

# PROJECT BASIS DATA

JASA ANTAR MAKANAN  
ONLINE

# ANGGOTA KELOMPOK 3

01



**Taufik Satria  
Nugraha**

H1D020028

02



**Maulana  
Zhahran**

H1D021070

03



**Naufal Althafi  
Handoyo**

H1D021087

# A. ROLE UTAMA BASIS DATA

Terdapat predefined roles dan user defined roles. Predefined roles merupakan role yang sudah disiapkan oleh DBMS Oracle, sedangkan user defined roles merupakan role yang ditentukan sendiri oleh programmer. User defined role dalam projek basis data ini ada 3 yaitu :

1. Admin : Mengatur dan merawat basis data
2. Driver : Membeli makanan yang sudah di pesan oleh pembeli, lalu mengantarkannya ke alamat pembeli
3. Pembeli : Memesan makanan yang diinginkan

## B. ENTITAS BASIS DATA

Entitas atau entity dalam database adalah benda, orang, tempat, unit, objek atau hal lainnya yang mempresentasikan data dan data tersebut akan disimpan ke dalam pangkal data. Setiap entitas memiliki beberapa atribut yang mendeskripsikan karakteristik dari objek. Entitas dalam projek basis data ini adalah sebagai berikut :

1. Pembeli (id\_pembeli, nama\_pembeli, alamat\_pembeli, no\_tel\_pembeli, email)
2. Pesanan (id\_pesanan, tanggal\_pemesanan, total\_harga)
3. Driver (id\_driver, nama\_driver, no\_tel\_driver, rating\_driver)
4. Makanan (id\_makanan, nama\_makanan, harga\_makanan, rating\_makanan)
5. Restoran (id\_restoran, nama\_restoran, alamat\_restoran, no\_tel\_restoran)

## C. ATURAN BISNIS

01

Satu pembeli bisa melakukan banyak pesanan.  
Satu pesanan hanya bisa dilakukan oleh satu pembeli.

02

Satu driver bisa mengantar banyak pesanan.  
Satu pesanan hanya bisa diantar oleh satu driver.

03

Satu pesanan dapat memiliki banyak makanan.  
Satu makanan dapat dimiliki oleh banyak pesanan.

04

Satu restoran menyediakan banyak makanan.  
Satu makanan hanya bisa disediakan oleh satu restoran.

05

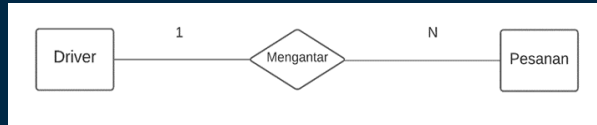
Satu pembeli menilai banyak driver.  
Satu driver dapat dinilai oleh banyak pembeli.

06

Satu pembeli menilai banyak makanan.  
Satu makanan dapat dinilai oleh banyak pembeli.

# D. RELASI BASIS DATA

## 1. Driver mengantar pesanan



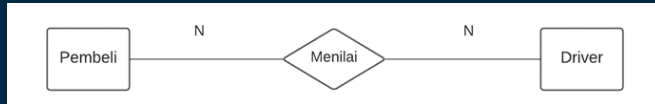
Tabel utama : Driver, Pesanan  
Tabel kedua : Pesanan  
Relasi : One to Many (1-N)  
Atribut penghubung : id\_driver (Foregin Key id\_driver di Pesanan)

## 2. Pembeli melakukan pesanan



Tabel utama : Pembeli, Pesanan  
Tabel kedua : Pesanan  
Relasi : One to Many (1-N)  
Atribut penghubung : id\_pembeli (Foregin Key id\_pembeli di Pesanan)

### 3. Pembeli menilai driver



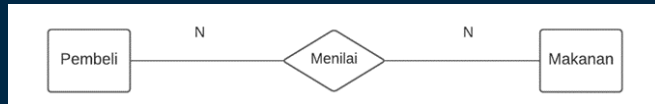
Tabel utama : Pembeli, Driver

Tabel kedua : rating\_driver

Relasi : Many to Many (N-N)

Atribut penghubung : id\_driver, id\_pembeli (Foreign Key id\_driver, id\_pembeli di rating\_driver)

