# Dokumentasi Ujian Tengah Semester 5 Implementasi Manajemen Basis Data dengan PostgreSQL dan PGADMIN4

|  |  |
| --- | --- |
| Nama : Dimas Rizqia Hidayat | Dosen Pengampu : Wildan Budiawan Z,S.T.,M.Kom |
| NIM : 1237050073 | Kelas : 5/A (Ganjil) |
| Prodi : Teknik Informatika | Tahun Akademik : 2025/2026 |
| Matakuliah : Manajemen Basis Data | Tanggal Ujian : 23 Oktober 2025 (tanggal pengumpulan) |

## Daftar Isi

[Dokumentasi Ujian Tengah Semester 5 Implementasi Manajemen Basis Data dengan PostgreSQL dan PGADMIN4 1](#_Toc212123072)

[Daftar Isi 2](#_Toc212123073)

[A. Set Up Projek 3](#_Toc212123074)

[1. Instalasi Docker 3](#_Toc212123075)

[2. Instalasi dan Konfigurasi PostgreSQL (Pgadmin dan DBeaver) 3](#_Toc212123076)

[a. Instalasi PostgreSQL via Docker + connect ke pgadmin 3](#_Toc212123077)

[b. Konfigurasi Akses PostgreSQL ke DBeaver (Eksternal) 6](#_Toc212123078)

[B. Buat Skema SALAM, Table Mahasiswas dan Contrainst Test 7](#_Toc212123079)

[a. Membuat skema SALAM 7](#_Toc212123080)

[b. Membuat *Table* mahasiswas 7](#_Toc212123081)

[c. Memasukan data pada *Table* mahasiswas 7](#_Toc212123082)

[d. Contrainst Test pada Table mahasiswas 9](#_Toc212123083)

[e. Full Script (01\_buat\_skema.sql) 13](#_Toc212123084)

[C. Membuat User/Role (backend\_dev, bi\_dev, dan data\_engineer), Konfigurasi Akses dan Tes Akses 15](#_Toc212123085)

[a. Membuat User/Role dan Memberikan Akses umum ke Skema 15](#_Toc212123086)

[b. Memberikan Akses CRUD kepada backend\_dev 15](#_Toc212123087)

[c. Memberikan Akses Select (Read-Only) kepada bi\_dev 15](#_Toc212123088)

[d. Memberikan Akses Full kepada data\_engineer 16](#_Toc212123089)

[e. Tes Akses masing masing User 16](#_Toc212123090)

[f. Full Script 02\_buat\_user.sql 20](#_Toc212123091)

[g. Full Script 02.1\_backend\_dev\_test.sql 21](#_Toc212123092)

[h. Full Script 02.2\_bi\_dev\_test.sql 22](#_Toc212123093)

[i. Full Script 02.3\_data\_engineer\_test.sql 22](#_Toc212123094)

[D. Fungsi Query Tambahan + Tes Fungsi 23](#_Toc212123095)

[a. Fungsi cek status mahasiswa 23](#_Toc212123096)

[b. Fungsi Hitung mahasiswa sesuai prodi 24](#_Toc212123097)

[c. Fungsi Kategori IPK 25](#_Toc212123098)

[d. Fungsi Pencarian sederhana (Nim atau nama) 26](#_Toc212123099)

[e. Tes 2 : Sample data mahasiswa dengan fungsi Ambil 5 mahasiswa aktif dengan IPK tertinggi beserta kategorinya 28](#_Toc212123100)

[f. Full Script 03\_buat\_fungsi.sql 28](#_Toc212123101)

[g. Full Script 03.1\_quick\_test\_fungsi.sql 30](#_Toc212123102)

## Set Up Projek

### Instalasi Docker

Langkah-langkah Instalasi Docker;

1. Download Installer Docker Desktop dari situs resmi : <https://www.docker.com/products/docker-desktop/>
2. Install Docker Desktop menggunakan docker installer, Lalu Restart PC/Laptop setelah proses instalasi selesai.
3. Verifikasi Instalasi

* Buka Docker Deskstop > Terminal
* Jalankan perintah berikut :

|  |
| --- |
| docker --version  docker-compose --version |

* Output seharusnya sebagai berikut:



### Instalasi dan Konfigurasi PostgreSQL (Pgadmin dan DBeaver)

#### Instalasi PostgreSQL via Docker + connect ke pgadmin

* Pertama, buat struktur direktori projek

|  |
| --- |
| mkdir uts-postgre-pgadmin4  cd uts-postgre-pgadmin4 |

* Kemudian, Buat file docker-compose.yml seperti berikut:

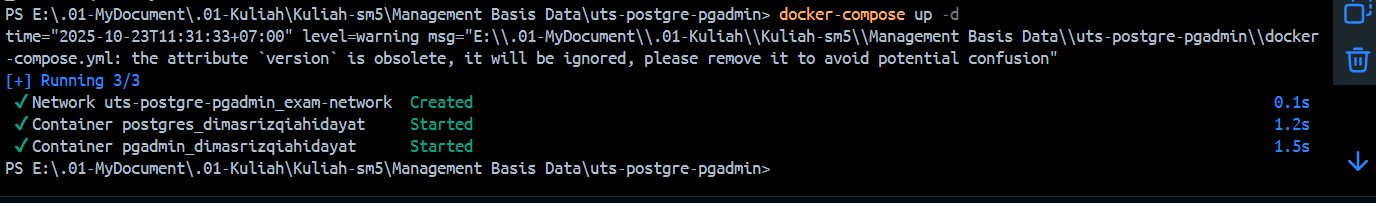
|  |
| --- |
| docker-compose.yml |
| version: '3.8'  services:    postgres\_dimasrizqiahidayat:      image: postgres:15      container\_name: postgres\_dimasrizqiahidayat      restart: always      env\_file:        - .env      ports:        - "22073:5432"      volumes:        - postgres\_data:/var/lib/postgresql/data      networks:        - exam-network    pgadmin\_dimasrizqiahidayat:      image: dpage/pgadmin4      container\_name: pgadmin\_dimasrizqiahidayat      restart: always      env\_file:        - .env      ports:        - "44073:80"      depends\_on:        - postgres\_dimasrizqiahidayat      volumes:        - pgadmin\_data:/var/lib/pgadmin      networks:        - exam-network  volumes:    postgres\_data:    pgadmin\_data:  networks:    exam-network:      driver: bridge |

* Buat file .env seperti berikut:

|  |
| --- |
| .env |
| POSTGRES\_USER=postgres  POSTGRES\_PASSWORD=ifunggul  POSTGRES\_DB=mahasiswa\_db  PGADMIN\_DEFAULT\_EMAIL=admin@admin.com  PGADMIN\_DEFAULT\_PASSWORD=admin123 |

* Jalankan docker-compose.yml   
  di Terminal jalankan command berikut;

