

**IF2240 - Basis Data**

# **Tugas Besar**

**Milestone 2: Implementasi Basis Data**



Disusun Oleh:

Hazelnut Milk Tea with Pudding

**Program Studi Teknik Informatika**

**Sekolah Teknik Elektro dan Informatika**

**Institut Teknologi Bandung**

**2024**

## IDENTITAS KELOMPOK

Nomor Kelompok : 10

Kode Kelompok : GWS

Nama Kelompok : Hazelnut Milk Tea with Pudding

Anggota Kelompok : 1. Dwi Kalam Amal Tauhid (13519210)  
2. Akbar Al Fattah (13522036)  
3. Imanuel Sebastian Girsang (13522058)  
4. Ahmad Mudabbir Arif (13522072)  
5. Sa'ad Abdul Hakim (13522092)

Kelas : K2

Dosen Pengampu : Dr. Fazat Nur Azizah, S.T, M.Sc

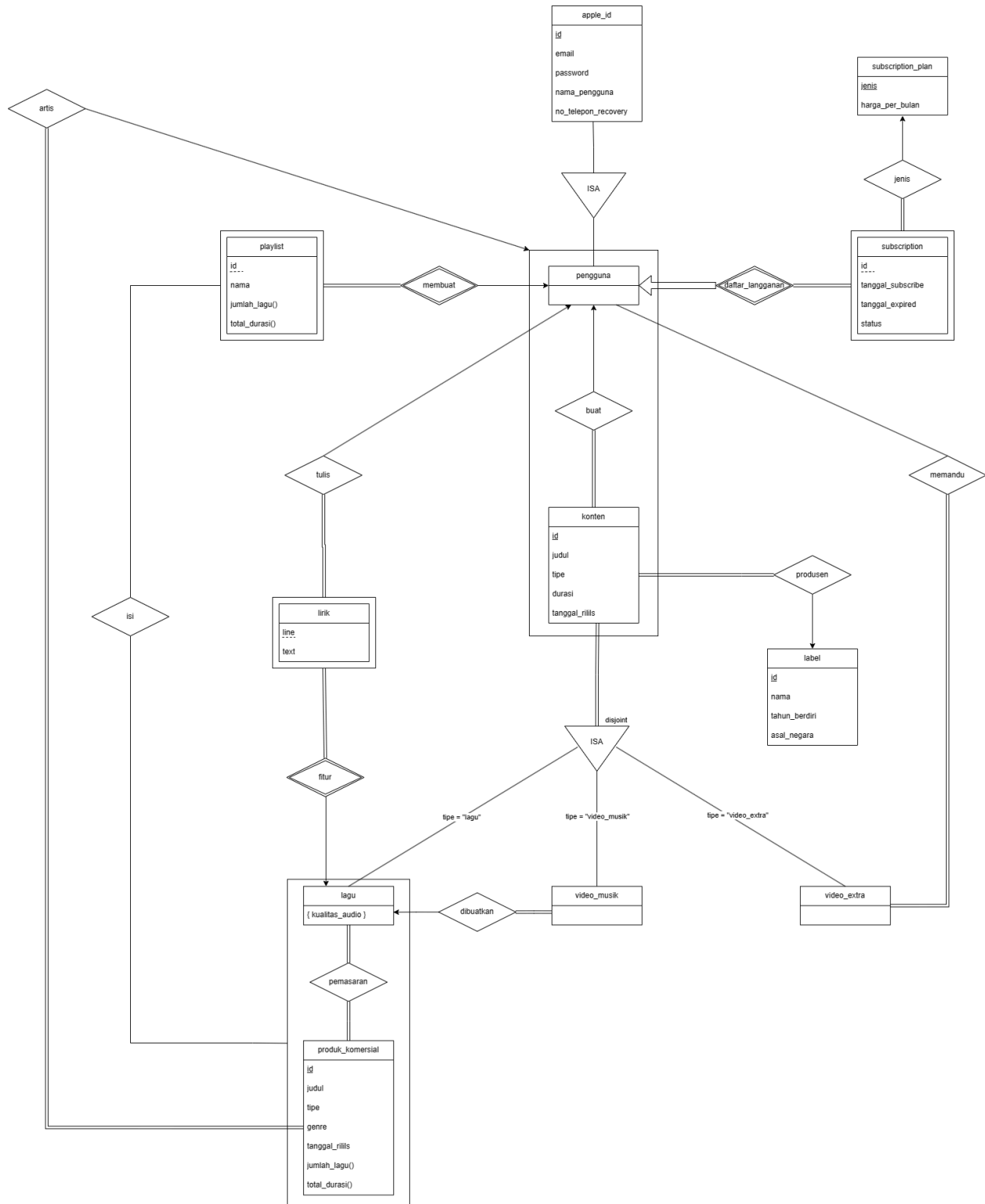
Problem Set : Apple Music

Nama Asisten : Anthonio Natthan Krishna (13521162)

Tanggal Pengumpulan : 7 Mei 2024

# 1. MODEL ENTITY-RELATIONSHIP

## a. Entity-Relationship Diagram



[Gambar 1.a.1. Entity-relationship diagram \(ERD\) Apple Music Hasil Revisi.](#)

b. Deskripsi

Berdasarkan ER *model* pada gambar 3.a.1. terdapat beberapa revisi yang dilakukan terhadap ER *model* sebelumnya.

Daftar *entity set* yang direvisi beserta penjelasannya dapat dilihat pada tabel 1.b.1.

Tabel 1.b.1. Daftar entity set revisi dan penjelasannya.

No	Entity Set	Penjelasan
1	pengguna	<p><b>pengguna</b> adalah <i>entity set</i> yang menyatakan Pengguna Apple Music, yang dalam hal ini, pernah atau sedang berlangganan terhadap suatu <b>subscription_plan</b>. Diasumsikan dapat diidentifikasi mana <b>pengguna</b> yang masih aktif dan tidak aktif berlangganan berdasarkan atribut pada <i>weak entity set</i> <b>subscription</b> yaitu <b>status</b> yang menyatakan status aktif dan tidak aktif berlangganan.</p> <p>Atribut:</p> <ul style="list-style-type: none"> <li>- <i>inherit</i> <b>apple_id</b></li> <li>- <del>tanggal_renewal</del>: not null</li> </ul>
2	<del>writer</del>	<i>Entity set <b>writer</b> ini dihilangkan.</i>
3	<del>artis</del>	<i>Entity set <b>artis</b> ini dihilangkan.</i>
4	<del>host</del>	<i>Entity set <b>host</b> ini dihilangkan.</i>
5	subscription	<p><b>subscription</b> adalah <i>weak entity set</i> yang menyatakan <b>subscription</b> atau daftar langganan yang dimiliki <b>pengguna</b>.</p> <p>Atribut:</p> <ul style="list-style-type: none"> <li>- Id: diskriminator, not null</li> <li>- tanggal_subscribe, not null</li> </ul>

		<ul style="list-style-type: none"> <li>- tanggal_expired, not null</li> <li>- status, enum ("aktif", "tidak aktif")</li> </ul>
6	label	<p><b>label</b> adalah <i>entity set</i> yang bertanggung jawab dalam merilis konten-konten di Apple Music.</p> <p>Atribut:</p> <ul style="list-style-type: none"> <li>- id: primary key, not null</li> <li>- nama: not null</li> <li>- jumlah_konten: derived, not null</li> <li>- total_durasi: derived, not null</li> <li>- tahun_berdiri: not null</li> <li>- asal_negara: not null</li> </ul>
7	lirik	<p><b>lirik</b> adalah <b>weak entity set</b> yang merepresentasikan lirik yang dimiliki oleh sebuah lagu.</p> <p>Atribut:</p> <ul style="list-style-type: none"> <li>- line: diskriminator, not null</li> <li>- text: not null</li> </ul>

Daftar *relationship set* yang direvisi beserta *entity sets* yang terhubung dan penjelasan atau justifikasinya dapat dilihat pada tabel 1.b.2.

Tabel 1.b.2. Daftar relationship set revisi beserta entity sets yang terhubung dan penjelasannya.

No	Relationship Set	Entity Sets yang Terhubung	Penjelasan
1	IS A	<ul style="list-style-type: none"> <li>- pengguna (higher level entity)</li> <li>- writer (lower level entity)</li> <li>- artis (lower level entity)</li> <li>- host (lower level entity)</li> </ul>	Relationship set IS A ini dihilangkan
2	daftar_langganan	<ul style="list-style-type: none"> <li>- pengguna</li> <li>- subscription</li> </ul>	Constraint dari <i>entity set</i> <b>pengguna</b> ke <i>weak entity</i>

			<p>set <b>subscription</b> berupa <b>many(total)-to-one(total)</b> dengan <i>many side</i> pada <i>weak entity</i> set <b>subscription</b> dan <i>one side</i> pada <i>entity</i> set <b>pengguna</b>. Ini menunjukkan bahwa yang termasuk pengguna dari apple music adalah pengguna yang memiliki daftar langganan terhadap <b>subscription</b>. <b>Total participation</b> pada kedua sisi menyatakan bahwa setiap pengguna apple music pasti memiliki langganan terhadap suatu <b>subscription</b> dan setiap <b>subscription</b> yang ada akan terikat pada suatu <b>pengguna</b>. <b>Constraint one side</b> menyatakan bahwa setiap <b>subscription</b> terikat pada satu pengguna dengan diskriminator <b>id</b> dari pengguna yang inheritance dari <b>apple_id</b> dan <b>constraint many side</b> menyatakan bahwa setiap pengguna dapat memiliki banyak <b>subscription</b>, tetapi yang memiliki status aktif hanya satu dari banyaknya <b>subscription</b> yang dimiliki <b>pengguna</b>.</p>
3	akses	<ul style="list-style-type: none"> <li>- pengguna</li> <li>- konten</li> </ul>	Relationship set akses ini dihilangkan
4	buat	<ul style="list-style-type: none"> <li>- pengguna</li> <li>- konten</li> </ul>	<p>Berdasarkan pernyataan, "... setiap konten diasumsikan hanya diproduksi oleh satu artis," membuat hubungan antara konten dengan <b>pengguna</b> <b>one(partial)-to-many(total)</b></p>