### 4. Pembeli menilai makanan



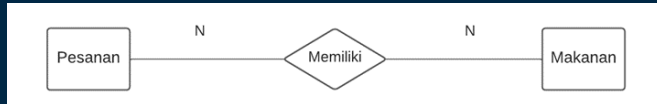
Tabel utama : Pembeli, Makanan

Tabel kedua : rating\_makanan

Relasi : Many to Many (N-N)

Atribut penghubung : id\_makanan, id\_pembeli (Foreign Key id\_makanan, id\_pembeli di rating\_makanan)

#### 5. Pesanan memiliki makanan



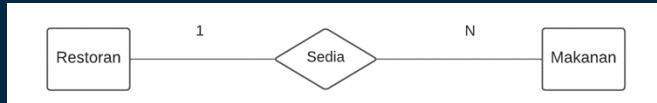
Tabel utama : Pesanan, makanan

Tabel kedua : detail\_pesanan

Relasi : Many to Many (N-N)

Atribut penghubung : id\_pesanan, id\_makanan (Foreign Key id\_pesanan, id\_makanan di detail\_pesanan)

#### 6. Restoran sedia makanan



Tabel utama : Restoran, Makanan

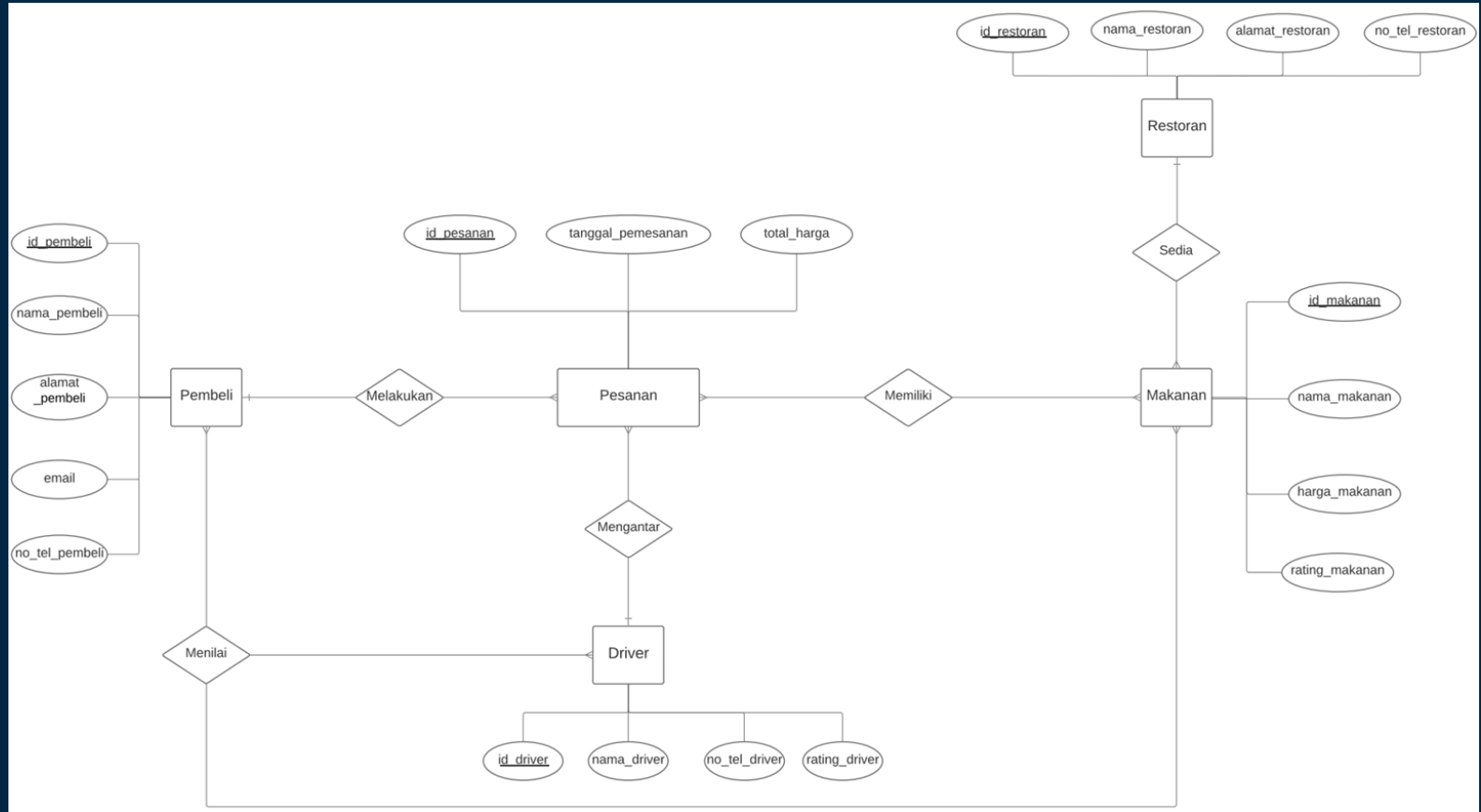
Tabel kedua : Makanan

Relasi : One to Many (1-N)

Atribut penghubung : id\_restoran (Foreign Key id\_restoran di Makanan)



# E. RANCANGAN ERD





# LANGKAH-LANGKAH PEMBUATAN SISTEM BASIS DATA JASA ANTAR MAKANAN

Pada pembuatan basis data ini, kami menggunakan dbms oracle version 10g express edition. Berikut ini langkah-langkah yang dilakukan dalam pembuatan basis data.

### 1. Membuat dan memberikan privileges ke user admin

```
1 create user admin identified by admin;
2 grant connect,resource to admin;
```

### 2. Membuat tabel pembeli

```
1 create table PEMBELI (
2     ID_PEMBELI          VARCHAR2(10)          not null,
3     NAMA_PEMBELI        VARCHAR2(50),
4     ALAMAT_PEMBELI      VARCHAR2(200),
5     NO_TEL_PEMBELI      VARCHAR2(12),
6     EMAIL                VARCHAR2(50),
7     constraint PK_PEMBELI primary key (ID_PEMBELI)
8 );
```

### 3. Membuat tabel driver

```
1 create table DRIVER (
2     ID_DRIVER           VARCHAR2(5)            not null,
3     NAMA_DRIVER         VARCHAR2(50),
