

**UJIAN AKHIR SEMESTER PEMROGRAMAN BERBASIS OBJEK**

**Mata Kuliah :**

Pemrograman Berbasis Objek

**Oleh :**

Faiz Maulana

24091397103

2024 C



**PROGRAM STUDI D4 MANAJEMEN INFOMATIKA**

**FAKULTAS VOKASI**

**UNIVERSITAS NEGERI SURABAYA**

**2025**

## DAFTAR ISI

DAFTAR ISI .....	ii
DAFTAR GAMBAR .....	iii
DAFTAR TABLE .....	iv
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang .....	1
1.2. Tujuan Pembuatan Aplikasi .....	1
1.3. Ruang Lingkup Aplikasi .....	1
BAB II PEMBAHASAN .....	2
2.1. Konsep OOP .....	2
2.2. Encapsulation.....	2
2.3. Inheritance.....	3
2.4. Polymorphism .....	4
BAB III PERANCANGAN SISTEM.....	5
3.1. Deskripsi Umum Sistem .....	5
3.2. Diagram Class .....	5
3.3. Relasi Antar Class .....	7
BAB IV IMPLEMENTASI APLIKASI.....	8
4.1. Fitur Aplikasi .....	8
4.2. Tampilan Aplikasi .....	8
4.3. Alur Aplikasi.....	10
BAB V PENUTUP.....	12
5.1. Kesimpulan .....	12

## DAFTAR GAMBAR

Gambar 2. 1 Gambar Kode Program Atribut Dengan Akses Protected .....	2
Gambar 2. 2 Gambar Kode Program Data Dibungkus Dalam Class .....	2
Gambar 2. 3 Gambar Kode Program Akses Data Melalui Method .....	3
Gambar 2. 4 Gambar Kode Program Pewarisan User .....	3
Gambar 2. 5 Gambar Kode Program Pewarisan Ruangan Kelas.....	3
Gambar 2. 6 Gmbar Kode Program Pewarisan Ruangan Rapat .....	3
Gambar 2. 7 Gambar Kode Program Method info() pada class Ruangan .....	4
Gambar 2. 8 Gmbar Kode Program Method info() pada class RuangKelas.....	4
Gambar 2. 9 Gambar Kode Program Method info() pada class RuangRapat.....	4
Gambar 3. 1 Gambar Class Diagram .....	5
Gambar 4. 1 Gambar Form Data Peminjam .....	8
Gambar 4. 2 Gambar Pilih Ruangan dan Tombol Tambah Reservasi.....	9
Gambar 4. 3 Gambar Daftar Reservasi dan Tombol Batalkan Reservasi .....	9
Gambar 4. 4 Gambar Alur Aplikasi .....	10

## DAFTAR TABLE

Table 3. 1 Table Relasi Inheritance .....	7
Table 3. 2 Table Relasi Association .....	7
Table 3. 3 Table Relasi Dependency .....	7

## **BAB I PENDAHULUAN**

### **1.1. Latar Belakang**

Perkembangan teknologi informasi saat ini telah mendorong berbagai institusi pendidikan untuk memanfaatkan sistem berbasis komputer dalam mendukung kegiatan operasional sehari-hari. Salah satu kebutuhan yang sering dijumpai di lingkungan perguruan tinggi adalah proses peminjaman atau reservasi ruangan, baik untuk kegiatan perkuliahan, rapat, maupun kegiatan akademik lainnya. Pada praktiknya, proses reservasi ruangan yang masih dilakukan secara manual berpotensi menimbulkan berbagai permasalahan, seperti bentrok jadwal penggunaan ruangan, kesulitan dalam pencatatan data peminjam, serta kurangnya informasi mengenai status ketersediaan ruangan. Hal tersebut dapat menghambat efektivitas dan efisiensi pengelolaan fasilitas kampus.

