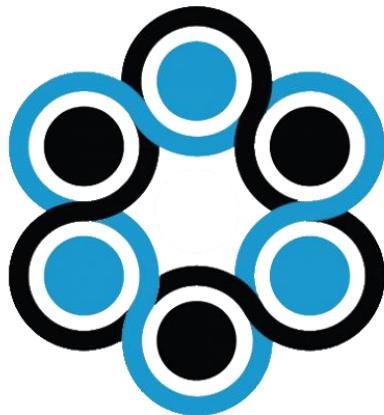


LAPORAN AKHIR
PRAKTIKUM BASIS DATA
Periode V

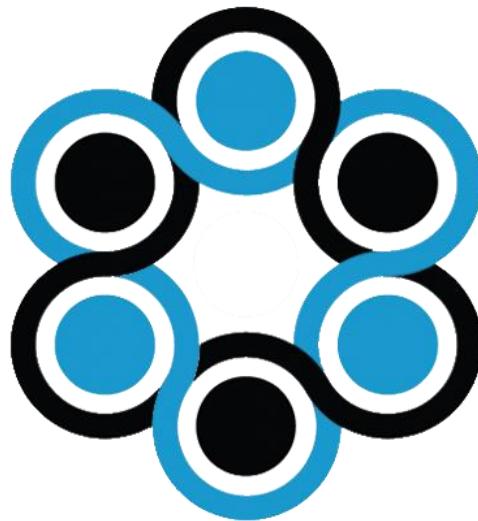


Disusun Oleh :

NAMA : Mochamad Khoirul Huda
NPM : 06.2018.1.06990

**FAKULTAS TEKNIK ELEKTRO DAN
TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI ADHI TAMA SURABAYA
SURABAYA
2020**

LAPORAN AKHIR
PRAKTIKUM BASIS DATA
Periode V



Disusun Oleh:

Mochamad Khoirul Huda 06.2018.1.06990
March Angga

**FAKULTAS TEKNIK ELEKTRO DAN
TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI ADHI TAMA SURABAYA
SURABAYA
2020**

LEMBAR PENGESAHAN

PRAKTIKUM BASIS DATA

Periode V

Di susun oleh :

Mochamad Khoirul Huda

06.2018.1.06990

Tanggal 07/Juli/2020

Dosen Pembimbing

Asisten Pembimbing

Septiawan Rosetya W, S.Kom., M.Kom
NIP. 173132

March Angga

NAMA : Mochamad Khoirul Huda
NPM : 06.2018.1.06990

BIMBINGAN DOSEN
PRAKTIKUM BASIS DATA PERIODE V

No.	TANGGAL	POKOK BAHASAN	PARAF DOSEN PEMBIMBING

**Surabaya,
Dosen Pembimbing**

Septiyawan Rosetya W, S.Kom., M.Kom

NAMA : Mochamad Khoirul Huda
NPM : 06.2018.1.06990

**BIMBINGAN ASISTEN
PRAKTIKUM BASIS DATA PERIODE V**

No.	TANGGAL	POKOK BAHASAN	PARAF ASISTEN PEMBIMBING

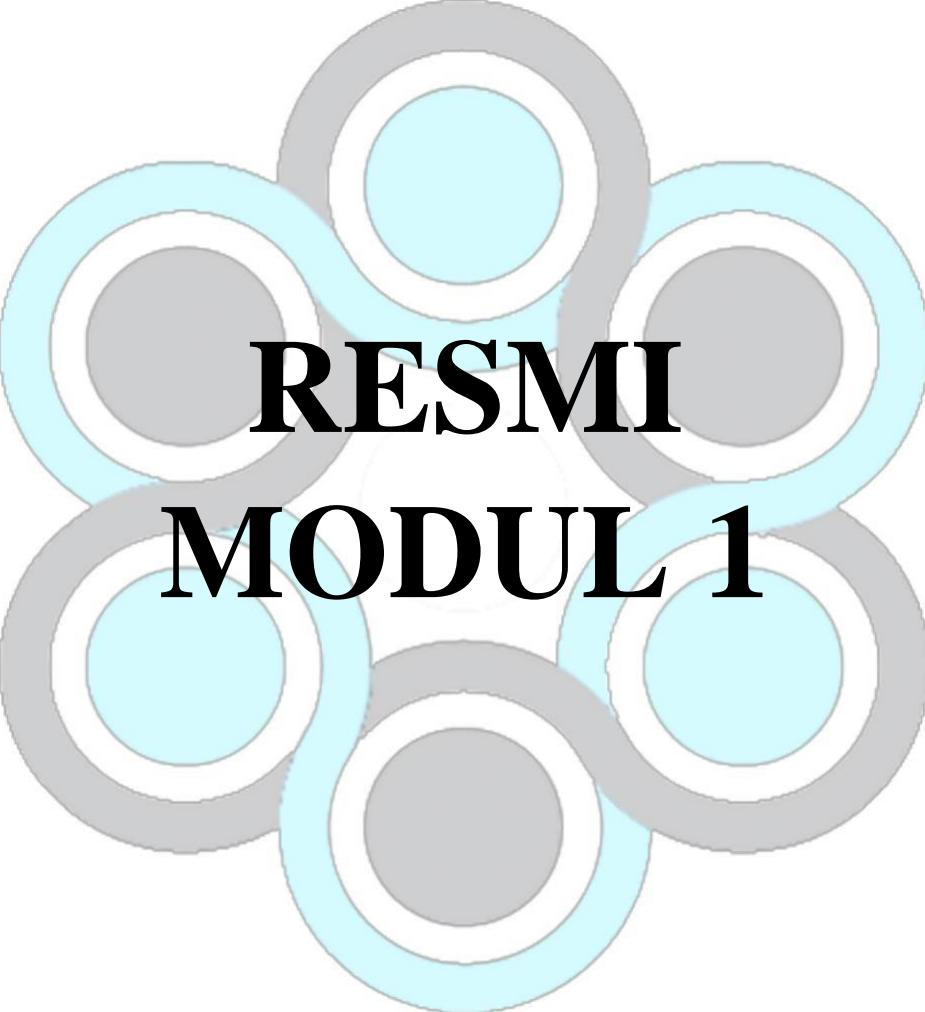
**Surabaya,
Asisten Pembimbing**

March Angga



LAPORAN RESMI

FAKULTAS TEKNIK ELEKTRO DAN
TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI ADHI TAMA
SURABAYA
2020

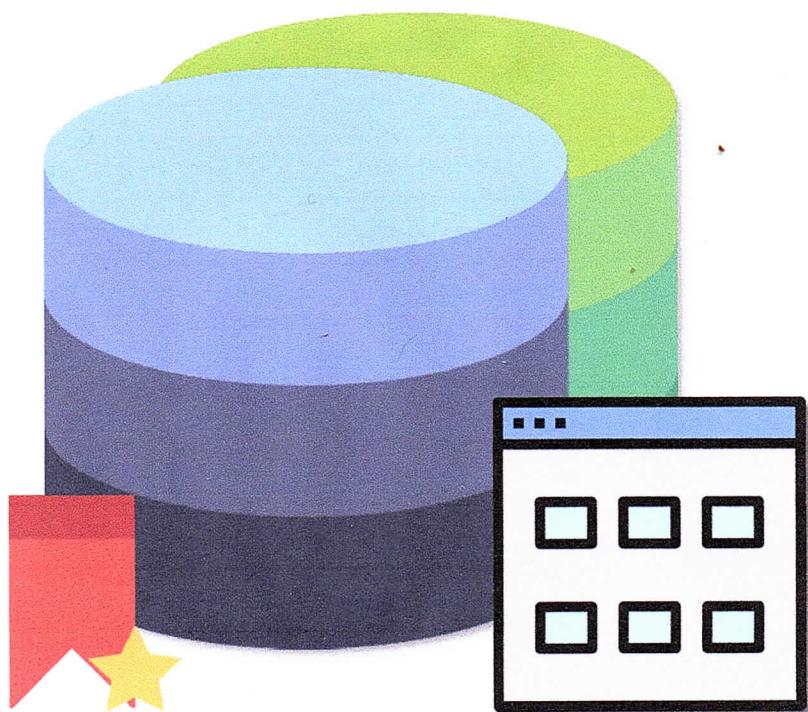


RESMI

MODUL 1

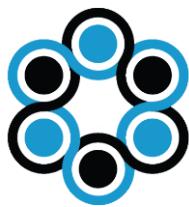
FAKULTAS TEKNIK ELEKTRO DAN
TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI ADHI TAMA
SURABAYA
2020

LAPORAN RESMI



PRAKTIKUM BASIS DATA PERIODE V - Tahun 2019/2020

NAMA : Mochamad Ichsanul Huda
NPM : 06.2018.1.06ggD
MODUL : I



Tugas Resmi

Soal Resmi

1. Buatlah user dan tablespace sesuai Nama dan NPM masing – masing praktikan.
2. Buatlah tabel sesuai Desain Database Proyek Kelas yang sudah di ACC oleh dosen.
3. Buatlah relasi antar tabel sesuai Desain Database dengan perintah ALTER.
4. Menerapkan Sequence pada salah satu tabel.
5. Ubah nama salah satu field pada salah satu tabel dengan menambah NPM didepan nama field.

Contoh : 06852_NamaProduk

6. Ubah salah satu field pada salah satu tabel dengan Attribute Constraint UNIQUE.
7. Ubah salah satu field pada salah satu tabel dengan Tipe data yang berbeda.

Langkah Pertama

- **Buat tablespace**

```
create tablespace <nama_table>
datafile '<Penempatan_file>\<Nama_Tablespace.dbf>'
size <ukuran_tablespace>;
```

- **Buat user**

```
create user <nama_user_NPM>
identified by <password>
default tablespace <nama_tablespace>
quota <banyaknya penyimpanan> on <nama_tablespace>;
```



Tugas Resmi

Query

```
create tablespace il_sogno
datafile
'D:\irul\BERKAS\kuliah\praktikum\basis
data\modul 1\IL_SOOGNO.dbf'
size 30M;

create user khoirul_huda_06990
identified by khoirulhuda
default tablespace IL_SOOGNO
quota 30M on IL_SOOGNO;
```

Screenshot

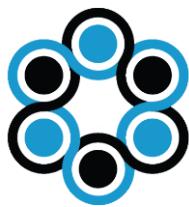
```
SQL> create tablespace il_sogno
2 datafile 'D:\irul\BERKAS\kuliah\praktikum\basis data\modul 1\IL_SOOGNO.dbf'
3 size 30M;

SQL> create user khoirul_huda_06990
2 identified by khoirulhuda
3 default tablespace IL_SOOGNO
4 quota 30M on IL_SOOGNO;

User created.
```

Analisa

Sebelum membuat user kita diharuskan membuat tablespace terlebih dahulu supaya saat kita membuat user, kita bisa memasukan langsung tablespacenya. Pembuatan tablespace kita bisa meletakan data file sesuai dengan keinginan kita dan jangan lupa untuk memberi ukuran data table space tersebut. Pembuatan user harus diawali dengan character, kita tidak bisa membuat user dengan angka atau simbol terlebih dahulu. Kata identified adalah kita memasukan password dari user yang sudah kita buat.



Tugas Resmi

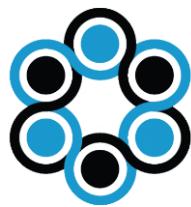
Langkah Kedua

- Buat table dengan nama Marketing, didalam table Marketing terdapat atribut Id_Marketing, Nama_Marketing, No_Telp.
- Buat table Profil_Aluminium yang didalam table Profil_Aluminium terdapat atribut Id_Marketing, Nama_Marketing, No_Telp.
- Buat table Arsitek yang didalam table Arsitek terdapat atribut Id_Arsitek, Nama_Arsitek, Telp_Kantor dengan Id_Arsitek sebagai primary key nya
- Buat table proyek yang didalam table Proyek

Query

```
create table Marketing
(
Id_Marketing int,
Nama_Marketing varchar(30),
No_Telp number(9),
constraint pk_marketing primary key
(Id_Marketing)
);
```

```
create table Profil_Aluminium
(
Id_Profil int,
Nama_Profil varchar(30),
Warna varchar (50),
constraint pk_profil primary key (Id_Profil)
);
```



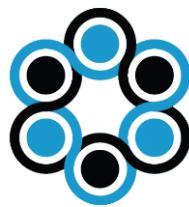
Tugas Resmi

```
create table Profil_Aluminium
(
Id_Profil int,
Nama_Profil varchar(30),
Warna varchar (50),
constraint pk_profil primary key (Id_Profil)
);
```

Screenshot

```
connected.
SQL> create table Marketing
  2  (
  3   Id_Marketing int,
  4   Nama_Marketing varchar(30),
  5   No_Telp number(9)
  6  );
Table created.
```

```
SQL> create table Profil_Aluminium
  2  (
  3   Id_Profil int,
  4   Nama_Profil varchar(30),
  5   Warna varchar (50),
  6   constraint pk_profil primary key (Id_Profil)
  7  );
Table created.
```



Tugas Resmi

```
SQL> create table Arsitek
  2  (
  3  Id_Arsitek integer not null,
  4  Nama_Arsitek varchar(30),
  5  Telp_Kantor number(9),
  6  constraint pk_Arsitek primary key(Id_Arsitek)
  7  );
Table created.
```

```
SQL> create table Proyek
  2  (
  3  Nama_Owner varchar(30),
  4  Alamat_Proyek Varchar(100),
  5  Alamat_Penyuratian varchar(100),
  6  No_Telp number(9),
  7  Id_Arsitek integer,
  8  Id_Marketing integer,
  9  Id_Profil integer,
 10  constraint pk_Proyek primary key(Alamat_Proyek)
 11  );
Table created.
```

Analisa :

Saat membuat table awal dari nama table diharuskan menggunakan character, jika kita menggunakan simbol atau angka terlebih dahulu maka akan terjadi error saat kita membuat table tersebut. Ada beberapa tipe data dalam database, sebagai contoh:

Varchar adalah sebuah tipe data character

Number adalah sebuah tipe data angka real

Integer adalah sebuah tipe data angka

Saat membuat primary key kita menggunakan syntax constraint



Tugas Resmi

Langkah Ketiga

Langkah selanjutnya adalah kita membuat relasi antar table dengan perintah alter

Sql > alter table <nama_tabel>

```
add constraint <nama_foreign_key> foreign key (<atribute_yang_dituju>)
references <nama_tabel_yang_atribute_dijadikan_refrensi>
(<nama_atribute_dari_table_yang_direfrensikan>)
;
```

Query

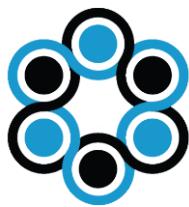
```
alter table Proyek
add constraint fk_Marketing foreign key
(Id_Marketing) references Marketing
(Id_Marketing)
add constraint fk_Arsitek foreign key
(Id_Arsitek) references Arsitek (Id_Arsitek)
add constraint fk_Profil foreign key
(Id_Profil) references Profil_Aluminium
(Id_Profil)
;
```

Screenshot

```
SQL> alter table Proyek
  2  add constraint fk_Marketing foreign key (Id_Marketing) references Marketing (Id_Marketing)
  3  add constraint fk_Arsitek foreign key (Id_Arsitek) references Arsitek (Id_Arsitek)
  4  add constraint fk_Profil foreign key (Id_Profil) references Profil_Aluminium (Id_Profil)
  5 ;
Table altered.
```

Analisa :

Saat kita akan merubah atau menambahkan sesuatu didalam tabel kita akan menggunakan perintah ALTER. Jika kita akan menambahkan sesuatu dalam table kita juga akan menggunakan perintah add. Pembuatan foreign key sama halnya dengan pembuatan primary key, akan tetapi perbedaan pembuatan foreign key terdapat dalam perintah REFERENCES yang artinya di refrensikan dari table yang mana dan didalam kurung itu adalah nama dari field yang direfrensikan.



Tugas Resmi

Langkah Keempat

Langkah ke-4 ini kita akan menggunakan sequence.

Sql > create sequence <nama_field_dalam_table>

```
minvalue <nilai_minimal>
maxvalue <nilai_maksimal>
start with <dimulai_angka>
increment by <dijumlahkan>
cache <kapasitas>;
```

Query

```
create sequence Id_Marketing
minvalue 1
maxvalue 9999
start with 1
increment by 1
cache 20;
```

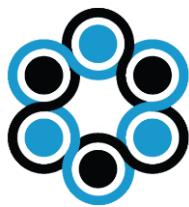
Screenshot

```
SQL> create sequence Id_Marketing
  2  minvalue 1
  3  maxvalue 9999
  4  start with 1
  5  increment by 1
  6  cache 20;

Sequence created.
```

Analisa :

Perintah minvalue ada perintah dari minimal value yang akan diberikan untuk max value adalah perintah pembatasan dari maksimalnya isi field. Start with adalah value yang dimulai dari 1 dikarenakan Id_Marketing adalah field dengan tipe data integer. Increment adalah penjumlah dari isi field yang sebelumnya.



Tugas Resmi

Langkah Kelima

merubah nama field di salah satu table

```
alter table <nama_tabel>
```

```
rename column <nama_field_lama> to <nama_field_baru>;
```

Query

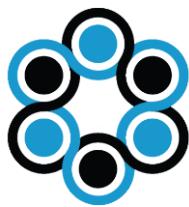
```
alter table Marketing
rename column Nama_Marketing to
06990_Nama_Marketing;
```

Screenshot

```
SQL> alter table Marketing
  2  rename column Nama_Marketing to 06990_Nama_Marketing;
Table altered.
```

Analisa :

Saat kita mengubah sebuah nama table maka kita akan menggunakan perintah ALTER dan di ikuti dengan nama dari sebuah table yang nama fieldnya akan kita rubah. Merubah sebuah ama dari field maka kita juga menggunakan perintah rename dan kata column adalah letak dari field lalu di ikuti nama awal dari sebuah field. Kata TO merujuk ke kata berganti lalu di ikuti dengan nama field yang baru. Akan tetapi seperti yang saya jelaskan sebelumnya jika penamaan sebuah table tidak bisa menggunakan angka terlebih dahulu. Maka dari itu angka 0 diawali nama field yang baru saya ganti dengan menggunakan huruf O.



Tugas Resmi

Langkah Keenam

Kita akan mengubah salah satu field di salah satu tablr dengan attribute Constraint UNIQUE.

```
SQL> ALTER TABLE <nama_tabel>
      add UNIQUE (<nama_field>);
```

Query

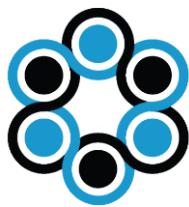
```
ALTER TABLE Proyek
add UNIQUE (No_Telp);
```

Screenshot

```
SQL> ALTER TABLE Proyek
      2 add UNIQUE (No_Telp);
      |
Table altered.
```

Analisa :

ALTER adalah syntax untuk merubah dan menambahkan sesatu. Jadi saat menambahakan UNIQUE kita juga menggunakan syntax ALTER dan diikuti dengan nama table yang dituju. Lalu kita menggunakan kata add diikuti UNIQUE lalu nama fieldnya. UNIQUE bisa dipakai seperti contoh nomer telephone dan email karena di dalam keduanya terdapat sebuah simbol.



Tugas Resmi

Langkah Ketujuh

kita akan mengubah salah satu field dengan tipe data yang berbeda dengan menggunakan perintah di bawah ini

```
alter table <nama_table> modify (<nama_field> <tipe_data>);
```

Query

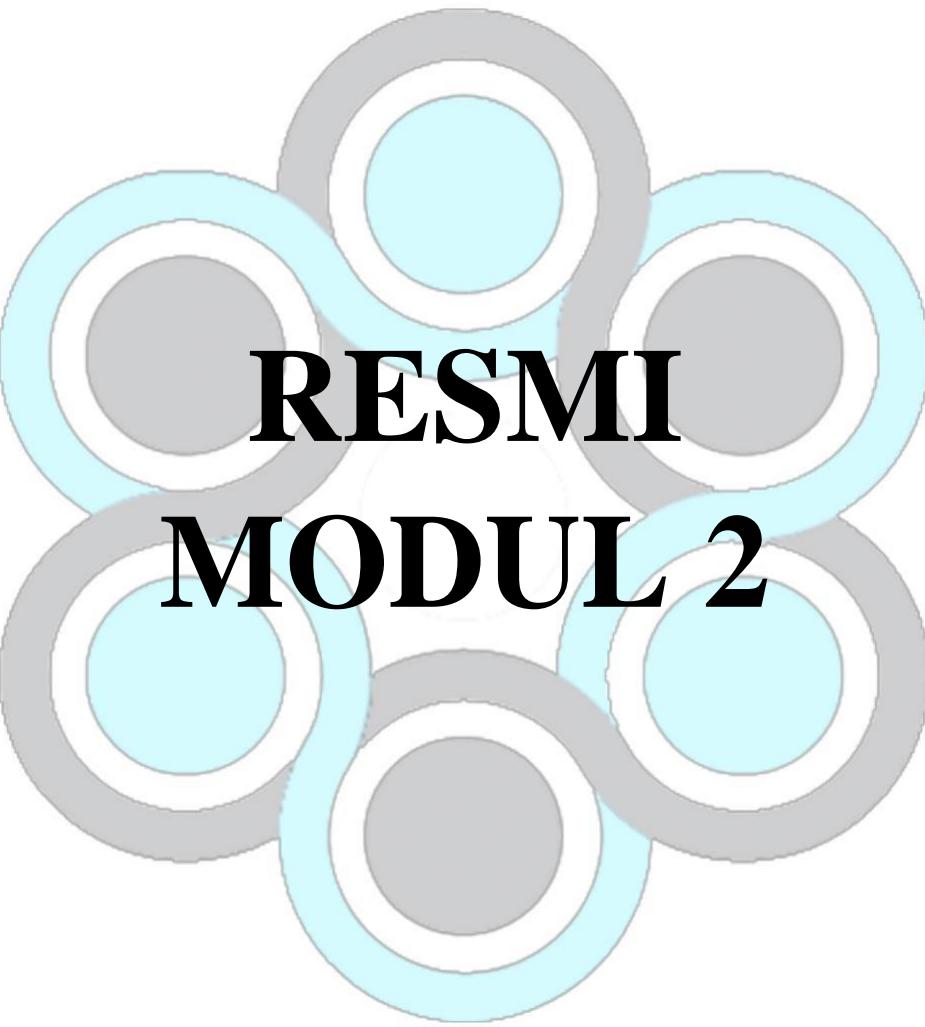
```
alter table Proyek modify (Id_Profil  
number(12));
```

Screenshot

```
SQL> alter table Proyek modify (Id_Profil number(12));  
Table altered.
```

Analisa :

Perintah **MODIFY** ini adalah perintah untuk mengubah tipe data dari sebuah field. Saya mengubah tipe data dari field **Id_Profil** yang sebelumnya integer menjadi number dikarenakan tipe data ini hampir sama.

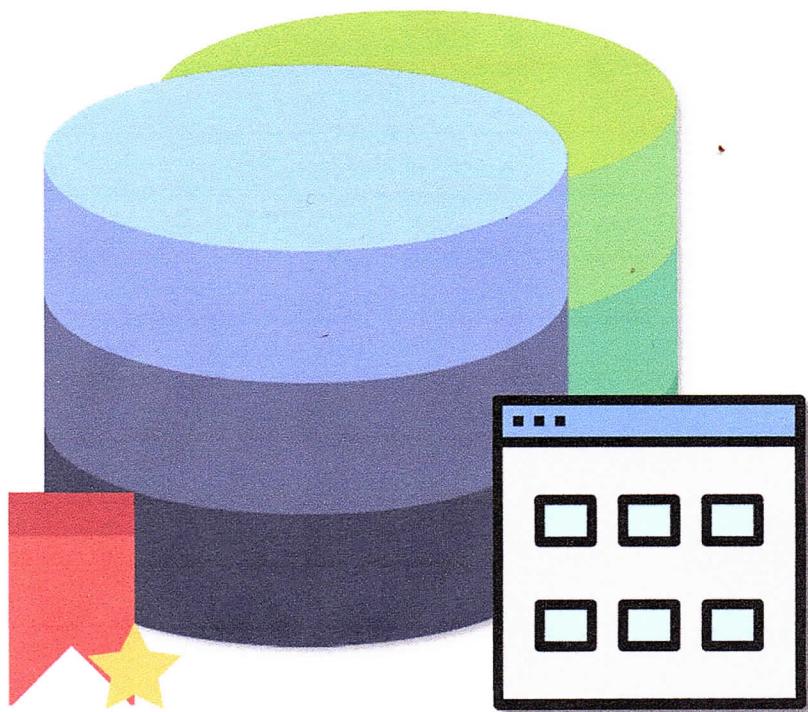


RESMI

MODUL 2

FAKULTAS TEKNIK ELEKTRO DAN
TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI ADHI TAMA
SURABAYA
2020

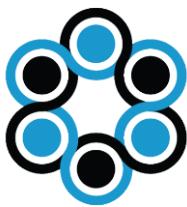
LAPORAN RESMI



PRAKTIKUM BASIS DATA PERIODE V - Tahun 2019/2020

NAMA : Machmud Kharraf Husdi
NPM : 06.2018.1.06990
MODUL : II

--



Tugas Resmi

Soal Resmi

1. Insert data pada setiap tabel yang telah Anda buat :
 - a. Single insert dengan 2 data
 - b. Multiple insert dengan 3 data
 - c. Terapkan insert dengan sequence yang telah Anda buat di Modul 1
2. Update 5 data pada tabel yang berbeda :
 - a. Semua data pada suatu kolom pada tabel
 - b. Menerapkan where clause
 - c. Menerapkan like
 - d. Menerapkan AND, OR dan NOT minimal 2 operator pada satu baris query
3. Delete minimal 3 data pada salah satu tabel dengan menerapkan 1 klausa dan 2 operator pada satu baris query yang berbeda pada setiap data
4. Aturlah transaksi dengan menerapkan :
 - a. Save Point
 - b. Commit
 - c. Rollback
5. Terapkan select dengan menerapkan :
 - a. Order By
 - b. Group By

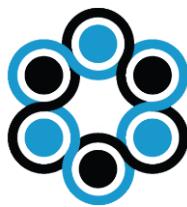
Langkah Pertama

Jika kita memasukan data secara satu persatu kita bisa menggunakan perintah

```
insert into nama_table (nama_kolom1, nama_kolom2, ... ,nama_kolom_ke-n)
values (isi_kolom1, isi_kolom2, ... , isi_kolom_ke-N);
```

Jika kita memasukan data secara bersama-sama maka kita menggunakan perintah

Insert all



Tugas Resmi

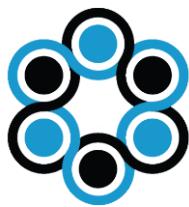
```
into nama_table (nama_kolom1, nama_kolom2, ... ,nama_kolom_ke-n) values  
(isi_kolom1, isi_kolom2, ... , isi_kolom_ke-N)  
into nama_table (nama_kolom1, nama_kolom2, ... ,nama_kolom_ke-n) values  
(isi_kolom1, isi_kolom2, ... , isi_kolom_ke-N)  
select 1 from dual;
```

Query

```
insert into profil_aluminium  
(id_profil, nama_profil, warna)  
values (1, 'strandart', 'iron grey');  
insert into profil_aluminium  
(id_profil, nama_profil, warna)  
values (2, 'standart', 'white');  
  
insert all  
into profil_aluminium (id_profil,  
nama_profil, warna)  
values (3, 'standart', 'black sahara')  
into profil_aluminium (id_profil,  
nama_profil, warna)  
values (4, 'penta 40', 'iron grey')  
into profil_aluminium (id_profil,  
nama_profil, warna)  
values (5, 'penta 40', 'white')  
select 1 from dual;
```

Screenshot

```
SQL> insert into profil_aluminium  
2  (id_profil, nama_profil, warna)  
3  values (1, 'strandart', 'iron grey');  
  
1 row created.  
  
SQL> insert into profil_aluminium  
2  (id_profil, nama_profil, warna)  
3  values (2, 'standart', 'white');  
  
1 row created.  
  
SQL>  
SQL> insert all  
2  into profil_aluminium (id_profil, nama_profil, warna)  
3  values (3, 'standart', 'black sahara')  
4  into profil_aluminium (id_profil, nama_profil, warna)  
5  values (4, 'penta 40', 'iron grey')  
6  into profil_aluminium (id_profil, nama_profil, warna)  
7  values (5, 'penta 40', 'white')  
8  select 1 from dual;  
  
3 rows created.
```



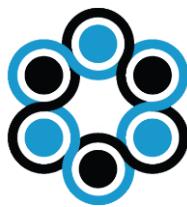
Tugas Resmi

Query

```
insert into marketing
(id_marketing, nama_marketing, no_telp)
values (id_marketing.nextval, 'rahmad',
'123456789');
insert into marketing
(id_marketing, nama_marketing, no_telp)
values (id_marketing.nextval, 'artani',
'234567891');

CREATE FUNCTION get_seq RETURN NUMBER IS
BEGIN
    RETURN id_marketing.nextval;
END;
/

insert all
into marketing (id_marketing, nama_marketing,
no_telp)
values (get_seq, 'hendro', '345678912')
into marketing
(id_marketing, nama_marketing, no_telp)
values (get_seq, 'france', '456789123')
into marketing (id_marketing, nama_marketing,
no_telp)
values (get_seq, 'yudha', '567891234')
select 1 from dual;
```



Tugas Resmi

Screenshot

```
SQL> insert into marketing
  2 (id_marketing, nama_marketing, no_telp)
  3 values (id_marketing.nextval, 'rahmad', '123456789');

1 row created.

SQL> insert into marketing
  2 (id_marketing, nama_marketing, no_telp)
  3 values (id_marketing.nextval, 'artani', '234567891');

1 row created.

SQL> CREATE FUNCTION get_seq RETURN NUMBER IS
  2 BEGIN
  3   RETURN id_marketing.nextval;
  4 END;
  5 /

Function created.

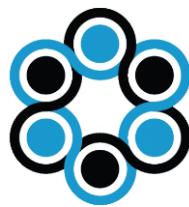
SQL>
SQL> insert all
  2 into marketing (id_marketing, nama_marketing, no_telp)
  3 values (get_seq, 'hendro', '345678912')
  4 into marketing
  5 (id_marketing, nama_marketing, no_telp)
  6 values (get_seq, 'france', '456789123')
  7 into marketing (id_marketing, nama_marketing, no_telp)
  8 values (get_seq, 'yudha', '567891234')
  9 select 1 from dual;

3 rows created.
```

Query

```
insert into arsitek
(id_arsitek, nama_arsitek, telp_kantor)
values (1, 'santoso', '1111111111');
insert into arsitek
(id_arsitek, nama_arsitek, telp_kantor)
values (2, 'budi kurniawan', '2222222222');

insert all
into arsitek
(id_arsitek, nama_arsitek, telp_kantor)
values (3, 'bambang handoko', '3333333333')
into arsitek
(id_arsitek, nama_arsitek, telp_kantor)
values (4, 'margono', '4444444444')
into arsitek
(id_arsitek, nama_arsitek, telp_kantor)
values (5, 'yudha prasetya', '5555555555')
select 1 from dual;
```



Tugas Resmi

Screenshot

```
SQL> insert into arsitek
  2  (id_arsitek, nama_arsitek, telp_kantor)
  3  values (1, 'santoso', '1111111111');

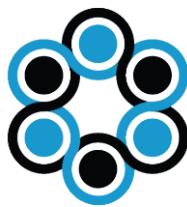
1 row created.

SQL> insert into arsitek
  2  (id_arsitek, nama_arsitek, telp_kantor)
  3  values (2, 'budi kurniawan', '2222222222');

1 row created.

SQL>
SQL> insert all
  2  into arsitek
  3  (id_arsitek, nama_arsitek, telp_kantor)
  4  values (3, 'bambang handoko', '3333333333')
  5  into arsitek
  6  (id_arsitek, nama_arsitek, telp_kantor)
  7  values (4, 'margono', '4444444444')
  8  into arsitek
  9  (id_arsitek, nama_arsitek, telp_kantor)
10  values (5, 'yudha prasetya', '5555555555')
11  select 1 from dual;

3 rows created.
```

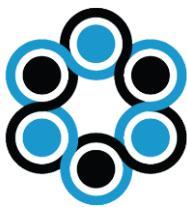


Tugas Resmi

Query

```
insert into proyek
(nama_owner, alamat_proyek,
alamat_penyuratan, no_telp, id_arsitek,
id_marketing, id_profil)
values ('kardono', 'graha family blok-k 62',
'Dusan', '1111111111', 1, 2, 2);
insert into proyek
(nama_owner, alamat_proyek,
alamat_penyuratan, no_telp, id_arsitek,
id_marketing, id_profil)
values ('sudarmono', 'graha family blok-h
18', 'graha family blok-h 18', '222222222',
5, 1, 1);

insert all
into proyek
(nama_owner, alamat_proyek,
alamat_penyuratan, no_telp, id_arsitek,
id_marketing, id_profil)
values ('yunus', 'BDG N-45', 'BK',
'333333333', 2, 1, 1)
into proyek
(nama_owner, alamat_proyek,
alamat_penyuratan, no_telp, id_arsitek,
id_marketing, id_profil)
values ('alex', 'Citraland blok-i4 20',
'embong malang', '444444444', 1, 2, 1)
into proyek
(nama_owner, alamat_proyek,
alamat_penyuratan, no_telp, id_arsitek,
id_marketing, id_profil)
values ('yos tanoyo', 'Citraland blok-h2 1',
'embong malang', '555555555', 1, 2, 2)
select 1 from dual;
```



Tugas Resmi

Screenshot

```
SQL> insert into proyek
  2  (nama_owner, alamat_proyek, alamat_penyuratkan, no_telp, id_arsitek, id_marketing, id_profil)
  3  values ('kardono', 'graha family blok-k 62', 'Dusan', '111111111', 1, 2, 2);
1 row created.

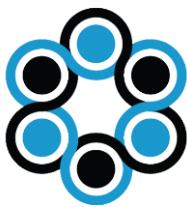
SQL> insert into proyek
  2  (nama_owner, alamat_proyek, alamat_penyuratkan, no_telp, id_arsitek, id_marketing, id_profil)
  3  values ('sudarmono', 'graha family blok-h 18', 'graha family blok-h 18', '222222222', 5, 1, 1);
1 row created.

SQL>
SQL> insert all
  2  into proyek
  3  (nama_owner, alamat_proyek, alamat_penyuratkan, no_telp, id_arsitek, id_marketing, id_profil)
  4  values ('yunus', 'BDG N-45', 'BK', '333333333', 2, 1, 1)
  5  into proyek
  6  (nama_owner, alamat_proyek, alamat_penyuratkan, no_telp, id_arsitek, id_marketing, id_profil)
  7  values ('alex', 'Citraland blok-i4 20', 'embong malang', '444444444', 1, 2, 1)
  8  into proyek
  9  (nama_owner, alamat_proyek, alamat_penyuratkan, no_telp, id_arsitek, id_marketing, id_profil)
 10  values ('yos tanoyo', 'Citraland blok-h2 1', 'embong malang', '555555555', 1, 2, 2)
 11  select 1 from dual;

3 rows created.
```

Analisa :

Saat kita memasukan data yang bersifat multiple atau yang biasa kita sebut dengan multiple insert yang didalam sebuah table terdapat sebuah sequence, maka kita tidak bisa memasukan data seperti multiple insert yang semestinya. Oleh karena itu kita di haruskan membuat sebuah function yang dapat mempermudah kita saat pengisian kolom sequence anda bisa lihat pada saat saya memasukan data di table marketing.



