

# **LAPORAN PROYEK AKHIR BASIS DATA WEBSITE UJIAN ONLINE**

**Laporan ini disusun untuk mata kuliah : Basis Data**

**Dosen pengampu : Nasrul, S.Pd.I., S.Kom., M.Kom.**



**Anggota Kelompok :**

**Achmad Rifa'i Ramadhan (0110223138)**

**Sri Lusiana (0110223129)**

**Tiara (0110223123)**

**Revaldi Eka Setiawan (0110223301)**

**Program Studi Teknik Informatika**

**STT Terpadu Nurul Fikri**

**2024**

## **KATA PENGANTAR**

Puji syukur kehadiran Allah SWT atas limpahan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan proyek akhir mata kuliah Basis Data. Penulis bersyukur atas kesempatan dan pengalaman berharga yang diperoleh selama proses pengerjaan proyek ini.

Penyusunan laporan ini dalam rangka memenuhi tugas akhir Mata Kuliah Basis Data. Selain itu, proyek ini bertujuan untuk membangun sebuah sistem ujian online berbasis web yang dapat digunakan untuk menyelenggarakan ujian secara online. Penulis mengucapkan terima kasih kepada Bapak Nasrul, S.Pd.I., S.Kom., M.Kom. selaku dosen dan Bang Daden Dharmawan selaku Asisten Dosen Mata Kuliah Basis Data.

Penulis menyadari bahwa proyek ini masih memiliki banyak kekurangan. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun dari semua pihak untuk perbaikan proyek ini di masa depan.

Bogor, 20 Juni 2024

Penyusun

## DAFTAR ISI

<b>KATA PENGANTAR.....</b>	<b>2</b>
<b>DAFTAR ISI.....</b>	<b>3</b>
<b>BAB I.....</b>	<b>5</b>
<b>PENDAHULUAN.....</b>	<b>5</b>
1.1. Latar Belakang.....	5
<b>BAB II.....</b>	<b>6</b>
2.1. Pengertian Basis Data.....	6
2.2. Pengertian DBMS.....	6
2.3. Pengertian MySQL.....	7
2.4. Pengertian MySQL Workbench.....	7
2.5. Entity Relationship Diagram (ERD).....	7
<b>BAB III.....</b>	<b>9</b>
<b>PEMBAHASAN.....</b>	<b>9</b>
3.1. Desain Database.....	9
3.2. Struktur Database.....	12
3.2.1. Tabel User.....	12
3.2.2. Tabel Admin.....	13
3.2.3. Tabel Teacher.....	13
3.2.4. Tabel Student.....	13
3.2.5. Tabel Course.....	14
3.2.6. Tabel Quiz.....	14
3.2.7. Tabel Question.....	15
3.2.8. Tabel Answer.....	15
3.2.9. Tabel Student_Quiz.....	15
3.2.10. Tabel Score.....	16
3.2.11. Tabel Student_Answer.....	16
3.3. Input Data.....	17
3.3.1. User.....	17
3.3.2. Admin.....	17
3.3.3. Teacher.....	17
3.3.4. Student.....	17
3.3.5. Course.....	18
3.3.8. Answer.....	18
3.3.9. Student_Quiz.....	18
3.3.10. Score.....	19
3.3.11. Student_Answer.....	19
3.4. Inner Join dan Outer Join.....	19
3.4.1. Inner Join.....	19
3.4.2. Outer Join.....	20

3.4.2.1. Left Outer Join.....	20
3.4.2.2. Right Outer Join.....	20
3.5. Index dan View.....	21
3.5.1. Index.....	21
3.5.2. View.....	22
3.6. Transaction.....	22
3.6.1. Transaction.....	22
3.6.2. Commit.....	23
3.6.3. Rollback.....	23
3.6.4. Savepoint.....	24
3.7. Procedure.....	25
3.8. Trigger.....	26
3.9. DCL.....	26
3.9.1. Grant All Privileges.....	26
3.9.2. Grant Select, Insert, Update dan delete.....	27
3.9.3 Query Revoke.....	28
3.10.1. Backup.....	28
<b>BAB IV.....</b>	<b>30</b>
<b>KESIMPULAN DAN SARAN.....</b>	<b>30</b>
4.1. Kesimpulan.....	30
4.2. Saran.....	30

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

MySQL merupakan *software database open source* yang populer di dunia, dimana saat ini digunakan lebih dari 100 juta pengguna diseluruh dunia. Dengan kehandalan, kecepatan dan kemudahan penggunaannya, MySQL menjadi pilihan utama bagi banyak pengembang *software* dan aplikasi baik di *platform* web maupun *desktop*. Pengguna MySQL tidak hanya sebatas pengguna perseorangan maupun perusahaan kecil, namun perusahaan seperti Yahoo!, Alcatel-Lucent, Google, Nokia, Youtube, Wordpress dan Facebook juga merupakan pengguna MySQL.

MySQL pertama kali dibuat dan dikembangkan di Swedia, yaitu oleh David Axmark, Allan Larson dan Michael “Monty” Widenius. Mereka mengembangkan MySQL sejak tahun 1980-an. Saat ini versi MySQL yang sudah stabil mencapai versi 5x dan sedang dikembangkan versi 6x. untuk lebih lengkapnya dapat dilihat di situs resmi MySQL.

Tujuan Laporan ini disusun dengan tujuan untuk memberikan pengetahuan yang mendalam mengenai proses desain dan implementasi basis data menggunakan alat bantu MySQL Workbench. Selain itu, laporan ini juga bertujuan untuk menguraikan secara rinci berbagai konsep dan perintah SQL yang esensial dalam pengelolaan basis data, seperti JOIN untuk menggabungkan tabel, INDEX untuk meningkatkan kinerja pencarian, VIEW untuk menyederhanakan query, TRANSACTION untuk memastikan integritas data, PROCEDURE untuk otomatisasi tugas, TRIGGER untuk penanganan peristiwa, DCL untuk kontrol hak akses, serta BACKUP dan RESTORE untuk perlindungan data. Dengan pemahaman yang komprehensif terhadap teknik-teknik ini, diharapkan pembaca mampu menerapkan pengetahuan tersebut dalam pengelolaan basis data secara lebih efisien dan efektif.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Pengertian Basis Data**

Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Kasus ini menggunakan basis data relasional yang diimplementasikan dengan tabel-tabel yang saling memiliki relasi.

Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data apapun bentuknya, entah berupa *file* teks ataupun *Database Management System* (DBMS). Kebutuhan basis data dalam sistem informasi meliputi memasukkan, menyimpan, dan mengambil data, serta membuat laporan berdasarkan data yang telah disimpan.

Tujuan dari dibuatnya tabel-tabel dikasus ini adalah untuk menyimpan data ke dalam tabel-tabel agar mudah diakses. Oleh karena itu, untuk merancang tabel-tabel yang akan dibuat maka dibutuhkan pola pikir penyimpanan data nantinya jika dalam bentuk baris-baris data (*record*) dimana setiap baris terdiri dari beberapa kolom.