Oleh karena itu, diperlukan sebuah aplikasi sistem reservasi ruangan yang mampu mengelola data peminjam, data ruangan, serta jadwal penggunaan ruangan secara terstruktur. Dalam pembuatan aplikasi ini, pendekatan Object Oriented Programming (OOP) dipilih karena mampu memodelkan permasalahan dunia nyata ke dalam bentuk objek, sehingga kode program menjadi lebih terorganisir, mudah dipahami, serta mudah dikembangkan di masa mendatang. Aplikasi Sistem Reservasi Ruang Fakultas Vokasi ini dibangun menggunakan bahasa pemrograman Python dengan antarmuka grafis (GUI) berbasis Tkinter. Aplikasi ini diharapkan dapat menjadi solusi sederhana untuk mengelola proses reservasi ruangan sekaligus sebagai media penerapan konsep-konsep OOP dalam pengembangan perangkat lunak.

### **1.2. Tujuan Pembuatan Aplikasi**

1. Menerapkan konsep Object Oriented Programming (OOP) dalam pengembangan aplikasi berbasis Python.
2. Membuat aplikasi GUI sederhana untuk mengelola reservasi ruangan di lingkungan fakultas.
3. Memudahkan pengguna dalam melakukan peminjaman ruangan tanpa terjadi bentrok jadwal.
4. Menyediakan informasi reservasi ruangan secara terstruktur dan mudah dipahami.
5. Melatih kemampuan analisis, perancangan, dan implementasi sistem berbasis objek.

### **1.3. Ruang Lingkup Aplikasi**

1. Aplikasi dikembangkan menggunakan bahasa pemrograman Python dengan library Tkinter sebagai antarmuka grafis.
2. Aplikasi menerapkan konsep OOP yang meliputi encapsulation, inheritance, dan polymorphism.
3. Sistem hanya melayani proses reservasi ruangan oleh satu jenis pengguna, yaitu peminjam (mahasiswa).
4. Data ruangan dan data reservasi disimpan sementara di dalam memori (tanpa menggunakan database).
5. Fitur aplikasi meliputi input data peminjam, pemilihan ruangan, penentuan tanggal dan waktu, serta pembatalan reservasi.

## BAB II PEMBAHASAN

### 2.1. Konsep OOP

Object Oriented Programming (OOP) merupakan paradigma pemrograman yang berorientasi pada objek. Dalam OOP, sebuah sistem dibangun dari kumpulan objek yang saling berinteraksi. Setiap objek merepresentasikan entitas dunia nyata yang memiliki atribut (data) dan method (perilaku). Penerapan OOP bertujuan untuk meningkatkan keterbacaan kode, mempermudah pemeliharaan program, serta memungkinkan pengembangan sistem secara modular. Dengan OOP, pengembang dapat memecah permasalahan kompleks menjadi bagian-bagian yang lebih kecil dan terstruktur dalam bentuk class dan objek. Dalam aplikasi Sistem Reservasi Ruangan Fakultas Vokasi, konsep OOP diterapkan untuk memodelkan entitas seperti pengguna (peminjam), ruangan, dan reservasi. Konsep OOP utama yang digunakan dalam aplikasi ini meliputi Encapsulation, Inheritance, dan Polymorphism.

### 2.2. Encapsulation

Encapsulation merupakan konsep OOP yang bertujuan untuk membungkus data dan method ke dalam sebuah class, serta membatasi akses langsung terhadap data tersebut. Dengan encapsulation, data menjadi lebih aman dan perubahan data dapat dikontrol melalui method yang telah disediakan.

