**LAPORAN TUGAS BESAR**

**Pemrograman Web**



**Disusun Oleh Kelompok 4:**

**Muh. Riflan 22650160**

**Dyah Chandra Priyankha 22650156**

**Amelia Nur Avivah 22650169**

**Saldin 22650177**

**Gavid Rizal 22650172**

**Afrizal 22650186**

**Alif Raihan 22650151**

**Jestia 22650167**

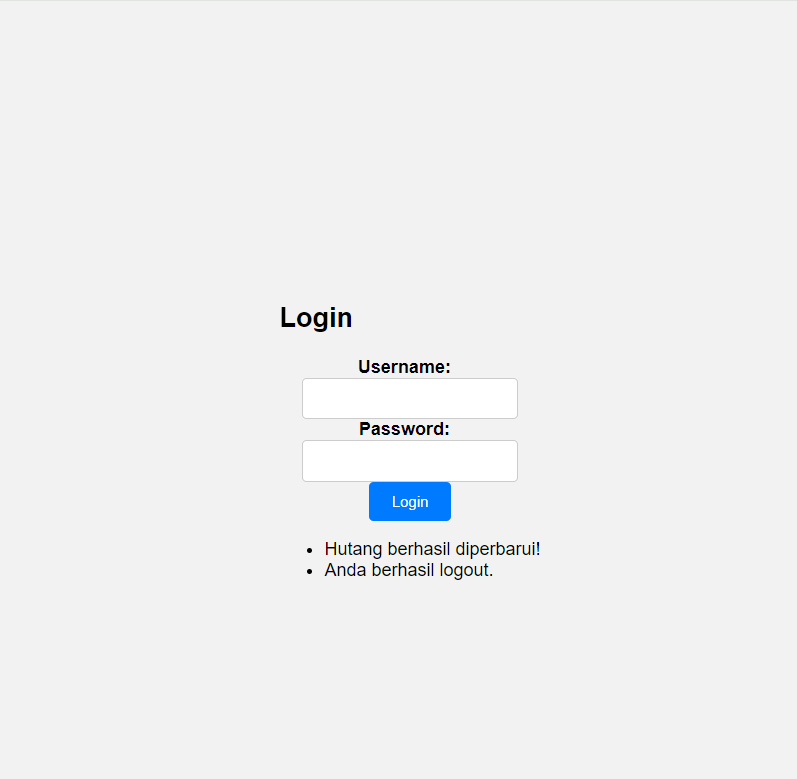
**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

