

**LAPORAN
TUGAS BESAR**

PEMROGRAMAN BERORIENTASI OBJEK



Kelas : SI06B
Kelompok : 7

No	NIM	Nama Mahasiswa
1	2211103037	Delia Pebina Br Ginting
2	2211103052	Wahyu Nurfida A
3	2211103056	Tanzil Aziim
4	2211103059	Maghda Luqyana
5	2211103065	Frisca Destar Suhertal

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
SEMESTER GANJIL 2022/2023**

DAFTAR ISI

1	PENDAHULUAN	2
2	RANCANGAN APLIKASI.....	2
2.1	FUNGSIONALITAS PRODUK	3
2.2	UML	5
2.2.1	Use Case	5
2.2.2	Class Diagram	6
2.2.3	Activity Diagram	6
2.2.4	Sequence Diagram	7
3	IMPLEMENTASI APLIKASI	8
3.1	LIST DAN DESKRIPSI END POINT.....	8
3.2	IMPLEMENTASI CODE END POINT	9
3.3	LINK GITHUB.....	10

1 Pendahuluan

Dalam mengelola keuangan, terutama pada skala kecil hingga menengah, penting bagi pemilik atau administrator di dalam suatu perusahaan untuk memiliki alat yang efektif dan efisien untuk mencatat, mengelola, dan memantau arus keuangan. Dengan perkembangan teknologi, aplikasi keuangan menjadi solusi yang tepat untuk membantu administrasi keuangan menjadi lebih terstruktur dan terorganisir. Terlebih lagi bagi kami yang memiliki brand Breadfast yang bergerak dalam bisnis di bidang FnB. Aplikasi keuangan yang dirancang khusus untuk memenuhi kebutuhan kami adalah aplikasi keuangan yang memfokuskan pada manajemen pemasukan, pengeluaran, dan saldo. Aplikasi ini dirancang dengan tujuan memberikan kemudahan bagi pemilik atau administrator, yang dalam hal ini disebut sebagai admin, untuk mengakses dan mengelola data keuangan secara terpusat. Dengan satu user, yaitu admin, memiliki hak akses penuh, aplikasi ini menyediakan fitur-fitur krusial yang mencakup pencatatan pemasukan, pengeluaran, serta pemantauan saldo. Pada bagian pemasukan dan pengeluaran, admin diberikan keleluasaan untuk melakukan sejumlah tindakan penting. Mulai dari menambah data transaksi, mengubah data yang sudah ada, melakukan update terhadap informasi transaksi, hingga menghapus data yang tidak relevan. Kemudahan-kemudahan ini memastikan bahwa admin dapat mengelola data keuangan dengan fleksibilitas dan akurasi. Selain itu, aplikasi ini juga memungkinkan admin untuk menyimpan data dengan aman, memastikan bahwa setiap transaksi yang dicatat tersimpan dengan baik dan dapat diakses kembali sewaktu-waktu. Proses penyimpanan ini tidak hanya memberikan keamanan terhadap data, tetapi juga memastikan keterjangkauan informasi untuk analisis dan keperluan lainnya. Pada aspek mengakses saldo, aplikasi ini menawarkan kemudahan bagi admin untuk melihat jumlah saldo secara transparan. Dengan informasi saldo yang dapat diakses dengan cepat, admin dapat mengambil keputusan yang tepat dan strategis dalam mengelola keuangan entitasnya.

Aplikasi catatan keuangan kasir ini memiliki beberapa kegunaan atau manfaat lain, antara lain:

- Memudahkan admin dalam mencatat pemasukan dan pengeluaran perusahaan.
- Membantu admin dalam memantau kondisi keuangan perusahaan.
- Membantu admin dalam membuat laporan keuangan perusahaan.

2 Rancangan Aplikasi

2.1 Fungsionalitas Produk

Nama	:	Login
Deskripsi	:	Fungsi login dalam aplikasi catatan keuangan kasir digunakan untuk memverifikasi identitas pengguna sebelum mengakses aplikasi. Fungsi ini menerima dua parameter, yaitu username dan password. Jika username dan password yang dimasukkan cocok dengan data pengguna yang tersimpan di database, maka pengguna akan berhasil login dan mendapatkan token. Token ini digunakan untuk autentikasi pengguna dalam setiap request yang dilakukan.

