

PROSES PEMBAYARAN PADA APLIKASI

LAPORAN TUGAS BESAR

Mata Kuliah Metode Formal



SE 46-03

1302220009	Marjauza Naswansyah
1302220037	Nizar Rasyiid
1302220031	Hasan Nurrahman Pane
1302223116	Matthew Theodore Lempas

PRODI S-1 REKAYASA PERANGKAT LUNAK

FAKULTAS INFORMATIKA

UNIVERSITAS TELKOM

2023

PENDAHULUAN

Latar Belakang

Proses pembayaran dalam aplikasi adalah serangkaian langkah elektronik yang memungkinkan pengguna untuk melakukan transaksi keuangan. Meskipun proses ini dirancang untuk memudahkan pengguna, beberapa kesalahan dapat terjadi. Salah satu kesalahan umum adalah kesalahan input informasi pembayaran, seperti memasukkan nomor kartu kredit yang salah. Ini dapat menghambat proses dan mengakibatkan transaksi ditolak.

Kesalahan lainnya berkaitan dengan keamanan. Penipuan elektronik dapat terjadi jika informasi pembayaran pengguna dicuri atau disalahgunakan. Oleh karena itu, aplikasi harus menyediakan lapisan keamanan yang kuat, termasuk enkripsi data dan penerapan tindakan pencegahan fraud. Selain itu, kurangnya transparansi terkait kebijakan pengembalian dan pembatalan juga dapat menjadi sumber kebingungan dan ketidakpuasan pengguna. Penting bagi pengguna untuk memahami syarat dan ketentuan sebelum menyelesaikan pembayaran. Proses pembayaran yang efisien memerlukan integrasi yang baik dengan penyedia layanan pembayaran dan gateway pembayaran. Gangguan pada sistem atau keterlambatan dalam pemrosesan transaksi juga dapat mengakibatkan ketidaknyamanan bagi pengguna.

GOALS

Rumusan Masalah

Bagaimana cara mengetahui proses pada fitur pembayaran di aplikasi menggunakan payment gateway?

Product Goals

Tujuan dalam dokumen ini adalah untuk mengetahui proses dari sebuah fitur pembayaran dalam aplikasi

Features

Fitur utama pada aplikasi ini adalah sebagai berikut:

- a. Aplikasi memiliki beragam metode pembayaran yang aman dan terpercaya.

End Goals

Manfaat yang bisa terwujud bagi berbagai pihak, diantaranya sebagai berikut:

- a. **Pengguna**, menciptakan rasa kepercayaan dari konsumen dengan mengedepankan keamanan dan privasi dalam pembayaran

Goals

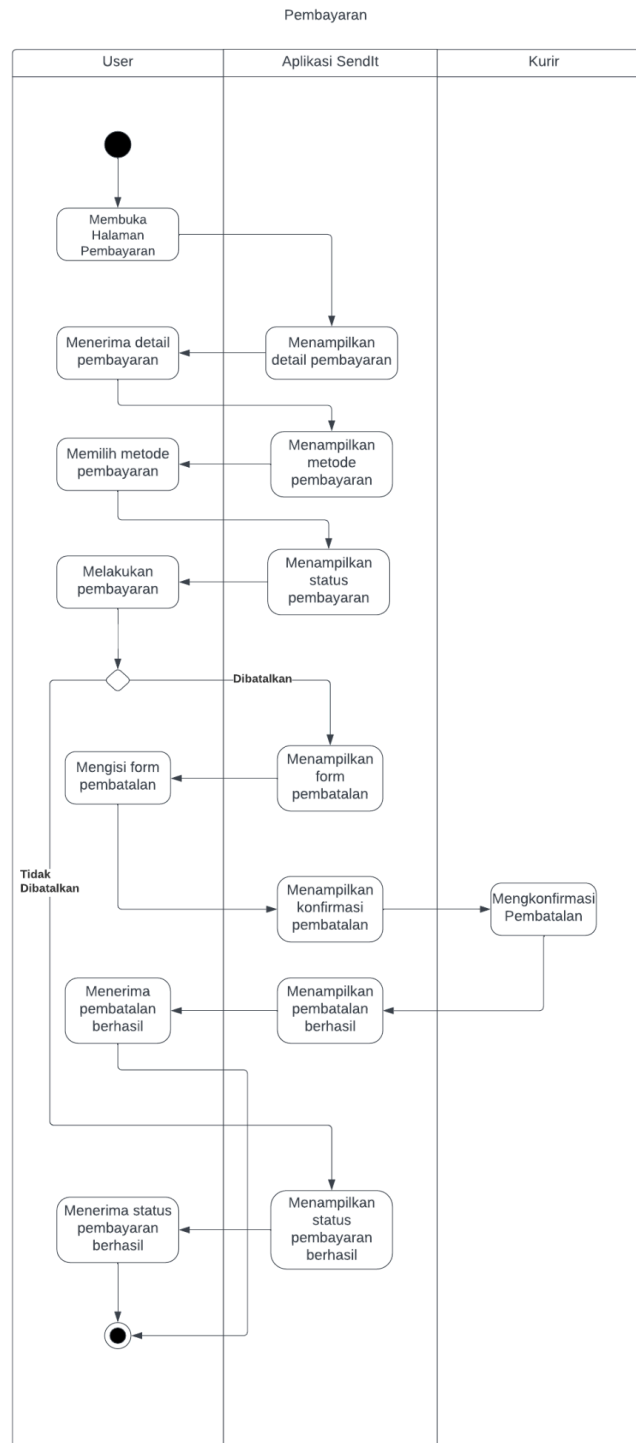
Manfaat yang bisa terwujud, diantaranya sebagai berikut:

- a. **Efisiensi Waktu dan Biaya**, dengan adanya fitur pembayaran yang efisien dan tidak memakan biaya, maka pengguna akan merasakan fitur ini dengan nyaman.
- b. **Keberagaman Opsi**, dalam proses pembayaran pada aplikasi ini, pengguna dapat memilih opsi metode pembayaran.
- c. **Keamanan**, menciptakan sebuah proses pembayaran yang aman dan tidak melanggar privasi pengguna.

Modelling

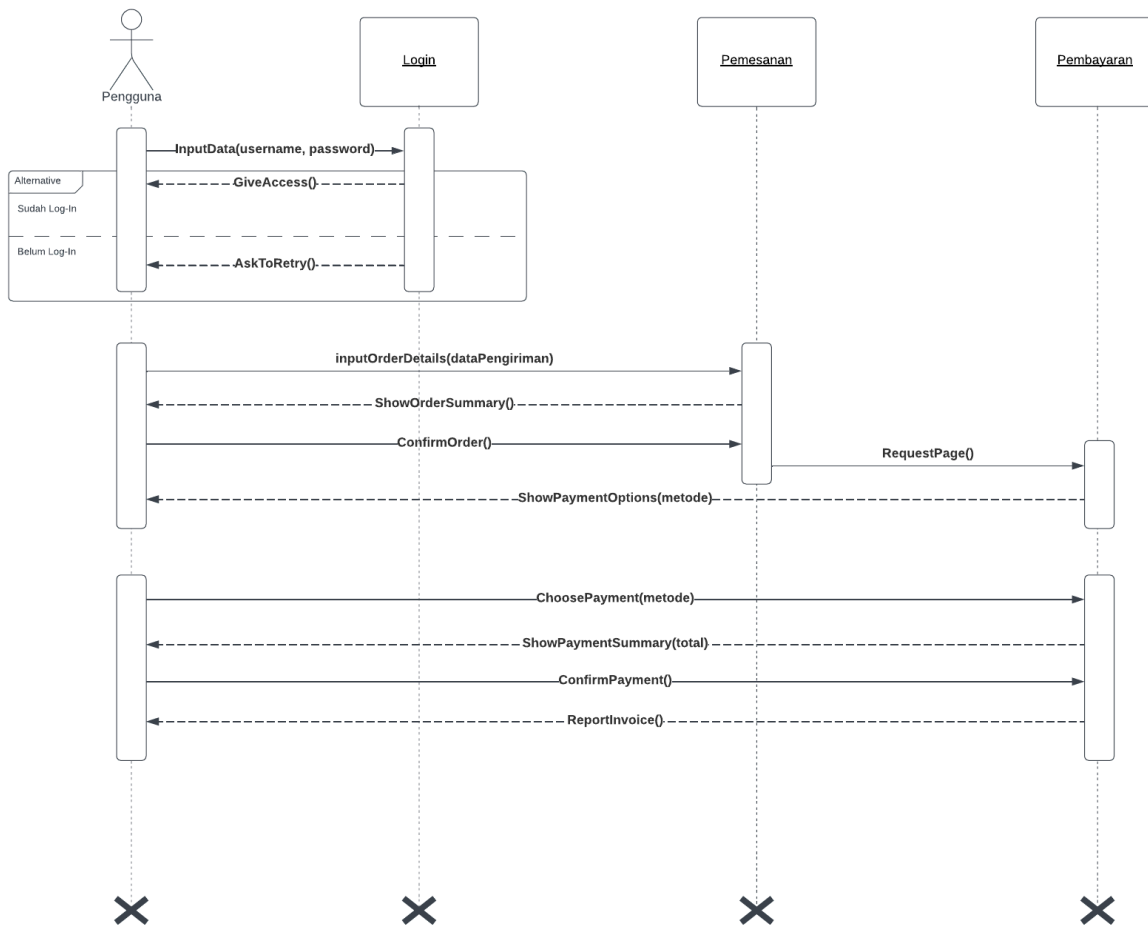
Activity Diagram:

Kami memilih untuk menjabarkan *activity diagram* pada *use case* pembayaran, berikut penggambarannya:



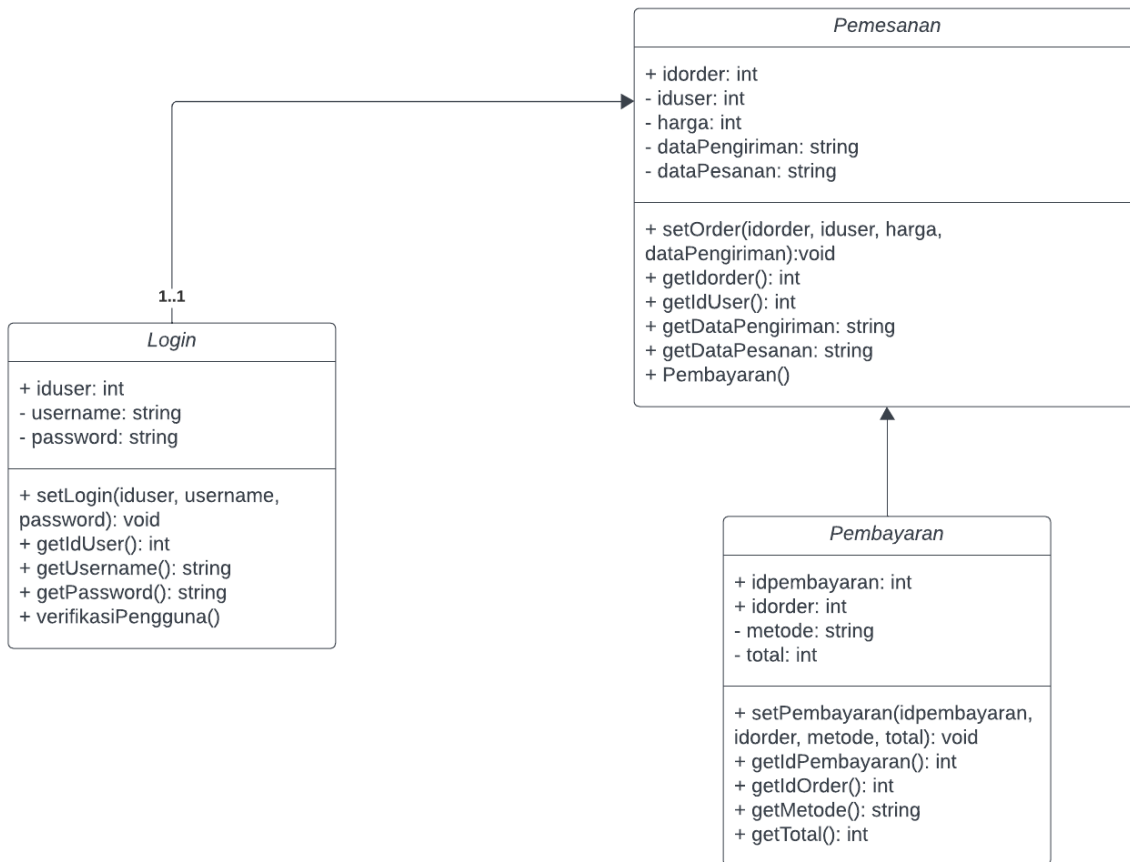
Sequence Diagram

Penggambaran *sequence diagram* dalam penggambaran *use case* pembayaran:



Class Diagram

Kami mencoba untuk melakukan penggambaran *class diagram* dari aplikasi ini, berikut penggambaran *class diagram*:



Alloy

Berikut merupakan implementasi alloy berdasarkan class diagram:

```
module Pemesanan

sig Pemesanan {
  idorder: Int
  iduser: Int
  harga: Int
  dataPengiriman: Text
  dataPesanan: Text
}

sig Login {
  iduser: Int
  username: Text
  password: EncryptedText
}

sig Pembayaran {
  idpembayaran: Int
  idorder: Int
  metode: Text
  total: Int
}

// Asosiasi
Pemesanan o-- 1..1 Login
Pemesanan o-- 1..1 Pembayaran
```

```
// Predikat

pred main() {
  // Setiap pemesanan harus memiliki id order, id user, harga, data pengiriman, dan data pesanan
  all p: Pemesanan | p.idorder > 0
  all p: Pemesanan | p.iduser > 0
  all p: Pemesanan | p.harga > 0
  all p: Pemesanan | p.dataPengiriman != ""
  all p: Pemesanan | p.dataPesanan != ""

  // Setiap login harus memiliki id user
  all l: Login | l.iduser > 0

  // Setiap pembayaran harus memiliki id order, metode, dan total
  all b: Pembayaran | b.idorder > 0
  all b: Pembayaran | b.metode != ""
  all b: Pembayaran | b.total > 0

  // Setiap pemesanan hanya boleh memiliki satu login
  no p1, p2: Pemesanan | p1 != p2 && p1.idorder = p2.idorder && p1.iduser != p2.iduser

  // Setiap pemesanan hanya boleh memiliki satu pembayaran
  no p1, p2: Pemesanan | p1 != p2 && p1.idorder = p2.idorder && p1.idpembayaran != p2.idpembayaran

  // Setiap login hanya boleh digunakan untuk satu pemesanan
  no l1, l2: Login | l1 != l2 && l1.iduser = l2.iduser

  // Setiap pembayaran hanya boleh digunakan untuk satu pemesanan
  no b1, b2: Pembayaran | b1 != b2 && b1.idpembayaran = b2.idpembayaran
}
```