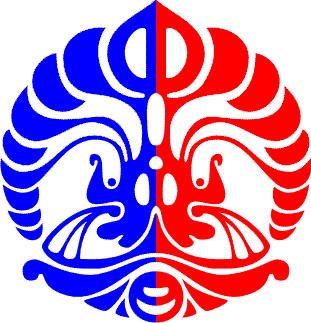
**TUGAS KELOMPOK BASIS DATA**

**LAPORAN TAHAP I**

**IMPLEMENTASI BASIS DATA**



**Disusun oleh:**

Arri Kurniawan 1606954735

Budi Indrawan 1606954754

Fadly Muhammad Ridho 1606954792

Riyandi Eko Pambudi 1406649403

**Program Studi S1 Ekstensi Sistem Informasi**

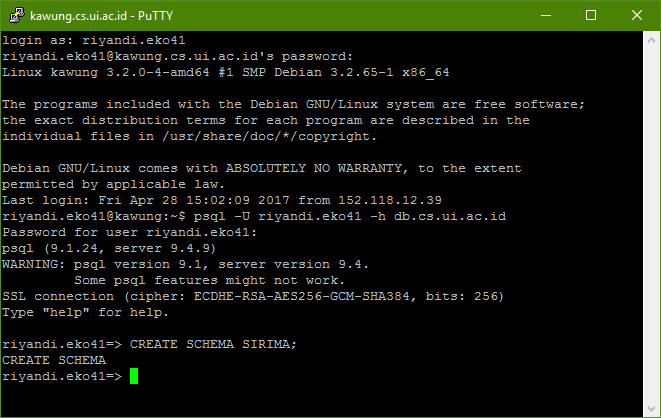
**Fakultas Ilmu Komputer**

**Universitas Indonesia**

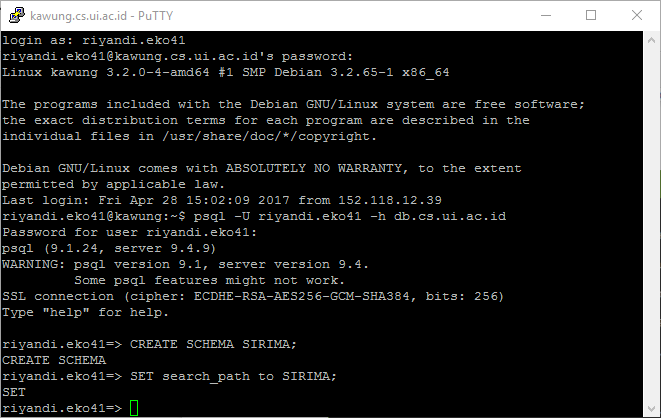
**2017**

1. **DDL script table**

CREATE SCHEMA SIRIMA;



SET search\_path to SIRIMA;



CREATE TABLE AKUN (

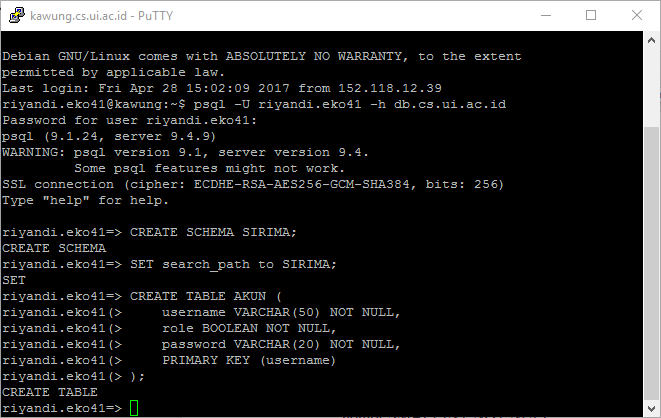
username VARCHAR(50) NOT NULL,

role BOOLEAN NOT NULL,

password VARCHAR(20) NOT NULL,

PRIMARY KEY (username)

);



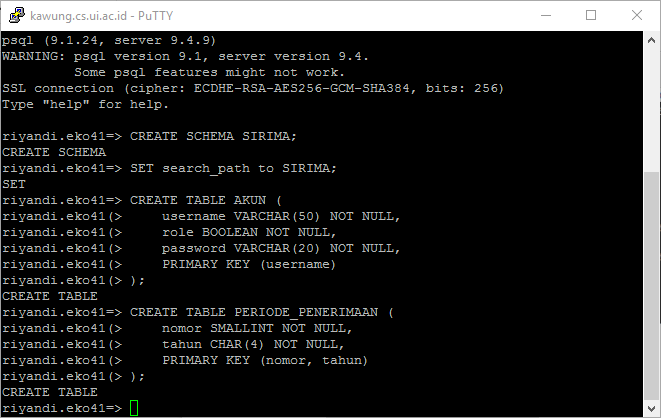
CREATE TABLE PERIODE\_PENERIMAAN (

nomor SMALLINT NOT NULL,

tahun CHAR(4) NOT NULL,

PRIMARY KEY (nomor, tahun)

);



CREATE TABLE JENJANG (

nama CHAR(2) NOT NULL,

PRIMARY KEY (nama)

);