Nama	:	Mengakses Keuangan
Deskripsi	:	<ul style="list-style-type: none">• Fungsi menambah data pemasukan digunakan untuk menambahkan data pemasukan baru ke dalam database. Data pemasukan yang harus dimasukkan adalah ID pemasukan, tanggal dan nominal.• Fungsi menyimpan data pemasukan digunakan untuk menyimpan data pemasukan ke dalam database. Fungsi ini dapat digunakan untuk menambahkan data pemasukan baru, mengubah data pemasukan, atau memperbarui data pemasukan.• Fungsi mengubah data pemasukan digunakan untuk mengubah data pemasukan yang sudah ada di database. Fungsi ini menerima data pemasukan sebagai parameter. Data pemasukan yang harus dimasukkan adalah ID pemasukan, tanggal, dan nominal.• Fungsi update data pemasukan digunakan untuk memperbarui data pemasukan yang sudah ada di database. Fungsi ini menerima data pemasukan sebagai parameter. Data pemasukan yang harus dimasukkan adalah ID pemasukan, field yang ingin diubah, dan nilai baru• Fungsi menghapus data pemasukan digunakan untuk menghapus data pemasukan dari database. Fungsi ini menerima ID pemasukan sebagai parameter.

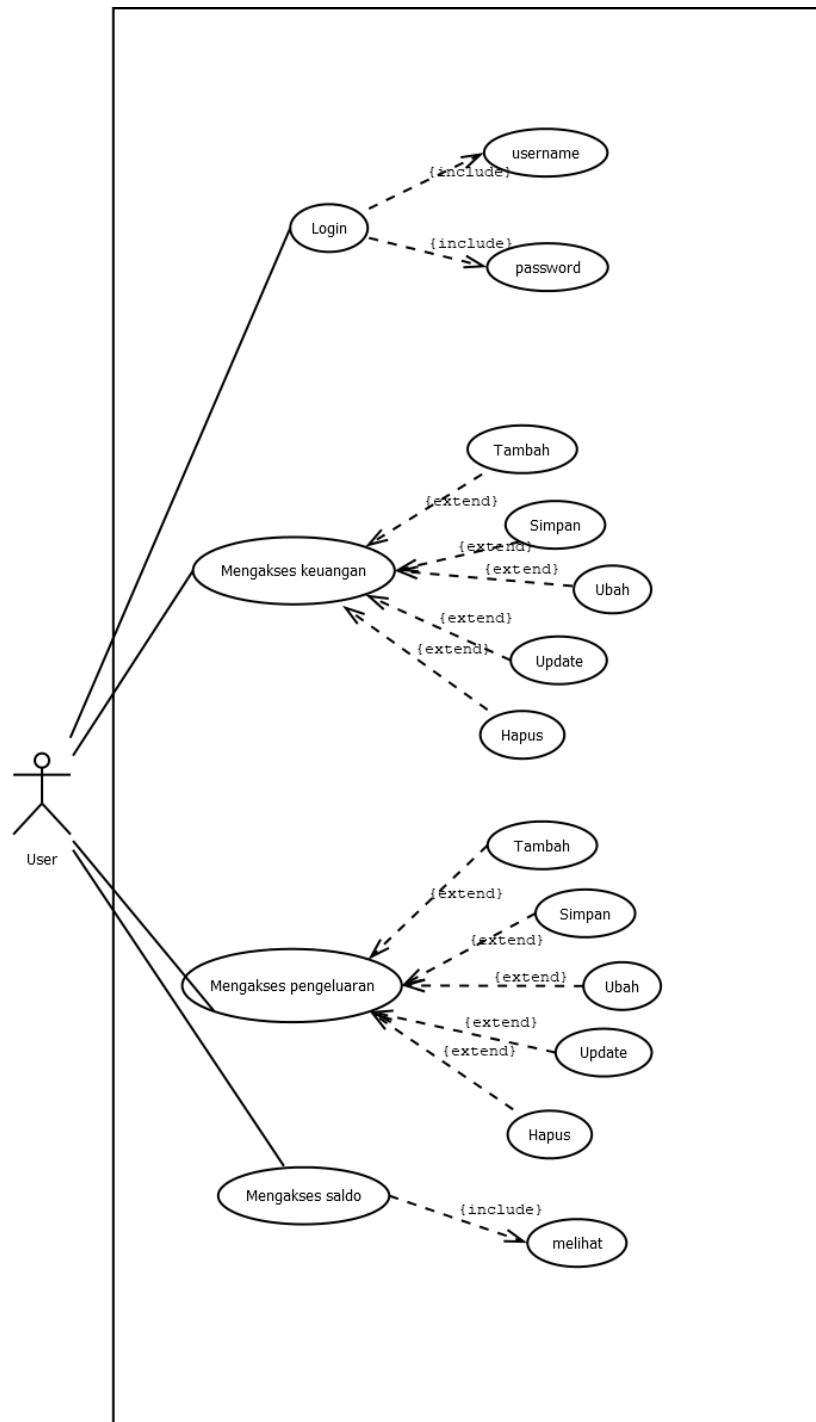
Nama	:	Mengakses Pengeluaran
Deskripsi	:	<ul style="list-style-type: none">• Fungsi menambah data pengeluaran digunakan untuk menambahkan data pengeluaran baru ke dalam database. Data pengeluaran yang harus dimasukkan adalah ID pengeluaran, tanggal dan nominal.

