

DATABASE MANAGEMENT SYSTEM OF ADMISSION OF NEW STUDENTS

LAPORAN PROJECT AKHIR

MATA KULIAH COMP6852004 – DATABASE TECHNOLOGY

KELAS BA20



Oleh :

2602199960 - RENDRA DWI PRASETYO

2602152982 – ALVINA RIAN TO PUTRA

2602124120 - IDA BAGUS SWAMBARA

Semester [Ganjil] 2023/2024

MALANG

BAB I

Pendahuluan

- Latar belakang

Seiring dengan peningkatan jumlah pendaftar mahasiswa dalam sebuah institusi atau universitas akan menyebabkan proses penerimaan mahasiswa yang semakin kompleks. Kompleksitas tersebut meliputi administrasi berkas, verifikasi dokumen, proses seleksi, dan pengelolaan informasi yang diperlukan dalam proses penerimaan mahasiswa baru.

Dalam upaya untuk meningkatkan efisiensi dan akurasi, institusi pendidikan membutuhkan sistem yang terkomputerisasi untuk mengelola semua tahapan penerimaan mahasiswa baru. Sistem semacam ini memungkinkan informasi dapat diakses dengan cepat, meminimalkan kesalahan manusiawi, dan mempercepat proses pengambilan keputusan.

Dengan adanya sistem database yang terintegrasi dengan baik, informasi mengenai penerimaan mahasiswa baru dapat dengan mudah diakses oleh berbagai departemen terkait seperti administrasi, keuangan, dan akademik. Hal ini akan membantu dalam meningkatkan layanan kepada mahasiswa dan mengoptimalkan pengelolaan informasi.

Sehingga dengan dibuatnya sistem database untuk penerimaan mahasiswa baru merupakan langkah strategis bagi institusi pendidikan dalam menghadapi tantangan volume data yang meningkat, kompleksitas proses penerimaan, serta kebutuhan akan pengelolaan dan pengambilan keputusan yang lebih efisien.

- Rumusan masalah

Terdapat suatu Universitas bernama Binus (Bina Negeri Univeristy) yang merupakan universitas fiktif-realistik di Indonesia. Universitas ini

ingin menyimpan informasi mengenai pendaftaran mahasiswa baru dalam bentuk database. Mereka ingin menyimpan informasi mengenai data diri calon mahasiswa. Data tersebut meliputi nama lengkap, alamat, tempat lahir, tanggal lahir, jenis kelamin, agama, email akademik, nomor induk siswa nasional, tahun lulus, nilai rata-rata rapor dan nilai dari Tes Potensi Keberhasilan Studi (TPKS). Kemudian database ini juga akan menyimpan informasi terkait kelengkapan pendaftaran berupa pengumpulan berkas dokumen ijazah dan Kartu keluarga (KK). Selain itu mereka juga ingin menyimpan informasi mengenai jurusan dan kampus yang dipilih, di Bina Negeri University ini setiap fakultas hanya berlokasi di 1 lokasi kampus, dan secara otomatis jurusan akan berlokasi di kampus yang menampung fakultas dari jurusan tersebut. Mereka juga ingin melihat daftar calon mahasiswa yang telah memenuhi syarat "diterima" dengan syarat nilai TPKS yang didapatkan (≥ 85) dan memiliki tahun lulus maksimal 2 tahun dari tahun pendaftaran.

- Tujuan

1. Meningkatkan pengelolaan data untuk memudahkan akses dan analisis informasi terkait calon mahasiswa baru yang mendaftar.
2. Membuat sistem yang aman untuk menyimpan dan memanipulasi data mahasiswa.

- Manfaat

1. Meningkatkan kecepatan dan ketepatan dalam proses pendaftaran mahasiswa baru
2. Meminimalisir kesalahan data serta duplikasi informasi

BAB II

Tinjauan Pustaka

A. Database dan DBMS

Database merupakan sekumpulan data yang berisi informasi yang terstruktur dan terhubung secara logis. Sehingga database didesain untuk memenuhi kebutuhan penyimpanan, pengelolaan, dan pengaksesan informasi oleh sebuah organisasi.

Sedangkan *Database Management System* (DBMS) merupakan seperangkat lunak yang memfasilitasi pengguna dalam mendefinisikan, membuat, memelihara, dan mengelola akses ke database.

B. MySQL

MySQL merupakan salah satu contoh dari Database Management System yang bersifat open source. MySQL menggunakan bahasa pemrograman berupa structured query language (SQL). SQL merupakan bahasa pemrograman yang digunakan untuk menyimpan dan memanipulasi informasi dalam basis data relasional.

MySQL memiliki beberapa kelebihan dalam menjamin keamanan, mendukung *multi-user*, penggunaan RAM yang murah, struktur tabel fleksibel, dan mendukung bahasa pemrograman yang lain selain SQL.

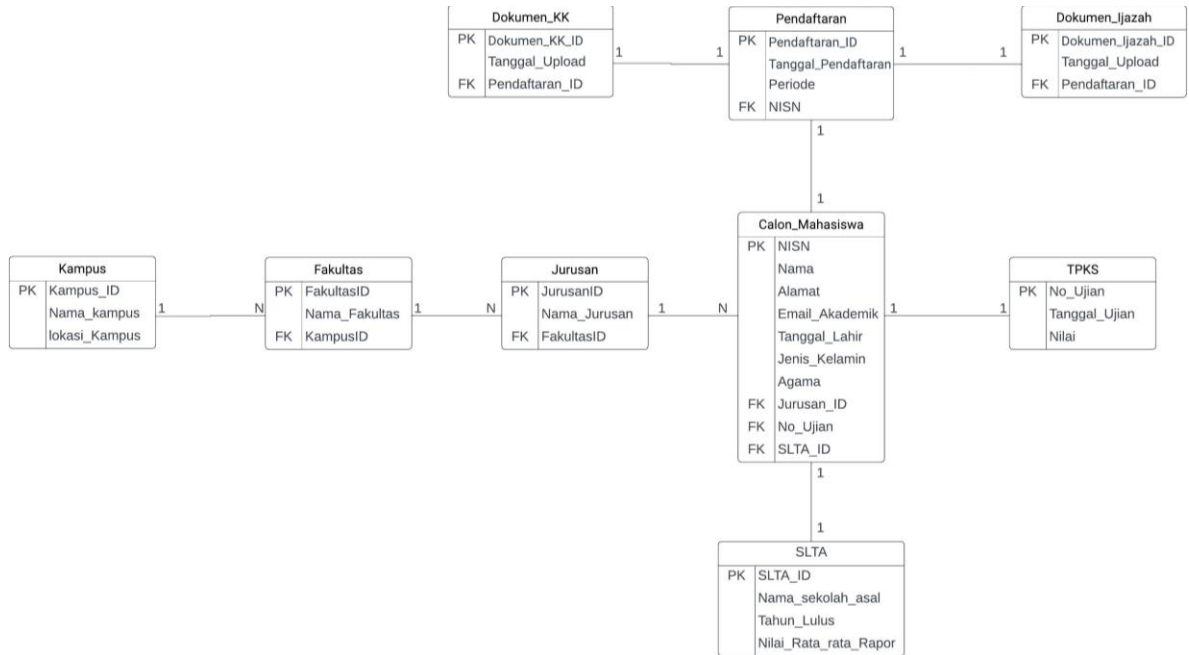
C. Entity Relational Diagram

Entity Relationship Diagram (ERD) adalah representasi grafis yang digunakan dalam perancangan database. Diagram ini memperlihatkan hubungan antara entitas, objek, atau elemen-elemen lain dalam suatu sistem database. Dengan menggunakan simbol-simbol khusus, ERD menggambarkan bagaimana entitas saling terhubung satu sama lain, serta atribut-atribut yang terkait dengan setiap entitas secara rinci. Diagram ini membantu dalam memvisualisasikan struktur data, relasi antar entitas, dan sifat atribut dalam suatu basis data, yang sangat penting dalam proses perancangan dan pemodelan database

BAB III

Hasil & Pembahasan

- Rancangan Database (ERD)



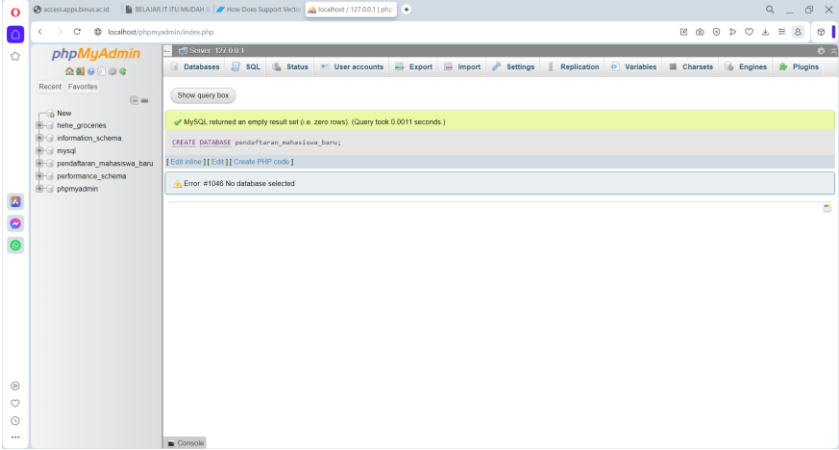
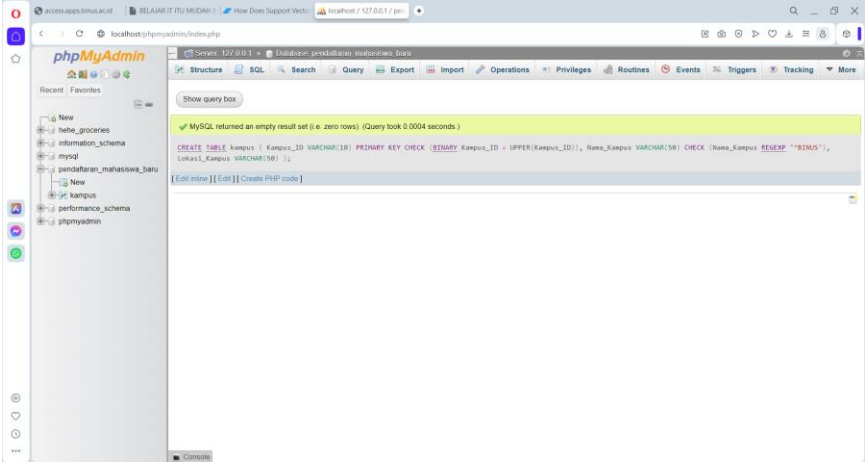
- List of atributes

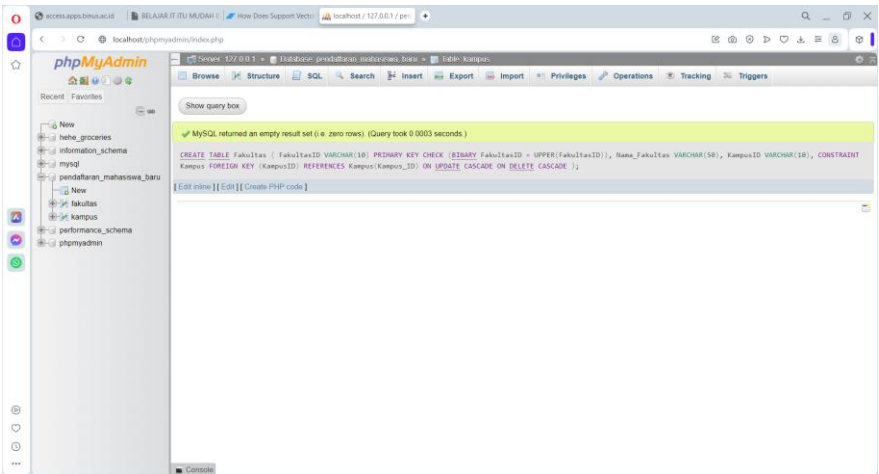
Entitas	Atributes	Tipe_Data & keterangan
Mahasiswa	NISN Nama Mahasiswa Alamat Email Tanggal Lahir Jenis Kelamin Agama Jurusan_ID SLTA_ID	VARCHAR(10), PK VARCHAR (30) VARCHAR(100) VARCHAR(100) DATE NUM VARCHAR(15) VARCHAR(10), FK VARCHAR(10), FK

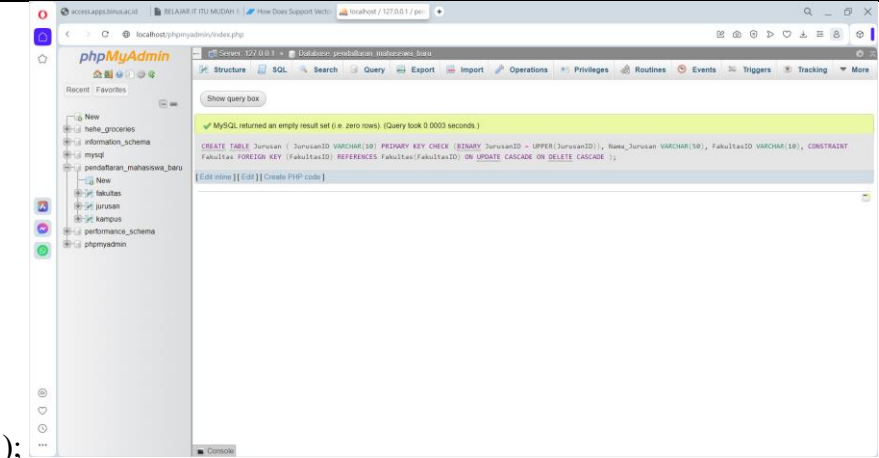
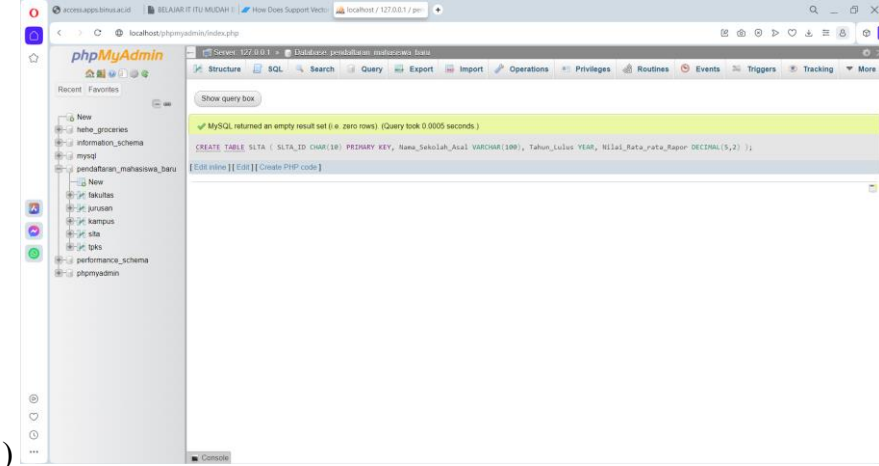
	No_Ujian	VARCHAR(10)
Jurusan	JurusanID Nama Jurusan FakultasID	VARCHAR(10), PK VARCHAR(50) VARCHAR(10), FK
Fakultas	FakultasID Nama Fakultas Kampus ID	VARCHAR(10), PK VARCHAR(50) VARCHAR(10), FK
Kampus	Kampus ID Nama Kampus Lokasi Kampus	VARCHAR(10), PK VARCHAR(50) VARCHAR(50)
Pendaftaran	Nomor pendaftaran Tanggal pendaftaran Periode (Tahun Masuk) NISN	VARCHAR(5), PK DATE YEAR VARCHAR(10), FK
Dokumen_KK	Dokumen_KK_ID Tanggal upload Pendaftaran ID	VARCHAR(5), PK DATE VARCHAR(5), FK
Dokumen_Ijazah	Dokumen_Ijazah_ID Tanggal upload Pendaftaran ID	VARCHAR(5), PK DATE VARCHAR(5), FK
TPKS	No_Ujian Nilai Tanggal_Ujian	VARCHAR(5), PK DECIMAL(5,2) DATE
SLTA	SLTA_ID Nama sekolah asal Tahun lulus Nilai Rata-rata rapor	CHAR(10), PK VARCHAR(100), YEAR, DECIMAL(5,2)

