PROJECT OBJECT ORIENTED PROGRAMMING

PENDIDIKAN FAKULTAS ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

Praktikum Pemrograman Dasar – Teknik Komputer D 2025

Dibuat Oleh :

Kelompok 12

MUHAMMAD FACHRIZA	255150307111053
KEVIN IBRAHIM	255150307111058
MARTUA DODY YOHANES SIMANJUTAK	255150300111042

UNIVERSITAS BRAWIJAYA

FAKULTAS ILMU KOMPUTER

A. Latar Belakang

Di sebuah Universitas Brawijaya, ada banyak jenis warga kampus seperti mahasiswa, dosen, dan karyawan. Setiap kelompok memiliki data yang berbeda-beda, misalnya mahasiswa punya NIM dan prodi, dosen punya NIDN, karyawan punya NIP dan divisi. Karena itulah dibutuhkan sebuah program yang bisa mengelola berbagai jenis data tersebut secara rapi dan mudah digunakan.

Pada tugas live coding ini, dibuat sebuah program berbasis OOP (Object-Oriented Programming) dengan bahasa C++. OOP dipilih karena memudahkan pengelompokan data berdasarkan jenisnya, serta membuat program lebih terstruktur dan gampang dikembangkan. Program ini memungkinkan pengguna untuk memasukkan data warga kampus sesuai jenisnya, lalu menampilkan kembali data tersebut dengan format yang sudah disesuaikan.

B. Gambaran Besar Project

Project ini adalah program sederhana untuk menginput dan menampilkan data tiga jenis warga kampus:

a. Mahasiswa

- Nama
- NIM
- Program Studi

b. Dosen

- Nama
- NIDN
- Homebase

c. Karyawan

- Nama
- NIP
- Divisi

Program ini dibuat menggunakan satu kelas induk bernama WargaKampus, yang berfungsi sebagai template dasar. Di dalamnya ada atribut nama dan dua fungsi virtual (input() dan display()) yang wajib diisi oleh setiap kelas turunan. Tiga kelas turunan Mahasiswa, Dosen, dan Karyawan mengisi sendiri cara input dan tampilan datanya masing-masing.

Di bagian menu, pengguna bisa memilih ingin memasukkan data siapa. Program lalu membuat objek sesuai pilihan tersebut, menampungnya ke dalam sebuah vector, dan menampilkan hasil inputnya.

Secara keseluruhan, project ini menunjukkan konsep dasar OOP:

- Class dan inheritance
- Polymorphism (menggunakan pointer ke class induk)
- Penyimpanan data dinamis dengan vector

Tujuan akhirnya kami ialah membuat program sesederhana mungkin yang bisa dipakai untuk mengelola data warga kampus secara fleksibel dan terorganisir.

C. Penjelasan setiap class, field, dan method

1. Class WargaKampus (Class Induk / Abstract Class)

Class ini menjadi dasar untuk semua jenis warga kampus (Mahasiswa, Dosen, Karyawan).

Field (Atribut):

- string nama
  Menyimpan nama warga kampus. Karena sifatnya umum, atribut ini diletakkan di class induk.

Method:

- Constructor WargaKampus()
  Menginisialisasi nama menjadi string kosong.
- void setNama(string x)
  Mengubah nilai atribut nama.
- string getNama()
  Mengambil nilai nama.
- virtual void input() = 0;
  Method *pure virtual*. Artinya, setiap class turunan wajib membuat versi input sendiri.
- virtual void display() = 0;
  Method *pure virtual* untuk menampilkan data—akan berbeda sesuai jenis warga kampus.
- Destructor virtual ~WargaKampus()
  Destructor virtual agar penghapusan objek polimorfik pada vector berjalan benar.

2. Class Mahasiswa (Turunan dari WargaKampus)

Class ini mewakili data untuk mahasiswa.

Field (Atribut):

- string nim
  Nomor Induk Mahasiswa.
- string prodi
  Program studi mahasiswa.

Method:

- Constructor Mahasiswa()
  Memanggil constructor class induk, kemudian mengosongkan nim dan prodi.
- void setNim(string n)
  Mengisi nilai NIM.
- void setProdi(string p)
  Mengisi nilai program studi.
  void input()
  Meminta pengguna memasukkan:
  - Nama
  - NIM
  - Program Studi
    Data yang diinput kemudian disimpan dengan setter.
- void display()
  Menampilkan data mahasiswa yang telah diinput dengan format khusus.

3. Class Dosen (Turunan dari WargaKampus)

Class ini digunakan untuk menyimpan data dosen.

Field (Atribut):

- string nidn
  Nomor Induk Dosen Nasional.
- string homebase
  Homebase atau jurusan asal dosen.