Implementasi Encapsulation pada Aplikasi :

- a. Atribut dengan akses protected

```
1 class User:
2     def __init__(self, user_id, nama):
3         self._user_id = user_id
4         self._nama = nama
```

Gambar 2. 1 Gambar Kode Program Atribut Dengan Akses Protected

- Atribut `_user_id` dan `_nama` menggunakan underscore (`_`). Ini menandakan protected attribute artinya atribut boleh diakses oleh class itu sendiri dan class turunannya

- b. Data dibungkus dalam class

```
1 class Reservasi:
2     def __init__(self, peminjam, ruangan, tanggal, waktu):
3         self.peminjam = peminjam
4         self.ruangan = ruangan
5         self.tanggal = tanggal
6         self.waktu = waktu
7         self.status = "Dikonfirmasi"
```

Gambar 2. 2 Gambar Kode Program Data Dibungkus Dalam Class

- Semua data reservasi tidak berdiri sendiri dan Dibungkus rapi dalam satu objek Reservasi

- c. Akses data melalui method

```
1 def batalkan(self):
2     self.status = "Dibatalkan"
```

Gambar 2. 3 Gambar Kode Program Akses Data Melalui Method

- Perubahan status tidak langsung diubah dari luar dan Dilakukan melalui method

## 2.3. Inheritance

Inheritance atau pewarisan merupakan konsep OOP yang memungkinkan sebuah class turunan (subclass) mewarisi atribut dan method dari class induk (superclass). Konsep ini bertujuan untuk menghindari pengulangan kode dan mempermudah pengembangan fitur baru.

Implementasi Encapsulation pada Aplikasi :

- a. Pewarisan User → Peminjam

```
1 class Peminjam(User):
2     def __init__(self, user_id, nama, no_identitas, prodi):
3         super().__init__(user_id, nama)
4         self.no_identitas = no_identitas
5         self.prodi = prodi
```

Gambar 2. 4 Gambar Kode Program Pewarisan User

- b. Pewarisan Ruangan

- RuangKelas

```
1 class RuangKelas(Ruangan):
2     def __init__(self, kode, nama, kapasitas, meja):
3         super().__init__(kode, nama, kapasitas)
4         self.meja = meja
5
6     def info(self):
7         return f"[KELAS] {self.nama} | Meja {self.meja}"
```

Gambar 2. 5 Gambar Kode Program Pewarisan Ruangan Kelas

- RuangRapat

```
1 class RuangRapat(Ruangan):
2     def __init__(self, kode, nama, kapasitas, fasilitas):
3         super().__init__(kode, nama, kapasitas)
4         self.fasilitas = fasilitas
5
6     def info(self):
7         return f"[RAPAT] {self.nama} | {self.fasilitas}"
```

Gambar 2. 6 Gmbar Kode Program Pewarisan Ruangan Rapat

## 2.4. Polymorphism

Polymorphism merupakan konsep OOP yang memungkinkan method dengan nama yang sama memiliki perilaku yang berbeda, tergantung pada objek yang memanggilnya. Polymorphism biasanya diterapkan melalui proses override method pada class turunan.

Implementasi Encapsulation pada Aplikasi :

- a. Method info() pada class Ruangan

```
1 class Ruangan:
2     def __init__(self, kode, nama, kapasitas):
3         self.kode = kode
4         self.nama = nama
5         self.kapasitas = kapasitas
6
7     def info(self):
8         return f"{self.nama} | Kapasitas {self.kapasitas}"
```

Gambar 2. 7 Gambar Kode Program Method info() pada class Ruangan

- b. Method info() dioverride di RuangKelas

```
1 class RuangKelas(Ruangan):
2     def __init__(self, kode, nama, kapasitas, meja):
3         super().__init__(kode, nama, kapasitas)
4         self.meja = meja
5
6     def info(self):
7         return f"[KELAS] {self.nama} | Meja {self.meja}"
```

Gambar 2. 8 Gmbar Kode Program Method info() pada class RuangKelas

- c. Method info() dioverride di RuangRapat

```
1 class RuangRapat(Ruangan):
2     def __init__(self, kode, nama, kapasitas, fasilitas):
3         super().__init__(kode, nama, kapasitas)
4         self.fasilitas = fasilitas
5
6     def info(self):
7         return f"[RAPAT] {self.nama} | {self.fasilitas}"
```