Tugas Resmi

Langkah Kedua

Pada saat kita update semua data pada suatu kolom pada table kita bisa menggunakan perintah

**Update from nama_table set nama_kolom = 'isi_data_terbaru' where
isi_table_lama operasional_kondisi kondisi;**

Dalam penerapan where clause kita bisa menggunakan cara seperti ini :

Update from nama_table set nama_kolom =

Penerapan LIKE adalah seperti ini

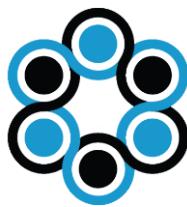
**Update from nama_table set nama_kolom = 'isi_data_baru' where
nama_kolom like '%y%';**

Penerapan dalam AND, OR dan NOT dan minimal 2 operator di dalam satu baris query

**Update from nama_table set nama_kolom = 'isi_data_baru' where
nama_kolom = 'isi_data_lama' operator_logika nama_kolom
operator_kondisi 'isi_data_lama'**

Query

```
update profil_aluminium set nama_profil =  
'penta 50' , warna = 'antico brown' where  
id_profil = 4;  
  
update profil_aluminium set nama_profil =  
'penta 50' , warna = 'black sahara' where  
id_profil = 5;  
  
update marketing set no_telp = '999999999'  
where nama_marketing = 'france';  
  
update arsitek set telp_kantor = '999999999'  
where nama_arsitek like '%y%';  
  
update proyek  
set alamat_penyuratan = 'Dusan'  
where nama_owner = 'yos tanoyo'  
AND id_marketing = 2  
OR alamat_proyek = 'Citraland blok-h2 1';
```



Tugas Resmi

Screenshot

```
SQL> update profil_aluminium set nama_profil = 'penta 50' , warna = 'antico brown' where id_profil = 4;
1 row updated.

SQL> update profil_aluminium set nama_profil = 'penta 50' , warna = 'black sahara' where id_profil = 5;
1 row updated.

SQL> select * from profil_aluminium;
select * from profil_aluminium
*
ERROR at line 1:
ORA-00942: table or view does not exist

SQL> select * from profil_aluminium;
ID_PROFIL NAMA_PROFIL          WARNA
-----  -----
1 strandart           iron grey
2 standart            white
3 standart            black sahara
4 penta 50             antico brown
5 penta 50             black sahara

SQL> update marketing set no_telp = '999999999' where nama_marketing = 'france';
1 row updated.

SQL> select * from marketing;
ID_MARKETING NAMA_MARKETING      NO_TELP
-----  -----
4 france              999999999
5 yudha               567891234
1 rahmad              123456789
2 artani               234567891
3 hendro              345678912

SQL> update arsitek set telp_kantor = '999999999' where nama_arsitek like '%y%';
1 row updated.

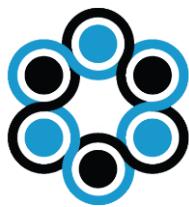
SQL> select * from arsitek;
ID_ARSITEK NAMA_ARSITEK          TELP_KANTOR
-----  -----
1 santoso              111111111
2 budi kurniawan        222222222
3 bambang handoko       333333333
4 margono               444444444
5 yudha prasetya        999999999

SQL> update proyek set alamat_penyurat = 'Dusan' where nama_owner = 'yos tanoyo' AND id_marketing = 2 OR alamat_proyek = 'Citraland blok-h2 1';
1 row updated.

SQL> select * from proyek;
NAMA_OWNER ID_PROFIL          ALAMAT_PROYEK          ALAMAT_PENYURATAN      NO_TELP ID_ARSITEK ID_MARKETING
-----  -----
kardono     2                  graha family blok-k 62   Dusan                 111111111   1       2
sudarmono   1                  graha family blok-h 18   graha family blok-h 18  222222222   5       1
yunus      1                  Bdg N-45                BK                   333333333   2       1
alex        1                  Citraland blok-i4 20   embong malang         444444444   1       2
yos tanoyo  2                  Citraland blok-h2 1      Dusan                 555555555   1       2
```

Analisa :

Untuk mengurangi kesalahan pada sebuah query langkah baiknya kita harus melihat isi pada table yang dituju. Dalam penggunaan operator logika atau pun operator kondisi sebaiknya di pahami terlebih dahulu apa yang kita inginkan supaya mempermudah kita untuk membuat sebuah query yang memiliki banyak operator



Tugas Resmi

Langkah Ketiga

Cara menghapus 3 data pada salah satu table dengan query

**Delete from nama_table where nama_kolom operator_kondisi 'isi_data'
operator_logika nama_kolom operator_kondisi 'isi_data'**

Query

```
delete from marketing
where nama_marketing != 'rahmad'
AND nama_marketing != 'artani'
OR id_marketing > 2;
```

Screenshot

```
SQL> delete from marketing
  2  where nama_marketing != 'rahmad'
  3  AND nama_marketing != 'artani'
  4  OR id_marketing > 2;

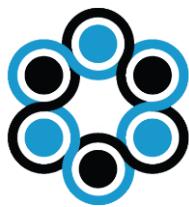
3 rows deleted.

SQL> select * from marketing;

ID_MARKETING NAMA_MARKETING          NO_TELP
-----  -----
      1 rahmad                      123456789
      2 artani                      234567891
```

Analisa :

Dalam pembuatan query untuk delete banyak data, kita harus benar-benar paham dengan kondisi apa yang kita pakai supaya menghindari apa yang tidak kita inginkan



Tugas Resmi

Langkah Keempat

untuk kita membuat savepoint kita bisa menggunakan cara berikut :

Savepoint nama_savepoint;

Saat kita akan mengulangi dari savepoint saat sudah terjadi perubahan dengan cara berikut :

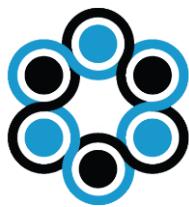
Rollback to savepoint nama_savepoint;

Untuk kita men-commit bisa menggunakan cara ini

Commit;

Query

```
savepoint pertama;
rollback to savepoint pertama;
commit;
```



Tugas Resmi

Screenshot

```
SQL> select * from marketing;
ID_MARKETING NAMA_MARKETING          NO_TELP
-----  -----
1 rahmad           123456789
2 artani          234567891

SQL> savepoint pertama;
Savepoint created.

SQL> update marketing
  2 set no_telp = '111111111'
  3 where id_marketing = 1;

1 row updated.

SQL> select * from marketing;
ID_MARKETING NAMA_MARKETING          NO_TELP
-----  -----
1 rahmad           111111111
2 artani          234567891

SQL> rollback savepoint pertama;
rollback savepoint pertama
*
ERROR at line 1:
ORA-02181: invalid option to ROLLBACK WORK

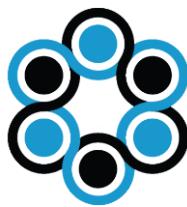
SQL> rollback to savepoint pertama;
Rollback complete.

SQL> select * from marketing;
ID_MARKETING NAMA_MARKETING          NO_TELP
-----  -----
1 rahmad           123456789
2 artani          234567891

SQL> commit;
Commit complete.
```

Analisa :

Kegunaan savepoint adalah menentukan sebuah titik aman, jika pada suatu saat kita melakukan kesalahan saat membuat query dan bertambah berubahnya sebuah data maka kita bisa menuju titik aman yang sudah kita buat dengan savepoint. Mengulangi ke titik aman kita menggunakan sebuah query rollback dan untuk commit adalah mem-fixkan sebuah data



Tugas Resmi

Langkah Kelima

Cara menggunakan order by dan group by

Query

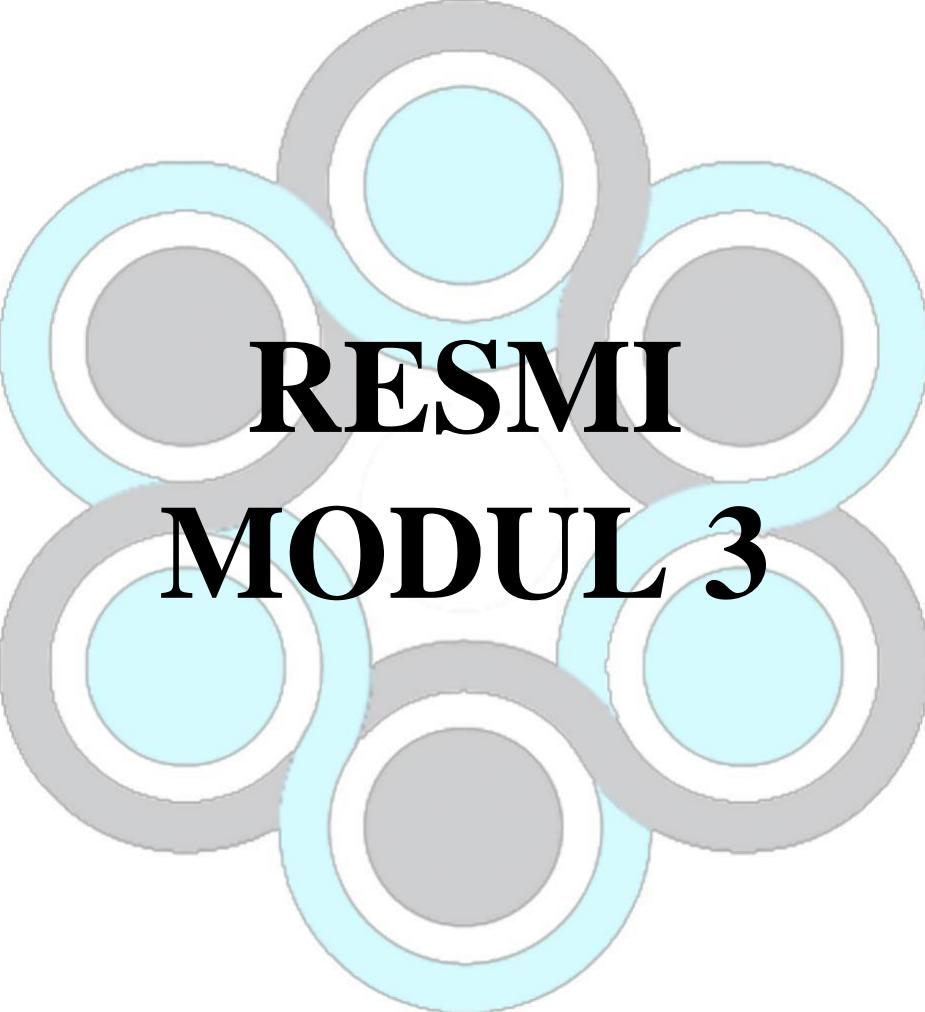
```
select nama_owner, alamat_penyuratan from  
proyek order by id_marketing;  
  
select warna, count(nama_profil) AS "banyak"  
from profil_aluminium  
group by warna;
```

Screenshot

```
SQL> select nama_owner, alamat_penyuratan from proyek order by id_marketing;  
NAMA_OWNER          ALAMAT_PENYURATAN  
-----  
yunus                BK  
sudarmono             graha family blok-h 18  
yos tanoyo            Dusan  
alex                  embong malang  
kardono               Dusan  
  
SQL> select warna, count(nama_profil) AS "banyak"  
  2  from profil_aluminium  
  3  group by warna;  
WARNA                 banyak  
-----  
black sahara           2  
white                  1  
antico brown           1  
iron grey               1
```

Analisa :

Order by berguna untuk melakukan request ke tabel yang dituju.
Group by berguna untuk mengelompokan sebuah data pada tabel

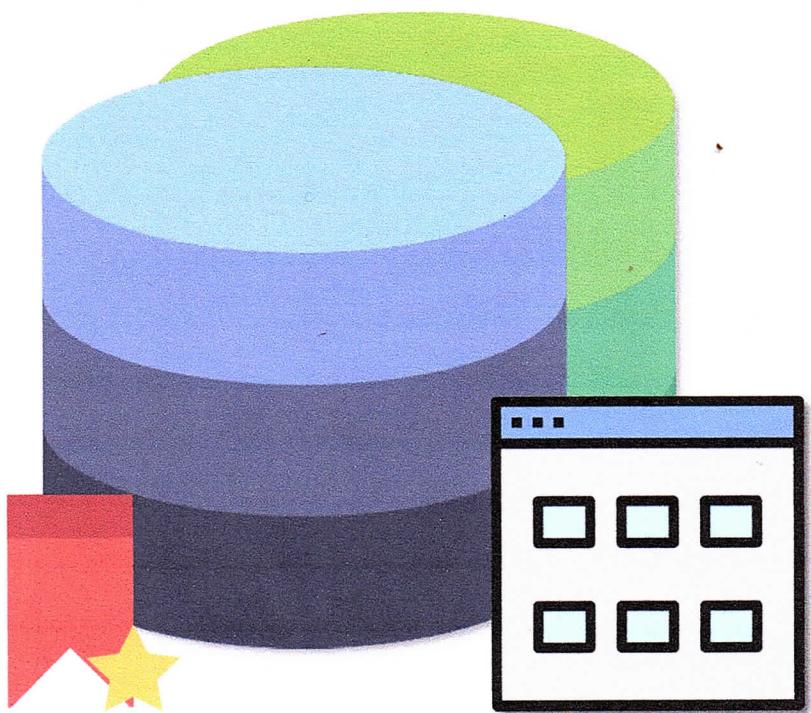


RESMI

MODUL 3

FAKULTAS TEKNIK ELEKTRO DAN
TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI ADHI TAMA
SURABAYA
2020

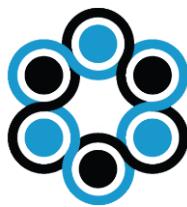
LAPORAN RESMI



PRAKTIKUM BASIS DATA PERIODE V - Tahun 2019/2020

NAMA : Mochamad Khoirul Huda
NPM : 06.2010.1.06990
MODUL : III

--



Tugas Resmi

Soal Resmi

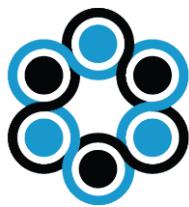
1. Insert minimal 5 data pada setiap tabel
2. Buatlah query dengan menerapkan fungsi aggregate :
 - a. Max dan Min dalam 1 Query
 - b. Count, 1 klausa dan 2 operator dalam 1 Query
 - c. Sum dalam 1 Query
 - d. Average pada soal 2C diatas
3. Buatlah Subquery dengan menerapkan :
 - a. Max dan Min
 - b. Count, 1 klausa dan 2 operator
 - c. Sum
 - d. Average
 - e. Group by
 - f. Select setelah where
 - g. Minimal 3 nested query

Langkah Pertama

insert data terlebih dahulu pada setiap tabel bisa menggunakan single insert atau pun multiple insert

Query

```
insert into profil_aluminium
(id_profil, nama_profil, warna)
values (1, 'strandart', 'iron grey');
insert into profil_aluminium
(id_profil, nama_profil, warna)
values (2, 'standart', 'white');
insert all
into profil_aluminium (id_profil,
nama_profil, warna)
values (3, 'standart', 'black sahara')
into profil_aluminium (id_profil,
nama_profil, warna)
values (4, 'penta 40', 'iron grey')
into profil_aluminium (id_profil,
nama_profil, warna)
values (5, 'penta 40', 'white')
select 1 from dual;
```



Tugas Resmi

Screenshot

```
SQL> insert into profil_aluminium
  2  (id_profil, nama_profil, warna)
  3  values (1, 'strandart', 'iron grey');

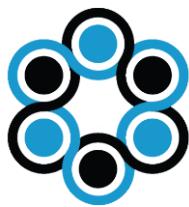
1 row created.

SQL> insert into profil_aluminium
  2  (id_profil, nama_profil, warna)
  3  values (2, 'standart', 'white');

1 row created.

SQL>
SQL> insert all
  2  into profil_aluminium (id_profil, nama_profil, warna)
  3  values (3, 'standart', 'black sahara')
  4  into profil_aluminium (id_profil, nama_profil, warna)
  5  values (4, 'penta 40', 'iron grey')
  6  into profil_aluminium (id_profil, nama_profil, warna)
  7  values (5, 'penta 40', 'white')
  8  select 1 from dual;

3 rows created.
```



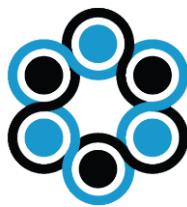
Tugas Resmi

Query

```
insert into marketing
(id_marketing, nama_marketing, no_telp)
values (id_marketing.nextval, 'rahmad',
'123456789');
insert into marketing
(id_marketing, nama_marketing, no_telp)
values (id_marketing.nextval, 'artani',
'234567891');

CREATE FUNCTION get_seq RETURN NUMBER IS
BEGIN
    RETURN id_marketing.nextval;
END;
/

insert all
into marketing (id_marketing, nama_marketing,
no_telp)
values (get_seq, 'hendro', '345678912')
into marketing
(id_marketing, nama_marketing, no_telp)
values (get_seq, 'france', '456789123')
into marketing (id_marketing, nama_marketing,
no_telp)
values (get_seq, 'yudha', '567891234')
select 1 from dual;
```



Tugas Resmi

Screenshot

```
SQL> insert into marketing
  2  (id_marketing, nama_marketing, no_telp)
  3  values (id_marketing.nextval, 'rahmad', '123456789');

1 row created.

SQL> insert into marketing
  2  (id_marketing, nama_marketing, no_telp)
  3  values (id_marketing.nextval, 'artani', '234567891');

1 row created.

SQL>
SQL> CREATE FUNCTION get_seq RETURN NUMBER IS
  2  BEGIN
  3    RETURN id_marketing.nextval;
  4  END;
  5  /

Function created.

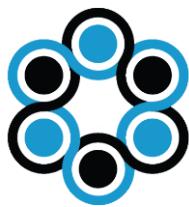
SQL> insert all
  2  into marketing (id_marketing, nama_marketing, no_telp)
  3  values (get_seq, 'hendro', '345678912')
  4  into marketing
  5  (id_marketing, nama_marketing, no_telp)
  6  values (get_seq, 'france', '456789123')
  7  into marketing (id_marketing, nama_marketing, no_telp)
  8  values (get_seq, 'yudha', '567891234')
  9  select 1 from dual;

3 rows created.
```

Query

```
insert into arsitek
(id_arsitek, nama_arsitek, telp_kantor)
values (1, 'santoso', '111111111');
insert into arsitek
(id_arsitek, nama_arsitek, telp_kantor)
values (2, 'budi kurniawan', '222222222');

insert all
into arsitek
(id_arsitek, nama_arsitek, telp_kantor)
values (3, 'bambang handoko', '333333333')
into arsitek
(id_arsitek, nama_arsitek, telp_kantor)
values (4, 'margono', '444444444')
into arsitek
(id_arsitek, nama_arsitek, telp_kantor)
values (5, 'yudha prasetya', '555555555')
select 1 from dual;
```



Tugas Resmi

Screenshot

```
SQL> insert into arsitek
  2 (id_arsitek, nama_arsitek, telp_kantor)
  3 values (1, 'santoso', '1111111111');

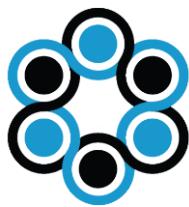
1 row created.

SQL> insert into arsitek
  2 (id_arsitek, nama_arsitek, telp_kantor)
  3 values (2, 'budi kurniawan', '2222222222');

1 row created.

SQL>
SQL> insert all
  2 into arsitek
  3 (id_arsitek, nama_arsitek, telp_kantor)
  4 values (3, 'bambang handoko', '3333333333')
  5 into arsitek
  6 (id_arsitek, nama_arsitek, telp_kantor)
  7 values (4, 'margono', '4444444444')
  8 into arsitek
  9 (id_arsitek, nama_arsitek, telp_kantor)
 10 values (5, 'yudha prasetya', '5555555555')
 11 select 1 from dual;

3 rows created.
```



Tugas Resmi

Query

```
insert into proyek
(nama_owner, alamat_proyek,
alamat_penyuratan, no_telp, id_arsitek,
id_marketing, id_profil)
values ('kardono', 'graha family blok-k 62',
'Dusan', '1111111111', 1, 2, 2);
insert into proyek
(nama_owner, alamat_proyek,
alamat_penyuratan, no_telp, id_arsitek,
id_marketing, id_profil)
values ('sudarmono', 'graha family blok-h
18', 'graha family blok-h 18', '222222222',
5, 3, 1);

insert all
into proyek
(nama_owner, alamat_proyek,
alamat_penyuratan, no_telp, id_arsitek,
id_marketing, id_profil)
values ('yunus', 'BDG N-45', 'BK',
'333333333', 2, 3, 1)
into proyek
(nama_owner, alamat_proyek,
alamat_penyuratan, no_telp, id_arsitek,
id_marketing, id_profil)
values ('alex', 'Citraland blok-i4 20',
'embong malang', '4444444444', 1, 2, 1)
into proyek
(nama_owner, alamat_proyek,
alamat_penyuratan, no_telp, id_arsitek,
id_marketing, id_profil)
values ('yos tanoyo', 'Citraland blok-h2 1',
'embong malang', '5555555555', 1, 2, 2)
select 1 from dual;
```



Tugas Resmi

Screenshot

```
SQL> insert into arsitek
  2 (id_arsitek, nama_arsitek, telp_kantor)
  3 values (1, 'santoso', '1111111111');

1 row created.

SQL> insert into arsitek
  2 (id_arsitek, nama_arsitek, telp_kantor)
  3 values (2, 'budi kurniawan', '2222222222');

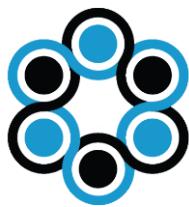
1 row created.

SQL>
SQL> insert all
  2 into arsitek
  3 (id_arsitek, nama_arsitek, telp_kantor)
  4 values (3, 'bambang handoko', '3333333333')
  5 into arsitek
  6 (id_arsitek, nama_arsitek, telp_kantor)
  7 values (4, 'margono', '4444444444')
  8 into arsitek
  9 (id_arsitek, nama_arsitek, telp_kantor)
 10 values (5, 'yudha prasetya', '5555555555')
 11 select 1 from dual;

3 rows created.
```

Analisa

Yang perlu di perhatikan dalam pengisian value pada table adalah tipe data colom apakah sudah sesuai dengan tipe data yang mau kita isikan. Kalian harus benar paham apakah colom tersebut mengandung nilai UNIX, jika iya maka nilai baris dari kolom tersebut tidak boleh sama dengan nilai baris lain dengan kolom yang sama



Tugas Resmi

Langkah Kedua

penerapan fungsi agregasi :

Max (nama_kolom) as “nama_alias” from nama_table
Min (nama_kolom) as “nama_alias” from nama_table
Avg (nama_kolom) as “nama_alias” from nama_table
Count (nama_kolom) as “nama_alias” from nama_table
Sum (nama_kolom) as “nama_alias” from nama_table

Query

```
select nama_owner,
       max (id_marketing) as "terbanyak",
       min (id_marketing) as "terendah"
  from proyek
 group by nama_owner;
```

Screenshot

```
SQL> select nama_owner,
  2  max (id_marketing) as "terbanyak",
  3  min (id_marketing) as "terendah"
  4  from proyek
  5  group by nama_owner;

NAMA_OWNER          terbanyak    terendah
-----              -----
sudarmono                  3          3
yunus                      3          3
alex                        2          2
kardono                     2          2
yos tanoyo                  2          2

SQL>
```



Tugas Resmi

Query

```
select warna, count(nama_profil) AS "banyak"
from profil_aluminium
where nama_profil = 'standart'
and warna != 'white'
group by warna;
```

Screenshot

```
SQL> select * from profil_aluminium;
ID_PROFIL NAMA_PROFIL          WARNA
-----  -----
1 strandart           iron grey
2 standart            white
3 standart            black sahara
4 penta 40             iron grey
5 penta 40             white

SQL> select warna, count(nama_profil) AS "banyak"
2  from profil_aluminium
3  where nama_profil = 'standart'
4  and warna != 'white'
5  group by warna;

WARNA          banyak
-----  -----
black sahara      1    . .

SQL>
```

Query

```
select id_arsitek,
sum (id_profil) as "jumlah"
from proyek
group by id_arsitek;
```

Screenshot

```
SQL> select id_arsitek,
2  sum (id_profil) as "jumlah"
3  from proyek
4  group by id_arsitek;

ID_ARSITEK      jumlah
-----  -----
1                  5
2                  1
5                  1

SQL>
```



Tugas Resmi

Query

```
select id_arsitek,
       avg (id_profil) as "rata2"
     from proyek
   group by id_arsitek;
```

Screenshot

```
SQL> select id_arsitek,
  2  avg (id_profil) as "rata2"
  3  from proyek
  4  group by id_arsitek;

ID_ARSITEK      rata2
----- -----
      1  1,66666667
      2          1
      5          1

SQL>
```

Analisa :

Arti dalam fungsi agregasi :

Max = mencari nilai maksimal dari suatu kolom

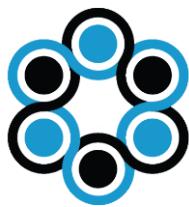
Min = mencari nilai minimal dari suatu kolom

Avg = mencari nilai rata-rata dari suatu kolom

Sum = mencari nilai total penjumlahan dari suatu kolom

Count = menghitung banyaknya kolom itu terisi.

Dalam penerapan fungsi agregasi yang perlu di perhatikan adalah apakah suatu kolom tersebut mempunyai nilai yang bisa untuk digunakan fungsi agregasi



Tugas Resmi

Langkah Ketiga

Menggunakan perintah sub query

Query

```
select nama_owner from proyek
where id_marketing != (select
min(id_marketing) as minimal from proyek)
and id_marketing = (select max(id_marketing)
as maksimal from proyek)
group by nama_owner;
```

Screenshot

```
SQL> select nama_owner from proyek
  2 where id_marketing != (select min(id_marketing) as minimal from proyek)
  3 and id_marketing = (select max(id_marketing) as maksimal from proyek)
  4 group by nama_owner;

NAMA_OWNER
-----
sudarmono
yunus
```

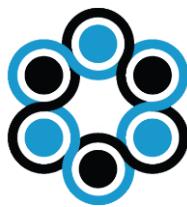
Query

```
select warna, count(nama_profil) AS "banyak"
from profil_aluminium
where warna = any (select warna from
profil_aluminium
where warna = any (select warna from
profil_aluminium
where warna !='white'))
group by warna;
```

Screenshot

```
SQL> select warna, count(nama_profil) AS "banyak"
  2 from profil_aluminium
  3 where warna = any (select warna from profil_aluminium
  4 where warna = any (select warna from profil_aluminium
  5 where warna !='white'))
  6 group by warna;

WARNA                      banyak
-----
black sahara                  1
iron grey                      2
```



Tugas Resmi

Query

```
select id_profil,rata2 from proyek,
(select sum (rata) as rata2 from
(select avg(id_profil) as rata from proyek
group by id_profil));
```

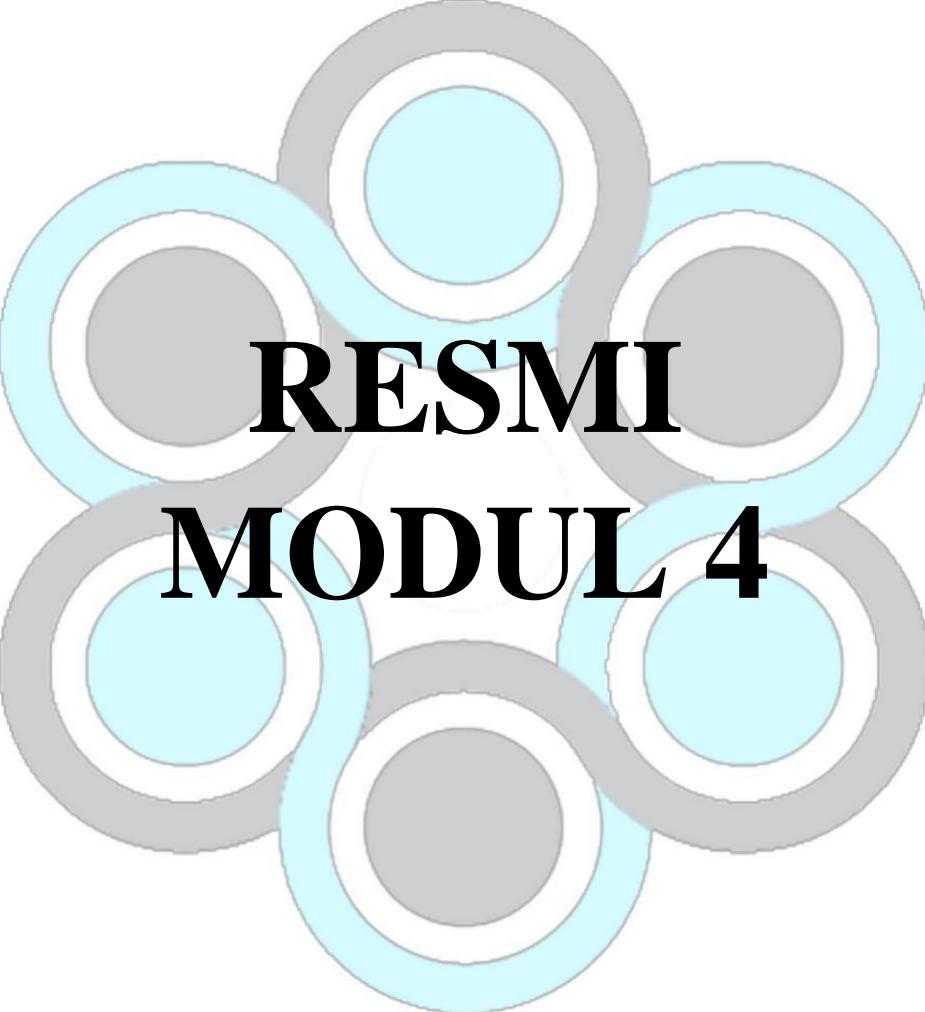
Screenshot

```
SQL> select id_profil,rata2 from proyek,
2  (select sum (rata) as rata2 from
3  (select avg(id_profil) as rata from proyek
4  group by id_profil));

ID_PROFIL      RATA2
-----  -----
          2          3
          1          3
          1          3
          1          3
          2          3
```

Analisa :

Sub query adalah suatu query yang ada di dalam query. Disini kalian bisa lihat penggunaan nama alias yang di beri tanda (“) tidak akan bisa dipanggil dalam select atau pun parameter fungsi agregasi, dikarenakan penamaan alias yang di beri tanda (“) hanya akan di tampilkan saja pada cmd. Jika kita tidak memberi tanda (“) maka nama alias tersebut bisa digunakan sebagai parameter atau bisa dibilang nama aliasnya bersifat temporari dalam satu kali sub query tersebut

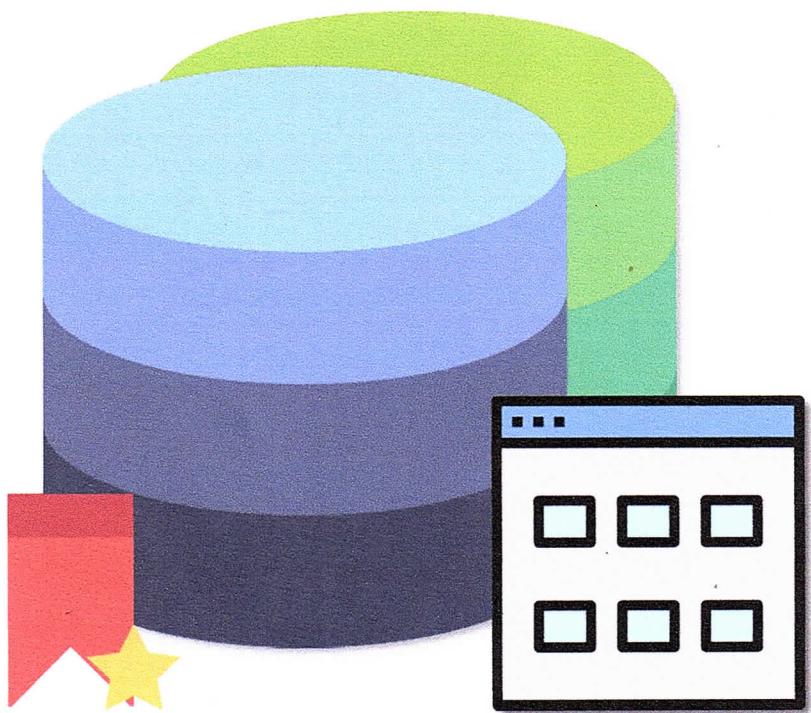


RESMI

MODUL 4

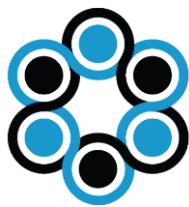
FAKULTAS TEKNIK ELEKTRO DAN
TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI ADHI TAMA
SURABAYA
2020

LAPORAN RESMI



PRAKTIKUM BASIS DATA PERIODE V - Tahun 2019/2020

NAMA : Mechamad Khoirul H
NPM : 06.2018.1.06gg0
MODUL : W



Tugas Resmi

Soal Resmi

1. Insert 5 data pada setiap tabel
2. Buatlah query laporan dengan jelas sesuai proyek Anda dengan ketentuan sebagai berikut :
 - a. Equi Join minimal 2 tabel dengan menerapkan Operator Pembanding dan Klausua
 - b. Left Join minimal 3 tabel dengan menerapkan Klausua dan Subquery
 - c. Right Join minimal 3 tabel dengan menerapkan Aggregate dan Operator Logika

Note : Tampilkan kolom pada tabel pertama dan kolom pada tabel kedua dengan memperhatikan kasus yang akan Anda laporakan.

