**BAB IV**

**PERANCANGAN SISTEM**

**4.1. Perancangan Sistem**

Perancangan sistem adalah Menyusun atau mendesain suatu sistem yang berisi Langkah-langkah operasi dalam proses pengolahan data dan proses prosedur-prosedur untuk mendukung operasi sistem. Tujuan dari perancangan sistem adalah memetakan semua kebutuhan sistem sehingga sistem berjalan dengan baik, memenuhi kebutuhan para pemakai sistem seta memberikan gambaran yang jelas dan rancang bangun yang lengkap kepada para programmer dan para ahli yang terlibat dalam pengembangan sistem.

Perancangan sistem ini mencakup *use case diagram*, *activity diagram, sequence diagram,* dan *class diagram* yang depat menjelaskan aliran data yang diproses hingga menghasilkan informasi yang diinginkan.

**4.1.1. Identifikasi Aktor**

Pada sistem Informasi pemesanan Persada Konveksi ini terdapat 3 pengguna atau actor. Adapun penjeasan dari pengguna atau actor adalah sebagai berikut:

**Tabel 4.1** Indentifikasi Aktor

|  |  |  |
| --- | --- | --- |
| **No.** | **Aktor** | **Deskripsi** |
| 1 | Administrator | Pengguna yang mempunyai hak akses untuk mengelola *website* secara keseluruhan. Pengguna ini dapat mengakses setiap data master, seperti data produk, data pelanggan, data pemesanan, data transaksi dan data penjualan. |
| 3 | Pelanggan | Pelanggan dapat melihat produk, dan melakukan pemesanan, |

**4.1.2. Identifikasi Use Case**

Pada Sistem Informasi Pemesanan Konveksi Persad aini terdapat beberapa *use case* untuk mengetahui fungsi dalam sistem yang akan dibangun. Adapun penjelasan dari *use case* tersebut adalah sebagai berikut:

**Table 4.2.** Identifikasi *Use Case*

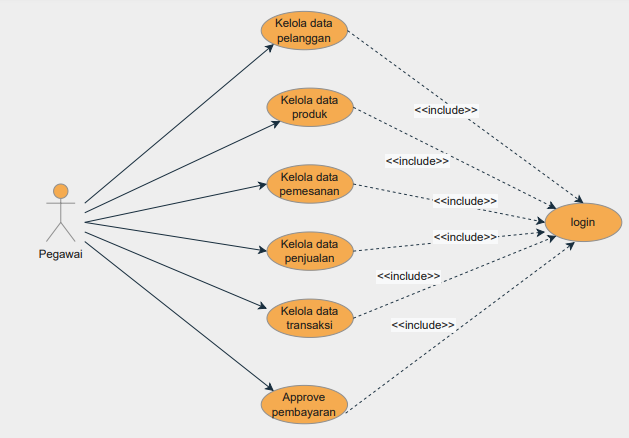
|  |  |  |
| --- | --- | --- |
| **No.** | **Aktor** | **Deskripsi** |
| 1 | Registrasi | [Pegawai, Pelanggan] Aktivitas yang diperlukan untuk membuat akun agar dapat mengakses aplikasi. |
| 2 | Login | [Pegawai, Pelanggan] Aktivitas untuk mengatur hak akses pengguna dan sebagai validasi untuk mendapatkan hak akses sistem. |
| 3 | Kelola data pelanggan | [Pegawai] Aktivitas untuk melihat data pelanggan. |
| 4 | Kelola data produk | [Pegawai] Aktivitas untuk mengelola data produk untuk aksi tambah dan ubah. |
| 5 | Kelola data pemesanan | [Pegawai] Aktivitas untuk mengelola data pemesanan yaitu ubah data pesanan. |
| 6. | Bayar pesanan | [Pelanggan] Aktivitas untuk memberikan informasi bahwa pelanggan telah melakukan pembayaran. |
| 7 | Lihat detail pesanan | [Pelanggan, pegawai] Aktivitas untuk melihat detail pesanan. |
| 8 | Lihat data produk | [Pelanggan] Aktivitas untuk melihat produk. |
| 9 | Lihat detail Produk | [Pelanggan, pegawai] Aktivitas untuk melihat detail produk. |
| 10 | Kelola data transaksi | [Pegawai] Aktivitas untuk mengeola transaksi pelanggan dengan cara menyetujui/menolak pembayaran yang diakukan oleh pelanggan. |
| 11 | Melihat laporan penjualan | [Pegawai] Aktivitas untuk memeriksa penjualan konveksi |

**4.2. Metode Perancangan Sistem**

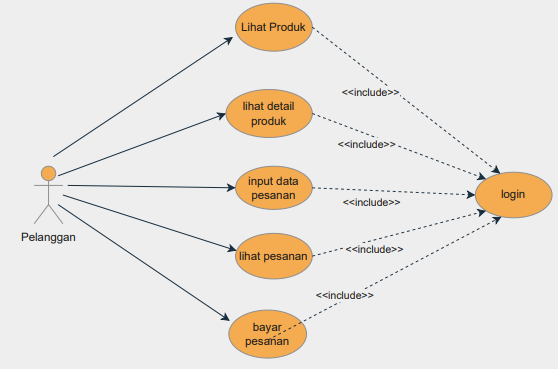
Untuk membantu proses perancangan sistem berorientasi objek, maka digunakan 4(empat) diagram yaitu *use case diagram*, *activity diagram, sequence diagram* dan *class diagram* sebagai berikut:

**4.2.1. *Use Case***

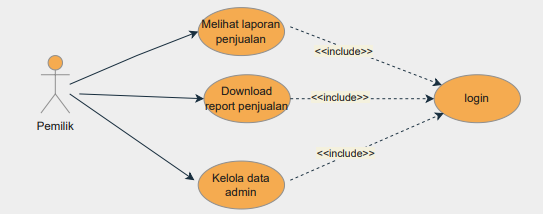
**4.2.1.1. Use Case Diagram**

****

**Gambar 4.1** Use Case Pegawai



**Gambar 4.2** Use Case Pelanggan

****

**Gambar 4.3** Use Case Pemilik

**4.2.1.2*. Use Case Spesification***

*1. Use Case Spesification Registrasi*

**Tabel 4.3** *Use Case Spesification* *Registrasi*

|  |  |  |
| --- | --- | --- |
| *Use Case Name* | Registrasi | |
| *Actor* | Pelanggan | |
| *Description* | *Use Case* ini digunakan oleh pelanggan untuk membuat akun pelanggan guna untuk login ke sisitem | |
| *Pre-condition* | Pengguna dalam keadaan logout dan sistem menampilkan *form login* | |
| *Basic Flow* | Aktor | Sistem |
| 1. *User* membuka *website* untuk pertama kali atau pengguna baru | 1. Menampilkan halaman login |
|  | 1. *User* klik link *registrasi* | 1. Menampilkan form inputan registrasi |
|  | 1. Mengisi form inputan data, kemudian menekan tombol *submit* | 1. Sistem melakukan validasi pada data yang diinputkan sebelum data dimasukan di dalam *database*. (*alternatif flow* 6.1) |
| Alternative Flow | 6.1. Data yang diinput tidak sesuai dengan type data yang diprogram dalam sistem(Kembali ke Langkah 4) | |
| *Requirement* | Berhasil login sebagai pegawai atau pelanggan | |
| *Post-condition* | Sistem melakukan *Data Manipulation Language* (DML) pada datebase sesuai dengan aksi yang dilakukan *actor*(tambah data pelanggan) | |