Gambar 2. 9 Gambar Kode Program Method info() pada class RuangRapat

## BAB III PERANCANGAN SISTEM

### 3.1. Deskripsi Umum Sistem

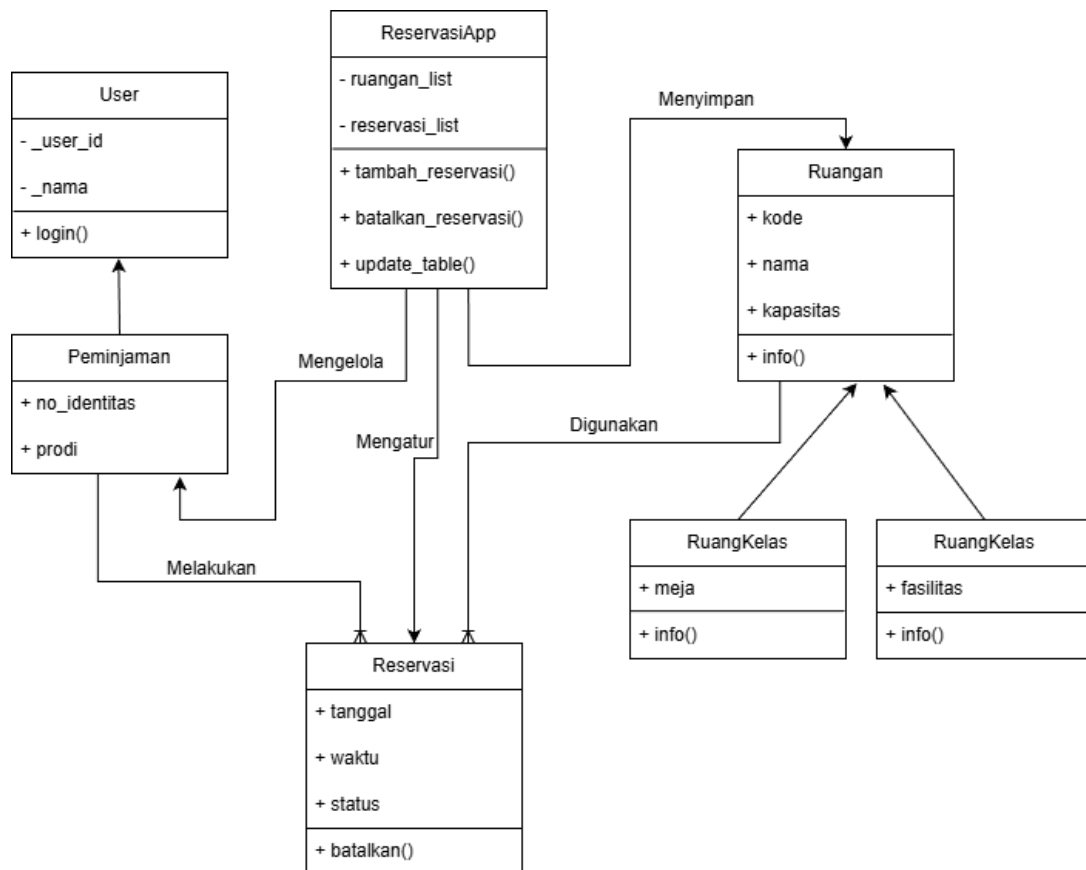
Aplikasi Sistem Reservasi Ruang Fakultas Vokasi merupakan aplikasi berbasis desktop yang dibangun menggunakan bahasa pemrograman Python dengan antarmuka grafis (GUI) berbasis Tkinter. Aplikasi ini dirancang untuk membantu proses peminjaman ruangan di lingkungan fakultas agar lebih terstruktur dan terhindar dari bentrok jadwal penggunaan ruangan. Sistem ini memungkinkan pengguna untuk menginput data peminjam, menentukan tanggal dan waktu peminjaman, memilih jenis ruangan yang tersedia, serta mengelola daftar reservasi yang telah dibuat. Seluruh proses reservasi dilakukan melalui satu antarmuka aplikasi yang sederhana dan mudah digunakan.

