# LAPORAN TUGAS BESAR
# PROYEK BASIS DATA LANJUT


**Disusun Oleh :**

1. Aulia Dwi Rahmadani (G1A023043)
2. Yohanes Adi Prasetya (G1A023049)
3. Waridhania As Syifa (G1A023075)
4. I Nyoman Dimas Kresna Adryan (G1A023077)

**Nama Asisten Dosen :**

1. Merly Yuni Purnama (G1A021006)
2. Reksi Hendra Pratama (G1A022032)
3. Sinta Ezra Wati Gulo (G1A022040)
4. Fadlan Dwi Febrio (G1A022051)
5. Torang Four Yones Manullang (G1A022052)
6. Wahyu Ozorah Manurung (G1A022060)
7. Shalaudin Muhammad Sah (G1A022070)
8. Dian Ardiyanti Saputri (G1A022084)

**Dosen Pengampu :**

1. Dr. Endina Putri Purwandari, S.T., M.Kom.
2. Ir. Tiara Eka Putri, S.T., M.Kom.

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS BENGKULU**


**2025**

# DAFTAR ISI

DAFTAR ISI ..... i
DAFTAR GAMBAR ..... ii
KATA PENGANTAR ..... 1
BAB I ..... 2
PENDAHULUAN ..... 2
1.1 Latar Belakang ..... 2
1.2 Rumusan Masalah ..... 4
1.3 Tujuan Penulisan ..... 4
BAB II ..... 5
LANDASAN TEORI ..... 5
2.1 Dasar Teori ..... 5
A. Basis Data Lanjut ..... 5
B. Sistem Akademik Dan Manajemen Pembayaran UKT Mahasiswa ..... 6
BAB III ..... 7
HASIL DAN PEMBAHASAN ..... 7
BAB IV ..... 59
PENUTUP ..... 59
4.1 Kesimpulan ..... 59
4.2 Saran ..... 59
DOKUMENTASI PENGERJAAN ..... 60
DAFTAR PUSTAKA ..... 61

## DAFTAR GAMBAR

Gambar 1 ERD Sistem Akademik & Manajemen Pembayaran UKT
Gambar 2 Source Code Tabel Fakultas, Program Studi, Golongan UKT
Gambar 4 Source Code Tabel Mahasiswa
Gambar 5 Source Code Create Table Dosen dan Mata Kuliah
Gambar 6 Source Code Create Table Semester dan Nilai
Gambar 7 Source Code Create Table Pembayaran UKT
Gambar 8 Source Code Insert Into Tabel Fakultas dan Program Studi
Gambar 9 Source Code Insert Into Tabel Program Studi
Gambar 10 Source Code Insert Into Tabel Program Studi
Gambar 11 Source Code Insert Into Tabel Golongan UKT
Gambar 12 Source Code Insert Into Tabel Golongan UKT
Gambar 13 Source Code Insert Into Tabel Mahasiswa
Gambar 14 Source Code Insert Into Tabel Mahasiswa
Gambar 15 Source Code Insert Into Tabel Mahasiswa
Gambar 16 Source Code Insert Into Tabel Mahasiswa
Gambar 17 Source Code Insert Into Tabel Dosen
Gambar 18 Source Code Insert Into Tabel Dosen
Gambar 19 Source Code Insert Into Tabel Mata Kuliah
Gambar 20 Source Code Insert Into Tabel Mata Kuliah
Gambar 21 Source Code Insert Into Tabel Pembayaran UKT
Gambar 22 Source Code Insert Into Tabel Pembayaran UKT
Gambar 23 Tabel Fakultas
Gambar 24 Tabel Program studi
Gambar 25 Tabel Golongan UKT
Gambar 26 Tabel Mahasiswa
Gambar 27 Tabel Dosen
Gambar 28 Tabel Mata Kuliah
Gambar 29 Tabel Semester
Gambar 30 Tabel Nilai
Gambar 31 Tabel Pembayaran
Gambar 32 Source Code Join
Gambar 33 Tabel Join
Gambar 34 Tabel Join
Gambar 35 Tabel Join

Gambar 36 Source Code Alias
Gambar 37 Source Code Alias
Gambar 38 Source Code Operator
Gambar 39 Tabel Alias
Gambar 40 Tabel Alias
Gambar 41 Alias
Gambar 42 Operator Between
Gambar 42 Operator In
Gambar 43 Operator Like
Gambar 44 Operator Not In
Gambar 45 Source Code Function Count
Gambar 46 Source Code Lower, Now, Upper
Gambar 47 Source Code Grouping
Gambar 48 Source Code Sorting
Gambar 49 Function Count
Gambar 50 Function Lower
Gambar 51 Function Now
Gambar 52 Function Upper
Gambar 53 Grouping
Gambar 54 Grouping
Gambar 55 Grouping
Gambar 56 Grouping
Gambar 57 Sorting
Gambar 58 Sorting
Gambar 59 Source Code View
Gambar 60 Source Code View
Gambar 61 View
Gambar 62 View
Gambar 63 View
Gambar 64 Source Code Store Procedure
Gambar 65 Source Code Store Procedure
Gambar 66 Source Code Store Procedure
Gambar 67 Store Procedure
Gambar 68 Store Procedure
Gambar 69 Store Procedure

Gambar 70 Source Code Nested Query
Gambar 71 Nested Query
Gambar 72 Nested Query
Gambar 73 Trigger
Gambar 74 Trigger
Gambar 75 Source Code DCL
Gambar 76 Grant
Gambar 77 Revoke
Gambar 78 Grant Dan Revoke di Cmd

# KATA PENGANTAR

Puji syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa atas segala rahmat, hidayah, dan karunia-Nya yang senantiasa melimpahkan keberkahan dalam setiap langkah perjalanan hidup kami. Dengan penuh rasa syukur, kami ingin mengucapkan terima kasih kepada semua pihak yang telah memberikan dukungan dan bantuan dalam penyusunan Laporan Tugas Besar ini. Laporan ini merupakan buah dari proses pembelajaran yang intensif dan penuh dedikasi selama mengikuti praktikum ini, yang tidak hanya memberikan pengetahuan teoritis, tetapi juga pengalaman praktis yang berharga

Laporan ini dibuat untuk memenuhi tugas mata kuliah proyek Basis Data Lanjut. Kami mengucapkan terima kasih kepada Abang Wahyu dan Abang Torang selaku asisten dosen kami yang telah membantu kami baik secara moral dan materi sehingga kami dapat menyelesaikan tugas laporan ini. Kami juga mengucapkan terima kasih kepada teman-teman kami yang telah membantu dalam pembuatan laporan ini.

Akhir kata, kami menyadari bahwa laporan ini jauh dari kesempurnaan. Oleh karena itu, kami sangat mengharapkan masukan dan saran yang membangun dari pembaca demi perbaikan dan pengembangan lebih lanjut. Semoga laporan ini bermanfaat bagi semua pihak yang terlibat dan dapat menjadi inspirasi bagi karya-karya yang lebih baik di masa mendatang.

Bengkulu, 6 Mei 2025

Kelompok 1

# BAB I
# PENDAHULUAN

## 1.1 Latar Belakang

Basis data merupakan sekumpulan data yang dikelola secara sistematis sehingga dapat diakses, dikelola, dan diperbarui dengan mudah. Dalam implementasinya, basis data digunakan untuk menyimpan informasi penting seperti data pelanggan, transaksi keuangan, inventaris barang, catatan medis, dan sebagainya. Sistem basis data relasional (Relational Database Management System/RDBMS) telah menjadi fondasi utama dalam pengelolaan data sejak diperkenalkan oleh Edgar F. Codd pada tahun 1970. Sistem ini memungkinkan organisasi untuk menyimpan data dalam bentuk tabel-tabel yang saling berelasi dan memudahkan dalam proses query menggunakan bahasa SQL (Structured Query Language). (Nanda Arista Rizki, 2019).

Namun, seiring dengan semakin kompleksnya kebutuhan organisasi dan meningkatnya volume data, pendekatan basis data tradisional sering kali tidak lagi mencukupi. Permasalahan seperti performa *query* yang lambat, kesulitan dalam menangani data tidak terstruktur, kebutuhan akan transaksi yang kompleks, serta kebutuhan integrasi dengan sistem lain mendorong munculnya konsep dan pendekatan yang lebih maju dalam pengelolaan data. Inilah yang menjadi dasar munculnya bidang kajian Basis Data Lanjut atau *advanced database*.

Basis Data Lanjut merupakan kelanjutan dari pemahaman dasar sistem basis data relasional, namun mencakup topik-topik yang lebih mendalam dan kompleks. Dalam mata kuliah atau kajian ini, mahasiswa atau praktisi akan mempelajari berbagai teknik dan pendekatan baru yang dirancang untuk meningkatkan efisiensi, skalabilitas, keamanan, dan kemampuan analitik dari sistem basis data. Di dalamnya mencakup pembahasan mengenai optimisasi *query*, basis data terdistribusi, basis data paralel, pemodelan data tingkat lanjut, basis data berorientasi objek, hingga teknologi basis data NoSQL. (Wibowo, 2019).

Optimisasi *query* menjadi salah satu fokus utama karena performa sistem basis data sangat ditentukan oleh bagaimana *query* dijalankan oleh sistem. Pada sistem dengan jutaan hingga miliaran baris data, *query* yang tidak dioptimalkan bisa memakan waktu sangat lama dan membebani server. Oleh karena itu, dalam basis data lanjut dipelajari cara-cara optimisasi seperti penggunaan indeks, pemrosesan *query* berbasis *cost*, serta pemilihan jalur eksekusi (query execution plan) yang paling efisien. Selain itu, konsep basis data terdistribusi dan paralel juga menjadi penting untuk menjawab tantangan penyimpanan dan pemrosesan data berskala besar. Dalam basis data terdistribusi, data disimpan di beberapa lokasi fisik yang terhubung melalui jaringan, namun tetap dikelola seolah-olah berada

dalam satu sistem logis. Hal ini memungkinkan organisasi untuk mempercepat pemrosesan data dan meningkatkan ketahanan sistem. Sedangkan basis data paralel memanfaatkan teknologi multiprosesor untuk melakukan operasi-operasi basis data secara bersamaan, sehingga meningkatkan kinerja sistem secara signifikan. (Rahman, 2011).

Tidak hanya terbatas pada pendekatan relasional, dalam Basis Data Lanjut juga dibahas mengenai pendekatan-pendekatan baru seperti basis data NoSQL (Not Only SQL). Dengan cakupan yang luas dan tantangan yang kompleks, penguasaan materi Basis Data Lanjut sangat penting bagi mahasiswa. Pemahaman yang mendalam terhadap struktur data, teknik penyimpanan, mekanisme query, dan teknologi modern akan memberikan landasan yang kuat untuk membangun sistem informasi yang handal dan siap menghadapi tantangan era digital. (Purwanto, 2017).

Dalam era digital saat ini, pengelolaan data menjadi aspek krusial dalam mendukung keberhasilan operasional berbagai institusi, termasuk perguruan tinggi. Kampus sebagai lembaga pendidikan tinggi modern tidak lagi mengandalkan pencatatan manual dalam mengelola data akademik dan keuangan mahasiswa. Sebaliknya, mereka mengimplementasikan sistem basis data yang canggih untuk menyimpan, mengelola, dan mengakses informasi akademik serta keuangan secara cepat dan akurat. Salah satu bentuk penerapan nyata basis data dalam dunia pendidikan tinggi adalah pada sistem akademik dan manajemen pembayaran Uang Kuliah Tunggal (UKT) mahasiswa.

Sistem ini harus mampu menangani berbagai transaksi penting seperti pengisian Kartu Rencana Studi (KRS), pencatatan nilai, jadwal perkuliahan, serta verifikasi status pembayaran UKT sebagai prasyarat pengambilan mata kuliah. Seiring dengan meningkatnya jumlah mahasiswa dan kompleksitas layanan kampus, sistem informasi akademik dan keuangan memerlukan struktur data dan proses yang tidak lagi cukup jika hanya mengandalkan sistem basis data relasional konvensional. Permintaan untuk akses data real-time, integrasi antar sistem (akademik, keuangan, perpustakaan, layanan kemahasiswaan), serta kebutuhan keamanan data yang tinggi menuntut pendekatan yang lebih maju dan efisien dalam pengelolaan basis data. (Sigit Indrawijaya, 2024).

Dengan pendekatan basis data lanjut, institusi pendidikan dapat menerapkan optimisasi *query* yang kompleks, pengelolaan data secara terdistribusi antar fakultas, hingga penggunaan sistem pencatatan transaksi (transaction log) yang menjamin integritas data ketika terjadi pemrosesan pembayaran UKT atau perubahan status akademik mahasiswa. Sistem pembayaran UKT yang terintegrasi dengan sistem akademik juga memerlukan fitur pelacakan histori transaksi, notifikasi keterlambatan, dan pencatatan cicilan atau beasiswa yang diberikan. Semua itu harus berjalan secara otomatis, konsisten, dan tersedia kapan

pun dibutuhkan oleh mahasiswa maupun staf administrasi. Penggunaan fitur-fitur seperti *stored procedure*, *trigger*, dan lain-lain, dalam basis data lanjut dapat menjaga agar data akademik dan keuangan mahasiswa tetap sinkron, bahkan ketika diakses oleh banyak pengguna secara bersamaan. Dengan demikian, penguasaan materi Basis Data Lanjut bukan hanya penting dari sisi teoritis, tetapi juga sangat aplikatif dalam membangun sistem informasi kampus yang modern, cerdas, dan adaptif terhadap kebutuhan dinamis dunia pendidikan. (Gupta, 2005).

Dalam implementasi sistem basis data, dibutuhkan alat bantu (tool) yang tidak hanya mempermudah proses pengembangan dan perancangan, tetapi juga mampu memvisualisasikan struktur data, menjalankan query SQL, serta mengelola koneksi ke server basis data. Salah satu *tools* yang paling populer dan banyak digunakan dalam dunia pendidikan dan industri adalah MySQL Workbench. MySQL Workbench merupakan *interface* grafis terpadu (GUI) resmi dari MySQL yang menyediakan berbagai fitur lengkap untuk keperluan desain, pengelolaan, dan administrasi basis data.

Dengan Workbench, pengembang basis data dapat melakukan pemodelan data secara visual menggunakan ERD (Entity Relationship Diagram), yang memungkinkan identifikasi entitas seperti *Mahasiswa*, *Dosen*, *Mata Kuliah*, *Pembayaran UKT*, serta hubungan antar tabel dengan lebih mudah dan terstruktur.

## 1.2 Rumusan Masalah

1. Bagaimana Basis Data Lanjut dapat diterapkan untuk amengoptimalkan sistem informasi akademik dan manajemen pembayaran UKT mahasiswa?
2. Bagaimana mengimplementasikan fitur-fitur basis data lanjut seperti stored procedure, trigger, dan view, untuk mendukung otomatisasi dan konsistensi sistem akademik dan manajemen pembayaran UKT?

## 1.3 Tujuan Penulisan

1. Untuk membangun sistem akademik dan sistem manajemen pembayaran UKT mahasiswa yang terintegrasi menggunakan pendekatan basis data lanjut. sehingga dapat mendukung proses pengelolaan data akademik dan keuangan secara efisien.
2. Untuk menguji efektivitas rancangan sistem basis data dalam menangani skenario nyata seperti verifikasi pembayaran, pengambilan mata kuliah, dan pelaporan data akademik mahasiswa, juga dilakukan analisis mendalam terhadap kinerja sistem untuk memastikan bahwa integritas data dan kecepatan respon tetap terjaga.
3. Untuk mengimplementasikan fitur-fitur basis data lanjut seperti stored procedure, trigger, dan view guna meningkatkan efisiensi, otomatisasi, konsistensi data dalam sistem, dan akurasi pengolahan data.

# BAB II
# LANDASAN TEORI

## 2.1 Dasar Teori

### A. Basis Data Lanjut

Basis data lanjut adalah salah satu cabang dalam ilmu komputer yang membahas tentang pengelolaan, penyimpanan, dan pemrosesan data secara efisien dengan memanfaatkan teknologi dan metode yang lebih kompleks dibandingkan dengan basis data dasar. Tujuan utama dari basis data lanjut adalah untuk memberikan solusi terhadap permasalahan yang tidak dapat diselesaikan dengan pendekatan konvensional dalam pengelolaan data. Basis data dasar mencakup model data relasional yang mengandalkan tabel untuk menyimpan informasi, namun, pada tingkat lanjut, dibutuhkan pemahaman yang lebih mendalam mengenai struktur data. (Wibowo, 2019).

Pada basis data lanjut, kita mengenal berbagai jenis model data selain relasional, seperti model hierarkis, jaringan, dan objek. Selain itu, topik-topik seperti normalisasi, pengindeksan, transaksi, dan query optimization menjadi hal yang sangat penting. Pengindeksan, misalnya, bertujuan untuk mempercepat proses pencarian data dengan menggunakan struktur data yang efisien, seperti pohon B (B-tree) atau pohon hash (hash tree). Transaksi pada basis data juga menjadi salah satu aspek penting dalam menjamin konsistensi dan integritas data, serta untuk mendukung kebutuhan aplikasi yang membutuhkan tingkat keandalan tinggi.

Di sisi lain, dengan berkembangnya teknologi informasi, munculnya data dalam jumlah besar (big data), serta kebutuhan akan pemrosesan data yang lebih kompleks, basis data lanjut juga mencakup sistem manajemen basis data terdistribusi (DDBMS), basis data objek (OODBMS), dan basis data NoSQL. Basis data terdistribusi memanfaatkan jaringan komputer untuk menyebarkan data pada berbagai lokasi fisik, sedangkan basis data NoSQL menawarkan fleksibilitas dalam menangani data yang tidak terstruktur atau semi-terstruktur, serta mendukung skalabilitas yang tinggi untuk aplikasi-aplikasi modern yang memerlukan penyimpanan data yang besar dan cepat. (Elmasri, 2015).

Secara keseluruhan, basis data lanjut bukan hanya berfokus pada penyimpanan dan pengambilan data yang efisien, tetapi juga berperan dalam meningkatkan kinerja aplikasi melalui optimasi *query*, pengelolaan transaksi yang lebih canggih, dan penerapan metode-metode yang lebih mutakhir dalam pengolahan data. Oleh karena itu, penguasaan teori dan praktik dalam basis data lanjut sangat diperlukan untuk dapat merancang sistem yang efektif, dapat diandalkan, dan juga mampu menangani volume data yang besar, menjaga konsistensi, serta memberikan performa yang optimal. (Papuangan, 2023).

## B. Sistem Akademik Dan Manajemen Pembayaran UKT Mahasiswa

Sistem akademik dan manajemen pembayaran Uang Kuliah Tunggal (UKT) mahasiswa merupakan dua aspek penting dalam mendukung kelancaran operasional suatu perguruan tinggi. Sistem akademik mencakup berbagai aspek yang berhubungan dengan administrasi pendidikan, seperti pendaftaran mata kuliah, perwalian akademik, pengisian KRS (Kartu Rencana Studi), penilaian, serta pengelolaan data mahasiswa. Tujuan utama dari sistem akademik adalah untuk memastikan proses pendidikan berjalan dengan efisien, transparan, dan terstruktur. Dengan adanya sistem ini, mahasiswa, dosen, dan pihak administrasi dapat mengakses data akademik secara cepat dan akurat, yang akan mendukung pengambilan keputusan yang tepat dalam pengelolaan pendidikan. (Wiwin Yuni Isnaeni, 2024).

Selain itu, sistem akademik juga berfungsi untuk memonitor kemajuan akademik mahasiswa, memastikan kelancaran proses belajar mengajar, dan mengintegrasikan berbagai layanan yang dibutuhkan oleh mahasiswa dalam menjalani kegiatan perkuliahan. Di sisi lain, manajemen pembayaran UKT mahasiswa merupakan bagian integral dari sistem administrasi perguruan tinggi yang bertujuan untuk mengatur dan mengelola pembayaran biaya pendidikan yang dibebankan kepada mahasiswa. UKT sendiri adalah sistem pembayaran tunggal yang mencakup seluruh biaya pendidikan, termasuk biaya perkuliahan, ujian, dan fasilitas lainnya. Sistem manajemen pembayaran UKT dirancang untuk memudahkan mahasiswa dalam melakukan pembayaran, baik secara online maupun offline, serta memastikan bahwa setiap transaksi tercatat dengan baik. (Annisa Mayasari, 2021).

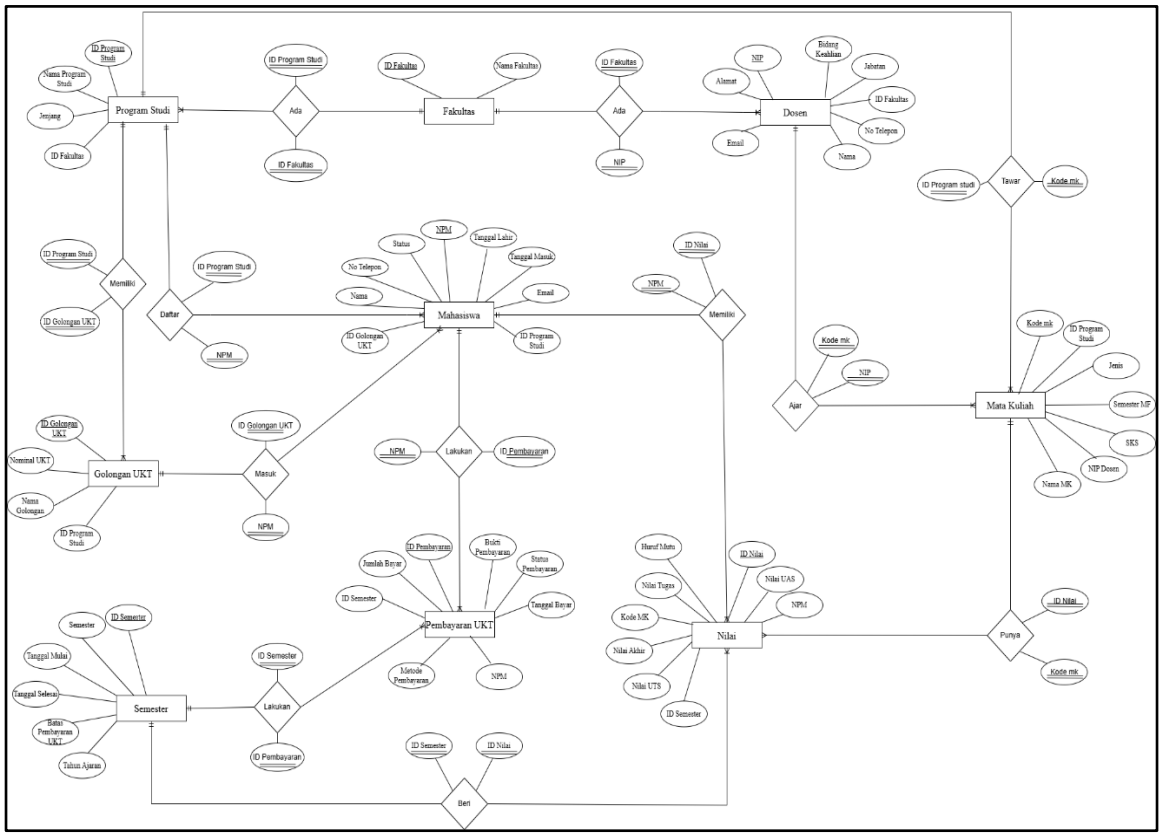
Kemudian, sistem ini juga berperan dalam memantau status keuangan mahasiswa, seperti pembayaran yang telah dilakukan, saldo yang masih harus dibayar, dan keterlambatan pembayaran. Dengan adanya sistem manajemen UKT yang terintegrasi, perguruan tinggi dapat lebih mudah dalam mengelola keuangan, menyusun laporan keuangan yang akurat, serta memberikan informasi yang transparan kepada mahasiswa mengenai status pembayaran mereka. (Sigit Indrawijaya, 2024).