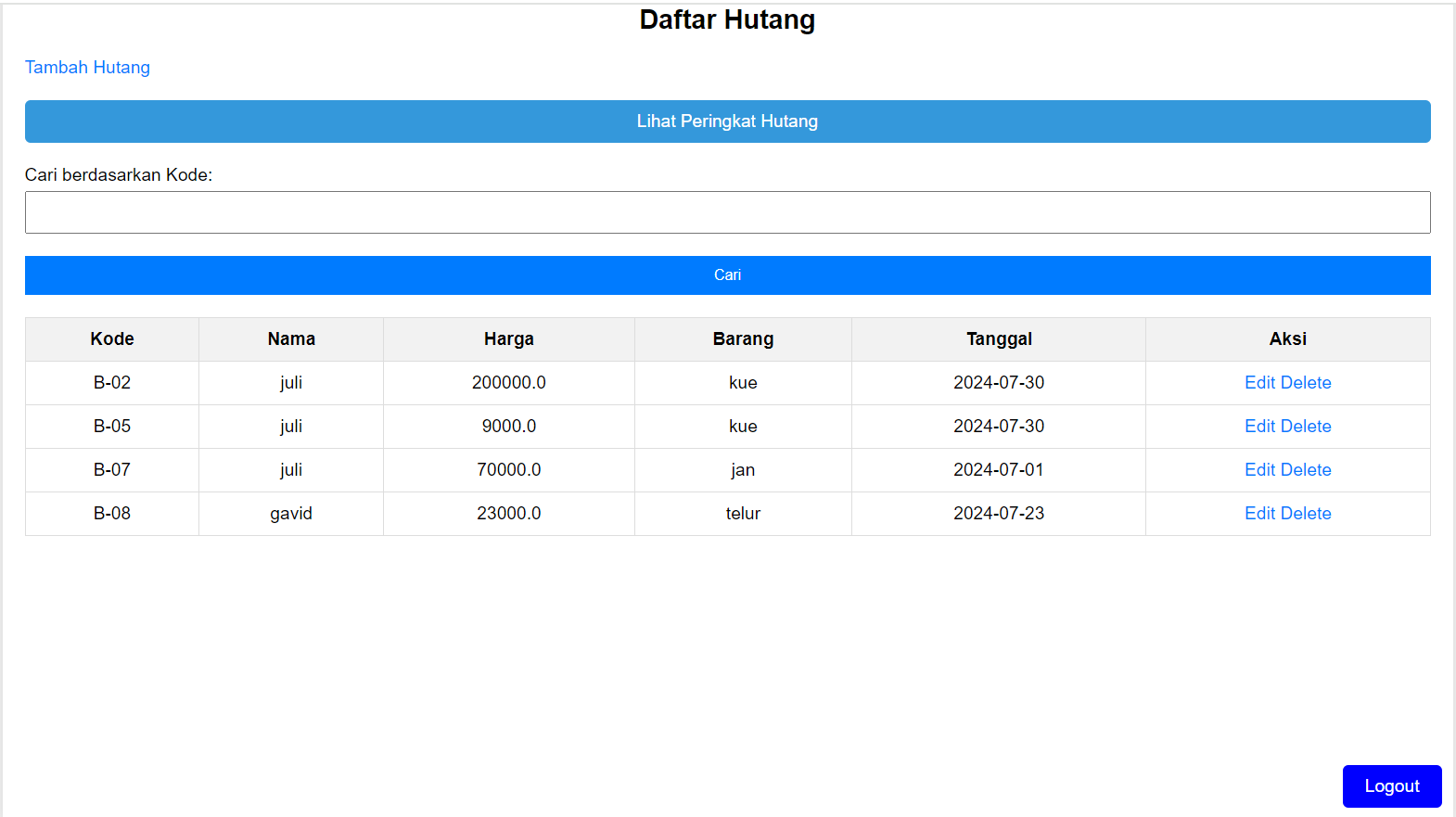
**UNIVERSITAS  DAYANU IKHSANUDDIN PASARWAJO**

**2024**

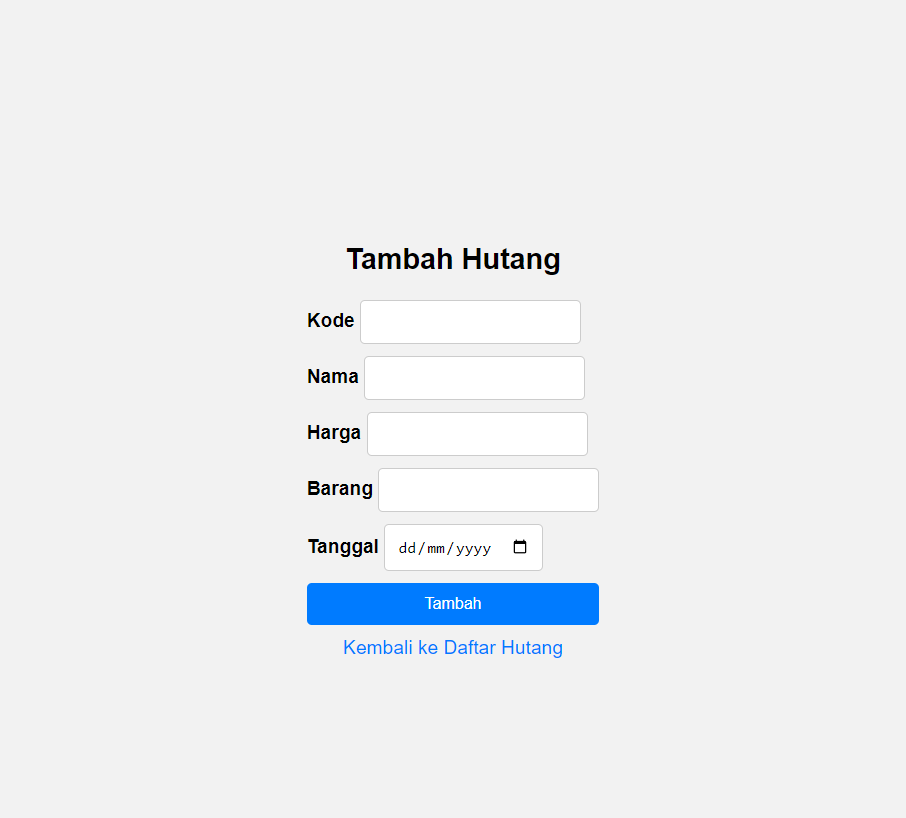
1. **Wireframe Buku Hutang**
2. **Menu Login**



