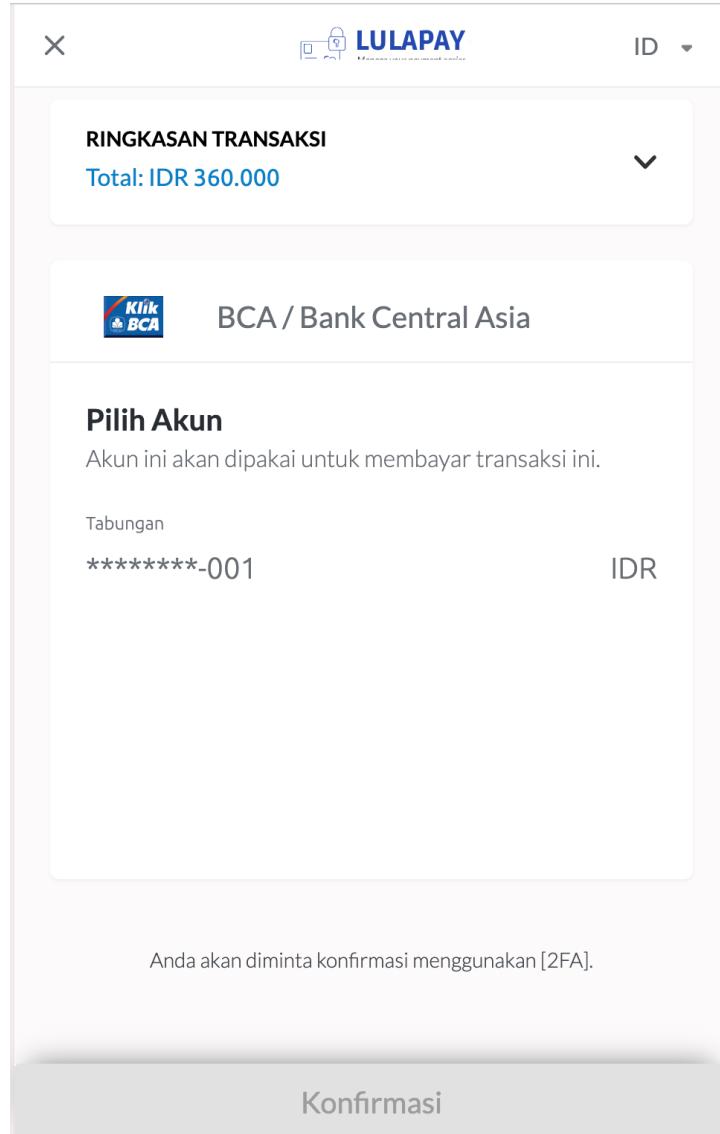
Berbeda dengan Midtrans, *payment gateway* mempunyai *output* berbeda dikarenakan transaksi tidak dilakukan di Aplikasi *Payment Gateway Aggregator* melainkan di website nya Brankas, berikut adalah tampilannya:



Gambar 5.27 Implementasi Halaman Pembayaran Brankas

Jika pembayaran dilakukan di luar aplikasi, akan ada satu tombol tambahan yaitu “Bayar Sekarang”, dimana jika di klik akan diarahkan ke website *payment gateway* untuk melakukan pembayaran:

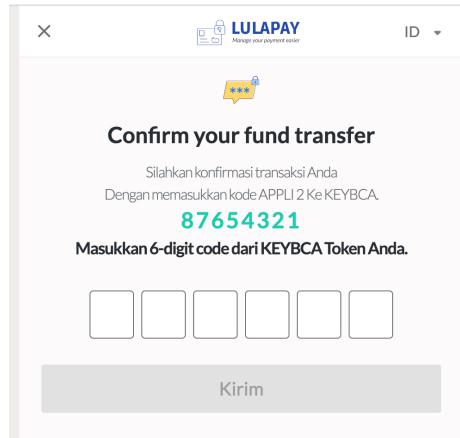
1. Halaman Konfirmasi Brankas



Gambar 5.28 Implementasi Halaman Konfirmasi Brankas

Akses URL: <https://tap.sandbox.brank.as/accounts?l=id>

2. Halaman PIN APPLI2

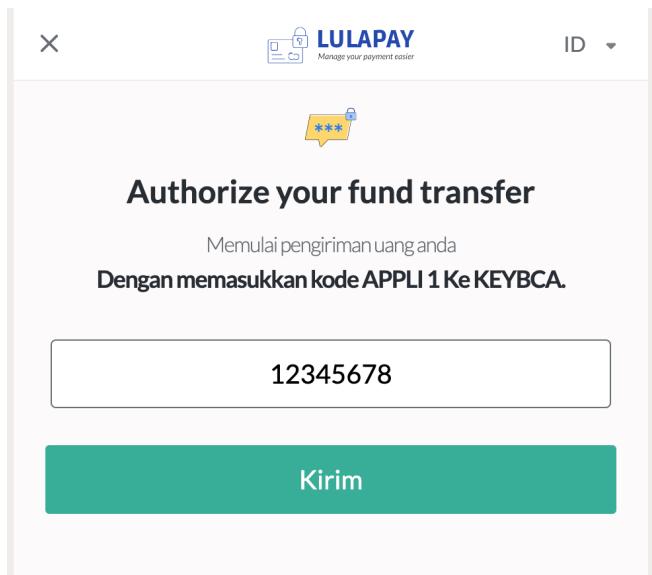


Gambar 5.29 Implementasi Halaman Konfirmasi APPLI 2 Brankas

Akses URL: <https://tap.sandbox.brank.as/tfa?type=transfer&l=id>

Berdasarkan informasi dari tim Brankas, untuk *environment* sandbox, pin APPLI2 adalah: 123456.

3. Halaman PIN APPLI 1

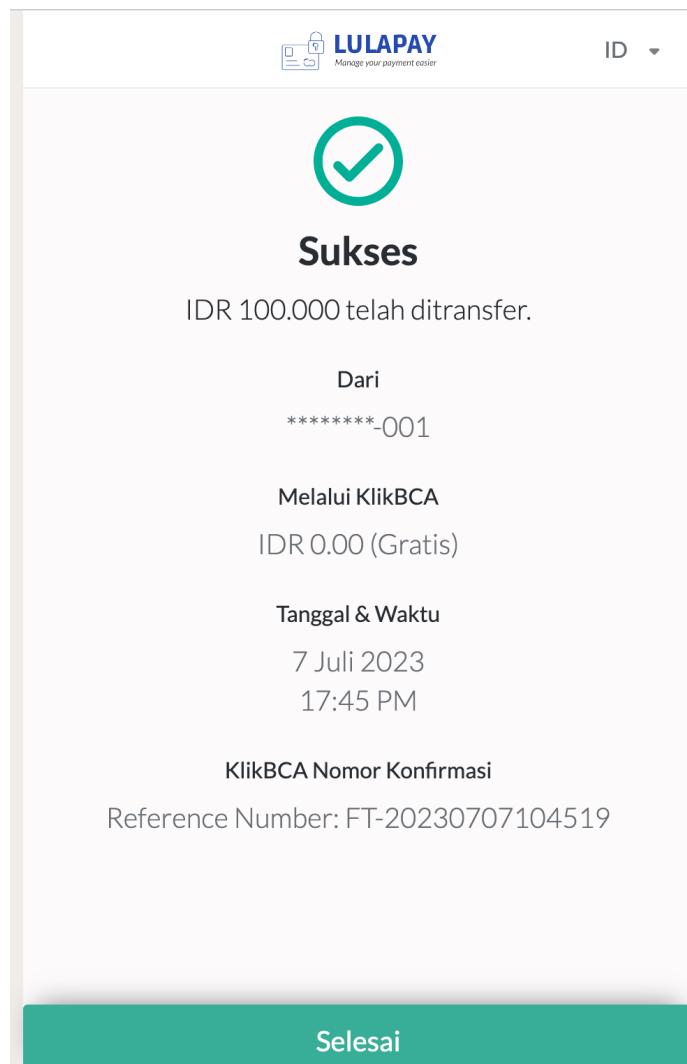


Gambar 5.30 Implementasi Halaman Konfirmasi APPLI 1 Brankas

Akses URL: <https://tap.sandbox.brank.as/tfa?type=transfer&l=id>

Berdasarkan informasi dari tim Brankas, untuk *environment sandbox*, PIN APPLI2 adalah: 12345678.

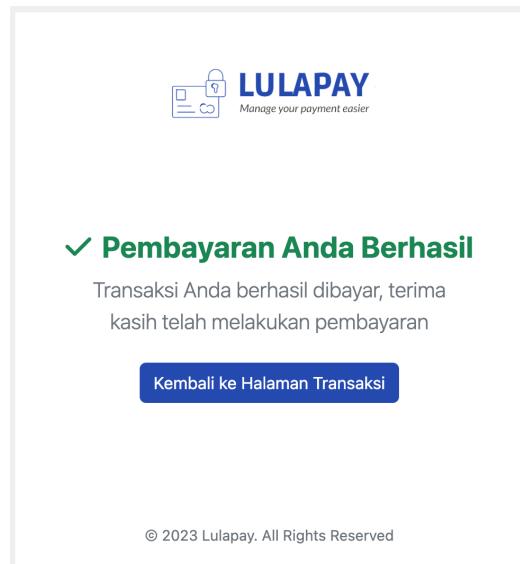
4. Halaman Informasi Pembayaran Brankas



Gambar 5.31 Halaman Informasi Pembayaran Success Brankas

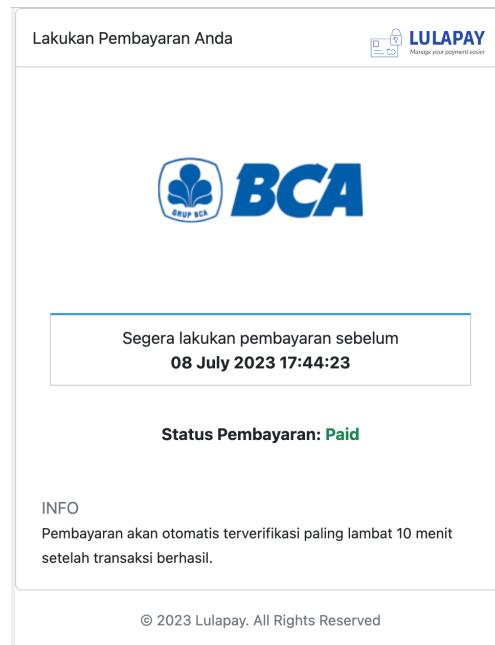
Akses URL: <https://tap.sandbox.brank.as/success?l=id>