Kedua sistem ini, yaitu sistem akademik dan manajemen pembayaran UKT, keduanya saling mendukung dalam menjalankan operasional perguruan tinggi. Sistem akademik yang efektif memerlukan integrasi dengan sistem pembayaran UKT, agar mahasiswa yang telah memenuhi kewajiban keuangan dapat melanjutkan studi mereka tanpa hambatan. Sebaliknya, sistem manajemen pembayaran UKT yang baik akan memastikan bahwa perguruan tinggi memiliki dana yang cukup untuk mendukung kegiatan akademik dan operasional lainnya. Oleh karena itu, pengembangan dan penerapan kedua sistem ini harus dilakukan dengan baik dan saling terhubung untuk menciptakan proses pendidikan yang lebih efisien, terstruktur, dan mudah diakses oleh semua pihak. (Purwanto, 2017).

# BAB III

# HASIL DAN PEMBAHASAN

## 3.1 Buat ERD




Gambar 1 ERD Sistem Akademik & Manajemen Pembayaran UKT

**Pembahasan Gambar :**

Pada gambar ERD (Entity Relationship Diagram) di atas menggambarkan sistem akademik yang mengelola data mahasiswa, dosen, mata kuliah, pembayaran UKT, dan nilai. Sistem ini terdiri dari beberapa entitas utama, yaitu mahasiswa, program studi, fakultas, golongan ukt, pembayaran ukt, semester, nilai, mata kuliah, dan dosen. Relasi antar entitas menunjukkan hubungan, beserta atribut-atribut pendukungnya. Entitas pertama yakni, mahasiswa yang memiliki atribut seperti npm, nama, status, tempat lahir, tanggal lahir, email, no telepon, dan ID program studi serta ID golongan UKT sebagai foreign key. Mahasiswa berelasi dengan entitas Program Studi melalui relasi daftar, yang memiliki kardinalitas *many to one* (>| - ||), artinya banyak mahasiswa dapat mendaftar di satu program studi.

Mahasiswa juga berelasi dengan golongan ukt yang memiliki kardinalitas *many to one* (>| - ||), artinya satu golongan UKT dapat digunakan oleh banyak mahasiswa. Mahasiswa juga memiliki hubungan dengan entitas pembayaran UKT melalui relasi lakukan, dengan kardinalitas *one to many* (|| - >|), menunjukkan bahwa satu mahasiswa dapat melakukan banyak pembayaran UKT. Mahasiswa juga berelasi dengan entitas Nilai melalui relasi

memiliki dengan kardinalitas *one to many* (|| - >|), menunjukkan bahwa satu mahasiswa dapat memiliki banyak data nilai. Entitas program studi memiliki atribut seperti id_program studi, nama program studi, dan jenjang. Program studi juga berelasi dengan entitas Mata Kuliah melalui relasi tawar dengan kardinalitas *one to many* (|| - >|), artinya satu program studi menawarkan banyak mata kuliah.

Entitas fakultas memiliki atribut id dan nama fakultas. Fakultas memiliki relasi *one to many* (|| - >|) dengan entitas program studi, karena satu fakultas menaungi banyak program studi. Selain itu, fakultas juga berelasi dengan Dosen melalui relasi ada, dengan kardinalitas *one to many* (|| - >|), artinya satu fakultas memiliki banyak dosen. Entitas golongan UKT memiliki atribut id golongan ukt dan nama golongan. Golongan UKT terhubung dengan program studi melalui relasi menentukan dengan kardinalitas *many to one* (>| - ||), menunjukkan bahwa banyak golongan dapat ditentukan oleh satu program studi. Entitas pembayaran UKT memiliki atribut seperti id pembayaran, jumlah bayar, metode pembayaran, bukti pembayaran, status pembayaran, dan tanggal bayar. Pembayaran UKT berhubungan dengan entitas semester melalui relasi berada dengan kardinalitas *many to one* (>| - ||), menunjukkan bahwa satu semester bisa memiliki banyak transaksi pembayaran.

Entitas Semester memiliki atribut id semester, semester, tanggal mulai, tanggal selesai, dan tahun ajaran. Semester juga berelasi dengan entitas nilai melalui relasi berada, menunjukkan bahwa setiap semester memuat banyak data nilai, dengan kardinalitas *one to many* (|| - >|). Entitas nilai mencatat data penilaian mahasiswa dalam suatu mata kuliah di semester tertentu, dengan atribut seperti id nilai, nilai tugas, nilai uts, nilai uas, nilai akhir, dan huruf mutu. Nilai berelasi dengan entitas mahasiswa, mata kuliah, dan semester, melalui relasi memiliki, punya, dan lakukan, dengan masing-masing kardinalitas nya *many to one* (>| - ||), karena banyak nilai bisa dimiliki oleh satu mahasiswa, satu mata kuliah, dan satu semester.

Entitas mata kuliah memiliki atribut seperti kode mk, nama mk, sks, jenis, dan semester mk. Setiap mata kuliah diajarkan oleh seorang dosen melalui relasi ajar, dengan kardinalitasnya yaitu *many to one* (>| - ||). Mata kuliah juga berhubungan dengan program studi, menunjukkan bahwa satu program studi dapat memiliki banyak mata kuliah dan berelasi punya nilai. Terakhir, entitas dosen memiliki atribut seperti nip, nama, alamat, email, no telepon, dan bidang keahlian. Dosen terhubung dengan entitas mata kuliah melalui relasi ajar, dengan kardinalitas *many to one* (>| - ||), menunjukkan bahwa satu dosen bisa mengajar banyak mata kuliah. Dosen juga berelasi dengan fakultas, karena setiap dosen berada di satu fakultas, dengan kardinalitas *many to one* (>| - ||).

3.2 Buat database, tabel dan masukkan data.

**Printscreen source code :**

```
-- Database sistem_akademik
CREATE DATABASE sistem_akademik;
USE sistem_akademik;
```

Gambar 3 Source Code Create Database

```
-- TABEL FAKULTAS
CREATE TABLE fakultas (
    id_fakultas INT PRIMARY KEY,
    nama_fakultas VARCHAR(100) NOT NULL
);
-- TABEL PROGRAM STUDI
CREATE TABLE program_studi (
    id_program_studi VARCHAR(10) PRIMARY KEY,
    nama_program_studi VARCHAR(100) NOT NULL,
    jenjang VARCHAR(20),
    id_fakultas INT,
    FOREIGN KEY (id_fakultas) REFERENCES fakultas(id_fakultas)
);
-- TABEL GOLONGAN UKT
CREATE TABLE golongan_ukt (
    id_golongan_ukt INT PRIMARY KEY,
    nama_golongan VARCHAR(50),
    nominal_ukt DECIMAL(12,2),
    id_program_studi VARCHAR(10),
    FOREIGN KEY (id_program_studi) REFERENCES program_studi(id_program_studi)
);
```

Gambar 2 Source Code Tabel Fakultas, Program Studi, Golongan UKT

```
-- TABEL MAHASISWA
CREATE TABLE mahasiswa (
    npm VARCHAR(15) PRIMARY KEY,
    nama VARCHAR(100),
    alamat TEXT,
    no_telepon VARCHAR(15),
    email VARCHAR(100),
    tanggal_lahir DATE,
    tanggal_masuk DATE,
    status VARCHAR(20),
    id_program_studi VARCHAR(10),
    id_golongan_ukt INT,
    FOREIGN KEY (id_program_studi) REFERENCES program_studi(id_program_studi),
    FOREIGN KEY (id_golongan_ukt) REFERENCES golongan_ukt(id_golongan_ukt)
);
```

Gambar 4 Source Code Tabel Mahasiswa

```sql
-- TABEL DOSEN
CREATE TABLE dosen (
    nip VARCHAR(20) PRIMARY KEY,
    nama VARCHAR(100),
    alamat TEXT,
    no_telepon VARCHAR(15),
    email VARCHAR(100),
    jabatan VARCHAR(50),
    bidang_keahlian VARCHAR(100),
    id_fakultas INT,
    FOREIGN KEY (id_fakultas) REFERENCES fakultas(id_fakultas)
);
-- TABEL MATA KULIAH
CREATE TABLE mata_kuliah (
    kode_mk VARCHAR(10) PRIMARY KEY,
    nama_mk VARCHAR(100),
    sks INT,
    semester_mf INT,
    jenis VARCHAR(50),
    id_program_studi VARCHAR(10),
    nip_dosen VARCHAR(20),
    FOREIGN KEY (id_program_studi) REFERENCES program_studi(id_program_studi),
    FOREIGN KEY (nip_dosen) REFERENCES dosen(nip)
);
```

Gambar 5 Source Code Create Table Dosen dan Mata Kuliah

```sql
-- TABEL SEMESTER
CREATE TABLE semester (
    id_semester VARCHAR(10) PRIMARY KEY,
    tahun_ajaran VARCHAR(20),
    semester VARCHAR(10),
    tanggal_mulai DATE,
    tanggal_selesai DATE,
    batas_pembayaran_ukt DATE
);
-- TABEL NILAI
CREATE TABLE nilai (
    id_nilai INT PRIMARY KEY,
    npm VARCHAR(15),
    kode_mk VARCHAR(10),
    id_semester VARCHAR(10),
    nilai_tugas FLOAT,
    nilai_uts FLOAT,
    nilai_uas FLOAT,
    nilai_akhir FLOAT,
    huruf_mutu VARCHAR(2),
    FOREIGN KEY (npm ) REFERENCES mahasiswa(npm ),
    FOREIGN KEY (kode_mk) REFERENCES mata_kuliah(kode_mk),
    FOREIGN KEY (id_semester) REFERENCES semester(id_semester)
);
```

Gambar 6 Source Code Create Table Semester dan Nilai

```sql
-- TABEL PEMBAYARAN UKT
CREATE TABLE pembayaran_ukt (
    id_pembayaran INT PRIMARY KEY,
    npm VARCHAR(15),
    id_semester VARCHAR(10),
    tanggal_bayar DATE,
    jumlah_bayar DECIMAL(12,2),
    status_pembayaran VARCHAR(20),
    metode_pembayaran VARCHAR(50),
    bukti_pembayaran TEXT,
    FOREIGN KEY (npm ) REFERENCES mahasiswa(npm ),
    FOREIGN KEY (id_semester) REFERENCES semester(id_semester)
);
```

Gambar 7 Source Code Create Table Pembayaran UKT

```sql
INSERT INTO fakultas (id_fakultas, nama_fakultas) VALUES
(1, 'Fakultas Teknik'),
(2, 'Fakultas Hukum'),
(3, 'Fakultas Ekonomi dan Bisnis'),
(4, 'Fakultas Ilmu Sosial dan Politik'),
(5, 'Fakultas Pertanian'),
(6, 'Fakultas Matematika dan Ilmu Pengetahuan Alam'),
(7, 'Fakultas Kedokteran'),
(8, 'Fakultas Keguruan dan Ilmu Pendidikan');

-- Untuk Program Studi // Fakultas Teknik --
INSERT INTO program_studi (id_program_studi, nama_program_studi, jenjang, id_fakultas) VALUES
('G1A', 'Informatika', 'S1', 1),
('G1B', 'Teknik Elektro', 'S1', 1),
('G1C', 'Teknik Sipil', 'S1', 1),
('G1D', 'Teknik Mesin', 'S1', 1),
('G1E', 'Arsitektur', 'S1', 1);

-- Fakultas Hukum
INSERT INTO program_studi (id_program_studi, nama_program_studi, jenjang, id_fakultas) VALUES
('B1A', 'Hukum Perdata', 'S1', 2),
('B1B', 'Hukum Pidana', 'S1', 2),
('B1C', 'Hukum Tata Negara', 'S1', 2),
('B1D', 'Hukum Internasional', 'S1', 2),
('B1E', 'Hukum Ekonomi', 'S1', 2);
```

Gambar 8 Source Code Insert Into Tabel Fakultas dan Program Studi

```sql
-- Fakultas Ekonomi dan Bisnis
INSERT INTO program_studi (id_program_studi, nama_program_studi, jenjang, id_fakultas) VALUES
('C1A', 'Manajemen', 'S1', 3),
('C1B', 'Akuntansi', 'S1', 3),
('C1C', 'Ekonomi Pembangunan', 'S1', 3),
('C1D', 'Ilmu Ekonomi', 'S1', 3),
('C1E', 'Bisnis Internasional', 'S1', 3);

-- Fakultas Ilmu Sosial dan Politik
INSERT INTO program_studi (id_program_studi, nama_program_studi, jenjang, id_fakultas) VALUES
('D1A', 'Ilmu Politik', 'S1', 4),
('D1B', 'Sosiologi', 'S1', 4),
('D1C', 'Antropologi', 'S1', 4),
('D1D', 'Ilmu Komunikasi', 'S1', 4),
('D1E', 'Psikologi', 'S1', 4);

-- Fakultas Pertanian
INSERT INTO program_studi (id_program_studi, nama_program_studi, jenjang, id_fakultas) VALUES
('E1A', 'Agribisnis', 'S1', 5),
('E1B', 'Teknologi Pertanian', 'S1', 5),
('E1C', 'Ilmu Tanah', 'S1', 5),
('E1D', 'Pendidikan Pertanian', 'S1', 5),
('E1E', 'Penyuluhan Pertanian', 'S1', 5);
```

Gambar 9 Source Code Insert Into Tabel Program Studi

```sql
-- Fakultas Matematika dan Ilmu Pengetahuan Alam
INSERT INTO program_studi (id_program_studi, nama_program_studi, jenjang, id_fakultas) VALUES
('F1A', 'Matematika', 'S1', 6),
('F1B', 'Fisika', 'S1', 6),
('F1C', 'Kimia', 'S1', 6),
('F1D', 'Biologi', 'S1', 6),
('F1E', 'Statistika', 'S1', 6);

-- Fakultas Kedokteran
INSERT INTO program_studi (id_program_studi, nama_program_studi, jenjang, id_fakultas) VALUES
('H1A', 'Kedokteran Umum', 'S1', 7),
('H1B', 'Kedokteran Gigi', 'S1', 7),
('H1C', 'Kedokteran Hewan', 'S1', 7),
('H1D', 'Farmasi', 'S1', 7),
('H1E', 'Keperawatan', 'S1', 7);

-- Fakultas Keguruan dan Ilmu Pendidikan
INSERT INTO program_studi (id_program_studi, nama_program_studi, jenjang, id_fakultas) VALUES
('A1A', 'Pendidikan Matematika', 'S1', 8),
('A1B', 'Pendidikan Bahasa Indonesia', 'S1', 8),
('A1C', 'Pendidikan Bahasa Inggris', 'S1', 8),
('A1D', 'Pendidikan Fisika', 'S1', 8),
('A1E', 'Pendidikan Kimia', 'S1', 8);
```

Gambar 10 Source Code Insert Into Tabel Program Studi

```sql
INSERT INTO golongan_ukt (id_golongan_ukt, nama_golongan, nominal_ukt, id_program_studi) VALUES
(1, 'Golongan A', 2000000.00, 'G1A'),
(2, 'Golongan B', 2500000.00, 'G1B'),
(3, 'Golongan C', 3000000.00, 'G1C'),
(4, 'Golongan D', 3500000.00, 'G1D'),
(5, 'Golongan E', 4000000.00, 'G1E');
-- Fakultas Hukum
INSERT INTO golongan_ukt (id_golongan_ukt, nama_golongan, nominal_ukt, id_program_studi) VALUES
(6, 'Golongan A', 2000000.00, 'B1A'),
(7, 'Golongan B', 2500000.00, 'B1B'),
(8, 'Golongan C', 3000000.00, 'B1C'),
(9, 'Golongan D', 3500000.00, 'B1D'),
(10, 'Golongan E', 4000000.00, 'B1E');
-- Fakultas Ekonomi dan Bisnis
INSERT INTO golongan_ukt (id_golongan_ukt, nama_golongan, nominal_ukt, id_program_studi) VALUES
(11, 'Golongan A', 2000000.00, 'C1A'),
(12, 'Golongan B', 2500000.00, 'C1B'),
(13, 'Golongan C', 3000000.00, 'C1C'),
(14, 'Golongan D', 3500000.00, 'C1D'),
(15, 'Golongan E', 4000000.00, 'C1E');
-- Fakultas Ilmu Sosial dan Politik
INSERT INTO golongan_ukt (id_golongan_ukt, nama_golongan, nominal_ukt, id_program_studi) VALUES
(16, 'Golongan A', 2000000.00, 'D1A'),
(17, 'Golongan B', 2500000.00, 'D1B'),
(18, 'Golongan C', 3000000.00, 'D1C'),
(19, 'Golongan D', 3500000.00, 'D1D'),
(20, 'Golongan E', 4000000.00, 'D1E');
```

Gambar 11 Source Code Insert Into Tabel Golongan UKT

```sql
-- Fakultas Pertanian
INSERT INTO golongan_ukt (id_golongan_ukt, nama_golongan, nominal_ukt, id_program_studi) VALUES
(21, 'Golongan A', 2000000.00, 'E1A'),
(22, 'Golongan B', 2500000.00, 'E1B'),
(23, 'Golongan C', 3000000.00, 'E1C'),
(24, 'Golongan D', 3500000.00, 'E1D'),
(25, 'Golongan E', 4000000.00, 'E1E');
-- Fakultas Matematika dan Ilmu Pengetahuan Alam
INSERT INTO golongan_ukt (id_golongan_ukt, nama_golongan, nominal_ukt, id_program_studi) VALUES
(26, 'Golongan A', 2000000.00, 'F1A'),
(27, 'Golongan B', 2500000.00, 'F1B'),
(28, 'Golongan C', 3000000.00, 'F1C'),
(29, 'Golongan D', 3500000.00, 'F1D'),
(30, 'Golongan E', 4000000.00, 'F1E');
-- Fakultas Kedokteran
INSERT INTO golongan_ukt (id_golongan_ukt, nama_golongan, nominal_ukt, id_program_studi) VALUES
(31, 'Golongan A', 2000000.00, 'H1A'),
(32, 'Golongan B', 2500000.00, 'H1B'),
(33, 'Golongan C', 3000000.00, 'H1C'),
(34, 'Golongan D', 3500000.00, 'H1D'),
(35, 'Golongan E', 4000000.00, 'H1E');
-- Fakultas Keguruan dan Ilmu Pendidikan
INSERT INTO golongan_ukt (id_golongan_ukt, nama_golongan, nominal_ukt, id_program_studi) VALUES
(36, 'Golongan A', 2000000.00, 'A1A'),
(37, 'Golongan B', 2500000.00, 'A1B'),
(38, 'Golongan C', 3000000.00, 'A1C'),
(39, 'Golongan D', 3500000.00, 'A1D'),
(40, 'Golongan E', 4000000.00, 'A1E');
```

Gambar 12 Source Code Insert Into Tabel Golongan UKT

```sql
-- Fakultas Teknik
INSERT INTO mahasiswa (npm, nama, alamat, no_telepon, email, tanggal_lahir,
tanggal_masuk, status, id_program_studi, id_golongan_ukt) VALUES
('G1A023049', 'Andi Rahmat', 'Jl. Raya No.1', '08123456789',
'andi@gmail.com', '2000-01-15', '2020-09-01', 'Aktif', 'G1A', 1),
('G1B023050', 'Budi Setiawan', 'Jl. Teknik No.3', '08234567890',
'budi@gmail.com', '1999-02-20', '2020-09-01', 'Aktif', 'G1B', 2),
('G1C023051', 'Cindy Yuliana', 'Jl. Sipil No.4', '08345678901',
'cindy@gmail.com', '1998-03-25', '2020-09-01', 'Aktif', 'G1C', 3),
('G1D023052', 'Diana Utami', 'Jl. Mesin No.5', '08456789012',
'diana@gmail.com', '2001-04-30', '2021-09-01', 'Aktif', 'G1D', 4),
('G1E023053', 'Erwin Pratama', 'Jl. Arsitektur No.6', '08567890123',
'erwin@gmail.com', '2000-05-10', '2021-09-01', 'Aktif', 'G1E', 5);
-- Fakultas Hukum
INSERT INTO mahasiswa (npm, nama, alamat, no_telepon, email, tanggal_lahir,
tanggal_masuk, status, id_program_studi, id_golongan_ukt) VALUES
('B1A023054', 'Farah Dewi', 'Jl. Hukum No.7', '08678901234',
'farah@gmail.com', '2000-06-12', '2020-09-01', 'Aktif', 'B1A', 6),
('B1B023055', 'Gilang Sutrisno', 'Jl. Pidana No.8', '08789012345',
'gilang@gmail.com', '1999-07-14', '2020-09-01', 'Aktif', 'B1B', 7),
('B1C023056', 'Hannah Nur', 'Jl. Tata Negara No.9', '08890123456',
'hannah@gmail.com', '1998-08-17', '2020-09-01', 'Aktif', 'B1C', 8),
('B1D023057', 'Ikhsan Mulyadi', 'Jl. Internasional No.10', '08901234567',
'ikhsan@gmail.com', '2001-09-21', '2021-09-01', 'Aktif', 'B1D', 9),
('B1E023058', 'Jasmine Amalia', 'Jl. Ekonomi No.11', '09012345678',
'jasmine@gmail.com', '2000-10-25', '2021-09-01', 'Aktif', 'B1E', 10);
```

