

DPPL-01

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

LelangKu

untuk:

Penjual dan pembeli barang lelang

Dipersiapkan oleh:

Kelompok#1 IF-44-11

Farhan Tirta Kesumah (1301204108)
Syamaidzar Dwi Novtiar (1301204273)
Galih Atha Faye Khansa (1301204028)
Zalfa Rania (1301204098)

Program Studi Teknik Informatika/Sistem dan Teknologi Informasi
Fakultas Informatika
Jl. Telekomunikasi 1, Dayeuhkolot Bandung

Telkom University	Prodi S1- Teknik Informatika Universitas Telkom	Nomor Dokumen	Halaman
		DPPL-01 <xx:no grp>	67
		Revisi	<nomor revisi>

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	
G	

INDEX TGL	-	A	B	C	D	E	F	G
Ditulis oleh								
Diperiksa oleh								
Disetujui oleh								

Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

Daftar Isi

1. Pendahuluan	5
1.1 Tujuan Penulisan Dokumen	6
1.2 Lingkup Masalah	6
1.3 Definisi dan Istilah	6
1.4 Aturan Penamaan dan Penomoran	6
1.5 Referensi	7
1.6 Ikhtisar Dokumen	7
2. Deskripsi Perancangan Global	7
2.1 Rancangan Lingkungan Implementasi	8
2.2 Deskripsi Arsitektural	8
2.3 Deskripsi Komponen	8
3. Perancangan Rinci	9
3.1 Realisasi Use Case	9
3.1.1 Use Case Top Up Deposit	10
3.2.1 Sequence Diagram Top Up Deposit	34
3.2.10 Sequence DiagramEdit Profile	38
3.2.11 Sequence Diagram Bank	38
3.2.12 Sequence Diagram Expedition	39
3.3 Diagram Kelas	39
3.4 Perancangan Detail Kelas	40
3.4.1 Kelas User	40
3.4.1 Kelas Admin	40
3.4.2 Kelas Category	40
3.4.3 Kelas Item	41
3.4.4 Kelas Transaction	41
3.4.5 Kelas Buyer	41
3.4.6 Kelas Seller	41
3.4.7 Kelas Shipment	42
3.4.8 Kelas Expedition	42
3.5 Diagram Kelas Keseluruhan	42
3.6 Algoritma/Query	43
3.7 Diagram Statechart	56
3.8 Perancangan Antarmuka	58
3.9 Perancangan Representasi Persistensi Kelas	66
4. Matriks Kerunutan	66

1. Pendahuluan

1.1 Tujuan Penulisan Dokumen

Tujuan pembuatan DPPL ini ditujukan untuk memenuhi tugas besar mata kuliah RPL: Implementasi dan Pengujian PL dan dokumentasi pengembangan LelangKu hingga tahap *testing*. Selain itu penulisan DPPL ini bertujuan sebagai acuan dalam pengembangan LelangKu.

1.2 Lingkup Masalah

Perangkat lunak LelangKu yang dikembangkan akan dipergunakan sebagai *platform* lelang. Perangkat lunak ini mampu menjembatani pelelang dan pembeli secara *online*, transaksi yang aman dan terpercaya. Dimana *seller* dapat melelang barang yang diinginkan, begitu juga *buyer* yang dapat memilih barang yang diinginkan dan mengikuti lelang.

1.3 Definisi dan Istilah

Table 1 Daftar definisi dan Akronim

Kata Kunci atau frase	Definisi dan atau akronim
DPPL	Dokumen Deskripsi Perancangan Perangkat Lunak yang bertujuan menjabarkan secara terperinci mengenai perancangan perangkat lunak yang akan dibangun
SKPL	Spesifikasi Kebutuhan Perangkat Lunak Dokumen hasil analisis yang berisi spesifikasi kebutuhan user
Bug	Suatu eror yang menyebabkan aplikasi atau software dalam kondisi tidak dapat dijalankan sebagaimana mestinya.
Bidding	Aktivitas dalam lelang dimana para peserta pelelangan memberikan penawaran tertinggi terhadap suatu objek lelang atau barang lelang yang sedang tampil.
Hack	Tindakan peretasan aplikasi yang mengakibatkan suatu aplikasi menjadi error dan dapat memungkinkan terjadinya kebocoran data pada aplikasi.
Back-up	Proses membuat data cadangan dengan cara menyalin atau membuat arsip data komputer sehingga data tersebut dapat digunakan kembali apabila terjadi kerusakan atau kehilangan.

1.4 Aturan Penamaan dan Penomoran

Pada penulisan dokumen DPPL ini menggunakan berbagai macam aturan penamaan dan penomoran yang berbeda - beda di beberapa bagian. Aturan penamaan dan penomoran yang digunakan berdasarkan hal atau bagian seperti yang tercantum pada tabel berikut :

Hal/Bagian	Aturan Penamaan dan Penomoran
Dokumen Deskripsi Perancangan Perangkat Lunak	DPPL-01
Matriks Kerunutan	FR-xx
Nomor untuk BAB	1, 2, 3, 4
Nomor untuk subbab	1., 2., 3., 4., dst

Nomor untuk anak subbab pertama	1.1., 2.1., 3.1., 4.1., dst
---------------------------------	-----------------------------

1.5 Referensi

1. Spesifikasi Kebutuhan Perangkat Lunak (SKPL) aplikasi Lelangku.

1.6 Ikhtisar Dokumen

Produk yang dirancang merupakan *platform* lelang yang berbasis *website*. perangkat lunak ini dapat mewadahi pelelang yang hendak melelangkan barangnya, membeli barang, melihat saldo, dan menarik uang. Perangkat lunak ini terhubung ke sistem (server pusat). Semua data user disimpan di basis data Lelangku dan hanya administrator di server yang dapat mengakses basis data tersebut. perangkat lunak Lelangku hanya berfungsi sebagai aplikasi *client* yang memfasilitasi kegiatan lelang.

Secara umum aplikasi lelangku berfungsi sebagai media perantara antara seller dan buyer dari suatu kegiatan lelang. Aktivitas ini hanya dapat dilakukan jika seller sudah selesai memposting barang yang hendak dijualnya di aplikasi lelangku. Aplikasi Lelangku juga mempermudah buyer dalam proses transaksi pembelian barang setelah dinyatakan dan dikonfirmasi telah memenangkan lelang yang diadakan. Maintenance aplikasi Lelangku dilakukan oleh administrator, misalnya memperbaiki bug yang muncul pada aplikasi. Untuk keperluan ini server dari aplikasi harus dinonaktifkan terlebih dahulu. Kemudian, setelah bug selesai diperbaiki server dapat diaktifkan kembali dan aplikasi dapat berjalan sebagaimana mestinya. Selain menangani bug pada aplikasi, administrator juga bertugas memperbarui tampilan dan visual aplikasi. Hal ini dilakukan guna mempermudah dan meningkatkan daya tarik dari pengguna aplikasi lelangku.

Dokumen DPPL ini berisi deskripsi spesifikasi/perancangan perangkat lunak berdasarkan dokumen SKPL yang akan diimplementasikan. Pada dokumen DPPL ini akan dijelaskan rincian dari rancangan perangkat lunak sehingga dapat diimplementasikan. Dokumen ini secara garis besar terdiri dari empat bab dengan perincian sebagai berikut:

A. Pendahuluan

Pendahuluan berisi penjelasan tentang dokumen DPPL yang mencakup tujuan pembuatan dokumen ini, lingkup masalah yang diselesaikan oleh perangkat lunak yang dikembangkan, definisi, aturan penamaan dan penomoran, referensi, dan ikhtisar dokumen.

B. Deskripsi Perancangan Global

Deskripsi perancangan global berisi tentang rancangan dari perangkat lunak yang akan dibangun meliputi, rancangan lingkungan implementasi, deskripsi arsitektural, dan deskripsi komponen.

C. Perancangan Rinci

Perancangan rinci pada dokumen ini berisi tentang realisasi use case, perancangan detil kelas, deskripsi diagram kelas, algoritma/query, diagram statechart, perancangan antarmuka, dan perancangan representasi persistensi kelas.

D. Matriks Keterunutan

Matriks keterunutan berisi tentang hal fungsional yang terdapat pada dokumen DPPL.

Prodi Teknik Informatika Tel-U	DPPL-01	Halaman 7 dari 67
Template dokumen ini dan informasi yang dimilikinya adalah milik ProdiTeknik Informatika Tel-U dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh ProdiTeknik Informatika Tel-U.		

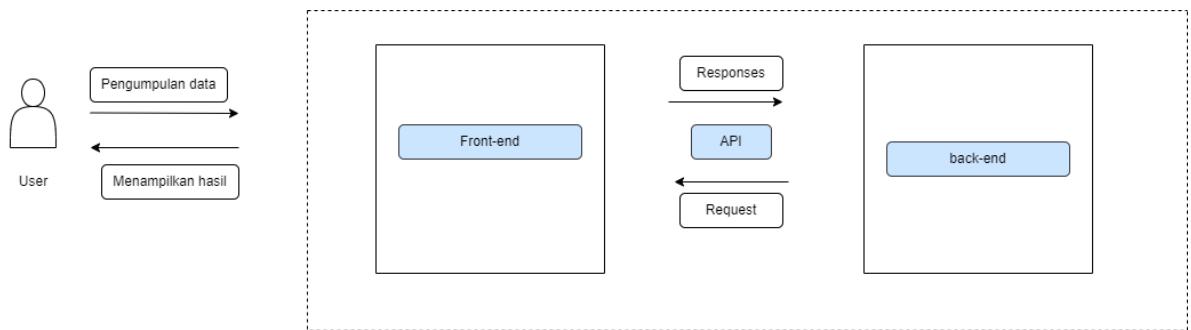
2. Deskripsi Perancangan Global

Deskripsi perancangan global meliputi penjelasan tentang rancangan lingkungan implementasi, deskripsi arsitektural, dan deskripsi komponen yang membangun perangkat lunak.

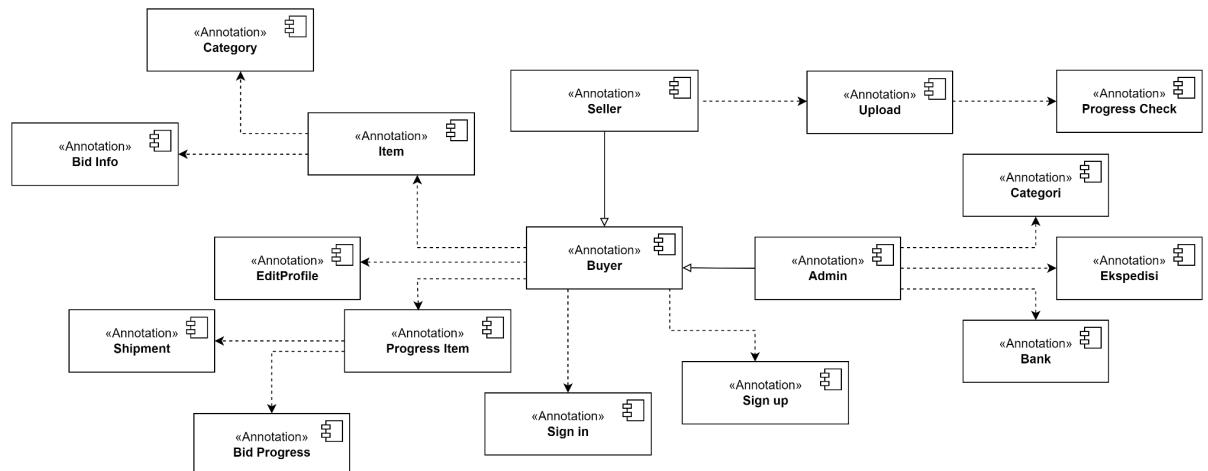
2.1 Rancangan Lingkungan Implementasi

No	Nama Komponen	Keterangan
1	Bahasa Pemrograman	HTML, CSS, Java Script
2	DBMS	MongoDB
3	Web Framework	ExpressJS NextJS
4	Development Tools	GitHub

2.2 Deskripsi Arsitektural



2.3 Deskripsi Komponen



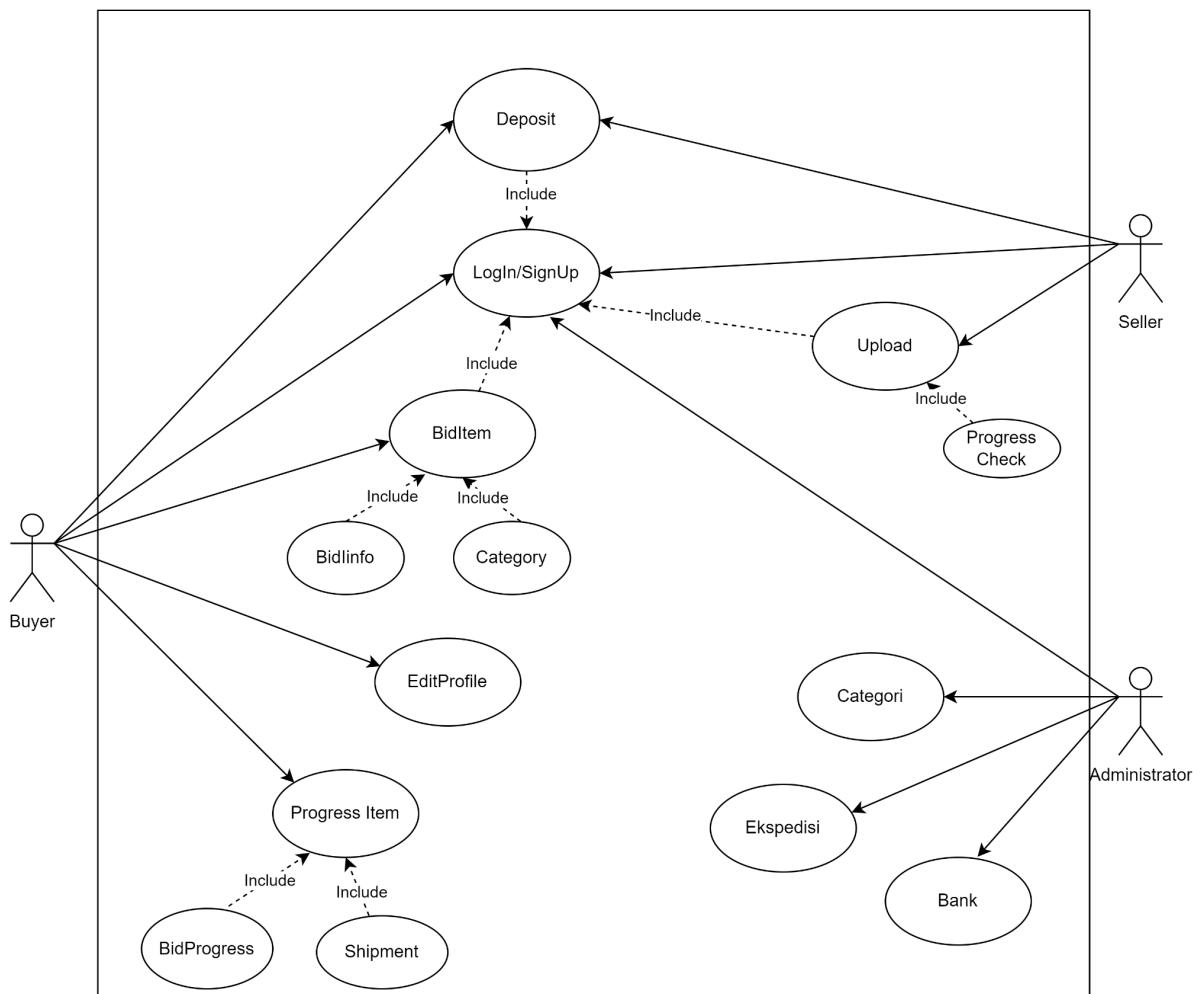
No	Nama Komponen	Keterangan
1	Bahasa Pemrograman	HTML, CSS, Java Script
2	DBMS	MongoDB
3	Web Framework	ExpressJS NextJS
4	Development Tools	GitHub

3. Perancangan Rinci

Dalam bab perancangan rinci akan dijelaskan mengenai realisasi use case, perancangan detail kelas, diagram kelas keseluruhan, algoritma/query, diagram statechart, antarmuka, dan perancangan representasi persistensi kelas.

3.1 Realisasi Use Case

Sub Bab ini menjelaskan tentang realisasi semua *use case* yang telah dirancang pada dokumen spesifikasi kebutuhan perangkat lunak.