*2. Use Case Spesification Login*

**Tabel 4.4** *Use Case Spesification* *Login*

|  |  |  |
| --- | --- | --- |
| *Use Case Name* | Login | |
| *Actor* | Pegawai dan pelanggan | |
| *Description* | Dalam proses ini sistem melakukan proses authentication dengan menyesuaikan *username* dan *password* yang dimasukkan oleh user dengan data berada di dalam database. Dan sistem juga melakukan proses *authorization*/menentukan hak akses yang dimiliki oleh user. | |
| *Pre-condition* | Pengguna dalam keadaan logout dan sistem menampilkan *form login* | |
| *Basic Flow* | Aktor | Sistem |
| 1. *Input username* dan *password* | Sistem melakukan validasi *username* dan *password* yang dimasukkan user |
| Alternative Flow | Jika Username dan password salah maka sistem akan menampilkan pesan “Username dan Password salah!”  Jika Username atau password kosong maka sistem akan menampilkan pesan “Username dan Password tidak boleh kosong” | |
| *Requirement* | Berhasil login sebagai pegawai atau pelanggan | |
| *Post-condition* | Sistem menampilkan halaman utama | |

3. *Use Case Spesification* Kelola Data Produk

**Tabel 4.5** *Use Case Spesification* Kelola Data Produk

|  |  |  |  |
| --- | --- | --- | --- |
| *Use Case Name* | Kelola Data Produk | | |
| *Actor* | Pegawai | | |
| *Description* | *Use Case* ini digunakan untuk mengelola data produk yang dapat diproduksi oleh konveksi. Fungsi-fungsi yang ada di dalam *use case* ini yaitu tambah, ubah dan hapus data produk. | | |
| *Pre-condition* | *User* dalam keadaan login | | |
| *Basic Flow* | Aksi | Aktor | Sistem |
|  | Tambah | 1. Klik menu form produk | 1. Sistem menampilkan form tambah data produk |
|  |  | 1. Mengisi *form* tambah data produk dan klik *button submit* | 1. Sistem melakukan validasi pada data yang diinputkan sebelum data dimasukan di dalam *database*. |
|  | Ubah | 1. Klik menu produk | 1. Menampilkan *form* yang berisi detail data yang akan diubah. |
|  |  | 1. Menginputkan perubahan data kemudian klik *button save* | 1. Sistem melakukan validasi data yang telah diinputkan user dan menyimpan data ke dalam *database.*   (*Alternatif flow* 4.1) |
| *Alternative Flow* | 4.1 Data yang diinputkan tidak sesuai dengan peraturan sistem (Kembali ke langkah 2) | | |
| *Requirement* | Berhasil login sebagai pegawai | | |
| *Post-condition* | Sistem melakukan Data Manipulation Language (DML) pada database sesuai dengan aksi yang dilakukan aktor (tambah data ubah data) | | |

4. *Use Case Spesification* Kelola Data Pesanan

**Tabel 4.6** *Use Case Spesification* Kelola Data Pesanan

|  |  |  |  |
| --- | --- | --- | --- |
| *Use Case Name* | Kelola Data Pesanan | | |
| *Actor* | Pegawai dan Pelanggan | | |
| *Description* | *Use Case* ini digunakan untuk mengelola data pesanan yang diinputkan oleh pelanggan dan pesanan yang diterima oleh konveksi. Fungsi-fungsi yang ada di dalam *use case* ini yaitu tambah dan ubah data pesanan. | | |
| *Pre-condition* | *User* dalam keadaan login | | |
| *Basic Flow* | Aksi | Aktor | Sistem |
|  | Tambah | 1. Klik *button* form pemesanan pesan | 1. Sistem menampilkan form tambah data untuk pesanan |
|  |  | 1. Mengisi *form* tambah data pesanan dan klik *button submit* | 1. Sistem melakukan validasi pada data yang diinputkan sebelum data dimasukan di dalam *database*. |
|  | Ubah | 1. Klik *button* data pesanan | 1. Sistem akan menampilkan data pesanan |
|  |  | 1. Klik u*pdate Progress* | 1. Sistem akan menampilkan form untuk *update progress* pesanan. |
|  |  | 1. Menginputkan perubahan data kemudian klik *button submit* | 1. Sistem melakukan validasi data yang telah diinputkan user dan menyimpan data ke dalam *database.*   (*Alternatif flow* 6.1) |
| *Alternative Flow* | 6.1 Data yang diinputkan tidak sesuai dengan peraturan sistem (Kembali ke langkah 4) | | |
| *Requirement* | Berhasil login sebagai pegawai atau pelanggan | | |
| *Post-condition* | Sistem melakukan Data Manipulation Language (DML) pada database sesuai dengan aksi yang dilakukan aktor (tambah, dan ubah data) | | |

1. *Use Case Spesification* Bayar Pesanan

**Tabel 4.7** *Use Case Spesification* Bayar Pesanan

|  |  |  |  |
| --- | --- | --- | --- |
| *Use Case Name* | Update Transaksi | | |
| *Actor* | Pegawai | | |
| *Description* | *Use Case* ini digunakan oleh pelanggan untuk memberikan informasi bahwa pelanggan telah melakukan pembayaran. | | |
| *Pre-condition* | *User* dalam keadaan login | | |
| *Basic Flow* | Aksi | Aktor | Sistem |
|  | 1. Klik menu pesanan | 1. Sistem menampilkan data pesanan. |
|  |  | 1. Klik button bayar | 1. Sistem menampilkan form bayar pesanan |
|  |  | 1. User mengisi form bayar pesanan kemudian klik *submit* | 1. Sistem akan menampilkan data pesanan telah diubah. |
| *Alternative Flow* |  | | |
| *Requirement* | User login sebagai pegawai | | |
| *Post-condition* | Sistem melakukan Data Manipulation Language (DML) pada database sesuai dengan aksi yang dilakukan aktor (ubah data) | | |

1. *Use Case Spesification* *Approve* Pembayaran

**Tabel 4.8** *Use Case Spesification* Approve Pembayaran

|  |  |  |  |
| --- | --- | --- | --- |
| *Use Case Name* | Approve Pembayaran | | |
| *Actor* | Pegawai | | |
| *Description* | *Use Case* ini digunakan oleh pegawai untuk mengkonfirmasi bahwa pelanggan telah melakukan pembayaran. | | |
| *Pre-condition* | *User* dalam keadaan login | | |
| *Basic Flow* | Aksi | Aktor | Sistem |
|  | 1. Klik menu transaksi | 1. Sistem menampilkan data transaksi. |
|  |  | 1. Klik button *update* | 1. Sistem menampilkan form transaksi pesanan |
|  |  | 1. User mengisi form approve pembayaran *submit* | 1. Sistem akan menampilkan data transaksi telah diubah. |
| *Alternative Flow* |  | | |
| *Requirement* | User login sebagai pegawai | | |
| *Post-condition* | Sistem melakukan Data Manipulation Language (DML) pada database sesuai dengan aksi yang dilakukan aktor (ubah data) | | |

1. *Use Case Spesification* Lihat Produk

**Tabel 4.9** *Use Case Spesification* Lihat Produk

|  |  |  |  |
| --- | --- | --- | --- |
| *Use Case Name* | Lihat Produk | | |
| *Actor* | Pegawai dan pelanggan | | |
| *Description* | *Use Case* ini digunakan oleh pelanggan untuk melihat produk | | |
| *Pre-condition* | *User* dalam keadaan login | | |
| *Basic Flow* | Aksi | Aktor | Sistem |
|  | 1. User klik *button* produk | 1. Sistem menampilkan data produk yang disediakan konveksi |
| *Alternative Flow* | - | | |
| *Requirement* | User login sebagai pelanggan | | |
| *Post-condition* | Data produk berhasil ditampilkan | | |

