

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **4.1 Analisis dan Pengumpulan Kebutuhan**

Berdasarkan hasil wawancara yang dilakukan pada Ibu Aprilia Kurnianti yang bertanggung jawab langsung pada perhitungan penggajian, data-data yang dibutuhkan berupa data-data dari para dosen, instruktur dan asisten dosen seperti nip, nama, alamat, nohp, email serta passwordnya. Serta diperlukannya data presensi yang berisikan nama pengajar, mata kuliah yang diajar dan sesi pertemuan. Lalu data penggajian yang akan dihitung dari banyaknya presensi kehadiran.

Wawancara yang dilakukan pada Saudara Andhy Kurniawan yang bertanggung jawab pada bagian presensi kurang lebihnya mengatakan hal yang sama. Seperti dibutuhkannya beberapa data dari dosen, instruktur dan asisten. Lalu data dari presensi yang memiliki batas maksimal dua puluh empat pertemuan, data-data mata kuliah dan data penggajian.

Berdasarkan hasil wawancara yang dilakukan kepada dua narasumber diatas, dapat disimpulkan bahwa data-data yang dibutuhkan dalam pembuatan *database* penggajian berupa data dosen, instruktur dan asisten dosen, data mata kuliah, data presensi dan data penggajian.

#### **4.2 Perancangan *Database* Konseptual**

Tahapan yang dilakukan pada perancangan *database* konseptual diawali melalui tahapan normalisasi. Adapun tabel yang di normalisasikan berdasarkan desain awal tabel yaitu tabel data gaji.

## 4.2.1 Normalisasi

### 1. Normalisasi Tabel Data Gaji

Berikut ini adalah desain tabel awal:

**Tabel 4. 1** Desain awal tabel Penggajian Staf

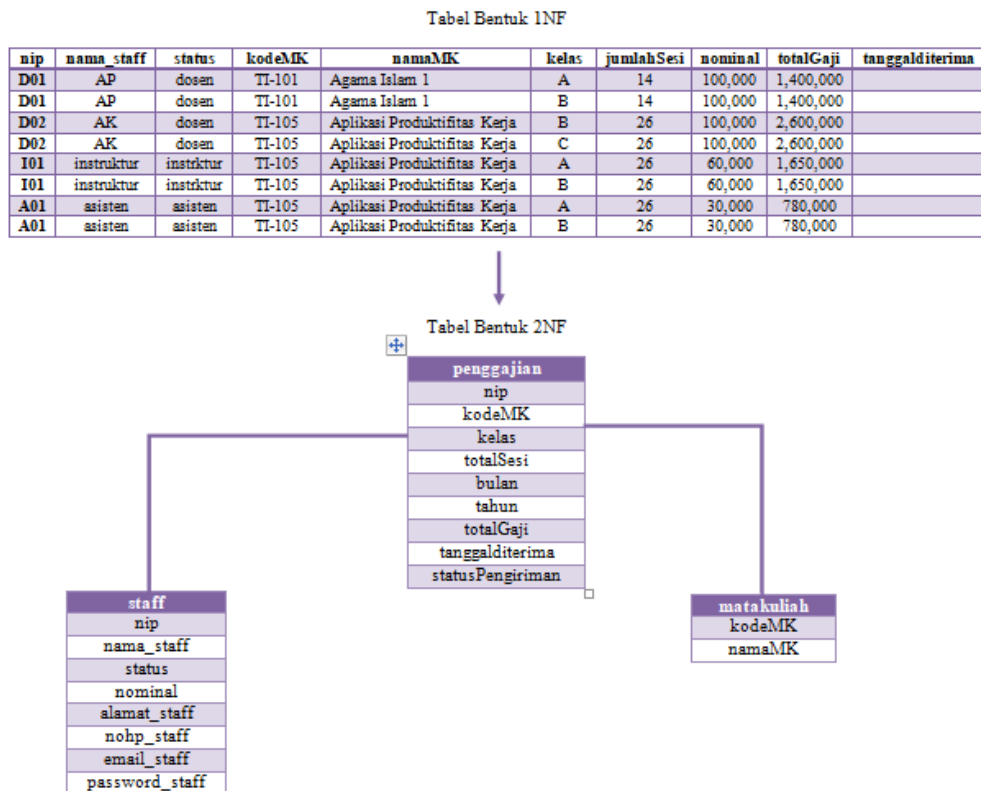
nip	nama	status	kode mk	nama mk	kelas	total sesi	nominal	total gaji	tgl terima
D01	AP	dosen	TI-101	Agama Islam 1	A	14	100,000	1,400,000	
					B	14	100,000	1,400,000	
D02	AK	dosen	TI-105	Aplikasi Produktifitas Kerja	B	26	100,000	2,600,000	
					C	26	100,000	2,600,000	
I01	Instruktur	instruktur	TI-105	Aplikasi Produktifitas Kerja	A	26	60,000	1,560,000	
					B	26	60,000	1,560,000	
A01	Asisten	Asisten	TI-105	Aplikasi Produktifitas Kerja	A	26	30,000	780,000	
					B	26	30,000	780,000	

Pada tabel tabel 4.1 masih terdapat data ganda, maka perlu dilakukannya normalisasi menjadi 1 NF seperti yang ditunjukkan pada Tabel 4.2

**Tabel 4. 2** Tabel Penggajian Staf Bentuk 1NF

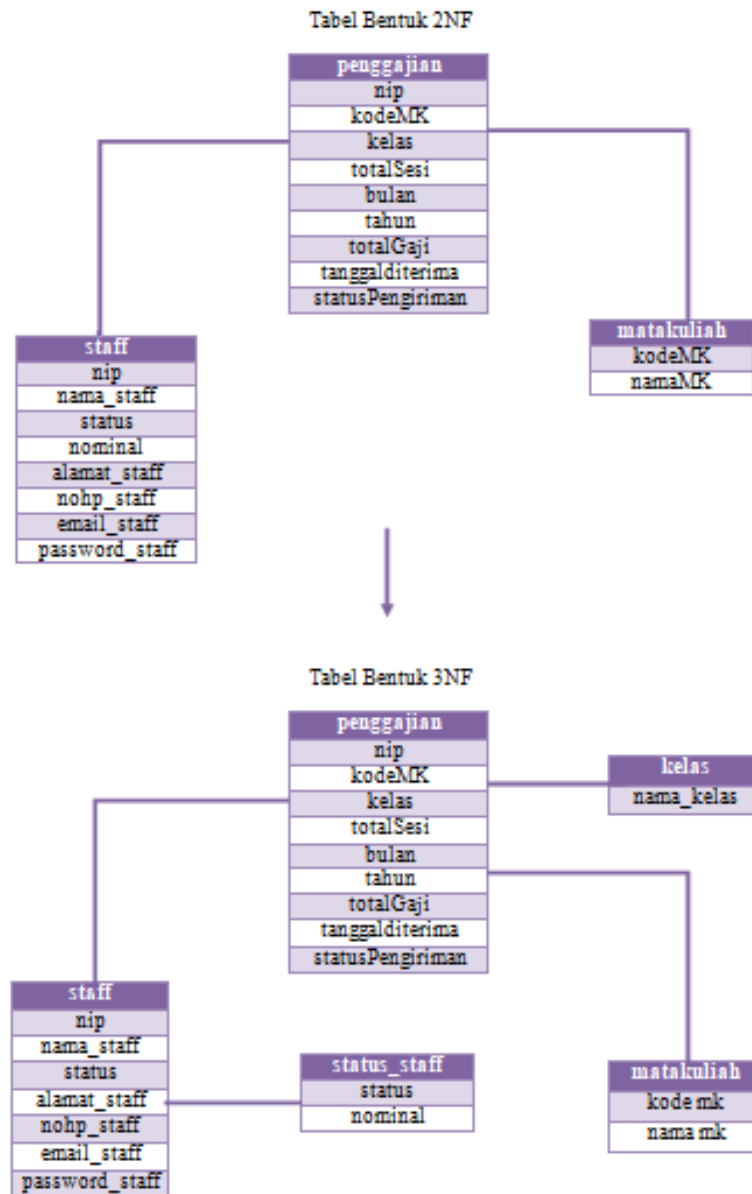
nip	nama staff	status	kodeMK	namaMK	kelas	jumlahSesi	nominal	totalGaji	tanggalditerima
D01	AP	dosen	TI-101	Agama Islam 1	A	14	100,000	1,400,000	
D01	AP	dosen	TI-101	Agama Islam 1	B	14	100,000	1,400,000	
D02	AK	dosen	TI-105	Aplikasi Produktifitas Kerja	B	26	100,000	2,600,000	
D02	AK	dosen	TI-105	Aplikasi Produktifitas Kerja	C	26	100,000	2,600,000	
I01	instruktur	instruktur	TI-105	Aplikasi Produktifitas Kerja	A	26	60,000	1,650,000	
I01	instruktur	instruktur	TI-105	Aplikasi Produktifitas Kerja	B	26	60,000	1,650,000	
A01	asisten	asisten	TI-105	Aplikasi Produktifitas Kerja	A	26	30,000	780,000	
A01	asisten	asisten	TI-105	Aplikasi Produktifitas Kerja	B	26	30,000	780,000	

Pada tabel Penggajian Staff pada Tabel 4.2 masih terdapat ketergantungan fungsional antara kolom nip, nama, status, nominal dan kolom kode mk, nama mk. Sehingga pada table Penggajian Staff dapat diubah menjadi 2NF, dengan cara memisahkan kolom yang memiliki ketergantungan fungsional, normalisasi bentuk 2NF dapat dilihat pada Gambar 4.1.



**Gambar 4. 1 Tabel Penggajian Staf Bentuk 2NF**

Pada Gambar 4.1 pada tabel Staff masih mempunyai ketergantungan fungsional transitif diantaranya pada kolom status dan nominal. Setelah dinormalisasikan ketergantungan fungsi transitif dihilangkan pada tabel Staff menjadi 3NF ditunjukkan pada gambar 4.2.



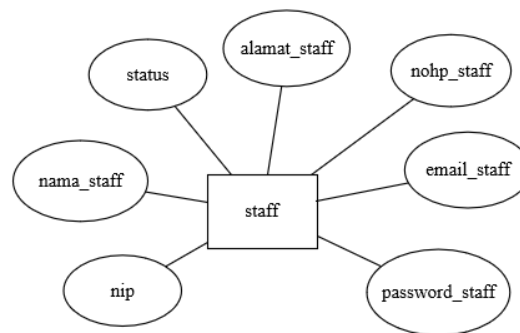
**Gambar 4. 2** Tabel Penggajian Staf Bentuk 3NF

#### 4.2.2 Entitas

Entitas merupakan tahap penentuan dan mengidentifikasi entitas yang akan digunakan. Entitas tersebut didapatkan berdasarkan hasil dari normalisasi yang sudah dilakukan sebelumnya. Berikut adalah entitas dan atribut dari sistem penggajian staff tersebut.

## 1. Entitas Staff

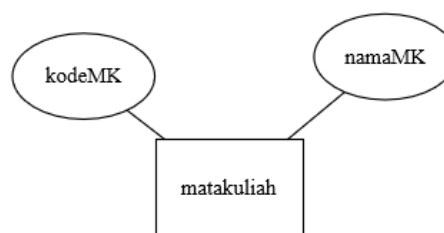
Entitas staff terdiri dari 7 atribut di antaranya nip, nama\_staff, status, alamat\_staff, nohp\_staff, email\_staff dan password\_staff. Pada atribut nip, nama\_staff, alamat\_staff dan nohp\_staff perlu dimasukkan ke dalam entitas staff, karena berfungsi untuk mengetahui informasi data dari staff. Atribut status perlu dimasukan kedalam entitas staff, karena untuk mengelompokkan status dari masing-masing staff. Sedangkan untuk atribut email\_staff dan password\_staff perlu dimasukan kedalam entitas staff karena berfungsi untuk login staff. Atribut tersebut dapat dilihat pada Gambar 4.3.



**Gambar 4. 3** Entitas Staff

## 2. Entitas Mata Kuliah

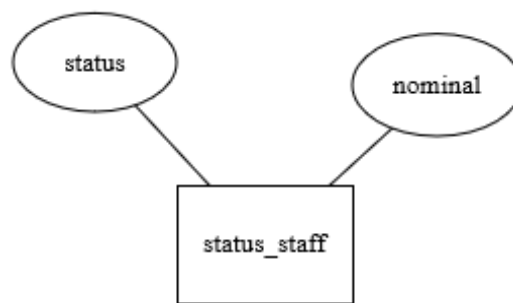
Entitas mata kuliah terdiri atas 2 atribut diantaranya adalah kodeMK dan namaMK. Pada atribut kodeMK dan namaMK perlu dimasukkan ke dalam entitas matakuliah karena berfungsi untuk mengetahui informasi dari data matakuliah tersebut. Atribut tersebut dapat dilihat pada Gambar 4.4.



**Gambar 4. 4** Entitas Mata Kuliah

### 3. Entitas Status Staff

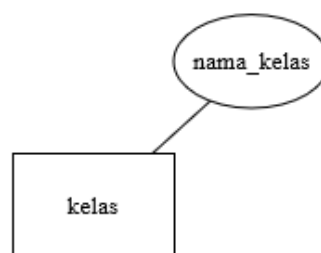
Entitas status staff terdiri atas 2 atribut di antaranya adalah status dan nominal. Pada atribut status perlu dimasukkan ke dalam entitas status staff adalah untuk membedakan status masing-masing staff dan nominal perlu dimasukkan ke dalam entitas matakuliah karena berfungsi untuk mengetahui gaji dari staff dalam satu kali pertemuan. Atribut tersebut dapat dilihat pada Gambar 4.5.



**Gambar 4. 5** Entitas Status Staff

### 4. Entitas Kelas

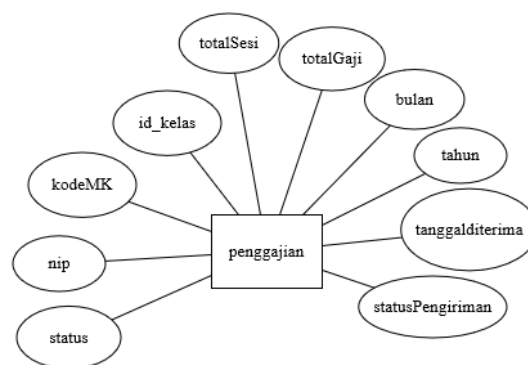
Entitas kelas hanya terdiri dari 1 atribut yaitu nama kelas. Pada atribut nama\_kelas perlu dimasukkan ke dalam entitas kelas karena berfungsi untuk membedakan kelas-kelas yang dilakukan pengajaran. Atribut tersebut dapat dilihat pada Gambar 4.6.



**Gambar 4. 6** Entitas Kelas

## 5. Entitas Penggajian

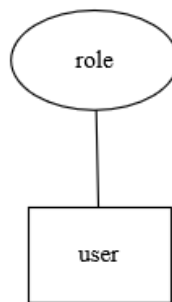
Entitas penggajian terdiri dari 10 atribut diantaranya status, nip, kodeMK, nama\_kelas, totalSesi, totalGaji, bulan, tahun, tanggalditerima dan statusPengiriman. Pada atribut status perlu dimasukkan ke dalam entitas penggajian adalah untuk mempermudah dalam mengelompokkan nip berdasarkan status yang akan ditampilkan. Atribut nip perlu dimasukkan ke dalam entitas penggajian adalah untuk menampilkan staff yang akan diberikan gaji. Atribut kodeMK di perlukan dalam entitas penggajian adalah untuk mengetahui mata kuliah apa yang diajarkan oleh staff. Atribut nama\_kelas perlu dimasukkan ke dalam entitas penggajian adalah untuk mengetahui staff mengajar pada kelas yang diajar. Atribut totalSesi perlu dimasukkan ke dalam entitas penggajian untuk mengetahui berapa banyak total pertemuan yang dilakukan oleh staff dalam mengajar. Atribut totalGaji perlu dimasukan ke dalam entitas penggajian untuk mengetahui total gaji yang didapatkan pengajar berdasarkan total sesi yang dihadiri oleh staff. Atribut bulan dan tahun perlu dimasukkan ke dalam entitas adalah untuk mempermudah dalam pendataan penggajian yang dikelompokkan berdasarkan bulan dan tahunnya. Attribute tanggalditerima dan statusPengiriman perlu dimasukkan ke dalam entitas penggajian adalah untuk mengetahui pada tanggal berapa gaji tersebut sudah dikirimkan serta status yang sudah terkirim atau belum. Atribut tersebut dapat dilihat pada Gambar 4.7.



**Gambar 4.7** Entitas Penggajian

## 6. Entitas *User*

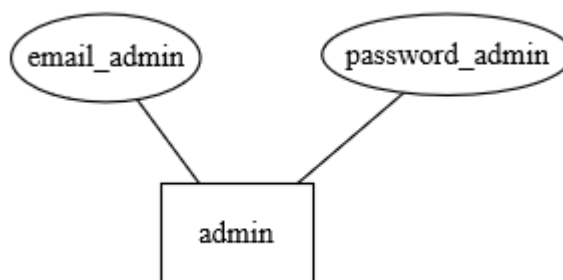
Entitas *user* terdiri dari 1 atribut diantaranya adalah atribut *role*. Atribut *role* perlu dimasukkan ke dalam entitas *user*, dimana atribut *role* berfungsi untuk mengetahui *role* dari kedua *user* yang ada di dalam *database* yaitu *staff* dan *admin*. Dengan adanya atribut *role database* dapat membagi *role* sesuai dengan fungsi masing-masing *user*. Atribut tersebut dapat dilihat pada Gambar 4.8.



**Gambar 4. 8** Entitas User

## 7. Entitas Admin

Entitas *admin* terdiri dari 2 atribut diantaranya adalah *email\_admin* dan *password\_admin*. Atribut *email\_admin* dan *password\_admin* perlu dimasukkan ke dalam entitas *admin* karena berfungsi untuk login. Atribut *admin* dapat dilihat ada Gambar 4.9.