3. Buatlah view dengan menerapkan DML.
4. Buatlah view dari masing-masing soal pada nomor 2.
5. Terapkan DML pada View pada nomor 4

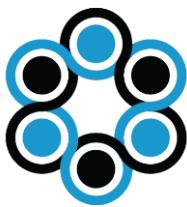
Langkah Pertama

Insert 5 data pada setiap tabel yang sudah anda buat.

Query

```
insert into profil_aluminium
(id_profil, nama_profil, warna)
values (1, 'strandart', 'iron grey');
insert into profil_aluminium
(id_profil, nama_profil, warna)
values (2, 'standart', 'white');

insert all
into profil_aluminium (id_profil,
nama_profil, warna)
values (3, 'standart', 'black sahara')
into profil_aluminium (id_profil,
nama_profil, warna)
values (4, 'penta 40', 'iron grey')
into profil_aluminium (id_profil,
nama_profil, warna)
values (5, 'penta 40', 'white')
select 1 from dual;
```



Tugas Resmi

Screenshot

```
SQL> insert into profil_aluminium (id_profil, nama_profil, warna) values (1, 'strandart', 'iron grey');
1 row created.

SQL> insert into profil_aluminium (id_profil, nama_profil, warna) values (2, 'standart', 'white');
1 row created.
```

```
SQL> insert all
  2  into profil_aluminium (id_profil, nama_profil, warna) values (3, 'standart', 'black sahara')
  3  into profil_aluminium (id_profil, nama_profil, warna) values (4, 'penta 40', 'iron grey')
  4  into profil_aluminium (id_profil, nama_profil, warna) values (5, 'penta 40', 'white')
  5  select 1 from dual;

3 rows created.
```

Query

```
insert into marketing (id_marketing, nama_marketing,
no_telp) values (id_marketing.nextval, 'rahmad',
'123456789');
insert into marketing (id_marketing, nama_marketing,
no_telp) values (id_marketing.nextval, 'artani',
'234567891');

insert into marketing (id_marketing, nama_marketing,
no_telp) values (id_marketing.nextval, 'hendro',
'345678912');
insert into marketing (id_marketing, nama_marketing,
no_telp) values (id_marketing.nextval, 'france',
'456789123');
insert into marketing (id_marketing, nama_marketing,
no_telp) values (id_marketing.nextval, 'yudha',
'567891234');
```

Screenshot

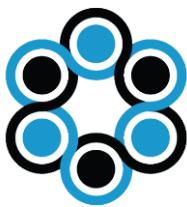
```
SQL> insert into marketing (id_marketing, nama_marketing, no_telp) values (id_marketing.nextval, 'rahmad', '123456789');
1 row created.

SQL> insert into marketing (id_marketing, nama_marketing, no_telp) values (id_marketing.nextval, 'artani', '234567891');
1 row created.
```

```
SQL> insert into marketing (id_marketing, nama_marketing, no_telp) values (id_marketing.nextval, 'hendro', '345678912');
1 row created.

SQL> insert into marketing (id_marketing, nama_marketing, no_telp) values (id_marketing.nextval, 'france', '456789123');
1 row created.

SQL> insert into marketing (id_marketing, nama_marketing, no_telp) values (id_marketing.nextval, 'yudha', '567891234');
1 row created.
```



Tugas Resmi

Query

```
insert into arsitek (id_arsitek, nama_arsitek,
telp_kantor) values (1, 'santoso', '1111111111');
insert into arsitek (id_arsitek, nama_arsitek,
telp_kantor) values (2, 'budi kurniawan',
'2222222222');

insert all
into arsitek (id_arsitek, nama_arsitek, telp_kantor)
values (3, 'bambang handoko', '3333333333')
into arsitek (id_arsitek, nama_arsitek, telp_kantor)
values (4, 'margono', '4444444444')
into arsitek (id_arsitek, nama_arsitek, telp_kantor)
values (5, 'yudha prasetya', '5555555555')
select 1 from dual;
```

Screenshot

```
SQL> insert into arsitek (id_arsitek, nama_arsitek, telp_kantor) values (1, 'santoso', '1111111111');
1 row created.

SQL> insert into arsitek (id_arsitek, nama_arsitek, telp_kantor) values (2, 'budi kurniawan', '2222222222');
1 row created.

SQL>
SQL> insert all
  2 into arsitek (id_arsitek, nama_arsitek, telp_kantor) values (3, 'bambang handoko', '3333333333')
  3 into arsitek (id_arsitek, nama_arsitek, telp_kantor) values (4, 'margono', '4444444444')
  4 into arsitek (id_arsitek, nama_arsitek, telp_kantor) values (5, 'yudha prasetya', '5555555555')
  5 select 1 from dual;

3 rows created.
```

Query

```
insert into proyek (alamat_proyek,
nama_owner, alamat_penyuratian, no_telp,
id_arsitek, id_marketing, id_profil)
values ('graha family blok-k 62', 'kardono',
'Dusan', '1111111111', 1, 3, 2);
insert into proyek (alamat_proyek,
nama_owner, alamat_penyuratian, no_telp,
id_arsitek, id_marketing, id_profil)
values ('graha family blok-h 18',
'sudarmono', 'graha family blok-h 18',
'2222222222', 5, 2, 1);
insert into proyek (alamat_proyek,
nama_owner, alamat_penyuratian, no_telp,
```



Tugas Resmi

Screenshot

```
SQL> insert into proyek (alamat_proyek, nama_owner, alamat_penyuratan, no_telp, id_arsitek, id_marketing, id_profil)
  2  values ('graha family blok-k 62', 'kardono', 'Dusan', '111111111', 1, 3, 2);
1 row created.

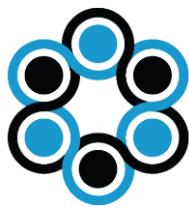
SQL> insert into proyek (alamat_proyek, nama_owner, alamat_penyuratan, no_telp, id_arsitek, id_marketing, id_profil)
  2  values ('graha family blok-h 18', 'sudarmono', 'graha family blok-h 18', '222222222', 5, 2, 1);
1 row created.

SQL> insert into proyek (alamat_proyek, nama_owner, alamat_penyuratan, no_telp, id_arsitek, id_marketing, id_profil) values ('BDG N-45', 'yunus', 'BK', '333333333', 2, 1, 1);
1 row created.

SQL> insert into proyek (alamat_proyek, nama_owner, alamat_penyuratan, no_telp, id_arsitek, id_marketing, id_profil) values ('Citraland blok-i4 20', 'alex', 'embong malang', '444444444', 1, 3, 1);
1 row created.
```

Analisa

Disini saya menggunakan 2 tipe insert table. Yang pertama adalah insert table secara partial dan yang kedua saya memasukan data secara bersamaan atau biasa kita sebut dengan multiple insert



Tugas Resmi

Langkah Kedua

Kita terpakan beberapa join condition dalam query kita diantaranya Inner join, Left join, dan Right join

Query

```
select m.nama_marketing, py.nama_owner
from proyek py join
marketing m
on py.id_marketing = m.id_marketing
where py.id_marketing = 2;
```

Screenshot

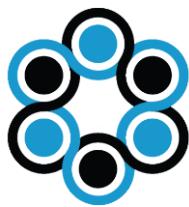
```
SQL> select m.nama_marketing, py.nama_owner
  2  from proyek py join
  3  marketing m
  4  on py.id_marketing = m.id_marketing
  5  where py.id_marketing = 2;

NAMA_MARKETING          NAMA_OWNER
-----  -----
rahmad                  kardono
rahmad                  alex
rahmad                  yos tanoyo

SQL>
```

Query

```
select m.nama_marketing, py.nama_owner,
a.nama_arsitek
from proyek py left join
marketing m
on py.id_marketing = m.id_marketing
left join arsitek a
on py.id_arsitek = a.id_arsitek
where py.id_marketing =
(select id_marketing from marketing
where nama_marketing = 'rahmad')
;
```



Tugas Resmi

Screenshot

```
SQL> select m.nama_marketing, py.nama_owner, a.nama_arsitek
  2  from proyek py left join
  3  marketing m
  4  on py.id_marketing = m.id_marketing
  5  left join arsitek a
  6  on py.id_arsitek = a.id_arsitek
  7  where py.id_marketing =
  8  (select id_marketing from marketing
  9  where nama_marketing = 'rahmad')
 10 ;
```

NAMA_MARKETING	NAMA_OWNER	NAMA_ARSITEK
rahmad	kardono	santoso
rahmad	alex	santoso
rahmad	yos tanoyo	santoso

```
SQL>
```

Query

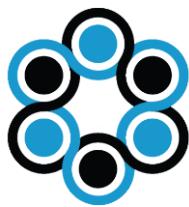
```
select m.nama_marketing, py.nama_owner,
a.nama_arsitek
from proyek py left join
marketing m
on py.id_marketing = m.id_marketing
left join arsitek a
on py.id_arsitek = a.id_arsitek
where py.id_marketing =
(select id_marketing from marketing
where nama_marketing = 'rahmad')
;
```

Screenshot

```
SQL> select m.nama_marketing, py.nama_owner, a.nama_arsitek
  2  from proyek py left join
  3  marketing m
  4  on py.id_marketing = m.id_marketing
  5  left join arsitek a
  6  on py.id_arsitek = a.id_arsitek
  7  where py.id_marketing =
  8  (select id_marketing from marketing
  9  where nama_marketing = 'rahmad')
 10 ;
```

NAMA_MARKETING	NAMA_OWNER	NAMA_ARSITEK
rahmad	kardono	santoso
rahmad	alex	santoso
rahmad	yos tanoyo	santoso

```
SQL>
```



Tugas Resmi

Query

```
select m.nama_marketing, pr.nama_profil,
count(m.nama_marketing) as jumlah
from proyek py
right join profil_aluminium pr
on pr.id_profil = py.id_profil
right join marketing m
on m.id_marketing = py.id_marketing
where pr.nama_profil = 'standart'
and pr.warna = 'iron grey'
group by m.nama_marketing, pr.nama_profil
;
```

Screenshot

```
SQL> select m.nama_marketing, pr.nama_profil,
2 count(m.nama_marketing) as jumlah
3 from proyek py
4 right join profil_aluminium pr
5 on pr.id_profil = py.id_profil
6 right join marketing m
7 on m.id_marketing = py.id_marketing
8 where pr.nama_profil = 'standart'
9 and pr.warna = 'iron grey'
10 group by m.nama_marketing, pr.nama_profil
11 ;

NAMA_MARKETING      NAMA_PROFIL      JUMLAH
-----  -----
rahmad            standart          1
artani            standart          2

SQL>
```

Analisa :

Di sini saya menggunakan Inner join pada query yang pertama. Query yang ke dua saya menerapkan Left join. Right join ada pada query yang ke tiga.



Tugas Resmi

Langkah Ketiga

Pada langkah ketiga ini kita akan menampilkan isi dalam sebuah tabel yang menggunakan perintah DML.

Query

```
select * from arsitek;
select * from marketing;
select * from profil_aluminium;
```

Screenshot

```
SQL> select * from arsitek;
ID_ARSITEK NAMA_ARSITEK          TELP_KANTOR
----- -----
1 santoso           1111111111
2 budi kurniawan   2222222222
3 bambang handoko 3333333333
4 margono          4444444444
5 yudha prasetya  5555555555

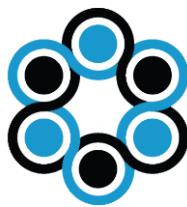
SQL> select * from marketing;
ID_MARKETING NAMA_MARKETING        NO_TELP
----- -----
2 rahmad           123456789
3 artani           234567891
4 hendro          345678912
5 france          456789123
6 yudha           567891234

SQL> select * from profil_aluminium;
ID_PROFIL NAMA_PROFIL            WARNA
----- -----
1 standart         iron grey
2 standart         white
3 standart         black sahara
4 penta 40         iron grey
5 penta 40         white

SQL>
```

Analisa :

Pada langkah ketiga ini saya menampilkan 3 table yang berhubungan dengan table proyek atau setiap id dalam 3 tabel tersebut adalah foreign key untuk beberapa kolom dari tabel proyek



Tugas Resmi

Langkah Keempat

Langkah keempat kita akan membuat sebuah view atau bisa disebut dengan virtual table dari masing-masing query yang ada pada langkah ketiga

Query

```
create view kepemilikan_proyek
as select m.nama_marketing, py.nama_owner
from proyek py join
marketing m
on py.id_marketing = m.id_marketing
where py.id_marketing = 2;
```

Screenshot

```
SQL> create view kepemilikan_proyek
  2  as select m.nama_marketing, py.nama_owner
  3  from proyek py join
  4  marketing m
  5  on py.id_marketing = m.id_marketing
  6  where py.id_marketing = 2;

View created.
```

Query

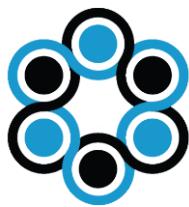
```
create view kepemilikan_proyek_arsitek
as select m.nama_marketing, py.nama_owner,
a.nama_arsitek
from proyek py left join
marketing m
on py.id_marketing = m.id_marketing
left join arsitek a
on py.id_arsitek = a.id_arsitek
where py.id_marketing =
(select id_marketing from marketing
where nama_marketing = 'rahmad');
```

Screenshot

```
SQL> create view kepemilikan_proyek_arsitek
  2  as select m.nama_marketing, py.nama_owner, a.nama_arsitek
  3  from proyek py left join
  4  marketing m
  5  on py.id_marketing = m.id_marketing
  6  left join arsitek a
  7  on py.id_arsitek = a.id_arsitek
  8  where py.id_marketing =
  9  (select id_marketing from marketing
10  where nama_marketing = 'rahmad')
11 ;

View created.

SQL>
```



Tugas Resmi

Query

```
create view jumlah_profil_digunakan
as select m.nama_marketing, pr.nama_profil,
count(m.nama_marketing) as jumlah
from proyek py
right join profil_aluminium pr
on pr.id_profil = py.id_profil
right join marketing m
on m.id_marketing = py.id_marketing
where pr.nama_profil = 'standart'
and pr.warna = 'iron grey'
group by m.nama_marketing, pr.nama_profil
;
```

Screenshot

```
SQL> create view jumlah_profil_digunakan
  2  as select m.nama_marketing, pr.nama_profil,
  3  count(m.nama_marketing) as jumlah
  4  from proyek py
  5  right join profil_aluminium pr
  6  on pr.id_profil = py.id_profil
  7  right join marketing m
  8  on m.id_marketing = py.id_marketing
  9  where pr.nama_profil = 'standart'
 10 and pr.warna = 'iron grey'
 11 group by m.nama_marketing, pr.nama_profil
 12 ;
View created.

SQL>
```

Analisa :

Pada langkah keempat ini kita membuat view



Tugas Resmi

Langkah Kelima

Menampilkan isi dari langkah keempat

Query

```
select * from kepemilikan_proyek;
select * from kepemilikan_proyek_arsitek;
select * from jumlah_profil_digunakan;
```

Screenshot

```
SQL> select * from kepemilikan_proyek;
NAMA_MARKETING      NAMA_OWNER
-----
rahmad              kardono
rahmad              alex
rahmad              yos tanoyo

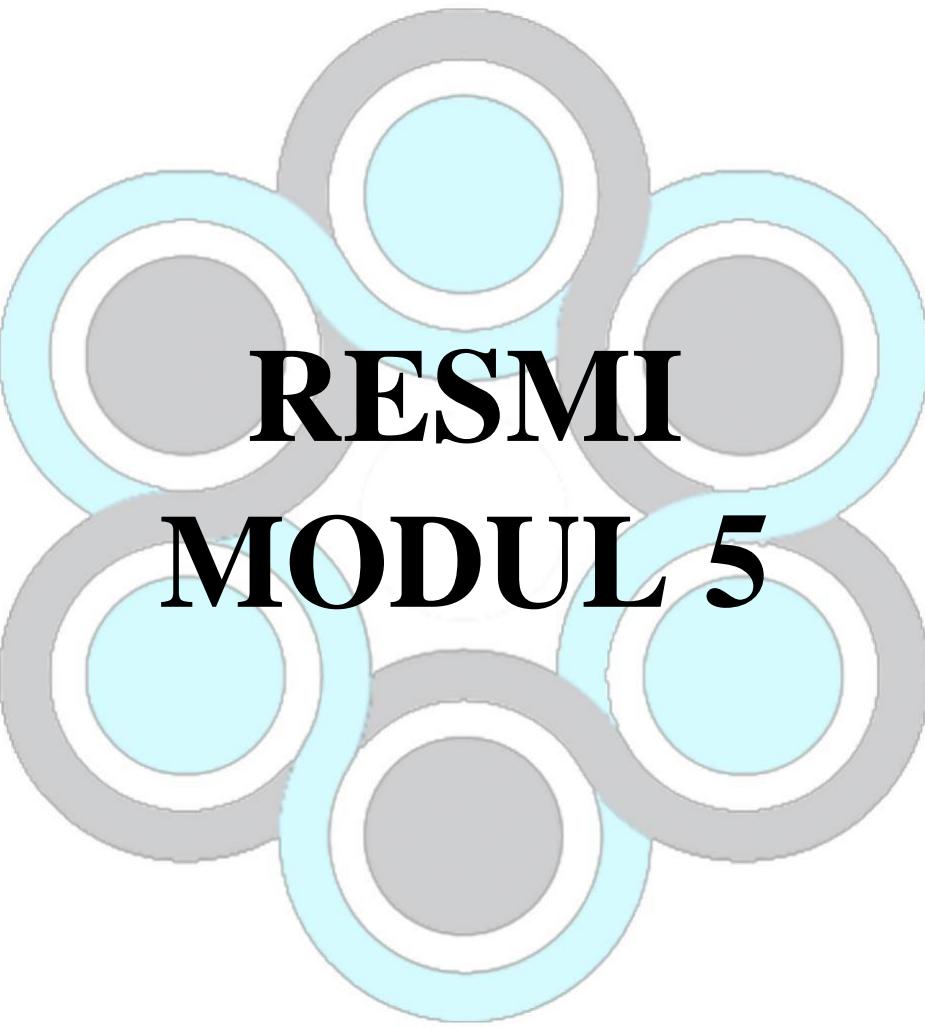
SQL> select * from kepemilikan_proyek_arsitek;
NAMA_MARKETING      NAMA_OWNER      NAMA_ARSITEK
-----
rahmad              kardono        santoso
rahmad              alex          santoso
rahmad              yos tanoyo    santoso

SQL> select * from jumlah_profil_digunakan;
NAMA_MARKETING      NAMA_PROFIL      JUMLAH
-----
rahmad              standart        1
artani              standart        2

SQL>
```

Analisa :

Disini kita akan menampilkan isi dari table virtual yang sudah kita buat pada langkah keempat

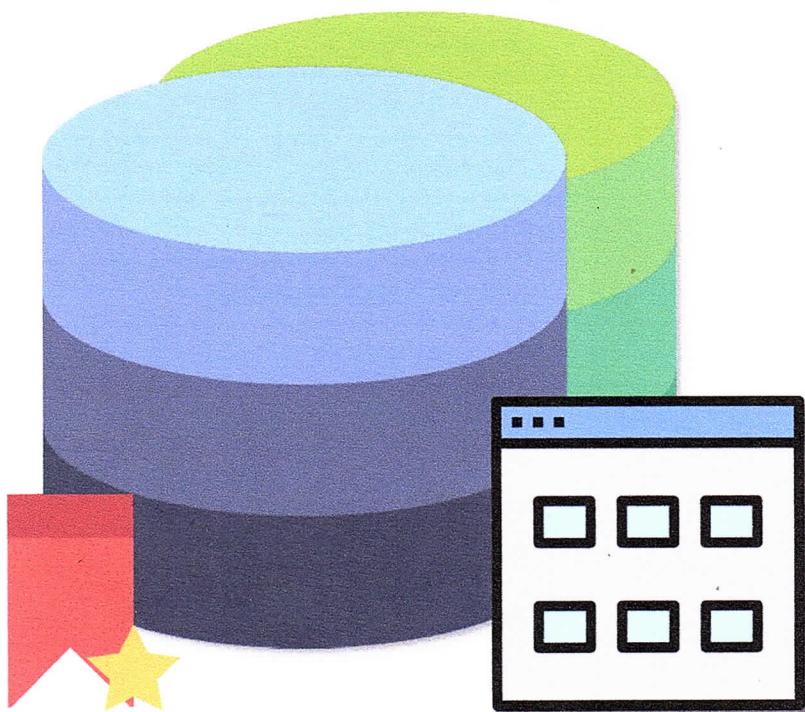


RESMI

MODUL 5

FAKULTAS TEKNIK ELEKTRO DAN
TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI ADHI TAMA
SURABAYA
2020

LAPORAN RESMI



PRAKTIKUM BASIS DATA PERIODE V - Tahun 2019/2020

NAMA : Mechamad Kharul Huda
NPM : 06.2010.1.06990
MODUL : ✓



Tugas Resmi

Soal Resmi

1. Buat koneksi database Oracle dengan Java/Web berdasarkan project masing-masing.
2. Buatlah CRUD pada salah satu tabel (boleh menggunakan GUI, Console atau Web).
3. Terapkan salah satu konsep JOIN dan tampilkan pada project Anda.
4. Terapkan salah satu konsep VIEW dan tampilkan pada project Anda.

Langkah Pertama

Buat file untuk konfigurasi koneksi dari database. Lalu kita import beberapa package dari java.sql. Untuk konfigurasinya seperti dibawah ini

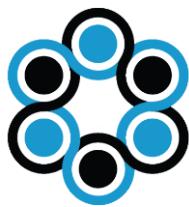
Query

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class Koneksi {
    private Connection connect;
    private Statement db;
    private String database = "oraclejava";

    public Koneksi() {
        try {

Class.forName("oracle.jdbc.driver.OracleDriver");
            System.out.println("Class Driver Ditemukan");
            try {
                connect =
DriverManager.getConnection("jdbc:oracle:thin
:@localhost:1521:xe","khoirul_huda_06990","kh
oirulhuda");
                System.out.println("Koneksi DataBase Ditemukan");
            }
        }
    }
}
```

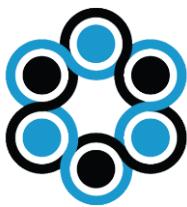


Tugas Resmi

```
 }catch (SQLException se) {
        System.out.println("Koneksi
Database Gagal : " + se);
    }
} catch (ClassNotFoundException err)
{
    System.out.println("Class Driver
Tidak Ditemukan, Terjadi Kesalahan Pada : " +
err);
}

public ResultSet GetData(String sql){
    try {
        db =
connect.createStatement(ResultSet.TYPE_SCROLL
_INSENSITIVE, ResultSet.CONCUR_UPDATABLE);
        return db.executeQuery(sql);
    } catch (SQLException e) {
        return null;
    }
}

public int ManipulasiData(String sql) {
    try {
        db = connect.createStatement();
        return db.executeUpdate(sql);
    } catch(SQLException e) {
        return 0;
    }
}
```



Tugas Resmi

Screenshot

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class Koneksi {
    private Connection connect;
    private Statement db;
    private String database = "oraclejava";

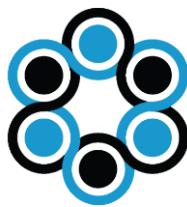
    public Koneksi() {
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            System.out.println("Class Driver Ditemukan");
            try {
                connect = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","Khoirul_huda_06990","Khoirulhuda");
                System.out.println("Koneksi DataBase Ditemukan");
            }catch (SQLException se) {
                System.out.println("Koneksi DataBase Gagal : " + se);
            }
        } catch (ClassNotFoundException err) {
            System.out.println("Class Driver Tidak Ditemukan, Terjadi Kesalahan Pada : " + err);
        }
    }

    public ResultSet GetData(String sql){
        try {
            db = connect.createStatement	ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_UPDATABLE);
            return db.executeQuery(sql);
        } catch (SQLException e) {
            return null;
        }
    }

    public int ManipulasiData(String sql) {
        try {
            db = connect.createStatement();
            return db.executeUpdate(sql);
        } catch(SQLException e) {
            return 0;
        }
    }
}
```

Analisa

Tahap pertama kita akan mengimport beberapa package yang ada dalam `java.sql`. Lalu kita akan mengkonfigurasikan sebuah DB dengan salah satu package. Untuk `resultset` adalah sebuah method yang beguna untuk mengatur menampilkan sebuah data. `ManipulasiData` adalah sebuah methoda yang mengatur sebuah peritah query.



Tugas Resmi

Langkah Kedua

Disini kita akan membuat sebuah method yang akan mengatur sebuah query
Create, Read, Update, Delete

Query

```
public ArrayList<Marketing> getMarketing()
() throws SQLException {
    this.marketing.clear();
    ResultSet rs =
this.koneksi.GetData("SELECT * FROM
MARKETING");
    while (rs.next()) {
        Marketing m = new Marketing();

        m.setId(rs.getInt("ID_MARKETING"));

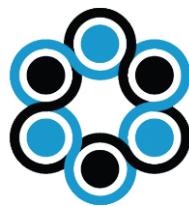
        m.setNama(rs.getString("NAMA_MARKETING"));
        m.setTelp(rs.getInt("NO_TELP"));
        this.marketing.add(m);
    }
    return this.marketing;
}
public void InsertAMP(String table,String
nama_sequence, String nama, String WT){
    String sql = "INSERT INTO "+table+
VALUES('"+nama_sequence+"','"+nama+"','"+WT+
's')";
    this.koneksi.ManipulasiData(sql);
}
public void UpdateAMP(String table,int
id, String update){
    String sql = null;
    if(null != table)switch (table) {
        case "arsitek":{
            sql = "UPDATE "+table+" set
telp_kantor = "+update+" where id_arsitek =
"+id;
            break;
        }
        case "marketing":{
            sql = "UPDATE "+table+" set
no_telp = "+update+" where id_marketing =
"+id;
            break;
        }
    }
}
```



Tugas Resmi

```
        break;
    }
    case "profil_aluminium":{
        sql = "UPDATE "+table+" set
warna = "+update+" where id_profil = "+id;
        break;
    }
    default:
        break;
}
this.koneksi.ManipulasiData(sql);
}

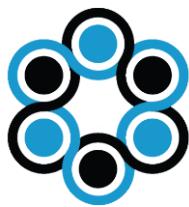
public void DeleteAMP(String table, int
id){
    String sql = null;
    switch (table) {
        case "arsitek":
            sql = "DELETE FROM "+table+
WHERE id_arsitek = "+id;
            break;
        case "marketing":
            sql = "DELETE FROM "+table+
WHERE id_marketing = "+id;
            break;
        case "profil_aluminium":
            sql = "DELETE FROM "+table+
WHERE id_profil = "+id;
            break;
        default:
            break;
    }
    this.koneksi.ManipulasiData(sql);
}
```



Tugas Resmi

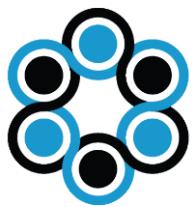
Screenshot

```
===== Selamat Datang =====
===== Menu =====
1 : Create
2 : Delete
3 : Update
4 : Select
5 : Select Join
6 : Select View
Masukan menu yang anda inginkan : 4
===== Select =====
ID      Nama    NO. TELP
=====
1       rahmad  123456789
2       artani   234567891
3       hendro   345678912
4       france   456789123
5       yudha   567891234
===== Ulang =====
Anda mau mengulangi (y/n)y
===== Selamat Datang =====
===== Menu =====
1 : Create
2 : Delete
3 : Update
4 : Select
5 : Select Join
6 : Select View
Masukan menu yang anda inginkan : 1
===== Create =====
Masukan nama : budi
Masukan NO. Telp : 987654321
===== Ulang =====
Anda mau mengulangi (y/n)y
===== Menu =====
1 : Create
2 : Delete
3 : Update
4 : Select
5 : Select Join
6 : Select View
Masukan menu yang anda inginkan : 4
===== Select =====
ID      Nama    NO. TELP
=====
1       rahmad  123456789
2       artani   234567891
3       hendro   345678912
4       france   456789123
5       yudha   567891234
21      budi     987654321
===== Ulang =====
Anda mau mengulangi (y/n)y
```



Tugas Resmi

```
===== Menu =====
1 : Create
2 : Delete
3 : Update
4 : Select
5 : Select Join
6 : Select View
Masukan menu yang anda inginkan : 3
===== Update =====
Masukan ID Marketing : 21
Masukan No. Telp baru : 999999999
===== Ulang =====
Anda mau mengulangi (y/n)y
===== Menu =====
1 : Create
2 : Delete
3 : Update
4 : Select
5 : Select Join
6 : Select View
Masukan menu yang anda inginkan : 4
===== Select =====
ID      Nama    NO. TELP
=====
1      rahmad  123456789
2      artani   234567891
3      hendro   345678912
4      france   456789123
5      yudha    567891234
21     budi     999999999
===== Ulang =====
Anda mau mengulangi (y/n)|
===== Menu =====
1 : Create
2 : Delete
3 : Update
4 : Select
5 : Select Join
6 : Select View
Masukan menu yang anda inginkan : 2
===== Delete =====
Masukan ID Marketing : 21
===== Ulang =====
Anda mau mengulangi (y/n)y
===== Menu =====
1 : Create
2 : Delete
3 : Update
4 : Select
5 : Select Join
6 : Select View
Masukan menu yang anda inginkan : 4
===== Select =====
ID      Nama    NO. TELP
=====
1      rahmad  123456789
2      artani   234567891
3      hendro   345678912
4      france   456789123
5      yudha    567891234
===== Ulang =====
Anda mau mengulangi (y/n)|
```



Tugas Resmi

Analisa :

Disini kita sudah membuat method yang akan dipergunakan create, read, update, dan delete. Dengan bantuan resultset yang ada pada java.