1. *Use Case Spesification* Lihat Detail Produk

**Tabel 4.10** *Use Case Spesification* Lihat Detail Produk

|  |  |  |  |
| --- | --- | --- | --- |
| *Use Case Name* | Lihat Detail Produk | | |
| *Actor* | Pegawai dan pelanggan | | |
| *Description* | *Use Case* ini digunakan oleh pelanggan untuk melihat detail produk | | |
| *Pre-condition* | *User* dalam keadaan login | | |
| *Basic Flow* | Aksi | Aktor | Sistem |
|  | 1. User klik *button* produk | 1. Sistem menampilkan data produk yang disediakan konveksi |
|  |  | 1. User mengklik *link* detail produk | 1. Menampilkan halaman berisi detail produk |
| *Alternative Flow* | - | | |
| *Requirement* | User login sebagai pelanggan | | |
| *Post-condition* | Detail produk berhasil ditampilkan | | |

1. *Use Case Spesification* Input Data Pesanan

**Tabel 4.11** *Use Case Spesification* Input Data Pesanan

|  |  |  |  |
| --- | --- | --- | --- |
| *Use Case Name* | Input Data Pesanan | | |
| *Actor* | Pelanggan | | |
| *Description* | *Use Case* ini digunakan oleh pelanggan untuk Menginputkan detail pesanan produk | | |
| *Pre-condition* | *User* dalam keadaan login | | |
| *Basic Flow* | Aksi | Aktor | Sistem |
|  | 1. User klik *button* produkpada *navbar* | 1. Sistem menampilkan data produk yang disediakan konveksi |
|  |  | 1. User mengklik *button* detail produk | 1. Menampilkan halaman berisi detail produk |
|  |  | 1. *User* mengklik *button* pesan | 1. Menampilkan form detail pesanan |
|  |  | 1. *User* menginputkan detail pesanan dan klik *submit* | 1. Melakukan validasi pada data yang telah dimasukan *user* sebelum dimasukan ke dalam database.(Alternatif flow 8.1) |
| *Alternative Flow* | 8.1 Data yang diinputkan tidak sesuai dengan *requirements* sistem(Kembali ke Langkah 6) | | |
| *Requirement* | User login sebagai pelanggan | | |
| *Post-condition* | Berhasil menyimpan data pesanan ke dalam database | | |

1. *Use Case Spesification* Lihat Laporan Penjualan

**Tabel 4.12** *Use Case Spesification* Lihat Laporan Penjualan

|  |  |  |  |
| --- | --- | --- | --- |
| *Use Case Name* | Lihat Laporan Penjualan | | |
| *Actor* | Pegawai | | |
| *Description* | *Use Case* ini digunakan oleh pemilik konveksi untuk melihat laporan penjualan | | |
| *Pre-condition* | *User* dalam keadaan login | | |
| *Basic Flow* | Aksi | Aktor | Sistem |
|  | 1. User klik *button* penjualan | 1. Sistem menampilkan data penjualan. |
|  |  | 1. User mengklik *button download* untuk mengunduh laporan berupa *PDF* | 1. Melakukan pengunduhan file |
| *Alternative Flow* | - | | |
| *Requirement* | User login sebagai pemilik | | |
| *Post-condition* | Data penjualan berhasil ditampilkan | | |

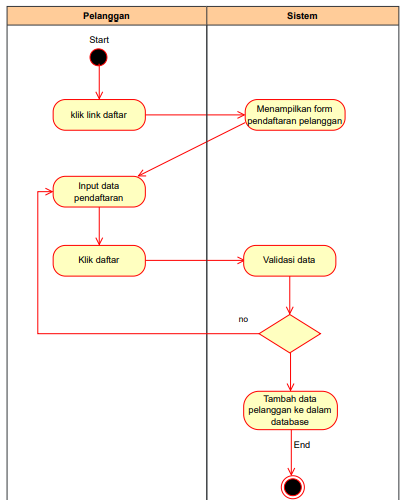
12 . *Use Case Spesification* Kelola Data Pelanggan

**Tabel 4.13** *Use Case Spesification* Kelola Data Pelanggan

|  |  |  |  |
| --- | --- | --- | --- |
| *Use Case Name* | Kelola Data Pelanggan | | |
| *Actor* | Pegawai | | |
| *Description* | *Use Case* ini digunakan untuk mengelola data pelanggan Persada Konveksi. Fungsi-fungsi yang ada di dalam *use case* ini yaitu lihat data pegawai. | | |
| *Pre-condition* | *User* dalam keadaan login | | |
| *Basic Flow* | Aksi | Aktor | Sistem |
|  | 1. Klik *button* data pelanggan | 1. Sistem menampilkan halaman data pelanggan |
| *Alternative Flow* |  | | |
| *Requirement* | Berhasil login sebagai pegawai | | |
| *Post-condition* | Data pelanggan berhasil ditampilkan | | |

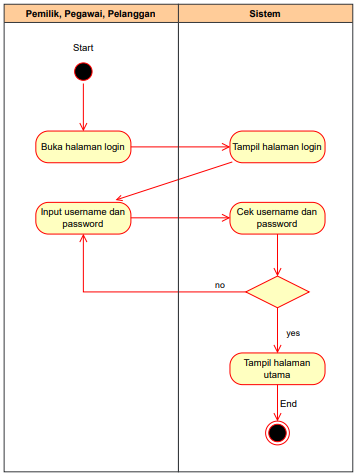
* + 1. **Activity Diagram**

1. *Activity Diagram* Registrasi



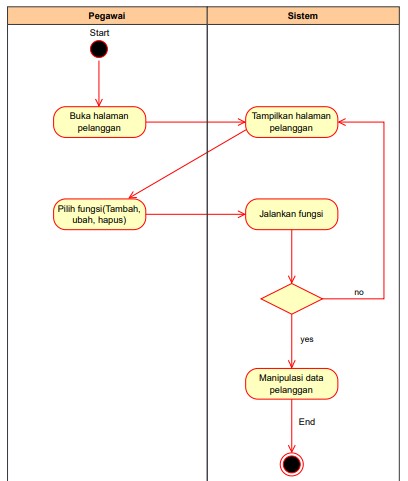
**Gambar 4.4** *Activity Diagram Registrasi*

1. *Activity Diagram* Login



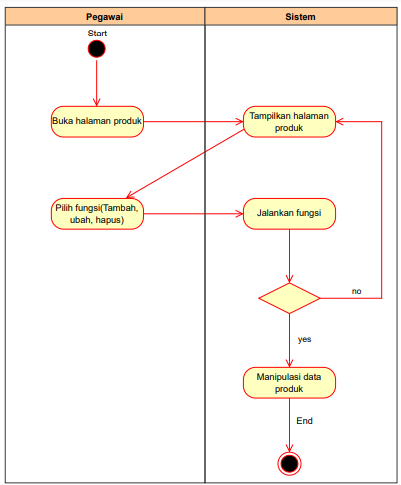
**Gambar 4.5** *Activity Diagram Login*

1. *Activity Diagram* Kelola Data Pelanggan



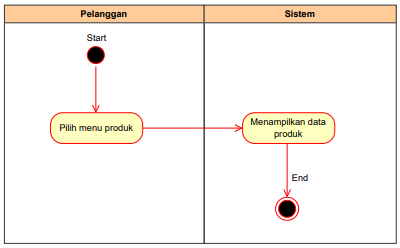
**Gambar 4.5** *Activity Diagram* Kelola Data Pelanggan