CREATE TABLE JADWAL\_PENTING (

nomor SMALLINT NOT NULL,

tahun CHAR(4) NOT NULL,

jenjang CHAR(2) NOT NULL,

waktu\_mulai TIMESTAMP NOT NULL,

waktu\_selesai TIMESTAMP NOT NULL,

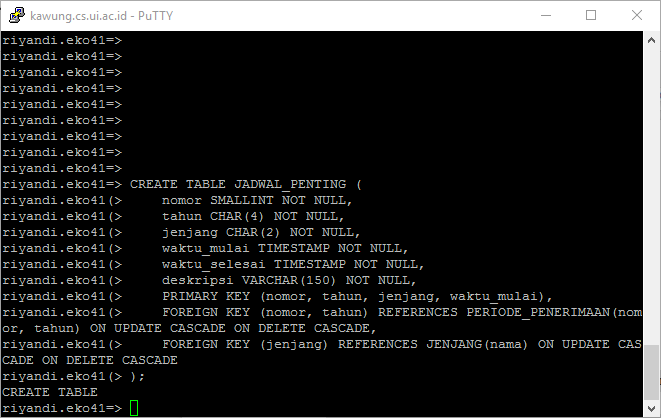
deskripsi VARCHAR(150) NOT NULL,

PRIMARY KEY (nomor, tahun, jenjang, waktu\_mulai),

FOREIGN KEY (nomor, tahun) REFERENCES PERIODE\_PENERIMAAN(nomor, tahun) ON UPDATE CASCADE ON DELETE CASCADE,

FOREIGN KEY (jenjang) REFERENCES JENJANG(nama) ON UPDATE CASCADE ON DELETE CASCADE

);



CREATE TABLE PROGRAM\_STUDI (

kode SERIAL NOT NULL,

nama VARCHAR(100) NOT NULL,

jenis\_kelas VARCHAR(50) NOT NULL,

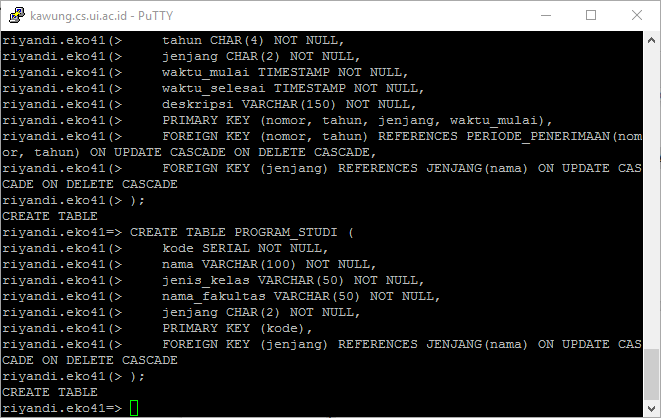
nama\_fakultas VARCHAR(50) NOT NULL,

jenjang CHAR(2) NOT NULL,

PRIMARY KEY (kode),

FOREIGN KEY (jenjang) REFERENCES JENJANG(nama) ON UPDATE CASCADE ON DELETE CASCADE

);



CREATE TABLE PENERIMAAN\_PRODI (

nomor\_periode INT NOT NULL,

tahun\_periode CHAR(4) NOT NULL,

kode\_prodi INT NOT NULL,

kuota INT NOT NULL,

jumlah\_pelamar INT,

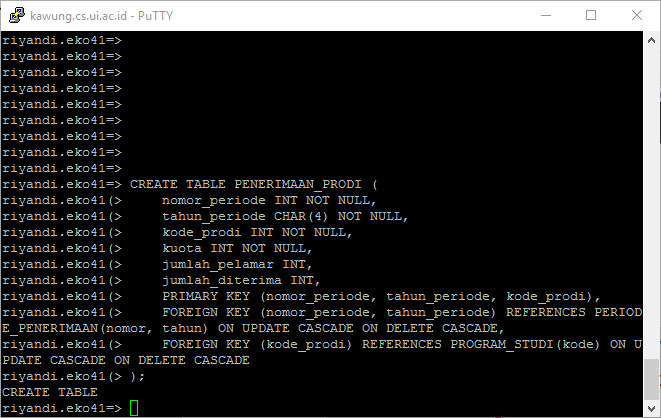
jumlah\_diterima INT,

PRIMARY KEY (nomor\_periode, tahun\_periode, kode\_prodi),

FOREIGN KEY (nomor\_periode, tahun\_periode) REFERENCES PERIODE\_PENERIMAAN(nomor, tahun) ON UPDATE CASCADE ON DELETE CASCADE,

FOREIGN KEY (kode\_prodi) REFERENCES PROGRAM\_STUDI(kode) ON UPDATE CASCADE ON DELETE CASCADE

);



CREATE TABLE PELAMAR (

username VARCHAR(50) NOT NULL,

nama\_lengkap VARCHAR(100) NOT NULL,

alamat TEXT NOT NULL,

jenis\_kelamin CHAR(1) NOT NULL,

tanggal\_lahir DATE NOT NULL,

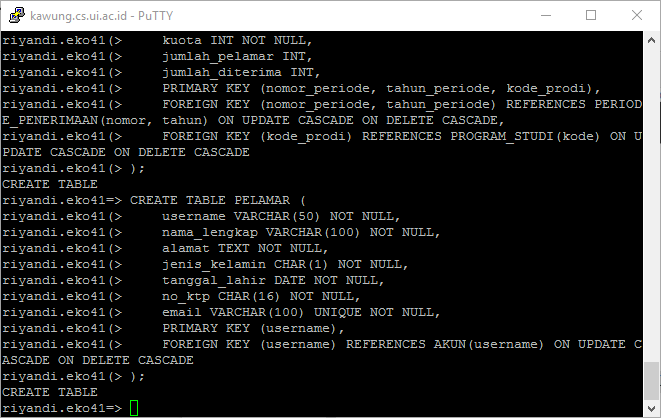
no\_ktp CHAR(16) NOT NULL,

email VARCHAR(100) UNIQUE NOT NULL,

PRIMARY KEY (username),

FOREIGN KEY (username) REFERENCES AKUN(username) ON UPDATE CASCADE ON DELETE CASCADE

