

**IMPLEMENTASI DAN PENGUJIAN PERANGKAT LUNAK**

**TUGAS #i - Convert Class Diagram ke Code**

**NYAMAN.IN**

**KELAS IF-10-03**



**Dosen: Muhamad Azrino Gustalika**

**Disusun Oleh:**

- |          |                                     |                   |
|----------|-------------------------------------|-------------------|
| <b>1</b> | <b>Muhammad Faiz Arkan Herwanto</b> | <b>2211102217</b> |
| <b>2</b> | <b>Muhammad Rizki Agustiansyah</b>  | <b>2211102325</b> |
| <b>3</b> | <b>Faaris Muwaffaq</b>              | <b>2211102032</b> |
| <b>4</b> | <b>Muhamad Raihan Annaji</b>        | <b>2211102358</b> |

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

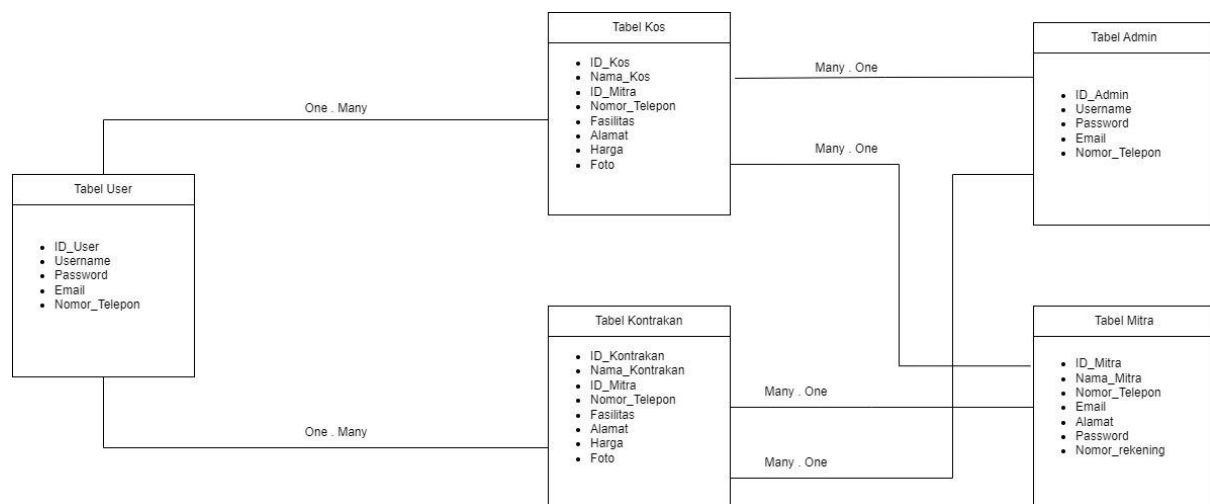
**FAKULTAS INFORMATIKA**

**UNIVERSITAS TELKOM**

**PURWOKERTO**

**2024**

## Class Diagram



## Implementasi Code dari Class Diagram

Source Code:

```
class User:
    def __init__(self, id_user, username, password, email, nomor_telepon):
        self.id_user = id_user
        self.username = username
        self.password = password
        self.email = email
        self.nomor_telepon = nomor_telepon

    def update_profile(self, username=None, email=None, nomor_telepon=None):
        """Update user profile"""
        if username:
            self.username = username
        if email:
            self.email = email
        if nomor_telepon:
            self.nomor_telepon = nomor_telepon
        print("Profile updated successfully.")

    def view_properties(self, properties):
        """View list of properties (Kos or Kontrakan)"""
        for property_item in properties:
            property_item.display_details()

    def book_property(self, property_item):
        """Method for booking a property"""
        print(f"{self.username} booked {property_item.nama}.")

class Property:
    def __init__(self, id_property, nama, id_mitra, nomor_telepon, fasilitas, alamat, harga, foto):
        self.id_property = id_property
        self.nama = nama
        self.id_mitra = id_mitra
        self.nomor_telepon = nomor_telepon
        self.fasilitas = fasilitas
```

```

        self.alamat = alamat
        self.harga = harga
        self.foto = foto

    def update_info(self, fasilitas=None, alamat=None, harga=None):
        """Update property information"""
        if fasilitas:
            self.fasilitas = fasilitas
        if alamat:
            self.alamat = alamat
        if harga:
            self.harga = harga
        print(f"Information for {self.nama} updated.")

    def display_details(self):
        """Display property details"""
        print(f>Nama: {self.nama}, Harga: {self.harga}, Alamat: {self.alamat}, Fasilitas: {self.fasilitas}")

class Kos(Property):
    pass

class Kontrakan(Property):
    pass

class Admin:
    def __init__(self, id_admin, username, password, email, nomor_telepon):
        self.id_admin = id_admin
        self.username = username
        self.password = password
        self.email = email
        self.nomor_telepon = nomor_telepon

    def add_property(self, property_list, new_property):
        """Add a new property (Kos or Kontrakan) to the system"""
        property_list.append(new_property)
        print(f"New property {new_property.nama} added by admin {self.username}.")

    def remove_property(self, property_list, property_id):
        """Remove a property by ID"""
        property_list[:] = [p for p in property_list if p.id_property != property_id]
        print(f"Property with ID {property_id} removed by admin {self.username}.")

class Mitra:
    def __init__(self, id_mitra, nama_mitra, nomor_telepon, email, alamat, password, nomor_rekening):
        self.id_mitra = id_mitra
        self.nama_mitra = nama_mitra
        self.nomor_telepon = nomor_telepon
        self.email = email
        self.alamat = alamat
        self.password = password
        self.nomor_rekening = nomor_rekening

    def add_property(self, property_list, new_property):
        """Add a new property (Kos or Kontrakan)"""
        property_list.append(new_property)