**Gambar 4. 9** Entitas Admin

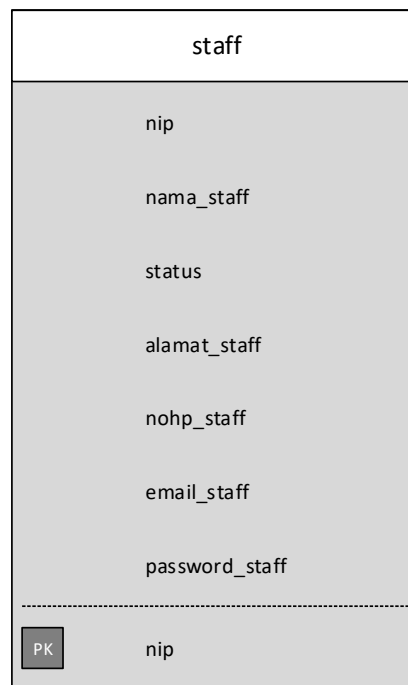


### 4.2.3 Primary Key

Pada tahap ini menentukan dan mengidentifikasi *primary key* dari setiap entitas dari Sistem Penggajian Staff. *Primary key* didapatkan dengan cara mengidentifikasi setiap atribut yang ada di dalam setiap entitas, apabila terdapat atribut yang memenuhi persyaratan yaitu *unique identifier* dan *non redundancy*, atribut tersebut menjadi *primary key*.

#### 1. Primary Key Entitas Staff

Entitas Staff memiliki atribut nip, nama\_staff, status, alamat\_staff, nohp\_staff, email\_staff dan password\_staff. Karena pada atribut nip telah memenuhi persyaratan sebagai *primary key* maka atribut nip terpilih sebagai *primary key*, entitas Staff dapat dilihat pada gambar 4.10.

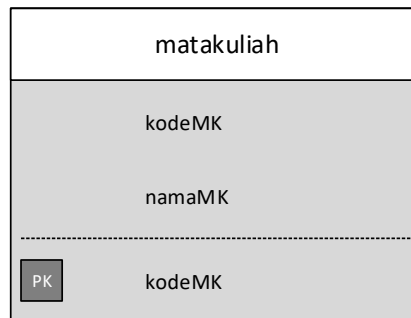


**Gambar 4. 10** Primary Key Entitas Staff

#### 2. Primary Key Entitas Mata Kuliah

Entitas Mata Kuliah memiliki attribute kodeMK dan namaMK. Karena pada atribut kodeMK telah memenuhi persyaratan sebagai *primary key* maka

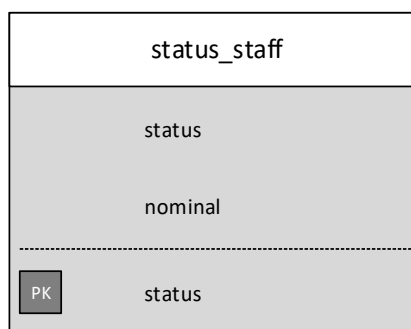
atribut kodeMK terpilih sebagai *primary key*, entitas Mata Kuliah dapat dilihat pada gambar 4.11.



**Gambar 4. 11** *Primary Key* Entitas Mata Kuliah

### 3. *Primary Key* Entitas Status Staff

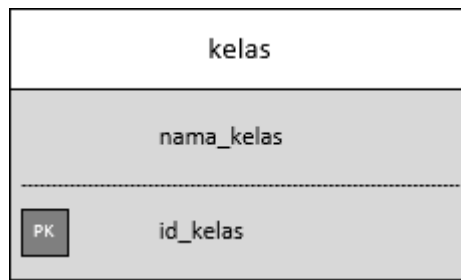
Entitas Status Staff memiliki attribute status dan nominal. Karena pada atribut status telah memenuhi persyaratan sebagai *primary key* maka atribut status terpilih sebagai *primary key*, entitas Status Staff dapat dilihat pada gambar 4.12.



**Gambar 4. 12** *Primary Key* Entitas Status Staff

### 4. *Primary Key* Entitas Kelas

Entitas Kelas memiliki atribut nama\_kelas. Karena belum adanya atribut yang memenuhi persyaratan sebagai *primary key* maka ditambahkan atribut id\_kelas sebagai *primary key*. Entitas Kelas dapat dilihat pada Gambar 4.13



**Gambar 4. 13** *Primary Key* Entitas Kelas

#### 5. *Primary Key* Entitas Penggajian

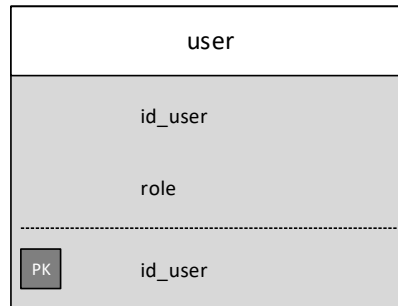
Entitas Penggajian memiliki atribut status, nip, kodeMK, kelas, totalSesi, totalGaji, bulan, tanggal, tanggalditerima dan statusPengiriman. Karena belum adanya atribut yang memenuhi persyaratan sebagai *primary key* maka ditambahkan atribut id\_penggajian sebagai *primary key*. Entitas Penggajian dapat dilihat pada Gambar 4.14.



**Gambar 4. 14** *Primary Key* Entitas Penggajian

#### 6. *Primary Key* Entitas *User*

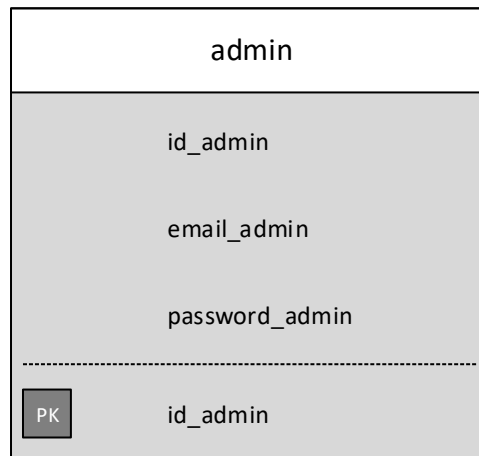
Entitas *User* memiliki atribut *role*, tetapi karena atribut tersebut belum memenuhi persyaratan sebagai *primary key* maka ditambahkan atribut *id\_user* sebagai *primary key*, entitas *User* dapat terlihat pada Gambar 4.15.



**Gambar 4.15** *Primary Key* Entitas *User*

#### 7. *Primary Key* Entitas *Admin*

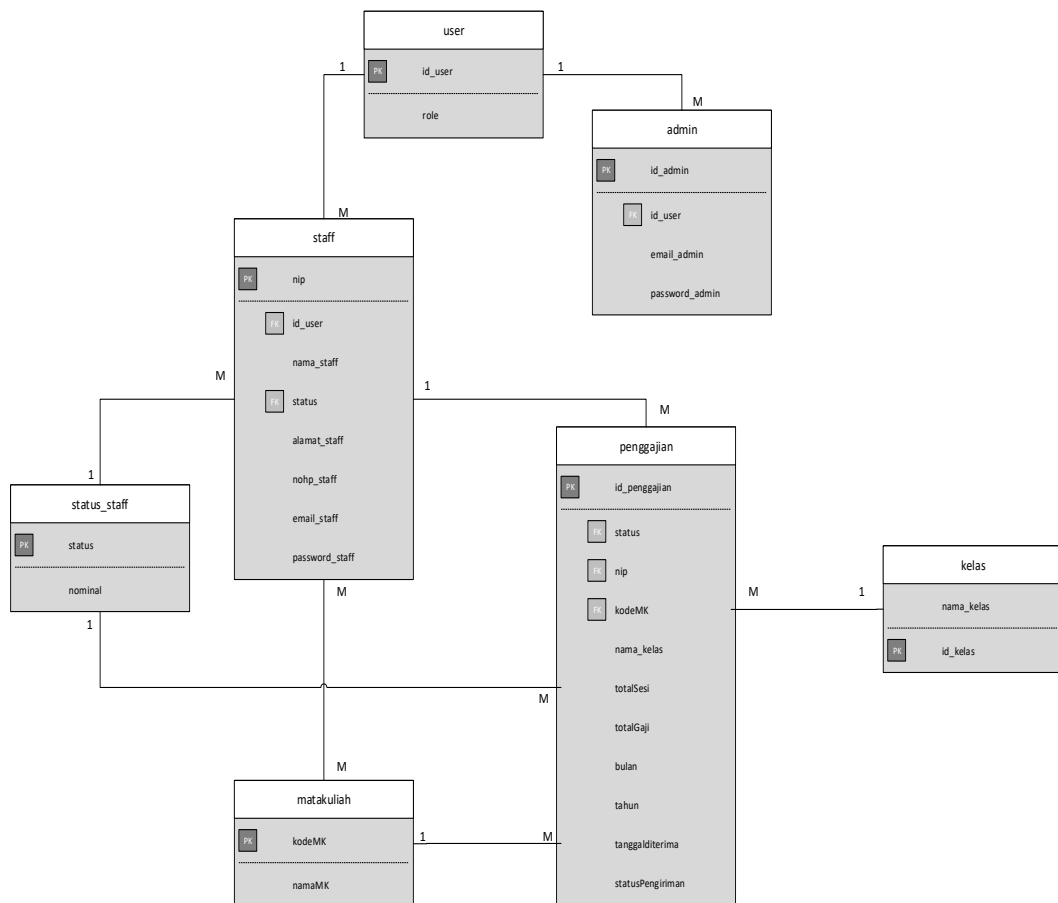
Entitas *Admin* memiliki atribut *email\_admin* dan *password\_admin*, tetapi karena atribut tersebut belum memenuhi persyaratan *primary key* maka ditambahkan atribut *id\_admin* sebagai *primary key*, entitas *Admin* dapat dilihat pada Gambar 4.16.



**Gambar 4.16** *Primary Key* Entitas *Admin*

#### 4.2.4 ERD (Entity Relationship Diagram)

Dalam sistem penggajian staf yang dibuat terdapat beberapa entitas yaitu entitas staff, entitas mata kuliah, entitas status staff, entitas penggajian, entitas *user* dan entitas admin. Dari semua entitas tersebut membentuk sebuah ERD diagram. ERD diagram dapat dilihat pada gambar 4.17.



**Gambar 4. 17** ERD Sistem Penggajian Staf

Penjelasan ERD pada Gambar 4.17 adalah sebagai berikut :

- Entitas *User* mempunyai relasi *one to many* dengan entitas Admin, yang artinya satu peran (*role*) dapat memiliki banyak *user* sesuai dengan peran dalam entitas Admin yaitu sebagai admin.

- b. Entitas *User* mempunyai relasi *one to many* dengan entitas Staff, yang artinya satu peran (*role*) dapat memiliki banyak *user* sesuai dengan peran dalam entitas Staff yaitu sebagai staff
- c. Entitas Staff mempunyai relasi *one to many* dengan entitas Penggajian, yang artinya satu staff hanya dapat melihat banyak data penggajian sesuai dengan data pribadinya.
- d. Entitas Status Staff mempunyai relasi *one to many* dengan entitas Staff, yang artinya satu status staff dapat dimiliki banyak staff dan banyaknya staff hanya memiliki satu status staff.
- e. Entitas Status Staff mempunyai relasi *one to many* dengan entitas Penggajian, yang artinya satu status staff dapat dimiliki banyak data penggajian dan banyaknya data penggajian hanya memiliki satu status staff.
- f. Entitas Staff mempunyai relasi *many to many* dengan entitas Mata Kuliah, yang artinya staff dapat mengajar lebih dari satu mata kuliah dan mata kuliah dapat diajar lebih dari satu staff, maka relasi ini akan menciptakan tabel baru.
- g. Entitas Mata Kuliah mempunyai relasi *one to many* dengan entitas Penggajian, yang artinya satu mata kuliah dapat memiliki banyak data penggajian.
- h. Entitas Kelas mempunyai relasi *one to many* dengan entitas Penggajian, yang artinya satu kelas dapat dimiliki banyak data penggajian.

### **4.3 Perancangan *Database* Logikal**

Perancangan *database* logikal dilakukan dengan tahapan konversi ERD ke dalam tabel. Berikut ini merupakan tahapannya :

#### **4.3.1 Konversi *ERD* ke dalam Tabel**

Konversi ERD ke dalam bentuk tabel adalah proses memasukan entitas menjadi nama tabel dan atribut menjadi nama kolom beserta relasinya ke dalam sebuah tabel yang saling berhubungan.

### 1. Konversi dengan relasi 1:M (*one to many*)

Konversi entitas yang memiliki relasi 1:M (*one to many*) yaitu menghubungkan dua buah entitas yang akan direpresntasikan dalam bentuk pemberian atau penambahan kunci atribut dari himpunan entitas pertama ke tabel yang mewakili himpunan entitas kedua. Kunci atribut dari himpunan entitas pertama ini menjadi atribut tambahan bagi himpunan entitas kedua. Pada ERD yang ditunjukkan pada Gambar 3.19 ditunjukkan terdapat relasi *one to many* yaitu antara entitas *user* dengan entitas admin, antara entitas *user* dengan entitas staff, antara entitas status staff dengan entitas staff, antara entitas status staff dengan entitas penggajian, antara entitas staff dengan entitas penggajian, antara entitas mata kuliah dengan entitas penggajian dan antara entitas kelas dengan entitas penggajian. Dari entitas tersebut maka akan direpresentasikan kedalam table yang dapat dilihat pada tabel-tabel berikut ini :

#### a. Relasi entitas *User* dengan entitas Admin

**Tabel 4. 3** Relasi antara entitas user dengan entitas admin

id_user	role

**Tabel 4. 4** Relasi antara entitas admin dengan entitas user

id_admin	id_user	email_admin	password_admin

Penjelasan konversi relasi antara entitas user pada table 4.3 dengan entitas admin pada tabel 4.4 adalah sebagai berikut :

Entitas user memiliki relasi *one to many* dengan entitas admin yang artinya satu id\_user yang terdapat di tabel user dapat dimiliki oleh banyak admin. Maka perlu ditambahkannya id\_user pada tabel admin yang berfungsi sebagai *foreign key* pada tabel admin.

b. Relasi entitas *User* dengan entitas *Staff*

**Tabel 4. 5** Relasi antara entitas user dengan entitas staff

id_user	role

**Tabel 4. 6** Relasi antara entitas staff dengan entitas user

nip	id_user	nama_staff	status	alamat_staff	nohp_staff	email_staff	password_staff

Penjelasan konversi relasi antara entitas user pada table 4.5 dengan entitas staff pada tabel 4.6 adalah sebagai berikut :

Entitas user memiliki relasi *one to many* dengan entitas staff yang artinya satu id\_user yang terdapat di tabel user dapat dimiliki oleh banyak staff. Maka perlu ditambahkan id\_user pada tabel staff yang berfungsi sebagai *foreign key* pada tabel staff.

c. Relasi entitas *Staff* dengan entitas *Penggajian*

**Tabel 4. 7** Relasi antara entitas staff dengan entitas penggajian

nip	id_user	nama_staff	status	alamat_staff	nohp_staff	email_staff	password_staff

**Tabel 4. 8** Relasi entitas penggajian dengan entitas staff

id_penggajian	status	nip	kodeMK	id_kelas	totalSesi	totalGaji	bulan	tahun	tanggalditerima	statusPengiriman

Penjelasan konversi relasi antara entitas staff pada table 4.7 dengan entitas penggajian pada tabel 4.8 adalah sebagai berikut :

Entitas staff memiliki relasi *one to many* dengan entitas penggajian yang artinya satu nip yang terdapat di tabel staff dapat memiliki banyak data pada tabel penggajian. Maka perlu ditambahkan nip pada tabel penggajian yang berfungsi sebagai *foreign key* pada tabel penggajian.



## d. Relasi entitas Status Staff dengan entitas Staff

**Tabel 4. 9** Relasi antara entitas status staff dengan entitas staff

status	nominal

**Tabel 4. 10** Relasi antara entitas staff dengan entitas status staff

nip	id_user	nama_staff	status	alamat_staff	nohp_staff	email_staff	password_staff

Penjelasan konversi relasi antara entitas status staff pada table 4.9 dengan entitas staff pada tabel 4.10 adalah sebagai berikut :

Entitas status staff memiliki relasi *one to many* dengan entitas staff yang artinya satu status yang terdapat di tabel status staff dapat dimiliki oleh banyak staff. Maka perlu ditambahkan status pada tabel staff yang berfungsi sebagai *foreign key* pada tabel staff.

## e. Relasi entitas Status Staff dengan entitas Penggajian

**Tabel 4. 11** Relasi antara entitas status staff dengan entitas penggajian

status	nominal

**Tabel 4. 12** Relasi antara entitas penggajian dengan entitas status staff

id_penggajian	status	nip	kodeMK	id_kelas	totalSesi	totalGaji	bulan	tahun	tanggalditerima	statusPengiriman

Penjelasan konversi relasi antara entitas status staff pada table 4.11 dengan entitas penggajian pada tabel 4.12 adalah sebagai berikut :

Entitas status staff memiliki relasi *one to many* dengan entitas penggajian yang artinya satu status yang terdapat di tabel status staff dapat dimiliki pada banyak data penggajian. Maka perlu ditambahkan status pada tabel penggajian yang berfungsi sebagai *foreign key* pada tabel penggajian.

## f. Relasi entitas Mata Kuliah dengan entitas Penggajian

**Tabel 4. 13** Relasi antara entitas mata kuliah dengan entitas penggajian

kodeMK	namaMK

**Tabel 4. 14** Relasi antara entitas penggajian dengan entitas mata kuliah

id_penggajian	status	nip	kodeMK	id_kelas	totalSesi	totalGaji	bulan	tahun	tanggalditerima	statusPengiriman

Penjelasan konversi relasi antara entitas mata kuliah pada table 4.13 dengan entitas penggajian pada tabel 4.14 adalah sebagai berikut :

Entitas mata kuliah memiliki relasi *one to many* dengan entitas penggajian yang artinya satu kodeMK yang terdapat di tabel mata kuliah dapat dimiliki pada banyak data penggajian. Maka perlu ditambahkan kodeMK pada tabel penggajian yang berfungsi sebagai *foreign key* pada tabel penggajian.