Gambar 13 Source Code Insert Into Tabel Mahasiswa

```sql
-- Fakultas Ekonomi dan Bisnis
INSERT INTO mahasiswa (npm, nama, alamat, no_telepon, email, tanggal_lahir,
tanggal_masuk, status, id_program_studi, id_golongan_ukt) VALUES
('C1A023059', 'Kevin Cahya', 'Jl. Manajemen No.12', '09123456789',
'kevin@gmail.com', '1999-11-30', '2020-09-01', 'Aktif', 'C1A', 11),
('C1B023060', 'Lina Mardiana', 'Jl. Akuntansi No.13', '09234567890',
'lina@gmail.com', '2000-12-14', '2020-09-01', 'Aktif', 'C1B', 12),
('C1C023061', 'Martha Desy', 'Jl. Ekonomi Pembangunan No.14', '09345678901',
'martha@gmail.com', '2001-01-17', '2021-09-01', 'Aktif', 'C1C', 13),
('C1D023062', 'Nadia Ananda', 'Jl. Ilmu Ekonomi No.15', '09456789012',
'nadia@gmail.com', '2000-02-22', '2021-09-01', 'Aktif', 'C1D', 14),
('C1E023063', 'Oscar Gunawan', 'Jl. Bisnis Internasional No.16', '09567890123',
'oscar@gmail.com', '1998-03-30', '2020-09-01', 'Aktif', 'C1E', 15);
-- Fakultas Ilmu Sosial dan Politik
INSERT INTO mahasiswa (npm, nama, alamat, no_telepon, email, tanggal_lahir,
tanggal_masuk, status, id_program_studi, id_golongan_ukt) VALUES
('D1A023064', 'Putu Satria', 'Jl. Ilmu Politik No.17', '09678901234',
'putu@gmail.com', '2000-04-02', '2020-09-01', 'Aktif', 'D1A', 16),
('D1B023065', 'Rina Ayu', 'Jl. Sosiologi No.18', '09789012345',
'rina@gmail.com', '1999-05-10', '2020-09-01', 'Aktif', 'D1B', 17),
('D1C023066', 'Siti Wulandari', 'Jl. Antropologi No.19', '09890123456',
'siti@gmail.com', '1998-06-18', '2020-09-01', 'Aktif', 'D1C', 18),
('D1D023067', 'Tina Ambarwati', 'Jl. Komunikasi No.20', '09901234567',
'tina@gmail.com', '2001-07-12', '2021-09-01', 'Aktif', 'D1D', 19),
('D1E023068', 'Umi Nabila', 'Jl. Psikologi No.21', '10012345678',
'umi@gmail.com', '2000-08-25', '2021-09-01', 'Aktif', 'D1E', 20);
```

Gambar 14 Source Code Insert Into Tabel Mahasiswa

```sql
-- Fakultas Pertanian
INSERT INTO mahasiswa (npm, nama, alamat, no_telepon, email, tanggal_lahir,
tanggal_masuk, status, id_program_studi, id_golongan_ukt) VALUES
('E1A023069', 'Vina Anjani', 'Jl. Agribisnis No.22', '10123456789',
'vina@gmail.com', '2000-09-10', '2020-09-01', 'Aktif', 'E1A', 21),
('E1B023070', 'Wawan Subagyo', 'Jl. Teknologi Pertanian No.23', '10234567890',
'wawan@gmail.com', '1999-10-15', '2020-09-01', 'Aktif', 'E1B', 22),
('E1C023071', 'Xenia Putri', 'Jl. Ilmu Tanah No.24', '10345678901',
'xenia@gmail.com', '2001-11-20', '2021-09-01', 'Aktif', 'E1C', 23),
('E1D023072', 'Yanti Amelia', 'Jl. Pendidikan Pertanian No.25', '10456789012',
 'yanti@gmail.com', '2000-12-05', '2021-09-01', 'Aktif', 'E1D', 24),
('E1E023073', 'Zaki Akbar', 'Jl. Penyuluhan Pertanian No.26', '10567890123',
'zaki@gmail.com', '1998-01-03', '2020-09-01', 'Aktif', 'E1E', 25);
-- Fakultas Matematika dan Ilmu Pengetahuan Alam
INSERT INTO mahasiswa (npm, nama, alamat, no_telepon, email, tanggal_lahir,
tanggal_masuk, status, id_program_studi, id_golongan_ukt) VALUES
('F1A023074', 'Alfiansyah Putra', 'Jl. Matematika No.27', '10678901234',
'alfiansyah@gmail.com', '1999-02-14', '2020-09-01', 'Aktif', 'F1A', 26),
('F1B023075', 'Budi Santosa', 'Jl. Fisika No.28', '10789012345',
'budi@gmail.com', '2000-03-25', '2020-09-01', 'Aktif', 'F1B', 27),
('F1C023076', 'Cahaya Indah', 'Jl. Kimia No.29', '10890123456',
'cahaya@gmail.com', '2001-04-18', '2021-09-01', 'Aktif', 'F1C', 28),
('F1D023077', 'Dani Rafi', 'Jl. Biologi No.30', '10901234567',
'dani@gmail.com', '2000-05-10', '2021-09-01', 'Aktif', 'F1D', 29),
('F1E023078', 'Elisa Dwi', 'Jl. Statistika No.31', '11012345678',
'elisa@gmail.com', '1998-06-22', '2020-09-01', 'Aktif', 'F1E', 30);
```

Gambar 15 Source Code Insert Into Tabel Mahasiswa

```sql
-- Fakultas Pertanian
INSERT INTO mahasiswa (npm, nama, alamat, no_telepon, email, tanggal_lahir,
tanggal_masuk, status, id_program_studi, id_golongan_ukt) VALUES
('E1A023069', 'Vina Anjani', 'Jl. Agribisnis No.22', '10123456789',
'vina@gmail.com', '2000-09-10', '2020-09-01', 'Aktif', 'E1A', 21),
('E1B023070', 'Wawan Subagyo', 'Jl. Teknologi Pertanian No.23', '10234567890',
'wawan@gmail.com', '1999-10-15', '2020-09-01', 'Aktif', 'E1B', 22),
('E1C023071', 'Xenia Putri', 'Jl. Ilmu Tanah No.24', '10345678901',
'xenia@gmail.com', '2001-11-20', '2021-09-01', 'Aktif', 'E1C', 23),
('E1D023072', 'Yanti Amelia', 'Jl. Pendidikan Pertanian No.25', '10456789012',
 'yanti@gmail.com', '2000-12-05', '2021-09-01', 'Aktif', 'E1D', 24),
('E1E023073', 'Zaki Akbar', 'Jl. Penyuluhan Pertanian No.26', '10567890123',
'zaki@gmail.com', '1998-01-03', '2020-09-01', 'Aktif', 'E1E', 25);
-- Fakultas Matematika dan Ilmu Pengetahuan Alam
INSERT INTO mahasiswa (npm, nama, alamat, no_telepon, email, tanggal_lahir,
tanggal_masuk, status, id_program_studi, id_golongan_ukt) VALUES
('F1A023074', 'Alfiansyah Putra', 'Jl. Matematika No.27', '10678901234',
'alfiansyah@gmail.com', '1999-02-14', '2020-09-01', 'Aktif', 'F1A', 26),
('F1B023075', 'Budi Santosa', 'Jl. Fisika No.28', '10789012345',
'budi@gmail.com', '2000-03-25', '2020-09-01', 'Aktif', 'F1B', 27),
('F1C023076', 'Cahaya Indah', 'Jl. Kimia No.29', '10890123456',
'cahaya@gmail.com', '2001-04-18', '2021-09-01', 'Aktif', 'F1C', 28),
('F1D023077', 'Dani Rafi', 'Jl. Biologi No.30', '10901234567',
'dani@gmail.com', '2000-05-10', '2021-09-01', 'Aktif', 'F1D', 29),
('F1E023078', 'Elisa Dwi', 'Jl. Statistika No.31', '11012345678',
'elisa@gmail.com', '1998-06-22', '2020-09-01', 'Aktif', 'F1E', 30);
```

Gambar 16 Source Code Insert Into Tabel Mahasiswa

```sql
-- Dosen Fakultas Teknik
INSERT INTO dosen (nip, nama, alamat, no_telepon, email, jabatan,
bidang_keahlian, id_fakultas) VALUES
('D1A01', 'Dr. Asep Anwar', 'Jl. Teknik No. 15', '081234567890',
'asep@universitas.com', 'Dosen Tetap', 'Teknik Informatika', 1),
('D1B01', 'Dr. Rina Sari', 'Jl. Komputer No. 10', '081234567891',
'rina@universitas.com', 'Dosen Tetap', 'Jaringan Komputer', 1);
-- Dosen Fakultas Hukum
INSERT INTO dosen (nip, nama, alamat, no_telepon, email, jabatan,
bidang_keahlian, id_fakultas) VALUES
('D2A01', 'Prof. Budi Pratama', 'Jl. Hukum No. 3', '081234567892',
'budi@universitas.com', 'Dosen Tetap', 'Hukum Pidana', 2),
('D2B01', 'Dr. Farah Diana', 'Jl. Hukum No. 7', '081234567893',
'farah@universitas.com', 'Dosen Tetap', 'Hukum Perdata', 2);
-- Dosen Fakultas Ekonomi dan Bisnis
INSERT INTO dosen (nip, nama, alamat, no_telepon, email, jabatan,
 bidang_keahlian, id_fakultas) VALUES
('D3A01', 'Dr. Joko Santoso', 'Jl. Ekonomi No. 5', '081234567894',
 'joko@universitas.com', 'Dosen Tetap', 'Manajemen Pemasaran', 3),
('D3B01', 'Dr. Siti Rahayu', 'Jl. Akuntansi No. 2', '081234567895',
 'siti@universitas.com', 'Dosen Tetap', 'Akuntansi Keuangan', 3);
-- Dosen Fakultas Ilmu Sosial dan Politik
INSERT INTO dosen (nip, nama, alamat, no_telepon, email, jabatan,
bidang_keahlian, id_fakultas) VALUES
('D4A01', 'Dr. Agus Wijaya', 'Jl. Sosial No. 8', '081234567896',
 'agus@universitas.com', 'Dosen Tetap', 'Ilmu Politik Dasar', 4),
('D4B01', 'Dr. Indah Puspita', 'Jl. Sosiologi No. 4',
'081234567897', 'indah@universitas.com', 'Dosen Tetap', 'Sosiologi Umum', 4);
```

Gambar 17 Source Code Insert Into Tabel Dosen

```sql
-- Dosen Fakultas Pertanian
INSERT INTO dosen (nip, nama, alamat, no_telepon, email, jabatan,
bidang_keahlian, id_fakultas) VALUES
('D5A01', 'Prof. Rudi Suryanto', 'Jl. Pertanian No. 1', '081234567898',
'rudi@universitas.com', 'Dosen Tetap', 'Agroekonomi', 5),
('D5B01', 'Dr. Amira Nasution', 'Jl. Tanah No. 6', '081234567899',
'amira@universitas.com', 'Dosen Tetap', 'Ilmu Tanah', 5);
-- Dosen Fakultas Matematika dan Ilmu Pengetahuan Alam
INSERT INTO dosen (nip, nama, alamat, no_telepon, email, jabatan,
bidang_keahlian, id_fakultas) VALUES
('D6A01', 'Dr. Alfi Fauzi', 'Jl. Matematika No. 9', '081234567900',
 'alfi@universitas.com', 'Dosen Tetap', 'Matematika Dasar', 6),
('D6B01', 'Dr. Rahmawati', 'Jl. Fisika No. 11', '081234567901',
'rahmawati@universitas.com', 'Dosen Tetap', 'Fisika Dasar', 6);
```

Gambar 18 Source Code Insert Into Tabel Dosen

```sql
-- Fakultas Teknik
INSERT INTO mata_kuliah (kode_mk, nama_mk, sks,
semester_mf, jenis, id_program_studi, nip_dosen) VALUES
('G1A001', 'Algoritma dan Pemrograman', 3, 1, 'Wajib', 'G1A', 'D1A01'),
('G1A002', 'Jaringan Komputer', 3, 2, 'Wajib', 'G1A', 'D1B01'),
('G1A003', 'Matematika Diskrit', 3, 1, 'Wajib', 'G1A', 'D1A01');
-- Fakultas Hukum
INSERT INTO mata_kuliah (kode_mk, nama_mk, sks,
semester_mf, jenis, id_program_studi, nip_dosen) VALUES
('B1A001', 'Hukum Pidana', 3, 1, 'Wajib', 'B1A', 'D1A01'),
('B1A002', 'Hukum Perdata', 3, 2, 'Wajib', 'B1A', 'D1B01'),
('B1A003', 'Hukum Administrasi Negara', 3, 3, 'Wajib', 'B1A', 'D1A01');
-- Fakultas Ekonomi dan Bisnis
INSERT INTO mata_kuliah (kode_mk, nama_mk, sks,
semester_mf, jenis, id_program_studi, nip_dosen) VALUES
('C1A001', 'Manajemen Pemasaran', 3, 1, 'Wajib', 'C1A', 'D1B01'),
('C1A002', 'Akuntansi Keuangan', 3, 2, 'Wajib', 'C1A', 'D1A01'),
('C1A003', 'Ekonomi Makro', 3, 1, 'Wajib', 'C1A', 'D1A01');
-- Fakultas Ilmu Sosial dan Politik
INSERT INTO mata_kuliah (kode_mk, nama_mk, sks,
semester_mf, jenis, id_program_studi, nip_dosen) VALUES
('D1A001', 'Ilmu Politik Dasar', 3, 1, 'Wajib', 'D1A', 'D1A01'),
('D1A002', 'Sosiologi Umum', 3, 2, 'Wajib', 'D1A', 'D1B01'),
('D1A003', 'Psikologi Sosial', 3, 3, 'Wajib', 'D1A', 'D1A01');
```

Gambar 19 Source Code Insert Into Tabel Mata Kuliah

```sql
-- Fakultas Pertanian
INSERT INTO mata_kuliah (kode_mk, nama_mk, sks,
semester_mf, jenis, id_program_studi, nip_dosen) VALUES
('E1A001', 'Agroekonomi', 3, 1, 'Wajib', 'E1A', 'D1B01'),
('E1A002', 'Ilmu Tanah', 3, 2, 'Wajib', 'E1A', 'D1A01'),
('E1A003', 'Pestisida dan Pengendalian Hama', 3, 3, 'Wajib', 'E1A', 'D1A01');
-- Fakultas Matematika dan Ilmu Pengetahuan Alam
INSERT INTO mata_kuliah (kode_mk, nama_mk, sks,
semester_mf, jenis, id_program_studi, nip_dosen) VALUES
('F1A001', 'Matematika Dasar', 3, 1, 'Wajib', 'F1A', 'D1A01'),
('F1A002', 'Fisika Dasar', 3, 2, 'Wajib', 'F1A', 'D1B01'),
('F1A003', 'Kimia Dasar', 3, 1, 'Wajib', 'F1A', 'D1A01');
```

Gambar 20 Source Code Insert Into Tabel Mata Kuliah

```sql
-- Semester Ganjil dan Genap 2020-2023
INSERT INTO semester (id_semester, tahun_ajaran, semester,
tanggal_mulai, tanggal_selesai, batas_pembayaran_ukt) VALUES
('20201', '2020/2021', 'Ganjil', '2020-08-01', '2021-01-31', '2020-09-15'),
('20202', '2020/2021', 'Genap', '2021-02-01', '2021-07-31', '2021-03-15'),
('20211', '2021/2022', 'Ganjil', '2021-08-01', '2022-01-31', '2021-09-15'),
('20212', '2021/2022', 'Genap', '2022-02-01', '2022-07-31', '2022-03-15'),
('20221', '2022/2023', 'Ganjil', '2022-08-15', '2023-01-20', '2022-09-10'),
('20222', '2022/2023', 'Genap',  '2023-02-06', '2023-07-24', '2023-03-06');
-- Tabel Nilai
INSERT INTO nilai (id_nilai, npm, kode_mk, id_semester,
nilai_tugas, nilai_uts, nilai_uas, nilai_akhir, huruf_mutu) VALUES
(1, 'G1A023049', 'G1A001', '20201', 80, 85, 90, 88, 'A'),
(2, 'G1A023049', 'G1A002', '20201', 75, 80, 85, 81, 'B+'),
(3, 'B1A023054', 'B1A001', '20201', 70, 75, 80, 78, 'B'),
(4, 'B1A023054', 'B1A002', '20201', 80, 85, 90, 88, 'A'),
(5, 'C1A023059', 'C1A001', '20201', 65, 70, 75, 74, 'B-'),
(6, 'G1A023049', 'G1A001', '20201', 80, 85, 90, 88, 'A'),
(7, 'G1A023049', 'G1A002', '20201', 75, 80, 85, 81, 'B+'),
(8, 'B1A023054', 'B1A001', '20201', 70, 75, 80, 78, 'B'),
(9, 'B1A023054', 'B1A002', '20201', 80, 85, 90, 88, 'A'),
(10, 'C1A023059', 'C1A001', '20201', 65, 70, 75, 74, 'B-'),
(11, 'D1A023064', 'D1A001', '20201', 85, 90, 88, 89, 'A'),
(12, 'D1A023064', 'D1A002', '20201', 78, 82, 85, 83, 'B+'),
(13, 'E1A023069', 'E1A001', '20201', 70, 75, 80, 78, 'B'),
(14, 'E1A023069', 'E1A002', '20201', 80, 85, 90, 88, 'A'),
(15, 'F1A023074', 'F1A001', '20201', 88, 90, 95, 92, 'A'),
(16, 'F1A023074', 'F1A002', '20201', 85, 88, 92, 89, 'A');
```

Gambar 21 Source Code Insert Into Tabel Pembayaran UKT

```sql
-- Data pembayaran UKT
INSERT INTO pembayaran_ukt (id_pembayaran, npm, id_semester, tanggal_bayar, jumlah_bayar,
status_pembayaran, metode_pembayaran, bukti_pembayaran) VALUES
(1, 'G1A023049', '20201', '2020-08-20', 1500000, 'Lunas', 'Transfer', 'bukti1.jpg'),
(2, 'B1A023054', '20201', '2020-08-21', 2000000, 'Lunas', 'Transfer', 'bukti2.jpg'),
(3, 'C1A023059', '20201', '2020-08-22', 1800000, 'Lunas', 'Transfer', 'bukti3.jpg'),
(4, 'D1A023064', '20201', '2020-08-23', 1600000, 'Lunas', 'Transfer', 'bukti4.jpg'),
(5, 'E1A023069', '20201', '2020-08-24', 1700000, 'Lunas', 'Transfer', 'bukti5.jpg'),
(6, 'G1A023049', '20201', '2020-08-20', 1500000, 'Lunas', 'Transfer', 'bukti1.jpg'),
(7, 'B1A023054', '20201', '2020-08-21', 2000000, 'Lunas', 'Transfer', 'bukti2.jpg'),
(8, 'C1A023059', '20201', '2020-08-22', 1800000, 'Lunas', 'Transfer', 'bukti3.jpg'),
(9, 'D1A023064', '20201', '2020-08-23', 1600000, 'Lunas', 'Transfer', 'bukti4.jpg'),
(10, 'E1A023069', '20201', '2020-08-24', 1700000, 'Lunas', 'Transfer', 'bukti5.jpg'),
(11, 'F1A023074', '20201', '2020-08-25', 1500000, 'Lunas', 'Transfer', 'bukti6.jpg');
```