5. Halaman Informasi Pembayaran Aplikasi *Payment Gateway Aggregator*



Gambar 5.32 Halaman Informasi Pembayaran Aplikasi

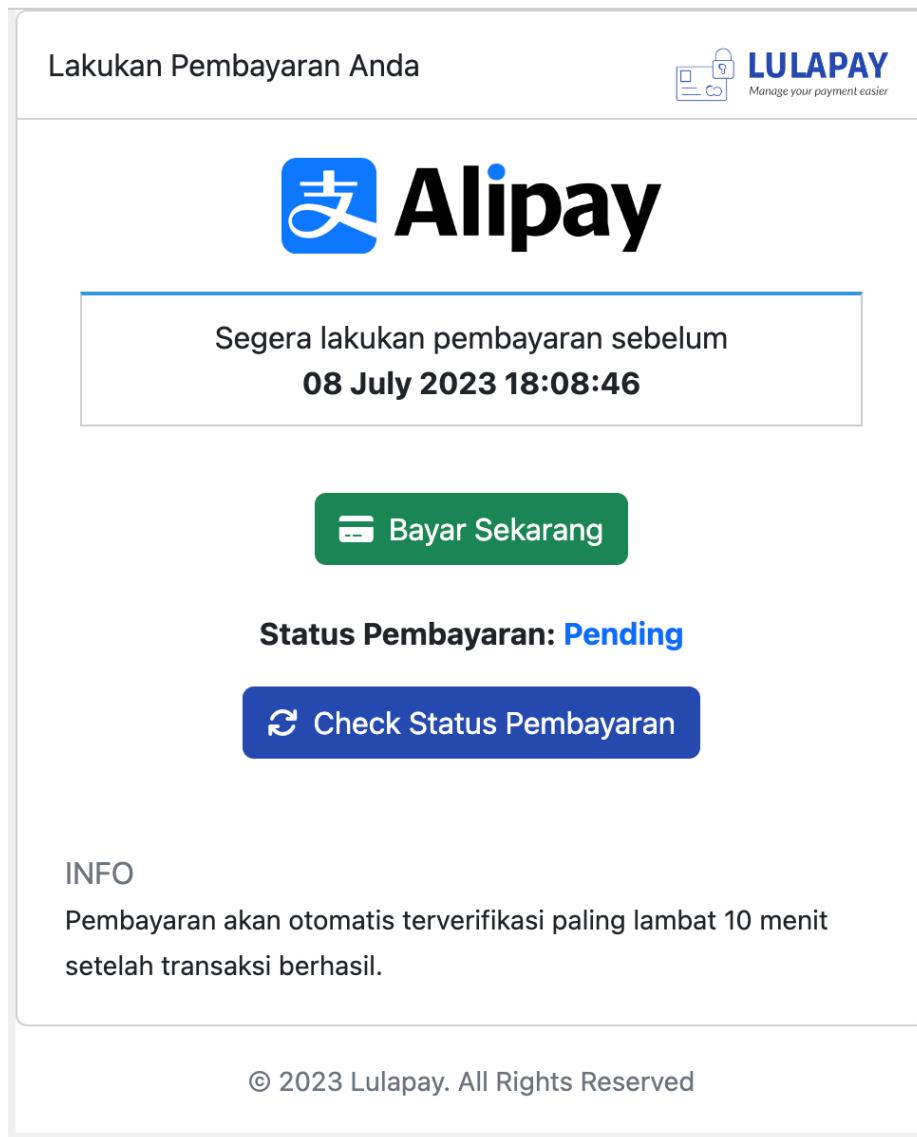
Jika sudah berhasil, akan otomatis diarahkan kembali ke aplikasi *payment gateway aggregator* seperti pada gambar di atas. Secara otomatis juga pada informasi pembayaran di bawah ini akan berubah statusnya menjadi “*Paid*”.



Gambar 5.33 Halaman Pembayaran

5.3.13 Halaman Pembayaran *Payment Gateway* Stripe

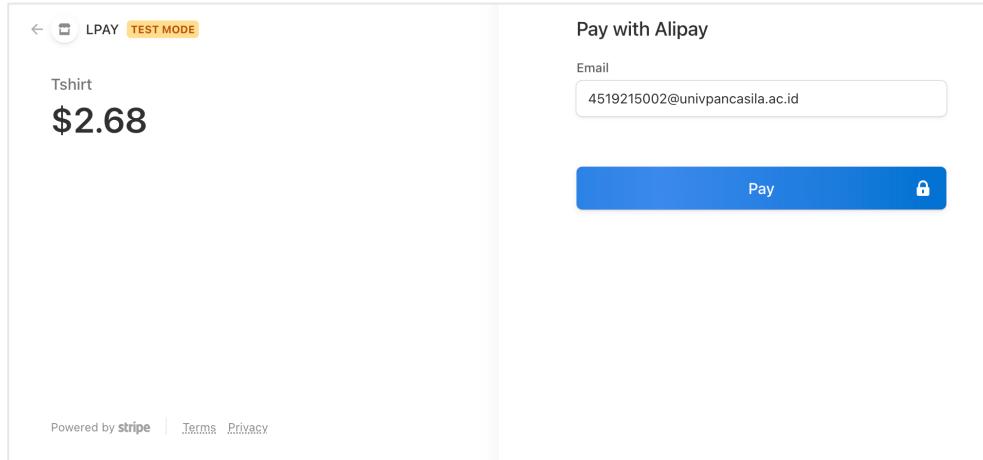
Sama seperti *payment gateway* Brankas, Stripe mempunyai *output* berbeda dikarenakan transaksi tidak dilakukan di aplikasi *payment gateway aggregator* melainkan di website Stripe, berikut adalah tampilannya:



Gambar 5.34 Implementasi Halaman Pembayaran Stripe

Jika pembayaran dilakukan di luar aplikasi, akan ada satu tombol tambahan yaitu “Bayar Sekarang”, dimana jika di klik akan diarahkan ke *website payment gateway* untuk melakukan pembayaran:

6. Halaman Konfirmasi Brankas

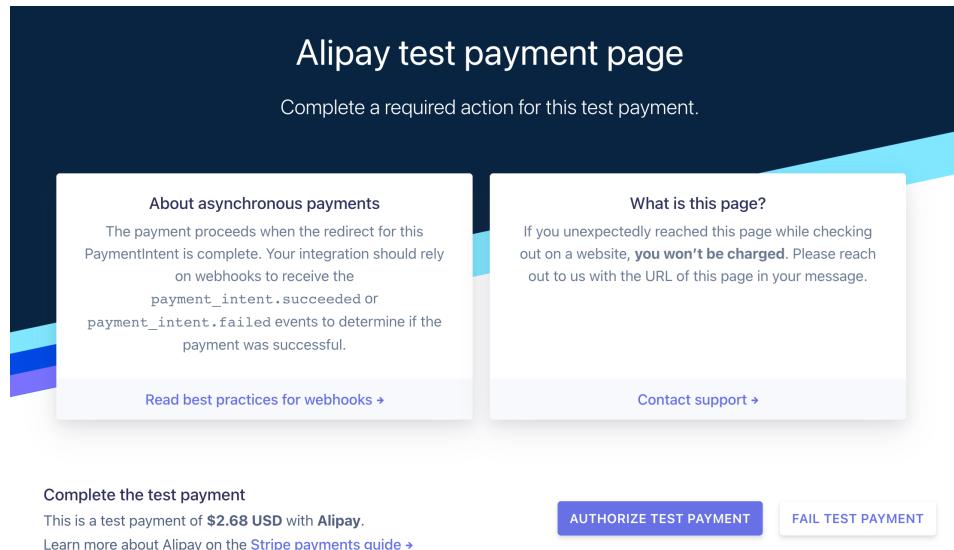


Gambar 5.35 Implementasi Halaman Konfirmasi Stripe

Akses URL:

https://checkout.stripe.com/c/pay/cs_test_a1CXoBeda1RwfoN8QPtVzT8wT4nPcCFo946Bdk1taM8HejiBCMDPrOif8F#fidkdWxOYHwnPyd1blpxYHZxWjA0S0BCckpMY0ZMaHRUc0Zxd2BWUWcxM0AydX9LcDxGb1VyN2k0YGFocDd8YWBUMVQyMV1KNUlmRDQ0TnxKMDV8ZH89TlIwYWNHPVddaU01UWJqPWByaWhGNTVDTE1qYXFOXCcpJ2N3amhWYHdzYHcnP3F3cGApJ2lkfGpwcVF8dWAnPyd2bGtiaWBabHFgaCcpJ2BrZGdpYFVpZGZgbWppYWb3dic%2FcXdwYHgl

7. Halaman Pembayaran Stripe



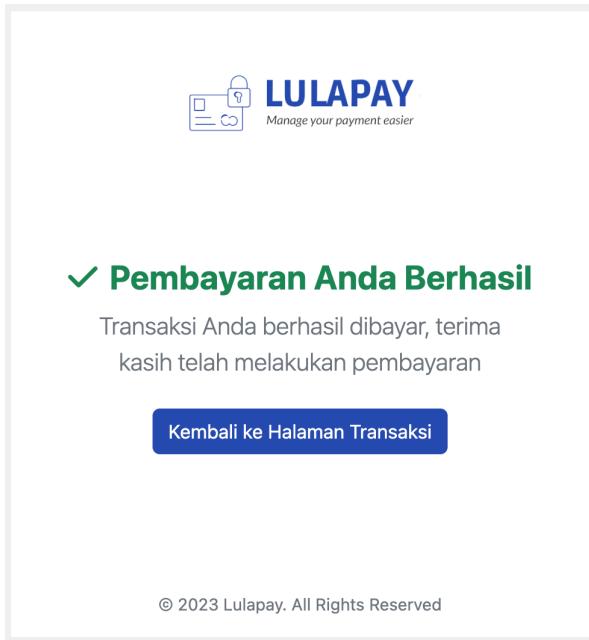
Gambar 5.36 Implementasi Halaman Pembayara Stripe

Akses URL:

```
https://stripe.com/sources/test_source?amount=268&currency=usd&livemode=false&return_url=https%3A%2F%2Fhooks.stripe.com%2Fadapter%2Falipay%2Fredirect%2Fcomplete%2Fsrc_1NR
CaCIIfCImqQvCtgXIhuGvr%2Fsrc_client_secret_PT9t0nF4M0P14hX8XdSNweRE&source_slug
=CBsaFwoVYWNjdF8xTkVHd09JZkNJbXFRdkN0KJLmn6UGMgY-
UlErFU46LdgnpK0z5oieyk5P0wEuAF3xvLZssBp20kU_lfxTPU8ydiwmlqjsHao--
_dTbg&type=alipay&usage=single_use
```

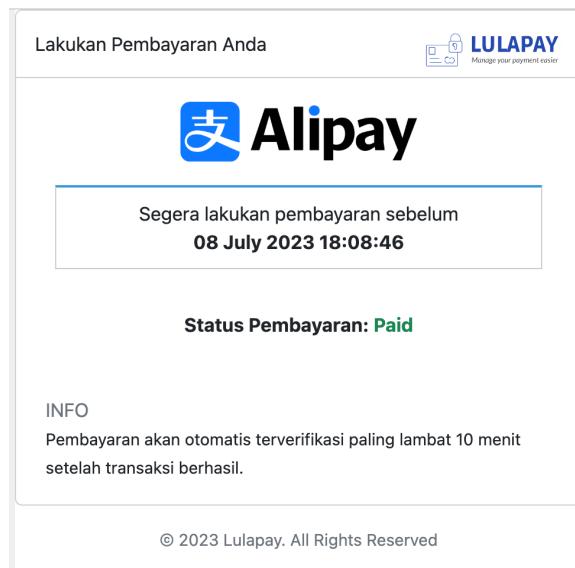
Halaman di atas adalah halaman *test* pembayaran pada *website* Stripe, untuk melakukan percobaan sukses bayar, klik “AUTHORIZE TEST PAYMENT”, sebaliknya, jika ingin transaksi gagal. Klik “FAIL TEST PAYMENT”. Pada kesempatan ini, saya akan klik “AUTHORIZE TEST PAYMENT”.

8. Halaman Informasi Pembayaran Aplikasi *Payment Gateway Aggregator*



Gambar 5.37 Halaman Informasi Pembayaran Aplikasi

Jika sudah berhasil, akan otomatis diarahkan kembali ke aplikasi *payment gateway aggregator* seperti pada gambar di atas. Secara otomatis juga pada informasi pembayaran di bawah ini akan berubah statusnya menjadi “Paid”.



Gambar 5.38 Halaman Pembayaran

5.4 Implementasi Input

5.4.1 Input Data Backend Users

The screenshot shows a 'New Administrator' form. At the top, there's a header 'Administrators' and a sub-header 'New Administrator'. The form has several input fields: 'Login *' and 'Email *' (both required), 'First Name' and 'Last Name', 'Password *' and 'Confirm Password *', and a 'Role' section with two options: 'Super Admin' (Site administrator with access to developer tools) and 'Merchant' (User's merchant). There's also an 'Avatar' field with a placeholder 'Avatar' and a file upload icon. At the bottom, there are three buttons: 'Create' (highlighted in blue), 'Create and close', and 'Cancel'.