Aktor utama dalam sistem ini adalah Peminjam, yaitu mahasiswa yang melakukan pemesanan ruangan. Peminjam berinteraksi langsung dengan sistem untuk :

- Mengisi data diri
- Memilih ruangan
- Melakukan reservasi ruangan
- Membatalkan reservasi yang telah dibuat

### 3.2. Diagram Class

Diagram class digunakan untuk menggambarkan struktur kelas dalam sistem beserta atribut, method, dan relasi antar class. Diagram ini membantu dalam memahami arsitektur sistem dan implementasi konsep Object Oriented Programming (OOP) pada aplikasi.



Gambar 3. 1 Gambar Class Diagram

Diagram class pada aplikasi ini terdiri dari beberapa class utama, yaitu User, Peminjam, Ruangan, RuangKelas, RuangRapat, Reservasi, dan ReservasiApp. Class-class tersebut saling berelasi untuk membentuk alur sistem reservasi ruangan yang terintegrasi.

**a. User**

Merupakan class induk (superclass) yang merepresentasikan pengguna sistem secara umum. Class ini menjadi dasar bagi pengguna yang akan melakukan reservasi.

Atribut :

- `_user_id` : ID unik pengguna
- `_nama` : Nama pengguna

Method :

- `login()` : Proses autentikasi pengguna

**b. Peminjam**

Merupakan turunan (inheritance) dari class User, yang secara spesifik adalah pengguna yang melakukan peminjaman ruangan. Peminjam pasti adalah User, tetapi User belum tentu Peminjam.

Atribut :

- `no_identitas` : Nomor identitas (misalnya NIM)
- `prodi` : Program studi peminjam

**c. ReservasiApp**

Merupakan class utama (controller/manager) yang mengelola keseluruhan proses reservasi. Class ini berperan sebagai pusat pengaturan sistem.

Atribut :

- `ruangan_list` : Daftar ruangan yang tersedia
- `reservasi_list` : Daftar data reservasi

Method :

- `tambah_reservasi()` : Menambahkan data reservasi
- `batalikan_reservasi()` : Membatalkan reservasi
- `update_table()` : Memperbarui tampilan data (misalnya tabel di UI)

**d. Reservasi**

Merepresentasikan data pemesanan ruangan.

Atribut :

- `tanggal` : Tanggal peminjaman
- `waktu` : Waktu peminjaman
- `status` : Status reservasi (aktif/batal)

Method :

- `batalikan()` : Mengubah status reservasi menjadi batal

**e. Ruangan**

Merupakan class induk untuk jenis ruangan.

Atribut :

- `kode` : Kode ruangan
- `nama` : Nama ruangan
- `kapasitas` : Kapasitas ruangan

Method :

- info() : Menampilkan informasi ruangan

**f. RuangKelas**

Merupakan turunan dari class Ruangan yang merepresentasikan ruang kelas.

Atribut :

- meja : Jumlah meja

Method :

- info() : Informasi spesifik ruang kelas

**g. RuangRapat**

Merupakan turunan dari class Ruangan yang merepresentasikan ruang rapat.

Atribut :

- fasilitas : Fasilitas ruang rapat

Method :

- info() : Informasi spesifik ruang rapat

### 3.3. Relasi Antar Class

Relasi antar class dalam sistem ini dirancang untuk mendukung penerapan konsep OOP secara optimal. Relasi yang digunakan meliputi inheritance, association, dan dependency.

**a. Inheritance (Pewarisan)**

Inheritance digunakan untuk menghindari duplikasi kode dan mempermudah pengembangan sistem.

Class Induk	Class Turunan
User	Peminjaman
Ruangan	RuangKelas
Ruangan	RuangRapat

Table 3. 1 Table Relasi Inheritance

**b. Association (Asosiasi)**

Association menunjukkan hubungan logis antar objek dalam sistem.

Class	Relasi
Peminjam → Reservasi	Peminjam dapat melakukan satu atau lebih reservasi
Ruangan → Reservasi	Satu ruangan dapat digunakan dalam beberapa reservasi

Table 3. 2 Table Relasi Association

**c. Dependency (Ketergantungan)**

Dependency menunjukkan bahwa class ReservasiApp bergantung pada class lain untuk menjalankan fungsionalitas aplikasi.