## g. Relasi entitas Kelas dengan entitas Penggajian

**Tabel 4. 15** Relasi antara entitas kelas dengan entitas penggajian

id_kelas	nama_kelas

**Tabel 4. 16** Relasi antara entitas penggajian dengan entitas kelas

id_penggajian	status	nip	kodeMK	id_kelas	totalSesi	totalGaji	bulan	tahun	tanggalditerima	statusPengiriman

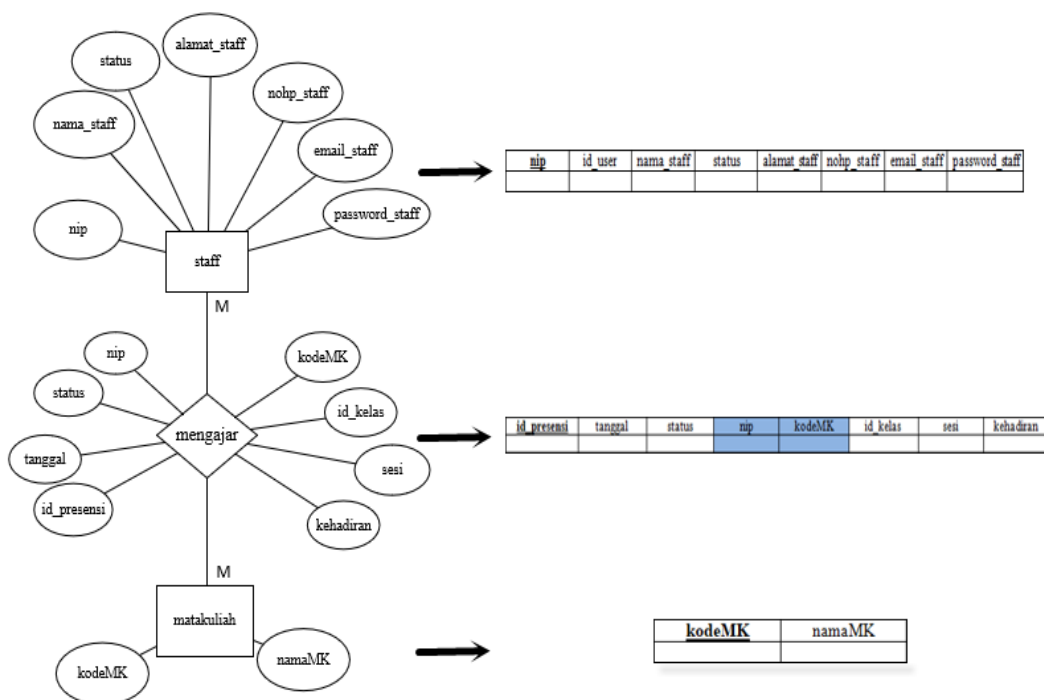
Penjelasan konversi relasi antara entitas kelas pada table 4.15 dengan entitas penggajian pada tabel 4.16 adalah sebagai berikut :

Entitas kelas memiliki relasi *one to many* dengan entitas penggajian yang artinya satu id\_kelas yang terdapat di tabel kelas dapat dimiliki pada banyak data penggajian. Maka perlu ditambahkan id\_kelas pada tabel penggajian yang berfungsi sebagai *foreign key* pada tabel penggajian.

## 2. Konversi dengan relasi M:M (*many to many*)

Konversi entitas yang memiliki relasi M:M (*many to many*) yaitu menghubungkan antara dua entitas, diwujudkan dalam bentuk tabel (*file data*) khusus yang memiliki *field (foreign key)* yang berasal dari kunci-kunci (*key*) dari entitas yang dihubungkan sehingga menciptakan tabel baru. Pada ERD yang ditunjukkan pada Gambar 3.19 ditunjukkan terdapat relasi *many to many* yaitu antara entitas staff dengan entitas mata kuliah. Dari entitas-entitas tersebut akan direpresentasikan ke dalam table, dapat dilihat pada Gambar 3.20 sebagai berikut :

### a. Relasi entitas Staff dengan entitas Mata Kuliah



**Gambar 4. 18** Konversi relasi antara entitas staff dengan entitas mata kuliah

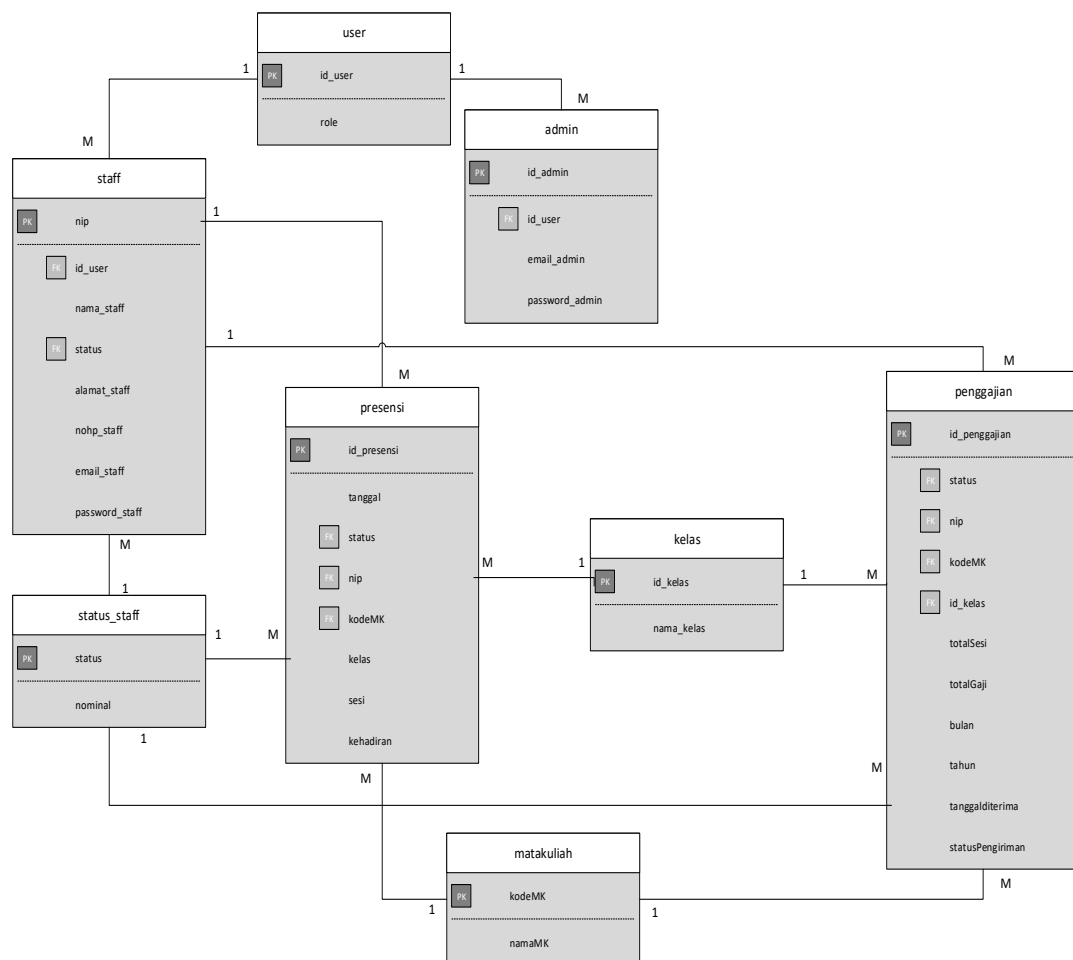
Penjelasan konversi relasi antara entitas staff dengan entitas mata kuliah pada Gambar 4.18 adalah sebagai berikut :

Entitas staff memiliki relasi *many to many* dengan entitas mata kuliah sehingga tercipta tabel baru yang menghubungkan antara entitas dosen dengan entitas mata kuliah yang diberikan nama tabel presensi, dimana kolom nip pada

tabel staff dan kolom kodeMK pada tabel mata kuliah ditambahkan ke dalam tabel presensi sebagai *foreign key*, maka perlu ditambahkan id\_presensi sebagai *primary key* dan kolom lain sebagai penambah informasi seperti tanggal, status, nama\_kelas, sesi dan kehadiran.

### 3. Relasi antar Tabel

Setelah dilakukan konversi ERD ke dalam tabel maka akan didapatkan relasi antar tabel yang dapat dilihat pada Gambar 4.19 sebagai berikut :



**Gambar 4. 19** Relasi antar tabel

#### 4. *Relationship* (Hubungan)

Tahapan selanjutnya adalah mengidentifikasi *relationship* (hubungan) antara entitas yang satu dengan entitas lain yang telah diidentifikasi sebelumnya. Setiap entitas mempunyai *relationship* atau hubungan. Berikut adalah *relationship* dari sistem penggajian staff yang dapat dilihat pada tabel 4.17

**Tabel 4. 17** Relasi Sistem Penggajian Staf

Nama Entitas	Hubungan	Nama Entitas
User	Memiliki	Admin
User	Memiliki	Staff
Staff	Melihat	Penggajian
Staff	Memiliki	Status Staff
Presensi	Memiliki	Status Staff
Penggajian	Memiliki	Status Staff
Presensi	Memiliki	Mata Kuliah
Penggajian	Memiliki	Mata Kuliah
Presensi	Memiliki	Kelas
Penggajian	Memiliki	Kelas

Penjelasan *relationship* atau hubungan pada tabel 4.17 adalah sebagai berikut :

- a. *User* memiliki admin yang artinya pada entitas Admin terdapat atribut *id\_user* sesuai dengan fungsi *role* dari admin.
- b. *User* memiliki staff yang artinya pada entitas Staff terdapat atribut *id\_user* sesuai dengan fungsi *role* dari staff
- c. Staff melihat penggajian yang artinya pada entitas penggajian, staff dapat melihat penggajian yang diterima disetiap bulannya. Staff dapat melihat berapa banyak sesi pertemuan dalam pengajaran dan status pengiriman penggajian tersebut yang disertakan dengan tanggal pengirimannya.
- d. Staff memiliki status staff yang artinya pada entitas staff terdapat atribut status, dimana satu staff memiliki satu status untuk dapat menggolongkan status dari masing-masing staff.
- e. Presensi memiliki status staff yang artinya pada entitas presensi terdapat atribut status, dimana satu data presensi memiliki satu status yang dapat menggolongkan status dari masing-masing staff.

- f. Penggajian memiliki status staff yang artinya pada entitas penggajian terdapat atribut status, dimana satu data penggajian memiliki satu status yang dapat menggolongkan status dari masing-masing staff.
- g. Presensi memiliki mata kuliah yang artinya pada entitas presensi terdapat atribut kodeMK, dimana pada presensi memiliki satu kode mata kuliah yang diajarkan oleh staff.
- h. Penggajian memiliki mata kuliah yang artinya pada entitas penggajian terdapat atribut kodeMK, dimana pada penggajian memiliki satu kode mata kuliah yang diajarkan oleh staff.
- i. Presensi memiliki kelas yang artinya pada entitas presensi terdapat atribut id\_kelas, dimana pada presensi memiliki satu kelas yang diajarkan oleh staff.
- j. Penggajian memiliki kelas yang artinya pada entitas penggajian terdapat atribut id\_kelas, dimana pada penggajian memiliki satu kelas yang diajarkan oleh staff.

Berdasarkan penjelasan diatas maka dapat ditentukan *foreign key* dari entitas yang saling berhubungan. Berikut ini *foreign key* yang terdapat pada entitas-entitas yang ada pada sistem penggajian staff, diantaranya adalah :

1. *User* memiliki Admin
  - a. Entitas *User* merupakan *parent table*
  - b. Entitas Admin merupakan *child table*
  - c. *Foreign Key* yaitu id\_user *reference* dari *parent table User*
2. *User* memiliki Staff
  - a. Entitas *User* merupakan *parent table*
  - b. Entitas Staff merupakan *child table*
  - c. *Foreign Key* yaitu id\_user *reference* dari *parent table User*
3. Staff melihat Penggajian
  - a. Entitas Staff merupakan *parent table*
  - b. Entitas Penggajian merupakan *child table*
  - c. *Foreign Key* yaitu nip *reference* dari *parent table Staff*
4. Staff memiliki Status Staff
  - a. Entitas Status Staff merupakan *parent table*

- b. Entitas Staff merupakan *child table*
  - c. *Foreign Key* yaitu status *reference* dari *parent table* Status Staff
5. Presensi Memiliki Status Staff
- a. Entitas Status Staff merupakan *parent table*
  - b. Entitas Presensi merupakan *child table*
  - c. *Foreign Key* yaitu status *reference* dari *parent table* Status Staff
6. Penggajian Memiliki Status Staff
- a. Entitas Status Staff merupakan *parent table*
  - b. Entitas Penggajian merupakan *child table*
  - c. *Foreign Key* yaitu status *reference* dari *parent table* Status Staff
7. Presensi Memiliki Mata Kuliah
- a. Entitas Mata Kuliah merupakan *parent table*
  - b. Entitas Presensi merupakan *child table*
  - c. *Foreign Key* yaitu kodeMK *reference* dari *parent table* Mata Kuliah
8. Penggajian Memiliki Mata Kuliah
- a. Entitas Mata Kuliah merupakan *parent table*
  - b. Entitas Penggajian merupakan *child table*
  - c. *Foreign Key* yaitu kodeMK *reference* dari *parent table* Mata Kuliah
9. Presensi Memiliki Kelas
- a. Entitas Kelas merupakan *parent table*
  - b. Entitas Presensi merupakan *child table*
  - c. *Foreign Key* yaitu id\_kelas *reference* dari *parent table* Kelas
10. Penggajian Memiliki Kelas
- a. Entitas Kelas merupakan *parent table*
  - b. Entitas Penggajian merupakan *child table*
  - c. *Foreign Key* yaitu id\_kelas *reference* dari *parent table* Kelas

#### **4.3.2 Kamus Data Database**

Kamus data (*data dictionary*) merupakan salah satu komponen kunci dari DBMS yang mencakup informasi mengenai struktur *database*. Setiap elemen

data yang disimpan dalam *database* memiliki catatan kamus data yang mendeskripsikan elemen-elemen tersebut.

Berikut ini merupakan kamus *database* dari tabel *User* :

### 1. Kamus Data *User*

**Tabel 4. 18** Kamus Data Tabel *User*

<b>Nama Atribut</b>	<b>Tipe Data</b>	<b>Key</b>	<b>Tabel Referensi</b>
id_user	int(11)	<i>Primary Key</i>	-
Role	varchar(5)	-	-

Penjelasan dari Tabel 4.18 Kamus Data Tabel *User* adalah sebagai berikut :

- a. id\_user adalah atribut yang menunjukkan id *user* dengan tipe data *int*, yang mempunyai panjang data 11 karena atribut id\_user berisikan angka (*numeric*).
- b. role adalah atribut yang menunjukkan nama dari peran masing-masing *user* dengan tipe data *varchar* yang mempunyai panjang data 5 karena atribut role berisikan karakter (*string*) dengan panjang yang bervariasi tergantung dengan datanya

### 2. Kamus Data Admin

**Tabel 4. 19** Kamus Data Tabel Admin

<b>Nama Atribut</b>	<b>Tipe Data</b>	<b>Key</b>	<b>Tabel Referensi</b>
id_admin	int(11)	<i>Primary Key</i>	-
id_user	int(11)	<i>Foreign Key</i>	Tabel <i>User</i>
email_admin	varchar(30)	-	-
password_admin	varchar(50)	-	-



Penjelasan dari Tabel 4.19 Kamus Data Tabel Admin adalah sebagai berikut :

- a. *id\_admin* adalah atribut yang menunjukkan id dari admin dengan tipe data *int* yang mempunyai panjang data 11 karena atribut *id\_admin* berisikan angka (*numeric*).
- b. *id\_user* adalah atribut yang menunjukkan *id\_user* dari tabel *user* dan merupakan *foreign key*.
- c. *email\_admin* adalah atribut yang menunjukkan *email* dari admin dengan tipe data *varchar* yang mempunyai panjang data 50 karena attribute *email\_admin* berisikan karakter (*string*) dengan panjang yang bervariasi tergantung dengan datanya.
- d. *password\_admin* adalah atribut yang menunjukkan *password* dari admin dengan tipe data *varchar* karena atribut *password\_admin* berisikan karakter (*string*) dengan panjang bervariasi tergantung dengan datanya. *password\_admin* mempunyai panjang data 50 karena menggunakan enkripsi data MD5.

### 3. Kamus Data Staff

**Tabel 4. 20** Kamus Data Tabel Admin

<b>Nama Atribut</b>	<b>Tipe Data</b>	<b>Key</b>	<b>Tabel Referensi</b>
nip	char(5)	<i>Primary Key</i>	-
id_user	int(11)	<i>Foreign Key</i>	Tabel <i>User</i>
nama_staff	varchar(50)	-	-
status	char(10)	<i>Foreign Key</i>	Tabel Status Staff
alamat_staff	text	-	-
nohp_staff	char(12)	-	-

<b>Nama Atribut</b>	<b>Tipe Data</b>	<b>Key</b>	<b>Tabel Referensi</b>
email_admin	varchar(30)	-	-
password_admin	varchar(50)	-	-

Penjelasan dari Tabel 4.20 Kamus Data Tabel Staff adalah sebagai berikut :

- a. nip adalah atribut Nomor Induk Pengajar atau sebagai inialisasi dari staff dengan tipe data *char* yang memiliki panjang data 5 karena atribut nip berisikan data dengan panjang yang tetap.
- b. id\_user adalah atribut yang menunjukkan id\_user dari tabel *user* dan merupakan *foreign key*.
- c. nama\_staff adalah atribut yang menunjukkan nama dari staff dengan tipe data *varchar* yang mempunyai panjang data 50 karena atribut nama\_staff berisikan karakter (*string*) dengan panjang bervariasi tergantung pada datanya.
- d. status adalah atribut yang menunjukan status dari tabel status staff dan merupakan *foreign key*.
- e. alamat\_staff adalah atribut yang menunjukkan alamat dari staff dengan tipe data *text* karena atribut alamat\_staff berisi karakter (*string*) dengan panjang yang *relative* lebih panjang dari data di atribut lainnya.
- f. nohp\_staff adalah atribut yang menunjukkan no hp dari staff dengan tipe data *char* yang mempunyai panjang data 12 karena nohp\_staff berisikan karakter (*string*) dengan panjang yang tetap.
- e. email\_admin adalah atribut yang menunjukkan *email* dari admin dengan tipe data *varchar* yang mempunyai panjang data 50 karena attribute email\_admin berisikan karakter (*string*) dengan panjang yang bervariasi tergantung dengan datanya.
- g. password\_admin adalah atribut yang menunjukkan *password* dari admin dengan tipe data *varchar* karena atribut *password\_admin* berisikan karakter (*string*) dengan panjang bervariasi tergantung dengan datanya.