Gambar 5.39 Implementasi Input Data Backend Users

Formulir di atas merupakan formulir data admin atau pengelola yang untuk masuk ke dalam aplikasi interna.

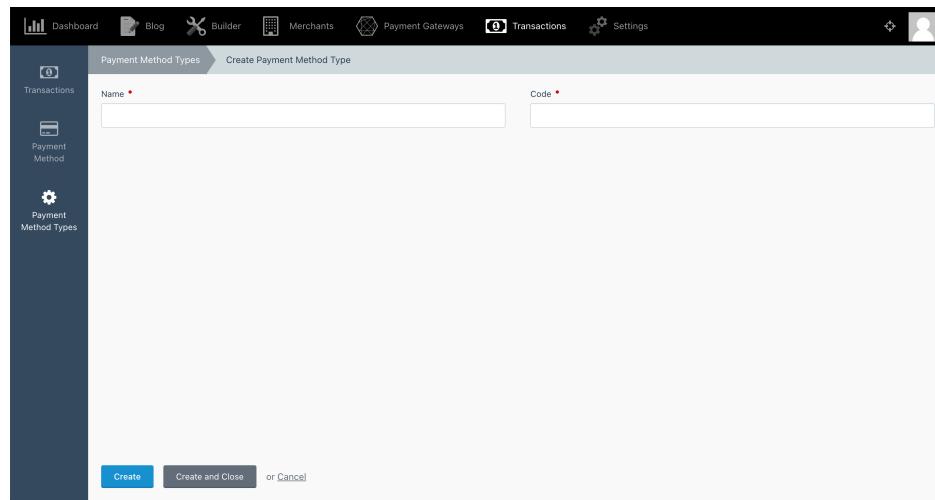
5.4.2 Input Data Metode Pembayaran

The screenshot shows a 'Create Payment Method' form. The header includes 'Transactions', 'Payment Method', and 'Payment Method Types'. The main form fields are: 'Provider *' (with a note 'Click the button to find a record'), 'Method Type *' (with a note 'Click the button to find a record'), 'Name *' and 'Code *', 'Description', 'Sandbox Simulator URL (if any)', and a 'Is Active?' checkbox. There's a 'Logo' field with a placeholder 'Logo' and a file upload icon. At the bottom, there are three buttons: 'Create' (highlighted in blue), 'Create and Close', and 'Cancel'.

Gambar 5.40 Implementasi Input Data Metode Pembayaran

Formulir di atas merupakan formulir metode pembayaran yang nantinya dapat digunakan digunakan oleh *merchant* sebagai opsi pembayarannya.

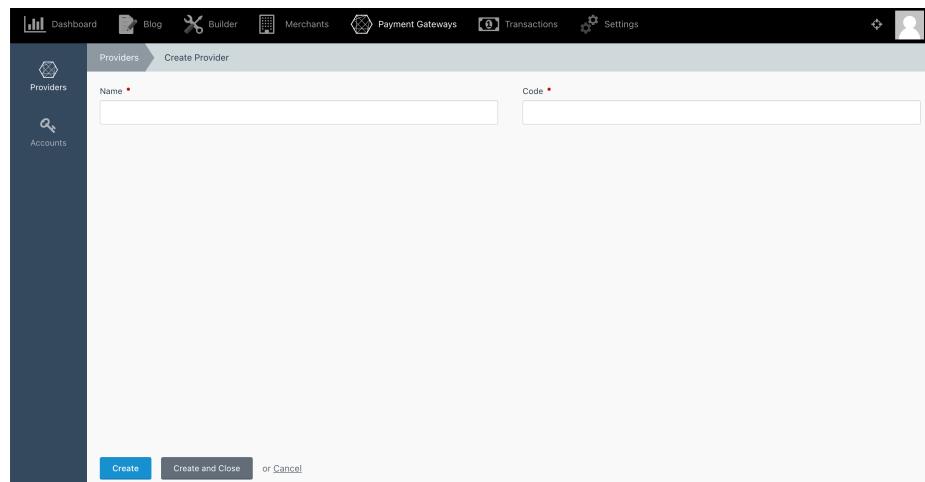
5.4.3 Input Data Tipe Metode Pembayaran



Gambar 5.41 Implementasi Input Data Tipe Metode Pembayaran

Formulir di atas adalah formulir data tipe metode pembayaran yang akan menjadi referensi dari formulir metode pembayaran.

5.4.4 Input Data Payment Gateway Provider



Gambar 5.42 Implementasi Input Data Payment Gateway Provider

Formulir di atas adalah formulir *payment gateway provider* yang dapat digunakan pada aplikasi. Nantinya data ini akan menjadi referensi pada formulir akun *payment gateway*, dan juga metode pembayaran.

5.4.5 Input Data Akun *Payment Gateway Provider*

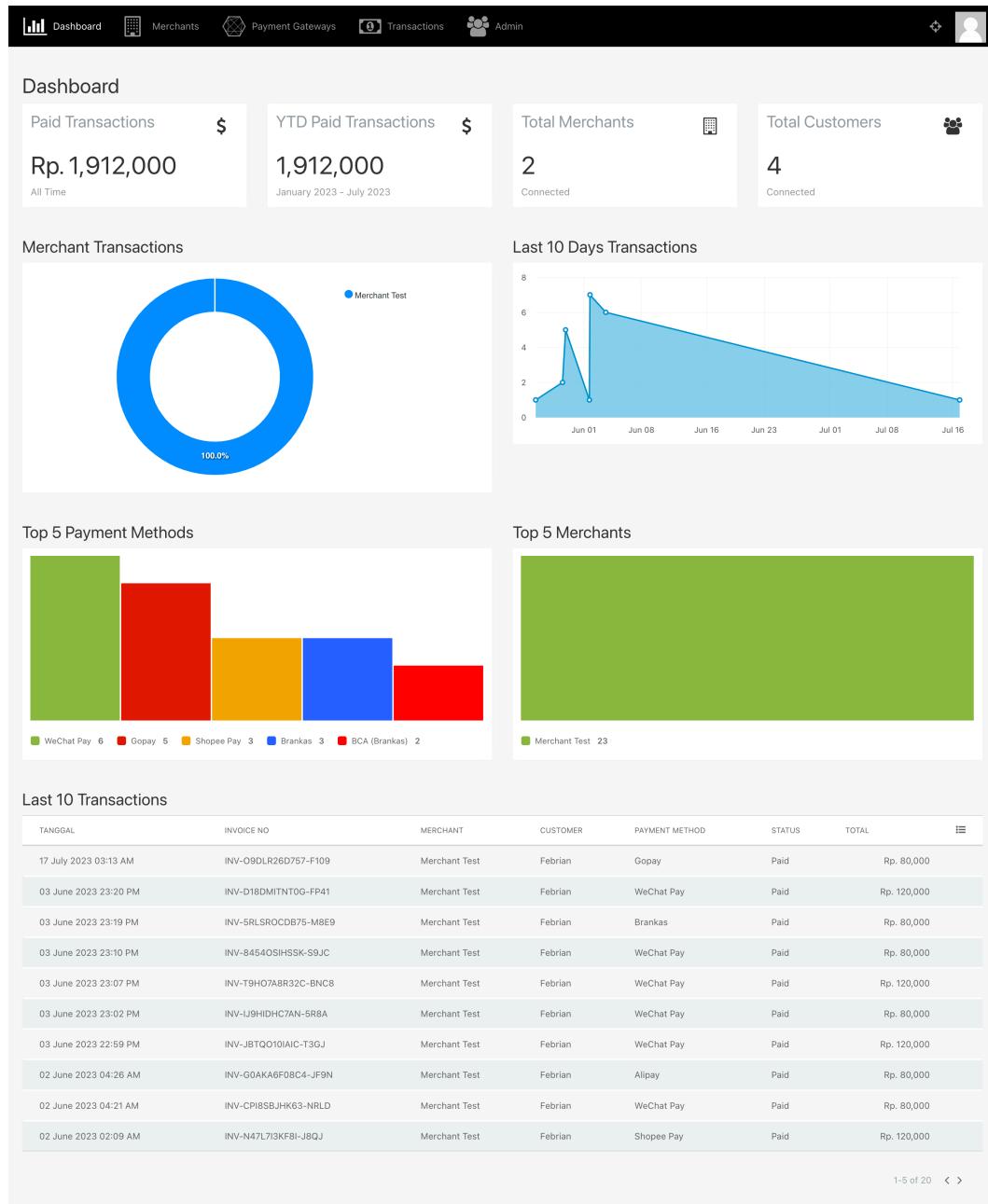
The screenshot shows a user interface for managing accounts. At the top, there's a navigation bar with links for Dashboard, Blog, Builder, Merchants, Payment Gateways (which is the active tab), Transactions, and Settings. On the left, a sidebar has sections for Providers and Accounts, with 'Accounts' being the active section. The main area is titled 'Create Account' under 'Accounts'. It has a search bar labeled 'Provider' with the placeholder 'Click the button to find a record'. Below it are several input fields: 'Name' (marked with a red asterisk), 'API Host (Base URL)' (marked with a red asterisk), 'Merchant Key', 'Client Key', and 'Server Key'. At the bottom, there are three buttons: a blue 'Create' button, a grey 'Create and Close' button, and a link 'or Cancel'.

Gambar 5.43 Implementasi Input Data Akun *Payment Gateway*

Formulir di atas adalah formulir akun *payment gateway provider* yang akan digunakan sebagai autentikasi aplikasi untuk terhubung dengan *payment gateway*.

5.5 Implementasi Menu

5.5.1 Dashboard

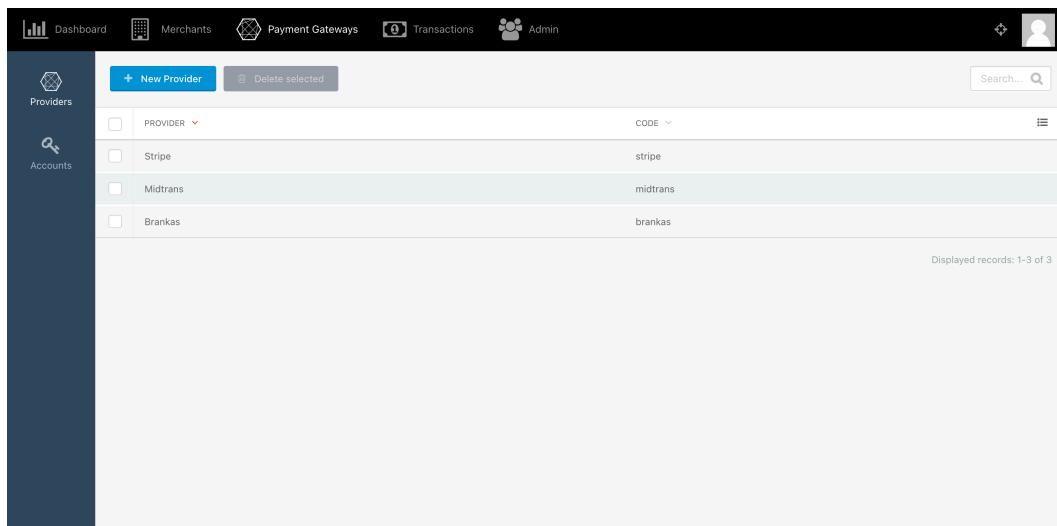


Gambar 5.44 Implementasi Menu Dashboard

Pada halaman ini, *admin* dapat melihat rangkuman data transaksi sukses yang terjadi pada aplikasi, data yang ditampilkan antara lain adalah:

1. *Paid Transactions*, merupakan data total transaksi secara keseluruhan.
2. *YTD Paid Transactions*, merupakan data total transaksi bulan ini.
3. *Total Merchants*, merupakan data total *merchant* yang sudah terhubung,
4. *Total Customers*, merupakan data total pelanggan yang sudah bertarnsaksi.
5. *Merchants Transaction*, data total transaksi berdasarkan *merchant*.
6. *Last 10 Days Transactions*, data transaksi 10 hari terakhir.
7. *Top 5 Payment Methods*, data metode pembayaran yang paling banyak digunakan berdasarkan total transaksi.
8. *Top 5 Merchants*, data transaksi *merchant* yang paling banyak melakukan transaksi berdasarkan total transaksi.
9. *Last 10 Transactions*, data 10 transaksi terakhir pada aplikasi.