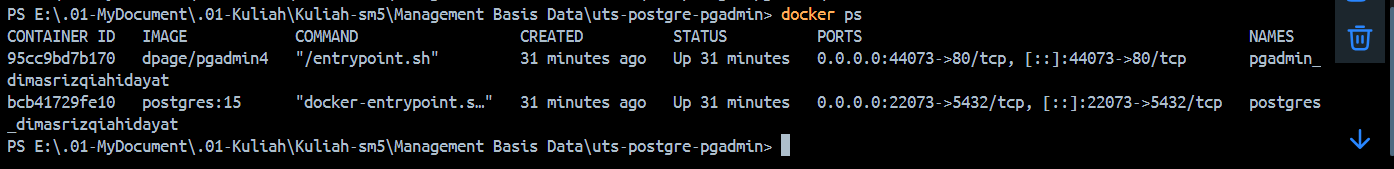
|  |
| --- |
| docker-compose up -d |

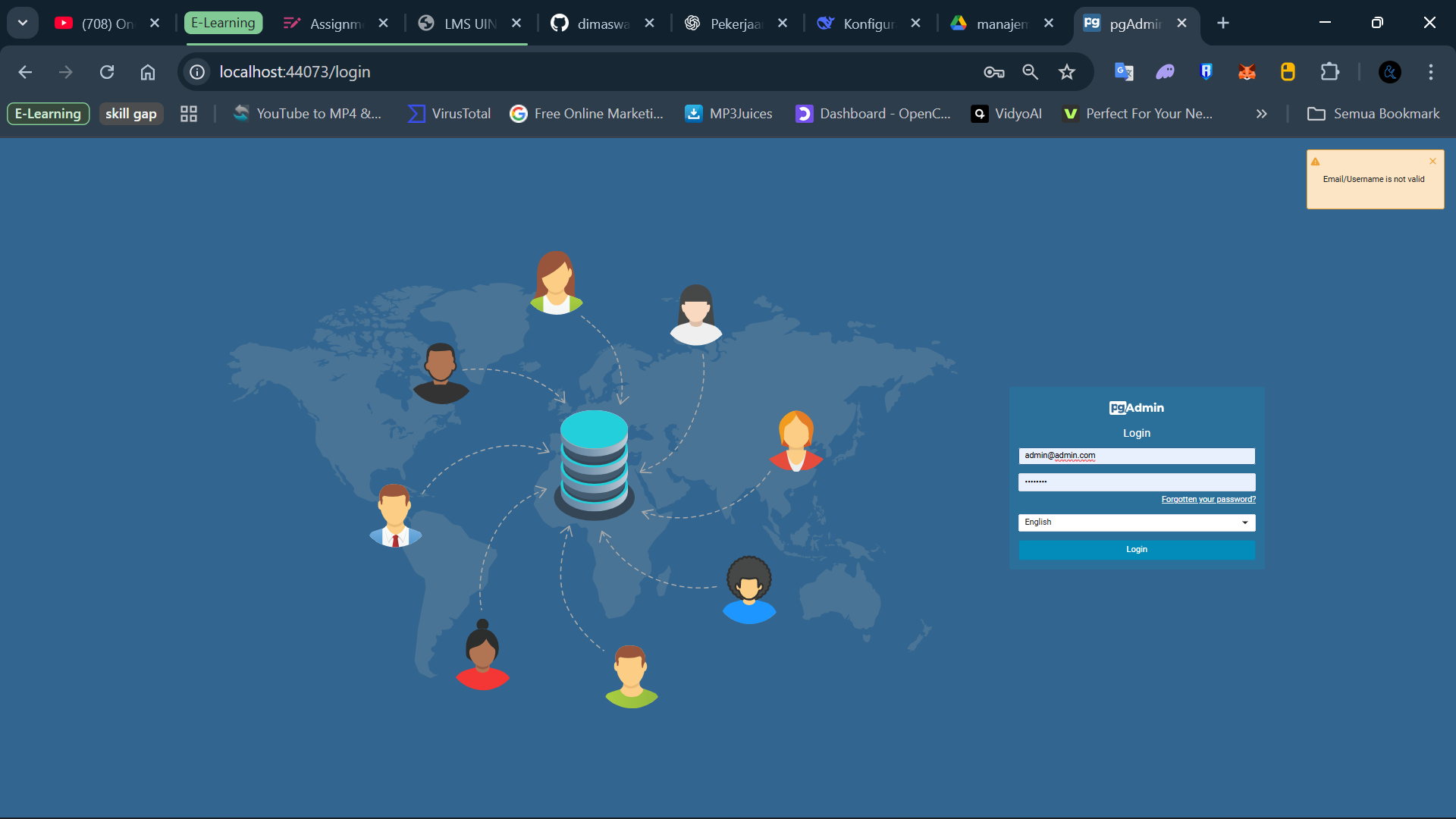


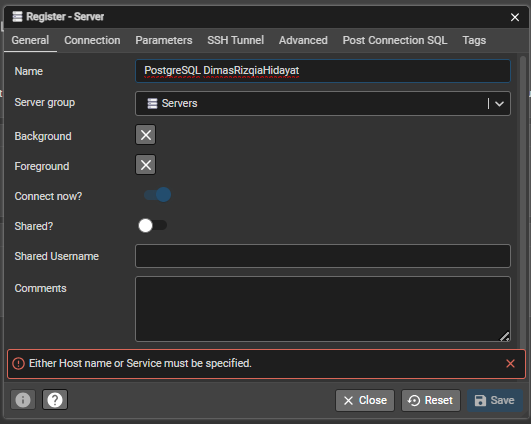
Lalu cek apakah Container sudah di jalankan

|  |
| --- |
| docker ps |

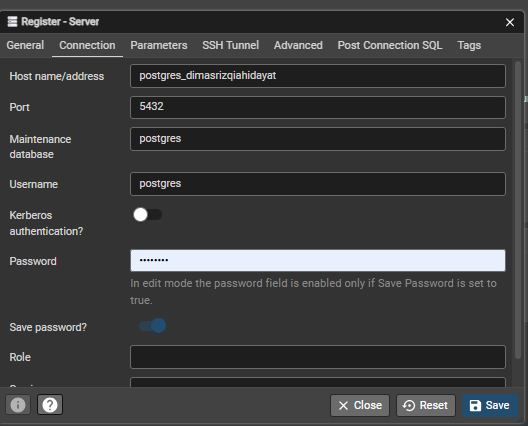
Harusnya muncul seperti ini



* Akses pgadmin via browser, lalu buka -> <http://localhost:44073>
* Lalu login dengan akun admin   
  Email : [admin@admin.com](mailto:admin@admin.com)  
  Password : admin123
* Hubungkan ke PostgreSQL (Container Internal)
  + Di sidebar kiri pgAdmin/Quick link Add new Server → Register → Server
  + Tab General:

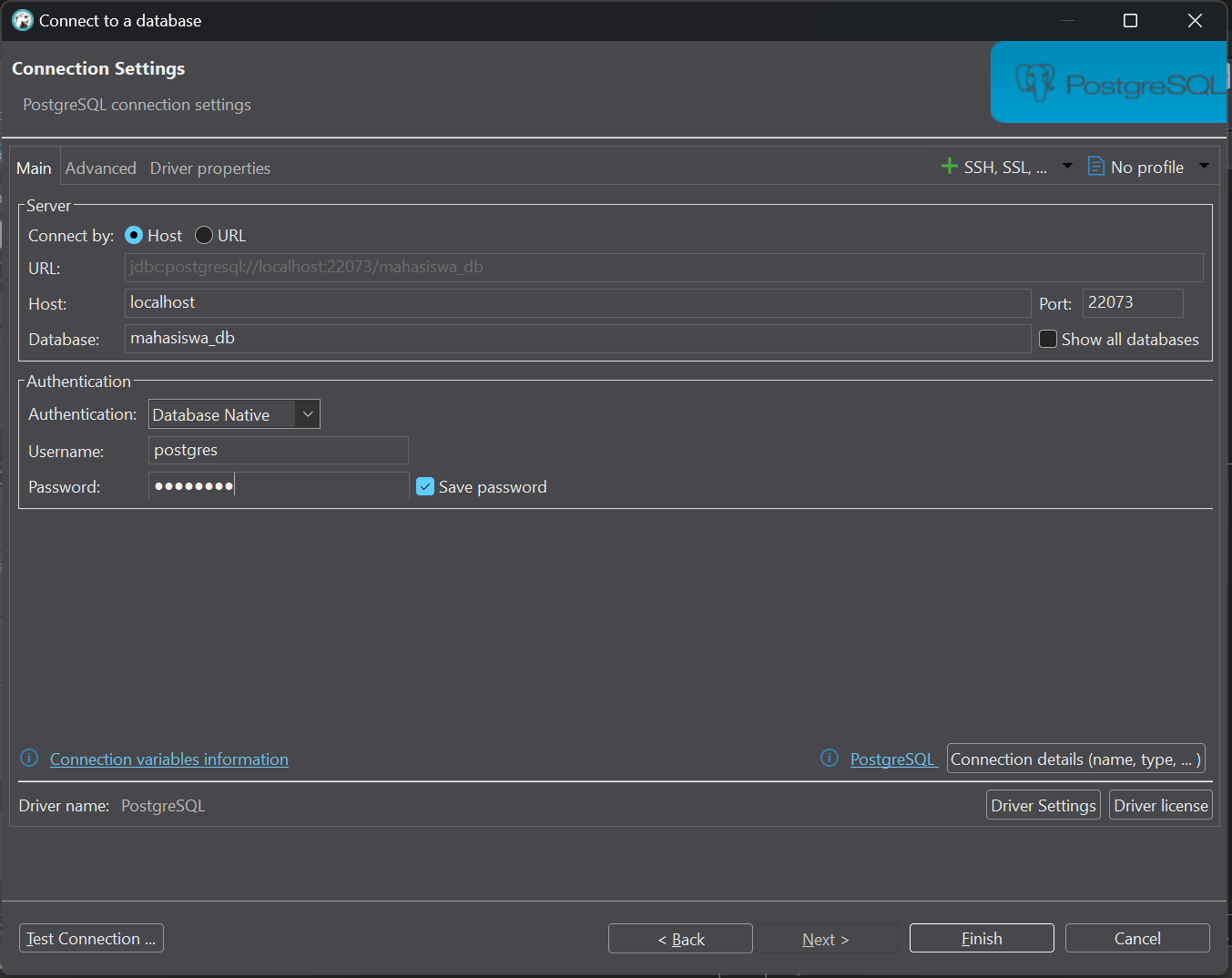
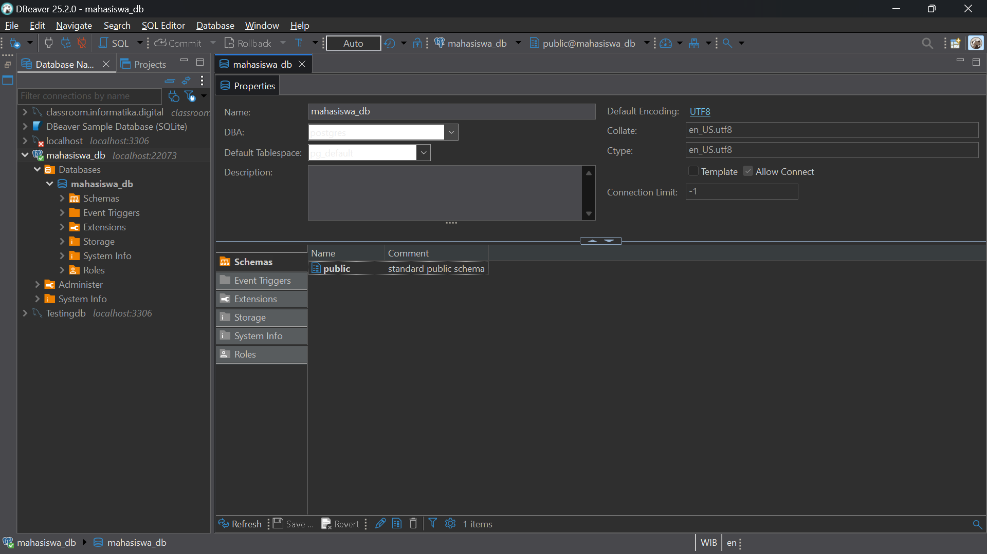


* + - Name: PostgreSQL DimasRizqiaHidayat
  + Tab Connection:



* + - Host name/address: postgres\_dimasrizqiahidayat  
      (ini sesuai dengan nama service di docker-compose.yml)
    - Port: 5432
    - Username: postgres
    - Password: ifunggul
  + Klik Save

#### Konfigurasi Akses PostgreSQL ke DBeaver (Eksternal)

* Pastikan Container berjalan. Jalankan docker ps di Terminal dan pastikan postgres\_dimasrizqiahidayat (port 22073:5432) berjalan.   
  Jika belum, jalankan docker-compose up -d
* Buka DBeaver, Lalu;
  + Klik ikon “New Database Connection” (ikon colokan listrik 🔌)
  + Pilih PostgreSQL
  + Klik Next
* Isi detail koneksi sesuai yang tertera pada docker-compose.yml dan .env  
  
  + Host : localhost
  + Port : 22073
  + Database : mahasiswa\_db
  + Username : postgres
  + Password : ifunggul
* Centang “Save password locally”, Lalu klik “Test Connection”
* Setelah berhasil klik “Finish”  
  

## Buat Skema SALAM, Table Mahasiswas dan Contrainst Test

Untuk tahap ini kita akan menggunakan Query Tools di pgadmin

#### Membuat skema SALAM

|  |
| --- |
| **CREATE** **SCHEMA** SALAM; |

#### Membuat *Table* mahasiswas

|  |
| --- |
| CREATE TABLE SALAM.mahasiswas (      nim CHAR(10) PRIMARY KEY,      nama VARCHAR(100) NOT NULL,      prodi VARCHAR(100) NOT NULL,      fakultas VARCHAR(100) NOT NULL,      semester INT CHECK (semester BETWEEN 1 AND 14),      angkatan INT CHECK (angkatan BETWEEN 2000 AND EXTRACT(YEAR FROM CURRENT\_DATE)),      email VARCHAR(100) UNIQUE,      usia INT CHECK (usia >= 16),      jenis\_kelamin CHAR(1) CHECK (jenis\_kelamin IN ('L', 'P')),      ipk NUMERIC(3,2) CHECK (ipk BETWEEN 0.00 AND 4.00) DEFAULT NULL,      status\_aktif VARCHAR(20) CHECK (status\_aktif IN ('Aktif', 'Cuti', 'Lulus', 'Drop Out')),        -- Jika IPK masih NULL (mahasiswa baru), maka semester HARUS 1      CONSTRAINT check\_semester\_mahasiswa\_baru CHECK (          (ipk IS NULL AND semester = 1) OR (ipk IS NOT NULL)      )  ); |