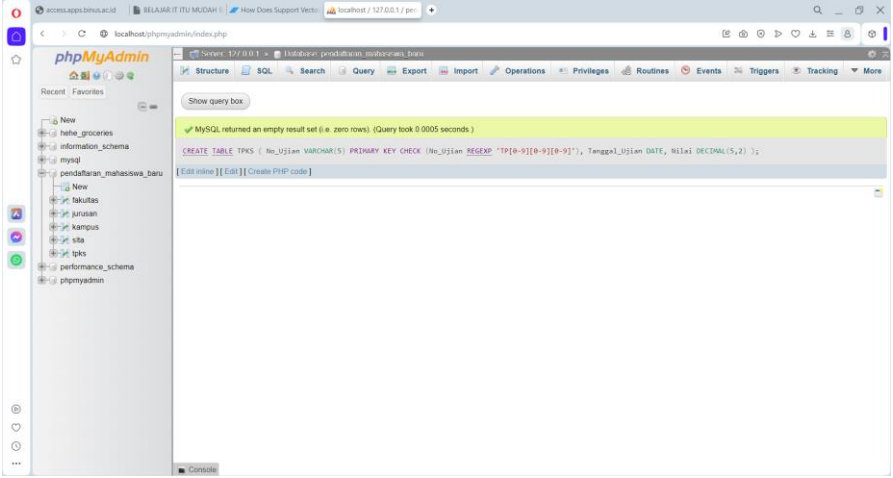
- SQL & Screenshot

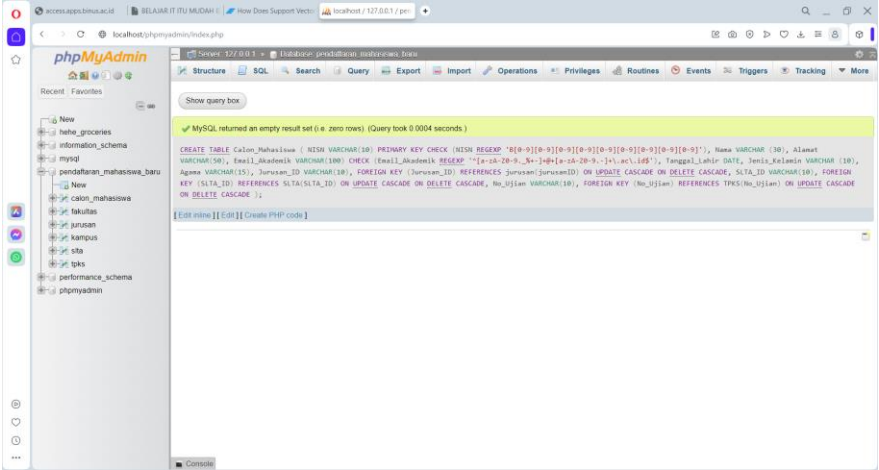
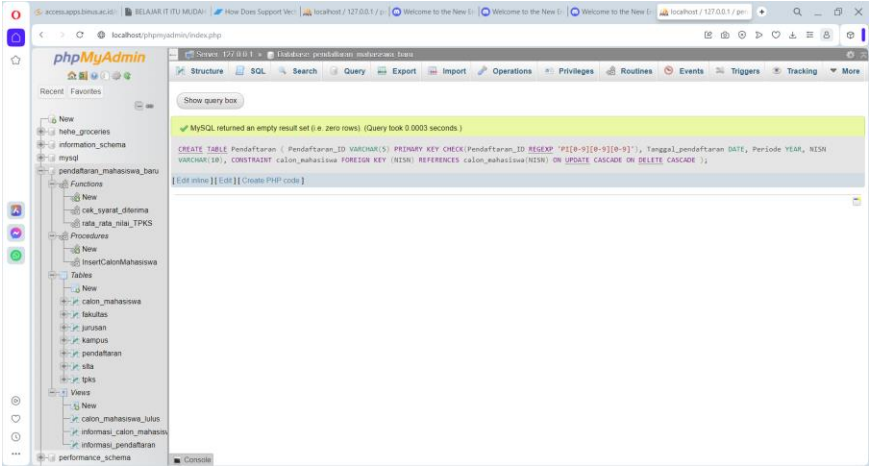
Membuat tabel dan dummy data

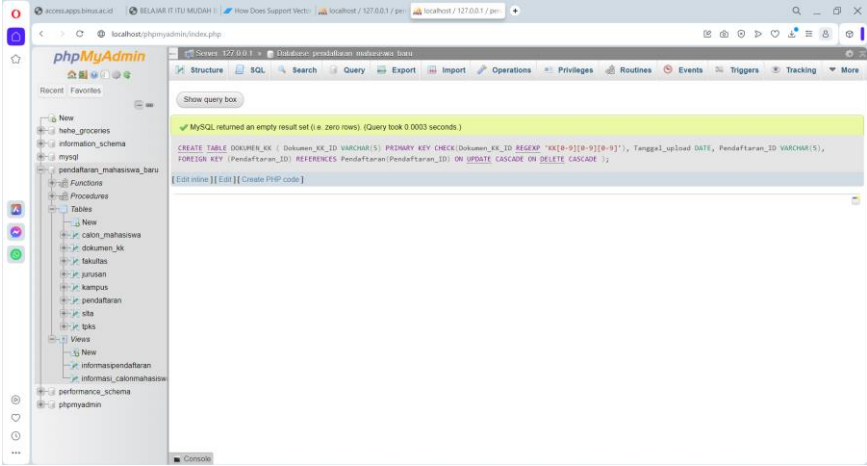
Perintah	SQL
Membuat Database Baru	<p>CREATE DATABASE pendaftaran_mahasiswa_baru</p> 
Membuat Tabel Kampus	<p>CREATE TABLE kampus (Kampus_ID VARCHAR(10) PRIMARY KEY CHECK (BINARY Kampus_ID = UPPER(Kampus_ID)), Nama_Kampus VARCHAR(50) CHECK (Nama_Kampus REGEXP '^BINUS'), Lokasi_Kampus VARCHAR(50))</p> 
Membuat Tabel Fakultas	<p>CREATE TABLE Fakultas (</p>

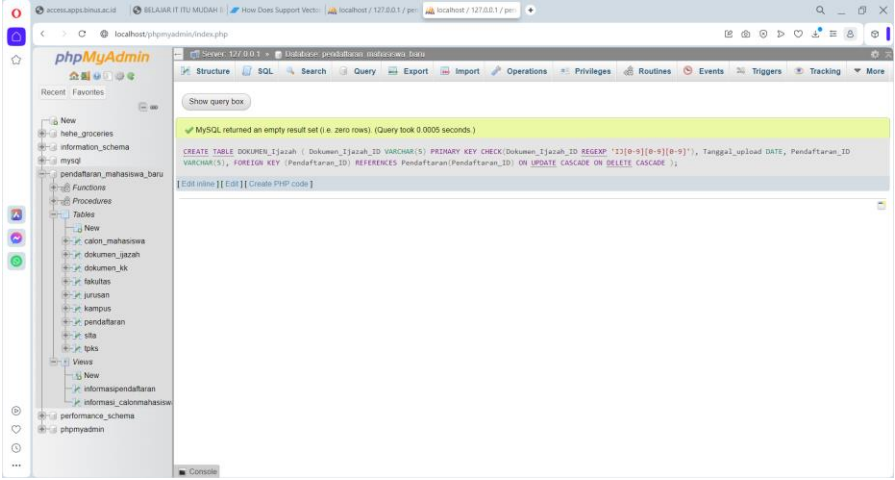
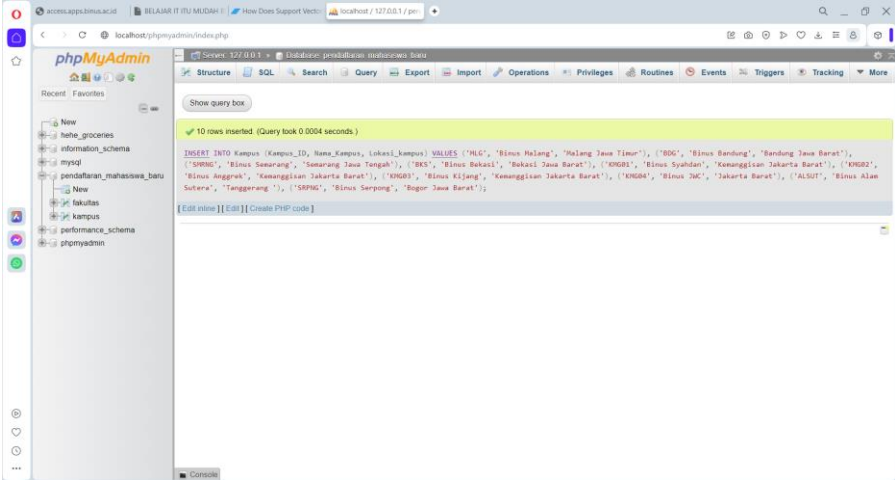
	<p>FakultasID VARCHAR(10) PRIMARY KEY CHECK (BINARY FakultasID = UPPER(FakultasID)),</p> <p>Nama_Fakultas VARCHAR(50),</p> <p>KampusID VARCHAR(10),</p> <p>CONSTRAINT Kampus FOREIGN KEY (KampusID) REFERENCES Kampus(Kampus_ID) ON UPDATE CASCADE ON DELETE CASCADE</p> <p>)</p> 
Membuat Tabel Jurusan	<p>CREATE TABLE Jurusan (</p> <p>JurusanID VARCHAR(10) PRIMARY KEY CHECK (BINARY JurusanID = UPPER(JurusanID)),</p> <p>Nama_Jurusan VARCHAR(50),</p> <p>FakultasID VARCHAR(10),</p> <p>CONSTRAINT Fakultas FOREIGN KEY (FakultasID) REFERENCES Fakultas(FakultasID) ON UPDATE CASCADE ON DELETE CASCADE</p>

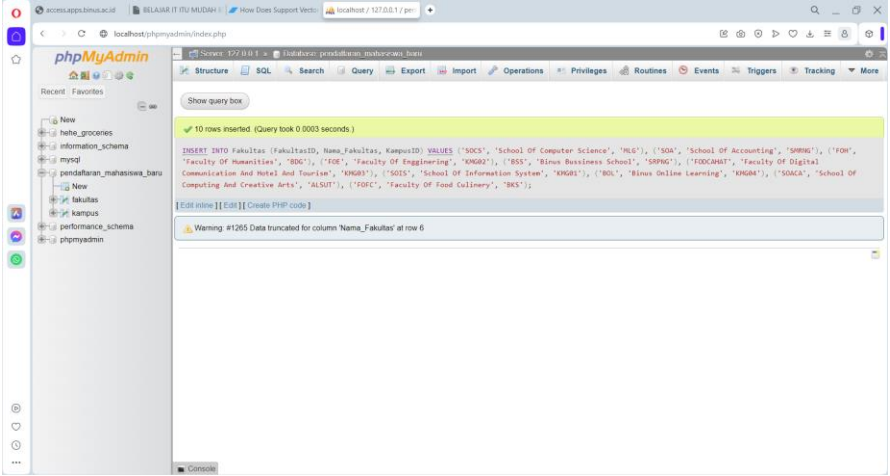
	 <p>);</p>
Membuat Tabel SLTA	<p>CREATE TABLE SLTA (</p> <p>SLTA_ID CHAR(10) PRIMARY KEY,</p> <p>Nama_Sekolah_Asal VARCHAR(100),</p> <p>Tahun_Lulus YEAR,</p> <p>Nilai_Rata_rata_Rapor DECIMAL(5,2)</p>  <p>)</p>
Membuat Tabel TPKS	<p>CREATE TABLE TPKS</p> <p>(</p> <p>No_Ujian VARCHAR(5) PRIMARY KEY CHECK (No_Ujian REGEXP 'TP[0-9][0-9][0-9]'),</p> <p>Tanggal_Ujian DATE,</p> <p>Nilai DECIMAL(5,2)</p>