*password\_admin* mempunyai panjang data 50 karena menggunakan enkripsi data MD5.

#### 4. Kamus Data Status Staff

**Tabel 4. 21** Kamus Data Tabel Status Staff

<b>Nama Atribut</b>	<b>Tipe Data</b>	<b>Key</b>	<b>Tabel Referensi</b>
status	char(10)	<i>Primary Key</i>	-
nominal	enum('100000', '60000', '30000')	-	-

Penjelasan dari Tabel 4.21 Kamus Data Tabel Status Staff adalah sebagai berikut :

- a. status adalah atribut yang menunjukkan status dari staff dengan tipe data *char* yang mempunyai panjang data 10 karena atribut status berisikan data dengan panjang yang tetap.
- b. nominal adalah atribut yang menunjukkan nominal gaji pada setiap status staff dengan tipe data *enum* karena nominal berisikan karakter atau nilai data yang sudah ditentukan sebelumnya yaitu '100000', '60000', '30000'.

#### 5. Kamus Data Mata Kuliah

**Tabel 4. 22** Kamus Data Tabel Mata Kuliah

<b>Nama Atribut</b>	<b>Tipe Data</b>	<b>Key</b>	<b>Tabel Referensi</b>
kodeMK	char(7)	<i>Primary Key</i>	-
namaMK	varchar(50)	-	-

Penjelasan dari Tabel 4.22 Kamus Data Tabel Mata Kuliah adalah sebagai berikut :

- a. kodeMK adalah atribut yang menunjukkan kode mata kuliah dari masing-masing mata kuliah dengan tipe data *char* yang mempunyai panjang data 7 karena atribut kodeMK berisikan data dengan panjang yang tetap.
- b. namaMK adalah atribut yang menunjukkan nama dari masing-masing mata kuliah dengan tipe data *varchar* yang mempunyai panjang data 50 karena atribut namaMK berisikan panjang rata-rata data dari nama mata kuliah tersebut.

## 6. Kamus Data Kelas

**Tabel 4. 23** Kamus Data Tabel Kelas

<b>Nama Atribut</b>	<b>Tipe Data</b>	<b>Key</b>	<b>Tabel Referensi</b>
id_kelas	int(11)	<i>Primary Key</i>	-
nama_kelas	char(1)	-	-

Penjelasan dari Tabel 4.23 Kamus Data Tabel kelas adalah sebagai berikut :

- a. id\_kelas adalah id dari masing-masing kelas yang bersifat *auto increment* dengan tipe data *int* yang mempunyai panjang data 11 karena atribut id\_kelas berisikan angka (*numeric*).
- b. nama\_kelas adalah atribut yang menunjukkan kelas yang di laksanakan pengajaran dengan tipe data *char* yang mempunyai panjang data 1 karena atribut kelas berisikan 1 karakter (*string*).

## 7. Kamus Data Presensi

**Tabel 4. 24** Kamus Data Tabel Presensi

<b>Nama Atribut</b>	<b>Tipe Data</b>	<b>Key</b>	<b>Tabel Referensi</b>
id_presensi	int(11)	<i>Primary Key</i>	-
tanggal	date	-	-
status	char(10)	<i>Foreign Key</i>	Tabel Status Staff
nip	char(5)	<i>Foreign Key</i>	Tabel Staff
kodeMK	char(7)	<i>Foreign Key</i>	Tabel Mata Kuliah
id_kelas	char(1)	<i>Foreign Key</i>	Tabel Kelas
sesi	int(11)	-	-
kehadiran	enum('1', '0')	-	-

Penjelasan dari Tabel 4.24 Kamus Data Tabel Presensi adalah sebagai berikut :

- a. id\_presensi adalah atribut yang menunjukkan id presensi dari masing-masing presensi yang bersifat *auto increment* dengan tipe data *int* yang mempunyai panjang data 11 karena atribut id\_presensi berisikan angka (*numeric*).
- b. tanggal adalah atribut yang menunjukkan tanggal presensi yang dilakukan dengan tipe data *date* karena pada atribut tanggal memiliki format tanggal.
- c. status adalah atribut yang menunjukkan atau mengelompokkan status dari tabel status staff dan merupakan *foreign key*.
- d. nip adalah atribut yang menunjukkan nip dari tabel staff dan merupakan *foreign key*.
- e. kodeMK adalah atribut yang menunjukkan kodeMK dari tabel matakuliah dan merupakan *foreign key*.

- f. `id_kelas` adalah atribut yang menunjukkan id dari suatu kelas di prodi dari tabel kelas dan merupakan *foreign key*.
- g. `sesi` adalah atribut yang menunjukkan pada sesi ke berapa mata kuliah tersebut dilaksanakan dengan tipe data *int* yang mempunyai panjang data 11 karena atribut `sesi` berisikan angka (*numeric*).
- h. `kehadiran` adalah atribut yang menunjukkan hadir atau tidaknya staff dalam pengajaran dengan tipe data *enum* karena karakter atau nilai data sudah ditentukan sebelumnya yaitu '1' dan '0'.

## 8. Kamus Data Penggajian

**Tabel 4. 25** Kamus Data Tabel Penggajian

<b>Nama Atribut</b>	<b>Tipe Data</b>	<b>Key</b>	<b>Tabel Referensi</b>
<code>id_penggajian</code>	<code>int(11)</code>	<i>Primary Key</i>	-
<code>status</code>	<code>char(10)</code>	<i>Foreign Key</i>	Tabel Status Staff
<code>nip</code>	<code>char(5)</code>	<i>Foreign Key</i>	Tabel Staff
<code>kodeMK</code>	<code>char(7)</code>	<i>Foreign Key</i>	Tabel Mata Kuliah
<code>id_kelas</code>	<code>char(1)</code>	<i>Foreign Key</i>	Tabel Kelas
<code>totalSesi</code>	<code>int(11)</code>	-	-
<code>totalGaji</code>	<code>float</code>	-	-
<code>bulan</code>	<code>char(2)</code>	-	-
<code>tahun</code>	<code>char(4)</code>	-	-
<code>tanggalditerima</code>	<code>date</code>	-	-
<code>statusPengiriman</code>	<code>Enum('1','0')</code>	-	-

Penjelasan dari Tabel 4.25 Kamus Data Tabel Penggajian adalah sebagai berikut :


- a. `id_penggajian` adalah atribut yang menunjukkan id penggajian dengan tipe data yang bersifat *auto increment* dengan tipe data *int* yang mempunyai panjang data 11 karena atribut `id_penggajian` berisikan angka (*numeric*).
- b. `status` adalah atribut yang menunjukkan atau mengelompokkan status dari tabel status staff dan merupakan *foreign key*.
- c. `nip` adalah atribut yang menunjukkan nip dari tabel staff dan merupakan *foreign key*.
- d. `kodeMK` adalah atribut yang menunjukkan kodeMK dari tabel matakuliah dan merupakan *foreign key*.
- e. `id_kelas` adalah atribut yang menunjukkan id dari suatu kelas di prodi dari tabel kelas dan merupakan *foreign key*.
- f. `totalSesi` adalah atribut yang menunjukkan jumlah total sesi atau pertemuan yang dilakukan staff dalam sebulan dengan tipe data *int* yang mempunyai panjang data 11 karena atribut `totalSesi` berisikan angka (*numeric*).
- g. `totalGaji` adalah atribut yang menunjukkan total gaji yang diterima staff dalam sebulan dengan tipe data *float* karena atribut `totalGaji` berisikan bilangan pecahan positif presisi tunggal.
- h. `bulan` adalah atribut yang menunjukkan bulan diterimanya gaji staff dengan tipe data *char* yang mempunyai panjang data 2 karena atribut `bulan` berisikan data dengan panjang yang tetap.
- i. `tahun` adalah atribut yang menunjukkan tahun diterimanya gaji staff dengan tipe data *char* yang mempunyai panjang data 4 karena atribut `tahun` berisikan data dengan panjang yang tetap.
- j. `tanggalditerima` adalah atribut yang menunjukkan tanggal di kirimkannya gaji tersebut dengan tipe data *date* karena pada atribut tanggal memiliki format tanggal.
- k. `statusPengiriman` adalah atribut yang menunjukkan terkirim atau belumnya gaji tersebut dengan tipe data *enum* karena karakter atau nilai data sudah ditentukan sebelumnya yaitu '1' dan '0'.

## 4.4 Perancangan *Database* Fisikal

Setelah melakukan beberapa tahapan dalam perancangan basis data, maka diperolehnya sebuah basis data yang bisa diimplementasikan. Adapun basis data yang dihasilkan terdiri dari 8 tabel yang masing-masing tabel memiliki kolom yang mewakili beberapa item data. Berikut adalah tabel-tabel yang dihasilkan dalam perancangan basis data.

### 4.4.1 Tabel

#### 4.4.1.1 Tabel *User*

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id_user 	int(11)			No	None	AUTO_INCREMENT
2	role	varchar(20)			No	None	

**Gambar 4. 20** Struktur Tabel *user*

Penjelasan dari Gambar 4.20 adalah sebagai berikut :

1. id\_user adalah kolom yang menunjukkan id dari masing-masing *user* sesuai dengan peran dari kedua *user* diantaranya admin dan staf. id\_user bersifat *auto increment* dan *primary key* dengan tipe data *int* karena pada kolom id\_user berisikan angka (*numeric*). Kolom id\_user memiliki panjang data 11 karena merupakan maksimal panjang data pada tipe data *int*.
2. role pada kolom yang menunjukkan nama *role* dari masing-masing *user* sesuai dengan peran dari kedua *user* diantaranya admin dan staf. Kolom *role* memiliki tipe data *varchar* karena berisikan karakter (*string*) dengan panjang data 20 karena karakter panjang dari nama *role* pada tabel *user* rata-rata terdiri dari 5 karakter.



#### 4.4.1.2 Tabel Admin





#	Name	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/> 1	<b>id_admin</b> 🔑	int(11)			No	None	AUTO_INCREMENT
<input type="checkbox"/> 2	<b>id_user</b> 🔑	int(11)			Yes	NULL	
<input type="checkbox"/> 3	<b>email_admin</b> 🔑	varchar(30)			Yes	NULL	
<input type="checkbox"/> 4	<b>password_admin</b>	varchar(50)			Yes	NULL	

**Gambar 4. 21** Struktur Tabel Admin

Penjelasan pada Gambar 4.21 adalah sebagai berikut :

1. `id_admin` adalah kolom yang menunjukkan id dari masing-masing admin yang bersifat *auto increment* dan *primary key* dengan tipe data *int* karena pada kolom `id_admin` berisikan angka (*numeric*). Kolom `id_admin` memiliki panjang data 11 karena merupakan maksimal panjang data pada tipe data *int*.
2. `id_user` adalah kolom yang menunjukkan `id_user` pada tabel `user` dan merupakan *foreign key*.
3. `email_admin` adalah kolom yang menunjukkan *email* dari masing-masing admin. Kolom `email_admin` memiliki tipe data *varchar* karena `email_admin` berisikan karakter (*string*) dengan panjang data 30 karena rata-rata panjang data maksimal dari email adalah 30 karakter. `email_admin` bersifat *unique key* agar tidak terdapat nama email yang sama.
4. `password_admin` adalah kolom yang menunjukkan password dari masing-masing admin. Kolom `password_admin` memiliki tipe data *varchar* karena `password_admin` berisikan karakter (*string*) dengan panjang data 50 karena password dari admin menggunakan enkripsi MD5 agar password menjadi lebih aman.

#### 4.4.1.3 Table Staff

	#	Name	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/>	1	nip 	char(5)			No	None	
<input type="checkbox"/>	2	id_user 	int(11)			No	None	
<input type="checkbox"/>	3	nama_staff	varchar(50)			No	None	
<input type="checkbox"/>	4	status 	char(10)			No	None	
<input type="checkbox"/>	5	alamat_staff	text			No	None	
<input type="checkbox"/>	6	nohp_staff	char(12)			No	None	
<input type="checkbox"/>	7	email_staff 	varchar(30)			No	None	
<input type="checkbox"/>	8	password_staff	varchar(50)			No	None	

**Gambar 4. 22** Struktur Tabel Staff

Penjelasan pada Gambar 4.22 adalah sebagai berikut :

1. nip adalah kolom yang menunjukkan nomor induk pengajar dari masing-masing staf dengan tipe data *char* karena pada kolom nip berisikan data dengan panjang yang tetap. Kolom nip memiliki panjang data 5 karena merupakan panjang data rata-rata nomor induk pengajar.
2. id\_user adalah kolom yang menunjukkan id\_user dari tabel *user* dan merupakan *foreign key*.
3. nama\_staff adalah kolom yang menunjukkan nama dari masing-masing staf dengan tipe data *varchar* karena pada kolom nama\_staff berisikan karakter (*string*). Kolom nama\_staff memiliki panjang data 50 karakter.
4. status adalah kolom yang menunjukkan status dari tabel *status\_staff* dan merupakan *foreign key*.
5. alamat\_staff adalah kolom yang menunjukkan alamat dari masing-masing staf dengan tipe data *text* karena alamat\_staff memiliki panjang data yang *relative* lebih panjang dari data di kolom lainnya.

6. nohp\_staff adalah kolom yang menunjukkan nomor *handphone* dari masing-masing staf dengan tipe data *char* karena nohp\_staff berisikan data dengan panjang yang tetap. Kolom nohp\_staff memiliki panjang data 12 karena merupakan panjang maksimal dari nomor *handphone* yang ada saat ini.
7. email\_staff adalah kolom yang menunjukkan email dari masing-masing staf dengan tipe data *varchar* karena email\_staff berisi karakter (*string*). Kolom email\_staff memiliki panjang data 30 karena rata-rata panjang data maksimal dari email adalah 30 karakter. email\_staff bersifat *unique key* agar tidak terdapat nama email yang sama.
8. password\_staff adalah kolom yang menunjukkan password dari masing-masing staf dengan tipe data *varchar* karena password\_staff berisikan karakter (*string*). Kolom password\_staff memiliki panjang data 50 karena password dari staf menggunakan enkripsi MD5 agar password menjadi lebih aman.

#### 4.4.1.4 Tabel Status Staff


#	Name	Type	Collation	Attributes	Null	Default	Extra
1	status	char(10)			No	None	
2	nominal	enum('100000', '60000', '30000')			No	None	

**Gambar 4. 23** Stuktur Table Status Staff

Penjelasan pada Gambar 4.23 adalah sebagai berikut :

1. status adalah kolom yang menunjukkan status dari staf dengan tipe data *char* karena karena pada kolom status berisikan data dengan panjang yang tetap. Kolom status memiliki panjang data 10 karena merupakan panjang data rata-rata dari status.
2. nominal adalah kolom yang menunjukkan nominal gaji dari masing-masing status dengan tipe data *enum* karena kolom nominal berisikan karakter atau nilai data yang sudah ditentukan sebelumnya yaitu '100000', '60000', '30000'.

#### 4.4.1.5 Tabel Mata Kuliah


#	Name	Type	Collation	Attributes	Null	Default	Extra
1	kodeMK 	char(7)			No	None	
2	namaMK	varchar(50)			Yes	NULL	

**Gambar 4. 24** Stuktur Table Status Staff

Penjelasan pada Gambar 4.24 adalah sebagai berikut :

1. kodeMK adalah kolom yang menunjukkan kode dari masing-masing mata kuliah dengan tipe data *char* karena berisikan data dengan panjang yang tetap. Kolom kodeMK memiliki panjang data 7 karena merupakan panjang tetap dari kode mata kuliah.
2. namaMK adalah kolom yang menunjukkan nama dari masing-masing mata kuliah dengan tipe data *varchar* karena pada kolom namaMK berisikan karakter (*string*). Kolom namaMK memiliki panjang data 50 kaena merupakan rata-rata panjang data dari mata kuliah.

#### 4.4.1.6 Tabel Kelas

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id_kelas 	int(11)			No	None	AUTO_INCREMENT
2	nama_kelas	char(1)			No	None	






**Gambar 4. 25** Struktur Tabel Kelas

Penjelasan pada Gambar 4.25 adalah sebagai berikut :

1. id\_kelas adalah kolom yang menunjukkan id dari masing-masing kelas. id\_kelas bersifat *auto increment* dan *primary key* dengan tipe data *int* karena pada kolom id\_kelas berisikan angka (*numeric*). Kolom id\_kelas memiliki panjang data 11 karena merupakan maksimal panjang data pada tipe data int.
2. nama\_kelas adalah kolom yang menunjukkan nama dari masing-masing kelas yang digunakan dengan tipe data *char* karena memiliki panjang data yang

tetap. Kolom kelas memiliki panjang data 1 karena berisikan satu karakter (*string*).

#### 4.4.1.7 Tabel Presensi

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id_presensi 	int(11)			No	None	AUTO_INCREMENT
2	tanggal	date			No	None	
3	status 	char(10)			No	None	
4	nip 	char(5)			No	None	
5	kodeMK 	char(7)			No	None	
6	id_kelas 	int(11)			No	None	
7	sesi	int(11)			No	None	
8	kehadiran	enum('0', '1')			No	None	

**Gambar 4. 26** Struktur Tabel Presensi

Penjelasan pada Gambar 4.26 adalah sebagai berikut :

1. id\_presensi adalah kolom yang menunjukkan id presensi dari setiap presensi yang diinputkan. id\_presensi bersifat *auto increment* dan *primary key* dengan tipe data *int* karena pada kolom id\_presensi berisikan angka (*numeric*). Kolom id\_presensi memiliki panjang data 11 karena merupakan maksimal panjang data pada tipe data *int*.
2. tanggal adalah kolom yang menunjukkan tanggal presensi tersebut diinputkan dengan tipe data *date* karena pada kolom tanggal memiliki format tanggal.
3. status adalah kolom yang menunjukkan status dari tabel status\_staff dan merupakan *foreign key*.
4. nip adalah kolom yang menunjukkan nip dari tabel staff dan merupakan *foreign key*.
5. kodeMK adalah kolom yang menunjukkan kodeMK dari tabel matakuliah dan merupakan *foreign key*.