#### Memasukan data pada *Table* mahasiswas

|  |
| --- |
| INSERT INTO SALAM.mahasiswas  (nim, nama, prodi, fakultas, semester, angkatan, email, usia, jenis\_kelamin, ipk, status\_aktif)  VALUES  ('1237150001', 'Ahmad Taufik', 'Informatika', 'Sains dan Teknologi', 5, 2023, 'ahmad112@mail.com', 20, 'L', 3.45, 'Aktif'); |

Atau bisa memasukan data dummy yang sudah saya siapkan pada file 01.1\_insert\_dummy.sql

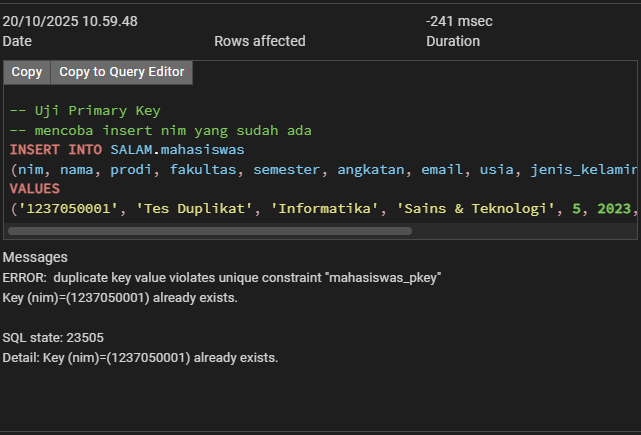
|  |
| --- |
| -- 40 data mahasiswa (semester disesuaikan dengan angkatan)  INSERT INTO SALAM.mahasiswas  (nim, nama, prodi, fakultas, semester, angkatan, email, usia, jenis\_kelamin, ipk, status\_aktif)  VALUES  -- Informatika (70-50)  ('1227050001','Ahmad Fauzi','Informatika','Sains & Teknologi',7,2022,'1227050001@student.uinsgd.ac.id',22,'L',3.45,'Aktif'),  ('1227050002','Siti Rahma','Informatika','Sains & Teknologi',7,2022,'1227050002@student.uinsgd.ac.id',21,'P',3.12,'Aktif'),  ('1237050003','Rafi Ramadhan','Informatika','Sains & Teknologi',5,2023,'1237050003@student.uinsgd.ac.id',20,'L',3.15,'Aktif'),  ('1237050004','Dewi Lestari','Informatika','Sains & Teknologi',5,2023,'1237050004@student.uinsgd.ac.id',22,'P',3.88,'Aktif'),  ('1247050005','Fahri Hidayat','Informatika','Sains & Teknologi',3,2024,'1247050005@student.uinsgd.ac.id',19,'L',3.20,'Aktif'),  ('1247050006','Nadia Aulia','Informatika','Sains & Teknologi',3,2024,'1247050006@student.uinsgd.ac.id',18,'P',3.55,'Aktif'),  ('1257050007','Bima Prasetyo','Informatika','Sains & Teknologi',1,2025,'1257050007@student.uinsgd.ac.id',18,'L',NULL,'Aktif'),  ('1257050008','Lina Marlina','Informatika','Sains & Teknologi',1,2025,'1257050008@student.uinsgd.ac.id',18,'P',NULL,'Aktif'),  -- Biologi (70-51)  ('1227051001','Rina Amalia','Biologi','Sains & Teknologi',7,2022,'1227051001@student.uinsgd.ac.id',22,'P',3.55,'Aktif'),  ('1227051002','Bayu Firmansyah','Biologi','Sains & Teknologi',7,2022,'1227051002@student.uinsgd.ac.id',21,'L',3.10,'Aktif'),  ('1237051003','Nia Rosita','Biologi','Sains & Teknologi',5,2023,'1237051003@student.uinsgd.ac.id',21,'P',3.78,'Aktif'),  ('1237051004','Andi Setiawan','Biologi','Sains & Teknologi',5,2023,'1237051004@student.uinsgd.ac.id',22,'L',3.33,'Aktif'),  ('1247051005','Citra Permata','Biologi','Sains & Teknologi',3,2024,'1247051005@student.uinsgd.ac.id',18,'P',2.90,'Aktif'),  ('1247051006','Rizky Maulana','Biologi','Sains & Teknologi',3,2024,'1247051006@student.uinsgd.ac.id',19,'L',3.0,'Aktif'),  ('1257051007','Ayu Wulandari','Biologi','Sains & Teknologi',1,2025,'1257051007@student.uinsgd.ac.id',18,'P',NULL,'Aktif'),  ('1257051008','Teguh Ramadhan','Biologi','Sains & Teknologi',1,2025,'1257051008@student.uinsgd.ac.id',18,'L',NULL,'Aktif'),  -- Aqidah & Filsafat Islam (71-52)  ('1227152001','Hasan Basri','Aqidah & Filsafat Islam','Ushuluddin',7,2022,'1227152001@student.uinsgd.ac.id',23,'L',3.60,'Aktif'),  ('1227152002','Nurul Hikmah','Aqidah & Filsafat Islam','Ushuluddin',7,2022,'1227152002@student.uinsgd.ac.id',21,'P',3.41,'Aktif'),  ('1237152003','Ridwan Kamil','Aqidah & Filsafat Islam','Ushuluddin',5,2023,'1237152003@student.uinsgd.ac.id',20,'L',3.10,'Aktif'),  ('1237152004','Maya Salsabila','Aqidah & Filsafat Islam','Ushuluddin',5,2023,'1237152004@student.uinsgd.ac.id',20,'P',3.25,'Aktif'),  ('1247152005','Hana Sari','Aqidah & Filsafat Islam','Ushuluddin',3,2024,'1247152005@student.uinsgd.ac.id',18,'P',3.35,'Aktif'),  ('1247152006','Faris Rahman','Aqidah & Filsafat Islam','Ushuluddin',3,2024,'1247152006@student.uinsgd.ac.id',18,'L',3.20,'Aktif'),  ('1257152007','Naila Qolbi','Aqidah & Filsafat Islam','Ushuluddin',1,2025,'1257152007@student.uinsgd.ac.id',18,'P',NULL,'Aktif'),  ('1257152008','Zain Fadhil','Aqidah & Filsafat Islam','Ushuluddin',1,2025,'1257152008@student.uinsgd.ac.id',18,'L',NULL,'Aktif'),  -- Ekonomi Syariah (72-53)  ('1227253001','Eka Nurhaliza','Ekonomi Syariah','Ekonomi & Bisnis Islam',7,2022,'1227253001@student.uinsgd.ac.id',23,'P',3.85,'Aktif'),  ('1227253002','Rangga Aditya','Ekonomi Syariah','Ekonomi & Bisnis Islam',7,2022,'1227253002@student.uinsgd.ac.id',21,'L',3.15,'Aktif'),  ('1237253003','Dini Larasati','Ekonomi Syariah','Ekonomi & Bisnis Islam',5,2023,'1237253003@student.uinsgd.ac.id',21,'P',3.75,'Aktif'),  ('1237253004','Taufik Hidayat','Ekonomi Syariah','Ekonomi & Bisnis Islam',5,2023,'1237253004@student.uinsgd.ac.id',20,'L',3.40,'Aktif'),  ('1247253005','Sarah Fadilah','Ekonomi Syariah','Ekonomi & Bisnis Islam',3,2024,'1247253005@student.uinsgd.ac.id',18,'P',3.9,'Aktif'),  ('1247253006','Reza Maulana','Ekonomi Syariah','Ekonomi & Bisnis Islam',3,2024,'1247253006@student.uinsgd.ac.id',19,'L',3.75,'Aktif'),  ('1257253007','Intan Rahayu','Ekonomi Syariah','Ekonomi & Bisnis Islam',1,2025,'1257253007@student.uinsgd.ac.id',18,'P',NULL,'Aktif'),  ('1257253008','Arif Wicaksono','Ekonomi Syariah','Ekonomi & Bisnis Islam',1,2025,'1257253008@student.uinsgd.ac.id',18,'L',NULL,'Aktif'),  -- Komunikasi & Penyiaran Islam (73-54)  ('1227354001','Dita Amelia','Komunikasi & Penyiaran Islam','Dakwah & Komunikasi',7,2022,'1227354001@student.uinsgd.ac.id',23,'P',3.50,'Aktif'),  ('1227354002','Fajar Nugraha','Komunikasi & Penyiaran Islam','Dakwah & Komunikasi',7,2022,'1227354002@student.uinsgd.ac.id',21,'L',3.20,'Aktif'),  ('1237354003','Silvia Rahmadani','Komunikasi & Penyiaran Islam','Dakwah & Komunikasi',5,2023,'1237354003@student.uinsgd.ac.id',20,'P',3.33,'Aktif'),  ('1237354004','Hafidz Fadlan','Komunikasi & Penyiaran Islam','Dakwah & Komunikasi',5,2023,'1237354004@student.uinsgd.ac.id',20,'L',3.12,'Aktif'),  ('1247354005','Aisyah Putri','Komunikasi & Penyiaran Islam','Dakwah & Komunikasi',3,2024,'1247354005@student.uinsgd.ac.id',18,'P',3.3,'Aktif'),  ('1247354006','Yoga Saputra','Komunikasi & Penyiaran Islam','Dakwah & Komunikasi',3,2024,'1247354006@student.uinsgd.ac.id',19,'L',3.55,'Aktif'),  ('1257354007','Luthfi Azhari','Komunikasi & Penyiaran Islam','Dakwah & Komunikasi',1,2025,'1257354007@student.uinsgd.ac.id',18,'L',NULL,'Aktif'),  ('1257354008','Nadia Khairunnisa','Komunikasi & Penyiaran Islam','Dakwah & Komunikasi',1,2025,'1257354008@student.uinsgd.ac.id',18,'P',NULL,'Aktif'); |