);



CREATE TABLE PENDAFTARAN (

id SERIAL NOT NULL,

status\_lulus BOOLEAN NOT NULL DEFAULT FALSE,

status\_verifikasi BOOLEAN NOT NULL DEFAULT FALSE,

npm CHAR(10) NOT NULL,

pelamar VARCHAR(50) NOT NULL,

nomor\_periode SMALLINT NOT NULL,

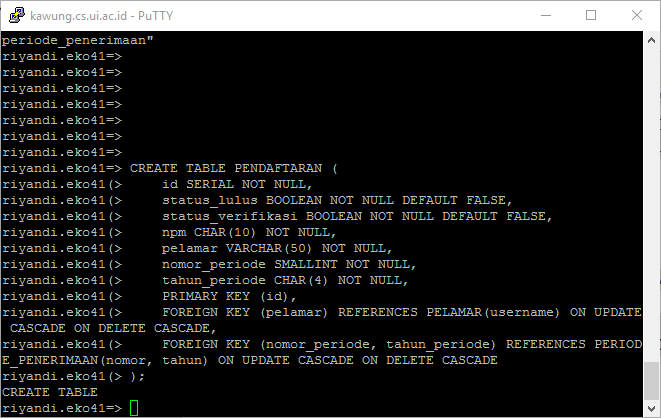
tahun\_periode CHAR(4) NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (pelamar) REFERENCES PELAMAR(username) ON UPDATE CASCADE ON DELETE CASCADE,

FOREIGN KEY (nomor\_periode, tahun\_periode) REFERENCES PERIODE\_PENERIMAAN(nomor, tahun) ON UPDATE CASCADE ON DELETE CASCADE

);



CREATE TABLE PENDAFTARAN\_UUI (

id\_pendaftaran INT NOT NULL,

rapot VARCHAR(100) NOT NULL,

surat\_rekomendasi VARCHAR(100) NOT NULL,

asal\_sekolah VARCHAR(100) NOT NULL,

jenis\_sma VARCHAR(50) NOT NULL,

alamat\_sekolah TEXT NOT NULL,

nisn VARCHAR(10) NOT NULL,

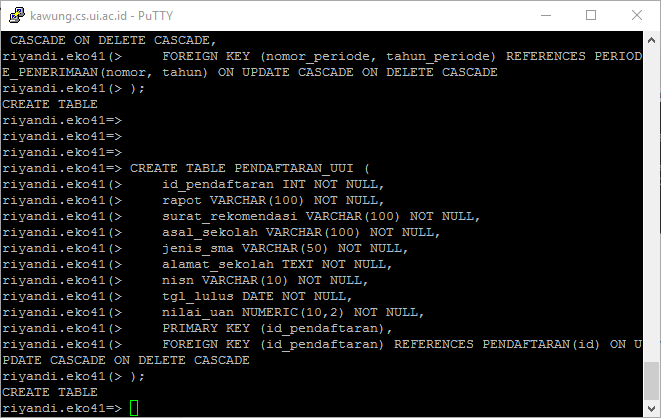
tgl\_lulus DATE NOT NULL,

nilai\_uan NUMERIC(10,2) NOT NULL,

PRIMARY KEY (id\_pendaftaran),

FOREIGN KEY (id\_pendaftaran) REFERENCES PENDAFTARAN(id) ON UPDATE CASCADE ON DELETE CASCADE

);



CREATE TABLE REKOMENDASI (

tgl\_review DATE NOT NULL,

id\_pendaftaran INT NOT NULL,

status BOOLEAN NOT NULL,

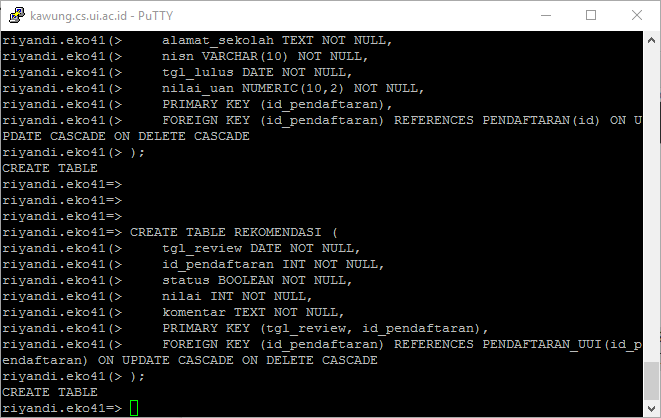
nilai INT NOT NULL,

komentar TEXT NOT NULL,

PRIMARY KEY (tgl\_review, id\_pendaftaran),

FOREIGN KEY (id\_pendaftaran) REFERENCES PENDAFTARAN\_UUI(id\_pendaftaran) ON UPDATE CASCADE ON DELETE CASCADE

);