			<p>dari entity set <b>pengguna</b> ke <b>konten</b> yang berarti setiap pengguna dapat membuat maupun tidak sebuah konten di Apple Music. Setiap pengguna dapat membuat lebih dari satu konten, sebaliknya setiap konten pasti hanya memiliki satu pengguna.</p> <p>Relationship set ini beserta dengan Entity sets yang terhubung dengannya dijadikan satu dalam sebuah agregasi. Agregasi tersebut menggambarkan sebuah entity baru yaitu artis.</p>
5	memandu	<p>- pengguna - video_extra</p>	<p>Hubungan antara entity set <b>video_extra</b> dan <b>pengguna</b> memiliki <b>constraint many(total)-to-many(partial)</b> dari entity set <b>video_extra</b> ke <b>pengguna</b> yang berarti bahwa <b>pengguna</b> Apple Music dapat memandu banyak <b>video_extra</b> dan <b>video_extra</b> dapat dipandu banyak <b>pengguna</b>, <b>pengguna</b> yang memandu suatu <b>video_extra</b> diasumsikan sebagai seorang <b>host</b>. <b>Constraint total participation</b> pada entity set <b>video_extra</b> berarti bahwa setiap satu <b>video_extra</b> minimal harus dipandu oleh satu orang <b>host</b> dan <b>constraint partial participation</b> pada entity set <b>pengguna</b> berarti bahwa setiap <b>pengguna</b> dapat memandu banyak</p>

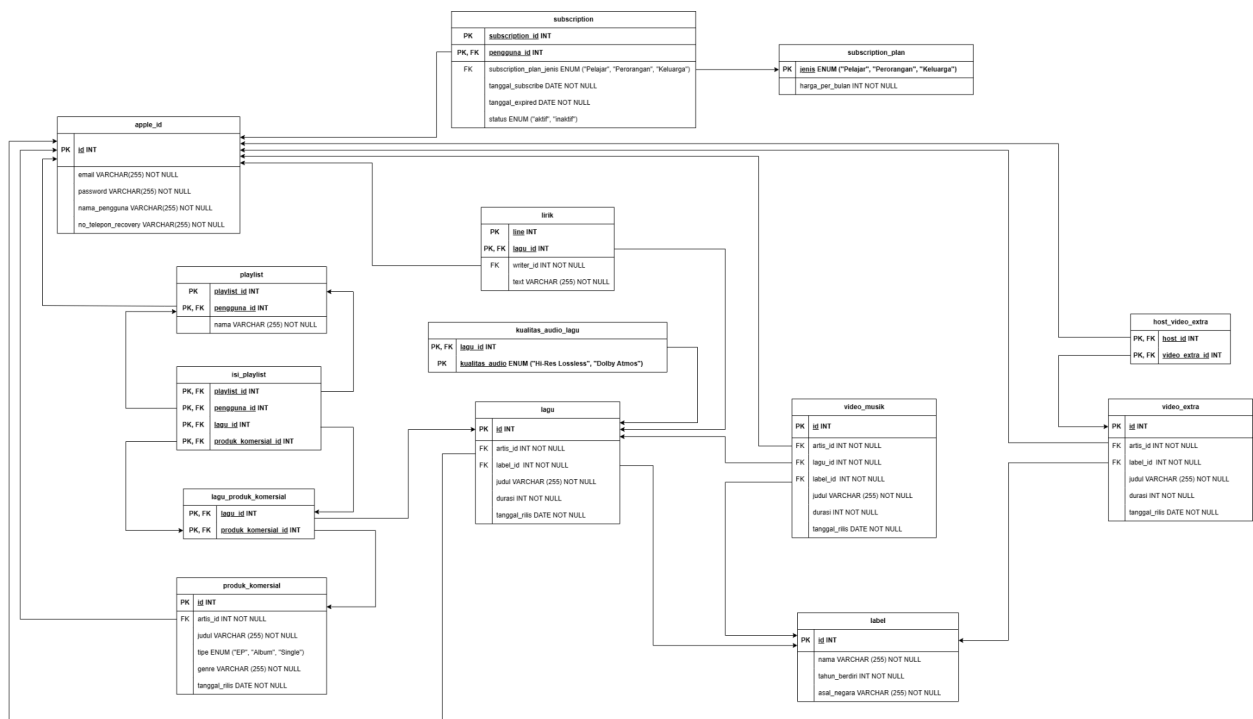
			<b>video_extra.</b>
6	fitur	- lagu - lirik	Constraint berupa <b>one(partial)-to-many(total)</b> dari <i>entity set lagu</i> ke <i>weak entity set lirik</i> menunjukkan bahwa sebuah <b>lagu</b> memungkinkan memiliki fitur berupa baris <b>lirik</b> . Sebuah lagu dapat memiliki lebih dari sebuah baris lirik ataupun tidak sama sekali. Sementara itu, sebuah <i>entity lirik</i> pasti digunakan atau menjadi fitur dari sebuah <b>lagu</b> .
7	tulis	- <b>pengguna</b> - lirik	Penetapan <i>constraint</i> <b>many(total)-to-one(partial)</b> dari <i>weak entity set lirik</i> ke <i>identifying entity set pengguna</i> disebabkan salah satunya oleh faktor penetapan <i>entity set lirik</i> sebagai <i>weak entity set</i> tersebut. Penetapan <i>constraint</i> tersebut menyatakan bahwa setiap <b>pengguna</b> dapat menulis banyak <b>lirik</b> dan setiap <b>lirik</b> pasti ditulis oleh seorang <b>pengguna</b> , <b>pengguna</b> yang menulis lirik diasumsikan sebagai seorang <b>writer</b> .
8	<b>artis</b>	- agregasi <b>pengguna dan konten</b> - produk komersial	Penetapan <i>constraint</i> <b>one(partial)-to-many(total)</b> pada relasi <b>artis</b> dengan one side pada agregasi pengguna dan konten dengan asumsi agregasi ini sebagai <b>artis</b> , dan many side pada <i>entity set produk_komersial</i> menyatakan bahwa setiap



			<p><b>pengguna</b> berupa <b>artis</b> dapat memiliki banyak <b>produk_komersial</b>, dan setiap <b>produk_komersial</b> yang ada pasti dimiliki oleh seorang <b>pengguna</b> yang dalam hal ini <b>artis</b>.</p>
--	--	--	--

## 2. MODEL RELASIONAL

### a. Diagram Relasional



[Gambar Relational Model](#)

### b. Deskripsi

Tabel 2.b.1. Daftar tabel relasional beserta atribut dan penjelasannya

NO	NAMA_TABEL	ATRIBUT	PENJELASAN
1	apple_id	<ul style="list-style-type: none"> <li><b>id</b> (PK) (INT)</li> <li>email (VARCHAR(255) NOT NULL)</li> <li>password (VARCHAR(255) NOT NULL)</li> <li>nama_pengguna (VARCHAR(255) NOT NULL)</li> </ul>	Tabel ini menyimpan semua data pengguna di apple_music.

		<ul style="list-style-type: none"> <li>no_telepon_recovery (VARCHAR(255) NOT NULL)</li> </ul>	
2	playlist	<ul style="list-style-type: none"> <li><u>playlist_id</u> (PK) (INT)</li> <li><u>pengguna_id</u> (PK,FK) (INT)</li> <li>nama (VARCHAR(255) NOT NULL)</li> </ul>	<p>Tabel ini menggambarkan playlist yang dibuat pengguna. Id play list ini dimulai dari 1 untuk setiap pengguna dan akan auto increment setelahnya.</p> <p>FK pengguna_id merujuk kepada id di table apple_id. Bersama dengan playlist_id, mereka menjadi PK yang menggambarkan misal playlist 1 pengguna 1 playlist 2 pengguna 1 dst.</p>
3	isi_playlist	<ul style="list-style-type: none"> <li><u>playlist_id</u> (PK,FK) (INT)</li> <li><u>pengguna_id</u> (PK,FK) (INT)</li> <li><u>lagu_id</u> (PK,FK)</li> <li><u>produk_komersial_id</u> (PK,FK)</li> </ul>	<p>Tabel ini menyimpan data isi playlist tertentu dari suatu pengguna.</p> <p>playlist_id merujuk pada playlist_id di tabel playlist, pengguna_id merujuk pada pengguna_id di tabel playlist, lagu_id merujuk pada lagu_id di tabel lagu, dan produk_komersial_id merujuk pada produk_komersial_id di tabel produk_komersial.</p> <p>Kombinasi keempatnya menjadi PK di tabel ini sebab lagu dengan id yang sama bisa berada di suatu playlist dua kali</p>

			apabila dipasarkan dengan produk komersial yang berbeda.
4	lagu_produk_komersial	<ul style="list-style-type: none"> <li>• <u>lagu_id</u> (PK,FK) (INT)</li> <li>• <u>produk_komersial_id</u> (PK,FK) (INT)</li> </ul>	<p>Tabel ini adalah penghubung antara tabel isi_playlist, lagu, dan produk_komersial.</p> <p>FK lagu_id merujuk ke atribut id di tabel lagu dan FK produk_komersial_id merujuk ke atribut id di tabel produk_komersial.</p>
5	produk_komersial	<ul style="list-style-type: none"> <li>• <u>id</u> (PK) (INT)</li> <li>• artis_id (FK) (INT NOT NULL)</li> <li>• judul (VARCHAR(255) NOT NULL)</li> <li>• tipe (ENUM ("EP", "Album", "Single"))</li> <li>• genre (VARCHAR (255) NOT NULL)</li> <li>• tanggal_rilis (DATE NOT NULL)</li> </ul>	<p>Tabel ini adalah tabel yang menyatakan data produk komersial yang ada di dalam basis data.</p> <p>FK artis_id merujuk ke atribut id di tabel apple_id yang menyatakan artis yang memiliki produk komersial yang bersangkutan.</p>
6	subscription	<ul style="list-style-type: none"> <li>• <u>subscription_id</u> (PK) (INT)</li> <li>• <u>pengguna_id</u> (PK,FK) (INT)</li> <li>• subscription_plan_jenis (FK) (ENUM ("Pelajar", "Perorangan", "Keluarga"))</li> <li>• tanggal_subscribe (DATE NOT NULL)</li> <li>• tanggal_expired (DATE NOT NULL)</li> <li>• status (ENUM ("aktif", "inaktif"))</li> </ul>	<p>Tabel ini adalah tabel yang menyimpan data subscription dari pengguna. Setiap pengguna dapat memiliki lebih dari satu data subscription</p> <p>FK pengguna_id merujuk ke atribut id di tabel apple_id yang menyatakan pengguna yang melakukan subscription dan FK</p>

