

BAB IV

HASIL DAN PEMBAHASAN

4.1 Perancangan *Database* Fisikal

Perancangan *Database* Fisikal merupakan suatu proses untuk menghasilkan gambaran dari implementasi basis data pada tempat penyimpanan, kemudian menjelaskan dasar dari relasi, organisasi *file* dan indeks yang digunakan untuk efisiensi data dan menghubungkan beberapa *integrity constraints* dan tindakan keamanan.

4.1.1 Tabel

Setelah melakukan proses normalisasi maka terbentuk beberapa tabel, diantaranya adalah tabel *Users*, tabel Administrator, tabel Mahasiswa, tabel Dosen, tabel Ide Skripsi dan tabel Pengumuman Ide Skripsi.

1. Tabel *Users*

Tabel 4. 1 *Users*

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	ID_USERS	int(11)			No	None		AUTO_INCREMENT
2	ROLE	varchar(10)	latin1_swedish_ci		No	None		

Penjelasan dari tabel 4.1 *Users* antara lain:

- 1) ID_USERS adalah kolom yang menunjukkan id *users* dari masing-masing *user* sesuai dengan peran dari ketiga *user* diantaranya administrator, mahasiswa dan dosen, bersifat *auto increment* dan *primary key* dengan tipe data *Int* karena pada

kolom `ID_USERS` berisi angka (*numeric*). Kolom `ID_USERS` memiliki panjang data 11 karena merupakan maksimal panjang data pada tipe data *Int*.

- 2) *ROLE* adalah kolom yang menunjukkan nama *role* dari masing-masing *users* sesuai dengan peran dari ketiga *user* diantaranya admin, mahasiswa dan dosen, memiliki tipe data *Varchar* karena pada kolom *ROLE* berisi karakter (*string*) dengan panjang bervariasi tergantung data dengan panjang. Data 10 karena karakter terpanjang dari nama role pada tabel Users adalah mahasiswa (10 karakter).

2. Tabel Administrator

Tabel 4. 2 Administrator

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<code>ID_ADM</code> 	int(11)			No	None		AUTO_INCREMENT
2	<code>ID_USERS</code> 	int(11)			No	None		
3	<code>NAMA_ADM</code>	varchar(30)	latin1_swedish_ci		No	None		
4	<code>ALAMAT_ADM</code>	text	latin1_swedish_ci		No	None		
5	<code>NOHP_ADM</code>	varchar(12)	latin1_swedish_ci		No	None		
6	<code>EMAIL_ADM</code>	varchar(30)	latin1_swedish_ci		No	None		
7	<code>USERNAME_ADM</code>	varchar(15)	latin1_swedish_ci		No	None		
8	<code>PASSWORD_ADM</code>	varchar(50)	latin1_swedish_ci		No	None		
9	<code>STATUS_ADM</code>	enum('0', '1')	latin1_swedish_ci		Yes	0		

Penjelasan dari tabel 4.2 Administrator antara lain:

- 1) `ID_ADM` adalah kolom yang menunjukkan id admin dari masing-masing administrator, bersifat *auto increment* dan *primary key* dengan tipe data *Int* karena pada kolom `ID_ADM` berisi angka (*numeric*). Kolom `ID_ADM` memiliki panjang data 11 karena merupakan maksimal panjang data pada tipe data *Int*.

- 2) ID_USERS adalah kolom yang menunjukkan ID_USER dari tabel *User* dan merupakan *foreign key*.
- 3) NAMA_ADM adalah kolom yang menunjukkan nama dari masing-masing administrator, dengan tipe data *Varchar* karena pada kolom NAMA_ADM berisi karakter (*string*) dengan panjang data yang bervariasi tergantung datanya. Kolom NAMA_ADM memiliki panjang data 30.
- 4) ALAMAT_ADM adalah kolom yang menunjukkan alamat dari masing-masing administrator, dengan tipe data *Text* karena ALAMAT_ADM memiliki panjang data yang *relative* lebih panjang dari data di kolom yang lain.
- 5) NOHP_ADM adalah kolom yang menunjukkan *nomor handphone* dari masing-masing administrator, dengan tipe data *Varchar* karena NOHP_ADM berisi karakter (*string*) dengan panjang data yang bervariasi tergantung datanya. Kolom NOHP_ADM memiliki panjang data 12 karena panjang maksimal dari *nomor handphone* yang ada saat ini adalah 12 karakter.
- 6) EMAIL_ADM adalah kolom yang menunjukkan *email* dari masing-masing administrator, dengan tipe data *Varchar* karena EMAIL_ADM berisi karakter (*string*) dengan panjang data yang bervariasi tergantung datanya. Kolom EMAIL_ADM memiliki panjang data 30 karena panjang data maksimal dari email adalah 30 karakter.
- 7) USERNAME_ADM adalah kolom yang menunjukkan *username* dari masing-masing administrator, dengan tipe data *Varchar* karena pada kolom USERNAME_ADM berisi karakter (*string*) dengan panjang data yang

bervariasi tergantung datanya. Kolom USERNAME_ADM memiliki panjang data 15 berdasarkan panjang rata-rata *username*.

- 8) PASSWORD_ADM adalah kolom yang menunjukkan password dari masing-masing administrator, dengan tipe data *Varchar* karena pada kolom PASSWORD_ADM berisi karakter (*string*) dengan panjang data yang bervariasi tergantung datanya. Kolom PASSWORD_ADM memiliki panjang data 50 karena password dari adminstartor menggunakan enkripsi MD5 agar password lebih aman karena yang mengetahui hanya pemilik dari akun saja.
- 9) STATUS_ADM adalah kolom yang menunjukkan status aktif atau tidaknya akun dari administrator, dengan tipe data *Enum* karena pada kolom STATUS_ADM berisi karakter atau nilai data yang sudah ditentukan sebelumnya yaitu '0' dan '1', nilai '0' untuk akun administrator yang belum aktif dan nilai '1' untuk akun administrator yang sudah aktif.

3. Tabel Mahasiswa

Tabel 4. 3 Mahasiswa

#	Name	Type	Collation	Attributes	Null	Default
1	NIM_MHS 🔑	char(11)	latin1_swedish_ci		No	None
2	ID_USERS 🔑	int(11)			No	None
3	NAMA_MHS	varchar(30)	latin1_swedish_ci		No	None
4	FAKULTAS_MHS	enum('Teknik')	latin1_swedish_ci		No	None
5	PRODI_MHS	enum('Teknik Informatika')	latin1_swedish_ci		No	None
6	SKS_MHS	char(3)	latin1_swedish_ci		No	None
7	IPK_MHS	float			No	None
8	ALAMAT_MHS	text	latin1_swedish_ci		No	None
9	NOHP_MHS	varchar(12)	latin1_swedish_ci		No	None
10	EMAIL_MHS	varchar(30)	latin1_swedish_ci		No	None
11	USERNAME_MHS	varchar(15)	latin1_swedish_ci		No	None
12	PASSWORD_MHS	varchar(50)	latin1_swedish_ci		No	None
13	STATUS_MHS	enum('0', '1')	latin1_swedish_ci		Yes	0

Penjelasan dari tabel 4.3 Mahasiswa antara lain:

- 1) NIM_MHS adalah kolom yang menunjukkan nomor induk mahasiswa dari masing-masing mahasiswa, dengan tipe data *Char* karena pada kolom NIM_MHS berisi data dengan panjang yang tetap. Kolom NIM_MHS memiliki panjang data 11 karena merupakan panjang data dari nomor induk mahasiswa yang ada di Universitas Muhammadiyah Yogyakarta.
- 2) ID_USERS adalah kolom yang menunjukkan ID_USERS dari tabel *Users* dan merupakan *foreign key*.