```

```

        print(f"New property {new_property.nama} added by mitra
{self.nama_mitra}.")

    def update_property(self, property_item, fasilitas=None, alamat=None,
harga=None):
        """Update property information"""
        property_item.update_info(fasilitas=fasilitas, alamat=alamat,
harga=harga)
        print(f"Property {property_item.nama} updated by mitra
{self.nama_mitra}.")

user1 = User(id_user=1, username="Paez", password="password123",
email="@Gmail.com", nomor_telepon="08123456789")

property1 = Property(id_property=101, nama="Kos Harmoni", id_mitra=201,
nomor_telepon="08123456789",
                    fasilitas="AC, WiFi", alamat="Jalan Mawar No. 10",
harga=1500000, foto="foto_kos.jpg")

admin1 = Admin(id_admin=1, username="admin_user", password="adminpass",
email="admin@example.com", nomor_telepon="08987654321")

properties = []
admin1.add_property(properties, property1)

user1.view_properties(properties)

user1.book_property(property1)

```

Output :

```

PS D:\BUKAN DATA FAIZ\Kuliah\Semester 5\Manajemen Projek TIK> & C:/Users/ASUS/AppData/Local/Program
New property Kos Harmoni added by admin admin_user.
Nama: Kos Harmoni, Harga: 1500000, Alamat: Jalan Mawar No. 10, Fasilitas: AC, WiFi
Paez booked Kos Harmoni.
PS D:\BUKAN DATA FAIZ\Kuliah\Semester 5\Manajemen Projek TIK> 

```

## 1. Kelas User

Kelas ini merepresentasikan pengguna sistem yang bisa melakukan operasi terkait profil mereka dan interaksi dengan properti.

- **Atribut:** id\_user, username, password, email, nomor\_telepon menyimpan informasi profil pengguna.
- **Metode:**
  - update\_profile: Memperbarui informasi profil pengguna (username, email, dan nomor\_telepon).
  - view\_properties: Menampilkan daftar properti yang tersedia dengan memanggil metode display\_details() dari setiap objek properti yang diberikan.
  - book\_property: Menampilkan pesan bahwa pengguna telah memesan properti tertentu.

## 2. Kelas Property

Kelas ini adalah *parent class* yang merepresentasikan properti umum, baik untuk Kos maupun Kontrakan.

- **Atribut:**
  - `id_property`: ID unik untuk properti.
  - `nama`: Nama properti.
  - `id_mitra`: ID mitra yang memiliki atau mengelola properti.
  - `nomor_telepon`: Nomor telepon mitra atau kontak yang dapat dihubungi.
  - `fasilitas, alamat, harga, foto`: Atribut untuk mendeskripsikan properti, seperti fasilitas yang tersedia, lokasi, harga, dan foto.
- **Metode:**
  - `update_info`: Memperbarui fasilitas, alamat, atau harga properti jika ada perubahan.
  - `display_details`: Menampilkan detail properti seperti nama, harga, alamat, dan fasilitas yang tersedia.

Kelas Property ini digunakan sebagai *parent class* untuk kelas Kos dan Kontrakan. Karena Kos dan Kontrakan hanya berbeda dari sisi nama, mereka tidak membutuhkan tambahan fungsi atau atribut baru, sehingga cukup dengan pewarisan dari Property.

## 3. Kelas Kos dan Kontrakan

Kelas Kos dan Kontrakan mewarisi dari Property, sehingga otomatis memiliki semua atribut dan metode yang ada di Property. Ini membuat kode lebih ringkas dan modular tanpa perlu mendefinisikan ulang hal-hal yang sama.

## 4. Kelas Admin

Kelas ini merepresentasikan administrator sistem yang memiliki hak untuk menambah dan menghapus properti dari sistem.

- **Atribut:** `id_admin, username, password, email, nomor_telepon` menyimpan informasi profil admin.
- **Metode:**
  - `add_property`: Menambahkan properti baru (baik Kos maupun Kontrakan) ke dalam daftar properti `property_list`.
  - `remove_property`: Menghapus properti dari `property_list` berdasarkan `id_property`. Di sini, filter diterapkan pada `property_list` untuk menghapus properti dengan `id_property` yang sesuai.

## 5. Kelas Mitra

Kelas ini merepresentasikan mitra penyedia properti yang bisa menambah atau memperbarui informasi properti mereka.

- **Atribut:** `id_mitra`, `nama_mitra`, `nomor_telepon`, `email`, `alamat`, `password`, `nomor_rekening` menyimpan informasi profil dan kontak mitra.
- **Metode:**
  - `add_property`: Menambahkan properti baru ke dalam daftar `property_list`, seperti halnya metode Admin.
  - `update_property`: Memperbarui detail properti yang dikelola mitra tersebut (fasilitas, alamat, dan harga).

### **Contoh Penggunaan**

Dengan struktur ini, Anda bisa melakukan operasi sebagai berikut:

- User dapat melihat dan memesan properti.
- Admin dapat menambah atau menghapus properti dari sistem.
- Mitra dapat menambah atau memperbarui properti yang mereka kelola.