	<ul style="list-style-type: none"> • Fungsi menyimpan data pengeluaran digunakan untuk menyimpan data pengeluaran ke dalam database. Fungsi ini dapat digunakan untuk menambahkan data pengeluaran baru, mengubah data pengeluaran, atau memperbarui data pengeluaran. • Fungsi mengubah data pengeluaran digunakan untuk mengubah data pengeluaran yang sudah ada di database. Fungsi ini menerima data pengeluaran sebagai parameter. Data pemasukan yang harus dimasukkan adalah ID pengeluaran, tanggal, dan nominal. • Fungsi update data pengeluaran digunakan untuk memperbarui data pengeluaran yang sudah ada di database. Fungsi ini menerima data pengeluaran sebagai parameter. Data pengeluaran yang harus dimasukkan adalah ID pengeluaran, field yang ingin diubah, dan nilai baru • Fungsi menghapus data pengeluaran digunakan untuk menghapus data pengeluaran dari database. Fungsi ini menerima ID pengeluaran sebagai parameter.
--	---

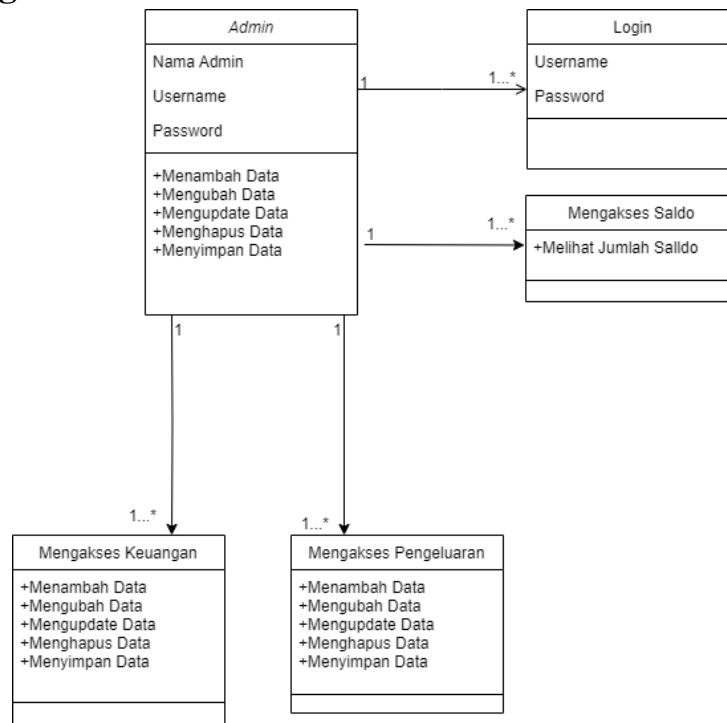
Nama	:	Mengakses saldo
Deskripsi	:	Fungsi ini dapat digunakan untuk melihat jumlah saldo toko secara real-time. Cara kerja fungsi adalah melakukan operasi pengurangan antara pemasukan dan pengeluaran kemudian hasil tersebut akan menjadi saldo terkini.