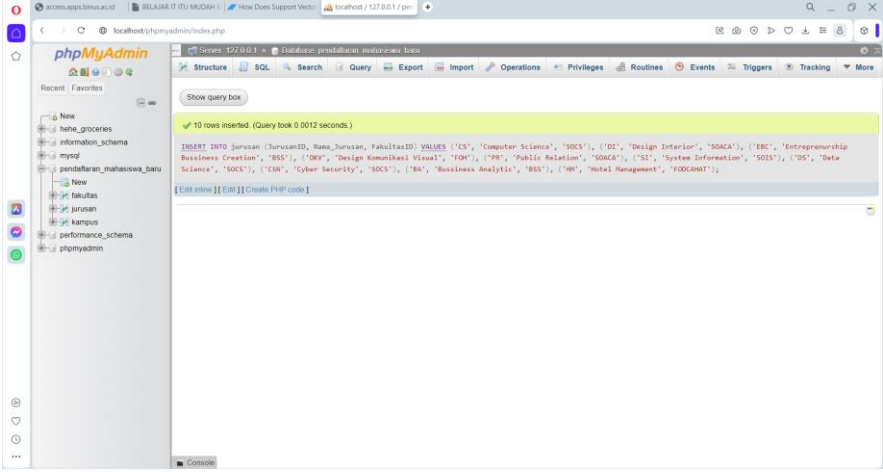
	<p>);</p> 
<p>Membuat Tabel Calon Mahasiswa</p>	<pre>CREATE TABLE Calon_Mahasiswa (NISN VARCHAR(10) PRIMARY KEY CHECK (NISN REGEXP 'B[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'), Nama VARCHAR (30), Alamat VARCHAR(100), Email_Akademik VARCHAR(100) CHECK (Email_Akademik REGEXP '[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.\ac\id\$'), Tanggal_Lahir DATE, Jenis_Kelamin VARCHAR (10), Agama VARCHAR(15), Jurusan_ID VARCHAR(10), FOREIGN KEY (Jurusan_ID) REFERENCES jurusan(jurusanID) ON UPDATE CASCADE ON DELETE CASCADE, SLTA_ID VARCHAR(10), FOREIGN KEY (SLTA_ID) REFERENCES SLTA(SLTA_ID) ON UPDATE CASCADE ON DELETE CASCADE, No_Ujian VARCHAR(10), FOREIGN KEY (No_Ujian) REFERENCES TPKS(No_Ujian) ON UPDATE CASCADE ON DELETE CASCADE</pre>

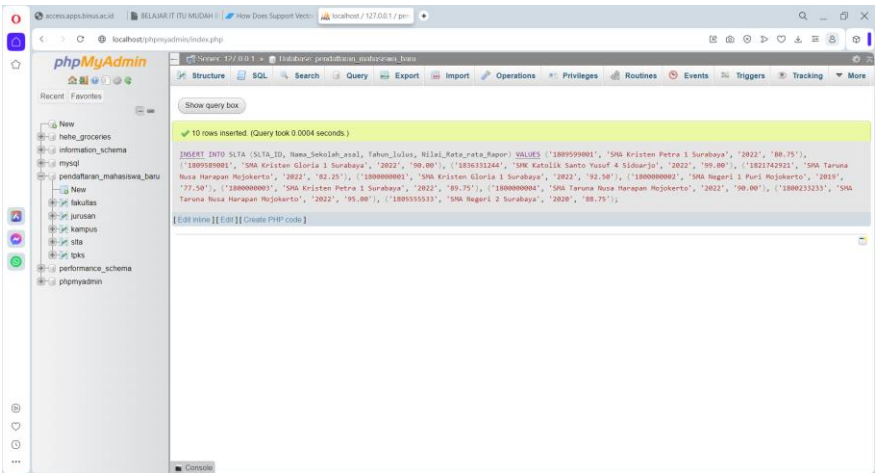
	<div><div>)</div><div></div></div>
Membuat Tabel Pendaftaran	<div><div>(</div><div><p>CREATE TABLE Pendaftaran</p><p>(</p><p> Pendaftaran_ID VARCHAR(5) PRIMARY KEY</p><p> CHECK(Pendaftaran_ID REGEXP ‘PI[0-9][0-9][0-9]’),</p><p> Tanggal_pendaftaran DATE,</p><p> Periode YEAR,</p><p> NISN VARCHAR(10),</p><p> CONSTRAINT calon_mahasiswa FOREIGN KEY (NISN)</p><p>REFERENCES calon_mahasiswa(NISN)</p><p>ON UPDATE CASCADE</p><p>ON DELETE CASCADE</p><p>)</p></div><div></div></div>

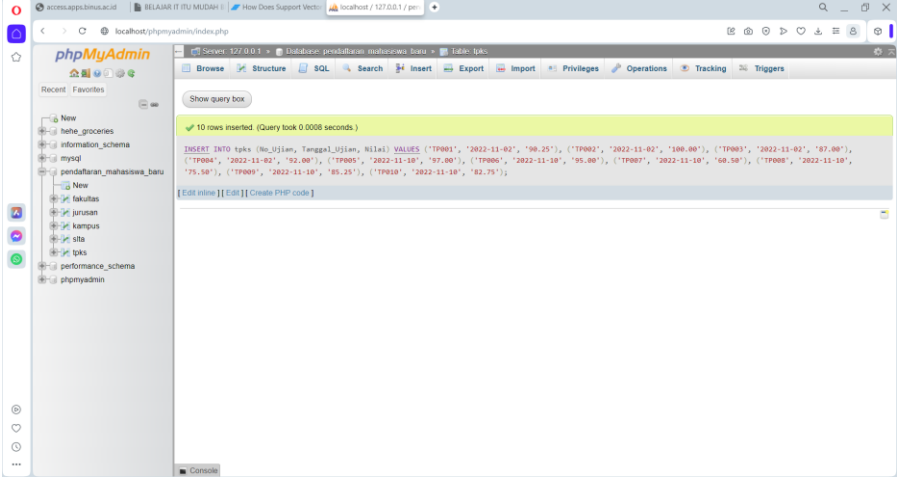
<p>Membuat Tabel Dokumen_KK</p>	<pre>CREATE TABLE DOKUMEN_KK (Dokumen_KK_ID VARCHAR(5) PRIMARY KEY CHECK(Dokumen_KK_ID REGEXP 'KK[0-9][0-9][0-9]'), Tanggal_upload DATE, Pendaftaran_ID VARCHAR(5), FOREIGN KEY (Pendaftaran_ID) REFERENCES Pendaftaran(Pendaftaran_ID) ON UPDATE CASCADE ON DELETE CASCADE)</pre> 
<p>Membuat Tabel Dokumen_Ijazah</p>	<pre>CREATE TABLE DOKUMEN_Ijazah (Dokumen_Ijazah_ID VARCHAR(5) PRIMARY KEY CHECK(Dokumen_Ijazah_ID REGEXP 'IJ[0-9][0-9][0-9]'), Tanggal_upload DATE, Pendaftaran_ID VARCHAR(5), FOREIGN KEY (Pendaftaran_ID) REFERENCES Pendaftaran(Pendaftaran_ID) ON UPDATE CASCADE ON DELETE CASCADE)</pre>

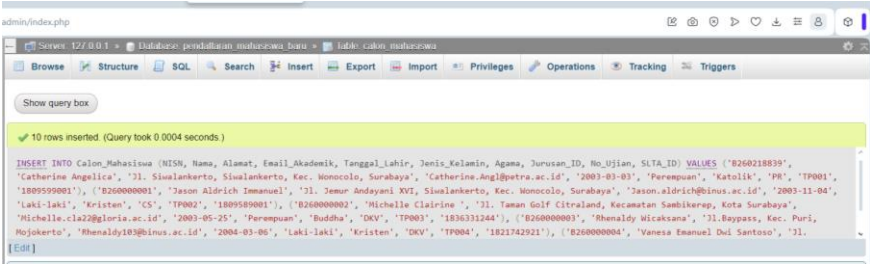
	
Memasukkan Dataset untuk tabel Kampus	<p>INSERT INTO Kampus (Kampus_ID, Nama_Kampus, Lokasi_kampus) VALUES</p> <p>(‘MLG’, ‘Binus Malang’, ‘Malang Jawa Timur’),</p> <p>(‘BDG’, ‘Binus Bandung’, ‘Bandung Jawa Barat’),</p> <p>(‘SMRNG’, ‘Binus Semarang’, ‘Semarang Jawa Tengah’),</p> <p>(‘BKS’, ‘Binus Bekasi’, ‘Bekasi Jawa Barat’),</p> <p>(‘KMG01’, ‘Binus Syahdan’, ‘Kemanggisan Jakarta Barat’),</p> <p>(‘KMG02’, ‘Binus Anggrek’, ‘Kemanggisan Jakarta Barat’),</p> <p>(‘KMG03’, ‘Binus Kijang’, ‘Kemanggisan Jakarta Barat’),</p> <p>(‘KMG04’, ‘Binus JWC’, ‘Jakarta Barat’),</p> <p>(‘ALSUT’, ‘Binus Alam Sutera’, ‘Tangerang ’),</p> <p>(‘SRPNG’, ‘Binus Serpong’, ‘Bogor Jawa Barat’);</p> 

<p>Memasukkan Dataset Untuk Tabel Fakultas</p>	<p>INSERT INTO Fakultas (FakultasID, Nama_Fakultas, KampusID) VALUES</p> <p>(‘SOCS’, ‘School Of Computer Science’, ‘MLG’),</p> <p>(‘SOA’, ‘School Of Accounting’, ‘SMRNG’),</p> <p>(‘FOH’, ‘Faculty Of Humanities’, ‘BDG’),</p> <p>(‘FOE’, ‘Faculty Of Engginering’, ‘KMG02’),</p> <p>(‘BSS’, ‘Binus Bussiness School’, ‘SRPNG’),</p> <p>(‘FODCAHAT’, ‘Faculty Of Digital Communication And Hotel And Tourism’, ‘KMG03’),</p> <p>(‘SOIS’, ‘School Of Information System’, ‘KMG01’),</p> <p>(‘BOL’, ‘Binus Online Learning’, ‘KMG04’),</p> <p>(‘SOACA’, ‘School Of Computing And Creative Arts’, ‘ALSUT’),</p> <p>(‘FOFC’, ‘Faculty Of Food Culinary’, ‘BKS’);</p> 
<p>Memasukkan Dataset Untuk Tabel Jurusan</p>	<p>INSERT INTO jurusan (JurusanID, Nama_Jurusan, FakultasID) VALUES</p> <p>(‘CS’, ‘Computer Science’, ‘SOCS’),</p> <p>(‘DI’, ‘Design Interior’, ‘SOACA’),</p> <p>(‘EBC’, ‘Entrepreneurship Bussiness Creation’, ‘BSS’),</p> <p>(‘DKV’, ‘Design Komunikasi Visual’, ‘FOH’),</p> <p>(‘PR’, ‘Public Relation’, ‘SOACA’),</p> <p>(‘SI’, ‘System Information’, ‘SOIS’),</p>

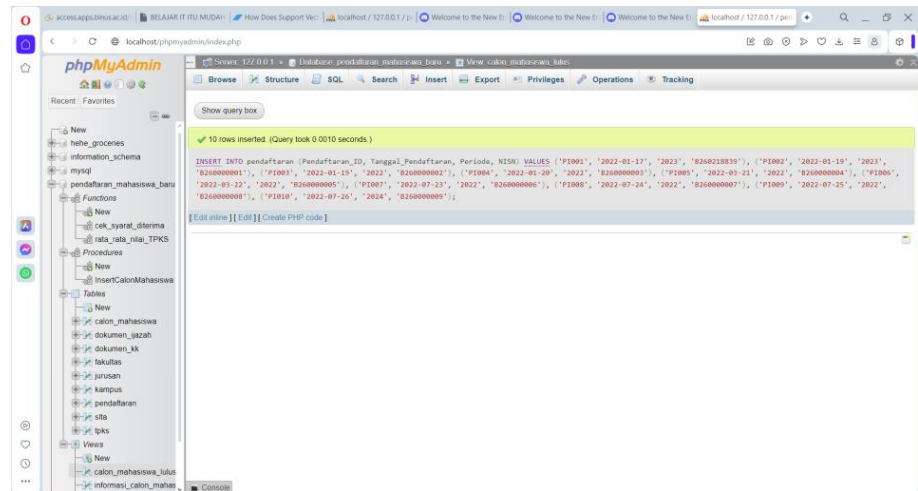
	<p>(‘DS’, ‘Data Science’, ‘SOCS’), (‘CSN’, ‘Cyber Security’, ‘SOCS’), (‘BA’, ‘Bussiness Analytic’, ‘BSS’), (‘HM’, ‘Hotel Management’, ‘FODCAHAT’);</p> 
<p>Memasukkan Dataset Untuk Tabel SLTA</p>	<pre>INSERT INTO SLTA (SLTA_ID, Nama_Sekolah_asal, Tahun_lulus, Nilai_Rata_rata_Rapor) VALUES ('1809599001', 'SMA Kristen Petra 1 Surabaya', '2022', '80.75'), ('1809589001', 'SMA Kristen Gloria 1 Surabaya', '2022', '90.00'), ('1836331244', 'SMK Katolik Santo Yusuf 4 Sidoarjo', '2022', '99.00'), ('1821742921', 'SMA Taruna Nusa Harapan Mojokerto', '2022', '82.25'), ('1800000001', 'SMA Kristen Gloria 1 Surabaya', '2022', '92.50'), ('1800000002', 'SMA Negeri 1 Puri Mojokerto', '2019', '77.50'), ('1800000003', 'SMA Kristen Petra 1 Surabaya', '2022', '89.75'), ('1800000004', 'SMA Taruna Nusa Harapan Mojokerto', '2022', '90.00'),</pre>

	<p>(‘1800233233’, ‘SMA Taruna Nusa Harapan Mojokerto’, ‘2022’, ‘95.00’),</p> <p>(‘1805555533’, ‘SMA Negeri 2 Surabaya’, ‘2020’, ‘88.75’);</p> 
Memasukkan dataset untuk tabel TPKS	<p>INSERT INTO tpks (No_Ujian, Tanggal_Ujian, Nilai) VALUES</p> <p>(‘TP001’, ‘2022-11-02’, ‘90.25’),</p> <p>(‘TP002’, ‘2022-11-02’, ‘100.00’),</p> <p>(‘TP003’, ‘2022-11-02’, ‘87.00’),</p> <p>(‘TP004’, ‘2022-11-02’, ‘92.00’),</p> <p>(‘TP005’, ‘2022-11-10’, ‘97.00’),</p> <p>(‘TP006’, ‘2022-11-10’, ‘95.00’),</p> <p>(‘TP007’, ‘2022-11-10’, ‘60.50’),</p> <p>(‘TP008’, ‘2022-11-10’, ‘75.50’),</p> <p>(‘TP009’, ‘2022-11-10’, ‘85.25’),</p> <p>(‘TP010’, ‘2022-11-10’, ‘82.75’);</p>