			subscription_plan_jenis merujuk ke atribut jenis di tabel subscription_plan yang menyatakan jenis subscription dari data subscription yang bersangkutan.
7	lirik	<ul style="list-style-type: none"> <li>• <u>line</u> (PK) (INT)</li> <li>• <u>lagu_id</u> (PK,FK) (INT)</li> <li>• writer_id (FK) (INT NOT NULL)</li> <li>• text VARCHAR (255) NOT NULL</li> </ul>	<p>Tabel ini adalah tabel yang menyimpan data lirik semua lagu berdasarkan barisnya.</p> <p>FK lagu_id merujuk ke atribut id pada tabel lagu, dan FK writer_id merujuk ke atribut id di tabel apple_id.</p>
8	kualitas_audio_lagu	<ul style="list-style-type: none"> <li>• <u>lagu_id</u> (PK,FK) (INT)</li> <li>• kualitas_audio (PK) (ENUM ("Hi-Res Lossless", "Dolby Atmos"))</li> </ul>	<p>Tabel ini adalah tabel yang menyimpan kualitas audio dari suatu lagu.</p> <p>FK lagu_id merujuk ke atribut id pada tabel lagu.</p>
9	lagu	<ul style="list-style-type: none"> <li>• <u>id</u> (PK) (INT)</li> <li>• artis_id (FK) (INT NOT NULL)</li> <li>• label_id (FK) (INT NOT NULL)</li> <li>• judul (VARCHAR (255) NOT NULL)</li> <li>• durasi (INT NOT NULL)</li> <li>• tanggal_rilis (DATE NOT NULL)</li> </ul>	<p>Tabel ini adalah tabel yang menyimpan data semua lagu yang ada di dalam database.</p> <p>FK artis_id merujuk ke atribut id yang berada pada tabel apple_id (pengguna yang menjadi artis lagu tersebut), dan FK label_id merujuk ke atribut id pada tabel label.</p>
10	subscription_plan	<ul style="list-style-type: none"> <li>• <u>jenis</u> (PK) (ENUM ("Pelajar", "Perorangan", "Keluarga"))</li> <li>• harga_per_bulan (INT NOT NULL)</li> </ul>	Tabel ini adalah tabel yang merepresentasikan

			jenis subscription yang tersedia.
11	video_musik	<ul style="list-style-type: none"> <li>• <u>id</u> (PK) (INT)</li> <li>• artis_id (FK) (INT NOT NULL)</li> <li>• lagu_id (FK) (INT NOT NULL)</li> <li>• label_id (FK) (INT NOT NULL)</li> <li>• judul (VARCHAR (255) NOT NULL)</li> <li>• durasi (INT NOT NULL)</li> <li>• tanggal_rilis (DATE NOT NULL)</li> </ul>	<p>Tabel ini adalah tabel yang menyimpan data semua video musik yang ada di basis data.</p> <p>FK artis_id merujuk ke atribut id yang berada pada tabel artis (pengguna yang menjadi artis dari video musik tersebut), FK lagu_id merujuk ke atribut id pada tabel lagu (lagu yang dimasukkan ke video musik), dan FK label_id merujuk ke atribut id pada tabel label.</p>
12	label	<ul style="list-style-type: none"> <li>• <u>id</u> (PK) (INT)</li> <li>• nama (VARCHAR (255) NOT NULL)</li> <li>• tahun_berdiri (INT NOT NULL)</li> <li>• asal_negara (VARCHAR (255) NOT NULL)</li> </ul>	Tabel ini adalah tabel yang menyimpan data label yang terdaftar di basis data Apple Music.
13	host_video_extra	<ul style="list-style-type: none"> <li>• <u>host_id</u> (PK,FK) (INT)</li> <li>• <u>video_extra_id</u> (PK,FK) (INT)</li> </ul>	<p>Tabel ini adalah tabel yang menyimpan data pengguna yang menjadi host suatu video extra.</p> <p>FK host_id merujuk ke atribut id pada tabel apple_id (pengguna yang menjadi host) dan FK video_extra_id merujuk ke atribut id di tabel video_extra.</p>
14	video_extra	<ul style="list-style-type: none"> <li>• <u>id</u> (PK) (INT)</li> <li>• artis_id (FK) (INT NOT NULL)</li> <li>• label_id (FK) (INT NOT NULL)</li> <li>• judul (VARCHAR (255) NOT NULL)</li> </ul>	Tabel ini adalah tabel yang menyimpan data semua video extra yang ada di basis data.

		<ul style="list-style-type: none"> <li>• durasi (INT NOT NULL)</li> <li>• tanggal_rilis (DATE NOT NULL)</li> </ul>	FK artis_id merujuk ke atribut id di tabel apple_id (pengguna yang menjadi artis yang membintangi video extra) dan FK label_id merujuk ke atribut id di tabel label.
--	--	--	--

### 3. IMPLEMENTASI BASIS DATA

#### a. Proses

Seeding dilakukan dengan memanfaatkan library Faker.js. Pembuatan data terhadap masing-masing table memperhatikan constraints seperti tipe data tiap atributnya yang relevan, referential integrity, cardinality dan participation constraint, dan lain-lain sehingga data yang dibangkitkan adalah data relevan untuk setiap table. Untuk memastikan *constraints-constraint* yang ada, dilakukan secara semi manual dengan melakukan modifikasi pada seeding dengan faker sehingga tidak sepenuhnya “random” melainkan harus saling terhubung. Untuk memenuhi query-query yang dibuat agar lebih tergambar, Data yang dibangkitkan untuk masing-masing table ter-output pada file berformat CSV dan setiap file tersebut di-import ke dalam database. Untuk melihat kode melakukan *faker* dengan lebih jelas, link berikut akan membawa ke github pembuatan kode [https://github.com/ImmanuelSG/Tubes\\_Basdat.git](https://github.com/ImmanuelSG/Tubes_Basdat.git). Platform basis data yang digunakan adalah Mariadb versi 11.3.2.

#### b. Hasil Implementasi

Tujuan	
Query	SHOW TABLES  describe host_video_extra; describe video_musik; describe kualitas_audio_lagu;

	<div>describe lirik; describe subscription; describe isi_playlist; describe lagu_produk_komersial; describe playlist; describe video_extra; describe produk_komersial; describe lagu; describe label; describe subscription_plan; describe apple_id;</div>
Tangkapan Layar Query	

```
MariaDB [TubesBasdat]> show tables;
```

Tables_in_tubesbasdat
apple_id
host_video_extra
isi_playlist
kualitas_audio_lagu
label
lagu
lagu_produk_komersial
lirik
playlist
produk_komersial
subscription
subscription_plan
video_extra
video_musik

```
14 rows in set (0.002 sec)
```

```
MariaDB [tesload2]> describe apple_id;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
email	varchar(255)	NO		NULL	
password	varchar(255)	NO		NULL	
nama_pengguna	varchar(255)	NO		NULL	
no_telepon_recovery	varchar(255)	NO		NULL	

```
5 rows in set (0.020 sec)
```



```
MariaDB [tesload2]> describe subscription_plan;
```

Field	Type	Null	Key	Default	Extra
jenis	enum('Pelajar', 'Perorangan', 'Keluarga')	NO	PRI	NULL	
harga_per_bulan	int(11)	NO		NULL	

```
2 rows in set (0.018 sec)
```

```
MariaDB [tesload2]> describe label;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
nama	varchar(255)	NO		NULL	
tahun_berdiri	int(11)	NO		NULL	
asal_negara	varchar(255)	NO		NULL	

```
4 rows in set (0.019 sec)
```

```
MariaDB [tesload2]> describe lagu;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
artis_id	int(11)	NO	MUL	NULL	
label_id	int(11)	NO	MUL	NULL	
judul	varchar(255)	NO		NULL	
durasi	int(11)	NO		NULL	
tanggal_rilis	date	NO		NULL	

```
6 rows in set (0.022 sec)
```

```
MariaDB [tesload2]> describe produk_komersial;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
artis_id	int(11)	NO	MUL	NULL	
judul	varchar(255)	NO		NULL	
tipe	enum('EP', 'Album', 'Single')	YES		NULL	
genre	varchar(255)	NO		NULL	
tanggal_rilis	date	NO		NULL	

```
6 rows in set (0.018 sec)
```

```
MariaDB [tesload2]> describe video_extra;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
artis_id	int(11)	NO	MUL	NULL	
label_id	int(11)	NO	MUL	NULL	
judul	varchar(255)	NO		NULL	
durasi	int(11)	NO		NULL	
tanggal_rilis	date	NO		NULL	

```
6 rows in set (0.021 sec)
```

```
MariaDB [tesload2]> describe playlist;
```

Field	Type	Null	Key	Default	Extra
playlist_id	int(11)	NO	PRI	NULL	
pengguna_id	int(11)	NO	PRI	NULL	
nama	varchar(255)	NO		NULL	

```
3 rows in set (0.019 sec)
```

```
MariaDB [tesload2]> describe lagu_produk_komersial;
```

Field	Type	Null	Key	Default	Extra
lagu_id	int(11)	NO	PRI	NULL	
produk_komersial_id	int(11)	NO	PRI	NULL	

```
2 rows in set (0.021 sec)
```

```
MariaDB [tesload2]> describe isi_playlist;
```

Field	Type	Null	Key	Default	Extra
playlist_id	int(11)	NO	PRI	NULL	
pengguna_id	int(11)	NO	PRI	NULL	
lagu_id	int(11)	NO	PRI	NULL	
produk_komersial_id	int(11)	NO	PRI	NULL	

