# BAB III ANALISIS DAN PERANCANGAN

## Analisis

### Analisa Data

Penelitian ini berfokus pada perancangan dan pengembangan sebuah sistem penjualan voucher game secara daring dengan penerapan metode antrian, yang bertujuan untuk menangani masalah kesalahan pencatatan mutasi saldo pengguna. Dalam era di mana permainan daring (game online) telah menjadi fenomena global, penyedia jasa daring menawarkan voucher game yang dapat diakses melalui platform daring. Namun, munculnya masalah dalam pencatatan mutasi saldo pengguna pada transaksi melalui sistem tersebut menjadi tantangan yang perlu diatasi.

Tujuan utama dari penelitian ini adalah merancang dan membangun sebuah platform penjualan voucher game daring yang berbasis website, serta menerapkan sistem antrian sebagai solusi untuk mengatasi permasalahan kesalahan pencatatan mutasi saldo pengguna. Dengan pendekatan ini, diharapkan pengguna dapat menghindari masalah pemotongan saldo yang tidak akurat, dan transaksi dapat dilakukan secara terstruktur dan efisien.

Glory Voucher menjadi objek penelitian selama 6 bulan, dimulai dari September 2022 hingga Februari 2023. Data dikumpulkan melalui wawancara dengan pemilik Glory Voucher untuk memahami secara mendalam proses transaksi dan kebutuhan data dalam pelaksanaan penjualan voucher game. Data ini kemudian dianalisis dengan metode deskriptif.

Hasil analisis data menunjukkan bahwa penerapan sistem antrian dalam transaksi merupakan solusi efektif untuk mengatasi permasalahan kesalahan pencatatan mutasi saldo pengguna. Dengan menerapkan sistem antrian, setiap transaksi dapat diproses secara teratur dan otomatis oleh sistem. Proses transaksi juga terintegrasi dengan Payment Gateway, yang memungkinkan validasi pembayaran otomatis dan memudahkan pengurus dalam melakukan verifikasi pembayaran.

Secara keseluruhan, penelitian ini akan menghasilkan sebuah platform penjualan voucher game daring yang menggunakan metode antrian. Platform ini diharapkan dapat memberikan kemudahan bagi pengguna dalam proses pembelian voucher game, serta membantu pengurus dalam mengelola keuangan dan mengatasi potensi kesalahan pencatatan mutasi saldo pengguna.

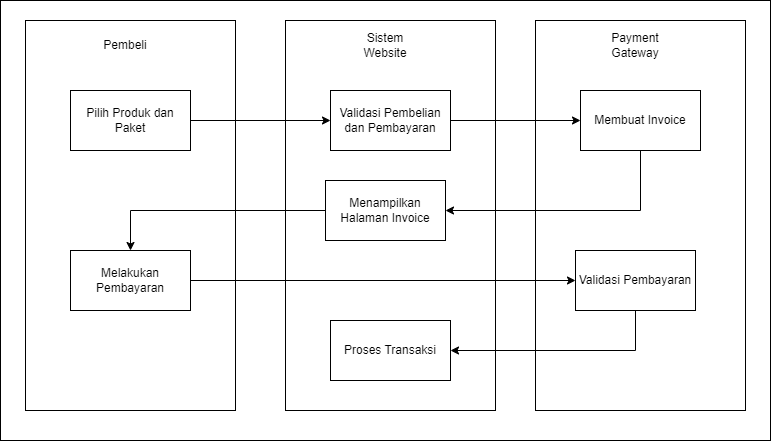
Dengan demikian, analisis deskriptif ini mencerminkan fokus dan tujuan dari penelitian yang telah Anda lakukan, serta memberikan gambaran yang jelas tentang proses perancangan sistem penjualan voucher game online dengan menerapkan metode antrian untuk menangani kesalahan pencatatan mutasi saldo pengguna.

### Identifikasi Masalah

Berdasarkan latar belakang yang telah di ungkapkan mengenai proses penjualan *voucher game* yang masih dilakukan secara manual. Meskipun dengan adanya website untuk menampung data pembelian namun proses transaksi tetap harus di proses oleh admin. Dan masalah yang sangat dirasa merugikan adalah ketika transaksi sedang padat dan satu *user* melakukan beberapa kali pembelian dalam satu waktu pemotongan saldo oleh sistem menjadi tidak akurat.

### Pemecahan Masalah

Berdasarkan permasalahan yang ada penulis akan membuatkan sebuah website yang dapat memproses segala jenis transaksi secara otomatis. Tidak hanya itu penulis juga akan membuat sistem antrian yang dapat mengatasi masalah pemotongan saldo yang tidak akurat. Pada Gambar 3.1 sampai Gambar 3.3 akan mejelaskan blok diagram proses pada aplikasi.

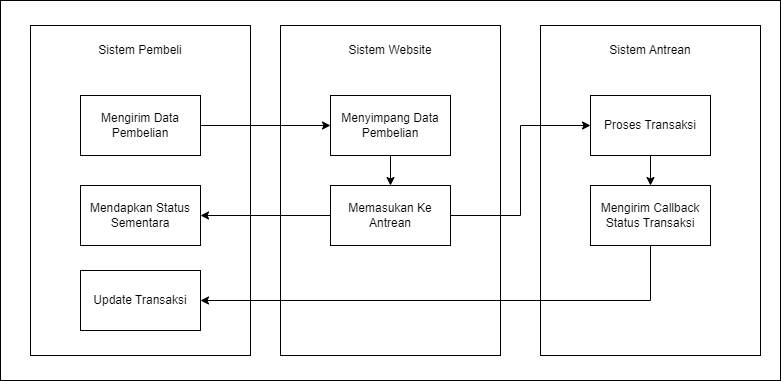


**Gambar 3.1** Blok Diagram Proses Pembelian dan Pembayaran

Gambar 3.1 mengilustrasikan blok diagram yang memaparkan langkah-langkah utama dalam proses ini, yang melibatkan tiga komponen utama: Pembeli, Sistem Website, dan Payment Gateway. Pada tahap awal, proses dimulai dengan tindakan pembelian yang diinisiasi oleh Pembeli. Kemudian, Pembeli mengirimkan permintaan pembelian ke Sistem Website, platform di mana transaksi dipicu dan dikelola. Setelah menerima permintaan tersebut, Sistem Website menginisiasi proses transaksi.

Setelah menerima permintaan, Sistem Website meneruskan informasi transaksi kepada Payment Gateway, yaitu entitas yang bertugas mengelola transaksi pembayaran. Fungsi utama Payment Gateway adalah untuk melakukan validasi terhadap pembayaran yang diajukan oleh Pembeli. Tahapan validasi ini mencakup verifikasi atas metode pembayaran yang digunakan, seperti kartu kredit, e-wallet, maupun merchant seperti alfamart.Apabila pembayaran berhasil melewati proses validasi yang dilakukan oleh Payment Gateway, Sistem Website akan menerima konfirmasi pembayaran yang sah.

Konfirmasi ini memungkinkan Sistem Website untuk melanjutkan dengan tahap pemrosesan transaksi pembelian voucher game online sesuai permintaan. Pembeli. Proses ini mencakup pembaruan saldo Pembeli dan penerbitan voucher game yang sesuai dengan permintaan.



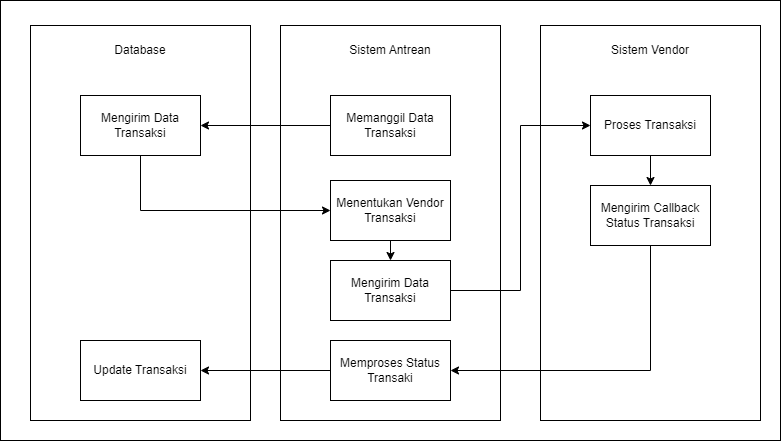
**Gambar 3.2** Blok Diagram Proses Pembelian Melalui API

Gambar 3.2 menggambarkan blok diagram yang menguraikan proses pembelian melalui API dalam Sistem Penjualan Voucher Game Online. Diagram ini terdiri dari tiga komponen utama: Sistem Pembeli, Sistem Website, dan Sistem Antrian.

Langkah pertama dalam proses ini dimulai ketika Sistem Pembeli, yang merupakan bagian dari sistem yang digunakan oleh pembeli, mengirimkan data pembelian ke Sistem Website. Sistem Pembeli bertanggung jawab atas inisiasi transaksi dan pengiriman informasi pembelian ke platform Sistem Website.

Setelah menerima data pembelian dari Sistem Pembeli, Sistem Website melakukan dua tugas utama. Pertama, ia menyimpan data transaksi pembelian yang diterima. Kedua, Sistem Website meneruskan data transaksi ke Sistem Antrian. Fungsi Sistem Antrian adalah untuk mengelola proses transaksi secara berurutan. Dalam konteks ini, setiap transaksi akan ditempatkan dalam antrian sesuai dengan urutannya untuk diproses lebih lanjut.

Saat tiba giliran transaksi dalam antrian, Sistem Antrian akan memproses transaksi tersebut. Proses ini melibatkan langkah-langkah untuk memastikan bahwa transaksi dilakukan dengan benar dan sesuai dengan data yang diberikan oleh Sistem Pembeli dan Sistem Website. Setelah transaksi selesai diproses, sistem akan mengirimkan callback ke Sistem Pembeli. Callback ini berisi informasi terbaru mengenai status transaksi, memberikan pembaruan kepada Sistem Pembeli mengenai transaksi yang telah selesai.



**Gambar 3.3** Blok Diagram Proses Transaksi Dalam Antrian

Pada gambar 3.3, ditampilkan blok diagram yang menggambarkan proses transaksi dalam antrian. Sistem antrian yang diterapkan dalam pembuatan website ini adalah jenis FIFO (First In First Out). Dalam sistem antrian FIFO ini, setiap transaksi yang masuk akan diprioritaskan untuk diproses berdasarkan urutan kedatangan. Dengan kata lain, transaksi yang pertama kali masuk ke antrian akan diberikan prioritas untuk diproses lebih dulu sebelum transaksi-transaksi yang masuk kemudian. Meskipun pengguna tidak secara langsung terlibat dalam mekanisme antrian ini yang berlangsung di lapisan backend, penerapan FIFO memastikan bahwa transaksi-transaksi diatasi dengan adil sesuai urutan masuknya. Ini membantu meminimalkan waktu tunggu, mengoptimalkan kinerja sistem, dan menjaga keselarasan antara permintaan masuk dan penanganan transaksi. Dengan adanya sistem antrian FIFO, proses transaksi akan dilakukan secara berurutan sehingga dapat mengatasi permasalah yang ada yaitu pemoongan saldo yang tidak akurat.

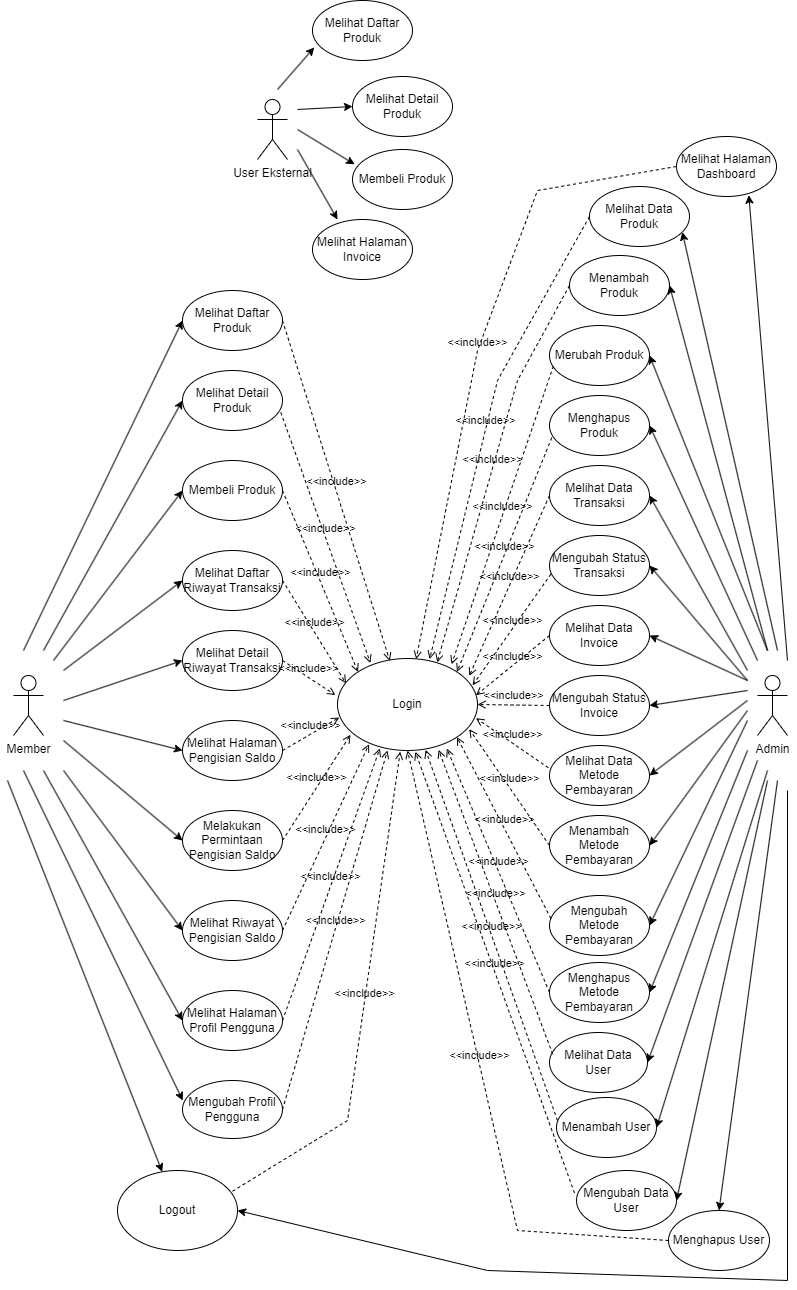
## Perancangan

Untuk perancangan sistem yang akan dibuat menggunakan model UML (*Unified Modeling Language*). Berikut merupakan diagram UML yang akan mendeskripsikan sistem yang akan dibangun:

### Perancangan Sistem

#### **Use Case**

*Use Case* merupakan sebuah interaksi antara aktor dan sistem. Dalam sistem yang akan dibuat, use case yang dibangun sebagai berikut:



**Gambar 3.4** *Use Case* Diagram

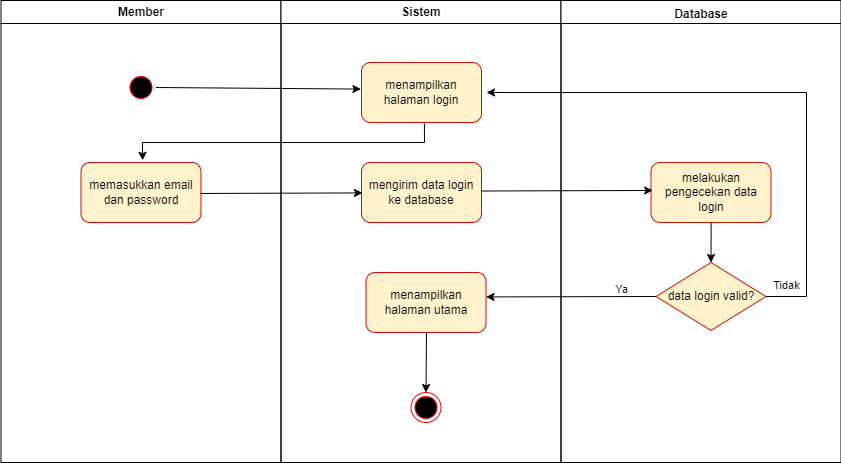
Sistem ini memiliki 3 aktor penting yaitu *Member*, Admin, dan *User* Eksternal. Setiap aktor memiliki peranan yang berbeda-beda yang dapat di deskripsikan sebagai berikut:

1. *Member*
2. *Member* dapat melakukan login ke dalam aplikasi.
3. *Member* dapat melihat daftar produk.
4. *Member* dapat melihat detail produk dan memilih paket.
5. *Member* dapat melakukan pembelian produk.
6. *Member* dapat melihat daftar riwayat transaksi.
7. *Member* dapat melihat detail riwayat transaksi.
8. *Member* dapat melihat halaman pengisian saldo.
9. *Member* dapat melakukan request pengisian saldo.
10. *Member* dapat melihat riwayat pengisian saldo.
11. *Member* dapat melihat halaman profil pengguna.
12. *Member* dapat mengubah profil pengguna.
13. *Member* dapat keluar dari aplikasi.
14. Admin
15. Admin dapat melihat halaman *dashboard.*
16. Admin dapat melihat daftar data produk.
17. Admin dapat menambah produk yang akan dijual.
18. Admin dapat mengubah produk yang telah ada.
19. Admin dapat menghapus produk.
20. Admin dapat melihat data transaksi.
21. Admin dapat merubah status transaksi.
22. Admin dapat melihat data *invoice* yang masuk.
23. Admin dapat mengubah status *invoice* yang telah ada.
24. Admin dapat melihat data metode pembayaran.
25. Admin dapat menambah metode pembayaran.
26. Admin dapat mengubah metode pembayaran yang telah ada.
27. Admin dapat menghapus metode pembayaran.
28. Admin dapat melihat data *user* yang telah terdaftar.
29. Admin dapat menambahkan data *user.*
30. Admin dapat mengubah data *user* yang telah terdaftar.
31. Admin dapat menghapus *user* yang telah terdaftar.
32. Admin dapat keluar dari aplikasi.
33. *User* Eksternal
34. *User* Eksternal dapat melihat list produk yang dijual.
35. *User* Eksternal dapat melihat detail produk serta paket yang tersedia.
36. *User* Eksternal dapat membeli paket dari produk yang tersedia.
37. *User* Eksternal dapat melihat detail *invoice* dari pembelian yang telah dilakukan.

#### **Activity Diagram**

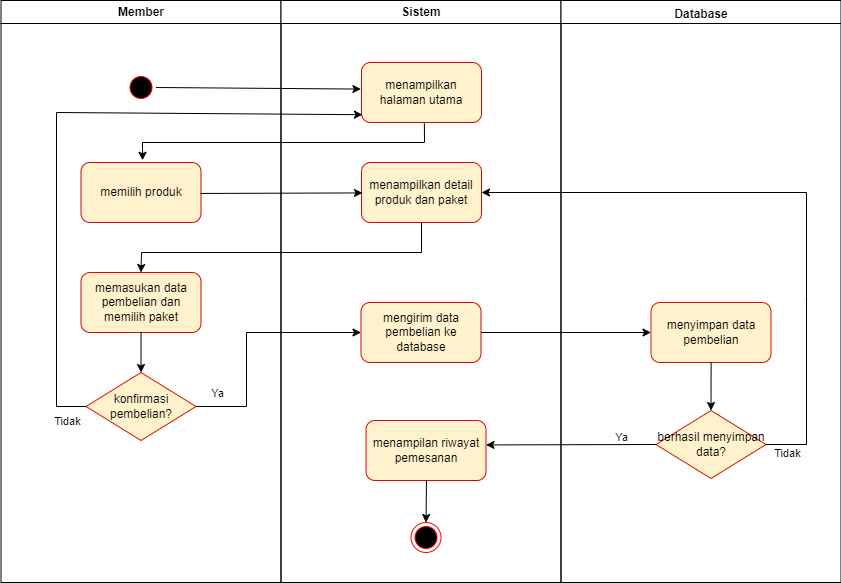
*Activity diagram* merupakan gambaran proses yang menerangkan tentang aktivitas-aktivitas pengguna yang akan diterapkan pada sistem informasi yang akan dibuat. Berikut *activity diagram* yang akan menggambarkan setiap kegiatan yang ada pada aplikasi yang akan dibuat:

1. *Activity Diagram Login Member*



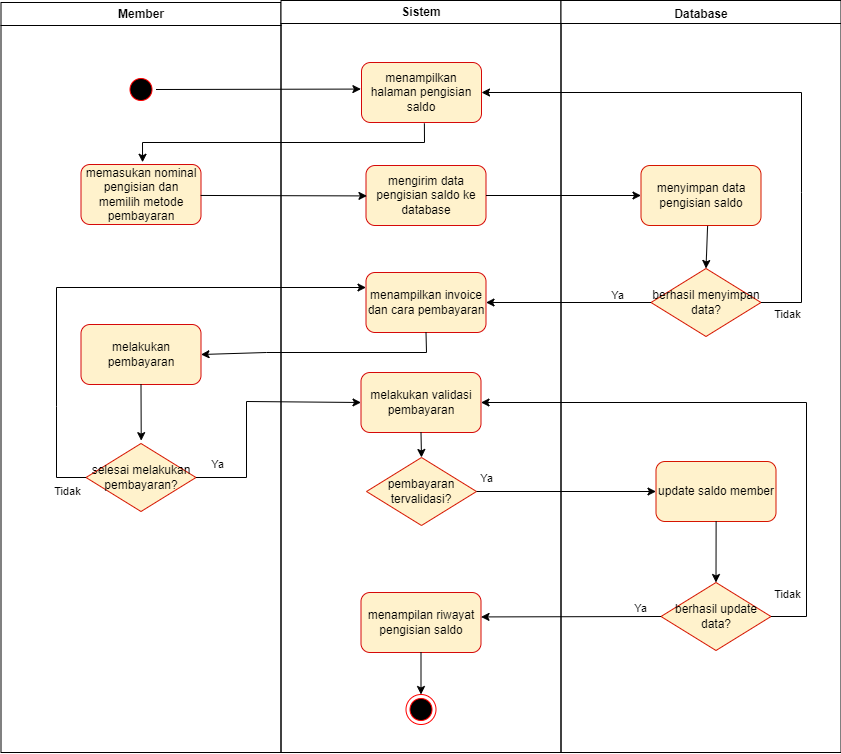
**Gambar 3.5** *Activity* Diagram Login Member

1. *Activity Diagram Member* Melakukan Pembelian



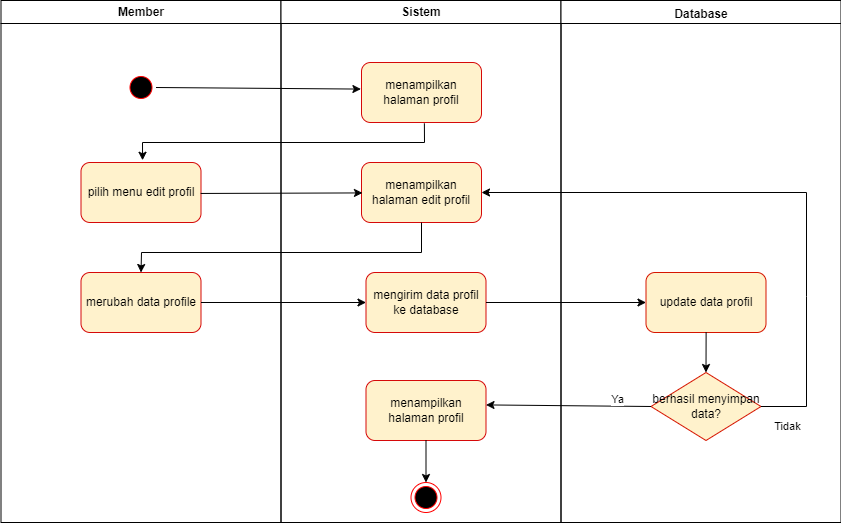
**Gambar 3.6** *Activity* Diagram Member Melakukan Pembelian

1. *Activity Diagram Member* Melakukan Pengisian Saldo



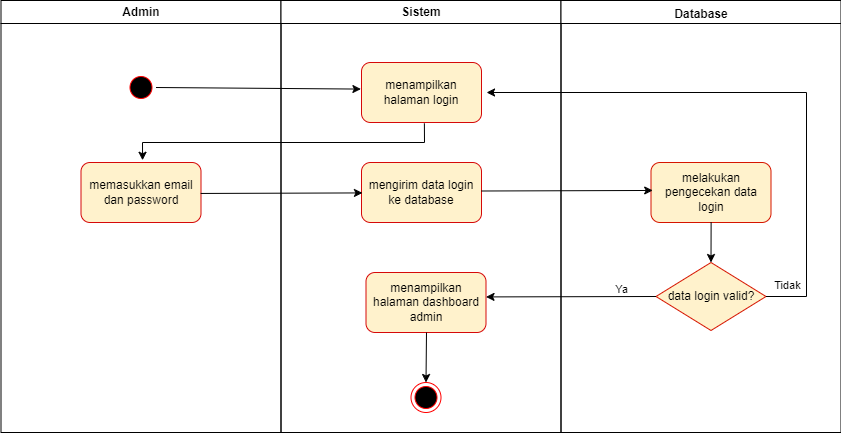
**Gambar 3.7** *Activity* Diagram Member Melakukan Pengisian Saldo

1. *Activity Diagram Member* Edit Profil



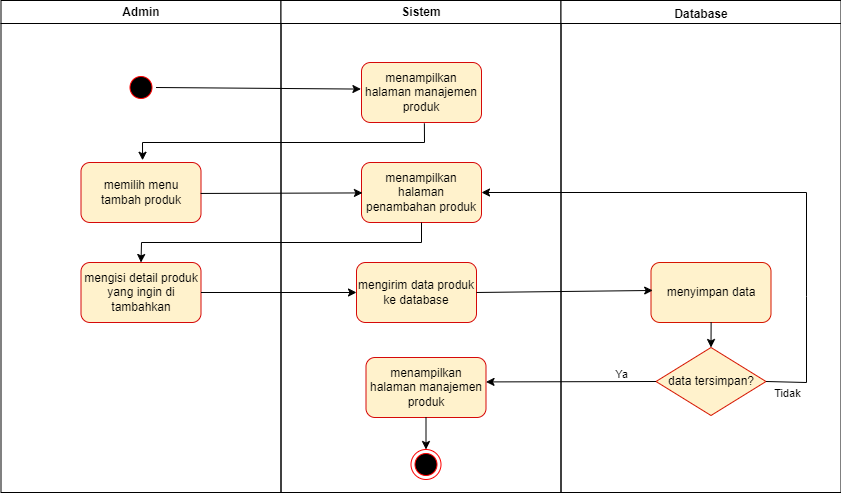
**Gambar 3.8** *Activity* Diagram Member Edit Profil

1. *Activity Diagram Login* Admin



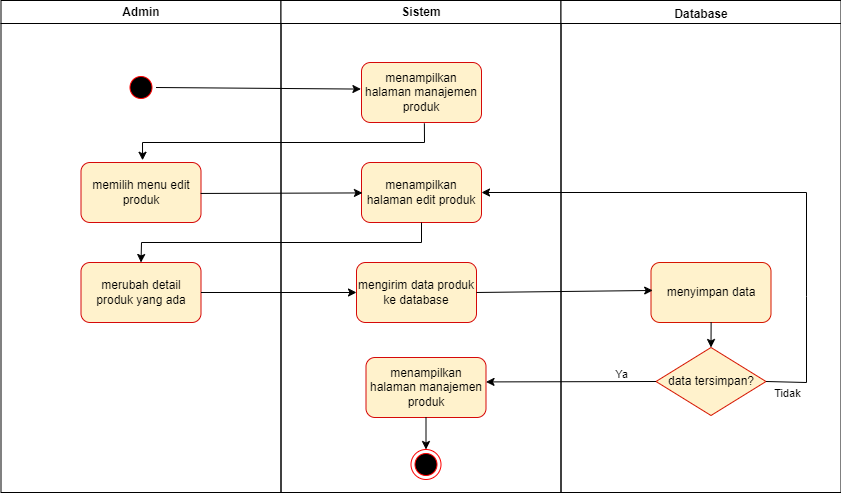
**Gambar 3.9** *Activity* Diagram *Login* Admin

1. *Activity Diagram* Admin Tambah Produk



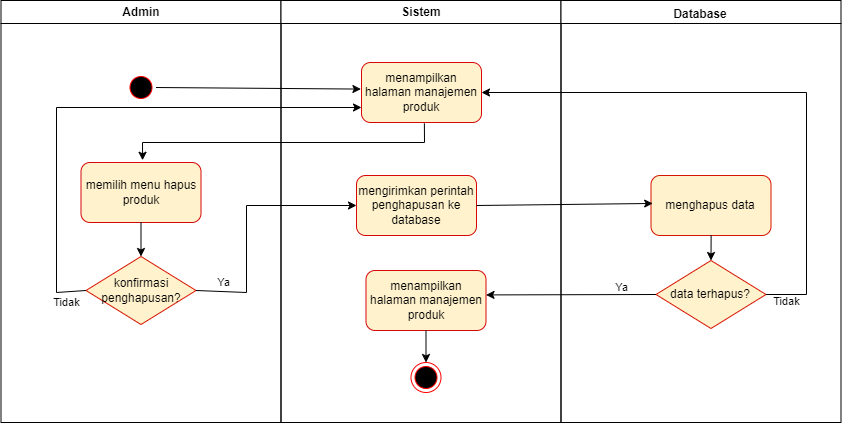
**Gambar 3.10** *Activity* Diagram Admin Tambah Produk

1. *Activity Diagram* Admin Edit Produk



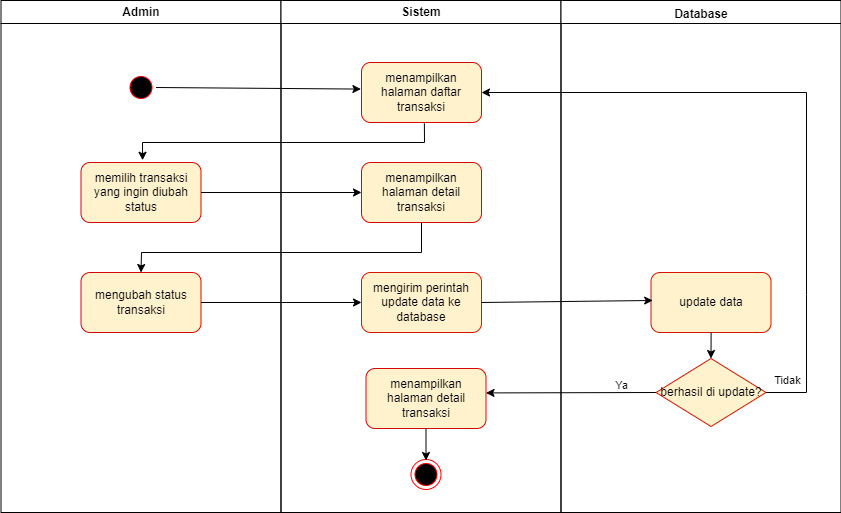
**Gambar 3.11** *Activity* Diagram Admin Edit Produk