CREATE TABLE LOKASI\_UJIAN (

kota VARCHAR(100) NOT NULL,

tempat VARCHAR(150) NOT NULL,

nomor\_periode SMALLINT NOT NULL,

tahun\_periode CHAR(4) NOT NULL,

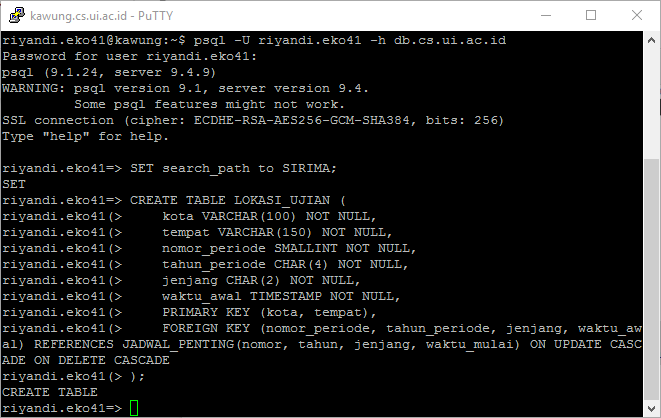
jenjang CHAR(2) NOT NULL,

waktu\_awal TIMESTAMP NOT NULL,

PRIMARY KEY (kota, tempat),

FOREIGN KEY (nomor\_periode, tahun\_periode, jenjang, waktu\_awal) REFERENCES JADWAL\_PENTING(nomor, tahun, jenjang, waktu\_mulai) ON UPDATE CASCADE ON DELETE CASCADE

);



CREATE TABLE PENDAFTARAN\_SEMAS (

id\_pendaftaran INT NOT NULL,

status\_hadir BOOLEAN NOT NULL,

nilai\_ujian INT NOT NULL,

no\_kartu\_ujian CHAR(10) NOT NULL,

lokasi\_kota VARCHAR(100) NOT NULL,

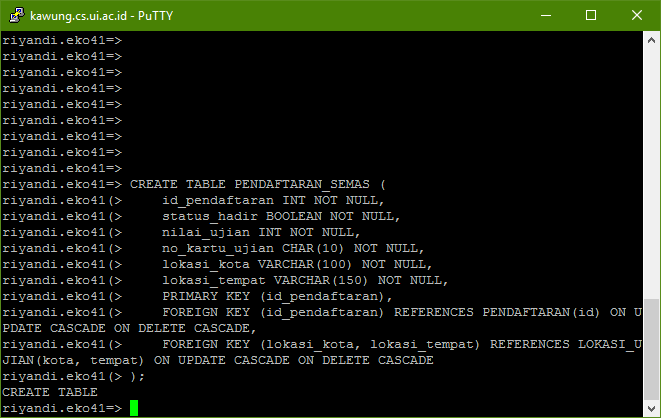
lokasi\_tempat VARCHAR(150) NOT NULL,

PRIMARY KEY (id\_pendaftaran),

FOREIGN KEY (id\_pendaftaran) REFERENCES PENDAFTARAN(id) ON UPDATE CASCADE ON DELETE CASCADE,

FOREIGN KEY (lokasi\_kota, lokasi\_tempat) REFERENCES LOKASI\_UJIAN(kota, tempat) ON UPDATE CASCADE ON DELETE CASCADE

);



CREATE TABLE PENDAFTARAN\_SEMAS\_SARJANA (

id\_pendaftaran INT NOT NULL,

asal\_sekolah VARCHAR(100) NOT NULL,

jenis\_sma VARCHAR(50) NOT NULL,

alamat\_sekolah TEXT NOT NULL,

nisn VARCHAR(10) NOT NULL,

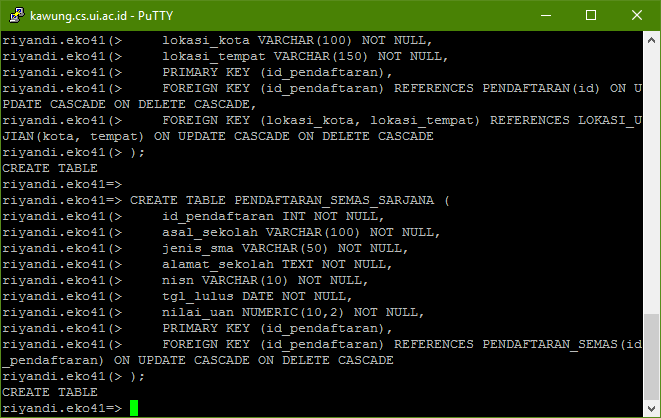
tgl\_lulus DATE NOT NULL,

nilai\_uan NUMERIC(10,2) NOT NULL,

PRIMARY KEY (id\_pendaftaran),

FOREIGN KEY (id\_pendaftaran) REFERENCES PENDAFTARAN\_SEMAS(id\_pendaftaran) ON UPDATE CASCADE ON DELETE CASCADE

);