- 3) NAMA_MHS adalah kolom yang menunjukkan nama dari masing-masing mahasiswa, dengan tipe data *Varchar* karena pada kolom NAMA_MHS berisi karakter (*string*) dengan panjang data yang bervariasi tergantung datanya. Kolom NAMA_MHS memiliki panjang data 30 karakter.
- 4) FAKULTAS_MHS adalah kolom yang menunjukkan nama fakultas dari masing-masing mahasiswa, dengan tipe data *Enum* karena pada kolom FAKULTAS_MHS berisi karakter atau nilai data yang sudah ditentukan sebelumnya yaitu 'Teknik', karena saat ini sistem pengambilan ide skripsi hanya untuk Fakultas Teknik. *Enum* dapat disesuaikan sesuai kebutuhan jika ada tambahan Fakultas.
- 5) PRODI_MHS adalah kolom yang menunjukkan nama prodi dari masing-masing mahasiswa, dengan tipe data *Enum* karena pada kolom PRODI_MHS berisi karakter atau nilai data yang sudah ditentukan sebelumnya yaitu 'Teknik Informatika', karena saat ini sistem pengambilan ide skripsi hanya untuk Prodi Teknik Informatika. *Enum* dapat disesuaikan sesuai kebutuhan jika ada tambahan Prodi.
- 6) SKS_MHS adalah kolom yang menunjukkan sks dari masing-masing mahasiswa, dengan tipe data *Char* karena pada kolom SKS_MHS berisi data dengan panjang yang sama. Kolom SKS_MHS memiliki panjang data 3 karena maksimal sks 145 (3 karakter).
- 7) IPK_MHS adalah kolom yang menunjukkan ipk dari masing-masing mahasiswa, dengan tipe data *Float* karena atribut IPK_MHS berisi bilangan pecahan positif presisi tunggal.

- 8) ALAMAT_MHS adalah kolom yang menunjukkan alamat dari masing-masing mahasiswa, dengan tipe data *Text* karena ALAMAT_MHS memiliki panjang data yang *relative* lebih panjang dari data di kolom yang lain.
- 9) NOHP_MHS adalah kolom yang menunjukkan *nomor handphone* dari masing-masing mahasiswa, dengan tipe data *Varchar* karena NOHP_MHS berisi karakter (*string*) dengan panjang data yang bervariasi tergantung datanya. Kolom NOHP_MHS memiliki panjang data 12 karena panjang maksimal dari *nomor handphone* yang ada saat ini adalah 12 karakter.
- 10) EMAIL_MHS adalah kolom yang menunjukkan *email* dari masing-masing mahasiswa, dengan tipe data *Varchar* karena EMAIL_MHS berisi karakter (*string*) dengan panjang data yang bervariasi tergantung datanya. Kolom EMAIL_MHS memiliki panjang data 30 karena panjang data maksimal dari email adalah 30 karakter.
- 11) USERNAME_MHS adalah kolom yang menunjukkan *username* dari masing-masing mahasiswa, dengan tipe data *Varchar* karena pada kolom USERNAME_MHS berisi karakter (*string*) dengan panjang data yang bervariasi tergantung datanya. Kolom USERNAME_MHS memiliki panjang data 15 berdasarkan panjang rata-rata *username*.
- 12) PASSWORD_MHS adalah kolom yang menunjukkan password dari masing-masing mahasiswa, dengan tipe data *Varchar* karena pada kolom PASSWORD_MHS berisi karakter (*string*) dengan panjang data yang bervariasi tergantung datanya. Kolom PASSWORD_MHS memiliki panjang

data 50 karena password dari mahasiswa menggunakan enkripsi MD5 agar password lebih aman karena yang mengetahui hanya pemilik dari akun saja.

13) STATUS_MHS adalah kolom yang menunjukkan status aktif atau tidaknya akun dari mahasiswa, dengan tipe data *Enum* karena pada kolom STATUS_MHS berisi karakter atau nilai data yang sudah ditentukan sebelumnya yaitu '0' dan '1', nilai '0' untuk akun mahasiswa yang belum aktif dan nilai '1' untuk akun mahasiswa yang sudah aktif.

4. Tabel Dosen

Tabel 4. 4 Dosen

#	Name	Type	Collation	Attributes	Null	Default
1	NIDN_DSN	char(10)	latin1_swedish_ci		No	None
2	ID_USERS	int(11)			No	None
3	NAMA_DSN	varchar(30)	latin1_swedish_ci		No	None
4	FAKULTAS_DSN	enum('Teknik')	latin1_swedish_ci		No	None
5	PRODI_DSN	enum('Teknik Informatika')	latin1_swedish_ci		No	None
6	ALAMAT_DSN	text	latin1_swedish_ci		No	None
7	NOHP_DSN	varchar(12)	latin1_swedish_ci		No	None
8	PENDIDIKAN_DSN	enum('S1', 'S2', 'S3')	latin1_swedish_ci		No	None
9	KEAHLIAN_DSN	varchar(100)	latin1_swedish_ci		No	None
10	EMAIL_DSN	varchar(30)	latin1_swedish_ci		No	None
11	USERNAME_DSN	varchar(15)	latin1_swedish_ci		No	None
12	PASSWORD_DSN	varchar(50)	latin1_swedish_ci		No	None
13	STATUS_DSN	enum('0', '1')	latin1_swedish_ci		Yes	0

Penjelasan dari tabel 4.3 Mahasiswa antara lain:

1) NIDN_MHS adalah kolom yang menunjukkan nomor induk dosen nasional dari masing-masing dosen, dengan tipe data *Char* karena pada kolom NIDN_MHS

berisi data dengan panjang yang tetap. Kolom NIDN_MHS memiliki panjang data 10 karena merupakan panjang data dari nomor induk dosen nasional.

- 2) ID_USERS adalah kolom yang menunjukkan ID_USERS dari tabel *Users* dan merupakan *foreign key*.
- 3) NAMA_DSN adalah kolom yang menunjukkan nama dari masing-masing dosen, dengan tipe data *Varchar* karena pada kolom NAMA_DSN berisi karakter (*string*) dengan panjang data yang bervariasi tergantung datanya. Kolom NAMA_DSN memiliki panjang data 30 karakter.
- 4) FAKULTAS_DSN adalah kolom yang menunjukkan nama fakultas dari masing-masing dosen, dengan tipe data *Enum* karena pada kolom FAKULTAS_DSN berisi karakter atau nilai data yang sudah ditentukan sebelumnya yaitu ‘Teknik’, karena saat ini sistem pengambilan ide skripsi hanya untuk Fakultas Teknik. *Enum* dapat disesuaikan sesuai kebutuhan jika ada tambahan Fakultas.
- 5) PRODI_DSN adalah kolom yang menunjukkan nama prodi dari masing-masing dosen, dengan tipe data *Enum* karena pada kolom PRODI_DSN berisi karakter atau nilai data yang sudah ditentukan sebelumnya yaitu ‘Teknik Informatika’, karena saat ini sistem pengambilan ide skripsi hanya untuk Prodi Teknik Informatika. *Enum* dapat disesuaikan sesuai kebutuhan jika ada tambahan Prodi.
- 6) ALAMAT_DSN adalah kolom yang menunjukkan alamat dari masing-masing dosen, dengan tipe data *Text* karena ALAMAT_DSN memiliki panjang data yang *relative* lebih panjang dari data di kolom yang lain.

- 7) NOHP_DSN adalah kolom yang menunjukkan *nomor handphone* dari masing-masing dosen, dengan tipe data *Varchar* karena NOHP_DSN berisi karakter (*string*) dengan panjang data yang bervariasi tergantung datanya. Kolom NOHP_DSN memiliki panjang data 12 karena panjang maksimal dari *nomor handphone* yang ada saat ini adalah 12 karakter.
- 8) PENDIDIKAN_DSN adalah kolom yang menunjukkan pendidikan terakhir dari masing-masing dosen, dengan tipe data *Enum* karena pada kolom PENDIDIKAN_DSN berisi karakter atau nilai data yang sudah ditentukan sebelumnya yaitu 'S1', 'S2', 'S3'.
- 9) KEAHLIAN_DSN adalah kolom yang menunjukkan keahlian yang dimiliki oleh masing-masing dosen dengan tipe data *Varchar* karena KEAHLIAN_DSN berisi karakter (*string*) dengan panjang data yang bervariasi tergantung datanya. Kolom KEAHLIAN_DSN memiliki panjang data 100 karakter karena satu dosen dapat mempunyai keahlian lebih dari satu keahlian, contoh: dosen A memiliki keahlian *programming, database* dan *networking*.
- 10) EMAIL_DSN adalah kolom yang menunjukkan *email* dari masing-masing dosen, dengan tipe data *Varchar* karena EMAIL_DSN berisi karakter (*string*) dengan panjang data yang bervariasi tergantung datanya. Kolom EMAIL_DSN memiliki panjang data 30 karena panjang data maksimal dari email adalah 30 karakter.
- 11) USERNAME_DSN adalah kolom yang menunjukkan *username* dari masing-masing dosen, dengan tipe data *Varchar* karena pada kolom USERNAME_DSN berisi karakter (*string*) dengan panjang data yang

bervariasi tergantung datanya. Kolom USERNAME_DSN memiliki panjang data 15 berdasarkan panjang rata-rata *username*.

