

**IF2240 - BASIS DATA  
MILESTONE 2  
TRANSFORMASI MODEL ER KE RELASIONAL DAN  
IMPLEMENTASI BASIS DATA**



**Kelompok 07 K-04:**

**13519175 Stefanus Jeremy Aslan  
13519179 Akifa Nabil Ufairah  
13519185 Richard Rivaldo  
13519199 Christian Gunawan  
13519201 Muhammad Rayhan Ravianda**

**Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung  
2021**

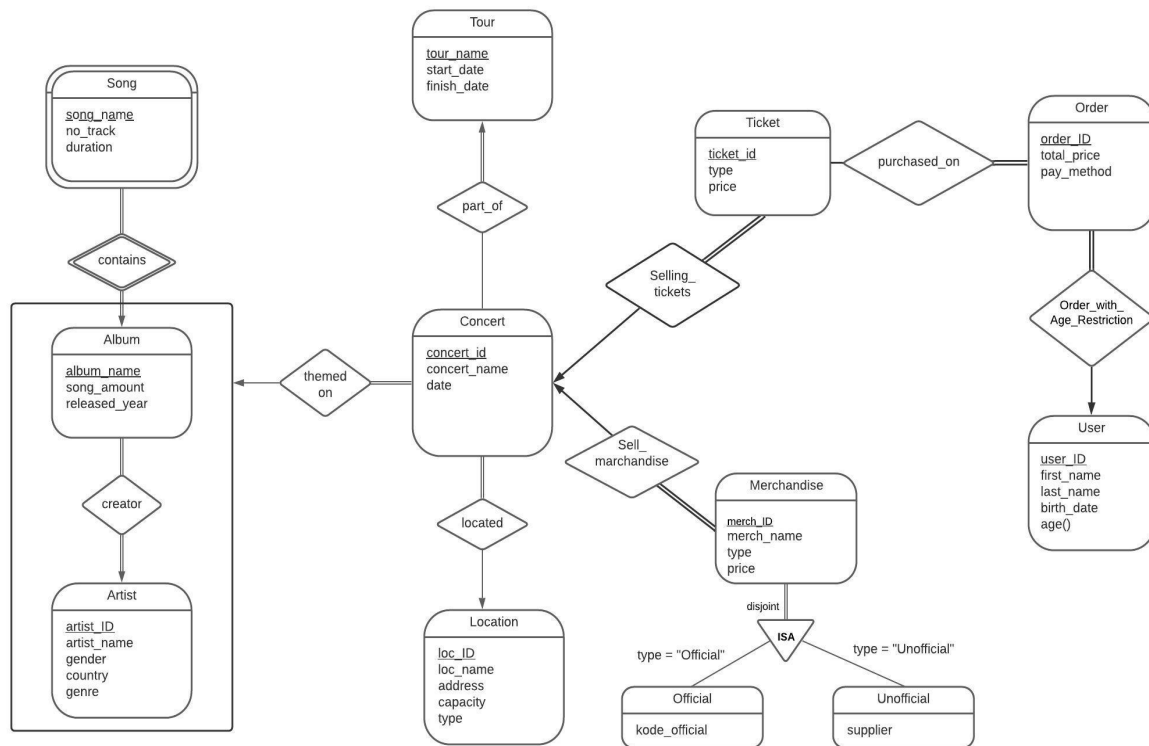
## 1. Deskripsi Sistem

Sebuah platform penjualan tiket bernama EventAja akan membuat sebuah section baru, dimana di section tersebut, user dapat membeli tiket konser. Untuk membuat section tersebut, dibutuhkan sebuah sistem basis data konser, dimana dalam pembuatannya pihak EventAja meminta bantuan mahasiswa informatika untuk pembuatan sistem basis data tersebut.

EventAja akan mencatat konser serta lokasi diadakannya. Setiap konser bisa jadi menjadi bagian dari sebuah tour. Dalam konser, akan dijual tiket yang dapat dipesan melalui pemesanan yang sekaligus dapat dilakukan untuk memesan *merchandise*. Setiap konser memiliki batas minimal umur untuk penontonnya. Merchandise yang dijual ada 2 jenis, yakni merch *official* dan yang *unofficial*. Pemesanan dibayar dengan suatu pembayaran yang hanya dapat dilakukan sekali untuk setiap pemesanan. Pemesanan dilakukan oleh seorang pemesan dengan umur yang memenuhi syarat.