5.5.2 Implementasi Menu *Payment Gateway Providers*



Gambar 5.45 Implementasi Menu Payment Gateway Providers

Menu ini digunakan untuk mengelola data *payment gateway provider* yang tersedia pada aplikasi.

5.5.3 Implementasi Menu Akun *Payment Gateway Providers*

Provider: all			
	NAME	MERCHANT KEY	PROVIDER
<input type="checkbox"/>	Stripe	Lulapay	Stripe
<input type="checkbox"/>	Febrian - Sandbox	G135392153	Midtrans
<input type="checkbox"/>	Brankas 1	Mohamad Febrian Mosil	Brankas

Displayed records: 1-3 of 3

Gambar 5.46 Implementasi Menu Payment Gateway Providers

Menu ini digunakan untuk mengelola data akun *payment gateway provider* yang tersedia pada aplikasi.

5.5.4 Implementasi Menu *Merchants*

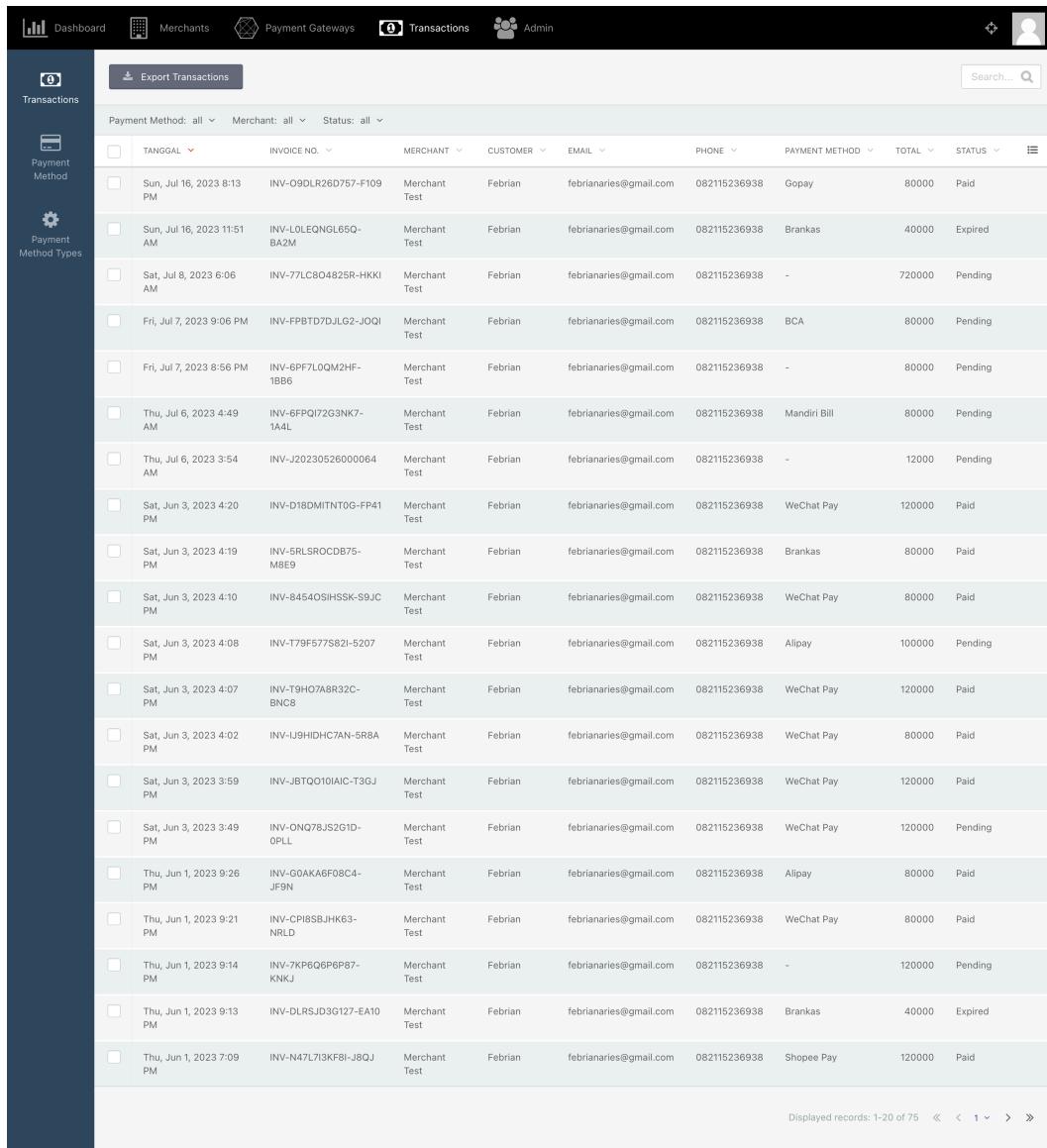
Merchant				
	NAME	EMAIL	PHONE	ADDRESS
<input type="checkbox"/>	Merchant Test	febianaries@gmail.com	082115236938	Test
<input type="checkbox"/>	GasMaj Bogor	gasmajbogor@localhost.com	08213123123	GasMaj Bogor

Displayed records: 1-2 of 2

Gambar 5.47 Implementasi Menu Merchant

Menu ini digunakan untuk mengelola data *merchant* yang tersedia pada aplikasi.

5.5.5 Implementasi Menu *Transactions*



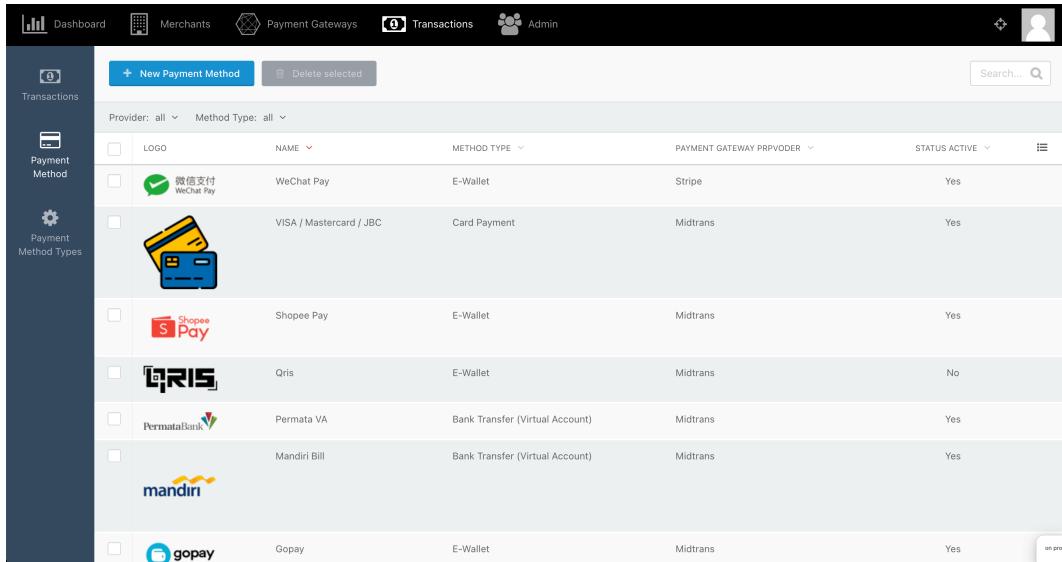
The screenshot shows the 'Transactions' section of a web application. The left sidebar has a dark blue background with icons for 'Transactions', 'Payment Method', and 'Payment Method Types'. The main area has a light gray header with tabs for 'Dashboard', 'Merchants', 'Payment Gateways', 'Transactions' (which is highlighted in blue), and 'Admin'. Below the header is a search bar with a magnifying glass icon and a 'Search...' placeholder. A button labeled 'Export Transactions' is also present. The main content area displays a table of transaction data with the following columns: TANGGAL, INVOICE NO., MERCHANT, CUSTOMER, EMAIL, PHONE, PAYMENT METHOD, TOTAL, and STATUS. The table contains 20 rows of transaction data, each with a checkbox in the first column. At the bottom right of the table, there is a message: 'Displayed records: 1-20 of 75 << < 1 > >>'.

	TANGGAL	INVOICE NO.	MERCHANT	CUSTOMER	EMAIL	PHONE	PAYMENT METHOD	TOTAL	STATUS	≡
<input type="checkbox"/>	Sun, Jul 16, 2023 8:13 PM	INV-O9DLR26D757-F109	Merchant Test	Febrian	febianaries@gmail.com	082115236938	Gopay	80000	Paid	
<input type="checkbox"/>	Sun, Jul 16, 2023 11:51 AM	INV-LOLEQNGL65Q-BA2M	Merchant Test	Febrian	febianaries@gmail.com	082115236938	Brankas	40000	Expired	
<input type="checkbox"/>	Sat, Jul 8, 2023 6:06 AM	INV-77LC804825R-HKKI	Merchant Test	Febrian	febianaries@gmail.com	082115236938	-	720000	Pending	
<input type="checkbox"/>	Fri, Jul 7, 2023 9:06 PM	INV-FPBTD7DJLG2-JOQI	Merchant Test	Febrian	febianaries@gmail.com	082115236938	BCA	80000	Pending	
<input type="checkbox"/>	Fri, Jul 7, 2023 8:56 PM	INV-6PF7L0QM2HF-1BB6	Merchant Test	Febrian	febianaries@gmail.com	082115236938	-	80000	Pending	
<input type="checkbox"/>	Thu, Jul 6, 2023 4:49 AM	INV-6FPQI7G3NK7-1A4L	Merchant Test	Febrian	febianaries@gmail.com	082115236938	Mandiri Bill	80000	Pending	
<input type="checkbox"/>	Thu, Jul 6, 2023 3:54 AM	INV-J20230526000064	Merchant Test	Febrian	febianaries@gmail.com	082115236938	-	12000	Pending	
<input type="checkbox"/>	Sat, Jun 3, 2023 4:20 PM	INV-D18DMITNT0G-FP4I	Merchant Test	Febrian	febianaries@gmail.com	082115236938	WeChat Pay	120000	Paid	
<input type="checkbox"/>	Sat, Jun 3, 2023 4:19 PM	INV-5RLSROCB75-M8E9	Merchant Test	Febrian	febianaries@gmail.com	082115236938	Brankas	80000	Paid	
<input type="checkbox"/>	Sat, Jun 3, 2023 4:10 PM	INV-84540SIHSSK-S9JC	Merchant Test	Febrian	febianaries@gmail.com	082115236938	WeChat Pay	80000	Paid	
<input type="checkbox"/>	Sat, Jun 3, 2023 4:08 PM	INV-T79F577S82I-5207	Merchant Test	Febrian	febianaries@gmail.com	082115236938	Alipay	100000	Pending	
<input type="checkbox"/>	Sat, Jun 3, 2023 4:07 PM	INV-T9HO7A8R32C-BNC8	Merchant Test	Febrian	febianaries@gmail.com	082115236938	WeChat Pay	120000	Paid	
<input type="checkbox"/>	Sat, Jun 3, 2023 4:02 PM	INV-IJ9HIDHC7AN-5R8A	Merchant Test	Febrian	febianaries@gmail.com	082115236938	WeChat Pay	80000	Paid	
<input type="checkbox"/>	Sat, Jun 3, 2023 3:59 PM	INV-JBTQO10AIC-T3GJ	Merchant Test	Febrian	febianaries@gmail.com	082115236938	WeChat Pay	120000	Paid	
<input type="checkbox"/>	Sat, Jun 3, 2023 3:49 PM	INV-ONQ78JS2G1D-OPLL	Merchant Test	Febrian	febianaries@gmail.com	082115236938	WeChat Pay	120000	Pending	
<input type="checkbox"/>	Thu, Jun 1, 2023 9:26 PM	INV-G0AKA6F08C4-JF9N	Merchant Test	Febrian	febianaries@gmail.com	082115236938	Alipay	80000	Paid	
<input type="checkbox"/>	Thu, Jun 1, 2023 9:21 PM	INV-CPI8SBJHK63-NRLD	Merchant Test	Febrian	febianaries@gmail.com	082115236938	WeChat Pay	80000	Paid	
<input type="checkbox"/>	Thu, Jun 1, 2023 9:14 PM	INV-7KP6Q6P6P87-KNKJ	Merchant Test	Febrian	febianaries@gmail.com	082115236938	-	120000	Pending	
<input type="checkbox"/>	Thu, Jun 1, 2023 9:13 PM	INV-DLRSJD3G127-EA10	Merchant Test	Febrian	febianaries@gmail.com	082115236938	Brankas	40000	Expired	
<input type="checkbox"/>	Thu, Jun 1, 2023 7:09 PM	INV-N47L73KF8I-J8QJ	Merchant Test	Febrian	febianaries@gmail.com	082115236938	Shopee Pay	120000	Paid	