CREATE TABLE PENDAFTARAN\_SEMAS\_PASCASARJANA (

id\_pendaftaran INT NOT NULL,

nilai\_tpa INT NOT NULL,

nilai\_toefl INT NOT NULL,

jenjang\_terakhir CHAR(2) NOT NULL,

asal\_univ VARCHAR(100) NOT NULL,

alamat\_univ TEXT NOT NULL,

prodi\_terakhir VARCHAR(10) NOT NULL,

nilai\_ipk NUMERIC(10,2) NOT NULL,

no\_ijazah VARCHAR(50) NOT NULL,

tgl\_lulus DATE NOT NULL,

jenjang CHAR(2) NOT NULL,

nama\_rekomender VARCHAR(100),

prop\_penelitian VARCHAR(100),

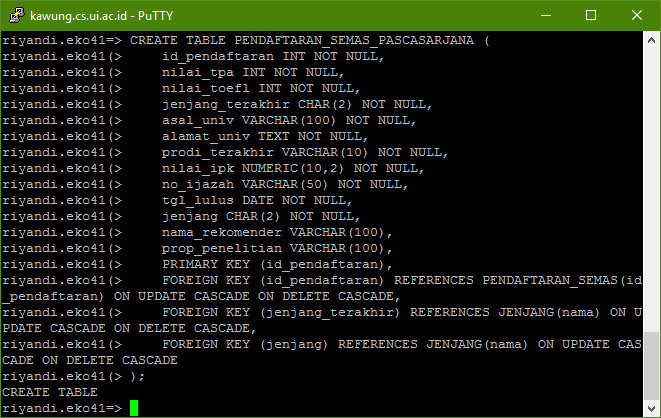
PRIMARY KEY (id\_pendaftaran),

FOREIGN KEY (id\_pendaftaran) REFERENCES PENDAFTARAN\_SEMAS(id\_pendaftaran) ON UPDATE CASCADE ON DELETE CASCADE,

FOREIGN KEY (jenjang\_terakhir) REFERENCES JENJANG(nama) ON UPDATE CASCADE ON DELETE CASCADE,

FOREIGN KEY (jenjang) REFERENCES JENJANG(nama) ON UPDATE CASCADE ON DELETE CASCADE

);



CREATE TABLE PEMBAYARAN (

id SERIAL NOT NULL,

waktu\_bayar TIMESTAMP NOT NULL,

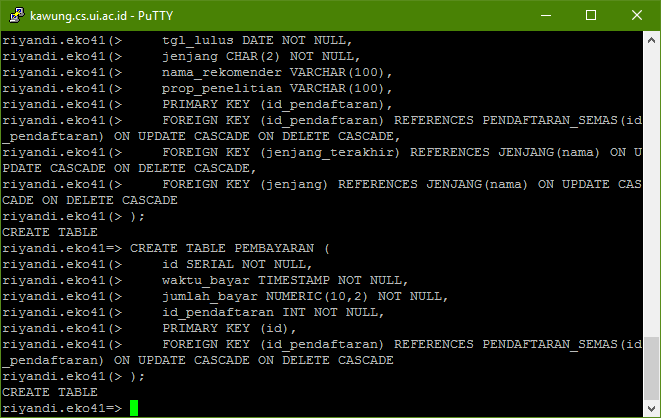
jumlah\_bayar NUMERIC(10,2) NOT NULL,

id\_pendaftaran INT NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (id\_pendaftaran) REFERENCES PENDAFTARAN\_SEMAS(id\_pendaftaran) ON UPDATE CASCADE ON DELETE CASCADE

);



CREATE TABLE RUANG\_UJIAN (

kota VARCHAR(100) NOT NULL,

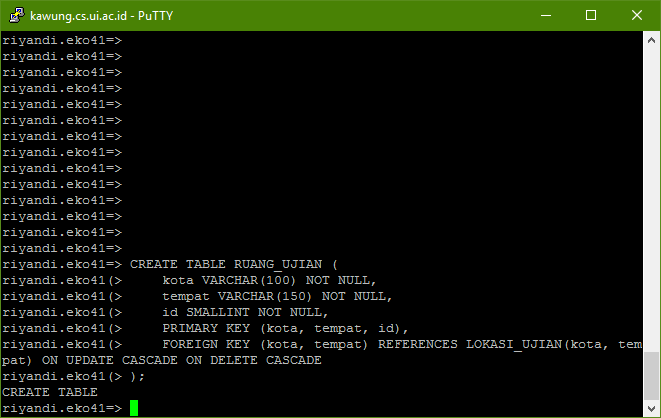
tempat VARCHAR(150) NOT NULL,

id SMALLINT NOT NULL,

PRIMARY KEY (kota, tempat, id),

FOREIGN KEY (kota, tempat) REFERENCES LOKASI\_UJIAN(kota, tempat) ON UPDATE CASCADE ON DELETE CASCADE

);



CREATE TABLE PENGAWAS (

nomor\_induk VARCHAR(16) NOT NULL,

nama VARCHAR(100) NOT NULL,

no\_telp TEXT NOT NULL,

lokasi\_kota VARCHAR(100) NOT NULL,

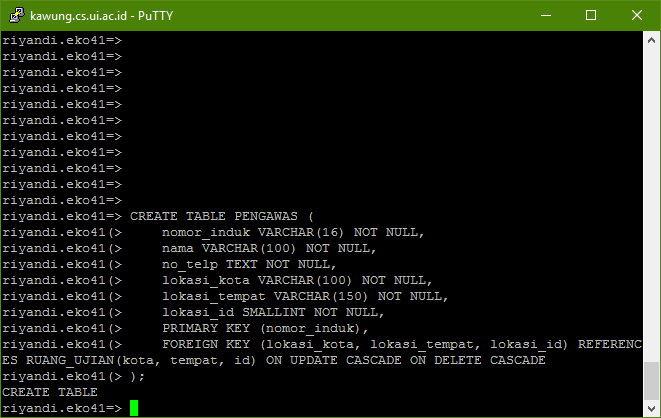
lokasi\_tempat VARCHAR(150) NOT NULL,

lokasi\_id SMALLINT NOT NULL,

PRIMARY KEY (nomor\_induk),

FOREIGN KEY (lokasi\_kota, lokasi\_tempat, lokasi\_id) REFERENCES RUANG\_UJIAN(kota, tempat, id) ON UPDATE CASCADE ON DELETE CASCADE