#### Contrainst Test pada Table mahasiswas

* Uji Primary Key

|  |
| --- |
| -- Uji Primary Key  -- mencoba insert nim yang sudah ada  INSERT INTO SALAM.mahasiswas  (nim, nama, prodi, fakultas, semester, angkatan, email, usia, jenis\_kelamin, ipk, status\_aktif)  VALUES  ('1237150001', 'Tes Duplikat', 'Informatika', 'Sains & Teknologi', 5, 2023, 'tes@mail.com', 20, 'L', 3.0, 'Aktif'); |

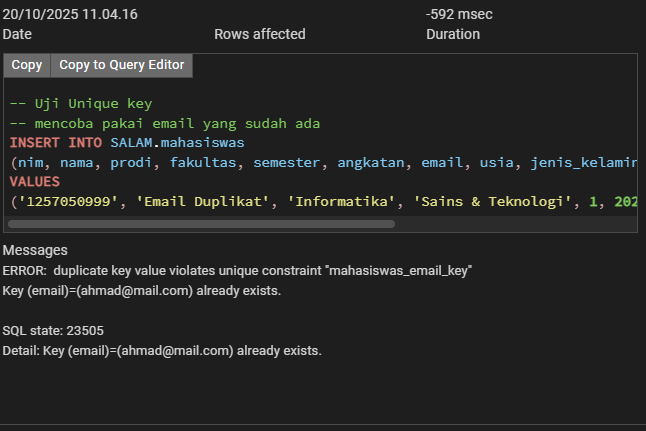
Hasil :



* Uji Unique Key

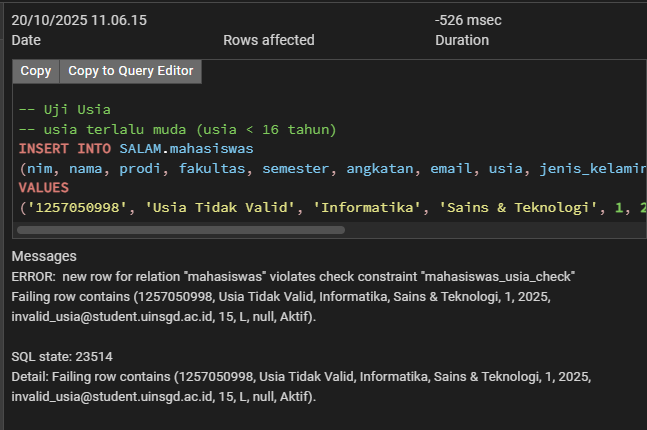
|  |
| --- |
| -- Uji Unique key  -- mencoba pakai email yang sudah ada  INSERT INTO SALAM.mahasiswas  (nim, nama, prodi, fakultas, semester, angkatan, email, usia, jenis\_kelamin, ipk, status\_aktif)  VALUES  ('1257050999', 'Email Duplikat', 'Informatika', 'Sains & Teknologi', 1, 2025, 'ahmad112@mail.com', 18, 'L', NULL, 'Aktif'); |

Hasil:



* Uji CHECK usia kurang dari 16 tahun

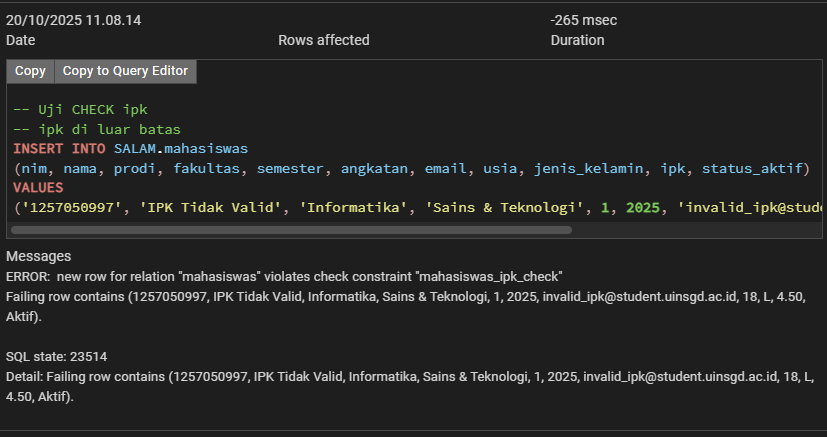
|  |
| --- |
| -- Uji CHECK Usia  -- usia terlalu muda (usia < 16 tahun)  INSERT INTO SALAM.mahasiswas  (nim, nama, prodi, fakultas, semester, angkatan, email, usia, jenis\_kelamin, ipk, status\_aktif)  VALUES  ('1257050998', 'Usia Tidak Valid', 'Informatika', 'Sains & Teknologi', 1, 2025, 'invalid\_usia@student.uinsgd.ac.id', 15, 'L', NULL, 'Aktif'); |

Hasil :  


* Uji CHECK ipk diluar batas

|  |
| --- |
| -- Uji CHECK ipk  -- ipk di luar batas  INSERT INTO SALAM.mahasiswas  (nim, nama, prodi, fakultas, semester, angkatan, email, usia, jenis\_kelamin, ipk, status\_aktif)  VALUES  ('1257050997', 'IPK Tidak Valid', 'Informatika', 'Sains & Teknologi', 1, 2025, 'invalid\_ipk@student.uinsgd.ac.id', 18, 'L', 4.5, 'Aktif'); |

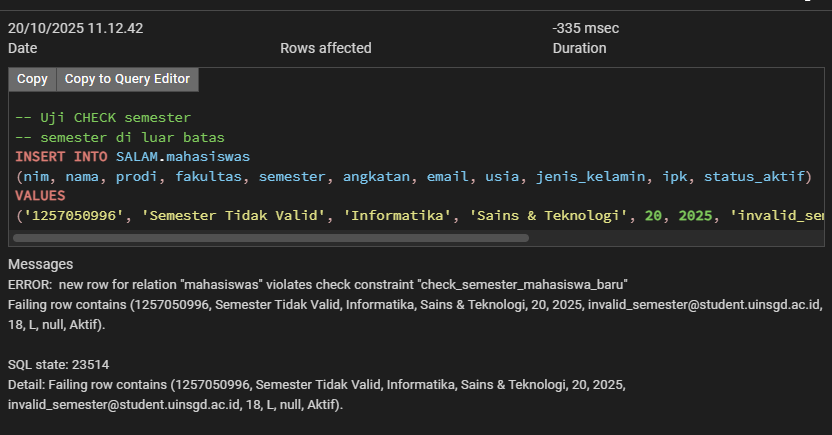
Hasil :



* Uji CHECK semester diluar batas

|  |
| --- |
| -- Uji CHECK semester  -- semester di luar batas  INSERT INTO SALAM.mahasiswas  (nim, nama, prodi, fakultas, semester, angkatan, email, usia, jenis\_kelamin, ipk, status\_aktif)  VALUES  ('1257050996', 'Semester Tidak Valid', 'Informatika', 'Sains & Teknologi', 20, 2025, 'invalid\_semester@student.uinsgd.ac.id', 18, 'L', NULL, 'Aktif'); |

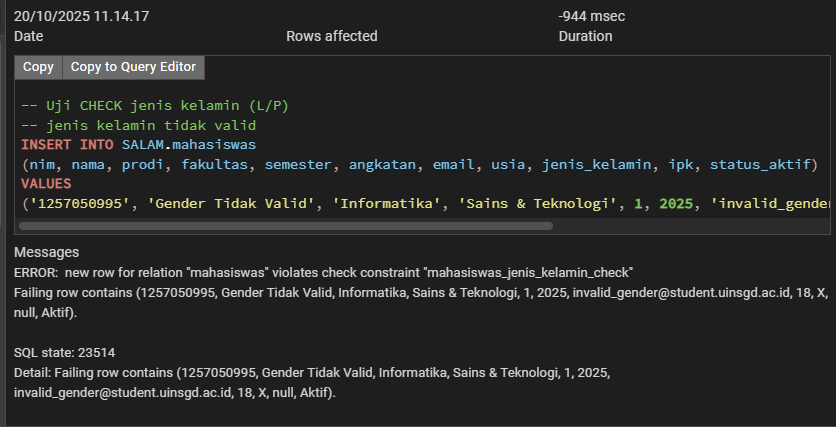
Hasil :



* Uji CHECK jenis kelamin (L/P)

|  |
| --- |
| -- Uji CHECK jenis kelamin (L/P)  -- jenis kelamin tidak valid  INSERT INTO SALAM.mahasiswas  (nim, nama, prodi, fakultas, semester, angkatan, email, usia, jenis\_kelamin, ipk, status\_aktif)  VALUES  ('1257050995', 'Gender Tidak Valid', 'Informatika', 'Sains & Teknologi', 1, 2025, 'invalid\_gender@student.uinsgd.ac.id', 18, 'X', NULL, 'Aktif'); |

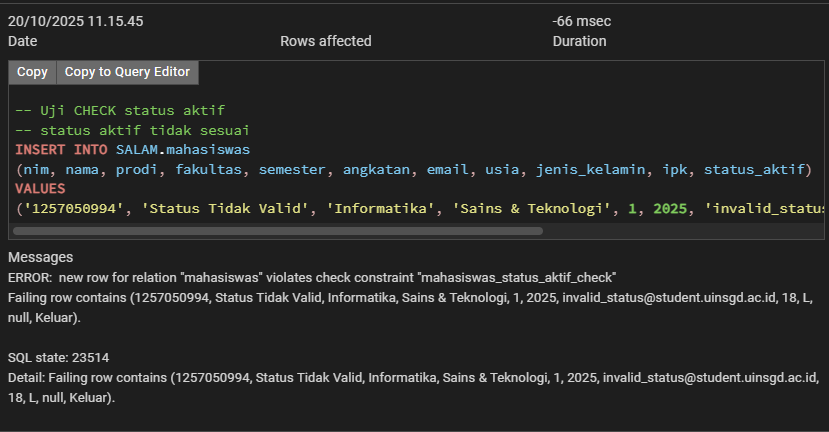
Hasil :



* Uji CHECK status aktif

|  |
| --- |
| -- Uji CHECK status aktif  -- status aktif tidak sesuai  INSERT INTO SALAM.mahasiswas  (nim, nama, prodi, fakultas, semester, angkatan, email, usia, jenis\_kelamin, ipk, status\_aktif)  VALUES  ('1257050994', 'Status Tidak Valid', 'Informatika', 'Sains & Teknologi', 1, 2025, 'invalid\_status@student.uinsgd.ac.id', 18, 'L', NULL, 'Keluar'); |