Tugas Resmi

Langkah Ketiga

Kita akan membuat sebuah query join pada program kita

Query

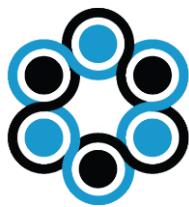
```
public ArrayList<Pengguna_Profil> getPp()
throws SQLException {
    this.pp.clear();
    ResultSet rs =
this.koneksi.GetData("SELECT
marketing.nama_marketing,
profil_aluminium.nama_profil,count(marketing.
nama_marketing) AS JUMLAH from proyek right
join profil_aluminium on
profil_aluminium.id_profil = proyek.id_profil
right join marketing on
marketing.id_marketing = proyek.id_marketing
where profil_aluminium.nama_profil =
'standard' and profil_aluminium.warna = 'iron
grey' group by marketing.nama_marketing,
profil_aluminium.nama_profil");
    while(rs.next()){
        Pengguna_Profil pProfil = new
Pengguna_Profil();

pProfil.setMarketing(rs.getString("NAMA_MARKE
TING"));

pProfil.setProfil(rs.getString("NAMA_PROFIL"))
;

pProfil.setJumlah(rs.getInt("JUMLAH"));

        this.pp.add(pProfil);
    }
    return this.pp;
}
```



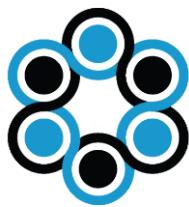
Tugas Resmi

Screenshot

```
===== Menu =====
1 : Create
2 : Delete
3 : Update
4 : Select
5 : Select Join
6 : Select View
Masukan menu yang anda inginkan : 5
===== Join =====
Marketing      Profil      Jumlah
=====
rahmad        standart      2
artani        standart      1
```

Analisa :

Disini kita membuat join dari beberapa table dan kita langsung menampilkan isinya



Tugas Resmi

Langkah Keempat

Pada langkah keempat ini kita sudah pernah membuat sebuah view pada database saat modul 4. Disini kita akan memanggil view tersebut

Query

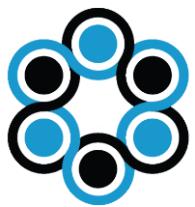
```
public ArrayList<Kepemilikan_Proyek>
getkp() throws SQLException {
    this.kp.clear();
    ResultSet rs =
this.koneksi.GetData("SELECT * FROM
KEPEMILIKAN_PROYEK");
    while (rs.next()) {
        Kepemilikan_Proyek kProyek = new
Kepemilikan_Proyek();

kProyek.setMarketing(rs.getString("NAMA_MARKE
TING"));

kProyek.setOwner(rs.getString("NAMA_OWNER"));
        this.kp.add(kProyek);
    }
    return this.kp;
}
```

Screenshot

```
===== Menu =====
1 : Create
2 : Delete
3 : Update
4 : Select
5 : Select Join
6 : Select View
Masukan menu yang anda inginkan : 6
===== View =====
Marketing          Owner
=====
artani            kardono
rahmad           sudarmono
rahmad           yunus
artani            alex
artani            yos tanoyo
```



Tugas Resmi

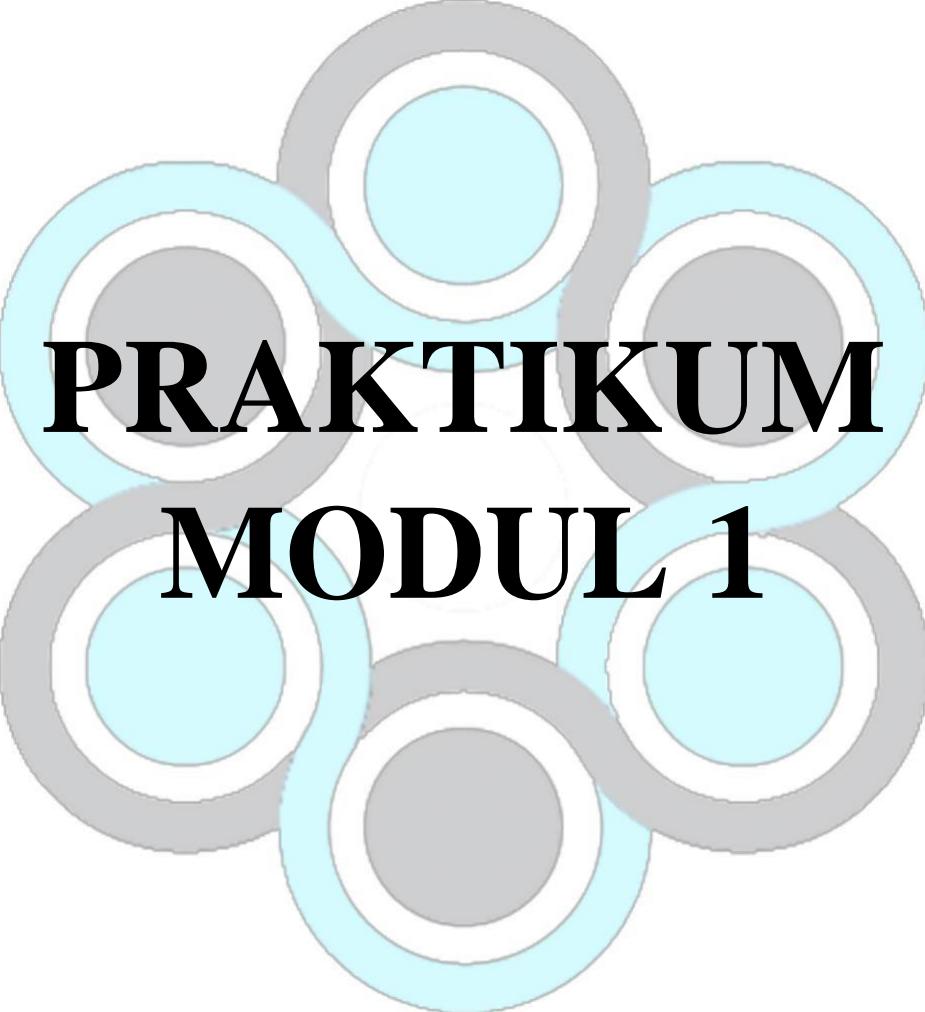
Analisa :

Disini kita hanya menampilkan bagian dari view yang sudah kita buat pada modul 4



LAPORAN PRAKTIKUM

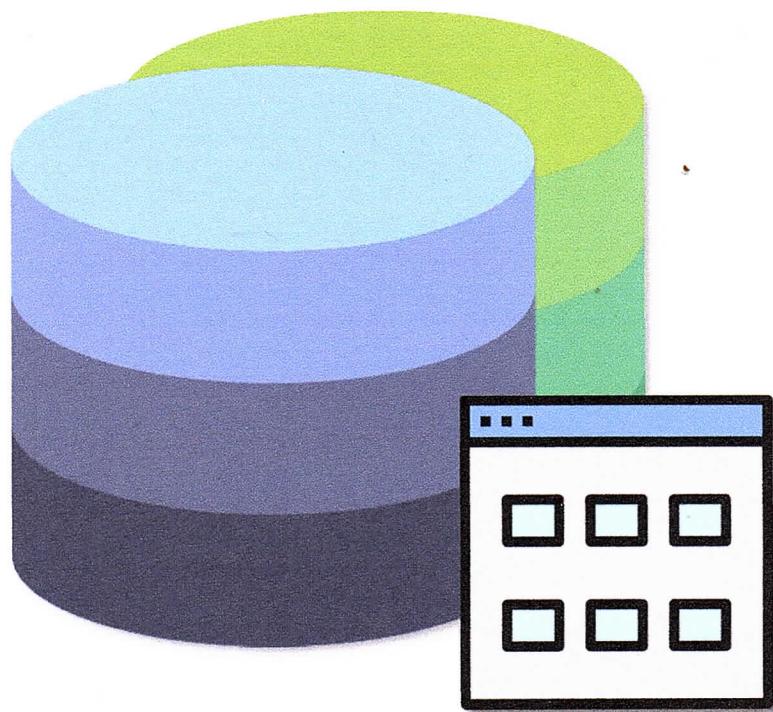
FAKULTAS TEKNIK ELEKTRO DAN
TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI ADHI TAMA
SURABAYA
2020



PRAKTIKUM MODUL 1

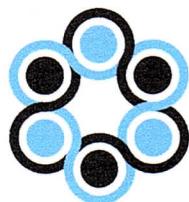
FAKULTAS TEKNIK ELEKTRO DAN
TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI ADHI TAMA
SURABAYA
2020

LAPORAN PENDAHULUAN



PRAKTIKUM BASIS DATA PERIODE V - Tahun 2019/2020

NAMA : Mochamad Khairul Huda
NPM : 06.2010.1.06990
MODUL : I



Tugas Pertanyaan

1. Buatlah Sistem Informasi Penjualan SeJelerha dimana terdapat 8 tabel yang harus saling berelasi. Yaitu tabel Pegawai, Pelanggan, kuit, Pemasok, kategori, Produk, Pemesanan dan Detail Pemesanan
Langkah Pertama:
 - Buatlah table space Seperti contoh dibawah :

```
CREATE TABLESPACE <nama_tablespace> DATAFILE  
'nama_tablespace.dbf' SIZE "Ukuran_table  
space";
```

SQL > CREATE TABLESPACE SistemJual
 - 2. datafile 'D:\ITATS\MATKUL\Semester 4\Basis Data
\Praktikum\System-jual.dbf'
 - 3. Size 30M;

Table space created

 - Buat User dengan ketentuan NAMA-NPM

```
CREATE USER <nama-user> IDENTIFIED BY <password->  
DEFAULT TABLESPACE <nama_tablespace>  
QUOTA <batas_ukuran_tablespace> ON <nama_table  
space>;
```

SQL > CREATE USER jennie - 012345

 2. IDENTIFIED BY jennie
 - 3 DEFAULT TABLESPACE SistemJual
 4. QUOTA 30M ON Sistem Jual;

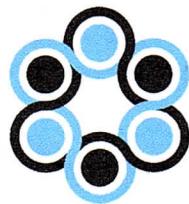
User created

 - Beri hak akses "All Privileges" untuk user

```
GRANT ALL PRIVILEGES TO <nama-user>;
```

SQL > GRANT ALL PRIVILEGES TO jennie - 012345;

Grant succeeded



Tugas Pertanyaan

- Logon ke user yang telah dibuat

CONN < nama-user > / < Password - users >

SQL > Conn jennie_012345 / jennie

Connected

Langkah kedua

Buat tabel Pegawai dengan nama < Pegawai - NPM > dan beberapa kolom id-Pegawai, nama-depan, nama-belakang, tanggal-lahir, alamat, kode-pos dan no-telp. Dimana id-Pegawai sebagai Primary key

SQL > Create table Pegawai - 012345

2. (

3. Id - Pegawai INTEGER not null,

4. Nama - depan VARCHAR (15),

5. Nama - belakang VARCHAR (15),

6. Tanggal - lahir DATE,

7. Alamat VARCHAR2 (100),

8. Kode - Pos NUMBER (7),

9. No - telp NUMBER (12),

10. Constraint PK - Pegawai Primary Key (Id - Pegawai)

11);

langkah ketiga

Buat tabel Pelanggan dengan nama < Pelanggan - NPM > dan beberapa kolom id-Pelanggan, nama-depan, nama-belakang, tanggal-lahir, alamat, kode-pos dan no-tlp.

Dimana id-Pelanggan, sebagai Primary key

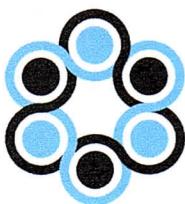
SQL > Create table Pelanggan - 012345

2 (

3. Id - Pelanggan INTEGER not null,

4. Nama - depan VARCHAR2 (15),

5. Nama - belakang VARCHAR2 (15),



Tugas Pertanyaan

6. Tanggal - lahir Date,
7. Alamat VARCHAR2(100),
8. Kode - Pos NUMBER (7),
9. NO - telp NUMBER (12),
10. Constraint PK - Pelanggan Primary key
(Id - Pelanggan)
11.);

(langkah keempat

Buat tabel kurir dengan nama < kurir - NPM > dan beberapa kolom Id - kurir , nama - perusahaan dan no - tlp .
Dimana Id - kurir , sebagai Primary key

SQL > Create table kurir_012345

2 (

3 Id - kurir INTEGER not null,
4 Nama - Perusahaan VARCHAR2(16),
5 No - tlp NUMBER (12),
6. Constraint PK - kurir Primary key (Id - kurir)

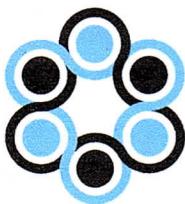
7.);

(langkah kelima

Buat tabel Pemasok dengan nama < Pemasok - NPM > dan beberapa kolom Id - kurir , nama - perusahaan , alamat ,
kode - pos dan no - tlp . Dimana Id - Pemasok sebagai Primary key .

SQL > Create table Pemasok_0egg0 (

2. Id - Pemasok INTEGER not null ,
3. Nama - Perusahaan VARCHAR2(15) ,
4. Alamat VARCHAR2 (100) ,
5. Kode - pos NUMBER (7) ,
6. NO - tlp NUMBER (12) ,
7. Constraint PK - Pemasok Primary key (Id - Pemasok)
8.);



Tugas Pertanyaan

Langkah keenam

Buat tabel kategori dengan nama `kategori_Npm` dan kolom `Id_kategori`, `Nama_kategori` dan `deskripsi`. Dimana `Id_kategori` sebagai Primary key.

SQL > Create table `kategori_06990`

```
2. (
3   Id_kategori      INTEGER          not null,
4   Nama_kategori    VARCHAR(16),
5   Deskripsi        VARCHAR(20),
6.  constraint PK_kategori primary key (Id_kategori)
7. );
```

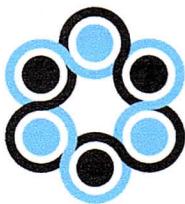
Table created

Langkah ke tujuh

Buat tabel Produk dengan nama `produk_Npm` dan beberapa kolom `Id_Produk`, `Id_Pemasok`, `Id_kategori`, `Nama_Produk`, `harga_Satuan` dan `Stock_Produk`. Dimana `Id_Produk` sebagai Primary key , `Id_Pemasok` dan `Id_kategori` sebagai Foreign key .

SQL > Create tabel `produk_06990`

```
2. (
3   Id_Produk        INTEGER          not null,
4   Id_Pemasok       INTEGER,
5   Id_kategori     INTEGER,
6.  Nama_Produk     VARCHAR(50),
7.  Harga_Satuan    NUMBER(9),
8.  Stock_Produk    NUMBER(3),
9.  constraint PK_Produk primary key (Id_Produk)
10. constraint FK_Pemasok FOREIGN key (Id_Pemasok)
11. REFERENCES Pemasok_06990 (Id_Pemasok),
12. constraint FK_kategori FOREIGN key (Id_kategori)
13. REFERENCES kategori_06990 (Id_kategori)
```



Tugas Pertanyaan

Table created.

Langkah kedelapan

Buat tabel Pemesanan dengan nama <Pemesanan - NPM> dan beberapa kolom Id_Pemesanan, Id_Pelanggan, Id_Pegawai, Id_kurir, tanggal_Pemesanan, tanggal_Pengiriman, alamat_Pengiriman dan harga_total. Dimana Id_Pemesanan sebagai Primary key.

SQL > Create table Pemesanan_Ogggo

2. C

3. Id_Pemesanan INTEGER not null

4. Id_Pelanggan INTEGER,

5. Id_Pegawai INTEGER,

6. Id_kurir INTEGER,

7. tanggal_Pemesanan DATE,

8. Tanggal_Pengiriman DATE,

9. Alamat_Pengiriman VARCHAR2(100),

10. Harga_Total NUMBER(9),

11. constraint PK_Pemesanan Primary key (Id_Pemesanan)

12. ;

Table Created.

Langkah kesembilan

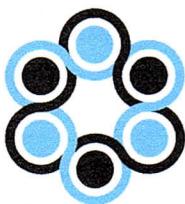
tambahkan foreign key ke tabel Pemesanan pada kolom Id_Pelanggan, Id_Pegawai, Id_kurir dengan menggunakan Alter Statement

SQL > alter table Pemesanan_Ogggo

2. add constraint fk_Id_Pelanggan FOREIGN key (Id_Pelanggan)

3. References Pelanggan_Ogggo (Id_Pelanggan)

4. add constraint fk_Id_Pegawai FOREIGN key (Id_Pegawai)



Tugas Pertanyaan

5. References Pegawai_Ogggo (Id_Pegawai)

6. add Constraint Pk_Id_kurir FOREIGN key (Id_kurir)

7. References kurir_Ogggo (Id_kurir);

Table altered

Langkah kesepuluh

Buat tabel Detail_Pemesanan dengan beberapa kolom Id_Produk, Id_Pemesanan, jumlah dan diskon.

SQL > Create table Detail_Pemesanan_Ogggo

2 (

3 Id_Produk INTEGER not null,

4 Id_Pemesanan INTEGER,

5 Jumlah NUMBER(3),

6 Diskon FLOAT(4)

7);

table created

Langkah kesembelas

Tambahkan foreign key ketable Detail_Pemesanan pada kolom Id_Produk dan Id_Pemesanan dengan menggunakan Alter Statement

SQL > alter table Detail_Pemesanan_Ogggo

2 add Constraint Pk_Id_Produk FOREIGN key (Id_Produk)

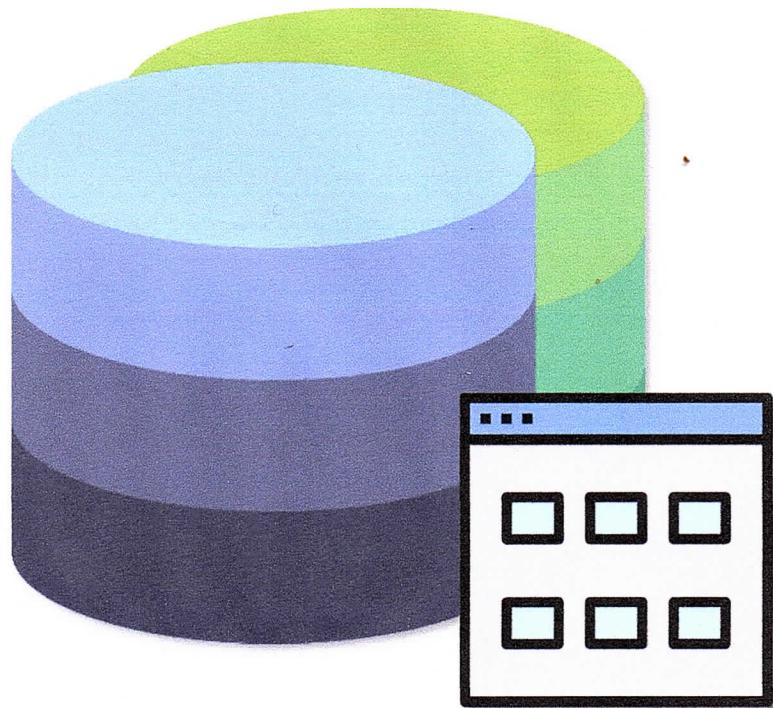
3 References Produk_Ogggo (Id_Produk)

4. add constraint Pk_Id_Pemesanan FOREIGN key
(Id_Pemesanan)

5. References Pemesanan_Ogggo (Id_Pemesanan);

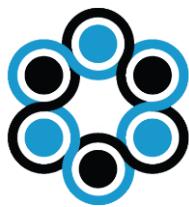
table altered

LAPORAN PRAKTIKUM



**PRAKTIKUM BASIS DATA
PERIODE V - Tahun 2019/2020**

NAMA : Mochamad Khairul Huda
NPM : 06.2018.1.06990
MODUL : I



Tugas Praktikum

Soal Praktikum

1. Buatlah user dan tablespace sesuai Nama dan NPM masing – masing praktikan.
2. Buatlah tabel sesuai Desain Database Proyek Kelas yang sudah di ACC oleh dosen.
3. Buatlah relasi antar tabel sesuai Desain Database dengan perintah ALTER.
4. Menerapkan Sequence pada salah satu tabel.
5. Ubah nama salah satu field pada salah satu tabel dengan menambah NPM didepan nama field.

Contoh : 06852_NamaProduk

6. Ubah salah satu field pada salah satu tabel dengan Attribute Constraint UNIQUE.
7. Ubah salah satu field pada salah satu tabel dengan Tipe data yang berbeda.

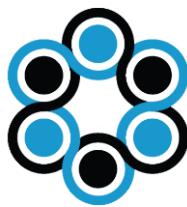
Langkah Pertama

- **Buat tablespace**

```
create tablespace <nama_table>
  datafile '<Penempatan_file>\<Nama_Tablespace.dbf>'
  size <ukuran_tablespace>;
```

- **Buat user**

```
create user <nama_user_NPM>
  identified by <password>
  default tablespace <nama_tablespace>
  quota <banyaknya penyimpanan> on <nama_tablespace>;
```



Tugas Praktikum

Query

```
create tablespace il_sogno
datafile
'D:\irul\BERKAS\kuliah\praktikum\basis
data\modul 1\IL_SOOGNO.dbf'
size 30M;

create user khoirul_huda_06990
identified by khoirulhuda
default tablespace IL_SOOGNO
quota 30M on IL_SOOGNO;
```

Screenshot

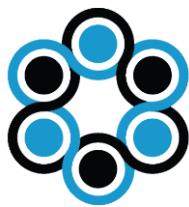
```
SQL> create tablespace il_sogno
2 datafile 'D:\irul\BERKAS\kuliah\praktikum\basis data\modul 1\IL_SOOGNO.dbf'
3 size 30M;

SQL> create user khoirul_huda_06990
2 identified by khoirulhuda
3 default tablespace IL_SOOGNO
4 quota 30M on IL_SOOGNO;

User created.
```

Analisa

Sebelum membuat user kita diharuskan membuat tablespace terlebih dahulu supaya saat kita membuat user, kita bisa memasukan langsung tablespacenya. Pembuatan tablespace kita bisa meletakan data file sesuai dengan keinginan kita dan jangan lupa untuk memberi ukuran data table space tersebut. Pembuatan user harus diawali dengan character, kita tidak bisa membuat user dengan angka atau simbol terlebih dahulu. Kata identified adalah kita memasukan password dari user yang sudah kita buat.



Tugas Praktikum

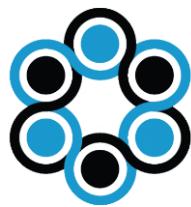
Langkah Kedua

- Buat table dengan nama Marketing, didalam table Marketing terdapat atribut Id_Marketing, Nama_Marketing, No_Telp.
- Buat table Profil_Aluminium yang didalam table Profil_Aluminium terdapat atribut Id_Marketing, Nama_Marketing, No_Telp.
- Buat table Arsitek yang didalam table Arsitek terdapat atribut Id_Arsitek, Nama_Arsitek, Telp_Kantor dengan Id_Arsitek sebagai primary key nya
- Buat table proyek yang didalam table Proyek

Query

```
create table Marketing
(
Id_Marketing int,
Nama_Marketing varchar(30),
No_Telp number(9),
constraint pk_marketing primary key
(Id_Marketing)
);
```

```
create table Profil_Aluminium
(
Id_Profil int,
Nama_Profil varchar(30),
Warna varchar (50),
constraint pk_profil primary key (Id_Profil)
);
```



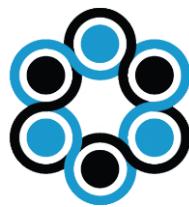
Tugas Praktikum

```
create table Profil_Aluminium
(
Id_Profil int,
Nama_Profil varchar(30),
Warna varchar (50),
constraint pk_profil primary key (Id_Profil)
);
```

Screenshot

```
connected.
SQL> create table Marketing
  2  (
  3   Id_Marketing int,
  4   Nama_Marketing varchar(30),
  5   No_Telp number(9)
  6  );
Table created.
```

```
SQL> create table Profil_Aluminium
  2  (
  3   Id_Profil int,
  4   Nama_Profil varchar(30),
  5   Warna varchar (50),
  6   constraint pk_profil primary key (Id_Profil)
  7  );
Table created.
```



Tugas Praktikum

```
SQL> create table Arsitek
  2  (
  3  Id_Arsitek integer not null,
  4  Nama_Arsitek varchar(30),
  5  Telp_Kantor number(9),
  6  constraint pk_Arsitek primary key(Id_Arsitek)
  7  );
Table created.
```

```
SQL> create table Proyek
  2  (
  3  Nama_Owner varchar(30),
  4  Alamat_Proyek Varchar(100),
  5  Alamat_Penyuratan varchar(100),
  6  No_Telp number(9),
  7  Id_Arsitek integer,
  8  Id_Marketing integer,
  9  Id_Profil integer,
 10  constraint pk_Proyek primary key(Alamat_Proyek)
 11  );
Table created.
```

Analisa :

Saat membuat table awal dari nama table diharuskan menggunakan character, jika kita menggunakan simbol atau angka terlebih dahulu maka akan terjadi error saat kita membuat table tersebut. Ada beberapa tipe data dalam database, sebagai contoh:

Varchar adalah sebuah tipe data character

Number adalah sebuah tipe data angka real

Integer adalah sebuah tipe data angka

Saat membuat primary key kita menggunakan syntax constraint



Tugas Praktikum

Langkah Ketiga

Langkah selanjutnya adalah kita membuat relasi antar table dengan perintah alter

Sql > alter table <nama_tabel>

```
add constraint <nama_foreign_key> foreign key (<atribute_yang_dituju>)
references <nama_tabel_yang_atribute_dijadikan_refrensi>
(<nama_atribute_dari_table_yang_direfrensikan>)
;
```

Query

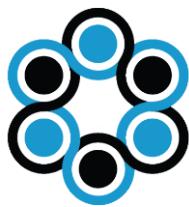
```
alter table Proyek
add constraint fk_Marketing foreign key
(Id_Marketing) references Marketing
(Id_Marketing)
add constraint fk_Arsitek foreign key
(Id_Arsitek) references Arsitek (Id_Arsitek)
add constraint fk_Profil foreign key
(Id_Profil) references Profil_Aluminium
(Id_Profil)
;
```

Screenshot

```
SQL> alter table Proyek
  2  add constraint fk_Marketing foreign key (Id_Marketing) references Marketing (Id_Marketing)
  3  add constraint fk_Arsitek foreign key (Id_Arsitek) references Arsitek (Id_Arsitek)
  4  add constraint fk_Profil foreign key (Id_Profil) references Profil_Aluminium (Id_Profil)
  5 ;
Table altered.
```

Analisa :

Saat kita akan merubah atau menambahkan sesuatu didalam tabel kita akan menggunakan perintah ALTER. Jika kita akan menambahkan sesuatu dalam table kita juga akan menggunakan perintah add. Pembuatan foreign key sama halnya dengan pembuatan primary key, akan tetapi perbedaan pembuatan foreign key terdapat dalam perintah REFERENCES yang artinya di refrensikan dari table yang mana dan didalam kurung itu adalah nama dari field yang direfrensikan.



Tugas Praktikum

Langkah Keempat

Langkah ke-4 ini kita akan menggunakan sequence.

Sql > create sequence <nama_field_dalam_table>

```
minvalue <nilai_minimal>
maxvalue <nilai_maksimal>
start with <dimulai_angka>
increment by <dijumlahkan>
cache <kapasitas>;
```

Query

```
create sequence Id_Marketing
minvalue 1
maxvalue 9999
start with 1
increment by 1
cache 20;
```

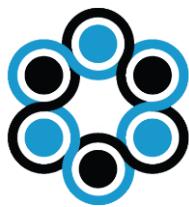
Screenshot

```
SQL> create sequence Id_Marketing
  2  minvalue 1
  3  maxvalue 9999
  4  start with 1
  5  increment by 1
  6  cache 20;

Sequence created.
```

Analisa :

Perintah minvalue ada perintah dari minimal value yang akan diberikan untuk max value adalah perintah pembatasan dari maksimalnya isi field. Start with adalah value yang dimulai dari 1 dikarenakan Id_Marketing adalah field dengan tipe data integer. Increment adalah penjumlah dari isi field yang sebelumnya.



Tugas Praktikum

Langkah Kelima

merubah nama field di salah satu table

```
alter table <nama_tabel>
```

```
rename column <nama_field_lama> to <nama_field_baru>;
```

Query

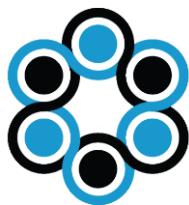
```
alter table Marketing
rename column Nama_Marketing to
06990_Nama_Marketing;
```

Screenshot

```
SQL> alter table Marketing
  2  rename column Nama_Marketing to 06990_Nama_Marketing;
Table altered.
```

Analisa :

Saat kita mengubah sebuah nama table maka kita akan menggunakan perintah ALTER dan di ikuti dengan nama dari sebuah table yang nama fieldnya akan kita rubah. Merubah sebuah ama dari field maka kita juga menggunakan perintah rename dan kata column adalah letak dari field lalu di ikuti nama awal dari sebuah field. Kata TO merujuk ke kata berganti lalu di ikuti dengan nama field yang baru. Akan tetapi seperti yang saya jelaskan sebelumnya jika penamaan sebuah table tidak bisa menggunakan angka terlebih dahulu. Maka dari itu angka 0 diawal nama field yang baru saya ganti dengan menggunakan huruf O.



Tugas Praktikum

Langkah Keenam

Kita akan mengubah salah satu field di salah satu tablr dengan attribute Constraint UNIQUE.

```
SQL> ALTER TABLE <nama_tabel>
      add UNIQUE (<nama_field>);
```

Query

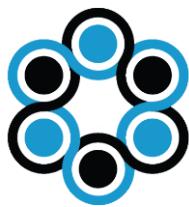
```
ALTER TABLE Proyek
add UNIQUE (No_Telp);
```

Screenshot

```
SQL> ALTER TABLE Proyek
      2 add UNIQUE (No_Telp);
      |
Table altered.
```

Analisa :

ALTER adalah syntax untuk merubah dan menambahkan sesatu. Jadi saat menambahakan UNIQUE kita juga menggunakan syntax ALTER dan diikuti dengan nama table yang dituju. Lalu kita menggunakan kata add diikuti UNIQUE lalu nama fieldnya. UNIQUE bisa dipakai seperti contoh nomer telephone dan email karena di dalam keduanya terdapat sebuah simbol.



Tugas Praktikum

Langkah Ketujuh

kita akan mengubah salah satu field dengan tipe data yang berbeda dengan menggunakan perintah di bawah ini

```
alter table <nama_table> modify (<nama_field> <tipe_data>);
```

Query

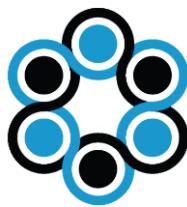
```
alter table Proyek modify (Id_Profil  
number(12));
```

Screenshot

```
SQL> alter table Proyek modify (Id_Profil number(12));  
Table altered.
```

Analisa :

Perintah **MODIFY** ini adalah perintah untuk mengubah tipe data dari sebuah field. Saya mengubah tipe data dari field **Id_Profil** yang sebelumnya integer menjadi number dikarenakan tipe data ini hampir sama.



Laporan Sementara

Soal Praktikum

1. Buatlah user dan tablespace sesuai Nama dan NPM masing – masing praktikan.
2. Buatlah tabel sesuai Desain Database Proyek Kelas yang sudah di ACC oleh dosen.
3. Buatlah relasi antar tabel sesuai Desain Database dengan perintah ALTER.
4. Menerapkan Sequence pada salah satu tabel.
5. Ubah nama salah satu field pada salah satu tabel dengan menambah NPM didepan nama field.

Contoh : 06852_NamaProduk

6. Ubah salah satu field pada salah satu tabel dengan Attribute Constraint UNIQUE.
7. Ubah salah satu field pada salah satu tabel dengan Tipe data yang berbeda.

Langkah Pertama

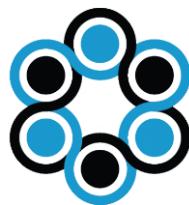
- Buat tablespace

```
create tablespace <nama_tablespace>
2 datafile <'penyimpanan.dbf'>
3 size 30M;
```

- Buat user

```
create user <nama_users_NPM>
2 identified by <password>
3 default tablespace <nama_tablespace>
4 quota 30M on <nama_tablespace>;
```

User created.



Laporan Sementara

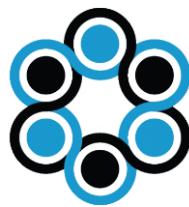
Query

```
create tablespace il_sogno
2 datafile
'D:\irul\BERKAS\kuliah\praktikum\basis
data\modul 1\IL_SOGLNO.dbf'
3 size 30M;

create user khoirul_huda_06990
2 identified by khoirulhuda
3 default tablespace IL_SOGLNO
4 quota 30M on IL_SOGLNO;
User created.
```

Screenshot

```
SQL> create tablespace il_sogno
2 datafile 'D:\irul\BERKAS\kuliah\praktikum\basis data\modul 1\IL_SOGLNO.dbf'
3 size 30M;
```



Laporan Sementara

Screenshot

```
SQL> create user khoirul_huda_06990
  2  identified by khoirulhuda
  3  default tablespace IL_SOOGNO
  4  quota 30M on IL_SOOGNO;

User created.
```

Analisa

(Ketik hasil analisa disini)



Laporan Sementara

Langkah Kedua

membuat table Marketing dengan atribut Id_Marketing, Nama_Marketing, No_Telp

Query

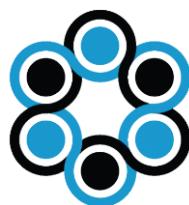
```
SQL> create table Marketing
2  (
3  Id_Marketing int,
4  Nama_Marketing varchar(30),
5  No_Telp number(9)
6  );
```

Screenshot

```
Connected.
SQL> create table Marketing
2  (
3  Id_Marketing int,
4  Nama_Marketing varchar(30),
5  No_Telp number(9)
6  );
Table created.
```

Analisa :

(Ketik hasil analisa disini)



Laporan Sementara

Langkah Ketiga

membuat table Profil_Aluminium dengan atribut Id_Profil, Nama_Profil, Warna

Query

```
SQL> create table Profil_Aluminium
2  (
3  Id_Profil int,
4  Nama_Profil varchar(30),
5  Warna varchar (50)
6  );
```

Screenshot

```
SQL> create table Profil_Aluminium
2  (
3  Id_Profil int,
4  Nama_Profil varchar(30),
5  Warna varchar (50)
6  );
Table created.
```

Analisa :

(Ketik hasil analisa disini)



Laporan Sementara

Langkah Keempat

membuat table Aristek dengan atribut Id_Arsitek, Nama_Arsitek, Telp_Kantor dengan syarat Id_Arsitek menjadi primary key

Query

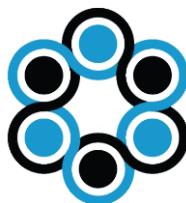
```
SQL> create table Arsitek
  2  (
  3  Id_Arsitek integer not null,
  4  Nama_Arsitek varchar(30),
  5  Telp_Kantor number(9),
  6  constraint pk_Arsitek primary
key(Id_Arsitek)
  7  );
```

Screenshot

```
SQL> create table Arsitek
  2  (
  3  Id_Arsitek integer not null,
  4  Nama_Arsitek varchar(30),
  5  Telp_Kantor number(9),
  6  constraint pk_Arsitek primary key(Id_Arsitek)
  7  );
Table created.
```

Analisa :

(Ketik hasil analisa disini)



Laporan Sementara

Langkah Kedua

membuat table proyek dengan atribut

Query

```
SQL> create table Proyek
(
    Nama_Owner varchar(30),
    Alamat_Proyek Varchar(100),
    Alamat_Penyuratan varchar(100),
    No_Telp number(9),
    Id_Arsitek integer,
    Id_Marketing integer,
    Id_Profil integer,
    constraint pk_Proyek primary
    key(Alamat_Proyek),
    constraint fk_Marketing foreign key
    (Id_Marketing) references Marketing
    (Id_Marketing),
    2 constraint fk_Arsitek foreign key
    (Id_Arsitek) references Arsitek (Id_Arsitek),
    3 constraint fk_Profil foreign key
    (Id_Profil) references Profil_Aluminium
    (Id_Profil)
    4 );
```

Screenshot

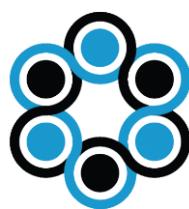
```
SQL> create table Proyek
          (
    Nama_Owner varchar(30),
    Alamat_Proyek Varchar(100),
    Alamat_Penyuratan varchar(100),
    No_Telp number(9),
    Id_Arsitek integer,
    Id_Marketing integer,
    Id_Profil integer,
    constraint pk_Proyek primary key(Alamat_Proyek),
    constraint fk_Marketing foreign key (Id_Marketing) references Marketing (Id_Marketi
ng),
    2 constraint fk_Arsitek foreign key (Id_Arsitek) references Arsitek (I
d_Arsitek),
    3 constraint fk_Profil foreign key (Id_Profil) references Profil_Alumi
nium (Id_Profil)
    4 );
Table created.
```