4     NO_TEL_DRIVER       VARCHAR2(12),
5     RATING_DRIVER       FLOAT,
6     constraint PK_DRIVER primary key (ID_DRIVER)
7 );
```

### 4. Membuat tabel restoran

```
1 create table RESTORAN (
2     ID_RESTORAN        VARCHAR2(10)           not null,
3     NAMA_RESTORAN      VARCHAR2(50),
4     ALAMAT_RESTORAN    VARCHAR2(200),
5     NO_TEL_RESTORAN    VARCHAR2(12),
6     constraint PK_RESTORAN primary key (ID_RESTORAN)
7 );
```

## 5. Membuat tabel makanan

```
1 create table MAKANAN (  
2   ID_MAKANAN      VARCHAR2(10)          not null,  
3   ID_RESTORAN     VARCHAR2(10),  
4   NAMA_MAKANAN    VARCHAR2(100),  
5   HARGA_MAKANAN   INTEGER,  
6   RATING_MAKANAN  FLOAT,  
7   constraint PK_MAKANAN primary key (ID_MAKANAN)  
8 );
```

## 6. Membuat tabel pesanan

```
1 create table PESANAN (  
2   ID_PESANAN      VARCHAR2(10)          not null,  
3   ID_PEMBELI      VARCHAR2(10),  
4   ID_DRIVER       VARCHAR2(5),  
5   TANGGAL_PEMESANAN  TIMESTAMP,  
6   TOTAL_HARGA      INTEGER,  
7   constraint PK_PESANAN primary key (ID_PESANAN)  
8 );
```

## 7. Membuat tabel detail pesanan

```
1 create table DETAIL_PESANAN (  
2   ID_PESANAN      VARCHAR2(10)          not null,  
3   ID_MAKANAN      VARCHAR2(10)          not null,  
4   JUMLAH_MAKANAN  NUMBER(2),  
5   constraint PK_DETAIL_PESANAN primary key (ID_PESANAN, ID_MAKANAN)  
6 );
```

## 8. Membuat tabel rating makanan

```
1 create table RATING_MAKANAN (  
2   ID_PEMBELI      VARCHAR2(10)          not null,  
3   ID_MAKANAN      VARCHAR2(10)          not null,  
4   RATING_MAKANAN  NUMBER(5),  
5   constraint PK_RATING_MAKANAN primary key (ID_PEMBELI, ID_MAKANAN)  
6 );
```