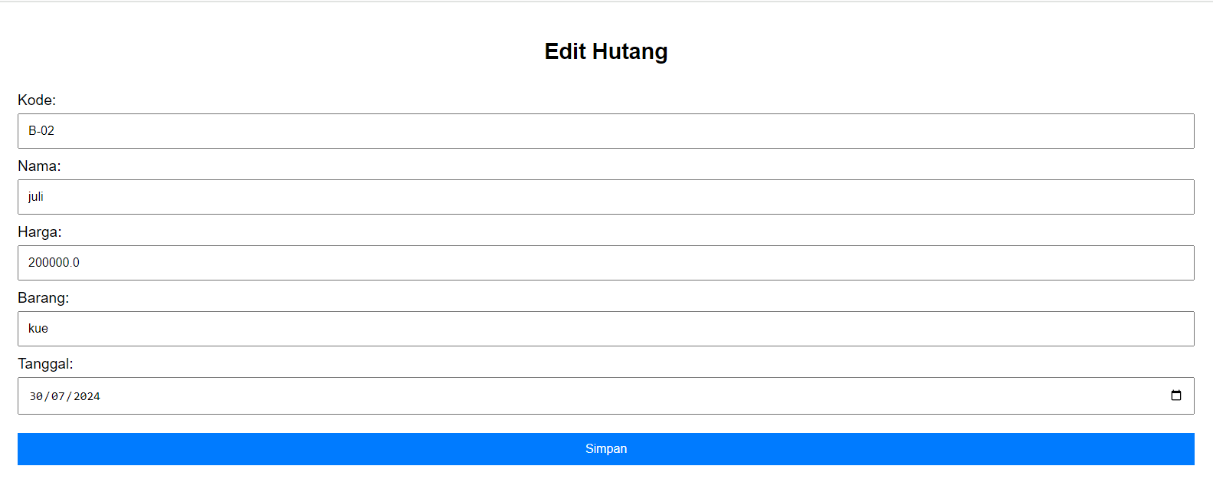
1. **Menu Daftar Hutang**

****

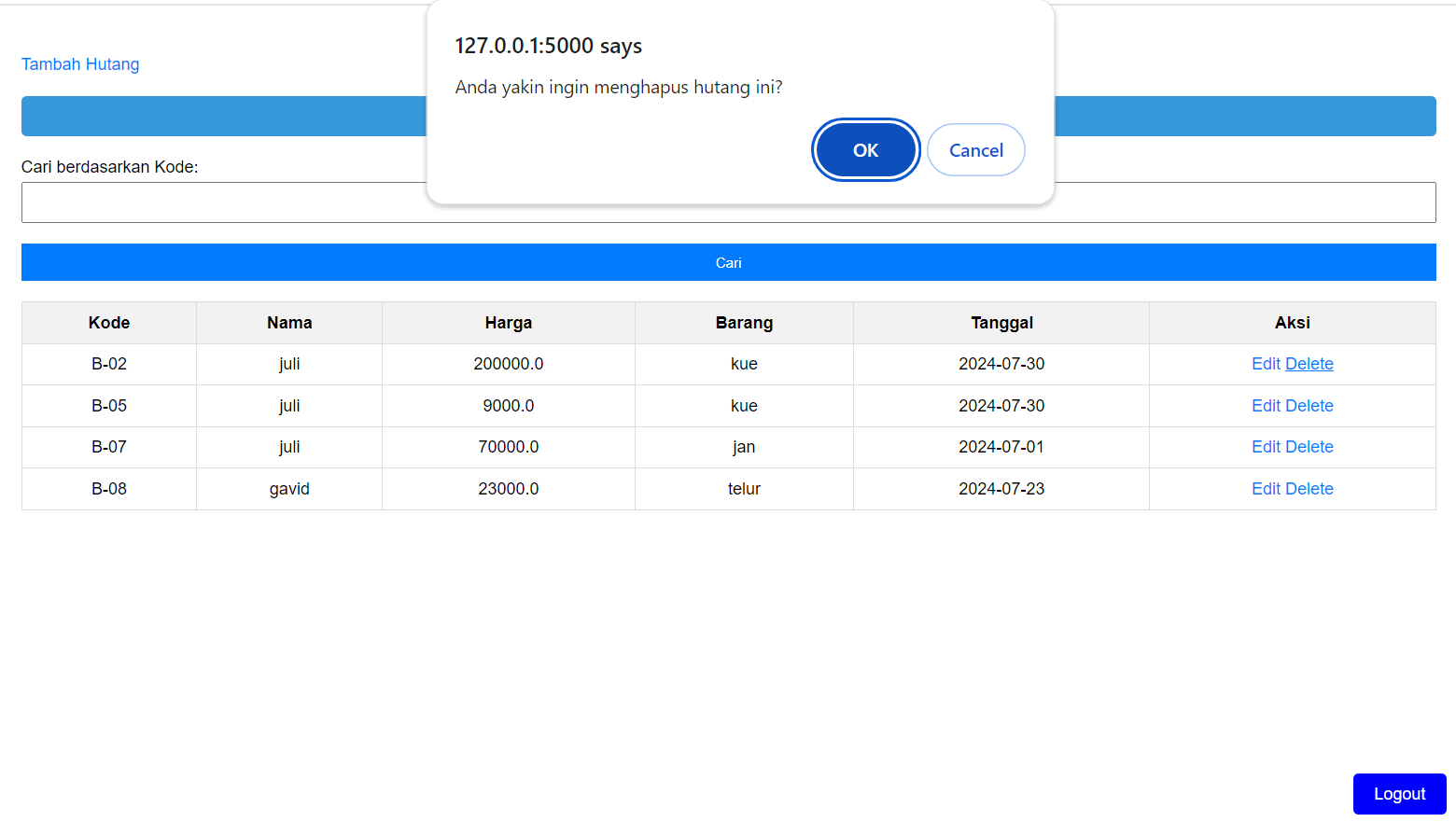