```
4 rows in set (0.018 sec)
```

```
MariaDB [tesload2]> describe subscription;
```

Field	Type	Null	Key	Default	Extra
subscription_id	int(11)	NO	PRI	NULL	
pengguna_id	int(11)	NO	PRI	NULL	
subscription_plan_jenis	enum('Pelajar', 'Perorangan', 'Keluarga')	YES	MUL	NULL	
tanggal_subscribe	date	NO		NULL	
tanggal_expired	date	NO		NULL	
status	enum('aktif', 'inaktif')	YES		NULL	

```
6 rows in set (0.018 sec)
```

```
MariaDB [tesload2]> describe lirik;
```

Field	Type	Null	Key	Default	Extra
line	int(11)	NO	PRI	NULL	
lagu_id	int(11)	NO	PRI	NULL	
writer_id	int(11)	NO	MUL	NULL	
text	varchar(255)	NO		NULL	

```
4 rows in set (0.026 sec)
```

MariaDB [tesload2]> describe kualitas_audio_lagu;						
Field	Type	Null	Key	Default	Extra	
lagu_id	int(11)	NO	PRI	NULL		
kualitas_audio	enum('Hi-Res Lossless','Dolby Atmos')	NO	PRI	NULL		

MariaDB [tesload2]> describe video_musik;						
Field	Type	Null	Key	Default	Extra	
id	int(11)	NO	PRI	NULL	auto_increment	
artis_id	int(11)	NO	MUL	NULL		
lagu_id	int(11)	NO	MUL	NULL		
label_id	int(11)	NO	MUL	NULL		
judul	varchar(255)	NO		NULL		
durasi	int(11)	NO		NULL		
tanggal_rilis	date	NO		NULL		

7 rows in set (0.018 sec)

MariaDB [tesload2]> describe host_video_extra;						
Field	Type	Null	Key	Default	Extra	
host_id	int(11)	NO	PRI	NULL		
video_extra_id	int(11)	NO	PRI	NULL		

2 rows in set (0.024 sec)

#### 4. PENGAMBILAN INSIGHT BASIS DATA

##### a. Query sederhana

Tujuan	<p>Menampilkan daftar pengguna Apple Music yang impactful terhadap Apple Music, yakni aktif berpartisipasi untuk setiap kesempatan yang tersedia pada Apple Music, ditandai dengan dirinya ikut serta menjadi artis, writer, dan/ atau host.</p> <p>Akan ditampilkan nama pengguna, is_artis, is_writer, is_host, dan total role tambahannya (artis, <i>writer</i>, dan/ atau <i>host</i>) yang total role tambahan tersebut minimal berjumlah 1 dan terurut mengecil berdasarkan total role tambahan tersebut.</p>
Query (Melibatkan setidaknya 3 relasi)	<pre>SELECT     aid.nama_pengguna,     CASE         WHEN a.artis_id IS NOT NULL THEN 1         ELSE 0</pre>

```

END AS is_artis,
CASE
    WHEN l.writer_id IS NOT NULL THEN 1
    ELSE 0
END AS is_writer,
CASE
    WHEN h.host_id IS NOT NULL THEN 1
    ELSE 0
END AS is_host,
((a.artis_id IS NOT NULL) + (l.writer_id IS NOT
NULL) + (h.host_id IS NOT NULL)) AS
total_extra_roles
FROM
apple_id AS aid
INNER JOIN (
    SELECT DISTINCT
        pengguna_id
    FROM
        subscription
) AS u
    ON (aid.id = u.pengguna_id)
LEFT OUTER JOIN (
    (
        SELECT DISTINCT
            artis_id
        FROM
            lagu
    )
    UNION
    (
        SELECT DISTINCT
            artis_id
        FROM
            video_extra
    )
) AS a
    ON (aid.id = a.artis_id)
LEFT OUTER JOIN (
    SELECT DISTINCT
        writer_id
    FROM
        lirik
) AS l
    ON (aid.id = l.writer_id)
LEFT OUTER JOIN (
    SELECT DISTINCT

```

	<pre>        host_id       FROM         host_video_extra       ) AS h       ON (aid.id = h.host_id) WHERE   a.artis_id IS NOT NULL   OR l.writer_id IS NOT NULL   OR h.host_id IS NOT NULL ORDER BY   total_extra_roles DESC,   aid.nama_pengguna ASC;</pre>
Tangkapan Layar Query	

```

MariaDB [TubesBasdat]> SELECT
-> aid.nama_pengguna,
-> CASE
-> WHEN a.artis_id IS NOT NULL THEN 1
-> ELSE 0
-> END AS is_artis,
-> CASE
-> WHEN l.writer_id IS NOT NULL THEN 1
-> ELSE 0
-> END AS is_writer,
-> CASE
-> WHEN h.host_id IS NOT NULL THEN 1
-> ELSE 0
-> END AS is_host,
-> ((a.artis_id IS NOT NULL) + (l.writer_id IS NOT NULL) + (h.host_id IS NOT NULL)) AS total_extra_ro
-> FROM
-> apple_id AS aid
-> INNER JOIN (
-> SELECT DISTINCT
-> pengguna_id
-> FROM
-> subscription
-> ) AS u
-> ON (aid.id = u.pengguna_id)
-> LEFT OUTER JOIN (
-> (
-> SELECT DISTINCT
-> artis_id
-> FROM
-> lagu
-> )
-> UNION
-> (
-> SELECT DISTINCT
-> artis_id
-> FROM
-> video_extra
-> )
-> ) AS a
-> ON (aid.id = a.artis_id)
-> LEFT OUTER JOIN (
-> SELECT DISTINCT
-> writer_id
-> FROM
-> lirik
-> ) AS l
-> ON (aid.id = l.writer_id)
-> LEFT OUTER JOIN (
-> SELECT DISTINCT
-> host_id
-> FROM
-> host_video_extra
-> ) AS h
-> ON (aid.id = h.host_id)
-> WHERE
-> a.artis_id IS NOT NULL
-> OR l.writer_id IS NOT NULL
-> OR h.host_id IS NOT NULL
-> ORDER BY
-> total_extra_roles DESC,
-> aid.nama_pengguna ASC;

```

nama_pengguna	is_artis	is_writer	is_host	total_extra_roles
Ajeng Salsabilla	1	1	1	3
Candrakanta Candrakanta Pradipta	1	1	1	3
Diah Diah Kusuma	1	1	1	3
Drajat Lazuardi	1	1	1	3
Irawan Raihan	1	1	1	3
Syahreza Ivan	1	1	1	3
Vera Yolanda	1	1	1	3
Yono Nugroho	1	1	1	3
Adhitama Caturangga	1	0	1	2
Anandya Elisa	1	0	1	2
Anandya Maya	1	1	0	2
Aryani Kani	1	0	1	2
Aryanti Janet	1	0	1	2
Atma Anggriawan	1	0	1	2
Aulia Aulia Anandya	1	0	1	2
Bakijan Marpaung	1	0	1	2
Cindy Nasyiah	1	1	0	2
Danang Mansur	1	1	0	2
Dodo Tampubolon	1	1	0	2
Ellis Pertiwi	1	0	1	2
Emong Samosir	1	0	1	2
Galiono Marpaung	1	0	1	2
Galuh Irawan	1	0	1	2
Gantar Gantar Iswahyudi	1	0	1	2
Ghani Ghani Adriansyah	1	1	0	2
Gilang Gilang Prasasta	1	0	1	2
Hani Hani Fernanda	1	0	1	2

Gitung Gitang Prasasta	1	0	1	2
Hani Hani Fernanda	1	0	1	2
Haryanto Jarwi	1	1	0	2
Hutagalung Raden	1	0	1	2
Intan Intan Kusuma	1	0	1	2
Ira Uyainah	1	0	1	2
Kahyang Septi	1	1	0	2
Kanda Kanda Mahendra	1	0	1	2
Karja Prasetyo	1	0	1	2
Kayla Kayla Aryani	1	0	1	2
Lamar Lamar Handoko	1	0	1	2
Maya Maya Kusmawati	1	0	1	2
Mustika Dinda	1	0	1	2
Nabila Nabila Winarsih	1	0	1	2
Narpati Opung	1	0	1	2
Nasyiah Zizi	1	0	1	2
Naura Naura Fernanda	1	0	1	2
Naura Naura Winarsih	1	0	1	2
Nugraha Akarsana	1	0	1	2
Paramita Paramita Yuniar	1	0	1	2
Purwanto Anggriawan	1	0	1	2
Rachel Rachel Lailasari	1	0	1	2
Rahman Purnama	1	0	1	2
Ramadhani Karen	1	0	1	2
Rosa Oktaviani	1	0	1	2
Safina Safina Usamah	1	0	1	2
Safira Safira Wijayanti	1	0	1	2
Safitri Mutia	1	0	1	2
Salma Rita	1	0	1	2
Shania Melani	1	0	1	2
Sinaga Dartono	0	1	1	2
Sintia Aryani	1	0	1	2
Suryatmi Febi	1	0	1	2
Talia Mardhiyah	1	0	1	2
Tami Tami Puspita	1	1	0	2
Utama Nainggolan	1	0	1	2
Uwais Mustika	1	0	1	2
Uyainah Dina	1	0	1	2
Wastuti Chelsea	1	0	1	2
Wicaksono Maryanto	1	0	1	2
Wijaya Karman	1	0	1	2
Winarsih Aurora	1	0	1	2
Yolanda Faizah	1	0	1	2
Abyasa Arifin	1	0	0	1
Ade Astuti	1	0	0	1
Balapati Abiputra	1	0	0	1
Bella Saputri	1	0	0	1
Caket Caket Saputra	1	0	0	1
Chandra Chandra Budiyo	1	0	0	1
Dina Dina Palastru	1	0	0	1
Dodo Hidayanto	0	0	1	1
Gamblang Gamblang Waskita	1	0	0	1
Gunarto Satya	1	0	0	1
Irawan Harimurti	1	0	0	1
Janet Janet Mustika	1	0	0	1
Jefri Syahreza	1	0	0	1
Laksita Nur	1	0	0	1
Manullang Bancar	1	0	0	1
Marbun Harja	0	0	1	1
Megantara Prasetyo	1	0	0	1
Murti Murti Rajasa	1	0	0	1
Nardi Widodo	1	0	0	1
Palastru Rita	1	0	0	1
Pangestu Jagapati	1	0	0	1
Ratih Ratih Aqila	1	0	0	1
Rini Zulaika	0	0	1	1
Salimah Salimah Mahestri	1	0	0	1
Sherly Sherly Yulianti	0	0	1	1
Suryatmi Pia	1	0	0	1
Vera Yuniar	0	0	1	1
Wasita Artanto	1	0	0	1
Yahya Pranowo	1	0	0	1

97 rows in set (0.003 sec)

b. Query dengan *set operation*

<p>Tujuan</p>	<p>Menampilkan daftar label yang track-nya ambisius terhadap setiap pasar konten, ditandai dengan label tersebut pernah memproduksi setiap tipe konten. Sayangnya, label-label tersebut tampak sedang "down", ditandai dengan label tersebut pada tahun 2024 tidak melakukan pemasaran terhadap lagu apa pun yang diproduserinya. Diasumsikan bahwa setiap produk komersial turut terkait dengan label berhubung setiap lagu pasti dipasarkan dalam bentuk produk komersial.</p> <p>Akan ditampilkan daftar nama, tahun berdiri, dan asal negara label yang pernah memproduksi setiap tipe konten, namun seluruh lagu yang diproduserinya tidak ada yang dipasarkan (dalam bentuk produk komersial) pada tahun 2024.</p>
<p>Query</p> <p>(Melibatkan setidaknya 3 relasi pada setiap klausa)</p>	<pre> SELECT     l.nama,     l.tahun_berdiri,     l.asal_negara FROM     label AS l     INNER JOIN (         (             SELECT DISTINCT                 l.label_id             FROM                 lagu AS l                 INNER JOIN video_musik AS                     vm                     ON (l.label_id =                         vm.label_id)                 INNER JOIN video_extra AS                     ve                     ON (l.label_id =                         ve.label_id)         )         EXCEPT         (             SELECT DISTINCT                 l.label_id             FROM                 lagu AS l                 INNER JOIN </pre>



	<pre>        lagu_produk_komersial AS lpk         ON (l.id = lpk.lagu_id)     INNER JOIN produk_komersial     AS pk         ON         (lpk.produk_komersial_id = pk.id)     WHERE         YEAR(pk.tanggal_rilis) = 2024     )     ) AS lid     ON (l.id = lid.label_id) ORDER BY     l.nama ASC;</pre>
Tangkapan Layar Query	