#### **2.2. Pengertian DBMS**

DBMS (*Database Management System*) atau dalam bahasa Indonesia sering disebut sebagai Sistem Manajemen Basis Data adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola, dan menampilkan data. Suatu sistem aplikasi disebut DBMS jika memenuhi persyaratan minimal sebagai berikut:

- 1) Menyediakan fasilitas untuk mengelola akses data
- 2) Mampu menangani integritas data
- 3) Mampu menangani akses data
- 4) Mampu menangani backup data

Karena pentingnya data bagi suatu organisasi/perusahaan, maka hampir sebagian besar perusahaan memanfaatkan DBMS dalam mengelola data yang mereka miliki. Pengelolaan DBMS sendiri biasanya ditangani oleh tenaga ahli yang spesialis mengenai DBMS yang disebut sebagai DBA (*Database Administrator*).

DBMS sudah mulai berkembang sejak tahun 1960an. Kemudian sekitar tahun 1970an mulai berkembang teknologi *Relational* DBMS yaitu DBMS berbasis relasional model. Relasional model pertama kali dikembangkan oleh Edgar J. Codd pada tahun

1970. Secara sederhana relasional model dapat dipahami sebagai suatu model yang memandang data sebagai sekumpulan tabel yang saling terkait. Hampir semua DBMS komersial dan *open source* saat ini berbasis *Relational* DBMS atau RDBMS.

### 2.3. Pengertian MySQL

SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk mengelola data pada RDBMS. SQL awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus. SQL mulai berkembang pada tahun 1970an. SQL mulai digunakan sebagai standar yang resmi pada tahun 1986 oleh ANSI (*American National Standards Institute*) dan pada tahun 1987 oleh ISO (*International Organization for Standardization*) dan disebut sebagai SQL-86. Pada perkembangannya, SQL beberapa kali dilakukan revisi. Berikut ini sejarah perkembangan SQL sampai saat ini:

- 1) Tahun 1986, SQL-86
- 2) Tahun 1989, SQL-89
- 3) Tahun 1992, SQL-92
- 4) Tahun 1999, SQL:1999
- 5) Tahun 2003, SQL:2003
- 6) Tahun 2006, SQL:2006
- 7) Tahun 2008, SQL:2008
- 8) Tahun 2011, SQL:2011

Meskipun SQL diadopsi dan diacu sebagai bahasa standar oleh hampir sebagian besar RDBMS yang beredar saat ini, tetapi tidak semua standar yang tercantum SQL diimplementasikan oleh seluruh DBMS tersebut. Sehingga kadang-kadang ada perbedaan perilaku (hasil yang ditampilkan) oleh DBMS yang berbeda padahal *query* yang dimasukkan sama.

### 2.4. Pengertian MySQL Workbench

MySQL Workbench adalah perangkat lunak yang powerful dan mudah digunakan untuk mengelola basis data MySQL. Perangkat ini menyediakan berbagai fitur untuk merancang, mengembangkan, dan mengelola basis data MySQL.

### 2.5. Entity Relationship Diagram (ERD)

*Entity Relationship Diagram* (ERD) adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak. *Entity*

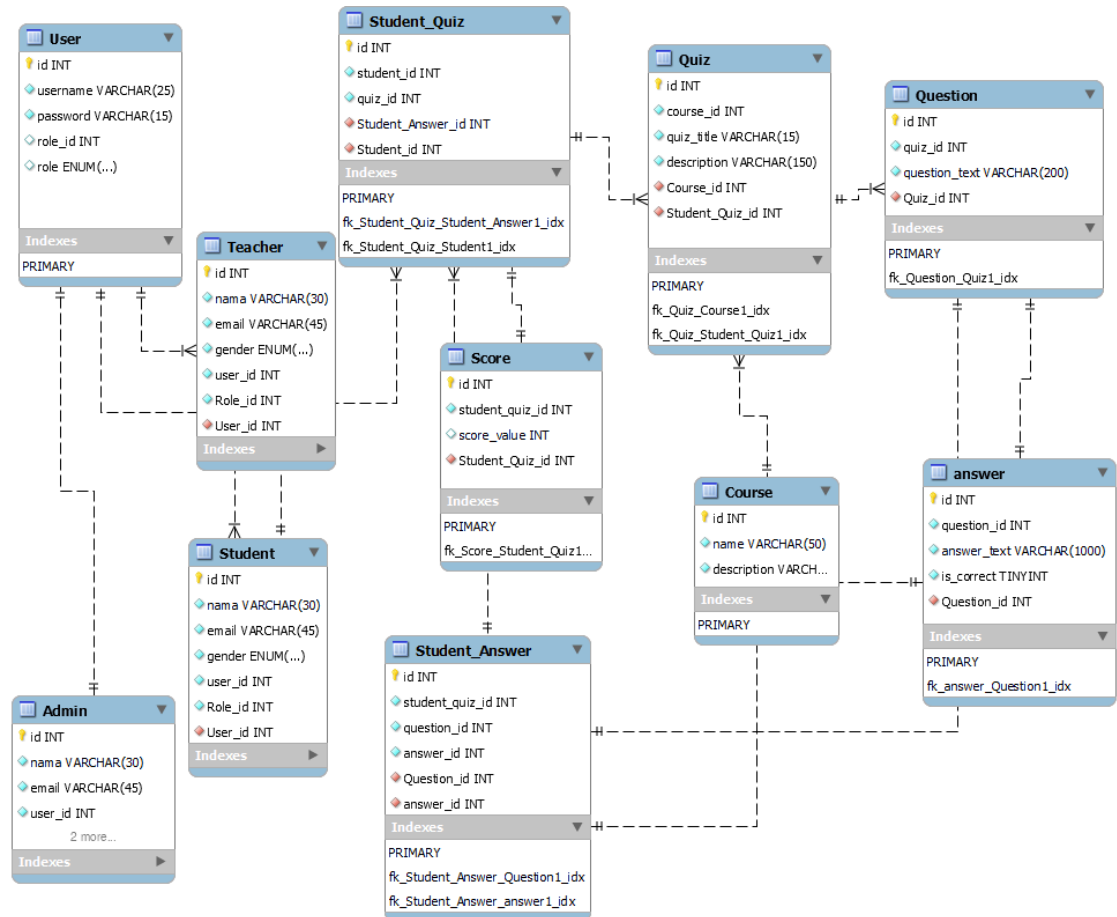
*Relationship Diagram* merupakan model jaringan yang menekankan pada struktur dan hubungan antardata. *Entity Relationship Diagram* juga memperlihatkan hubungan antardata store pada *Data Flow Diagram*. *Entity Relationship Diagram* atau lebih dikenal dengan E-R adalah notasi grafik dari sebuah model data atau sebuah model jaringan yang menjelaskan tentang data yang disimpan (storage data) dalam sistem secara abstrak. *Entity Relationship Diagram* tidak menyatakan bagaimana memanfaatkan data, membuat data, mengubah data dan menghapus data.