1. **Menu Tambah Hutang**

****

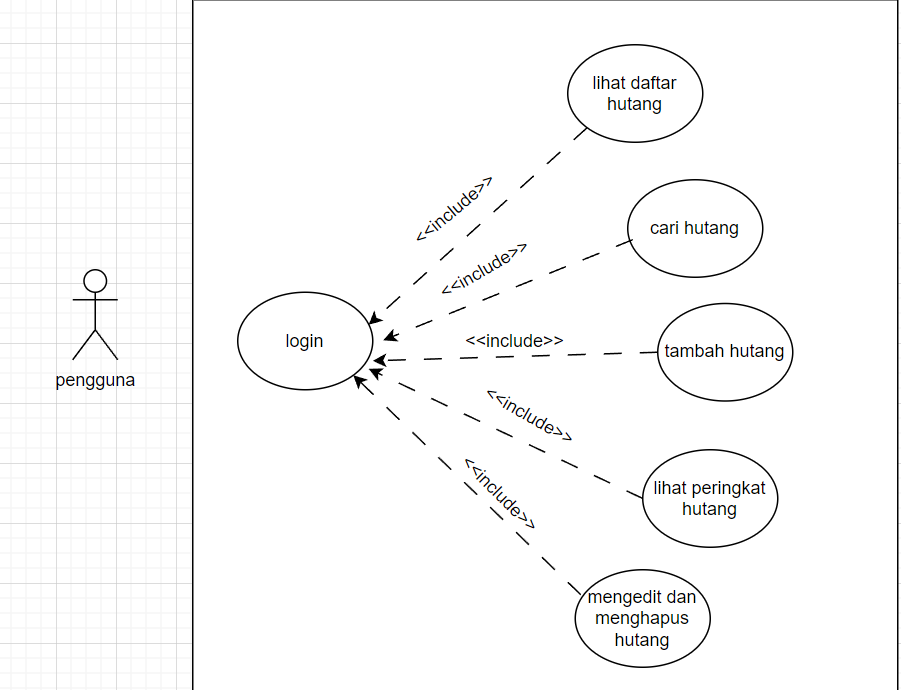
1. **Menu Peringkat Hutang**
2. **Menu Edit Hutang**