Untuk mengadakan sebuah konser, artis harus membuat album terlebih dahulu sehingga sebuah konser diadakan berdasarkan album yang dikeluarkan artis. Sebuah album berisikan lagu yang memiliki nomor track sebagai urutan khusus dalam album. Tidak ada lagu yang tidak berada di dalam album.

## 2. Model *Entity Relationship Diagram*



**Gambar Diagram Entity-Relationship EventAja**

Di dalam ERD tersebut diasumsikan bahwa relasi *purchased\_on* telah menangani kasus pembelian tiket dan *merchandise* secara bersamaan. Hal ini sesuai dengan *requirement gathering* yang dilakukan sebelumnya, yaitu tidak diperlukannya atribut untuk menyimpan *merch* yang dibeli.

## 3. Deskripsi Revisi

Pada entitas *Ticket*, ditambahkan suatu atribut yang menjadi pembeda antara suatu tiket dengan tiket lainnya, yaitu *ticket\_id*. Selain itu, relasi *creator* yang menghubungkan entitas *Album* dan *Artist* dijadikan sebagai sebuah agregat, dan relasi *themed on* dengan entitas *Concert* akan mengarah pada agregat ini karena agregat ini menjadi sebuah *prerequisite* untuk relasi tersebut.

#### 4. Pemetaan Model *Entity-Relationship* ke Model Relasional

Ketika entitas *Tour* diubah ke dalam model relasional, maka semua atribut yang ada di entitas ini akan masuk ke dalam relasi tersebut. Karena entitas *Tour* hanya memiliki relasi yang bersifat *One-To-Many* dengan entitas *Concert*, maka tidak ada atribut tambahan yang perlu ditambahkan ke dalam relasi *Tour*. Relasi *Tour* ini akan menjadi sebagai berikut.

$\text{Tour} = (\text{tour\_name}, \text{start\_date}, \text{finish\_date})$

Hal serupa terjadi terhadap entitas *Location*, yang semua atributnya akan dimasukkan ke dalam relasi *Location*. Entitas ini juga hanya berhubungan dengan entitas *Concert* dalam hubungan *One-To-Many*, sehingga tidak ada atribut tambahan yang perlu ditambahkan ke dalam relasi *Location* ini. Relasi *Location* ini akan menjadi sebagai berikut.

$\text{Location} = (\text{loc\_ID}, \text{loc\_name}, \text{address}, \text{capacity}, \text{type})$

Namun, karena entitas *Concert* memiliki hubungan *Many-To-One* dengan banyak entitas lainnya, seperti *Location* dan *Tour* tadi, maka *Primary Key* dari *Location* dan *Tour* akan dititipkan ke *Concert* dan menjadi atribut tambahan untuk relasi ini. Selain itu, karena *Concert* juga berhubungan dengan entitas hasil agregasi *Album* dan *Artist*, maka *Primary Key* kedua entitas ini juga dititipkan ke dalam *Concert*. Transformasi entitas *Concert* akan menghasilkan relasi berikut.

$\text{Concert} = (\text{concert\_id}, \text{tour\_name}, \text{loc\_ID}, \text{album\_name}, \text{artist\_ID}, \text{concert\_name}, \text{date})$

Kelas entitas *Song* diubah ke dalam model relasional. Entitas *Song* bersifat sebagai *weak entity* dan memiliki relasi *Many-To-One* dengan entitas *Album*. Maka pada relasi *Song* ada tambahan *Primary Key* dari *Album* yang nantinya akan juga menjadi *Primary Key* dari relasi *Song*. Relasi ini akan menjadi

$\text{Song} = (\text{song\_name}, \text{album\_name}, \text{no\_track}, \text{duration})$

Kelas entitas *Artist* yang berada dalam hubungan dengan *Album* dengan hubungannya *One-To-Many* sehingga tidak ada atribut yang perlu ditambahkan. Maka relasi *Artist* menjadi

$\text{Artist} = (\text{artist\_ID}, \text{artist\_name}, \text{gender}, \text{country}, \text{genre})$