## BAB III

### PEMBAHASAN

#### 3.1. Desain Database



Desain database yang kami buat adalah sebuah desain yang dirancang untuk memenuhi kebutuhan sistem ujian online. Desain ini terdiri dari beberapa tabel yang saling berhubungan untuk memungkinkan pengelolaan data yang lebih efektif dan efisien. Tabel User berisi informasi pengguna umum, seperti nama pengguna, kata sandi, dan peran (admin, guru, atau siswa). Tabel Admin, Teacher, dan Student berisi informasi yang spesifik untuk tiap peran, seperti nama, email, dan user\_id yang merujuk ke tabel User. Tabel Admin berisi informasi admin secara spesifik, termasuk nama, email, dan user\_id yang merujuk ke tabel User. Tabel Teacher berisi informasi guru, termasuk nama, email, jenis kelamin, dan user\_id yang merujuk ke tabel User. Tabel Student berisi informasi siswa, termasuk nama, email, jenis kelamin, dan user\_id yang merujuk ke tabel User.

Tabel Course berisi informasi mata pelajaran, termasuk nama dan deskripsi, sedangkan tabel Quiz berisi informasi kuis, termasuk judul, deskripsi, mata pelajaran terkait (melalui `course_id`), dan guru yang membuat kuis (melalui `teacher_id`). Tabel Question berisi informasi pertanyaan dalam kuis, termasuk teks pertanyaan dan kuis terkait (melalui `quiz_id`), serta tabel Answer berisi informasi jawaban untuk pertanyaan kuis, termasuk teks jawaban, apakah jawaban benar (`is_correct`), dan pertanyaan terkait (melalui `question_id`).

Tabel Student\_Quiz berisi informasi kuis yang diambil oleh siswa, termasuk siswa yang mengikuti kuis (melalui `student_id`) dan kuis yang diambil (melalui `quiz_id`). Tabel Score berisi nilai yang diperoleh siswa untuk kuis tertentu, termasuk nilai (`score_value`) dan informasi kuis yang diikuti (melalui `student_quiz_id`). Tabel Student\_Answer berisi jawaban yang dipilih siswa untuk pertanyaan dalam kuis, termasuk informasi kuis yang diikuti (melalui `student_quiz_id`), pertanyaan yang dijawab (melalui `question_id`), dan jawaban yang dipilih (melalui `answer_id`).

Dengan menggunakan desain ini, sistem dapat dengan mudah mengelola data yang lebih efektif dan efisien, serta memungkinkan pengguna untuk melakukan pendaftaran, mengakses kursus, dan mengikuti quiz. Desain ini juga memungkinkan pengelolaan data yang lebih efektif dan efisien dengan menggunakan query yang lebih kompleks dan efisien. Oleh karena itu, desain database ini diharapkan dapat membantu meningkatkan fungsi sistem dan memenuhi kebutuhan pengguna yang lebih efektif dan efisien.

Pada MySQL Workbench, relasi antar tabel digunakan untuk menghubungkan satu tabel dengan tabel lainnya dalam basis data. Relasi ini mendefinisikan bagaimana data dalam tabel satu terkait dengan data dalam tabel lainnya. Diantaranya adalah One-to-One, One-to-Many dan Many-to-Many.

One-to-One atau Relasi satu-ke-satu terjadi ketika setiap baris dalam tabel A hanya dapat terhubung dengan satu baris dalam tabel B, dan sebaliknya. Contoh umumnya adalah hubungan antara tabel User dan UserProfile, di mana setiap pengguna memiliki satu profil pengguna.

One-to-Many atau Relasi satu-ke-banyak terjadi ketika satu baris dalam tabel A dapat terhubung dengan beberapa baris dalam tabel B, tetapi setiap baris dalam tabel B hanya dapat terhubung dengan satu baris dalam tabel A. Contoh kasusnya adalah hubungan antara tabel Orders (pesanan) dan Customers (pelanggan), di mana satu pelanggan dapat memiliki banyak pesanan.

Many-to-Many atau Relasi banyak-ke-banyak terjadi ketika setiap baris dalam tabel A dapat terhubung dengan beberapa baris dalam tabel B, dan sebaliknya. Contoh klasiknya adalah hubungan antara tabel Students (mahasiswa) dan Courses (mata kuliah), di mana satu mahasiswa dapat mendaftar ke banyak mata kuliah, dan setiap mata kuliah dapat diikuti oleh banyak mahasiswa.

Jenis relasi yang digunakan pada desain database sistem ujian online ini adalah sebagai berikut:

1) Relasi User dan Admin/Teacher/Student

Tabel User terhubung dengan tabel Admin, Teacher, dan Student melalui `user_id`. Setiap pengguna di tabel User dapat memiliki peran sebagai admin, guru, atau siswa. Relasi ini bersifat one-to-one karena setiap `user_id` hanya dapat memiliki satu peran, dan setiap peran hanya dapat dihubungkan dengan satu `user_id`.

2) Relasi Course dan Quiz

Tabel Course terhubung dengan tabel Quiz melalui `course_id`. Setiap kuis terkait dengan satu mata pelajaran. Relasi ini bersifat one-to-many karena satu mata pelajaran dapat memiliki banyak kuis, sedangkan satu kuis hanya dapat terhubung dengan satu mata pelajaran.

3) Relasi Quiz dan Question

Tabel Quiz terhubung dengan tabel Question melalui `quiz_id`. Setiap kuis memiliki beberapa pertanyaan. Relasi ini bersifat one-to-many karena satu kuis dapat memiliki banyak pertanyaan, sedangkan satu pertanyaan hanya dapat terhubung dengan satu kuis.

4) Relasi Question dan Answer

Tabel Question terhubung dengan tabel Answer melalui `question_id`. Setiap pertanyaan memiliki beberapa jawaban. Relasi ini bersifat one-to-many karena satu pertanyaan dapat memiliki banyak jawaban, sedangkan satu jawaban hanya dapat terhubung dengan satu pertanyaan.

5) Relasi Student dan Student\_Quiz

Tabel Student terhubung dengan tabel Student\_Quiz melalui `student_id`. Tabel Student\_Quiz menyimpan data kuis yang diambil oleh siswa. Relasi ini bersifat many-to-many karena satu siswa dapat mengikuti banyak kuis, dan satu kuis dapat diikuti oleh banyak siswa.

#### 6) Relasi Student\_Quiz dan Score

Tabel Student\_Quiz terhubung dengan tabel Score melalui student\_quiz\_id. Tabel Score menyimpan nilai yang diperoleh siswa untuk kuis tertentu. Relasi ini bersifat one-to-one karena setiap informasi kuis yang diikuti oleh siswa hanya dapat memiliki satu nilai, dan setiap nilai hanya dapat terhubung dengan satu informasi kuis yang diikuti oleh siswa.

#### 7) Relasi Student\_Quiz dan Student\_Answer

Tabel Student\_Quiz terhubung dengan tabel Student\_Answer melalui student\_quiz\_id. Tabel Student\_Answer menyimpan jawaban yang dipilih siswa dalam kuis. Relasi ini bersifat many-to-many karena satu informasi kuis yang diikuti oleh siswa dapat memiliki banyak jawaban yang dipilih, dan satu jawaban yang dipilih dapat terhubung dengan banyak informasi kuis yang diikuti oleh siswa.