****

1. **Tampilan Hapus Data Hutang**

****

1. **Desain use case, Sequence diagram Dan Class Diagram.**
   * 1. **Use Case Diagram**

****

Berikut adalah penjelasan lengkap mengenai setiap elemen dalam diagram tersebut:

* Pengguna (Aktor): Pengguna adalah entitas yang akan menggunakan sistem pengelolaan hutang. Dalam diagram ini, pengguna diwakili oleh aktor bernama "pengguna".
  + 1. **Class Diagram**

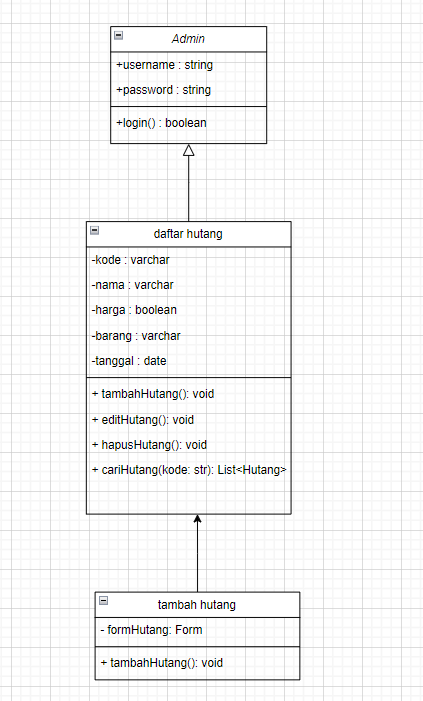
****

Diagram kelas ini menjelaskan struktur dari sistem pengelolaan hutang dengan tiga kelas utama: Admin, `daftar hdaftar hutang, dan `tambahkan hutambah hutang. Menjadi

1. **Kelas Admin**
2. Atribut:

* username: string: Menyimpan nama pengguna untuk login.
* password: string: Menyimpan kata sandi untuk login

1. Metode:

* login(): boolean: Memverifikasi kredensial pengguna dan mengembalikan nilai boolean yang menunjukkan keberhasilan login.a

1. **Kelas Daftar Hutang**
2. Atribut:

* kode: varchar: Menyimpan kode hutang sebagai pengenal unik.
* nama: varchar: Menyimpan nama orang yang berhutang.
* harga: boolean: Menyimpan jumlah hutang.
* barang: varchar: Menyimpan barang yang terkait dengan hutang.
* tanggal: date: Menyimpan tanggal transaksi hutang.

1. Metode:

* tambahHutang(): void: Menambahkan hutang baru ke dalam daftar.
* editHutang(): void: Mengedit informasi hutang yang sudah ada.
* hapusHutang(): void: Menghapus hutang dari daftar.
* cariHutang(kode: str): List<Hutang>: Mencari hutang berdasarkan kode dan mengembalikan daftar hutang yang ditemukan.

1. Kelas Tambah Hutang
2. Atribut

* formHutang: Form: Menyimpan form untuk menambah hutang.