*Gambar 5.48 Implementasi Menu *Transactions**

Menu ini digunakan untuk melihat data transaksi pada aplikasi. Berbeda dari menu yang lain, pada menu ini, admin hanya dapat melihat transaksi dan tidak disediakan akses untuk melakukan perubahan atau menghapus transaksi.

5.5.6 Implementasi Menu *Payment Methods*



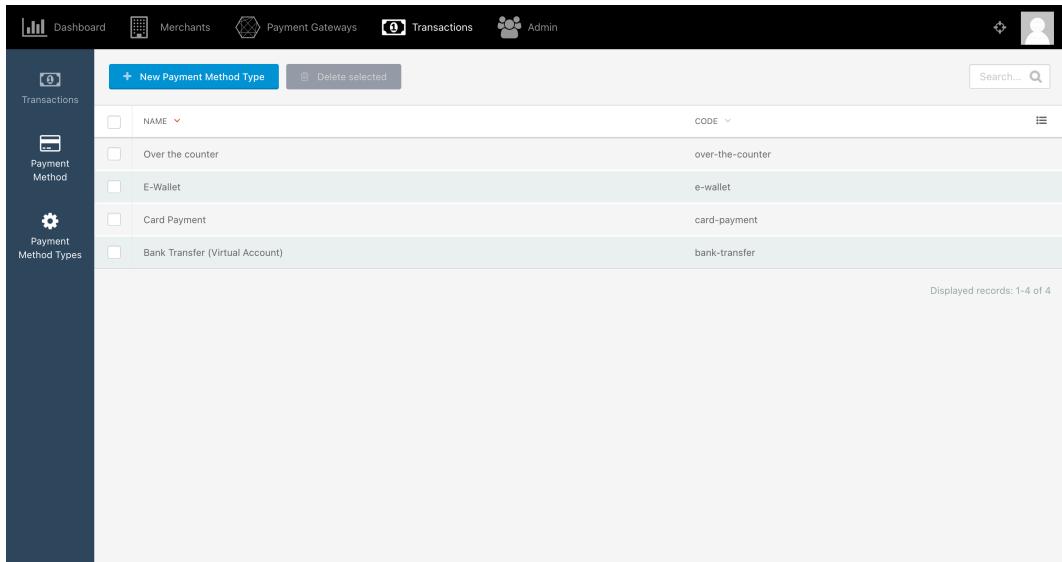
The screenshot shows a list of payment methods in a management interface. The columns include LOGO, NAME, METHOD TYPE, PAYMENT GATEWAY PROVIDER, and STATUS ACTIVE. The listed providers are WeChat Pay (E-Wallet, Stripe), VISA / Mastercard / JBC (Card Payment, Midtrans), Shopee Pay (E-Wallet, Midtrans), Qris (E-Wallet, Midtrans), PermataVA (Bank Transfer (Virtual Account), Midtrans), Mandiri Bill (Bank Transfer (Virtual Account), Midtrans), and Gopay (E-Wallet, Midtrans).

<input type="checkbox"/> LOGO	NAME	METHOD TYPE	PAYMENT GATEWAY PROVIDER	STATUS ACTIVE	
<input type="checkbox"/>	WeChat Pay	E-Wallet	Stripe	Yes	
<input type="checkbox"/>	VISA / Mastercard / JBC	Card Payment	Midtrans	Yes	
<input type="checkbox"/>	Shopee Pay	E-Wallet	Midtrans	Yes	
<input type="checkbox"/>	Qris	E-Wallet	Midtrans	No	
<input type="checkbox"/>	PermataVA	Bank Transfer (Virtual Account)	Midtrans	Yes	
<input type="checkbox"/>	Mandiri Bill	Bank Transfer (Virtual Account)	Midtrans	Yes	
<input type="checkbox"/>	Gopay	E-Wallet	Midtrans	Yes	

Gambar 5.49 Implementasi Menu *Payment Methods*

Menu ini digunakan untuk mengelola data metode *pembayaran* yang tersedia pada aplikasi.

5.5.7 Implementasi Menu *Payment Method Types*



The screenshot shows a list of payment method types in a management interface. The columns include NAME and CODE. The listed types are Over the counter (over-the-counter), E-Wallet (e-wallet), Card Payment (card-payment), and Bank Transfer (Virtual Account) (bank-transfer).

<input type="checkbox"/> NAME	CODE	
<input type="checkbox"/>	over-the-counter	
<input type="checkbox"/>	e-wallet	
<input type="checkbox"/>	card-payment	
<input type="checkbox"/>	bank-transfer	

Displayed records: 1-4 of 4

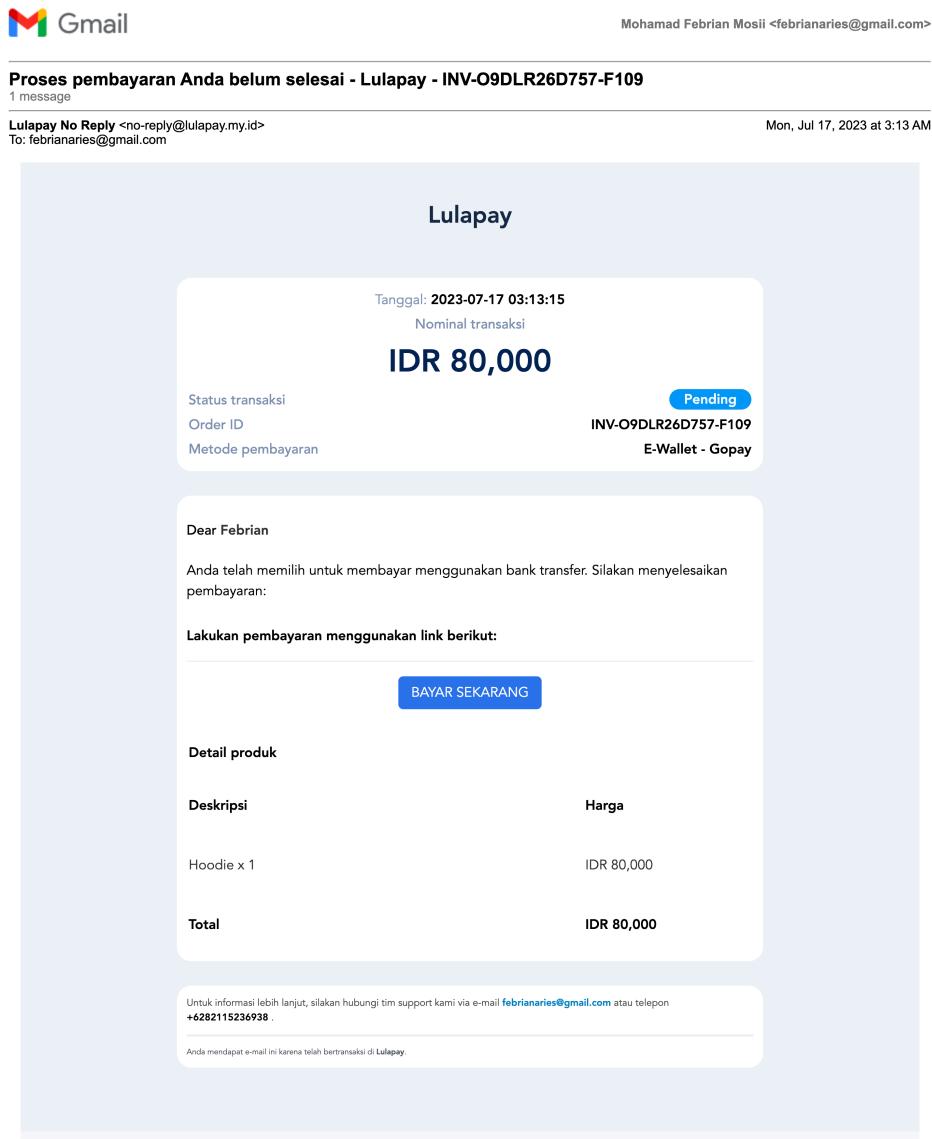
Gambar 5.50 Implementasi Menu *Payment Method Types*

Menu ini digunakan untuk mengelola data tipe metode *pembayaran* yang tersedia pada aplikasi.

5.5.8 Implementasi Email Notifikasi

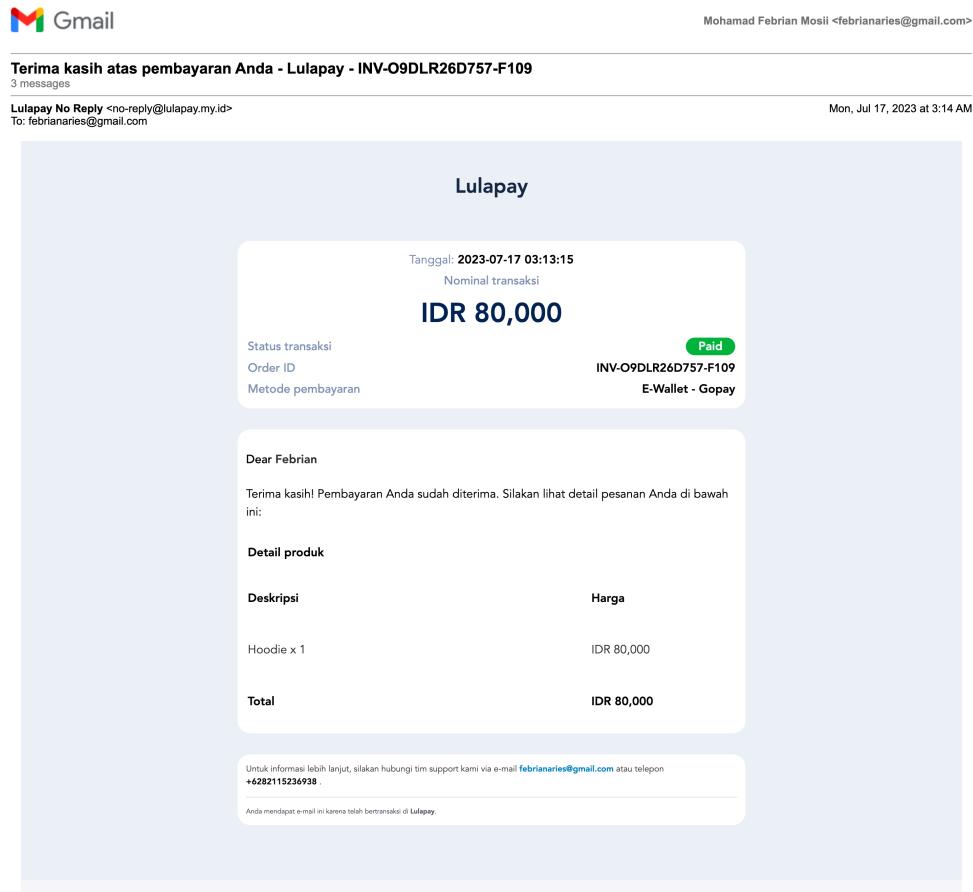
Setelah melakukan transaksi, pengguna juga akan menerima email baik pertama kali membuat transaksi dan pada setiap perubahan status pada transaksi terjadi, berikut adalah notifikasi email yang sudah diimplementasikan.

