

LAPORAN PROJECT UAS

1. Pendahuluan

Laporan ini merupakan hasil penggerjaan Tugas Akhir Semester (UAS) yang mengintegrasikan empat modul utama dalam sistem manajemen kampus. Proyek ini mendemonstrasikan kemampuan dalam mengelola struktur data kompleks menggunakan List of Dictionaries, logika alur program (looping & conditional), serta dokumentasi teknis melalui UML Class Diagram.

2. Analisis Sistem & UML Diagram

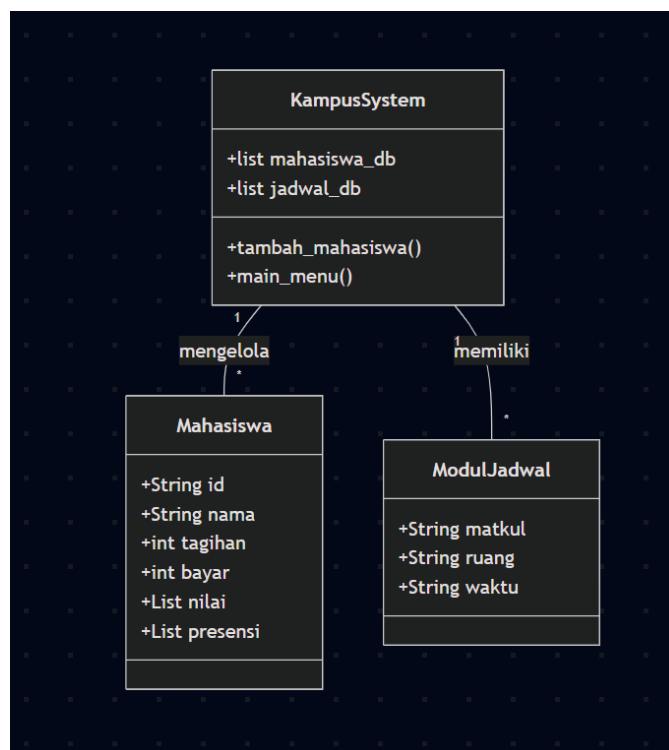
Sistem dibangun dengan arsitektur terpusat di mana KampusSystem bertindak sebagai pengelola utama database mahasiswa dan jadwal.

Deskripsi Class Diagram:

KampusSystem: Kelas utama yang menampung database dan logika menu.

Mahasiswa: Entitas yang memiliki atribut ID, Nama, Tagihan, dan histori (Presensi & Nilai).

ModulJadwal: Entitas penyimpan informasi akademik terkait waktu dan ruangan.



3. Penjelasan Modul Terintegrasi

3.1 Modul Registrasi & Database

Sistem menggunakan mahasiswa_db sebagai database in-memory. Fitur registrasi memastikan tidak ada duplikasi ID (NIM) saat mahasiswa baru daftarkan.

3.2 Modul Presensi (QR Code)

Mensimulasikan proses scan dengan mencocokkan ID. Sistem menggunakan library datetime untuk mencatat waktu kehadiran secara presisi (tahun, bulan, hari, hingga detik).

3.3 Modul Keuangan (SPP)

Mengelola arus kas masuk dari mahasiswa. Sistem akan secara otomatis mengurangkan jumlah bayar dari total tagihan tetap (Rp5.000.000) dan menampilkan sisa tunggakan secara real-time.

3.4 Modul Nilai & Transkrip (IPK)

Modul ini mengimplementasikan logika konversi nilai angka (0-100) menjadi bobot standar akademik (1.0 - 4.0). IPK dihitung berdasarkan rata-rata bobot nilai yang telah dimasukkan.

3.5 Modul Jadwal & Ruangan

Menampilkan informasi jadwal kuliah dalam format tabel yang rapi di terminal, mencakup Nama Mata Kuliah, Kode Ruangan, dan Waktu.