1. *Activity Diagram* Admin Hapus Produk



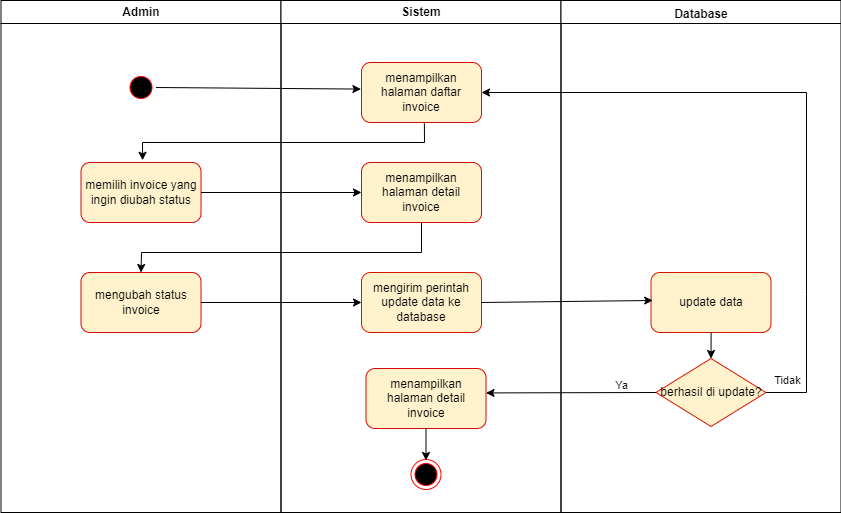
**Gambar 3.12** *Activity Diagram* Admin Hapus Produk

1. *Activity Diagram* Admin Ubah Status Transaksi



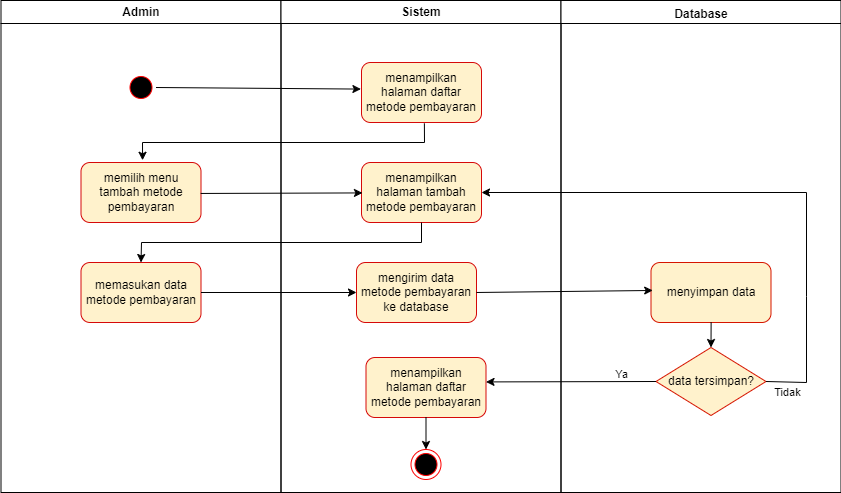
**Gambar 3.13** *Activity* Diagram Admin Ubah Status Transaksi

1. *Activity Diagram* Admin Ubah Status *Invoice*



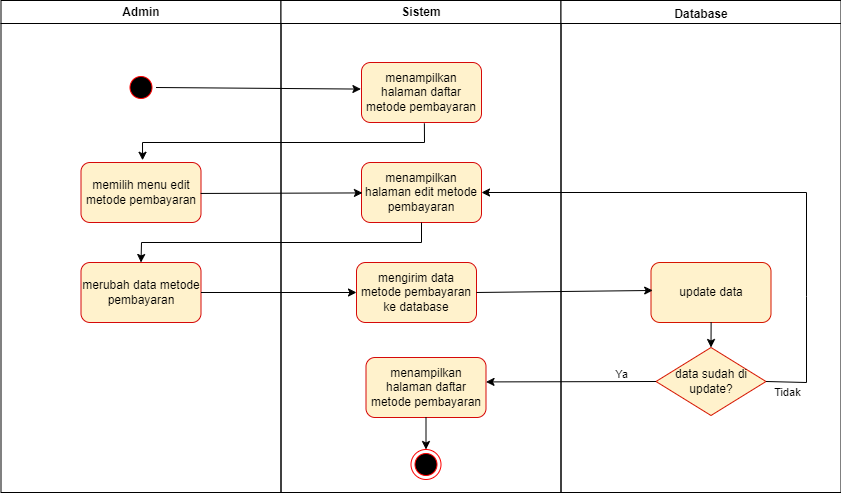
**Gambar 3.14** *Activity* Diagram Admin Ubah Status *Invoice*

1. *Activity Diagram* Admin Tambah Metode Pembayaran



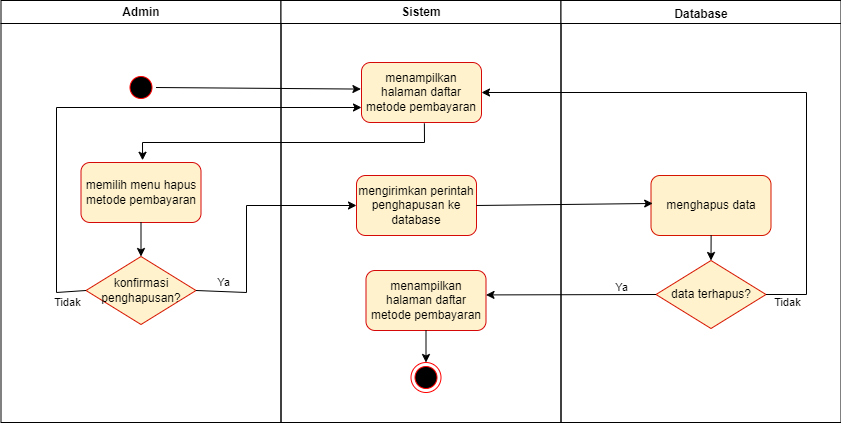
**Gambar 3.15** *Activity* Diagram Admin Tambah Metode Pembayaran

1. *Activity Diagram* Admin Edit Metode Pembayaran



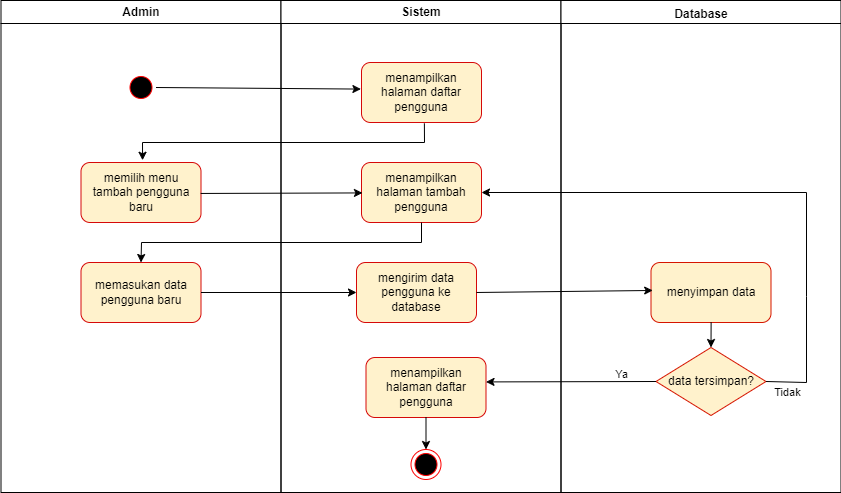
**Gambar 3.16** *Activity* Diagram Admin Edit Metode Pembayaran

1. *Activity Diagram* Admin Hapus Metode Pembayaran



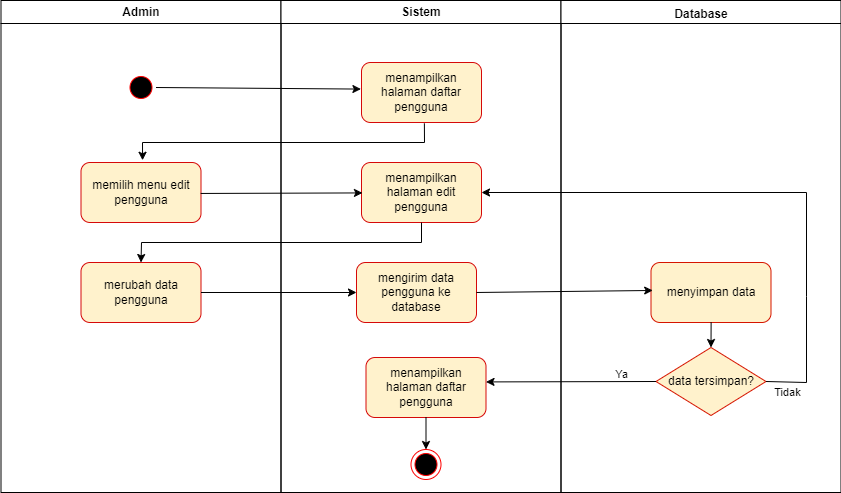
**Gambar 3.17** *Activity* Diagram Admin Hapus Metode Pembayaran

1. *Activity Diagram* Admin Tambah Pengguna Baru



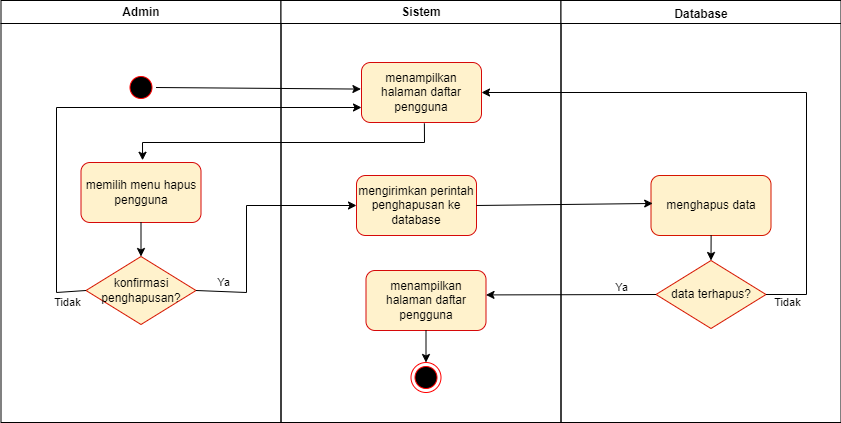
**Gambar 3.18** *Activity* Diagram Admin Tambah Pengguna Baru

1. *Activity Diagram* Admin Edit Data Pengguna



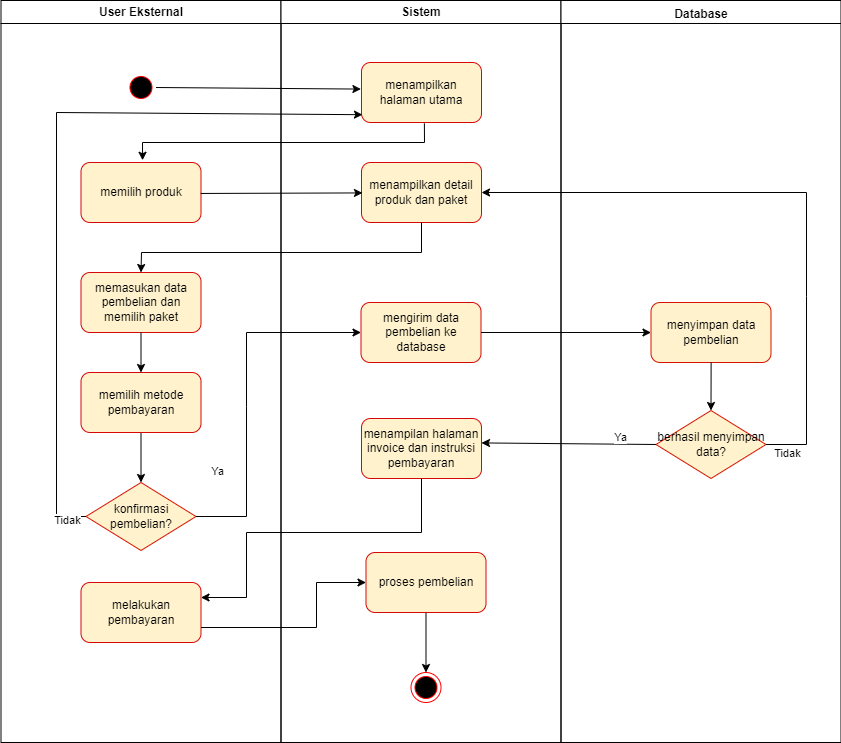
**Gambar 3.19** *Activity* Diagram Admin Edit Data Pengguna

1. *Activity Diagram* Admin Hapus Pengguna



**Gambar 3.20** *Activity* Diagram Admin Hapus Pengguna

1. *Activity Diagram User* Eksternal Melakukan Pembelian



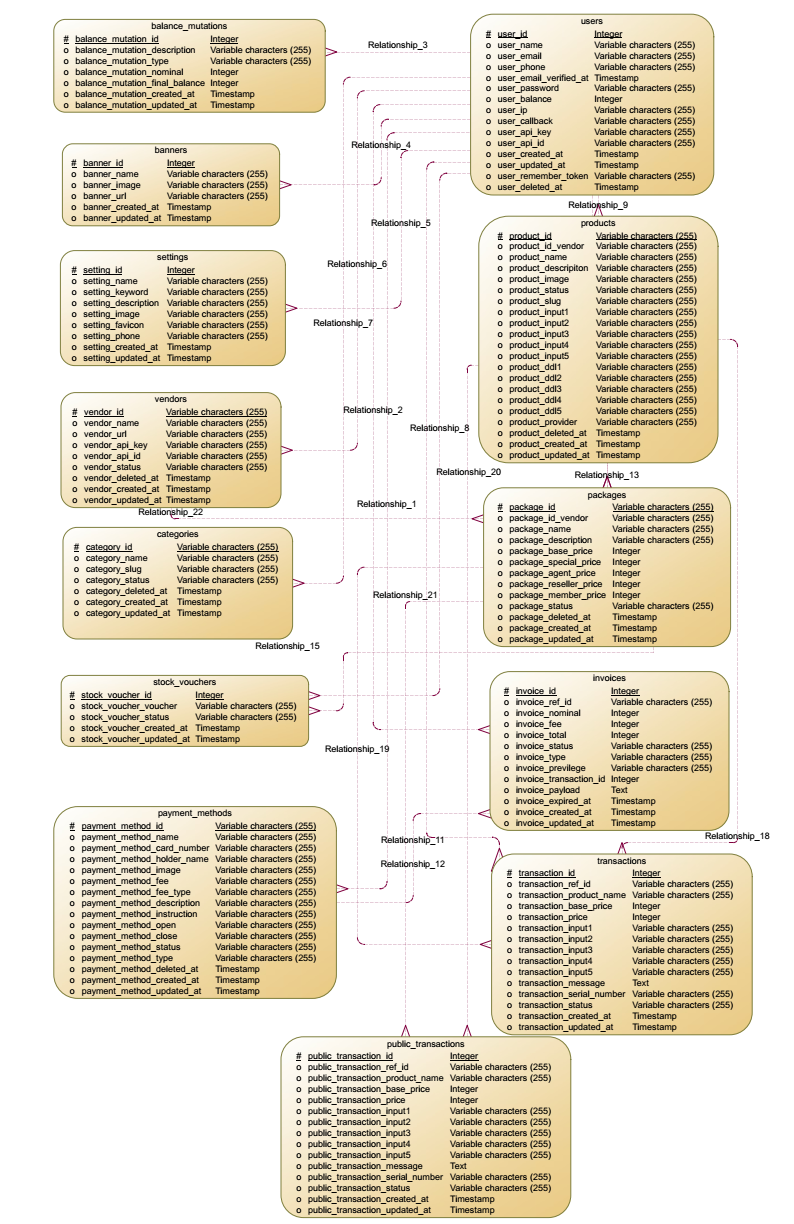
**Gambar 3.21** *Activity* Diagram User Eksternal Melakukan Pembelian