Gambar 22 Source Code Insert Into Tabel Pembayaran UKT

**Source code :**

```sql
CREATE DATABASE sistem_akademik;
USE sistem_akademik;
CREATE TABLE fakultas (
   id_fakultas INT PRIMARY KEY,
   nama_fakultas VARCHAR(100) NOT NULL
```

```
);
CREATE TABLE program_studi (
    id_program_studi VARCHAR(10) PRIMARY KEY,
    nama_program_studi VARCHAR(100) NOT NULL,
    jenjang VARCHAR(20),
    id_fakultas INT,
    FOREIGN KEY (id_fakultas) REFERENCES fakultas(id_fakultas)
);
CREATE TABLE golongan_ukt (
    id_golongan_ukt INT PRIMARY KEY,
    nama_golongan VARCHAR(50),
    nominal_ukt DECIMAL(12,2),
    id_program_studi VARCHAR(10),
    FOREIGN KEY (id_program_studi) REFERENCES program_studi(id_program_studi)
);
CREATE TABLE mahasiswa (
    npm VARCHAR(15) PRIMARY KEY,
    nama VARCHAR(100),
    alamat TEXT,
    no_telepon VARCHAR(15),
    email VARCHAR(100),
    tanggal_lahir DATE,
    tanggal_masuk DATE,
    status VARCHAR(20),
    id_program_studi VARCHAR(10),
    id_golongan_ukt INT,
    FOREIGN KEY (id_program_studi) REFERENCES
program_studi(id_program_studi),
    FOREIGN KEY (id_golongan_ukt) REFERENCES golongan_ukt(id_golongan_ukt)
);
CREATE TABLE dosen (
    nip VARCHAR(20) PRIMARY KEY,
    nama VARCHAR(100),
    alamat TEXT,
    no_telepon VARCHAR(15),
```

```
    email VARCHAR(100),
    jabatan VARCHAR(50),
    bidang_keahlian VARCHAR(100),
    id_fakultas INT,
    FOREIGN KEY (id_fakultas) REFERENCES fakultas(id_fakultas)
);
CREATE TABLE mata_kuliah (
    kode_mk VARCHAR(10) PRIMARY KEY,
    nama_mk VARCHAR(100),
    sks INT,
    semester_mf INT,
    jenis VARCHAR(50),
    id_program_studi VARCHAR(10),
    nip_dosen VARCHAR(20),
    FOREIGN KEY (id_program_studi) REFERENCES
program_studi(id_program_studi),
    FOREIGN KEY (nip_dosen) REFERENCES dosen(nip)
);
CREATE TABLE semester (
    id_semester VARCHAR(10) PRIMARY KEY,
    tahun_ajaran VARCHAR(20),
    semester VARCHAR(10),
    tanggal_mulai DATE,
    tanggal_selesai DATE,
    batas_pembayaran_ukt DATE
);
CREATE TABLE nilai (
    id_nilai INT PRIMARY KEY,
    npm VARCHAR(15),
    kode_mk VARCHAR(10),
    id_semester VARCHAR(10),
    nilai_tugas FLOAT,
    nilai_uts FLOAT,
    nilai_uas FLOAT,
    nilai_akhir FLOAT,
```

```
    huruf_mutu VARCHAR(2),
    FOREIGN KEY (npm) REFERENCES mahasiswa(npm),
    FOREIGN KEY (kode_mk) REFERENCES mata_kuliah(kode_mk),
    FOREIGN KEY (id_semester) REFERENCES semester(id_semester)
);
CREATE TABLE pembayaran_ukt (
    id_pembayaran INT PRIMARY KEY,
    npm VARCHAR(15),
    id_semester VARCHAR(10),
    tanggal_bayar DATE,
    jumlah_bayar DECIMAL(12,2),
    status_pembayaran VARCHAR(20),
    metode_pembayaran VARCHAR(50),
    bukti_pembayaran TEXT,
    FOREIGN KEY (npm) REFERENCES mahasiswa(npm),
    FOREIGN KEY (id_semester) REFERENCES semester(id_semester)
);
INSERT INTO fakultas (id_fakultas, nama_fakultas) VALUES
(1, 'Fakultas Teknik'),
(2, 'Fakultas Hukum'),
(3, 'Fakultas Ekonomi dan Bisnis'),
(4, 'Fakultas Ilmu Sosial dan Politik'),
(5, 'Fakultas Pertanian'),
(6, 'Fakultas Matematika dan Ilmu Pengetahuan Alam'),
(7, 'Fakultas Kedokteran'),
(8, 'Fakultas Keguruan dan Ilmu Pendidikan');
INSERT INTO program_studi (id_program_studi, nama_program_studi, jenjang,
id_fakultas) VALUES
('G1A', 'Informatika', 'S1', 1),
('G1B', 'Teknik Elektro', 'S1', 1),
('G1C', 'Teknik Sipil', 'S1', 1),
('G1D', 'Teknik Mesin', 'S1', 1),
('G1E', 'Arsitektur', 'S1', 1);
INSERT INTO program_studi (id_program_studi, nama_program_studi, jenjang,
id_fakultas) VALUES
```

```
('B1A', 'Hukum Perdata', 'S1', 2),
('B1B', 'Hukum Pidana', 'S1', 2),
('B1C', 'Hukum Tata Negara', 'S1', 2),
('B1D', 'Hukum Internasional', 'S1', 2),
('B1E', 'Hukum Ekonomi', 'S1', 2);
INSERT INTO program_studi (id_program_studi, nama_program_studi, jenjang, id_fakultas) VALUES
('C1A', 'Manajemen', 'S1', 3),
('C1B', 'Akuntansi', 'S1', 3),
('C1C', 'Ekonomi Pembangunan', 'S1', 3),
('C1D', 'Ilmu Ekonomi', 'S1', 3),
('C1E', 'Bisnis Internasional', 'S1', 3);
INSERT INTO program_studi (id_program_studi, nama_program_studi, jenjang, id_fakultas) VALUES
('D1A', 'Ilmu Politik', 'S1', 4),
('D1B', 'Sosiologi', 'S1', 4),
('D1C', 'Antropologi', 'S1', 4),
('D1D', 'Ilmu Komunikasi', 'S1', 4),
('D1E', 'Psikologi', 'S1', 4);
INSERT INTO program_studi (id_program_studi, nama_program_studi, jenjang, id_fakultas) VALUES
('E1A', 'Agribisnis', 'S1', 5),
('E1B', 'Teknologi Pertanian', 'S1', 5),
('E1C', 'Ilmu Tanah', 'S1', 5),
('E1D', 'Pendidikan Pertanian', 'S1', 5),
('E1E', 'Penyuluhan Pertanian', 'S1', 5);
INSERT INTO program_studi (id_program_studi, nama_program_studi, jenjang, id_fakultas) VALUES
('F1A', 'Matematika', 'S1', 6),
('F1B', 'Fisika', 'S1', 6),
('F1C', 'Kimia', 'S1', 6),
('F1D', 'Biologi', 'S1', 6),
('F1E', 'Statistika', 'S1', 6);
INSERT INTO program_studi (id_program_studi, nama_program_studi, jenjang, id_fakultas) VALUES
```

```
('H1A', 'Kedokteran Umum', 'S1', 7),
('H1B', 'Kedokteran Gigi', 'S1', 7),
('H1C', 'Kedokteran Hewan', 'S1', 7),
('H1D', 'Farmasi', 'S1', 7),
('H1E', 'Keperawatan', 'S1', 7);
INSERT INTO program_studi (id_program_studi, nama_program_studi, jenjang, id_fakultas) VALUES
('A1A', 'Pendidikan Matematika', 'S1', 8),
('A1B', 'Pendidikan Bahasa Indonesia', 'S1', 8),
('A1C', 'Pendidikan Bahasa Inggris', 'S1', 8),
('A1D', 'Pendidikan Fisika', 'S1', 8),
('A1E', 'Pendidikan Kimia', 'S1', 8);
INSERT INTO golongan_ukt (id_golongan_ukt, nama_golongan, nominal_ukt, id_program_studi) VALUES
(1, 'Golongan A', 2000000.00, 'G1A'),
(2, 'Golongan B', 2500000.00, 'G1B'),
(3, 'Golongan C', 3000000.00, 'G1C'),
(4, 'Golongan D', 3500000.00, 'G1D'),
(5, 'Golongan E', 4000000.00, 'G1E');
INSERT INTO golongan_ukt (id_golongan_ukt, nama_golongan, nominal_ukt, id_program_studi) VALUES
(6, 'Golongan A', 2000000.00, 'B1A'),
(7, 'Golongan B', 2500000.00, 'B1B'),
(8, 'Golongan C', 3000000.00, 'B1C'),
(9, 'Golongan D', 3500000.00, 'B1D'),
(10, 'Golongan E', 4000000.00, 'B1E');
INSERT INTO golongan_ukt (id_golongan_ukt, nama_golongan, nominal_ukt, id_program_studi) VALUES
(11, 'Golongan A', 2000000.00, 'C1A'),
(12, 'Golongan B', 2500000.00, 'C1B'),
(13, 'Golongan C', 3000000.00, 'C1C'),
(14, 'Golongan D', 3500000.00, 'C1D'),
(15, 'Golongan E', 4000000.00, 'C1E');
INSERT INTO golongan_ukt (id_golongan_ukt, nama_golongan, nominal_ukt, id_program_studi) VALUES
```

```
(16, 'Golongan A', 2000000.00, 'D1A'),
(17, 'Golongan B', 2500000.00, 'D1B'),
(18, 'Golongan C', 3000000.00, 'D1C'),
(19, 'Golongan D', 3500000.00, 'D1D'),
(20, 'Golongan E', 4000000.00, 'D1E');
INSERT INTO golongan_ukt (id_golongan_ukt, nama_golongan, nominal_ukt, id_program_studi) VALUES
(21, 'Golongan A', 2000000.00, 'E1A'),
(22, 'Golongan B', 2500000.00, 'E1B'),
(23, 'Golongan C', 3000000.00, 'E1C'),
(24, 'Golongan D', 3500000.00, 'E1D'),
(25, 'Golongan E', 4000000.00, 'E1E');
INSERT INTO golongan_ukt (id_golongan_ukt, nama_golongan, nominal_ukt, id_program_studi) VALUES
(26, 'Golongan A', 2000000.00, 'F1A'),
(27, 'Golongan B', 2500000.00, 'F1B'),
(28, 'Golongan C', 3000000.00, 'F1C'),
(29, 'Golongan D', 3500000.00, 'F1D'),
(30, 'Golongan E', 4000000.00, 'F1E');
INSERT INTO golongan_ukt (id_golongan_ukt, nama_golongan, nominal_ukt, id_program_studi) VALUES
(31, 'Golongan A', 2000000.00, 'H1A'),
(32, 'Golongan B', 2500000.00, 'H1B'),
(33, 'Golongan C', 3000000.00, 'H1C'),
(34, 'Golongan D', 3500000.00, 'H1D'),
(35, 'Golongan E', 4000000.00, 'H1E');
INSERT INTO golongan_ukt (id_golongan_ukt, nama_golongan, nominal_ukt, id_program_studi) VALUES
(36, 'Golongan A', 2000000.00, 'A1A'),
(37, 'Golongan B', 2500000.00, 'A1B'),
(38, 'Golongan C', 3000000.00, 'A1C'),
(39, 'Golongan D', 3500000.00, 'A1D'),
(40, 'Golongan E', 4000000.00, 'A1E');
INSERT INTO mahasiswa (npm, nama, alamat, no_telepon, email, tanggal_lahir, tanggal_masuk, status, id_program_studi, id_golongan_ukt) VALUES
```

('G1A023049', 'Andi Rahmat', 'Jl. Raya No.1', '08123456789', 'andi@gmail.com', '2000-01-15', '2020-09-01', 'Aktif', 'G1A', 1),
('G1B023050', 'Budi Setiawan', 'Jl. Teknik No.3', '08234567890', 'budi@gmail.com', '1999-02-20', '2020-09-01', 'Aktif', 'G1B', 2),
('G1C023051', 'Cindy Yuliana', 'Jl. Sipil No.4', '08345678901', 'cindy@gmail.com', '1998-03-25', '2020-09-01', 'Aktif', 'G1C', 3),
('G1D023052', 'Diana Utami', 'Jl. Mesin No.5', '08456789012', 'diana@gmail.com', '2001-04-30', '2021-09-01', 'Aktif', 'G1D', 4),
('G1E023053', 'Erwin Pratama', 'Jl. Arsitektur No.6', '08567890123', 'erwin@gmail.com', '2000-05-10', '2021-09-01', 'Aktif', 'G1E', 5);
INSERT INTO mahasiswa (npm, nama, alamat, no_telepon, email, tanggal_lahir, tanggal_masuk, status, id_program_studi, id_golongan_ukt) VALUES
('B1A023054', 'Farah Dewi', 'Jl. Hukum No.7', '08678901234', 'farah@gmail.com', '2000-06-12', '2020-09-01', 'Aktif', 'B1A', 6),
('B1B023055', 'Gilang Sutrisno', 'Jl. Pidana No.8', '08789012345', 'gilang@gmail.com', '1999-07-14', '2020-09-01', 'Aktif', 'B1B', 7),
('B1C023056', 'Hannah Nur', 'Jl. Tata Negara No.9', '08890123456', 'hannah@gmail.com', '1998-08-17', '2020-09-01', 'Aktif', 'B1C', 8),
('B1D023057', 'Ikhsan Mulyadi', 'Jl. Internasional No.10', '08901234567', 'ikhsan@gmail.com', '2001-09-21', '2021-09-01', 'Aktif', 'B1D', 9),
('B1E023058', 'Jasmine Amalia', 'Jl. Ekonomi No.11', '09012345678', 'jasmine@gmail.com', '2000-10-25', '2021-09-01', 'Aktif', 'B1E', 10);

INSERT INTO mahasiswa (npm, nama, alamat, no_telepon, email, tanggal_lahir, tanggal_masuk, status, id_program_studi, id_golongan_ukt) VALUES
('C1A023059', 'Kevin Cahya', 'Jl. Manajemen No.12', '09123456789', 'kevin@gmail.com', '1999-11-30', '2020-09-01', 'Aktif', 'C1A', 11),
('C1B023060', 'Lina Mardiana', 'Jl. Akuntansi No.13', '09234567890', 'lina@gmail.com', '2000-12-14', '2020-09-01', 'Aktif', 'C1B', 12),
('C1C023061', 'Martha Desy', 'Jl. Ekonomi Pembangunan No.14', '09345678901', 'martha@gmail.com', '2001-01-17', '2021-09-01', 'Aktif', 'C1C', 13),
('C1D023062', 'Nadia Ananda', 'Jl. Ilmu Ekonomi No.15', '09456789012', 'nadia@gmail.com', '2000-02-22', '2021-09-01', 'Aktif', 'C1D', 14),
('C1E023063', 'Oscar Gunawan', 'Jl. Bisnis Internasional No.16', '09567890123', 'oscar@gmail.com', '1998-03-30', '2020-09-01', 'Aktif', 'C1E', 15);

```
INSERT INTO mahasiswa (npm, nama, alamat, no_telepon, email, tanggal_lahir, tanggal_masuk, status, id_program_studi, id_golongan_ukt) VALUES
('D1A023064', 'Putu Satria', 'Jl. Ilmu Politik No.17', '09678901234', 'putu@gmail.com', '2000-04-02', '2020-09-01', 'Aktif', 'D1A', 16),
('D1B023065', 'Rina Ayu', 'Jl. Sosiologi No.18', '09789012345', 'rina@gmail.com', '1999-05-10', '2020-09-01', 'Aktif', 'D1B', 17),
('D1C023066', 'Siti Wulandari', 'Jl. Antropologi No.19', '09890123456', 'siti@gmail.com', '1998-06-18', '2020-09-01', 'Aktif', 'D1C', 18),
('D1D023067', 'Tina Ambarwati', 'Jl. Komunikasi No.20', '09901234567', 'tina@gmail.com', '2001-07-12', '2021-09-01', 'Aktif', 'D1D', 19),
('D1E023068', 'Umi Nabila', 'Jl. Psikologi No.21', '10012345678', 'umi@gmail.com', '2000-08-25', '2021-09-01', 'Aktif', 'D1E', 20);
INSERT INTO mahasiswa (npm, nama, alamat, no_telepon, email, tanggal_lahir, tanggal_masuk, status, id_program_studi, id_golongan_ukt) VALUES
('E1A023069', 'Vina Anjani', 'Jl. Agribisnis No.22', '10123456789', 'vina@gmail.com', '2000-09-10', '2020-09-01', 'Aktif', 'E1A', 21),
('E1B023070', 'Wawan Subagyo', 'Jl. Teknologi Pertanian No.23', '10234567890', 'wawan@gmail.com', '1999-10-15', '2020-09-01', 'Aktif', 'E1B', 22),
('E1C023071', 'Xenia Putri', 'Jl. Ilmu Tanah No.24', '10345678901', 'xenia@gmail.com', '2001-11-20', '2021-09-01', 'Aktif', 'E1C', 23),
('E1D023072', 'Yanti Amelia', 'Jl. Pendidikan Pertanian No.25', '10456789012', 'yanti@gmail.com', '2000-12-05', '2021-09-01', 'Aktif', 'E1D', 24),
('E1E023073', 'Zaki Akbar', 'Jl. Penyuluhan Pertanian No.26', '10567890123', 'zaki@gmail.com', '1998-01-03', '2020-09-01', 'Aktif', 'E1E', 25);

INSERT INTO mahasiswa (npm, nama, alamat, no_telepon, email, tanggal_lahir, tanggal_masuk, status, id_program_studi, id_golongan_ukt) VALUES
('F1A023074', 'Alfiansyah Putra', 'Jl. Matematika No.27', '10678901234', 'alfiansyah@gmail.com', '1999-02-14', '2020-09-01', 'Aktif', 'F1A', 26),
('F1B023075', 'Budi Santosa', 'Jl. Fisika No.28', '10789012345', 'budi@gmail.com', '2000-03-25', '2020-09-01', 'Aktif', 'F1B', 27),
('F1C023076', 'Cahaya Indah', 'Jl. Kimia No.29', '10890123456', 'cahaya@gmail.com', '2001-04-18', '2021-09-01', 'Aktif', 'F1C', 28),
('F1D023077', 'Dani Rafi', 'Jl. Biologi No.30', '10901234567', 'dani@gmail.com', '2000-05-10', '2021-09-01', 'Aktif', 'F1D', 29),
```

('F1E023078', 'Elisa Dwi', 'Jl. Statistika No.31', '11012345678', 'elisa@gmail.com', '1998-06-22', '2020-09-01', 'Aktif', 'F1E', 30);

INSERT INTO dosen (nip, nama, alamat, no_telepon, email, jabatan, bidang_keahlian, id_fakultas) VALUES

('D1A01', 'Dr. Asep Anwar', 'Jl. Teknik No. 15', '081234567890', 'asep@universitas.com', 'Dosen Tetap', 'Teknik Informatika', 1),

('D1B01', 'Dr. Rina Sari', 'Jl. Komputer No. 10', '081234567891', 'rina@universitas.com', 'Dosen Tetap', 'Jaringan Komputer', 1);

INSERT INTO dosen (nip, nama, alamat, no_telepon, email, jabatan, bidang_keahlian, id_fakultas) VALUES

('D2A01', 'Prof. Budi Pratama', 'Jl. Hukum No. 3', '081234567892', 'budi@universitas.com', 'Dosen Tetap', 'Hukum Pidana', 2),

('D2B01', 'Dr. Farah Diana', 'Jl. Hukum No. 7', '081234567893', 'farah@universitas.com', 'Dosen Tetap', 'Hukum Perdata', 2);

INSERT INTO dosen (nip, nama, alamat, no_telepon, email, jabatan, bidang_keahlian, id_fakultas) VALUES

('D3A01', 'Dr. Joko Santoso', 'Jl. Ekonomi No. 5', '081234567894', 'joko@universitas.com', 'Dosen Tetap', 'Manajemen Pemasaran', 3),

('D3B01', 'Dr. Siti Rahayu', 'Jl. Akuntansi No. 2', '081234567895', 'siti@universitas.com', 'Dosen Tetap', 'Akuntansi Keuangan', 3);

INSERT INTO dosen (nip, nama, alamat, no_telepon, email, jabatan, bidang_keahlian, id_fakultas) VALUES

('D4A01', 'Dr. Agus Wijaya', 'Jl. Sosial No. 8', '081234567896', 'agus@universitas.com', 'Dosen Tetap', 'Ilmu Politik Dasar', 4),

('D4B01', 'Dr. Indah Puspita', 'Jl. Sosiologi No. 4', '081234567897', 'indah@universitas.com', 'Dosen Tetap', 'Sosiologi Umum', 4);

INSERT INTO dosen (nip, nama, alamat, no_telepon, email, jabatan, bidang_keahlian, id_fakultas) VALUES

('D5A01', 'Prof. Rudi Suryanto', 'Jl. Pertanian No. 1', '081234567898', 'rudi@universitas.com', 'Dosen Tetap', 'Agroekonomi', 5),

('D5B01', 'Dr. Amira Nasution', 'Jl. Tanah No. 6', '081234567899', 'amira@universitas.com', 'Dosen Tetap', 'Ilmu Tanah', 5);

INSERT INTO dosen (nip, nama, alamat, no_telepon, email, jabatan, bidang_keahlian, id_fakultas) VALUES

```
('D6A01', 'Dr. Alfi Fauzi', 'Jl. Matematika No. 9', '081234567900', 'alfi@universitas.com',
'Dosen Tetap', 'Matematika Dasar', 6),
('D6B01', 'Dr. Rahmawati', 'Jl. Fisika No. 11', '081234567901',
'rahmawati@universitas.com', 'Dosen Tetap', 'Fisika Dasar', 6);
INSERT INTO mata_kuliah (kode_mk, nama_mk, sks, semester_mf, jenis,
id_program_studi, nip_dosen) VALUES
('G1A001', 'Algoritma dan Pemrograman', 3, 1, 'Wajib', 'G1A', 'D1A01'),
('G1A002', 'Jaringan Komputer', 3, 2, 'Wajib', 'G1A', 'D1B01'),
('G1A003', 'Matematika Diskrit', 3, 1, 'Wajib', 'G1A', 'D1A01');
INSERT INTO mata_kuliah (kode_mk, nama_mk, sks, semester_mf, jenis,
id_program_studi, nip_dosen) VALUES
('B1A001', 'Hukum Pidana', 3, 1, 'Wajib', 'B1A', 'D1A01'),
('B1A002', 'Hukum Perdata', 3, 2, 'Wajib', 'B1A', 'D1B01'),
('B1A003', 'Hukum Administrasi Negara', 3, 3, 'Wajib', 'B1A', 'D1A01');
INSERT INTO mata_kuliah (kode_mk, nama_mk, sks, semester_mf, jenis,
id_program_studi, nip_dosen) VALUES
('C1A001', 'Manajemen Pemasaran', 3, 1, 'Wajib', 'C1A', 'D1B01'),
('C1A002', 'Akuntansi Keuangan', 3, 2, 'Wajib', 'C1A', 'D1A01'),
('C1A003', 'Ekonomi Makro', 3, 1, 'Wajib', 'C1A', 'D1A01');
INSERT INTO mata_kuliah (kode_mk, nama_mk, sks, semester_mf, jenis,
id_program_studi, nip_dosen) VALUES
('D1A001', 'Ilmu Politik Dasar', 3, 1, 'Wajib', 'D1A', 'D1A01'),
('D1A002', 'Sosiologi Umum', 3, 2, 'Wajib', 'D1A', 'D1B01'),
('D1A003', 'Psikologi Sosial', 3, 3, 'Wajib', 'D1A', 'D1A01');
INSERT INTO mata_kuliah (kode_mk, nama_mk, sks, semester_mf, jenis,
id_program_studi, nip_dosen) VALUES
('E1A001', 'Agroekonomi', 3, 1, 'Wajib', 'E1A', 'D1B01'),
('E1A002', 'Ilmu Tanah', 3, 2, 'Wajib', 'E1A', 'D1A01'),
('E1A003', 'Pestisida dan Pengendalian Hama', 3, 3, 'Wajib', 'E1A', 'D1A01');

INSERT INTO mata_kuliah (kode_mk, nama_mk, sks, semester_mf, jenis,
id_program_studi, nip_dosen) VALUES
('F1A001', 'Matematika Dasar', 3, 1, 'Wajib', 'F1A', 'D1A01'),
('F1A002', 'Fisika Dasar', 3, 2, 'Wajib', 'F1A', 'D1B01'),
('F1A003', 'Kimia Dasar', 3, 1, 'Wajib', 'F1A', 'D1A01');
```

```
INSERT INTO semester (id_semester, tahun_ajaran, semester, tanggal_mulai, tanggal_selesai, batas_pembayaran_ukt) VALUES
('20201', '2020/2021', 'Ganjil', '2020-08-01', '2021-01-31', '2020-09-15'),
('20202', '2020/2021', 'Genap', '2021-02-01', '2021-07-31', '2021-03-15'),
-- Semester Ganjil dan Genap 2021-2022
('20211', '2021/2022', 'Ganjil', '2021-08-01', '2022-01-31', '2021-09-15'),
('20212', '2021/2022', 'Genap', '2022-02-01', '2022-07-31', '2022-03-15'),
-- Semester Ganjil dan Genap 2022-2023
('20221', '2022/2023', 'Ganjil', '2022-08-15', '2023-01-20', '2022-09-10'),
('20222', '2022/2023', 'Genap', '2023-02-06', '2023-07-24', '2023-03-06');
INSERT INTO nilai (id_nilai, npm, kode_mk, id_semester, nilai_tugas, nilai_uts, nilai_uas, nilai_akhir, huruf_mutu) VALUES
(1, 'G1A023049', 'G1A001', '20201', 80, 85, 90, 88, 'A'),
(2, 'G1A023049', 'G1A002', '20201', 75, 80, 85, 81, 'B+'),
(3, 'B1A023054', 'B1A001', '20201', 70, 75, 80, 78, 'B'),
(4, 'B1A023054', 'B1A002', '20201', 80, 85, 90, 88, 'A'),
(5, 'C1A023059', 'C1A001', '20201', 65, 70, 75, 74, 'B-'),
(6, 'G1A023049', 'G1A001', '20201', 80, 85, 90, 88, 'A'),
(7, 'G1A023049', 'G1A002', '20201', 75, 80, 85, 81, 'B+'),
(8, 'B1A023054', 'B1A001', '20201', 70, 75, 80, 78, 'B'),
(9, 'B1A023054', 'B1A002', '20201', 80, 85, 90, 88, 'A'),
(10, 'C1A023059', 'C1A001', '20201', 65, 70, 75, 74, 'B-'),
(11, 'D1A023064', 'D1A001', '20201', 85, 90, 88, 89, 'A'),
(12, 'D1A023064', 'D1A002', '20201', 78, 82, 85, 83, 'B+'),
(13, 'E1A023069', 'E1A001', '20201', 70, 75, 80, 78, 'B'),
(14, 'E1A023069', 'E1A002', '20201', 80, 85, 90, 88, 'A'),
(15, 'F1A023074', 'F1A001', '20201', 88, 90, 95, 92, 'A'),
(16, 'F1A023074', 'F1A002', '20201', 85, 88, 92, 89, 'A');
INSERT INTO pembayaran_ukt (id_pembayaran, npm, id_semester, tanggal_bayar, jumlah_bayar, status_pembayaran, metode_pembayaran, bukti_pembayaran) VALUES
(1, 'G1A023049', '20201', '2020-08-20', 1500000, 'Lunas', 'Transfer', 'bukti1.jpg'),
(2, 'B1A023054', '20201', '2020-08-21', 2000000, 'Lunas', 'Transfer', 'bukti2.jpg'),
(3, 'C1A023059', '20201', '2020-08-22', 1800000, 'Lunas', 'Transfer', 'bukti3.jpg'),
(4, 'D1A023064', '20201', '2020-08-23', 1600000, 'Lunas', 'Transfer', 'bukti4.jpg'),
(5, 'E1A023069', '20201', '2020-08-24', 1700000, 'Lunas', 'Transfer', 'bukti5.jpg'),
```

(6, 'G1A023049', '20201', '2020-08-20', 1500000, 'Lunas', 'Transfer', 'bukti1.jpg'),
(7, 'B1A023054', '20201', '2020-08-21', 2000000, 'Lunas', 'Transfer', 'bukti2.jpg'),
(8, 'C1A023059', '20201', '2020-08-22', 1800000, 'Lunas', 'Transfer', 'bukti3.jpg'),
(9, 'D1A023064', '20201', '2020-08-23', 1600000, 'Lunas', 'Transfer', 'bukti4.jpg'),
(10, 'E1A023069', '20201', '2020-08-24', 1700000, 'Lunas', 'Transfer', 'bukti5.jpg'),
(11, 'F1A023074', '20201', '2020-08-25', 1500000, 'Lunas', 'Transfer', 'bukti6.jpg');

**Pembahasan source code :**

Pada code di atas digunakan untuk membuat *database* sistem_akademik menggunakan CREATE DATABASE sistem_akademik; yang kemudian diaktifkan dengan USE sistem_akademik;. Terdapat beberapa tabel yakni fakultas, program_studi, golongan_ukt, mahasiswa, dosen, mata_kuliah, semester, nilai, dan pembayaran_ukt. Tabel pertama adalah fakultas, menggunakan CREATE TABLE untuk membuat tabel seperti yang lainnya, yang terdiri dari dua kolom yaitu id_fakultas yang bertipe INT dan *primary key* untuk kode unik nya, nama_fakultas VARCHAR(100), yang digunakan untuk menyimpan informasi setiap fakultas di perguruan tinggi. Tabel kedua adalah program_studi, yang mencakup kolom id_program_studi VARCHAR(10) PRIMARY KEY, nama_program_studi VARCHAR(100) NOT NULL, jenjang VARCHAR(20), id_fakultas INT, FOREIGN KEY (id_fakultas) REFERENCES fakultas(id_fakultas), agar bisa berelasi ke fakultas menggunakan *foreign key*.