2.2 UML

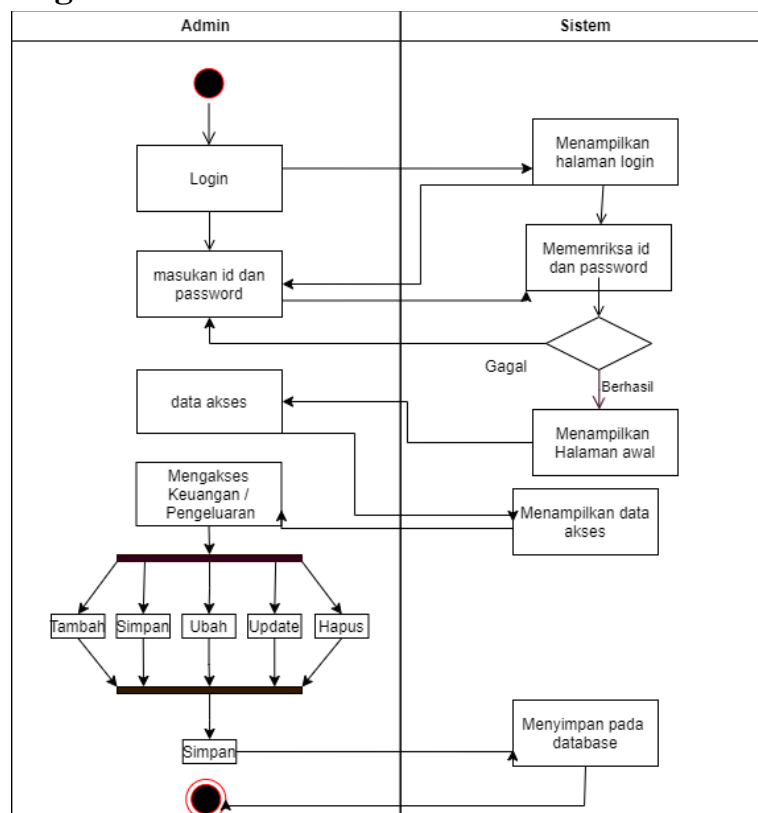
2.2.1 Use Case



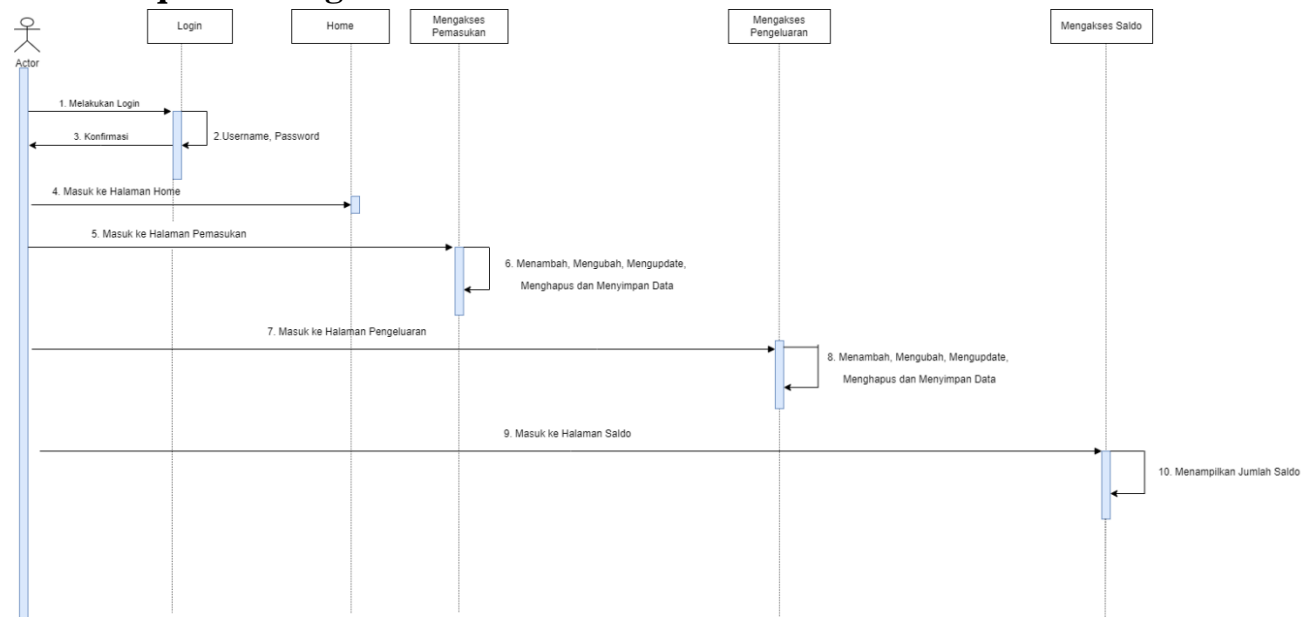
2.2.2 Class Diagram



2.2.3 Activity Diagram



2.2.4 Sequence Diagram



3 Implementasi Aplikasi

3.1 List dan Deskripsi End Point

- **User API**
 - **Register User**
Endpoint: POST /api/user
 - **Login User**
Endpoint: POST /api/auth/login
 - **Get User**
Endpoint: GET /api/users/current
 - **Update User**
Endpoint: PATCH /api/users/current
 - **Logout User**
Endpoint: DELETE /api/auth/logout
- **Income API**
 - **Create Income**
Endpoint : POST /api/incomes
 - **Update Income**
Endpoint : PUT /api/incomes/{idIncome}
 - **Get Income**
Endpoint : GET /api/incomes/{idIncome}
 - **List Income**
Endpoint : GET /api/incomes/{idIncome}
 - **Remove Income**
Endpoint : DELETE /api/incomes/{idIncome}
- **Outcome API**
 - **Create Outcome**
Endpoint : POST /api/outcomes
 - **Update Outcome**
Endpoint : PUT /api/outcomes/{idOutcome}
 - **Get Outcome**
Endpoint : GET /api/outcomes/{idOutcome}
 - **List Outcome**
Endpoint : GET /api/incomes/{idIncome}

- **Remove Outcome**

Endpoint : DELETE /api/outcomes/{idOutcome}

3.2 Implementasi Code End Point

User Entity

```
package tubespbokelompok7.restful.entity;

import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.Id;
import jakarta.persistence.Table;
import lombok.AllArgsConstructor;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;

@Getter
@Setter
@AllArgsConstructor