```

MariaDB [TubesBasdat]> SELECT
-> l.nama,
-> l.tahun_berdiri,
-> l.asal_negara
-> FROM
-> label AS l
-> INNER JOIN (
-> (
-> SELECT DISTINCT
-> l.label_id
-> FROM
-> lagu AS l
-> INNER JOIN video_musik AS vm
-> ON (l.label_id = vm.label_id)
-> INNER JOIN video_extra AS ve
-> ON (l.label_id = ve.label_id)
-> )
-> EXCEPT
-> (
-> SELECT DISTINCT
-> l.label_id
-> FROM
-> lagu AS l
-> INNER JOIN lagu_produk_komersial AS lpk
-> ON (l.id = lpk.lagu_id)
-> INNER JOIN produk_komersial AS pk
-> ON (lpk.produk_komersial_id = pk.id)
-> WHERE
-> YEAR(pk.tanggal_rilis) = 2024
-> )
-> ) AS lid
-> ON (l.id = lid.label_id)
-> ORDER BY
-> l.nama ASC;

```

nama	tahun_berdiri	asal_negara
Abshire Tbk	2016	Falkland Islands (Malvinas)
Champlin (Persero) Tbk	2019	Guyana
CV Bogisich	2017	Norway
CV Cronin	2017	Lebanon
CV Goldner (Persero) Tbk	2014	Jamaica
CV Rohan Tbk	2011	Uganda
Dooley Tbk	2013	Saint Barthelemy
Durgan (Persero) Tbk	2012	Mayotte
Fa Becker Tbk	2011	United States of America
Fa Feest Tbk	2016	Gabon
Fa Little	2019	Bahrain
Fa Walter Tbk	2015	Tajikistan
Fritsch (Persero) Tbk	2018	Sudan
Gleichner Tbk	2019	Malaysia
Hoppe (Persero) Tbk	2014	Macao
KKB Lakin	2011	Cote d'Ivoire
Klein (Persero) Tbk	2017	Rwanda
Leffler Tbk	2018	Benin
Little (Persero) Tbk	2018	Uruguay
PD Keeling (Persero) Tbk	2015	Cote d'Ivoire
Perum Ankunding Tbk	2019	Guyana
Perum Denesik	2017	United Kingdom
Perum Kerluke	2014	Japan
PT Franecki (Persero) Tbk	2018	Moldova
PT Littel Tbk	2019	Venezuela
PT Oberbrunner (Persero) Tbk	2014	Paraguay
PT Wolff Tbk	2017	Guernsey
Swaniawski (Persero) Tbk	2013	Sint Maarten
Tillman (Persero) Tbk	2011	Curacao
Turcotte Tbk	2016	Samoa
UD Huels	2011	El Salvador
UD Orn	2015	Qatar
UD Rolfson	2010	Mauritius
UD Stanton	2013	Guadeloupe
UD Thiel (Persero) Tbk	2017	Bouvet Island
Williamson Tbk	2011	Cameroon
Yayasan Murazik	2016	Svalbard & Jan Mayen Islands

37 rows in set (0.002 sec)

c. Query dengan agregasi

<p>Tujuan</p>	<p>Menampilkan daftar lagu dan artis pembuatnya yang lagu tersebut tampak dirilis secara "low budget", yakni tidak memiliki fitur lirik, tidak dibuatkan video musik, dan tidak memiliki kualitas audio apa pun. Walaupun begitu, lagu-lagu tersebut tampak laku di pasar, ditandai dengan lagu tersebut dimasukkan ke dalam playlist pengguna. Oleh karena itu, akan dilihat lagu-lagu "low budget" tersebut yang paling laku di pasaran.</p> <p>Akan ditampilkan judul, durasi, dan tanggal rilis lagu, serta nama artis lagu tersebut yang lagunya tidak memiliki fitur lirik, tidak dibuatkan video musik, dan tidak memiliki kualitas audio apa pun, namun dimasukkan sebagai playlist pengguna. Terurut mengecil berdasarkan jumlah lagu dimasukkan ke dalam playlist.</p>
<p>Query</p> <p>(Melibatkan setidaknya 4 relasi dengan <i>group by</i> dan <i>having</i>)</p>	<pre> SELECT     aid.nama_pengguna,     l.judul,     l.durasi,     l.tanggal_rilis,     COUNT(ip.playlist_id) AS total_in_playlist FROM     apple_id AS aid     INNER JOIN lagu AS l         ON (aid.id = l.artis_id)     INNER JOIN isi_playlist AS ip         ON (l.id = ip.lagu_id) WHERE     l.id NOT IN (         (             SELECT DISTINCT                 lagu_id             FROM                 lirik         )         UNION         (             SELECT DISTINCT                 lagu_id             FROM                 video_musik         )     ) </pre>

	<pre>UNION (     SELECT DISTINCT         lagu_id     FROM         kualitas_audio_lagu ) GROUP BY     l.id HAVING     total_in_playlist &gt; 0 ORDER BY     total_in_playlist DESC,     aid.nama_pengguna ASC;</pre>
Tangkapan Layar Query	

