# NEGERI STATES

### SMK NEGERI 2 KARANGANYAR JOBSHEET DASAR-DASAR PENGEMBANGAN

#### OBSHEET DASAR-DASAR PENGEMBANGAN PERANGKAT LUNAK DAN GIM

Disusun Oleh:
Oktiana Nuraini

DDL (Data Definition Language)
Revisi: 01
September 2022

September 2022 Kelas X

 $4 \times 45$ "

#### A. Tujuan

Peserta didik dapat mengenal salah satu bahasa SQL (*Structured Query Language*), yaitu DDL (*Data Definition Language*) untuk melakukan operasi *database*, operasi tabel, dan untuk mengelola *field* (kolom).

#### B. Dasar Teori

#### 1. Bahasa SQL

SQL (Structured Query Language) adalah sebuah bahasa yang digunakan untuk mengakses data dalam software DBMS. Bahasa ini merupakan bahasa standar yang digunakan dalam manajemen basis data relasional yang memungkinkan kita untuk berkomunikasi dengan basis data, membaca, menulis, dan memperoleh informasi yang berguna dari basis data bahasa SQL. Secara umum, SQL terdiri dari tiga bahasa, yaitu Data Definition Language (DDL), Data Manipulation Language (DML), dan Data Control Language (DCL).

#### 2. DDL (Data Definition Language)

DDL adalah sebuah metode *query* SQL yang berguna untuk mendefinisikan data pada sebuah basis data. *Query* yang dimiliki DDL adalah:

- a. CREATE: Digunakan untuk membuat basis data dan tabel
- b. DROP : Digunakan untuk menghapus basis data dan tabel
- c. ALTER: Digunakan untuk melakukan perubahan struktur tabel yang telah dibuat, baik menambah *field* (ADD), mengganti nama *field* (CHANGE) ataupun menamakannya kembali (RENAME) dan menghapus *field* (DROP).

#### C. Alat dan Bahan

Alat

- 1. PC/Laptop dengan Sistem Operasi Windows
- 2. Koneksi Internet
- 3. Software XAMPP
- 4. Command Prompt atau CMD

Bahan :

Jobsheet Dasar-Dasar Pengembangan Perangkat Lunak dan Gim – DDL (Data Definition Language)



#### SMK NEGERI 2 KARANGANYAR JOBSHEET DASAR-DASAR PENGEMBANGAN

#### BSHEET DASAR-DASAR PENGEMBANGAN PERANGKAT LUNAK DAN GIM

Semester Gasal **DDL** (*Data Definition Language*)  $4 \times 45$ "

Disusun Oleh: Revisi: 01 September 2022 Kelas X

#### D. Langkah Kerja

#### **LATIHAN**

#### 1. Operasi Database

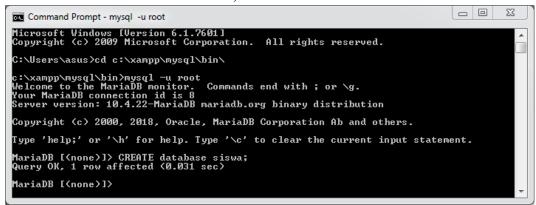
a. Membuat database

Untuk membuat *database* baru, kita dapat menggunakan perintah:

#### **CREATE database** nama\_database;

Oktiana Nuraini

Contoh: CREATE database siswa;



#### b. Melihat seluruh database

Untuk menampilkan daftar *database* di server, kita dapat menggunakan perintah:

#### **SHOW** databases;



#### c. Mengakses database

Untuk memilih *database* sebagai *database* aktif, kita dapat menggunakan perintah:

#### **USE** nama\_database;

Contoh: USE siswa;





## SMK NEGERI 2 KARANGANYAR JOBSHEET DASAR-DASAR PENGEMBANGAN PERANGKAT LUNAK DAN GIM Semester Gasal DDL (Data Definition Language) 4 × 45" Disusun Oleh: Oktiana Nuraini Revisi : 01 September 2022 Kelas X

#### d. Menghapus database

Untuk menghapus *database*, kita dapat menggunakan perintah:

**DROP** database nama database;

Contoh: DROP database siswa;



Pada gambar di atas dapat dilihat bahwa setelah menjalankan perintah **DROP** untuk menghapus *database* siswa, maka saat perintah **SHOW databases** dijalankan, *database* siswa sudah tidak ada.