Class	Relasi
ReservasiApp → Peminjam	Mengelola data peminjam
ReservasiApp → Ruangan	Mengelola data ruangan
ReservasiApp → Reservasi	Mengatur proses reservasi

Table 3. 3 Table Relasi Dependency

## BAB IV IMPLEMENTASI APLIKASI

### 4.1. Fitur Aplikasi

Aplikasi Sistem Reservasi Ruangan Fakultas Vokasi memiliki beberapa fitur utama yang dirancang untuk memudahkan proses peminjaman ruangan serta pengelolaan data reservasi. Fitur-fitur yang tersedia dalam aplikasi ini antara lain :

#### 1. Input Data Peminjam

Aplikasi menyediakan form untuk menginput data peminjam yang meliputi nama peminjam, nomor identitas (NIM), dan program studi. Data ini digunakan sebagai identitas peminjam pada setiap reservasi yang dilakukan.

#### 2. Penentuan Tanggal dan Waktu Reservasi

Pengguna dapat menentukan tanggal dan waktu peminjaman ruangan sesuai dengan kebutuhan kegiatan yang akan dilakukan.

#### 3. Pemilihan Ruangan

Aplikasi menampilkan daftar ruangan yang tersedia yang dibedakan menjadi dua kategori, yaitu ruang kelas dan ruang rapat. Pengguna hanya dapat memilih satu ruangan untuk setiap reservasi.

#### 4. Validasi Bentrok Jadwal

Sistem secara otomatis melakukan pengecekan jadwal untuk mencegah terjadinya bentrok pemesanan ruangan pada tanggal dan waktu yang sama.

#### 5. Menampilkan Daftar Reservasi

Semua reservasi yang telah dibuat akan ditampilkan dalam bentuk tabel yang berisi informasi peminjam, ruangan, tanggal, waktu, dan status reservasi.

#### 6. Pembatalan Reservasi

Pengguna dapat membatalkan reservasi yang telah dibuat. Sistem tidak menghapus data reservasi, melainkan mengubah status reservasi menjadi Dibatalkan.

#### 7. Antarmuka Scrollable

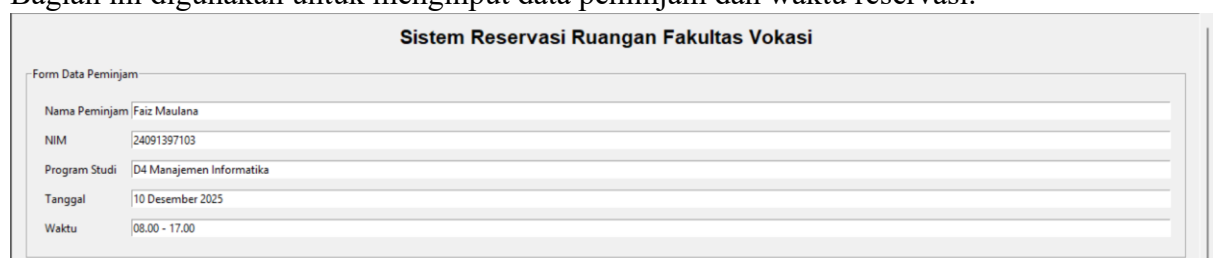
Aplikasi dilengkapi dengan fitur scroll untuk memudahkan pengguna dalam menavigasi tampilan ketika jumlah data atau ukuran layar terbatas.

### 4.2. Tampilan Aplikasi

Antarmuka aplikasi dirancang sederhana dan mudah digunakan, sehingga pengguna dapat memahami alur penggunaan aplikasi tanpa memerlukan panduan yang kompleks. Tampilan aplikasi terdiri dari beberapa bagian utama sebagai berikut :

#### 1. Tampilan Form Data Peminjam

Bagian ini digunakan untuk menginput data peminjam dan waktu reservasi.