Tabel ini menyimpan data mengenai setiap program studi, tingkatan pendidikan, serta fakultas pengampunya. Tabel golongan_ukt menyimpan informasi mengenai kelompok pembayaran UKT dengan kolom id_golongan_ukt (INT, PRIMARY KEY), nama_golongan VARCHAR(50), nominal_ukt DECIMAL(12,2), dan id_program_studi (VARCHAR(10), FOREIGN KEY (id_program_studi) REFERENCES program_studi(id_program_studi), untuk mencatat berbagai tingkatan UKT berdasarkan program studi. Selanjutnya, tabel mahasiswa memiliki kolom npm VARCHAR(15) PRIMARY KEY, nama VARCHAR(100), alamat TEXT, no_telepon VARCHAR(15), email VARCHAR(100), tanggal_lahir DATE, tanggal_masuk DATE, status VARCHAR(20), id_program_studi VARCHAR(10) FOREIGN KEY ke program_studi.id_program_studi, dan id_golongan_ukt INT FOREIGN KEY golongan_ukt.id_golongan_ukt.

Tabel ini mencakup seluruh data pribadi, akademik, dan keuangan mahasiswa. Tabel dosen menyimpan informasi mengenai dosen dan memiliki kolom nip VARCHAR(20) PRIMARY KEY, nama VARCHAR(100), alamat TEXT, no_telepon VARCHAR(15), email VARCHAR(100), jabatan VARCHAR(50), bidang_keahlian VARCHAR(100), dan

id_fakultas INT FOREIGN KEY fakultas.id_fakultas, Tabel mata_kuliah mencakup kolom kode_mk VARCHAR(10) PRIMARY KEY), nama_mk VARCHAR(100), sks (INT), semester_mf (INT), jenis VARCHAR(50), id_program_studi VARCHAR(10) FOREIGN KEY program_studi.id_program_studi, dan nip_dosen VARCHAR(20) FOREIGN KEY ke dosen.nip. Tabel ini menyimpan informasi lengkap tentang mata kuliah yang ditawarkan oleh program studi dan diajarkan oleh dosen tertentu. Tabel semester menyimpan periode akademik, dengan kolom id_semester VARCHAR(10) PRIMARY KEY untuk kode unik nya, tahun_ajaran VARCHAR(20), semester VARCHAR(10), tanggal_mulai DATE, tanggal_selesai DATE, dan batas_pembayaran_ukt DATE. Untuk memudahkan penjadwalan waktu akademik dan batas pembayaran secara terstruktur.

Tabel nilai berfungsi menyimpan hasil studi mahasiswa. Kolomnya adalah id_nilai INT PRIMARY KEY npm VARCHAR(15) FOREIGN KEY mahasiswa.npm, kode_mk VARCHAR(10) FOREIGN KEY mata_kuliah.kode_mk, id_semester VARCHAR(10) FOREIGN KEY semester.id_semester, nilai_tugas FLOAT, nilai_uts FLOAT tipe data ini digunakan untuk menyimpan bilangan decimal, nilai_uas FLOAT, nilai_akhir FLOAT, dan huruf_mutu VARCHAR(2). Data ini digunakan untuk melihat pencapaian akademik mahasiswa di setiap semester nya.

Terakhir, tabel pembayaran UKT yang mencatat transaksi pembayaran kuliah mahasiswa. Terdiri dari kolom id_pembayaran INT, PRIMARY KEY, npm VARCHAR(15) FOREIGN KEY mahasiswa.npm, id_semester VARCHAR(10) FOREIGN KEY semester.id_semester, tanggal_bayar DATE, jumlah_bayar DECIMAL(12,2), status_pembayaran VARCHAR(20), metode_pembayaran VARCHAR(50), dan bukti_pembayaran TEXT. Tabel ini digunakan untuk memudahkan pencatatan detail pembayaran UKT per semester untuk setiap mahasiswa.

Selanjutnya, data dimasukkan ke dalam tabel menggunakan INSERT INTO yang berfungsi menambahkan data baru ke dalam sebuah tabel di basis data yang berisi delapan fakultas, seperti Fakultas Teknik, Fakultas Hukum, Fakultas Ekonomi dan Bisnis, Fakultas Ilmu Sosial dan Politik, Fakultas Pertanian, Fakultas Matematika dan Ilmu Pengetahuan Alam, Fakultas Kedokteran dan Fakultas Keguruan dan Ilmu Pendidikan. Setelah itu, data program studi (program_studi) dimasukkan dan dikelompokkan berdasarkan fakultasnya. Masing-masing program studi memiliki ID unik menggunakan *primary key*, nama, jenjang pendidikan, dan dihubungkan ke fakultas melalui id_fakultas. Selanjutnya, data UKT (Uang Kuliah Tunggal) dimasukkan ke tabel golongan_ukt, yang menentukan besar nominal UKT berdasarkan golongan untuk setiap program studi. Setiap program studi memiliki lima golongan UKT, dari A hingga E, dengan nominal bertingkat dari 2 juta

hingga 4 juta rupiah. Terakhir, data mahasiswa dimasukkan ke dalam tabel mahasiswa, dengan masing-masing mahasiswa memiliki informasi lengkap seperti NPM, nama, alamat, kontak, email, tanggal lahir, tanggal masuk, status (semuanya "Aktif"), serta relasi ke program studi dan golongan UKT yang sesuai.

**Printscreen output :**

| id_fakultas | nama_fakultas |
|---|---|
| 1 | Fakultas Teknik |
| 2 | Fakultas Hukum |
| 3 | Fakultas Ekonomi dan Bisnis |
| 4 | Fakultas Ilmu Sosial dan Politik |
| 5 | Fakultas Pertanian |
| 6 | Fakultas Matematika dan Ilmu Pengetahuan Alam |
| 7 | Fakultas Kedokteran |
| 8 | Fakultas Keguruan dan Ilmu Pendidikan |
| NULL | NULL |

Gambar 23 Tabel Fakultas

| id_program_studi | nama_program_studi | jenjang | id_fakultas |
|---|---|---|---|
| A1A | Pendidikan Matematika | S1 | 8 |
| A1B | Pendidikan Bahasa Indonesia | S1 | 8 |
| A1C | Pendidikan Bahasa Inggris | S1 | 8 |
| A1D | Pendidikan Fisika | S1 | 8 |
| A1E | Pendidikan Kimia | S1 | 8 |
| B1A | Hukum Perdata | S1 | 2 |
| B1B | Hukum Pidana | S1 | 2 |
| B1C | Hukum Tata Negara | S1 | 2 |
| B1D | Hukum Internasional | S1 | 2 |
| B1E | Hukum Ekonomi | S1 | 2 |
| C1A | Manajemen | S1 | 3 |

Gambar 24 Tabel Program studi

| id_golongan_ukt | nama_golongan | nominal_ukt | id_program_studi |
|---|---|---|---|
| 1 | Golongan A | 2000000.00 | G1A |
| 2 | Golongan B | 2500000.00 | G1B |
| 3 | Golongan C | 3000000.00 | G1C |
| 4 | Golongan D | 3500000.00 | G1D |
| 5 | Golongan E | 4000000.00 | G1E |
| 6 | Golongan A | 2000000.00 | B1A |
| 7 | Golongan B | 2500000.00 | B1B |
| 8 | Golongan C | 3000000.00 | B1C |
| 9 | Golongan D | 3500000.00 | B1D |
| 10 | Golongan E | 4000000.00 | B1E |
| 11 | Golongan A | 2000000.00 | C1A |
| 12 | Golongan B | 2500000.00 | C1B |
| 13 | Golongan C | 3000000.00 | C1C |
| 14 | Golongan D | 3500000.00 | C1D |
| 15 | Golongan E | 4000000.00 | C1E |

Gambar 25 Tabel Golongan UKT

| npm | nama | alamat | no_telepon | email | tanggal_lahir | tanggal_masuk | status | id_program_studi | id_golongan_ukt |
|---|---|---|---|---|---|---|---|---|---|
| D1A023064 | Putu Satria | Jl. Ilmu Politik No.17 | 09678901234 | putu@example.com | 2000-04-02 | 2020-09-01 | Aktif | D1A | 16 |
| D1B023065 | Rina Ayu | Jl. Sosiologi No.18 | 09789012345 | rina@example.com | 1999-05-10 | 2020-09-01 | Aktif | D1B | 17 |
| D1C023066 | Siti Wulandari | Jl. Antropologi No.19 | 09890123456 | siti@example.com | 1998-06-18 | 2020-09-01 | Aktif | D1C | 18 |
| D1D023067 | Tina Ambarwati | Jl. Komunikasi No.20 | 09901234567 | tina@example.com | 2001-07-12 | 2021-09-01 | Aktif | D1D | 19 |
| D1E023068 | Umi Nabila | Jl. Psikologi No.21 | 10012345678 | umi@example.com | 2000-08-25 | 2021-09-01 | Aktif | D1E | 20 |
| E1A023069 | Vina Anjani | Jl. Agribisnis No.22 | 10123456789 | vina@example.com | 2000-09-10 | 2020-09-01 | Aktif | E1A | 21 |
| E1B023070 | Wawan Subagyo | Jl. Teknologi Pertanian No.23 | 10234567890 | wawan@example.com | 1999-10-15 | 2020-09-01 | Aktif | E1B | 22 |
| E1C023071 | Xenia Putri | Jl. Ilmu Tanah No.24 | 10345678901 | xenia@example.com | 2001-11-20 | 2021-09-01 | Aktif | E1C | 23 |
| E1D023072 | Yanti Amelia | Jl. Pendidikan Pertanian No.25 | 10456789012 | yanti@example.com | 2000-12-05 | 2021-09-01 | Aktif | E1D | 24 |
| E1E023073 | Zaki Akbar | Jl. Penyuluhan Pertanian No.26 | 10567890123 | zaki@example.com | 1998-01-03 | 2020-09-01 | Aktif | E1E | 25 |
| F1A023074 | Alfiansyah Putra | Jl. Matematika No.27 | 10678901234 | alfiansyah@example.... | 1999-02-14 | 2020-09-01 | Aktif | F1A | 26 |
| F1B023075 | Budi Santosa | Jl. Fisika No.28 | 10789012345 | budi@example.com | 2000-03-25 | 2020-09-01 | Aktif | F1B | 27 |
| F1C023076 | Cahaya Indah | Jl. Kimia No.29 | 10890123456 | cahaya@example.com | 2001-04-18 | 2021-09-01 | Aktif | F1C | 28 |
| F1D023077 | Dani Rafi | Jl. Biologi No.30 | 10901234567 | dani@example.com | 2000-05-10 | 2021-09-01 | Aktif | F1D | 29 |
| F1E023078 | Elisa Dwi | Jl. Statistika No.31 | 11012345678 | elisa@example.com | 1998-06-22 | 2020-09-01 | Aktif | F1E | 30 |
| G1A023049 | Andi Rahmat | Jl. Raya No.1 | 08123456789 | andi@example.com | 2000-01-15 | 2020-09-01 | Aktif | G1A | 1 |
| G1B023050 | Budi Setiawan | Jl. Teknik No.3 | 08234567890 | budi@example.com | 1999-02-20 | 2020-09-01 | Aktif | G1B | 2 |
| G1C023051 | Cindy Yuliana | Jl. Sipil No.4 | 08345678901 | cindy@example.com | 1998-03-25 | 2020-09-01 | Aktif | G1C | 3 |
| G1D023052 | Diana Utami | Jl. Mesin No.5 | 08456789012 | diana@example.com | 2001-04-30 | 2021-09-01 | Aktif | G1D | 4 |
| G1E023053 | Erwin Pratama | Jl. Arsitektur No.6 | 08567890123 | erwin@example.com | 2000-05-10 | 2021-09-01 | Aktif | G1E | 5 |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

Gambar 26 Tabel Mahasiswa

| nip | nama | alamat | no_telepon | email | jabatan | bidang_keahlian | id_fakultas |
|---|---|---|---|---|---|---|---|
| D1A01 | Dr. Asep Anwar | Jl. Teknik No. 15 | 081234567890 | asep@universitas.com | Dosen Tetap | Teknik Informatika | 1 |
| D1B01 | Dr. Rina Sari | Jl. Komputer No. 10 | 081234567891 | rina@universitas.com | Dosen Tetap | Jaringan Komputer | 1 |
| D2A01 | Prof. Budi Pratama | Jl. Hukum No. 3 | 081234567892 | budi@universitas.com | Dosen Tetap | Hukum Pidana | 2 |
| D2B01 | Dr. Farah Diana | Jl. Hukum No. 7 | 081234567893 | farah@universitas.com | Dosen Tetap | Hukum Perdata | 2 |
| D3A01 | Dr. Joko Santoso | Jl. Ekonomi No. 5 | 081234567894 | joko@universitas.com | Dosen Tetap | Manajemen Pemasaran | 3 |
| D3B01 | Dr. Siti Rahayu | Jl. Akuntansi No. 2 | 081234567895 | siti@universitas.com | Dosen Tetap | Akuntansi Keuangan | 3 |
| D4A01 | Dr. Agus Wijaya | Jl. Sosial No. 8 | 081234567896 | agus@universitas.com | Dosen Tetap | Ilmu Politik Dasar | 4 |

Gambar 27 Tabel Dosen

| kode_mk | nama_mk | sks | semester_mf | jenis | id_program_studi | nip_dosen |
|---|---|---|---|---|---|---|
| B1A001 | Hukum Pidana | 3 | 1 | Wajib | B1A | D1A01 |
| B1A002 | Hukum Perdata | 3 | 2 | Wajib | B1A | D1B01 |
| B1A003 | Hukum Administrasi Negara | 3 | 3 | Wajib | B1A | D1A01 |
| C1A001 | Manajemen Pemasaran | 3 | 1 | Wajib | C1A | D1B01 |
| C1A002 | Akuntansi Keuangan | 3 | 2 | Wajib | C1A | D1A01 |
| C1A003 | Ekonomi Makro | 3 | 1 | Wajib | C1A | D1A01 |
| D1A001 | Ilmu Politik Dasar | 3 | 1 | Wajib | D1A | D1A01 |
| D1A002 | Sosiologi Umum | 3 | 2 | Wajib | D1A | D1B01 |
| D1A003 | Psikologi Sosial | 3 | 3 | Wajib | D1A | D1A01 |

Gambar 28 Tabel Mata Kuliah

| id_semester | tahun_ajaran | semester | tanggal_mulai | tanggal_selesai | batas_pembayaran_ukt |
|---|---|---|---|---|---|
| 20201 | 2020/2021 | Ganjil | 2020-08-01 | 2021-01-31 | 2020-09-15 |
| 20202 | 2020/2021 | Genap | 2021-02-01 | 2021-07-31 | 2021-03-15 |
| 20211 | 2021/2022 | Ganjil | 2021-08-01 | 2022-01-31 | 2021-09-15 |
| 20212 | 2021/2022 | Genap | 2022-02-01 | 2022-07-31 | 2022-03-15 |
| 20221 | 2022/2023 | Ganjil | 2022-08-15 | 2023-01-20 | 2022-09-10 |
| 20222 | 2022/2023 | Genap | 2023-02-06 | 2023-07-24 | 2023-03-06 |
| NULL | NULL | NULL | NULL | NULL | NULL |

Gambar 28 Tabel Semester

| id_nilai | npm | kode_mk | id_semester | nilai_tugas | nilai_uts | nilai_uas | nilai_akhir | huruf_mutu |
|---|---|---|---|---|---|---|---|---|
| 1 | G1A023049 | G1A001 | 20201 | 80 | 85 | 90 | 88 | A |
| 2 | G1A023049 | G1A002 | 20201 | 75 | 80 | 85 | 81 | B+ |
| 3 | B1A023054 | B1A001 | 20201 | 70 | 75 | 80 | 78 | B |
| 4 | B1A023054 | B1A002 | 20201 | 80 | 85 | 90 | 88 | A |
| 5 | C1A023059 | C1A001 | 20201 | 65 | 70 | 75 | 74 | B- |
| 6 | G1A023049 | G1A001 | 20201 | 80 | 85 | 90 | 88 | A |
| 7 | G1A023049 | G1A002 | 20201 | 75 | 80 | 85 | 81 | B+ |
| 8 | B1A023054 | B1A001 | 20201 | 70 | 75 | 80 | 78 | B |
| 9 | B1A023054 | B1A002 | 20201 | 80 | 85 | 90 | 88 | A |
| 10 | C1A023059 | C1A001 | 20201 | 65 | 70 | 75 | 74 | B- |

Gambar 29 Tabel Nilai

| id_pembayaran | npm | id_semester | tanggal_bayar | jumlah_bayar | status_pembayaran | metode_pembayaran | bukti_pembayaran |
|---|---|---|---|---|---|---|---|
| 1 | G1A023049 | 20201 | 2020-08-20 | 1500000.00 | Lunas | Transfer | bukti1.jpg |
| 2 | B1A023054 | 20201 | 2020-08-21 | 2000000.00 | Lunas | Transfer | bukti2.jpg |
| 3 | C1A023059 | 20201 | 2020-08-22 | 1800000.00 | Lunas | Transfer | bukti3.jpg |
| 4 | D1A023064 | 20201 | 2020-08-23 | 1600000.00 | Lunas | Transfer | bukti4.jpg |
| 5 | E1A023069 | 20201 | 2020-08-24 | 1700000.00 | Lunas | Transfer | bukti5.jpg |
| 6 | G1A023049 | 20201 | 2020-08-20 | 1500000.00 | Lunas | Transfer | bukti1.jpg |
| 7 | B1A023054 | 20201 | 2020-08-21 | 2000000.00 | Lunas | Transfer | bukti2.jpg |
| 8 | C1A023059 | 20201 | 2020-08-22 | 1800000.00 | Lunas | Transfer | bukti3.jpg |
| 9 | D1A023064 | 20201 | 2020-08-23 | 1600000.00 | Lunas | Transfer | bukti4.jpg |
| 10 | E1A023069 | 20201 | 2020-08-24 | 1700000.00 | Lunas | Transfer | bukti5.jpg |
| 11 | F1A023074 | 20201 | 2020-08-25 | 1500000.00 | Lunas | Transfer | bukti6.jpg |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

Gambar 30 Tabel Pembayaran

**Pembahasan output :**

Pada output di atas menampilkan beberapa tabel utama. Pertama tabel fakultas yang memiliki dua kolom, yaitu id_fakultas dan nama_fakultas, yang menyimpan informasi mengenai masing-masing fakultas. Tabel program studi terdiri dari kolom id_program_studi, nama_program_studi, jenjang, dan id_fakultas, yang digunakan untuk menyimpan data setiap program studi beserta jenjang pendidikan dan fakultas pengampunya.

Tabel golongan_ukt berisi kolom id_golongan_ukt, nama_golongan, nominal_ukt, dan id_program_studi, yang menyimpan informasi tentang kategori UKT berdasarkan program studi. Selanjutnya, tabel mahasiswa memiliki kolom npm, nama, alamat, no_telepon, email, tanggal_lahir, tanggal_masuk, status, id_program_studi, dan id_golongan_ukt, yang mencatat data lengkap mahasiswa termasuk latar belakang akademik dan golongan UKT nya. Lalu, tabel dosen memuat kolom nip, nama, alamat, no_telepon, email, jabatan, bidang_keahlian, dan id_fakultas, yang mencakup informasi pribadi dosen, serta keterkaitannya dengan fakultas. Selanjutnya yaitu tabel mata_kuliah yang memiliki kolom kode_mk, nama_mk, sks, semester_mf, jenis, id_program_studi, dan nip_dosen, yang digunakan untuk mendeskripsikan setiap mata kuliah beserta program studi dan dosen pengampunya.