12) PASSWORD_DSN adalah kolom yang menunjukkan password dari masing-masing dosen, dengan tipe data *Varchar* karena pada kolom PASSWORD_DSN berisi karakter (*string*) dengan panjang data yang bervariasi tergantung datanya. Kolom PASSWORD_DSN memiliki panjang data 50 karena password dari dosen menggunakan enkripsi MD5 agar password lebih aman karena yang mengetahui hanya pemilik dari akun saja.

13) STATUS_DSN adalah kolom yang menunjukkan status aktif atau tidaknya akun dari dosen, dengan tipe data *Enum* karena pada kolom STATUS_DSN berisi karakter atau nilai data yang sudah ditentukan sebelumnya yaitu '0' dan '1', nilai '0' untuk akun dosen yang belum aktif dan nilai '1' untuk akun dosen yang sudah aktif.

5. Tabel Ide Skripsi

Tabel 4.5 Ide Skripsi

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	ID_IDE 	int(11)			No	None		AUTO_INCREMENT
2	NIDN_DSN 	char(10)	latin1_swedish_ci		No	None		
3	IDE_SKRIPSI	varchar(60)	latin1_swedish_ci		No	None		
4	DESKRIPSI	text	latin1_swedish_ci		No	None		
5	KUOTA_AMBIL	int(11)			No	None		
6	MATKUL	varchar(30)	latin1_swedish_ci		No	None		

Penjelasan dari gambar 4.5 pada tabel Ide Skripsi antara lain:

1) ID_IDE adalah kolom yang menunjukkan id ide skripsi dari masing-masing ide skripsi, bersifat *auto increment* dan *primary key* dengan tipe data *Int* karena

pada kolom ID_IDE berisi angka (numeric) dengan panjang data 11 karena merupakan maksimal panjang data pada tipe data *Int*.

- 2) NIDN_DSN adalah kolom yang menunjukkan NIDN_DSN dari tabel Dosen dan merupakan *foreign key*.
- 3) IDE_SKRIPSI adalah kolom yang menunjukkan nama ide skripsi dari masing-masing ide skripsi, dengan tipe data *Varchar* karena pada kolom IDE_SKRIPSI berisi karakter (*string*) dengan panjang data yang bervariasi tergantung datanya. Kolom IDE_SKRIPSI memiliki panjang data 60 berdasarkan panjang rata-rata dari ide skripsi.
- 4) DESKRIPSI adalah kolom yang menunjukkan deskripsi ide skripsi dari masing-masing ide skripsi, dengan tipe data *Text* karena DESKRIPSI memiliki panjang data yang *relative* lebih panjang dari data di kolom yang lain.
- 5) KUOTA adalah kolom yang menunjukkan kuota ide skripsi dari masing-masing ide skripsi, dengan tipe data *Int* karena pada kolom KUOTA berisi angka (numeric) dengan panjang data 11 karena merupakan maksimal panjang data pada tipe data *Int*.
- 6) MATKUL adalah kolom yang menunjukkan mata kuliah yang diampu oleh dosen yang memberikan ide skripsi, dengan tipe data *Varchar* karena pada kolom MATKUL berisi karakter (*string*) dengan panjang data yang bervariasi tergantung datanya. Kolom MATKUL memiliki panjang data 30 berdasarkan panjang rata-rata dari beberapa mata kuliah.

6. Tabel Pengumuman Ide Skripsi

Tabel 4. 6 Pengumuman Ide Skripsi

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	ID_PIS 	int(11)			No	None		AUTO_INCREMENT
2	ID_IDE 	int(11)			No	None		
3	NIM_MHS 	char(11)	latin1_swedish_ci		No	None		
4	STATUS_PIS	enum('Y', 'N')	latin1_swedish_ci		Yes	NULL		
5	NILAI_MATKUL	varchar(30)	latin1_swedish_ci		No	None		

Penjelasan dari gambar 4.6 pada tabel Mahasiswa antara lain:

- 1) ID_PIS adalah kolom yang menunjukkan id pengambilan ide skripsi dari masing-masing administrator, bersifat *auto increment* dan *primary key* dengan tipe data *Int* karena pada kolom ID_ADM berisi angka(numeric) dengan panjang data 11 karena merupakan maksimal panjang data pada tipe data *Int*.
- 2) ID_IDE adalah kolom yang menunjukkan ID_IDE dari tabel Ide Skripsi dan merupakan *foreign key*.
- 3) NIM_MHS adalah kolom yang menunjukkan NIM_MHS dari tabel Mahasiswa dan merupakan *foreign key*.
- 4) STATUS adalah kolom yang menunjukkan status diterima atau belum diterimanya masing-masing mahasiswa untuk mengerjakan ide skripsi oleh dosen, dapat bersifat *Null* karena pada saat mahasiswa mengambil ide skripsi pada tabel pengumuman ide skripsi, status dari mahasiswa masih *Null* dan ketika dosen telah menyeleksi mahasiswa yang akan mengerjakan ide skripsi maka status dari mahasiswa tersebut dari *Null* akan berganti menjadi 'Y' yang berarti diterima dan 'N' yang berarti belum diterima. Tipe data dari kolom STATUS adalah *Enum* karena karakter atau nilai data sudah ditentukan sebelumnya yaitu 'Y' dan 'N'.

7) NILAI_MATKUL adalah kolom yang menunjukkan nilai mata kuliah mahasiswa sebagai salah satu alternative dosen pada saat memilih mahasiswa yang berhak mengerjakan ide skripsi yang telah diberikan oleh dosen tersebut, dengan tipe data *Varchar* karena pada kolom NILAI_MATKUL berisi karakter (*string*) dengan panjang data yang bervariasi tergantung datnya. Kolom NILAI_MATKUL memiliki panjang data 30 berdasarkan panjang rata-rata dari beberapa mata kuliah.

4.1.2 Foreign Key Constraints

Foreign Key Constrains adalah *constrains* 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 Constrains*, seperti tabel Administrator, tabel Mahasiswa, tabel Dosen, tabel Ide Skripsi dan tabel Pengumuman Ide Skripsi.

1. Foreign Key Constraints Tabel Administrator

Constraints tabel Administrator dapat dilihat pada gambar 4.1



Gambar 4. 1 Foreign Key Constraints Tabel Administrator

Berikut adalah penjelasan dari gambar 4.1:

Tabel Administrator mempunyai *constraints ON DELETE RESTRICT* dan terhubung dengan tabel *Users* yang artinya kolom pada tabel *Users* tidak dapat dihapus sebelum menghapus kolom tabel Administrator terlebih dahulu. Tabel Administrator mempunyai *constraints ON UPDATE CASCADE* dan terhubung

dengan tabel *Users* yang artinya jika kolom tabel *Users* diubah maka kolom tabel Administrator ikut berubah datanya.

2. Foreign Key Constraints Tabel Dosen

Constraints tabel Dosen dapat dilihat pada gambar 4.2



Gambar 4. 2 Foreign Key Constraints Tabel Dosen

Berikut adalah penjelasan dari gambar 4.2:

Tabel Dosen mempunyai *constraints ON DELETE RESTRICT* dan terhubung dengan tabel *Users* yang artinya kolom pada tabel *Users* tidak dapat dihapus sebelum menghapus kolom tabel Dosen terlebih dahulu. Tabel Dosen mempunyai *constraints ON UPDATE CASCADE* dan terhubung dengan tabel *Users* yang artinya jika kolom tabel *Users* diubah maka kolom tabel Dosen ikut berubah datanya.

3. Foreign Key Constraints Tabel Mahasiswa

Constraints tabel *Users* dapat dilihat pada gambar 4.3



Gambar 4. 3 Foreign Key Constraints Tabel Mahasiswa

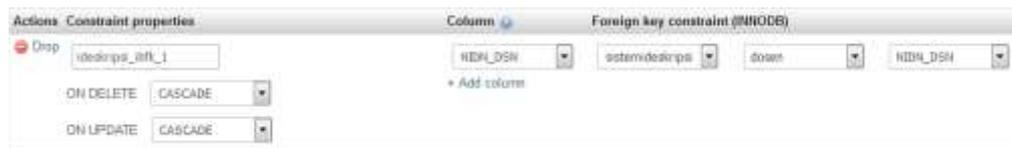
Berikut adalah penjelasan dari gambar 4.3:

Tabel Mahasiswa mempunyai *constraints ON DELETE RESTRICT* dan terhubung dengan tabel *Users* yang artinya kolom pada tabel *Users* tidak dapat

dihapus sebelum menghapus kolom tabel Mahasiswa terlebih dahulu. Tabel Mahasiswa mempunyai *constraints ON UPDATE CASCADE* dan terhubung dengan tabel *Users* yang artinya jika kolom tabel *Users* diubah maka kolom tabel Mahasiswa ikut berubah datanya.