5.5.8.1 Transaksi *Pending*



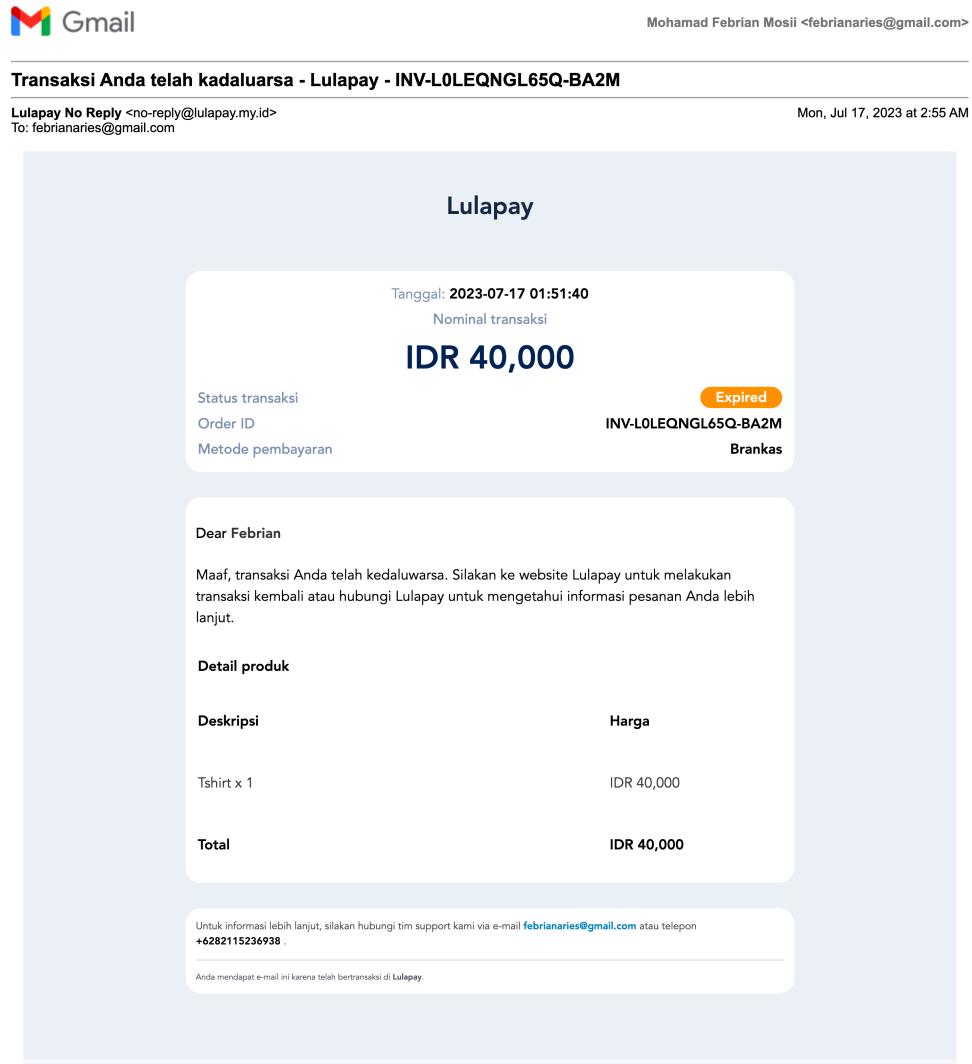
Gambar 5.51 Email Notifikasi Pending

5.5.8.2 Transaksi Sukses (*Paid*)



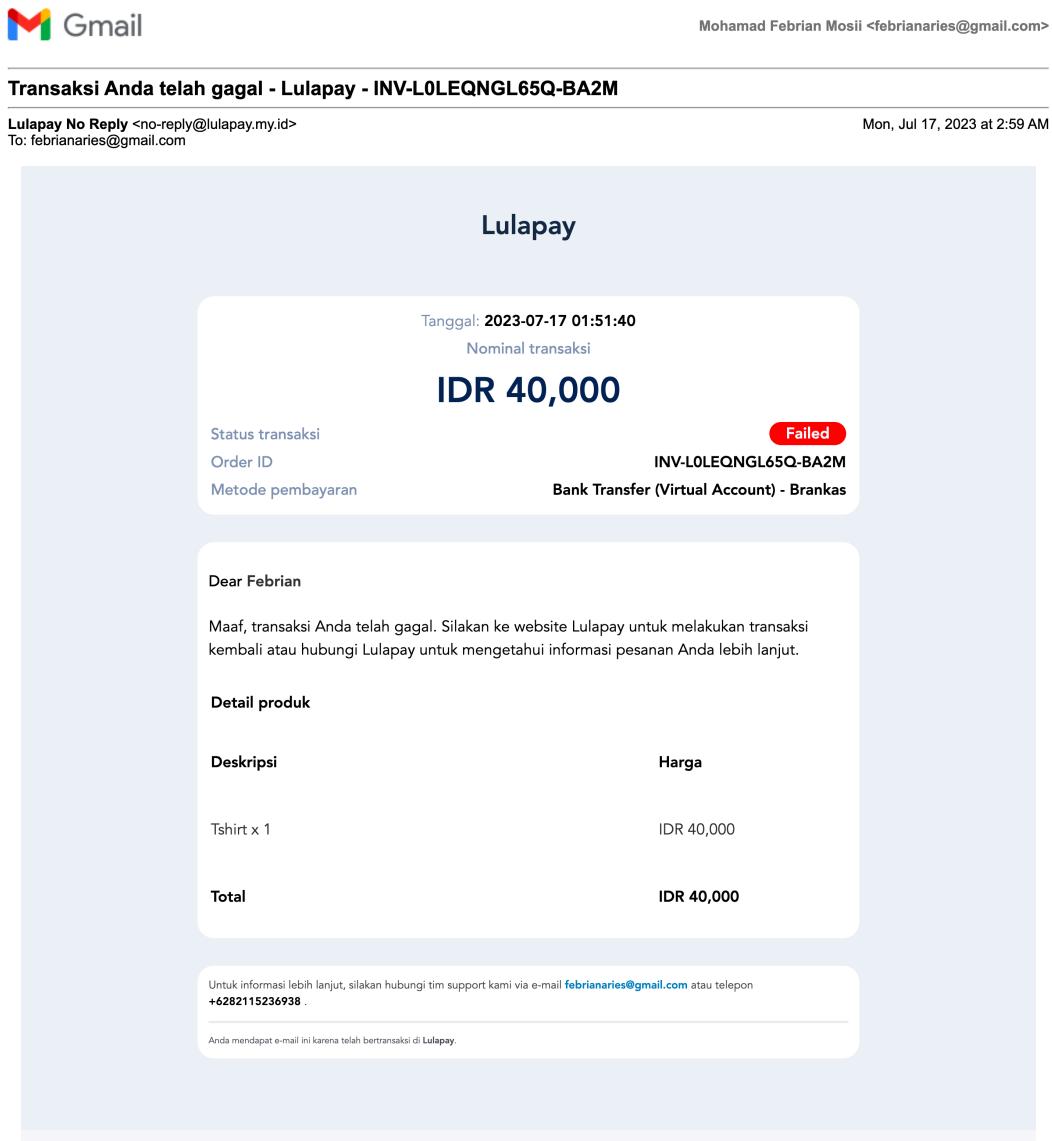
Gambar 5.52 Email Notifikasi Paid

5.5.8.3 Transaksi Kadaluarsa



Gambar 5.53 Email Notifikasi Expired

5.5.8.4 Transaksi Gagal



Gambar 5.54 Email Notifikasi Failed

5.6 Evaluasi Kepuasan Pengguna

Setelah melakukan uji coba, selanjutnya ada kuesioner untuk mengetahui tingkat kepuasan pengguna yang telah menggunakan aplikasi *payment gateway aggregator*. Hasil kuesioner didapatkan menggunakan Google Form yang beralamat:

- <https://forms.gle/avcXzrg3Cjz177at8>, untuk pihak pengelola aplikasi (CV Himalaya Teknologi)
- <https://forms.gle/smj95WeXD9JdPAmN7>, untuk calon pelanggan yang melakukan pembayaran, disebarluaskan melalui media sosial.

5.6.1 Kuesioner Pengelola

Berikut dibawah ini diuraikan hasil dari kuesioner yang telah terisi oleh 4 responden *user* pengelola.

Tabel 5.1 Kuesioner untuk pengelola merchant

No	Kuesioner	SS	S	CS	KS	TS
1	Aplikasi mudah digunakan	4	0	0	0	0
2	Integrasi aplikasi mudah	2	2	0	0	0
3	Proses mengelola transaksi mudah	3	1	0	0	0
4	Dapat meningkatkan efisiensi operasional perusahaan	4	0	0	0	0
5	Sudah memenuhi kebutuhan perusahaan	3	0	1	0	0

Keterangan : Sangat Setuju (SS), Setuju (S), Cukup Setuju (CS), Tidak Setuju (TS)

5.6.1.1 Kuesioner Calon Pelanggan

Berikut dibawah ini diuraikan hasil dari kuesioner yang telah terisi oleh 14 responden calon pelanggan yang telah menggunakan Aplikasi *Payment Gateway Aggregator*.

Tabel 5.2 Kuesioner untuk calon pelanggan

No	Kuesioner	SS	S	CS	KS	TS
1	Aplikasi mudah digunakan	9	5	0	0	0
2	Tampilan aplikasi menarik	12	1	0	1	0
3	Proses transaksi mudah	10	2	1	1	0
4	Opsi pembayaran yang biasa saya gunakan tersedia	9	3	0	2	0
5	Aplikasi layak digunakan sebagai media pembayaran <i>online</i>	9	3	2	0	0

Keterangan : Sangat Setuju (SS), Setuju (S), Cukup Setuju (CS), Tidak Setuju (TS)

5.6.2 Hasil Perhitungan Kuesioner

Tabel 5.3 Hasil Perhitungan Kuesioner

No	Indeks (%)	Kesimpulan
1	0 – 19.99	Tidak Setuju
2	20 – 39.99	Kurang Setuju
3	40 – 59.99	Cukup Setuju
4	60 – 79.99	Setuju
5	80 – 100	Sangat Setuju

Setelah menguji kuesioner dengan beberapa pertanyaan kepada responden yang merupakan pengguna aplikasi. Penulis dapat menyimpulkan dari seluruh pertanyaan bahwa untuk pengelola aplikasi memiliki indeks sebesar 96% dan untuk calon pelanggan memiliki indeks sebesar 91.42%. “Sangat Setuju” adalah pilihan yang diminati dan disarankan untuk diimplementasikan pada produk CV Himalaya Teknologi, yaitu Gasmaj. *Detail* perhitungan skala likert dapat dilihat pada LAMPIRAN 1.