Tabel semester mencakup kolom id_semester, tahun_ajaran, semester, tanggal_mulai, tanggal_selesai, dan batas_pembayaran_ukt, yang menyimpan informasi detail tentang setiap periode semester akademik. Tabel nilai terdiri dari kolom id_nilai, npm mahasiswa, kode_mk, id_semester, nilai_tugas, nilai_uts, nilai_uas, nilai_akhir, dan huruf_mutu, yang digunakan untuk mencatat nilai akademik mahasiswa dalam setiap mata kuliah dan semester. Dan yang terakhir, tabel pembayaran_ukt memiliki kolom id_pembayaran, npm, id_semester, tanggal_bayar, jumlah_bayar, status_pembayaran, metode_pembayaran, dan bukti_pembayaran, yang menyimpan data transaksi pembayaran UKT mahasiswa setiap semester.

3.3 Lakukan join

**Printscreen source code :**

```
-- JOIN `mahasiswa` dan `program_studi`
SELECT m.nama AS mahasiswa, p.nama_program_studi AS program_studi
FROM mahasiswa m
JOIN program_studi p ON m.id_program_studi = p.id_program_studi;
-- JOIN mahasiswa dengan golongan_ukt
SELECT m.nama, g.nama_golongan, g.nominal_ukt
FROM mahasiswa m
JOIN golongan_ukt g ON m.id_golongan_ukt = g.id_golongan_ukt;
-- JOIN nilai dengan mata_kuliah dan dosen
SELECT n.npm, mk.nama_mk, d.nama AS dosen_pengampu, n.nilai_akhir
FROM nilai n
JOIN mata_kuliah mk ON n.kode_mk = mk.kode_mk
JOIN dosen d ON mk.nip_dosen = d.nip;
```

Gambar 31 Source Code Join

**Source code :**

```
SELECT m.nama AS mahasiswa, p.nama_program_studi AS program_studi
FROM mahasiswa m
JOIN program_studi p ON m.id_program_studi = p.id_program_studi
    SELECT m.nama, g.nama_golongan, g.nominal_ukt
    FROM mahasiswa m
    JOIN golongan_ukt g ON m.id_golongan_ukt = g.id_golongan_ukt;
SELECT n.npm, mk.nama_mk, d.nama AS dosen_pengampu, n.nilai_akhir
FROM nilai n
JOIN mata_kuliah mk ON n.kode_mk = mk.kode_mk
JOIN dosen d ON mk.nip_dosen = d.nip;
```

**Pembahasan source code :**

Pada code di atas digunakan untuk penggabungan tabel dengan JOIN, pada SELECT pertama digunakan untuk menampilkan daftar mahasiswa beserta program studi yang mereka ambil. Tabel mahasiswa (diberi alias m) digabungkan dengan tabel program_studi

(alias p) menggunakan kolom id_program_studi sebagai *foreign key* relasinya. Output nya adalah nama mahasiswa dan nama program studi yang bersangkutan. Lalu, SELECT kedua mengambil data nama mahasiswa, nama golongan UKT (Uang Kuliah Tunggal), dan nominal UKT yang dibayarkan. Pada tabel mahasiswa digabung dengan tabel golongan_ukt berdasarkan id_golongan_ukt, untuk mengetahui besaran UKT yang dibebankan kepada setiap mahasiswa berdasarkan golongannya. Dan SELECT ketiga berfungsi untuk menampilkan nilai akhir mahasiswa dalam suatu mata kuliah. Tabel nilai digabung dengan tabel mata_kuliah melalui kolom kode_mk, lalu tabel mata_kuliah digabung lagi dengan tabel dosen berdasarkan nip_dosen (Nomor Induk Pegawai dosen pengampu).

**Printscreen output :**

| mahasiswa | program_studi |
|---|---|
| Oscar Gunawan | Bisnis Internasional |
| Putu Satria | Ilmu Politik |
| Rina Ayu | Sosiologi |
| Siti Wulandari | Antropologi |
| Tina Ambarwati | Ilmu Komunikasi |
| Umi Nabila | Psikologi |
| Vina Anjani | Agribisnis |
| Wawan Subagyo | Teknologi Pertanian |
| Xenia Putri | Ilmu Tanah |
| Yanti Amelia | Pendidikan Pertanian |
| Zaki Akbar | Penyuluhan Pertanian |
| Alfiansyah Putra | Matematika |
| Budi Santosa | Fisika |
| Cahaya Indah | Kimia |
| Dani Rafi | Biologi |
| Elisa Dwi | Statistika |
| Andi Rahmat | Informatika |

Gambar 32 Tabel Join

| nama | nama_golongan | nominal_ukt |
|---|---|---|
| Oscar Gunawan | Golongan E | 4000000.00 |
| Putu Satria | Golongan A | 2000000.00 |
| Rina Ayu | Golongan B | 2500000.00 |
| Siti Wulandari | Golongan C | 3000000.00 |
| Tina Ambarwati | Golongan D | 3500000.00 |
| Umi Nabila | Golongan E | 4000000.00 |
| Vina Anjani | Golongan A | 2000000.00 |
| Wawan Subagyo | Golongan B | 2500000.00 |
| Xenia Putri | Golongan C | 3000000.00 |
| Yanti Amelia | Golongan D | 3500000.00 |
| Zaki Akbar | Golongan E | 4000000.00 |
| Alfiansyah Putra | Golongan A | 2000000.00 |
| Budi Santosa | Golongan B | 2500000.00 |
| Cahaya Indah | Golongan C | 3000000.00 |

Gambar 33 Tabel Join

| npm | nama_mk | dosen_pengampu | nilai_akhir |
|---|---|---|---|
| G1A023049 | Algoritma dan Pemrograman | Dr. Asep Anwar | 88 |
| G1A023049 | Jaringan Komputer | Dr. Rina Sari | 81 |
| B1A023054 | Hukum Pidana | Dr. Asep Anwar | 78 |
| B1A023054 | Hukum Perdata | Dr. Rina Sari | 88 |
| C1A023059 | Manajemen Pemasaran | Dr. Rina Sari | 74 |
| G1A023049 | Algoritma dan Pemrograman | Dr. Asep Anwar | 88 |
| G1A023049 | Jaringan Komputer | Dr. Rina Sari | 81 |
| B1A023054 | Hukum Pidana | Dr. Asep Anwar | 78 |
| B1A023054 | Hukum Perdata | Dr. Rina Sari | 88 |
| C1A023059 | Manajemen Pemasaran | Dr. Rina Sari | 74 |
| D1A023064 | Ilmu Politik Dasar | Dr. Asep Anwar | 89 |
| D1A023064 | Sosiologi Umum | Dr. Rina Sari | 83 |

Gambar 34 Tabel Join

**Pembahasan output :**

Pada output tabel diatas merupakan penggunaan join untuk menggabungkan beberapa tabel. Pada tabel pertama, ditampilkan dua kolom yaitu nama mahasiswa dan nama program studi, yang menunjukkan setiap mahasiswa dan program studi tempat mereka terdaftar, misalnya: “Oscar, Bisnis Internasional” atau “Putu, Ilmu Politik”. Output ini berguna untuk mengetahui identitas akademik mahasiswa beserta jurusan atau program studi yang mereka ambil.

Pada output perintah kedua, akan ditampilkan tiga kolom yaitu nama mahasiswa, nama golongan UKT, dan nominal UKT. Contoh hasilnya bisa berupa: “Andi - Golongan 3 - 3.500.000”. Output ini memberikan informasi mengenai klasifikasi biaya kuliah masing-masing mahasiswa berdasarkan golongan yang ditentukan, serta jumlah nominal UKT yang harus dibayarkan setiap semester. Sementara itu, output dari perintah ketiga akan menyajikan empat kolom yakni, npm mahasiswa, nama mata kuliah, nama dosen pengampu, dan nilai akhir. Contohnya: “23421001 - Basis Data - Ibu Rina - 87”. Output ini memberikan gambaran akademik mahasiswa dalam suatu mata kuliah tertentu, termasuk siapa dosen yang mengampu mata kuliah tersebut.

3.4 Lakukan Alias dan Operator

**Printscreen source code :**

```
-- Alias digunakan untuk membuat query lebih mudah dibaca
SELECT m.nama AS mahasiswa, p.nama_program_studi AS program_studi
FROM mahasiswa m
JOIN program_studi p ON m.id_program_studi = p.id_program_studi;
-- Alias untuk nominal pembayaran
SELECT p.jumlah_bayar AS nominal, s.tahun_ajaran AS tahun
FROM pembayaran_ukt p
JOIN semester s ON p.id_semester = s.id_semester;
```

Gambar 35 Source Code Alias

```sql
-- Alias untuk nama fakultas dan prodi
SELECT f.nama_fakultas AS fakultas, ps.nama_program_studi AS prodi
FROM program_studi ps
JOIN fakultas f ON ps.id_fakultas = f.id_fakultas;
```

Gambar 36 Source Code Alias

```sql
-- Operator `BETWEEN` untuk mencari nilai dalam rentang tertentu
SELECT nama, tanggal_lahir
FROM mahasiswa
WHERE tanggal_lahir BETWEEN '2000-01-01' AND '2005-12-31';
-- Operator `IN` untuk memilih nilai yang ada dalam daftar
SELECT nama, status
FROM mahasiswa
WHERE status IN ('Aktif', 'Cuti');
-- Operator LIKE
SELECT nama
FROM mahasiswa
WHERE nama LIKE 'A%';
-- Operator NOT IN
SELECT nama, status
FROM mahasiswa
WHERE status NOT IN ('Lulus', 'DO');
```

Gambar 37 Source Code Operator

**Source code :**

```sql
SELECT m.nama AS mahasiswa, p.nama_program_studi AS program_studi
FROM mahasiswa m
JOIN program_studi p ON m.id_program_studi = p.id_program_studi;
    SELECT p.jumlah_bayar AS nominal, s.tahun_ajaran AS tahun
    FROM pembayaran_ukt p
    JOIN semester s ON p.id_semester = s.id_semester;
SELECT f.nama_fakultas AS fakultas, ps.nama_program_studi AS prodi
FROM program_studi ps
JOIN fakultas f ON ps.id_fakultas = f.id_fakultas;

SELECT nama, tanggal_lahir
FROM mahasiswa
WHERE tanggal_lahir BETWEEN '2000-01-01' AND '2005-12-31';
    SELECT nama, status
    FROM mahasiswa
    WHERE status IN ('Aktif', 'Cuti');
SELECT nama
```

```
FROM mahasiswa
WHERE nama LIKE 'A%';
    SELECT nama, status
    FROM mahasiswa
    WHERE status NOT IN ('Lulus', 'DO');
```

**Pembahasan source code :**

Pada code di atas digunakan untuk mengambil data dari berbagai tabel dengan penggunaan alias dan operator, baik dengan menggunakan JOIN untuk menggabungkan antar tabel, dan kondisi WHERE untuk menyaring data berdasarkan kriteria tertentu. Seperti halnya saat menggunakan alias, bisa menyingkat penulisan seperti m.nama alih-alih mahasiswa.nama. Pada SELECT pertama melakukan JOIN antara tabel mahasiswa dan program_studi berdasarkan kolom id_program_studi, untuk menampilkan dua kolom, yaitu nama mahasiswa dan nama program studi yang diambil oleh mahasiswa tersebut. Tujuannya adalah untuk mengetahui setiap mahasiswa terdaftar di program studi yang mana. Perintah SELECT kedua menggabungkan tabel pembayaran_ukt dan semester melalui kolom id_semester, untuk mengetahui jumlah nominal pembayaran UKT mahasiswa dalam tiap semester.

Perintah SELECT ketiga melakukan JOIN antara tabel program_studi dan fakultas berdasarkan id_fakultas untuk menampilkan nama fakultas dan program studi di bawah naungannya. Selanjutnya, WHERE tanggal_lahir BETWEEN ‘2000-01-01’ AND ‘2005-12-31’, digunakan operator BETWEEN untuk mengambil data dalam rentang tanggal tertentu. WHERE status IN (‘Aktif’, ‘Cuti’), digunakan operator IN untuk memfilter data yang memiliki status tertentu. Kemudian, menampilkan nama-nama mahasiswa yang diawali dengan huruf “A” menggunakan WHERE nama LIKE ‘A%’, digunakan operator LIKE untuk mencari data berdasarkan pola karakter. Dalam hal ini, ‘A%’ berarti mencari nama yang dimulai dengan huruf “A”. Simbol % berfungsi sebagai wildcard yang mewakili nol atau lebih karakter setelah huruf “A”. Terakhir, menampilkan nama dan status mahasiswa yang tidak termasuk dalam status “Lulus” atau “DO” (Drop Out), menggunakan kondisi NOT IN. Hasil ini menyajikan mahasiswa yang masih terdaftar dalam sistem kecuali yang sudah menyelesaikan studi atau keluar.

**Printscreen output :**

| mahasiswa | program_studi |
|---|---|
| Farah Dewi | Hukum Perdata |
| Gilang Sutrisno | Hukum Pidana |
| Hannah Nur | Hukum Tata Negara |
| Ikhsan Mulyadi | Hukum Internasional |
| Jasmine Amalia | Hukum Ekonomi |

Gambar 38 Tabel Alias

| nominal | tahun |
|---|---|
| 1500000.00 | 2020/2021 |
| 2000000.00 | 2020/2021 |
| 1800000.00 | 2020/2021 |
| 1600000.00 | 2020/2021 |
| 1700000.00 | 2020/2021 |
| 1500000.00 | 2020/2021 |
| 2000000.00 | 2020/2021 |
| 1800000.00 | 2020/2021 |
| 1600000.00 | 2020/2021 |
| 1700000.00 | 2020/2021 |
| 1500000.00 | 2020/2021 |

Gambar 39 Tabel Alias

| fakultas | prodi |
|---|---|
| Fakultas Teknik | Informatika |
| Fakultas Teknik | Teknik Elektro |
| Fakultas Teknik | Teknik Sipil |
| Fakultas Teknik | Teknik Mesin |
| Fakultas Teknik | Arsitektur |
| Fakultas Hukum | Hukum Perdata |
| Fakultas Hukum | Hukum Pidana |
| Fakultas Hukum | Hukum Tata Negara |

Gambar 40 Alias

| nama | tanggal_lahir |
|---|---|
| Farah Dewi | 2000-06-12 |
| Ikhsan Mulyadi | 2001-09-21 |
| Jasmine Amalia | 2000-10-25 |
| Lina Mardiana | 2000-12-14 |
| Martha Desy | 2001-01-17 |
| Nadia Ananda | 2000-02-22 |
| Putu Satria | 2000-04-02 |
| Tina Ambarwati | 2001-07-12 |
| Umi Nabila | 2000-08-25 |
| Vina Anjani | 2000-09-10 |

Gambar 41 Operator Between

| nama | status |
|---|---|
| Lina Mardiana | Aktif |
| Martha Desy | Aktif |
| Nadia Ananda | Aktif |
| Oscar Gunawan | Aktif |
| Putu Satria | Aktif |
| Rina Ayu | Aktif |
| Siti Wulandari | Aktif |
| Tina Ambarwati | Aktif |
| Umi Nabila | Aktif |
| Vina Anjani | Aktif |
| Wawan Subagyo | Aktif |
| Xenia Putri | Aktif |

Gambar 42 Operator In

| | nama |
|---|---|
| ▶ | Alfiansyah Putra |
| | Andi Rahmat |

Gambar 43 Operator Like

| nama | status |
|---|---|
| Ikhsan Mulyadi | Aktif |
| Jasmine Amalia | Aktif |
| Kevin Cahya | Aktif |
| Lina Mardiana | Aktif |
| Martha Desy | Aktif |
| Nadia Ananda | Aktif |

Gambar 44 Operator Not In

**Pembahasan output :**

Pada *query* pertama, ditampilkan daftar mahasiswa beserta program studi yang mereka ambil. Misalnya, hasilnya bisa seperti "Farah - Hukum Perdata" atau "Gilang - Hukum Pidana", untuk melihat hubungan antara mahasiswa dan jurusan tempat mereka terdaftar. Output *query* kedua akan menunjukkan nominal pembayaran UKT yang dilakukan mahasiswa pada semester tertentu, seperti "1.500.000 - 2020/2021 Ganjil", untuk mengevaluasi jumlah pembayaran setiap semester.

Selanjutnya, output dari *query* ketiga akan menampilkan data nama fakultas beserta nama program studi yang berada di bawah naungan masing-masing fakultas. Sebagai contoh, "Fakultas Teknik - Teknik Informatika" atau "Fakultas Hukum - Hukum Pidana". Output *query* keempat, yang menyaring mahasiswa berdasarkan tanggal lahir antara tahun 2000 hingga 2005, akan menghasilkan daftar mahasiswa yang berada dalam rentang usia tertentu, misalnya: "Farah - 2000-06-12", untuk mengetahui usia mahasiswa.

Pada output query kelima, ditampilkan mahasiswa yang berstatus Aktif atau Cuti, misalnya "Lina - Aktif", untuk menyusun data mahasiswa yang masih berstatus aktif secara administratif, baik sedang mengikuti perkuliahan atau sedang cuti. Output *query* keenam akan menampilkan semua mahasiswa yang namanya diawali huruf "A", contohnya "Alfiansyah", "Andi", untuk pencarian berdasarkan nama tertentu. Terakhir, output *query* ketujuh menyajikan nama dan status mahasiswa yang tidak termasuk dalam kategori "Lulus" atau "DO" (Drop Out), untuk mengidentifikasi siapa saja yang masih tercatat sebagai mahasiswa aktif maupun yang belum menyelesaikan studi.

3.5 Lakukan Function, Grouping, Sorting

**Printscreen Source Code :**

```
-- Fungsi `COUNT` untuk menghitung jumlah mahasiswa
SELECT COUNT(*) AS total_mahasiswa
FROM mahasiswa;
```

Gambar 45 Source Code Function Count

```sql
-- Fungsi LOWER
SELECT LOWER(email) AS email_kecil
FROM dosen;
-- Fungsi NOW untuk tanggal saat ini
SELECT NOW() AS tanggal_sekarang;
-- Fungsi `UPPER` untuk mengubah teks menjadi huruf kapital
SELECT UPPER(nama) AS nama_mahasiswa
FROM mahasiswa;
```

Gambar 46 Source Code Lower, Now, Upper

```sql
-- Menghitung jumlah mahasiswa berdasarkan status
SELECT status, COUNT(*) AS jumlah_mahasiswa
FROM mahasiswa
GROUP BY status;
-- Total pembayaran per semester
SELECT s.semester, SUM(p.jumlah_bayar) AS total_pembayaran
FROM pembayaran_ukt p
JOIN semester s ON p.id_semester = s.id_semester
GROUP BY s.semester;
-- Jumlah mahasiswa per golongan UKT
SELECT g.nama_golongan, COUNT(*) AS jumlah_mahasiswa
FROM mahasiswa m
JOIN golongan_ukt g ON m.id_golongan_ukt = g.id_golongan_ukt
GROUP BY g.nama_golongan;
-- Menghitung nilai rata-rata berdasarkan program studi
SELECT p.nama_program_studi, AVG(n.nilai_akhir) AS rata_rata_nilai
FROM nilai n
JOIN mahasiswa m ON n.npm = m.npm
JOIN program_studi p ON m.id_program_studi = p.id_program_studi
GROUP BY p.nama_program_studi;
```

Gambar 47 Source Code Grouping

```sql
-- Mengurutkan mahasiswa berdasarkan nama
SELECT nama
FROM mahasiswa
ORDER BY nama ASC;
-- Mengurutkan mahasiswa berdasarkan nilai_akhir secara menurun
SELECT m.nama, n.nilai_akhir
FROM mahasiswa m
JOIN nilai n ON m.npm = n.npm
ORDER BY n.nilai_akhir DESC;
```

Gambar 48 Source Code Sorting

**Source code :**

```sql
SELECT COUNT(*) AS total_mahasiswa
FROM mahasiswa;
    SELECT LOWER(email) AS email_kecil
    FROM dosen;
SELECT NOW() AS tanggal_sekarang;
    SELECT UPPER(nama) AS nama_mahasiswa
    FROM mahasiswa;
```

```
SELECT status, COUNT(*) AS jumlah_mahasiswa
FROM mahasiswa
GROUP BY status;
    SELECT s.semester, SUM(p.jumlah_bayar) AS total_pembayaran
    FROM pembayaran_ukt p
    JOIN semester s ON p.id_semester = s.id_semester
    GROUP BY s.semester;
SELECT g.nama_golongan, COUNT(*) AS jumlah_mahasiswa
FROM mahasiswa m
JOIN golongan_ukt g ON m.id_golongan_ukt = g.id_golongan_ukt
GROUP BY g.nama_golongan;
    SELECT p.nama_program_studi, AVG(n.nilai_akhir) AS rata_rata_nilai
    FROM nilai n
    JOIN mahasiswa m ON n.npm = m.npm
    JOIN program_studi p ON m.id_program_studi = p.id_program_studi
    GROUP BY p.nama_program_studi;
SELECT nama
FROM mahasiswa
ORDER BY nama ASC;
    SELECT m.nama, n.nilai_akhir
    FROM mahasiswa m
    JOIN nilai n ON m.npm = n.npm
    ORDER BY n.nilai_akhir DESC;
```

**Pembahasan source code :**

Pada SELECT COUNT(*) AS total_mahasiswa FROM mahasiswa; menghasilkan angka total mahasiswa yang terdaftar, untuk mengetahui jumlah keseluruhan mahasiswa. Sementara itu, SELECT LOWER(email) AS email_kecil FROM dosen; menampilkan daftar email dosen dengan huruf kecil untuk memastikan konsistensi data. Perintah SELECT NOW() AS tanggal_sekarang; memberikan tanggal dan waktu saat ini yang berguna sebagai acuan waktu laporan. Kemudian, SELECT UPPER(nama) AS nama_mahasiswa FROM mahasiswa; menampilkan nama mahasiswa dengan huruf kapital yang cocok untuk keperluan tampilan resmi.

Selanjutnya, SELECT status, COUNT(*) AS jumlah_mahasiswa FROM mahasiswa GROUP BY status; menyajikan jumlah mahasiswa berdasarkan status mereka seperti Aktif, Cuti, atau Lulus, yang sangat bermanfaat dalam evaluasi akademik. Pada perintah

SELECT s.semester, SUM(p.jumlah_bayar) AS total_pembayaran, ditampilkan total pembayaran UKT pada tiap semester sebagai acuan pemasukan institusi per periode akademik. Kemudian, SELECT g.nama_golongan, COUNT(*) AS jumlah_mahasiswa menunjukkan sebaran jumlah mahasiswa berdasarkan golongan UKT, yang membantu dalam analisis pemerataan pembiayaan pendidikan.

Perintah SELECT p.nama_program_studi, AVG(n.nilai_akhir) AS rata_rata_nilai memberikan nilai rata-rata per program studi untuk menilai performa akademik mahasiswa dalam setiap jurusan. Sementara itu, SELECT nama FROM mahasiswa ORDER BY nama ASC; menghasilkan daftar nama mahasiswa yang terurut dari A ke Z dan sangat berguna untuk pencarian atau daftar presensi. Terakhir, SELECT m.nama, n.nilai_akhir ORDER BY n.nilai_akhir DESC; menyajikan peringkat mahasiswa berdasarkan nilai akhir, dari yang tertinggi hingga terendah, sehingga dapat diketahui siapa mahasiswa terbaik atau yang perlu pendampingan akademik.