```

MariaDB [TubesBasdat]> SELECT
-> aid.nama_pengguna,
-> l.judul,
-> l.durasi,
-> l.tanggal_rilis,
-> COUNT(ip.playlist_id) AS total_in_playlist
-> FROM
-> apple_id AS aid
-> INNER JOIN lagu AS l
-> ON (aid.id = l.artis_id)
-> INNER JOIN isi_playlist AS ip
-> ON (l.id = ip.lagu_id)
-> WHERE
-> l.id NOT IN (
-> (
-> SELECT DISTINCT
-> lagu_id
-> FROM
-> lirik
-> )
-> UNION
-> (
-> SELECT DISTINCT
-> lagu_id
-> FROM
-> video_musik
-> )
-> UNION
-> (
-> SELECT DISTINCT
-> lagu_id
-> FROM
-> kualitas_audio_lagu
-> )
-> )
-> GROUP BY
-> l.id
-> HAVING
-> total_in_playlist > 0
-> ORDER BY
-> total_in_playlist DESC,
-> aid.nama_pengguna ASC;

```

nama_pengguna	judul	durasi	tanggal_rilis	total_in_playlist
Ade Astuti	On the Atchison	406	2021-04-26	1
Anandya Elisa	Dancing in the Street	349	2022-04-03	1
Aryani Mani	Twist & Shout	598	2022-05-10	1
Balapati Abiputra	One of These Nights	344	2021-12-18	1
Candrakanta Candrakanta Pradipta	Walk Like a Man	140	2021-11-10	1
Dodo Tampubolon	Bad Romance	375	2020-08-08	1
Drajat Lazuardi	Another One Bites the Dust	424	2021-04-13	1
Emong Samosir	Will It Go Round In Circles	164	2022-08-19	1
Galiono Marpaung	Walk This Way	512	2021-08-19	1
Galuh Irawan	Whole Lotta Love	576	2021-06-07	1
Gantar Gantar Iswahyudi	White Rabbit	504	2022-10-05	1
Gilang Gilang Prasasta	Escape (The Pina Colada Song)	133	2021-01-11	1
Hani Hani Fernanda	Under the Boardwalk	211	2022-01-14	1
Haryanto Jarwi	Music	329	2022-05-29	1
Irawan Raihan	Careless Whisper	480	2021-10-04	1
Irawan Raihan	Mood Indigo	493	2021-10-04	1
Laksita Nur	Manana (Is Soon Enough For Me)	504	2023-01-08	1
Manullang Bancar	Soldier Boy	451	2022-07-01	1
Manullang Bancar	How Much is That Doggy in the Window?	247	2022-07-01	1
Megantara Prasetyo	I Kissed A Girl	535	2020-01-26	1
Mustika Dinda	Walk Like an Egyptian	334	2021-05-15	1
Nabila Nabila Winarsih	Superstition	216	2022-07-30	1
Nabila Nabila Winarsih	The Letter	552	2022-07-30	1
Naura Naura Winarsih	ABC	313	2022-01-06	1
Nugraha Akarsana	I Shot the Sheriff	306	2021-10-29	1

Palastri Rita	Gypsies	332	2022-02-13	1
Pangestu Jagapati	Goodnight	433	2022-07-14	1
Rahman Purnama	I Feel Love	477	2020-07-15	1
Rahman Purnama	He's a Rebel	469	2020-07-15	1
Ramadhani Karen	A Thousand Miles	365	2021-01-09	1
Safira Safira Wijayanti	Higher Love	425	2021-03-29	1
Salimah Salimah Mahestri	Sweet Georgia Brown	368	2021-11-16	1
Salma Rita	You're Still the One	412	2020-12-29	1
Shania Melani	Cry	546	2021-09-17	1
Sintia Aryani	Stay (I Missed You)	280	2020-03-20	1
Sintia Aryani	Unchained Melody	420	2020-03-20	1
Suryatmi Febi	Apologize	128	2021-10-12	1
Suryatmi Pia	I Can't Go For That (No Can Do)	360	2022-02-02	1
Talia Mardhiyah	Careless Whisper	194	2022-02-04	1
Talia Mardhiyah	St Louis Blues	153	2022-02-04	1
Talia Mardhiyah	Papa Was a Rolling Stone	573	2022-02-04	1
Tami Tami Puspita	Faith	552	2020-03-16	1
Tami Tami Puspita	This Land is Your Land	278	2020-03-16	1
Utama Nainggolan	The Love You Save	260	2020-03-31	1
Uyainah Dina	On My Own	478	2021-01-01	1
Uyainah Dina	My Blue Heaven	593	2021-01-01	1
Vera Yolanda	Hard to Say I'm Sorry	314	2022-10-01	1
Yahya Pranowo	Touch Me	245	2021-05-10	1
Yahya Pranowo	There goes my baby	387	2021-05-10	1
Yolanda Faizah	(I Can't Get No) Satisfaction	262	2021-06-25	1

50 rows in set (0.005 sec)

d. Query dengan *subquery*

<p>Tujuan</p>	<p>Menampilkan pengguna Apple Music yang relatif masih muda, ditandai dengan status subscription aktifnya adalah subscription berjenis "Pelajar", namun dirinya dinilai sangat potensial bagi Apple Music. Hal tersebut turut ditandai dengan, pada seusianya, dirinya sudah menjadi artis, writer, dan host pada Apple Music.</p> <p>Akan ditampilkan nama pengguna yang <i>subscription</i> aktifnya berjenis "Pelajar" dan turut termasuk <i>writer</i>, artis, dan juga <i>host</i>.</p>
<p>Query</p> <p>(Melibatkan setidaknya 3 relasi pada subquery dan 2 relasi pada query utama)</p>	<pre> SELECT     aid.nama_pengguna FROM     apple_id AS aid     INNER JOIN (         SELECT DISTINCT             pengguna_id         FROM             subscription         WHERE             subscription_plan_jenis = "Pelajar"             AND status = "aktif"         ) AS s     ON (aid.id = s.pengguna_id)     INNER JOIN (         (             SELECT DISTINCT                 writer_id AS pengguna_id             FROM                 lirik         )         INTERSECT         (             (                 SELECT DISTINCT                     artis_id AS                         pengguna_id                 FROM                     lagu             )             UNION             (                 SELECT DISTINCT </pre>

	<pre>        artis_id AS         pengguna_id       FROM         video_extra     )   )   INTERSECT   (     SELECT DISTINCT       host_id AS pengguna_id     FROM       host_video_extra   )   ) AS uid   ON (aid.id = uid.pengguna_id) ORDER BY   aid.nama_pengguna ASC;</pre>
Tangkapan Layar Query	

```

MariaDB [TubesBasdat]> SELECT
  -> aid.nama_pengguna
  -> FROM
  -> apple_id AS aid
  -> INNER JOIN (
  -> SELECT DISTINCT
  -> pengguna_id
  -> FROM
  -> subscription
  -> WHERE
  -> subscription_plan_jenis = "Pelajar"
  -> AND status = "aktif"
  -> ) AS s
  -> ON (aid.id = s.pengguna_id)
  -> INNER JOIN (
  -> (
  -> SELECT DISTINCT
  -> writer_id AS pengguna_id
  -> FROM
  -> lirik
  -> )
  -> INTERSECT
  -> (
  -> (
  -> SELECT DISTINCT
  -> artis_id AS pengguna_id
  -> FROM
  -> lagu
  -> )
  -> UNION
  -> (
  -> SELECT DISTINCT
  -> artis_id AS pengguna_id
  -> FROM
  -> video_extra
  -> )
  -> )
  -> INTERSECT
  -> (
  -> SELECT DISTINCT
  -> host_id AS pengguna_id
  -> FROM
  -> host_video_extra
  -> )
  -> ) AS uid
  -> ON (aid.id = uid.pengguna_id)
  -> ORDER BY
  -> aid.nama_pengguna ASC;
+-----+
| nama_pengguna |
+-----+
| Yono Nugroho  |
+-----+
1 row in set (0.008 sec)

```



e. Query kompleks

<p>Tujuan</p>	<p>Menampilkan daftar artis yang relatif sangat ambisius dan multitalenta, ditandai dengan dirinya telah membuat konten yang totalnya lebih dari atau sama dengan rata-rata total konten setiap artis. Tak hanya itu, artis yang termasuk juga harus pernah membuat lirik terhadap lagunya sendiri dan pernah menjadi host. Bahkan lebih dari itu, artis tersebut harus memiliki selera pasar yang relatif bagus, ditandai dengan lagu-lagunya laku di pasaran, yakni total lagunya yang dijadikan playlist harus lebih dari atau sama dengan rata-rata total lagu setiap artis yang dimasukkan playlist pengguna Apple Music.</p> <p>Akan ditampilkan nama artis, total lagunya, total video musiknya, total <i>video extra</i>-nya, total kontennya, dan total lagunya yang dimasukkan playlist, dengan total konten yang pernah dibuatnya lebih dari atau sama dengan rata-rata total konten seluruh artis. Artis tersebut juga harus pernah membuat lirik terhadap lagunya sendiri dan pernah menjadi <i>host</i>. Tak hanya itu, jumlah lagunya (pasangan lagu dan produk komersialnya) yang dijadikan playlist oleh orang-orang juga harus lebih banyak atau sama dengan rata-rata total lagu (pasangan lagu dan produk komersialnya) setiap artis yang dimasukkana playlist.</p>
<p>Query</p> <p>(Melibatkan setidaknya kombinasi dua dari b/c/d. Ketentuan setiap komponen tetap harus terpenuhi)</p> <p>(c dan d)</p>	<pre> SELECT     aid.nama_pengguna,     IFNULL(la.total_lagu, 0) AS total_lagu,     IFNULL(vm.total_video_musik, 0) AS     total_video_musik,     IFNULL(ve.total_video_extra, 0) AS     total_video_extra,     (IFNULL(la.total_lagu, 0) +     IFNULL(vm.total_video_musik, 0) +     IFNULL(ve.total_video_extra, 0)) AS total_konten,     COUNT(ip.playlist_id) AS     total_lagu_dimasukkan_playlist FROM     (         SELECT             aid.id,             aid.nama_pengguna         FROM             apple_id AS aid     ) </pre>

```

INNER JOIN (
    SELECT DISTINCT
        li.writer_id
    FROM
        lirik AS li
        INNER JOIN lagu AS la
            ON (li.lagu_id =
                la.id)
    WHERE
        li.writer_id = la.artis_id
) AS w
    ON (aid.id = w.writer_id)
INNER JOIN (
    SELECT DISTINCT
        host_id
    FROM
        host_video_extra
) AS h
    ON (aid.id = h.host_id)
) AS aid
LEFT OUTER JOIN (
    SELECT
        artis_id,
        COUNT(id) AS total_lagu
    FROM
        lagu
    GROUP BY
        artis_id
) AS la
    ON (aid.id = la.artis_id)
LEFT OUTER JOIN (
    SELECT
        artis_id,
        COUNT(id) AS total_video_musik
    FROM
        video_musik
    GROUP BY
        artis_id
) AS vm
    ON (aid.id = vm.artis_id)
LEFT OUTER JOIN (
    SELECT
        artis_id,
        COUNT(id) AS total_video_extra
    FROM
        video_extra

```

```

GROUP BY
    artis_id
) AS ve
ON (aid.id = ve.artis_id)
INNER JOIN (
    SELECT
        artis_id,
        id
    FROM
        lagu
) AS l
ON (aid.id = l.artis_id)
INNER JOIN (
    SELECT
        lagu_id,
        produk_komersial_id,
        playlist_id
    FROM
        isi_playlist
) AS ip
ON (l.id = ip.lagu_id)
WHERE
(IFNULL(la.total_lagu, 0) +
IFNULL(vm.total_video_musik, 0) +
IFNULL(ve.total_video_extra, 0)) >= (
    SELECT
        AVG (IFNULL(l.total_lagu, 0) +
            IFNULL(vm.total_video_musik, 0) +
            IFNULL(ve.total_video_extra, 0))
    FROM
        (
            SELECT
                id
            FROM
                apple_id
        ) AS aid
    LEFT OUTER JOIN (
        SELECT
            artis_id,
            COUNT(id) AS
            total_lagu
        FROM
            lagu
        GROUP BY
            artis_id
    ) AS l

```