1. *Activity Diagram* Kelola Data Produk

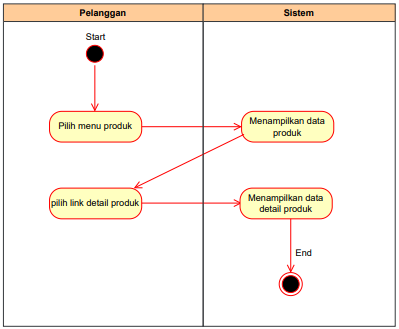


**Gambar 4.5** *Activity Diagram* Kelola Data Produk

1. *Activity Diagram* Lihat Produk

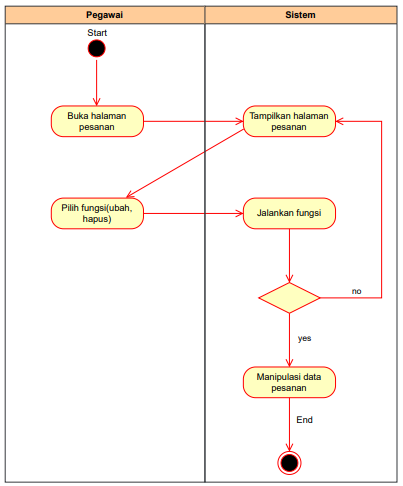


**Gambar 4.8** *Activity Diagram* Lihat Produk

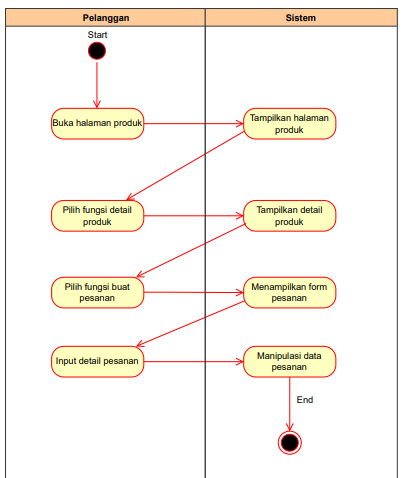


**Gambar 4.9** *Activity Diagram* Lihat Detail Produk

1. *Activity Diagram* Kelola Data Pesanan

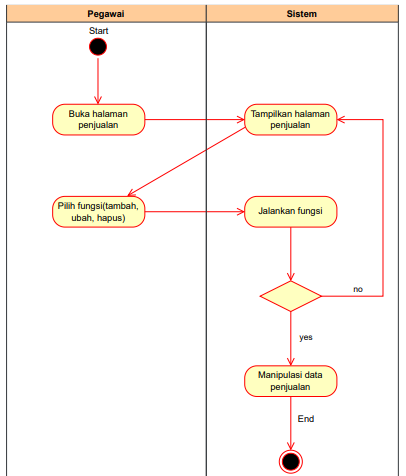


**Gambar 4.5** *Activity Diagram* Kelola Data Pesanan oleh Pegawai



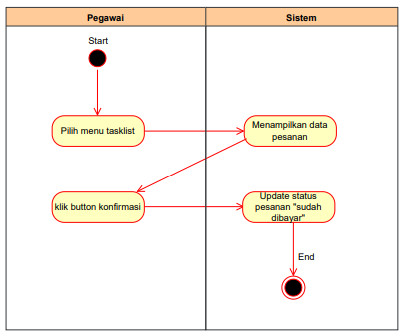
**Gambar 4.6** *Activity Diagram* Kelola Data Pesanan oleh Pelanggan

1. *Activity Diagram* Kelola Data Penjualan



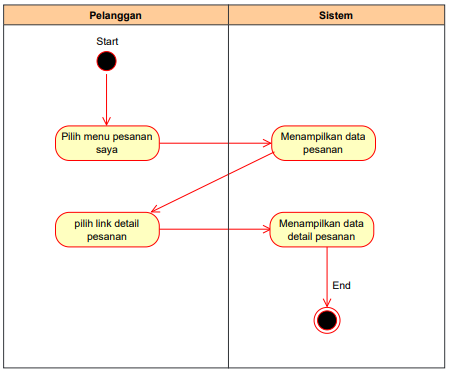
**Gambar 4.7** *Activity Diagram* Kelola Data Penjualan

1. *Activity Diagram* Konfirmasi Pembayaran



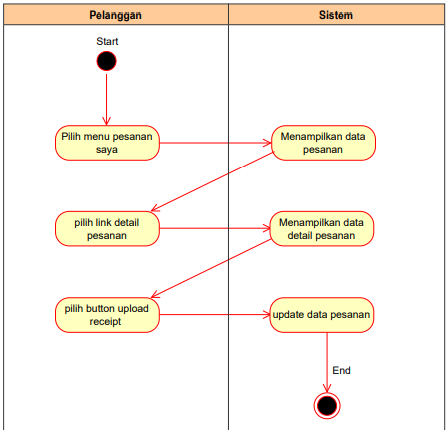
**Gambar 4.10** *Activity Diagram Approve* Pembayaran

1. *Activity Diagram* Lihat Progress Pesanan



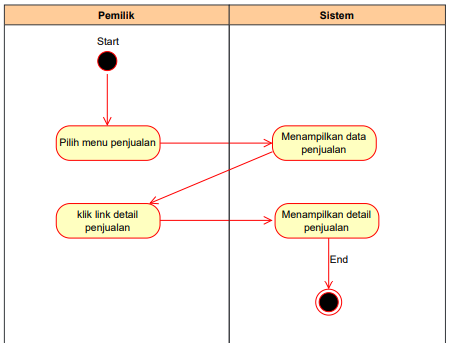
**Gambar 4.11** *Activity Diagram* Lihat Progress Pesanan

1. *Activity Diagram* *Bayar* Pesanan



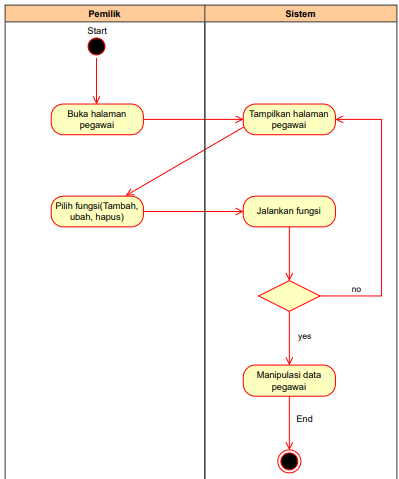
**Gambar 4.12** *Activity Diagram Upload* Bukti Pembayaran