6. `id_kelas` adalah kolom yang menunjukkan `id_kelas` dari tabel kelas dan merupakan *foreign key*
7. `sesi` adalah kolom yang menunjukkan pada sesi ke berapa mata kuliah tersebut dilaksanakan dengan tipe data *int* karena pada kolom sesi berisikan angka (*numeric*). Kolom sesi memiliki panjang data 11 karena merupakan maksimal panjang data pada tipe data *int*.
8. `kehadiran` adalah kolom yang menunjukkan hadir atau tidaknya staf dalam pengajaran. Kolom kehadiran memiliki tipe data *enum* karena berisikan karakter atau nilai data yang sudah ditentukan sebelumnya yaitu '1' dan '0'.

#### 4.4.1.8 Tabel Penggajian

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	<code>id_penggajian</code> 🔑	<code>int(11)</code>			No	None	AUTO_INCREMENT
2	<code>status</code> 🔑	<code>char(10)</code>			No	None	
3	<code>nip</code> 🔑	<code>char(5)</code>			No	None	
4	<code>kodeMK</code> 🔑	<code>char(7)</code>			No	None	
5	<code>id_kelas</code> 🔑	<code>int(11)</code>			No	None	
6	<code>totalSesi</code>	<code>int(11)</code>			No	None	
7	<code>totalGaji</code>	<code>float</code>			No	None	
8	<code>bulan</code>	<code>char(2)</code>			No	None	
9	<code>tahun</code>	<code>char(4)</code>			No	None	
10	<code>tanggalditerima</code>	<code>date</code>			No	None	
11	<code>statusPengiriman</code>	<code>enum('0','1')</code>			No	None	

**Gambar 4. 27** Struktur Tabel Penggajian

Pengejelasan pada Gambar 4.27 adalah sebagai berikut :

1. `id_penggajian` adalah kolom yang menunjukkan id penggajian dari setiap penggajian yang diinputkan. `id_penggajian` bersifat *auto increment* dan *primary key* dengan tipe data *int* karena pada kolom `id_penggajian` berisikan

- angka (*numeric*). Kolom `id_penggajian` memiliki panjang data 11 karena merupakan maksimal panjang data pada tipe data *int*.
2. `status` adalah kolom yang menunjukkan status dari tabel `status_staff` dan merupakan *foreign key*.
  3. `nip` adalah kolom yang menunjukkan nip dari tabel `staff` dan merupakan *foreign key*.
  4. `kodeMK` adalah kolom yang menunjukkan kodeMK dari tabel `matakuliah` dan merupakan *foreign key*.
  5. `id_kelas` adalah kolom yang menunjukkan `id_kelas` dari tabel `kelas` dan merupakan *foreign key*.
  6. `totalSesi` adalah kolom yang menunjukkan jumlah total sesi atau pertemuan yang dilakukan staf dalam sebulan dengan tipe data *int* karena pada kolom `totalSesi` berisikan angka (*numeric*). Kolom `totalSesi` memiliki panjang data 11 karena merupakan maksimal panjang data pada tipe data *int*.
  7. `totalGaji` adalah kolom yang menunjukkan total gaji yang diterima staf dalam sebulan dengan tipe data *float* karena kolom `totalGaji` berisikan bilangan pecahan positif presisi tunggal.
  8. `bulan` adalah kolom yang menunjukkan bulan diterimanya gaji staf dengan tipe data *char* karena pada kolom `bulan` berisikan data dengan panjang yang tetap. Kolom `tahun` memiliki panjang data 2.
  9. `tahun` adalah kolom yang menunjukkan tahun diterimanya gaji staf dengan tipe data *char* karena pada kolom `tahun` berisikan data dengan panjang yang tetap. Kolom `tahun` memiliki panjang data 4.
  10. `tanggalditerima` adalah kolom yang menunjukkan tanggal di kirimkannya gaji tersebut dengan tipe data *date* karena pada kolom `tanggalditerima`.
  11. `statusPengiriman` adalah kolom yang menunjukkan status terkirim atau belum dikirimnya gaji tersebut. Kolom `statusPengiriman` memiliki tipe data *enum* karena berisikan karakter atau nilai data yang sudah ditentukan sebelumnya yaitu '1' dan '0'.

## 4.4.2 Foreign Key Constraints

*Foreign Key Constraint* adalah *constraints* suatu kolom yang ada pada suatu tabel, dimana kolom tersebut juga dimiliki oleh tabel yang lain sebagai suatu *Primary Key*. Adapun beberapa tabel yang memiliki *Foreign Key Constraints* seperti tabel admin, tabel staff, tabel presensi dan tabel penggajian.

### 4.4.2.1 Foreign Key Constraints Tabel Admin

Actions	Constraint properties	Column	Foreign key constraint (INNODB)			
Drop	admin_ibfk_1	id_user	penggajianstaff	user	id_user	
	ON DELETE	CASCADE				
	ON UPDATE	CASCADE				

**Gambar 4. 28** Foreign Key Constraints Tabel Admin

Pengejelasan pada Gambar 4.28 adalah sebagai berikut :

Tabel admin mempunyai *constraints ON DELETE CASCADE* dan terhubung dengan tabel *user* yang artinya jika kolom pada tabel *user* dihapus, maka kolom pada tabel admin ikut terhapus. Tabel admin mempunyai *constraints ON UPDATE CASCADE* dan terhubung dengan tabel *user* yang artinya jika kolom pada tabel *user* diubah, maka kolom pada tabel admin ikut berubah datanya.

### 4.4.2.2 Foreign Key Constraints Tabel Staff

Actions	Constraint properties	Column	Foreign key constraint (INNODB)			
Drop	staff_ibfk_1	id_user	penggajianstaff	user	id_user	
	ON DELETE	CASCADE				
	ON UPDATE	CASCADE				
Drop	staff_ibfk_2	status	penggajianstaff	status_staff	status	
	ON DELETE	CASCADE				
	ON UPDATE	CASCADE				

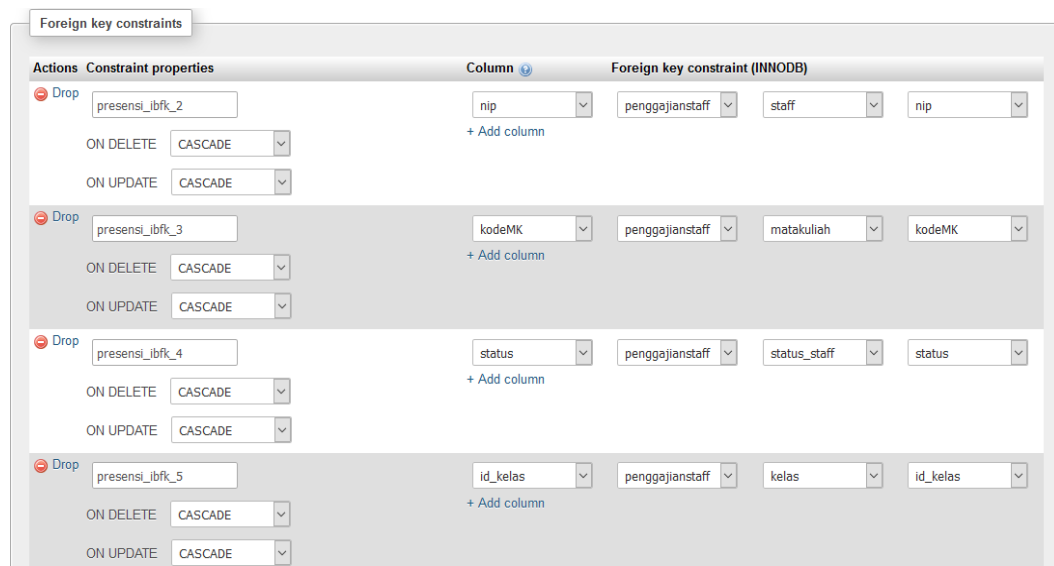
**Gambar 4. 29** Foreign Key Constraints Tabel Staff



Tabel *staff* mempunyai *constraints ON DELETE CASCADE* dan terhubung dengan tabel *user* yang artinya jika kolom pada tabel *user* dihapus, maka kolom pada tabel *staff* ikut terhapus. Tabel *staff* mempunyai *constraints ON UPDATE CASCADE* dan terhubung dengan tabel *user* yang artinya jika kolom pada tabel *user* diubah, maka kolom pada tabel *staff* ikut berubah datanya.

Tabel *staff* mempunyai *constraints ON DELETE CASCADE* dan terhubung dengan tabel *status\_staff* yang artinya jika kolom pada tabel *status\_staff* dihapus, maka kolom pada tabel *staff* ikut terhapus. Tabel *staff* mempunyai *constraints ON UPDATE CASCADE* dan terhubung pada tabel *status\_staff* yang artinya jika pada kolom tabel *status\_staff* diubah, maka kolom pada tabel *staff* ikut berubah datanya.

#### 4.4.2.3 Foreign Key Constraints Tabel Presensi



**Gambar 4. 30** *Foreign Key Constraints* Tabel Presensi

Tabel *presensi* mempunyai *constraints ON DELETE CASCADE* dan terhubung dengan tabel *staff* yang artinya jika kolom pada tabel *staff* dihapus, maka kolom pada tabel *presensi* ikut terhapus. Tabel *presensi* mempunyai *constraints ON UPDATE CASCADE* dan terhubung pada tabel *staff* yang artinya

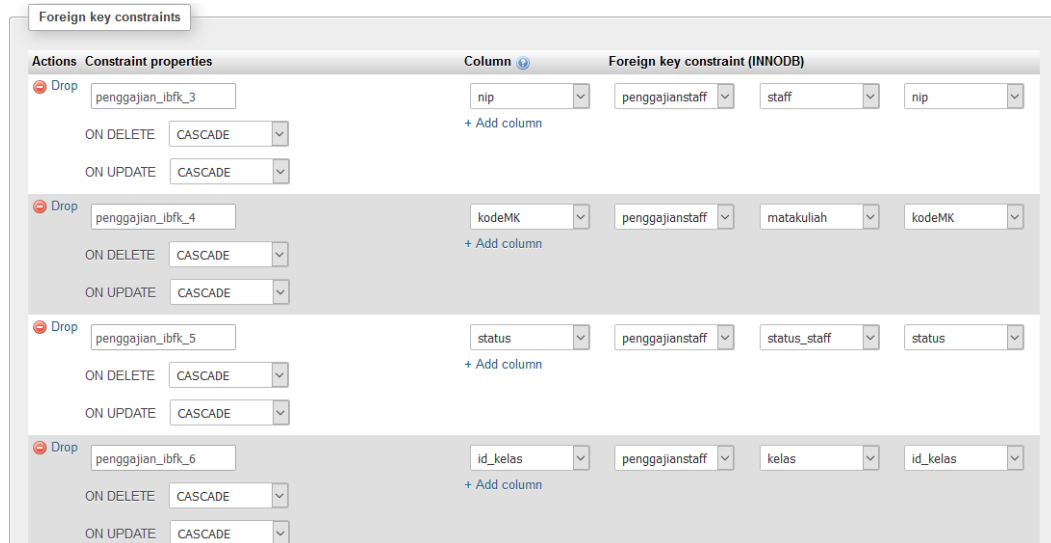
jika pada kolom tabel staff diubah, maka kolom pada tabel presensi ikut berubah datanya.

Tabel presensi mempunyai *constraints ON DELETE CASCADE* dan terhubung dengan tabel matakuliah yang artinya jika kolom pada tabel matakuliah dihapus, maka kolom pada tabel presensi ikut terhapus. Tabel presensi mempunyai *constraints ON UPDATE CASCADE* dan terhubung pada tabel matakuliah yang artinya jika pada kolom tabel matakuliah diubah, maka kolom pada tabel presensi ikut berubah datanya.

Tabel presensi mempunyai *constraints ON DELETE CASCADE* dan terhubung dengan tabel status\_staff yang artinya jika kolom pada tabel status\_staff dihapus, maka kolom pada tabel presensi ikut terhapus. Tabel presensi mempunyai *constraints ON UPDATE CASCADE* dan terhubung pada tabel status\_staff yang artinya jika pada kolom tabel status\_staff diubah, maka kolom pada tabel presensi ikut berubah datanya.

Tabel presensi mempunyai *constraints ON DELETE CASCADE* dan terhubung dengan tabel kelas yang artinya jika kolom pada tabel kelas dihapus, maka kolom pada tabel presensi ikut terhapus. Tabel presensi mempunyai *constraints ON UPDATE CASCADE* dan terhubung pada tabel kelas yang artinya jika pada kolom tabel kelas diubah, maka kolom pada tabel presensi ikut berubah datanya.

#### 4.4.2.4 Foreign Key Constraints Tabel Penggajian



**Gambar 4. 31** Foreign Key Constraints Tabel Penggajian

Tabel penggajian mempunyai *constraints ON DELETE CASCADE* dan terhubung dengan tabel staff yang artinya jika kolom pada tabel staff dihapus, maka kolom pada tabel penggajian ikut terhapus. Tabel penggajian mempunyai *constraints ON UPDATE CASCADE* dan terhubung pada tabel staff yang artinya jika pada kolom tabel staff diubah, maka kolom pada tabel penggajian ikut berubah datanya.

Tabel penggajian mempunyai *constraints ON DELETE CASCADE* dan terhubung dengan tabel matakuliah yang artinya jika kolom pada tabel matakuliah dihapus, maka kolom pada tabel penggajian ikut terhapus. Tabel penggajian mempunyai *constraints ON UPDATE CASCADE* dan terhubung pada tabel matakuliah yang artinya jika pada kolom tabel matakuliah diubah, maka kolom pada tabel penggajian ikut berubah datanya.

Tabel penggajian mempunyai *constraints ON DELETE CASCADE* dan terhubung dengan tabel status\_staff yang artinya jika kolom pada tabel status\_staff dihapus, maka kolom pada tabel penggajian ikut terhapus. Tabel penggajian mempunyai *constraints ON UPDATE CASCADE* dan terhubung pada

tabel `status_staff` yang artinya jika pada kolom tabel `status_staff` diubah, maka kolom pada tabel `penggajian` ikut berubah datanya.



Tabel `penggajian` mempunyai *constraints ON DELETE CASCADE* dan terhubung dengan tabel `kelas` yang artinya jika kolom pada tabel `kelas` dihapus, maka kolom pada tabel `penggajian` ikut terhapus. Tabel `penggajian` mempunyai *constraints ON UPDATE CASCADE* dan terhubung pada tabel `kelas` yang artinya jika pada kolom tabel `kelas` diubah, maka kolom pada tabel `penggajian` ikut berubah datanya.

#### 4.4.3 Kebijakan Akses User

Kebijakan Akses *User* dapat digunakan untuk mengamankan *database* agar data yang ada di dalamnya menjadi lebih aman. Kebijakan akses pada *MySQL* menyediakan berbagai tingkatan level kebijakan akses. Setiap *user* dapat dibatasi untuk dapat mengakses baik itu sebuah *database*, tabel atau bahkan kolom tertentu saja. Pada Sistem `Penggajian Staf` terdapat 3 jenis *user* diantaranya adalah :

##### 4.4.3.1 Kebijakan Akses Master

Pada kebijakan akses `Master` mempunyai hak penuh atas *database*. `Master` dapat melihat, mengubah, menghapus data pada *database* `Sistem Peggajian Staf` seperti pada gambar 4.32.

Database-specific privileges				
Database	Privileges	Grant	Table-specific privileges	Action
penggajianstaff	ALL PRIVILEGES	Yes	No	 Edit privileges  Revoke

Gambar 4. 32 *Privileges Master*

Pada gambar 4.33 merupakan kejelasan hak akses spesifik master adalah sebagai berikut :

The image shows a screenshot of the MySQL Database-specific privileges configuration interface. At the top, there is a header "Database-specific privileges" with a checked checkbox "Check all". Below this, a note states "Note: MySQL privilege names are expressed in English." The interface is divided into three main sections: "Data", "Structure", and "Administration", each with a checked checkbox. The "Data" section includes SELECT, INSERT, UPDATE, and DELETE. The "Structure" section includes CREATE, ALTER, INDEX, DROP, CREATE TEMPORARY TABLES, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, EXECUTE, CREATE VIEW, EVENT, and TRIGGER. The "Administration" section includes GRANT, LOCK TABLES, and REFERENCES.

Category	Privileges
Data	SELECT, INSERT, UPDATE, DELETE
Structure	CREATE, ALTER, INDEX, DROP, CREATE TEMPORARY TABLES, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, EXECUTE, CREATE VIEW, EVENT, TRIGGER
Administration	GRANT, LOCK TABLES, REFERENCES

**Gambar 4. 33** Hak Akses Spesifik Master

#### 4.4.3.2 Kebijakan Akses Admin

Pada kebijakan akses Admin dilakukan untuk keamanan data, apabila ada seorang yang dapat membobol page admin. Maka kemungkinan data yang ada didalam *database* masih tetap aman.