**Printscreen output :**

| status | jumlah_mahasiswa |
|---|---|
| Aktif | 30 |

Gambar 49 Function Count

| email_kecil |
|---|
| asep@universitas.com |
| rina@universitas.com |
| budi@universitas.com |
| farah@universitas.com |
| joko@universitas.com |
| siti@universitas.com |
| agus@universitas.com |
| indah@universitas.com |
| rudi@universitas.com |
| amira@universitas.com |
| alfi@universitas.com |
| rahmawati@universita... |

Gambar 50 Function Lower

| tanggal_sekarang |
|---|
| 2025-05-09 01:45:21 |

Gambar 51 Function Now

| nama_mahasiswa |
|---|
| IKHSAN MULYADI |
| JASMINE AMALIA |
| KEVIN CAHYA |
| LINA MARDIANA |
| MARTHA DESY |

Gambar 52 Function Upper

| status | jumlah_mahasiswa |
|---|---|
| Aktif | 30 |

Gambar 53 Grouping

| semester | total_pembayaran |
|---|---|
| Ganjil | 18700000.00 |

Gambar 54 Grouping

| nama_golongan | jumlah_mahasiswa |
|---|---|
| Golongan A | 6 |
| Golongan B | 6 |
| Golongan C | 6 |
| Golongan D | 6 |
| Golongan E | 6 |

Gambar 55 Grouping

| nama_program_studi | rata_rata_nilai |
|---|---|
| Informatika | 84.5 |
| Hukum Perdata | 83 |
| Manajemen | 74 |
| Ilmu Politik | 86 |
| Agribisnis | 83 |
| Matematika | 90.5 |

Gambar 56 Grouping

| nama |
|---|
| Alfiansyah Putra |
| Andi Rahmat |
| Budi Santosa |
| Budi Setiawan |
| Cahaya Indah |
| Cindy Yuliana |
| Dani Rafi |
| Diana Utami |
| Elisa Dwi |

Gambar 57 Sorting

| nama | nilai_akhir |
|---|---|
| Alfiansyah Putra | 92 |
| Putu Satria | 89 |
| Alfiansyah Putra | 89 |
| Andi Rahmat | 88 |
| Farah Dewi | 88 |
| Andi Rahmat | 88 |
| Farah Dewi | 88 |
| Vina Anjani | 88 |

Gambar 58 Sorting

**Pembahasan output :**

Pada code memanfaatkan fungsi agregat, fungsi manipulasi teks, fungsi waktu, serta pengurutan data. Yang pertama menggunakan COUNT(*) untuk menghitung jumlah total

mahasiswa yang terdaftar di tabel mahasiswa. Hasilnya berupa satu nilai angka yang mewakili keseluruhan entri mahasiswa dalam database. SELECT kedua mengambil data email dari tabel dosen dan mengubah semua hurufnya menjadi kecil menggunakan fungsi LOWER(email), untuk membuat huruf nya menjadi kecil sesuai dengan standarisasi data email. Selanjutnya, SELECT ketiga menggunakan fungsi NOW() untuk menampilkan tanggal dan waktu saat ini ketika *query* dijalankan, dan biasa digunakan sebagai *timestamp* untuk laporan. Pada SELECT keempat, nama mahasiswa diambil dari tabel mahasiswa, lalu diubah ke huruf kapital seluruhnya menggunakan UPPER(nama). Fungsi untuk menampilkan nama dengan format seragam, misalnya untuk keperluan pencetakan dokumen. SELECT kelima menampilkan data status mahasiswa seperti "Aktif", "Cuti", atau "Lulus" beserta jumlah mahasiswa untuk masing-masing status tersebut, menggunakan fungsi COUNT(*) dan GROUP BY status;, untuk memberikan statistik sebaran mahasiswa berdasarkan status akademiknya. SELECT keenam menjumlahkan pembayaran UKT yang dilakukan mahasiswa pada tiap semester, menggunakan SUM(p.jumlah_bayar) dan pengelompokan berdasarkan semester dari tabel semester, utnuk menunjukkan total uang yang diterima kampus per semester.

Kemudian, ketujuh menghitung jumlah mahasiswa dalam masing-masing golongan UKT, dengan menggunakan JOIN antara mahasiswa dan golongan_ukt, lalu dikelompokkan berdasarkan nama_golongan, untuk melihat distribusi jumlah mahasiswa per kategori pembiayaan. Lalu, kedelapan menghitung nilai rata-rata menggunakan AVG (nilai_akhir) mahasiswa untuk setiap program studi. Penggabungan tabel nilai, mahasiswa, dan program_studi untuk mengelompokkan nilai berdasarkan program studi tempat mahasiswa tersebut terdaftar.

Selanjutnya yang kesembilan, ditampilkan seluruh nama mahasiswa dan disusun berdasarkan abjad secara menaik menggunakan ORDER BY nama ASC, untuk tampilan daftar nama yang rapi dan mudah dicari. Terakhir, kesepuluh menampilkan nama mahasiswa beserta nilai akhir mereka, kemudian diurutkan dari nilai tertinggi ke terendah ORDER BY n.nilai_akhir DESC, agar dapat melihat peringkat mahasiswa berdasarkan nilai akademik nya.

3.6 Lakukan View

**Printscreen source code :**

```
CREATE VIEW view_mahasiswa_aktif AS
SELECT nama, status
FROM mahasiswa
WHERE status = 'Aktif';
SELECT * FROM view_mahasiswa_aktif;
```

Gambar 59 Source Code View

```sql
-- View untuk melihat total pembayaran tiap mahasiswa
CREATE VIEW view_total_pembayaran_mahasiswa AS
SELECT m.npm, m.nama, SUM(p.jumlah_bayar) AS total_pembayaran
FROM mahasiswa m
JOIN pembayaran_ukt p ON m.npm = p.npm
GROUP BY m.npm, m.nama;

SELECT * FROM view_total_pembayaran_mahasiswa;

-- View untuk nilai mahasiswa per semester
CREATE VIEW view_nilai_mahasiswa_per_semester AS
SELECT n.npm, mk.nama_mk, s.semester, n.nilai_akhir
FROM nilai n
JOIN mata_kuliah mk ON n.kode_mk = mk.kode_mk
JOIN semester s ON n.id_semester = s.id_semester;

SELECT * FROM view_nilai_mahasiswa_per_semester;
```

Gambar 60 Source Code View

**Source code :**

```sql
CREATE VIEW view_mahasiswa_aktif AS
SELECT nama, status
FROM mahasiswa
WHERE status = 'Aktif';
SELECT * FROM view_mahasiswa_aktif;

-- View untuk melihat total pembayaran tiap mahasiswa
CREATE VIEW view_total_pembayaran_mahasiswa AS
SELECT m.npm, m.nama, SUM(p.jumlah_bayar) AS total_pembayaran
FROM mahasiswa m
JOIN pembayaran_ukt p ON m.npm = p.npm
GROUP BY m.npm, m.nama;
SELECT * FROM view_total_pembayaran_mahasiswa;

-- View untuk nilai mahasiswa per semester
CREATE VIEW view_nilai_mahasiswa_per_semester AS
SELECT n.npm, mk.nama_mk, s.semester, n.nilai_akhir
FROM nilai n
JOIN mata_kuliah mk ON n.kode_mk = mk.kode_mk
JOIN semester s ON n.id_semester = s.id_semester;
SELECT * FROM view_nilai_mahasiswa_per_semester;
```

**Pembahasan source code :**

Pada *code* di atas digunakan untuk membuat *view* agar dapat untuk menyederhanakan pengambilan data, menyajikan informasi spesifik, dan meningkatkan keamanan dengan membatasi akses ke data tertentu. Yang pertama, CREATE VIEW view_mahasiswa_aktif AS lalu SELECT nama, status FROM mahasiswa WHERE status = 'Aktif'; akan membuat sebuah view bernama view_mahasiswa_aktif yang hanya menampilkan nama dan status mahasiswa yang sedang aktif. Ketika SELECT * FROM view_mahasiswa_aktif; dijalankan, maka hasilnya adalah daftar mahasiswa aktif saja, untuk memudahkan pihak akademik untuk melihat siapa saja yang sedang menempuh studi.

Selanjutnya, CREATE VIEW view_total_pembayaran_mahasiswa AS untuk menampilkan total pembayaran UKT dari setiap mahasiswa, menggabungkan tabel mahasiswa dengan pembayaran_ukt melalui npm, dan menjumlahkan seluruh pembayaran dengan fungsi SUM() lalu mengelompokkan berdasarkan nama mahasiswa. SELECT * FROM view_total_pembayaran_mahasiswa; dijalankan, outputnya akan ditampilkan daftar mahasiswa beserta total pembayaran yang telah mereka lakukan. Terakhir, CREATE VIEW view_nilai_mahasiswa_per_semester AS membuat view untuk menampilkan nilai mahasiswa berdasarkan semester, digunkanan untuk menggabungkan tabel nilai, mata_kuliah, dan semester untuk menampilkan npm, nama mata kuliah, semester, dan nilai akhir.

**Printscreen output :**

| nama | status |
|---|---|
| Lina Mardiana | Aktif |
| Martha Desy | Aktif |
| Nadia Ananda | Aktif |
| Oscar Gunawan | Aktif |
| Putu Satria | Aktif |
| Rina Ayu | Aktif |
| Siti Wulandari | Aktif |
| Tina Ambarwati | Aktif |
| Umi Nabila | Aktif |
| Vina Anjani | Aktif |

Gambar 61 View

| npm | nama | total_pembayaran |
|---|---|---|
| G1A023049 | Andi Rahmat | 3000000.00 |
| B1A023054 | Farah Dewi | 4000000.00 |
| C1A023059 | Kevin Cahya | 3600000.00 |
| D1A023064 | Putu Satria | 3200000.00 |
| E1A023069 | Vina Anjani | 3400000.00 |
| F1A023074 | Alfiansyah Putra | 1500000.00 |

Gambar 62 View

| npm | nama_mk | semester | nilai_akhir |
|---|---|---|---|
| G1A023049 | Algoritma dan Pemrograman | Ganjil | 88 |
| G1A023049 | Jaringan Komputer | Ganjil | 81 |
| B1A023054 | Hukum Pidana | Ganjil | 78 |
| B1A023054 | Hukum Perdata | Ganjil | 88 |
| C1A023059 | Manajemen Pemasaran | Ganjil | 74 |
| G1A023049 | Algoritma dan Pemrograman | Ganjil | 88 |
| G1A023049 | Jaringan Komputer | Ganjil | 81 |
| B1A023054 | Hukum Pidana | Ganjil | 78 |
| B1A023054 | Hukum Perdata | Ganjil | 88 |
| C1A023059 | Manajemen Pemasaran | Ganjil | 74 |
| D1A023064 | Ilmu Politik Dasar | Ganjil | 89 |
| D1A023064 | Sosiologi Umum | Ganjil | 83 |
| E1A023069 | Agroekonomi | Ganjil | 78 |
| E1A023069 | Ilmu Tanah | Ganjil | 88 |

Gambar 63 View

**Pembahasan output :**

Pada tabel output diatas merupakan penggunaan *view*. view_mahasiswa_aktif akan menghasilkan daftar mahasiswa yang statusnya masih aktif, hanya menampilkan kolom nama dan status, sehingga mempermudah pihak kampus dalam melihat siapa saja yang masih terdaftar aktif tanpa harus memfilter data secara manual dari tabel mahasiswa. Kemudian, dari view_total_pembayaran_mahasiswa akan menampilkan kolom npm, nama, dan total_pembayaran yang merupakan hasil penjumlahan seluruh pembayaran UKT yang telah dilakukan masing-masing mahasiswa, untuk keperluan rekap administrasi pembayaran dan evaluasi keuangan mahasiswa. Sedangkan output dari view_nilai_mahasiswa_per_semester menyajikan daftar nilai akhir mahasiswa dalam setiap mata kuliah dan semester, dengan kolom npm, nama_mk, semester, dan nilai_akhir, sehingga memudahkan dosen atau bagian akademik dalam mengevaluasi capaian pembelajaran mahasiswa pada semester tertentu

3.7 Lakukan Store Procedure

**Printscreen source code :**

```
-- Mendapatkan mahasiswa berdasarkan status
DELIMITER $$

CREATE PROCEDURE get_mahasiswa_by_status(IN mahasiswa_status VARCHAR(20))
BEGIN
    SELECT nama, status
    FROM mahasiswa
    WHERE status = mahasiswa_status;
END $$

DELIMITER ;

CALL get_mahasiswa_by_status('Aktif');
```

Gambar 64 Source Code Store Procedure

```
-- Menampilkan total pembayaran UKT per mahasiswa
DELIMITER $$
CREATE PROCEDURE total_pembayaran_semua_mahasiswa()
BEGIN
    SELECT m.npm, m.nama, SUM(p.jumlah_bayar) AS total_bayar
    FROM mahasiswa m
    JOIN pembayaran_ukt p ON m.npm = p.npm
    GROUP BY m.npm, m.nama;
END$$
DELIMITER ;
CALL total_pembayaran_semua_mahasiswa();
```

Gambar 65 Source Code Store Procedure

```
-- Menampilkan nilai mahasiswa per semester
DELIMITER $$
CREATE PROCEDURE nilai_per_semester(IN semester_input VARCHAR(10))
BEGIN
    SELECT
        m.npm,
        m.nama,
        mk.nama_mk,
        n.nilai_akhir
    FROM nilai n
    JOIN mahasiswa m ON m.npm = n.npm
    JOIN mata_kuliah mk ON n.kode_mk = mk.kode_mk
    WHERE n.id_semester = semester_input;
END$$
DELIMITER ;
CALL nilai_per_semester('20201');
```

Gambar 66 Source Code Store Procedure

**Source code :**

```
DELIMITER $$
CREATE PROCEDURE get_mahasiswa_by_status(IN mahasiswa_status
    VARCHAR(20))
BEGIN
    SELECT nama, status
    FROM mahasiswa
    WHERE status = mahasiswa_status;
END $$
DELIMITER ;
CALL get_mahasiswa_by_status('Aktif');

-- Menampilkan total pembayaran UKT per mahasiswa
DELIMITER $$
CREATE PROCEDURE total_pembayaran_semua_mahasiswa()
```

```
BEGIN
   SELECT m.npm, m.nama, SUM(p.jumlah_bayar) AS total_bayar
   FROM mahasiswa m
   JOIN pembayaran_ukt p ON m.npm = p.npm
   GROUP BY m.npm, m.nama;
END$$
DELIMITER ;
CALL total_pembayaran_semua_mahasiswa();

-- Menampilkan nilai mahasiswa untuk 1 semester tertentu
DELIMITER $$
CREATE PROCEDURE nilai_per_semester(IN semester_input VARCHAR(10))
BEGIN
   SELECT
      m.npm,
      m.nama,
      mk.nama_mk,
      n.nilai_akhir
   FROM nilai n
   JOIN mahasiswa m ON m.npm = n.npm
   JOIN mata_kuliah mk ON n.kode_mk = mk.kode_mk
   WHERE n.id_semester = semester_input;
END$$
DELIMITER ;
CALL nilai_per_semester('20201');
```

**Pembahasan source code :**

Pada code di atas digunakan untuk beberapa stored procedures yang dirancang untuk mengambil data dari beberapa tabel dalam database yang berhubungan dengan mahasiswa, pembayaran UKT, dan nilai per semester. get_mahasiswa_by_status, mengambil nama dan status mahasiswa berdasarkan status yang diberikan sebagai parameter input. Prosedur ini memanfaatkan parameter mahasiswa_status untuk memilih mahasiswa yang memiliki status sesuai dengan input, misalnya Aktif. Prosedur kedua, total_pembayaran_semua_mahasiswa, untuk menghitung total pembayaran UKT untuk setiap mahasiswa dengan menggabungkan tabel mahasiswa dan pembayaran_ukt menggunakan JOIN berdasarkan npm, kemudian mengelompokkan hasilnya berdasarkan

npm dan nama mahasiswa. Prosedur terakhir, nilai_per_semester, digunakan untuk menampilkan nilai mahasiswa berdasarkan semester yang diberikan sebagai input, lalu menggabungkan tabel nilai, mahasiswa, dan mata_kuliah untuk menampilkan npm, nama, nama_mk, dan nilai_akhir mahasiswa yang terdaftar di semester yang dimaksud. Setiap prosedur dibuat dengan menggunakan DELIMITER untuk memastikan bahwa perintah SQL yang digunakan dalam prosedur diproses dengan benar. Setelah prosedur dibuat, prosedur-prosedur ini dapat dipanggil menggunakan perintah CALL dengan memberikan parameter yang sesuai, seperti 'Aktif', '20201', dan lain-lain, untuk menampilkan hasil yang diinginkan.

**Printscreen output :**

| nama | status |
|---|---|
| Farah Dewi | Aktif |
| Gilang Sutrisno | Aktif |
| Hannah Nur | Aktif |
| Ikhsan Mulyadi | Aktif |
| Jasmine Amalia | Aktif |
| Kevin Cahya | Aktif |
| Lina Mardiana | Aktif |
| Martha Desy | Aktif |
| Nadia Ananda | Aktif |

Gambar 67 Store Procedure

| npm | nama | total_bayar |
|---|---|---|
| G1A023049 | Andi Rahmat | 3000000.00 |
| B1A023054 | Farah Dewi | 4000000.00 |
| C1A023059 | Kevin Cahya | 3600000.00 |
| D1A023064 | Putu Satria | 3200000.00 |
| E1A023069 | Vina Anjani | 3400000.00 |
| F1A023074 | Alfiansyah Putra | 1500000.00 |

Gambar 68 Store Procedure

| npm | nama | nama_mk | nilai_akhir |
|---|---|---|---|
| G1A023049 | Andi Rahmat | Algoritma dan Pemrograman | 88 |
| G1A023049 | Andi Rahmat | Jaringan Komputer | 81 |
| B1A023054 | Farah Dewi | Hukum Pidana | 78 |
| B1A023054 | Farah Dewi | Hukum Perdata | 88 |
| C1A023059 | Kevin Cahya | Manajemen Pemasaran | 74 |
| G1A023049 | Andi Rahmat | Algoritma dan Pemrograman | 88 |
| G1A023049 | Andi Rahmat | Jaringan Komputer | 81 |
| B1A023054 | Farah Dewi | Hukum Pidana | 78 |
| B1A023054 | Farah Dewi | Hukum Perdata | 88 |
| C1A023059 | Kevin Cahya | Manajemen Pemasaran | 74 |
| D1A023064 | Putu Satria | Ilmu Politik Dasar | 89 |

Gambar 69 Store Procedure

**Pembahasan output :**

Pada Saat prosedur CALL get_mahasiswa_by_status('Aktif') dijalankan, output yang ditampilkan adalah daftar nama mahasiswa beserta statusnya yang berstatus "Aktif", sehingga kita bisa mengetahui siapa saja mahasiswa yang masih aktif dalam sistem. Kemudian, saat prosedur CALL total_pembayaran_semua_mahasiswa() dijalankan, hasilnya berupa daftar mahasiswa lengkap dengan npm dan nama, serta total pembayaran UKT masing-masing mahasiswa yang telah dijumlahkan dari tabel pembayaran_ukt. Ini berguna untuk melihat seberapa banyak pembayaran yang telah dilakukan setiap mahasiswa selama masa studi mereka. Terakhir, saat CALL nilai_per_semester('20201') dipanggil, output yang ditampilkan adalah daftar mahasiswa yang mengikuti perkuliahan pada semester 20201, dengan informasi berupa npm, nama mahasiswa, nama mata kuliah yang diambil, dan nilai akhir yang diperoleh.

3.8 Lakukan Nested Query

**Printscreen source code :**

```
-- Nilai mahasiswa yang nilai akhir lebih tinggi dari rata-rata nilai
SELECT nama
FROM mahasiswa
WHERE nim IN (
    SELECT nim
    FROM nilai
    WHERE nilai_akhir > (
        SELECT AVG(nilai_akhir) FROM nilai
    )
);
-- Nilai tertinggi per mahasiswa
SELECT nim, MAX(nilai_akhir) AS nilai_tertinggi
FROM nilai
GROUP BY nim;
-- Mahasiswa dengan UKT lebih tinggi dari rata-rata program studi-nya
SELECT m.nim, m.nama, g.nominal_ukt
FROM mahasiswa m
JOIN golongan_ukt g ON m.id_golongan_ukt = g.id_golongan_ukt
WHERE g.nominal_ukt > (
    SELECT AVG(g2.nominal_ukt)
    FROM golongan_ukt g2
    WHERE g2.id_program_studi = m.id_program_studi
);
```

Gambar 70 Source Code Nested Query

**Source code :**

--Menampilkan nama mahasiswa yang memiliki nilai akhir lebih tinggi dari rata-rata semua nilai

SELECT m.nama, n.nilai_akhir

FROM mahasiswa m

JOIN nilai n ON m.npm = n.npm

```
  WHERE n.npm IN (
    SELECT npm
    FROM nilai
    WHERE nilai_akhir > (
        SELECT AVG(nilai_akhir)
        FROM nilai
    )
);
-- Menampilkan nilai tertinggi yang diperoleh oleh setiap mahasiswa
SELECT npm, MAX(nilai_akhir) AS nilai_tertinggi
FROM nilai
GROUP BY npm;
```

**Pembahasan source code :**

Pada code di atas digunakan untuk menampilkan data terkait nilai mahasiswa. Bagian pertama merupakan *nested query* yang menampilkan nama mahasiswa dan nilai akhir yang lebih tinggi dari rata-rata seluruh nilai yang tercatat di tabel nilai. Dalam *query* ini, tabel mahasiswa *join* dengan tabel nilai melalui kolom npm, lalu menggunakan WHERE dengan membandingkan setiap nilai_akhir mahasiswa terhadap hasil dari SELECT AVG(nilai_akhir) FROM nilai, yaitu rata-rata semua nilai. Hanya mahasiswa yang memiliki nilai lebih tinggi dari rata-rata tersebut yang ditampilkan dalam hasil. Bagian kedua dari kode berfungsi untuk menampilkan nilai tertinggi MAX(nilai_akhir) yang pernah dicapai oleh setiap mahasiswa, tabel nilai dikelompokkan menggunakan GROUP BY berdasarkan npm untuk memastikan setiap mahasiswa hanya muncul satu kali dalam hasil, dengan nilai tertingginya.