1. *Activity Diagram* Lihat Laporan Penjualan

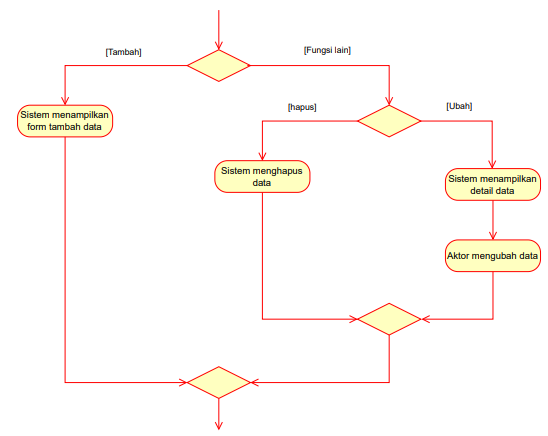


**Gambar 4.13** *Activity Diagram* Lihat Laporan Penjualan

1. *Activity Diagram* Kelola Data Pegawai

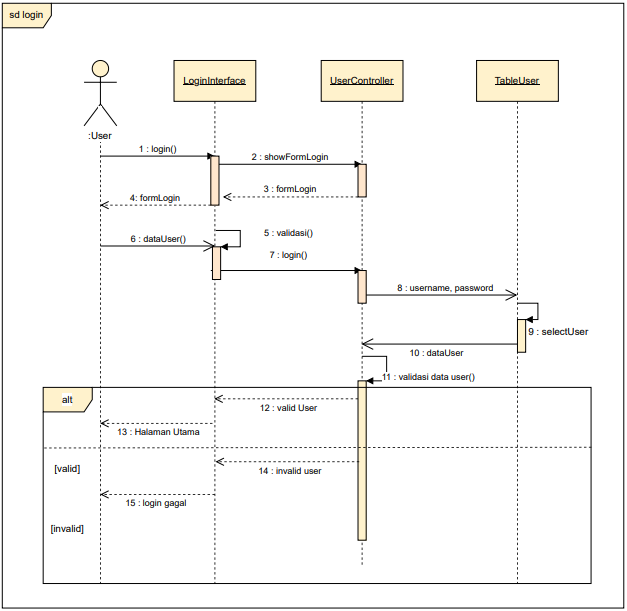


**Gambar 4.13** *Activity Diagram* Kelola Data Pegawai

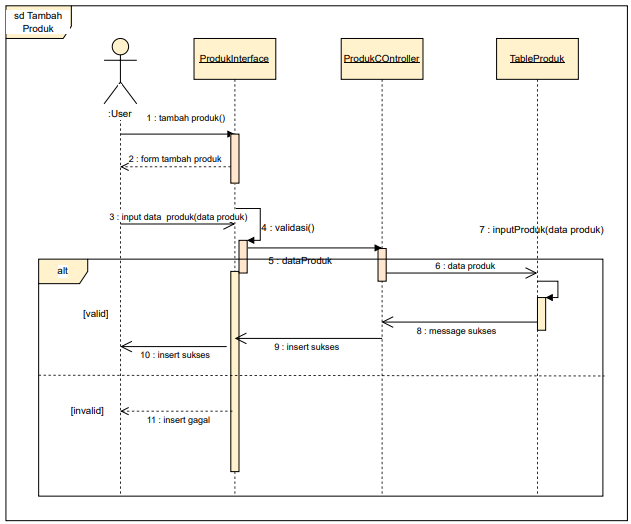
1. *SubActivity* Diagram Jalankan Fungsi

**Gambar 4.14** *SubActivity* Diagram Jalankan Fungsi

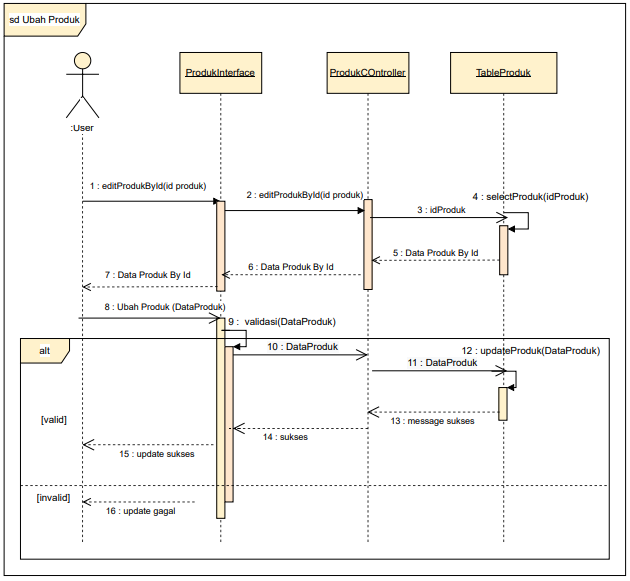
* + 1. **Sequence Diagram**



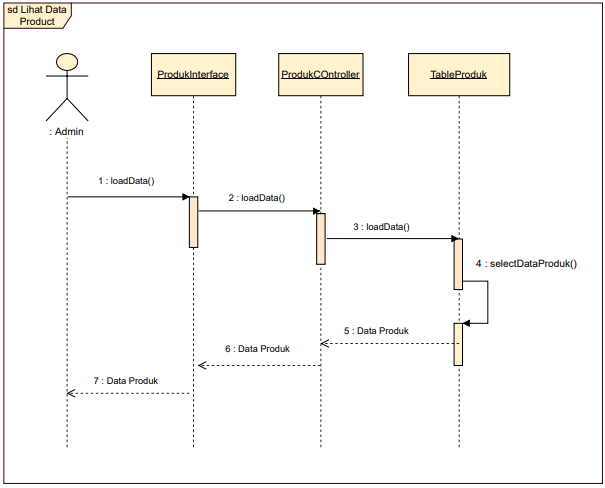
**Gambar 4.15** *Sequence Diagram Login*

**

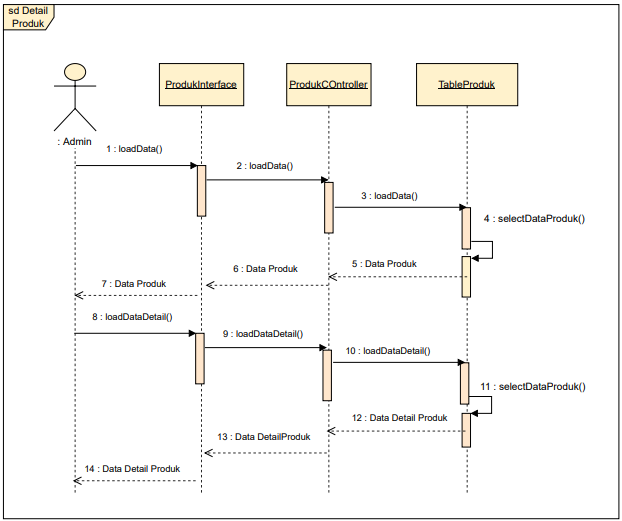
**Gambar 4.16** *Sequence Diagram* Tambah Produk



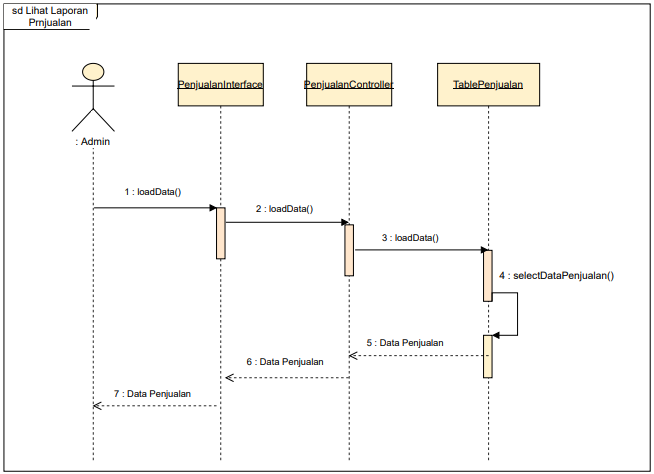
**Gambar 4.17** *Sequence Diagram* Ubah Produk



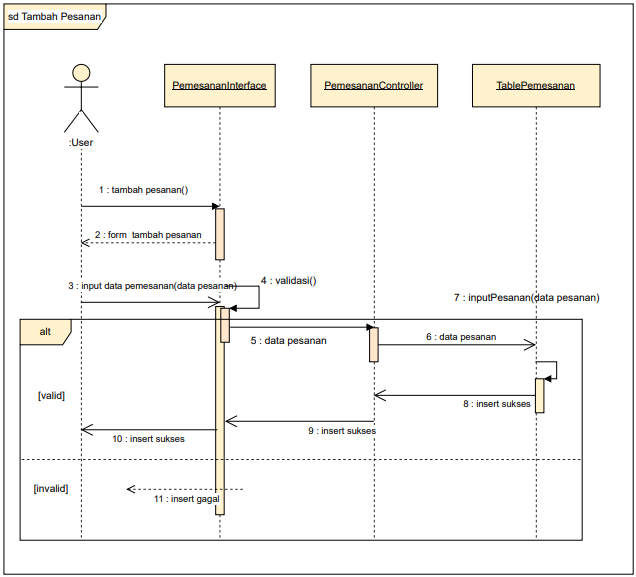
**Gambar 4.19** *Sequence Diagram* Lihat Data Produk