No	Nama Use Case	Deskripsi Use Case

1	<i>Top up deposit</i>	Buyer dapat menambahkan saldo yang dapat digunakan sebagai alat pembayaran ketika memenangkan pelelangan
2	<i>Login</i>	Buyer memasukkan <i>username</i> dan <i>password</i> yang telah dibuat agar dapat mengakses layanan aplikasi LelangKu
3	<i>SignUp</i>	Buyer yang pertama kali mengakses aplikasi LelangKu harus melakukan pendaftaran terlebih dahulu untuk mendapatkan akses aplikasi
4	<i>Upload</i>	Seller menambahkan barang yang akan dijual pada aplikasi LelangKu
5	<i>Bid Item</i>	Buyer memberikan penawaran harga dan bersaing dengan buyer lain hingga didapat penawaran tertinggi yang akan memenangkan pelelangan
6	<i>Bid Info</i>	Mengetahui informasi <i>bid</i> dari suatu barang lelang
7	<i>kategori</i>	Buyer dapat melihat item yang dibutuhkan dalam suatu kategori
8	<i>Progres Check</i>	Seller memeriksa yang sudah selesai melakukan lelang dan yang masih menjalankan lelang
9	<i>On Progres Item</i>	Buyer dapat memilih item yang di inginkan
10	<i>Bid Progres</i>	Buyer melakukan bid untuk mendapatkan barang lelang
11	<i>Shipment</i>	Proses pengiriman barang yang sudah di <i>bid</i> oleh <i>buyer</i>
13	<i>Update</i>	Administrator melakukan <i>update</i> ke <i>buyer</i> dan <i>user</i>
14	<i>Edit Profile</i>	User dapat mengedit informasi mulai dari nama, Nomor HP, Provinsi, Kota, Kecamatan, Kelurahan, Alamat, Kode Pos, dan photo profile
15	<i>Bank</i>	Bank berguna sebagai fitur pendukung dalam melakukan operasional transaksi
16	<i>Ekspedisi</i>	Ekspedisi berguna sebagai fitur pendukung dalam melakukan operasional pengiriman barang.
17	<i>Maintance</i>	Memperbaiki <i>bug – bug</i> yang ditemukan dalam jangka waktu tertentu secara rutin

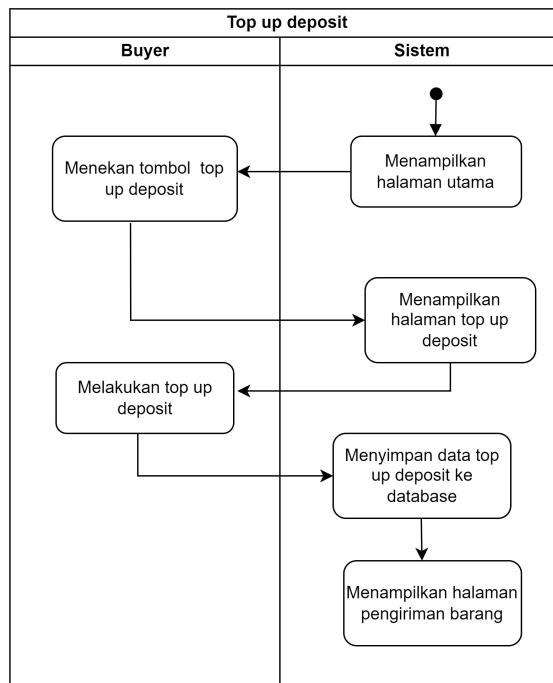
3.1.1 Use Case Top Up Deposit

Nama	<i>Top up deposit</i>
Deskripsi	<i>Buyer</i> dapat menambahkan saldo yang dapat digunakan sebagai alat pembayaran ketika memenangkan pelelangan.

Pre-kondisi	Saldo belum bertambah
Post-kondisi	Saldo sudah bertambah
Platform	<i>Website</i>

Table Typical Course of Event

Aktor	Sistem
	1. Sistem menampilkan halaman utama
2. Buyer menekan tombol melakukan <i>top up</i> deposit	
	3. Sistem menampilkan halaman <i>top up</i> deposit
4. Buyer melakukan <i>top up</i> deposit	
	5. Menyimpan data <i>top up</i> deposit ke <i>database</i>
	6. Menampilkan hasil <i>top up</i> deposit



Top Up

Mau top-up berapa?

Metode Pembayaran

BCA (11111)

BSI (222222)

Mandiri (1234543252)

BRI (1231234)

CIMB (23523532)

Bayar

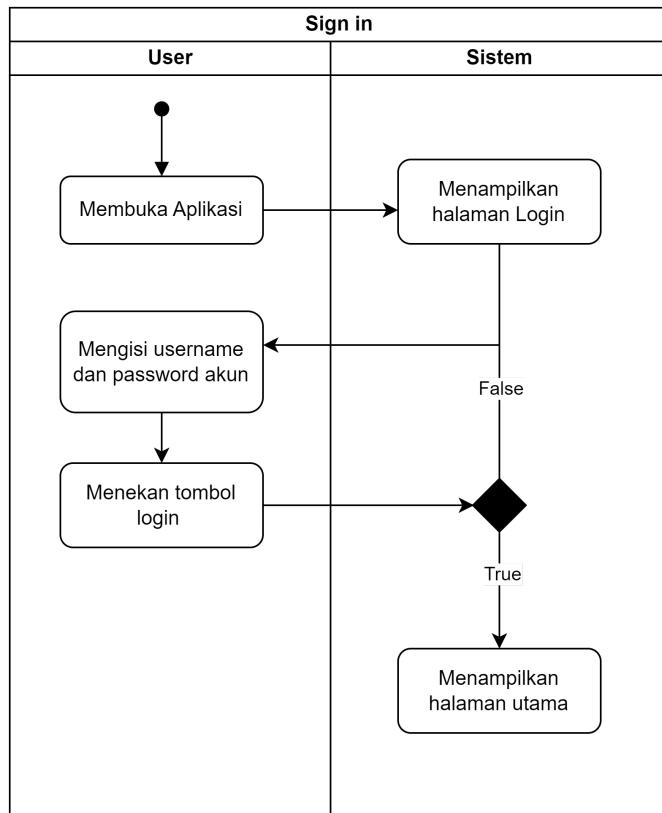
3.1.2 Use Case *Sign In*

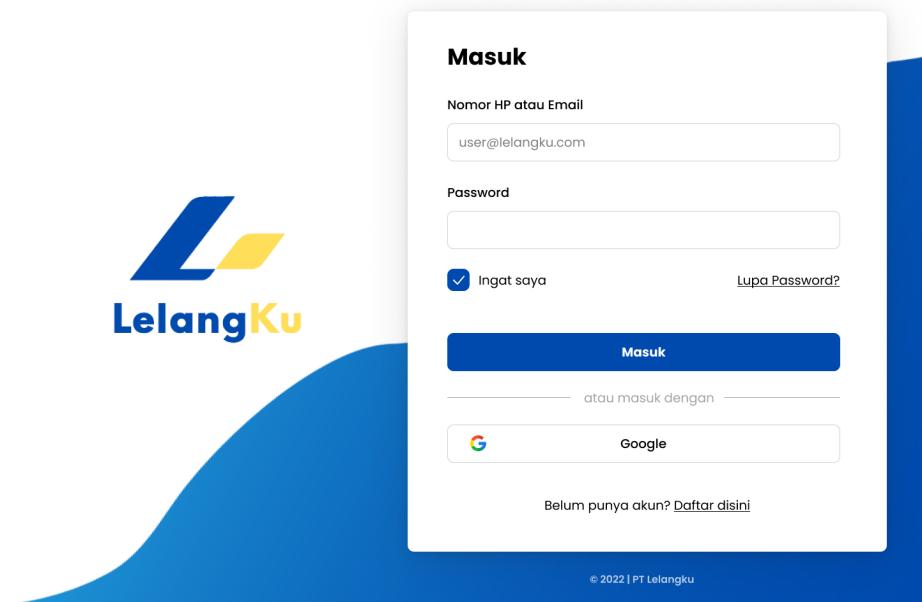
Nama	<i>Sign In</i>
Deskripsi	Buyer memasukkan <i>username</i> dan <i>password</i> yang telah dibuat agar dapat mengakses layanan aplikasi LelangKu.
Pre-kondisi	Aplikasi menampilkan <i>form Login</i> .
Post-kondisi	Aplikasi menampilkan menu utama
Platform	<i>Website</i>

Table Typical Course of Event

Aktor	Sistem
1. User membuka aplikasi	
	2. Sistem menampilkan halaman <i>Login</i>
3. User mengisi <i>email</i> dan <i>password</i>	
	4. Sistem melakukan pengecekan data. Jika data akun yang dimasukkan salah maka akan kembali ke nomor 3. jika data sudah benar maka sistem akan menampilkan

	halaman utama
5. User berhasil masuk ke halaman utama	





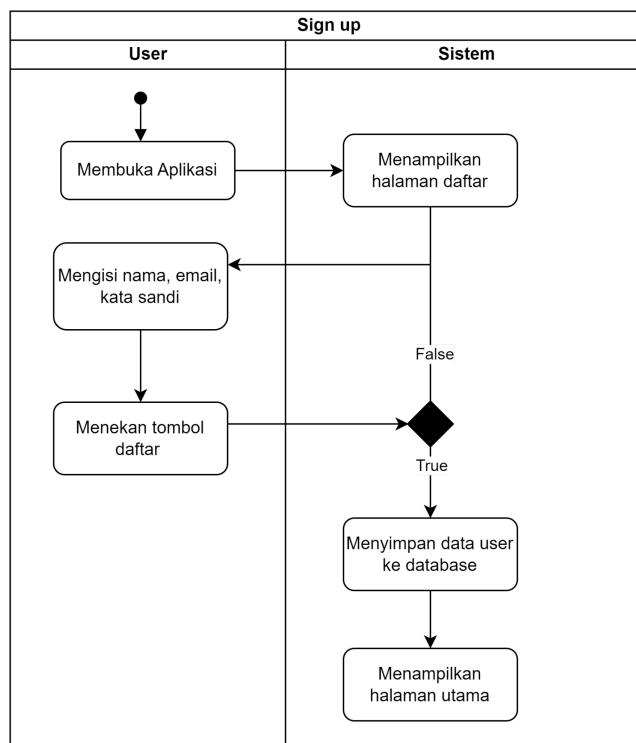
3.1.3 Use Case Sign Up

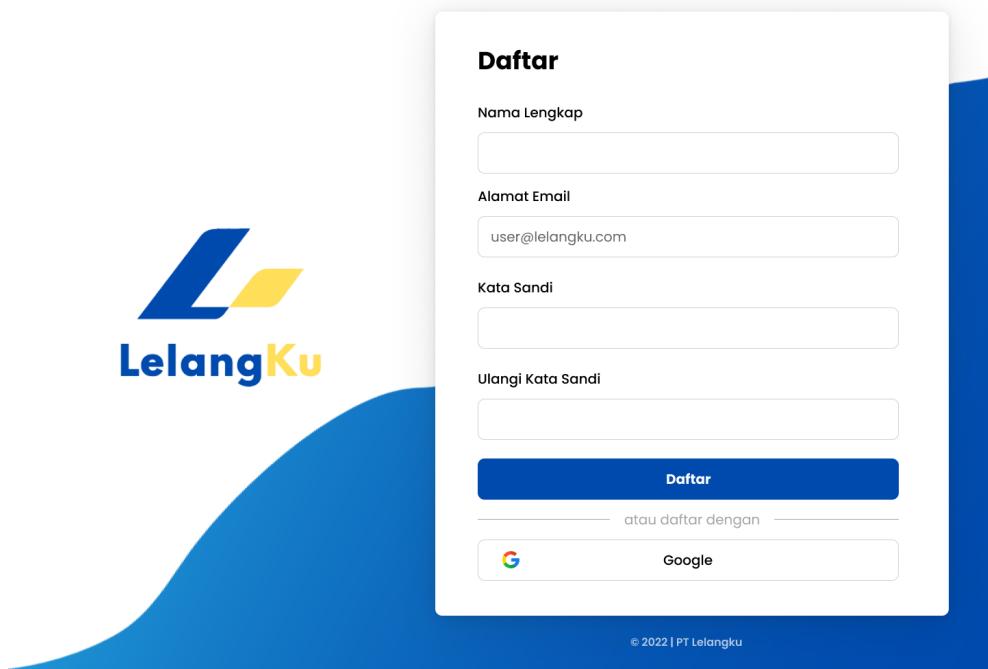
Nama	<i>SignUp</i>
Deskripsi	<i>Buyer</i> yang pertama kali mengakses aplikasi LelangKu harus melakukan pendaftaran terlebih dahulu untuk mendapatkan akses aplikasi
Pre-kondisi	Aplikasi menampilkan <i>form SignUp</i>
Post-kondisi	Aplikasi menampilkan menu utama
Platform	<i>Website</i>

Table Typical Course of Event

Aktor	Sistem
1. <i>User</i> membuka aplikasi	
	2. Sistem menampilkan halaman sign up
3. <i>user</i> mengisi nama, <i>email</i> , dan <i>password</i>	
4. <i>User</i> menekan tombol daftar	

	5. Sistem melakukan pengecekan data, jika data <i>username</i> dan <i>email</i> telah terdaftar maka kembali ke langkah 2
	6. Jika belum tersedia maka lanjut ke langkah 7
	7. Sistem merekam data daftar yang <i>diinput</i> oleh user
	8. Memproses dan menyimpan data daftar ke dalam <i>database</i>
	9. sistem menampilkan halaman <i>login</i>





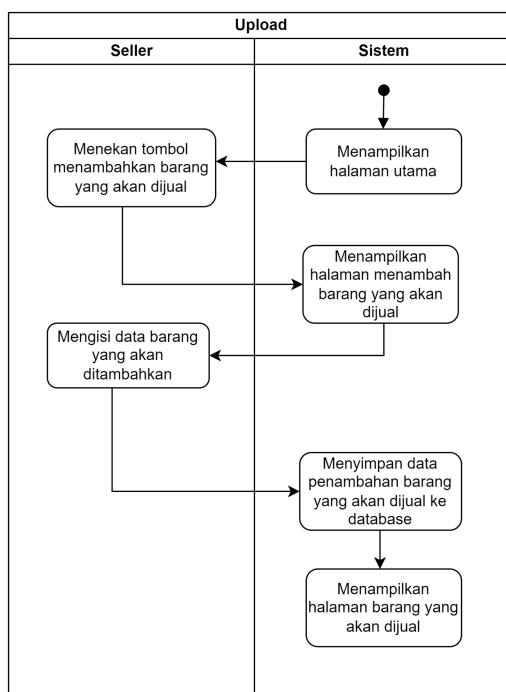
3.1.4 Use Case Upload

Nama	<i>Upload</i>
Deskripsi	<i>Seller</i> menambahkan barang yang akan dijual pada aplikasi LelangKu
Pre-kondisi	Barang tidak dapat dilihat pada aplikasi
Post-kondisi	Barang sudah muncul pada aplikasi
Platform	<i>Website</i>

Table Typical Course of Event

Aktor	Sistem
	1. Sistem menampilkan halaman utama
2. <i>seller</i> menekan tombol menambahkan barang yang akan dijual	
	3. Menampilkan halaman menambah barang yang akan

	di jual
4. Mengisi data barang yang akan ditambahkan	
	5. Menyimpan data penambahan barang yang akan di jual ke database
	6. Menampilkan halaman barang yang akan dijual



LelangKu Kategori Cari barang... Toko | test

test @ Bandung ★ 4.9/5 Rating & Ulasan

Barang yang di lelang

Basreng Rp. 10000 Baru

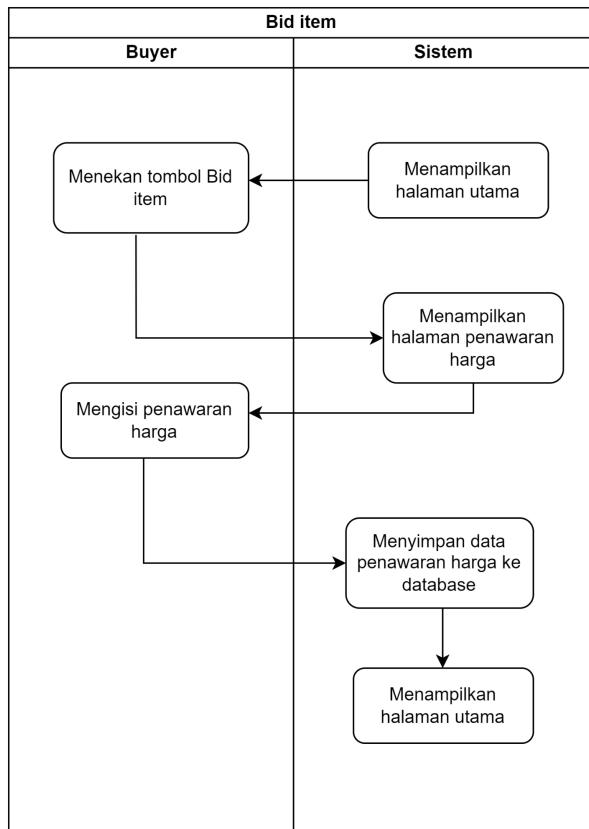
Berakhir pada

3.1.5 Use Case Bid Item

Nama	<i>Bid Item</i>
Deskripsi	<i>Buyer</i> memberikan penawaran harga dan bersaing dengan <i>buyer</i> lain sehingga didapat penawaran tertinggi yang akan memenangkan pelelangan
Pre-kondisi	Aplikasi memulai lelang
Post-kondisi	<i>Buyer</i> melakukan <i>bid</i>
Platform	<i>Website</i>