@NoArgsConstructor
@Entity
@Table(name = "users")
public class User {
    @Id
    private String username;
    private String password;
    private String name;
    private String token;
    @Column(name = "token_expired_at")
    private Long tokenExpiredAt;
}
```

Income Entity

```
package tubespbokelompok7.restful.entity;

import jakarta.persistence.Entity;
import jakarta.persistence.Id;
import jakarta.persistence.Table;
import lombok.AllArgsConstructor;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;

import java.math.BigDecimal;

@Getter
@Setter
@AllArgsConstructor
@NoArgsConstructor
@Entity
@Table(name = "incomes")
```

```

public class Income {
    @Id
    private String id;
    private BigDecimal balance;
    private String itemName;
    private BigDecimal price;
    private int amount;
}

```

Outcome Entity

```

package tubespbokelompok7.restful.entity;

import jakarta.persistence.Entity;
import jakarta.persistence.Id;
import jakarta.persistence.Table;
import lombok.AllArgsConstructor;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;

import java.math.BigDecimal;

@Getter
@Setter
@AllArgsConstructor
@NoArgsConstructor
@Entity
@Table(name = "outcomes")
public class Outcome {
    @Id
    private String id;
    private BigDecimal balance;
    private String outcomeName;
    private BigDecimal outcomeTotal;
}

```

3.3 Link github

<https://github.com/tanzilaziim/TugasBesarPBOKel7.git> (belum selesai)