```

        ON (aid.id = l.artis_id)
LEFT OUTER JOIN (
    SELECT
        artis_id,
        COUNT(id) AS
        total_video_musik
    FROM
        video_musik
    GROUP BY
        artis_id
) AS vm
    ON (aid.id = vm.artis_id)
LEFT OUTER JOIN (
    SELECT
        artis_id,
        COUNT(id) AS
        total_video_extra
    FROM
        video_extra
    GROUP BY
        artis_id
) AS ve
    ON (aid.id = ve.artis_id)
WHERE
    l.artis_id IS NOT NULL
    OR vm.artis_id IS NOT NULL
    OR ve.artis_id IS NOT NULL
)
GROUP BY
    aid.id
HAVING
    total_lagu_dimasukkan_playlist >= (
        SELECT
            AVG(p.total_lagu_dimasukkan_playlis
            t)
        FROM (
            SELECT
                l.artis_id,
                COUNT(ip.playlist_id) AS
                total_lagu_dimasukkan_playlist
            FROM
                isi_playlist AS ip
                INNER JOIN lagu AS l
                    ON (ip.lagu_id = l.id)
            GROUP BY
                l.artis_id
        )
    )

```

```

) AS p
)
ORDER BY
total_lagu_dimasukkan_playlist DESC,
total_konten DESC,
aid.nama_pengguna ASC;

```

## Tangkapan Layar Query

```

MariaDB [Tubes8asdat]> SELECT
-> aid.nama_pengguna,
-> IFNULL(la.total_lagu, 0) AS total_lagu,
-> IFNULL(vm.total_video_musik, 0) AS total_video_musik,
-> IFNULL(ve.total_video_extra, 0) AS total_video_extra,
-> (IFNULL(la.total_lagu, 0) + IFNULL(vm.total_video_musik, 0) + IFNULL(ve.total_video_extra, 0)) AS total_konten,
-> COUNT(ip.playlist_id) AS total_lagu_dimasukkan_playlist
-> FROM
-> (
-> SELECT
-> aid.id,
-> aid.nama_pengguna
-> FROM
-> apple_id AS aid
-> INNER JOIN (
-> SELECT DISTINCT
-> li.writer_id
-> FROM
-> lirik AS li
-> INNER JOIN lagu AS la
-> ON (li.lagu_id = la.id)
-> WHERE
-> li.writer_id = la.artis_id
-> ) AS w
-> ON (aid.id = w.writer_id)
-> INNER JOIN (
-> SELECT DISTINCT
-> host_id
-> FROM
-> host_video_extra
-> ) AS h
-> ON (aid.id = h.host_id)
-> ) AS aid
-> LEFT OUTER JOIN (
-> SELECT
-> artis_id,
-> COUNT(id) AS total_lagu
-> FROM
-> lagu
-> GROUP BY
-> artis_id
-> ) AS la
-> ON (aid.id = la.artis_id)
-> LEFT OUTER JOIN (
-> SELECT
-> artis_id,
-> COUNT(id) AS total_video_musik
-> FROM
-> video_musik
-> GROUP BY
-> artis_id
-> ) AS vm
-> ON (aid.id = vm.artis_id)
-> LEFT OUTER JOIN (
-> SELECT
-> artis_id,
-> COUNT(id) AS total_video_extra
-> FROM
-> video_extra
-> GROUP BY
-> artis_id
-> ) AS ve
-> ON (aid.id = ve.artis_id)
-> INNER JOIN (
-> SELECT
-> artis_id,
-> id
-> FROM
-> lagu
-> ) AS l
-> ON (aid.id = l.artis_id)
-> INNER JOIN (
-> SELECT
-> lagu_id,
-> produk_komersial_id,
-> playlist_id
-> FROM
-> isi_playlist
-> ) AS ip
-> ON (l.id = ip.lagu_id)
-> WHERE
-> (IFNULL(la.total_lagu, 0) + IFNULL(vm.total_video_musik, 0) + IFNULL(ve.total_video_extra, 0)) >= (
-> SELECT
-> AVG (IFNULL(l.total_lagu, 0) + IFNULL(vm.total_video_musik, 0) + IFNULL(ve.total_video_extra, 0))

```

```

-> FROM
-> (
-> SELECT
-> id
-> FROM
-> apple_id
-> ) AS aid
-> LEFT OUTER JOIN (
-> SELECT
-> artis_id,
-> COUNT(id) AS total_lagu
-> FROM
-> lagu
-> GROUP BY
-> artis_id
-> ) AS l
-> ON (aid.id = l.artis_id)
-> LEFT OUTER JOIN (
-> SELECT
-> artis_id,
-> COUNT(id) AS total_video_musik
-> FROM
-> video_musik
-> GROUP BY
-> artis_id
-> ) AS vm
-> ON (aid.id = vm.artis_id)
-> LEFT OUTER JOIN (
-> SELECT
-> artis_id,
-> COUNT(id) AS total_video_extra
-> FROM
-> video_extra
-> GROUP BY
-> artis_id
-> ) AS ve
-> ON (aid.id = ve.artis_id)
-> WHERE
-> l.artis_id IS NOT NULL
-> OR vm.artis_id IS NOT NULL
-> OR ve.artis_id IS NOT NULL
-> )
-> GROUP BY
-> aid.id
-> HAVING
-> total_lagu_dimasukkan_playlist >= (
-> SELECT
-> AVG(p.total_lagu_dimasukkan_playlist)
-> FROM (
-> SELECT
-> l.artis_id,
-> COUNT(ip.playlist_id) AS total_lagu_dimasukkan_playlist
-> FROM
-> isi_playlist AS ip
-> INNER JOIN lagu AS l
-> ON (ip.lagu_id = l.id)
-> GROUP BY
-> l.artis_id
-> ) AS p
-> )
-> ORDER BY
-> total_lagu_dimasukkan_playlist DESC,
-> total_konten DESC,
-> aid.nama_pengguna ASC;

```

nama_pengguna	total_lagu	total_video_musik	total_video_extra	total_konten	total_lagu_dimasukkan_playlist
Ajeng Salsabilla	4	7	0	11	9
Drajat Lazuardi	3	3	1	7	9
Irawan Raihan	4	3	1	8	8

3 rows in set (0.012 sec)

**LAMPIRAN**  
**ASISTENSI MODEL ER**

**a. Deskripsi Asistensi**

Nomor Kelompok	10
Anggota yang hadir saat wawancara	1. Dwi Kalam Amal Tauhid (13519210) 2. Akbar Al Fattah (13522036) 3. Imanuel Sebastian Girsang (13522058) 4. Ahmad Mudabbir Arif (13522072) 5. Sa'ad Abdul Hakim (13522092)
Tempat, tanggal	Laboratorium Basis Data, 30 April 2024
Waktu	11.00 - 12.00 WIB
Nama Asisten	Anthonio Natthan Krishna (13521162)

**b. Notulensi**

No.	Pertanyaan	Jawaban
1.	Bagaimana mendefinisikan tanggal berakhir dari subscription?	<p>Untuk kasus umum, cara mendefinisikan tanggal berakhir adalah bulannya bertambah 1/tulis bulan setelahnya tanpa mengubah tanggalnya sama sekali. Contoh: jika tanggal mulai subscription adalah tanggal 1 April, maka tanggal berakhir subscription adalah 1 Mei.</p> <p>Untuk kasus khusus, yaitu tanggal mulai subscription 30 dan 31 Januari, tanggal berakhir dari subscriptionnya adalah 29 Februari.</p>
2	Bagaimana skema penyimpanan lirik yang baru?	Lirik akan berubah menjadi menyimpan tiap line.
3	Apa saja yang harus dihapus dari model ER kami?	Relationship IS A pada pengguna untuk spesialisasi pengguna harus dihapus karena bisa saja pengguna apple music merangkap dua jenis pengguna secara bersamaan.
4	Apa yang harus kami lakukan terlebih dahulu?	Ubah model ER-nya terlebih dahulu, lalu buat model relasionalnya, dan buat database di file SQL.

**c. Dokumentasi**





**LAMPIRAN**  
**ASISTENSI MODEL RELASIONAL**

**a. Deskripsi Asistensi**

Nomor Kelompok	10
Anggota yang hadir saat wawancara	1. Dwi Kalam Amal Tauhid (13519210) 2. Akbar Al Fattah (13522036) 3. Imanuel Sebastian Girsang (13522058) 4. Ahmad Mudabbir Arif (13522072) 5. Sa'ad Abdul Hakim (13522092)
Tempat, tanggal	Online, 4 April 2024
Waktu	11.00 - 12.00 WIB
Nama Asisten	Anthonio Natthan Krishna (13521162)

**b. Notulensi**

No.	Pertanyaan	Jawaban
1.	Apakah model relasional harus ada tipe data?	Tidak perlu, cukup PK dan FK saja
2	Apa yang salah dari gambar panah pada model relasional kami?	Panahnya tidak boleh ditimpa, harus dipisah
3	Apakah harus dibuat tabel per entitas yang dispesialisasi?	Tidak perlu
4	Apakah entitas masih diperlukan di model relasional?	Pengguna tidak perlu karena pemandu cukup mereferensi ke entitas Apple_ID
5	Apakah penulis lirik perlu menjadi primary key?	Penulis lirik tidak perlu menjadi primary key
6	Apakah keempat atribut playlist semuanya PK?	Ya, semuanya PK
7	View ada jumlah durasi, apakah ditulis di model?	Ref attribute akan mengambil durasi dari query, tidak diperlu ditulis.
8	Apa saja yang harus diimplementasi ke SQL?	Yang perlu diimplementasi di SQL hanya relational model.



## **LAMPIRAN**

### **// Query untuk menghapus semua table**

```
DROP TABLE host_video_extra;  
DROP TABLE video_musik;  
DROP TABLE kualitas_audio_lagu;  
DROP TABLE lirik;  
DROP TABLE subscription;  
DROP TABLE isi_playlist;  
DROP TABLE lagu_produk_komersial;  
DROP TABLE playlist;  
DROP TABLE video_extra;  
DROP TABLE produk_komersial;  
DROP TABLE lagu;  
DROP TABLE label;  
DROP TABLE subscription_plan;  
DROP TABLE apple_id;
```

**// Query untuk melakukan load CSV ke table**

```
LOAD DATA INFILE 'apple_id.csv' INTO TABLE apple_id FIELDS  
TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\r\n'  
IGNORE 1 ROWS;
```