Table Typical Course of Event

Aktor	Sistem
	1. Sistem menampilkan halaman utama
2. buyer menekan tombol <i>bid item</i>	
	3. Menampilkan halaman penawaran harga
4. buyer mengisi penawaran harga	
	5. Sistem menyimpan data penawaran harga ke <i>database</i>
	6. Sistem menampilkan penawaran harga



LelangKu Kategori Cari barang... Toko | member

Kaos Polos Premium

⌚ Sisa waktu :2 Hari lagi | Banyak bid :0

Harga Terkini
Rp. 20000

Bid | + Rp. 50.000

Deskripsi Produk
Lorem ipsum is simply dummy text of the printing and typesetting industry. Lorem ipsum has been the industry's standard dummy text ever since the 1500s.

hasbulia SELLER
© bandung | ★ 4.9 / 5.0

Barang Lainnya

-
-
-
-

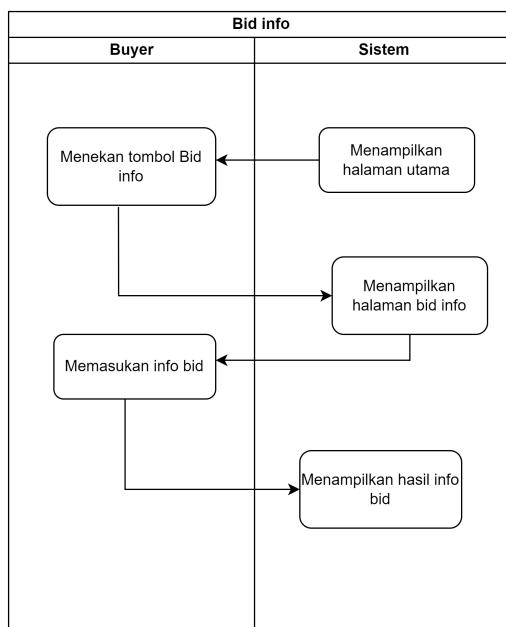
3.1.6 Use Case Bid Info

Nama	<i>Bid Info</i>
Deskripsi	Mengetahui <i>bid</i> dari suatu barang lelang

Pre-kondisi	Memasukan info <i>bid</i>
Post-kondisi	Dapat melihat info <i>bid</i>
Platform	<i>Website</i>

Table Typical Course of Event

Aktor	Sistem
	1. Sistem menampilkan halaman utama
	2. Buyer menekan tombol <i>bid info</i>
	3. Menampilkan halaman <i>bid info</i>
	4. Memasukan <i>info bid</i>
	5. Menampilkan hasil <i>info bid</i>



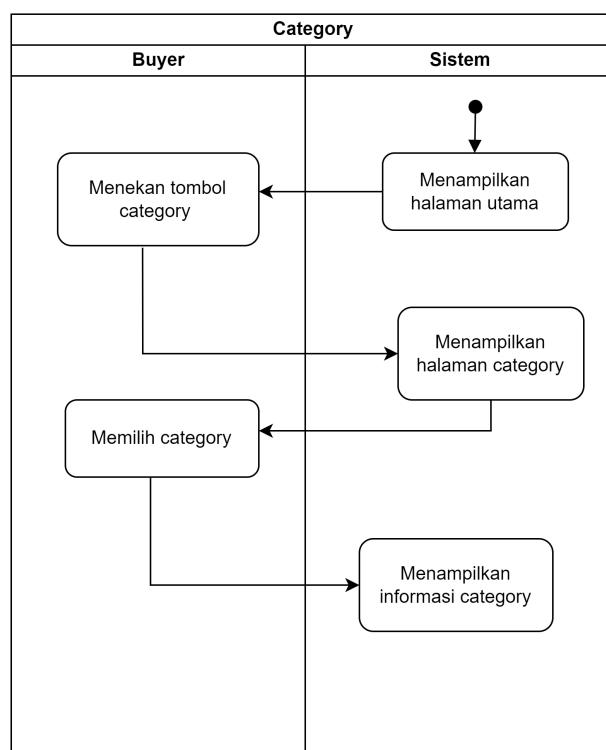
3.1.6 Use Case Category

Nama	<i>Category</i>
Deskripsi	<i>Buyer</i> dapat melihat <i>item</i> yang dibutuhkan dalam suatu kategori
Pre-kondisi	memilih kategori

Post-kondisi	Mendapatkan item yang diinginkan
Platform	<i>Website</i>

Table Typical Course of Event

Aktor	Sistem
	1. Sistem menampilkan halaman utama
2. buyer menekan tombol <i>category</i>	
	3. Sistem menampilkan halaman <i>category</i>
4. Buyer dapat memilih <i>category</i>	
	5. Menampilkan informasi <i>category</i>



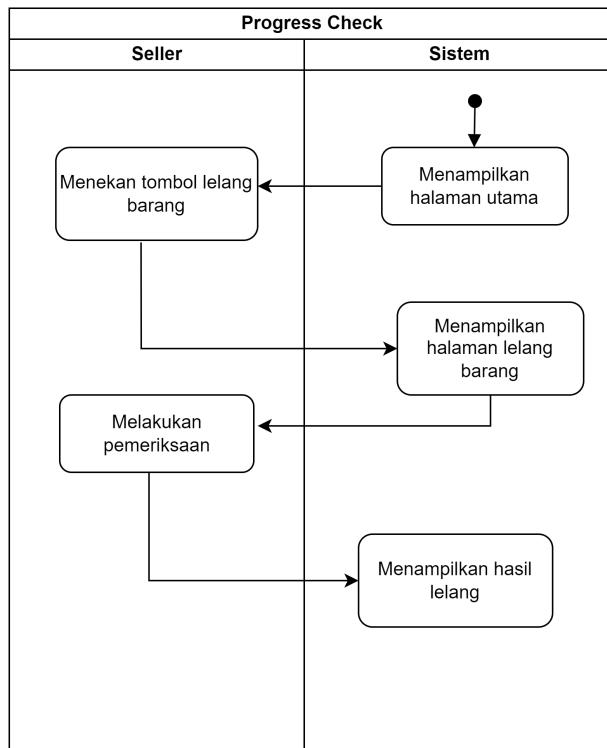
Category		
Expandable Table		
No	Nama	Action
1	Laptop	Delete
2	Fashion	Delete
3	Makanan	Delete
4	Handphone	Delete
5	Lainnya berhasil edit	Delete

3.1.7 Use Case Progress Check

Nama	Progress Check
Deskripsi	<i>Seller</i> memeriksa yang sudah selesai melakukan lelang dan yang masih menjalankan lelang
Pre-kondisi	Melakukan lelang barang
Post-kondisi	Sudah selesai di <i>upload</i>
Platform	<i>Website</i>

Table Typical Course of Event

Aktor	Sistem
	1. Sistem menampilkan halaman utama
2. <i>Seller</i> menekan tombol lelang barang	
	3. Menampilkan halaman lelang barang
4. Melakukan pemeriksaan	
	5. Menampilkan hasil lelang



3.1.8 Use Case Progress Item

Nama	<i>On Progres Item</i>
Deskripsi	<i>Buyer</i> dapat memilih <i>item</i> yang di inginkan
Pre-kondisi	<i>Buyer</i> memilih <i>item</i>
Post-kondisi	<i>Buyer</i> melakukan <i>bid</i>
Platform	<i>Website</i>

Table Typical Course of Event

Aktor	Sistem
	1. Sistem menampilkan halaman utama
2. <i>Buyer</i> menekan tombol <i>item</i>	
	3. Menampilkan halaman <i>item</i>
4. <i>Buyer</i> memilih <i>item</i>	
	5. Menampilkan halaman <i>item</i>



Kategori

[Handphone] [Handphone] [Handphone] [Handphone] [Handphone] [Handphone] [Handphone]

Sebentar lagi Lihat semua



Sebentar lagi Lihat semua



Ikuti kami



Bantuan dan Panduan

Tutorial Pengguna
Pendaftaran dan Log In



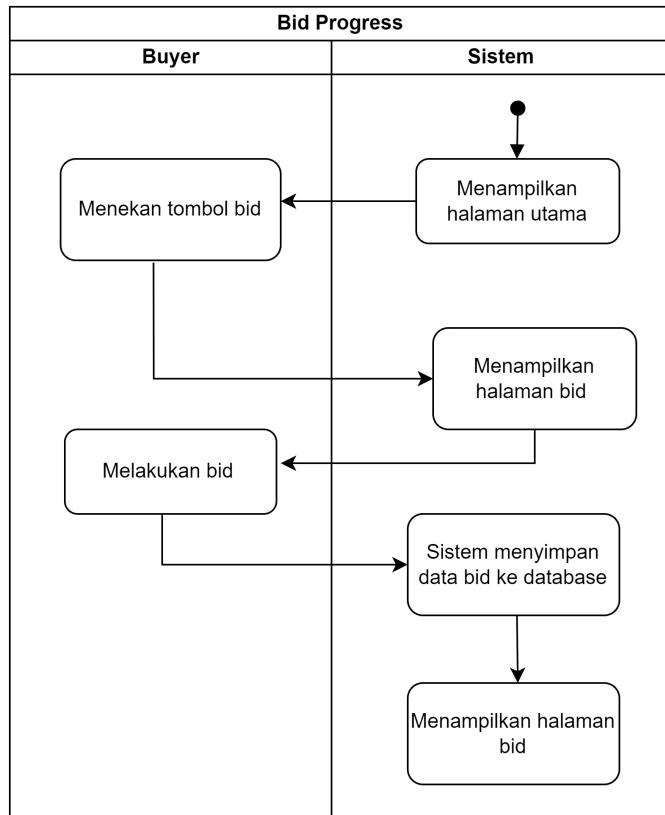
© 2022 PT Lelangku

3.1.9 Use Case Bid Progress

Nama	<i>Bid Progres</i>
Deskripsi	<i>Buyer</i> melakukan <i>bid</i> untuk mendapatkan barang lelang
Pre-kondisi	Melakukan <i>bid</i>
Post-kondisi	Mendapatkan hasil <i>bid</i>
Platform	<i>Website</i>

Table Typical Course of Event

Aktor	Sistem
	1. Sistem menampilkan halaman utama
2. Buyer menekan tombol <i>bid</i>	
	3. Menampilkan halaman <i>bid</i>
4. <i>Buyer</i> melakukan <i>bid</i>	
	5. Sistem menyimpan data <i>bid</i> ke <i>database</i>
	6. sistem menampilkan halaman <i>bid</i>



The screenshot shows a product listing for a "Redmi Note 10 - Mulus - No minus - Fungsi HP Normal - Layar Normal". The phone is shown in two colors: black and blue. The current bid is displayed as "Rp. 3.000.000,00". A prominent blue button labeled "Bid | + Rp. 50.000" is visible. Below the main image, there are five smaller thumbnail images of the phone from different angles. The page includes standard e-commerce features like a search bar, user profile, and navigation links.

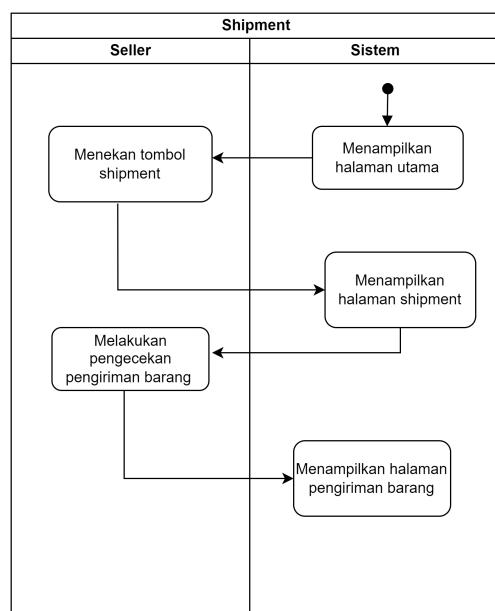
3.1.10 Use Case Shipment

Nama	Shipment
------	----------

Deskripsi	Untuk pengiriman barang
Pre-kondisi	Membayar barang yang sudah di-bid
Post-kondisi	Barang dikirim ke <i>buyer</i>
Platform	<i>Website</i>

Table Typical Course of Event

Aktor	Sistem
	1. Sistem menampilkan halaman utama
2. Buyer menekan tombol Toko	
	3. Sistem menampilkan halaman Toko
4. Melakukan pengecekan pengiriman barang	
	5. Menampilkan halaman pengiriman barang



LelangKu Kategori

Daftar Progress Bid

Cari transaksi di sini

Status: Semua, Belum Dikirim, Dikemas, Dikirim, Selesai

No. lote: 1234567890123456
Redmi Note 10 - Mulus - No minus - Fungsi HP Normal - Layar Normal
Tiba sebelum 27 Oktober 2022
Rp. 3.000.000,00

Look Barang

LelangKu
Bantuan dan Panduan
Syarat & Ketentuan
Kebijakan Privasi

Ikuti kami
Facebook, Instagram, Twitter

© 2022 PT Lelangku

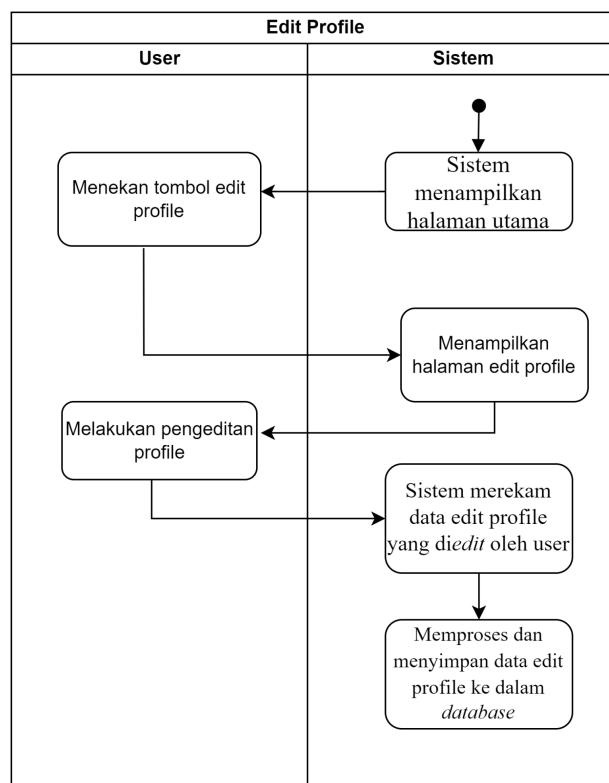
3.1.11 Use Case Edit Profile

Nama	<i>Edit Profile</i>
Deskripsi	User dapat mengedit informasi mulai dari nama, Nomor HP, Provinsi, Kota, Kecamatan, Kelurahan, Alamat, Kode Pos, dan photo profile
Pre-kondisi	User mengedit profile
Post-kondisi	Profile akan terupdate
Platform	<i>Website</i>

Table Typical Course of Event

Aktor	Sistem
	1. Sistem menampilkan halaman utama
2. User menekan tombol editprofile	

	3. Sistem menampilkan halaman edit profile
4. Melakukan pengeditan profile	
	5. Sistem merekam data edit profile yang <i>diinput</i> oleh user
	6. Memproses dan menyimpan data edit profile ke dalam <i>database</i>