Table	Privileges	Grant	Column-specific privileges	Action
admin	SELECT, INSERT, UPDATE	No	No	Edit privileges  Revoke
kelas	SELECT, INSERT, UPDATE, DELETE	No	No	Edit privileges  Revoke
matakuliah	SELECT, INSERT, UPDATE, DELETE	No	No	Edit privileges  Revoke
penggajian	SELECT, INSERT, UPDATE, DELETE	No	No	Edit privileges  Revoke
presensi	SELECT, INSERT, UPDATE, DELETE	No	No	Edit privileges  Revoke
staff	SELECT, INSERT, UPDATE, DELETE	No	No	Edit privileges  Revoke
status_staff	SELECT, UPDATE	No	No	Edit privileges  Revoke

**Gambar 4. 34** Privileges Admin

1. Login, melihat, menambah dan mengubah data pada tabel admin

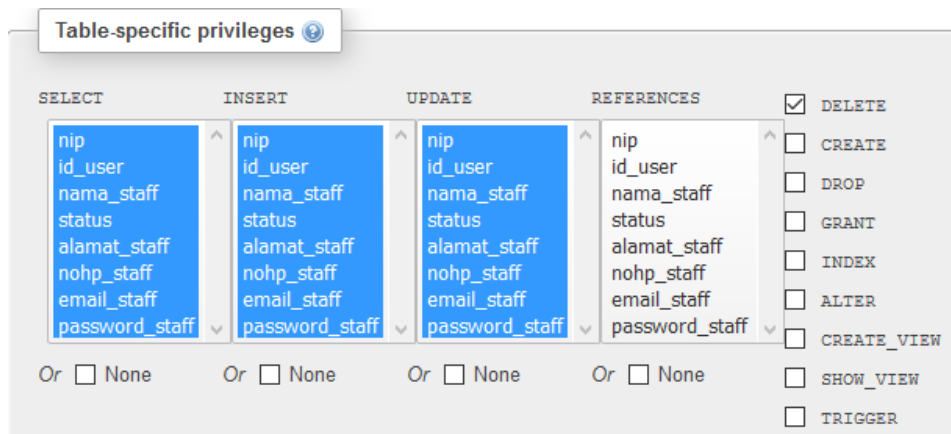
Pada gambar 4.35 merupakan batasan-batasan hak akses dari admin adalah sebagai berikut :

SELECT	INSERT	UPDATE	REFERENCES	DELETE
id_admin id_user email_admin password_admin	id_admin id_user email_admin password_admin	id_admin id_user email_admin password_admin	id_admin id_user email_admin password_admin	<input type="checkbox"/>
Or <input type="checkbox"/> None	Or <input type="checkbox"/> None	Or <input type="checkbox"/> None	Or <input type="checkbox"/> None	<input type="checkbox"/> CREATE
				<input type="checkbox"/> DROP
				<input type="checkbox"/> GRANT
				<input type="checkbox"/> INDEX
				<input type="checkbox"/> ALTER
				<input type="checkbox"/> CREATE_VIEW
				<input type="checkbox"/> SHOW_VIEW
				<input type="checkbox"/> TRIGGER

**Gambar 4. 35** Privileges Specific Admin tabel Admin

2. Melihat, menambah, mengubah, menghapus data pada tabel staff

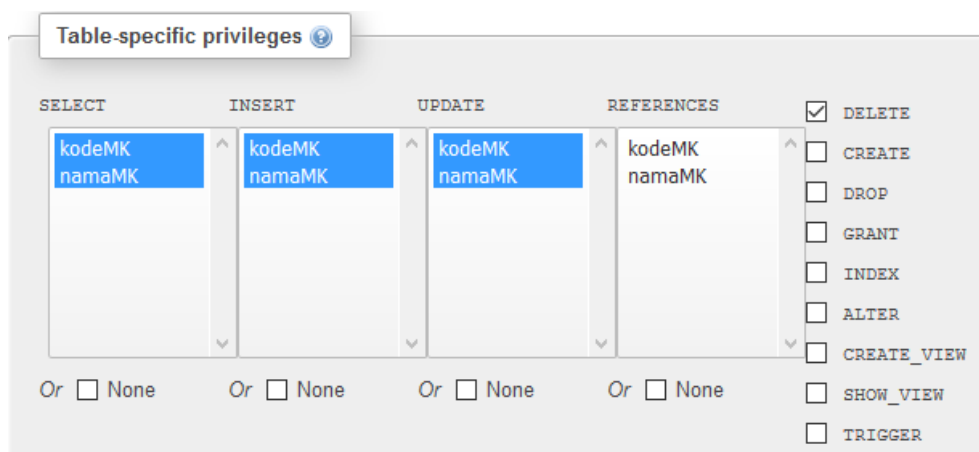
Pada gambar 4.36 merupakan batasan-batasan hak akses dari admin adalah sebagai berikut :



**Gambar 4. 36** Privileges Specific Admin tabel Staff

3. Melihat, menambah, mengubah, menghapus data pada tabel matakuliah

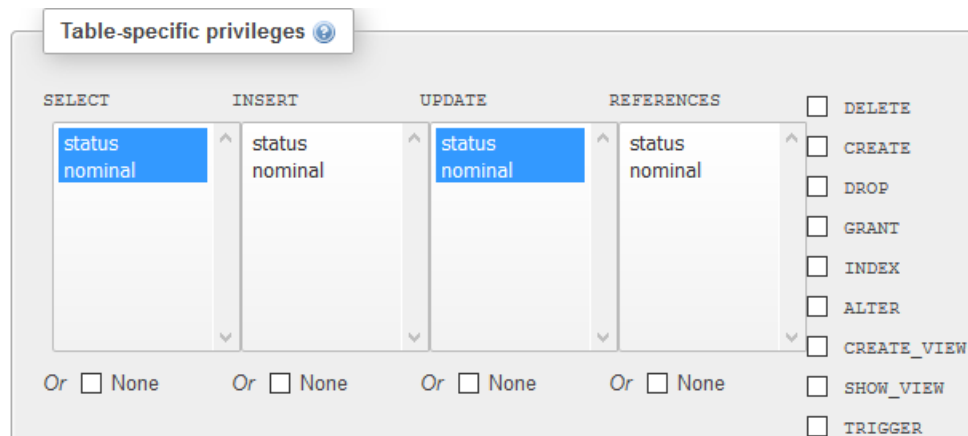
Pada gambar 4.37 merupakan batasan-batasan hak akses dari admin adalah sebagai berikut :



**Gambar 4. 37** Privileges Specific Admin tabel Staff

4. Melihat dan mengubah data pada tabel status\_staff

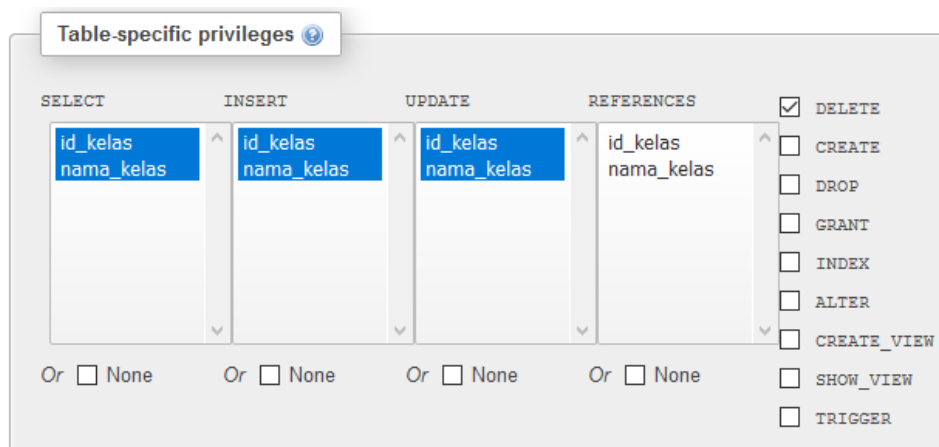
Pada gambar 4.38 merupakan batasan-batasan hak akses dari admin adalah sebagai berikut :



**Gambar 4. 38** *Privileges Specific Admin* tabel Status Staff

5. Melihat, menambah, mengubah, menghapus data pada tabel kelas

Pada gambar 4.39 merupakan batasan-batasan hak akses dari admin adalah sebagai berikut :

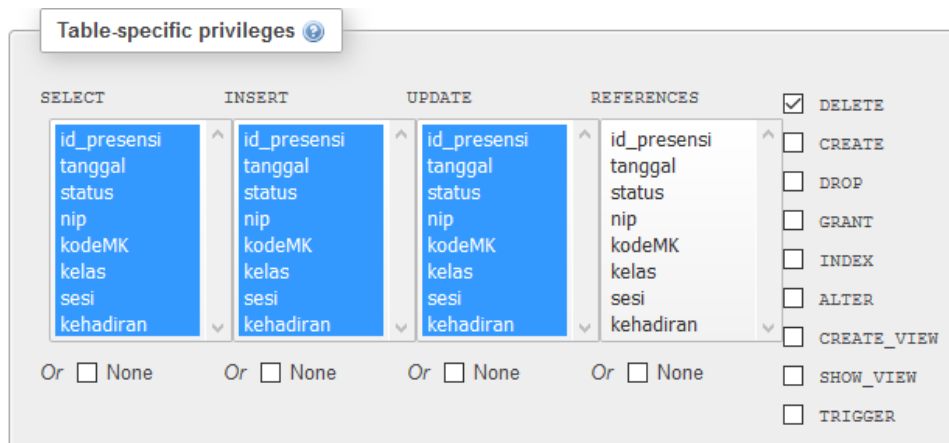


**Gambar 4. 39** *Privileges Specific Admin* tabel Kelas



6. Melihat, menambah, mengubah, menghapus data pada tabel presensi

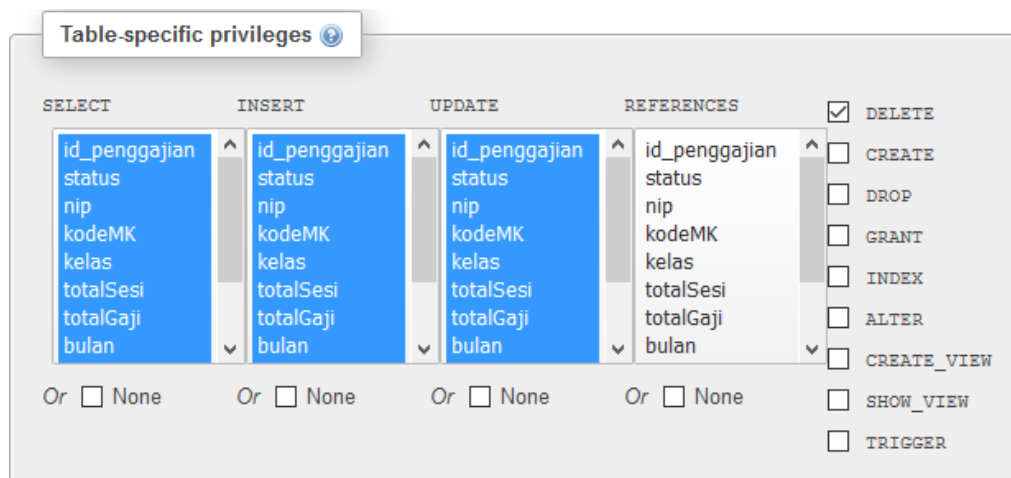
Pada gambar 4.40 merupakan batasan-batasan hak akses dari admin adalah sebagai berikut :



**Gambar 4. 40** Privileges Specific Admin tabel Presensi

7. Melihat, menambah, mengubah, menghapus data pada tabel penggajian

Pada gambar 4.41 merupakan batasan-batasan hak akses dari admin adalah sebagai berikut :



**Gambar 4. 41** Privileges Specific Admin tabel Penggajian

#### 4.4.3.1 Kebijakan Akses Staf

Pada kebijakan akses Staf, staf hanya dapat melakukan satu kebijakan akses ke tabel penggajian yang ada di *database* sistem penggajian staf.

Table-specific privileges				
Table	Privileges	Grant	Column-specific privileges	Action
penggajian	SELECT	No	No	Edit privileges  Revoke
staff	USAGE	No	Yes	Edit privileges  Revoke

**Gambar 4. 42** *Privileges Staff*

Berikut adalah kebijakan akses yang hanya bisa dilakukan staf :

1. Login dan melihat data pada tabel penggajian

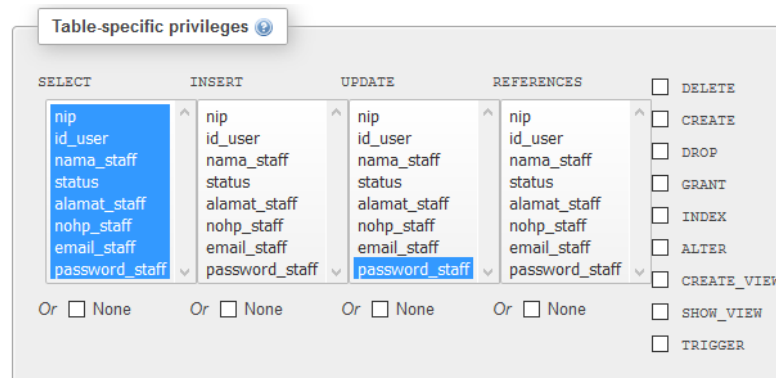
Pada gambar 4.43 merupakan batasan-batasan hak akses dari staf adalah sebagai berikut :

Table-specific privileges				
SELECT	INSERT	UPDATE	REFERENCES	
<input checked="" type="checkbox"/> id_penggajian	<input checked="" type="checkbox"/> id_penggajian	<input checked="" type="checkbox"/> id_penggajian	<input checked="" type="checkbox"/> id_penggajian	<input type="checkbox"/> DELETE
<input checked="" type="checkbox"/> status	<input checked="" type="checkbox"/> status	<input checked="" type="checkbox"/> status	<input checked="" type="checkbox"/> status	<input type="checkbox"/> CREATE
<input checked="" type="checkbox"/> nip	<input checked="" type="checkbox"/> nip	<input checked="" type="checkbox"/> nip	<input checked="" type="checkbox"/> nip	<input type="checkbox"/> DROP
<input checked="" type="checkbox"/> kodeMK	<input checked="" type="checkbox"/> kodeMK	<input checked="" type="checkbox"/> kodeMK	<input checked="" type="checkbox"/> kodeMK	<input type="checkbox"/> GRANT
<input checked="" type="checkbox"/> kelas	<input checked="" type="checkbox"/> kelas	<input checked="" type="checkbox"/> kelas	<input checked="" type="checkbox"/> kelas	<input type="checkbox"/> INDEX
<input checked="" type="checkbox"/> totalSesi	<input checked="" type="checkbox"/> totalSesi	<input checked="" type="checkbox"/> totalSesi	<input checked="" type="checkbox"/> totalSesi	<input type="checkbox"/> ALTER
<input checked="" type="checkbox"/> totalGaji	<input checked="" type="checkbox"/> totalGaji	<input checked="" type="checkbox"/> totalGaji	<input checked="" type="checkbox"/> totalGaji	<input type="checkbox"/> CREATE_VIEW
<input checked="" type="checkbox"/> tanggalditerima	<input checked="" type="checkbox"/> tanggalditerima	<input checked="" type="checkbox"/> tanggalditerima	<input checked="" type="checkbox"/> tanggalditerima	<input type="checkbox"/> SHOW_VIEW
Or <input type="checkbox"/> None	Or <input type="checkbox"/> None	Or <input type="checkbox"/> None	Or <input type="checkbox"/> None	<input type="checkbox"/> TRIGGER

**Gambar 4. 43** *Privileges Specific Staff tabel Penggajian*

## 2. Melihat dan mengubah password pada tabel staff

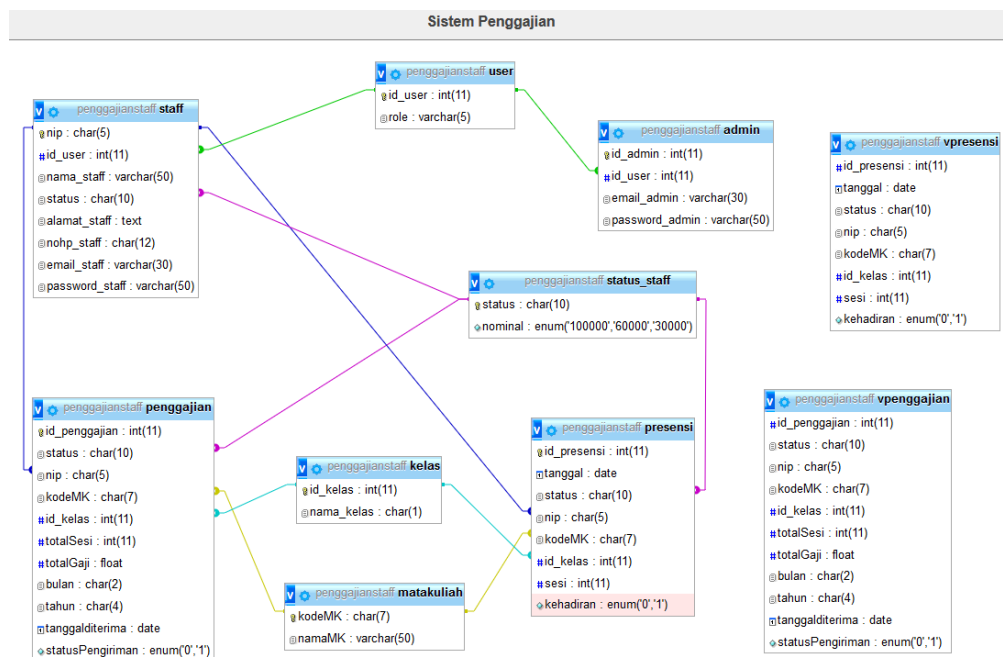
Pada gambar 4.44 merupakan batasan-batasan hak akses dari staf adalah sebagai berikut :



Gambar 4. 44 Privileges Specific Staff tabel Staff

### 4.4.4 RAT (Rancangan Antar Tabel)

Rancangan Antar Tabel pada Sistem Penggajian Staf dapat dilihat pada Gambar 4.45 :



Gambar 4. 45 RAT Sistem Penggajian Staf

Penjelasan dari rancangan antar tabel pada Gambar 4.45 adalah sebagai berikut :

1. Garis warna hijau menunjukkan adanya relasi dari tabel *user* dengan tabel admin dan tabel staff.
2. Garis warna biru tua menunjukkan adanya relasi dari tabel staff dengan tabel presensi dan tabel penggajian.
3. Garis warna kuning menunjukkan adanya relasi dari tabel matakuliah dengan tabel presensi dan tabel penggajian.
4. Garis berwarna ungu menunjukkan adanya relasi dari tabel status\_staff dengan tabel staff, tabel presensi dan tabel penggajian.
5. Garis berwarna biru muda menunjukkan adanya relasi dari tabel kelas dengan tabel presensi dan tabel penggajian.

## **4.5 Pengujian**

### **4.5.1 Metode Pengujian**

Pengujian perangkat lunak dilakukan untuk memperoleh informasi serta mengevaluasi kualitas dari produk atau yang layanan yang sedang diuji. Tujuan pengujian dalam pengembangan *database* adalah untuk mengetahui apakah *database* yang diujikan dapat memenuhi kebutuhan *admin* dan *user* dengan mendasari pada rancangan dan pengembangan perangkat lunak. Berikut ini hasil pengujian anomali dan pengujian *view check* dilakukan:

### **4.5.2 Pengujian Anomali**

Pengujian *anomali* dilakukan terhadap tabel-tabel yang memiliki relasi dari tabel satu ke tabel lainnya. Pengujian dilakukan sebanyak 3 kali yaitu *insert anomali*, *delete anomali* dan *update anomali*. Berikut adalah tabel yang memiliki relasi dengan tabel lain diantaranya adalah :

1. Tabel *user* memiliki relasi dengan tabel admin dan tabel staff.
2. Tabel staff memiliki relasi dengan tabel presensi dan tabel penggajian.
3. Tabel matakuliah memiliki relasi dengan tabel presensi dan tabel penggajian.