#### 8) Relasi Question dan Student\_Answer

Tabel Question terhubung dengan tabel Student\_Answer melalui question\_id. Setiap jawaban siswa terkait dengan pertanyaan tertentu. Relasi ini bersifat many-to-many karena satu pertanyaan dapat memiliki banyak jawaban yang dipilih oleh siswa.

### 3.2. Struktur Database

#### 3.2.1. Tabel User

```
MariaDB [ujianonline]> desc user;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
username	varchar(25)	NO		NULL	
password	varchar(15)	NO		NULL	
role_id	int(11)	YES		NULL	
role	enum('Student','Teacher')	YES		NULL	

5 rows in set (0.032 sec)

- Kolom: id, username, password, role\_id
- Keterangan: Tabel ini menyimpan informasi pengguna umum seperti siswa, guru, atau admin. role\_id menentukan peran pengguna (role).

### 3.2.2. Tabel Admin

```
MariaDB [ujianonline]> desc admin;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
nama	varchar(30)	NO		NULL	
email	varchar(45)	NO		NULL	
user_id	int(11)	NO	MUL	NULL	

4 rows in set (0.024 sec)

- Kolom: id, nama, email, user\_id, role\_id
- Keterangan: Tabel ini menyimpan informasi admin. user\_id adalah foreign key yang menghubungkan ke tabel User untuk mengidentifikasi pengguna yang terkait.

### 3.2.3. Tabel Teacher

```
MariaDB [ujianonline]> desc teacher;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
nama	varchar(30)	NO		NULL	
email	varchar(45)	NO		NULL	
gender	enum('Laki-Laki', 'Perempuan')	NO		NULL	
user_id	int(11)	NO	MUL	NULL	
course_id	int(11)	YES		NULL	

6 rows in set (0.028 sec)

- Kolom: id, nama, email, gender, user\_id, role\_id
- Keterangan: Tabel ini menyimpan informasi guru. user\_id adalah foreign key ke tabel User untuk menghubungkan data guru dengan pengguna yang terkait.

### 3.2.4. Tabel Student

```
MariaDB [ujianonline]> desc student;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
nama	varchar(30)	NO		NULL	
email	varchar(45)	NO		NULL	
gender	enum('Laki-Laki', 'Perempuan')	NO		NULL	
user_id	int(11)	NO	MUL	NULL	

5 rows in set (0.061 sec)

- Kolom: id, nama, email, gender, user\_id, role\_id

- Keterangan: Tabel ini menyimpan informasi siswa. user\_id adalah foreign key ke tabel User untuk menghubungkan data siswa dengan pengguna yang terkait.

### 3.2.5. Tabel Course

```
MariaDB [ujianonline]> desc course;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
name	varchar(50)	NO		NULL	
description	varchar(150)	NO		NULL	

3 rows in set (0.031 sec)

- Kolom: id, name, description
- Keterangan: Tabel ini menyimpan informasi mengenai mata pelajaran atau kursus yang tersedia.

### 3.2.6. Tabel Quiz

```
MariaDB [ujianonline]> desc quiz;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
course_id	int(11)	NO	MUL	NULL	
quiz_title	varchar(15)	NO		NULL	
description	varchar(150)	NO		NULL	

4 rows in set (0.026 sec)

- Kolom: id, course\_id, quiz\_title, description, teacher\_id
- Keterangan: Tabel ini menyimpan informasi mengenai kuis yang terkait dengan suatu kursus. course\_id adalah foreign key ke tabel Course untuk mengidentifikasi kursus yang terkait, dan teacher\_id adalah foreign key ke tabel Teacher untuk mengidentifikasi guru yang membuat kuis tersebut.

### 3.2.7. Tabel Question

```
MariaDB [ujianonline]> desc question;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+-----+
| id         | int(11)   | NO   | PRI | NULL    | auto_increment |
| quiz_id    | int(11)   | NO   | MUL | NULL    |              |
| question_text | varchar(200) | NO   |     | NULL    |              |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.026 sec)
```

- Kolom: id, quiz\_id, question\_text
- Keterangan: Tabel ini menyimpan informasi pertanyaan yang terkait dengan suatu kuis. quiz\_id adalah foreign key ke tabel Quiz untuk mengidentifikasi kuis yang pertanyaannya terkait.

### 3.2.8. Tabel Answer

```
MariaDB [ujianonline]> desc answer;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+-----+
| id         | int(11)   | NO   | PRI | NULL    | auto_increment |
| question_id | int(11)   | NO   | MUL | NULL    |              |
| answer_text | varchar(1000) | NO   |     | NULL    |              |
| is_correct  | tinyint(4) | NO   |     | NULL    |              |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.031 sec)
```

- Kolom: id, question\_id, answer\_text, is\_correct
- Keterangan: Tabel ini menyimpan informasi jawaban untuk setiap pertanyaan kuis. question\_id adalah foreign key ke tabel Question untuk mengidentifikasi pertanyaan yang jawabannya terkait. Kolom is\_correct menandakan apakah jawaban tersebut benar atau salah.

### 3.2.9. Tabel Student\_Quiz

```
MariaDB [ujianonline]> desc student_quiz;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+-----+
| id         | int(11)   | NO   | PRI | NULL    | auto_increment |
| student_id | int(11)   | NO   | MUL | NULL    |              |
| quiz_id    | int(11)   | NO   |     | NULL    |              |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.021 sec)
```

- Kolom: id, student\_id, quiz\_id
- Keterangan: Tabel ini menyimpan informasi mengenai kuis yang diambil oleh siswa. student\_id adalah foreign key ke tabel Student untuk mengidentifikasi siswa yang mengambil kuis tersebut, dan quiz\_id adalah foreign key ke tabel Quiz untuk mengidentifikasi kuis yang diambil.

### 3.2.10. Tabel Score

```
MariaDB [ujianonline]> desc score;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
student_quiz_id	int(11)	NO	MUL	NULL	
score_value	int(11)	YES		NULL	

3 rows in set (0.032 sec)

- Kolom: id, student\_quiz\_id, score\_value
- Keterangan: Tabel ini menyimpan nilai yang diperoleh siswa untuk suatu kuis tertentu. student\_quiz\_id adalah foreign key ke tabel Student\_Quiz untuk mengidentifikasi kuis yang hasilnya terkait.

### 3.2.11. Tabel Student\_Answer

```
MariaDB [ujianonline]> desc student_answer;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
student_quiz_id	int(11)	NO		NULL	
question_id	int(11)	NO	MUL	NULL	
answer_id	int(11)	NO	MUL	NULL	

4 rows in set (0.029 sec)

- Kolom: id, student\_quiz\_id, question\_id, answer\_id
- Keterangan: Tabel ini menyimpan informasi mengenai jawaban yang dipilih oleh siswa untuk setiap pertanyaan dalam suatu kuis. student\_quiz\_id adalah foreign key ke tabel Student\_Quiz untuk mengidentifikasi kuis yang jawabannya terkait. question\_id adalah foreign key ke tabel Question untuk mengidentifikasi pertanyaan yang jawabannya dipilih. answer\_id adalah foreign key ke tabel Answer untuk mengidentifikasi jawaban yang dipilih oleh siswa.