Method:

- Constructor Dosen()
  Menginisialisasi nidn dan homebase menjadi string kosong.
- void setNidn(string n)
  Mengisi NIDN.
- void setHomebase(string h)
  Mengisi homebase dosen.
- void input()
  Meminta pengguna memasukkan:
  - Nama
  - NIDN
  - Homebase
- void display()
  Menampilkan data dosen secara lengkap.

4. Class Karyawan (Turunan dari WargaKampus)

Class ini mewakili data karyawan kampus.

Field (Atribut):

- string nip
  Nomor Induk Pegawai.
- string divisi
  Divisi tempat karyawan Universitas.

Method:

- Constructor Karyawan()
  Mengosongkan atribut nip dan divisi.
- void input()
  Meminta input berupa:
  - Nama
  - NIP
  - Divisi

- void display()
  Menampilkan data karyawan dengan format rapi.

5. Fungsi main()

Bagian utama program.

Penjelasan:

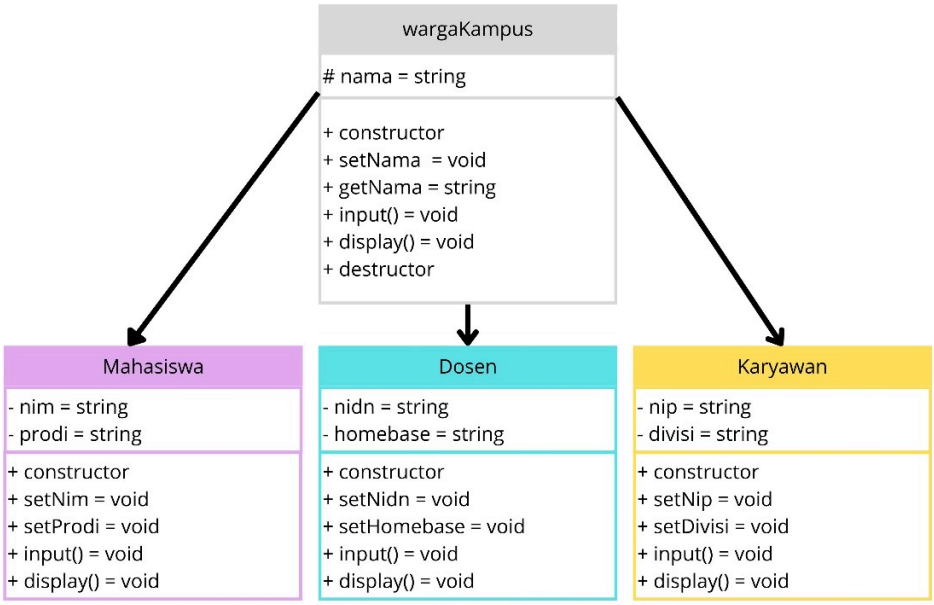
- Membuat sebuah vector<WargaKampus*> semuaData;
  Vector ini menyimpan pointer ke object dari berbagai class turunan.
- Program menampilkan menu:
    - Input data mahasiswa
    - Input data dosen
    - Input data karyawan
    - Keluar
- Berdasarkan pilihan, program akan:
    - Membuat object baru (new Mahasiswa, new Dosen, atau new Karyawan)
    - Memanggil input()
    - Memanggil display()
    - Menyimpan object tersebut ke vector
- Ketika keluar, program melakukan delete untuk setiap pointer agar tidak terjadi kebocoran memori.

D. Source Code

Berikut ialah link source code yang dipush ke Github :

https://github.com/anesmartua-ans/OOPKelompok12.git.

E. Pembuatan Class Diagaram



F. Kesimpulan

Dari program yang sudah dibuat, bisa disimpulkan bahwa kode ini bertujuan untuk mencatat dan menampilkan data dari tiga jenis warga kampus, yaitu mahasiswa, dosen, dan karyawan. Program dibuat dengan konsep OOP supaya lebih rapi dan gampang diatur.

Dengan adanya class induk WargaKampus dan class turunan seperti Mahasiswa, Dosen, dan Karyawan, program jadi bisa menangani berbagai jenis data yang berbeda, tapi tetap memakai pola yang sama. Setiap class bisa menginput dan menampilkan data sesuai kebutuhan masing-masing.

Selain itu, program juga memakai vector untuk menyimpan data sebanyak yang dimasukkan oleh pengguna, jadi jumlah datanya bisa fleksibel. Penggunaan fungsi input() dan display() yang berbeda pada setiap class juga menunjukkan bagaimana OOP membuat program lebih mudah dikembangkan.