### Perancangan Data

Perancangan data terdiri dari *database* (basis data) yang menentukan data yang dibutuhkan. Berikut rancangan data yang akan digunakan:

#### **Conceptual Data Model (CDM)**

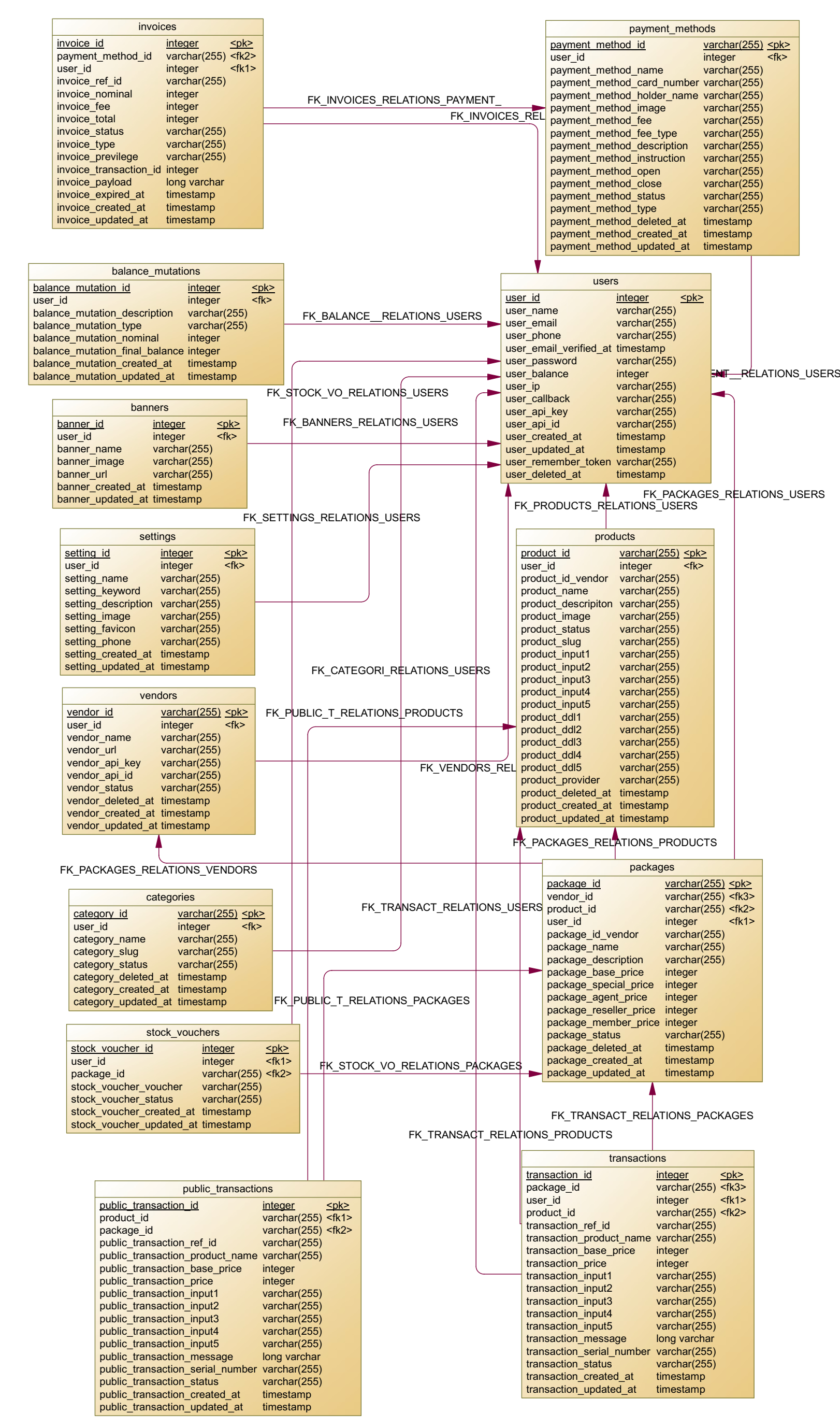
*Conceptual Data Model (CDM)* adalah model yang dibuat berdasarkan anggapan bahwa dunia nyata merupakan kumpulan obyek-obyek yang dinamakan entitas (*entity*)yang berhubungan (*relation)* dengan entitas-entitas lain.



**Gambar 3.22** *Conceptual Data Model (CDM)*

#### **Physical Data Model (PDM)**

*Physical Data Model (PDM)* merupakan hasil dari perancangan *database* secara konseptual dalam bentuk *Conceptual Data Model (CDM)*, digunakan untuk melakukan perancangan *database* secara fisik menggunakan Physical Data Model. PDM merupakan model relasional dari beberapa tabel untuk menggambarkan data yang saling terelasi. Setiap tabel memiliki nama kolom yang unik serta merupakan bentuk secara fisik dari *database* dan dapat langsung di implementasikan sebagai *database sistem*. Gambar dari *physical data model* tersebut dapat dilihat pada gambar 3.23.



**Gambar 3.23** *Physical Data Model (PDM)*

#### **Struktur Tabel**

Struktur tabel yang akan digunakan pada rancang bangun sistem penjualan voucher game online adalah sebagai berikut:

1. Tabel *Users*

Tabel *Users* digunakan untuk menyimpan data pengguna mulai dari member, reseller, maupun admin.

**Tabel 3.1** Tabel *Users*

|  |  |  |
| --- | --- | --- |
| Kolom | Tipe Data | Keterangan |
| user\_id | *integer* | *Primary Key* |
| user\_name | *varchar(255)* | - |
| user\_email | *varchar(255)* | - |
| user\_phone | *varchar(255)* | - |
| user\_email\_verified\_at | *long varchar* | - |
| user\_password | *long varchar* | - |
| user\_balance | *integer* | - |
| user\_level | *varchar(255)* | - |
| user\_ip | *varchar(255)* | - |
| user\_callback | *varchar(255)* | - |
| user\_api\_key | *varchar(255)* | - |
| user\_api\_id | *varchar(255)* | - |
| user\_created\_at | *timestamp* | - |
| user\_updated\_at | *timestamp* | - |
| user\_deleted\_at | *timestamp* | - |

1. Tabel *Products*

Tabel *Praducts* digunakan untuk menyimpan data produk yang akan ditampilkan.

**Tabel 3.2** Tabel *Products*

| Kolom | Tipe | Keterangan |
| --- | --- | --- |
| product\_id | *varchar(255)* | *Primary Key* |
| user\_id | *integer* | *Foreign Key* |
| product\_id\_vendor | *varchar(255)* | - |
| product\_name | *varchar(255)* | - |
| product\_description | *varchar(255)* | - |
| product\_image | *long varchar* | - |
| product\_status | *varchar(255)* | - |
| product\_slug | *long varchar* | - |
| product\_input1 | *varchar(255)* | - |
| product\_input2 | *varchar(255)* | - |
| product\_input3 | *varchar(255)* | - |
| product\_input4 | *varchar(255)* | - |
| product\_input5 | *varchar(255)* | - |
| product\_ddl1 | *varchar(255)* | - |
| product\_ddl2 | *varchar(255)* | - |
| product\_ddl3 | *varchar(255)* | - |
| product\_ddl4 | *varchar(255)* | - |
| product\_ddl5 | *varchar(255)* | - |
| product\_provider | *varchar(255)* | - |
| product\_created\_at | *timestamp* | - |
| product\_updated\_at | *timestamp* | - |
| product\_deleted\_at | *timestamp* | - |

1. Tabel *Packages*

Tabel *Packages* digunakan untuk menyimpan data paket yang tersedia untuk setiap produk yang ada.

**Tabel 3.3** Tabel *Packages*.

| Kolom | *Tipe* | Keterangan |
| --- | --- | --- |
| package\_id | *varchar(255)* | *Primary Key* |
| user\_id | *integer* | *Foreign Key* |
| product\_id | *varchar(255)* | *Foreign Key* |
| vendor\_id | *varchar(255)* | *Foreign Key* |
| package\_id\_vendor | *varchar(255)* | *-* |
| package\_name | *varchar(255)* | - |
| package\_description | *varchar(255)* | - |
| package\_base\_price | *integer* | - |
| package\_spesial\_price | *integer* | - |
| package\_agent\_price | *integer* | - |
| package\_reseller\_price | *integer* | - |
| package\_member\_price | *integer* | - |
| package\_status | *varchar(255)* | - |
| package\_created\_at | *timestamp* | - |
| package\_updated\_at | *timestamp* | - |
| package\_deleted\_at | *timestamp* | - |

1. Tabel *Payment Methods*

Tabel *Payment Methods* digunakan untuk menyimpan data metode pembayaran.

**Tabel 3.4** Tabel *Payment Methods*

|  |  |  |
| --- | --- | --- |
| Kolom | Tipe | Keterangan |
| payment\_method\_id | *varchar(255)* | *Primary Key* |
| user\_id | *integer* | *Foreign Key* |
| payment\_name | *varchar(255)* | - |
| payment\_card\_number | *varchar(255)* | - |
| payment\_holder\_name | *varchar(255)* | - |
| payment\_image | *long varchar* | - |
| payment\_fee | *integer* | - |
| payment\_fee\_type | *varchar(255)* | - |
| payment\_description | *varchar(255)* | - |
| payment\_instruction | *long varchar* | - |
| payment\_open | *time* | - |
| payment\_close | *time* | - |
| payment\_status | *varchar(255)* | - |
| payment\_type | *varchar(255)* | - |
| payment\_created\_at | *timestamp* | - |
| payment\_updated\_at | *timestamp* | - |
| payment\_deleted\_at | *timestamp* | - |

1. Tabel *Invoices*

Tabel *Invoices* digunakan untuk menyimpan data *invoice* pembelian dari pembeli.

**Tabel 3.5** Tabel *Invoices*

|  |  |  |
| --- | --- | --- |
| Kolom | Tipe | Keterangan |
| invoice\_id | *integer* | *Primary Key* |
| payment\_method\_id | *varchar(255)* | *Foreign Key* |
| user\_id | *Integer* | *Foreign Key* |
| invoice\_ref\_id | *varchar(255)* | - |
| invoice\_transaction\_id | *integer* | - |
| invoice\_nominal | *integer* | - |
| invoice\_fee | *integer* | - |
| invoice\_total | *integer* | - |
| invoice\_status | *varchar(255)* | - |
| invoice\_type | *varchar(255)* | - |
| invoice\_previlege | *varchar(255)* | - |
| invoice\_payload | *long varchar* | - |
| invoice\_created\_at | *timestamp* | - |
| invoice\_updated\_at | *timestamp* | - |

1. Tabel *Transactions*

Tabel *Transactions* digunakan untuk menyimpan data transaksi yang dilaksanakan oleh pengguna terdaftar.

**Tabel 3.6** Tabel *Transactions*

| Kolom | Tipe | *Keterangan* |
| --- | --- | --- |
| transaction\_id | *integer* | *Primary Key* |
| user\_id | *integer* | *Foreign Key* |
| product\_id | *varchar(255)* | *Foreign Key* |
| package\_id | *varchar(255)* | *Foreign Key* |
| transaction\_ref\_id | *varchar(255)* | *-* |
| transaction\_product\_name | *varchar(255)* | *-* |
| transaction\_base\_price | *integer* | *-* |
| transaction\_price | *integer* | *-* |
| transaction\_input1 | *long varchar* | *-* |
| transaction\_input2 | *long varchar* | *-* |
| transaction\_input3 | *long varchar* | *-* |
| transaction\_input4 | *long varchar* | *-* |
| transaction\_input5 | *long varchar* | *-* |
| transaction\_message | *long varchar* | *-* |
| transaction\_serial\_number | *long varchar* | *-* |
| transaction\_status | *varchar(255)* | *-* |
| transaction\_created\_at | *timestamp* | *-* |
| transaction\_updated\_at | *timestamp* | *-* |

1. Tabel *Public Transactions*

Tabel *Public Transactions* digunakan untuk menyimpan data transaksi *user* eksternal atau user yang tidak terdaftar dan melakukan pembelian secara langsung tanpa login.

**Tabel 3.7** Tabel *Public Transactions*

|  |  |  |
| --- | --- | --- |
| Kolom | Tipe | *Keterangan* |
| public\_transaction\_id | *integer* | *Primary Key* |
| product\_id | *varchar(255)* | *Foreign Key* |
| package\_id | *varchar(255)* | *Foreign Key* |
| public\_transaction\_ref\_id | *varchar(255)* | *-* |
| public\_transaction\_product\_name | *varchar(255)* | *-* |
| public\_transaction\_base\_price | *integer* | *-* |
| public\_transaction\_price | *integer* | *-* |
| public\_transaction\_input1 | *long varchar* | *-* |
| public\_transaction\_input2 | *long varchar* | *-* |
| public\_transaction\_input3 | *long varchar* | *-* |
| public\_transaction\_input4 | *long varchar* | *-* |
| public\_transaction\_input5 | *long varchar* | *-* |
| public\_transaction\_message | *long varchar* | *-* |
| public\_transaction\_serial\_number | *long varchar* | *-* |
| public\_transaction\_status | *varchar(255)* | *-* |
| public\_transaction\_created\_at | *timestamp* | *-* |
| public\_transaction\_updated\_at | *timestamp* | *-* |

1. Tabel *Categories*

Tabel *Categories* digunakan untuk menyimpan data kategori atau jenis produk.

**Tabel 3.8** Tabel *Categories*

|  |  |  |
| --- | --- | --- |
| Kolom | Tipe | *Keterangan* |
| category\_id | *varchar(255)* | *Primary Key* |
| user\_id | *integer* | *Foreign Key* |
| category\_name | *varchar(255)* | *-* |
| category\_slug | *varchar(255)* | *-* |
| category\_status | *varchar(255)* | *-* |
| category\_created\_at | *timestamp* | *-* |
| category\_created\_at | *timestamp* | *-* |
| category\_updated\_at | *timestamp* | *-* |

1. Tabel *Vendors*

Tabel *Vendors* adalah tabel yang digunakan untuk menyimpan data *vendor* atau data jalur transaksi yang ada.

**Tabel 3.9** Tabel *Vendors*

|  |  |  |
| --- | --- | --- |
| Kolom | Tipe | *Keterangan* |
| vendor\_id | *varchar(255)* | *Primary Key* |
| user\_id | *integer* | *Foreign Key* |
| vendor\_name | *varchar(255)* | *-* |
| vendor\_api\_key | *varchar(255)* | *-* |
| vendor\_api\_id | *varchar(255)* | *-* |
| vendor\_status | *varchar(255)* | *-* |
| vendor\_deleted\_at | *timestamp* | *-* |
| vendor\_created\_at | *timestamp* | *-* |
| vendor\_updated\_at | *timestamp* | *-* |

1. Tabel *Stock Vouchers*

Tabel *Stock Voucher* digunakan untuk menyimpan data *voucher game* yang akan di jual.