Kelas entitas *Album* yang berada dalam relasi *Album*. Entitas ini juga berhubungan dengan *Artist* dan *Song* dengan hubungan keduanya adalah *Many-To-One* dan *One-To-Many*. Maka relasi *Album* ditambahkan *primary key* dari relasi *Artist* menjadi

$\text{Album} = (\text{album\_name}, \text{artist\_ID}, \text{song\_amount}, \text{released\_year})$

Entitas *Merchandise* pada ERD merupakan sebuah entitas dengan dua entitas yang diturunkan, yaitu *Official* dan *Unofficial* berdasarkan atribut *type*. Karena itu, *Merchandise* dapat menjadi sebuah relasi pada model relasional dengan tambahan atribut dari relasi *Concert* untuk hubungan *Many-To-One*, sebagai berikut

Merchandise = (**merch\_id**, concert\_ID, merch\_name, type, price)

Entitas *Official* sebagai salah satu entitas yang diturunkan sebagai bentuk spesialisasi dari entitas *Merchandise* dapat menjadi sebuah relasi pada model relasional dengan atribut adalah semua atribut dari entitas *Official* sendiri ditambah *primary key* dari relasi *Merchandise*, yaitu *merch\_id*.

Official = (**merch\_id**, kode\_official)

Sama halnya dengan entitas *Official*, *Unofficial* sebagai entitas yang merupakan spesialisasi dari entitas *Merchandise* dapat menjadi sebuah relasi pada model relasional dengan atribut adalah semua atribut dari entitas *Official* sendiri ditambah *primary key* dari relasi *Merchandise*.

Unofficial = (**merch\_id**, supplier)

Entitas *Ticket* dipetakan menjadi relasi *Ticket* pada model relasional dengan atribut mencakup seluruh atribut pada entitas *ticket* serta tambahan atribut, yaitu *concert\_ID* yang merupakan *primary key* dari entitas *Concert*. Atribut ini ditambahkan karena entitas *Ticket* dan entitas *Concert* memiliki hubungan *Many-to-One* dengan *Many side* berada pada entitas *Ticket*. Karena itu, *primary key* dari relasi *Concert* dititipkan pada relasi *Ticket*

Ticket = (**ticket\_id**, concert\_ID, type, price)

Entitas *User* dipetakan menjadi relasi *User* yang mencakup semua atribut entitas *User* kecuali fungsi *age()* yang dihilangkan pada pemetaan karena merupakan atribut turunan atau derivasi. Relasi *User* dapat dilihat sebagai berikut.

User = (**user\_ID**, first\_name, last\_name, birth\_date)

Entitas *Order* dan entitas *Ticket* memiliki relasi *Many-to-Many* di bawah relasi entitas *Purchase*, sehingga pada model relasional, diciptakan relasi bernama *Purchase* dengan *attribute* berupa *primary key* relasi *Order* dan relasi *Ticket*. Relasi *Purchase* dapat dilihat sebagai berikut.

Purchase = (**order\_ID**, **ticket\_id**)

Entitas *Order* dipetakan menjadi relasi *Order* yang mencakup semua atribut entitas *Order*. Karena relasi *Many-to-One* antara entitas *Order* dengan entitas *User*, terdapat tambahan attribute pada relasi *Order* yaitu *reference* terhadap *user\_ID* pada relasi *User*. Relasi *Order* dapat dilihat sebagai berikut.

Order = (**order\_ID**, user\_ID, total\_price, pay\_method)

Atribut *artist\_ID* pada entitas *Album* merupakan *foreign key* dari entitas *Artist*. *Foreign keys reference* dapat dilihat sebagai berikut.

Album(artist\_ID) → Artist(artist\_ID)

Atribut *album\_name* pada entitas *Song* merupakan *foreign key* dari entitas *Album*. *Foreign keys reference* dapat dilihat sebagai berikut.

Song(album\_name) → Album(album\_name)

Atribut *tour\_name* pada entitas *Concert* merupakan *foreign key* dari entitas *Tour*. *Foreign keys reference* dapat dilihat sebagai berikut.

Concert (tour\_name) → Tour(tour\_name)

Atribut *loc\_ID* pada entitas *Concert* merupakan *foreign key* dari entitas *Location*. *Foreign keys reference* dapat dilihat sebagai berikut.