## 9. Membuat tabel rating\_driver

```
1 create table RATING_DRIVER (  
2   ID_PEMBELI      VARCHAR2(10)          not null,  
3   ID_DRIVER       VARCHAR2(5)           not null,  
4   RATING_DRIVER   NUMBER(5),  
5   constraint PK_RATING_DRIVER primary key (ID_PEMBELI, ID_DRIVER)  
6 );
```

## 10. Membuat tabel c\_rating\_driver

```
1 create table C_RATING_DRIVER (  
2   ID_PEMBELI      VARCHAR2(10)          not null,  
3   ID_DRIVER       VARCHAR2(5)           not null,  
4   RATING_DRIVER   NUMBER(5),  
5   constraint PK_C_RATING_DRIVER primary key (ID_PEMBELI, ID_DRIVER)  
6 );
```

## 11. Membuat tabel c\_rating\_makanan

```
1 create table C_RATING_MAKANAN (  
2   ID_PEMBELI      VARCHAR2(10)          not null,  
3   ID_MAKANAN      VARCHAR2(10)          not null,  
4   RATING_MAKANAN  NUMBER(5),  
5   constraint PK_C_RATING_MAKANAN primary key (ID_PEMBELI, ID_MAKANAN)  
6 );
```

## 12. Membuat tabel c\_detail\_pesanan

```
1 create table C_DETAIL_PESANAN (  
2   ID_PESANAN      VARCHAR2(10)          not null,  
3   ID_MAKANAN      VARCHAR2(10)          not null,  
4   JUMLAH_MAKANAN  NUMBER(2),  
5   constraint PK_C_DETAIL_PESANAN primary key (ID_PESANAN, ID_MAKANAN)  
6 );
```

### 13. Membuat procedure add\_makanan

```
1 create or replace procedure add_makanan(nama_makanan in varchar2
2 , harga_makanan in number, id_restoran in varchar2)
3 as
4 begin
5     insert into
6     MAKANAN(id_makanan, nama_makanan, harga_makanan, id_restoran) values
7     ('MKN' || MAKANAN_SEQ.nextval,
8      nama_makanan,
9      harga_makanan,
10     id_restoran);
11 end;
12 /
```

### 14. Membuat procedure add\_restoran

```
1 create or replace procedure add_restoran(nama_restoran in varchar2
2 , alamat_restoran in varchar2, no_tel_restoran in varchar2)
3 as
4 begin
5     insert into RESTORAN values ('RSTRN' || RESTORAN_SEQ.nextval,
6     nama_restoran,
7     alamat_restoran,
8     no_tel_restoran);
9 end;
10 /
```

### 15. Membuat procedure add\_pesanan

```
1 create or replace procedure add_pesanan(id_pembeli in varchar2
2 , id_driver in varchar2)
3 as
4 begin
5     insert into
6     PESANAN(id_pesanan, id_pembeli, id_driver, tanggal_pemesanan) values
7     ('PSN' || PESANAN_SEQ.nextval,
8     id_pembeli,
9     id_driver,
10    systimestamp);
11 end;
12 /
```

### 16. Membuat procedure add\_detail\_pesanan

```
1 create or replace procedure add_detail_pesanan(id_pesanan in varchar2
2 , id_makanan in varchar2, jumlah_makanan in number)
3 as
4 begin
5     insert into
6     DETAIL_PESANAN(id_pesanan, id_makanan, jumlah_makanan) values (
7     id_pesanan,
8     id_makanan,
9     jumlah_makanan);
10 end;
11 /
```

## 17. Membuat procedure add\_rating\_driver

```
1 create or replace procedure add_rating_driver(id_pembeli in varchar2
2 , id_driver in varchar2, rating in number)
3 as
4 begin
5     if (rating > 0 and rating <= 5) then
6         insert into RATING_DRIVER(ID_PEMBELI, ID_DRIVER, RATING_DRIVER)
7         values(id_pembeli, id_driver, rating);
8     else
9         raise_application_error(-20000, 'Rating harus antara 1-5');
10    end if;
11 end;
12 /
```