**Tabel 3.10** Tabel *Stock Vouchers*

|  |  |  |
| --- | --- | --- |
| Kolom | Tipe | *Keterangan* |
| stock\_voucher\_id | *integer* | *Primary Key* |
| user\_id | *integer* | *Foreign Key* |
| package\_id | *integer* | *Foreign Key* |
| stock\_voucher\_voucher | *varchar(255)* | *-* |
| stock\_voucher\_status | *varchar(255)* | *-* |
| stock\_voucher\_created\_at | *timestamp* | *-* |
| stock\_voucher\_updated\_at | *timestamp* | *-* |

1. Tabel *Balance Mutations*

Tabel *Blance Mutations* digunakan untuk menyimpan data mutasi saldo pengguna ketika melakukan pembelian maupun pengembalian saldo ketika transaksi gagal.

**Tabel 3.11** Tabel *Balance Mutations*

|  |  |  |
| --- | --- | --- |
| Kolom | Tipe | *Keterangan* |
| balance\_mutation\_id | *integer* | *Primary Key* |
| user\_id | *integer* | *Foreign Key* |
| balance\_mutation\_description | *varchar(255)* | *-* |
| balance\_mutation\_type | *varchar(255)* | *-* |
| balance\_mutation\_nominal | *integer* | *-* |
| balance\_mutation\_final\_balance | *integer* | *-* |
| balance\_mutation\_created\_at | *timestamp* | *-* |
| balance\_mutation\_updated\_at | *timestamp* | *-* |

1. Tabel *Banners*

Tabel *Banners* merupakan table yang menyimpang data banner yang bertujuan untuk di tampilkan di halaman utama website.

**Tabel 3.12** Tabel *Banners*

|  |  |  |
| --- | --- | --- |
| Kolom | Tipe | *Keterangan* |
| banner\_id | *integer* | *Primary Key* |
| user\_id | *integer* | *Foreign Key* |
| banner\_name | *varchar(255)* | *-* |
| banner\_url | *varchar(255)* | *-* |
| banner\_image | *varchar(255)* | *-* |
| banner\_created\_at | *timestamp* | *-* |
| banner\_updated\_at | *timestamp* | *-* |

1. Tabel *Settings*

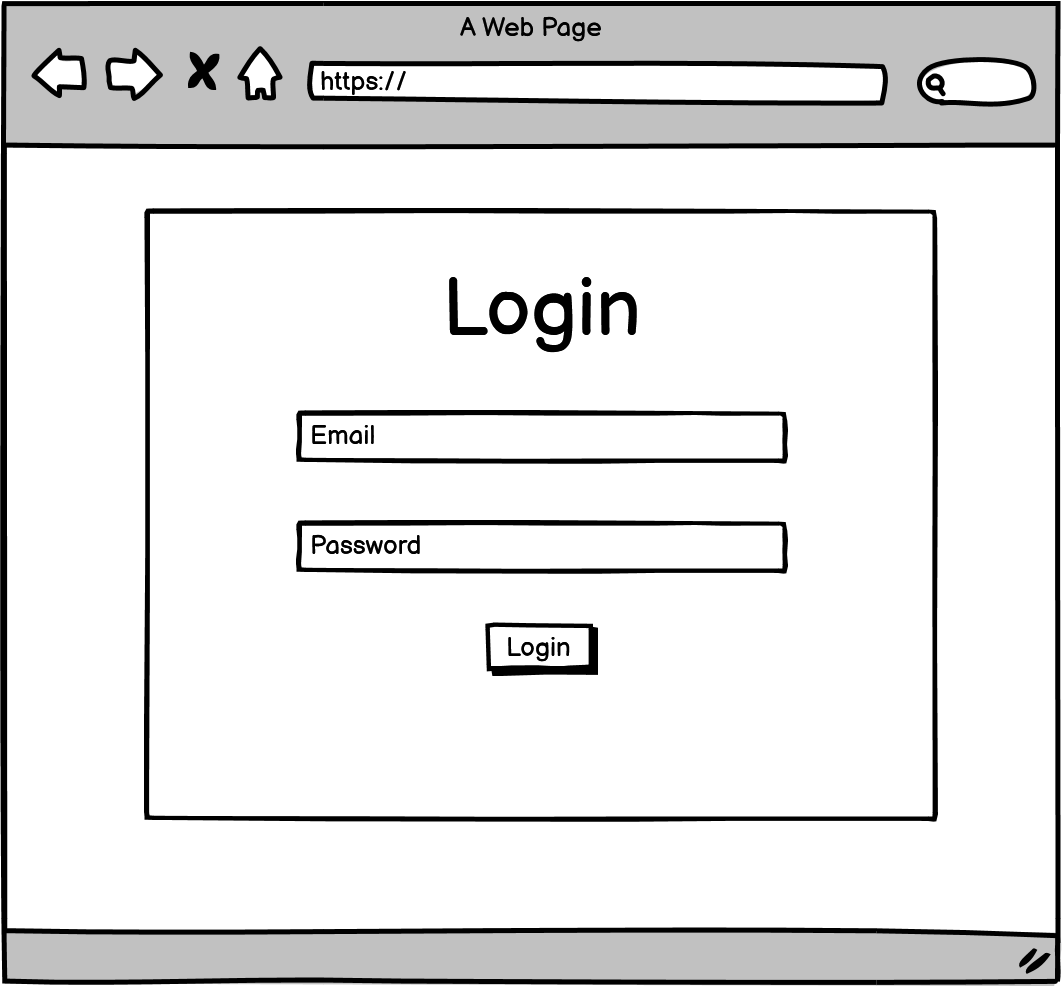
Tabel *Settings* digunakan untuk menyimpan data konfigurasi website seperti nama website, logo, dan juga deskripsi.

**Tabel 3.13** Tabel *Settings*

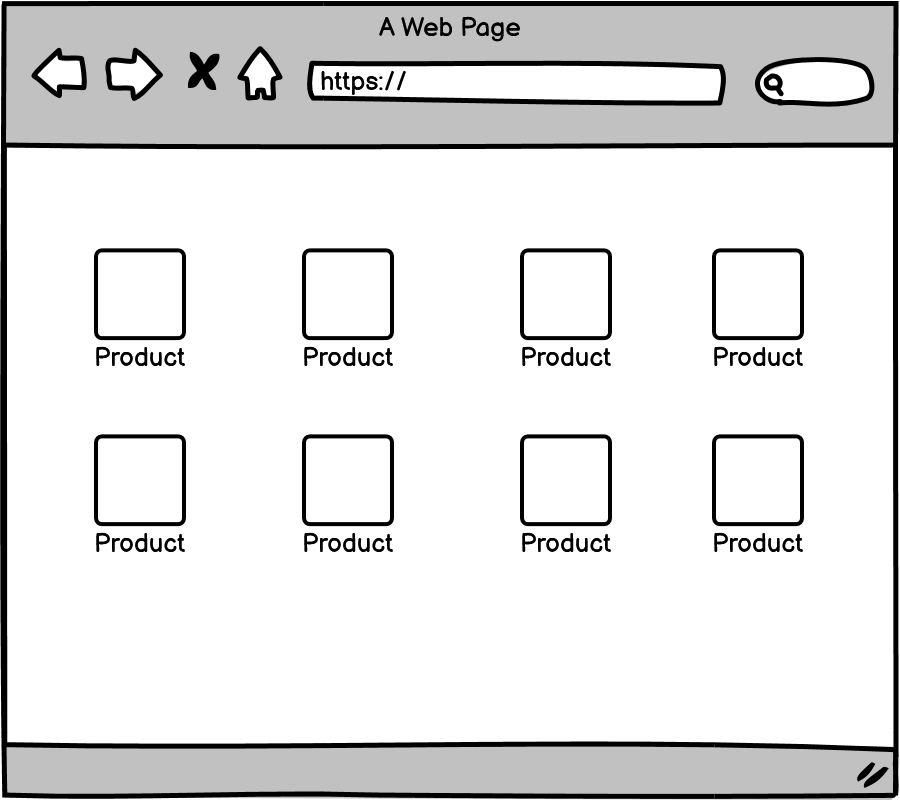
|  |  |  |
| --- | --- | --- |
| Kolom | Tipe | *Keterangan* |
| setting\_id | *integer* | *Primary Key* |
| user\_id | *integer* | *Foreign Key* |
| setting\_name | *varchar(255)* | *-* |
| setting\_descripiton | *varchar(255)* | *-* |
| setting\_keyword | *varchar(255)* | *-* |
| setting\_image | *varchar(255)* | *-* |
| setting\_favicon | *varchar(255)* | *-* |
| setting\_phone | *varchar(255)* | *-* |
| setting\_created\_at | *timestamp* | *-* |
| setting\_updated\_at | *timestamp* | *-* |

### Perancangan User Interface / *Mock-up* aplikasi

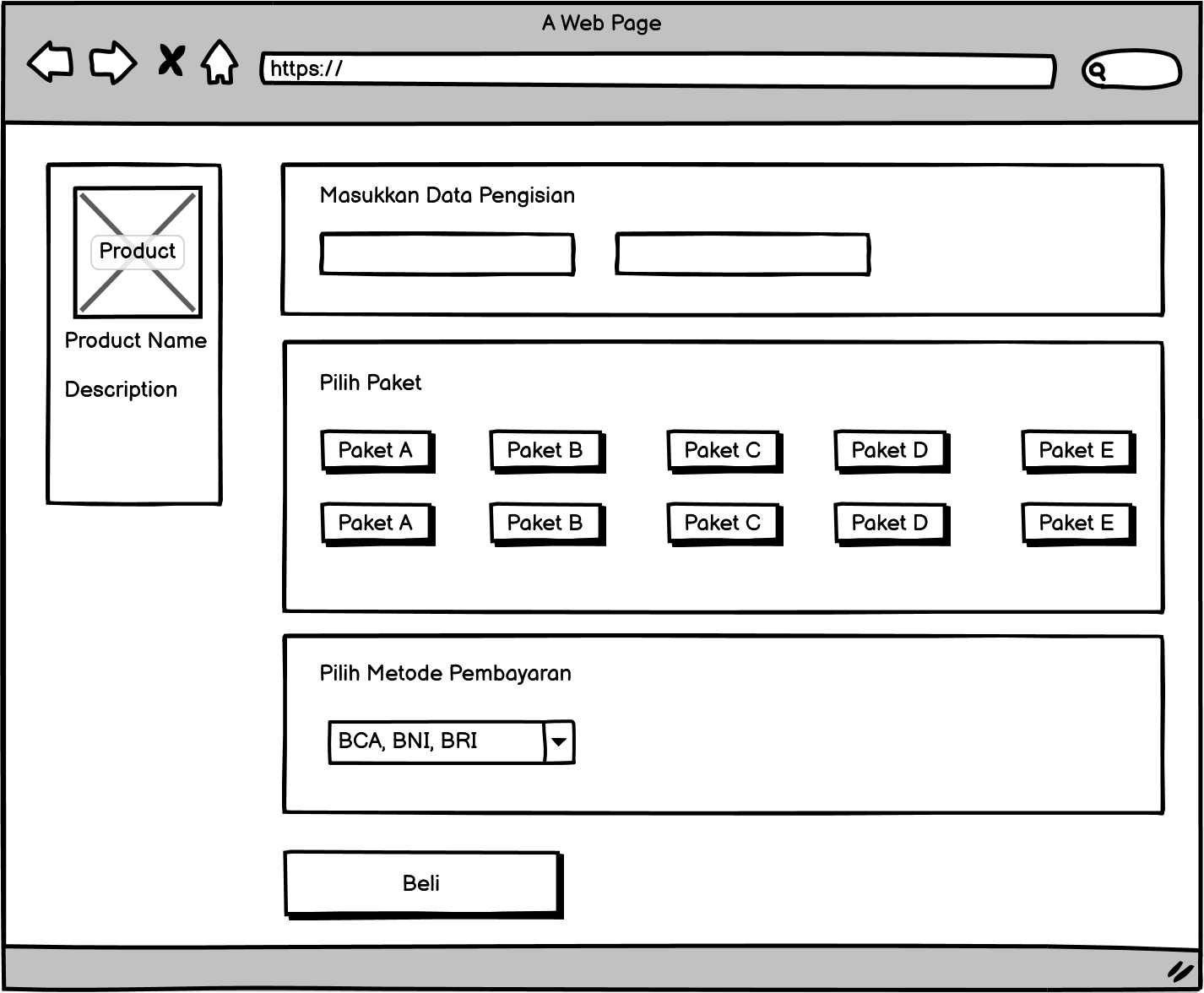
Berikut rancangan user interface yang akan diimplementasikan terhadap program sistem penjualan voucher game ini.