BAB VI

KESIMPULAN DAN SARAN

Dengan merujuk pada penjelasan dan rumusan masalah yang telah diuraikan pada bab sebelumnya, berikut adalah simpulan yang dapat ditarik oleh penulis dalam poin-poin berikut:

6.1 Kesimpulan

Dengan terintegrasinya Gasmaj dengan Aplikasi *Payment Gateway Aggregator*:

1. Dapat meningkatkan efisiensi operasional dalam pengelolaan pembayaran di aplikasi Gasmaj CV Himalaya Teknologi, karena tanpa harus menggunakan sumber daya manusia tambahan, verifikasi pada setiap pembayaran sudah bisa dilakukan secara otomatis.
2. CV Himalaya Teknologi dapat memperluas opsi metode pembayaran, karena melalui penggabungan tiga *payment gateway*, yakni Midtrans, Brankas, dan Stripe, dapat menghasilkan opsi metode pembayaran yang lebih banyak, sehingga dapat memenuhi preferensi pelanggan.
3. CV Himalaya Teknologi tidak perlu mengelola persyaratan teknis yang berbeda dari setiap *Payment Gateway* secara terpisah jika ingin memiliki banyak metode pembayaran. Karena dengan satu kali integrasi melalui Aplikasi *Payment Gateway Aggregator*, sudah bisa terhubung ke tiga *payment gateway* sekaligus.

6.2 Saran

Adapun saran yang diusulkan oleh penulis sebagai berikut:

1. Saat ini belum ada fitur perbandingan antara metode pembayaran yang sama antara *payment gateway*. *Payment gateway* umumnya memiliki standar kebijakan yang berbeda terkait biaya layanan, dengan adanya fitur ini, pemilik usaha dapat lebih mudah melakukan perbandingan harga dan menentukan keputusan.
2. Aplikasi *payment gateway aggregator* dikembangkan menjadi SaaS (*Software as a service*). Dengan kelebihan yang dimiliki dan proses integrasi yang mudah, dapat memungkinkan bukan hanya merchant Gasmaj yang dapat terintegrasi, melainkan segala jenis *merchant* yang ingin menyediakan fitur pembayaran *online* terintegrasi.

DAFTAR PUSTAKA

- [1] M. I. Fajrin, I. A. Kautsar, and S. Aji, “Design and Build *Payment Gateways* in Sharia-Based E-Commerce (Case Study: Murabahah Ijabqabul.Id Contract),” *Procedia of Engineering and Life Science*, vol. 2, no. 2, Sep. 2022, doi: 10.21070/pels.v2i2.1262.
- [2] T. M. M. Puspitasari and D. Maulina, “Implementasi *Payment Gateway* Menggunakan Midtrans Pada Marketplace Travnesia.Com,” *Mobile and Forensics*, vol. 1, no. 1, p. 22, Sep. 2019, doi: 10.12928/mf.v1i1.997.
- [3] G. Tang, L. Wu, and L. Guo, “The impact of e-commerce platform merchants’ reputation on consumer decision making,” *E3S Web of Conferences*, vol. 292, p. 02023, Jan. 2021, doi: 10.1051/e3sconf/202129202023.
- [4] J. Sausi, J. Mtebe, and J. Mbelwa, “Evaluating user satisfaction with the e-*payment gateway* system in Tanzania,” *SA Journal of Information Management*, vol. 23, Dec. 2021, doi: 10.4102/sajim.v23i1.1430.
- [5] M. Hassan, Z. Shukur, and M. Hasan, “An Efficient Secure Electronic Payment System for E-Commerce,” *Computers*, vol. 9, p. 13, Aug. 2020, doi: 10.3390/computers9030066.
- [6] S. Mukhopadhyay, S. Pingali, and A. Satyam, “Razorpay: Providing Payment Convenience to Disruptors,” 2021, pp. 237–260. doi: 10.4324/9781003155270-12.
- [7] D. Sugiarti and R. Iskandar, “Pengaruh Consumer Review Terhadap Keputusan Pembeli Terhadap Toko Online Shopee,” *Jurnal Sosial*

- Teknologi*, vol. 1, pp. 954–962, Sep. 2021, doi: 10.36418/jurnalsostech.v1i9.195.
- [8] A. Ghoni and T. Bodroastuti, “Pengaruh Faktor Budaya, Sosial, Pribadi Dan Psikologi Terhadap Perilaku Konsumen (Studi Pada Pembelian Rumah Di Perumahan Griya Utama Banjardowo Semarang),” *Jurnal Kajian Akuntansi dan Bisnis*, vol. 1, no. 1, 2012.
- [9] A. Frihatni, “Analisis Perbandingan Pendapatan Pedagang Pakaian Tradisional dengan Pendapatan Bisnis Online,” *Jurnal Ilmiah Akuntansi Manajemen*, vol. 5, pp. 1–8, Apr. 2022, doi: 10.35326/jiam.v5i1.1980.
- [10] Z. Zulkifli and M. Solot, “Pengaruh Customer Offline dan Customer Online Terhadap Penjualan Pada Toko 3 Second Plaza Mulia Samarinda”,” *Jurnal Ekonomika : Manajemen, Akuntansi, dan Perbankan Syari’ah*, vol. 7, p. 43, Feb. 2019, doi: 10.24903/je.v7i1.445.
- [11] Д. Бабич, В. Иванова, and Л. Сайбонова, *THE ESSENCE OF B2B-SALES IN IT*. 2023. doi: 10.37539/230123.2023.35.38.007.
- [12] M. Yudono *et al.*, “Perbandingan Brick and Mortar dan Click and Mortar,” Jan. 2023.
- [13] B. Bangun, “STUDI SOSIO-LEGAL TERHADAP PENGATURAN DAN POLA PERDAGANGAN LINTAS BATAS NEGARA DI POS LINTAS BATAS NEGARA (PLBN) ENTIKONG,” *Jurnal Komunikasi Hukum (JKH)*, vol. 8, pp. 139–160, Jan. 2022, doi: 10.23887/jkh.v8i1.43606.
- [14] R. Rachman and R. Oktavianti, “Pengaruh Kepercayaan Konsumen terhadap Loyalitas Pelanggan dalam Penggunaan Sistem Pembayaran Online (Survei

- Pengguna Produk Unipin)," *Prologia*, vol. 5, no. 1, pp. 148–153, Mar. 2021, doi: 10.24912/pr.v5i1.8200.
- [15] D. A. Lestari, E. D. Purnamasari, and B. Setiawan, "Pengaruh *Payment Gateway* terhadap Kinerja Keuangan UMKM," *Jurnal Bisnis, Manajemen, dan Ekonomi*, vol. 1, no. 1, pp. 1–10, Sep. 2020, doi: 10.47747/jbme.v1i1.20.
- [16] D. Wahyuningsih, P. Romadiana, and L. Tommy, "Integrasi Sistem Organisasi Mahasiswa Atma Luhur Dengan Pendekatan Service Oriented Architecture," 2020.
- [17] T. M. M. Puspitasari and D. Maulina, "Implementasi *Payment Gateway* Menggunakan Midtrans Pada Marketplace Travnesia.Com," *Mobile and Forensics*, vol. 1, no. 1, p. 22, Sep. 2019, doi: 10.12928/mf.v1i1.997.
- [18] D. Account, "This is test MS 6 Feb ENG-1606 Testing Stripe Payment for Prepaid Credits," *JMIR Form Res*, Feb. 2023, doi: 10.2196/45971.
- [19] A. Nabila, F. Oktavianti, and N. Putri, "Using Paypal As E-Payment In The International Payment System," *ASIAN Economic and Business Development*, vol. 4, pp. 14–19, Jun. 2022, doi: 10.54204/AEBD/Vol4No1July2022003.
- [20] S. Markovich, N. Achwal, and E. Queathem, "Stripe: Helping Money Move on the Internet," *Kellogg School of Management Cases*, pp. 1–12, Oct. 2017, doi: 10.1108/case.kellogg.2021.000073.

LAMPIRAN 1:
PERHITUNGAN EVALUASI KEPUASAN PELANGGAN

No	Pilihan Jawab	Bobot Nilai
1	Sangat Setuju	5
2	Setuju	4
3	Cukup Setuju	3
4	Kurang Setuju	2
5	Tidak Setuju	1

A. Kuesioner Pengelola

Kriteria interpretasi skor berdasarkan interval:

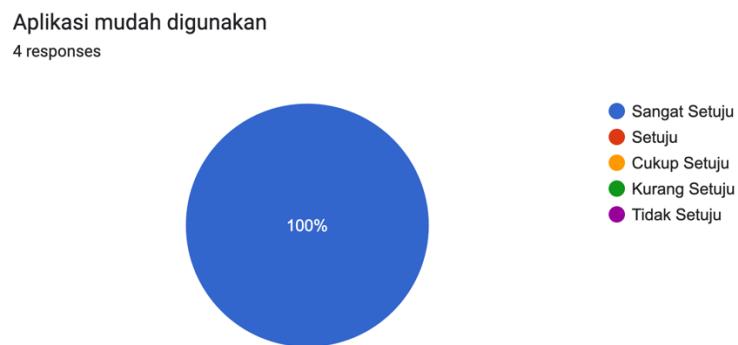
- Skor tertinggi: $5 \times 4 = 20$. Dimana $20/20 \times 100\% = 100\%$ (Sangat Setuju).
- Skor terendah: $1 \times 4 = 4$. Dimana $4/20 \times 100\% = 20\%$ (Tidak Setuju).

Sehingga dihasilkan bahwa:

- Angka 0% - 20% = Tidak Setuju
- Angka 21% - 40% = Kurang Setuju
- Angka 41% - 60% = Cukup Setuju
- Angka 61% - 80% = Setuju
- Angka 81% - 100% = Sangat setuju

Berikut pertanyaan yang diajukan penulis kepada 4 responden untuk selaku pihak pengelola:

1. Aplikasi mudah digunakan



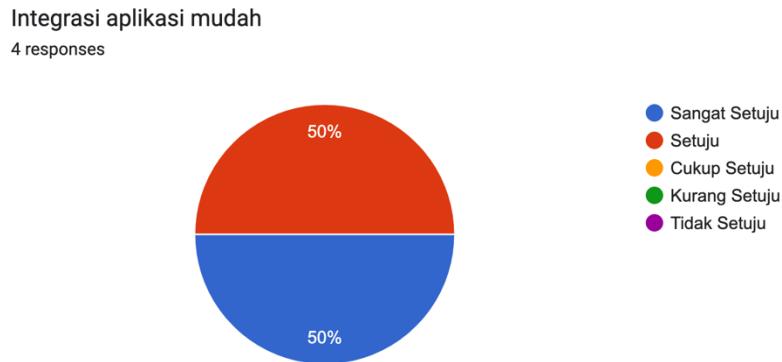
Berdasarkan hasil kuesioner dari 4 responden menjawab, 4 sangat setuju (SS), 0 responden menjawab setuju (S), 0 responden menjawab cukup setuju (CS), 0 menjawab kurang setuju (KS), dan 0 menjawab tidak setuju (TS).

Hasil pengukuran skala likert, maka:

- $SS = 5 \times 4 = 20$
- $S = 4 \times 0 = 0$
- $CS = 3 \times 0 = 0$
- $KS = 2 \times 0 = 0$
- $TS = 1 \times 0 = 0$

Jumlah skor ideal untuk kuesioner nomor satu adalah 20, maka $20/20 \times 100\% = 100\%$ termasuk kategori sangat setuju.

2. Integrasi aplikasi mudah



Berdasarkan hasil kuesioner dari 4 responden menjawab, 2 sangat setuju (SS), 2 responden menjawab setuju (S), 0 responden menjawab cukup setuju (CS), 0 menjawab kurang setuju (KS), dan 0 menjawab tidak setuju (TS).