	
<p>Memasukkan Dataset Calon Mahasiswa</p>	<p>INSERT INTO Calon_Mahasiswa (NISN, Nama, Alamat, Email_Akademik, Tanggal_Lahir, Jenis_Kelamin, Agama, Jurusan_ID, No_Ujian, SLTA_ID) VALUES</p> <p>(‘B260218839’, ‘Catherine Angelica’, ‘Jl. Siwalankerto, Siwalankerto, Kec. Wonocolo, Surabaya’, ‘Catherine.Angl@petra.ac.id’, ‘2003-03-03’, ‘Perempuan’, ‘Katolik’, ‘PR’, ‘TP001’, ‘1809599001’),</p> <p>(‘B260000001’, ‘Jason Aldrich Immanuel’, ‘Jl. Jemur Andayani XVI, Siwalankerto, Kec. Wonocolo, Surabaya’, ‘Jason.aldrich@binus.ac.id’, ‘2003-11-04’, ‘Laki-laki’, ‘Kristen’, ‘CS’, ‘TP002’, ‘1809589001’),</p> <p>(‘B260000002’, ‘Michelle Clairine’, ‘Jl. Taman Golf Citraland, Kecamatan Sambikerep, Kota Surabaya’, ‘Michelle.cla22@gloria.ac.id’, ‘2003-05-25’, ‘Perempuan’, ‘Buddha’, ‘DKV’, ‘TP003’, ‘1836331244’),</p> <p>(‘B260000003’, ‘Rhenaldy Wicaksana’, ‘Jl.Baypass, Kec. Puri, Mojokerto’, ‘Rhenaldy103@binus.ac.id’, ‘2004-03-06’, ‘Laki-laki’, ‘Kristen’, ‘DKV’, ‘TP004’, ‘1821742921’),</p> <p>(‘B260000004’, ‘Vanessa Emanuel Dwi Santoso’, ‘Jl. Rafless Blvd Citraland, Kecamatan Sambikerep, Kota Surabaya’, ‘Vanesss134@tnh.ac.id’, ‘2005-09-17’, ‘Perempuan’, ‘Buddha’, ‘DI’, ‘TP005’, ‘1800000001’),</p>

	<p>(‘B260000005’, ‘Ricky Jonathan’, ‘Jl. Pahlawan, Kec. Margersari, Mojokerto’, ‘Ricky.Jonathan@binus.ac.id’, ‘2003-03-08’, ‘Laki-laki’, ‘Kristen’, ‘SI’, ‘TP006’, ‘1800000002’),</p> <p>(‘B260000006’, ‘Wilson Setiawan’, ‘Jl. W.R.Supratman, Mojokerto’, ‘WilsonSetiawan@binus.ac.id’, ‘2003-09-09’, ‘Laki-laki’, ‘Kristen’, ‘CS’, ‘TP007’, ‘1800000003’),</p> <p>(‘B260000007’, ‘Margareth Nyoto Wijaya’, ‘Jl. Pisang No 12 Kec. Margersari, Mojokerto’, ‘Bimbing@binus.ac.id’, ‘2003-10-10’, ‘Perempuan’, ‘Kristen’, ‘EBC’, ‘TP008’, ‘1800000004’),</p> <p>(‘B260000008’, ‘Galih Rawangga’, ‘Gg. IV, Lidah Kulon, Kec. Lakarsantri, Surabaya,’ ‘Galih.rawangga@binus.ac.id’, ‘2003-01-11’, ‘Laki-laki’, ‘Islam’, ‘CS’, ‘TP009’, ‘1805555533’),</p> <p>(‘B260000009’, ‘Evelyn Wahyu Callista’, ‘Jl. Apel No 9 Kec. Margersari, Mojokerto’, ‘Epelinw134@tnh.ac.id’, ‘2003-12-12’, ‘Perempuan’, ‘Katolik’, ‘EBC’, ‘TP010’, ‘1800233233’);</p> 
Memasukan Dataset untuk Tabel Pendaftaran	<p>INSERT INTO pendaftaran (Pendaftaran_ID, Tanggal_Pendaftaran, Periode, NISN) VALUES</p> <p>(‘PI001’, ‘2022-01-17’, ‘2023’, ‘B260218839’),</p> <p>(‘PI002’, ‘2022-01-19’, ‘2023’, ‘B260000001’),</p> <p>(‘PI003’, ‘2022-01-19’, ‘2022’, ‘B260000002’),</p> <p>(‘PI004’, ‘2022-01-20’, ‘2022’, ‘B260000003’),</p> <p>(‘PI005’, ‘2022-03-21’, ‘2022’, ‘B260000004’),</p> <p>(‘PI006’, ‘2022-03-22’, ‘2022’, ‘B260000005’),</p> <p>(‘PI007’, ‘2022-07-23’, ‘2022’, ‘B260000006’),</p> <p>(‘PI008’, ‘2022-07-24’, ‘2022’, ‘B260000007’),</p> <p>(‘PI009’, ‘2022-07-25’, ‘2022’, ‘B260000008’),</p>

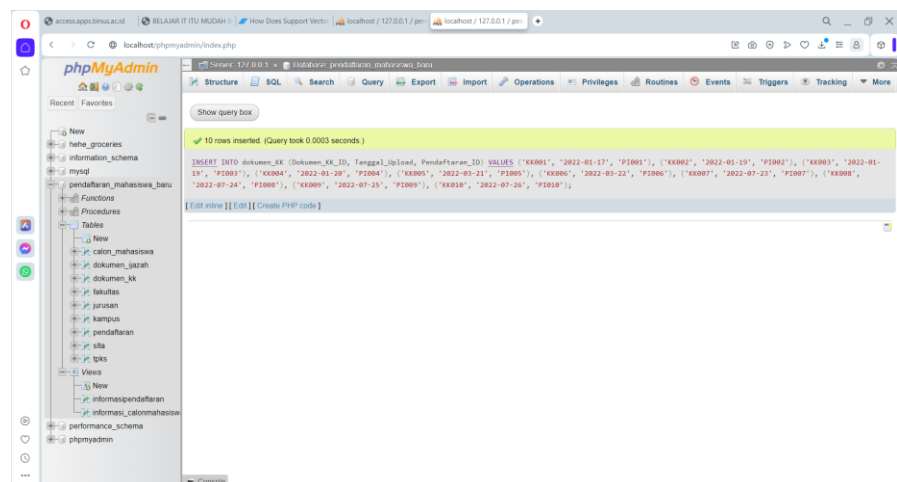
(‘PI010’, ‘2022-07-26’, ‘2024’, ‘B260000009’);

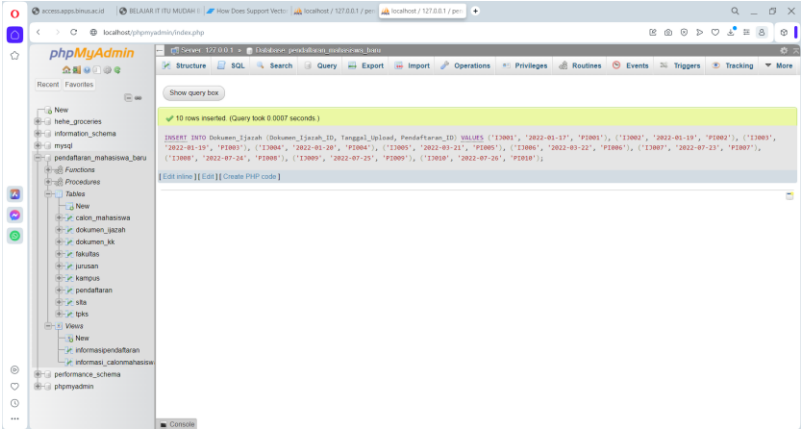


Memasukan
Dataset untuk
Tabel Dokumen
KK

INSERT INTO dokumen_KK (Dokumen_KK_ID, Tanggal_Upload,
Pendaftaran_ID) VALUES

(‘KK001’, ‘2022-01-17’, ‘PI001’),
(‘KK002’, ‘2022-01-19’, ‘PI002’),
(‘KK003’, ‘2022-01-19’, ‘PI003’),
(‘KK004’, ‘2022-01-20’, ‘PI004’),
(‘KK005’, ‘2022-03-21’, ‘PI005’),
(‘KK006’, ‘2022-03-22’, ‘PI006’),
(‘KK007’, ‘2022-07-23’, ‘PI007’),
(‘KK008’, ‘2022-07-24’, ‘PI008’),
(‘KK009’, ‘2022-07-25’, ‘PI009’),
(‘KK010’, ‘2022-07-26’, ‘PI010’);



Memasukan Dataset untuk Tabel Dokumen Ijazah	<p>INSERT INTO Dokumen_Ijazah (Dokumen_Ijazah_ID, Tanggal_Upload, Pendaftaran_ID) VALUES</p> <p>(‘IJ001’, ‘2022-01-17’, ‘PI001’),</p> <p>(‘IJ002’, ‘2022-01-19’, ‘PI002’),</p> <p>(‘IJ003’, ‘2022-01-19’, ‘PI003’),</p> <p>(‘IJ004’, ‘2022-01-20’, ‘PI004’),</p> <p>(‘IJ005’, ‘2022-03-21’, ‘PI005’),</p> <p>(‘IJ006’, ‘2022-03-22’, ‘PI006’),</p> <p>(‘IJ007’, ‘2022-07-23’, ‘PI007’),</p> <p>(‘IJ008’, ‘2022-07-24’, ‘PI008’),</p> <p>(‘IJ009’, ‘2022-07-25’, ‘PI009’),</p> <p>(‘IJ010’, ‘2022-07-26’, ‘PI010’);</p> 
----------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

- **Membuat Procedure, Function dan view**

Function Untuk mengecek syarat diterima	<p>DELIMITER \$\$</p> <p>CREATE FUNCTION check_status_diterima(tpks_nilai DECIMAL(5,2), slta_tahun_lulus YEAR, pendaftaran_periode YEAR)</p> <p>RETURNS VARCHAR(50)</p> <p>BEGIN</p> <p>DECLARE result VARCHAR(50);</p>
-----------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

```

IF tpks_nilai >= 85 AND slta_tahun_lulus >=
pendaftaran_periode - 2 THEN
    SET result = 'DITERIMA';
ELSE
    SET result = 'DITOLAK';
END IF;

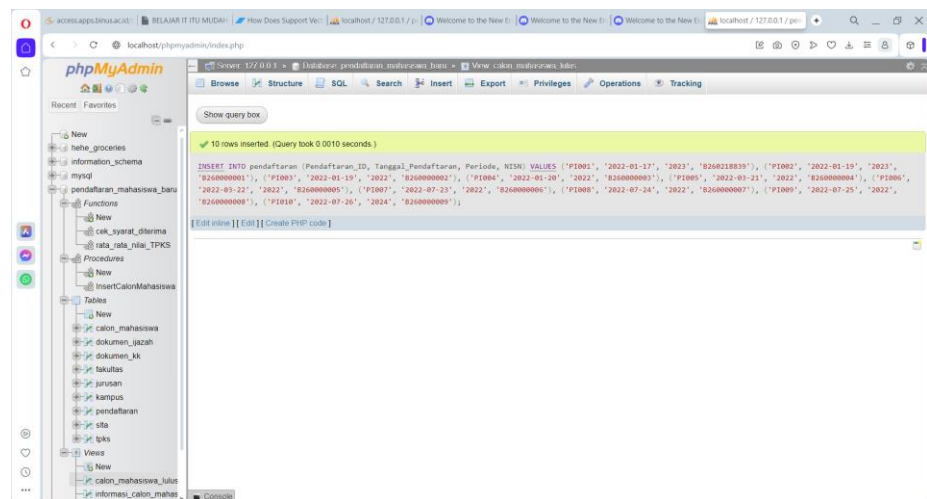
RETURN result;

END$$

```

DELIMITER ;

DELIMITER ;

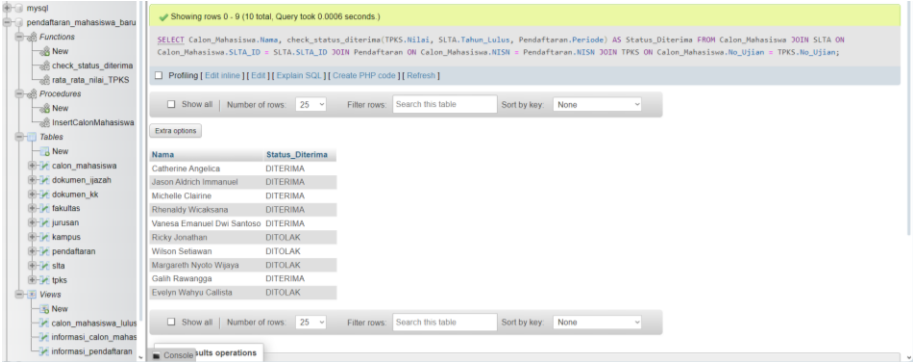


(untuk memanggil fungsi :)

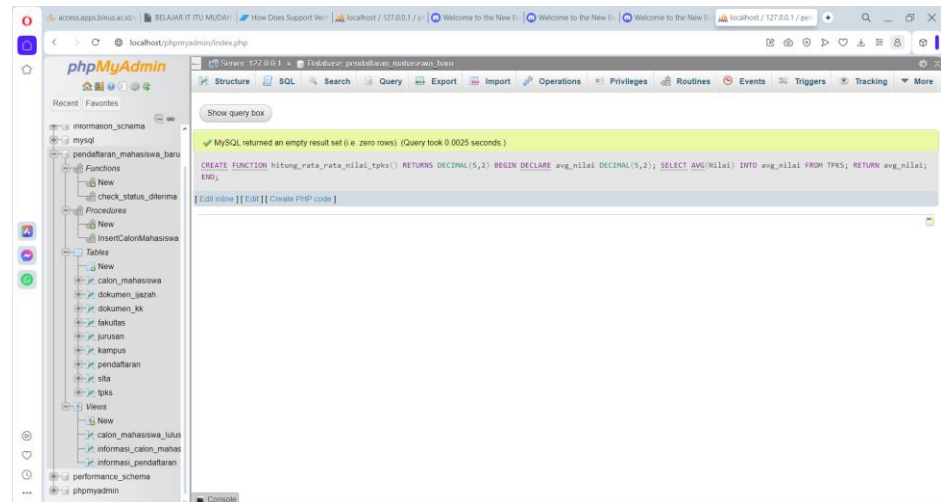
```

SELECT Calon_Mahasiswa>Nama,
        check_status_diterima(TPKS.Nilai, SLTA.Tahun_Lulus,
Pendaftaran.Periode) AS Status_Diterima
FROM Calon_Mahasiswa
JOIN SLTA ON Calon_Mahasiswa.SLTA_ID = SLTA.SLTA_ID
JOIN Pendaftaran ON Calon_Mahasiswa.NISN =
Pendaftaran.NISN