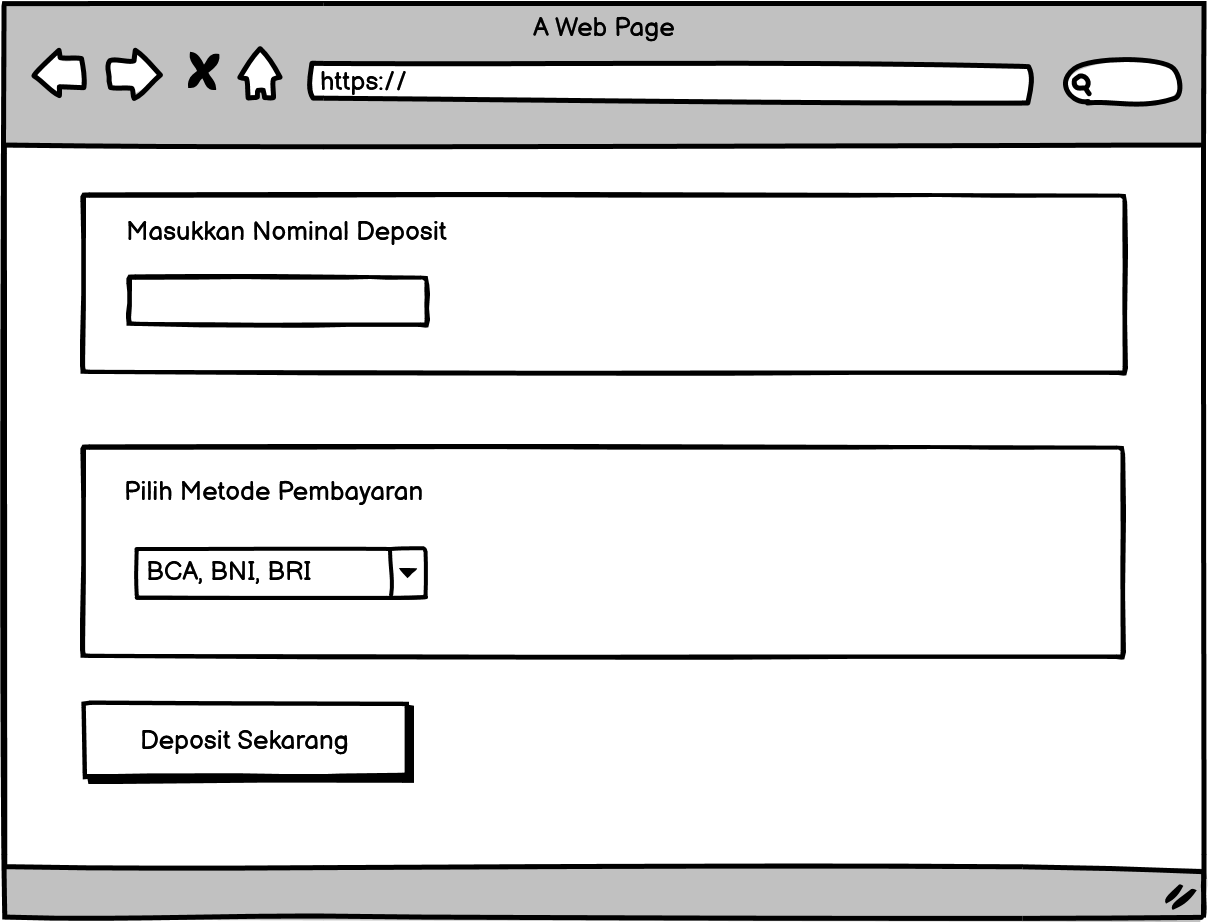
**Gambar 3.24** Halaman Login



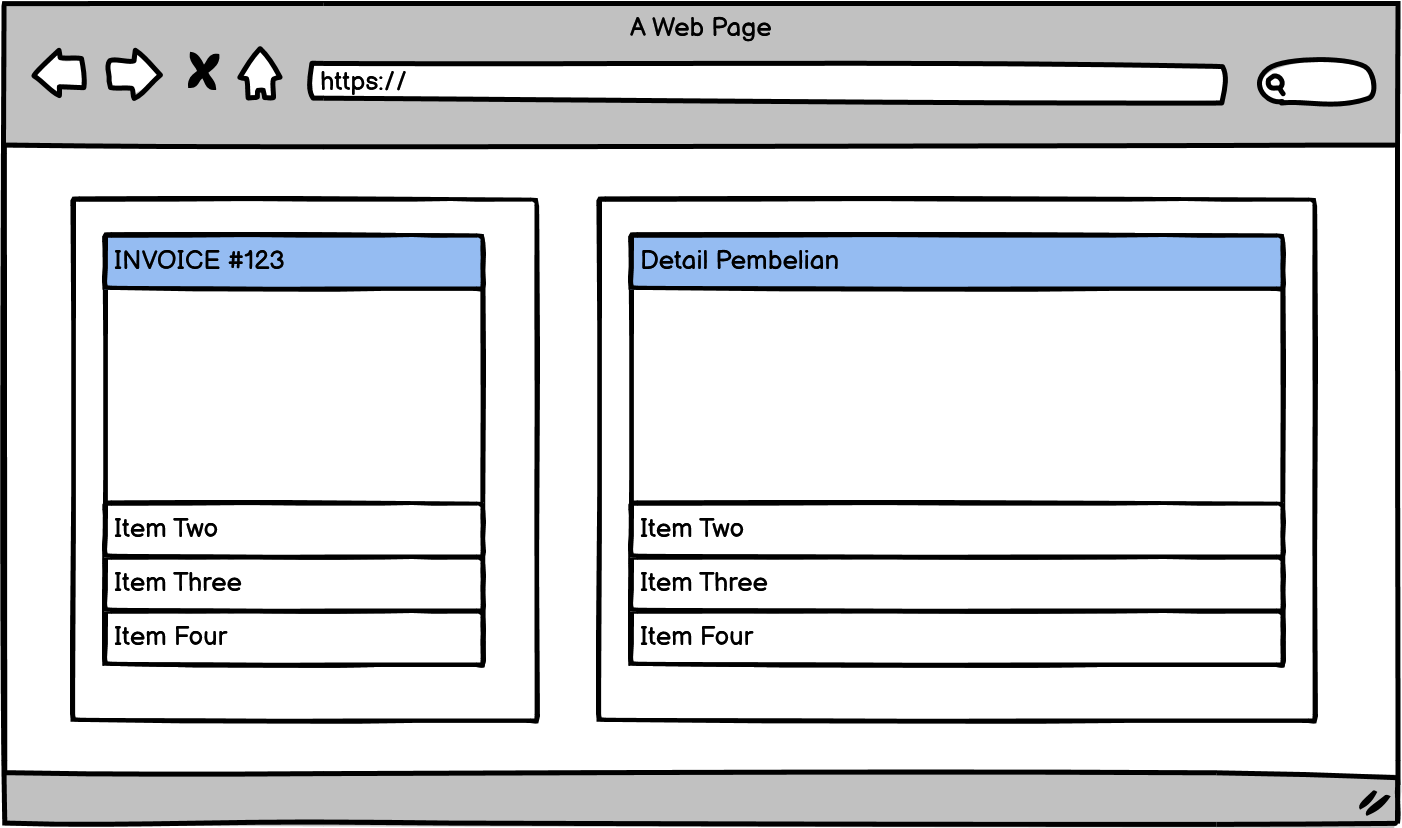
**Gambar 3.25** Halaman Homepage



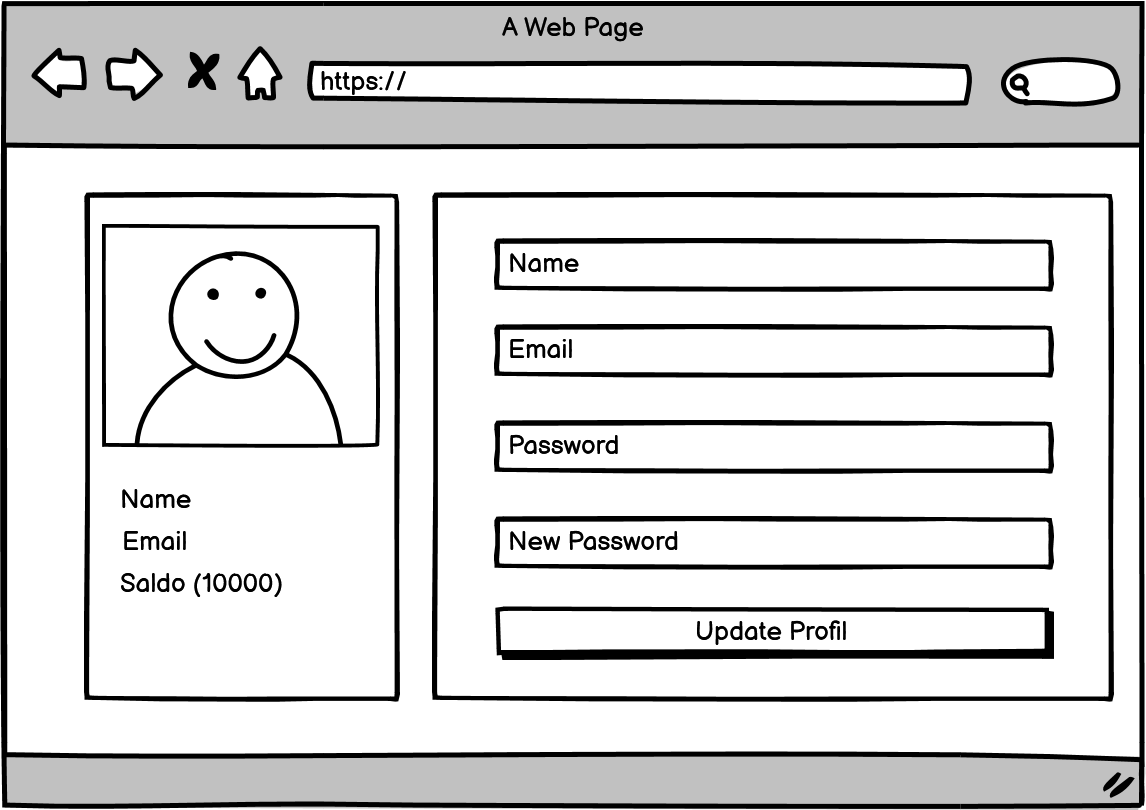
**Gambar 3.26** Halaman Detail Produk

**

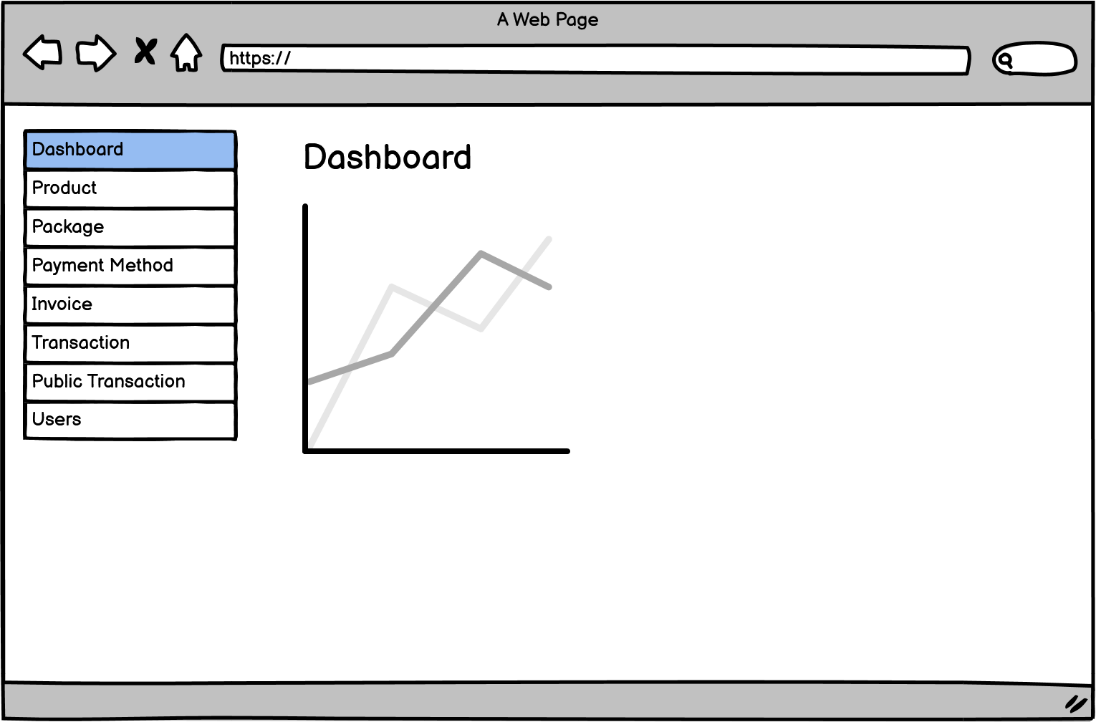
**Gambar 3.27** Halaman Deposit

**

**Gambar 3.28** Halaman Detail *Invoice*

**

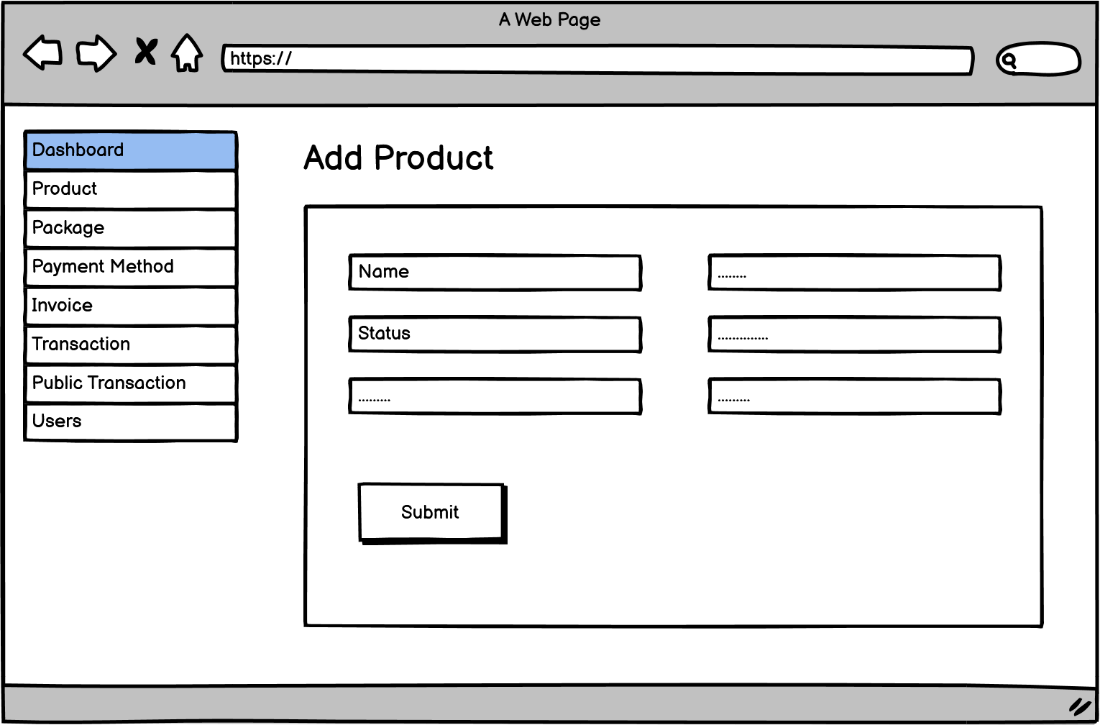
**Gambar 3.29** Halaman Profile



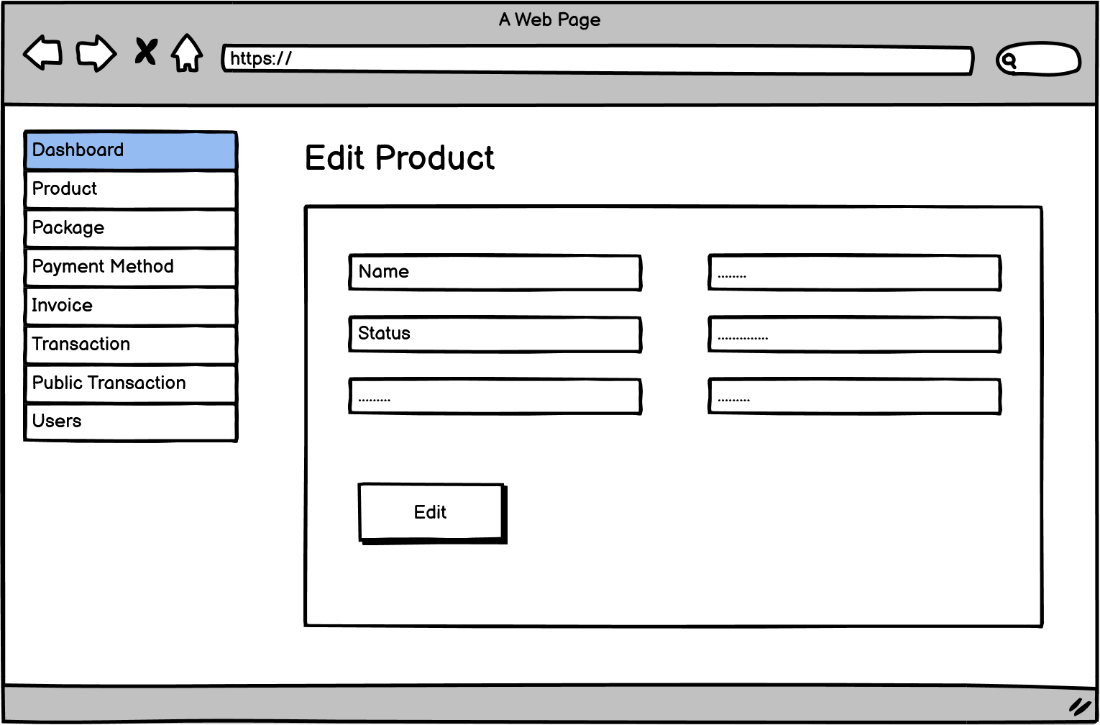
**Gambar 3.30** Halaman *Dashboard* Admin