);



CREATE TABLE PENDAFTARAN\_PRODI (

id\_pendaftaran INT NOT NULL,

kode\_prodi INT NOT NULL,

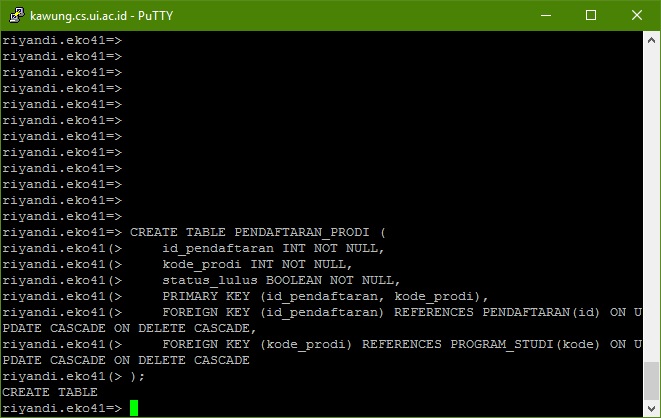
status\_lulus BOOLEAN NOT NULL,

PRIMARY KEY (id\_pendaftaran, kode\_prodi),

FOREIGN KEY (id\_pendaftaran) REFERENCES PENDAFTARAN(id) ON UPDATE CASCADE ON DELETE CASCADE,

FOREIGN KEY (kode\_prodi) REFERENCES PROGRAM\_STUDI(kode) ON UPDATE CASCADE ON DELETE CASCADE

);



1. **DDL script trigger/stored procedure**

CREATE OR REPLACE FUNCTION hitung\_pelamar()

RETURNS trigger AS

$$

BEGIN

IF (TG\_OP = 'INSERT') THEN

UPDATE PENERIMAAN\_PRODI PNP

SET jumlah\_pelamar = (coalesce(jumlah\_pelamar,0) + 1)

FROM PERIODE\_PENERIMAAN PRP

JOIN PENDAFTARAN P

ON P.NOMOR\_PERIODE = PRP.NOMOR

AND P.TAHUN\_PERIODE = PRP.TAHUN

WHERE PRP.NOMOR = PNP.NOMOR\_PERIODE

AND PRP.TAHUN = PNP.TAHUN\_PERIODE

AND P.ID = NEW.ID\_PENDAFTARAN

AND PNP.KODE\_PRODI = NEW.kode\_prodi;

ELSEIF (TG\_OP = 'UPDATE') THEN

UPDATE PENERIMAAN\_PRODI PNP

SET jumlah\_pelamar = (coalesce(jumlah\_pelamar,0) + 1)

FROM PERIODE\_PENERIMAAN PRP

JOIN PENDAFTARAN P

ON P.NOMOR\_PERIODE = PRP.NOMOR

AND P.TAHUN\_PERIODE = PRP.TAHUN

WHERE PRP.NOMOR = PNP.NOMOR\_PERIODE

AND PRP.TAHUN = PNP.TAHUN\_PERIODE

AND P.ID = NEW.ID\_PENDAFTARAN

AND PNP.KODE\_PRODI = NEW.kode\_prodi;

UPDATE PENERIMAAN\_PRODI PNP

SET jumlah\_pelamar = (jumlah\_pelamar - 1)

FROM PERIODE\_PENERIMAAN PRP

JOIN PENDAFTARAN P

ON P.NOMOR\_PERIODE = PRP.NOMOR

AND P.TAHUN\_PERIODE = PRP.TAHUN

WHERE PRP.NOMOR = PNP.NOMOR\_PERIODE

AND PRP.TAHUN = PNP.TAHUN\_PERIODE

AND P.ID = OLD.ID\_PENDAFTARAN

AND PNP.KODE\_PRODI = OLD.kode\_prodi;

ELSEIF (TG\_OP = 'DELETE') THEN

UPDATE PENERIMAAN\_PRODI PNP

SET jumlah\_pelamar = (jumlah\_pelamar - 1)

FROM PERIODE\_PENERIMAAN PRP

JOIN PENDAFTARAN P

ON P.NOMOR\_PERIODE = PRP.NOMOR

AND P.TAHUN\_PERIODE = PRP.TAHUN

WHERE PRP.NOMOR = PNP.NOMOR\_PERIODE

AND PRP.TAHUN = PNP.TAHUN\_PERIODE

AND P.ID = OLD.ID\_PENDAFTARAN

AND PNP.KODE\_PRODI = OLD.kode\_prodi;