#### 2. Operasi Tabel

a. Membuat tabel

Untuk membuat tabel baru, kita dapat menggunakan perintah:

```
CREATE table nama_database(namaField_1 tipeData_1, namaField_2 tipeData_2, dst);
```

Nama tabel tidak boleh mengandung spasi (*space*). Field\_1 dan tipeData\_1 merupakan nama kolom pertama dan tipe data untuk kolom pertama. Jika ingin membuat tabel dengan kolom lebih dari satu, maka setelah pendefinisian tipe data sebelumnya diberikan tanda koma (,).

Contoh: Berikut ini perintah untuk membuat tabel siswa pada *database* siswa:

#### **CREATE** table siswa(

```
nis CHAR(10) NOT NULL,
nama VARCHAR(60) NOT NULL,
jenis_kelamin CHAR(1) NOT NULL,
alamat VARCHAR(120),
PRIMARY KEY(nis)
```

);



#### SMK NEGERI 2 KARANGANYAR

#### JOBSHEET DASAR-DASAR PENGEMBANGAN PERANGKAT LUNAK DAN GIM

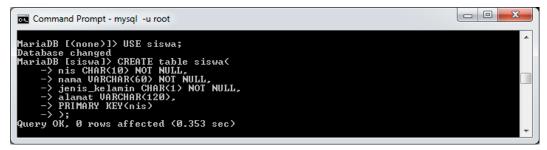
Semester Gasal **DDL** (*Data Definition Language*) 4 × 45"

Disusun Oleh:

Oktiana Nuraini

Revisi: 01 September 2022

Kelas X



Jelaskan apa kegunaan CHAR, VARCHAR, NOT NULL, dan PRIMARY KEY yang digunakan di atas!

#### b. Menampilkan tabel

Untuk menampilkan daftar nama tabel yang ada pada *database* yang sedang aktif/digunakan, kita dapat menggunakan perintah:

#### **SHOW** tables;

```
MariaDB [siswal> SHOW tables;

Tables_in_siswa |

siswa |

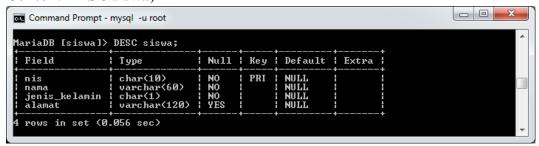
row in set (0.002 sec)
```

#### c. Menampilkan atribut tabel

Untuk menampilkan deskripsi tabel, kita dapat menggunakan perintah:

**DESC** nama\_tabel; atau **DESCRIBE** nama tabel;

#### Contoh: DESC siswa;



#### d. Memodifikasi nama tabel

Untuk memodifikasi atau mengubah nama tabel, kita dapat menggunakan perintah:

#### ALTER TABLE nama\_tabel\_lama RENAME nama\_tabel\_baru;

Contoh: Berikut ini perintah untuk mengubah nama tabel siswa menjadi identitas\_siswa:

#### ALTER TABLE siswa RENAME identitas\_siswa;



#### SMK NEGERI 2 KARANGANYAR

#### JOBSHEET DASAR-DASAR PENGEMBANGAN PERANGKAT LUNAK DAN GIM

Semester Gasal **DDL** (*Data Definition Language*) 4 × 45"

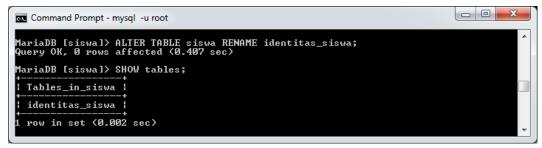
Disusun Oleh:

Oktiana Nuraini

Revisi: 01

September 2022

Kelas X



#### 3. Mengelola Field (Kolom)

1. Menambah *field* baru

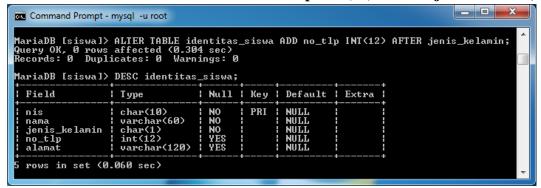
Untuk menambah *field* baru, kita dapat menggunakan perintah:

**ALTER TABLE** nama\_tabel **ADD** namaFieldBaru tipeDataBaru FIRST/AFTER namaField;

Perintah **FIRST** akan membuat *field* baru ditambahkan sebagai *field* pertama sedangkan perintah **AFTER namaFiel**d akan membuat *field* baru ditambahkan setelah *field* yang ditentukan. Apabila perintah FIRST/AFTER tidak dicantumkan, maka secara *default*, *field* baru akan ditambahkan sebagai *field* terakhir.

Contoh: Berikut akan menambah *field* no\_tlp pada tabel identitas\_siswa dan diletakkan setelah *field* jenis\_kelamin:

#### ALTER TABLE identitas\_siswa ADD no\_tlp INT(12) AFTER jenis\_kelamin;



#### 2. Mengubah nama field

Untuk memodifikasi atau mengubah nama *field*, kita dapat menggunakan perintah:

**ALTER TABLE** nama\_tabel **CHANGE** namaFieldLama namaFieldBaru tipeDataBaru;

Contoh: Berikut akan mengubah *field* no\_tlp pada tabel identitas\_siswa menjadi no\_hp:

ALTER TABLE identitas\_siswa CHANGE no\_tlp no\_hp INT(13);



### SMK NEGERI 2 KARANGANYAR

#### JOBSHEET DASAR-DASAR PENGEMBANGAN PERANGKAT LUNAK DAN GIM

Semester Gasal **DDL** (*Data Definition Language*) 4 × 45"

Disusun Oleh: Revisi: 01 September 2022 Kelas X

MariaDB [siswa]> ALTER TABLE identitas\_siswa CHANGE no\_tlp no\_hp INT(13);
Query OK, Ø rows affected (0.124 sec)
Records: Ø Duplicates: Ø Warnings: Ø

MariaDB [siswa]> DESC identitas\_siswa;

Field | Type | Null | Key | Default | Extra |

inis | char(10) | NO | PRI | NULL |

inama | varchar(60) | NO | NULL |

jenis\_kelamin | char(1) | NO | NULL |

ino\_hp | int(13) | YES | NULL |

alamat | varchar(120) | YES | NULL |

5 rows in set (0.013 sec)

#### 3. Menghapus field

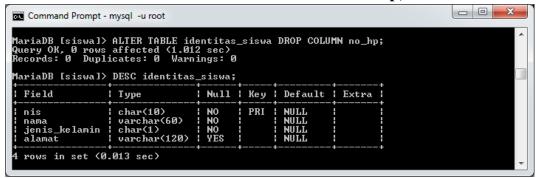
Untuk menghapus *field*, kita dapat menggunakan perintah:

Oktiana Nuraini

#### ALTER TABLE nama\_tabel DROP COLUMN namaField;

Contoh: Berikut merupakan perintah untuk menghapus *field* no\_hp dari tabel identitas\_siswa:

#### ALTER TABLE identitas\_siswa DROP COLUMN no\_hp;



#### **TUGAS**

- 1. Buat tabel 'mata pelajaran' yang memiliki struktur tabel terdiri dari *fields* ('kode mata pelajaran', 'nama mata pelajaran', dan 'durasi jam pelajaran'). *Screenshot* hasil struktur tabel yang terbentuk!
- 2. Tambahkan *field* 'guru pengampu' pada tabel 'mata pelajaran' yang telah kamu buat. *Screenshot* hasil struktur tabel yang terbentuk!
- 3. Tambahkan *field* 'nomor' pada tabel 'mata pelajaran' yang telah kamu buat, di mana *field* 'nomor' berada pada posisi *field* yang pertama. *Screenshot* hasil struktur tabel yang terbentuk!
- 4. Tambahkan *field* 'ruang' pada tabel 'mata pelajaran', di mana field 'ruang' berada pada posisi setelah field 'durasi jam pelajaran'. *Screenshot* hasil struktur tabel yang terbentuk!