```

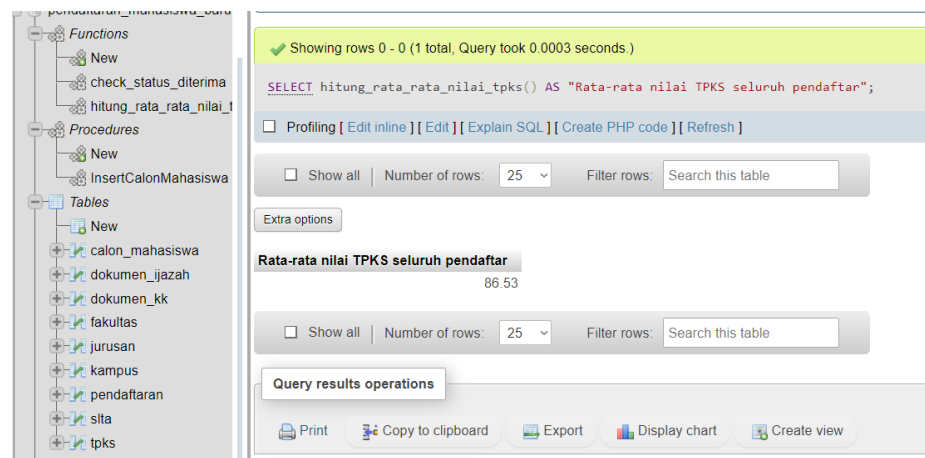
	<div>JOIN TPKS ON Calon_Mahasiswa.No_Ujian = TPKS.No_Ujian;</div> <div><p>The screenshot shows the MySQL Workbench interface. On the left is the 'Schemas' tree with 'mysql' selected. The main area displays a SQL query: <code>SELECT Calon_Mahasiswa>Nama, check_status_diterima(TPKS.Nilai, SLTA.Tahun_Lulus, Pendaftaran.Periode) AS Status_Diterima FROM Calon_Mahasiswa JOIN SLTA ON Calon_Mahasiswa.SLTA_ID = SLTA.SLTA_ID JOIN Pendaftaran ON Calon_Mahasiswa.NISN = Pendaftaran.NISN JOIN TPKS ON Calon_Mahasiswa.No_Ujian = TPKS.No_Ujian;</code>. Below the query, a table of results is shown with columns 'Nama' and 'Status_Diterima'. The results list 10 students and their status (DITERIMA or DITOLAK).</p><table><thead><tr><th>Nama</th><th>Status_Diterima</th></tr></thead><tbody><tr><td>Catherine Angelica</td><td>DITERIMA</td></tr><tr><td>Jason Aldrich Immanuel</td><td>DITERIMA</td></tr><tr><td>Michelle Ciarine</td><td>DITERIMA</td></tr><tr><td>Shenaisy Wicakarna</td><td>DITERIMA</td></tr><tr><td>Vanessa Emanuel Dwi Santoso</td><td>DITERIMA</td></tr><tr><td>Ricky Jonathan</td><td>DITOLAK</td></tr><tr><td>Wilson Setiawan</td><td>DITOLAK</td></tr><tr><td>Margareth Nycto Wijaya</td><td>DITOLAK</td></tr><tr><td>Gaith Rawangga</td><td>DITERIMA</td></tr><tr><td>Evelyn Wahyuni Cahida</td><td>DITOLAK</td></tr></tbody></table></div>	Nama	Status_Diterima	Catherine Angelica	DITERIMA	Jason Aldrich Immanuel	DITERIMA	Michelle Ciarine	DITERIMA	Shenaisy Wicakarna	DITERIMA	Vanessa Emanuel Dwi Santoso	DITERIMA	Ricky Jonathan	DITOLAK	Wilson Setiawan	DITOLAK	Margareth Nycto Wijaya	DITOLAK	Gaith Rawangga	DITERIMA	Evelyn Wahyuni Cahida	DITOLAK
Nama	Status_Diterima																						
Catherine Angelica	DITERIMA																						
Jason Aldrich Immanuel	DITERIMA																						
Michelle Ciarine	DITERIMA																						
Shenaisy Wicakarna	DITERIMA																						
Vanessa Emanuel Dwi Santoso	DITERIMA																						
Ricky Jonathan	DITOLAK																						
Wilson Setiawan	DITOLAK																						
Margareth Nycto Wijaya	DITOLAK																						
Gaith Rawangga	DITERIMA																						
Evelyn Wahyuni Cahida	DITOLAK																						
<div>FUNCTION Hitung Rata- rata nilai TPKS dari semua calon mahasiswa</div>	<div>DELIMITER //</div> <div>CREATE FUNCTION hitung_rata_rata_nilai_tpks() RETURNS DECIMAL(5,2) BEGIN DECLARE avg_nilai DECIMAL(5,2); SELECT AVG(Nilai) INTO avg_nilai FROM TPKS; RETURN avg_nilai; END//</div>																						

DELIMITER ;



(untuk memanggil fungsi :)

SELECT hitung_rata_rata_nilai_tpks() AS "Rata-rata nilai TPKS seluruh pendaftar";



FUNCTION :
Menghitung
rata-rata usia
calon
pendaftar

DELIMITER //

```
CREATE FUNCTION hitung_rata_rata_usia_pendaftar()  
RETURNS INT  
BEGIN  
    DECLARE avg_usia INT;
```

```

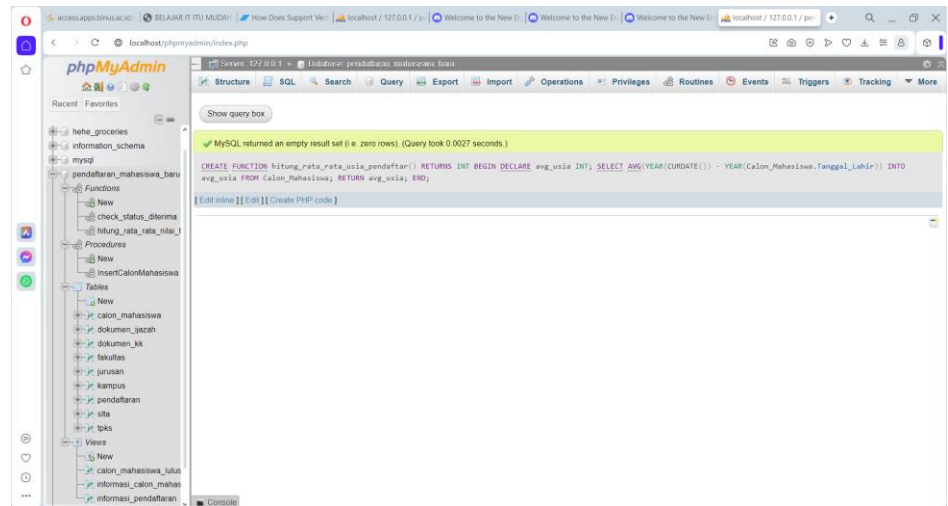
SELECT AVG(YEAR(CURDATE()) -
YEAR(Calon_Mahasiswa.Tanggal_Lahir)) INTO avg_usia
FROM Calon_Mahasiswa;

RETURN avg_usia;

END//

DELIMITER ;

```

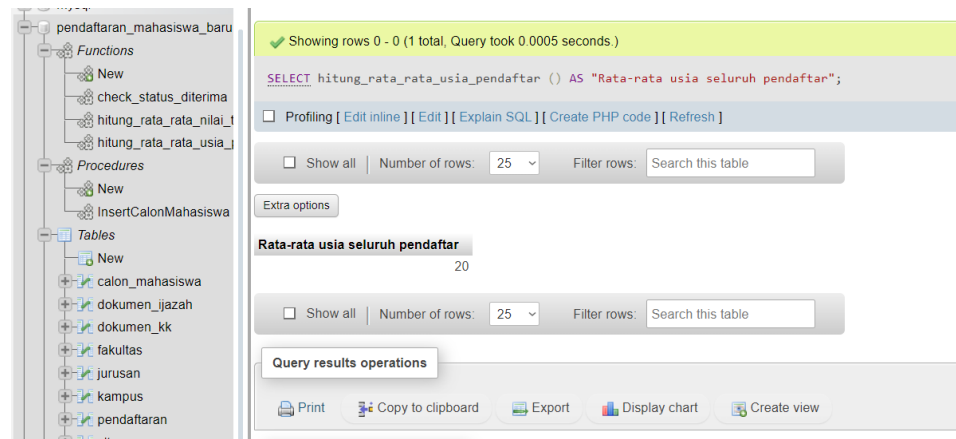


(untuk memanggil fungsi :)

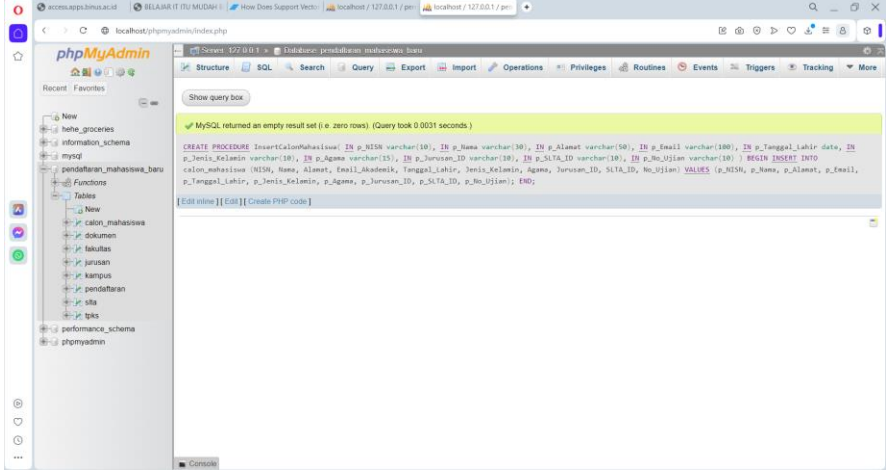
```

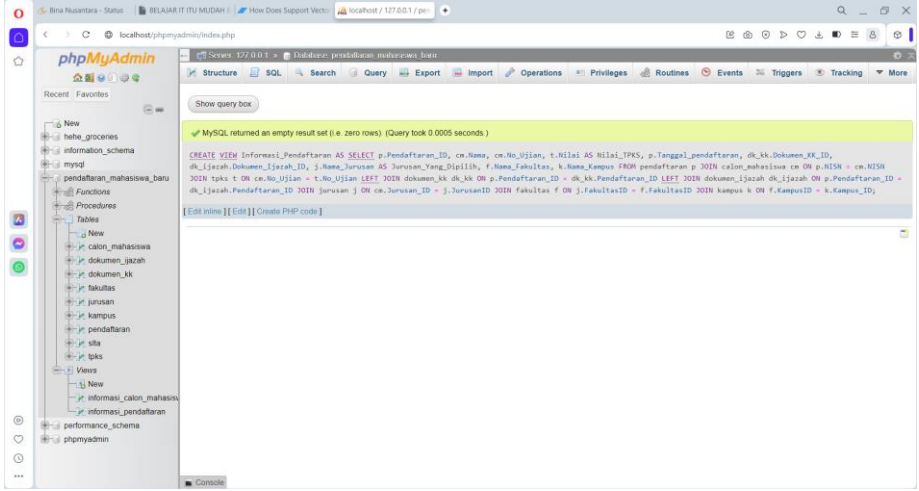
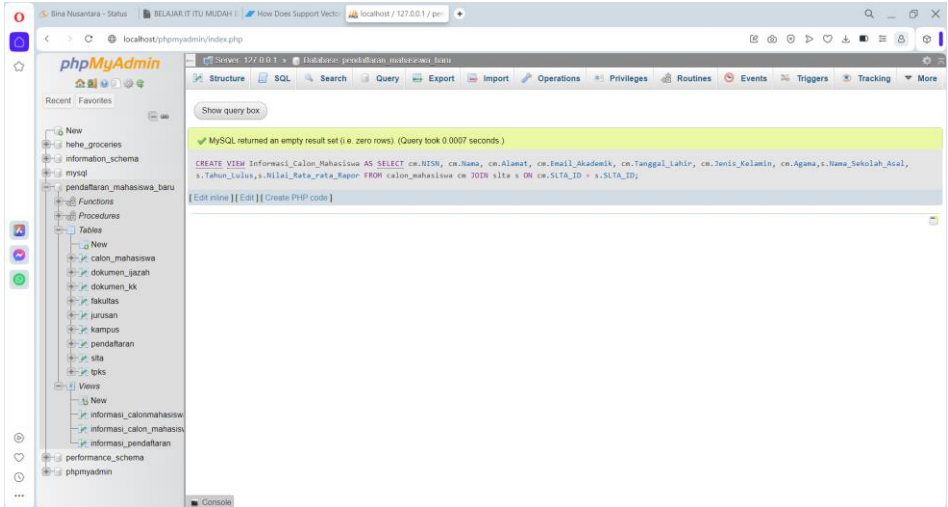
SELECT hitung_rata_rata_usia_pendaftar () AS "Rata-rata usia seluruh
pendaftar";

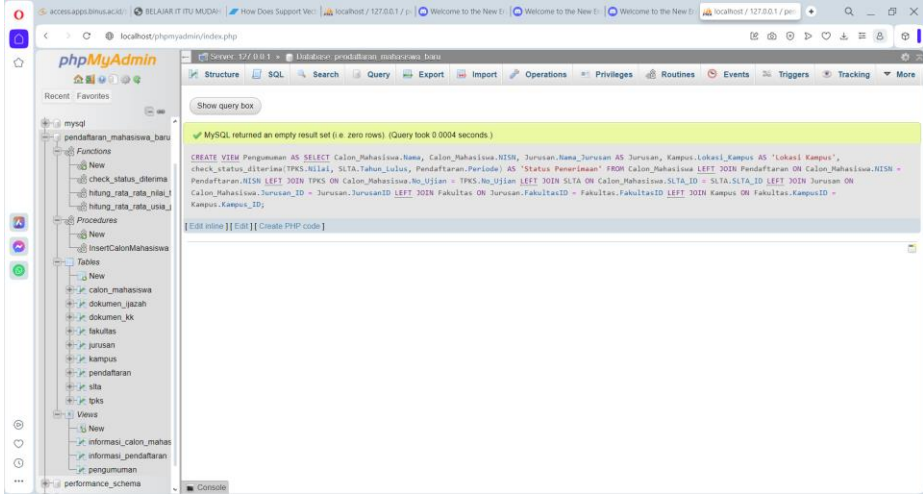
```