**

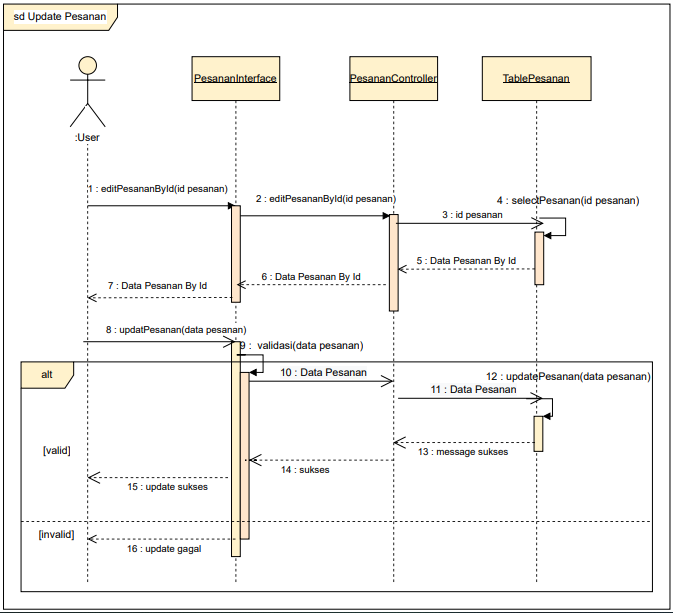
**Gambar 4.20** *Sequence Diagram* Lihat Detail Produk



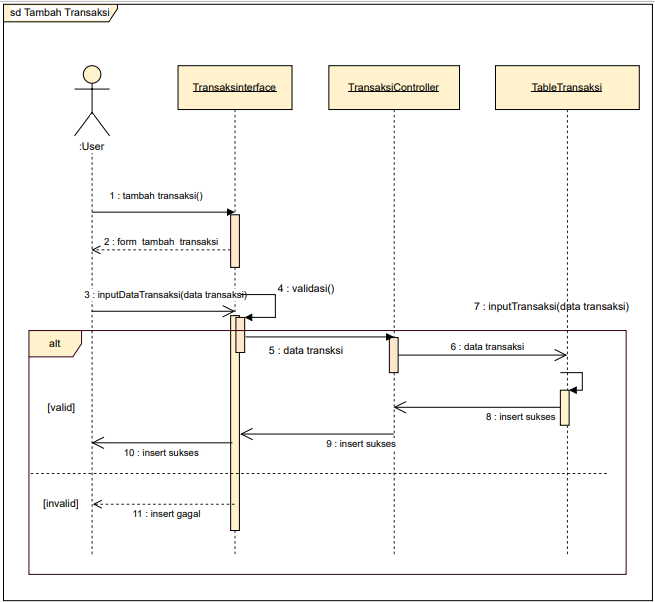
**Gambar 4.23** *Sequence Diagram* Lihat Laporan Penjualan



**Gambar 4.24** *Sequence Diagram* Tambah Pesanan

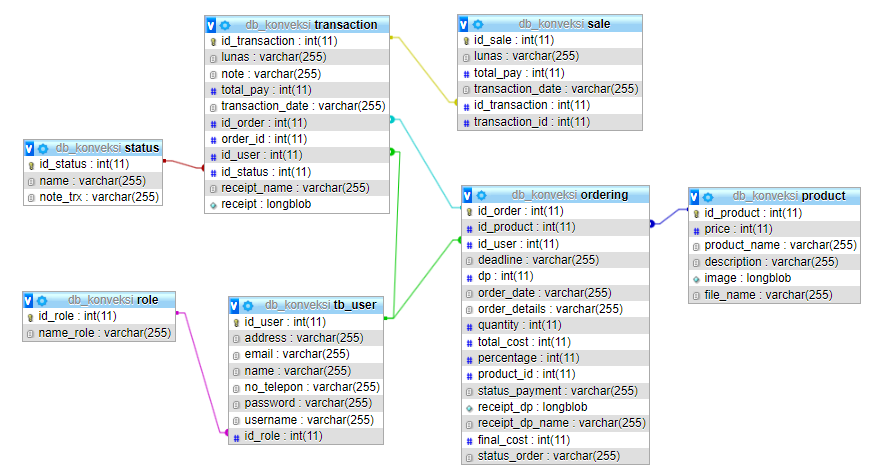


**Gambar 4.25** *Sequence Diagram* Ubah Pesanan



**Gambar 4.26** *Sequence Diagram* Tambah Transaksi

* + 1. Class Diagram
  1. **Perancangan Basis Data**
     1. **Skema Relasi Basis Data**

**

**Gambar 4**. Skema Relasi Basis Data

* + 1. **Deskripsi Tabel**

1. Table User

**Tabel 4.26** Deskripsi Tabel Login

|  |  |  |
| --- | --- | --- |
| Nama Kolom | Tipe Data | Keterangan |
| id\_user | int (11) | Primary Key |
| address | varchar(255) | - |
| email | varchar(255) | - |
| name | varchar(255) | - |
| no\_telepon | varchar(255) | - |
| password | varchar(255) | - |
| username | varchar(255) | - |
| id\_role | Int(11) | Foreign Key |

1. Table Role

|  |  |  |
| --- | --- | --- |
| Nama Kolom | Tipe Data | Keterangan |
| id\_role | int (11) | Primary Key |
| Name\_role | varchar(255) | - |

1. Table Product

|  |  |  |
| --- | --- | --- |
| Nama Kolom | Tipe Data | Keterangan |
| id\_product | int (11) | Primary Key |
| price | varchar(255) | - |
| product\_name | varchar(255) |  |
| description | varchar(255) |  |
| image | longblob |  |
| file\_name | varchar(255) |  |

1. Table Ordering

|  |  |  |
| --- | --- | --- |
| Nama Kolom | Tipe Data | Keterangan |
| id\_order | int (11) | Primary Key |
| id\_product | int (11) | Foreign Key |
| id\_user | int (11) | Foreign Key |
| deadline | varchar(255) | - |
| dp | int (11) | - |
| order\_date | varchar(255) | - |
| order\_details | varchar(255) | - |
| quantity | int (11) | - |
| total\_cost | varchar(255) | - |
| percentage | int (11) | - |
| status\_payment | varchar(255) | - |
| receipt\_dp | varchar(255) | - |
| receipt\_dp\_name | longlob | - |
| final\_cost | int (11) | - |
| status\_order | varchar(255) | - |

1. Table Transaction

|  |  |  |
| --- | --- | --- |
| Nama Kolom | Tipe Data | Keterangan |
| id\_transaction | int (11) | Primary Key |
| lunas | varchar(255) | - |
| note | varchar(255) | - |
| total\_pay | varchar(255) | - |
| transaction\_date | varchar(255) | - |
| Id\_order | varchar(255) | Foreign Key |
| Id\_user | int(11) | Foreign Key |
| id\_status | int(11) | Foreign Key |
| receipt\_name | varchar(255) | - |
| receipt | longlob | - |