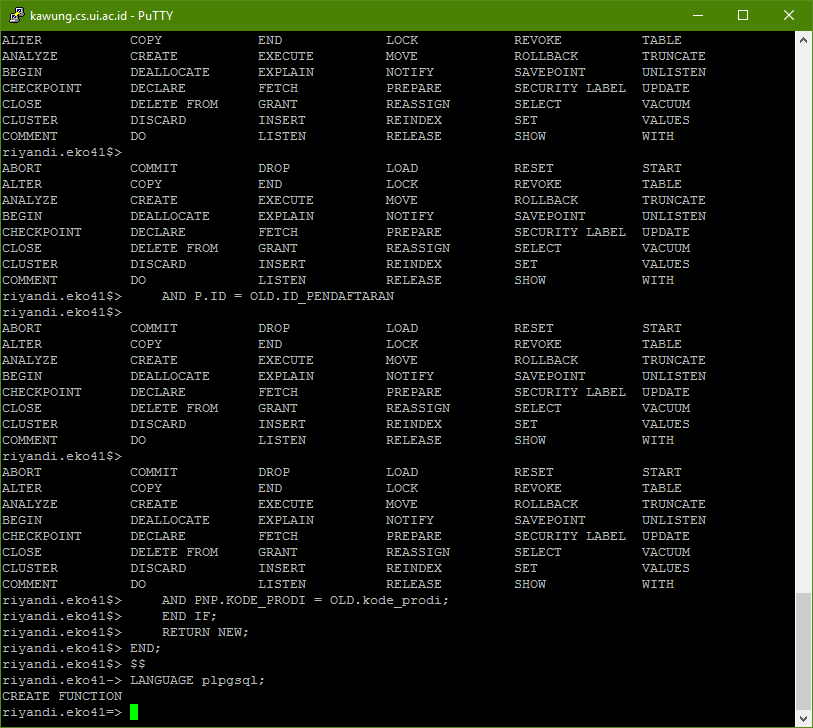
END IF;

RETURN NEW;

END;

$$

LANGUAGE plpgsql;

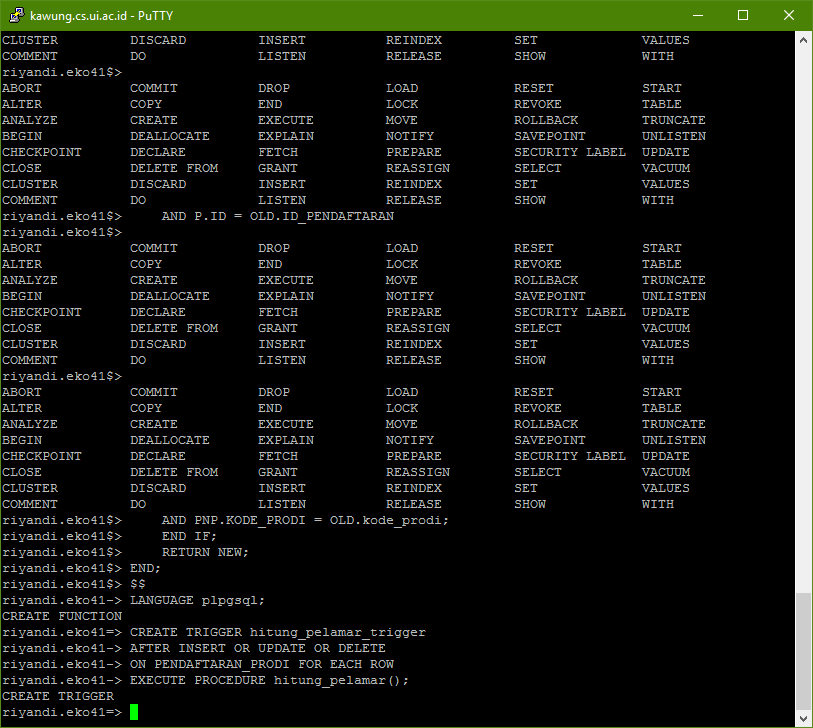


CREATE TRIGGER hitung\_pelamar\_trigger

AFTER INSERT OR UPDATE OR DELETE

ON PENDAFTARAN\_PRODI FOR EACH ROW

EXECUTE PROCEDURE hitung\_pelamar();



CREATE OR REPLACE FUNCTION hitung\_pelamar\_diterima()

RETURNS trigger AS

$$

DECLARE

jumlah\_lulus INT;

BEGIN

SELECT COUNT(PP.status\_lulus) INTO jumlah\_lulus

FROM PENDAFTARAN\_PRODI PP

WHERE PP.ID\_PENDAFTARAN = NEW.ID\_PENDAFTARAN

AND PP.status\_lulus = TRUE;

IF (TG\_OP = 'UPDATE') THEN

IF(NEW.status\_lulus IS TRUE AND OLD.status\_lulus IS FALSE) THEN

IF(jumlah\_lulus = 1) THEN

UPDATE PENERIMAAN\_PRODI PNP

SET jumlah\_diterima = (coalesce(jumlah\_diterima,0) + 1)

FROM PERIODE\_PENERIMAAN PRP

JOIN PENDAFTARAN P

ON P.NOMOR\_PERIODE = PRP.NOMOR

AND P.TAHUN\_PERIODE = PRP.TAHUN

WHERE PRP.NOMOR = PNP.NOMOR\_PERIODE

AND PRP.TAHUN = PNP.TAHUN\_PERIODE

AND P.ID = NEW.ID\_PENDAFTARAN

AND PNP.KODE\_PRODI = NEW.kode\_prodi;

ELSE

RAISE EXCEPTION 'Error : status\_lulus untuk id\_pendaftaran yang sama maksimal berjumlah 1';

END IF;

ELSEIF(NEW.status\_lulus IS FALSE AND OLD.status\_lulus IS TRUE) THEN

UPDATE PENERIMAAN\_PRODI PNP

SET jumlah\_diterima = (jumlah\_diterima - 1)