<p>PROCEDURE</p> <p>Insert_Calon_Mahasiswa</p>	<p>DELIMITER //</p> <p>CREATE PROCEDURE InsertCalonMahasiswa(IN p_NISN varchar(10), IN p_Nama varchar(30), IN p_Alamat varchar(50), IN p_Email varchar(100), IN p_Tanggal_Lahir date, IN p_Jenis_Kelamin varchar(10), IN p_Agama varchar(15), IN p_Jurusan_ID varchar(10), IN p_SLTA_ID varchar(10), IN p_No_Ujian varchar(10)) BEGIN INSERT INTO calon_mahasiswa (NISN, Nama, Alamat, Email_Akademik, Tanggal_Lahir, Jenis_Kelamin, Agama, Jurusan_ID, SLTA_ID, No_Ujian) VALUES (p_NISN, p_Nama, p_Alamat, p_Email, p_Tanggal_Lahir, p_Jenis_Kelamin, p_Agama, p_Jurusan_ID, p_SLTA_ID, p_No_Ujian); END //</p> <p>DELIMITER ;</p>
------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	
<p>View 1 :</p> <p>Informasi</p> <p>Pendaftaran</p>	<pre> CREATE VIEW Informasi_Pendaftaran AS SELECT p.Pendaftaran_ID, cm.Nama, cm.No_Ujian, t.Nilai AS Nilai_TPKS, p.Tanggal_pendaftaran, p.Periode, dk_kk.Dokumen_KK_ID, dk_ijazah.Dokumen_Ijazah_ID, j.Nama_Jurusan, f.Nama_Fakultas, k.Nama_Kampus FROM pendaftaran p JOIN calon_mahasiswa cm ON p.NISN = cm.NISN JOIN tpks t ON cm.No_Ujian = t.No_Ujian LEFT JOIN dokumen_kk dk_kk ON p.Pendaftaran_ID = dk_kk.Pendaftaran_ID LEFT JOIN dokumen_ijazah dk_ijazah ON p.Pendaftaran_ID = dk_ijazah.Pendaftaran_ID JOIN jurusan j ON cm.Jurusan_ID = j.JurusanID JOIN fakultas f ON j.FakultasID = f.FakultasID </pre>

	<p>JOIN kampus k ON f.KampusID = k.Kampus_ID;</p>  <p>The screenshot shows the phpMyAdmin interface with a SQL query entered in the query box. The query is a JOIN operation between a table named 'f' and a table named 'k' on the condition 'f.KampusID = k.Kampus_ID'. The query has been executed, and the result is an empty set (0 rows).</p>
<p>View 2 :</p> <p>Informasi Calon Mahasiswa</p>	<p>CREATE VIEW Informasi_Calon_Mahasiswa AS</p> <p>SELECT cm.NISN, cm>Nama, cm.Alatmat, cm.Email_Akademik, cm.Tanggal_Lahir, cm.Jenis_Kelamin, cm.Agama,s>Nama_Sekolah_Asal, s.Tahun_Lulus,s.Nilai_Rata_rata_Rapor</p> <p>FROM calon_mahasiswa cm</p> <p>JOIN slta s ON cm.SLTA_ID = s.SLTA_ID</p>  <p>The screenshot shows the phpMyAdmin interface with a SQL query entered in the query box. The query is a CREATE VIEW statement for a view named 'Informasi_Calon_Mahasiswa'. The view is created by selecting columns from the 'calon_mahasiswa' table (cm) and the 'slta' table (s) on the condition 'cm.SLTA_ID = s.SLTA_ID'. The query has been executed, and the result is an empty set (0 rows).</p>
<p>View 3 :</p> <p>Informasi pengumuman penerimaan</p>	<p>CREATE VIEW Pengumuman AS</p> <p>SELECT</p> <p>Calon_Mahasiswa>Nama,</p> <p>Calon_Mahasiswa.NISN,</p>

<p>mahasiswa baru</p>	<pre> Jurusan>Nama_Jurusan AS Jurusan, Kampus.Lokasi_Kampus AS 'Lokasi Kampus', check_status_diterima(TPKS.Nilai, SLTA.Tahun_Lulus, Pendaftaran.Periode) AS 'Status Penerimaan' FROM Calon_Mahasiswa LEFT JOIN Pendaftaran ON Calon_Mahasiswa.NISN = Pendaftaran.NISN LEFT JOIN TPKS ON Calon_Mahasiswa.No_Ujian = TPKS.No_Ujian LEFT JOIN SLTA ON Calon_Mahasiswa.SLTA_ID = SLTA.SLTA_ID LEFT JOIN Jurusan ON Calon_Mahasiswa.Jurusan_ID = Jurusan.JurusanID LEFT JOIN Fakultas ON Jurusan.FakultasID = Fakultas.FakultasID LEFT JOIN Kampus ON Fakultas.KampusID = Kampus.Kampus_ID; </pre> 
---------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

- Pertanyaan dan jawaban saat presentasi**

<p>Pertanyaan (Kelompok)</p>	<p>Membuat view yang menampilkan informasi berdasarkan jurusan mulai dari rata-rata nilai tpks (dibulatkan) calon mahasiswa yang mendaftar, jumlah mahasiswa, dan nilai tpks tertinggi serta terendah (dibulatkan).</p>
<p>QUERY & Penjelasan</p>	<pre> CREATE VIEW Informasi_Jurusan AS SELECT j>Nama_Jurusan, </pre>

```

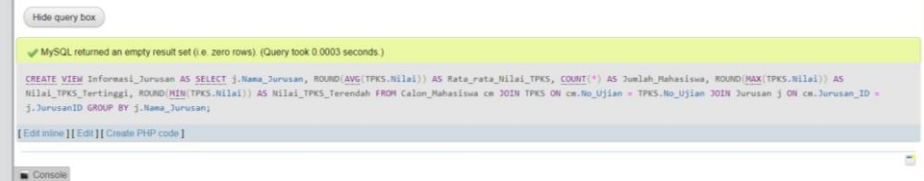
ROUND(AVG(TPKS.Nilai)) AS Rata_rata_Nilai_TPKS,
COUNT(*) AS Jumlah_Mahasiswa,
ROUND(MAX(TPKS.Nilai)) AS Nilai_TPKS_Tertinggi,
ROUND(MIN(TPKS.Nilai)) AS Nilai_TPKS_Terendah
FROM Calon_Mahasiswa cm
JOIN TPKS ON cm.No_Ujian = TPKS.No_Ujian
JOIN Jurusan j ON cm.Jurusan_ID = j.JurusanID
GROUP BY j>Nama_Jurusan;

```

PENJELASAN QUERY :

- CREATE VIEW Informasi_Jurusan AS → Membuat view dengan diberi nama Informasi_Jurusan.
- SELECT → untuk mengambil data dari atribut yang akan diambil/ ditampilkan dalam view
- ROUND(AVG(TPKS.Nilai)) AS Rata_rata_Nilai_TPKS → Round untuk pembulatan, AVG untuk menghitung rata-rata dari kolom nilai dari tabel TPKS
- COUNT(*) → untuk menghitung jumlah semua siswa
- Kemudian MAX dan MIN untuk mendapatkan nilai tertinggi dan nilai terendah dari kolom Nilai dalam tabel TPKS dan hasilnya dibulatkan (ROUND)
- FROM Calon_Mahasiswa cm JOIN TPKS ON cm.No_Ujian = TPKS.No_Ujian JOIN Jurusan j ON cm.Jurusan_ID = j.JurusanID → Menggabungkan tabel Calon_Mahasiswa, TPKS, dan Jurusan berdasarkan relasi yang sesuai (Primary Key dan Foreign Key masing-masing tabel).
- GROUP BY j>Nama_Jurusan → Mengelompokkan hasil berdasarkan Nama Jurusan

Screenshot



Hide query box

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0003 seconds.)

```

CREATE VIEW Informasi_Jurusan AS SELECT j>Nama_Jurusan, ROUND(AVG(TPKS.Nilai)) AS Rata_rata_Nilai_TPKS, COUNT(*) AS Jumlah_Mahasiswa, ROUND(MAX(TPKS.Nilai)) AS Nilai_TPKS_Tertinggi, ROUND(MIN(TPKS.Nilai)) AS Nilai_TPKS_Terendah FROM Calon_Mahasiswa cm JOIN TPKS ON cm.No_Ujian = TPKS.No_Ujian JOIN Jurusan j ON cm.Jurusan_ID = j.JurusanID GROUP BY j>Nama_Jurusan;

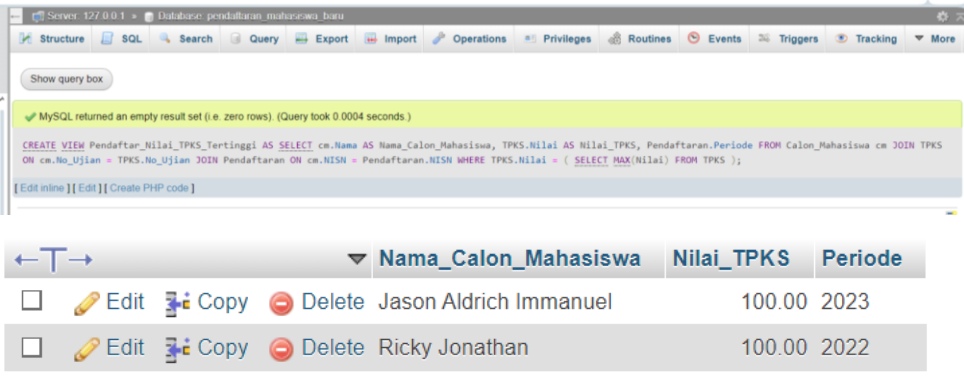
```

[Edit inline] [Edit] [Create PHP code]

Console

Extra options

Nama_Jurusan	Rata_rata_Nilai_TPKS	Jumlah_Mahasiswa	Nilai_TPKS_Tertinggi	Nilai_TPKS_Terendah
Computer Science	82	3	100	61
Design Interior	97	1	97	97
Design Komunikasi Visual	90	2	92	87
Entrepreneurship Bussiness Creation	79	2	83	76
Public Relation	90	1	90	90
System Information	100	1	100	100

Pertanyaan 1 (Individu)	Membuat view yang menampilkan nama pendaftar dengan nilai tpks tertinggi, beserta nilainya dan periode pendaftaran.
QUERY & penjelasan	<div>CREATE VIEW Pendaftar_Nilai_TPKS_Tertinggi AS SELECT cm>Nama AS Nama_Calon_Mahasiswa, TPKS.Nilai AS Nilai_TPKS, Pendaftaran.Periode AS Periode_Pendaftaran FROM Calon_Mahasiswa cm JOIN TPKS ON cm.No_Ujian = TPKS.No_Ujian JOIN Pendaftaran ON cm.NISN = Pendaftaran.NISN WHERE TPKS.Nilai = (SELECT MAX(Nilai) FROM TPKS)</div> <div>Penjelasan Query :<ul style="list-style-type: none">• CREATE VIEW Pendaftar_Nilai_TPKS_Tertinggi AS → untuk membuat view baru.• SELECT cm>Nama AS Nama_Calon_Mahasiswa, TPKS.Nilai AS Nilai_TPKS, Pendaftaran.Periode AS Periode_Pendaftaran → Itu untuk memilih kolom/atribut nama dari tabel Calon_mahasiswa (cm), Nilai dari tabel TPKS, dan Periode dari tabel pendaftaran. → AS itu untuk memberikan nama lain untuk kolom yang dipilih• FROM dan JOIN → Digunakan untuk menghubungkan tabel yang sesuai berdasarkan relasi (Primary Key dan Foreign Key).• WHERE TPKS.Nilai = (SELECT MAX(Nilai) FROM TPKS) → Untuk hanya memilih baris dengan nilai TPKS yang paling tinggi (tertinggi) yang ditemukan dalam tabel TPKS.</div>
Screenshot	<div></div>

Pertanyaan 2 (Individu)	Membuat view yang menampilkan nama pendaftar dengan nilai tpks terendah, beserta nilainya dan periode pendaftaran.
QUERY & PENJELASAN	<pre>CREATE VIEW Pendaftar_Nilai_TPKS_Terendah AS SELECT cm.Nama, TPKS.Nilai AS Nilai_TPKS, Pendaftaran.Periode AS Periode_Pendaftaran FROM Calon_Mahasiswa cm JOIN TPKS ON cm.No_Ujian = TPKS.No_Ujian JOIN Pendaftaran ON cm.NISN = Pendaftaran.NISN WHERE TPKS.Nilai = (SELECT MIN(Nilai) FROM TPKS)</pre> <p>Penjelasan QUERY :</p> <ul style="list-style-type: none"> • CREATE VIEW Pendaftar_Nilai_TPKS_Terendah AS → untuk membuat view baru yang menampilkan para pendaftar yang memiliki nilai terendah. • SELECT cm.Nama, TPKS.Nilai AS Nilai_TPKS, Pendaftaran.Periode AS Periode_Pendaftaran → Itu untuk memilih kolom/atribut nama dari tabel Calon_mahasiswa (cm), Nilai dari tabel TPKS, dan Periode dari tabel pendaftaran. → AS itu untuk memberikan nama lain untuk kolom yang dipilih • FROM dan JOIN → Digunakan untuk menghubungkan tabel yang sesuai berdasarkan relasi (Primary Key dan Foreign Key). Jadi FROM tabel Calon_Mahasiswa akan di joinkan dengan tabel TPKS dengan relasi No_Ujian (FK di tabel Calon_Mahasiswa dan menjadi PK di tabel TPKS) begitu juga dengan mengabungkan From tabel Calon_Mahasiswa dengan tabel Pendaftaran melalui NISN. • WHERE TPKS.Nilai = (SELECT MIN(Nilai) FROM TPKS) → Untuk hanya memilih baris dengan nilai TPKS yang paling rendah (terendah) yang ditemukan dalam tabel TPKS.