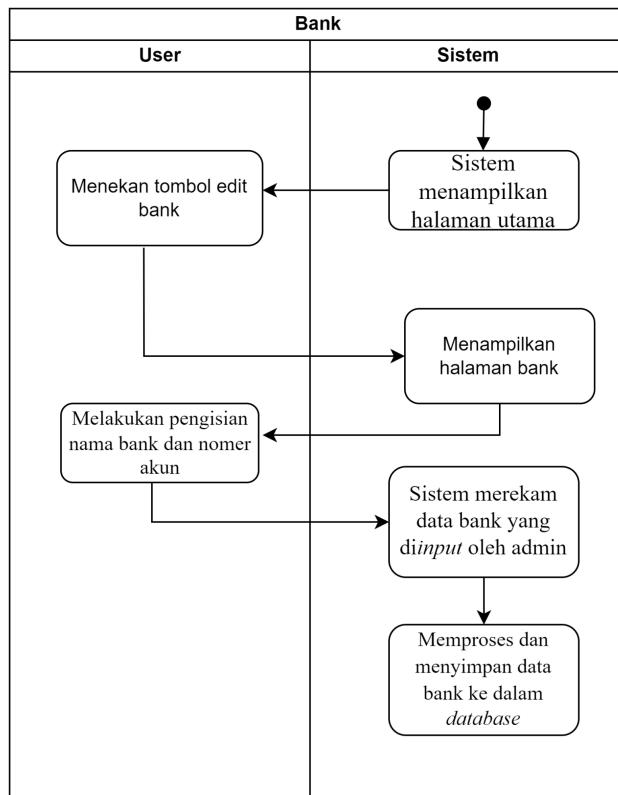
3.1.12 Use Case Bank

Nama	<i>Bank</i>
Deskripsi	sebagai fitur pendukung dalam melakukan operasional transaksi
Pre-kondisi	Mengisi nama bank dan nomer akun
Post-kondisi	data bank akan tersimpan pada database
Platform	<i>Website</i>

Table Typical Course of Event

Aktor	Sistem
	1. Sistem menampilkan halaman utama
2. Admin menekan tombol bank	
	3. Sistem menampilkan halaman bank
4. Melakukan pengisian nama bank dan nomer akun	
	7. Sistem merekam data bank

	yang diinput oleh admin
	8. Memproses dan menyimpan data bank ke dalam <i>database</i>

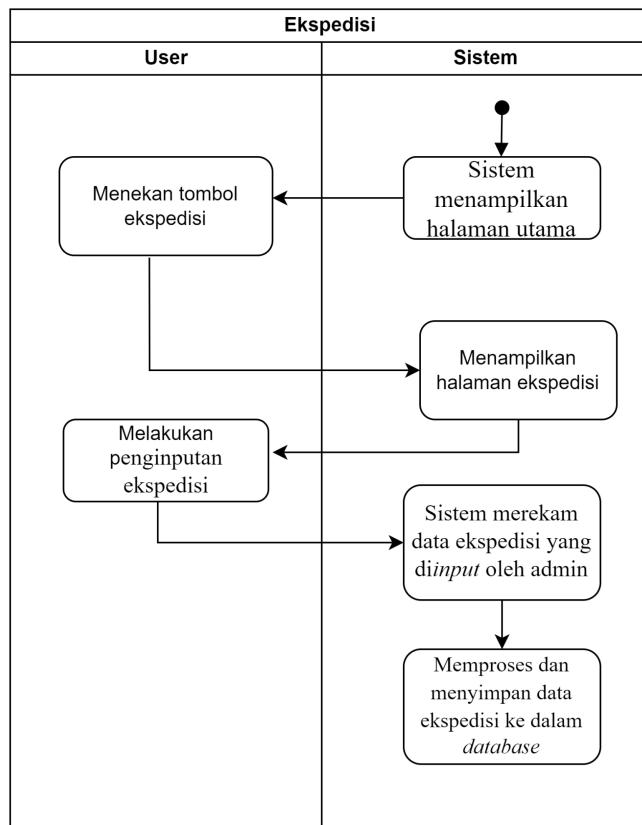


Bank			
No	Bank Name	Account Number	Action
1	BCA	1111111	<button>Delete</button>
2	BSI	2222222	<button>Delete</button>
3	Mandiri	1234543252	<button>Delete</button>
4	BRI	1231234	<button>Delete</button>
5	CIMB	23523532	<button>Delete</button>

3.1.13 Use Case Ekspedisi

Nama	<i>Ekspedisi</i>
Deskripsi	sebagai fitur pendukung dalam melakukan operasional pengiriman barang.
Pre-kondisi	Admin menginput ekspedisi
Post-kondisi	data ekspedisi akan tersimpan pada database
Platform	<i>Website</i>

Aktor	Sistem
	1. Sistem menampilkan halaman utama
2. Admin menekan tombol ekspedisi	
	3. Sistem menampilkan halaman ekspedisi
4. Melakukan penginputan ekspedisi	
	5. Sistem merekam data ekspedisi yang <i>diinput</i> oleh admin
	6. Memproses dan menyimpan data ekspedisi ke dalam <i>database</i>

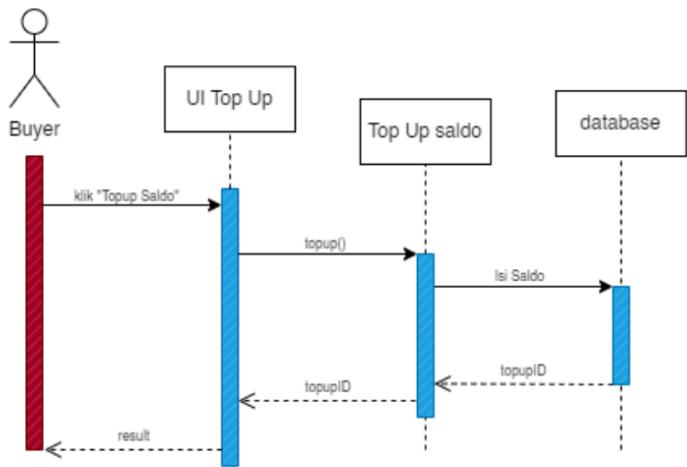


Expedition

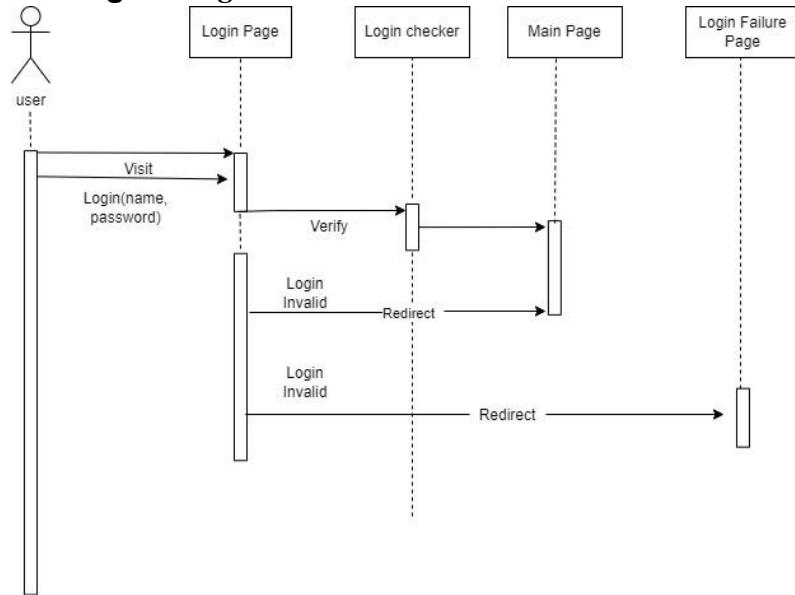
Expandable Table

No	Nama	Action
1	Si Cepat	<button style="background-color: #007bff; color: white; border: none; padding: 5px 10px; border-radius: 5px;">Delete</button>
2	JNE	<button style="background-color: #007bff; color: white; border: none; padding: 5px 10px; border-radius: 5px;">Delete</button>
3	JNT	<button style="background-color: #007bff; color: white; border: none; padding: 5px 10px; border-radius: 5px;">Delete</button>
4	Paxel	<button style="background-color: #007bff; color: white; border: none; padding: 5px 10px; border-radius: 5px;">Delete</button>
5	Pos Indonesia	<button style="background-color: #007bff; color: white; border: none; padding: 5px 10px; border-radius: 5px;">Delete</button>
6	Shoppe	<button style="background-color: #007bff; color: white; border: none; padding: 5px 10px; border-radius: 5px;">Delete</button>