FROM PERIODE\_PENERIMAAN PRP

JOIN PENDAFTARAN P

ON P.NOMOR\_PERIODE = PRP.NOMOR

AND P.TAHUN\_PERIODE = PRP.TAHUN

WHERE PRP.NOMOR = PNP.NOMOR\_PERIODE

AND PRP.TAHUN = PNP.TAHUN\_PERIODE

AND P.ID = NEW.ID\_PENDAFTARAN

AND PNP.KODE\_PRODI = NEW.kode\_prodi;

END IF;

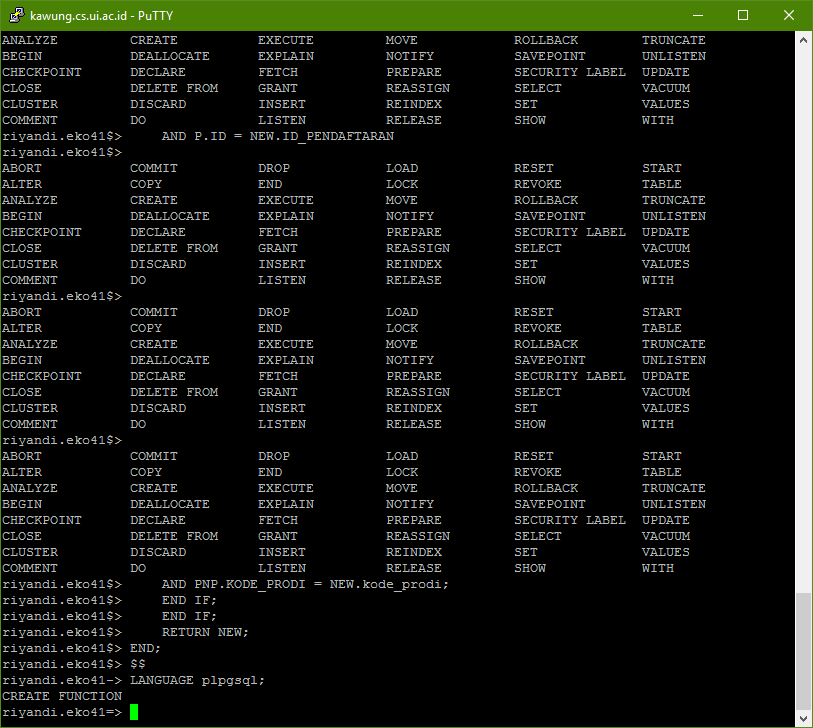
END IF;

RETURN NEW;

END;

$$

LANGUAGE plpgsql;

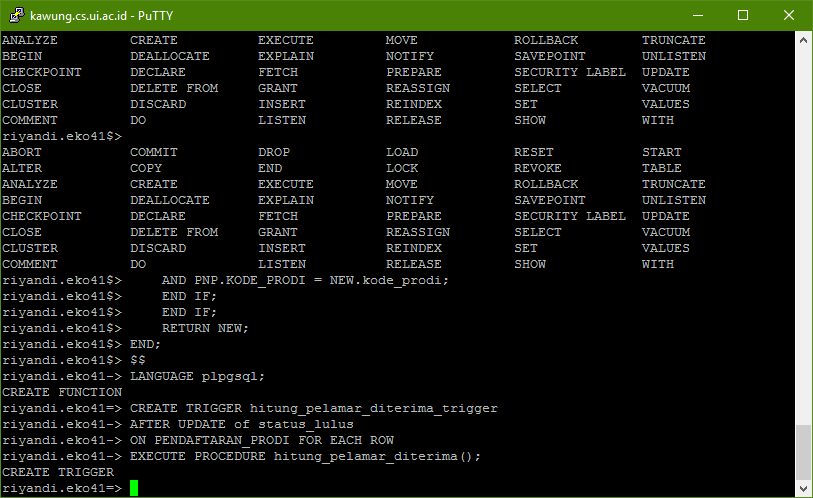


CREATE TRIGGER hitung\_pelamar\_diterima\_trigger

AFTER UPDATE of status\_lulus

ON PENDAFTARAN\_PRODI FOR EACH ROW

EXECUTE PROCEDURE hitung\_pelamar\_diterima();



1. **Table state**

\d AKUN

\d JADWAL\_PENTING

\d JENJANG

\d LOKASI\_UJIAN

\d PELAMAR

\d PEMBAYARAN

\d PENDAFTARAN

\d PENDAFTARAN\_PRODI

\d PENDAFTARAN\_SEMAS

\d PENDAFTARAN\_SEMAS\_PASCASARJANA

\d PENDAFTARAN\_SEMAS\_SARJANA

\d PENDAFTARAN\_UUI

\d PENERIMAAN\_PRODI

\d PENGAWAS

\d PERIODE\_PENERIMAAN

\d PROGRAM\_STUDI

\d REKOMENDASI

\d RUANG\_UJIAN

1. **Pembagian tugas dan kontribusi**

|  |  |  |  |
| --- | --- | --- | --- |
| **NPM** | **Nama** | **Deskripsi Tugas** | **Kontribusi** |
| 1606954735 | Arri Kurniawan | Membuat script DDL untuk create table, stored procedure dan trigger | 25% |
| 1606954754 | Budi Indrawan | Membuat script DDL untuk insert query dan program untuk generate data acak (9 table) | 25% |
| 1606954792 | Fadly Muhammad Ridho | Membuat script DDL untuk insert query dan program untuk generate data acak (9 table) | 25% |
| 1406649403 | Riyandi Eko Pambudi | Menyusun laporan akhir, melakukan koreksi query serta eksekusi script DDL ke dalam sistem | 25% |