Screenshot	
------------	------------------------------------------------------------------------------------

Pertanyaan 3 (Individu)	Membuat view yang menampilkan informasi tentang calon mahasiswa yang memiliki nilai tpks di atas rata-rata.
QUEY & Penjelasan	<p>CREATE VIEW Calon_Mahasiswa_Diatas_Rataan AS SELECT cm>Nama, TPKS.Nilai AS Nilai_TPKS, Pendaftaran.Periode AS Periode_Pendaftaran FROM Calon_Mahasiswa cm JOIN TPKS ON cm.No_Ujian = TPKS.No_Ujian JOIN Pendaftaran ON cm.NISN = Pendaftaran.NISN WHERE TPKS.Nilai > (SELECT AVG(Nilai) FROM TPKS)</p> <p>Penjelasan QUERY :</p> <ul style="list-style-type: none"> • CREATE VIEW Calon_Mahasiswa_Diatas_Rataan AS → untuk membuat view baru yang menampilkan informasi pendaftar yang memiliki nilai TPKS di atas rata-rata. • SELECT cm>Nama, TPKS.Nilai AS Nilai_TPKS, Pendaftaran.Periode AS Periode_Pendaftaran → Itu untuk memilih kolom/atribut nama dari tabel Calon_mahasiswa (cm), Nilai dari tabel TPKS, dan Periode dari tabel pendaftaran. → AS itu untuk memberikan nama lain untuk kolom yang dipilih

- FROM dan JOIN → Digunakan untuk menghubungkan tabel yang sesuai berdasarkan relasi (Primary Key dan Foreign Key). Jadi FROM tabel Calon_Mahasiswa akan di joinkan dengan tabel TPKS dengan relasi No_Ujian (FK di tabel Calon_Mahasiswa dan menjadi PK di tabel TPKS) begitu juga dengan mengabungkan From tabel Calon_Mahasiswa dengan tabel Pendaftaran melalui NISN.
- WHERE TPKS.Nilai = (SELECT AVG(Nilai) FROM TPKS) → Supaya hanya memilih baris dengan nilai TPKS yang diatas dari rata-rata dari seluruh nilai TPKS yang ditemukan dalam tabel TPKS.

ScreenShot

Server: 127.0.0.1 x Database: pendaftaran_mahasiswa_satu x View: pendaftaran_tpk_satu_bringg

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0004 seconds)

```
CREATE VIEW Calon_Mahasiswa_Diatas_Rataan AS SELECT cm.Nama, TPKS.Nilai AS Nilai_TPKS, Pendaftaran.Periode AS Periode_Pendaftaran FROM Calon_Mahasiswa cm JOIN TPKS ON cm.No_Ujian = TPKS.No_Ujian JOIN Pendaftaran ON cm.NISN = Pendaftaran.NISN WHERE TPKS.Nilai > (SELECT AVG(Nilai) FROM TPKS);
```

Extra options

				Nama	Nilai_TPKS	Periode_Pendaftaran
<input type="checkbox"/>	Edit	Copy	Delete	Catherine Angelica	90.25	2023
<input type="checkbox"/>	Edit	Copy	Delete	Jason Aldrich Immanuel	100.00	2023
<input type="checkbox"/>	Edit	Copy	Delete	Rhenaldy Wicaksana	92.00	2022
<input type="checkbox"/>	Edit	Copy	Delete	Vanesa Emanuel Dwi Santoso	97.00	2022
<input type="checkbox"/>	Edit	Copy	Delete	Ricky Jonathan	100.00	2022

Pertanyaan 4 (Individu) : Membuat view yang menampilkan informasi tentang calon mahasiswa yang memiliki nilai tpks di bawah rata-rata.	
Query	<pre>CREATE VIEW calon_Mahasiswa_Bawah_Rata_Rata AS SELECT cm.NISN, cm.Nama, cm.Alamat, cm.Email_Akademik, cm.Tanggal_Lahir, cm.Jenis_Kelamin, cm.Agama, s.Nama_Sekolah_Asal,</pre>

	<div>s.Tahun_Lulus,</div> <div>s.Nilai_Rata_rata_Rapor,</div> <div>t.Nilai AS Nilai_TPKS</div> <div>FROM calon_mahasiswa cm</div> <div>JOIN slta s ON cm.SLTA_ID = s.SLTA_ID</div> <div>JOIN tpks t ON cm.No_Ujian = t.No_Ujian</div> <div>WHERE t.Nilai < (hitung_rata_rata_nilai_tpks());</div>																																																							
Penjelasan	<div>1. CREATE VIEW calon_Mahasiswa_Bawah_Rata_Rata AS :</div> <div>Mendefinisikan pembuatan view dengan nama calon_Mahasiswa_Bawah_Rata_Rata.</div> <div>2. SELECT cm.NISN, cm>Nama, cm.Alat, cm.Email_Akademik, cm.Tanggal_Lahir, cm.Jenis_Kelamin, cm.Agama, s>Nama_Sekolah_Asal, s.Tahun_Lulus, s.Nilai_Rata_rata_Rapor, t.Nilai AS Nilai_TPKS :</div> <div>Menentukan kolom-kolom yang akan ditampilkan dalam view. Seperti Informasi calon mahasiswa, Informasi dari sekolah asal, dan Nilai TPKS yang diambil dari tabel tpks dengan alias t.</div> <div>3. WHERE t.Nilai < (hitung_rata_rata_nilai_tpks()):</div> <div>Untuk menampilkan view dengan Nilai dibawah rata-rata menggunakan fungsi hitung_rata_rata_nilai_tpks() yang digunakan untuk menghitung rata-rata nilai TPKS.</div>																																																							
Screenshot	<div><div><div>Struktur</div><div>SQL</div><div>Cari</div><div>Kueri</div><div>Ekspor</div><div>Impor</div><div>Operasi</div><div>Hak Akses</div><div>Routine</div><div>Event</div><div>Trigger</div><div>Pelacakan</div><div>Desainer</div></div><div>Tampilkan kotak kueri</div><div><div>MySQL memberikan hasil kosong (atau nol baris). (Pencarian dilakukan dalam 0.0007 detik.)</div><div>CREATE VIEW calon_Mahasiswa_Bawah_Rata_Rata AS SELECT cm.NISN, cm>Nama, cm.Alat, cm.Email_Akademik, cm.Tanggal_Lahir, cm.Jenis_Kelamin, cm.Agama, s>Nama_Sekolah_Asal, s.Tahun_Lulus, s.Nilai_Rata_rata_Rapor, t.Nilai AS Nilai_TPKS FROM calon_mahasiswa cm JOIN slta s ON cm.SLTA_ID = s.SLTA_ID JOIN tpks t ON cm.No_Ujian = t.No_Ujian WHERE t.Nilai < (hitung_rata_rata_nilai_tpks());</div><div>[Edit dikotak] [Ubah] [Buat kode PHP]</div></div><div>Hasil view :</div><table><thead><tr><th>NISN</th><th>Nama</th><th>Alamat</th><th>Email_Akademik</th><th>Tanggal_Lahir</th><th>Jenis_Kelamin</th><th>Agama</th><th>Nama_Sekolah_Asal</th><th>Tahun_Lulus</th><th>Nilai_Rata_rata_Rapor</th><th>Nilai_TPKS</th></tr></thead><tbody><tr><td>B260000006</td><td>Wilson Setiawan</td><td>Jl W R Supratman, Mojokerto</td><td>WilsonSetiawan@binus.ac.id</td><td>2003-09-09</td><td>Laki-laki</td><td>Kristen</td><td>SMA Kristen Petra 1 Surabaya</td><td>2022</td><td>89.75</td><td>60.50</td></tr><tr><td>B260000007</td><td>Margareth Nyoto Wijaya</td><td>Jl Pisang No 12 Kec. Margersari, Mojokerto</td><td>Bimbing@binus.ac.id</td><td>2003-10-10</td><td>Perempuan</td><td>Kristen</td><td>SMA Taruna Nusa Harapan Mojokerto</td><td>2022</td><td>90.00</td><td>75.50</td></tr><tr><td>B260000008</td><td>Galih Rawangga</td><td>Gg. IV, Lidah Kulon, Kec. Lakarsantri, Surabaya,</td><td>Galih.rawangga@binus.ac.id</td><td>2003-01-11</td><td>Laki-laki</td><td>Islam</td><td>SMA Negeri 2 Surabaya</td><td>2020</td><td>88.75</td><td>85.25</td></tr><tr><td>B260000009</td><td>Evelyn Wahyu Callista</td><td>Jl. Apel No 9 Kec. Margersari, Mojokerto</td><td>Epelinw134@tnh.ac.id</td><td>2003-12-12</td><td>Perempuan</td><td>Katolik</td><td>SMA Taruna Nusa Harapan Mojokerto</td><td>2022</td><td>95.00</td><td>82.75</td></tr></tbody></table></div>	NISN	Nama	Alamat	Email_Akademik	Tanggal_Lahir	Jenis_Kelamin	Agama	Nama_Sekolah_Asal	Tahun_Lulus	Nilai_Rata_rata_Rapor	Nilai_TPKS	B260000006	Wilson Setiawan	Jl W R Supratman, Mojokerto	WilsonSetiawan@binus.ac.id	2003-09-09	Laki-laki	Kristen	SMA Kristen Petra 1 Surabaya	2022	89.75	60.50	B260000007	Margareth Nyoto Wijaya	Jl Pisang No 12 Kec. Margersari, Mojokerto	Bimbing@binus.ac.id	2003-10-10	Perempuan	Kristen	SMA Taruna Nusa Harapan Mojokerto	2022	90.00	75.50	B260000008	Galih Rawangga	Gg. IV, Lidah Kulon, Kec. Lakarsantri, Surabaya,	Galih.rawangga@binus.ac.id	2003-01-11	Laki-laki	Islam	SMA Negeri 2 Surabaya	2020	88.75	85.25	B260000009	Evelyn Wahyu Callista	Jl. Apel No 9 Kec. Margersari, Mojokerto	Epelinw134@tnh.ac.id	2003-12-12	Perempuan	Katolik	SMA Taruna Nusa Harapan Mojokerto	2022	95.00	82.75
NISN	Nama	Alamat	Email_Akademik	Tanggal_Lahir	Jenis_Kelamin	Agama	Nama_Sekolah_Asal	Tahun_Lulus	Nilai_Rata_rata_Rapor	Nilai_TPKS																																														
B260000006	Wilson Setiawan	Jl W R Supratman, Mojokerto	WilsonSetiawan@binus.ac.id	2003-09-09	Laki-laki	Kristen	SMA Kristen Petra 1 Surabaya	2022	89.75	60.50																																														
B260000007	Margareth Nyoto Wijaya	Jl Pisang No 12 Kec. Margersari, Mojokerto	Bimbing@binus.ac.id	2003-10-10	Perempuan	Kristen	SMA Taruna Nusa Harapan Mojokerto	2022	90.00	75.50																																														
B260000008	Galih Rawangga	Gg. IV, Lidah Kulon, Kec. Lakarsantri, Surabaya,	Galih.rawangga@binus.ac.id	2003-01-11	Laki-laki	Islam	SMA Negeri 2 Surabaya	2020	88.75	85.25																																														
B260000009	Evelyn Wahyu Callista	Jl. Apel No 9 Kec. Margersari, Mojokerto	Epelinw134@tnh.ac.id	2003-12-12	Perempuan	Katolik	SMA Taruna Nusa Harapan Mojokerto	2022	95.00	82.75																																														

Pertanyaan 5 (Individu) : Membuat view yang menampilkan nama calon mahasiswa yang memiliki nilai tpks tertinggi dan nilai rata-rata rapor di atas 90.

Query	<pre> CREATE VIEW calon_Mahasiswa_Tertinggi_Dan_Rapor_Atas_90 AS SELECT cm>Nama, t.Nilai AS Nilai_TPKS, s.Nilai_Rata_rata_Rapor FROM calon_mahasiswa cm JOIN slta s ON cm.SLTA_ID = s.SLTA_ID JOIN tpks t ON cm.No_Ujian = t.No_Ujian WHERE (t.Nilai = (SELECT MAX(Nilai) FROM tpks)) AND s.Nilai_Rata_rata_Rapor > 90; </pre>
Penjelasan	<ol style="list-style-type: none"> 1. CREATE VIEW calon_Mahasiswa_Tertinggi_Dan_Rapor_Atas_90 AS: Mendefinisikan pembuatan view dengan nama calon_Mahasiswa_Tertinggi_Dan_Rapor_Atas_90. 2. SELECT cm>Nama, t.Nilai AS Nilai_TPKS, s.Nilai_Rata_rata_Rapor: Menentukan kolom-kolom yang akan ditampilkan dalam view. <ul style="list-style-type: none"> cm>Nama: Nama calon mahasiswa. t.Nilai AS Nilai_TPKS: Nilai TPKS dari tabel tpks. s.Nilai_Rata_rata_Rapor: Nilai rata-rata rapor dari tabel slta. 3. FROM calon_mahasiswa cm JOIN slta s ON cm.SLTA_ID = s.SLTA_ID JOIN tpks t ON cm.No_Ujian = t.No_Ujian WHERE t.Nilai = (SELECT MAX(Nilai) FROM tpks) AND s.Nilai_Rata_rata_Rapor > 90; : <ul style="list-style-type: none"> calon_mahasiswa cm: Menggunakan tabel calon_mahasiswa dengan alias cm. JOIN slta s ON cm.SLTA_ID = s.SLTA_ID: Melakukan operasi JOIN dengan tabel slta menggunakan kondisi SLTA_ID. JOIN tpks t ON cm.No_Ujian = t.No_Ujian: Melakukan operasi JOIN dengan tabel tpks menggunakan kondisi No_Ujian.