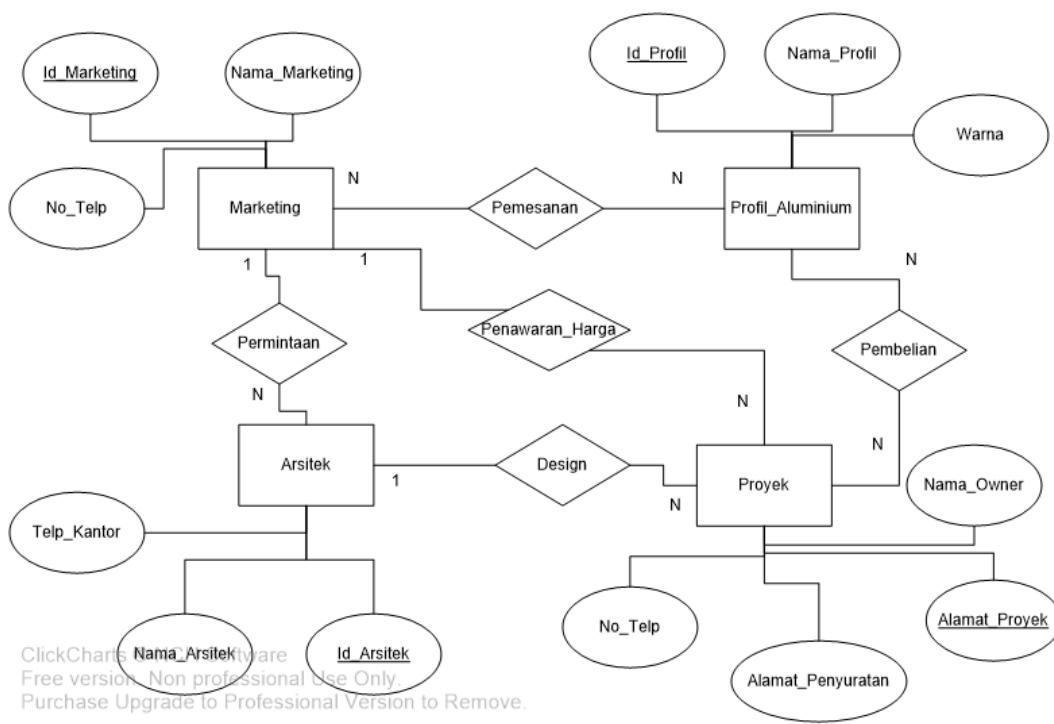
Laporan Sementara

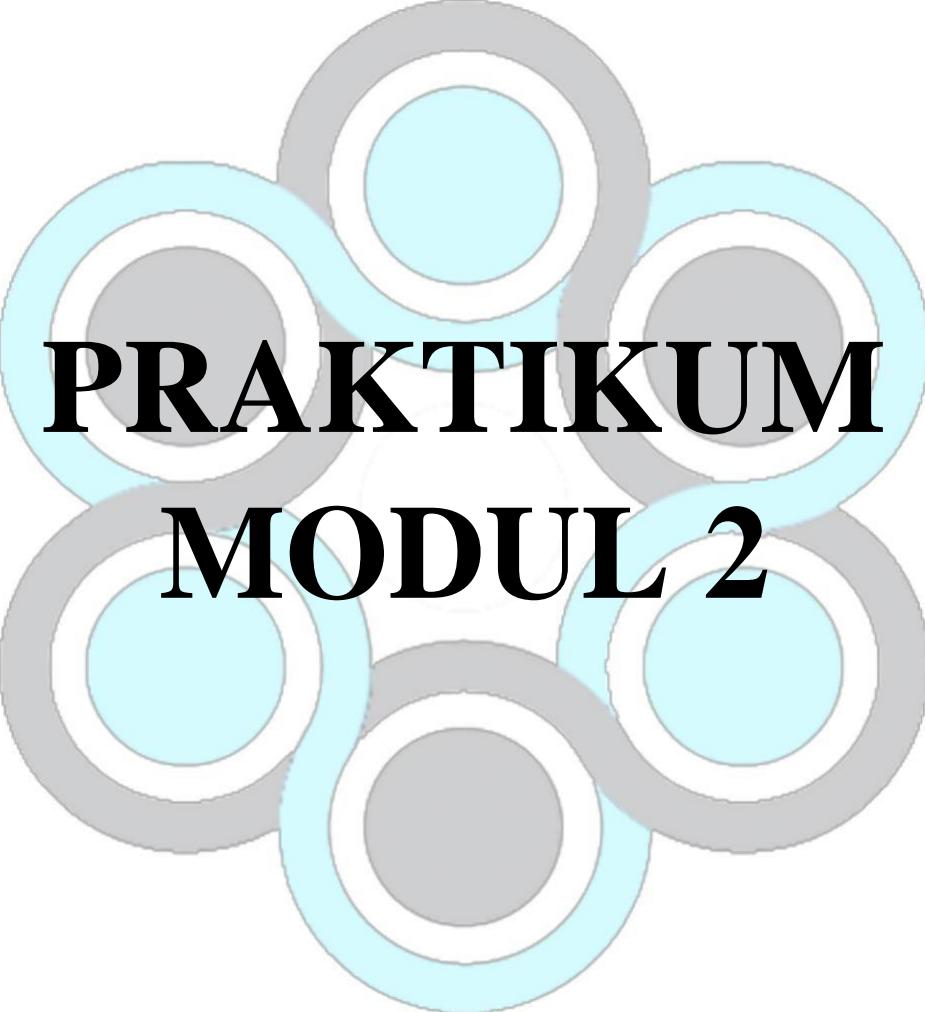
Analisa :

(Ketik hasil analisa disini)



Laporan Sementara

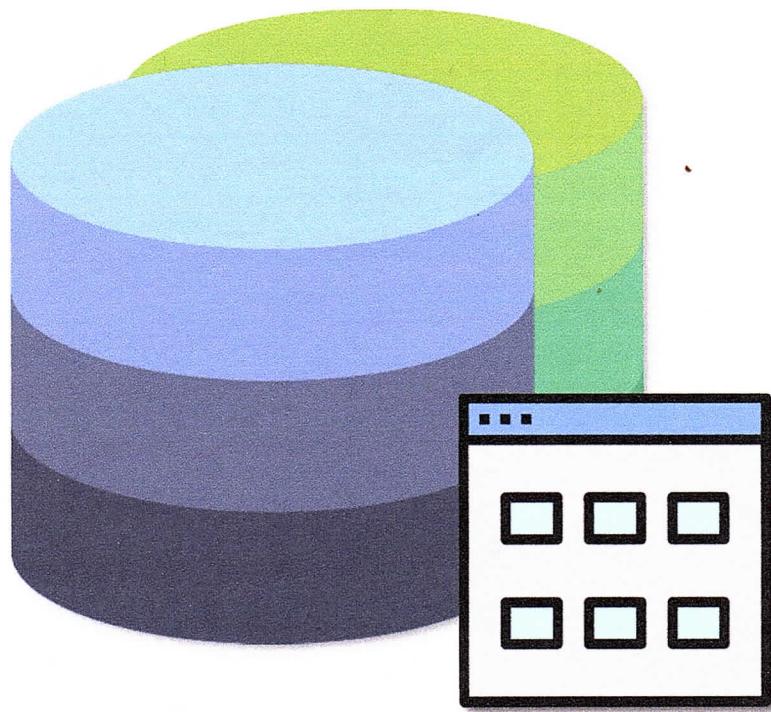




PRAKTIKUM MODUL 2

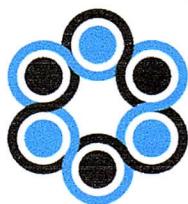
FAKULTAS TEKNIK ELEKTRO DAN
TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI ADHI TAMA
SURABAYA
2020

LAPORAN PENDAHULUAN



PRAKTIKUM BASIS DATA PERIODE V - Tahun 2019/2020

NAMA : M. Khoirul Huda
NPM : 06.2018.1.06990
MODUL : II



Tugas Pertanyaan

Modul 2.

1. Insert semua table berdasarkan PDM pada Modul 1 dengan syarat : 5 bars data pada table Pegawai, Pelanggan, Iuran, Pemasok, Kategori, Produk, Pemesanan dan Detal Pemesanan.

- Table Pegawai

SQL > Insert all

2. Into Pegawai_06ggo (Id-Pegawai, Nama-Depan, Nama-Belakang, Tanggal-lahir, Alamat, kode-Pas, No-telp)

Value (1, 'Bego', 'Subego' to_date ('01/10/1998', 'dd/mm/yyyy'), 'Bandung', 69198, '123456789')

3. Into Pegawai_06ggo (Id-Pegawai, Nama-Depan, Nama-Belakang, Tanggal-lahir, Alamat, kode-Pas, No-telp) Value (2, 'Budi', 'Subakti', to_date ('25/01/1997', 'dd/mm/yyyy'), 'Madiura', 6971, '2345678912')

4. Into Pegawai_06ggo (Id-Pegawai, Nama-Depan, Nama-Belakang, Tanggal-lahir, Alamat, kode-Pas, No-telp) Value (3, 'Bachti', 'Seharti', to_date ('08/01/1995', 'dd/mm/yyyy'), 'Soloargo', 6916, '345678912')

5. Into Pegawai_06ggo (Id-Pegawai, Nama-Depan, Nama-Belakang, Tanggal-lahir, Alamat, kode-Pas, No-Telp) Value (4, 'Rini', 'Aini', to_date ('06/11/1998', 'dd/mm/yyyy'), 'Rungkut', 67911, '456789123')

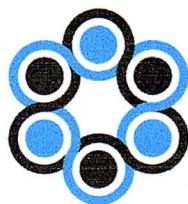
6. Into Pegawai_06ggo (Id-Pegawai, Nama-Depan, Nama-Belakang, Tanggal-lahir, Alamat, kode-Pas, No-Telp) Value (5, 'Beni', 'Pratama', to_date ('31/05/1998', 'dd/mm/yyyy'), 'Sukesirto', 6901, '567891234')

7. Select 1 from dual;

- Table Pelanggan

SQL > Insert all

2. Into Pelanggan_06ggo (Id-Pelanggan, Nama-Depan,



Tugas Pertanyaan

Nama Belakang, tanggal_lahir, Alamat, kode_pos, No_telp)

Values (1, 'Tomy', 'maulana', to_date ('31/10/1998', 'dd/mm/yyyy'), 'lombok', Gg18, '123456789')

3. Insert Pelanggan_06gggo (Id_Pelanggan, Nama_Depan, Nama_Belakang, Tanggal_lahir, Alamat, kode_pos, No_Telp)

Values (2, 'Indah', 'Nuraini', to_date ('25/01/1988', 'dd/mm/yyyy'), 'madura', Gg71, '234567891')

4. Insert Pelanggan_06gggo (Id_Pelanggan, Nama_Depan, Nama_Belakang, tanggal_lahir, Alamat, kode_pos, No_Telp) Values (3,

'Abdul', 'Falah', to_date ('08/01/1994', 'dd/mm/yyyy'), 'Sriwijaya', Gg16, '345678912')

5. Insert Pelanggan_06gggo (Id_Pelanggan, Nama_Depan, Nama_Belakang, tanggal_lahir, Alamat, kode_pos, No_Telp)

Values (4, 'Reno', 'ubaidillah', to_date ('06/11/1998', 'dd/mm/yyyy'), 'Juppner', Gg11, '456789123')

6. Insert Pelanggan_06gggo (Id_Pelanggan, Nama_Depan, Nama_Belakang, tanggal_lahir, Alamat, kode_pos, No_Telp)

Values (5, 'Aritzen', 'Suhartono', to_date ('31/05/1998', 'dd/mm/yyyy'), 'Pluto', Gg01, '567891234')

7. Select * From dual

- Tabel kurir

SQL> Insert all

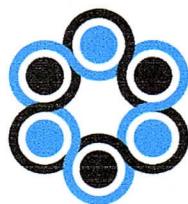
2. Insert kurir_06gggo (Id_kurir, Nama_Perusahaan, No_Telp) Values (1, 'Indomart', '123456789')

3. Insert kurir_06gggo (Id_kurir, Nama_Perusahaan, No_Telp) Values (2, 'Alfamart', '234567891')

4. Insert kurir_06gggo (Id_kurir, Nama_Perusahaan, No_Telp) Values (3, 'Giant', '345678912')

5. Insert kurir_06gggo (Id_kurir, Nama_Perusahaan, No_Telp) Values (4, 'Trans mart', '456789123')

6. Insert kurir_06gggo (Id_kurir, Nama_Perusahaan, No_Telp)



Tugas Pertanyaan

TELP) VALUES (5, 'SuperIndo', '567891234')

7. Select 1 From dual;

- Tabel Pemasok

SQL > Insert all

2. Into Pemasok - obggo (Id- Pemasok , Nama- Perusahaan,

Alamat , kode- Pos , No-telp) values (1, 'Distri

'Indomart' , 'Lombok' , 6919 , '123456989')

3. Into Pemasok - obggo (Id- Pemasok , Nama- Perusahaan,

Alamat , kode- Pos , No-telp) values (2, 'Distri

'Alamet' , 'Makura' , 6971 , '234567891')

4. Into Pemasok - obggo (Id- Pemasok , Nama- Perusahaan,

Alamat , kode- Pos , No-telp) values (3, 'Distri

'Grant' , 'Sidoarjo' , 6916 , '345678912')

5. Into Pemasok - obggo (Id- Pemasok , Nama- Perusahaan,

Alamat , kode- Pos , No-telp) values (4, 'Distri

'Trans mart' , 'Jupiter' , 6911 , '456789123')

6. Into Pemasok - obggo (Id- Pemasok , Nama- Perusahaan,

Alamat , kode- Pos , No-telp) values (5, 'Distri

'SuperIndo' , 'Pluto' , 6901 , '567891234')

7. Select 1 From dual;

- Tabel kategori

SQL > Insert into kategori - obggo (Id- kategori , Nama-

kategori) values (1, 'makanan');

1 row created

SQL > Insert into kategori - obggo (Id- kategori , Nama-

kategori) values (2, 'minuman');

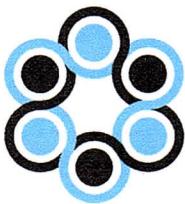
1 row created

SQL > Insert into kategori - OGGO (Id- kategori , Nama-

kategori) values (3, 'Buah-sayur');

1 row created

SQL > Insert into kategori - obggo (Id- kategori , Nama- kategori)



Tugas Pertanyaan

Values (4, 'Electronic');

1 row created.

SQL > Insert Into kategori_ogggo (Id-kategori, Nama-kategori) Values (5, 'Baju');

1 row created.

- Table Produk

SQL > Insert all

2. Into Produk_ogggo (Id-produk, Id-Pemasok, Id-kategori, Nama-produk, Harga-Satuan, Stock-Produk)
Values (1, 1, 1, 'Indomie', '2500', '50')

3. Into Produk_ogggo (Id-produk, Id-Pemasok, Id-kategori, Nama-produk, Harga-Satuan, Stock-Produk)
Values (2, 2, 2, 'Martimas', '2000', '40')

4. Into Produk_ogggo (Id-produk, Id-Pemasok, Id-kategori, Nama-produk, Harga-Satuan, Stock-Produk) Values (3, 3, 3, 'Jeruk', '5000', '100')

5. Into Produk_ogggo (Id-produk, Id-Pemasok, Id-kategori, Nama-produk, Harga-Satuan, Stock-Produk) Values (4, 4, 4, 'Samsung', '8000000', '35')

6. Into Produk_ogggo (Id-produk, Id-Pemasok, Id-kategori, Nama-produk, Harga-Satuan, Stock-Produk)
Values (5, 5, 5, 'Supreme', '35000', '95')

7. Select 1 From dual;

- Tabel Pemesanan

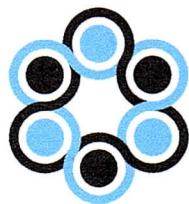
SQL > Insert all

2. Into Pemesanan_ogggo (Id-Pemesanan, Id-Kelanggan, Id-Pegawai, Id-kurir, Tanggal-Pemesanan, Tanggal-Pengiriman, Alamat-Pengiriman, Harga-Total) Values

(1, 1, 1, 1, to_date ('02/01/2017', 'dd/mm/yyyy'),

to_date ('02/01/2017', 'dd/mm/yyyy'), 'Botol',

'6000')



Tugas Pertanyaan

3. Info Pemesanan - Ogggo (Id-Pemesanan, Id-Pelanggan, Id-Pegawai, Id-kurir, Tanggal_Pemesanan, Tanggal_Pengiriman, Alamat_Pengiriman, Harga_Total) Values
(2,2,2,2, to_date('02/02/2017', 'dd/mm/yyyy'),
to_date('03/02/2017', 'dd/mm/yyyy'), 'Rungkut',
'50000')

4. Info Pemesanan - Ogggo (Id-Pemesanan, Id-Pelanggan, Id-Pejuggerai, Id-kurir, tanggal_Pemesanan, tanggal_Pengiriman, Alamat_Pengiriman, Harga_Total) Values
(3,3,3,3, to_date('03/03/2017', 'dd/mm/yyyy'),
to_date('04/03/2017', 'dd/mm/yyyy'), 'Sukorlo',
'153000')

5. Info Pemesanan - Ogggo (Id-Pemesanan, Id-Pelanggan, Id-Pegawai, Id-kurir, Tanggal_Pemesanan, Tanggal_Pengiriman, Alamat_Pengiriman, Harga_Total) Values
(4,4,4,4, to_date('04/04/2017', 'dd/mm/yyyy'),
to_date('05/04/2017', 'dd/mm/yyyy'), 'Sukorlo',
'100000')

6. Info Pemesanan - Ogggo (Id-Pemesanan, Id-Pelanggan, Id-Pegawai, Id-kurir, tanggal_Pemesanan, tanggal_Pengiriman, Alamat_Pengiriman, Harga_Total) Values
(5,5,5,5, to_date('05/05/2017', 'dd/mm/yyyy'),
to_date('06/05/2017', 'dd/mm/yyyy'), 'Sukorlo',
'200000')

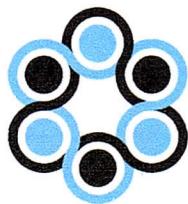
7. Select 1 from dual;

- tabel Detail_Pemesanan

SQL > Insert all

2. Info Detail - Pemesanan - Ogggo (Id_Produk, Id_Pemesanan, jumlah, Diskon) Values ('1','11','1','10')

3. Info Detail - Pemesanan - Ogggo (Id_Produk, Id_Pemesanan, jumlah, Diskon) Values ('2','2','2','20')



Tugas Pertanyaan

4. Insert Detail_Pemesanan_Gggo (Id_Prod, Id_Pemesanan, Jumlah, Diskon) Values ('3', '3', '3', '30')

5. Insert Detail_Pemesanan_Gggo (Id_Prod, Id_Pemesanan, Jumlah, Diskon) Values ('4', '4', '4', '40')

6. Insert Detail_Pemesanan_Gggo (Id_Prod, Id_Pemesanan, Jumlah, Diskon) Values ('5', '5', '5', '50')

7. Select 1 From Dual;

2. Updatekan 2 Record nama depan dan nama belakang pada table Pegawai

- Tampilkan table yang ingin di update terlebih dahulu

SQL > SELECT * FROM Pegawai_Gggo

ID_Pegawai Nama_Depan Nama_Belakang Tanggal_L_Alamat

Kode_Pos No_Telp

1. Lalaq Manoban 01-oct-98 Boboh 69198 123456789

2. Kim Angga 25-jan-97 Nlaura 6971 234567891

3. Bakti Sebakti 08-jan-95 Sidoarjo 6916 345678912

4. Rini Atni 06-nov-98 Rungkut 6791 456789123

5. Beni Pratama 31-may-98 Sukolilo 6901 867891234

- Update tabel yang diinginkan

SQL > UPDATE Pegawai_Gggo

2. SET nama_Depan = 'Kim',

3. nama_Belakang = 'jseid',

4. WHERE Id_Pegawai = 1;

1 row updated

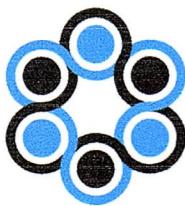
SQL >

SQL > UPDATE Pegawai_Gggo

2. SET nama_Depan = 'Lee',

3. nama_Belakang = 'kurniawan'

4. WHERE Id_Pegawai = 2;



Tugas Pertanyaan

1 row updated

- Cek ulang pada tabel yang telah diupdate

SQL > SELECT * FROM Pegawai - 06990

ID-Pegawai Nama-Depan Nama-Belakang Tanggal-L Kode-Pes

NO-TELP

1. Kim Jisoo 01-oct-98 Babon 69198 123456789

2. Lee Kunrawan 25-jan-97 Morauna 6971 234567891

3. Bakti Sebakti 08-jan-95 Sidoarjo 6916 2345678912

4. Rini Atini 06-nov-98 Rungkut 6911 23456789123

5. Beni Pratama 31-may-98 Sukorlo 6901 567891234

3. Hapus dua record pada tabel detar_Pemesanan !

- Tampilkan tabel yang ingin dihapus

SQL > SELECT * FROM detar_Pemesanan - 06990 ;

ID-Product ID-Pemesanan jumlah Diskon

1	1	1	10
2	2	2	20
3	3	3	30
4	4	4	40
5	5	5	50

- Hapus tabel yang dituju

SQL > DELETE FROM detar_Pemesanan - 06990

2. WHERE Id-Product = 4;

1 row deleted

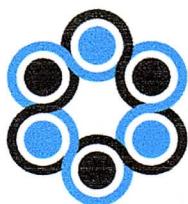
SQL>

SQL > DELETE FROM detar_Pemesanan - 06990

2. WHERE Id-Product = 5;

1 row deleted.

- Cek ulang tabel yang telah dihapus



Tugas Pertanyaan

SQL > SELECT * FROM detail_Pemesanan_06gg0;

ID_Proses	ID_Pemesanan	Jumlah	Diskon
1	1	1	10
2	2	2	20
3	3	3	30

4. Buatlah query dimana menampilkan kode_pos, alamat, no_telp, tanggal_lahir dari setiap Pegawai yang nama depannya memiliki huruf a (semua) dan mempunyai nama belakang e (semua) dengan Id_Pegawai kurang dari 5 ?

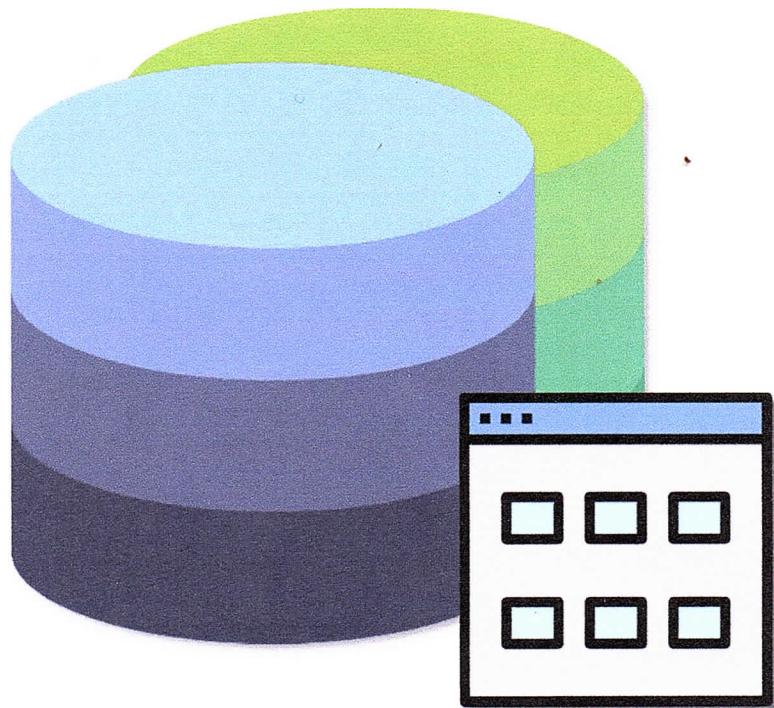
SQL > SELECT kode_pos, alamat, no_telp, tanggal_lahir
from Pegawai_06gg0

2. Untuk Nama-depan LIKE '%a%' AND nama-
belakang LIKE '%e%' AND Id_Pegawai < 5 ;

kode_pos Alamat NO_TELP Tanggal_L

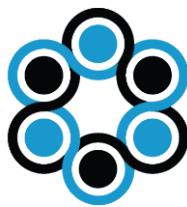
6916 Sracargo 3815678912 08-Jan-95

LAPORAN PRAKTIKUM



PRAKTIKUM BASIS DATA PERIODE V - Tahun 2019/2020

NAMA : Mochamad khoirulhusn
NPM : 06.2010.1.06gg0
MODUL : II



Tugas Praktikum

Soal Praktikum

1. Insert data pada setiap tabel yang telah Anda buat :
 - a. Single insert dengan 2 data
 - b. Multiple insert dengan 3 data
 - c. Terapkan insert dengan sequence yang telah Anda buat di Modul 1
2. Update 5 data pada tabel yang berbeda :
 - a. Semua data pada suatu kolom pada tabel
 - b. Menerapkan where clause
 - c. Menerapkan like
 - d. Menerapkan AND, OR dan NOT minimal 2 operator pada satu baris query
3. Delete minimal 3 data pada salah satu tabel dengan menerapkan 1 klausa dan 2 operator pada satu baris query yang berbeda pada setiap data
4. Aturlah transaksi dengan menerapkan :
 - a. Save Point
 - b. Commit
 - c. Rollback
5. Terapkan select dengan menerapkan :
 - a. Order By
 - b. Group By

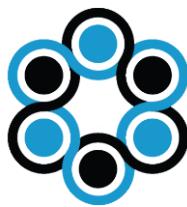
Langkah Pertama

Jika kita memasukan data secara satu persatu kita bisa menggunakan perintah

```
insert into nama_table (nama_kolom1, nama_kolom2, ... ,nama_kolom_ke-n)
values (isi_kolom1, isi_kolom2, ... , isi_kolom_ke-N);
```

Jika kita memasukan data secara bersama-sama maka kita menggunakan perintah

Insert all



Tugas Praktikum

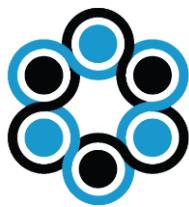
```
into nama_table (nama_kolom1, nama_kolom2, ... ,nama_kolom_ke-n) values  
(isi_kolom1, isi_kolom2, ... , isi_kolom_ke-N)  
into nama_table (nama_kolom1, nama_kolom2, ... ,nama_kolom_ke-n) values  
(isi_kolom1, isi_kolom2, ... , isi_kolom_ke-N)  
select 1 from dual;
```

Query

```
insert into profil_aluminium  
(id_profil, nama_profil, warna)  
values (1, 'strandart', 'iron grey');  
insert into profil_aluminium  
(id_profil, nama_profil, warna)  
values (2, 'standart', 'white');  
  
insert all  
into profil_aluminium (id_profil,  
nama_profil, warna)  
values (3, 'standart', 'black sahara')  
into profil_aluminium (id_profil,  
nama_profil, warna)  
values (4, 'penta 40', 'iron grey')  
into profil_aluminium (id_profil,  
nama_profil, warna)  
values (5, 'penta 40', 'white')  
select 1 from dual;
```

Screenshot

```
SQL> insert into profil_aluminium  
2  (id_profil, nama_profil, warna)  
3  values (1, 'strandart', 'iron grey');  
  
1 row created.  
  
SQL> insert into profil_aluminium  
2  (id_profil, nama_profil, warna)  
3  values (2, 'standart', 'white');  
  
1 row created.  
  
SQL>  
SQL> insert all  
2  into profil_aluminium (id_profil, nama_profil, warna)  
3  values (3, 'standart', 'black sahara')  
4  into profil_aluminium (id_profil, nama_profil, warna)  
5  values (4, 'penta 40', 'iron grey')  
6  into profil_aluminium (id_profil, nama_profil, warna)  
7  values (5, 'penta 40', 'white')  
8  select 1 from dual;  
  
3 rows created.
```



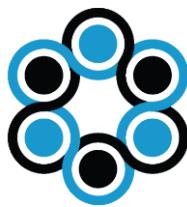
Tugas Praktikum

Query

```
insert into marketing
(id_marketing, nama_marketing, no_telp)
values (id_marketing.nextval, 'rahmad',
'123456789');
insert into marketing
(id_marketing, nama_marketing, no_telp)
values (id_marketing.nextval, 'artani',
'234567891');

CREATE FUNCTION get_seq RETURN NUMBER IS
BEGIN
    RETURN id_marketing.nextval;
END;
/

insert all
into marketing (id_marketing, nama_marketing,
no_telp)
values (get_seq, 'hendro', '345678912')
into marketing
(id_marketing, nama_marketing, no_telp)
values (get_seq, 'france', '456789123')
into marketing (id_marketing, nama_marketing,
no_telp)
values (get_seq, 'yudha', '567891234')
select 1 from dual;
```



Tugas Praktikum

Screenshot

```
SQL> insert into marketing
  2 (id_marketing, nama_marketing, no_telp)
  3 values (id_marketing.nextval, 'rahmad', '123456789');

1 row created.

SQL> insert into marketing
  2 (id_marketing, nama_marketing, no_telp)
  3 values (id_marketing.nextval, 'artani', '234567891');

1 row created.

SQL> CREATE FUNCTION get_seq RETURN NUMBER IS
  2 BEGIN
  3   RETURN id_marketing.nextval;
  4 END;
  5 /

Function created.

SQL>
SQL> insert all
  2 into marketing (id_marketing, nama_marketing, no_telp)
  3 values (get_seq, 'hendro', '345678912')
  4 into marketing
  5 (id_marketing, nama_marketing, no_telp)
  6 values (get_seq, 'france', '456789123')
  7 into marketing (id_marketing, nama_marketing, no_telp)
  8 values (get_seq, 'yudha', '567891234')
  9 select 1 from dual;

3 rows created.
```

Query

```
insert into arsitek
(id_arsitek, nama_arsitek, telp_kantor)
values (1, 'santoso', '1111111111');
insert into arsitek
(id_arsitek, nama_arsitek, telp_kantor)
values (2, 'budi kurniawan', '2222222222');

insert all
into arsitek
(id_arsitek, nama_arsitek, telp_kantor)
values (3, 'bambang handoko', '3333333333')
into arsitek
(id_arsitek, nama_arsitek, telp_kantor)
values (4, 'margono', '4444444444')
into arsitek
(id_arsitek, nama_arsitek, telp_kantor)
values (5, 'yudha prasetya', '5555555555')
select 1 from dual;
```



Tugas Praktikum

Screenshot

```
SQL> insert into arsitek
  2  (id_arsitek, nama_arsitek, telp_kantor)
  3  values (1, 'santoso', '1111111111');

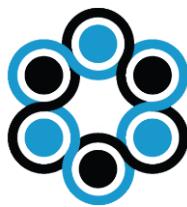
1 row created.

SQL> insert into arsitek
  2  (id_arsitek, nama_arsitek, telp_kantor)
  3  values (2, 'budi kurniawan', '2222222222');

1 row created.

SQL>
SQL> insert all
  2  into arsitek
  3  (id_arsitek, nama_arsitek, telp_kantor)
  4  values (3, 'bambang handoko', '3333333333')
  5  into arsitek
  6  (id_arsitek, nama_arsitek, telp_kantor)
  7  values (4, 'margono', '4444444444')
  8  into arsitek
  9  (id_arsitek, nama_arsitek, telp_kantor)
10  values (5, 'yudha prasetya', '5555555555')
11  select 1 from dual;

3 rows created.
```

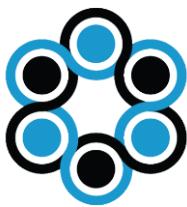


Tugas Praktikum

Query

```
insert into proyek
(nama_owner, alamat_proyek,
alamat_penyuratan, no_telp, id_arsitek,
id_marketing, id_profil)
values ('kardono', 'graha family blok-k 62',
'Dusan', '1111111111', 1, 2, 2);
insert into proyek
(nama_owner, alamat_proyek,
alamat_penyuratan, no_telp, id_arsitek,
id_marketing, id_profil)
values ('sudarmono', 'graha family blok-h
18', 'graha family blok-h 18', '222222222',
5, 1, 1);

insert all
into proyek
(nama_owner, alamat_proyek,
alamat_penyuratan, no_telp, id_arsitek,
id_marketing, id_profil)
values ('yunus', 'BDG N-45', 'BK',
'333333333', 2, 1, 1)
into proyek
(nama_owner, alamat_proyek,
alamat_penyuratan, no_telp, id_arsitek,
id_marketing, id_profil)
values ('alex', 'Citraland blok-i4 20',
'embong malang', '444444444', 1, 2, 1)
into proyek
(nama_owner, alamat_proyek,
alamat_penyuratan, no_telp, id_arsitek,
id_marketing, id_profil)
values ('yos tanoyo', 'Citraland blok-h2 1',
'embong malang', '555555555', 1, 2, 2)
select 1 from dual;
```



Tugas Praktikum

Screenshot

```
SQL> insert into proyek
  2  (nama_owner, alamat_proyek, alamat_penyuratan, no_telp, id_arsitek, id_marketing, id_profil)
  3  values ('kardono', 'graha family blok-k 62', 'Dusan', '111111111', 1, 2, 2);
1 row created.

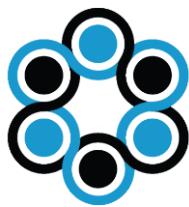
SQL> insert into proyek
  2  (nama_owner, alamat_proyek, alamat_penyuratan, no_telp, id_arsitek, id_marketing, id_profil)
  3  values ('sudarmono', 'graha family blok-h 18', 'graha family blok-h 18', '222222222', 5, 1, 1);
1 row created.

SQL>
SQL> insert all
  2  into proyek
  3  (nama_owner, alamat_proyek, alamat_penyuratan, no_telp, id_arsitek, id_marketing, id_profil)
  4  values ('yunus', 'BDG N-45', 'BK', '333333333', 2, 1, 1)
  5  into proyek
  6  (nama_owner, alamat_proyek, alamat_penyuratan, no_telp, id_arsitek, id_marketing, id_profil)
  7  values ('alex', 'Citraland blok-i4 20', 'embong malang', '444444444', 1, 2, 1)
  8  into proyek
  9  (nama_owner, alamat_proyek, alamat_penyuratan, no_telp, id_arsitek, id_marketing, id_profil)
 10  values ('yos tanoyo', 'Citraland blok-h2 1', 'embong malang', '555555555', 1, 2, 2)
 11  select 1 from dual;

3 rows created.
```

Analisa :

Saat kita memasukan data yang bersifat multiple atau yang biasa kita sebut dengan multiple insert yang didalam sebuah table terdapat sebuah sequence, maka kita tidak bisa memasukan data seperti multiple insert yang semestinya. Oleh karena itu kita di haruskan membuat sebuah function yang dapat mempermudah kita saat pengisian kolom sequence anda bisa lihat pada saat saya memasukan data di table marketing.