Concert (loc\_ID) → Location(loc\_ID)

Atribut *album\_name* pada entitas *Concert* merupakan *foreign key* dari entitas *Album*. *Foreign keys reference* dapat dilihat sebagai berikut.

Concert(album\_name) → Album(album\_name)

Atribut *artist\_ID* pada entitas *Concert* merupakan *foreign key* dari entitas *Artist*. *Foreign keys reference* dapat dilihat sebagai berikut.

Concert(artist\_ID) → Artist(artist\_ID)

Atribut *concert\_ID* pada entitas *Merchandise* merupakan *foreign key* dari entitas *Concert*. *Foreign keys reference* dapat dilihat sebagai berikut.

Merchandise(concert\_ID) → Concert(concert\_ID)

Atribut *concert\_ID* pada entitas *Ticket* merupakan *foreign key* dari entitas *Concert*. *Foreign keys reference* dapat dilihat sebagai berikut.

Ticket(concert\_ID) → Concert(concert\_ID)

Atribut *merch\_id* pada entitas *Official* merupakan *foreign key* dari entitas *Merchandise*. *Foreign keys reference* dapat dilihat sebagai berikut.

Official(merch\_id) → Merchandise(merch\_id)

Atribut *merch\_id* pada entitas *Unofficial* merupakan *foreign key* dari entitas *Merchandise*. *Foreign keys reference* dapat dilihat sebagai berikut.

Unofficial(merch\_id) → Merchandise(merch\_id)

Atribut *order\_ID* pada entitas *Purchase* merupakan *foreign key* dari entitas *Order*. *Foreign keys reference* dapat dilihat sebagai berikut.

Purchase(order\_ID) → Order(order\_ID)

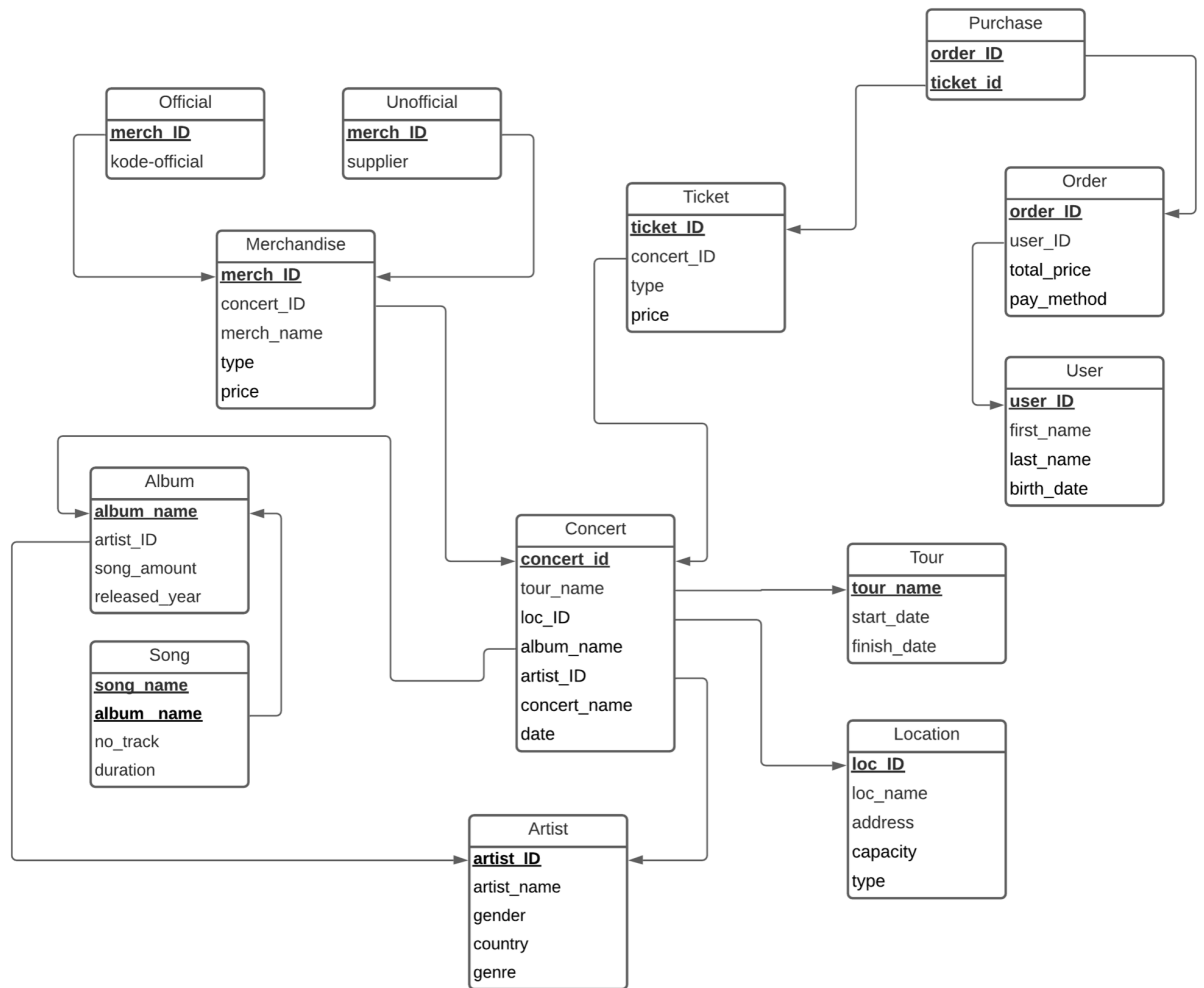
Atribut *ticket\_ID* pada entitas *Purchase* merupakan *foreign key* dari entitas *Ticket*. *Foreign keys reference* dapat dilihat sebagai berikut.