Form Data Peminjam	
Nama Peminjam	Faiz Maulana
NIM	24091397103
Program Studi	D4 Manajemen Informatika
Tanggal	10 Desember 2025
Waktu	08.00 - 17.00

Gambar 4. 1 Gambar Form Data Peminjam

#### 2. Tampilan Pemilihan Ruangan dan Tambah Reservasi

Pada bagian ini, aplikasi menampilkan dua tabel ruangan yang disusun secara berdampingan. Pengguna hanya dapat memilih satu ruangan pada satu waktu. Jika pengguna memilih ruangan pada tabel ruang kelas, maka pilihan pada tabel ruang rapat akan otomatis dibatalkan, dan sebaliknya. Dan Tombol *Tambah Reservasi* digunakan untuk menyimpan data reservasi berdasarkan input yang telah diisi. Setelah tombol ditekan, sistem akan melakukan validasi data dan menambahkan reservasi ke dalam daftar jika tidak terjadi kesalahan.

Ruangan Kelas			Ruangan Rapat		
Nama	Kapasitas	Meja	Nama	Kapasitas	Fasilitas
Ruang K2.01.02	40	20	Ruang K9.02.01	40	Proyektor, AC
Ruang K2.01.03	40	20	Ruang K9.02.02	40	Proyektor, AC
Ruang K2.01.04	40	20	Ruang K9.02.03	40	Proyektor, AC
Ruang K2.01.05	40	20	Ruang K9.02.04	30	Proyektor, AC
Ruang K10.02.01	40	20	Ruang K9.02.05	40	Proyektor, AC

**Tambah Reservasi**

Gambar 4. 2 Gambar Pilih Ruangan dan Tombol Tambah Reservasi

### 3. Tampilan Daftar Reservasi dan Batalkan Reservasi

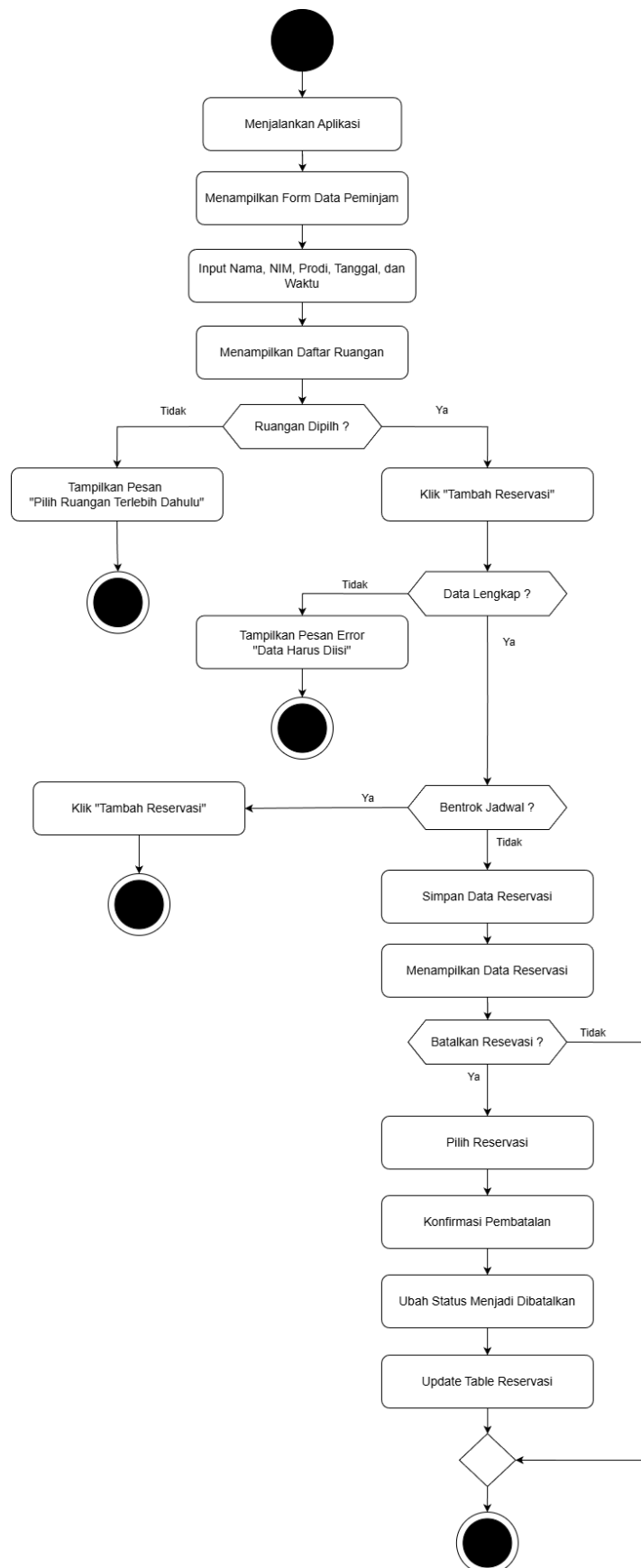
Bagian ini menampilkan daftar seluruh reservasi yang telah dibuat dalam bentuk table. Dan Pengguna dapat memilih salah satu reservasi pada tabel, kemudian menekan tombol *Batalkan Reservasi*. Status reservasi akan berubah menjadi *Dibatalkan* setelah pengguna mengonfirmasi tindakan tersebut.