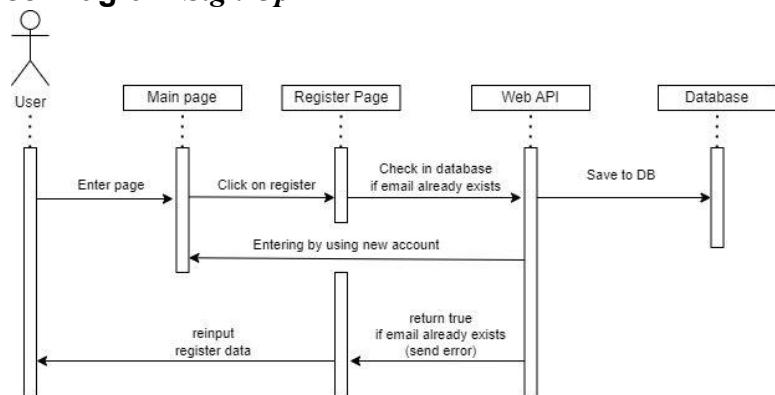
3.2.1 Sequence Diagram Top Up Deposit



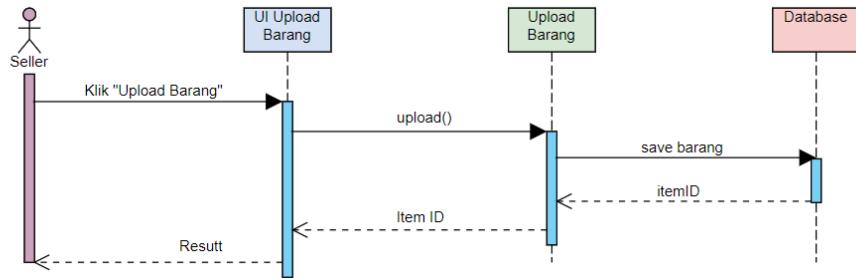
3.2.2 Sequence Diagram Sign In



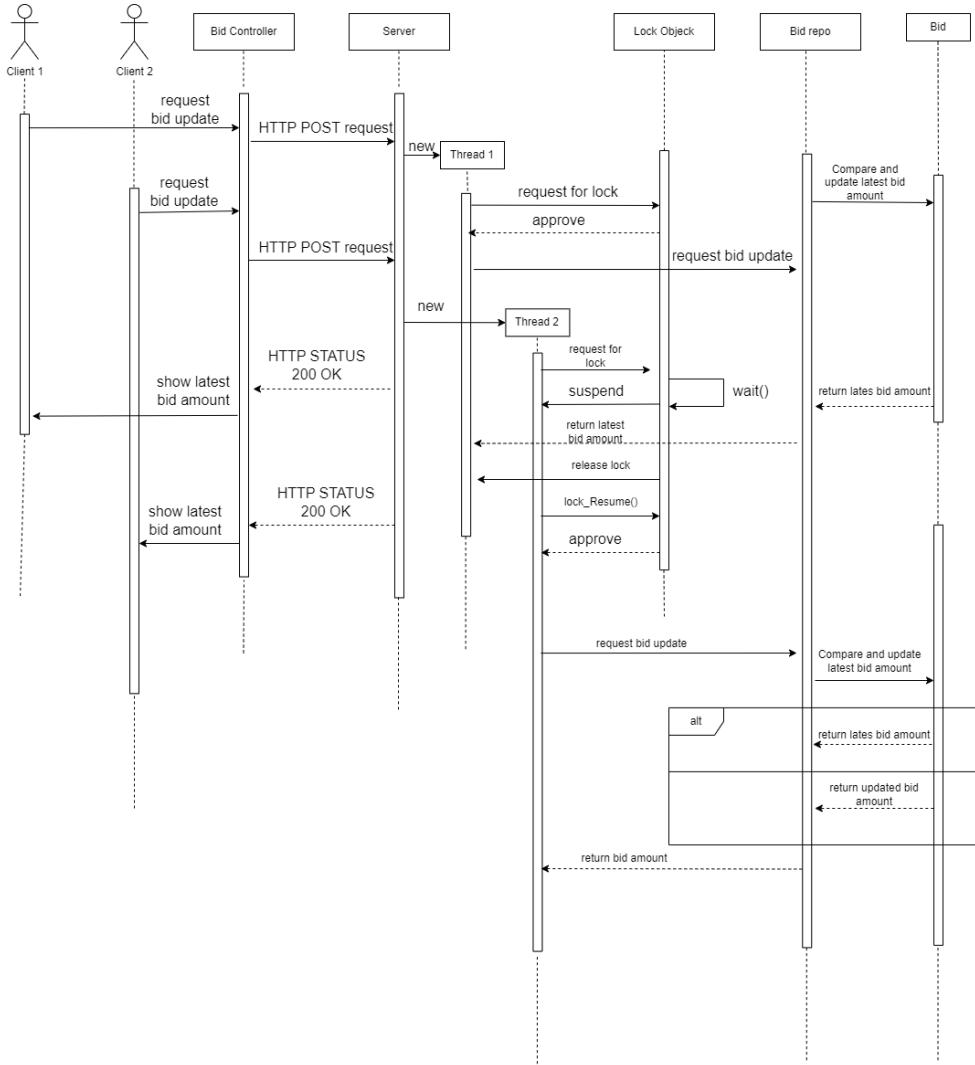
3.2.3 Sequence Diagram Sign Up



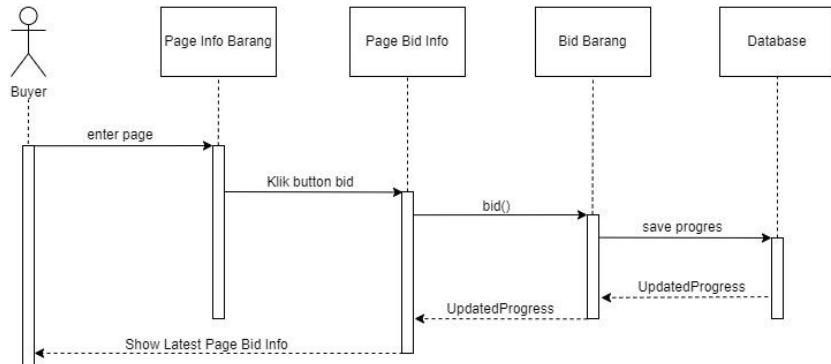
3.2.4 Sequence Diagram Upload



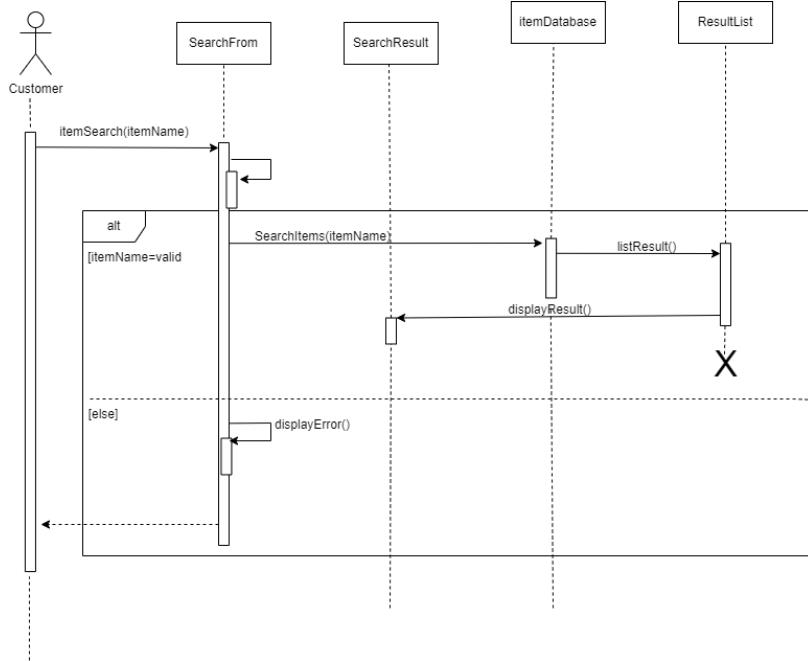
3.2.5 Sequence Diagram Bid Item



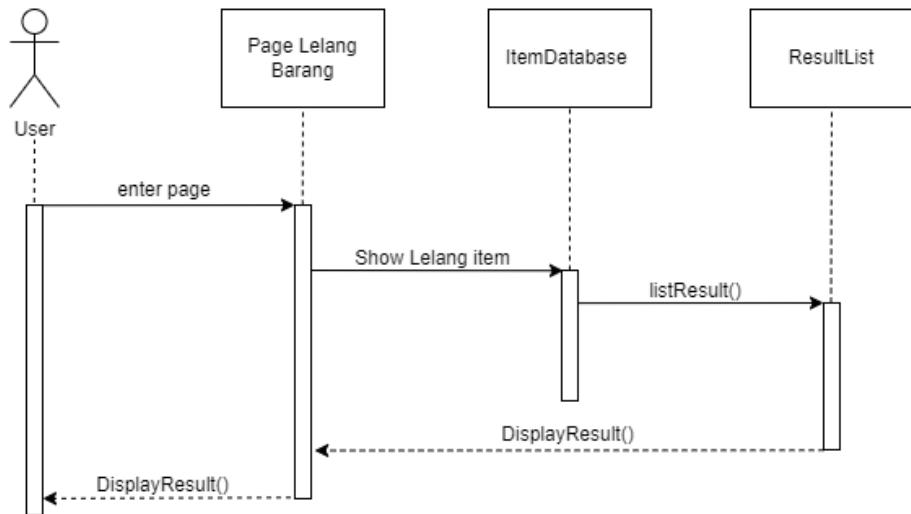
3.2.6 Sequence Diagram Bid progress



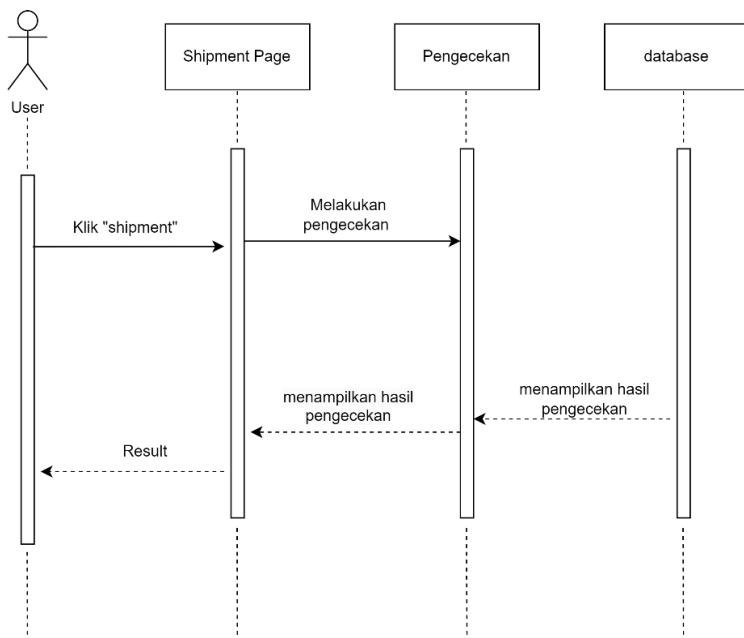
3.2.7 Sequence Diagram Category



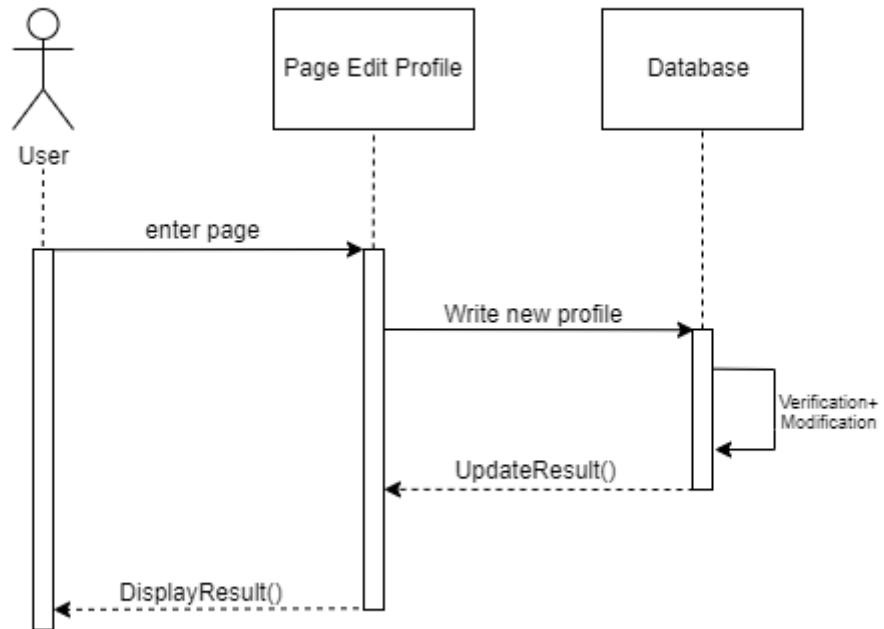
3.2.8 Sequence Diagram Progress Check



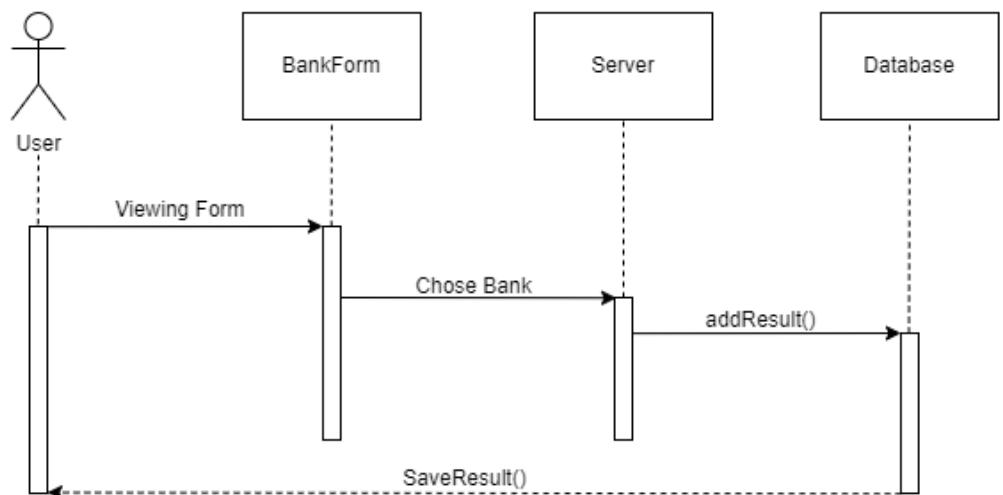
3.2.9 Sequence Diagram Shipment



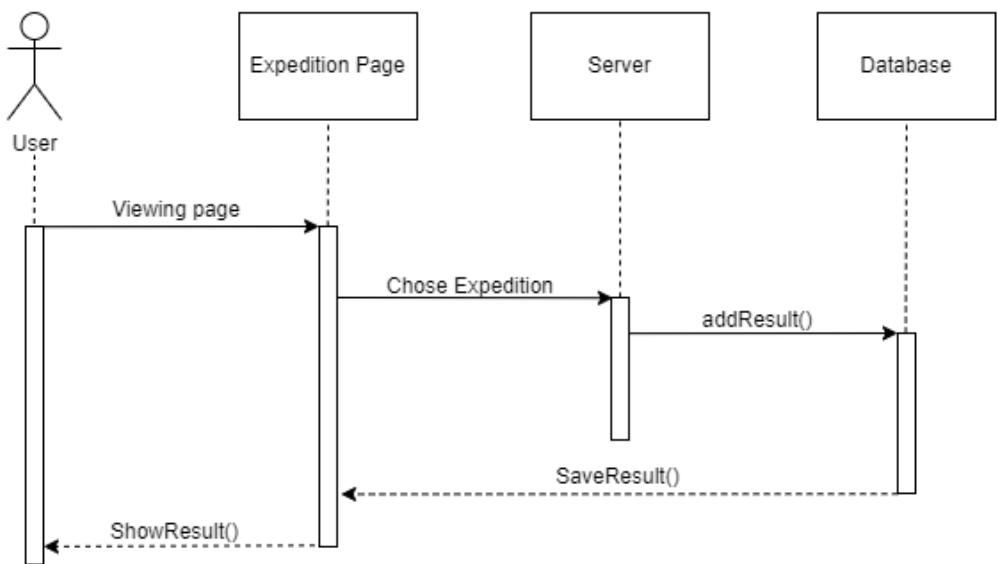
3.2.10 Sequence Diagram Edit Profile



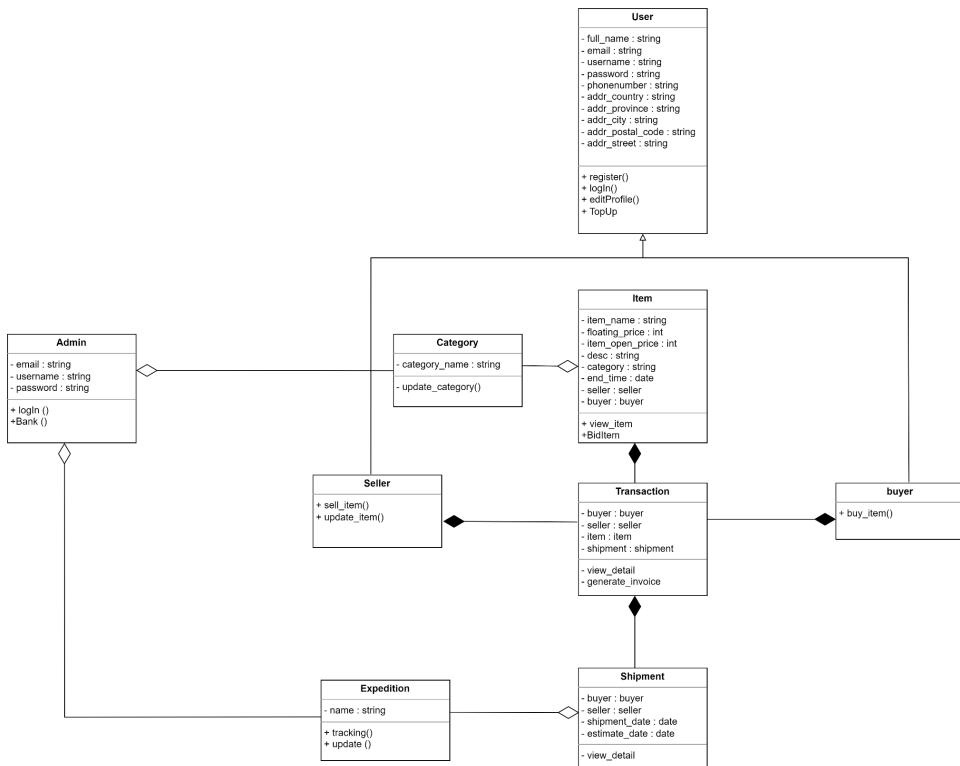
3.2.11 Sequence Diagram Bank



3.2.12 Sequence Diagram Expedition



3.3 Diagram Kelas



3.4 Perancangan Detail Kelas

No	Nama Kelas Perancangan	Nama Kelas Analisis Terkait
1.	User	<i>buyer, seller</i>
2.	Admin	<i>Category, expedition</i>
3.	Buyer	<i>Transaction, user, seller</i>
4.	Seller	<i>Transaction, user, buyer</i>
5.	Category	<i>Item, admin</i>
6.	Shipment	<i>Transaction, expedition</i>
7.	Item	<i>Item, Transaction</i>
8.	Transaction	<i>Item, buyer, seller, Shipment</i>
9.	Expedition	<i>Admin, shipment</i>

3.4.1 Kelas User

Nama Kelas : User

<i>Nama Operasi</i>	<i>Visibility (-private, +public)</i>	<i>Keterangan</i>
Register	<i>public</i>	<i>Method</i>
Login	<i>public</i>	<i>Method</i>
Update profile	<i>private</i>	<i>Method</i>
TopUp	<i>public</i>	<i>Method</i>
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
<i>Full name</i>	<i>Private</i>	<i>String</i>
<i>email</i>	<i>Private</i>	<i>String</i>
<i>Username</i>	<i>Private</i>	<i>String</i>
<i>Password</i>	<i>Private</i>	<i>String</i>
<i>Phonenumber</i>	<i>Private</i>	<i>String</i>
<i>addr country</i>	<i>Private</i>	<i>String</i>
<i>addr province</i>	<i>Private</i>	<i>String</i>

3.4.1 Kelas Admin

Nama Kelas : Admin

<i>Nama Operasi</i>	<i>Visibility (-private, +public)</i>	<i>Keterangan</i>
Login	<i>public</i>	<i>Method</i>
Bank	<i>public</i>	<i>Method</i>
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
<i>email</i>	<i>Private</i>	<i>String</i>
<i>Username</i>	<i>Private</i>	<i>String</i>
<i>Password</i>	<i>Private</i>	<i>String</i>

3.4.2 Kelas Category

Nama Kelas : Category

<i>Nama Operasi</i>	<i>Visibility (-private, +public)</i>	<i>Keterangan</i>

<i>Update category</i>	<i>private</i>	<i>Method</i>
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
<i>Category name</i>	<i>Private</i>	<i>String</i>

3.4.3 Kelas Item

Nama Kelas : Item

<i>Nama Operasi</i>	<i>Visibility (-private, +public)</i>	<i>Keterangan</i>
<i>View item</i>	<i>public</i>	<i>Method</i>
<i>BidItem</i>		
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
<i>Item name</i>	<i>Private</i>	<i>String</i>
<i>price</i>	<i>Private</i>	<i>int</i>
<i>Desc</i>	<i>Private</i>	<i>String</i>
<i>Category</i>	<i>Private</i>	<i>String</i>
<i>End time</i>	<i>Private</i>	<i>Date</i>
<i>Seller</i>	<i>Private</i>	<i>Seller</i>
<i>Buyer</i>	<i>Private</i>	<i>Buyer</i>

3.4.4 Kelas Transaction

Nama Kelas : Transaction

<i>Nama Operasi</i>	<i>Visibility (-private, +public)</i>	<i>Keterangan</i>
<i>View detail</i>	<i>private</i>	<i>Method</i>
<i>generate invoice</i>	<i>private</i>	<i>Method</i>
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
<i>Buyer</i>	<i>Private</i>	<i>Buyer</i>
<i>Seller</i>	<i>Private</i>	<i>Seller</i>
<i>Item</i>	<i>Private</i>	<i>Item</i>
<i>Shipment</i>	<i>Private</i>	<i>Shipment</i>

3.4.5 Kelas Buyer

Nama Kelas : Transaction

<i>Nama Operasi</i>	<i>Visibility (-private, +public)</i>	<i>Keterangan</i>
<i>Buy item</i>	<i>Public</i>	<i>Method</i>

3.4.6 Kelas Seller

Nama Kelas : Seller

<i>Nama Operasi</i>	<i>Visibility (-private, +public)</i>	<i>Keterangan</i>
<i>Sell item</i>	<i>Public</i>	<i>Method</i>
<i>Update item</i>	<i>Public</i>	<i>Method</i>

3.4.7 Kelas *Shipment*

Nama Kelas : *Shipment*

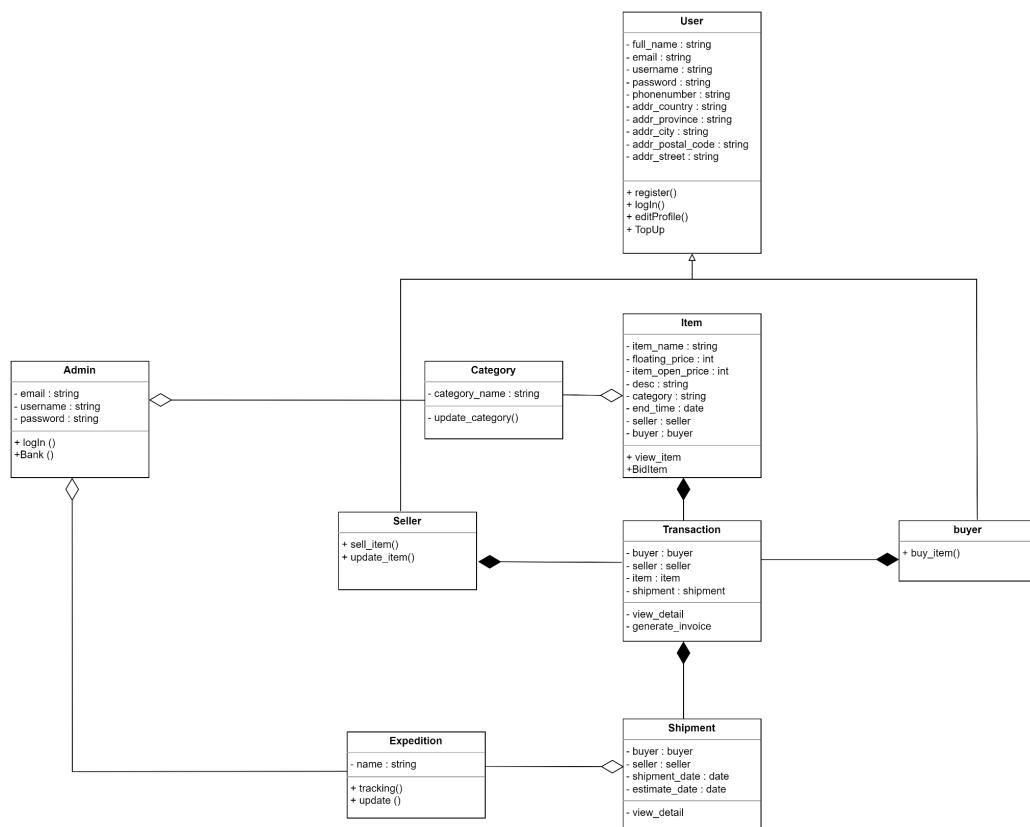
<i>Nama Operasi</i>	<i>Visibility (-private, +public)</i>	<i>Keterangan</i>
<i>View detail</i>	<i>private</i>	<i>Method</i>
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
<i>Buyer</i>	<i>Private</i>	<i>Buyer</i>
<i>Seller</i>	<i>Private</i>	<i>Seller</i>
<i>Shipment date</i>	<i>Private</i>	<i>date</i>
<i>Item</i>	<i>Private</i>	<i>Item</i>