4. Tabel `status_staff` memiliki relasi dengan tabel `staff`, tabel `presensi` dan tabel `penggajian`.
5. Tabel `kelas` memiliki relasi dengan tabel `presensi` dan tabel `penggajian`

**a. Pengujian Anomali pada Tabel user**

1. *Insert* anomali

*Insert* anomali dilakukan pada tabel `user` karena tabel `user` memiliki relasi dengan tabel `admin` dan tabel `staff`. Pengujian *insert* anomali dapat dilihat pada tabel 4.26.

**Tabel 4. 26** Pengujian *Insert* Tabel User

Tabel <i>User</i>			
<code>id_user</code>	<code>role</code>		
1	Admin		
2	Staff		
3	TU		

Tabel Admin			
<code>id_admin</code>	<code>id_user</code>	<code>email_admin</code>	<code>password_admin</code>
4545	1	adminnie@gmail.com	098f6bcd4621d373cade4e832627b4f6
4546	1	nadyaaprillia@gmail.com	babd26213f50f2906fae75698e24d1c1
4547	1	hartikarahay@gmail.com	0c14e488a51a54689d0f8a9d2000c5ff
4548	1	nisrinaarp@gmail.com	d7f51cb5be62883c3570409bba87bc64
4549	3	budi@gmail.com	750462c088103c9632e5a746a4b0dba5

Tabel Staff							
nip	id_user	nama_staff	status	alamat_staff	nohp_staff	email_staff	password_staff
A01	2	Ohno Satoshi	Asisten	Tokyo	934567891	satoshihno@yahoo.jp	7c978c1228b60cc41787020695f8cb77
A02	2	Ninomiya Kazunari	Asisten	Tokyo	86523456890	ninomiyakazunari@yahoo.jp	d5433b40b38af99be049364c502c88dd
D01	2	Aprillia Kumiati	Dosen	Jalan Magelang	987654322123	apriiakumiati@gmail.com	e4b3a0f4e7348df27ce5b124c09f01ac
D02	2	Slamet Riyadi	Dosen	Yogyakarta	87654323678	slametriyadi@gmail.com	92a968ca99da34abeeb3f2042f973cab
I01	2	Sakurai Sho	Instruktur	Tokyo, Japan	975367112	sakuraisho@yahoo.jp	6ec28e0d3a75899f36a7b7c56bac3eec
I02	2	Matshumoto Jun	Instruktur	Japan	23456789098	matshumotojun@yahoo.jp	ce5ec93770844b25ef0223b169a2f08a

Setelah dilakukan *insert* data pada tabel *user*, tidak adanya perubahan yang terjadi pada tabel admin dan tabel staff. Maka tidak ada terjadi anomali pada pemasukan data.

## 2. *Update* anomali

*Update* anomali dilakukan pada tabel *user* karena tabel *user* memiliki relasi dengan tabel admin dan tabel staff. Pengujian *update* anomali dapat dilihat pada tabel 4.27.

**Tabel 4. 27** Pengujian *Update* Tabel User

Tabel <i>User</i>	
id_user	role
12	Admin
22	Staff
33	TU

Tabel Admin

id_admin	id_user	email_admin	password_admin
4545	11	adminnie@gmail.com	098f6bcd4621d373cade4e832627b4f6
4546	11	nadyaaprillia@gmail.com	babd26213f50f2906fae75698e24d1c1
4547	11	hartikarahay@gmail.com	0c14e488a51a54689d0f8a9d2000c5ff
4548	11	nisrinaarp@gmail.com	d7f51cb5be62883c3570409bba87bc64
4549	33	budi@gmail.com	750462c088103c9632e5a746a4b0dba5

Tabel Staff

nip	id_user	nama_staff	status	alamat_staff	nohp_staff	email_staff	password_staff
A01	22	Ohno Satoshi	Asisten	Tokyo	934567891	satoshiohno@yahoo.jp	7c978c1228b60cc41787020695f8cb77
A02	22	Ninomiya Kazunari	Asisten	Tokyo	86523456890	ninomiyakazunari@yahoo.jp	d5433b40b38af99be049364c502c88dd
D01	22	Aprillia Kurniati	Dosen	Jalan Magelang	987654322123	aprilliakurniati@gmail.com	e4b3a0f4e7348df27ce5b124c09f01ac
D02	22	Slamet Riyadi	Dosen	Yogyakarta	87654323678	slametriyadi@gmail.com	92a968ca99da34abeeb3f2042f973cab
I01	22	Sakurai Sho	Instruktur	Tokyo, Japan	975367112	sakuraisho@yahoo.jp	6ec28e0d3a75899f36a7b7c56bac3eec
I02	22	Matshumoto Jun	Instruktur	Japan	23456789098	matshumotojun@yahoo.jp	ce5ec93770844b25ef0223b169a2f08a

Setelah dilakukan *update* data pada tabel *user*, tabel yang memiliki relasi dan memiliki data yang sama akan berubah mengikuti data yang di *update* pada tabel *user*, menandakan bahwa tidak terjadi anomali saat data di *update*.

### 3. Delete anomali

*Delete* anomali dilakukan pada tabel *user* karena tabel *user* memiliki relasi dengan tabel *admin* dan tabel *staff*. Pengujian *delete* anomali dapat dilihat pada tabel 4.28.

**Tabel 4. 28** Pengujian *Delete* Tabel User

Tabel User			
<b>id_user</b>	<b>role</b>		
1	Admin		
2	Staff		

Tabel Admin			
<b>id_admin</b>	<b>id_user</b>	<b>email_admin</b>	<b>password_admin</b>
4545	11	adminnie@gmail.com	098f6bcd4621d373cade4e832627b4f6
4546	11	nadyaaprilla@gmail.com	babd26213f50f2906fae75698e24d1c1
4547	11	hartikarahay@gmail.com	0c14e488a51a54689d0f8a9d2000c5ff
4548	11	nisrinaarp@gmail.com	d7f51cb5be62883c3570409bba87bc64

Tabel Staff							
<b>nip</b>	<b>id_user</b>	<b>nama_staff</b>	<b>status</b>	<b>alamat_staff</b>	<b>nohp_staff</b>	<b>email_staff</b>	<b>password_staff</b>
A01	22	Ohno Satoshi	Asisten	Tokyo	934567891	satoshihno@yahoo.jp	7c978c1228b60cc41787020695f8cb77
A02	22	Ninomiya Kazunari	Asisten	Tokyo	86523456890	ninomiya kazunari@yahoo.jp	d5433b40b38af99be049364c502c88dd
D01	22	Aprillia Kurniati	Dosen	Jalan Magelang	987654322123	apriilikurniati@gmail.com	e4b3a0f4e7348df27ce5b124c09f01ac
D02	22	Slamet Riyadi	Dosen	Yogyakarta	87654323678	slametriyadi@gmail.com	92a968ca99da34abeeb3f2042f973cab
I01	22	Sakurai Sho	Instruktur	Tokyo, Japan	975367112	sakuraisho@yahoo.jp	6ec28e0d3a75899f36a7b7c56bac3eec
I02	22	Matshumoto Jun	Instruktur	Japan	23456789098	matshumotojun@yahoo.jp	ce5ec93770844b25ef0223b169a2f08a

Setelah dilakukan *delete* pada salah satu data dari tabel *user*, data dari tabel *admin* dan tabel *staff* dapat terhapus. Data yang terhapus pada tabel



admin dan tabel staff hanya data yang memiliki relasi dengan data yang dihapus pada tabel *user*, maka tidak terjadi anomali saat data dihapus.

## b. Pengujian Anomali pada Tabel staff

### 1. *Insert* anomali

*Insert* anomali dilakukan pada tabel staff karena tabel staff memiliki relasi dengan tabel presensi dan tabel penggajian. Pengujian *insert* anomali dapat dilihat pada tabel 4.29.

**Tabel 4. 29** Pengujian *Insert* Tabel Staff

Tabel Staff							
nip	id_user	nama_staff	status	alamat_staff	nohp_staff	email_staff	password_staff
A01	2	Ohno Satoshi	Asisten	Tokyo	934567891	satoshihno@yahoo.jp	7c978c1228b60cc41787020695f8cb77
A02	2	Ninomiya Kazunari	Asisten	Tokyo	86523456890	ninomiyakazunari@yahoo.jp	d5433b40b38af99be049364c502c88dd
D01	2	Aprillia Kurniati	Dosen	Jalan Magelang	987654322123	apriiakurniati@gmail.com	e4b3a0f4e7348df27ce5b124c09f01ac
D02	2	Slamet Riyadi	Dosen	Yogyakarta	87654323678	slametriyadi@gmail.com	92a968ca99da34abeeb3f2042f973cab
I01	2	Sakurai Sho	Instruktur	Tokyo, Japan	975367112	sakuraisho@yahoo.jp	6ec28e0d3a75899f36a7b7c56bac3eec
I02	2	Matshumoto Jun	Instruktur	Japan	23456789098	matshumotojun@yahoo.jp	ce5ec93770844b25ef0223b169a2f08a

Tabel Presensi							
id_presensi	tanggal	status	nip	kodeMK	id_kelas	sesi	kehadiran
1	2017-12-04	Dosen	D01	TI-102	3	1	1
2	2017-12-11	Dosen	D01	TI-102	3	2	1
3	2017-12-18	Dosen	D02	TI-101	1	1	1
4	2017-12-26	Dosen	D02	TI-101	1	1	1
5	2017-11-01	Instruktur	I01	TI-105	2	1	1
6	2017-11-01	Instruktur	I01	TI-105	2	2	1
7	2017-12-08	Instruktur	I01	TI-105	2	3	1
8	2017-12-04	Asisten	A01	TI-103	4	1	1
9	2017-12-11	Asisten	A01	TI-103	4	2	1

Tabel Penggajian										
id_penggajian	status	nip	kodeMK	id_kelas	totalSesi	totalGaji	bulan	tahun	tanggalditerima	statusPengiriman
1	Dosen	D01	TI-102	3	4	400000	12	2017	2017-12-09	1
2	Asisten	A01	TI-103	4	2	60000	12	2017	2017-12-15	1
3	Instruktur	I01	TI-105	2	3	180000	12	2017	2017-12-08	1

Setelah dilakukan *insert* data pada tabel staff, tidak adanya perubahan yang terjadi pada tabel presensi dan tabel penggajian. Maka tidak ada terjadi anomali pada pemasukan data.

## 2. *Update* anomali

*Update* anomali dilakukan pada tabel staff karena tabel staff memiliki relasi dengan tabel presensi dan tabel penggajian. Pengujian *update* anomlai dapat dilihat pada tabel 4.30.

**Tabel 4. 30** Pengujian *Update* Tabel Staff

Tabel Staff							
nip	id_user	nama_staff	status	alamat_staff	nohp_staff	email_staff	password_staff
A01	2	Ohno Satoshi	Asisten	Tokyo	934567891	satoshiohno@yahoo.jp	7c978c1228b60cc41787020695f8cb77
A02	2	Ninomiya Kazunari	Asisten	Tokyo	86523456890	ninomiyakazunari@yahoo.jp	d5433b40b38af99be049364c502c88dd
D02	2	Slamet Riyadi	Dosen	Yogyakarta	87654323678	slametriyadi@gmail.com	92a968ca99da34abeeb3f2042f973cab
D04	2	Aprillia Kurniati	Dosen	Jalan Magelang	987654322123	apriiliakurniati@gmail.com	e4b3a0f4e7348df27ce5b124c09f01ac
I01	2	Sakurai Sho	Instruktur	Tokyo, Japan	975367112	sakuraisho@yahoo.jp	6ec28e0d3a75899f36a7b7c56bac3eec
I02	2	Matshumoto Jun	Instruktur	Japan	23456789098	matshumotojun@yahoo.jp	ce5ec93770844b25ef0223b169a2f08a

Tabel Presensi							
id_presensi	tanggal	status	nip	kodeMK	id_kelas	sesi	kehadiran
1	2017-12-04	Dosen	D04	TI-102	3	1	1
2	2017-12-11	Dosen	D04	TI-102	3	2	1
3	2017-12-18	Dosen	D02	TI-101	1	1	1
4	2017-12-26	Dosen	D02	TI-101	1	1	1
5	2017-11-01	Instruktur	I01	TI-105	2	1	1
6	2017-11-01	Instruktur	I01	TI-105	2	2	1
7	2017-12-08	Instruktur	I01	TI-105	2	3	1
8	2017-12-04	Asisten	A01	TI-103	4	1	1
9	2017-12-11	Asisten	A01	TI-103	4	2	1

Tabel Penggajian										
id_penggajian	status	nip	kodeMK	id_kelas	totalSesi	totalGaji	bulan	tahun	tanggalditerima	statusPengiriman
1	Dosen	D04	TI-102	3	4	400000	12	2017	2017-12-09	1
2	Asisten	A01	TI-103	4	2	60000	12	2017	2017-12-15	1
3	Instruktur	I01	TI-105	2	3	180000	12	2017	2017-12-08	1

Setelah dilakukan *update* data pada tabel staff, tabel yang memiliki relasi dan memiliki data yang sama akan berubah mengikuti data yang di *update* pada tabel staff, menandakan bahwa tidak terjadi anomali saat data di *update*.

### 3. Delete anomali

*Delete* anomali dilakukan pada tabel staff karena tabel staff memiliki relasi dengan tabel presensi dan tabel penggajian. Pengujian *delete* anomali dapat dilihat pada tabel 4.31.

**Tabel 4. 31** Pengujian *Delete* Tabel Staff

Tabel Staff							
nip	id_user	nama_staff	status	alamat_staff	nohp_staff	email_staff	password_staff
A01	2	Ohno Satoshi	Asisten	Tokyo	934567891	satoshihno@yahoo.jp	7c978c1228b60cc41787020695f8cb77
A02	2	Ninomiya Kazunari	Asisten	Tokyo	86523456890	ninomiyakazunari@yahoo.jp	d5433b40b38af99be049364c502c88dd
D02	2	Slamet Riyadi	Dosen	Yogyakarta	87654323678	slametriyadi@gmail.com	92a968ca99da34abeeb3f2042f973cab
I01	2	Sakurai Sho	Instruktur	Tokyo, Japan	975367112	sakuraisho@yahoo.jp	6ec28e0d3a75899f36a7b7c56bac3eec
I02	2	Matshumoto Jun	Instruktur	Japan	23456789098	matshumotojun@yahoo.jp	ce5ec93770844b25ef0223b169a2f08a

Tabel Presensi							
id_presensi	tanggal	status	nip	kodeMK	id_kelas	sesi	kehadiran
3	2017-12-18	Dosen	D02	TI-101	1	1	1
4	2017-12-26	Dosen	D02	TI-101	1	1	1
5	2017-11-01	Instruktur	I01	TI-105	2	1	1
6	2017-11-01	Instruktur	I01	TI-105	2	2	1
7	2017-12-08	Instruktur	I01	TI-105	2	3	1
8	2017-12-04	Asisten	A01	TI-103	4	1	1
9	2017-12-11	Asisten	A01	TI-103	4	2	1

Tabel Penggajian										
id_penggajian	status	nip	kodeMK	id_kelas	totalSesi	totalGaji	bulan	tahun	tanggalditerima	statusPengiriman
2	Asisten	A01	TI-103	4	2	60000	12	2017	2017-12-15	1
3	Instruktur	I01	TI-105	2	3	180000	12	2017	2017-12-08	1

Setelah dilakukan *delete* pada salah satu data dari tabel staff, data dari tabel staff dapat terhapus. Data yang terhapus pada tabel presensi dan tabel penggajian hanya data yang memiliki relasi dengan data yang dihapus pada tabel staff, maka tidak terjadi anomali saat data dihapus.

### c. Pengujian Anomali pada Tabel matakuliah

#### 1. *Insert* anomali

*Insert* anomali dilakukan pada tabel mata kuliah karena tabel mata kuliah memiliki relasi dengan tabel presensi dan tabel penggajian. Pengujian *insert* anomali dapat dilihat pada tabel 4.32.

**Tabel 4. 32** Pengujian *Insert* Tabel Mata Kuliah

Tabel Mata Kuliah	
kodeMK	namaMK
TI-101	Agama Islam 1
TI-102	Bahasa Inggris 1
TI-103	Fisika dan Elektronika
TI-104	Dasar Teknologi Informasi
TI-105	Applikasi Produktifitas Kerja

Tabel Presensi							
id_presensi	tanggal	status	nip	kodeMK	id_kelas	sesi	kehadiran
1	2017-11-06	Dosen	D04	TI-102	1	1	1
2	2017-11-13	Dosen	D04	TI-102	1	2	1
3	2017-11-20	Dosen	D04	TI-102	1	3	1
4	2017-11-27	Dosen	D04	TI-103	1	4	1
5	2017-11-01	Instruktur	I01	TI-105	2	1	1
6	2017-11-01	Instruktur	I01	TI-105	2	2	1
9	2017-11-08	Instruktur	I01	TI-105	2	3	1

Tabel Penggajian										
id_penggajian	status	nip	kodeMK	id_kelas	totalSesi	totalGaji	bulan	tahun	tanggalditerima	statusPengiriman
1	Dosen	D04	TI-102	1	4	400000	11	2017	2017-11-30	1
2	Instruktur	I01	TI-105	2	9	540000	11	2017	2017-11-30	1
4	Dosen	D02	TI-101	1	1	100000	11	2017	2017-10-31	1

Setelah dilakukan *insert* data pada tabel mata kuliah, tidak adanya perubahan yang terjadi pada tabel presensi dan tabel penggajian. Maka tidak ada terjadi anomali pada pemasukan data.

## 2. Update anomali

*Update* anomali dilakukan pada tabel mata kuliah karena tabel mata kuliah memiliki relasi dengan tabel presensi dan tabel penggajian. Pengujian *update* anomlai dapat dilihat pada tabel 4.33.