**Printscreen output :**

| nama | nilai_akhir |
|---|---|
| Andi Rahmat | 88 |
| Andi Rahmat | 81 |
| Andi Rahmat | 88 |
| Andi Rahmat | 81 |
| Farah Dewi | 78 |
| Farah Dewi | 88 |
| Farah Dewi | 78 |
| Farah Dewi | 88 |
| Putu Satria | 89 |
| Putu Satria | 83 |
| Vina Anjani | 78 |
| Vina Anjani | 88 |
| Alfiansyah ... | 92 |

Gambar 71 Nested Query

| npm | nilai_tertinggi |
|---|---|
| B1A023054 | 88 |
| C1A023059 | 74 |
| D1A023064 | 89 |
| E1A023069 | 88 |
| F1A023074 | 92 |
| G1A023049 | 88 |

Gambar 72 Nested Query

**Pembahasan output :**

Pada tabel pertama akan menampilkan nama mahasiswa dan nilai akhirnya yang lebih tinggi dari rata-rata nilai seluruh mahasiswa yang terdaftar di tabel nilai. Hasil ini akan menunjukkan mahasiswa yang memiliki pencapaian akademik di atas rata-rata, memberikan gambaran tentang siapa saja yang unggul dalam nilai akhir. Kolom nama akan menampilkan nama mahasiswa, sedangkan kolom nilai_akhir akan menampilkan nilai akhir mereka yang lebih tinggi dari nilai rata-rata. Hasil ini bisa mencakup lebih dari satu mahasiswa jika ada beberapa yang memiliki nilai lebih tinggi dari rata-rata tersebut.

Output dari *query* kedua akan menampilkan nilai tertinggi yang diperoleh oleh setiap mahasiswa. Hasilnya adalah sebuah daftar yang menunjukkan npm mahasiswa bersama dengan nilai tertinggi yang tercatat dalam tabel nilai. Setiap mahasiswa akan muncul satu kali dalam hasil dengan nilai akhir terbaik yang mereka capai. Jika seorang mahasiswa memiliki beberapa nilai untuk mata kuliah yang berbeda, maka hanya nilai tertinggi yang akan ditampilkan.

3.9 Lakukan Triger

**Printscreen source code :**

```
-- Trigger untuk memastikan bahwa setiap pembayaran UKT yang diterima melebihi 500.000
DELIMITER $$
CREATE TRIGGER after_pembayaran_insert
AFTER INSERT ON pembayaran_ukt
FOR EACH ROW
BEGIN

UPDATE mahasiswa
    SET status_pembayaran = 'Lunas'
    WHERE npm = NEW.npm;
END $$
DELIMITER ;

INSERT INTO pembayaran_ukt (npm, id_semester, jumlah_bayar)
VALUES ('E1A023069', '20201', 600000);
SELECT * FROM pembayaran_ukt;
```

Gambar 73 Source Code Trigger

**Source code :**

```
DELIMITER $$

CREATE TRIGGER after_pembayaran_insert
AFTER INSERT ON pembayaran_ukt
FOR EACH ROW
BEGIN

    UPDATE mahasiswa
        SET status_pembayaran = 'Lunas'
        WHERE npm = NEW.npm;
    END $$

DELIMITER ;

INSERT INTO pembayaran_ukt (npm, id_smester, jumlah_bayar)
VALUES ('E1A023069', '20201', 600000);

SELECT * FROM pembayaran_ukt;
```

**Pembahasan source code :**

Pada code di atas digunakan untuk mengelola proses data pembayaran UKT mahasiswa secara otomatis, menggunakan fitur *trigger* di dalam sistem manajemen basis data. Pertama, digunakan DELIMITER $$ untuk mengganti tanda akhir perintah default (;) menjadi $$, agar memungkinkan penulisan blok kode trigger dengan mudah. Dengan demikian, DELIMITER $$ mempermudah dalam menyusun dan menjalankan logika pemrograman Kemudian dibuat sebuah *trigger* bernama after_pembayaran_insert yang dapat dieksekusi setelah ada data baru yang dimasukkan ke dalam tabel pembayaran_ukt AFTER INSERT.

Trigger ini berlaku untuk setiap baris yang dimasukkan FOR EACH ROW, Ketika data baru dimasukkan ke pembayaran_ukt, trigger akan mencari mahasiswa yang memiliki npm (Nomor Pokok Mahasiswa) yang sama dengan nilai npm pada baris yang baru dimasukkan, dan secara otomatis mengubah nilai status_pembayaran mahasiswa tersebut menjadi 'Lunas'. Pemanfaatan NEW.npm dalam *trigger* menunjukkan bahwa *trigger* mengambil data dari baris yang baru ditambahkan (yang memicu trigger), dan menjadikannya acuan untuk proses pembaruan di tabel lain. Mekanisme ini sangat berguna

untuk menjaga konsistensi data antar tabel, tanpa perlu melibatkan logika tambahan di aplikasi atau input manual dari pengguna.

Setelah *trigger* didefinisikan, INSERT INTO digunakan untuk menambahkan data pembayaran baru ke tabel pembayaran_ukt dengan npm E1A023069, semester 20201, dan jumlah pembayaran sebesar 600000. Terakhir, SELECT * FROM pembayaran_ukt digunakan untuk menampilkan seluruh isi tabel pembayaran_ukt guna memverifikasi bahwa data telah berhasil dimasukkan dan trigger telah berhasil dijalankan.

**Printscreen output :**

| id_pembayaran | npm | id_semester | tanggal_bayar | jumlah_bayar | status_pembayaran | metode_pembayaran | bukti_pembayaran |
|---|---|---|---|---|---|---|---|
| 1 | G1A023049 | 20201 | 2020-08-20 | 1500000.00 | Lunas | Transfer | bukti1.jpg |
| 2 | B1A023054 | 20201 | 2020-08-21 | 2000000.00 | Lunas | Transfer | bukti2.jpg |
| 3 | C1A023059 | 20201 | 2020-08-22 | 1800000.00 | Lunas | Transfer | bukti3.jpg |
| 4 | D1A023064 | 20201 | 2020-08-23 | 1600000.00 | Lunas | Transfer | bukti4.jpg |
| 5 | E1A023069 | 20201 | 2020-08-24 | 1700000.00 | Lunas | Transfer | bukti5.jpg |
| 6 | G1A023049 | 20201 | 2020-08-20 | 1500000.00 | Lunas | Transfer | bukti1.jpg |
| 7 | B1A023054 | 20201 | 2020-08-21 | 2000000.00 | Lunas | Transfer | bukti2.jpg |
| 8 | C1A023059 | 20201 | 2020-08-22 | 1800000.00 | Lunas | Transfer | bukti3.jpg |
| 9 | D1A023064 | 20201 | 2020-08-23 | 1600000.00 | Lunas | Transfer | bukti4.jpg |
| 10 | E1A023069 | 20201 | 2020-08-24 | 1700000.00 | Lunas | Transfer | bukti5.jpg |
| 11 | F1A023074 | 20201 | 2020-08-25 | 1500000.00 | Lunas | Transfer | bukti6.jpg |
| 21 | E1A023069 | 20201 | NULL | 600000.00 | NULL | NULL | NULL |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

Gambar 74 Trigger

**Pembahasan output :**

Pada tabel pembayaran_ukt menunjukkan hasil setelah dilakukan penambahan data dengan menggunakan INSERT, dan data sudah berhasil masuk ke baris ke-21 dengan npm E1A023069, id_semester 20201, dan jumlah_bayar sebesar 600000.00. Pada baris tersebut, kolom id_pembayaran masih bernilai NULL, yang mungkin disebabkan karena kolom tersebut belum diset sebagai auto increment atau sistem belum memperbaruinya secara otomatis.

Dari data ini, bisa dilihat bahwa mahasiswa dengan npm tersebut sudah melakukan pembayaran, dan jumlahnya sudah tercatat dengan benar di sistem. Selain itu, gambar ini juga menunjukkan bahwa trigger bernama after_pembayaran_insert sudah berjalan sesuai fungsinya. Trigger bekerja otomatis setelah data pembayaran ditambahkan, dan langsung mengubah status pembayaran mahasiswa di tabel mahasiswa menjadi Lunas. Artinya, sistem sudah berjalan otomatis dan tidak perlu lagi update manual untuk status pembayaran. Mekanisme ini sangat membantu agar data selalu konsisten dan memudahkan pengelolaan pembayaran mahasiswa dalam sistem akademik

3.10 Lakukan DCL

**Printscreen source code :**

```
-- Memberikan hak akses SELECT kepada user 'kelompok1' pada tabel 'mahasiswa'
GRANT SELECT ON sistem_akademik.* TO 'kelompok1'@'localhost';
FLUSH PRIVILEGES;

-- Mencabut hak akses SELECT dari user 'kelompok1'
REVOKE SELECT ON sistem_akademik.* FROM 'kelompok1'@'localhost';
FLUSH PRIVILEGES;
```

Gambar 75 Source CodeDCL

**Source code :**

-- Memberikan hak akses SELECT kepada user ‘kelompok1’ pada tabel ‘mahasiswa’

GRANT SELECT ON sistem_akademik.* TO ‘kelompok1’@’localhost’;

FLUSH PRIVILEGES;

-- Mencabut hak akses SELECT dari user ‘kelompok1’

REVOKE SELECT ON sistem_akademik.* FROM ‘kelompok1’@’localhost’;

FLUSH PRIVILEGES;

**Pembahasan source code :**

Pada code di atas digunakan untuk mengelola hak akses pengguna bernama kelompok1 terhadap database sistem_akademik. GRANT SELECT ON sistem_akademik.* TO 'kelompok1'@'localhost';dijalankan, pengguna bernama kelompok1akan memiliki izin untuk membaca data dari semua tabel dalam *database* sistem_akademik, dan perubahan ini segera diterapkan setelah perintah FLUSH PRIVILEGES dijalankan. Sebaliknya, ketika REVOKE SELECT ON sistem_akademik.* FROM 'kelompok1'@'localhost'; dijalankan, maka hak akses bacaan tersebut dicabut, dan pengguna tidak lagi dapat melakukan operasi SELECT pada *database* tersebut. Perintah FLUSH PRIVILEGES kembali dijalankan agar penabutan hak akses segera berlaku.

**Printscreen output :**

| | | | |
|---|---|---|---|
| ✓ | 207 | 04:22:12 | GRANT SELECT ON sistem_akademik.* TO 'kelompok1'@'localhost' |
| ✓ | 208 | 04:22:12 | FLUSH PRIVILEGES |

Gambar 76 Grant

| | | | |
|---|---|---|---|
| ✓ | 210 | 04:23:41 | REVOKE SELECT ON sistem_akademik.* FROM 'kelompok1'@'localhost' |
| ✓ | 211 | 04:23:41 | FLUSH PRIVILEGES |

Gambar 77 Revoke

```
mysql> GRANT SELECT ON sistem_akademik.mahasiswa TO 'root'@'localhost';
Query OK, 0 rows affected (0.00 sec)

mysql> REVOKE SELECT ON sistem_akademik.mahasiswa FROM 'root'@'localhost';
Query OK, 0 rows affected (0.00 sec)

mysql>
```

Gambar 78 Grant Dan Revoke di Mysql

**Pembahasan output :**

Pada tabel output diatas untuk mengelola hak akses pengguna bernama kelompok1 terhadap database sistem_akademik. GRANT digunakan untuk memberikan hak akses kepada pengguna ‘kelompok1’@’localhost’, yang berarti pengguna tersebut dapat melihat atau membaca seluruh data dari semua tabel dalam database sistem_akademik. FLUSH PRIVILEGES digunakan untuk memperbarui dan menerapkan perubahan hak akses tersebut secara langsung ke sistem.

Selanjutnya, jika hak akses tersebut tidak lagi diperlukan, REVOKE digunakan untuk mencabut hak dari pengguna yang sama, sehingga kelompok1tidak lagi memiliki izin untuk melihat data pada database tersebut. FLUSH PRIVILEGES kembali dijalankan untuk memastikan perubahan hak akses tersebut segera aktif.

Gambar tersebut menampilkan dua perintah SQL di Mysql yang digunakan untuk mengatur hak akses (privileges) pengguna terhadap sebuah tabel dalam basis data. Perintah pertama adalah GRANT SELECT ON sistem_akademik.mahasiswa TO 'root'@'localhost'; yang berfungsi untuk memberikan hak akses SELECT kepada *user root* yang terhubung dari localhost. Hak akses SELECT ini memungkinkan user tersebut untuk melihat atau membaca data dari tabel mahasiswa yang berada di dalam database sistem_akademik.

Setelah perintah dijalankan, sistem memberikan respons Query OK, 0 rows affected (0.00 sec), yang menandakan bahwa perintah berhasil dieksekusi meskipun tidak ada baris data yang langsung terpengaruh karena ini hanya menyangkut pengaturan hak akses. Perintah kedua adalah REVOKE SELECT ON sistem_akademik.mahasiswa FROM 'root'@'localhost'; yang berfungsi untuk mencabut kembali hak akses SELECT dari *user* yang sama, yaitu *root* pada localhost. Dengan perintah ini, *user root* tidak lagi memiliki izin untuk membaca data dari tabel mahasiswa. Seperti sebelumnya, sistem memberikan respons yang sama, menandakan bahwa perintah telah dijalankan dengan sukses. Kedua perintah ini menunjukkan proses pemberian dan pencabutan hak akses terhadap suatu objek basis data untuk mengatur keamanan dan kontrol terhadap data. Pengelolaan hak akses seperti ini penting untuk menjaga integritas dan kerahasiaan data di lingkungan basis data yang digunakan oleh institusi seperti Universitas.

# BAB IV
# PENUTUP

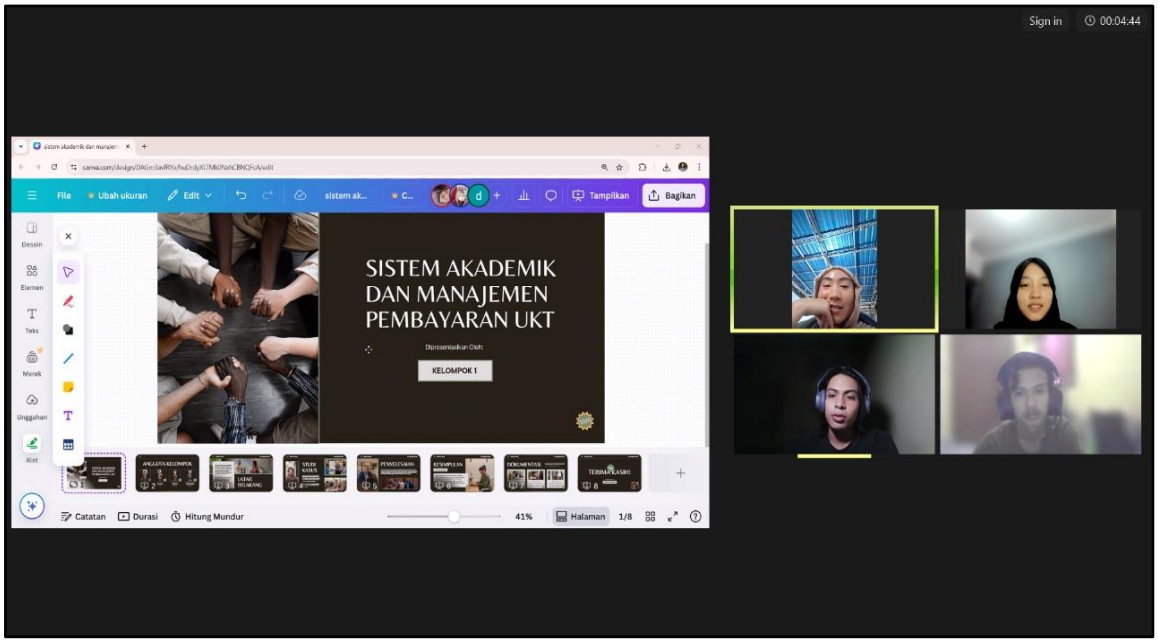
## 4.1 Kesimpulan

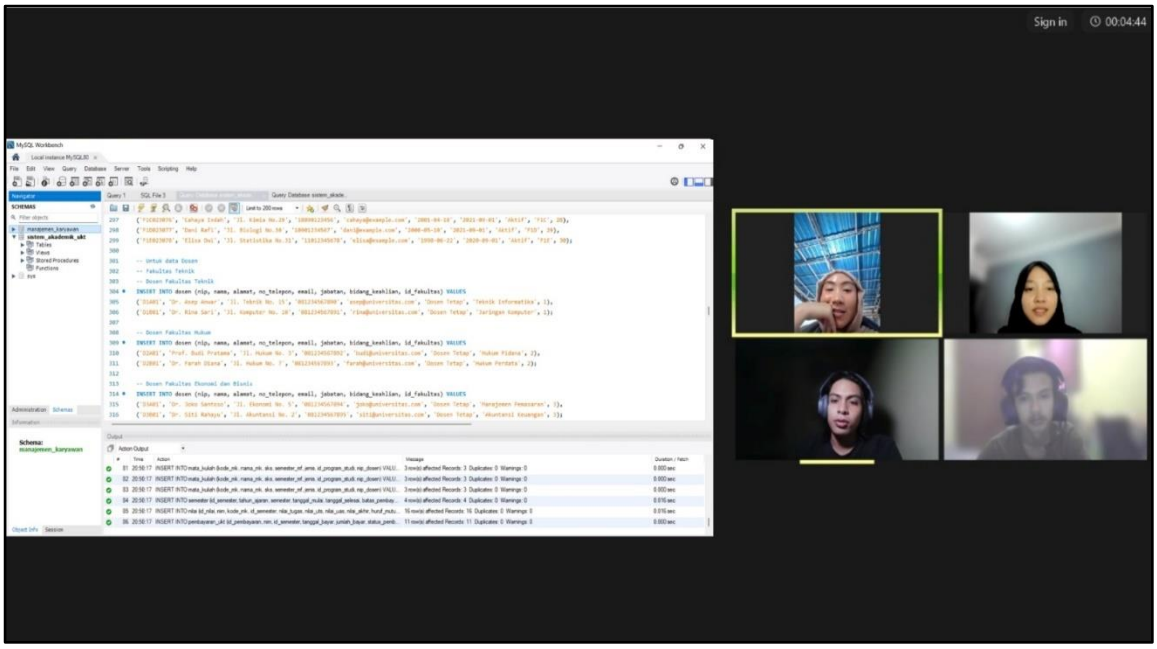
Sistem akademik dan manajemen pembayaran UKT mahasiswa merupakan teknologi yang dirancang untuk mengelola, memproses, dan menyebarkan informasi terkait kegiatan akademik serta administrasi keuangan mahasiswa secara efisien dan transparan. Sistem akademik mencakup pendaftaran mata kuliah, perwalian akademik, penilaian, serta pengelolaan data mahasiswa yang memungkinkan pihak kampus untuk memonitor perkembangan akademik mahasiswa secara akurat. Sedangkan, sistem manajemen pembayaran UKT bertujuan untuk memfasilitasi proses pembayaran biaya kuliah mahasiswa, memastikan pembayaran tercatat dengan baik, dan menjaga transparansi dalam pengelolaan keuangan. Dengan adanya sistem ini, efisiensi dalam pengelolaan administrasi akademik dan keuangan dapat meningkat, serta mengurangi potensi kesalahan atau kecurangan dalam proses administrasi.

## 4.2 Saran

Sistem akademik dan manajemen pembayaran UKT mahasiswa Sistem akademik dan manajemen pembayaran UKT mahasiswa ini penting untuk terus dikembangkan dan diperbarui kedua sistem ini dengan teknologi yang lebih canggih untuk meningkatkan kemudahan akses, akurasi data, dan efisiensi operasional. Integrasi antara sistem akademik dan manajemen pembayaran UKT perlu dilakukan secara optimal agar informasi yang dibutuhkan dapat diperoleh secara cepat dan akurat. Selain itu, pelatihan dan sosialisasi yang efektif kepada mahasiswa dan staf administrasi sangat penting untuk memastikan sistem ini dapat dimanfaatkan dengan baik dan memberikan manfaat yang maksimal.

## DOKUMENTASI PENGERJAAN




Jobdesk : 1. Yohanes membuat code

2. Yohanes membuat code

3. Aulia membuat Erd

4. Dimas membuat Erd

5. Syifa membuat Laporan.

## DAFTAR PUSTAKA


Annisa Mayasari, Y. S. (2021). Implementasi Sistem Informasi Manajemen Akademik Berbasis Teknologi Informasi dalam Meningkatkan Mutu Pelayanan Pembelajaran di SMK. STKIP Yapis Dompu.

Gupta, H. &. (2005). *Database Triggers: A Survey and Evaluation*. Association for Computing Machinery.

Nanda Arista Rizki, F. D. (2019). *Database System*. Universitas Mulawarman.

Papuangan, M. (2023). Sistem Basis Data. CV. Keranjang Teknologi Media.

Purwanto, R. (2017). Penerapan Sistem Informasi Akademik (SIA) sebagai Upaya Peningkatan Efektivitas dan Efisiensi Pengelolaan Akademik Sekolah. Politeknik Negeri Indramayu.

Rahman, M. &. (2011). Enhancing Database Triggers for Complex Data Management. IGI Global.

Sigit Indrawijaya, R. A. (2024). Pengembangan Fitur Manajemen Uang Kuliah Tunggal (UKT) Mahasiswa pada Sistem Registrasi Universitas Jambi. Asosiasi Perguruan Tinggi Informatika dan Komputer (APTIKOM).

Wibowo, A. P. (2019). Sistem Basis Data Lanjut. UniversitasTeknologiYogyakarta.

Wiwin Yuni Isnaeni, Z. Z. (2024). Pengelolaan Pembayaran Ukt Melalui Botclaim Tingkatkan Kepuasan Mahasiswa. Universitas KH. Mukhtar Syafaat Blokagung.

**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN**
**RISET DAN TEKNOLOGI**
**UNIVERSITAS BENGKULU**
**FAKULTAS TEKNIK**
**PROGRAM STUDI INFORMATIKA**
Jl. Wr. Supratman Kandang Limun, Bengkulu Bengkulu 38371 A
Telp: (0736) 344087, 22105 - 227

## LEMBAR ASISTENSI

## TUGAS BESAR PRAKTIKUM PROYEK BASIS DATA LANJUT

| | | |
|---|---|---|
| Nama Mahasiswa | : 1. Aulia Dwi Rahmadani | (G1A023043) |
| | 2. Yohanes Adi Prasetya | (G1A023049) |
| | 3. Waridhania As Syifa | (G1A023075) |
| | 4. I Nyoman Dimas Kresna Adryan | (G1A023077) |
| Dosen | : 1. Dr. Endina Putri Purwandari, S.T., M.Kom. | |
| | 2. Ir. Tiara Eka Putri, S.T., M.Kom. | |
| Asisten | : 1. Merly Yuni Purnama | (G1A021006) |
| | 2. Reksi Hendra Pratama | (G1A022032) |
| | 3. Sinta Ezra Wati Gulo | (G1A022040) |
| | 4. Fadlan Dwi Febrio | (G1A022051) |
| | 5. Torang Four Yones Manullang | (G1A022052) |
| | 6. Wahyu Ozorah Manurung | (G1A022060) |
| | 7. Shalaudin Muhammad Sah | (G1A022070) |
| | 8. Dian Ardiyanti Saputri | (G1A022084) |

| Laporan Tugas Besar | Catatan dan Tanda Tangan |
|---|---|
| Laporan Tugas Besar | |