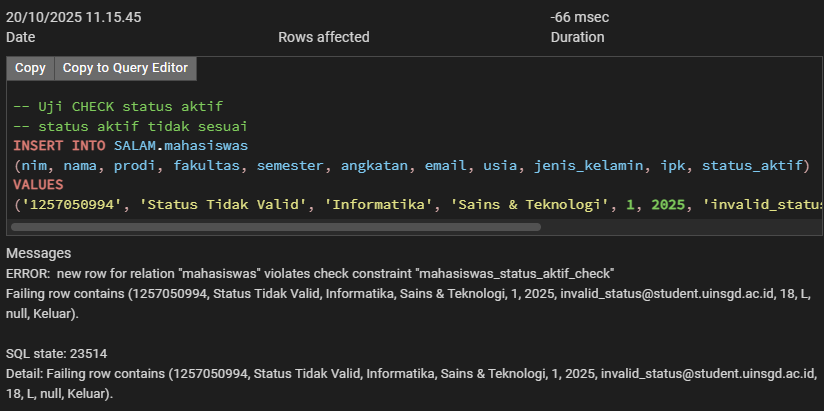
Hasil :



* Uji CHECK NULL Value

|  |
| --- |
| -- Uji CHECK NULL  -- nama kosong  INSERT INTO SALAM.mahasiswas  (nim, nama, prodi, fakultas, semester, angkatan, email, usia, jenis\_kelamin, ipk, status\_aktif)  VALUES  ('1257050993', NULL, 'Informatika', 'Sains & Teknologi', 1, 2025, 'invalid\_null@student.uinsgd.ac.id', 18, 'L', NULL, 'Aktif'); |

Hasil :



#### Full Script (01\_buat\_skema.sql)

|  |
| --- |
| -- Script untuk membuat skema dan tabel mahasiswas dengan berbagai constraint  CREATE SCHEMA SALAM;  GO  CREATE TABLE SALAM.mahasiswas (      nim CHAR(10) PRIMARY KEY,      nama VARCHAR(100) NOT NULL,      prodi VARCHAR(100) NOT NULL,      fakultas VARCHAR(100) NOT NULL,      semester INT CHECK (semester BETWEEN 1 AND 14),      angkatan INT CHECK (angkatan BETWEEN 2000 AND EXTRACT(YEAR FROM CURRENT\_DATE)),      email VARCHAR(100) UNIQUE,      usia INT CHECK (usia >= 16),      jenis\_kelamin CHAR(1) CHECK (jenis\_kelamin IN ('L', 'P')),      ipk NUMERIC(3,2) CHECK (ipk BETWEEN 0.00 AND 4.00) DEFAULT NULL,      status\_aktif VARCHAR(20) CHECK (status\_aktif IN ('Aktif', 'Cuti', 'Lulus', 'Drop Out')),        -- Jika IPK masih NULL (mahasiswa baru), maka semester HARUS 1      CONSTRAINT check\_semester\_mahasiswa\_baru CHECK (          (ipk IS NULL AND semester = 1) OR (ipk IS NOT NULL)      )  );  INSERT INTO SALAM.mahasiswas  (nim, nama, prodi, fakultas, semester, angkatan, email, usia, jenis\_kelamin, ipk, status\_aktif)  VALUES  ('1237150001', 'Ahmad Taufik', 'Informatika', 'Sains dan Teknologi', 5, 2023, 'ahmad112@mail.com', 20, 'L', 3.45, 'Aktif');  -- Uji Primary Key  -- mencoba insert nim yang sudah ada  INSERT INTO SALAM.mahasiswas  (nim, nama, prodi, fakultas, semester, angkatan, email, usia, jenis\_kelamin, ipk, status\_aktif)  VALUES  ('1237050001', 'Tes Duplikat', 'Informatika', 'Sains & Teknologi', 5, 2023, 'tes@mail.com', 20, 'L', 3.0, 'Aktif');  -- Uji Unique key  -- mencoba pakai email yang sudah ada  INSERT INTO SALAM.mahasiswas  (nim, nama, prodi, fakultas, semester, angkatan, email, usia, jenis\_kelamin, ipk, status\_aktif)  VALUES  ('1257050999', 'Email Duplikat', 'Informatika', 'Sains & Teknologi', 1, 2025, 'ahmad@mail.com', 18, 'L', NULL, 'Aktif');  -- Uji CHECK Usia  -- usia terlalu muda (usia < 16 tahun)  INSERT INTO SALAM.mahasiswas  (nim, nama, prodi, fakultas, semester, angkatan, email, usia, jenis\_kelamin, ipk, status\_aktif)  VALUES  ('1257050998', 'Usia Tidak Valid', 'Informatika', 'Sains & Teknologi', 1, 2025, 'invalid\_usia@student.uinsgd.ac.id', 15, 'L', NULL, 'Aktif');  -- Uji CHECK ipk  -- ipk di luar batas  INSERT INTO SALAM.mahasiswas  (nim, nama, prodi, fakultas, semester, angkatan, email, usia, jenis\_kelamin, ipk, status\_aktif)  VALUES  ('1257050997', 'IPK Tidak Valid', 'Informatika', 'Sains & Teknologi', 1, 2025, 'invalid\_ipk@student.uinsgd.ac.id', 18, 'L', 4.5, 'Aktif');  -- Uji CHECK semester  -- semester di luar batas  INSERT INTO SALAM.mahasiswas  (nim, nama, prodi, fakultas, semester, angkatan, email, usia, jenis\_kelamin, ipk, status\_aktif)  VALUES  ('1257050996', 'Semester Tidak Valid', 'Informatika', 'Sains & Teknologi', 20, 2025, 'invalid\_semester@student.uinsgd.ac.id', 18, 'L', NULL, 'Aktif');  -- Uji CHECK jenis kelamin (L/P)  -- jenis kelamin tidak valid  INSERT INTO SALAM.mahasiswas  (nim, nama, prodi, fakultas, semester, angkatan, email, usia, jenis\_kelamin, ipk, status\_aktif)  VALUES  ('1257050995', 'Gender Tidak Valid', 'Informatika', 'Sains & Teknologi', 1, 2025, 'invalid\_gender@student.uinsgd.ac.id', 18, 'X', NULL, 'Aktif');  -- Uji CHECK status aktif  -- status aktif tidak sesuai  INSERT INTO SALAM.mahasiswas  (nim, nama, prodi, fakultas, semester, angkatan, email, usia, jenis\_kelamin, ipk, status\_aktif)  VALUES  ('1257050994', 'Status Tidak Valid', 'Informatika', 'Sains & Teknologi', 1, 2025, 'invalid\_status@student.uinsgd.ac.id', 18, 'L', NULL, 'Keluar');  -- Uji CHECK NULL  -- nama kosong  INSERT INTO SALAM.mahasiswas  (nim, nama, prodi, fakultas, semester, angkatan, email, usia, jenis\_kelamin, ipk, status\_aktif)  VALUES  ('1257050993', NULL, 'Informatika', 'Sains & Teknologi', 1, 2025, 'invalid\_null@student.uinsgd.ac.id', 18, 'L', NULL, 'Aktif'); |

## Membuat User/Role (backend\_dev, bi\_dev, dan data\_engineer), Konfigurasi Akses dan Tes Akses

#### Membuat User/Role dan Memberikan Akses umum ke Skema

|  |
| --- |
| -- Pastikan kamu login sebagai superuser (misal: postgres)  CREATE USER backend\_dev WITH PASSWORD 'backend123';  CREATE USER bi\_dev WITH PASSWORD 'bidev123';  CREATE USER data\_engineer WITH PASSWORD 'dataeng123';  -- Berikan akses umum ke skema  GRANT USAGE ON SCHEMA SALAM TO backend\_dev, bi\_dev, data\_engineer; |

#### Memberikan Akses CRUD kepada backend\_dev

|  |
| --- |
| -- backend\_dev access account  -- CRUD Access : dapat membaca dan memodifikasi data di semua tabel  GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA SALAM TO backend\_dev; |

#### Memberikan Akses Select (Read-Only) kepada bi\_dev

|  |
| --- |
| -- bi\_dev access account  -- Hanya dapat membaca (SELECT)  GRANT SELECT ON ALL TABLES IN SCHEMA SALAM TO bi\_dev;  -- Termasuk view  GRANT SELECT ON ALL SEQUENCES IN SCHEMA SALAM TO bi\_dev;  ALTER DEFAULT PRIVILEGES IN SCHEMA SALAM  GRANT SELECT ON TABLES TO bi\_dev; |

#### Memberikan Akses Full kepada data\_engineer

|  |
| --- |
| -- data\_engineer account (Full Access)  -- Dapat membuat, mengubah, dan menghapus objek dalam skema  GRANT CREATE, USAGE ON SCHEMA SALAM TO data\_engineer;  -- CRUD ke seluruh tabel  GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA SALAM TO data\_engineer;  -- Hak tambahan  ALTER DEFAULT PRIVILEGES IN SCHEMA SALAM  GRANT ALL PRIVILEGES ON TABLES TO data\_engineer; |

#### Tes Akses masing masing User

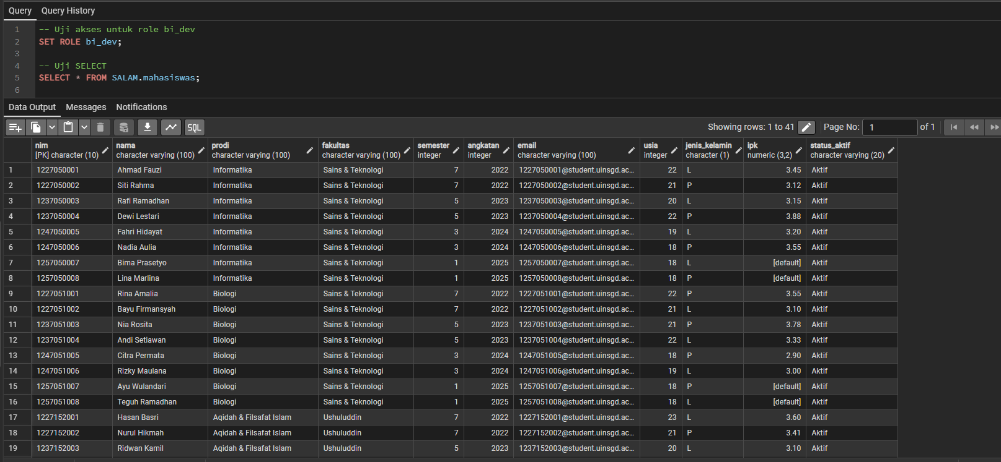
Jangan lupa untuk ubah ke role yang ingin di test!