4. Foreign Key Constraints Tabel Ide Skripsi

Constraints tabel Ide Skripsi dapat dilihat pada gambar 4.4



Gambar 4. 4 Foreign Key Constraint Tabel Ide Skripsi

Berikut adalah penjelasan dari gambar 4.4:

Tabel Ide Skripsi wisata mempunyai *constraints ON DELETE CASCADE* dan terhubung dengan tabel Dosen yang artinya jika kolom tabel Dosen dihapus, maka kolom tabel Ide Skripsi ikut juga terhapus. Tabel Ide Skripsi mempunyai *constraints ON UPDATE CASCADE* dan terhubung dengan tabel Dosen yang artinya jika kolom tabel Dosen diubah, maka kolom tabel Ide Skripsi ikut juga berubah.

5. Foreign Key Constraints Tabel Pengumuman Ide Skripsi

Constraints tabel Pengumuman Ide Skripsi dapat dilihat pada gambar 4.5



Gambar 4. 5 Foreign Key Constraints Table Pengumuman Ide Skripsi

Berikut adalah penjelasan dari gambar 4.5:

Tabel Pengumuman Ide Skripsi mempunyai *constraints ON DELETE CASCADE* dan terhubung dengan tabel Ide Skripsi yang artinya jika kolom tabel Ide Skripsi dihapus, maka kolom tabel Pengumuman Ide Skripsi ikut juga terhapus. Tabel Pengumuman Ide Skripsi mempunyai *constraints ON UPDATE CASCADE* dan terhubung dengan tabel Ide Skripsi yang artinya jika kolom tabel Ide Skripsi diubah, maka kolom tabel Pengumuman Ide Skripsi ikut juga berubah.

Tabel Pengumuman Ide Skripsi mempunyai *constraints ON DELETE CASCADE* dan terhubung dengan tabel Mahasiswa yang artinya jika kolom tabel Mahasiswa dihapus, maka kolom tabel Pengumuman Ide Skripsi ikut juga terhapus. Tabel Pengumuman Ide Skripsi mempunyai *constraints ON UPDATE CASCADE* dan terhubung dengan tabel Mahasiswa yang artinya jika kolom tabel Mahasiswa diubah, maka kolom tabel Pengumuman Ide Skripsi ikut juga berubah.

4.1.3 Store Procedure

Stored procedure adalah salah satu objek *routine* yang tersimpan pada *database* MySQL dan dapat digunakan untuk menggantikan berbagai kumpulan perintah yang sering digunakan. Pada *database* sistem pengambilan ide skripsi terdapat 3 perintah *procedure* yaitu *procedure* Hapus Users Administrator, Hapus Users Dosen dan Hapus Users Mahasiswa.

1. *Procedure* Hapus Administrator

Constraints tabel Administrator dapat dilihat pada gambar 4.6

```

1 DELIMITER |
2
3 CREATE PROCEDURE HapusUsersADM(IN users_adm_param INT(11))
4 BEGIN
5     DELETE FROM admin WHERE ID_USERS = users_adm_param;
6     DELETE FROM users WHERE ID_USERS = users_adm_param;
7 END;
8
9 |
10
11 DELIMITER ;

```

Gambar 4. 6 *Procedure* Hapus Administrator

Penjelasan pada gambar 4.6 sebagai berikut ini:

Untuk menghapus *Users* diperlukan *procedure* hapus *Users* Admin, yang dimana perintah tersebut berfungsi untuk menghapus kolom tabel Admin terlebih dahulu, baru menghapus tabel *Users*.

Perintah untuk menghapus *Users* Administrator dapat dilihat pada gambar 4.7

```
CALL HapusUsersADM(1);
```

Gambar 4. 7 *Call Procedure* UsersADM

Penjelasan pada gambar 4.7 sebagai berikut ini:

Perintah pada gambar 4.6 ditujukan untuk menghapus *Users* Administrator. *CALL* untuk memanggil *procedure* yang telah dibuat pada gambar 4.7 yaitu *HapusUsersADM*, angka 1 pada perintah diatas menunjukkan user id pada kolom Administrator dari tabel *Users*.

2. Procedure Hapus Dosen

Constraints tabel Dosen dapat dilihat pada gambar 4.8

```

1 DELIMITER |
2
3 CREATE PROCEDURE HapusUsersDSN(IN users_dsn_param INT(11))
4 BEGIN
5     DELETE FROM dosen WHERE ID_USERS = users_dsn_param;
6     DELETE FROM users WHERE ID_USERS = users_dsn_param;
7 END;
8
9 |
10
11 DELIMITER ;|

```

Gambar 4. 8 Procedure Hapus Dosen

Penjelasan pada gambar 4.8 sebagai berikut ini:

Untuk menghapus *Users* diperlukan *procedure* hapus Users Dosen, yang dimana perintah tersebut berfungsi untuk menghapus kolom tabel Dosen terlebih dahulu, baru menghapus tabel *Users*.

Perintah untuk menghapus *Users* Dosen dapat dilihat pada gambar 4.9

```
CALL HapusUsersDSN(3);
```

Gambar 4. 9 Call Procedure UsersDSN

Penjelasan pada gambar 4.9 sebagai berikut ini:

Perintah pada gambar 4.8 ditujukan untuk menghapus *Users* Dosen. CALL untuk memanggil *procedure* yang telah dibuat pada gambar 4.9 yaitu HapusUsersDSN, angka 3 pada perintah diatas menunjukkan user id pada kolom Dosen dari tabel *Users*.

3. *Procedure* Hapus Mahasiswa

Constraints tabel Dosen dapat dilihat pada gambar 4.10

```

1 DELIMITER |
2
3 CREATE PROCEDURE HapusUsersMHS(IN users_mhs_param INT(11))
4 BEGIN
5     DELETE FROM mahasiswa WHERE ID_USERS = users_mhs_param;
6     DELETE FROM users WHERE ID_USERS = users_mhs_param;
7 END;
8
9 |
10
11 DELIMITER ;
12

```

Gambar 4. 10 *Procedure* Hapus Mahasiswa

Penjelasan pada gambar 4.10 sebagai berikut ini:

Untuk menghapus *Users* diperlukan *procedure* hapus *Users* Mahasiswa, yang dimana perintah tersebut berfungsi untuk menghapus kolom tabel Mahasiswa terlebih dahulu, baru menghapus tabel *Users*.

Perintah untuk menghapus *Users* Mahasiswa dapat dilihat pada gambar 4.11

```
CALL HapusUsersMHS (2);
```

Gambar 4. 11 *Call Procedure* UsersMHS

Penjelasan pada gambar 4.11 sebagai berikut ini:

Perintah pada gambar 4.11 ditujukan untuk menghapus *Users* Mahasiswa. *CALL* untuk memanggil *procedure* yang telah dibuat pada gambar 4.10 yaitu *HapusUsersMHS*, angka 2 pada perintah diatas menunjukkan user id pada kolom Mahasiswa dari tabel *Users*.

4.1.4 Kebijakan Akses *Users*

Kebijakan Akses *Users* 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 tertentu saja, tabel tertentu, atau bahkan hanya kolom tertentu. Pada Sistem Pengambilan Ide Skripsi terdapat 4 jenis *user* diantaranya adalah:

1. Kebijakan Akses Master

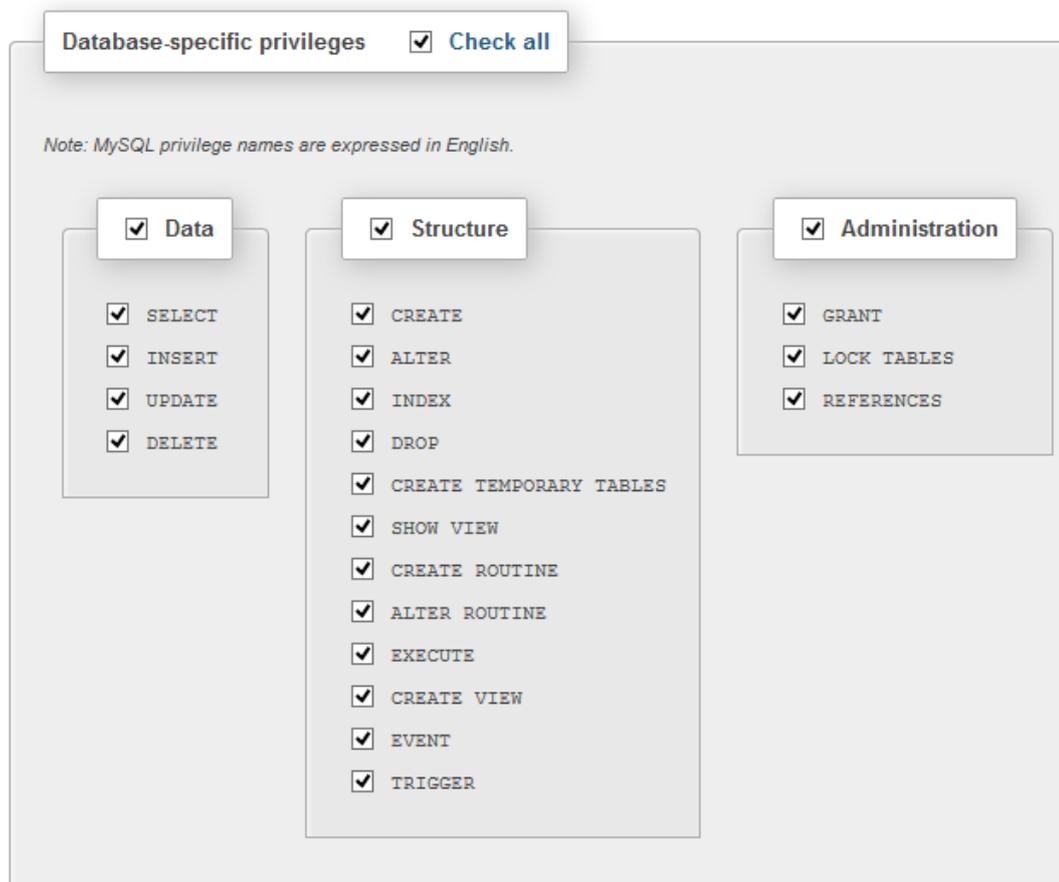
Pada kebijakan akses Master mempunyai hak penuh atas *database*. Master dapat melihat, mengubah, menghapus data pada *database* Sistem Pengambilan Ide Skripsi.

```
CREATE USER 'master_sis'@'localhost' IDENTIFIED BY 'master123';
```

Gambar 4. 12 Create Users Hak Akses Master

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

Gambar 4. 13 Privileges Administrator



Gambar 4. 14 Hak Akses Spesifik Master

2. Kebijakan Akses Administrator

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

```
CREATE USER 'tatania_adm'@'localhost' IDENTIFIED BY 'tatania123';
```

Gambar 4. 15 Create Users Hak Akses Administrator

Table-specific privileges					
Table	Privileges	Grant	Column-specific privileges	Action	
admin	SELECT	No	Yes	Edit privileges	Revoke
dosen	SELECT, DELETE	No	Yes	Edit privileges	Revoke
mahasiswa	SELECT, DELETE	No	Yes	Edit privileges	Revoke

Gambar 4. 16 Privileges Administrator

Berikut adalah kebijakan akses yang hanya bisa dilakukan admin, diantaranya:

- Login, melihat dan mengubah data pada tabel Administrator

Table-specific privileges					
<i>Note: MySQL privilege names are expressed in English.</i>					
SELECT	INSERT	UPDATE	REFERENCES		
<input type="checkbox"/> ID_ADM	<input type="checkbox"/> ID_ADM	<input type="checkbox"/> ID_ADM	<input type="checkbox"/> ID_ADM	<input type="checkbox"/>	<input type="checkbox"/> DELETE
<input type="checkbox"/> ID_USERS	<input type="checkbox"/> ID_USERS	<input type="checkbox"/> ID_USERS	<input type="checkbox"/> ID_USERS	<input type="checkbox"/>	<input type="checkbox"/> CREATE
<input type="checkbox"/> NAMA_ADM	<input type="checkbox"/> NAMA_ADM	<input type="checkbox"/> NAMA_ADM	<input type="checkbox"/> NAMA_ADM	<input type="checkbox"/>	<input type="checkbox"/> DROP
<input type="checkbox"/> ALAMAT_ADM	<input type="checkbox"/> ALAMAT_ADM	<input type="checkbox"/> ALAMAT_ADM	<input type="checkbox"/> ALAMAT_ADM	<input type="checkbox"/>	<input type="checkbox"/> GRANT
<input type="checkbox"/> NOHP_ADM	<input type="checkbox"/> NOHP_ADM	<input type="checkbox"/> NOHP_ADM	<input type="checkbox"/> NOHP_ADM	<input type="checkbox"/>	<input type="checkbox"/> INDEX
<input type="checkbox"/> EMAIL_ADM	<input type="checkbox"/> EMAIL_ADM	<input type="checkbox"/> EMAIL_ADM	<input type="checkbox"/> EMAIL_ADM	<input type="checkbox"/>	<input type="checkbox"/> ALTER
<input type="checkbox"/> USERNAME_ADM	<input type="checkbox"/> USERNAME_ADM	<input type="checkbox"/> USERNAME_ADM	<input type="checkbox"/> USERNAME_ADM	<input type="checkbox"/>	<input type="checkbox"/> CREATE_VIEW
<input type="checkbox"/> PASSWORD_ADM	<input type="checkbox"/> PASSWORD_ADM	<input type="checkbox"/> PASSWORD_ADM	<input type="checkbox"/> PASSWORD_ADM	<input type="checkbox"/>	<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"/>	<input type="checkbox"/> TRIGGER

Gambar 4. 17 Privileges Specific Admin tabel Administrator

- b. Melihat, mengubah *password* dan menghapus data pada tabel Mahasiswa

Table-specific privileges

Note: MySQL privilege names are expressed in English.

SELECT	INSERT	UPDATE	REFERENCES	<input checked="" type="checkbox"/> DELETE
NIM_MHS ID_USERS NAMA_MHS FAKULTAS_MHS PRODI_MHS SKS_MHS IPK_MHS ALAMAT_MHS	NIM_MHS ID_USERS NAMA_MHS FAKULTAS_MHS PRODI_MHS SKS_MHS IPK_MHS ALAMAT_MHS	PRODI_MHS SKS_MHS IPK_MHS ALAMAT_MHS NOHP_MHS EMAIL_MHS USERNAME_MHS PASSWORD_MHS STATUS_MHS	NIM_MHS ID_USERS NAMA_MHS FAKULTAS_MHS PRODI_MHS SKS_MHS IPK_MHS ALAMAT_MHS	<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
Or <input type="checkbox"/> None	Or <input type="checkbox"/> None	Or <input type="checkbox"/> None	Or <input type="checkbox"/> None	

Gambar 4. 18 Privileges Spesific Admin tabel Mahasiswa

- c. Melihat, mengubah *password*, mengubah status aktif atau tidaknya akun dosen dan menghapus data pada tabel Dosen

Table-specific privileges

Note: MySQL privilege names are expressed in English.

SELECT	INSERT	UPDATE	REFERENCES	<input checked="" type="checkbox"/> DELETE
NIDN_DSN ID_USERS NAMA_DSN FAKULTAS_DSN PRODI_DSN ALAMAT_DSN NOHP_DSN PENDIDIKAN_DS	NIDN_DSN ID_USERS NAMA_DSN FAKULTAS_DSN PRODI_DSN ALAMAT_DSN NOHP_DSN PENDIDIKAN_DS	PRODI_DSN ALAMAT_DSN NOHP_DSN PENDIDIKAN_DS KEAHLIAN_DSN EMAIL_DSN USERNAME_DSN PASSWORD_DS STATUS_DSN	NIDN_DSN ID_USERS NAMA_DSN FAKULTAS_DSN PRODI_DSN ALAMAT_DSN NOHP_DSN PENDIDIKAN_DS	<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
Or <input type="checkbox"/> None	Or <input type="checkbox"/> None	Or <input type="checkbox"/> None	Or <input type="checkbox"/> None	

Gambar 4. 19 Privileges Spesific Admin tabel Mahasiswa

3. Kebijakan Akses Mahasiswa

Pada kebijakan akses Mahasiswa, mahasiswa dapat melakukan beberapa kebijakan akses ke beberapa tabel yang ada di *database* Sistem Pengambilan Ide skripsi.

```
CREATE USER 'mahasiswa'@'localhost' IDENTIFIED BY 'mahasiswa123';
```

Gambar 4. 20 Create Users Hak Akses Mahasiswa

Table-specific privileges				
Table	Privileges	Grant	Column-specific privileges	Action
ideskripsi	SELECT	No	No	Edit privileges Revoke
mahasiswa	SELECT	No	Yes	Edit privileges Revoke
pengumuman	SELECT	No	Yes	Edit privileges Revoke

Gambar 4. 21 Privileges Mahasiswa

Berikut adalah kebijakan akses yang hanya bisa dilakukan admin, diantaranya:

- Login, melihat dan mengubah data pada tabel Mahasiswa kecuali *password* dan status mahasiswa.