Tugas Praktikum

Langkah Kedua

Pada saat kita update semua data pada suatu kolom pada table kita bisa menggunakan perintah

**Update from nama_table set nama_kolom = 'isi_data_terbaru' where
isi_table_lama operasional_kondisi kondisi;**

Dalam penerapan where clause kita bisa menggunakan cara seperti ini :

Update from nama_table set nama_kolom =

Penerapan LIKE adalah seperti ini

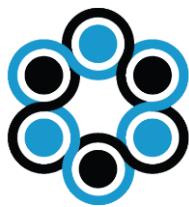
**Update from nama_table set nama_kolom = 'isi_data_baru' where
nama_kolom like '%y%';**

Penerapan dalam AND, OR dan NOT dan minimal 2 operator di dalam satu baris query

**Update from nama_table set nama_kolom = 'isi_data_baru' where
nama_kolom = 'isi_data_lama' operator_logika nama_kolom
operator_kondisi 'isi_data_lama'**

Query

```
update profil_aluminium set nama_profil =  
'penta 50' , warna = 'antico brown' where  
id_profil = 4;  
  
update profil_aluminium set nama_profil =  
'penta 50' , warna = 'black sahara' where  
id_profil = 5;  
  
update marketing set no_telp = '999999999'  
where nama_marketing = 'france';  
  
update arsitek set telp_kantor = '999999999'  
where nama_arsitek like '%y%';  
  
update proyek  
set alamat_penyuratan = 'Dusan'  
where nama_owner = 'yos tanoyo'  
AND id_marketing = 2  
OR alamat_proyek = 'Citraland blok-h2 1';
```



Tugas Praktikum

Screenshot

```
SQL> update profil_aluminium set nama_profil = 'penta 50' , warna = 'antico brown' where id_profil = 4;
1 row updated.

SQL> update profil_aluminium set nama_profil = 'penta 50' , warna = 'black sahara' where id_profil = 5;
1 row updated.

SQL> select * from profil_aluminium;
select * from profil_aluminium
*
ERROR at line 1:
ORA-00942: table or view does not exist

SQL> select * from profil_aluminium;
ID_PROFIL NAMA_PROFIL          WARNA
-----  -----
1 strandart           iron grey
2 standart            white
3 standart            black sahara
4 penta 50             antico brown
5 penta 50             black sahara

SQL> update marketing set no_telp = '999999999' where nama_marketing = 'france';
1 row updated.

SQL> select * from marketing;
ID_MARKETING NAMA_MARKETING      NO_TELP
-----  -----
4 france              999999999
5 yudha               567891234
1 rahmad              123456789
2 artani               234567891
3 hendro              345678912

SQL> update arsitek set telp_kantor = '999999999' where nama_arsitek like '%y%';
1 row updated.

SQL> select * from arsitek;
ID_ARSITEK NAMA_ARSITEK          TELP_KANTOR
-----  -----
1 santoso              111111111
2 budi kurniawan        222222222
3 bambang handoko       333333333
4 margono               444444444
5 yudha prasetya        999999999

SQL> update proyek set alamat_penyuratannya = 'Dusan' where nama_owner = 'yos tanoyo' AND id_marketing = 2 OR alamat_proyek = 'Citraland blok-h2 1';
1 row updated.

SQL> select * from proyek;
NAMA_OWNER ID_PROFIL          ALAMAT_PROYEK          ALAMAT_PENYURATAN      NO_TELP ID_ARSITEK ID_MARKETING
-----  -----
kardono     2                  graha family blok-k 62   Dusan                 111111111  1       2
sudarmono   1                  graha family blok-h 18   graha family blok-h 18  222222222  5       1
yunus       1                  Bdg N-45                BK                   333333333  2       1
alex         1                  Citraland blok-i4 20   embong malang          444444444  1       2
yos tanoyo  2                  Citraland blok-h2 1      Dusan                 555555555  1       2
```

Analisa :

Untuk mengurangi kesalahan pada sebuah query langkah baiknya kita harus melihat isi pada table yang dituju. Dalam penggunaan operator logika atau pun operator kondisi sebaiknya di pahami terlebih dahulu apa yang kita inginkan supaya mempermudah kita untuk membuat sebuah query yang memiliki banyak operator



Tugas Praktikum

Langkah Ketiga

Cara menghapus 3 data pada salah satu table dengan query

**Delete from nama_table where nama_kolom operator_kondisi 'isi_data'
operator_logika nama_kolom operator_kondisi 'isi_data'**

Query

```
delete from marketing
where nama_marketing != 'rahmad'
AND nama_marketing != 'artani'
OR id_marketing > 2;
```

Screenshot

```
SQL> delete from marketing
  2  where nama_marketing != 'rahmad'
  3  AND nama_marketing != 'artani'
  4  OR id_marketing > 2;

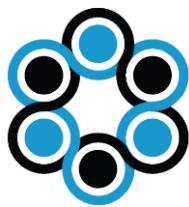
3 rows deleted.

SQL> select * from marketing;

ID_MARKETING NAMA_MARKETING          NO_TELP
-----  -----
      1 rahmad                      123456789
      2 artani                      234567891
```

Analisa :

Dalam pembuatan query untuk delete banyak data, kita harus benar-benar paham dengan kondisi apa yang kita pakai supaya menghindari apa yang tidak kita inginkan



Tugas Praktikum

Langkah Keempat

untuk kita membuat savepoint kita bisa menggunakan cara berikut :

Savepoint nama_savepoint;

Saat kita akan mengulangi dari savepoint saat sudah terjadi perubahan dengan cara berikut :

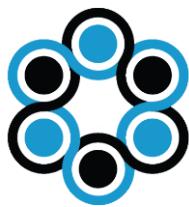
Rollback to savepoint nama_savepoint;

Untuk kita men-commit bisa menggunakan cara ini

Commit;

Query

```
savepoint pertama;
rollback to savepoint pertama;
commit;
```



Tugas Praktikum

Screenshot

```
SQL> select * from marketing;
ID_MARKETING NAMA_MARKETING          NO_TELP
-----  -----
1 rahmad                         123456789
2 artani                          234567891

SQL> savepoint pertama;
Savepoint created.

SQL> update marketing
  2 set no_telp = '111111111'
  3 where id_marketing = 1;

1 row updated.

SQL> select * from marketing;
ID_MARKETING NAMA_MARKETING          NO_TELP
-----  -----
1 rahmad                         111111111
2 artani                          234567891

SQL> rollback savepoint pertama;
rollback savepoint pertama
*
ERROR at line 1:
ORA-02181: invalid option to ROLLBACK WORK

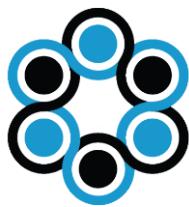
SQL> rollback to savepoint pertama;
Rollback complete.

SQL> select * from marketing;
ID_MARKETING NAMA_MARKETING          NO_TELP
-----  -----
1 rahmad                         123456789
2 artani                          234567891

SQL> commit;
Commit complete.
```

Analisa :

Kegunaan savepoint adalah menentukan sebuah titik aman, jika pada suatu saat kita melakukan kesalahan saat membuat query dan bertambah berubahnya sebuah data maka kita bisa menuju titik aman yang sudah kita buat dengan savepoint. Mengulangi ke titik aman kita menggunakan sebuah query rollback dan untuk commit adalah mem-fixkan sebuah data



Tugas Praktikum

Langkah Kelima

Cara menggunakan order by dan group by

Query

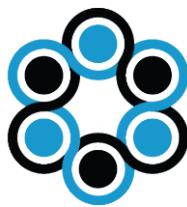
```
select nama_owner, alamat_penyuratan from  
proyek order by id_marketing;  
  
select warna, count(nama_profil) AS "banyak"  
from profil_aluminium  
group by warna;
```

Screenshot

```
SQL> select nama_owner, alamat_penyuratan from proyek order by id_marketing;  
NAMA_OWNER          ALAMAT_PENYURATAN  
-----  
yunus                BK  
sudarmono             graha family blok-h 18  
yos tanoyo            Dusan  
alex                  embong malang  
kardono               Dusan  
  
SQL> select warna, count(nama_profil) AS "banyak"  
  2  from profil_aluminium  
  3  group by warna;  
WARNA                 banyak  
-----  
black sahara           2  
white                  1  
antico brown           1  
iron grey               1
```

Analisa :

Order by berguna untuk melakukan request ke tabel yang dituju.
Group by berguna untuk mengelompokan sebuah data pada tabel



Laporan Sementara

Soal Praktikum

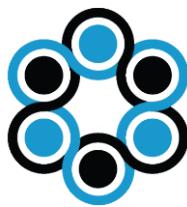
1. Insert data pada setiap tabel yang telah Anda buat :
 - a. Single insert dengan 2 data
 - b. Multiple insert dengan 3 data
 - c. Terapkan insert dengan sequence yang telah Anda buat di Modul 1
2. Update 5 data pada tabel yang berbeda :
 - a. Semua data pada suatu kolom pada tabel
 - b. Menerapkan where clause
 - c. Menerapkan like
 - d. Menerapkan AND, OR dan NOT minimal 2 operator pada satu baris query
3. Delete minimal 3 data pada salah satu tabel dengan menerapkan 1 klausa dan 2 operator pada satu baris query yang berbeda pada setiap data
4. Aturlah transaksi dengan menerapkan :
 - a. Save Point
 - b. Commit
 - c. Rollback
5. Terapkan select dengan menerapkan :
 - a. Order By
 - b. Group By

Langkah Pertama

menginput data pada semua table yang ada.

Menginput meinput dengan perintah `insert into (field 1 table, filed 2 table,..)`
`values (isi 1, isi 2,..);`

Jika kita menginput table langsung banyak atau biasa di sebut dengan multiple insert bisa menggunakan **insert all**



Laporan Sementara

Query

```
insert into profil_aluminium (id_profil, nama_profil, warna) values (1, 'strandart', 'iron grey');
insert into profil_aluminium (id_profil, nama_profil, warna) values (2, 'standart', 'white');

insert all
into profil_aluminium (id_profil, nama_profil, warna)
values (3, 'standart', 'black sahara')
into profil_aluminium (id_profil, nama_profil, warna)
values (4, 'penta 40', 'iron grey')
into profil_aluminium (id_profil, nama_profil, warna)
values (5, 'penta 40', 'white')
select 1 from dual;
```

Screenshot

```
SQL> insert into profil_aluminium (id_profil, nama_profil, warna) values (1, 'strandart', 'iron grey');

1 row created.

SQL> insert into profil_aluminium (id_profil, nama_profil, warna) values (2, 'standart', 'white');

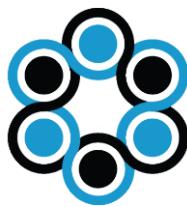
1 row created.
```

```
SQL> insert all
  2  into profil_aluminium (id_profil, nama_profil, warna) values (3, 'standart', 'black sahara')
  3  into profil_aluminium (id_profil, nama_profil, warna) values (4, 'penta 40', 'iron grey')
  4  into profil_aluminium (id_profil, nama_profil, warna) values (5, 'penta 40', 'white')
  5  select 1 from dual;

3 rows created.
```

Analisa

(Ketik hasil analisa disini)



Laporan Sementara

Query

```
insert into marketing (id_marketing, nama_marketing,
no_telp) values (id_marketing.nextval, 'rahmad',
'123456789');
insert into marketing (id_marketing, nama_marketing,
no_telp) values (id_marketing.nextval, 'artani',
'234567891');

insert into marketing (id_marketing, nama_marketing,
no_telp) values (id_marketing.nextval, 'hendro',
'345678912');
insert into marketing (id_marketing, nama_marketing,
no_telp) values (id_marketing.nextval, 'france',
'456789123');
insert into marketing (id_marketing, nama_marketing,
no_telp) values (id_marketing.nextval, 'yudha',
'567891234');
```

Screenshot

```
SQL> insert into marketing (id_marketing, nama_marketing, no_telp) values (id_marketing.nextval, 'rahmad', '123456789');
1 row created.

SQL> insert into marketing (id_marketing, nama_marketing, no_telp) values (id_marketing.nextval, 'artani', '234567891');
1 row created.

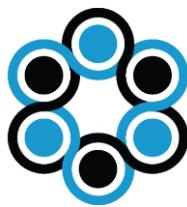
SQL> insert into marketing (id_marketing, nama_marketing, no_telp) values (id_marketing.nextval, 'hendro', '345678912');
1 row created.

SQL> insert into marketing (id_marketing, nama_marketing, no_telp) values (id_marketing.nextval, 'france', '456789123');
1 row created.

SQL> insert into marketing (id_marketing, nama_marketing, no_telp) values (id_marketing.nextval, 'yudha', '567891234');
1 row created.
```

Analisa :

(Ketik hasil analisa disini)



Laporan Sementara

Query

```
insert into arsitek (id_arsitek, nama_arsitek,
telp_kantor) values (1, 'santoso', '1111111111');
insert into arsitek (id_arsitek, nama_arsitek,
telp_kantor) values (2, 'budi kurniawan',
'2222222222');

insert all
into arsitek (id_arsitek, nama_arsitek, telp_kantor)
values (3, 'bambang handoko', '3333333333')
into arsitek (id_arsitek, nama_arsitek, telp_kantor)
values (4, 'margono', '4444444444')
into arsitek (id_arsitek, nama_arsitek, telp_kantor)
values (5, 'yudha prasetya', '5555555555')
select 1 from dual;
```

Screenshot

```
SQL> insert into arsitek (id_arsitek, nama_arsitek, telp_kantor) values (1, 'santoso', '1111111111');
1 row created.

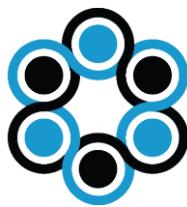
SQL> insert into arsitek (id_arsitek, nama_arsitek, telp_kantor) values (2, 'budi kurniawan', '2222222222');
1 row created.

SQL>
SQL> insert all
  2  into arsitek (id_arsitek, nama_arsitek, telp_kantor) values (3, 'bambang handoko', '3333333333')
  3  into arsitek (id_arsitek, nama_arsitek, telp_kantor) values (4, 'margono', '4444444444')
  4  into arsitek (id_arsitek, nama_arsitek, telp_kantor) values (5, 'yudha prasetya', '5555555555')
  5  select 1 from dual;

3 rows created.
```

Analisa :

(Ketik hasil analisa disini)



Laporan Sementara

Query

```
insert into proyek (alamat_proyek,
nama_owner, alamat_penyuratan, no_telp,
id_arsitek, id_marketing, id_profil)
values ('graha family blok-k 62', 'kardono',
'Dusan', '1111111111', 1, 3, 2);
insert into proyek (alamat_proyek,
nama_owner, alamat_penyuratan, no_telp,
id_arsitek, id_marketing, id_profil)
values ('graha family blok-h 18',
'sudarmono', 'graha family blok-h 18',
'2222222222', 5, 2, 1);
insert into proyek (alamat_proyek,
nama owner, alamat penyuratan, no telp.
```

Screenshot

```
SQL> insert into proyek (alamat_proyek, nama_owner, alamat_penyuratan, no_telp, id_arsitek, id_marketing, id_profil)
2 values ('graha family blok-k 62', 'kardono', 'Dusan', '1111111111', 1, 3, 2);

1 row created.

SQL> insert into proyek (alamat_proyek, nama_owner, alamat_penyuratan, no_telp, id_arsitek, id_marketing, id_profil)
2 values ('graha family blok-h 18', 'sudarmono', 'graha family blok-h 18', '2222222222', 5, 2, 1);

1 row created.

SQL> insert into proyek (alamat_proyek, nama_owner, alamat_penyuratan, no_telp, id_arsitek, id_marketing, id_profil) values ('B06 N-45', 'yunus', 'BK', '3333333333', 2, 1);

1 row created.

SQL> insert into proyek (alamat_proyek, nama_owner, alamat_penyuratan, no_telp, id_arsitek, id_marketing, id_profil) values ('Citraland blok-i4 20', 'alex', 'embong malang', '4444444444', 1, 3, 1);

1 row created.
```

Analisa :

(Ketik hasil analisa disini)



Laporan Sementara

Langkah Kedua

update dengan perintah update nama_table set nama_kolom_yang_mau_diupdate = isi_yang_mau_update where = kondisinya apa

Query

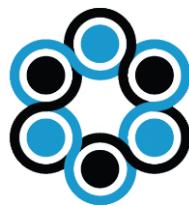
```
update from profil_aluminium set nama_profil  
= 'penta 50' , warna = 'antico brown' where  
id_profil = 4;  
update from profil_aluminium set nama_profil  
= 'penta 50' , warna = 'black sahara' where  
id_profil = 5;
```

Screenshot

```
SQL> update profil_aluminium set nama_profil = 'penta 50' , warna = 'antico brown' where id_profil = 4;  
1 row updated.  
  
SQL> update profil_aluminium set nama_profil = 'penta 50' , warna = 'black sahara' where id_profil = 5;  
1 row updated.
```

Analisa :

(Ketik hasil analisa disini)



Laporan Sementara

Query

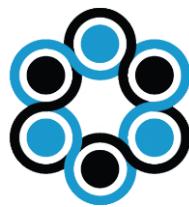
```
update from marketing set no_telp =  
'999999999' where nama_marketing = 'france';
```

Screenshot

```
SQL> update marketing set no_telp = '999999999' where nama_marketing = 'france';  
1 row updated.
```

Analisa :

(Ketik hasil analisa disini)



Laporan Sementara

Query

```
update from arsitek set no_kantor =
'999999999' where nama_arsitek like %y%;
```

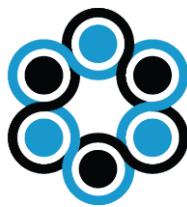
Screenshot

```
SQL> update arsitek set telp_kantor = '999999999' where nama_arsitek like '%y%';

1 row updated.
```

Analisa :

(Ketik hasil analisa disini)



Laporan Sementara

Query

```
update from proyek set alamat_penyuratan =
Dusan where nama_owner = 'yos tanoyo' AND
id_marketing = 2 OR alamat_proyek =
'Citraland blok-h2 1';
```

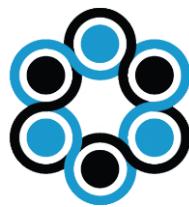
Screenshot

```
SQL> update proyek set alamat_penyuratan = 'Dusan' where nama_owner = 'alex' AND id_marketing = 3 OR alamat_proyek = 'Citraland blok-i4 20';
update proyek set alamat_penyuratan = 'Dusan' where nama_owner = 'alex' AND id_marketing = 3 OR alamat_proyek = 'Citraland blok-i4 20'
*
ERROR at line 1:
ORA-00001: unique constraint (KHOIRUL_HUDA.06990.SYS_C0011580) violated
```

Activate

Analisa :

(Ketik hasil analisa disini)



Laporan Sementara

Langkah Ketiga

(Ketik penjelasan langkah – langkah disini)

Query

```
delete from marketing
where nama_marketing != 'rahmad'
AND nama_marketing != 'artani'
OR id_marketing < 3;
```

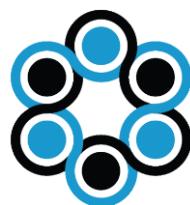
Screenshot

```
SQL> delete from marketing
  2  where nama_marketing != 'rahmad'
  3  AND nama_marketing != 'artani'
  4  OR id_marketing > 3;

3 rows deleted.
```

Analisa :

(Ketik hasil analisa disini)



Laporan Sementara

Langkah Keempat

(Ketik penjelasan langkah – langkah disini)

Query

```
savepoint pertama;  
commit;  
rollback to pertama;
```

Screenshot

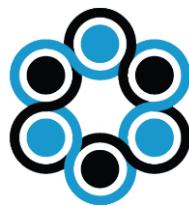
```
SQL> savepoint fisrt;  
Savepoint created.
```

```
SQL> commit  
 2 ;  
Commit complete.
```

```
SQL> rollback to first;  
rollback to first  
*  
ERROR at line 1:  
ORA-01086: savepoint 'FIRST' never established in this session or is invalid
```

Analisa :

(Ketik hasil analisa disini)



Laporan Sementara

Langkah Kelima

(Ketik penjelasan langkah – langkah disini)

Query

```
select nama_owner, alamat_proyek from proyek  
order by id_marketing;
```

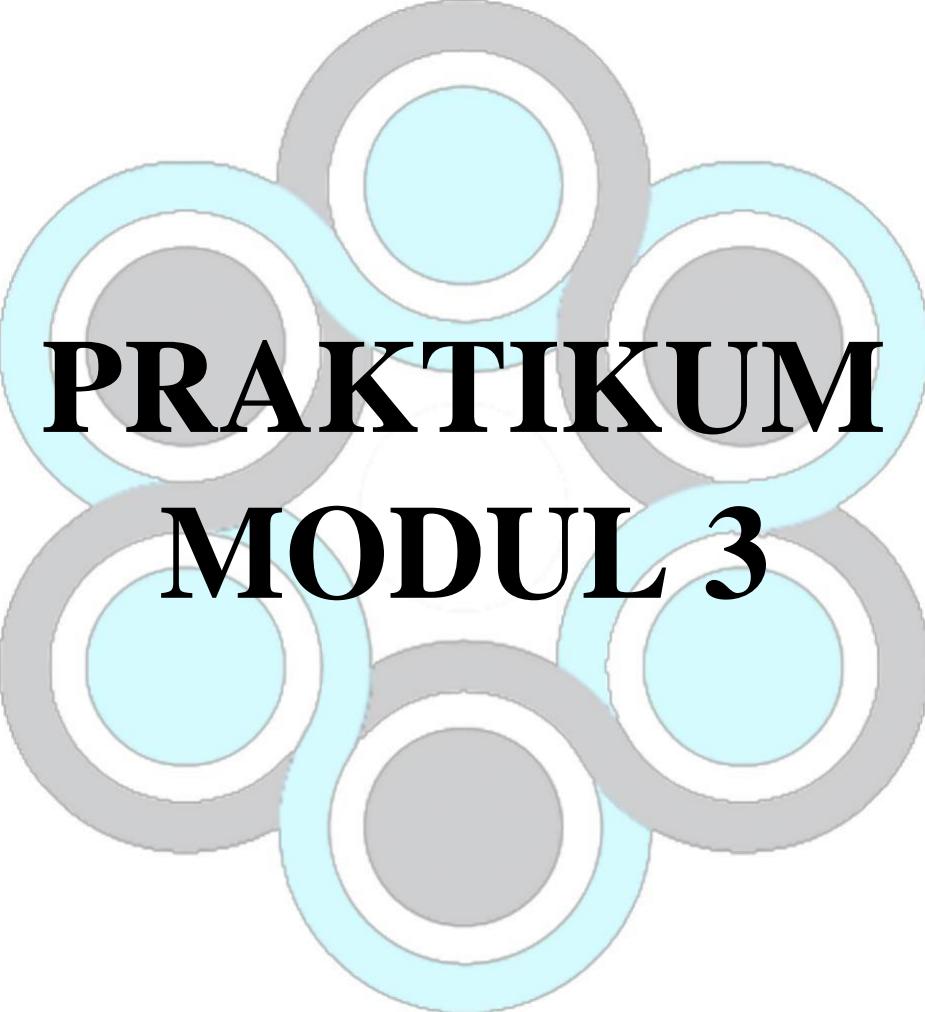
```
select nama_owner from proyek group by  
nama_owner;
```

Screenshot

```
SQL> select nama_owner, alamat_proyek from proyek order by id_marketing;  
  
NAMA_OWNER  
-----  
ALAMAT_PROYEK  
-----  
sudarmono  
graha family blok-h 18  
  
yunus  
BDG N-45  
  
kardono  
graha family blok-k 62  
  
SQL> select nama_owner from proyek group by nama_owner;  
  
NAMA_OWNER  
-----  
sudarmono  
yunus  
kardono
```

Analisa :

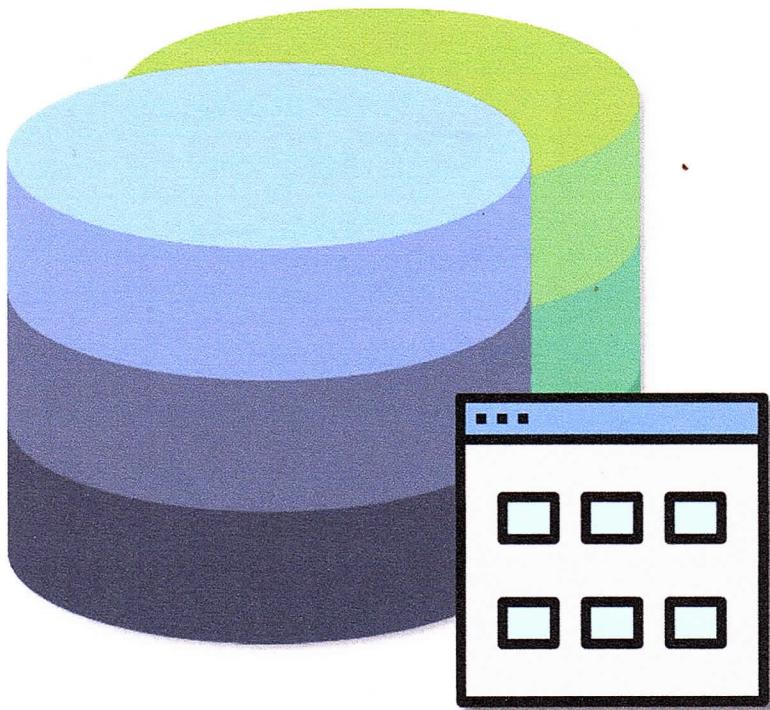
(Ketik hasil analisa disini)



PRAKTIKUM MODUL 3

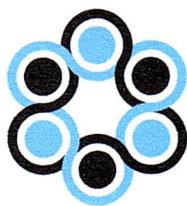
FAKULTAS TEKNIK ELEKTRO DAN
TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI ADHI TAMA
SURABAYA
2020

LAPORAN PENDAHULUAN



PRAKTIKUM BASIS DATA PERIODE V - Tahun 2019/2020

NAMA : M. Khorul Huda
NPM : 06.2018.1.0690
MODUL : III



Tugas Pertanyaan

1. Buatlah sebuah query untuk menampilkan data berikut :

ID - Pemesanan TOTAL_ITEM_YG_DIPESAN

3 3

- Tampilkan data terlebih dahulu pada tabel yang dituju

SQL > Select * from detail_Pemesanan_06990;

ID-PRODUK ID-PEMESANAN JUMLAH DISKON

1	1	1	10
2	2	2	20
3	3	3	30

- Tuliskan query seperti dibawah ini

SQL > SELECT Id_Pemesanan , (SELECT SUM(Jumlah) FROM detail_Pemesanan_06990

2. WHERE detail_Pemesanan_06990 . Id_Pemesanan = Pemesanan_06990 . Id_Pemesanan)

3. AS total_item_yg_dipesan FROM Pemesanan_06990
WHERE Id_Pemesanan = 3;

2. Buatlah sebuah query untuk menampilkan data berikut :

ID-KATEGORI TERBANYAK TERENDAH

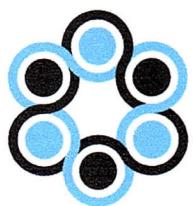
1	90	10
2	40	40
4	35	35
5	95	95
3	100	100

- Query untuk menampilkan seperti diatas adalah :

SQL > SELECT Id_kategori , MAX(STOK_PRODUK) AS

2. terbanyak , MIN(STOK_PRODUK) AS terendah FROM Produk_06990

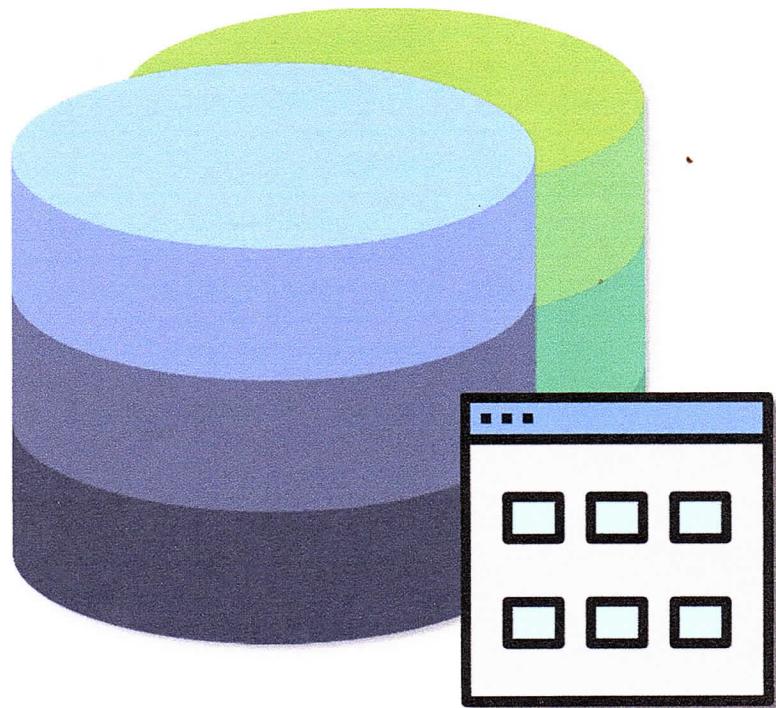
3. GROUP BY Id_kategori ;



Tugas Pertanyaan

- Pada query diatas akan merekap dan menampilkan nilai terendah dan tertinggi gaji produk yang terdapat di tabel produk.

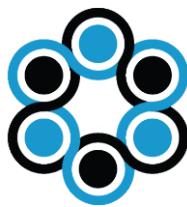
LAPORAN PRAKTIKUM



**PRAKTIKUM BASIS DATA
PERIODE V - Tahun 2019/2020**

NAMA : Mochamad Khoirul Huda
NPM : 06.2018.1.06990
MODUL : III

--



Tugas Praktikum

Soal Praktikum

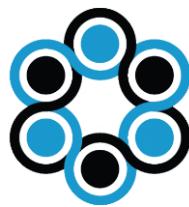
1. Insert minimal 5 data pada setiap tabel
2. Buatlah query dengan menerapkan fungsi aggregate :
 - a. Max dan Min dalam 1 Query
 - b. Count, 1 klausa dan 2 operator dalam 1 Query
 - c. Sum dalam 1 Query
 - d. Average pada soal 2C diatas
3. Buatlah Subquery dengan menerapkan :
 - a. Max dan Min
 - b. Count, 1 klausa dan 2 operator
 - c. Sum
 - d. Average
 - e. Group by
 - f. Select setelah where
 - g. Minimal 3 nested query

Langkah Pertama

insert data terlebih dahulu pada setiap tabel bisa menggunakan single insert atau pun multiple insert

Query

```
insert into profil_aluminium
(id_profil, nama_profil, warna)
values (1, 'strandart', 'iron grey');
insert into profil_aluminium
(id_profil, nama_profil, warna)
values (2, 'standart', 'white');
insert all
into profil_aluminium (id_profil,
nama_profil, warna)
values (3, 'standart', 'black sahara')
into profil_aluminium (id_profil,
nama_profil, warna)
values (4, 'penta 40', 'iron grey')
into profil_aluminium (id_profil,
nama_profil, warna)
values (5, 'penta 40', 'white')
select 1 from dual;
```



Tugas Praktikum

Screenshot

```
SQL> insert into profil_aluminium
  2  (id_profil, nama_profil, warna)
  3  values (1, 'strandart', 'iron grey');

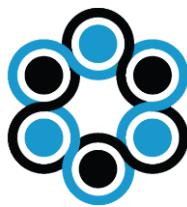
1 row created.

SQL> insert into profil_aluminium
  2  (id_profil, nama_profil, warna)
  3  values (2, 'standart', 'white');

1 row created.

SQL>
SQL> insert all
  2  into profil_aluminium (id_profil, nama_profil, warna)
  3  values (3, 'standart', 'black sahara')
  4  into profil_aluminium (id_profil, nama_profil, warna)
  5  values (4, 'penta 40', 'iron grey')
  6  into profil_aluminium (id_profil, nama_profil, warna)
  7  values (5, 'penta 40', 'white')
  8  select 1 from dual;

3 rows created.
```



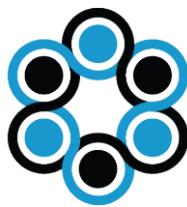
Tugas Praktikum

Query

```
insert into marketing
(id_marketing, nama_marketing, no_telp)
values (id_marketing.nextval, 'rahmad',
'123456789');
insert into marketing
(id_marketing, nama_marketing, no_telp)
values (id_marketing.nextval, 'artani',
'234567891');

CREATE FUNCTION get_seq RETURN NUMBER IS
BEGIN
    RETURN id_marketing.nextval;
END;
/

insert all
into marketing (id_marketing, nama_marketing,
no_telp)
values (get_seq, 'hendro', '345678912')
into marketing
(id_marketing, nama_marketing, no_telp)
values (get_seq, 'france', '456789123')
into marketing (id_marketing, nama_marketing,
no_telp)
values (get_seq, 'yudha', '567891234')
select 1 from dual;
```



Tugas Praktikum

Screenshot

```
SQL> insert into marketing
  2  (id_marketing, nama_marketing, no_telp)
  3  values (id_marketing.nextval, 'rahmad', '123456789');

1 row created.

SQL> insert into marketing
  2  (id_marketing, nama_marketing, no_telp)
  3  values (id_marketing.nextval, 'artani', '234567891');

1 row created.

SQL>
SQL> CREATE FUNCTION get_seq RETURN NUMBER IS
  2  BEGIN
  3    RETURN id_marketing.nextval;
  4  END;
  5  /

Function created.

SQL> insert all
  2  into marketing (id_marketing, nama_marketing, no_telp)
  3  values (get_seq, 'hendro', '345678912')
  4  into marketing
  5  (id_marketing, nama_marketing, no_telp)
  6  values (get_seq, 'france', '456789123')
  7  into marketing (id_marketing, nama_marketing, no_telp)
  8  values (get_seq, 'yudha', '567891234')
  9  select 1 from dual;

3 rows created.
```

Query

```
insert into arsitek
(id_arsitek, nama_arsitek, telp_kantor)
values (1, 'santoso', '111111111');
insert into arsitek
(id_arsitek, nama_arsitek, telp_kantor)
values (2, 'budi kurniawan', '222222222');

insert all
into arsitek
(id_arsitek, nama_arsitek, telp_kantor)
values (3, 'bambang handoko', '333333333')
into arsitek
(id_arsitek, nama_arsitek, telp_kantor)
values (4, 'margono', '444444444')
into arsitek
(id_arsitek, nama_arsitek, telp_kantor)
values (5, 'yudha prasetya', '555555555')
select 1 from dual;
```