Nama (NIM)	Prodi	Ruangan	Tanggal	Waktu	Status
Faiz Maulana (24091397103)	D4 Manajemen Informatika	Ruang K2.01.02	10 Desember 2025	08.00 - 17.00	Dikonfirmasi

**Batalkan Reservasi**

Gambar 4. 3 Gambar Daftar Reservasi dan Tombol Batalkan Reservasi

### 4.3. Alur Aplikasi



Gambar 4. 4 Gambar Alur Aplikasi

Alur aplikasi Sistem Reservasi Ruangan Fakultas Vokasi dapat dijelaskan sebagai berikut :

1. Pengguna menjalankan aplikasi.
2. Aplikasi menampilkan form input data peminjam.
3. Pengguna mengisi data peminjam, tanggal, dan waktu reservasi.
4. Pengguna memilih ruangan yang tersedia.
5. Pengguna menekan tombol Tambah Reservasi.
6. Sistem melakukan validasi data dan pengecekan bentrok jadwal.
7. Jika data valid, reservasi disimpan dan ditampilkan pada tabel daftar reservasi.
8. Pengguna dapat membatalkan reservasi dengan memilih data pada tabel dan menekan tombol Batalkan Reservasi.
9. Sistem memperbarui status reservasi dan menampilkan hasil perubahan pada tabel.

## **BAB V PENUTUP**

### **5.1. Kesimpulan**

Berdasarkan hasil perancangan dan implementasi aplikasi Sistem Reservasi Ruangan Fakultas Vokasi, dapat disimpulkan bahwa aplikasi ini berhasil dibangun dengan menerapkan konsep Object Oriented Programming (OOP) secara baik dan terstruktur. Konsep encapsulation, inheritance, dan polymorphism telah diimplementasikan melalui penggunaan class, pewarisan antar class, serta penggunaan method yang memiliki perilaku berbeda pada class turunan.

Aplikasi ini mampu memfasilitasi proses peminjaman ruangan dengan menyediakan fitur input data peminjam, pemilihan ruangan, penentuan tanggal dan waktu, serta pengelolaan data reservasi. Selain itu, sistem juga dapat melakukan validasi data dan mencegah terjadinya bentrok jadwal peminjaman ruangan.

Dengan adanya aplikasi ini, proses reservasi ruangan menjadi lebih tertata, efisien, dan mudah digunakan. Antarmuka berbasis GUI juga membantu pengguna dalam memahami alur penggunaan aplikasi tanpa memerlukan keahlian teknis khusus.