### 3.3. Input Data

#### 3.3.1. User

```
MariaDB [ujianonline]> insert into user values
-> (null, 'student3', 'pass753', null, 'student'),
-> (null, 'student4', 'pass984', null, 'student'),
-> (null, 'student5', 'pass584', null, 'student');
Query OK, 3 rows affected (0.007 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

#### 3.3.2. Admin

```
MariaDB [ujianonline]> insert into admin values
-> (null, 'yanto', 'yanto@gmail.com', 1);
Query OK, 1 row affected (0.010 sec)
```

#### 3.3.3. Teacher

```
MariaDB [ujianonline]> insert into teacher values
-> (null, 'Rusmanto', 'Rus22@teacher.com', 'Laki-Laki', 4, 5),
-> (null, 'Reza', 'Reza762@teacher.com', 'Laki-Laki', 5, 3),
-> (null, 'Sirojul', 'Sirojul33@teacher.com', 'Laki-Laki', 5, 4),
-> (null, 'Sirojul', 'Sirojul33@teacher.com', 'Laki-Laki', 5, 6);
Query OK, 4 rows affected (0.011 sec)
Records: 4  Duplicates: 0  Warnings: 0
```

#### 3.3.4. Student

```
MariaDB [ujianonline]> insert into student values
-> (null, 'Queena Sabila', 'Queen32@student.com', 'Perempuan', 8),
-> (null, 'Akila', 'Akila72@student.com', 'Perempuan', 9),
-> (null, 'Mattiew Baker', 'baker45@student.com', 'Laki-Laki', 10);
Query OK, 3 rows affected (0.007 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

### 3.3.5. Course

```
MariaDB [ujianonline]> insert into course values
-> (null, 'Matematika', 'Basic Matematika Course'),
-> (null, 'Bahasa Inggris', 'Basic Bahasa Inggris Course'),
-> (null, 'IPA', 'Basic IPA Course'),
-> (null, 'IPS', 'Basic IPS Course'),
-> (null, 'PPKN', 'Basic PPKN Course'),
-> (null, 'Bahasa Indonesia', 'Basic Bahasa Indonesia Course'),
-> (null, 'Seni dan Prakarya', 'Basic Seni dan Prakarya Course'),
-> (null, 'Pendidikan Agama', 'Basic Pendidikan Agama Course'),
-> (null, 'PJOK', 'Basic PJOK Course');
Query OK, 9 rows affected (0.007 sec)
Records: 9 Duplicates: 0 Warnings: 0
```

### 3.3.8. Answer

```
MariaDB [ujianonline]> insert into answer values
-> (null, 2, 'The capital of France is Paris.', 1),
-> (null, 3, 'Fotosintesis adalah suatu kejadian alamiah yang muncul secara berulang pada tumbuhan, mulai dari daun sebagai organ utama yang menjadi tempat berlangsungnya proses ini.', 1),
-> (null, 4, 'garis khatulistiwa adalah garis khayal lintang nol derajat yang membagi bumi menjadi dua belahan sama besar yaitu belahan bumi utara dan belahan bumi selatan.', 1),
-> (null, 5, 'Pancasila adalah sebuah ideologi dan dasar negara yang untuk mewujudkan sebuah kenakmuran bagi masyarakat Indonesia.', 1),
-> (null, 6, 'Unsur intrinsik adalah unsur pembangun dari dalam cerpen.', 1),
-> (null, 7, 'Kerajinan tangan adalah produk yang dibuat secara manual dengan menggunakan alat sederhana dan keterampilan tangan.', 1),
-> (null, 8, 'Konsep tauhid dalam agama Islam merupakan prinsip fundamental yang mengajarkan bahwa hanya ada satu Tuhan, yaitu Allah, dan bahwa Dia adalah satu-satunya yang layak disembah.', 1);
Query OK, 7 rows affected (0.016 sec)
Records: 7 Duplicates: 0 Warnings: 0
```

### 3.3.9. Student Quiz

```
MariaDB [ujianonline]> insert into student_quiz values
-> (null, 2, 2),
-> (null, 3, 3),
-> (null, 4, 4),
-> (null, 5, 5),
-> (null, 1, 6),
-> (null, 2, 7),
-> (null, 3, 8),
-> (null, 4, 1),
-> (null, 5, 2);
Query OK, 9 rows affected (0.009 sec)
Records: 9 Duplicates: 0 Warnings: 0
```

### 3.3.10. Score

```
MariaDB [ujianonline]> insert into score values
-> (null, 2, 75),
-> (null, 3, 80),
-> (null, 4, 82),
-> (null, 5, 79),
-> (null, 6, 87),
-> (null, 7, 83),
-> (null, 8, 89),
-> (null, 9, 78),
-> (null, 10, 91);
Query OK, 9 rows affected (0.009 sec)
Records: 9  Duplicates: 0  Warnings: 0
```

### 3.3.11. Student\_Answer

```
MariaDB [ujianonline]> insert into student_answer values
-> (null, 2, 2, 1),
-> (null, 3, 3, 1),
-> (null, 4, 4, 1),
-> (null, 5, 5, 1),
-> (null, 6, 6, 1),
-> (null, 7, 7, 1),
-> (null, 8, 8, 1),
-> (null, 1, 2, 1);
Query OK, 8 rows affected (0.006 sec)
Records: 8  Duplicates: 0  Warnings: 0
```

## 3.4. Inner Join dan Outer Join

### 3.4.1. Inner Join

```
MariaDB [ujianonline]> select quiz.*, course.name as kategori
-> from quiz inner join course
-> on course.id = quiz.course_id;
```

id	course_id	quiz_title	description	kategori
1	1	Quiz Matematika	First Matematika Quiz	Matematika
2	2	Quiz Bahasa Ing	First English Quiz	Bahasa inggris
3	3	Quiz IPA 1	First IPA Quiz	IPA
4	4	Quiz IPS 1	First IPS Quiz	IPS
5	5	Quiz PPKN 1	First PPKN Quiz	PPKN
6	6	Quiz Bahasa Ind	First Bahasa Quiz	Bahasa Indonesia
7	7	Quiz Seni dan P	First Arts and Crafts Quiz	Seni dan Prakarya
8	8	Quiz Pendidikan	First Religious Education Quiz	Pendidikan Agama

```
8 rows in set (0.106 sec)
```

**Pengertian :** Inner Join adalah jenis join dalam SQL yang mengembalikan baris yang memiliki nilai yang cocok di kedua tabel yang bergabung.

**Tujuan :** Untuk menggabungkan baris-baris dari dua atau lebih tabel berdasarkan kondisi yang ditentukan, sehingga hanya baris-baris yang memenuhi kondisi penggabungan tersebut yang akan dimasukkan ke dalam hasil kueri.

**Penjelasan :** Pada bagian ini menjalani perintah inner join antara tabel course dengan quiz.course\_id. Dengan hasil menampilkan nama quiz, deskripsi dan kategori.