|  |
| --- |
| SET ROLE backend\_dev;  Atau  SET ROLE bi\_dev;  Atau  SET ROLE data\_engineer; |

* CRUD Akses backend\_dev
  + Uji SELECT

|  |
| --- |
| -- Uji SELECT  SELECT \* FROM SALAM.mahasiswas; |

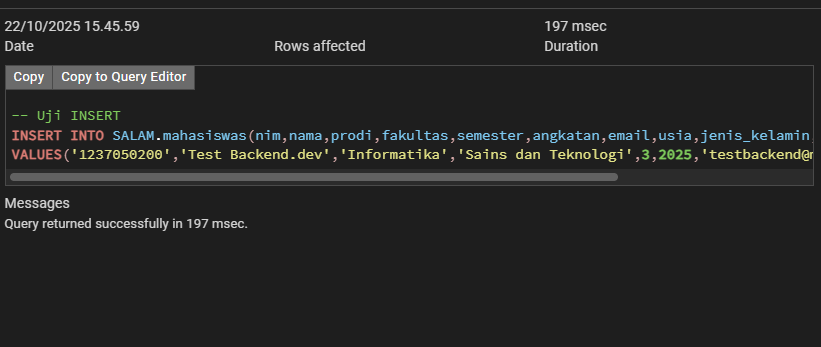
Hasil :



* + Uji INSERT

|  |
| --- |
| -- Uji INSERT  INSERT INTO SALAM.mahasiswas(nim,nama,prodi,fakultas,semester,angkatan,email,usia,jenis\_kelamin,ipk,status\_aktif)  VALUES('1237050200','Test Backend.dev','Informatika','Sains dan Teknologi',3,2025,'testbackend@mail.com',20,'L',3.5,'Aktif'); |

Hasil :



* + Uji UPDATE

|  |
| --- |
| -- Uji UPDATE  UPDATE SALAM.mahasiswas SET ipk=3.6 WHERE nim='1237050200'; |

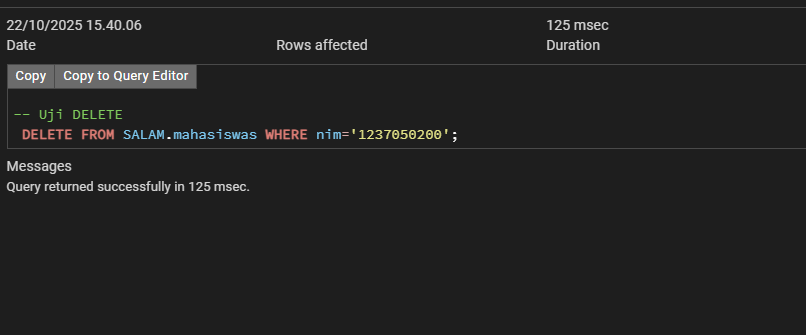
Hasil :



* + Uji DELETE

|  |
| --- |
| -- Uji DELETE   DELETE FROM SALAM.mahasiswas WHERE nim='1237050200'; |

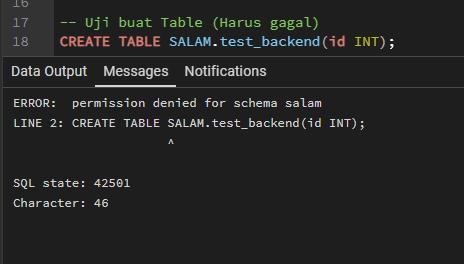
Hasil :



* + Uji Buat Table (Harus gagal)

|  |
| --- |
| -- Uji buat Table (Harus gagal)  CREATE TABLE SALAM.test\_backend(id INT); |

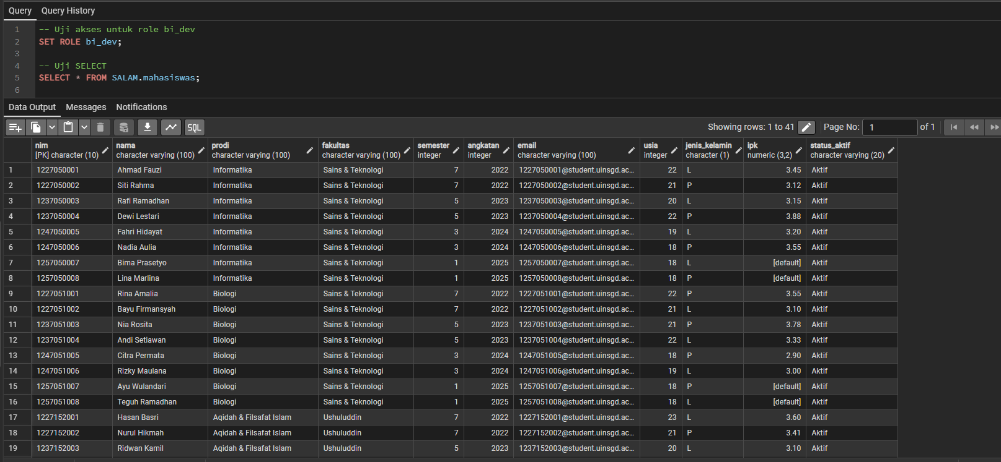
Hasil :



* Read-Only Akses bi\_dev
  + Uji SELECT

|  |
| --- |
| -- Uji SELECT  SELECT \* FROM SALAM.mahasiswas; |

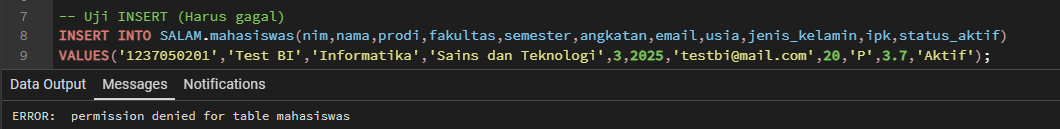
Hasil :



* + Uji INSERT (Harus gagal)

|  |
| --- |
| -- Uji INSERT (Harus gagal)  INSERT INTO SALAM.mahasiswas(nim,nama,prodi,fakultas,semester,angkatan,email,usia,jenis\_kelamin,ipk,status\_aktif)  VALUES('1237050201','Test BI','Informatika','Sains dan Teknologi',3,2025,'testbi@mail.com',20,'P',3.7,'Aktif'); |

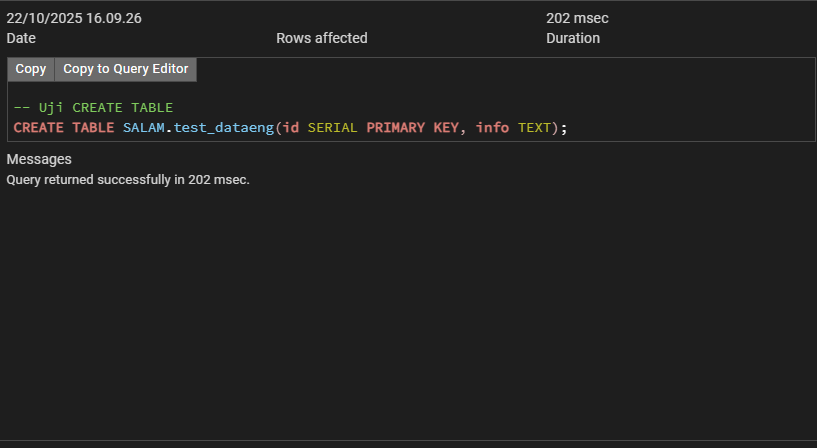
Hasil :



* Full Akses data\_engineer
  + Uji CREATE TABLE

|  |
| --- |
| -- Uji CREATE TABLE  CREATE TABLE SALAM.test\_dataeng(id SERIAL PRIMARY KEY, info TEXT); |

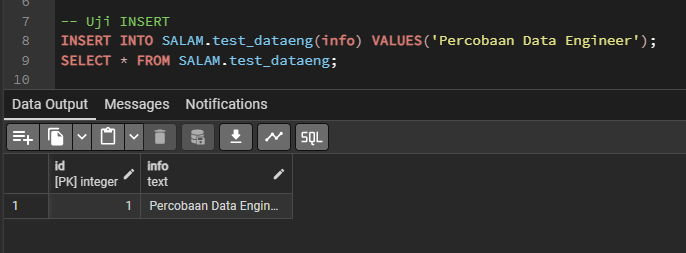
Hasil :



* + Uji INSERT

|  |
| --- |
| -- Uji INSERT  INSERT INTO SALAM.test\_dataeng(info) VALUES('Percobaan Data Engineer');  -- Lihat Hasil  SELECT \* FROM SALAM.test\_dataeng; |

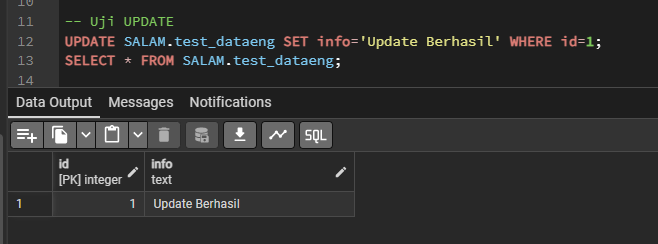
Hasil :



* + Uji UPDATE

|  |
| --- |
| -- Uji UPDATE  UPDATE SALAM.test\_dataeng SET info='Update Berhasil' WHERE id=1;  SELECT \* FROM SALAM.test\_dataeng; |

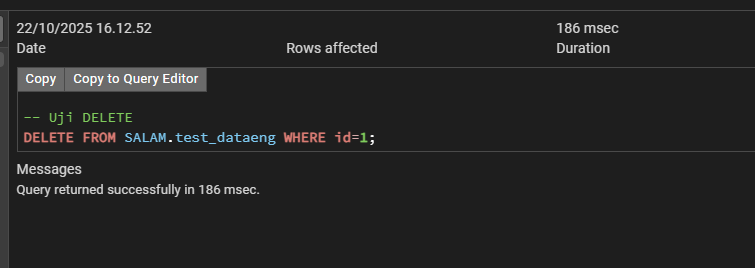
Hasil :