```
LOAD DATA INFILE 'subscription_plan.csv' INTO TABLE subscription_plan  
FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY  
'\r\n' IGNORE 1 ROWS;
```

```
LOAD DATA INFILE 'label.csv' INTO TABLE label FIELDS TERMINATED  
BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\r\n' IGNORE 1 ROWS;
```

```
LOAD DATA INFILE 'lagu.csv' INTO TABLE lagu FIELDS TERMINATED  
BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\r\n' IGNORE 1 ROWS;
```

```
LOAD DATA INFILE 'produk_komersial.csv' INTO TABLE produk_komersial  
FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY  
'\r\n' IGNORE 1 ROWS;
```

```
LOAD DATA INFILE 'video_extra.csv' INTO TABLE video_extra FIELDS  
TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\r\n'  
IGNORE 1 ROWS;
```

```
LOAD DATA INFILE 'playlist.csv' INTO TABLE playlist FIELDS  
TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\r\n'  
IGNORE 1 ROWS;
```

```
LOAD DATA INFILE 'lagu_produk_komersial.csv' INTO TABLE  
lagu_produk_komersial FIELDS TERMINATED BY ',' ENCLOSED BY ''"  
LINES TERMINATED BY '\r\n' IGNORE 1 ROWS;
```

```
LOAD DATA INFILE 'isi_playlist.csv' INTO TABLE isi_playlist FIELDS  
TERMINATED BY ',' ENCLOSED BY '' " LINES TERMINATED BY '\r\n'  
IGNORE 1 ROWS;
```

```
LOAD DATA INFILE 'subscription.csv' INTO TABLE subscription FIELDS  
TERMINATED BY ',' ENCLOSED BY '' " LINES TERMINATED BY '\r\n'  
IGNORE 1 ROWS;
```

```
LOAD DATA INFILE 'lirik.csv' INTO TABLE lirik FIELDS TERMINATED BY  
, ' ENCLOSED BY '' " LINES TERMINATED BY '\r\n' IGNORE 1 ROWS;
```

```
LOAD DATA INFILE 'kualitas_audio_lagu.csv' INTO TABLE  
kualitas_audio_lagu FIELDS TERMINATED BY ',' ENCLOSED BY '' " LINES  
TERMINATED BY '\r\n' IGNORE 1 ROWS;
```

```
LOAD DATA INFILE 'video_musik.csv' INTO TABLE video_musik FIELDS  
TERMINATED BY ',' ENCLOSED BY '' " LINES TERMINATED BY '\r\n'  
IGNORE 1 ROWS;
```

```
LOAD DATA INFILE 'host_video_extra.csv' INTO TABLE host_video_extra  
FIELDS TERMINATED BY ',' ENCLOSED BY '' " LINES TERMINATED BY  
'\r\n' IGNORE 1 ROWS;
```

**// Query untuk melakukan pembuatan TABLE**

```
CREATE TABLE apple_id (  
    id INT AUTO_INCREMENT,  
    email VARCHAR (255) NOT NULL,  
    password VARCHAR (255) NOT NULL,  
    nama_pengguna VARCHAR (255) NOT NULL,  
    no_telepon_recovery VARCHAR (255) NOT NULL,  
    PRIMARY KEY (id)  
);
```

```
CREATE TABLE subscription_plan (  
    jenis ENUM ("Pelajar", "Perorangan", "Keluarga"),  
    harga_per_bulan INT NOT NULL,  
  
    PRIMARY KEY (jenis)  
);
```

```
CREATE TABLE label (  
    id INT AUTO_INCREMENT,  
    nama VARCHAR (255) NOT NULL,  
    tahun_berdiri INT NOT NULL,  
    asal_negara VARCHAR (255) NOT NULL,  
  
    PRIMARY KEY (id)  
);
```

```
CREATE TABLE lagu (  

```

```
id INT AUTO_INCREMENT,  
artis_id INT NOT NULL,  
label_id INT NOT NULL,  
judul VARCHAR (255) NOT NULL,  
durasi INT NOT NULL,  
tanggal_rilis DATE NOT NULL,  
  
PRIMARY KEY (id),  
FOREIGN KEY (artis_id) REFERENCES apple_id (id)  
    ON UPDATE CASCADE  
    ON DELETE RESTRICT,  
FOREIGN KEY (label_id) REFERENCES label (id)  
    ON UPDATE CASCADE  
    ON DELETE RESTRICT  
);
```

```
CREATE TABLE produk_komersial (  
    id INT AUTO_INCREMENT,  
    artis_id INT NOT NULL,  
    judul VARCHAR (255) NOT NULL,  
    tipe ENUM ("EP", "Album", "Single"),  
    genre VARCHAR (255) NOT NULL,  
    tanggal_rilis DATE NOT NULL,  
  
    PRIMARY KEY (id),  
    FOREIGN KEY (artis_id) REFERENCES apple_id (id)  
        ON UPDATE CASCADE
```

```
        ON DELETE RESTRICT
    );

CREATE TABLE video_extra (
    id INT AUTO_INCREMENT,
    artis_id INT NOT NULL,
    label_id INT NOT NULL,
    judul VARCHAR (255) NOT NULL,
    durasi INT NOT NULL,
    tanggal_rilis DATE NOT NULL,

    PRIMARY KEY (id),
    FOREIGN KEY (artis_id) REFERENCES apple_id (id)
        ON UPDATE CASCADE
        ON DELETE RESTRICT,
    FOREIGN KEY (label_id) REFERENCES label (id)
        ON UPDATE CASCADE
        ON DELETE RESTRICT
);

CREATE TABLE playlist (
    playlist_id INT,
    pengguna_id INT,
    nama VARCHAR (255) NOT NULL,

    PRIMARY KEY (playlist_id, pengguna_id),
    FOREIGN KEY (pengguna_id) REFERENCES apple_id (id)
```



```
        ON UPDATE CASCADE
        ON DELETE RESTRICT
);

CREATE TABLE lagu_produk_komersial (
    lagu_id INT,
    produk_komersial_id INT,

    PRIMARY KEY (lagu_id, produk_komersial_id),
    FOREIGN KEY (lagu_id) REFERENCES lagu (id)
        ON UPDATE CASCADE
        ON DELETE RESTRICT,
    FOREIGN KEY (produk_komersial_id) REFERENCES
    produk_komersial (id)
        ON UPDATE CASCADE
        ON DELETE RESTRICT
);
```

```
CREATE TABLE isi_playlist (
    playlist_id INT,
    pengguna_id INT,
    lagu_id INT,
    produk_komersial_id INT,

    PRIMARY KEY (playlist_id, pengguna_id, lagu_id,
    produk_komersial_id),
```

```
FOREIGN KEY (playlist_id, pengguna_id) REFERENCES playlist
(playlist_id, pengguna_id)
    ON UPDATE CASCADE
    ON DELETE RESTRICT,
FOREIGN KEY (lagu_id, produk_komersial_id) REFERENCES
lagu_produk_komersial (lagu_id, produk_komersial_id)
    ON UPDATE CASCADE
    ON DELETE RESTRICT
);
```

```
CREATE TABLE subscription (
    subscription_id INT,
    pengguna_id INT,
    subscription_plan_jenis ENUM ("Pelajar", "Perorangan", "Keluarga"),
    tanggal_subscribe DATE NOT NULL,
    tanggal_expired DATE NOT NULL,
    status ENUM ("aktif", "inaktif"),

    PRIMARY KEY (subscription_id, pengguna_id),
    FOREIGN KEY (pengguna_id) REFERENCES apple_id (id)
        ON UPDATE CASCADE
        ON DELETE RESTRICT,
    FOREIGN KEY (subscription_plan_jenis) REFERENCES
subscription_plan (jenis)
        ON UPDATE CASCADE
        ON DELETE RESTRICT
);
```

```
CREATE TABLE lirik (  
    line INT,  
    lagu_id INT,  
    writer_id INT NOT NULL,  
    text VARCHAR (255) NOT NULL,  
  
    PRIMARY KEY (line, lagu_id),  
    FOREIGN KEY (lagu_id) REFERENCES lagu (id)  
        ON UPDATE CASCADE  
        ON DELETE RESTRICT,  
    FOREIGN KEY (writer_id) REFERENCES apple_id (id)  
        ON UPDATE CASCADE  
        ON DELETE RESTRICT  
);
```

```
CREATE TABLE kualitas_audio_lagu (  
    lagu_id INT,  
    kualitas_audio ENUM ("Hi-Res Lossless", "Dolby Atmos"),  
  
    PRIMARY KEY (lagu_id, kualitas_audio),  
    FOREIGN KEY (lagu_id) REFERENCES lagu (id)  
        ON UPDATE CASCADE  
        ON DELETE RESTRICT  
);
```

```
CREATE TABLE video_musik (
```

```
id INT AUTO_INCREMENT,  
artis_id INT NOT NULL,  
lagu_id INT NOT NULL,  
label_id INT NOT NULL,  
judul VARCHAR (255) NOT NULL,  
durasi INT NOT NULL,  
tanggal_rilis DATE NOT NULL,  
  
PRIMARY KEY (id),  
FOREIGN KEY (artis_id) REFERENCES apple_id (id)  
    ON UPDATE CASCADE  
    ON DELETE RESTRICT,  
FOREIGN KEY (lagu_id) REFERENCES lagu (id)  
    ON UPDATE CASCADE  
    ON DELETE RESTRICT,  
FOREIGN KEY (label_id) REFERENCES label (id)  
    ON UPDATE CASCADE  
    ON DELETE RESTRICT  
);
```

```
CREATE TABLE host_video_extra (  
    host_id INT,  
    video_extra_id INT,  
  
    PRIMARY KEY (host_id, video_extra_id),  
    FOREIGN KEY (host_id) REFERENCES apple_id (id)  
        ON UPDATE CASCADE
```

```
        ON DELETE RESTRICT,  
FOREIGN KEY (video_extra_id) REFERENCES video_extra (id)  
        ON UPDATE CASCADE  
        ON DELETE RESTRICT  
);
```