1. Metode

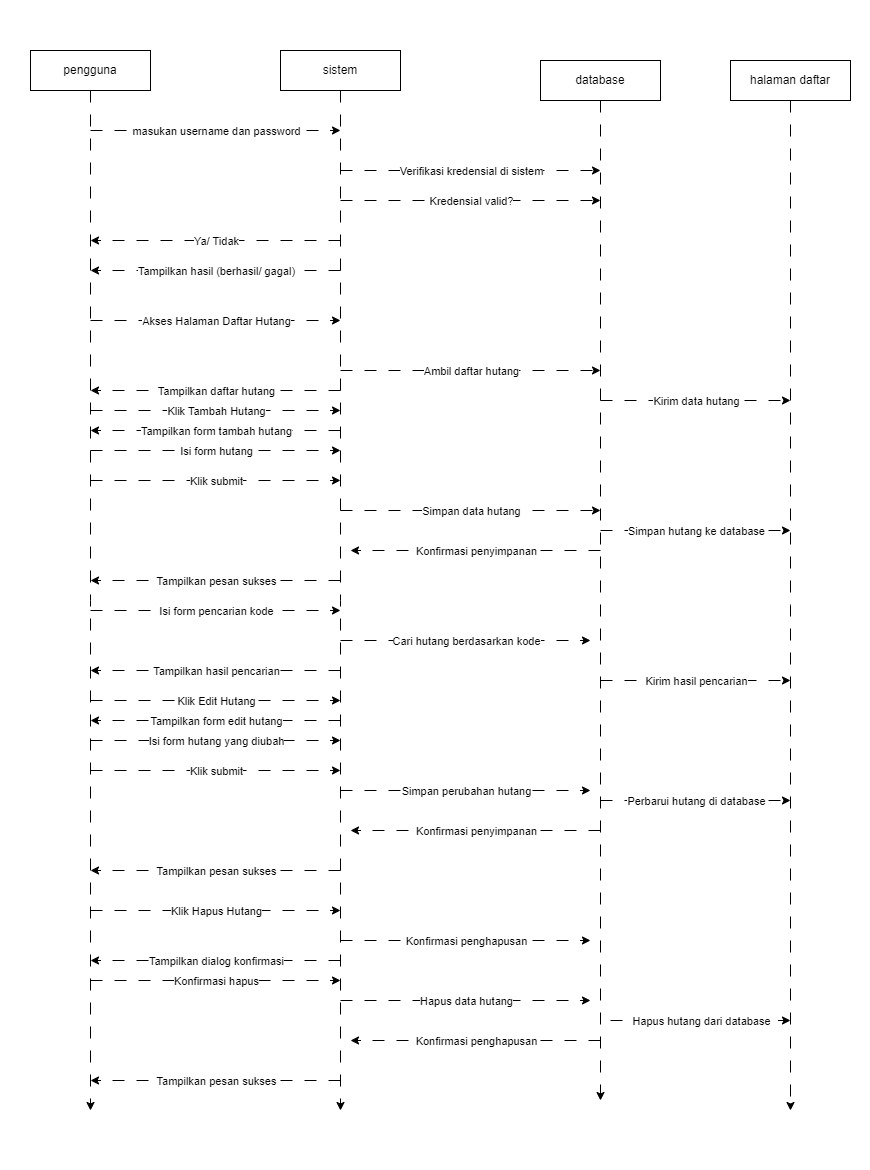
* tambahHutang(): void: Memproses form dan menambah hutang baru ke dalam daftar.

1. **Hubungan Antar Kelas**:

* Kelas Admin memiliki hubungan asosiasi dengan Kelas Daftar Hutang yang menunjukkan bahwa admin memiliki akses untuk mengelola daftar hutang setelah berhasil login.
* Kelas Tambah Hutang memiliki hubungan asosiasi dengan Kelas Daftar Hutang yang menunjukkan bahwa proses penambahan hutang menggunakan metode dari kelas Daftar Hutang.