Table-specific privileges				
<i>Note: MySQL privilege names are expressed in English.</i>				
SELECT	INSERT	UPDATE	REFERENCES	DELETE
NIM_MHS ID_USERS NAMA_MHS FAKULTAS_MHS PRODI_MHS SKS_MHS IPK_MHS ALAMAT_MHS	NIM_MHS ID_USERS NAMA_MHS FAKULTAS_MHS PRODI_MHS SKS_MHS IPK_MHS ALAMAT_MHS	NAMA_MHS FAKULTAS_MHS PRODI_MHS SKS_MHS ALAMAT_MHS NOHP_MHS EMAIL_MHS	NIM_MHS ID_USERS NAMA_MHS FAKULTAS_MHS PRODI_MHS SKS_MHS IPK_MHS ALAMAT_MHS	<input type="checkbox"/> DELETE <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
Or <input type="checkbox"/> None	Or <input type="checkbox"/> None	Or <input type="checkbox"/> None	Or <input type="checkbox"/> None	

Gambar 4. 22 Privileges Spesifik Mahasiswa tabel Mahasiswa

- b. Melihat, menambahkan data kecuali status pada tabel Pengumuman Ide Skripsi

Table-specific privileges

Note: MySQL privilege names are expressed in English.

SELECT	INSERT	UPDATE	REFERENCES	
ID_PIS ID_IDE NIM_MHS STATUS_PIS NILAI_MATKUL	ID_PIS ID_IDE NIM_MHS STATUS_PIS NILAI_MATKUL	ID_PIS ID_IDE NIM_MHS STATUS_PIS NILAI_MATKUL	ID_PIS ID_IDE NIM_MHS STATUS_PIS NILAI_MATKUL	<input type="checkbox"/> DELETE
				<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
Or <input type="checkbox"/> None	<input type="checkbox"/> SHOW_VIEW			
				<input type="checkbox"/> TRIGGER

Gambar 4. 23 Privileges Spesific Mahasiswa tabel PIS

- c. Hanya dapat melihat data pada tabel Ide Skripsi

Table-specific privileges

Note: MySQL privilege names are expressed in English.

SELECT	INSERT	UPDATE	REFERENCES	
ID_IDE NIDN_DSN IDE_SKRIPSI DESKRIPSI KUOTA_AMBIL MATKUL	ID_IDE NIDN_DSN IDE_SKRIPSI DESKRIPSI KUOTA_AMBIL MATKUL	ID_IDE NIDN_DSN IDE_SKRIPSI DESKRIPSI KUOTA_AMBIL MATKUL	ID_IDE NIDN_DSN IDE_SKRIPSI DESKRIPSI KUOTA_AMBIL MATKUL	<input type="checkbox"/> DELETE
				<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
Or <input type="checkbox"/> None	<input type="checkbox"/> SHOW_VIEW			
				<input type="checkbox"/> TRIGGER

Gambar 4. 24 Privileges Spesific Mahasiswa tabel Ide Skripsi

4. Kebijakan Akses Dosen

Pada kebijakan akses Dosen, dosen dapat melakukan beberapa kebijakan akses ke beberapa tabel yang ada di *database* sistem pengambilan ide skripsi.

```
CREATE USER 'dosen'@'localhost' IDENTIFIED BY 'dosen123';
```

Gambar 4. 25 Create Users Hak Akses Dosen

Table-specific privileges				
Table	Privileges	Grant	Column-specific privileges	Action
dosen	SELECT	No	Yes	Edit privileges Revoke
ideskripsi	SELECT, INSERT, DELETE	No	Yes	Edit privileges Revoke
pengumuman	SELECT	No	Yes	Edit privileges Revoke

Gambar 4. 26 Privileges Dosen

Berikut adalah kebijakan akses yang hanya bisa dilakukan admin, diantaranya:

- Login, melihat dan mengubah data pada tabel Dosen kecuali *password* dan status Dosen.

Table-specific privileges				
<i>Note: MySQL privilege names are expressed in English.</i>				
SELECT	INSERT	UPDATE	REFERENCES	<input type="checkbox"/> DELETE
<input checked="" type="checkbox"/> NIDN_DSN <input checked="" type="checkbox"/> ID_USERS <input checked="" type="checkbox"/> NAMA_DSN <input checked="" type="checkbox"/> FAKULTAS_DSN <input checked="" type="checkbox"/> PRODI_DSN <input checked="" type="checkbox"/> ALAMAT_DSN <input checked="" type="checkbox"/> NOHP_DSN <input checked="" type="checkbox"/> PENDIDIKAN_DS	<input checked="" type="checkbox"/> NIDN_DSN <input checked="" type="checkbox"/> ID_USERS <input checked="" type="checkbox"/> NAMA_DSN <input checked="" type="checkbox"/> FAKULTAS_DSN <input checked="" type="checkbox"/> PRODI_DSN <input checked="" type="checkbox"/> ALAMAT_DSN <input checked="" type="checkbox"/> NOHP_DSN <input checked="" type="checkbox"/> PENDIDIKAN_DS	<input checked="" type="checkbox"/> NIDN_DSN <input checked="" type="checkbox"/> ID_USERS <input checked="" type="checkbox"/> NAMA_DSN <input checked="" type="checkbox"/> FAKULTAS_DSN <input checked="" type="checkbox"/> PRODI_DSN <input checked="" type="checkbox"/> ALAMAT_DSN <input checked="" type="checkbox"/> NOHP_DSN <input checked="" type="checkbox"/> PENDIDIKAN_DS	<input checked="" type="checkbox"/> NIDN_DSN <input checked="" type="checkbox"/> ID_USERS <input checked="" type="checkbox"/> NAMA_DSN <input checked="" type="checkbox"/> FAKULTAS_DSN <input checked="" type="checkbox"/> PRODI_DSN <input checked="" type="checkbox"/> ALAMAT_DSN <input checked="" type="checkbox"/> NOHP_DSN <input checked="" type="checkbox"/> PENDIDIKAN_DS	<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
Or <input type="checkbox"/> None				

Gambar 4. 27 Privileges Specific Dosen tabel Dosen

- b. Melihat, menambahkan, mengubah dan menghapus data pada tabel Ide Skripsi

Table-specific privileges

Note: MySQL privilege names are expressed in English.

SELECT	INSERT	UPDATE	REFERENCES	
ID_IDE NIDN_DSN IDE_SKRIPSI DESKRIPSI KUOTA_AMBIL MATKUL	ID_IDE NIDN_DSN IDE_SKRIPSI DESKRIPSI KUOTA_AMBIL MATKUL	ID_IDE NIDN_DSN IDE_SKRIPSI DESKRIPSI KUOTA_AMBIL MATKUL	ID_IDE NIDN_DSN IDE_SKRIPSI DESKRIPSI KUOTA_AMBIL MATKUL	<input checked="" type="checkbox"/> DELETE
				<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
Or <input type="checkbox"/> None	<input type="checkbox"/> SHOW_VIEW			
				<input type="checkbox"/> TRIGGER

Gambar 4. 28 Privileges Spesific Dosen tabel Ide Skripsi

- c. Melihat dan mengubah kolom status pada tabel Pengumuman Ide Skripsi

Table-specific privileges

Note: MySQL privilege names are expressed in English.

SELECT	INSERT	UPDATE	REFERENCES	
ID_PIS ID_IDE NIM_MHS STATUS_PIS NILAI_MATKUL	ID_PIS ID_IDE NIM_MHS STATUS_PIS NILAI_MATKUL	ID_PIS ID_IDE NIM_MHS STATUS_PIS NILAI_MATKUL	ID_PIS ID_IDE NIM_MHS STATUS_PIS NILAI_MATKUL	<input type="checkbox"/> DELETE
				<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
Or <input type="checkbox"/> None	<input type="checkbox"/> SHOW_VIEW			
				<input type="checkbox"/> TRIGGER

Gambar 4. 29 Privileges Spesific Dosen tabel PIS

4.2 RAT (Rancangan Antar Tabel)

Rancangan Antar Tabel pada Sistem Pengambilan Ide Skripsi dapat dilihat pada gambar 4.29:



Gambar 4. 30 RAT Sistem Pengambilan Ide Skripsi

Penjelasan dari rancangan antar tabel pada Gambar 4.29 adalah:

- 1) Garis warna hijau menunjukkan adanya relasi dari tabel *Users* ke tabel Administrator, tabel Mahasiswa dan tabel Dosen.
- 2) Garis warna biru menunjukkan adanya relasi dari tabel Dosen ke tabel Ide Skripsi.
- 3) Garis warna kuning menunjukkan adanya relasi dari tabel Ide Skripsi ke tabel Pengumuman Ide Skripsi
- 4) Garis warna ungu menunjukkan adanya relasi dari tabel Mahasiswa ke tabel Pengumuman Ide Skripsi

- 5) Tabel *vdosen* merupakan hasil dari create tabel *View Dosen* untuk *view check* Pendidikan Dosen.
- 6) Tabel *vpengumuman* merupakan hasil dari create tabel *View Pengambilan Ide Skripsi* untuk *view check* Status Mahasiswa.