**Tabel 4. 33** Pengujian *Update* Tabel Mata Kuliah

Tabel Mata Kuliah							
kodeMK	namaMK						
TI-101	Agama Islam 1						
TI-102P	Bahasa Inggris 1						
TI-103	Fisika dan Elektronika						
TI-104	Dasar Teknologi Informasi						
TI-105	Applikasi Produktifitas Kerja						

Tabel Presensi							
id_presensi	tanggal	status	nip	kodeMK	id_kelas	sesi	kehadiran
1	2017-11-06	Dosen	D04	TI-102P	1	1	1
2	2017-11-13	Dosen	D04	TI-102P	1	2	1
3	2017-11-20	Dosen	D04	TI-102P	1	3	1
4	2017-11-27	Dosen	D04	TI-103	1	4	1
5	2017-11-01	Instruktur	I01	TI-105	2	1	1
6	2017-11-01	Instruktur	I01	TI-105	2	2	1
9	2017-11-08	Instruktur	I01	TI-105	2	3	1

Tabel Penggajian

id_penggajian	status	nip	kodeMK	id_kelas	totalSesi	totalGaji	bulan	tahun	tanggalditerima	statusPengiriman
1	Dosen	D04	TI-102P	1	4	400000	11	2017	2017-11-30	1
2	Instruktur	I01	TI-105	2	9	540000	11	2017	2017-11-30	1
4	Dosen	D02	TI-101	1	1	100000	11	2017	2017-10-31	1

Setelah dilakukan *update* pada tabel mata kuliah, tabel yang memiliki relasi dan memiliki data yang sama akan berubah mengikuti data yang di *update* pada tabel mata kuliah, menandakan bahwa tidak terjadi anomali saat data di *update*.

### 3. *Delete* anomali

*Delete* anomali dilakukan pada tabel mata kuliah karena tabel mata kuliah memiliki relasi dengan tabel presensi dan tabel penggajian. Pengujian *delete* anomali dapat dilihat pada tabel 4.34.

**Tabel 4. 34** Pengujian *Delete* Tabel Mata Kuliah

Tabel Mata Kuliah

kodeMK	namaMK
TI-101	Agama Islam 1
TI-103	Fisika dan Elektronika
TI-104	Dasar Teknologi Informasi
TI-105	Applikasi Produktifitas Kerja



Tabel Presensi							
id_presensi	tanggal	status	nip	kodeMK	id_kelas	sesi	kehadiran
4	2017-11-27	Dosen	D04	TI-103	1	4	1
5	2017-11-01	Instruktur	I01	TI-105	2	1	1
6	2017-11-01	Instruktur	I01	TI-105	2	2	1
9	2017-11-08	Instruktur	I01	TI-105	2	3	1

Tabel Penggajian										
id_penggajian	status	nip	kodeMK	id_kelas	totalSesi	totalGaji	bulan	tahun	tanggalditerima	statusPengiriman
2	Instruktur	I01	TI-105	2	9	540000	11	2017	2017-11-30	1
4	Dosen	D02	TI-101	1	1	100000	11	2017	2017-10-31	1

Setelah dilakukan *delete* data pada tabel mata kuliah, data dari tabel mata kuliah dapat terhapus. Data yang terhapus pada tabel presensi dan tabel penggajian hanya data yang memiliki relasi dengan data yang dihapus pada tabel mata kuliah, maka tidak terjadi anomali saat data dihapus.

#### d. Pengujian Anomali pada Tabel status\_staff

##### 1. *Insert* anomali

*Insert* anomali dilakukan pada tabel status staff karena tabel status staff memiliki relasi dengan tabel staff, tabel presensi dan tabel penggajian. Pengujian *insert* anomali dapat dilihat pada tabel 4.35.

**Tabel 4. 35** Pengujian *Insert* Tabel Status Staff

Tabel Status Staff							
<b>status</b>	<b>nominal</b>						
Asisten	30000						
Dosen	100000						
Instruktur	60000						

Tabel Staff							
nip	id_user	nama_staff	status	alamat_staff	nohp_staff	email_staff	password_staff
A01	2	Ohno Satoshi	Asisten	Tokyo	934567891	satoshiohno@yahoo.jp	7c978c1228b60cc41787020695f8cb77
A02	2	Ninomiya Kazunari	Asisten	Tokyo	86523456890	ninomiyakazunari@yahoo.jp	d5433b40b38af99be049364c502c88dd
D02	2	Slamet Riyadi	Dosen	Yogyakarta	87654323678	slametriyadi@gmail.com	92a968ca99da34abeeb3f2042f973cab
I01	2	Sakurai Sho	Instruktur	Tokyo, Japan	975367112	sakuraisho@yahoo.jp	6ec28e0d3a75899f36a7b7c56bac3eec
I02	2	Matshumoto Jun	Instruktur	Japan	23456789098	matshumotojun@yahoo.jp	ce5ec93770844b25ef0223b169a2f08a

Tabel Presensi							
id_presensi	tanggal	status	nip	kodeMK	id_kelas	sesi	kehadiran
4	2017-11-27	Dosen	D04	TI-103	1	4	1
5	2017-11-01	Instruktur	I01	TI-105	2	1	1
6	2017-11-01	Instruktur	I01	TI-105	2	2	1
9	2017-11-08	Instruktur	I01	TI-105	2	3	1

Tabel Penggajian										
id_penggajian	status	nip	kodeMK	id_kelas	totalSesi	totalGaji	bulan	tahun	tanggalditerima	statusPengiriman
2	Instruktur	I01	TI-105	2	9	540000	11	2017	2017-11-30	1
4	Dosen	D02	TI-101	1	1	100000	11	2017	2017-10-31	1

Setelah dilakukan *insert* data pada tabel status staff, tidak adanya perubahan yang terjadi pada tabel staff, tabel presensi dan tabel penggajian. Maka tidak ada terjadi anomali pada pemasukan data

## 2. Update anomali

*Update* anomali dilakukan pada tabel status staff karena tabel status staff memiliki relasi dengan tabel staff, tabel presensi dan tabel penggajian. Pengujian *update* anomali dapat dilihat pada tabel 4.36.

**Tabel 4. 36** Pengujian *Update* Tabel Status Staff

Tabel Status Staff							
<b>status</b>	<b>nominal</b>						
Asisten	30000						
Doseen	100000						
Instruktur	60000						
Tabel Staff							
nip	id_user	nama_staff	status	alamat_staff	nohp_staff	email_staff	password_staff
A01	2	Ohno Satoshi	Asisten	Tokyo	934567891	satoshihno@yahoo.jp	7c978c1228b60cc41787020695f8cb77
A02	2	Ninomiya Kazunari	Asisten	Tokyo	86523456890	ninomiyakazunari@yahoo.jp	d5433b40b38af99be049364c502c88dd
D02	2	Slamet Riyadi	Doseen	Yogyakarta	87654323678	slametriyadi@gmail.com	92a968ca99da34abeeb3f2042f973cab
I01	2	Sakurai Sho	Instruktur	Tokyo, Japan	975367112	sakuraisho@yahoo.jp	6ec28e0d3a75899f36a7b7c56bac3eec
I02	2	Matshumoto Jun	Instruktur	Japan	23456789098	matshumotojun@yahoo.jp	ce5ec93770844b25ef0223b169a2f08a

Tabel Presensi							
id_presensi	tanggal	status	nip	kodeMK	id_kelas	sesi	kehadiran
3	2017-12-18	Doseen	D02	TI-101	1	1	1
4	2017-12-26	Doseen	D02	TI-101	1	1	1
5	2017-11-01	Instruktur	I01	TI-105	2	1	1
6	2017-11-01	Instruktur	I01	TI-105	2	2	1
7	2017-12-08	Instruktur	I01	TI-105	2	3	1
8	2017-12-04	Asisten	A01	TI-103	4	1	1
9	2017-12-11	Asisten	A01	TI-103	4	2	1

Tabel Penggajian										
id_penggajian	status	nip	kodeMK	id_kelas	totalSesi	totalGaji	bulan	tahun	tanggalditerima	statusPengiriman
1	Doseen	D02	TI-101	1	2	200000	12	2017	2017-12-15	1
2	Asisten	A01	TI-103	4	2	60000	12	2017	2017-12-15	1
3	Instruktur	I01	TI-105	2	3	180000	12	2017	2017-12-08	1

Setelah dilakukan *update* data pada tabel status staff, tabel yang memiliki relasi dan memiliki data yang sama akan berubah mengikuti data yang di *update* pada tabel staff, tabel presensi dan tabel penggajian, menandakan bahwa tidak terjadi anomali saat data di *update*.

### 3. *Delete* anomali

*Delete* anomali dilakukan pada tabel status staff karena tabel status staff memiliki relasi dengan tabel staff, tabel presensi dan tabel penggajian. Pengujian *delete* anomali dapat dilihat pada tabel 4.37.

**Tabel 4. 37** Pengujian *Delete* Tabel Status Staff

Tabel Status Staff	
status	nominal
Asisten	30000
Instruktur	60000

Tabel Staff							
nip	id_user	nama_staff	status	alamat_staff	nohp_staff	email_staff	password_staff
A01	2	Ohno Satoshi	Asisten	Tokyo	934567891	satoshiohno@yahoo.jp	satoshiiohno
A04	2	Ninomiya Kazunari	Asisten	Tokyo	86523456890	ninomiyakazunari@yahoo.jp	kazunarininomiya
I01	2	Sakurai Sho	Instruktur	Tokyo, Japan	975367112	sakuraisho@yahoo.jp	shosakurairi

Tabel Presensi							
id_presensi	tanggal	status	nip	kodeMK	id_kelas	sesi	kehadiran
5	2017-11-01	Instruktur	I01	TI-105	2	1	1
6	2017-11-01	Instruktur	I01	TI-105	2	2	1
9	2017-11-08	Instruktur	I01	TI-105	2	3	1

Tabel Penggajian										
id_penggajian	status	nip	kodeMK	id_kelas	totalSesi	totalGaji	bulan	tahun	tanggalditerima	statusPengiriman
2	Instruktur	I01	TI-105	2	9	540000	11	2017	2017-11-30	1

Setelah dilakukan *delete* data pada tabel status staff, data dari tabel status staff dapat terhapus. Data yang terhapus pada tabel staff, tabel presensi dan tabel penggajian hanya data yang memiliki relasi dengan data yang dihapus pada tabel status staff, maka tidak terjadi anomali saat data dihapus.

#### e. Pengujian Anomali pada Tabel kelas

##### 1. *Insert* Anomali

*Insert* anomali dilakukan pada tabel kelas karena tabel kelas memiliki relasi dengan tabel presensi dan tabel penggajian. Pengujian *insert* anomali dapat dilihat pada tabel 4.38.

**Tabel 4. 38** Pengujian *Insert* Tabel Kelas

Tabel Kelas	
id_kelas	nama_kelas
1	A
2	B
3	C
4	D

Tabel Presensi							
id_presensi	tanggal	status	nip	kodeMK	id_kelas	sesi	kehadiran
1	2017-12-04	Dosen	D01	TI-102	3	1	1
2	2017-12-11	Dosen	D01	TI-102	3	2	1
3	2017-12-18	Dosen	D01	TI-102	3	3	1
4	2017-12-25	Dosen	D01	TI-102	3	4	1
5	2017-11-01	Instruktur	I01	TI-105	2	1	1
6	2017-11-01	Instruktur	I01	TI-105	2	2	1

Tabel Penggajian										
id_penggajian	status	nip	kodeMK	id_kelas	totalSesi	totalGaji	bulan	tahun	tanggalditerima	statusPengiriman
2	Instruktur	I01	TI-105	2	9	540000	11	2017	2017-11-30	1

Setelah dilakukan *insert* data pada tabel kelas, tidak adanya perubahan yang terjadi pada tabel presensi dan tabel penggajian. Maka tidak ada terjadi anomali pada pemasukan data.

## 2. Update Anomali

*Update* anomali dilakukan pada tabel kelas karena tabel kelas memiliki relasi dengan tabel presensi dan tabel penggajian. Pengujian *update* anomali dapat dilihat pada tabel 4.39..

**Tabel 4. 39** Pengujian *Update* Tabel Kelas

Tabel Kelas	
id_kelas	nama_kelas
1	A
3	C
4	D
5	B

Tabel Presensi							
id_presensi	tanggal	status	nip	kodeMK	id_kelas	sesi	kehadiran
1	2017-12-04	Dosen	D01	TI-102	3	1	1
2	2017-12-11	Dosen	D01	TI-102	3	2	1
3	2017-12-18	Dosen	D01	TI-102	3	3	1
4	2017-12-26	Dosen	D01	TI-102	3	4	1
5	2017-11-01	Instruktur	I01	TI-105	5	1	1
6	2017-11-01	Instruktur	I01	TI-105	5	2	1
9	2017-12-08	Instruktur	I01	TI-105	5	3	1

Tabel Penggajian										
id_penggajian	status	nip	kodeMK	id_kelas	totalSesi	totalGaji	bulan	tahun	tanggalditerima	statusPengiriman
1	Dosen	D01	TI-102	3	4	400000	12	2017	2017-12-09	1
2	Instruktur	I01	TI-105	5	9	540000	11	2017	2017-11-30	1

Setelah dilakukan *update* data pada tabel kelas, tabel yang memiliki relasi dan memiliki data yang sama akan berubah mengikuti data yang di *update* pada tabel presensi dan tabel penggajian, menandakan bahwa tidak terjadi anomali saat data di *update*.

### 3. Delete Anomali

*Delete* anomali dilakukan pada tabel kelas karena tabel kelas memiliki relasi dengan tabel presensi dan tabel penggajian. Pengujian *delete* anomali dapat dilihat pada tabel 4.40.

**Tabel 4. 40** Pengujian *Delete* Tabel Kelas

Tabel Kelas										
id_kelas	nama_kelas									
1	A									
3	C									
4	D									

Tabel Presensi										
id_presensi	tanggal	status	nip	kodeMK	id_kelas	sesi	kehadiran			
1	2017-12-04	Dosen	D01	TI-102	3	1	1			
2	2017-12-11	Dosen	D01	TI-102	3	2	1			
3	2017-12-18	Dosen	D01	TI-102	3	3	1			
4	2017-12-26	Dosen	D01	TI-102	3	4	1			

Tabel Penggajian										
id_penggajian	status	nip	kodeMK	id_kelas	totalSesi	totalGaji	bulan	tahun	tanggalditerima	statusPengiriman
1	Dosen	D01	TI-102	3	4	400000	12	2017	2017-12-09	1



Setelah dilakukan *delete* data pada tabel kelas, data dari tabel kelas dapat terhapus. Data yang terhapus pada tabel presensi dan tabel penggajian hanya data yang memiliki relasi dengan data yang dihapus pada tabel status staff, maka tidak terjadi anomali saat data dihapus.

### 4.5.3 Pengujian View

*View* adalah salah satu perintah *query* yang disimpan pada *database* dengan suatu nama tertentu, sehingga dapat digunakan setiap saat untuk melihat data tanpa menuliskan ulang *query* tersebut. Keuntungan penggunaan *view* adalah dapat membatasi akses data, menyediakan data yang independen, memudahkan *query* yang kompleks dan bisa juga dipakai untuk validasi suatu data.

#### 1. *View Check Option* tabel presensi

*View Check Option* di implementasikan pada tabel presensi kolom kehadiran, pada kolom ini hanya diisi dengan angka biner 1 dan 0. Berikut adalah *query* untuk *View Check* pada tabel presensi dapat dilihat pada Gambar 4.46. Dan pada gambar 4.46 merupakan *query* untuk *insert* data pada table penggajian

```
CREATE VIEW vpresensi AS
  SELECT * FROM presensi WHERE kehadiran IN ('1','0')
  WITH CHECK OPTION;
```

**Gambar 4. 46** *Query Create View Check* tabel presensi

```
INSERT INTO vpresensi VALUES (25, '2017-11-15', 'Dosen', 'D02', 'TI-103', 1, 1, '3')
```

**Gambar 4. 47** *Query Insert* vpresensi tabel presensi

```

Error
SQL query:

INSERT INTO vpresensi VALUES (25, '2017-11-15', 'Dosen', 'D02', 'TI-103', 1, 1, '3')

```

**Gambar 4. 48** Validasi *View Check* vpresensi

Pada gambar 4.48, saat pengujian *insert* data yang tidak sesuai dengan pembatasan data pada kolom kehadiran yaitu '1' dan '0', maka *query* tidak bisa masuk ke tabel presensi dan mengalami *error Check Option*. Hal ini berarti *View Check Option* telah berhasil di implementasikan pada tabel presensi.

## 2. *View Check Option* tabel penggajian

*View Check Option* di implementasikan pada tabel presensi kolom statusPengiriman, pada kolom ini hanya diisi dengan angka biner 1 dan 0. Berikut adalah *query* untuk *View Check* pada tabel penggajian dapat dilihat pada Gambar 4.49. Dan pada gambar 4.50 merupakan *query* untuk *insert* data pada table penggajian.

```

CREATE VIEW vpenggajian AS
SELECT * FROM penggajian WHERE statusPengiriman IN ('1','0')
WITH CHECK OPTION;

```

**Gambar 4. 49** *Query Create View Check* tabel penggajian

```

INSERT INTO vpenggajian VALUES (4, 'Dosen', 'D02', 'TI-101', 1,
1, '100000', '11', '2017', '2017-10-31', '3')

```

**Gambar 4. 50** *Query Insert* vpenggajian tabel penggajian

```
Error
SQL query:
INSERT INTO vpenggajian VALUES (4, 'Dosen', 'D02', 'TI-101', 1, 1, '100000', '11', '2017', '2017-10-31', '3')
MySQL said: ⓘ
#1369 - CHECK OPTION failed 'penggajianstaff.vpenggajian'
```

**Gambar 4. 51** Validasi *View Check* vpenggajian

Pada gambar 4.51, saat pengujian *insert* data yang tidak sesuai dengan pembatasan data pada kolom statusPengiriman yaitu '1' dan '0', maka *query* tidak bisa masuk ke tabel penggajian dan mengalami *error Check Option*. Hal ini berarti *View Check Option* telah berhasil di implementasikan pada tabel penggajian.