Purchase(ticket\_ID) → Ticket(ticket\_ID)

Atribut *user\_ID* pada entitas *Order* merupakan *foreign key* dari entitas *User*. *Foreign keys reference* dapat dilihat sebagai berikut.

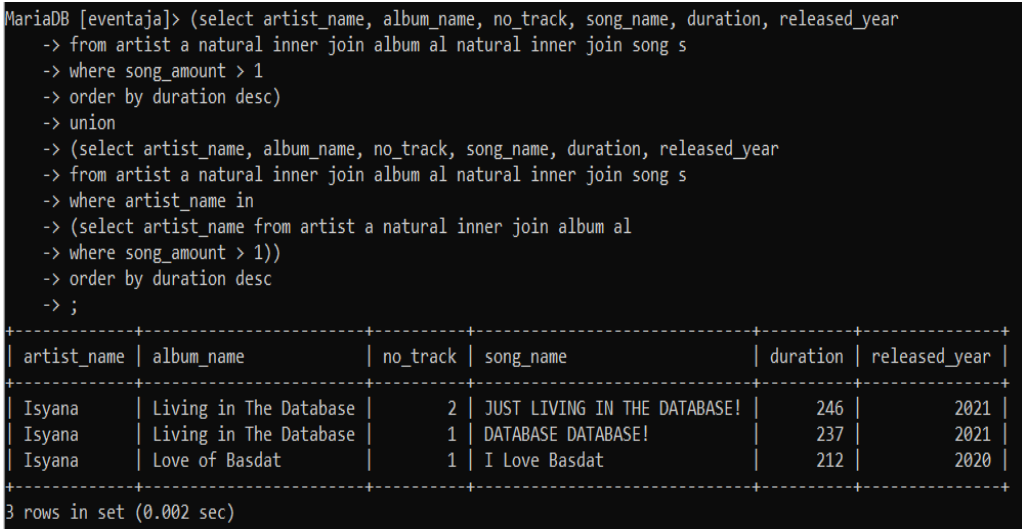
Order(user\_ID) → User(user\_ID)