1. Table Sale

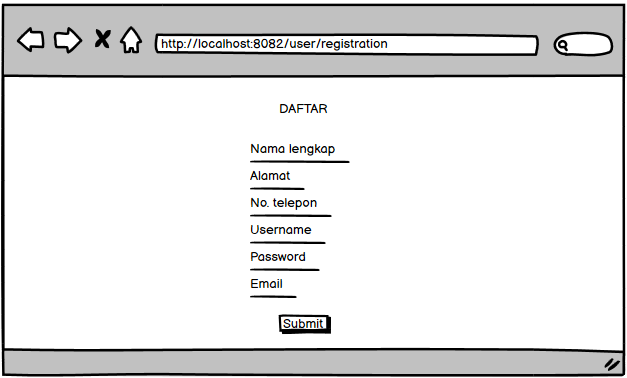
|  |  |  |
| --- | --- | --- |
| Nama Kolom | Tipe Data | Keterangan |
| id\_sale | int (11) | Primary Key |
| lunas | varchar(255) | - |
| total\_pay | varchar(255) | - |
| transaction\_date | varchar(255) | - |
| id\_transaction | int(11) | Foreign Key |

1. Table Status

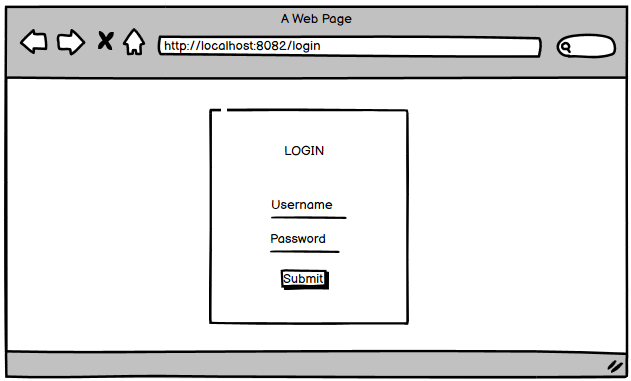
|  |  |  |
| --- | --- | --- |
| Nama Kolom | Tipe Data | Keterangan |
| id\_status | int (11) | Primary Key |
| name | varchar(255) | - |
| note\_trx | varchar(255) | - |

* 1. Perancangan Perangkat Lunak
     1. **Struktur Design Graphic User Interface (GUI) Umum**

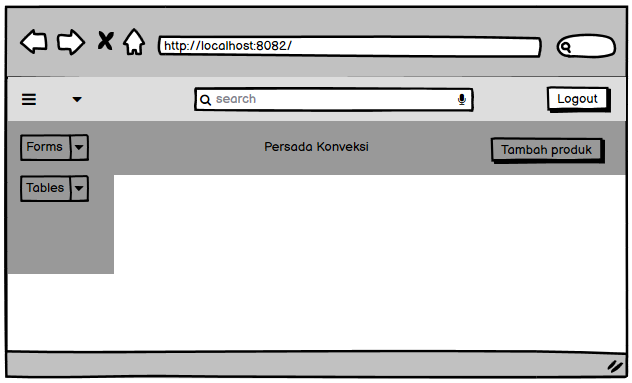
1. *User Interface* Daftar



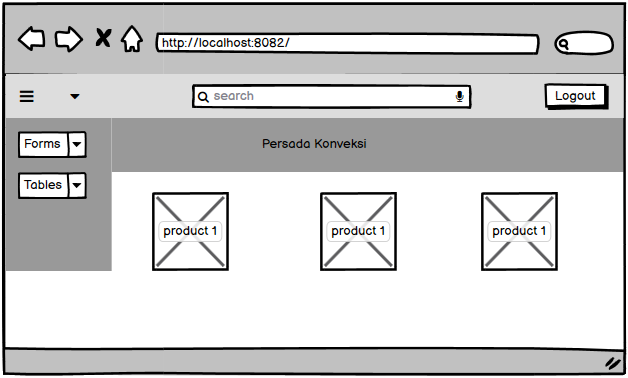
1. *User Interface* Login



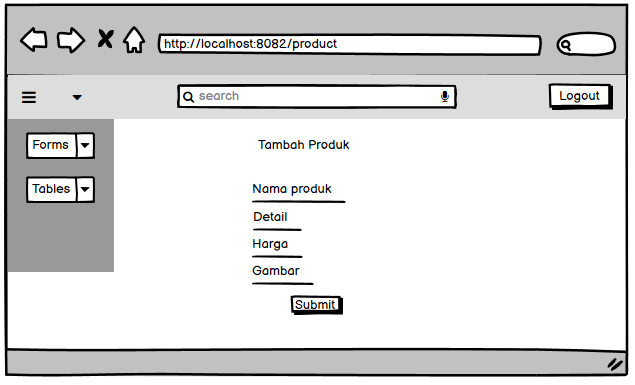
1. *User Interface Dashboard* Admin

**

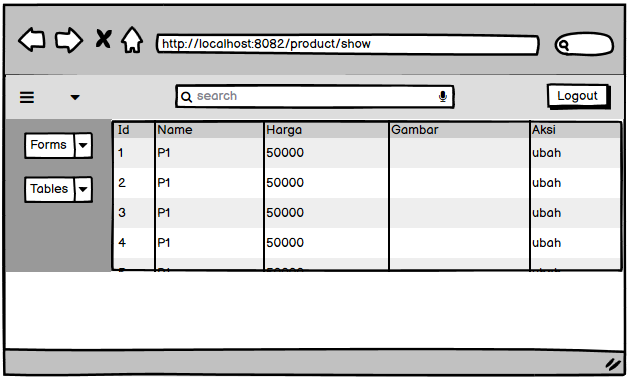
1. *User Interface Dashboard* Pelanggan

**

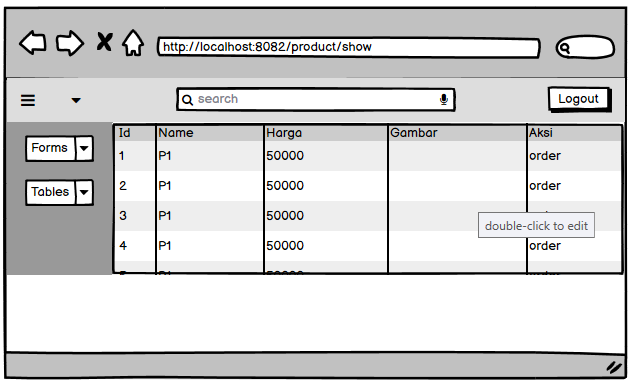
1. *User Interface* Tambah Produk

**

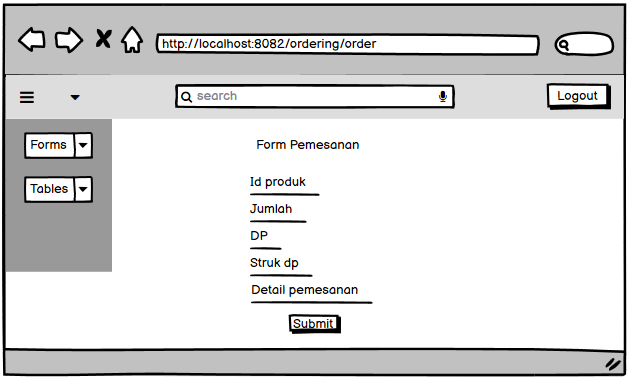
1. *User Interface* Tabel Produk Admin

**

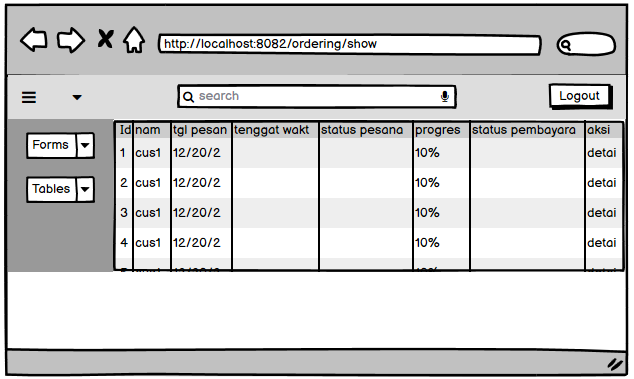
1. *User Interface* Tabel Produk Pelanggan