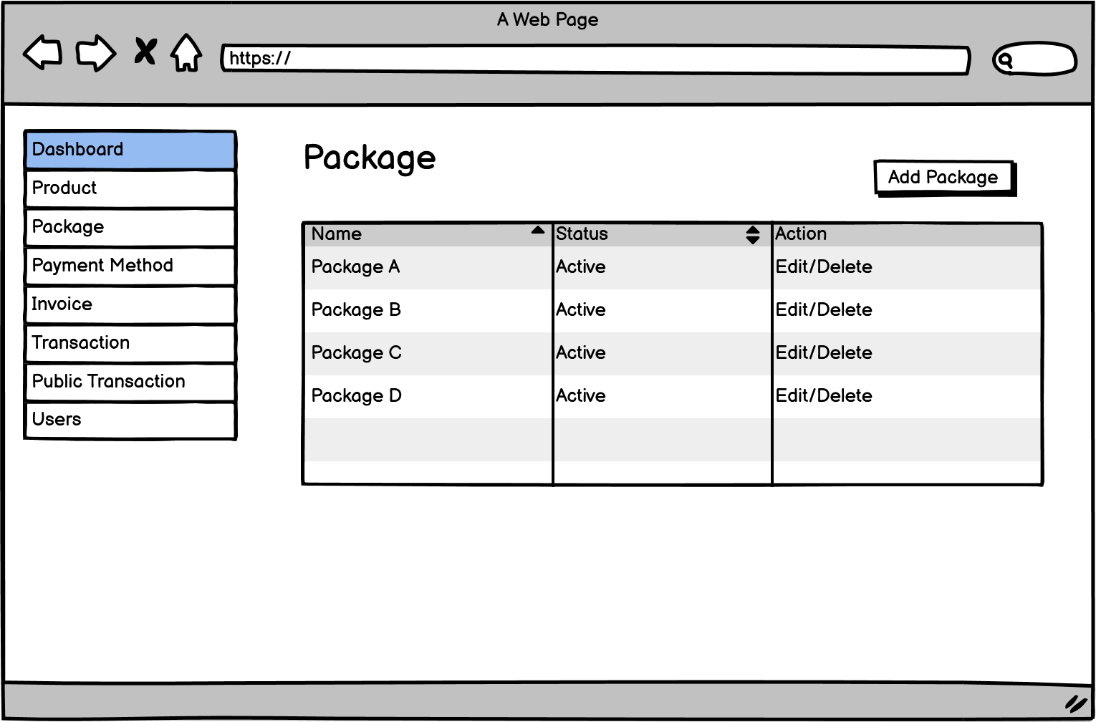
**Gambar 3.31** Halaman Manajemen Produk



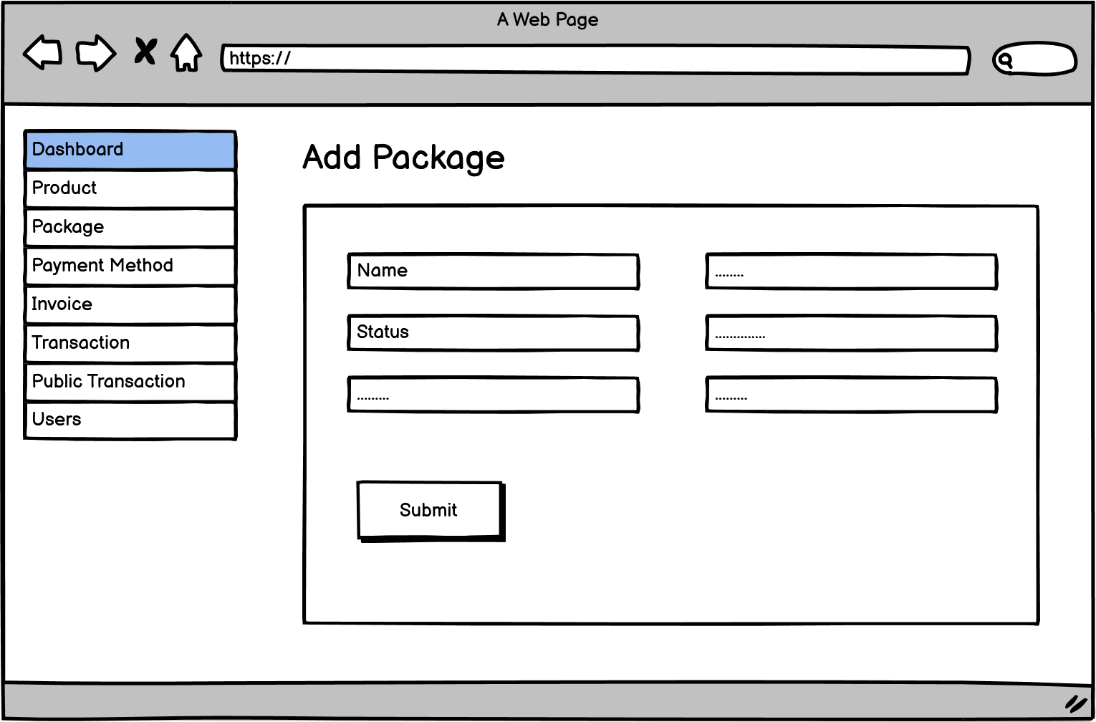
**Gambar 3.32** Halaman Tambah Produk



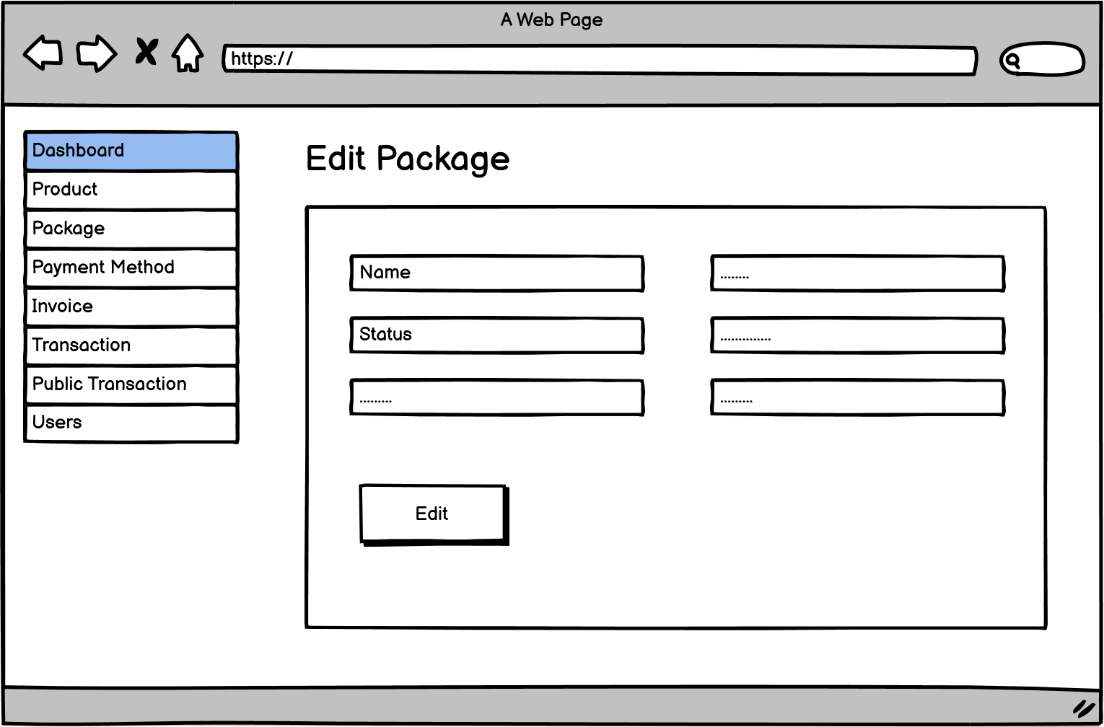
**Gambar 3.33** Halaman Edit Produk



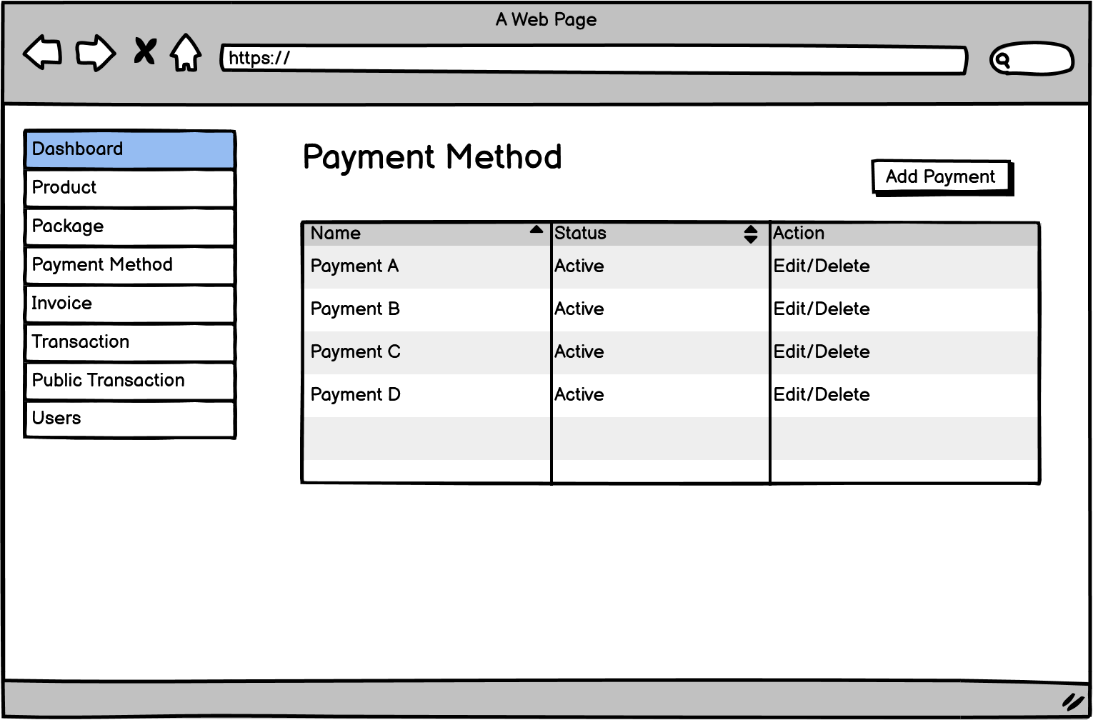
**Gambar 3.34** Halaman Manajemen Paket



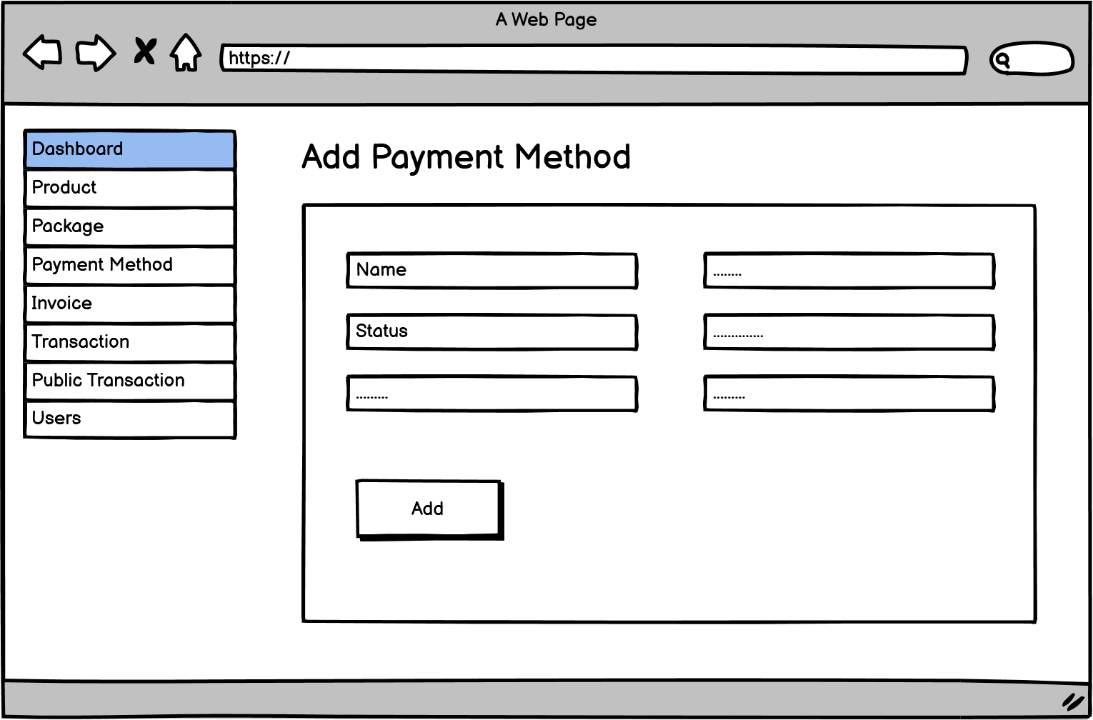
**Gambar 3.35** Halaman Tambah Paket

**

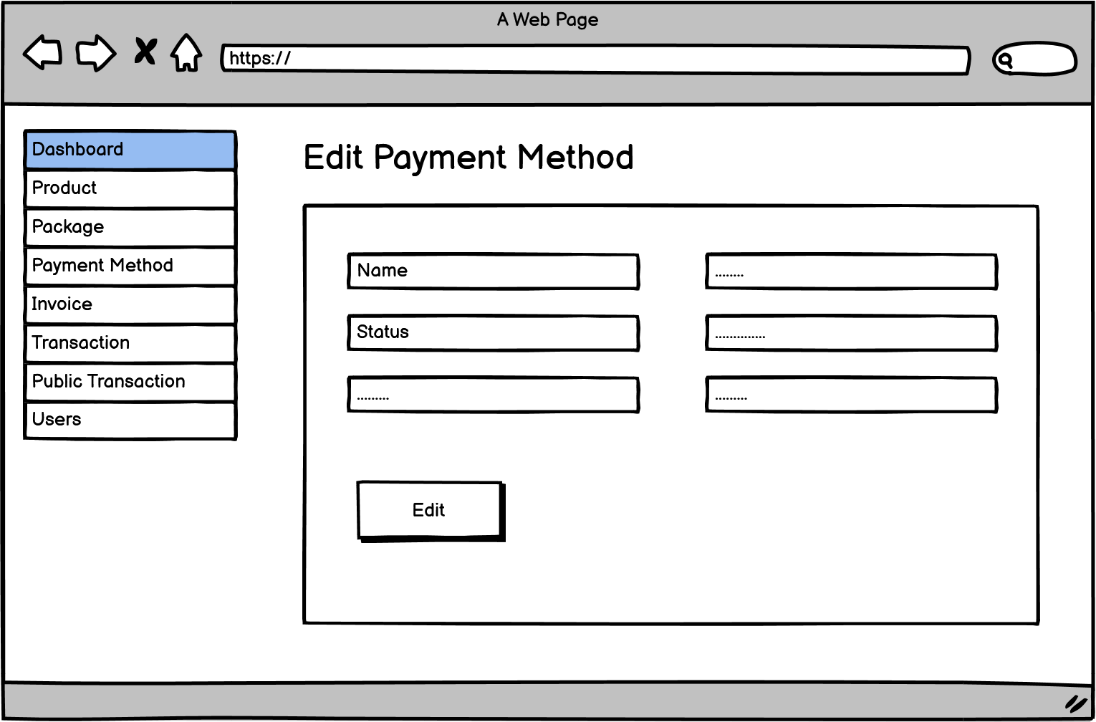
**Gambar 3.36** Halaman Edit Paket

**

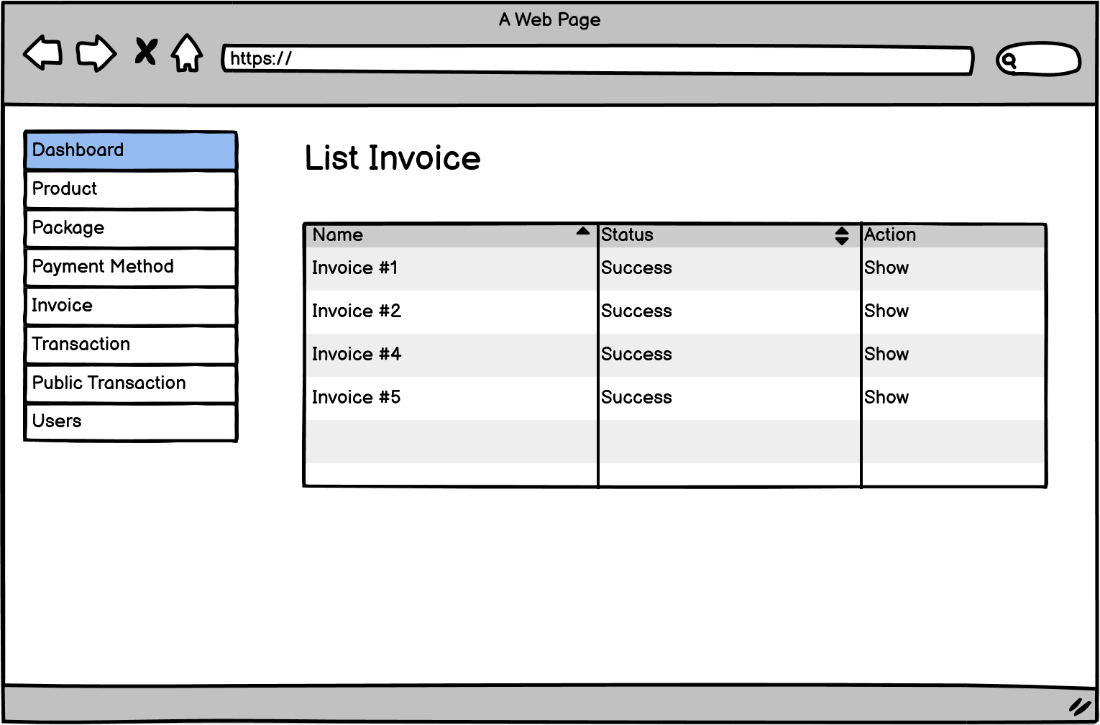
**Gambar 3.37** Halaman Manajemen Metode Pembayaran