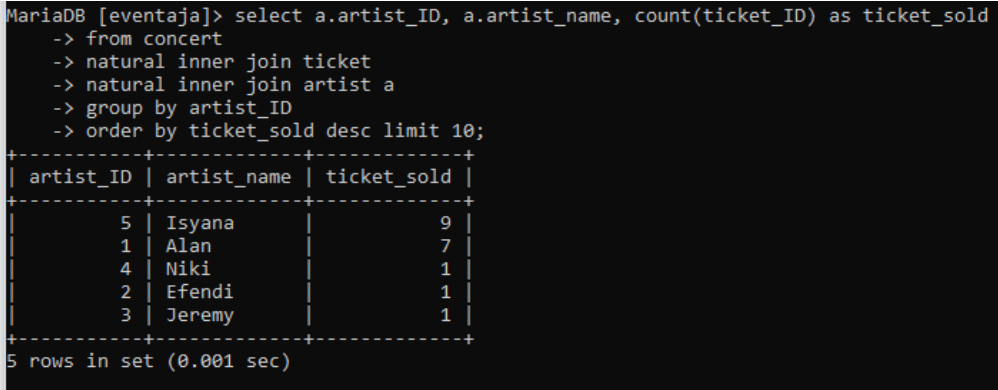
## 5. Model Relasional

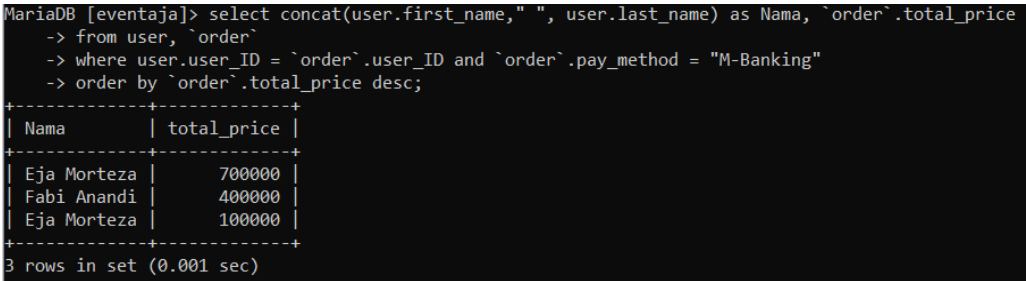
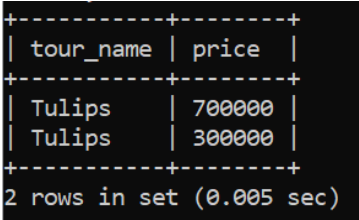




## 6. *Insight dan Query Basis Data*

Case	Insight																								
1	<p>Menampilkan nama penyanyi, nama album, <i>nomor track</i>, nama lagu, durasi, dan tahun rilis dari lagu yang terkandung dalam album yang memiliki lebih dari satu buah lagu, diurutkan berdasarkan durasi dari yang terlama sampai yang tersingkat. Gabungkan hasil pencarian dengan album lain yang dimiliki oleh penyanyi yang sama jika ada. Urutkan lagi data yang dihasilkan dengan cara yang serupa.</p> <p><i>Query:</i></p> <pre>(select artist_name, album_name, no_track, song_name, duration, released_year from artist a natural inner join album al natural inner join song s where song_amount &gt; 1 order by duration desc)  union  (select artist_name, album_name, no_track, song_name, duration, released_year from artist a natural inner join album al natural inner join song s where artist_name in (select artist_name from artist a natural inner join album al where song_amount &gt; 1))  order by duration desc;</pre> <p><i>Screenshot Query:</i></p>  <pre>MariaDB [eventaja]&gt; (select artist_name, album_name, no_track, song_name, duration, released_year -&gt; from artist a natural inner join album al natural inner join song s -&gt; where song_amount &gt; 1 -&gt; order by duration desc) -&gt; union -&gt; (select artist_name, album_name, no_track, song_name, duration, released_year -&gt; from artist a natural inner join album al natural inner join song s -&gt; where artist_name in -&gt; (select artist_name from artist a natural inner join album al -&gt; where song_amount &gt; 1)) -&gt; order by duration desc -&gt; ;</pre> <table><tr><th>artist_name</th><th>album_name</th><th>no_track</th><th>song_name</th><th>duration</th><th>released_year</th></tr><tr><td>Isyana</td><td>Living in The Database</td><td>2</td><td>JUST LIVING IN THE DATABASE!</td><td>246</td><td>2021</td></tr><tr><td>Isyana</td><td>Living in The Database</td><td>1</td><td>DATABASE DATABASE!</td><td>237</td><td>2021</td></tr><tr><td>Isyana</td><td>Love of Basdat</td><td>1</td><td>I Love Basdat</td><td>212</td><td>2020</td></tr></table> <p>3 rows in set (0.002 sec)</p>	artist_name	album_name	no_track	song_name	duration	released_year	Isyana	Living in The Database	2	JUST LIVING IN THE DATABASE!	246	2021	Isyana	Living in The Database	1	DATABASE DATABASE!	237	2021	Isyana	Love of Basdat	1	I Love Basdat	212	2020
artist_name	album_name	no_track	song_name	duration	released_year																				
Isyana	Living in The Database	2	JUST LIVING IN THE DATABASE!	246	2021																				
Isyana	Living in The Database	1	DATABASE DATABASE!	237	2021																				
Isyana	Love of Basdat	1	I Love Basdat	212	2020																				