Tugas Praktikum

Screenshot

```
SQL> insert into arsitek
  2 (id_arsitek, nama_arsitek, telp_kantor)
  3 values (1, 'santoso', '1111111111');

1 row created.

SQL> insert into arsitek
  2 (id_arsitek, nama_arsitek, telp_kantor)
  3 values (2, 'budi kurniawan', '2222222222');

1 row created.

SQL>
SQL> insert all
  2 into arsitek
  3 (id_arsitek, nama_arsitek, telp_kantor)
  4 values (3, 'bambang handoko', '3333333333')
  5 into arsitek
  6 (id_arsitek, nama_arsitek, telp_kantor)
  7 values (4, 'margono', '4444444444')
  8 into arsitek
  9 (id_arsitek, nama_arsitek, telp_kantor)
 10 values (5, 'yudha prasetya', '5555555555')
 11 select 1 from dual;

3 rows created.
```

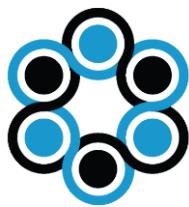


Tugas Praktikum

Query

```
insert into proyek
(nama_owner, alamat_proyek,
alamat_penyuratan, no_telp, id_arsitek,
id_marketing, id_profil)
values ('kardono', 'graha family blok-k 62',
'Dusan', '111111111', 1, 2, 2);
insert into proyek
(nama_owner, alamat_proyek,
alamat_penyuratan, no_telp, id_arsitek,
id_marketing, id_profil)
values ('sudarmono', 'graha family blok-h
18', 'graha family blok-h 18', '222222222',
5, 3, 1);

insert all
into proyek
(nama_owner, alamat_proyek,
alamat_penyuratan, no_telp, id_arsitek,
id_marketing, id_profil)
values ('yunus', 'BDG N-45', 'BK',
'333333333', 2, 3, 1)
into proyek
(nama_owner, alamat_proyek,
alamat_penyuratan, no_telp, id_arsitek,
id_marketing, id_profil)
values ('alex', 'Citraland blok-i4 20',
'embong malang', '444444444', 1, 2, 1)
into proyek
(nama_owner, alamat_proyek,
alamat_penyuratan, no_telp, id_arsitek,
id_marketing, id_profil)
values ('yos tanoyo', 'Citraland blok-h2 1',
'embong malang', '555555555', 1, 2, 2)
select 1 from dual;
```



Tugas Praktikum

Screenshot

```
SQL> insert into arsitek
  2 (id_arsitek, nama_arsitek, telp_kantor)
  3 values (1, 'santoso', '1111111111');

1 row created.

SQL> insert into arsitek
  2 (id_arsitek, nama_arsitek, telp_kantor)
  3 values (2, 'budi kurniawan', '2222222222');

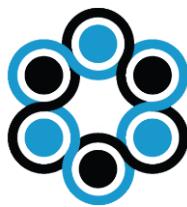
1 row created.

SQL>
SQL> insert all
  2 into arsitek
  3 (id_arsitek, nama_arsitek, telp_kantor)
  4 values (3, 'bambang handoko', '3333333333')
  5 into arsitek
  6 (id_arsitek, nama_arsitek, telp_kantor)
  7 values (4, 'margono', '4444444444')
  8 into arsitek
  9 (id_arsitek, nama_arsitek, telp_kantor)
 10 values (5, 'yudha prasetya', '5555555555')
 11 select 1 from dual;

3 rows created.
```

Analisa

Yang perlu di perhatikan dalam pengisian value pada table adalah tipe data colom apakah sudah sesuai dengan tipe data yang mau kita isikan. Kalian harus benar paham apakah colom tersebut mengandung nilai UNIX, jika iya maka nilai baris dari kolom tersebut tidak boleh sama dengan nilai baris lain dengan kolom yang sama



Tugas Praktikum

Langkah Kedua

penerapan fungsi agregasi :

Max (nama_kolom) as “nama_alias” from nama_table

Min (nama_kolom) as “nama_alias” from nama_table

Avg (nama_kolom) as “nama_alias” from nama_table

Count (nama_kolom) as “nama_alias” from nama_table

Sum (nama_kolom) as “nama_alias” from nama_table

Query

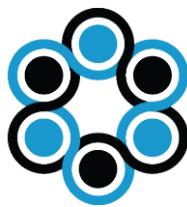
```
select nama_owner,
       max (id_marketing) as "terbanyak",
       min (id_marketing) as "terendah"
  from proyek
 group by nama_owner;
```

Screenshot

```
SQL> select nama_owner,
  2  max (id_marketing) as "terbanyak",
  3  min (id_marketing) as "terendah"
  4  from proyek
  5  group by nama_owner;

NAMA_OWNER          terbanyak    terendah
-----              -----
sudarmono                  3          3
yunus                      3          3
alex                        2          2
kardono                     2          2
yos tanoyo                  2          2

SQL>
```



Tugas Praktikum

Query

```
select warna, count(nama_profil) AS "banyak"
from profil_aluminium
where nama_profil = 'standart'
and warna != 'white'
group by warna;
```

Screenshot

```
SQL> select * from profil_aluminium;
ID_PROFIL NAMA_PROFIL          WARNA
-----  -----
1 strandart           iron grey
2 standart            white
3 standart            black sahara
4 penta 40             iron grey
5 penta 40             white

SQL> select warna, count(nama_profil) AS "banyak"
2  from profil_aluminium
3  where nama_profil = 'standart'
4  and warna != 'white'
5  group by warna;

WARNA          banyak
-----  -----
black sahara      1    . .

SQL>
```

Query

```
select id_arsitek,
sum (id_profil) as "jumlah"
from proyek
group by id_arsitek;
```

Screenshot

```
SQL> select id_arsitek,
2  sum (id_profil) as "jumlah"
3  from proyek
4  group by id_arsitek;

ID_ARSITEK      jumlah
-----  -----
1                  5
2                  1
5                  1

SQL>
```



Tugas Praktikum

Query

```
select id_arsitek,
       avg (id_profil) as "rata2"
     from proyek
   group by id_arsitek;
```

Screenshot

```
SQL> select id_arsitek,
  2  avg (id_profil) as "rata2"
  3  from proyek
  4  group by id_arsitek;

ID_ARSITEK      rata2
----- -----
      1  1,66666667
      2          1
      5          1

SQL>
```

Analisa :

Arti dalam fungsi agregasi :

Max = mencari nilai maksimal dari suatu kolom

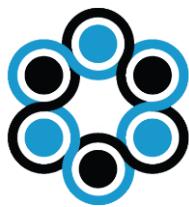
Min = mencari nilai minimal dari suatu kolom

Avg = mencari nilai rata-rata dari suatu kolom

Sum = mencari nilai total penjumlahan dari suatu kolom

Count = menghitung banyaknya kolom itu terisi.

Dalam penerapan fungsi agregasi yang perlu di perhatikan adalah apakah suatu kolom tersebut mempunyai nilai yang bisa untuk digunakan fungsi agregasi



Tugas Praktikum

Langkah Ketiga

Menggunakan perintah sub query

Query

```
select nama_owner from proyek
where id_marketing != (select
min(id_marketing) as minimal from proyek)
and id_marketing = (select max(id_marketing)
as maksimal from proyek)
group by nama_owner;
```

Screenshot

```
SQL> select nama_owner from proyek
  2 where id_marketing != (select min(id_marketing) as minimal from proyek)
  3 and id_marketing = (select max(id_marketing) as maksimal from proyek)
  4 group by nama_owner;

NAMA_OWNER
-----
sudarmono
yunus
```

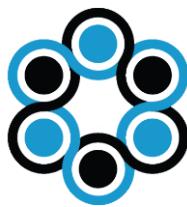
Query

```
select warna, count(nama_profil) AS "banyak"
from profil_aluminium
where warna = any (select warna from
profil_aluminium
where warna = any (select warna from
profil_aluminium
where warna !='white'))
group by warna;
```

Screenshot

```
SQL> select warna, count(nama_profil) AS "banyak"
  2 from profil_aluminium
  3 where warna = any (select warna from profil_aluminium
  4 where warna = any (select warna from profil_aluminium
  5 where warna !='white'))
  6 group by warna;

WARNA                      banyak
-----
black sahara                  1
iron grey                      2
```



Tugas Praktikum

Query

```
select id_profil,rata2 from proyek,
(select sum (rata) as rata2 from
(select avg(id_profil) as rata from proyek
group by id_profil));
```

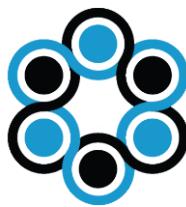
Screenshot

```
SQL> select id_profil,rata2 from proyek,
2  (select sum (rata) as rata2 from
3  (select avg(id_profil) as rata from proyek
4  group by id_profil));

ID_PROFIL      RATA2
-----  -----
          2          3
          1          3
          1          3
          1          3
          2          3
```

Analisa :

Sub query adalah suatu query yang ada di dalam query. Disini kalian bisa lihat penggunaan nama alias yang di beri tanda (“) tidak akan bisa dipanggil dalam select atau pun parameter fungsi agregasi, dikarenakan penamaan alias yang di beri tanda (“) hanya akan di tampilkan saja pada cmd. Jika kita tidak memberi tanda (“) maka nama alias tersebut bisa digunakan sebagai parameter atau bisa dibilang nama aliasnya bersifat temporari dalam satu kali sub query tersebut



Laporan Sementara

Soal Praktikum

1. Insert minimal 5 data pada setiap tabel
2. Buatlah query dengan menerapkan fungsi aggregate :
 - a. Max dan Min dalam 1 Query
 - b. Count, 1 klausa dan 2 operator dalam 1 Query
 - c. Sum dalam 1 Query
 - d. Average pada soal 2C diatas
3. Buatlah Subquery dengan menerapkan :
 - a. Max dan Min
 - b. Count, 1 klausa dan 2 operator
 - c. Sum
 - d. Average
 - e. Group by
 - f. Select setelah where
 - g. Minimal 3 nested query

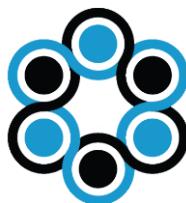
Langkah Pertama

insert data terlebih dahulu pada setiap tabel

Query

```
insert into profil_aluminium
(id_profil, nama_profil, warna)
values (1, 'strandart', 'iron grey');
insert into profil_aluminium
(id_profil, nama_profil, warna)
values (2, 'standart', 'white');

insert all
into profil_aluminium (id_profil,
nama_profil, warna)
values (3, 'standart', 'black sahara')
into profil_aluminium (id_profil,
nama_profil, warna)
values (4, 'penta 40', 'iron grey')
into profil_aluminium (id_profil,
nama_profil, warna)
values (5, 'penta 40', 'white')
select 1 from dual;
```



Laporan Sementara

Screenshot

```
SQL> insert into profil_aluminium
  2  (id_profil, nama_profil, warna)
  3  values (1, 'strandart', 'iron grey');

1 row created.

SQL> insert into profil_aluminium
  2  (id_profil, nama_profil, warna)
  3  values (2, 'standart', 'white');

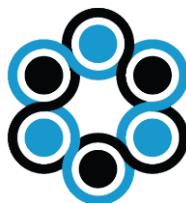
1 row created.

SQL>
SQL> insert all
  2  into profil_aluminium (id_profil, nama_profil, warna)
  3  values (3, 'standart', 'black sahara')
  4  into profil_aluminium (id_profil, nama_profil, warna)
  5  values (4, 'penta 40', 'iron grey')
  6  into profil_aluminium (id_profil, nama_profil, warna)
  7  values (5, 'penta 40', 'white')
  8  select 1 from dual;

3 rows created.
```

Query

```
insert into marketing
(id_marketing, nama_marketing, no_telp)
values (id_marketing.nextval, 'rahmad',
'123456789');
insert into marketing
(id_marketing, nama_marketing, no_telp)
values (id_marketing.nextval, 'artani',
'234567891');
CREATE FUNCTION get_seq RETURN NUMBER IS
BEGIN
    RETURN id_marketing.nextval;
END;
/
insert all
into marketing (id_marketing, nama_marketing,
no_telp)
values (get_seq, 'hendro', '345678912')
into marketing
(id_marketing, nama_marketing, no_telp)
values (get_seq, 'france', '456789123')
into marketing (id_marketing, nama_marketing,
no_telp)
values (get_seq, 'yudha', '567891234')
select 1 from dual;
```



Laporan Sementara

Screenshot

```
SQL> insert into marketing
  2 (id_marketing, nama_marketing, no_telp)
  3 values (id_marketing.nextval, 'rahmad', '123456789');

1 row created.

SQL> insert into marketing
  2 (id_marketing, nama_marketing, no_telp)
  3 values (id_marketing.nextval, 'artani', '234567891');

1 row created.

SQL>
SQL> CREATE FUNCTION get_seq RETURN NUMBER IS
  2 BEGIN
  3   RETURN id_marketing.nextval;
  4 END;
  5 /

Function created.

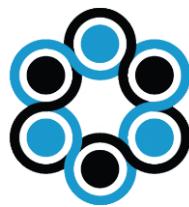
SQL> insert all
  2 into marketing (id_marketing, nama_marketing, no_telp)
  3 values (get_seq, 'hendro', '345678912')
  4 into marketing
  5 (id_marketing, nama_marketing, no_telp)
  6 values (get_seq, 'france', '456789123')
  7 into marketing (id_marketing, nama_marketing, no_telp)
  8 values (get_seq, 'yudha', '567891234')
  9 select 1 from dual;

3 rows created.
```

Query

```
insert into arsitek
(id_arsitek, nama_arsitek, telp_kantor)
values (1, 'santoso', '111111111');
insert into arsitek
(id_arsitek, nama_arsitek, telp_kantor)
values (2, 'budi kurniawan', '222222222');

insert all
into arsitek
(id_arsitek, nama_arsitek, telp_kantor)
values (3, 'bambang handoko', '333333333')
into arsitek
(id_arsitek, nama_arsitek, telp_kantor)
values (4, 'margono', '444444444')
into arsitek
(id_arsitek, nama_arsitek, telp_kantor)
values (5, 'yudha prasetya', '555555555')
select 1 from dual;
```



Laporan Sementara

Screenshot

```
SQL> insert into arsitek
  2  (id_arsitek, nama_arsitek, telp_kantor)
  3  values (1, 'santoso', '1111111111');

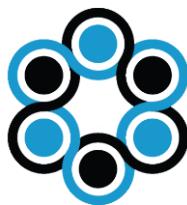
1 row created.

SQL> insert into arsitek
  2  (id_arsitek, nama_arsitek, telp_kantor)
  3  values (2, 'budi kurniawan', '2222222222');

1 row created.

SQL>
SQL> insert all
  2  into arsitek
  3  (id_arsitek, nama_arsitek, telp_kantor)
  4  values (3, 'bambang handoko', '3333333333')
  5  into arsitek
  6  (id_arsitek, nama_arsitek, telp_kantor)
  7  values (4, 'margono', '4444444444')
  8  into arsitek
  9  (id_arsitek, nama_arsitek, telp_kantor)
 10 values (5, 'yudha prasetya', '5555555555')
 11 select 1 from dual;

3 rows created.
```

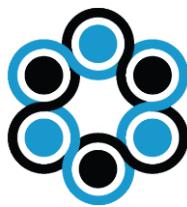


Laporan Sementara

Query

```
insert into proyek
(nama_owner, alamat_proyek,
alamat_penyuratan, no_telp, id_arsitek,
id_marketing, id_profil)
values ('kardono', 'graha family blok-k 62',
'Dusan', '1111111111', 1, 2, 2);
insert into proyek
(nama_owner, alamat_proyek,
alamat_penyuratan, no_telp, id_arsitek,
id_marketing, id_profil)
values ('sudarmono', 'graha family blok-h
18', 'graha family blok-h 18', '222222222',
5, 3, 1);

insert all
into proyek
(nama_owner, alamat_proyek,
alamat_penyuratan, no_telp, id_arsitek,
id_marketing, id_profil)
values ('yunus', 'BDG N-45', 'BK',
'333333333', 2, 3, 1)
into proyek
(nama_owner, alamat_proyek,
alamat_penyuratan, no_telp, id_arsitek,
id_marketing, id_profil)
values ('alex', 'Citraland blok-i4 20',
'embong malang', '4444444444', 1, 2, 1)
into proyek
(nama_owner, alamat_proyek,
alamat_penyuratan, no_telp, id_arsitek,
id_marketing, id_profil)
values ('yos tanoyo', 'Citraland blok-h2 1',
'embong malang', '5555555555', 1, 2, 2)
select 1 from dual;
```



Laporan Sementara

Screenshot

```
SQL> insert into arsitek
  2 (id_arsitek, nama_arsitek, telp_kantor)
  3 values (1, 'santoso', '1111111111');

1 row created.

SQL> insert into arsitek
  2 (id_arsitek, nama_arsitek, telp_kantor)
  3 values (2, 'budi kurniawan', '2222222222');

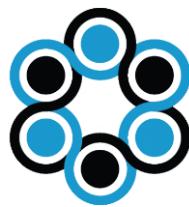
1 row created.

SQL>
SQL> insert all
  2 into arsitek
  3 (id_arsitek, nama_arsitek, telp_kantor)
  4 values (3, 'bambang handoko', '3333333333')
  5 into arsitek
  6 (id_arsitek, nama_arsitek, telp_kantor)
  7 values (4, 'margono', '4444444444')
  8 into arsitek
  9 (id_arsitek, nama_arsitek, telp_kantor)
 10 values (5, 'yudha prasetya', '5555555555')
 11 select 1 from dual;

3 rows created.
```

Analisa

(Ketik hasil analisa disini)



Laporan Sementara

Langkah Kedua

penerapan fungsi agregasi

Query

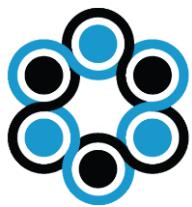
```
select nama_owner,
       max (id_marketing) as "terbanyak",
       min (id_marketing) as "terendah"
  from proyek
 group by nama_owner;
```

Screenshot

```
SQL> select nama_owner,
  2  max (id_marketing) as "terbanyak",
  3  min (id_marketing) as "terendah"
  4  from proyek
  5  group by nama_owner;

NAMA_OWNER          terbanyak    terendah
-----  -----
sudarmono                3            3
yunus                     3            3
alex                      2            2
kardono                   2            2
yos tanoyo                 2            2

SQL>
```



Laporan Sementara

Query

```
select warna, count(nama_profil) AS "banyak"  
from profil_aluminium  
where nama_profil = 'standart'  
and warna != 'white'  
group by warna;
```

Screenshot

```

SQL> select * from profil_aluminium;
ID_PROFIL NAMA_PROFIL                               WARNA
-----
1 strandart                                     iron grey
2 standart                                      white
3 standart                                     black sahara
4 penta 40                                       iron grey
5 penta 40                                       white

SQL> select warna, count(nama_profil) AS "banyak"
  2 from profil_aluminium
  3 where nama_profil = 'standart'
  4 and warna != 'white'
  5 group by warna;

WARNA                                banyak
-----
black sahara                           1      .
SQL>

```

Query

```
select id_arsitek,  
sum (id_profil) as "jumlah"  
from proyek  
group by id_arsitek;
```

Screenshot

```
SQL> select id_arsitek,
  2 sum (id_profil) as "jumlah"
  3 from proyek
  4 group by id_arsitek;

ID_ARSITEK      jumlah
-----  -----
          1            5
          2            1
          5            1

SQL>
```



Laporan Sementara

Query

```
select id_arsitek,
       avg (id_profil) as "rata2"
     from proyek
   group by id_arsitek;
```

Screenshot

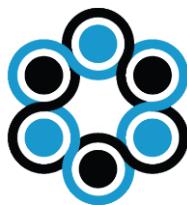
```
SQL> select id_arsitek,
  2  avg (id_profil) as "rata2"
  3  from proyek
  4  group by id_arsitek;

ID_ARSITEK      rata2
----- -----
      1 1,66666667
      2          1
      5          1

SQL>
```

Analisa :

(Ketik hasil analisa disini)



Laporan Sementara

Langkah Ketiga

menggunakan perintah sub query

Query

```
select id_marketing, nama_marketing, no_telp
from marketing where id_marketing =
(select max(id_marketing)
from marketing);

select id_marketing, nama_marketing, no_telp
from marketing where id_marketing =
(select min(id_marketing)
from marketing);
```

Screenshot

```
SQL> select id_marketing, nama_marketing, no_telp
  2  from marketing where id_marketing =
  3  (select max(id_marketing)
  4  from marketing);

ID_MARKETING NAMA_MARKETING          NO_TELP
----- -----
      6 yudha                      567891234

SQL>

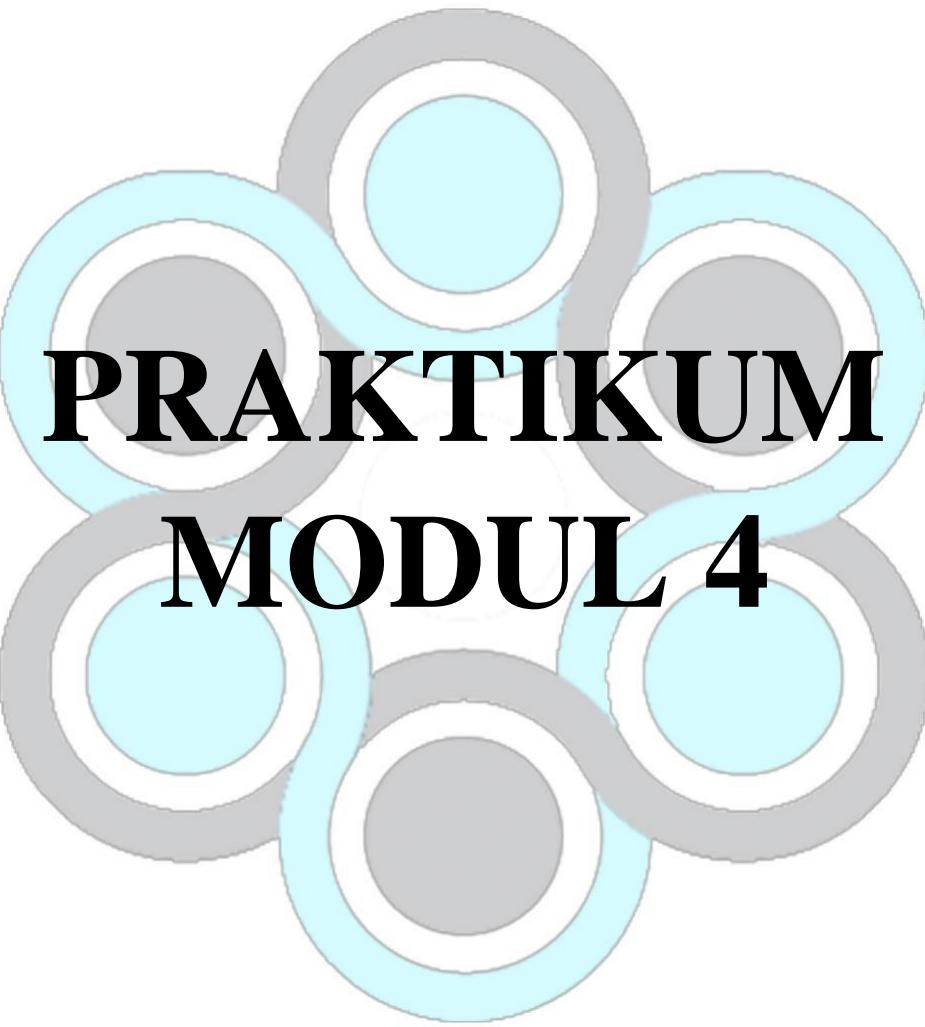
SQL> select id_marketing, nama_marketing, no_telp
  2  from marketing where id_marketing =
  3  (select min(id_marketing)
  4  from marketing);

ID_MARKETING NAMA_MARKETING          NO_TELP
----- -----
      2 rahmad                     123456

SQL>
```

Analisa :

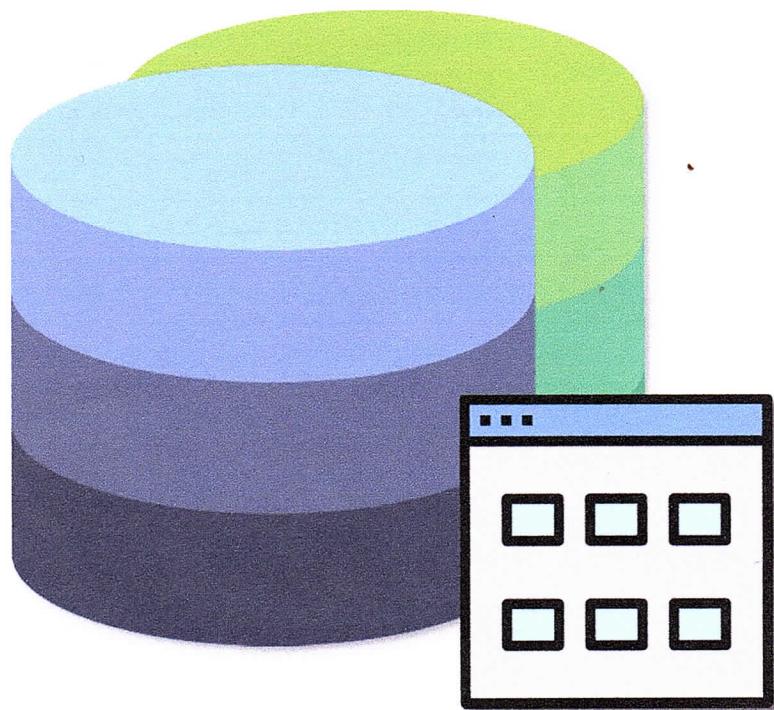
(Ketik hasil analisa disini)



PRAKTIKUM MODUL 4

FAKULTAS TEKNIK ELEKTRO DAN
TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI ADHI TAMA
SURABAYA
2020

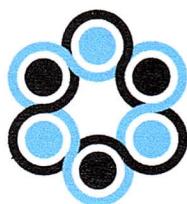
LAPORAN PENDAHULUAN



PRAKTIKUM BASIS DATA PERIODE V - Tahun 2019/2020

NAMA : Mochamad Khornul Muza
NPM : 06.2018.1.06gg0
MODUL : IV

--



Tugas Pertanyaan

1. Perhatikan tabel berikut :

Nama-Produk	Nama-Kategori	Harga-Satuan	Stok
Produk			
Indomie	Makanan	2500	50
Manmas	Minuman	2000	40
Jeruk	Buah-Sayur	8000	100
Samsung	Electronik	81.000.000	35
Supreme	Pakaian	35.000	95
Roti Kukus	Makanan	2500	10
Kue Belok	Makanan	8500	90
Cranberries	Makanan	2500	70
Semangka	Minuman	10.500	40
Kue Ku	Makanan	2500	50

Buatlah sebuah query untuk menampilkan tabel seperti diatas !

- Cek terlebih dahulu tabel yang memiliki Primary key dan Foreign key yang saling terhubung
- a. Tabel Produk

`SELECT DESC Produk_O6gg0`

Name

ID - PRODUK

ID - PEMASOK

ID - KATEGORI

NAMA - PRODUK

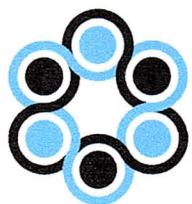
HARGA - SATUAN

STOK - PRODUK

- b. Tabel Kategori

`SELECT DESC Kategori - O6gg0`

Name



Tugas Pertanyaan

1D - KATEGORI

NAMA - KATEGORI

DESKRIPSI

- Buat query untuk melakukan join tabel

SQL> SELECT a. nama - Produk , b. nama - kategori ,
a. harga - satuan , a. stok - Produk

2. FROM Produk - 06gg0 a JOIN

3. kategori - 06gg0 b .

4. ON a. Id - kategori = b. Id - kategori

5. WHERE rownum <= 10;

2. Perhatikan gambar berikut :

ID - PEMASOK NAMA - PERUSAHAAN ALAMAT

NAMA - PRODUK

1. Distri Indomaret Lombok Indomre

2. Distri Alfamart Madura Marimas

3. Distri Giant Sidoarjo Jeruk

4. Distri Transmart Jupiter Samsung

5. Distri Superindo Pluto Supreme

6. Distri Indomart Lombok Roti Kukus

7. Distri Indomart Lombok Kue Balok

8. Distri Indomart Lombok Crem berries

9. Distri Alfamart Madura Semangka

10. Distri Superindo Pluto Kue Ku

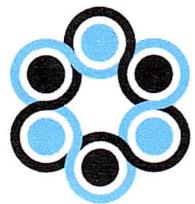
Buatlah sebuah view untuk menampilkan tabel

seperti di atas !

- Query untuk View

SQL> CREATE VIEW list - Pemasok

2. AS SELECT a. Id - Pemasok , a. nama - Pemasok



Tugas Pertanyaan

a. nama - Perusahaan , alamat , b. nama - Produk

3 FROM Pemasok - 06gg0 a.

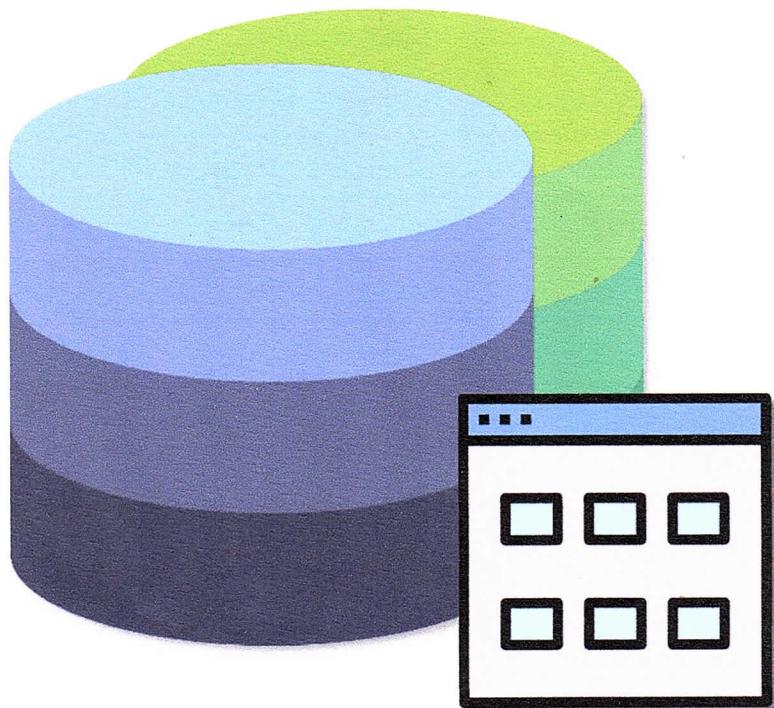
4. JOIN Produk - 06gg0

5 ON a. Id - Pemasok = b. Id - Pemasok

6. JOIN kategori - 06gg0 c.

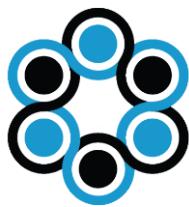
7. ON b. Id - kategori = C. Id - kategori ;

LAPORAN PRAKTIKUM



PRAKTIKUM BASIS DATA PERIODE V - Tahun 2019/2020

NAMA : Mochamad Khairul Huda
NPM : 06.2010.1.06990
MODUL : W



Tugas Praktikum

Soal Praktikum

1. Insert 5 data pada setiap tabel
2. Buatlah query laporan dengan jelas sesuai proyek Anda dengan ketentuan sebagai berikut :
 - a. Equi Join minimal 2 tabel dengan menerapkan Operator Pembanding dan Klausula
 - b. Left Join minimal 3 tabel dengan menerapkan Klausula dan Subquery
 - c. Right Join minimal 3 tabel dengan menerapkan Aggregate dan Operator Logika
- Note : Tampilkan kolom pada tabel pertama dan kolom pada tabel kedua dengan memperhatikan kasus yang akan Anda lapor.
3. Buatlah view dengan menerapkan DML.
4. Buatlah view dari masing-masing soal pada nomor 2.
5. Terapkan DML pada View pada nomor 4

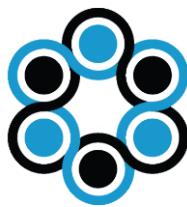
Langkah Pertama

Insert 5 data pada setiap tabel yang sudah anda buat.