4.3 Pengujian

4.3.1 Metode Pengujian

Pengujian perangkat lunak dilakukan untuk memperoleh informasi serta mengevaluasi kualitas dari produk atau layanan yang sedang diuji. Tujuan pengujian dalam pengembangan *database* adalah untuk mengetahui apakah *database* yang diuji dapat memenuhi kebutuhan *admin* dan *users* dengan mendasari pada rancangan dan pengembangan perangkat lunak.

Metode pengujian yang dipakai dalam pengembangan *database* adalah *anomali testing* dan *view check*. Pengujian *anomali testing* dan *view check* dilakukan melalui *page* admin. *Anomali testing* berfungsi untuk mengetahui apakah proses basis data yang *memberikan* efek samping yang tidak diharapkan (misalnya menyebabkan tidak konsistennya data atau membuat suatu data menjadi hilang ketika data dihapus dan *view check* berfungsi untuk validasi data.

4.3.2 Pengujian Anomali

Pengujian *anomali* di lakukan terhadap tabel tabel memiliki relasi dari tabel satu ke tabel lainnya. Pengujian di lakukan sebanyak 5 kali *insert anomali*, *delete anomali* dan *update anomali*. Berikut adalah tabel yang memiliki relasi dengan tabel lain diantaranya adalah:

- 1) Tabel *Users* memiliki relasi dengan Tabel Administrator, Tabel Dosen dan Tabel Mahasiswa.
- 2) Tabel Dosen memiliki relasi dengan Tabel Ide Skripsi
- 3) Tabel Mahasiswa memiliki relasi dengan Tabel Pengumuman Ide Skripsi
- 4) Tabel Ide Skripsi memiliki relasi dengan tabel Pengumuman Ide Skripsi

1. Pengujian Anomali pada Tabel *Users*

a. *Insert* anomali

Insert anomali dilakukan pada tabel *Users* karena tabel *Users* memiliki relasi dengan Tabel Admin, Tabel Mahasiswa dan Tabel Dosen. Pengujian *insert* anomali dapat dilihat pada tabel 4.7.

Tabel 4. 7 Pengujian *Insert* Tabel *Users*

Tabel <i>Users</i>	Tabel yang berhubungan		
	Tabel Admin	Tabel Mahasiswa	Tabel Dosen
<i>Insert</i> pertama	Tidak ada perubahan	Tidak ada perubahan	Tidak ada perubahan
<i>Insert</i> kedua	Tidak ada perubahan	Tidak ada perubahan	Tidak ada perubahan
<i>Insert</i> ketiga	Tidak ada perubahan	Tidak ada perubahan	Tidak ada perubahan

Setelah dilakukan *insert* data sebanyak 3 kali pada tabel *Users*, tidak ada perubahan yang terjadi pada tabel Administrator, tabel Mahasiswa dan tabel Dosen. Maka tidak ada terjadi anomali pada pemasukan data.

b. *Update* anomali

Update anomali dilakukan pada tabel *Users* karena tabel *Users* memiliki relasi dengan Tabel Admin, Tabel Mahasiswa dan Tabel Dosen. Pengujian *update* anomali dapat dilihat pada tabel 4.8.

Tabel 4. 8 Pengujian *Update* Tabel *Users*

Tabel Users	Tabel yang berhubungan		
	Tabel Admin	Tabel Mahasiswa	Tabel Dosen
<i>Update</i> pertama	Ada perubahan	Ada perubahan	Ada perubahan
<i>Update</i> kedua	Ada perubahan	Ada perubahan	Ada perubahan
<i>Update</i> ketiga	Ada perubahan	Ada perubahan	Ada perubahan

Setelah dilakukan *update* data sebanyak 3 kali pada tabel *Users*, tabel yang memiliki relasi dan memiliki data yang sama akan berubah mengikuti data yang di *update* pada tabel *Users*, menandakan bahwa tidak terjadi anomali saat data di *update*.

c. *Delete* anomali

Update anomali dilakukan pada tabel *Users* karena tabel *Users* memiliki relasi dengan Tabel Admin, Tabel Mahasiswa dan Tabel Dosen. Pengujian *delete* anomali dapat dilihat pada tabel 4.9.

Tabel 4. 9 Pengujian *Delete* Tabel *Users*

Tabel Users	Tabel yang berhubungan		
	Tabel Admin	Tabel Mahasiswa	Tabel Dosen
<i>Delete</i> pertama	Tidak ada perubahan	Tidak ada perubahan	Tidak ada perubahan
<i>Delete</i> kedua	Tidak ada perubahan	Tidak ada perubahan	Tidak ada perubahan
<i>Delete</i> ketiga	Tidak ada perubahan	Tidak ada perubahan	Tidak ada perubahan

Setelah dilakukan *delete* data sebanyak 3 kali pada tabel *Users*, data dari tabel *Users* tidak dapat dihapus karena menggunakan *foreign key constraint ON*

DELETE RESTRICT, maka dari itu tidak ada perubahan yang terjadi di tabel mahasiswa dan tabel dosen walaupun memiliki data yang sama. Maka tidak terjadi anomali saat data dihapus.

2. Pengujian Anomali pada Tabel Dosen

a. *Insert* anomali

Insert anomali dilakukan pada tabel Dosen karena tabel Dosen memiliki relasi dengan tabel Ide Skripsi. Pengujian *insert* anomali dapat dilihat pada tabel 4.10.

Tabel 4. 10 Pengujian *Insert* Tabel Dosen

Tabel Dosen	Tabel yang berhubungan
	Tabel Ide Skripsi
<i>Insert</i> pertama	Tidak ada perubahan
<i>Insert</i> kedua	Tidak ada perubahan
<i>Insert</i> ketiga	Tidak ada perubahan

Setelah dilakukan *insert* data sebanyak 3 kali pada tabel Dosen, tidak ada perubahan yang terjadi pada tabel Ide Skripsi. Maka tidak ada terjadi anomali pada pemasukan data.

b. *Update* anomali

Update anomali dilakukan pada tabel Dosen karena tabel Dosen memiliki relasi dengan tabel Ide Skripsi. Pengujian *update* anomali dapat dilihat pada tabel 4.11.

Tabel 4. 11 Pengujian *Update* Tabel Dosen

Tabel Dosen	Tabel yang berhubungan
	Tabel Ide Skripsi
<i>Update</i> pertama	Ada perubahan
<i>Update</i> kedua	Ada perubahan
<i>Update</i> ketiga	Ada perubahan

Setelah dilakukan *update* data sebanyak 3 kali pada tabel Dosen, tabel yang memiliki relasi dan memiliki data yang sama akan berubah mengikuti data yang di *update* pada tabel Dosen, menandakan bahwa tidak terjadi anomali saat data di *update*.

c. *Delete* anomali

Delete anomali dilakukan pada tabel Dosen karena tabel Dosen memiliki relasi dengan Tabel Ide Skripsi. Pengujian *delete* anomali dapat dilihat pada tabel 4.12.

Tabel 4. 12 Pengujian *Delete* Tabel Dosen

Tabel Dosen	Tabel yang berhubungan
	Tabel Ide Skripsi
<i>Delete</i> pertama	Ada perubahan
<i>Delete</i> kedua	Ada perubahan
<i>Delete</i> ketiga	Ada perubahan

Setelah dilakukan *delete* data sebanyak 3 kali pada tabel *Dosen*, data dari tabel *Dosen* dapat dihapus karena menggunakan *foreign key constraint ON DELETE CASCADE*, data yang terhapus pada tabel *Ide Skripsi* hanya data yang memiliki relasi dengan data yang dihapus pada tabel *Dosen*, maka tidak terjadi anomali saat data dihapus.