2	<p>Daftar tour diurutkan dari tour dengan total pendapatan terbanyak</p> <p><i>Query :</i></p> <pre>select t.tour_name, sum(o.total_price) as 'Income' from tour t,concert c, location l, purchase p, ticket tc, `order` o where t.tour_name = c.tour_name and c.loc_ID = l.loc_ID and c.concert_ID = tc.concert_ID and p.ticket_ID = tc.ticket_ID and p.order_ID = o.order_ID group by t.tour_name order by sum(o.total_price) desc</pre> <p><i>Screenshot Query :</i></p> <table><tr><th>tour_name</th><th>Income</th></tr><tr><td>Tulips</td><td>3500000</td></tr><tr><td>Basis</td><td>500000</td></tr><tr><td>Roses</td><td>200000</td></tr><tr><td>Violets</td><td>200000</td></tr><tr><td>Lilacs</td><td>100000</td></tr></table> <p>5 rows in set (0.001 sec)</p>	tour_name	Income	Tulips	3500000	Basis	500000	Roses	200000	Violets	200000	Lilacs	100000						
tour_name	Income																		
Tulips	3500000																		
Basis	500000																		
Roses	200000																		
Violets	200000																		
Lilacs	100000																		
3	<p>Daftar (hingga) 10 artist terpopuler berdasarkan jumlah tiket konser yang terjual.</p> <p><i>Query :</i></p> <pre>select a.artist_ID, a.artist_name, count(ticket_ID) as ticket_sold from concert natural inner join ticket natural inner join artist a group by artist_ID order by ticket_sold desc limit 10;</pre> <p><i>Screenshot Query :</i></p>  <pre>MariaDB [eventaja]&gt; select a.artist_ID, a.artist_name, count(ticket_ID) as ticket_sold -&gt; from concert -&gt; natural inner join ticket -&gt; natural inner join artist a -&gt; group by artist_ID -&gt; order by ticket_sold desc limit 10;</pre> <table><tr><th>artist_ID</th><th>artist_name</th><th>ticket_sold</th></tr><tr><td>5</td><td>Isyana</td><td>9</td></tr><tr><td>1</td><td>Alan</td><td>7</td></tr><tr><td>4</td><td>Niki</td><td>1</td></tr><tr><td>2</td><td>Efendi</td><td>1</td></tr><tr><td>3</td><td>Jeremy</td><td>1</td></tr></table> <p>5 rows in set (0.001 sec)</p>	artist_ID	artist_name	ticket_sold	5	Isyana	9	1	Alan	7	4	Niki	1	2	Efendi	1	3	Jeremy	1
artist_ID	artist_name	ticket_sold																	
5	Isyana	9																	
1	Alan	7																	
4	Niki	1																	
2	Efendi	1																	
3	Jeremy	1																	