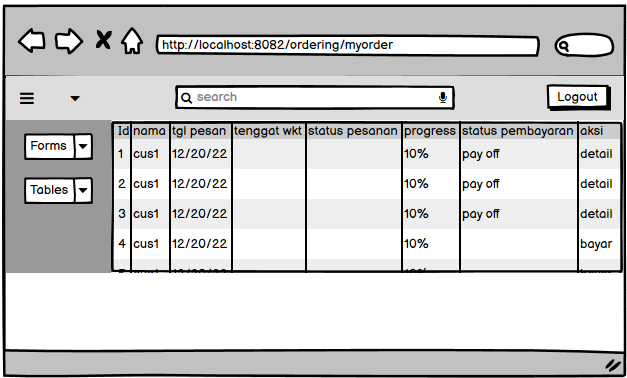
1. *User Interface* Form Pemesanan

**

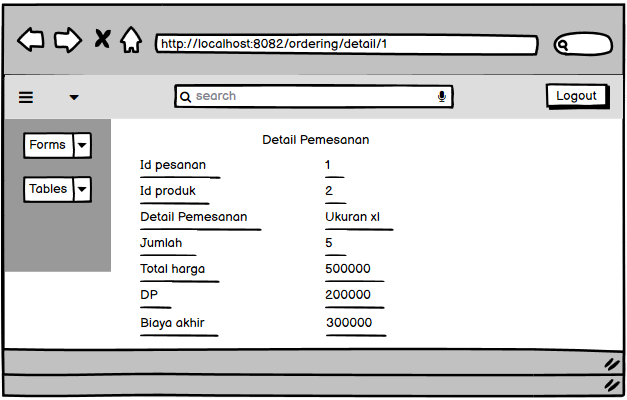
1. *User Interface* Tabel Pesanan Admin

**

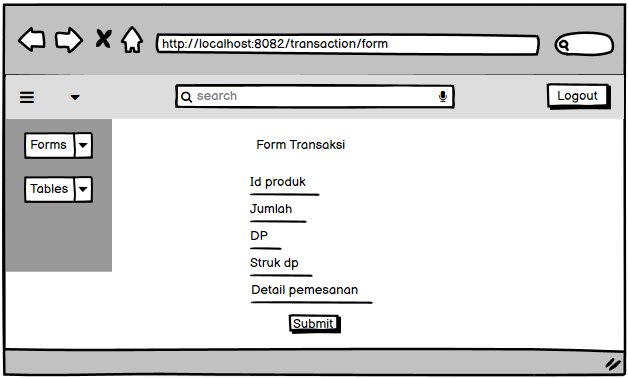
1. *User Interface* Tabel Pesanan Pelanggan

**

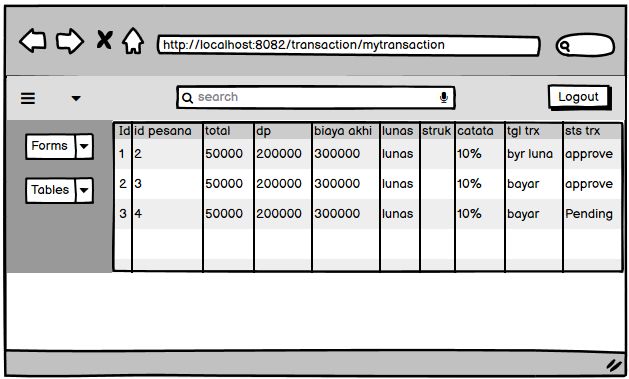
1. *User Interface* Tabel Detail Pesanan



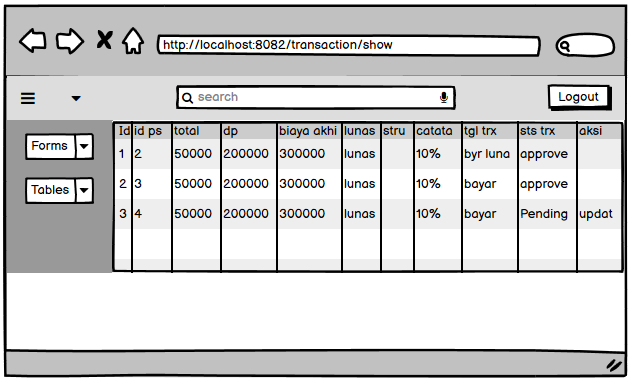
1. *User Interface* Form Transaksi

**

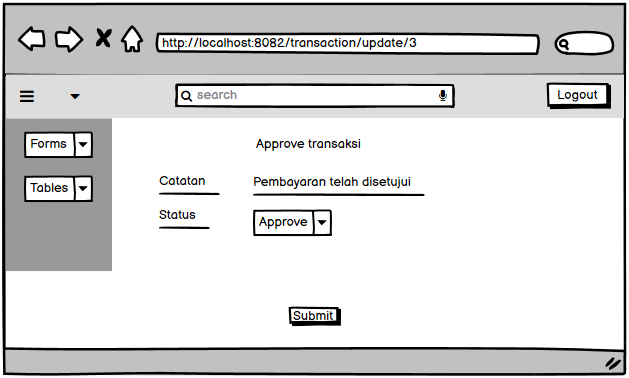
1. *User Interface* Transaksi Pelanggan



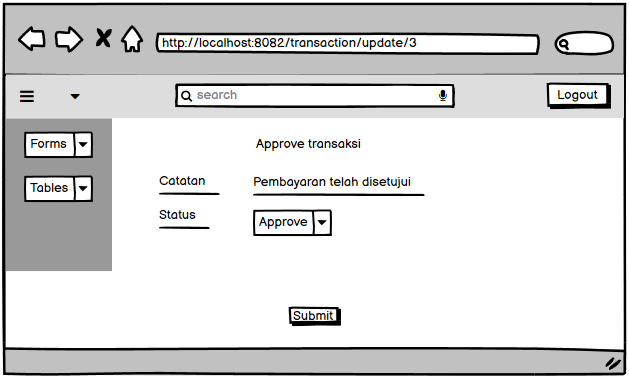
1. *User Interface* Transaksi Admin



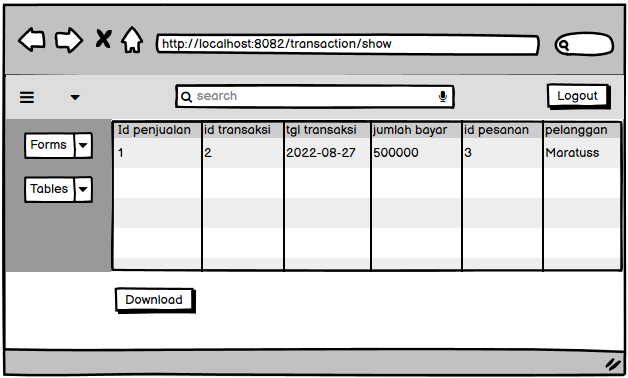
1. *User Interface Approve* Traksaksi



1. *User Interface Update*  Traksaksi



1. *User Interface Table* *Penjualan*



1. User Interface *Table* Pelanggan