	<ul style="list-style-type: none"> • WHERE t.Nilai = (SELECT MAX(nilai) FROM tpks) AND s.Nilai_Rata_rata_Rapor > 90; : Memilih hanya baris yang memiliki nilai TPKS tertinggi dan nilai rata-rata rapor di atas 90.
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Screenshot

A screenshot of a database management interface. At the top, there is a navigation bar with icons and labels for 'Struktur', 'SQL', 'Cari', 'Kueri', 'Ekspor', 'Impor', 'Operasi', 'Hak Akses', 'Routine', 'Event', 'Trigger', 'Pelacakan', and 'Des'. Below this is a search bar with the text 'Tampilkan kotak kueri'. The main area displays a MySQL query: 'CREATE VIEW calon_Mahasiswa_Tertinggi_Dan_Rapor_Atas_90 AS SELECT cm>Nama, t.Nilai AS Nilai_TPKS, s.Nilai_Rata_rata_Rapor FROM calon_mahasiswa cm JOIN slta s ON cm.SLTA_ID = s.SLTA_ID JOIN tpks t ON cm.No_Ujian = t.No_Ujian WHERE (SELECT MAX(Nilai) FROM tpks) AND s.Nilai_Rata_rata_Rapor > 90;'. Above the query, a green message box states: 'MySQL memberikan hasil kosong (atau nol baris) (Pencarian dilakukan dalam 0,0008 detik)'. Below the query, there are links for '[Edit dikotak]' and '[Buat kode PHP]'.

Hasil view :

Nama	Nilai_TPKS	Nilai_Rata_rata_Rapor
------	------------	-----------------------

Penjelasan :

Dari hasil view diatas, tidak ada calon mahasiswa yang memiliki nilai tpks tertinggi sekaligus mempunyai rata-rata rapor diatas 90.

Soal 6 : Membuat view yang menampilkan jumlah calon mahasiswa yang diterima dan ditolak berdasarkan jurusan.	
Query	<pre>CREATE VIEW Status_Diterima_Ditolak_By_Jurusan AS SELECT Jurusan>Nama_Jurusan, SUM(CASE WHEN TPKS.Nilai >= 85 THEN 1 ELSE 0 END) AS Jumlah_Diterima, SUM(CASE WHEN TPKS.Nilai < 85 THEN 1 ELSE 0 END) AS Jumlah_Ditolak FROM Calon_Mahasiswa JOIN TPKS ON Calon_Mahasiswa.No_Ujian = TPKS.No_Ujian</pre>

	JOIN Pendaftaran ON Calon_Mahasiswa.NISN = Pendaftaran.NISN JOIN Jurusan ON Calon_Mahasiswa.Jurusan_ID = Jurusan.JurusanID GROUP BY Jurusan>Nama_Jurusan;
Penjelasan	<ol style="list-style-type: none"> 1. SELECT Jurusan>Nama_Jurusan: Memilih kolom Nama_Jurusan dari tabel Jurusan. 2. SUM(CASE WHEN TPKS.Nilai >= 85 THEN 1 ELSE 0 END) AS Jumlah_Diterima: Menggunakan fungsi SUM untuk menghitung jumlah calon mahasiswa yang diterima (nilai TPKS di atas atau sama dengan 85). Fungsi CASE digunakan untuk menentukan apakah nilai TPKS memenuhi kriteria diterima. 3. SUM(CASE WHEN TPKS.Nilai < 85 THEN 1 ELSE 0 END) AS Jumlah_Ditolak: Menggunakan fungsi agregat SUM untuk menghitung jumlah calon mahasiswa yang ditolak (nilai TPKS di bawah 85). Fungsi CASE digunakan untuk menentukan apakah nilai TPKS memenuhi kriteria ditolak. 4. FROM Calon_Mahasiswa: Mengambil data dari tabel Calon_Mahasiswa. 5. JOIN TPKS ON Calon_Mahasiswa.No_Ujian = TPKS.No_Ujian: Menggabungkan tabel TPKS dengan Calon_Mahasiswa berdasarkan kolom No_Ujian. 6. JOIN Pendaftaran ON Calon_Mahasiswa.NISN = Pendaftaran.NISN: Menggabungkan tabel Pendaftaran dengan Calon_Mahasiswa berdasarkan kolom NISN. 7. JOIN Jurusan ON Calon_Mahasiswa.Jurusan_ID = Jurusan.JurusanID: Menggabungkan tabel Jurusan dengan Calon_Mahasiswa berdasarkan kolom Jurusan_ID. 8. GROUP BY Jurusan>Nama_Jurusan: Mengelompokkan hasil berdasarkan nama jurusan, sehingga perhitungan jumlah diterima dan ditolak dilakukan per jurusan.

Screenshot

Struktur
SQL
Cari
Kueri
Ekspor
Impor
Operasi
Hak Akses
Routine
Event
Trigger
Pelacakan
De

Tampilkan kotak kueri

MySQL memberikan hasil kosong (atau nol baris). (Pencarian dilakukan dalam 0,0007 detik.)

```
CREATE VIEW Status_Diterima_Ditolak_By_Jurusan AS SELECT Jurusan>Nama_Jurusan, SUM(CASE WHEN TPKS.Nilai >= 85 THEN 1 ELSE 0 END) AS Jumlah_Diterima, SUM(CASE WHEN TPKS.Nilai < 85 THEN 1 ELSE 0 END) AS Jumlah_Ditolak FROM Calon_Mahasiswa JOIN TPKS ON Calon_Mahasiswa.No_Ujian = TPKS.No_Ujian JOIN Pendaftaran ON Calon_Mahasiswa.Pendaftaran.NISN JOIN Jurusan ON Calon_Mahasiswa.Jurusan_ID = Jurusan.JurusanID GROUP BY Jurusan>Nama_Jurusan;
```

Edit dikotak
Ubah
Buat kode PHP


Hasil view :

☐ Show all
Number of rows: 25
Filter rows: Search this table


Extra options

Nama_Jurusan	Jumlah_Diterima	Jumlah_Ditolak
Computer Science	2	1
Design Interior	1	0
Design Komunikasi Visual	2	0
Entrepreneurship Bussiness Creation	0	2
Public Relation	1	0
System Information	1	0

Pertanyaan 7 (Individu)	Membuat procedure yang dapat menampilkan informasi mahasiswa berdasarkan NISN.
Query & Penjelasan	<pre> DELIMITER // CREATE PROCEDURE Tampilkan_Info_Mahasiswa_By_NISN (IN p_NISN VARCHAR(10)) BEGIN SELECT * FROM Calon_Mahasiswa WHERE NISN = p_NISN; END // </pre>

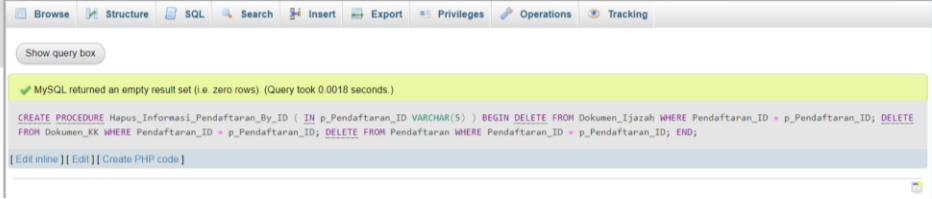
	<p>DELIMITER ;</p> <p>Penjelasan Query :</p> <ul style="list-style-type: none"> • DELIMITER // → Mengubah delimiter (pemisah perintah) agar dapat menggunakan multiple statements. • CREATE PROCEDURE Tampilkan_Info_Mahasiswa_By_NISN → Membuat prosedur dengan nama Tampilkan_Info_Mahasiswa_By_NISN. • IN p_NISN VARCHAR(10) → Mendefinisikan parameter input yang akan menjadi NISN calon mahasiswa yang ingin ditampilkan informasinya. • BEGIN...END → Area di mana perintah-perintah akan dijalankan. • SELECT * FROM Calon_Mahasiswa WHERE NISN = p_NISN → Memilih semua kolom dari tabel Calon_Mahasiswa di mana NISN-nya sama dengan nilai parameter yang diberikan.
Screenshot	 <p>✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0030 seconds)</p> <pre>CREATE PROCEDURE Tampilkan_Info_Mahasiswa_By_NISN (IN p_NISN VARCHAR(10)) BEGIN SELECT * FROM Calon_Mahasiswa WHERE NISN = p_NISN; END;</pre> <p>[Edit inline] [Edit] [Create PHP code]</p>

Pertanyaan 8 (Individu)	Membuat procedure yang dapat mengubah alamat calon mahasiswa berdasarkan NISN.
Query & Penjelasan	<p>DELIMITER //</p> <p>CREATE PROCEDURE Ubah_Alamat_Calon_Mahasiswa (</p> <p> IN p_NISN VARCHAR(10),</p> <p> IN p_Alamat_Barul VARCHAR(100)</p> <p>)</p> <p>BEGIN</p> <p> UPDATE Calon_Mahasiswa</p>

	<pre> SET Alamat = p_Alamat_Barul WHERE NISN = p_NISN; END // DELIMITER ; </pre> <p>Penjelasan Query :</p> <ul style="list-style-type: none"> • DELIMITER // → Mengubah delimiter (pemisah perintah) agar dapat menggunakan multiple statements. • CREATE PROCEDURE Ubah_Alamat_Calon_Mahasiswa → Membuat prosedur dengan nama Ubah_Alamat_Calon_Mahasiswa. • IN p_NISN VARCHAR(10), IN p_Alamat_Barul VARCHAR(100): Mendefinisikan dua parameter input: NISN calon mahasiswa dan alamat baru yang akan diubah. • BEGIN...END → Area di mana perintah-perintah akan dijalankan. • UPDATE Calon_Mahasiswa SET Alamat = p_Alamat_Barul WHERE NISN = p_NISN → Memperbarui alamat calon mahasiswa dengan NISN yang sesuai dengan nilai parameter yang diberikan. <p>Pastikan struktur tabel dan relasi yang tepat dalam perintah UPDATE agar perubahan alamat berjalan dengan benar.</p>
Screenshot	 <p>The screenshot shows a MySQL query execution window. At the top, it says 'Show query box'. Below that, a green status bar indicates 'MySQL returned an empty result set (i.e. zero rows) (Query took 0.0025 seconds)'. The main area displays the SQL code: 'CREATE PROCEDURE Ubah_Alamat_Calon_Mahasiswa (IN p_NISN VARCHAR(10), IN p_Alamat_Barul VARCHAR(100)) BEGIN UPDATE Calon_Mahasiswa SET Alamat = p_Alamat_Barul WHERE NISN = p_NISN; END;'. At the bottom, there are links for '[Edit mine]', '[Edit]', and '[Create PHP code]'.</p>

Pertanyaan 9 (Individu)	Membuat procedure yang dapat menghapus informasi pendaftaran calon mahasiswa berdasarkan ID pendaftaran.
--------------------------------	----------------------------------------------------------------------------------------------------------

<p>SQL</p> <p>&</p> <p>Penjelasan</p>	<pre> DELIMITER // CREATE PROCEDURE Hapus_Informasi_Pendaftaran_By_ID (IN p_Pendaftaran_ID VARCHAR(5)) BEGIN DELETE FROM Dokumen_Ijazah WHERE Pendaftaran_ID = p_Pendaftaran_ID; DELETE FROM Dokumen_KK WHERE Pendaftaran_ID = p_Pendaftaran_ID; DELETE FROM Pendaftaran WHERE Pendaftaran_ID = p_Pendaftaran_ID; END // DELIMITER ; </pre> <p>Penjelasan :</p> <ul style="list-style-type: none"> • DELIMITER // → Mengubah delimiter (pemisah perintah) agar dapat menggunakan multiple statements. • CREATE PROCEDURE Hapus_Pendaftaran_By_ID → Membuat prosedur dengan nama Hapus_Pendaftaran_By_ID. • IN p_Pendaftaran_ID VARCHAR(5) → Mendefinisikan parameter input yang akan menjadi ID pendaftaran yang ingin dihapus. • BEGIN...END → Area di mana perintah-perintah akan dijalankan. • DELETE FROM ... WHERE Pendaftaran_ID = p_Pendaftaran_ID → Menghapus entri pendaftaran, dokumen Ijazah, dan dokumen KK yang terkait dengan ID pendaftaran yang diberikan.
----------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	<ul style="list-style-type: none"> • DELIMITER ; → Mengembalikan delimiter ke keadaan semula.
Screenshot	 <p>The screenshot shows a MySQL query execution interface. The query executed is: <code>CREATE PROCEDURE Hapus_Informasi_Pendaftaran_By_ID (IN p_Pendaftaran_ID VARCHAR(5)) BEGIN DELETE FROM Dokumen_Ijazah WHERE Pendaftaran_ID = p_Pendaftaran_ID; DELETE FROM Dokumen_KK WHERE Pendaftaran_ID = p_Pendaftaran_ID; DELETE FROM Pendaftaran WHERE Pendaftaran_ID = p_Pendaftaran_ID; END;</code>. The result shows 'MySQL returned an empty result set (i.e. zero rows) (Query took 0.0018 seconds)'. The interface includes tabs for Browse, Structure, SQL, Search, Insert, Export, Privileges, Operations, and Tracking. Below the query box are links for 'Edit inline', 'Edit', and 'Create PHP code'.</p>

BAB IV

Kesimpulan

Proyek ini berhasil mengimplementasikan database management system untuk mendukung proses pendaftaran mahasiswa baru. Melalui berbagai tabel yang terhubung melalui relasi, validasi yang diterapkan, serta fungsionalitas tambahan seperti prosedur, fungsi, dan views, sistem ini memberikan dasar yang kokoh untuk pengelolaan data pendaftaran mahasiswa dengan lebih efisien dan terorganisir.

BAB V

Daftar Pustaka

- [1] “Perbedaan Database, Database Management System, dan Database Application Program,” Himpunan Mahasiswa Sistem Informasi. <https://student-activity.binus.ac.id/himsisfo/2017/05/perbedaan-database-database-management-system-dan-database-application-program/>
- [2] “Mengenal MySQL, Definisi, Fungsi, hingga Cara Kerjanya | Biznet Gio,” [www.biznetgio.com. https://www.biznetgio.com/news/apa-itu-mysql](https://www.biznetgio.com/news/apa-itu-mysql)
- [3] “Apa itu Bahasa Kueri Terstruktur (SQL)? - Penjelasan tentang SQL - AWS,” Amazon Web Services, Inc. <https://aws.amazon.com/id/what-is/sql/>
- [4] R. Setiawan, “Bagaimana Cara Membuat ERD dan Contohnya,” Dicoding Blog, Aug. 27, 2021. <https://www.dicoding.com/blog/cara-membuat-erd-dan-contohnya/>