4. Implementasi Kode Python

```
import datetime

# Database Sederhana (List of Dictionaries)
mahasiswa_db = [
    {"id": "M001", "nama": "Budi", "tagihan": 5000000, "bayar": 0, "nilai": [], "presensi": []},
    {"id": "M002", "nama": "Ani", "tagihan": 5000000, "bayar": 0, "nilai": [], "presensi": []}
]

jadwal_db = [
    {"matkul": "Pemrograman Python", "ruang": "Lab 01", "waktu": "Senin 08:00"},
    {"matkul": "Struktur Data", "ruang": "R. 302", "waktu": "Selasa 10:00"}
]

# --- Fitur Tambahan: Registrasi Mahasiswa ---
def tambah_mahasiswa():
    """Fungsi untuk mendaftarkan mahasiswa baru ke database."""
    print("\n--- Registrasi Mahasiswa Baru ---")
    nim = input("Masukkan ID/NIM Baru: ")

    # Cek duplikasi ID
```

```

for mhs in mahasiswa_db:
    if mhs['id'] == nim:
        print("Gagal: ID sudah terdaftar!")
        return

    nama = input("Masukkan Nama Lengkap: ")
    mhs_baru = {
        "id": nim,
        "nama": nama,
        "tagihan": 5000000,
        "bayar": 0,
        "nilai": [],
        "presensi": []
    }
    mahasiswa_db.append(mhs_baru)
    print(f"Berhasil! Mahasiswa {nama} ({nim}) telah terdaftar di sistem.")

# --- 1. Modul Presensi (Simulasi QR) ---
def presensi_qr():
    """Mencatat kehadiran mahasiswa berdasarkan ID dengan timestamp otomatis."""
    print("\n--- Simulasi Scan QR Code ---")
    nim = input("Masukkan ID Mahasiswa: ")
    for mhs in mahasiswa_db:
        if mhs['id'] == nim:
            waktu_sekarang = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
            mhs['presensi'].append(waktu_sekarang)
            print(f"Berhasil! {mhs['nama']} tercatat hadir pada {waktu_sekarang}")
            return
    print("ID Mahasiswa tidak ditemukan.")

# --- 2. Modul Keuangan (SPP) ---
def keuangan_spp():
    """Mengelola input pembayaran dan menghitung sisa tunggakan."""
    print("\n--- Pembayaran SPP (Total Tagihan: Rp5.000.000) ---")
    nim = input("Masukkan ID Mahasiswa: ")
    for mhs in mahasiswa_db:
        if mhs['id'] == nim:
            try:
                bayar = int(input(f"Masukkan jumlah bayar untuk {mhs['nama']}: "))
                mhs['bayar'] += bayar
                sisa = mhs['tagihan'] - mhs['bayar']
                print(f"Pembayaran diterima. Sisa tunggakan {mhs['nama']}: Rp{sisa}")
            except ValueError:
                print("Input harus berupa angka nominal!")
            return
    print("ID Mahasiswa tidak ditemukan.")

# --- 3. Modul Nilai & Transkrip ---
def hitung_bobot(nilai):
    if nilai >= 80: return 4.0
    elif nilai >= 70: return 3.0
    elif nilai >= 60: return 2.0
    else: return 1.0

def input_nilai_ipk():
    """Input nilai, konversi ke bobot, dan hitung rata-rata (IPK)."""
    print("\n--- Input Nilai & Hitung IPK ---")
    nim = input("Masukkan ID Mahasiswa: ")
    for mhs in mahasiswa_db:
        if mhs['id'] == nim:
            try:
                n = int(input("Masukkan nilai angka (0-100): "))
                bobot = hitung_bobot(n)
                mhs['nilai'].append(bobot)
                ipk = sum(mhs['nilai']) / len(mhs['nilai'])
                print(f"Nilai tersimpan. IPK Sementara {mhs['nama']}: {ipk:.2f}")
            except ValueError:

```

```

        print("Input harus berupa angka!")
        return
    print("ID Mahasiswa tidak ditemukan.")

# --- 4. Modul Jadwal & Ruangan ---
def lihat_jadwal():
    """Menampilkan jadwal kuliah yang tersedia."""
    print("\n--- Jadwal Kuliah & Ruangan ---")
    print(f'{Mata Kuliah}<20} | {Ruang}<10} | {Waktu}')
    print("-" * 45)
    for j in jadwal_db:
        print(f'{j['matkul']}<20} | {j['ruang']}<10} | {j['waktu']}')

# --- Menu Utama ---
def main():
    while True:
        print("\n===== SISTEM KAMPUS TERINTEGRASI =====")
        print("0. Tambah Mahasiswa Baru (Registrasi ID)")
        print("1. Presensi Mahasiswa (Scan QR)")
        print("2. Pembayaran SPP")
        print("3. Input Nilai & IPK")
        print("4. Lihat Jadwal Kuliah")
        print("5. Keluar")
        pilihan = input("Pilih menu (0-5): ")

        if pilihan == '0': tambah_mahasiswa()
        elif pilihan == '1': presensi_qr()
        elif pilihan == '2': keuangan_spp()
        elif pilihan == '3': input_nilai_ipk()
        elif pilihan == '4': lihat_jadwal()
        elif pilihan == '5':
            print("Keluar dari sistem. Terima kasih!")
            break
        else:
            print("Pilihan tidak valid.")

if __name__ == "__main__":
    main()

```

5. Kesimpulan

Proyek UAS ini berhasil menyatukan berbagai fungsi manajemen kampus ke dalam satu aplikasi berbasis terminal. Penggunaan struktur data list of dictionaries terbukti fleksibel untuk menangani data yang saling terkait (seperti menghubungkan ID mahasiswa dengan histori nilai dan presensi mereka).