Hasil pengukuran skala likert, maka:

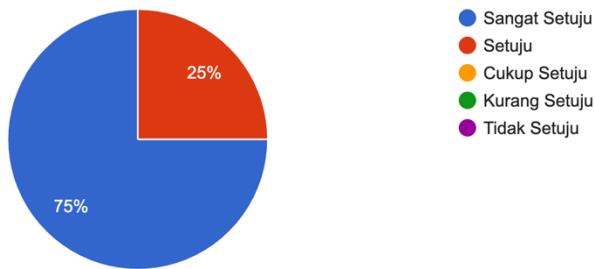
- $SS = 5 \times 2 = 10$
- $S = 4 \times 2 = 8$
- $CS = 3 \times 0 = 0$
- $KS = 2 \times 0 = 0$
- $TS = 1 \times 0 = 0$

Jumlah skor ideal untuk kuesioner nomor dua adalah 18, maka $18/20 \times 100\% = 90\%$ termasuk kategori sangat setuju.

3. Proses mengelola transaksi mudah

Proses mengelola transaksi mudah

4 responses



Berdasarkan hasil kuesioner dari 4 responden menjawab, 3 sangat setuju (SS), 1 responden menjawab setuju (S), 0 responden menjawab cukup setuju (CS), 0 menjawab kurang setuju (KS), dan 0 menjawab tidak setuju (TS).

Hasil pengukuran skala likert, maka:

- SS = $5 \times 3 = 15$
- S = $4 \times 1 = 4$
- CS = $3 \times 0 = 0$
- KS = $2 \times 0 = 0$
- TS = $1 \times 0 = 0$

Jumlah skor ideal untuk kuesioner nomor tiga adalah 19, maka $19/20 \times 100\% = 95\%$ termasuk kategori sangat setuju.

4. Dapat meningkatkan efisiensi operasional perusahaan

Dapat meningkatkan efisiensi operasional perusahaan
4 responses



Berdasarkan hasil kuesioner dari 4 responden menjawab, 4 sangat setuju (SS), 0 responden menjawab setuju (S), 0 responden menjawab cukup setuju (CS), 0 menjawab kurang setuju (KS), dan 0 menjawab tidak setuju (TS).

Hasil pengukuran skala likert, maka:

- $SS = 5 \times 4 = 20$
- $S = 4 \times 0 = 0$
- $CS = 3 \times 0 = 0$
- $KS = 2 \times 0 = 0$
- $TS = 1 \times 0 = 0$

Jumlah skor ideal untuk kuesioner nomor empat adalah 20, maka $20/20 \times 100\% = 100\%$ termasuk kategori sangat setuju.

5. Sudah memenuhi kebutuhan perusahaan



Berdasarkan hasil kuesioner dari 4 responden menjawab, 3 sangat setuju (SS), 1 responden menjawab setuju (S), 0 responden menjawab cukup setuju (CS), 0 menjawab kurang setuju (KS), dan 0 menjawab tidak setuju (TS).

Hasil pengukuran skala likert, maka:

- SS = $5 \times 3 = 15$
- S = $4 \times 1 = 4$
- CS = $3 \times 0 = 0$
- KS = $2 \times 0 = 0$
- TS = $1 \times 0 = 0$

Jumlah skor ideal untuk kuesioner nomor lima adalah 19, maka $19/20 \times 100\% = 95\%$ termasuk kategori sangat setuju.

B. Kuesioner Calon Pelanggan

No	Pilihan Jawab	Bobot Nilai
1	Sangat Setuju	5
2	Setuju	4
3	Cukup Setuju	3
4	Kurang Setuju	2
5	Tidak Setuju	1

Kriteria interpretasi skor berdasarkan interval:

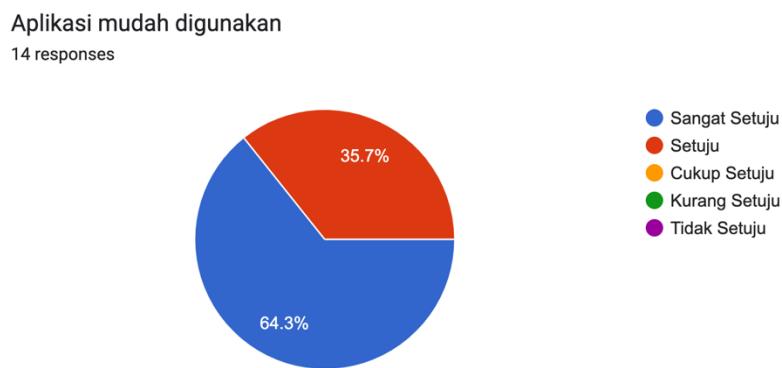
- Skor tertinggi: $5 \times 14 = 70$. Dimana $70/70 \times 100\% = 100\%$ (Sangat Setuju).
- Skor terendah: $1 \times 14 = 14$. Dimana $14/70 \times 100\% = 20\%$ (Tidak Setuju).

Sehingga dihasilkan bahwa:

- Angka 0% - 20% = Tidak Setuju
- Angka 21% - 40% = Kurang Setuju
- Angka 41% - 60% = Cukup Setuju
- Angka 61% - 80% = Setuju
- Angka 81% - 100% = Sangat setuju

Berikut pertanyaan yang diajukan penulis kepada 14 responden untuk selaku calon pelanggan:

1. Aplikasi mudah digunakan



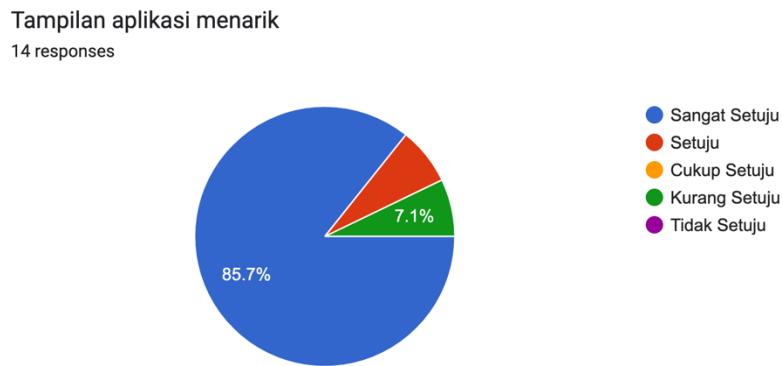
Berdasarkan hasil kuesioner dari 14 responden menjawab, 9 sangat setuju (SS), 5 responden menjawab setuju (S), 0 responden menjawab cukup setuju (CS), 0 menjawab kurang setuju (KS), dan 0 menjawab tidak setuju (TS).

Hasil pengukuran skala likert, maka:

- $SS = 5 \times 9 = 45$
- $S = 4 \times 5 = 20$
- $CS = 3 \times 0 = 0$
- $KS = 2 \times 0 = 0$
- $TS = 1 \times 0 = 0$

Jumlah skor ideal untuk kuesioner nomor satu adalah 65, maka $65/70 \times 100\% = 92,85\%$ termasuk kategori sangat setuju.

2. Integrasi aplikasi mudah



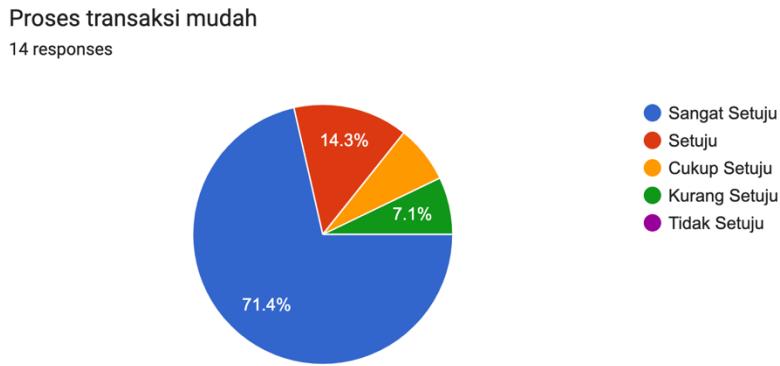
Berdasarkan hasil kuesioner dari 14 responden menjawab, 12 sangat setuju (SS), 1 responden menjawab setuju (S), 0 responden menjawab cukup setuju (CS), 1 menjawab kurang setuju (KS), dan 0 menjawab tidak setuju (TS).

Hasil pengukuran skala likert, maka:

- $SS = 5 \times 12 = 60$
- $S = 4 \times 1 = 4$
- $CS = 3 \times 0 = 0$
- $KS = 2 \times 1 = 2$
- $TS = 1 \times 0 = 0$

Jumlah skor ideal untuk kuesioner nomor dua adalah 66, maka $66/70 \times 100\% = 94,28\%$ termasuk kategori sangat setuju.

3. Proses mengelola transaksi mudah



Berdasarkan hasil kuesioner dari 14 responden menjawab, 10 sangat setuju (SS), 2 responden menjawab setuju (S), 1 responden menjawab cukup setuju (CS), 1 menjawab kurang setuju (KS), dan 0 menjawab tidak setuju (TS).

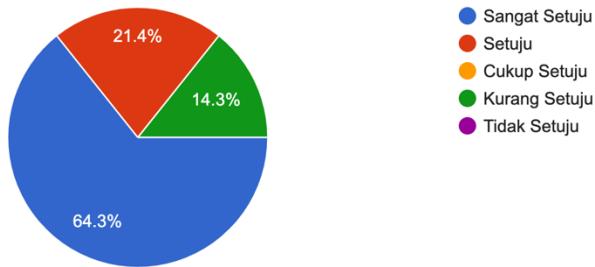
Hasil pengukuran skala likert, maka:

- SS = $5 \times 10 = 50$
- S = $4 \times 2 = 8$
- CS = $3 \times 1 = 3$
- KS = $2 \times 1 = 2$
- TS = $1 \times 0 = 0$

Jumlah skor ideal untuk kuesioner nomor tiga adalah 63, maka $63/70 \times 100\% = 90\%$ termasuk kategori sangat setuju.

4. Dapat meningkatkan efisiensi operasional perusahaan

Opsi pembayaran yang biasa saya gunakan tersedia
14 responses



Berdasarkan hasil kuesioner dari 14 responden menjawab, 9 sangat setuju (SS), 3 responden menjawab setuju (S), 2 responden menjawab cukup setuju (CS), 0 menjawab kurang setuju (KS), dan 0 menjawab tidak setuju (TS).

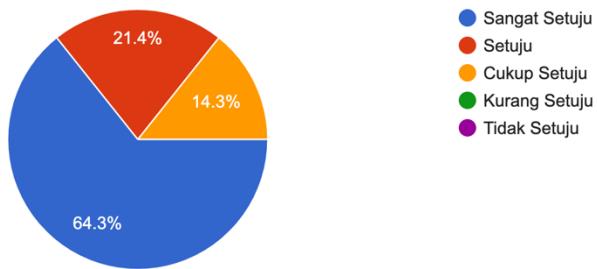
Hasil pengukuran skala likert, maka:

- $SS = 5 \times 9 = 45$
- $S = 4 \times 3 = 12$
- $CS = 3 \times 2 = 6$
- $KS = 2 \times 0 = 0$
- $TS = 1 \times 0 = 0$

Jumlah skor ideal untuk kuesioner nomor empat adalah 63, maka $63/70 \times 100\% = 90\%$ termasuk kategori sangat setuju.

5. Sudah memenuhi kebutuhan perusahaan

Aplikasi layak digunakan sebagai media pembayaran online
14 responses



Berdasarkan hasil kuesioner dari 14 responden menjawab, 9 sangat setuju (SS), 3 responden menjawab setuju (S), 2 responden menjawab cukup setuju (CS), 0 menjawab kurang setuju (KS), dan 0 menjawab tidak setuju (TS).

Hasil pengukuran skala likert, maka:

- SS = 5 x 9 = 45
- S = 4 x 3 = 12
- CS = 3 x 2 = 6
- KS = 2 x 0 = 0
- TS = 1 x 0 = 0

Jumlah skor ideal untuk kuesioner nomor lima adalah 63, maka $63/70 \times 100\% = 90\%$ termasuk kategori sangat setuju.