### 3.4.2. Outer Join

#### 3.4.2.1. Left Outer Join

```
MariaDB [ujianonline]> select quiz.*, question.question_text as pertanyaan
-> from quiz left join question
-> on question.quiz_id = quiz.id;
```

	id	course_id	quiz_title	description	pertanyaan
1	1	1	Quiz Matematika	First Matematika Quiz	berapakah hasil dari 17 + 2 - 3?
2	2	2	Quiz Bahasa Ing	First English Quiz	What skills do you want to improve?
3	3	3	Quiz IPA 1	First IPA Quiz	Apa itu fotosintesis?
4	4	4	Quiz IPS 1	First IPS Quiz	Apa itu garis khatulistiwa?
5	5	5	Quiz PPKN 1	First PPKN Quiz	Apa itu Pancasila dan apa fungsinya dalam kehidupan berbangsa dan bernegara di Indonesia?
6	6	6	Quiz Bahasa Ind	First Bahasa Quiz	Apa pengertian dari unsur intrinsik dan ekstrinsik dalam sebuah cerpen? Berikan contoh dari masing-masing unsur.
7	7	7	Quiz Seni dan D	First Arts and Crafts Quiz	Bagaimana proses pembuatan seni grafis menggunakan teknik cetak? Jelaskan langkah-langkahnya.
8	8	8	Quiz Pendidikan	First Religious Education Quiz	Apa pengertian dari konsep tauhid dalam agama Islam? Bagaimana tauhid mempengaruhi kehidupan sehari-hari umat Muslim?

8 rows in set (0.001 sec)

**Pengertian :** Left Outer Join mengembalikan semua baris dari tabel kiri (left table) dan baris yang cocok dari tabel kanan. Jika tidak ada kecocokan, hasilnya tetap menampilkan semua baris dari tabel kiri dengan nilai NULL untuk kolom dari tabel kanan.

**Tujuan :** Untuk menggabungkan baris-baris dari dua atau lebih tabel berdasarkan kondisi yang ditentukan, sambil mempertahankan semua baris dari tabel yang berada di sebelah kiri dalam penggabungan, bahkan jika tidak ada nilai kunci yang sesuai dari tabel lainnya.

**Penjelasan :** Pada tahap ini adalah menjalani kueri Left Outer Join dengan menggabungkan antara tabel quiz dengan tabel question. Lalu menampilkan kolom quiz\_title, description dan pertanyaan.

#### 3.4.2.2. Right Outer Join

```
MariaDB [ujianonline]> select question.*, quiz.description
-> from question right join quiz
-> on question.quiz_id = quiz.id;
```

	id	quiz_id	question_text	description
1	1	1	berapakah hasil dari 17 + 2 - 3?	First Matematika Quiz
2	2	2	What skills do you want to improve?	First English Quiz
3	3	3	Apa itu fotosintesis?	First IPA Quiz
4	4	4	Apa itu garis khatulistiwa?	First IPS Quiz
5	5	5	Apa itu Pancasila dan apa fungsinya dalam kehidupan berbangsa dan bernegara di Indonesia?	First PPKN Quiz
6	6	6	Apa pengertian dari unsur intrinsik dan ekstrinsik dalam sebuah cerpen? Berikan contoh dari masing-masing unsur.	First Bahasa Quiz
7	7	7	Bagaimana proses pembuatan seni grafis menggunakan teknik cetak? Jelaskan langkah-langkahnya.	First Arts and Crafts Quiz
8	8	8	Apa pengertian dari konsep tauhid dalam agama Islam? Bagaimana tauhid mempengaruhi kehidupan sehari-hari umat Muslim?	First Religious Education Quiz

8 rows in set (0.001 sec)

**Pengertian :** Right Outer Join mengembalikan semua baris dari tabel kanan (right

table) dan baris yang cocok dari tabel kiri. Jika tidak ada kecocokan, hasilnya tetap menampilkan semua baris dari tabel kanan dengan nilai NULL untuk kolom dari tabel kiri.

**Tujuan :** Untuk menggabungkan baris-baris dari dua atau lebih tabel berdasarkan kondisi yang ditentukan, sambil mempertahankan semua baris dari tabel yang berada di sebelah kanan dalam penggabungan, bahkan jika tidak ada nilai kunci yang sesuai dari tabel lainnya.

**Penjelasan :** Pada tahapan ini menggabungkan antara tabel question dengan tabel quiz dan menampilkan data kolom question\_text dan description.

### 3.5. Index dan View

#### 3.5.1. Index

```
MariaDB [ujianonline]> CREATE INDEX idx_nama_student on student(nama);
Query OK, 0 rows affected (0.016 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [ujianonline]>
```

**Pengertian :** Indeks dalam database adalah struktur data yang meningkatkan kecepatan operasi pengambilan data pada tabel database. Indeks bekerja seperti indeks di buku, yang memungkinkan pencarian cepat untuk menemukan baris yang diinginkan.

**Tujuan :** untuk meningkatkan kinerja pencarian dan pengurutan data dengan menyediakan akses cepat ke baris-baris yang sesuai dengan kriteria pencarian tertentu.

**Penjelasan :** Pada tahapan ini menjalani index nama student pada tabel student(nama).

### 3.5.2. View

```
MariaDB [ujianonline]> CREATE VIEW detailstudent AS SELECT
-> student.nama,
-> student.email,
-> student.gender,
-> user.username,
-> user.password
-> from student inner join user
-> on student.user_id = user.id;
Query OK, 0 rows affected (0.005 sec)

MariaDB [ujianonline]> SELECT * from detailstudent;
+-----+-----+-----+-----+-----+
| nama          | email                | gender  | username | password |
+-----+-----+-----+-----+-----+
| Muhammad Rifki | Rifki23@student.com | Laki-Laki | student1 | pass123 |
| Nadia Mustika  | Nadia34@student.com | Perempuan | student2 | pass789 |
| Queena Sabila  | Queen32@student.com | Perempuan | student3 | pass753 |
| Akila          | Akila72@student.com | Perempuan | student4 | pass984 |
| Mattiew Baker  | baker45@student.com | Laki-Laki | student5 | pass584 |
+-----+-----+-----+-----+-----+
5 rows in set (0.001 sec)

MariaDB [ujianonline]>
```

**Pengertian :** View dalam database adalah tabel virtual yang terdiri dari hasil query yang disimpan. View tidak menyimpan data fisik tetapi menyimpan query SQL yang menghasilkan data setiap kali view diakses.

**Tujuan :** untuk menyediakan pandangan virtual atau representasi terstruktur dari satu atau lebih tabel dalam basis data.

**Penjelasan :** Pada tahapan ini menjalani perintah view dengan menampilkan detail student dengan kolom nama, email, gender, username dan password.

## 3.6. Transaction

### 3.6.1. Transaction

```
MariaDB [ujianonline]> START TRANSACTION;
Query OK, 0 rows affected (0.000 sec)
```

**Pengertian :** Transaction dalam database adalah serangkaian operasi database yang dieksekusi sebagai satu unit kerja yang atomik. Transaksi memastikan bahwa semua operasi dalam satu unit kerja berhasil atau semuanya gagal, sehingga integritas data tetap terjaga.

**Tujuan :** untuk memastikan keakuratan, konsistensi, dan integritas data selama proses manipulasi data yang melibatkan operasi seperti INSERT, UPDATE, dan DELETE.