3. Pengujian Anomali pada Tabel Mahasiswa

a. *Insert* anomali

Insert anomali dilakukan pada tabel *Mahasiswa* karena tabel *Mahasiswa* memiliki relasi dengan tabel *Pengumuman Ide Skripsi*. Pengujian *insert* anomali dapat dilihat pada tabel 4.13.

Tabel 4. 13 Pengujian *Insert* Tabel Mahasiswa

Tabel Mahasiswa	Tabel yang berhubungan
	Tabel Pengumuman Ide Skripsi
<i>Insert</i> pertama	Tidak ada perubahan
<i>Insert</i> kedua	Tidak ada perubahan
<i>Insert</i> ketiga	Tidak ada perubahan

Setelah dilakukan *insert* data sebanyak 3 kali pada tabel *Mahasiswa*, tidak ada perubahan yang terjadi pada tabel *Pengumuman Ide Skripsi*. Maka tidak ada terjadi anomali pada pemasukan data.

b. *Update* anomali

Update anomali dilakukan pada tabel *Mahasiswa* karena tabel *Mahasiswa* memiliki relasi dengan tabel *Pengumuman Ide Skripsi*. Pengujian *update* anomali dapat dilihat pada tabel 4.14.

Tabel 4. 14 Pengujian *Update* Tabel Mahasiswa

Tabel Mahasiswa	Tabel yang berhubungan
	Tabel Pengumuman Ide Skripsi
<i>Update</i> pertama	Ada perubahan
<i>Update</i> kedua	Ada perubahan
<i>Update</i> ketiga	Ada perubahan

Setelah dilakukan *update* data sebanyak 3 kali pada tabel Mahasiswa, tabel yang memiliki relasi dan memiliki data yang sama akan berubah mengikuti data yang di *update* pada tabel Mahasiswa, menandakan bahwa tidak terjadi anomali saat data di *update*.

c. *Delete* anomali

Delete anomali dilakukan pada tabel Mahasiswa karena tabel Mahasiswa memiliki relasi dengan Tabel Pengumuman Ide Skripsi. Pengujian *delete* anomali dapat dilihat pada tabel 4.15.

Tabel 4. 15 Pengujian *Delete* Tabel Mahasiswa

Tabel Mahasiswa	Tabel yang berhubungan
	Tabel Pengumuman Ide Skripsi
<i>Delete</i> pertama	Ada perubahan
<i>Delete</i> kedua	Ada perubahan
<i>Delete</i> ketiga	Ada perubahan

Setelah dilakukan *delete* data sebanyak 3 kali pada tabel Mahasiswa, data dari tabel Mahasiswa dapat dihapus karena menggunakan *foreign key constraint ON DELETE CASCADE*, data yang terhapus pada tabel Pengumuman Ide Skripsi hanya data yang memiliki relasi dengan data yang dihapus pada tabel Mahasiswa, maka tidak terjadi anomali saat data dihapus.

4. Pengujian Anomali pada Tabel Ide Skripsi

a. *Insert* anomali

Insert anomali dilakukan pada tabel Ide Skripsi karena tabel Ide Skripsi memiliki relasi dengan tabel Pengumuman Ide Skripsi. Pengujian *insert* anomali dapat dilihat pada tabel 4.16.

Tabel 4. 16 Pengujian *Insert* Tabel Ide Skripsi

Tabel Ide Skripsi	Tabel yang berhubungan
	Tabel Pengumuman Ide Skripsi
<i>Insert</i> pertama	Tidak ada perubahan
<i>Insert</i> kedua	Tidak ada perubahan
<i>Insert</i> ketiga	Tidak ada perubahan

Setelah dilakukan *insert* data sebanyak 3 kali pada tabel Ide Skripsi, tidak ada perubahan yang terjadi pada tabel Pengumuman Ide Skripsi. Maka tidak ada terjadi anomali pada pemasukan data.

b. *Update* anomali

Update anomali dilakukan pada tabel Ide Skripsi karena tabel Ide Skripsi memiliki relasi dengan tabel Pengumuman Ide Skripsi. Pengujian *update* anomali dapat dilihat pada tabel 4.17.

Tabel 4. 17 Pengujian *Update* Tabel Ide Skripsi

Tabel Ide Skripsi	Tabel yang berhubungan
	Tabel Pengumuman Ide Skripsi
<i>Update</i> pertama	Ada perubahan
<i>Update</i> kedua	Ada perubahan
<i>Update</i> ketiga	Ada perubahan

Setelah dilakukan *update* data sebanyak 3 kali pada tabel Ide Skripsi, tabel yang memiliki relasi dan memiliki data yang sama akan berubah mengikuti data yang di *update* pada tabel Ide Skripsi, menandakan bahwa tidak terjadi anomali saat data di *update*.

c. *Delete* anomali

Delete anomali dilakukan pada tabel Ide Skripsi karena tabel Ide Skripsi memiliki relasi dengan Tabel Pengumuman Ide Skripsi. Pengujian *delete* anomali dapat dilihat pada tabel 4.18.

Tabel 4. 18 Pengujian *Delete* Tabel Ide Skripsi

Tabel Ide Skripsi	Tabel yang berhubungan
	Tabel Pengumuman Ide Skripsi
<i>Delete</i> pertama	Ada perubahan
<i>Delete</i> kedua	Ada perubahan
<i>Delete</i> ketiga	Ada perubahan

Setelah dilakukan *delete* data sebanyak 3 kali pada tabel Ide Skripsi, data dari tabel Ide Skripsi dapat dihapus karena menggunakan *foreign key constraint ON DELETE CASCADE*, data yang terhapus pada tabel Pengumuman Ide Skripsi hanya data yang memiliki relasi dengan data yang dihapus pada tabel Ide Skripsi, maka tidak terjadi anomali saat data dihapus.

4.3.3 Pengujian View

View adalah perintah *query* yang disimpan pada *database* dengan suatu nama tertentu, sehingga bisa 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 data.

1. *View Check Option* tabel Pengumuman Ide Skripsi

View Check Option di implementasikan pada tabel Pengumuman Ide Skripsi kolom STATUS, pada kolom ini hanya diisi karakter Y atau N. Berikut adalah *query* untuk *View Check* pada tabel Pengumuman Ide Skripsi dapat dilihat pada gambar 4.30

```
CREATE VIEW vPengumuman AS
SELECT* FROM pengumuman WHERE STATUS_PIS IN ('Y','N')
WITH CHECK OPTION;|
```

Gambar 4. 31 Query Create View Check tabel PIS

```
INSERT INTO vpengumuman VALUES (12,5,'20130140012','A','PKW=A, IBD=A, HCI=B');
```

Gambar 4. 32 Query Insert vdosen tabel Dosen

```

Error

SQL query:

INSERT INTO vpengumuman VALUES (12,5,'20130140012','A','PKW=A, IBD=A, HCI=B')

MySQL said: ⓘ
#1369 - CHECK OPTION failed 'sistemideskripsi.vpengumuman'

```

Gambar 4. 33 Validasi View Check vpengumuman

Pada saat pengujian insert data yang tidak sesuai dengan pembatas data pada kolom STATUS yaitu ‘Y’ atau ‘N’, maka *query* tidak bisa masuk ke tabel pengumuman ide skripsi dan mengalami *error Check Option*. Hal ini berarti *View Check Option* telah berhasil di implementasikan pada tabel pengumuman ide skripsi.

2. View Check Option tabel Dosen

View Check Option di implementasikan pada tabel Dosen kolom PENDIDIKAN_DSN, pada kolom ini hanya diisi karakter S1, S2 dan S3. Berikut adalah *query* untuk *View Check* pada tabel Dosen dapat dilihat pada gambar 4.33

```

create view vDosen as
select * from dosen where PENDIDIKAN_DSN IN ('S1','S2','S3')
with check option;|

```

Gambar 4. 34 Query Create View Check tabel Dosen

```
INSERT INTO vdosen VALUES ('0526047404', 3, 'Rarira Rira Yuma', 'Teknik', 'Teknologi
Informasi', 'Sleman', '085794209081', 'TK', 'Programming', 'safira@gmail.com',
'safira', 'safira123');
```

Gambar 4. 35 Query insert vdosen tabel Dosen



Gambar 4. 36 Validasi View Check vdosen

Pada saat pengujian *insert* data yang tidak sesuai dengan pembatas data pada kolom PENDIDIKAN_DSN yaitu 'S1', 'S2' atau 'S3', maka *query* tidak bisa masuk ke tabel dosen dan mengalami *error Check Option*. Hal ini berarti *View Check Option* telah berhasil di implementasikan pada tabel dosen.