3.4.8 Kelas *Expedition*

Nama Kelas : *Shipment*

<i>Nama Operasi</i>	<i>Visibility (-private, +public)</i>	<i>Keterangan</i>
<i>Tracking</i>	<i>Public</i>	<i>Method</i>
<i>Update</i>	<i>Private</i>	<i>Method</i>
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
<i>Name</i>	<i>Private</i>	<i>String</i>

3.5 Diagram Kelas Keseluruhan



3.6 Algoritma/Query

3.6.1 Algoritma #1

Nama Kelas : User

Nama Operasi : Sign up

Algoritma : Sign up

(Algo-xxx)

```
register: async(req,res,next)=>{
    try {
        const payload = req.body
        if(payload.password.length > 8){
            if(payload.username.length != 0){
                if(payload.name.length != 0){
                    if(payload.email.length != 0){
                        let user = new User(payload)
                        await user.save()
                        delete user._doc.password
                        req.flash('alertMessage', "Berhasil mendaftar, silahkan masuk")
                        req.flash('alertStatus', "success")
                        res.redirect('/signin')
                    }else{
                        req.flash('alertMessage', "Email harus diisi")
                        req.flash('alertStatus', "danger")
                        res.redirect('/signup')
                    }
                }else{
                    req.flash('alertMessage', "Nama harus diisi")
                    req.flash('alertStatus', "danger")
                    res.redirect('/signup')
                }
            }else{
                req.flash('alertMessage', "Username harus diisi")
                req.flash('alertStatus', "danger")
                res.redirect('/signup')
            }
        }else{
            req.flash('alertMessage', "Panjang kata sandi harus lebih dari 8 karakter")
            req.flash('alertStatus', "danger")
            res.redirect('/signup')
        }
    } catch (err) {
        if(err && err.name === "ValidationError"){
            req.flash('alertMessage', "Email sudah terdaftar")
            req.flash('alertStatus', "danger")
            res.redirect('/signup')
        }
        next(err)
    }
}
```

{Jika mengacu query tertentu, lengkapi tabel query di bawah}

3.6.2 Algoritma #2

Nama Kelas : User

Nama Operasi : Sign in
 Algoritma : Sign in
 (Algo-xxx)

```
actionSignIn: async(req, res)=>{
    try {
        const {email, password} = req.body
        const admin = await Admin.findOne({
            email: email
        })
        if(admin) {
            const checkPass = await bcrypt.compare(password, admin.password)
            if(checkPass){
                req.session.admin = {
                    id: admin._id,
                    email: admin.email,
                    username: admin.username
                }
                console.log(req.session.admin)
                res.redirect('/admin/dashboard')
            }else{
                req.flash('alertMessage', "Kata sandi salah")
                req.flash('alertStatus', "danger")
                res.redirect('/admin')
            }
        }else{
            req.flash('alertMessage', "Email tidak terdaftar sebagai admin")
            req.flash('alertStatus', "danger")
            res.redirect('/admin')
        }
    } catch (error) {
        console.log(error)
        res.redirect('/admin')
    },
},
```

```
signIn: async(req, res, next)=>{
    try {
        const {email, password} = req.body
        const user = await User.findOne({
            email: email
        })
        if(email.length != ""){
            if(user) {
                const checkPass = await bcrypt.compare(password, user.password)
                if(checkPass){
                    req.session.user = {
                        id: user._id,
                        email: user.email,
                        username: user.username,
                        balance: user.balance
                    }
                    console.log(req.session.user)
                    res.redirect('/')
                }else{
                    req.flash('alertMessage', "Kata sandi salah")
                    req.flash('alertStatus', "danger")
                    res.redirect('/signin')
                }
            }
        }
    }
},
```

```

        }else{
            req.flash('alertMessage', "Email belum terdaftar")
            req.flash('alertStatus', "danger")
            res.redirect('/signin')
        }
    }else{
        req.flash('alertMessage', "Email belum diisi")
        req.flash('alertStatus', "danger")
        res.redirect('/signin')
    }
} catch (error) {
    console.log(error)
    res.redirect('/signin')
}
},
actionSignOut: async(req, res)=>{
    req.session.destroy()
    res.redirect('/')
}

```

3.6.3 Algoritma #3

Nama Kelas : *User*
 Nama Operasi : *Sell item*
 Algoritma : *Sell item*

(Algo-xxx)

```

createItem: async(req, res)=>{
    try {
        const {name, desc, category, price, limit, condition} = req.body
        const seller = req.session.user.id
        if(req.file){
            if(name.length > 0){
                if(desc.length > 0){
                    if(category != ""){
                        if(price>0){
                            if(limit == '3 hari'||limit == '7 hari'||limit == '14 hari'||limit == '21 hari'||limit == '28
hari'){
                                if(condition == "Baru" || condition == "Bekas"){
                                    const image = req.file.path.split('\\').slice(1).join('\\');
                                    const now = new Date();
                                    if(limit=='3 hari'){
                                        timeLimit = date.addSeconds(now, 3*86400);
                                    }else if(limit=='7 hari'){
                                        timeLimit = date.addSeconds(now, 7*86400);
                                    }else if(limit=='14 hari'){
                                        timeLimit = date.addSeconds(now, 14*86400);
                                    }else if(limit=='21 hari'){
                                        timeLimit = date.addSeconds(now, 21*86400);
                                    }else{
                                        timeLimit = date.addSeconds(now, 28*86400);
                                    }
                                    let item = new Item({ name, desc, category, price, timeLimit, condition, seller,
image:image})
                                    await item.save()
                                    req.flash('alertMessage', "Berhasil melelang barang")
                                    req.flash('alertStatus', "success")
                                    res.redirect("/")
                                }else{
                                    req.flash('alertMessage', "Kondisi barang hasur diisi")
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

        req.flash('alertStatus', "danger")
        res.redirect("/shop/sellitem")
    }
} else{
    req.flash('alertMessage', "Rentang waktu pelelangan harus diisi")
    req.flash('alertStatus', "danger")
    res.redirect("/shop/sellitem")
}
} else{
    req.flash('alertMessage', "Harga harus diisi")
    req.flash('alertStatus', "danger")
    res.redirect("/shop/sellitem")
}
} else{
    req.flash('alertMessage', "Kategori harus dipilih")
    req.flash('alertStatus', "danger")
    res.redirect("/shop/sellitem")
}
} else{
    req.flash('alertMessage', "Deskripsi barang harus diisi")
    req.flash('alertStatus', "danger")
    res.redirect("/shop/sellitem")
}
} else{
    req.flash('alertMessage', "Nama barang harus diisi")
    req.flash('alertStatus', "danger")
    res.redirect("/shop/sellitem")
}
} else{
    req.flash('alertMessage', "Gambar harus diisi")
    req.flash('alertStatus', "danger")
    res.redirect("/shop/sellitem")
}
} catch (err) {
    console.log(err)
}
}

```

3.6.4 Algoritma #4

Nama Kelas : *Transaction*
 Nama Operasi : *Bid item*
 Algoritma : *Bid item*

(Algo-xxx)

```

bidItem: async(req, res)=>{
    try {
        console.log("heyaa")
        const buyer = req.session.user.id
        const {id} = req.params
        const user = await User.findOne({
            _id: buyer
        })
        const item = await Item.findOne({
            _id: id
        })
        if(user.balance >= (item.price+50000)){
            await Item.findOneAndUpdate({
                id:id
            }, {buyer:buyer, $inc:{bidCount:1, price:50000}})
        }
    }
}

```

```

        req.flash('alertMessage', "Selamat bid berhasil dilakukan")
        req.flash('alertStatus', "success")
        res.redirect('/')
    }else{
        req.flash('alertMessage', "Saldo anda tidak mencukupi, silahkan isi saldo terlebih dahulu")
        req.flash('alertStatus', "danger")
        res.redirect('/')
    }
} catch (err) {
    req.flash('alertMessage', err.message)
    req.flash('alertStatus', "danger")
    res.redirect('/')
}
}

```

3.6.5 Algoritma #5

Nama Kelas : *Transaction*
 Nama Operasi : *Bid info*
 Algoritma : *Bid info*

(Algo-xxx)

```

detailItem: async(req, res)=>{
    try {
        const alertMessage = req.flash("alertMessage")
        const alertStatus = req.flash("alertStatus")
        const alert = {message:alertMessage, status:alertStatus}

        const user = await User.findOne({_id: req.session.user.id})
        const {id} = req.params
        let item = await Item.findOne({
            _id:id
        }).populate('seller')
        const item_2 = await Item.find({post:"Y"})
        let limit = item.timeLimit
        const today = new Date();
        let reameningTime = date.subtract(limit, today).toHours();
        let timeConv = ""
        if(date.subtract(limit, today).toDays()>=1){
            reameningTime = date.subtract(limit, today).toDays();
            timeConv = Math.floor(reameningTime)+" Hari lagi"
        }else if(date.subtract(limit, today).toDays()<1){
            reameningTime = date.subtract(limit, today).toHours();
            timeConv = Math.floor(reameningTime)+" Jam lagi"
        }else if(date.subtract(limit, today).toHours()<1){
            reameningTime = date.subtract(limit, today).toMinutes();
            timeConv = Math.floor(reameningTime)+" Menit lagi"
        }
        if(limit < today){
            console.log("first")
            timeConv = " Selesai"
            await User.findOneAndUpdate({
                _id: req.session.user.id
            }, {post:"N", status:"Mengirim"})
        }
        res.render('client/detailItem/index',{
            user;
        })
    }
}

```

```

        id: req.session.user.id,
        title:item.name,
        name:user.name,
        balance: user.balance,
        itemName: item.name,
        desc: item.desc,
        price: item.price,
        item,
        item_2,
        alert,
        date:timeConv
    })
} catch (err) {
    res.status(500).json({
        message: err.message
    })
}
}
}

```

3.6.6 Algoritma #6

Nama Kelas : *Item*
 Nama Operasi : *Category*
 Algoritma : *Category*

(Algo-xxx)

```

index:async(req, res)=>{
    try {
        const category = await Category.find()
        res.render('admin/category/index', {
            title: 'Category',
            category,
            active: "category",
        })
    } catch (err) {
        console.log(err)
        res.redirect('/admin/category')
    }
},
actionCreate:async(req, res)=>{
    try {
        const {name} = req.body
        let category = await Category({name})
        await category.save()
        res.redirect('/admin/category')
    } catch (err) {
        console.log(err)
    }
},
actionEdit:async(req, res)=>{
    try {
        const {id} = req.params
        const {name}= req.body
        await Category.findOneAndUpdate(
            {id:id},
            {name})
        res.redirect('/admin/category')
    } catch (err) {
        console.log(err)
    }
}

```

```

},
actionDelete:async(req, res)=>{
  try {
    const {id} = req.params
    await Category.findOneAndRemove({
      _id:id
    })
    res.redirect('/admin/category')
  } catch (err) {
    console.log(err)
  }
}

```

3.6.7 Algoritma #7

Nama Kelas : *Transaction*
 Nama Operasi : *Progress check*
 Algoritma : *Progress check*

(Algo-xxx)

```

viewShop: async(req, res)=>{
  try {
    const expedition = await Expedition.find()
    const user = await User.findOne({_id: req.session.user.id})
    let item = await Item.find(
      {seller: req.session.user.id, post:"Y", status:"Bid",
       $or: [ { seller: req.session.user.id }, { post:"N" }, { status:"Bid" }, { buyyer:{ $eq: null } }]}
    )
    let item2 = await Item.find(
      {seller: req.session.user.id, post:"N", status:"Bid", buyyer:{ $ne: null }}
    ).populate('buyyer')
    let item3 = await Item.find(
      {seller: req.session.user.id, post:"N", status:"Mengirim"}
    ).populate('buyyer')
    let item4 = await Item.find(
      {seller: req.session.user.id, post:"N", status:"Selesai"}
    )
    res.render('client/itemToko/index',{
      user,
      id: req.session.user.id,
      name : user.name,
      email: user.email,
      balance: user.balance,
      phoneNumber: user.phoneNumber,
      province: user.province,
      city: user.city,
      district: user.district,
      ward: user.ward,
      postalCode: user.postalCode,
      fullAddr: user.fullAddr,
      image: user.image,
      item,
      item2,
      item3,
      item4,
      expedition
    })
  } catch (err) {
    res.status(500).json({
      message: err.message
    })
  }
}

```

```
}
```

3.6.8 Algoritma #8

Nama Kelas : *Item*
Nama Operasi : *On progress item*
Algoritma : *On progress item*

(Algo-xxx)

```
viewShop: async(req, res)=>{
    try {
        const expedition = await Expedition.find()
        const user = await User.findOne({_id: req.session.user.id})
        let item = await Item.find(
            {seller: req.session.user.id, post:"Y", status:"Bid",
            $or: [ { seller: req.session.user.id }, { post:"N" }, { status:"Bid" }, { buyyer:{ $eq: null } } ]}
        )
        let item2 = await Item.find(
            {seller: req.session.user.id, post:"N", status:"Bid", buyyer:{ $ne: null }}
        ).populate('buyyer')
        let item3 = await Item.find(
            {seller: req.session.user.id, post:"N", status:"Mengirim"}
        ).populate('buyyer')
        let item4 = await Item.find(
            {seller: req.session.user.id, post:"N", status:"Selesai"}
        )
        res.render('client/itemToko/index',{
            user,
            id: req.session.user.id,
            name : user.name,
            email: user.email,
            balance: user.balance,
            phoneNumber: user.phoneNumber,
            province: user.province,
            city: user.city,
            district: user.district,
            ward: user.ward,
            postalCode: user.postalCode,
            fullAddr: user.fullAddr,
            image: user.image,
            item,
            item2,
            item3,
            item4,
            expedition
        })
    } catch (err) {
        res.status(500).json({
            message: err.message
        })
    }
}
```

3.6.9 Algoritma #9

Nama Kelas : *Transaction*
Nama Operasi : *Bid progress*

Algoritma : Bid progress (Algo-xxx)

```
viewTransaction: async(req,res)=>{
  try {
    const user = await User.findOne({_id: req.session.user.id})
    const item = await Item.find({buyer:req.session.user.id})
    res.render('client/transaction/index', {
      user,
      item,
      id:req.session.user.id,
      title: "Transaction",
      username : user.username,
      name : user.name,
      balance: user.balance
    })
  } catch (err) {
    console.log(err)
  }
},
finishItem: async(req, res)=>{
  try {
    const {id} = req.params
    await Item.findOneAndUpdate({
      _id:id},{status:"Selesai"})
    res.redirect('/user/transaction')
  } catch (err) {
    console.log(err)
  }
}
```

3.6.10 Algoritma #10

Nama Kelas : Shipment

Nama Operasi : Shipment

Algoritma : Shipment

(Algo-xxx)

```
sendItem: async(req, res)=>{
  try {
    const {id} = req.params
    const {expedition, receiptNumber} = req.body
    console.log(expedition)
    let receipt = await Invoice({
      expedition, item:id, receiptNumber
    })
    await receipt.save()
    await Item.findOneAndUpdate({_id:id},{status:"Mengirim"})
    res.redirect('/')
  } catch (err) {
    console.log(err)
  }
}
```