**Penjelasan :** Tahapan awal pada transaction dengan menjalani perintah “START TRANSACTION”.

### 3.6.2. Commit

```
MariaDB [ujianonline]> START TRANSACTION;
Query OK, 0 rows affected (0.000 sec)

MariaDB [ujianonline]> UPDATE student set nama = 'asep' where id = 1;
Query OK, 1 row affected (0.001 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [ujianonline]> commit;
Query OK, 0 rows affected (0.001 sec)

MariaDB [ujianonline]>
```

**Pengertian :** Commit dalam database adalah perintah yang digunakan untuk mengakhiri sebuah transaksi dan memastikan bahwa semua perubahan data yang terjadi selama transaksi tersebut disimpan secara permanen di dalam database.

**Tujuan :** untuk mengonfirmasi dan menerapkan perubahan yang dilakukan dalam sebuah transaksi SQL ke dalam basis data secara permanen.

**Penjelasan :** Tahapan kedua dalam transaction dengan mengupdate data student menjadi nama ‘asep’ yang mempunyai id=1.

### 3.6.3. Rollback

```
MariaDB [ujianonline]> UPDATE student set gender = 'Laki-laki' where id = 4;
Query OK, 1 row affected (0.001 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [ujianonline]> rollback;
Query OK, 0 rows affected (0.044 sec)

MariaDB [ujianonline]>
```

**Pengertian :** Rollback dalam database adalah perintah yang digunakan untuk membatalkan semua perubahan yang dilakukan dalam sebuah transaksi yang belum di-commit. Ketika rollback dijalankan, database dikembalikan ke keadaan seperti sebelum transaksi dimulai.

**Tujuan :** untuk membatalkan perubahan yang dilakukan dalam transaksi yang sedang berlangsung sehingga mengembalikan database ke keadaan sebelum transaksi dimulai.

**Penjelasan :** Pada tahapan ini mengubah data student dengan id=4 menjadi gender laki-laki.

#### 3.6.4. Savepoint

```
Database changed
MariaDB [ujianonline]> start transaction;
Query OK, 0 rows affected (0.001 sec)

MariaDB [ujianonline]> savepoint update_student;
Query OK, 0 rows affected (0.002 sec)
```

**Pengertian :** adalah titik dalam transaksi di mana Anda dapat menyimpan status transaksi saat ini sehingga Anda dapat kembali ke titik tersebut jika diperlukan.

**Tujuan :** untuk menyimpan titik tertentu dalam transaksi, sehingga jika terjadi kesalahan atau jika Anda perlu membatalkan sebagian dari perubahan yang dilakukan, Anda dapat mengembalikan transaksi ke savepoint tersebut tanpa harus membatalkan seluruh transaksi.

**Penjelasan :** Menyimpan data yang telah di buat agar tidak hilang semua saat ada perubahan.



### 3.7. Procedure

```
MariaDB [ujianonline]> delimiter ##
MariaDB [ujianonline]> create procedure showCourse()
    -> begin
    -> select * from course;
    -> end##
Query OK, 0 rows affected (0.050 sec)

MariaDB [ujianonline]> delimiter ;
MariaDB [ujianonline]> call showCourse();
```

id	name	description
1	Matematika	Basic Matematika Course
2	Bahasa inggris	Bahasa Inggris Basic Course
3	IPA	IPA Basic Course
4	IPS	IPS Basic Course
5	PPKN	PPKN Basic Course
6	Bahasa Indonesia	Bahasa Indonesia Basic Course
7	Seni dan Prakarya	Seni dan Prakarya Basic Course
8	Pendidikan Agama	Pendidikan Agama Basic Course

```
8 rows in set (0.102 sec)

Query OK, 0 rows affected (0.116 sec)
```

**Pengertian :** Procedure dalam database adalah mengacu pada objek yang mengandung serangkaian pernyataan SQL yang telah didefinisikan dan disimpan di dalam database.

**Tujuan :** Untuk mengotomatisasi tugas rutin, meningkatkan efisiensi, dan memastikan konsistensi operasi dalam basis data.

**Penjelasan :** Membuat sebuah procedure bernama showCourse untuk melihat isi dari table course.

### 3.8. Trigger

```
MariaDB [ujianonline]> delimiter //
```

```
MariaDB [ujianonline]> create or replace trigger updateUser
```

```
  -> after update on student
```

```
  -> for each row
```

```
  -> begin
```

```
    -> update user set password = 'pw12345' where id = 10;
```

```
  -> end//
```

```
Query OK, 0 rows affected (0.020 sec)
```

```
MariaDB [ujianonline]> update student set nama = 'Aqila Rista' where id = 4 //
```

```
Query OK, 1 row affected (0.010 sec)
```

```
Rows matched: 1 Changed: 1 Warnings: 0
```

```
MariaDB [ujianonline]> select * from student//
```

id	nama	email	gender	user_id
1	Muhammad Rifki	Rifki23@student.com	Laki-Laki	1
2	Nadia Mustika	Nadia34@student.com	Perempuan	3
3	Queen Sabila	Queen32@student.com	Perempuan	8
4	Aqila Rista	Akila72@student.com	Perempuan	9
5	Mattiew Baker	baker45@student.com	Laki-Laki	10

```
5 rows in set (0.001 sec)
```

```
MariaDB [ujianonline]> select * from user//
```

id	username	password	role_id	role
1	student1	pass123	NULL	Student
2	teacher1	pass456	NULL	Teacher
3	student2	pass789	NULL	Student
4	teacher2	pass321	NULL	Teacher
5	teacher3	pass654	NULL	Teacher
6	teacher4	pass987	NULL	Teacher
7	teacher5	pass531	NULL	Teacher
8	student3	pass753	NULL	Student
9	student4	pass984	NULL	Student
10	student5	pw12345	NULL	Student

```
10 rows in set (0.001 sec)
```

```
MariaDB [ujianonline]>
```

**Pengertian :** Trigger dalam database adalah jenis objek yang digunakan untuk memantau peristiwa tertentu yang terjadi pada tabel (seperti penambahan, pembaruan, atau penghapusan data) dan secara otomatis menjalankan serangkaian perintah SQL ketika peristiwa tersebut terjadi.

**Tujuan :** Untuk memastikan integritas data, otomatisasi tugas, dan menegakkan aturan bisnis dalam basis data.

**Penjelasan :** Membuat sebuah trigger untuk mengupdate data yang ada di dalam tabel user. namun, perintah update tersebut dapat berjalan setelah menjalankan perintah update pada tabel student.

### 3.9. DCL

#### 3.9.1. Grant All Privileges

```
MariaDB [ujianonline]> GRANT ALL PRIVILEGES ON ujianonline.student to student@localhost IDENTIFIED BY '123';
```

```
Query OK, 0 rows affected (0.041 sec)
```

```
MariaDB [ujianonline]> FLUSH PRIVILEGES;
```

```
Query OK, 0 rows affected (0.009 sec)
```

**Pengertian dan Tujuan :** Perintah Grant All Privileges dalam sistem manajemen basis data seperti MySQL digunakan untuk memberikan semua hak akses atau izin kepada pengguna tertentu pada basis data tertentu. Hak akses ini mencakup kemampuan untuk melakukan operasi seperti SELECT, INSERT, UPDATE, DELETE, CREATE.