## 18. Membuat procedure add\_rating\_makanan

```
1 create or replace procedure add_rating_makanan(id_pembeli in varchar2
2 , id_makanan in varchar2, rating in number)
3 as
4 begin
5     if (rating > 0 and rating <= 5) then
6         insert into RATING_MAKANAN(ID_PEMBELI, ID_MAKANAN, RATING_MAKANAN)
7         values(id_pembeli, id_makanan, rating);
8     else
9         raise_application_error(-20000, 'Rating harus antara 1-5');
10    end if;
11 end;
12 /
```

## 19. Membuat procedure calculate\_rating\_driver

```
1 create or replace procedure calculate_rating_driver(id_driver1 in
2 varchar2)
3 as
4 begin
5     update DRIVER
6     set RATING_DRIVER = (select round(avg(RATING_DRIVER),2)
7     from C_RATING_DRIVER
8     where ID_DRIVER = id_driver1)
9     where ID_DRIVER = id_driver1;
10 end;
11 /
```

## 20. Membuat procedure calculate\_rating\_makanan

```
1 create or replace procedure calculate_rating_makanan(id_makanan1 in
2 varchar2)
3 as
4 begin
5     update MAKANAN
6     set RATING_MAKANAN = (select round(avg(RATING_MAKANAN),2)
7     from C_RATING_MAKANAN
8     where ID_MAKANAN = id_makanan1)
9     where ID_MAKANAN = id_makanan1;
10 end;
11 /
```

## 21. Membuat function calculate\_harga\_makanan

```
1 create or replace function calculate_harga_makanan(id_pesanan1 in
  varchar2)
2   return INTEGER as
3     total INTEGER := 0;
4     i INTEGER;
5     cursor c_makanan is
6       select HARGA_MAKANAN*JUMLAH_MAKANAN
7         from C_DETAIL_PESANAN join MAKANAN on
8         C_DETAIL_PESANAN.ID_MAKANAN = MAKANAN.ID_MAKANAN
9         where ID_PESANAN = id_pesanan1;
10    begin
11      open c_makanan;
12      loop
13        fetch c_makanan into i;
14        exit when c_makanan%notfound;
15        total := total + i;
16      end loop;
17      close c_makanan;
18      return total;
19    end;
```

## 22. Membuat trigger rating\_driver\_trigger

```
1 create or replace trigger rating_driver_trigger
2   after insert or update on RATING_DRIVER
3   for each row
4   declare
5     jumlah integer;
6     cursor c_jumlah is select count(*) from C_RATING_DRIVER where
7       ID_DRIVER = :new.ID_DRIVER and id_pembeli = :new.ID_PEMBELI;
8   begin
9     open c_jumlah;
10    fetch c_jumlah into jumlah;
11    if jumlah = 1 then
12      update C_RATING_DRIVER
13        set RATING_DRIVER = :new.RATING_DRIVER
14        where ID_DRIVER = :new.ID_DRIVER and id_pembeli = :new.ID_PEMBELI;
15      calculate_rating_driver(:new.ID_DRIVER);
16    else
17      insert into C_RATING_DRIVER values
18        (:new.ID_PEMBELI, :new.ID_DRIVER, :new.RATING_DRIVER);
19      calculate_rating_driver(:new.ID_DRIVER);
20    end if;
21    close c_jumlah;
22  end;
```

## 23. Membuat trigger rating\_makanan\_trigger

```
1 create or replace trigger rating_makanan_trigger
2   after insert or update on RATING_MAKANAN
3   for each row
4   declare
5     jumlah integer;
6     cursor c_jumlah is select count(*) from C_RATING_MAKANAN where
7       ID_MAKANAN = :new.ID_MAKANAN and id_pembeli = :new.ID_PEMBELI;
8   begin
9     open c_jumlah;
10    fetch c_jumlah into jumlah;
11    if jumlah = 1 then
12      update C_RATING_MAKANAN
13        set RATING_MAKANAN = :new.RATING_MAKANAN
14        where ID_MAKANAN = :new.ID_MAKANAN and id_pembeli = :new.ID_PEMBELI;
15      calculate_rating_makanan(:new.ID_MAKANAN);
16    else
17      insert into C_RATING_MAKANAN values
18        (:new.ID_PEMBELI, :new.ID_MAKANAN, :new.RATING_MAKANAN);
19      calculate_rating_makanan(:new.ID_MAKANAN);
20    end if;
21    close c_jumlah;
22  end;
```