3.6.11 Algoritma #11

Nama Kelas : Admin

Nama Operasi : SignIn

Algoritma : SignIn

(Algo-xxx)

```
viewSignin: async(req, res)=>{
  try {
    const alertMessage = req.flash("alertMessage")
```

```

const alertStatus = req.flash("alertStatus")
const alert = {message:alertMessage, status:alertStatus}
if (req.session.admin === null || req.session.admin === undefined) {
    res.render('admin/auth/signIn',{
        title: "Admin | LelangKu",
        alert
    })
} else {
    res.redirect('/admin/dashboard')
}
} catch (err) {
    console.log(err)
    res.redirect('/admin')
}
},
actionSignIn: async(req, res)=>{
try {
    const {email, password} = req.body
    const admin = await Admin.findOne({
        email: email
    })
    if(admin) {
        const checkPass = await bcrypt.compare(password, admin.password)
        if(checkPass){
            req.session.admin = {
                id: admin._id,
                email: admin.email,
                username: admin.username
            }
            console.log(req.session.admin)
            res.redirect('/admin/dashboard')
        }else{
            req.flash('alertMessage', "Kata sandi salah")
            req.flash('alertStatus', "danger")
            res.redirect('/admin')
        }
    }else{
        req.flash('alertMessage', "Email tidak terdaftar sebagai admin")
        req.flash('alertStatus', "danger")
        res.redirect('/admin')
    }
} catch (error) {
    console.log(error)
    res.redirect('/admin')
}
}
}

```

3.6.12 Algoritma #12

Nama Kelas : Admin
 Nama Operasi : SignOut
 Algoritma : SignOut

(Algo-xxx)

```

actionSignOut: async(req, res)=>{
    req.session.destroy()
    res.redirect('/admin')
}

```

3.6.13 Algoritma #13

Nama Kelas : Admin

Nama Operasi : *createBank*
 Algoritma : *createBank* (Algo-xxx)

```
actionCreate: async(req, res)=>{
  try {
    const {bankName, accountNumber} = req.body
    let bank = await Bank({
      bankName, accountNumber
    })
    await bank.save()
    res.redirect('/admin/bank')
  } catch (err) {
    console.log(err)
    res.redirect('/admin/bank')
  }
}
```

3.6.14 Algoritma #14

Nama Kelas : *Admin*
 Nama Operasi : *editBank*
 Algoritma : *editBank* (Algo-xxx)

```
actionEdit: async(req, res)=>{
  try {
    const { id } = req.params;
    const {bankName, accountNumber } = req.body

    await Bank.findOneAndUpdate({
      _id: id
    }, { bankName, accountNumber });
    res.redirect('/admin/bank')

  } catch (err) {
    console.log(err)
    res.redirect('/admin/bank')
  }
}
```

3.6.15 Algoritma #15

Nama Kelas : *Admin*
 Nama Operasi : *deleteBank*
 Algoritma : *deleteBank* (Algo-xxx)

```
actionDelete: async(req, res)=>{
  try {
    const { id } = req.params;

    await Bank.findOneAndRemove({
      _id: id
    });
    res.redirect('/admin/bank')
  } catch (err) {
    console.log(err)
    res.redirect('/admin/bank')
  }
}
```

3.6.16 Algoritma #16

Nama Kelas : *Admin*

Nama Operasi : *createCategory*
Algoritma : *createCategory* (Algo-xxx)

```
actionCreate:async(req, res)=>{
  try {
    const {name} = req.body
    let category = await Category({name})
    await category.save()
    res.redirect('/admin/category')
  } catch (err) {
    console.log(err)
  }
}
```

3.6.17 Algoritma #17

Nama Kelas : *Admin*
Nama Operasi : *editCategory*
Algoritma : *editCategory* (Algo-xxx)

```
actionEdit:async(req, res)=>{
  try {
    const {id} = req.params
    const {name}= req.body
    await Category.findOneAndUpdate({
      id:id
    }, {name})
    res.redirect('/admin/category')
  } catch (err) {
    console.log(err)
  }
}
```

3.6.18 Algoritma #18

Nama Kelas : *Admin*
Nama Operasi : *deleteCategory*
Algoritma : *deleteCategory* (Algo-xxx)

```
actionDelete:async(req, res)=>{
  try {
    const {id} = req.params
    await Category.findOneAndRemove({
      id:id
    })
    res.redirect('/admin/category')
  } catch (err) {
    console.log(err)
  }
}
```

3.6.19 Algoritma #19

Nama Kelas : *Admin*
Nama Operasi : *createExpedition*
Algoritma : *createExpedition* (Algo-xxx)

```
actionCreate:async(req, res)=>{
  try {
    const {name} = req.body
    let expedition = await Expedition({name})
    await expedition.save()
    res.redirect('/admin/expedition')
  } catch (err) {
  }
}
```

```
        console.log(err)
    }
}
```

3.6.20 Algoritma #20

Nama Kelas : Admin
Nama Operasi : *editExpedition*
Algoritma : *editExpedition*

(Algo-xxx)

```
actionEdit:async(req, res)=>{
    try {
        const {id} = req.params
        const {name}= req.body
        await Expedition.findOneAndUpdate({
            id:id
        }, {name})
        res.redirect('/admin/expedition')
    } catch (err) {
        console.log(err)
    }
}
```

3.6.21 Algoritma #21

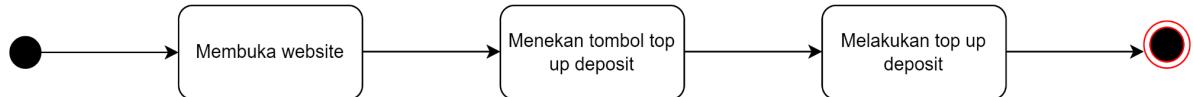
Nama Kelas : Admin
Nama Operasi : *deleteExpedition*
Algoritma : *deleteExpedition*

(Algo-xxx)

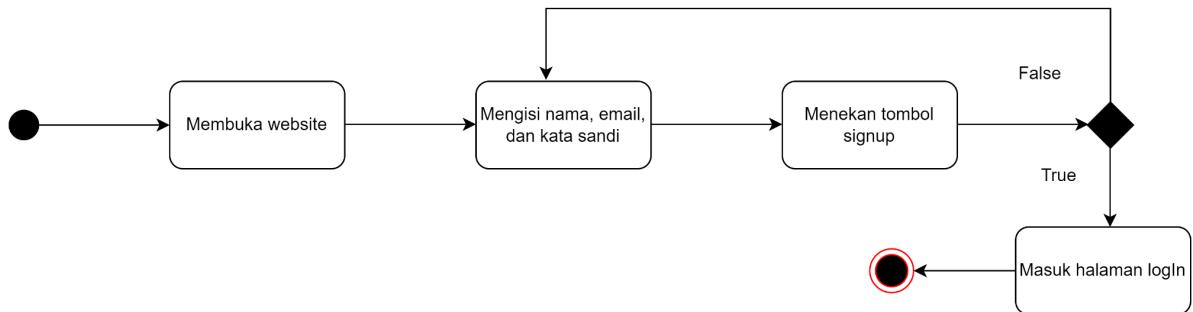
```
actionDelete:async(req, res)=>{
    try {
        const {id} = req.params
        await Expedition.findOneAndRemove({
            id:id
        })
        res.redirect('/admin/expedition')
    } catch (err) {
        console.log(err)
    }
}
```

3.7 Diagram Statechart

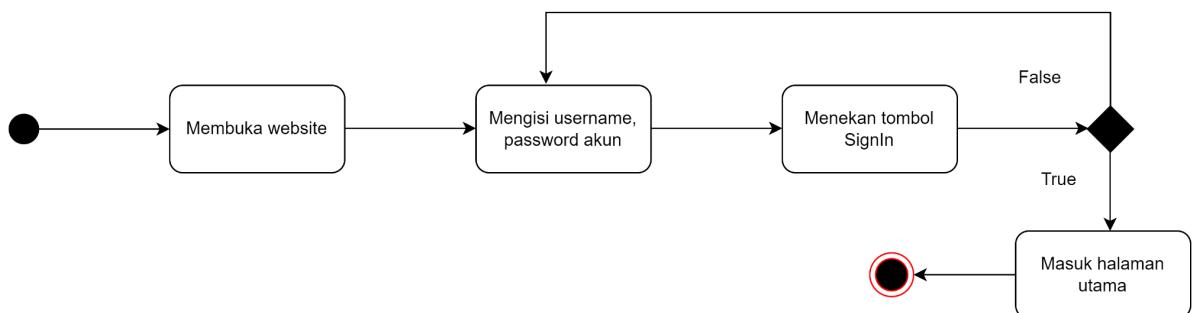
3.7.1 Top Up Deposit



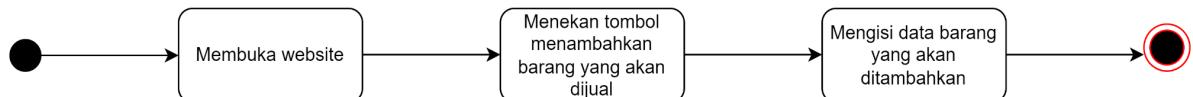
3.7.2 Sign Up



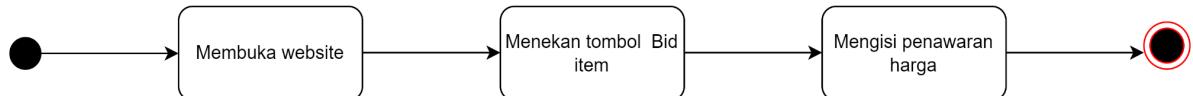
3.7.3 Sign In



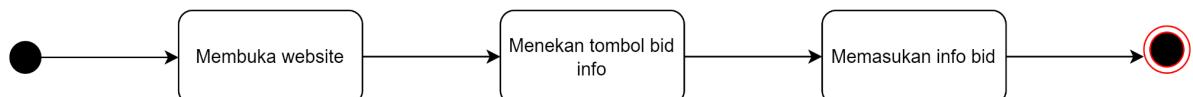
3.7.4 Upload / Sell item



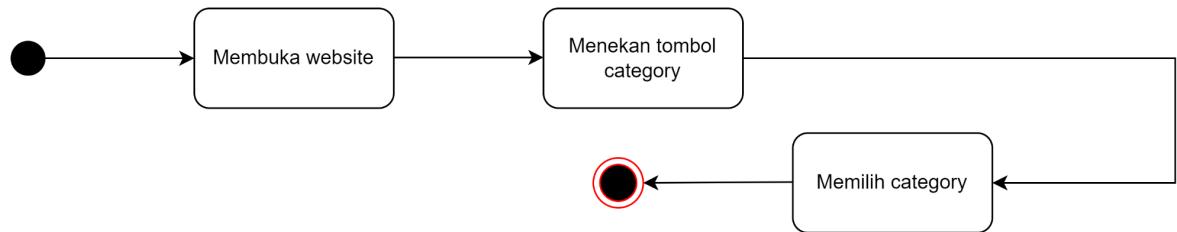
3.7.5 Bid item



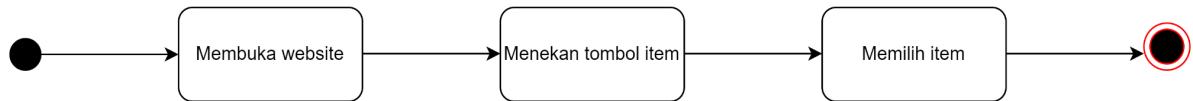
3.7.6 Bid info



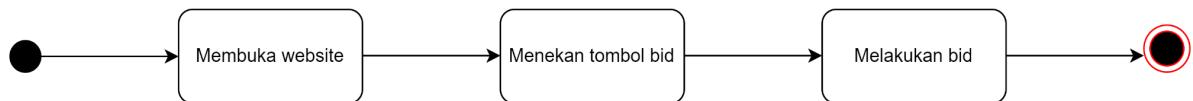
3.7.7 Category



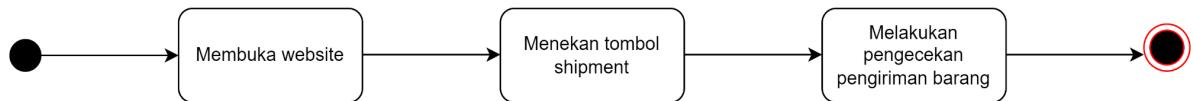
3.7.8 Progress check



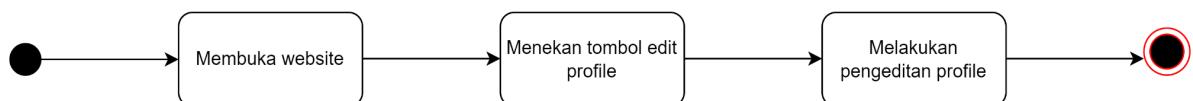
3.7.9 Bid Progress



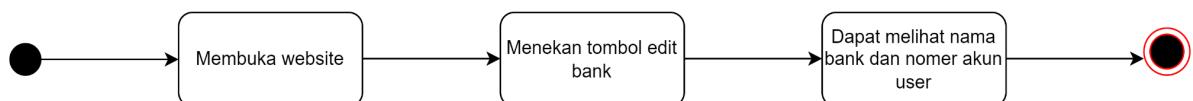
3.7.10 Shipment



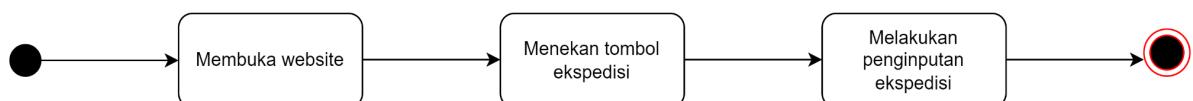
3.7.11 Edit Profile



3.7.12 Bank



3.7.13 Ekspedition



3.8 Perancangan Antarmuka

3.8.1 Halaman Sign In



The LelangKu logo features a stylized blue and yellow 'L' shape above the word 'LelangKu' in blue and yellow.

Masuk

Nomor HP atau Email

Password

Ingat saya [Lupa Password?](#)

Masuk

———— atau masuk dengan ———

 Google

[Belum punya akun? Daftar disini](#)

© 2022 | PT Lelangku

Id Objek	Jenis	Nama	Keterangan
Nomor Hp atau Email	Text Field	Nomor Hp atau Email	Mengisi data nomor Hp atau email
Password	Text Field	Password	Mengisi password.
Masuk	Button	Masuk	Jika diklik akan menuju tampilan Dashboard
Ingat Saya	Radio Button	Ingat Saya	
Lupa Password	Redirect Link	Lupa Password	jika diklik akan menuju tampilan lupa password
Google	Button	Google	jika diklik akan menuju tampilan google

3.8.2 Halaman Sign Up



The LelangKu logo features a stylized blue and yellow 'L' shape above the word 'LelangKu' in blue and yellow.