**Penjelasan :** Pada tahap ini adalah membuat Grant All Privileges dengan nama student dengan password 123. Lalu tak lupa juga menjalankan perintah FLUSH.

### 3.9.2. Grant Select, Insert, Update dan delete

```
MariaDB [ujianonline]> grant select, insert, update, delete on ujianonline.student to student1@localhost identified by '123';
Query OK, 0 rows affected (0.041 sec)

MariaDB [ujianonline]> flush privileges;
Query OK, 0 rows affected (0.041 sec)
```

```
# mysql -u student1 ujianonline -p
Enter password: ***
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 81
Server version: 10.4.28-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [ujianonline]> desc student;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id    | int(11)       | NO   | PRI | NULL    | auto_increment |
| nama  | varchar(30)   | NO   |     | NULL    |                |
| email | varchar(45)   | NO   |     | NULL    |                |
| gender | enum('Laki-Laki','Perempuan') | NO   |     | NULL    |                |
| user_id | int(11)      | NO   | MUL | NULL    |                |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.088 sec)
```

```
MariaDB [ujianonline]> insert into student
-> values
-> (6, 'Nazila', 'Naji87@student.com', 'Perempaun', 4);
Query OK, 1 row affected, 1 warning (0.042 sec)

MariaDB [ujianonline]> update student set email = 'Muha23@student.com' where id = 1;
Query OK, 1 row affected (0.046 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [ujianonline]> delete from student where id = 4;
ERROR 1142 (42000): DELETE command denied to user 'student1'@'localhost' for table 'ujianonline`.`student`
MariaDB [ujianonline]>
```

**Pengertian dan Tujuan :** Dalam konteks database SQL, perintah-perintah GRANT digunakan untuk memberikan izin akses tertentu kepada pengguna atau peran tertentu terhadap objek database seperti tabel, view atau bahkan database itu sendiri.

**Penjelasan :** Pada tahapan ini adalah membuat grant select, insert, update, dan delete, dengan menambahkan mana student baru, mengupdate email, dan menghapus student.

### 3.9.3 Query Revoke

```
MariaDB [ujianonline]> revoke delete on ujianonline.student from student1@localhost;
Query OK, 0 rows affected (0.041 sec)

MariaDB [ujianonline]> flush privileges;
Query OK, 0 rows affected (0.002 sec)
```

```
MariaDB [ujianonline]> show grants for student1@localhost;
+-----+
| Grants for student1@localhost |
+-----+
| GRANT USAGE ON *.* TO 'student1'@'localhost' IDENTIFIED BY PASSWORD '*23AE809DDACAF96AF0FD78ED04B6A265E05AA257' |
| GRANT SELECT, INSERT, UPDATE ON 'ujianonline'.'student' TO 'student1'@'localhost' |
+-----+
2 rows in set (0.000 sec)
```

**Pengertian dan Tujuan :** Perintah Revoke dalam sistem manajemen basis data seperti MySQL digunakan untuk mencabut atau menghapus hak akses atau izin yang sebelumnya telah diberikan kepada pengguna tertentu pada basis data tertentu.

**Penjelasan :** Pada tahapan ini adalah membuat perintah revoke “delete” pada grant yang telah dibuat sebelumnya, dan tak lupa juga perintah FLUSH. Lalu tahapan berikutnya menjalani perintah show untuk melihat grant yang sudah dibuat sebelumnya.

## 3.10. Backup dan Restore

### 3.10.1. Backup

```
LENOVO Series@LAPTOP-0AM5M1EN d:\xampp
# mysqldump -u root -p ujianonline > D:\backup_ujianonline.sql
Enter password:
```

**Pengertian dan Tujuan :** Backup adalah proses membuat salinan data dari basis data bertujuan untuk pemulihan dan mencegah kehilangan data akibat kerusakan sistem, kesalahan pengguna, atau bencana lainnya.

**Penjelasan :** pada tahapan ini adalah membuat perintah backup dimana backup ini yaitu untuk menyimpan semua tabel.

### 3.10.2. Restore

```
# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 9
Server version: 10.4.28-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database ujianonline_restored;
Query OK, 1 row affected (0.003 sec)
```

```
# mysql -u root -p ujianonline_restored < D:\backup_ujianonline.sql
Enter password:
```

```
# mysql -u root -p ujianonline_restored
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 12
Server version: 10.4.28-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [ujianonline_restored]> show tables;
+-----+
| Tables_in_ujianonline_restored |
+-----+
| admin                           |
| answer                         |
| course                         |
| question                       |
| quiz                           |
| score                          |
| student                        |
| student_answer                 |
| student_quiz                   |
| teacher                        |
| user                           |
+-----+
11 rows in set (0.001 sec)
```

**Pengertian dan Tujuan :** Restore adalah proses mengembalikan data dari backup ke basis data setelah terjadi kehilangan data atau kerusakan. Restore memastikan data yang telah dibackup dapat digunakan kembali. Proses restore biasanya dilakukan setelah backup terakhir.

**Penjelasan :** Pada perintah yang terakhir ini adalah proses pengembalian data yang sudah di backup sebelumnya ke dalam SQL. Lalu pada tahapan ini juga menampilkan tabel yang ada pada database.

## **BAB IV**

### **KESIMPULAN DAN SARAN**

#### **4.1. Kesimpulan**

MySQL adalah sistem basis data open source yang populer dan andal, digunakan oleh lebih dari 100 juta pengguna termasuk perusahaan besar seperti Google dan Facebook. Basis data, khususnya yang bersifat relasional, mengorganisasikan data dalam tabel-tabel yang saling berhubungan, dan diatur oleh Database Management System (DBMS) seperti MySQL yang membantu mengelola akses, menjaga integritas, dan menangani backup data. Structured Query Language (SQL) adalah bahasa standar untuk mengelola data dalam RDBMS, dan MySQL Workbench adalah alat yang sangat berguna untuk merancang, mengembangkan, dan mengelola basis data MySQL dengan fitur visualisasi, forward engineering, dan reverse engineering. Entity Relationship Diagram (ERD) sangat membantu dalam visualisasi struktur dan hubungan antar data dalam sistem.

#### **4.2. Saran**

Untuk pengembangan dan pengelolaan basis data di masa depan, penting untuk mengikuti pelatihan berkelanjutan agar tetap up-to-date dengan teknologi terbaru. Implementasi indeks pada kolom yang sering digunakan dapat meningkatkan kinerja query, sementara strategi backup dan recovery yang kuat akan mencegah kehilangan data. Langkah-langkah keamanan seperti enkripsi data dan pengaturan hak akses sangat penting untuk melindungi data dari akses tidak sah. Pemantauan dan optimasi kinerja database secara rutin akan memastikan performa yang optimal, dan penggunaan alat bantu seperti MySQL Workbench akan memudahkan pengelolaan basis data secara efisien dan efektif. Dengan menerapkan langkah-langkah ini, basis data yang dirancang dapat berfungsi optimal dan memberikan manfaat maksimal bagi pengguna.