## 24. Membuat trigger harga\_makanan\_trigger

```
1 create or replace trigger harga_makanan_trigger
2   after insert on DETAIL_PESANAN
3   for each row
4   begin
5     insert into C_DETAIL_PESANAN values
6       (:new.ID_PESANAN, :new.ID_MAKANAN, :new.JUMLAH_MAKANAN);
7     update PESANAN
8       set TOTAL_HARGA = calculate_harga_makanan(:new.ID_PESANAN)
9       where ID_PESANAN = :new.ID_PESANAN;
10  end;
```





### 39. Insert data detail ke tabel detail\_pesanan

```
SQL> select * from detail_pesanan;
```

ID_PESANAN	ID_MAKANAN	JUMLAH_MAKANAN
PSN1	MKN1	1
PSN1	MKN2	1
PSN1	MKN3	1
PSN2	MKN4	2
PSN2	MKN5	2
PSN3	MKN6	3
PSN4	MKN7	1
PSN4	MKN8	1
PSN5	MKN9	1
PSN5	MKN10	1

10 rows selected.

32. Insert data rating setiap minuman ke tabel rating\_makanan

```
SQL> select * from rating_makanan;
```

ID_PEMBELI	ID_MAKANAN	RATING_MAKANAN
PBLI1	MKN1	3
PBLI1	MKN2	5
PBLI1	MKN3	4
PBLI1	MKN4	5
PBLI1	MKN5	1
PBLI1	MKN6	2
PBLI1	MKN7	5
PBLI1	MKN8	5
PBLI1	MKN9	3
PBLI1	MKN10	5
PBLI2	MKN1	5
PBLI2	MKN2	5
PBLI2	MKN3	5
PBLI2	MKN4	3
PBLI2	MKN5	5
PBLI2	MKN6	4
PBLI2	MKN7	5
PBLI2	MKN8	5
PBLI2	MKN9	2
PBLI2	MKN10	4
PBLI2	MKN11	3

21 rows selected.

### 33. Melihat info pesanan dengan view info\_pesanan

```
1 create or replace view info_pesanan as
2 select id_pesanan, nama_pembeli, nama_driver, alamat_pembeli, tanggal_pemesanan, total_harga
3      from pesanan join pembeli on pesanan.id_pembeli = pembeli.id_pembeli
4      join driver on pesanan.id_driver = driver.id_driver;
```

```
SQL> select * from info_pesanan;
```

ID_PESANAN	NAMA_PEMBELI	NAMA_DRIVER	ALAMAT_PEMBELI	TANGGAL_PEMESANAN	TOTAL_HARGA
PSN1	Nate Creer	Fred Siggins	9 Sheridan Hill	05-06-2022 09.24.41,871000	60000
PSN2	Layne Claybourn	Tuesday Bernette	5132 Killdeer Center	05-06-2022 09.24.41,874000	108000
PSN3	Sim Lyford	Cati Malcher	51992 Fisk Parkway	05-06-2022 09.24.41,880000	30000
PSN4	Ketty McNae	Natka Scoterbosh	48 Trailsway Road	05-06-2022 09.24.41,881000	35000
PSN5	Innis Warke	Jens Sepey	73331 Valley Edge Terrace	05-06-2022 09.24.41,896000	28000
PSN6	Nert Downton	Nathalie Scyner	57 Fallview Lane	05-06-2022 09.24.41,911000	
PSN7	Ruprecht Beany	Nixie Dast	2 Beilfuss Crossing	05-06-2022 09.24.41,927000	
PSN8	Bailie Storey	Margaretthe Killelea	45 Corben Hill	05-06-2022 09.24.41,928000	
PSN9	Nicholas Phelan	Gates O'Shiel	32 Sachs Park	05-06-2022 09.24.41,942000	
PSN10	Angie Winnard	Edie Keegan	6 Shelley Court	05-06-2022 09.24.43,186000	

```
10 rows selected.
```

# TERIMA KASIH

CREDITS: This presentation template was created by [Slidesgo](#),  
including icons by [Flaticon](#), and infographics & images by [Freepik](#)