* + Uji DELETE

|  |
| --- |
| -- Uji DELETE  DELETE FROM SALAM.test\_dataeng WHERE id=1; |

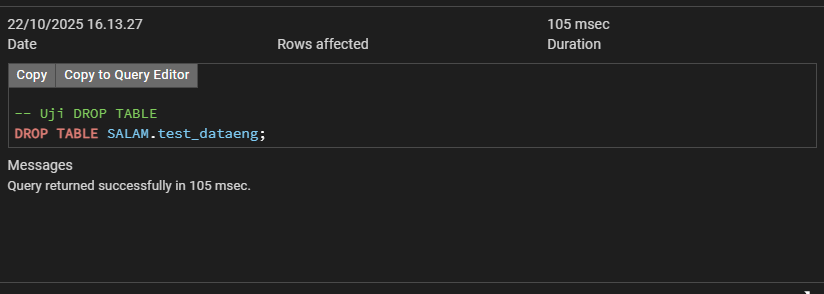
Hasil :



* + Uji DROP TABLE

|  |
| --- |
| -- Uji DROP TABLE  DROP TABLE SALAM.test\_dataeng; |

Hasil :



#### Full Script 02\_buat\_user.sql

|  |
| --- |
| -- Script new user etc --  -- Pastikan kamu login sebagai superuser (misal: postgres)  CREATE USER backend\_dev WITH PASSWORD 'backend123';  CREATE USER bi\_dev WITH PASSWORD 'bidev123';  CREATE USER data\_engineer WITH PASSWORD 'dataeng123';  -- ============================================================================== --  -- Berikan akses umum ke skema  GRANT USAGE ON SCHEMA SALAM TO backend\_dev, bi\_dev, data\_engineer;  -- backend\_dev access account  -- CRUD Access : dapat membaca dan memodifikasi data di semua tabel  GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA SALAM TO backend\_dev;  -- Agar hak akses ini otomatis berlaku untuk tabel baru  ALTER DEFAULT PRIVILEGES IN SCHEMA SALAM  GRANT SELECT, INSERT, UPDATE, DELETE ON TABLES TO backend\_dev;  -- bi\_dev access account  -- Hanya dapat membaca (SELECT)  GRANT SELECT ON ALL TABLES IN SCHEMA SALAM TO bi\_dev;  -- Termasuk view  GRANT SELECT ON ALL SEQUENCES IN SCHEMA SALAM TO bi\_dev;  ALTER DEFAULT PRIVILEGES IN SCHEMA SALAM  GRANT SELECT ON TABLES TO bi\_dev;  -- data\_engineer account (Full Access)  -- Dapat membuat, mengubah, dan menghapus objek dalam skema  GRANT CREATE, USAGE ON SCHEMA SALAM TO data\_engineer;  -- CRUD ke seluruh tabel  GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA SALAM TO data\_engineer;  -- Hak tambahan  ALTER DEFAULT PRIVILEGES IN SCHEMA SALAM  GRANT ALL PRIVILEGES ON TABLES TO data\_engineer; |

#### Full Script 02.1\_backend\_dev\_test.sql

|  |
| --- |
| -- Uji Testing user backend\_dev  SET ROLE backend\_dev;  -- Uji SELECT  SELECT \* FROM SALAM.mahasiswas;  -- Uji INSERT  INSERT INTO SALAM.mahasiswas(nim,nama,prodi,fakultas,semester,angkatan,email,usia,jenis\_kelamin,ipk,status\_aktif)  VALUES('1237050200','Test Backend.dev','Informatika','Sains dan Teknologi',3,2025,'testbackend@mail.com',20,'L',3.5,'Aktif');  -- Uji UPDATE  UPDATE SALAM.mahasiswas SET ipk=3.6 WHERE nim='1237050200';  -- Uji DELETE   DELETE FROM SALAM.mahasiswas WHERE nim='1237050200';  -- Uji buat Table (Harus gagal)  CREATE TABLE SALAM.test\_backend(id INT);  -- Kembali ke superuser  SET ROLE postgres; |

#### Full Script 02.2\_bi\_dev\_test.sql

|  |
| --- |
| -- Uji akses untuk role bi\_dev  SET ROLE bi\_dev;  -- Uji SELECT  SELECT \* FROM SALAM.mahasiswas;  -- Uji INSERT (Harus gagal)  INSERT INTO SALAM.mahasiswas(nim,nama,prodi,fakultas,semester,angkatan,email,usia,jenis\_kelamin,ipk,status\_aktif)  VALUES('1237050201','Test BI','Informatika','Sains dan Teknologi',3,2025,'testbi@mail.com',20,'P',3.7,'Aktif');  -- kembali ke superuser  SET ROLE postgres; |

#### Full Script 02.3\_data\_engineer\_test.sql

|  |
| --- |
| -- Uji akses untuk role data\_engineer  SET ROLE data\_engineer;  -- Uji CREATE TABLE  CREATE TABLE SALAM.test\_dataeng(id SERIAL PRIMARY KEY, info TEXT);  -- Uji INSERT  INSERT INTO SALAM.test\_dataeng(info) VALUES('Percobaan Data Engineer');  SELECT \* FROM SALAM.test\_dataeng;  -- Uji UPDATE  UPDATE SALAM.test\_dataeng SET info='Update Berhasil' WHERE id=1;  SELECT \* FROM SALAM.test\_dataeng;  -- Uji DELETE  DELETE FROM SALAM.test\_dataeng WHERE id=1;  -- Uji DROP TABLE  DROP TABLE SALAM.test\_dataeng;  -- kembali ke superuser  SET ROLE postgres; |

## Fungsi Query Tambahan + Tes Fungsi

#### Fungsi cek status mahasiswa

* Fungsi :

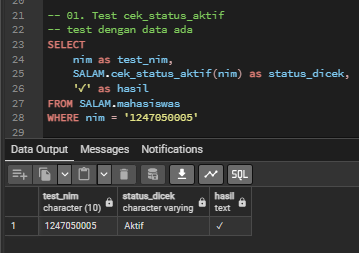
|  |
| --- |
| -- 01. Fungsi sederhana untuk cek status mahasiswa  CREATE OR REPLACE FUNCTION SALAM.cek\_status\_aktif(p\_nim CHAR(10))  RETURNS VARCHAR  LANGUAGE plpgsql  AS $$  DECLARE      status\_mhs VARCHAR(20);  BEGIN      SELECT status\_aktif INTO status\_mhs      FROM SALAM.mahasiswas      WHERE nim = p\_nim;        IF status\_mhs IS NULL THEN          RETURN 'MAHASISWA TIDAK DITEMUKAN';      ELSE          RETURN status\_mhs;      END IF;  END;  $$; |

* Testing :

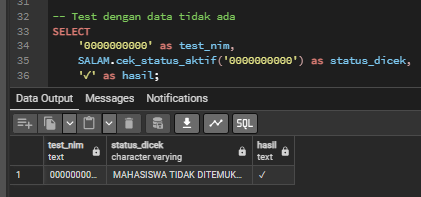
|  |
| --- |
| -- 01. Test cek\_status\_aktif  -- test dengan data ada  SELECT      nim as test\_nim,      SALAM.cek\_status\_aktif(nim) as status\_dicek,      '✓' as hasil  FROM SALAM.mahasiswas  WHERE nim = '1247050005'  UNION ALL  -- Test dengan data tidak ada  SELECT      '0000000000' as test\_nim,      SALAM.cek\_status\_aktif('0000000000') as status\_dicek,      '✓' as hasil; |

Hasil :

* + Tes data ada :



* + Tes data tidak ada :



#### Fungsi Hitung mahasiswa sesuai prodi

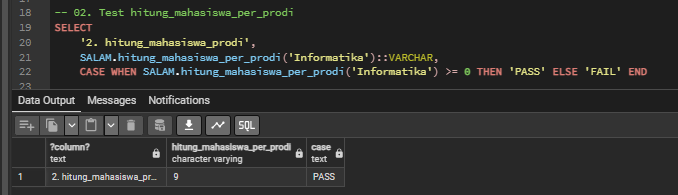
* Fungsi :

|  |
| --- |
| -- 02. Fungsi hitung mahasiswa per prodi  CREATE OR REPLACE FUNCTION SALAM.hitung\_mahasiswa\_per\_prodi(p\_prodi VARCHAR)  RETURNS INTEGER  LANGUAGE plpgsql  AS $$  DECLARE      total INTEGER;  BEGIN      SELECT COUNT(\*) INTO total      FROM SALAM.mahasiswas      WHERE prodi = p\_prodi      AND status\_aktif = 'Aktif';        RETURN total;  END;  $$; |

* Testing

|  |
| --- |
| -- 02. Test hitung\_mahasiswa\_per\_prodi  SELECT      '2. hitung\_mahasiswa\_prodi',      SALAM.hitung\_mahasiswa\_per\_prodi('Informatika')::VARCHAR,      CASE WHEN SALAM.hitung\_mahasiswa\_per\_prodi('Informatika') >= 0 THEN 'PASS' ELSE 'FAIL' END |

Hasil :



#### Fungsi Kategori IPK

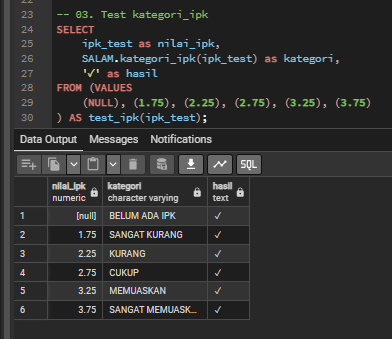
* Fungsi :

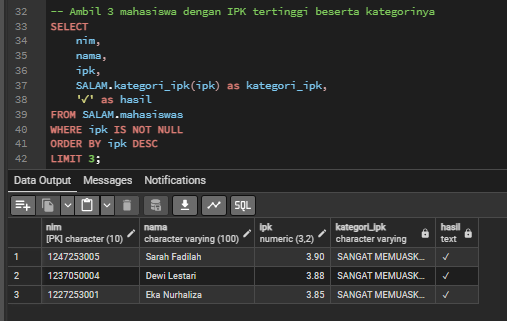
|  |
| --- |
| -- 03. Fungsi kategorisasi IPK  CREATE OR REPLACE FUNCTION SALAM.kategori\_ipk(p\_ipk NUMERIC)  RETURNS VARCHAR  LANGUAGE plpgsql  AS $$  BEGIN      IF p\_ipk IS NULL THEN          RETURN 'BELUM ADA IPK';      ELSIF p\_ipk >= 3.5 THEN          RETURN 'SANGAT MEMUASKAN';      ELSIF p\_ipk >= 3.0 THEN          RETURN 'MEMUASKAN';      ELSIF p\_ipk >= 2.5 THEN          RETURN 'CUKUP';      ELSIF p\_ipk >= 2.0 THEN          RETURN 'KURANG';      ELSE          RETURN 'SANGAT KURANG';      END IF;  END;  $$; |