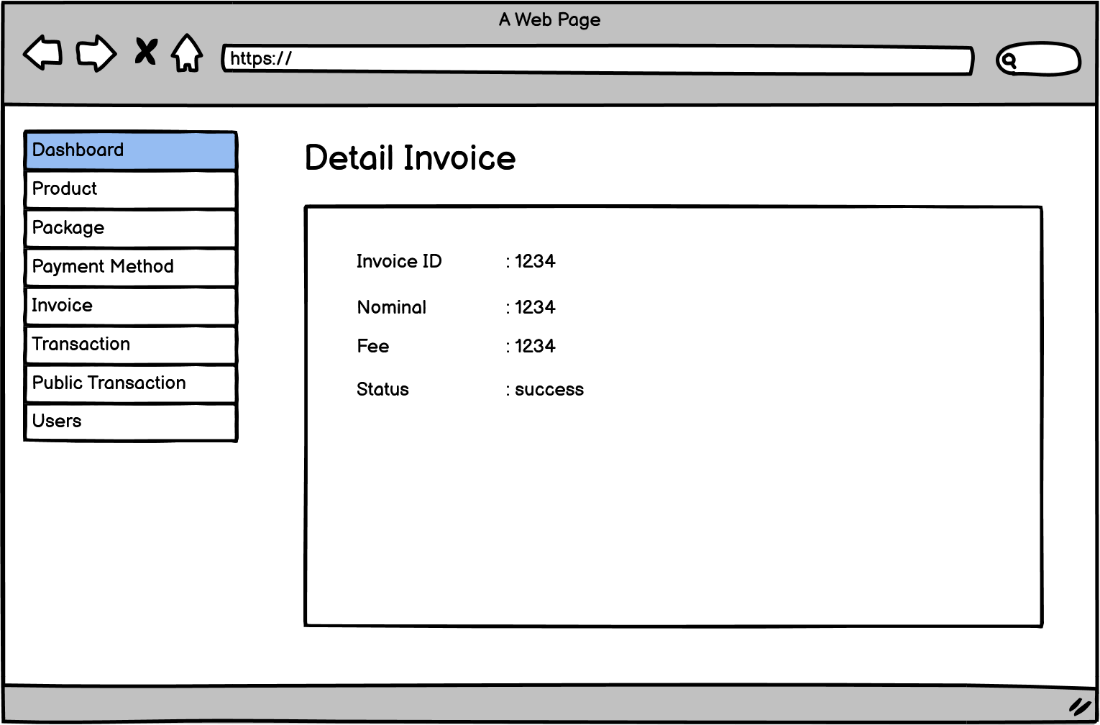
**Gambar 3.38** Halaman Tambah Metode Pembayaran



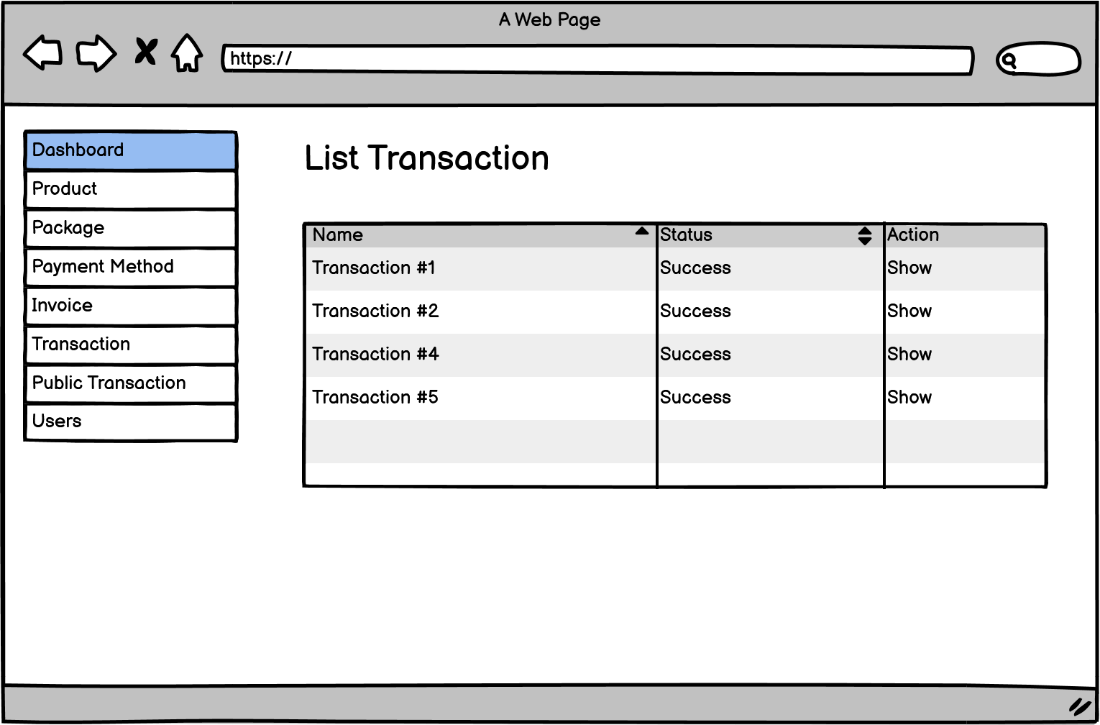
**Gambar 3.39** Halaman Edit Metode Pembayaran



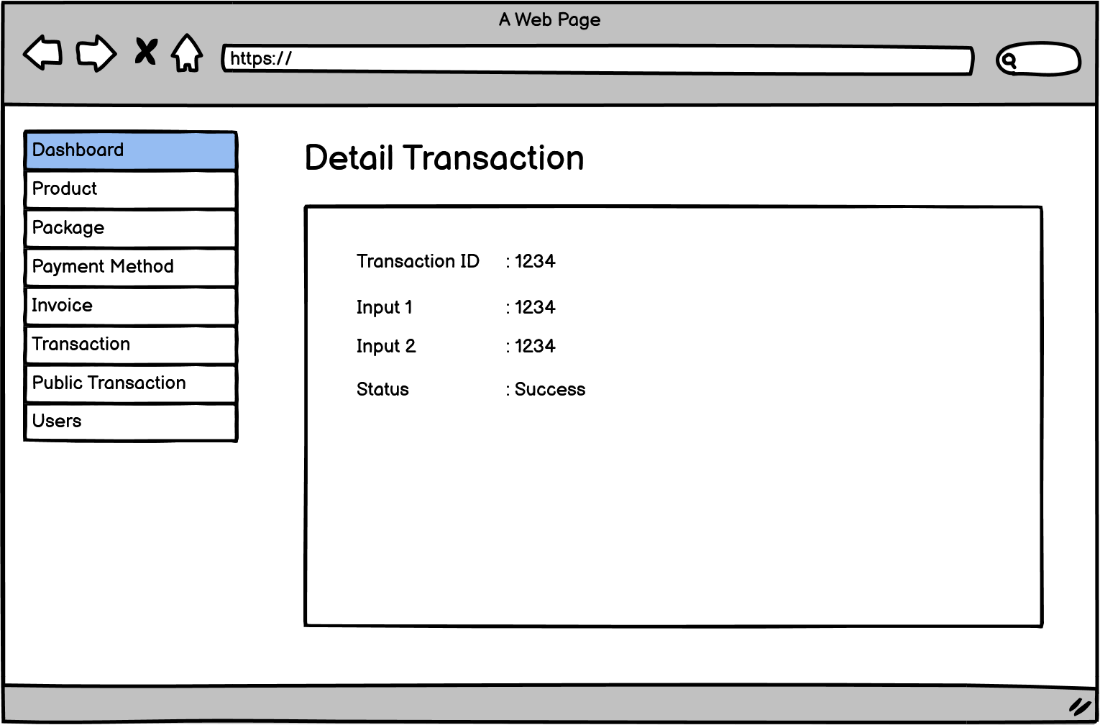
**Gambar 3.40** Halaman Daftar *Invoice*

**

**Gambar 3.41** Halaman Detail *Invoice*

**

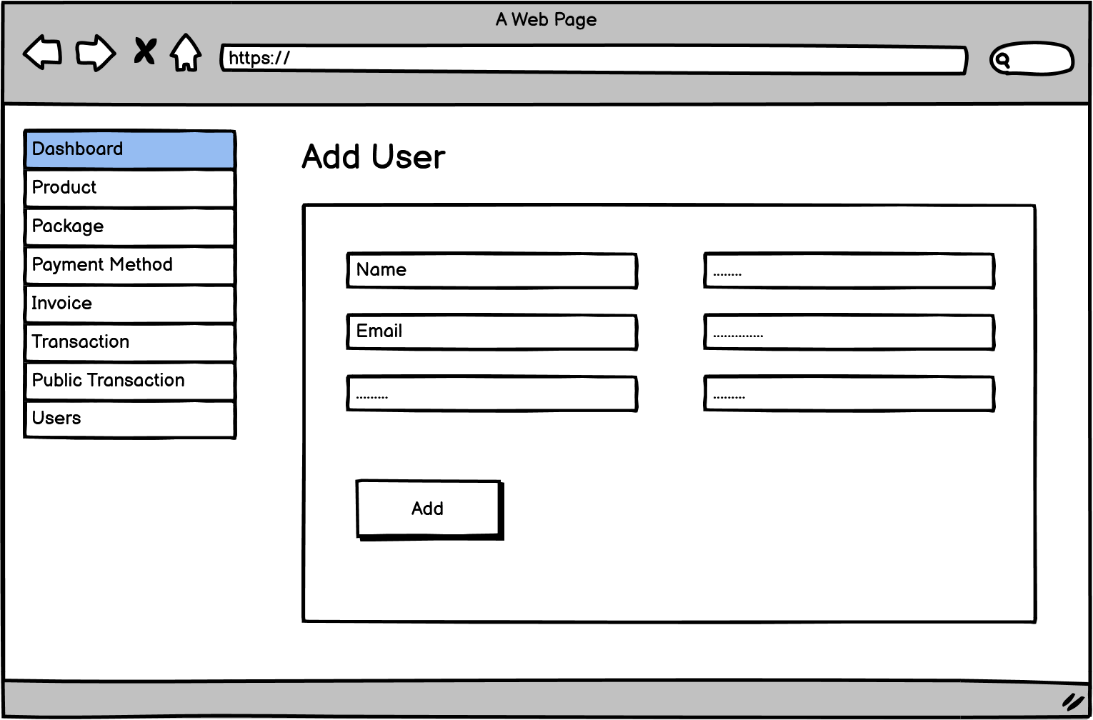
**Gambar 3.42** Halaman Daftar Transaksi

**

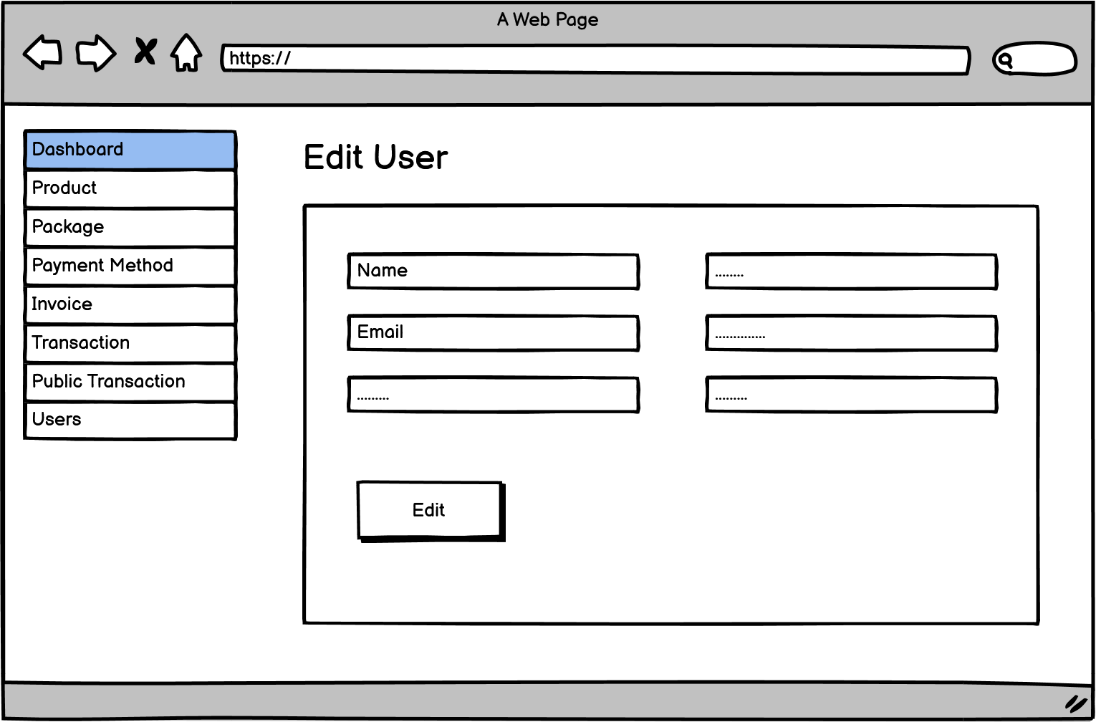
**Gambar 3.43** Halaman Detail Transaksi

**

**Gambar 3.44** Halaman Daftar Pengguna



**Gambar 3.45** Halaman Tambah Pengguna Baru



**Gambar 3.46** Halaman Edit Data Pengguna

## Rancangan Pengujian

Pada penelitian ini, metode pengujian yang digunakan oleh penulis untuk melakukan analisis adalah *blackbox testing*. *Blacbox testing* atau dapat disebut pengujian fungsional, merupakan metode pengujian yang dilakukan hanya untuk menganalisis hasil dari pengujian melalui data uji yang telah didapatkan tanpa harus mengetahui struktur kode dari perangkat lunak yang digunakan ataupun memiliki pengetahuan atas bagaimana melakukannya bedasarkan algoritma dari perangkat lunak tersebut.