4	<p>Daftar transaksi user berisi nama lengkap user dan uang yang dibayarkan dengan metode M-Banking diurutkan berdasarkan pembayaran terbesar</p> <p><i>Query :</i></p> <pre>select concat(user.first_name," ", user.last_name) as Nama, `order`.total_price from user, `order` where user.user_ID = `order`.user_ID and `order`.pay_method = "M-Banking" order by `order`.total_price desc;</pre> <p><i>Screenshot Query :</i></p>  <pre>MariaDB [eventaja]&gt; select concat(user.first_name," ", user.last_name) as Nama, `order`.total_price -&gt; from user, `order` -&gt; where user.user_ID = `order`.user_ID and `order`.pay_method = "M-Banking" -&gt; order by `order`.total_price desc; +-----+-----+   Nama        total_price   +-----+-----+   Eja Morteza   700000          Fabi Anandi   400000          Eja Morteza   100000        +-----+-----+ 3 rows in set (0.001 sec)</pre>
5	<p>Daftar nama tour dan harga tiket dengan metode pembayaran ATM dan diurutkan dari yang terbesar</p> <p><i>Query :</i></p> <pre>select c.tour_name, t.price from concert as c, ticket as t, purchase as p, `order` as o where c.concert_ID = t.concert_ID and t.ticket_ID = p.ticket_ID and p.order_ID = o.order_ID and o.pay_method = "ATM" order by t.price desc;</pre> <p><i>Screenshot Query :</i></p>  <pre>+-----+-----+   tour_name   price   +-----+-----+   Tulips      700000     Tulips      300000   +-----+-----+ 2 rows in set (0.005 sec)</pre>

## 7. Referensi

Database Design Using E-R Model oleh Tim Pengajar IF2140 TA 2020/2021. Diakses melalui

[https://stei19.kuliah.itb.ac.id/pluginfile.php/80724/mod\\_resource/content/1/IF2240%20-%20Design%20using%20E-R%20Model%20-%20part%201.pdf](https://stei19.kuliah.itb.ac.id/pluginfile.php/80724/mod_resource/content/1/IF2240%20-%20Design%20using%20E-R%20Model%20-%20part%201.pdf) pada 18 Maret 2021 pukul 12.31 WIB.

Database Design Using E-R Model Bagian 2 oleh Tim Pengajar IF2140 TA 2020/2021. Diakses melalui

[https://stei19.kuliah.itb.ac.id/pluginfile.php/80848/mod\\_resource/content/1/IF2240%20-%20Design%20using%20E-R%20Model%20-part%202.pdf](https://stei19.kuliah.itb.ac.id/pluginfile.php/80848/mod_resource/content/1/IF2240%20-%20Design%20using%20E-R%20Model%20-part%202.pdf) pada 18 Maret 2021 pukul 12.37 WIB.

Weak Entity Set in E-R Diagrams oleh Mithlesh Upadhyay. Diakses melalui <https://www.geeksforgeeks.org/weak-entity-set-in-er-diagrams/> pada 18 Maret 2021 pukul 13.25 WIB.

## 8. Lampiran MoM

# Moments of Meeting

### 1. Deskripsi Wawancara

Nomor Kelompok	07 K04
Anggota yang hadir saat wawancara	1. Stefanus Jeremy Aslan 2. Akifa Nabil Ufairah 3. Richard Rivaldo 4. Christian Gunawan 5. Muhammad Rayhan Ravianda
Tempat, tanggal	<a href="https://meet.google.com/kun-kkqd-gsg?hs=224">https://meet.google.com/kun-kkqd-gsg?hs=224</a> , 2 April 2021.
Waktu	13.30 - 14.30 WIB.
Nama Asisten	Chandrika Azharyanti

### 2. Hasil Wawancara

No.	Pertanyaan	Jawaban
1.	Revisi ERD	Tiket, kasih pembeda, misalnya nomor kursi supaya bisa membedakan tiket yang satu dengan yang lain.
2	Revisi ERD	Album dan Artis prereq/syarat mengadakan suatu konser, sebaiknya di agregasi, karena wajib ada untuk konser.
3	Song perlu dimasukkan ke agregat?	Tidak perlu karena agregasi satu relasi aja.
4	Insight dan query, apakah kita bikin data dan query sendiri ?	Iya, sesuai poin 3. Tujuannya untuk memberikan/menampilkan informasi. Tabel acuan minimal 5 data, tapi tabel transaksi harusnya lebih banyak datanya.
5	Langkah-langkah transformasi menjelaskan sesuai teori?	Iya sesuai teori aja.
6	Model Relasional diisi apa?	Screenshot tabel, atau bikin kayak struktur biasa di draw io dulu

Tanda Tangan Asisten

A handwritten signature in black ink, consisting of a large, stylized 'O' followed by a series of loops and a final horizontal stroke.