1. Penjelasan Detail:

* Admin adalah pengguna yang memiliki hak akses untuk masuk ke sistem dengan menggunakan nama pengguna dan kata sandi. Setelah login berhasil, admin dapat mengelola daftar hutang.
* Daftar Hutang adalah inti dari sistem manajemen hutang yang menyimpan semua data hutang beserta informasi terkait. Kelas ini memiliki metode untuk menambah, mengedit, menghapus, dan mencari hutang.
* Tambah Hutang adalah fitur yang memungkinkan admin menambah hutang baru ke dalam sistem melalui form yang diisi dengan data hutang baru.
  + 1. **Sequence Diagram**



Gambar diagram ini menunjukkan diagram urutan (sequence diagram) untuk sistem pengelolaan hutang. Berikut adalah penjelasan dari masing-masing langkah dalam diagram tersebut:

* Masukkan username dan password : Pengguna memasukkan kredensial mereka dan mengirimkannya ke sistem.
* Verifikasi kredensial di sistem : Sistem memverifikasi kredensial pengguna dengan mengirimkan permintaan ke database.
* Kredensial valid: Database memeriksa kredensial dan mengirimkan hasilnya kembali ke sistem.
* Tampilkan hasil (berhasil/gagal): Sistem menampilkan pesan kepada pengguna apakah login berhasil atau gagal.
* Akses Halaman Daftar Hutang : Pengguna yang berhasil login meminta akses ke halaman daftar hutang.
* Ambil daftar hutang: Sistem mengambil data daftar hutang dari database.
* Kirim data hutang: Database mengirimkan data daftar hutang ke sistem.
* Tampilkan daftar hutang: Sistem menampilkan daftar hutang kepada pengguna.
* Klik Tambah Hutang: Pengguna mengklik tombol untuk menambah hutang baru.
* Tampilkan form tambah hutang: Sistem menampilkan formulir untuk menambah hutang kepada pengguna.
* Isi form hutang: Pengguna mengisi formulir dengan data hutang baru.
* Klik submit : Pengguna mengirimkan formulir yang telah diisi ke sistem.
* Simpan data hutang : Sistem menyimpan data hutang baru ke database.
* Simpan hutang ke database: Database menyimpan data hutang baru.
* Konfirmasi penyimpanan :Database mengirimkan konfirmasi penyimpanan ke sistem.
* Tampilkan pesan sukses :Sistem menampilkan pesan sukses kepada pengguna.
* Isi form pencarian kode : Pengguna mengisi formulir untuk mencari hutang berdasarkan kode.
* Cari hutang berdasarkan kode : Sistem mencari hutang berdasarkan kode yang diberikan dengan mengirimkan permintaan ke database.
* Kirim hasil pencarian : Database mengirimkan hasil pencarian ke sistem.
* Tampilkan hasil pencarian: Sistem menampilkan hasil pencarian kepada pengguna.
* Klik Edit Hutang : Pengguna mengklik tombol untuk mengedit hutang.
* Tampilkan form edit hutang : Sistem menampilkan formulir untuk mengedit hutang kepada pengguna.
* Isi form hutang yang diubah : Pengguna mengisi formulir dengan data hutang yang telah diubah.
* Klik submit : Pengguna mengirimkan formulir yang telah diisi ke sistem.
* Simpan perubahan hutang : Sistem menyimpan perubahan hutang ke database.
* Perbarui hutang di database : Database memperbarui data hutang yang diubah.
* Konfirmasi penyimpanan : Database mengirimkan konfirmasi penyimpanan ke sistem.
* Tampilkan pesan sukses : Sistem menampilkan pesan sukses kepada pengguna.
* Klik Hapus Hutang : Pengguna mengklik tombol untuk menghapus hutang.
* Tampilkan dialog konfirmasi : Sistem menampilkan dialog konfirmasi penghapusan kepada pengguna.
* Konfirmasi hapus : Pengguna mengkonfirmasi penghapusan hutang.
* Hapus data hutang : Sistem mengirimkan permintaan penghapusan hutang ke database.
* Hapus hutang dari database : Database menghapus data hutang.
* Konfirmasi penghapusan : Database mengirimkan konfirmasi penghapusan ke sistem.
* Tampilkan pesan sukses : Sistem menampilkan pesan sukses kepada pengguna.