* Testing

|  |
| --- |
| -- 03. Test kategori\_ipk  SELECT      ipk\_test as nilai\_ipk,      SALAM.kategori\_ipk(ipk\_test) as kategori,      '✓' as hasil  FROM (VALUES      (NULL), (1.75), (2.25), (2.75), (3.25), (3.75)  ) AS test\_ipk(ipk\_test); |
| -- Ambil 3 mahasiswa dengan IPK tertinggi beserta kategorinya  SELECT      nim,      nama,      ipk,      SALAM.kategori\_ipk(ipk) as kategori\_ipk,      '✓' as hasil  FROM SALAM.mahasiswas  WHERE ipk IS NOT NULL  ORDER BY ipk DESC  LIMIT 3; |

Hasil :

* + 
  + Ambil 3 IPK teratas



#### Fungsi Pencarian sederhana (Nim atau nama)

* Fungsi :

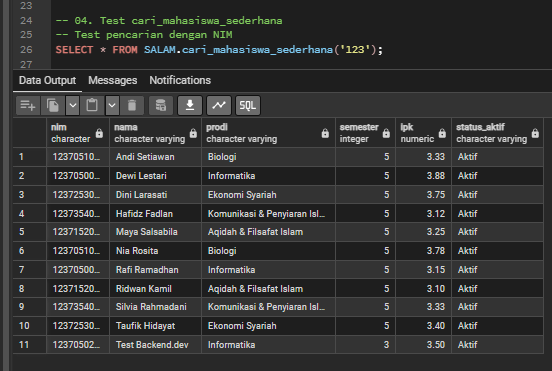
|  |
| --- |
| -- 04. Fungsi pencarian sederhana  CREATE OR REPLACE FUNCTION SALAM.cari\_mahasiswa\_sederhana(p\_keyword VARCHAR)  RETURNS TABLE(      nim CHAR(10),      nama VARCHAR(100),      prodi VARCHAR(100),      semester INTEGER,      ipk NUMERIC,      status\_aktif VARCHAR(20)  )  LANGUAGE plpgsql  AS $$  BEGIN      RETURN QUERY      SELECT          m.nim, m.nama, m.prodi, m.semester, m.ipk, m.status\_aktif      FROM SALAM.mahasiswas m      WHERE m.nim LIKE '%' || p\_keyword || '%'         OR m.nama ILIKE '%' || p\_keyword || '%'      ORDER BY m.nama;  END;  $$; |

* Testing

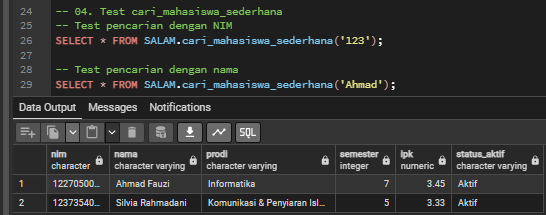
|  |
| --- |
| -- 04. Test cari\_mahasiswa\_sederhana  -- Test pencarian dengan NIM  SELECT \* FROM SALAM.cari\_mahasiswa\_sederhana('123');  -- Test pencarian dengan nama  SELECT \* FROM SALAM.cari\_mahasiswa\_sederhana('Ahmad'); |

Hasil :

* + Cari dengan NIM



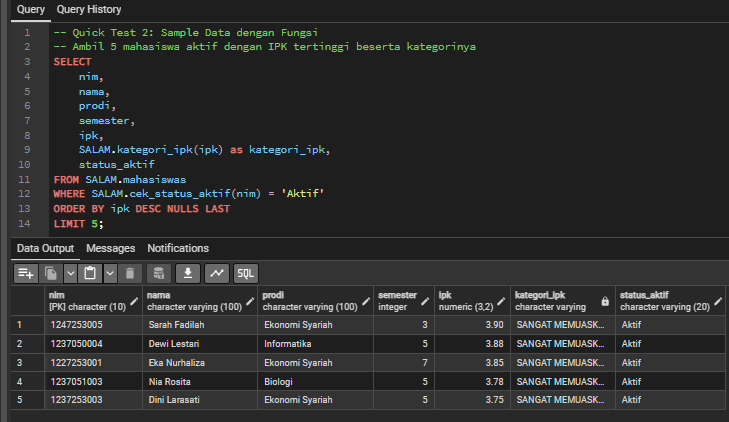
* + Cari dengan Nama. Silvia terpanggil karna ada “ahmad” dalam nama nya



#### Tes 2 : Sample data mahasiswa dengan fungsi Ambil 5 mahasiswa aktif dengan IPK tertinggi beserta kategorinya

|  |
| --- |
| -- Quick Test 2: Sample Data dengan Fungsi  -- Ambil 5 mahasiswa aktif dengan IPK tertinggi beserta kategorinya  SELECT      nim,      nama,      prodi,      semester,      ipk,      SALAM.kategori\_ipk(ipk) as kategori\_ipk,      status\_aktif  FROM SALAM.mahasiswas  WHERE SALAM.cek\_status\_aktif(nim) = 'Aktif'  ORDER BY ipk DESC NULLS LAST  LIMIT 5; |

Hasil :



#### Full Script 03\_buat\_fungsi.sql

|  |
| --- |
| -- 01. Fungsi sederhana untuk cek status mahasiswa  CREATE OR REPLACE FUNCTION SALAM.cek\_status\_aktif(p\_nim CHAR(10))  RETURNS VARCHAR  LANGUAGE plpgsql  AS $$  DECLARE      status\_mhs VARCHAR(20);  BEGIN      SELECT status\_aktif INTO status\_mhs      FROM SALAM.mahasiswas      WHERE nim = p\_nim;        IF status\_mhs IS NULL THEN          RETURN 'MAHASISWA TIDAK DITEMUKAN';      ELSE          RETURN status\_mhs;      END IF;  END;  $$;  -- 02. Fungsi hitung mahasiswa per prodi  CREATE OR REPLACE FUNCTION SALAM.hitung\_mahasiswa\_per\_prodi(p\_prodi VARCHAR)  RETURNS INTEGER  LANGUAGE plpgsql  AS $$  DECLARE      total INTEGER;  BEGIN      SELECT COUNT(\*) INTO total      FROM SALAM.mahasiswas      WHERE prodi = p\_prodi      AND status\_aktif = 'Aktif';        RETURN total;  END;  $$;  -- 03. Fungsi kategorisasi IPK  CREATE OR REPLACE FUNCTION SALAM.kategori\_ipk(p\_ipk NUMERIC)  RETURNS VARCHAR  LANGUAGE plpgsql  AS $$  BEGIN      IF p\_ipk IS NULL THEN          RETURN 'BELUM ADA IPK';      ELSIF p\_ipk >= 3.5 THEN          RETURN 'SANGAT MEMUASKAN';      ELSIF p\_ipk >= 3.0 THEN          RETURN 'MEMUASKAN';      ELSIF p\_ipk >= 2.5 THEN          RETURN 'CUKUP';      ELSIF p\_ipk >= 2.0 THEN          RETURN 'KURANG';      ELSE          RETURN 'SANGAT KURANG';      END IF;  END;  $$;  -- 04. Fungsi pencarian sederhana  CREATE OR REPLACE FUNCTION SALAM.cari\_mahasiswa\_sederhana(p\_keyword VARCHAR)  RETURNS TABLE(      nim CHAR(10),      nama VARCHAR(100),      prodi VARCHAR(100),      semester INTEGER,      ipk NUMERIC,      status\_aktif VARCHAR(20)  )  LANGUAGE plpgsql  AS $$  BEGIN      RETURN QUERY      SELECT          m.nim, m.nama, m.prodi, m.semester, m.ipk, m.status\_aktif      FROM SALAM.mahasiswas m      WHERE m.nim LIKE '%' || p\_keyword || '%'         OR m.nama ILIKE '%' || p\_keyword || '%'      ORDER BY m.nama;  END;  $$; |

#### Full Script 03.1\_quick\_test\_fungsi.sql

|  |
| --- |
| -- =============================================================================  -- QUICK TEST - FUNGSI SEDERHANA  -- =============================================================================  SET search\_path TO SALAM;  -- Quick Test 1: Fungsi Dasar  SELECT 'QUICK TEST RESULTS:' as test\_header;  -- 01. Test cek\_status\_aktif  -- test dengan data ada  SELECT      nim as test\_nim,      SALAM.cek\_status\_aktif(nim) as status\_dicek,      '✓' as hasil  FROM SALAM.mahasiswas  WHERE nim = '1247050005'  UNION ALL  -- Test dengan data tidak ada  SELECT      '0000000000' as test\_nim,      SALAM.cek\_status\_aktif('0000000000') as status\_dicek,      '✓' as hasil;  UNION ALL  -- 02. Test hitung\_mahasiswa\_per\_prodi  SELECT      '2. hitung\_mahasiswa\_prodi',      SALAM.hitung\_mahasiswa\_per\_prodi('Informatika')::VARCHAR,      CASE WHEN SALAM.hitung\_mahasiswa\_per\_prodi('Informatika') >= 0 THEN 'PASS' ELSE 'FAIL' END  UNION ALL  -- 03. Test kategori\_ipk  SELECT      ipk\_test as nilai\_ipk,      SALAM.kategori\_ipk(ipk\_test) as kategori,      '✓' as hasil  FROM (VALUES      (NULL), (1.75), (2.25), (2.75), (3.25), (3.75)  ) AS test\_ipk(ipk\_test);  -- Ambil 3 mahasiswa dengan IPK tertinggi beserta kategorinya  SELECT      nim,      nama,      ipk,      SALAM.kategori\_ipk(ipk) as kategori\_ipk,      '✓' as hasil  FROM SALAM.mahasiswas  WHERE ipk IS NOT NULL  ORDER BY ipk DESC  LIMIT 3;  UNION ALL  -- 04. Test cari\_mahasiswa\_sederhana  -- Test pencarian dengan NIM  SELECT \* FROM SALAM.cari\_mahasiswa\_sederhana('123');  -- Test pencarian dengan nama  SELECT \* FROM SALAM.cari\_mahasiswa\_sederhana('Ahmad');  -- Quick Test 2: Sample Data dengan Fungsi  -- Ambil 5 mahasiswa aktif dengan IPK tertinggi beserta kategorinya  SELECT      nim,      nama,      prodi,      semester,      ipk,      SALAM.kategori\_ipk(ipk) as kategori\_ipk,      status\_aktif  FROM SALAM.mahasiswas  WHERE SALAM.cek\_status\_aktif(nim) = 'Aktif'  ORDER BY ipk DESC NULLS LAST  LIMIT 5; |