Daftar

Nama Lengkap

Alamat Email

Kata Sandi

Ulangi Kata Sandi

Daftar

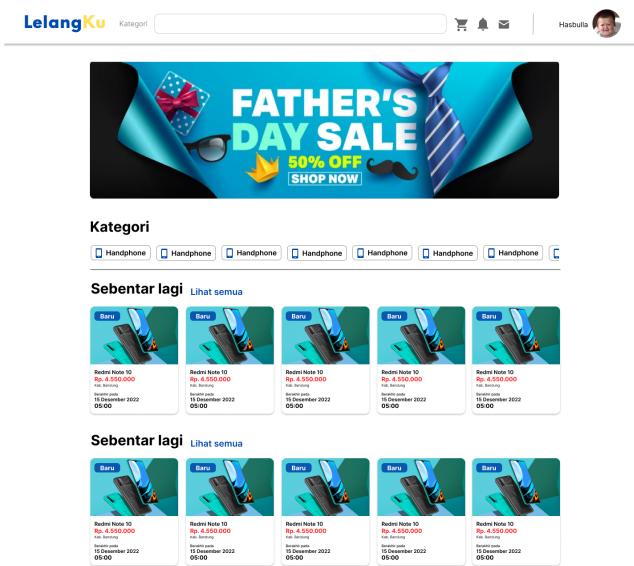
———— atau daftar dengan ———

 Google

© 2022 | PT Lelangku

Id Objek	Jenis	Nama	Keterangan
<i>Nama Lengkap</i>	<i>Text Field</i>	<i>Nama Lengkap</i>	<i>Mengisi data nama lengkap</i>
<i>Alamat Email</i>	<i>Text Field</i>	<i>Alamat Email</i>	<i>Mengisi data alamat email</i>
<i>Kata Sandi</i>	<i>Text Field</i>	<i>Kata Sandi</i>	<i>Mengisi data kata sandi</i>
<i>Ulangi Kata Sandi</i>	<i>Text Field</i>	<i>Ulangi Kata Sandi</i>	<i>Mengisi data ulangi kata sandi</i>
<i>Daftar</i>	<i>Button</i>	<i>Daftar</i>	<i>Jika diklik akan menuju tampilan SignIn</i>
<i>Google</i>	<i>Button</i>	<i>Google</i>	<i>Jika diklik akan menuju tampilan google</i>

3.8.3 Halaman ProgressItem



LelangKu
LelangKu is simply dummy text of the printing and typesetting industry. Lorem ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

Bantuan dan Panduan
Sosial & Keterbukaan
Kelebihan Privasi



© 2022 | PT Lelangku

Id Objek	Jenis	Nama	Keterangan
<i>Kategori</i>	<i>Text Field</i>	<i>Kategori</i>	<i>Mengisi kategori yang ingin di cari</i>

3.8.4 Halaman Progress Check

LelangKu Kategori Hasbullah

Daftar Progress Bid

Cari transaksi mu di sini

Status Service Berlangsung Dikirim Dikemas Selesai

 Redmi Note 10 – Mulus – No minus – Fungsi HP Normal – Layar Normal
Rp. 3.000.000,00
🕒 7j 13m
Anda pelelang tertinggi!! [Lihat Selengkapnya](#)

 Redmi Note 10 – Mulus – No minus – Fungsi HP Normal – Layar Normal
Rp. 3.000.000,00
🕒 7j 13m
Anda bukan pelelang tertinggi! [Lihat Selengkapnya](#)

Id Objek	Jenis	Nama	Keterangan
Kategori	Text Field	Kategori	Mengisi kategori yang ingin dicari
Status	Button	Status	jika mengklik akan menuju status pengiriman
Lihat Selengkapnya	Redirect Link	Lihat Selengkapnya	Jika mengklik akan menuju tampilan info lengkap
Cari transaksi	Text Field	Cari transaksi	Mengisi pencarian transaksi

3.8.5 Halaman Bid Progress

Id Objek	Jenis	Nama	Keterangan
<i>Kategori</i>	<i>Text Field</i>	<i>Kategori</i>	<i>Mengisi kategori yang ingin dicari</i>

Id Objek	Jenis	Nama	Keterangan
<i>Bid</i>	<i>Button</i>	<i>Bid</i>	<i>Melakukan bid</i>

3.8.6 Halaman Shipment

LelangKu Kategori Hasil Pemua

Cari transaksimu di sini

Status: Tiba sebelum 27 Oktober 2022 Total Harga: Rp. 3.000.000,00 Lacak Barang

LelangKu
Loren ipsum is simply dummy text of the printing and typesetting industry. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s, when it became a type specimen book.

Bantuan dan Panduan Syarat & Ketentuan Kebijakan Privasi

Ikuti kami

© 2022 PT LelangKu

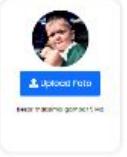
Id Objek	Jenis	Nama	Keterangan
<i>Kategori</i>	<i>Text Field</i>	<i>Kategori</i>	<i>Mengisi kategori yang ingin dicari</i>
<i>Status</i>	<i>Button</i>	<i>Status</i>	<i>jika mengklik akan menuju status pengiriman</i>
<i>Lacak Barang</i>	<i>Redirect Link</i>	<i>Lacak Barang</i>	<i>Jika mengklik akan menuju tampilan lacak barang</i>
<i>Cari transaksi</i>	<i>Text Field</i>	<i>Cari transaksi</i>	<i>Mengisi pencarian transaksi</i>

3.8.7 Halaman Edit Profile

LelangKu | Log In | Daftar

Follow | Hasibullah Magomedov | 

Ubah Profil Saya



Username: hasibullahMILLER
Email: hasibullah@gmail.com
Nama: Hasibullah Magomedov
Nomor HP: 08099209111
Provinsi: Jawa Barat
Kota: Bandung
Kecamatan: Cibiru
Kelurahan: Cipeudong
Alamat Lengkap: Komp. Mayura no.70
Kode Pos: 49007

Simpan

LelangKu
Latin Ipsum is simply dummy text of the printing and typesetting industry. Latin Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

Bantuan dan Panduan
[Syarat & Ketentuan](#)
[Kebijakan Privasi](#)

LelangKu
© 2022 PT Lelangku

Ikuti Kami
  

Id Objek	Jenis	Nama	Keterangan
<i>Username</i>	<i>Text Field</i>	<i>Kategori</i>	<i>Berisi Username dari pengguna</i>
<i>Email</i>	<i>Button</i>	<i>Status</i>	<i>Mengisi Email baru</i>
<i>Nama</i>	<i>Text Field</i>	<i>Nama</i>	<i>Mengisi Nama</i>
<i>Nomor HP</i>	<i>Text Field</i>	<i>Nomor HP</i>	<i>Mengisi Nomor Hp baru</i>
<i>Provinsi</i>	<i>Text Field</i>	<i>Provinsi</i>	<i>Mengisi nama Provinsi tempat tinggal</i>
<i>Kota</i>	<i>Text Field</i>	<i>Kota</i>	<i>Mengisi nama Kota tempat tinggal</i>
<i>Kecamatan</i>	<i>Text Field</i>	<i>Kecamatan</i>	<i>Mengisi nama Kecamatan tempat tinggal</i>
<i>Kelurahan</i>	<i>Text Field</i>	<i>Kelurahan</i>	<i>Mengisi nama Kelurahan tempat tinggal</i>
<i>Alamat Lengkap</i>	<i>Text Field</i>	<i>Alamat Lengkap</i>	<i>Mengisi Alamat Lengkap</i>
<i>Kode Pos</i>	<i>Text Field</i>	<i>Kode Pos</i>	<i>Mengisi Kode Pos</i>
<i>Simpan</i>	<i>Button</i>	<i>Simpan</i>	<i>Menyimpan Profile baru</i>
<i>Upload Foto</i>	<i>Button</i>	<i>Upload Foto</i>	<i>Mengupload foto profile baru</i>

3.8.8 Halaman Top Up

Top Up

Mau top-up berapa?

Metode Pembayaran

- BCA (11111)
- BSI (222222)
- Mandiri (1234543282)
- BRI (1231234)
- CIMB (1231231331)

Bayar**LelangKu**

This page is simply dummy text of the printing and typesetting industry. Lorem ipsum has been the industry's standard dummy text ever since the 1500s, when an anonymous printer took a galley of type and scrambled it to make a type specimen book.

Bantuan dan Panduan
[Syarat & Ketentuan](#)
[Kebijakan Privasi](#)


© 2023 PT Lelangku

Ikuti Kami!

Id Objek	Jenis	Nama	Keterangan
<i>Mau top-up Berapa?</i>	<i>Text Field</i>	<i>Mau top-up Berapa?</i>	<i>Mengisi jumlah nominal top up saldo yang diinginkan</i>
<i>Metode Pembayaran</i>	<i>RadioButt on</i>	<i>Metode Pembayaran</i>	<i>Memilih salah satu metode pembayaran yang tersedia</i>

3.8.9 Halaman Admin Transaksi**Good Morning, Admin**

Latest Transactions					
Name	Harga	Penjual	Status		
Redmi Note 11 Pro	Rp. 5000000	hosbulia SELLER	Bid		
Oppo A85	Rp. 3500000	hosbulia SELLER	Bid		
Redmi Note 9	Rp. 3000000	hosbulia SELLER	Bid		
Samsung S20	Rp. 7000000	hosbulia SELLER	Bid		
Asus Vivobook X412A	Rp. 7000000	hosbulia SELLER	Bid		
Acer Swift 7	Rp. 5000000	hosbulia SELLER	Bid		
Asus ProArt	Rp. 10000000	hosbulia SELLER	Bid		
Koper Polos Premium	Rp. 70000	hosbulia SELLER	Selesai		
Botol Tumbler Encello	Rp. 10000	hosbulia SELLER	Bid		
Borrong	Rp. 10000	test	Bid		
Borrong	Rp. 60000	test	Bid		

Id Objek	Jenis	Nama	Keterangan
<i>Nama</i>	<i>Text Field</i>	<i>Nama</i>	<i>Berisi Nama Barang</i>
<i>Harga</i>	<i>Text Field</i>	<i>Harga</i>	<i>Berisi Harga dari barang yang dilelang</i>
<i>Penjual</i>	<i>Text Field</i>	<i>Penjual</i>	<i>Berisi Nama dari Penjual barang yang dilelang</i>
<i>Status</i>	<i>Text Field</i>	<i>Status</i>	<i>Berisi Status dari barang yang dilelang</i>

3.8.10 Halaman Bank

The screenshot shows the 'Bank' section of the LelangKu Admin interface. On the left, there's a sidebar with links for Dashboard, My Account, Category, Bank (selected), Expedition, and Log Out. The main area has a header 'Good Morning, Admin'. Below it, a table lists existing banks with columns for No, Bank Name, Account Number, and Action (Delete). A modal window titled 'Edit Bank' is open, showing fields for Bank Name (BSI) and Account Number (222222), with a 'Submit' button. At the bottom, there's a footer with the text '©2022 LelangKu. All rights reserved.'

Id Objek	Jenis	Nama	Keterangan
Bank Name	Text Field	Bank Name	Berisi Nama Bank yang sudah ada
Account Number	Text Field	Account Number	Berisi Nomor rekening dari bank
Delete	Button	Delete	Menghapus Bank dan Account Number yang ada

Id Objek	Jenis	Nama	Keterangan
Bank Name	Text Field	Bank Name	Mengisi Nama Bank baru
Account Number	Text Field	Account Number	Mengisi Nomor rekening dari bank baru yang akan dibuat
Submit	Button	Submit	Menambahkan Bank dan Account Number

Id Objek	Jenis	Nama	Keterangan
Bank Name	Text Field	Bank Name	Mengubah Nama Bank yang sudah ada
Account Number	Text Field	Account Number	Mengubah Nomor rekening dari bank yang sudah ada
Submit	Button	Submit	Mengedit Bank dan Account Number yang sudah ada

3.8.11 Halaman Expedition

The screenshot shows the 'Expedition' section of the LelangKu Admin interface. On the left, there's a sidebar with links for Dashboard, My Account, Category, Bank, Expedition (selected), and Log Out. The main area has a header 'Good Morning, Admin'. Below it, a table lists existing delivery services with columns for No, Name, and Action (Delete). A modal window titled 'Edit name' is open, showing a field for 'kategori' (Enter name) and a 'Submit' button. At the bottom, there's a footer with the text '©2022 LelangKu. All rights reserved.'

Id Objek	Jenis	Nama	Keterangan
<i>Nama</i>	<i>Text Field</i>	<i>Nama</i>	<i>Menampilkan Nama Ekspedisi yang sudah ada</i>
<i>Delete</i>	<i>Button</i>	<i>Delete</i>	<i>Menghapus Ekspedisi yang sudah ada</i>
<i>Kategori</i>	<i>Text Field</i>	<i>Kategori</i>	<i>Menambahkan Ekspedisi baru</i>

3.8.12 Halaman Kategori

No	Nama	Action
1	Laptop	Delete
2	Fashion	Delete
3	Handphone	Delete
4	Lainnya	Delete

Modal for editing 'Laptop':

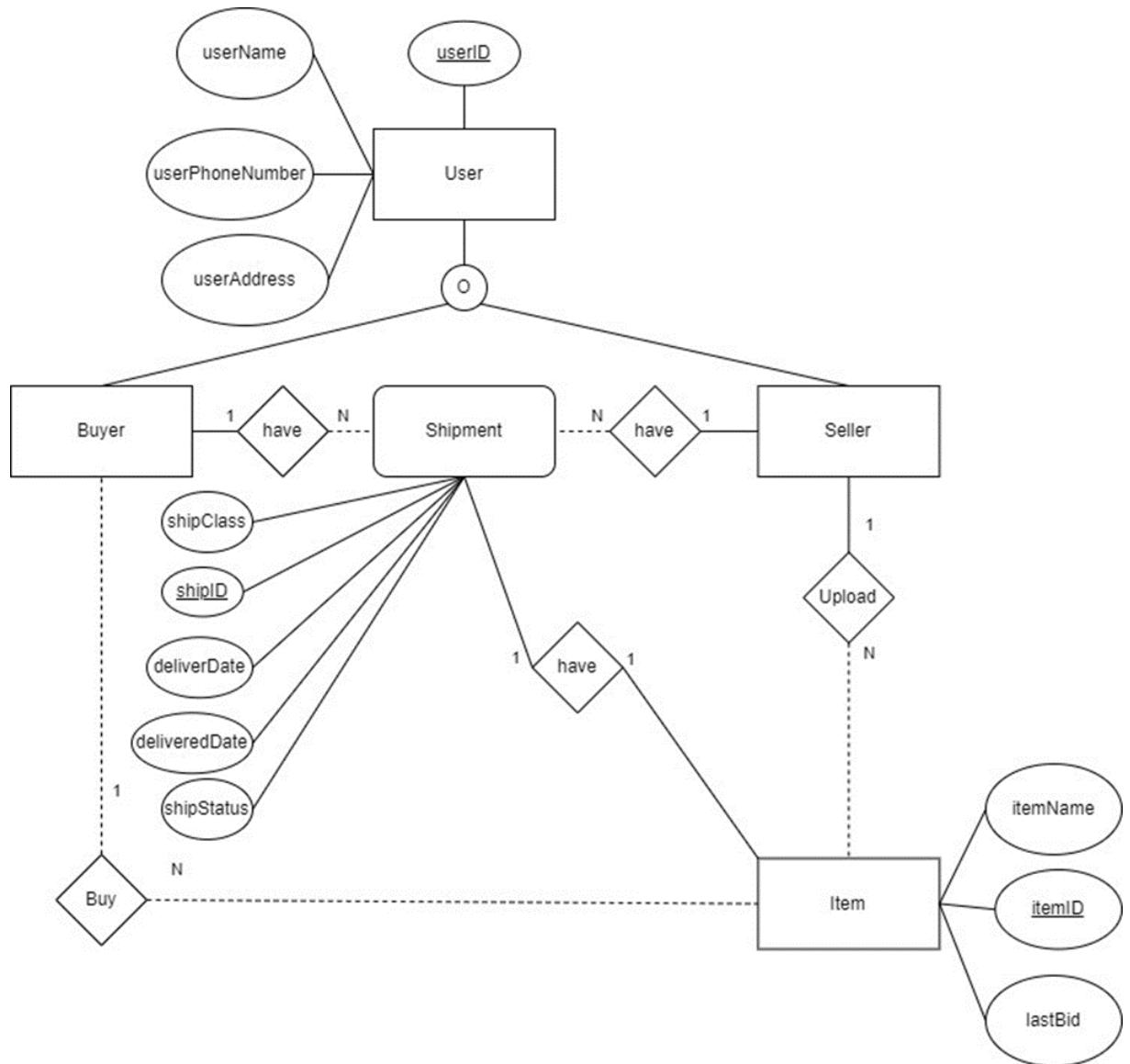
- Input field: kategori (Laptop)
- Submit button

Id Objek	Jenis	Nama	Keterangan
<i>Nama</i>	<i>Text Field</i>	<i>Nama</i>	<i>Menampilkan Nama Kategori yang sudah ada</i>
<i>Delete</i>	<i>Button</i>	<i>Delete</i>	<i>Menghapus Kategori yang sudah ada</i>

Id Objek	Jenis	Nama	Keterangan
<i>Kategori</i>	<i>Text Field</i>	<i>Kategori</i>	<i>Mengisi Nama Kategori baru</i>
<i>Submit</i>	<i>Button</i>	<i>Submit</i>	<i>Menambahkan Kategori baru</i>

Id Objek	Jenis	Nama	Keterangan
<i>Kategori</i>	<i>Text Field</i>	<i>Kategori</i>	<i>Mengedit Nama Kategori yang sudah ada</i>
<i>Submit</i>	<i>Button</i>	<i>Submit</i>	<i>Mengedit Kategori yang sudah ada</i>

3.9 Perancangan Representasi Persistensi Kelas



4. Matriks Kerunutan

Requirement	Use Case Terkait	Kelas
FR-01	Sign up	User
FR-02	Sign in	User,admin
FR-03	Upload	Seller
FR-04	Bid item	buyer
FR-05	Bid info	buyer
FR-06	Category	admin, item
FR-07	Progress check	seller
FR-08	Bid progress	Buyer
FR-09	Shipment	Transaction, expedition