Query

```
insert into profil_aluminium
(id_profil, nama_profil, warna)
values (1, 'strandart', 'iron grey');
insert into profil_aluminium
(id_profil, nama_profil, warna)
values (2, 'standart', 'white');

insert all
into profil_aluminium (id_profil,
nama_profil, warna)
values (3, 'standart', 'black sahara')
into profil_aluminium (id_profil,
nama_profil, warna)
values (4, 'penta 40', 'iron grey')
into profil_aluminium (id_profil,
nama_profil, warna)
values (5, 'penta 40', 'white')
select 1 from dual;
```



Tugas Praktikum

Screenshot

```
SQL> insert into profil_aluminium (id_profil, nama_profil, warna) values (1, 'strandart', 'iron grey');
1 row created.

SQL> insert into profil_aluminium (id_profil, nama_profil, warna) values (2, 'standart', 'white');
1 row created.
```

```
SQL> insert all
  2  into profil_aluminium (id_profil, nama_profil, warna) values (3, 'standart', 'black sahara')
  3  into profil_aluminium (id_profil, nama_profil, warna) values (4, 'penta 40', 'iron grey')
  4  into profil_aluminium (id_profil, nama_profil, warna) values (5, 'penta 40', 'white')
  5  select 1 from dual;

3 rows created.
```

Query

```
insert into marketing (id_marketing, nama_marketing,
no_telp) values (id_marketing.nextval, 'rahmad',
'123456789');
insert into marketing (id_marketing, nama_marketing,
no_telp) values (id_marketing.nextval, 'artani',
'234567891');

insert into marketing (id_marketing, nama_marketing,
no_telp) values (id_marketing.nextval, 'hendro',
'345678912');
insert into marketing (id_marketing, nama_marketing,
no_telp) values (id_marketing.nextval, 'france',
'456789123');
insert into marketing (id_marketing, nama_marketing,
no_telp) values (id_marketing.nextval, 'yudha',
'567891234');
```

Screenshot

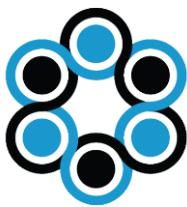
```
SQL> insert into marketing (id_marketing, nama_marketing, no_telp) values (id_marketing.nextval, 'rahmad', '123456789');
1 row created.

SQL> insert into marketing (id_marketing, nama_marketing, no_telp) values (id_marketing.nextval, 'artani', '234567891');
1 row created.
```

```
SQL> insert into marketing (id_marketing, nama_marketing, no_telp) values (id_marketing.nextval, 'hendro', '345678912');
1 row created.

SQL> insert into marketing (id_marketing, nama_marketing, no_telp) values (id_marketing.nextval, 'france', '456789123');
1 row created.

SQL> insert into marketing (id_marketing, nama_marketing, no_telp) values (id_marketing.nextval, 'yudha', '567891234');
1 row created.
```



Tugas Praktikum

Query

```
insert into arsitek (id_arsitek, nama_arsitek,
telp_kantor) values (1, 'santoso', '1111111111');
insert into arsitek (id_arsitek, nama_arsitek,
telp_kantor) values (2, 'budi kurniawan',
'2222222222');

insert all
into arsitek (id_arsitek, nama_arsitek, telp_kantor)
values (3, 'bambang handoko', '3333333333')
into arsitek (id_arsitek, nama_arsitek, telp_kantor)
values (4, 'margono', '4444444444')
into arsitek (id_arsitek, nama_arsitek, telp_kantor)
values (5, 'yudha prasetya', '5555555555')
select 1 from dual;
```

Screenshot

```
SQL> insert into arsitek (id_arsitek, nama_arsitek, telp_kantor) values (1, 'santoso', '1111111111');
1 row created.

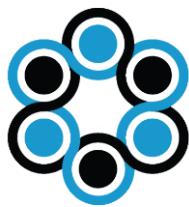
SQL> insert into arsitek (id_arsitek, nama_arsitek, telp_kantor) values (2, 'budi kurniawan', '2222222222');
1 row created.

SQL>
SQL> insert all
  2 into arsitek (id_arsitek, nama_arsitek, telp_kantor) values (3, 'bambang handoko', '3333333333')
  3 into arsitek (id_arsitek, nama_arsitek, telp_kantor) values (4, 'margono', '4444444444')
  4 into arsitek (id_arsitek, nama_arsitek, telp_kantor) values (5, 'yudha prasetya', '5555555555')
  5 select 1 from dual;

3 rows created.
```

Query

```
insert into proyek (alamat_proyek,
nama_owner, alamat_penyuratan, no_telp,
id_arsitek, id_marketing, id_profil)
values ('graha family blok-k 62', 'kardono',
'Dusan', '1111111111', 1, 3, 2);
insert into proyek (alamat_proyek,
nama_owner, alamat_penyuratan, no_telp,
id_arsitek, id_marketing, id_profil)
values ('graha family blok-h 18',
'sudarmono', 'graha family blok-h 18',
'2222222222', 5, 2, 1);
insert into proyek (alamat_proyek,
nama_owner, alamat_penyuratan, no_telp,
```



Tugas Praktikum

Screenshot

```
SQL> insert into proyek (alamat_proyek, nama_owner, alamat_penyuratan, no_telp, id_arsitek, id_marketing, id_profil)
  2  values ('graha family blok-k 62', 'kardono', 'Dusan', '111111111', 1, 3, 2);
1 row created.

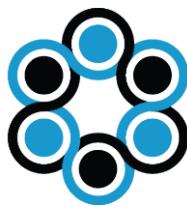
SQL> insert into proyek (alamat_proyek, nama_owner, alamat_penyuratan, no_telp, id_arsitek, id_marketing, id_profil)
  2  values ('graha family blok-h 18', 'sudarmono', 'graha family blok-h 18', '222222222', 5, 2, 1);
1 row created.

SQL> insert into proyek (alamat_proyek, nama_owner, alamat_penyuratan, no_telp, id_arsitek, id_marketing, id_profil) values ('BDG N-45', 'yunus', 'BK', '333333333', 2, 1, 1);
1 row created.

SQL> insert into proyek (alamat_proyek, nama_owner, alamat_penyuratan, no_telp, id_arsitek, id_marketing, id_profil) values ('Citraland blok-i4 20', 'alex', 'embong malang', '444444444', 1, 3, 1);
1 row created.
```

Analisa

Disini saya menggunakan 2 tipe insert table. Yang pertama adalah insert table secara partial dan yang kedua saya memasukan data secara bersamaan atau biasa kita sebut dengan multiple insert



Tugas Praktikum

Langkah Kedua

Kita terpakan beberapa join condition dalam query kita diantaranya Inner join, Left join, dan Right join

Query

```
select m.nama_marketing, py.nama_owner
from proyek py join
marketing m
on py.id_marketing = m.id_marketing
where py.id_marketing = 2;
```

Screenshot

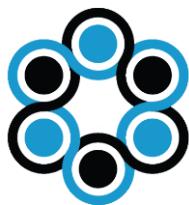
```
SQL> select m.nama_marketing, py.nama_owner
  2  from proyek py join
  3  marketing m
  4  on py.id_marketing = m.id_marketing
  5  where py.id_marketing = 2;

NAMA_MARKETING          NAMA_OWNER
-----          -----
rahmad                  kardono
rahmad                  alex
rahmad                  yos tanoyo

SQL>
```

Query

```
select m.nama_marketing, py.nama_owner,
a.nama_arsitek
from proyek py left join
marketing m
on py.id_marketing = m.id_marketing
left join arsitek a
on py.id_arsitek = a.id_arsitek
where py.id_marketing =
(select id_marketing from marketing
where nama_marketing = 'rahmad')
;
```



Tugas Praktikum

Screenshot

```
SQL> select m.nama_marketing, py.nama_owner, a.nama_arsitek
  2  from proyek py left join
  3  marketing m
  4  on py.id_marketing = m.id_marketing
  5  left join arsitek a
  6  on py.id_arsitek = a.id_arsitek
  7  where py.id_marketing =
  8  (select id_marketing from marketing
  9  where nama_marketing = 'rahmad')
 10 ;

      NAMA_MARKETING      NAMA_OWNER      NAMA_ARSITEK
-----  -----  -----
rahmad          kardono        santoso
rahmad          alex           santoso
rahmad          yos tanoyo     santoso

SQL>
```

Query

```
select m.nama_marketing, py.nama_owner,
a.nama_arsitek
from proyek py left join
marketing m
on py.id_marketing = m.id_marketing
left join arsitek a
on py.id_arsitek = a.id_arsitek
where py.id_marketing =
(select id_marketing from marketing
where nama_marketing = 'rahmad')
;
```

Screenshot

```
SQL> select m.nama_marketing, py.nama_owner, a.nama_arsitek
  2  from proyek py left join
  3  marketing m
  4  on py.id_marketing = m.id_marketing
  5  left join arsitek a
  6  on py.id_arsitek = a.id_arsitek
  7  where py.id_marketing =
  8  (select id_marketing from marketing
  9  where nama_marketing = 'rahmad')
 10 ;

      NAMA_MARKETING      NAMA_OWNER      NAMA_ARSITEK
-----  -----  -----
rahmad          kardono        santoso
rahmad          alex           santoso
rahmad          yos tanoyo     santoso

SQL>
```



Tugas Praktikum

Query

```
select m.nama_marketing, pr.nama_profil,
count(m.nama_marketing) as jumlah
from proyek py
right join profil_aluminium pr
on pr.id_profil = py.id_profil
right join marketing m
on m.id_marketing = py.id_marketing
where pr.nama_profil = 'standart'
and pr.warna = 'iron grey'
group by m.nama_marketing, pr.nama_profil
;
```

Screenshot

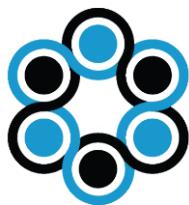
```
SQL> select m.nama_marketing, pr.nama_profil,
2 count(m.nama_marketing) as jumlah
3 from proyek py
4 right join profil_aluminium pr
5 on pr.id_profil = py.id_profil
6 right join marketing m
7 on m.id_marketing = py.id_marketing
8 where pr.nama_profil = 'standart'
9 and pr.warna = 'iron grey'
10 group by m.nama_marketing, pr.nama_profil
11 ;

NAMA_MARKETING      NAMA_PROFIL      JUMLAH
-----  -----
rahmad            standart          1
artani            standart          2

SQL>
```

Analisa :

Di sini saya menggunakan Inner join pada query yang pertama. Query yang ke dua saya menerapkan Left join. Right join ada pada query yang ke tiga.



Tugas Praktikum

Langkah Ketiga

Pada langkah ketiga ini kita akan menampilkan isi dalam sebuah tabel yang menggunakan perintah DML.

Query

```
select * from arsitek;
select * from marketing;
select * from profil_aluminium;
```

Screenshot

```
SQL> select * from arsitek;
ID_ARSITEK NAMA_ARSITEK          TELP_KANTOR
----- -----
1 santoso           1111111111
2 budi kurniawan   2222222222
3 bambang handoko 3333333333
4 margono          4444444444
5 yudha prasetya  5555555555

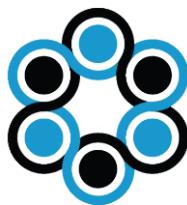
SQL> select * from marketing;
ID_MARKETING NAMA_MARKETING      NO_TELP
----- -----
2 rahmad           123456789
3 artani           234567891
4 hendro          345678912
5 france          456789123
6 yudha           567891234

SQL> select * from profil_aluminium;
ID_PROFIL NAMA_PROFIL          WARNA
----- -----
1 standart         iron grey
2 standart         white
3 standart         black sahara
4 penta 40         iron grey
5 penta 40         white

SQL>
```

Analisa :

Pada langkah ketiga ini saya menampilkan 3 table yang berhubungan dengan table proyek atau setiap id dalam 3 tabel tersebut adalah foreign key untuk beberapa kolom dari tabel proyek



Tugas Praktikum

Langkah Keempat

Langkah keempat kita akan membuat sebuah view atau bisa disebut dengan virtual table dari masing-masing query yang ada pada langkah ketiga

Query

```
create view kepemilikan_proyek
as select m.nama_marketing, py.nama_owner
from proyek py join
marketing m
on py.id_marketing = m.id_marketing
where py.id_marketing = 2;
```

Screenshot

```
SQL> create view kepemilikan_proyek
  2  as select m.nama_marketing, py.nama_owner
  3  from proyek py join
  4  marketing m
  5  on py.id_marketing = m.id_marketing
  6  where py.id_marketing = 2;

View created.
```

Query

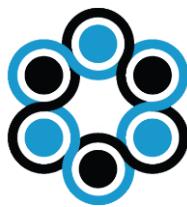
```
create view kepemilikan_proyek_arsitek
as select m.nama_marketing, py.nama_owner,
a.nama_arsitek
from proyek py left join
marketing m
on py.id_marketing = m.id_marketing
left join arsitek a
on py.id_arsitek = a.id_arsitek
where py.id_marketing =
(select id_marketing from marketing
where nama_marketing = 'rahmad');
```

Screenshot

```
SQL> create view kepemilikan_proyek_arsitek
  2  as select m.nama_marketing, py.nama_owner, a.nama_arsitek
  3  from proyek py left join
  4  marketing m
  5  on py.id_marketing = m.id_marketing
  6  left join arsitek a
  7  on py.id_arsitek = a.id_arsitek
  8  where py.id_marketing =
  9  (select id_marketing from marketing
10  where nama_marketing = 'rahmad')
11 ;

View created.

SQL >
```



Tugas Praktikum

Query

```
create view jumlah_profil_digunakan
as select m.nama_marketing, pr.nama_profil,
count(m.nama_marketing) as jumlah
from proyek py
right join profil_aluminium pr
on pr.id_profil = py.id_profil
right join marketing m
on m.id_marketing = py.id_marketing
where pr.nama_profil = 'standart'
and pr.warna = 'iron grey'
group by m.nama_marketing, pr.nama_profil
;
```

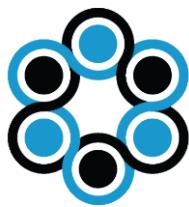
Screenshot

```
SQL> create view jumlah_profil_digunakan
  2  as select m.nama_marketing, pr.nama_profil,
  3  count(m.nama_marketing) as jumlah
  4  from proyek py
  5  right join profil_aluminium pr
  6  on pr.id_profil = py.id_profil
  7  right join marketing m
  8  on m.id_marketing = py.id_marketing
  9  where pr.nama_profil = 'standart'
 10 and pr.warna = 'iron grey'
 11 group by m.nama_marketing, pr.nama_profil
 12 ;
View created.

SQL>
```

Analisa :

Pada langkah keempat ini kita membuat view



Tugas Praktikum

Langkah Kelima

Menampilkan isi dari langkah keempat

Query

```
select * from kepemilikan_proyek;
select * from kepemilikan_proyek_arsitek;
select * from jumlah_profil_digunakan;
```

Screenshot

```
SQL> select * from kepemilikan_proyek;
NAMA_MARKETING      NAMA_OWNER
-----
rahmad              kardono
rahmad              alex
rahmad              yos tanoyo

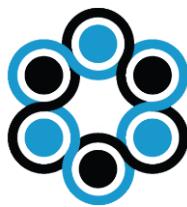
SQL> select * from kepemilikan_proyek_arsitek;
NAMA_MARKETING      NAMA_OWNER      NAMA_ARSITEK
-----
rahmad              kardono        santoso
rahmad              alex          santoso
rahmad              yos tanoyo    santoso

SQL> select * from jumlah_profil_digunakan;
NAMA_MARKETING      NAMA_PROFIL      JUMLAH
-----
rahmad              standart        1
artani              standart        2

SQL>
```

Analisa :

Disini kita akan menampilkan isi dari table virtual yang sudah kita buat pada langkah keempat



Laporan Sementara

Soal Praktikum

1. Insert 5 data pada setiap tabel
2. Buatlah query laporan dengan jelas sesuai proyek Anda dengan ketentuan sebagai berikut :
 - a. Equi Join minimal 2 tabel dengan menerapkan Operator Pembanding dan Klausua
 - b. Left Join minimal 3 tabel dengan menerapkan Klausua dan Subquery
 - c. Right Join minimal 3 tabel dengan menerapkan Aggregate dan Operator Logika

Note : Tampilkan kolom pada tabel pertama dan kolom pada tabel kedua dengan memperhatikan kasus yang akan Anda lapor.

3. Buatlah view dengan menerapkan DML.
4. Buatlah view dari masing-masing soal pada nomor 2.
5. Terapkan DML pada View pada nomor 4

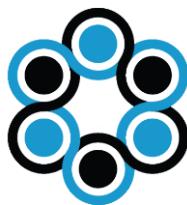
Langkah Pertama

Insert 5 data pada setiap tabel yang sudah anda buat.

Query

```
insert into profil_aluminium
(id_profil, nama_profil, warna)
values (1, 'strandart', 'iron grey');
insert into profil_aluminium
(id_profil, nama_profil, warna)
values (2, 'standart', 'white');

insert all
into profil_aluminium (id_profil,
nama_profil, warna)
values (3, 'standart', 'black sahara')
into profil_aluminium (id_profil,
nama_profil, warna)
values (4, 'penta 40', 'iron grey')
into profil_aluminium (id_profil,
nama_profil, warna)
values (5, 'penta 40', 'white')
select 1 from dual;
```



Laporan Sementara

Screenshot

```
SQL> insert into profil_aluminium (id_profil, nama_profil, warna) values (1, 'strandart', 'iron grey');
1 row created.

SQL> insert into profil_aluminium (id_profil, nama_profil, warna) values (2, 'standart', 'white');
1 row created.
```

```
SQL> insert all
  2  into profil_aluminium (id_profil, nama_profil, warna) values (3, 'standart', 'black sahara')
  3  into profil_aluminium (id_profil, nama_profil, warna) values (4, 'penta 40', 'iron grey')
  4  into profil_aluminium (id_profil, nama_profil, warna) values (5, 'penta 40', 'white')
  5  select 1 from dual;

3 rows created.
```

Query

```
insert into marketing (id_marketing, nama_marketing,
no_telp) values (id_marketing.nextval, 'rahmad',
'123456789');
insert into marketing (id_marketing, nama_marketing,
no_telp) values (id_marketing.nextval, 'artani',
'234567891');

insert into marketing (id_marketing, nama_marketing,
no_telp) values (id_marketing.nextval, 'hendro',
'345678912');
insert into marketing (id_marketing, nama_marketing,
no_telp) values (id_marketing.nextval, 'france',
'456789123');
insert into marketing (id_marketing, nama_marketing,
no_telp) values (id_marketing.nextval, 'yudha',
'567891234');
```

Screenshot

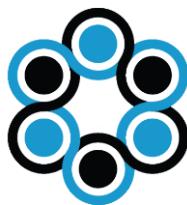
```
SQL> insert into marketing (id_marketing, nama_marketing, no_telp) values (id_marketing.nextval, 'rahmad', '123456789');
1 row created.

SQL> insert into marketing (id_marketing, nama_marketing, no_telp) values (id_marketing.nextval, 'artani', '234567891');
1 row created.

SQL> insert into marketing (id_marketing, nama_marketing, no_telp) values (id_marketing.nextval, 'hendro', '345678912');
1 row created.

SQL> insert into marketing (id_marketing, nama_marketing, no_telp) values (id_marketing.nextval, 'france', '456789123');
1 row created.

SQL> insert into marketing (id_marketing, nama_marketing, no_telp) values (id_marketing.nextval, 'yudha', '567891234');
1 row created.
```



Laporan Sementara

Query

```
insert into arsitek (id_arsitek, nama_arsitek,
telp_kantor) values (1, 'santoso', '1111111111');
insert into arsitek (id_arsitek, nama_arsitek,
telp_kantor) values (2, 'budi kurniawan',
'2222222222');

insert all
into arsitek (id_arsitek, nama_arsitek, telp_kantor)
values (3, 'bambang handoko', '3333333333')
into arsitek (id_arsitek, nama_arsitek, telp_kantor)
values (4, 'margono', '4444444444')
into arsitek (id_arsitek, nama_arsitek, telp_kantor)
values (5, 'yudha prasetya', '5555555555')
select 1 from dual;
```

Screenshot

```
SQL> insert into arsitek (id_arsitek, nama_arsitek, telp_kantor) values (1, 'santoso', '1111111111');
1 row created.

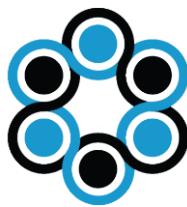
SQL> insert into arsitek (id_arsitek, nama_arsitek, telp_kantor) values (2, 'budi kurniawan', '2222222222');
1 row created.

SQL>
SQL> insert all
  2 into arsitek (id_arsitek, nama_arsitek, telp_kantor) values (3, 'bambang handoko', '3333333333')
  3 into arsitek (id_arsitek, nama_arsitek, telp_kantor) values (4, 'margono', '4444444444')
  4 into arsitek (id_arsitek, nama_arsitek, telp_kantor) values (5, 'yudha prasetya', '5555555555')
  5 select 1 from dual;

3 rows created.
```

Query

```
insert into proyek (alamat_proyek,
nama_owner, alamat_penyuratan, no_telp,
id_arsitek, id_marketing, id_profil)
values ('graha family blok-k 62', 'kardono',
'Dusan', '1111111111', 1, 3, 2);
insert into proyek (alamat_proyek,
nama_owner, alamat_penyuratan, no_telp,
id_arsitek, id_marketing, id_profil)
values ('graha family blok-h 18',
'sudarmono', 'graha family blok-h 18',
'2222222222', 5, 2, 1);
insert into proyek (alamat_proyek,
nama_owner, alamat_penyuratan, no_telp,
```



Laporan Sementara

Screenshot

```
SQL> insert into proyek (alamat_proyek, nama_owner, alamat_penyuratan, no_telp, id_arsitek, id_marketing, id_profil)
  2  values ('graha family blok-k 62', 'kardono', 'Dusan', '111111111', 1, 3, 2);
1 row created.

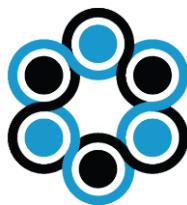
SQL> insert into proyek (alamat_proyek, nama_owner, alamat_penyuratan, no_telp, id_arsitek, id_marketing, id_profil)
  2  values ('graha family blok-h 18', 'sudarmono', 'graha family blok-h 18', '222222222', 5, 2, 1);
1 row created.

SQL> insert into proyek (alamat_proyek, nama_owner, alamat_penyuratan, no_telp, id_arsitek, id_marketing, id_profil) values ('BDG N-45', 'yunus', 'BK', '333333333', 2, 1, 1);
1 row created.

SQL> insert into proyek (alamat_proyek, nama_owner, alamat_penyuratan, no_telp, id_arsitek, id_marketing, id_profil) values ('Citraland blok-i4 20', 'alex', 'embong malang', '444444444', 1, 3, 1);
1 row created.
```

Analisa

(Ketik hasil analisa disini)



Laporan Sementara

Langkah Kedua

(Ketik penjelasan langkah – langkah disini)

Query

```
select m.nama_marketing, py.nama_owner
from proyek py join
marketing m
on py.id_marketing = m.id_marketing
where py.id_marketing = 2;
```

Screenshot

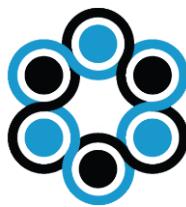
```
SQL> select m.nama_marketing, py.nama_owner
  2  from proyek py join
  3  marketing m
  4  on py.id_marketing = m.id_marketing
  5  where py.id_marketing = 2;

NAMA_MARKETING          NAMA_OWNER
-----  -----
rahmad                  kardono
rahmad                  alex
rahmad                  yos tanoyo

SQL>
```

Query

```
select m.nama_marketing, py.nama_owner,
a.nama_arsitek
from proyek py left join
marketing m
on py.id_marketing = m.id_marketing
left join arsitek a
on py.id_arsitek = a.id_arsitek
where py.id_marketing =
(select id_marketing from marketing
where nama_marketing = 'rahmad')
;
```



Laporan Sementara

Screenshot

```
SQL> select m.nama_marketing, py.nama_owner, a.nama_arsitek
  2  from proyek py left join
  3  marketing m
  4  on py.id_marketing = m.id_marketing
  5  left join arsitek a
  6  on py.id_arsitek = a.id_arsitek
  7  where py.id_marketing =
  8  (select id_marketing from marketing
  9  where nama_marketing = 'rahmad')
 10 ;
```

NAMA_MARKETING	NAMA_OWNER	NAMA_ARSITEK
rahmad	kardono	santoso
rahmad	alex	santoso
rahmad	yos tanoyo	santoso

```
SQL>
```

Query

```
select m.nama_marketing, py.nama_owner,
a.nama_arsitek
from proyek py left join
marketing m
on py.id_marketing = m.id_marketing
left join arsitek a
on py.id_arsitek = a.id_arsitek
where py.id_marketing =
(select id_marketing from marketing
where nama_marketing = 'rahmad')
;
```

Screenshot

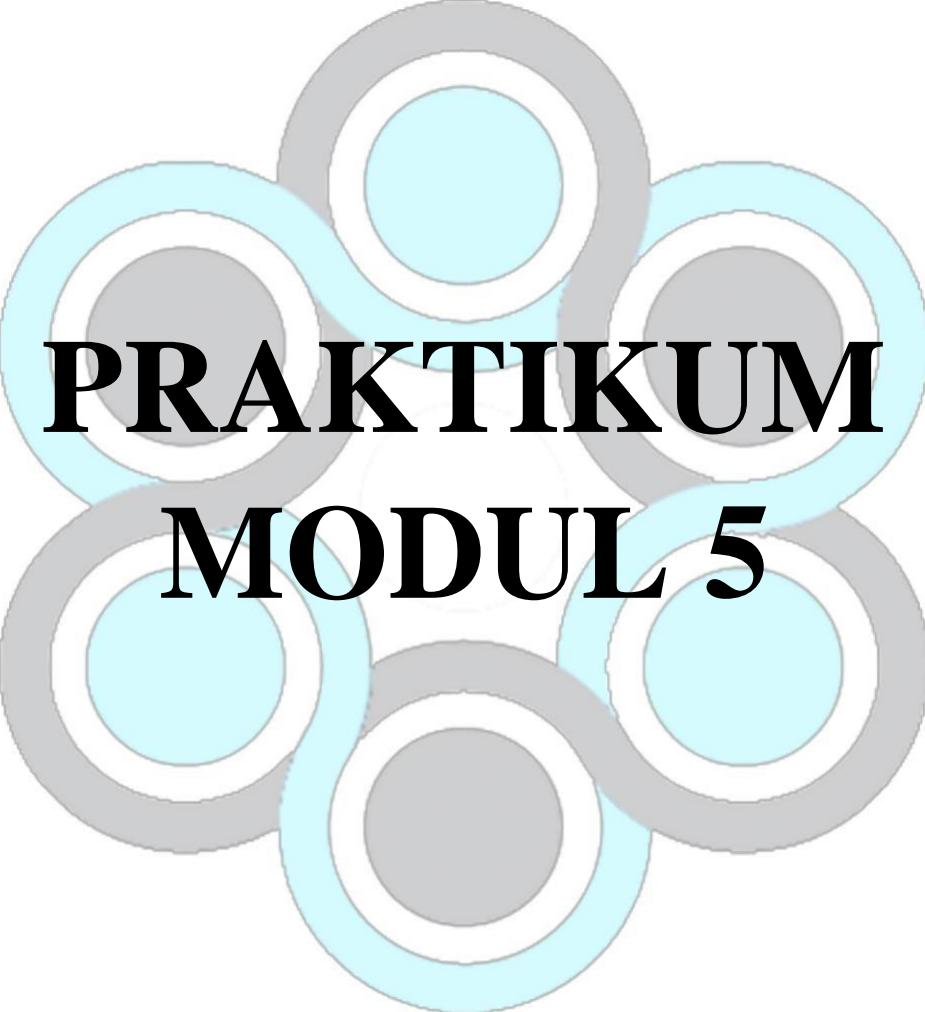
```
SQL> select m.nama_marketing, py.nama_owner, a.nama_arsitek
  2  from proyek py left join
  3  marketing m
  4  on py.id_marketing = m.id_marketing
  5  left join arsitek a
  6  on py.id_arsitek = a.id_arsitek
  7  where py.id_marketing =
  8  (select id_marketing from marketing
  9  where nama_marketing = 'rahmad')
 10 ;
```

NAMA_MARKETING	NAMA_OWNER	NAMA_ARSITEK
rahmad	kardono	santoso
rahmad	alex	santoso
rahmad	yos tanoyo	santoso

```
SQL>
```

Analisa :

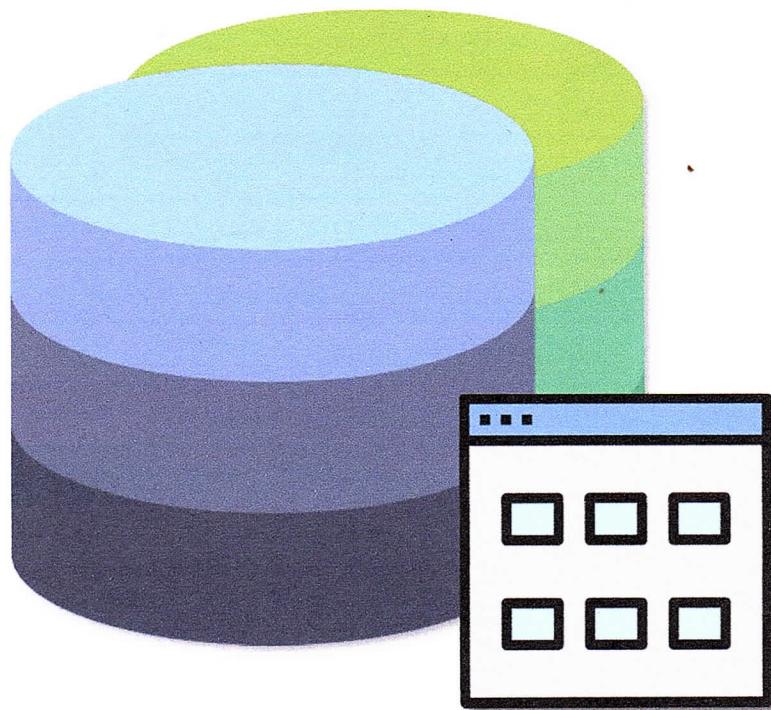
(Ketik hasil analisa disini)



PRAKTIKUM MODUL 5

FAKULTAS TEKNIK ELEKTRO DAN
TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI ADHI TAMA
SURABAYA
2020

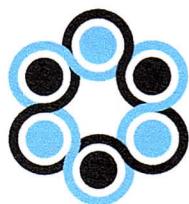
LAPORAN PENDAHULUAN



PRAKTIKUM BASIS DATA PERIODE V - Tahun 2019/2020

NAMA : Mochamava Khamru Huza
NPM : 06.2018-1.06990
MODUL : V

--



Tugas Pertanyaan

S.3. Soal Pendahuluan

Pada bab ini kita akan mempelajari tentang cara mengkoneksikan program yang akan kita bangun dengan database. Pada kasus ini kita akan membuat program dengan bahasa Pemrograman Java dan Database. Yang digunakan adalah Oracle Database. Kasus yang akan kita gunakan adalah membuat aplikasi management transaksi dengan design database yang sudah disediakan di modul.

Sebelum membuat aplikasi kita harus mempersiapkan library / package yang kita gunakan untuk manajemen koneksi Database Oracle yakni OJDBC.

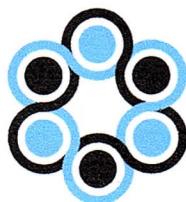
Setelah kita menyiapkan library tersebut, kita dapat langsung membuat proyek baru di IDE Netbeans dan buat folder dengan nama "lib" yang berisi file library yang kita siapkan sebelumnya.

Untuk persiapan database pastikan sudah menelepasikan langkah-langkah pada modul sebelumnya dan pastikan sudah ada Sequence ID-PEMESANAN. Pada kasus ini kita akan melakukan hal yang sama sesuai modul tapi dengan user baru yakni oracle java dengan password oracle java 123

Struktur folder akan menjadi seperti berikut:

```
1 -- lib  
  | -- nb Project  
  | -- SRC  
  | -- build.xml  
  | -- manifest.mf
```

Selanjutnya, siapkan buat Package dan class sesuai struktur berikut:

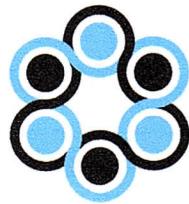


Tugas Pertanyaan

| -- Controller
| ----- Transaksi.java
| -- database
| ---- Koneksi.java
| -- model
| ----- Detail_Pemesanan.java
| ----- Kategori.java
| ----- Kurir.java
| ----- Pegawai.java
| ----- Pelanggan.java
| ----- Pemasok.java
| ----- Pemesanan.java
| ----- Produk.java
| -- oracle.java
| ----- Oracle.java.java
| -- View
| ----- TransaksiView.java

Untuk transaksiView.java classnya adalah bentuk
JFrame form dengan tampilan seperti berikut:

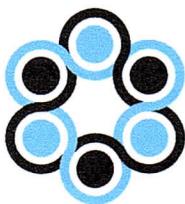
Pelanggan :	<input type="text"/>
Pegawai :	<input type="text"/>
Kurir :	<input type="text"/>
Alamat :	<input type="text"/>
tgj Pengiriman:	
Produk :	<input type="text"/>
Jumlah :	<input type="text"/>
Diskon :	<input type="text"/>
<input type="button" value="Tambah"/>	



Tugas Pertanyaan

Keterangan :

Komponen	Nama Variabel
Combo Box Pelanggan	cb Pelanggan
Combo Box Pegawai	cb Pegawai
Combo Box Kurir	cb kurir
Text Field Alamat	tf Alamat
Text Field Tgl Pengiriman	tf Tgl Pengiriman
Combo Box Produk	cb Produk
Text Field Jumlah	tf jumlah
Text Field Diskon	tf Diskon
Button Tambah	btn tambah
Button Simpan	btn Simpan

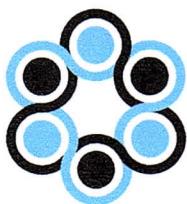


Tugas Pertanyaan

Button Input Detarz	btn Detarz
Text Preca total Harga	tf Total harga
Table kerangjang (tabel tengah)	tbl kerangjang
Table Transaksi (tabel kanan atas)	tbl Pesanan
Text Area Detarz Transaksi	ta Detarz Pesanan

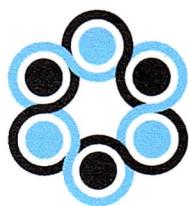
Setelah itu kita siapkan logisca programnya. Pertama kita akan menulis kode untuk koneksi dengan database di class koneksi.java pada package database. Tulis kodengnya seperti dibawah ini:

1. Package database;
- 2.
3. Import java.sql.Connection;
4. Import java.sql.DriverManager;
5. Import java.sql.ResultSet;
6. Import java.sql.SQLException;
7. Import java.sql.Statement;
- 8.
9. Public class koneksi {
- 10.
11. Private Connection connect;
12. Private Statement db;
13. Private String database = "oraclejava";
- 14.
15. Public koneksi () {
16. try {
17. Class.forName ("oracle.jdbc.driver.OracleDriver");
18. System.out.println ("Class Driver Ditemukan");
19. try {
20. Connect = DriverManager.getConnection ("jdbc:oracle:



Tugas Pertanyaan

```
    thn : @localhost : 1521 : xe", "oracle java", "oracle java  
    (23" );  
21. System.out.println (" koneksi databases sukses ");  
22. } catch (SQLException se) {  
23.     System.out.println (" koneksi Database Gagal : " + se );  
24. }  
25. } catch (ClassNotFoundException err) {  
26.     System.out.println (" class Driver tidak ditemukan,  
        Terjadi kesalahan pada : " + err );  
27. }  
28. }  
29.  
30.  
31. public ResultSet GetData ( String sql ) {  
32.     try {  
33.         db = connect.createStatement ( ResultSet.TYPE_  
            SCROLL_INSENSITIVE, ResultSet.CONCUR_UPDATABLE  
            TABLE );  
34.         return db.executeQuery ( sql );  
35.     } catch (SQLException e) {  
36.         return null;  
37.     }  
38. }  
39.  
40. public int manipulasiData ( String sql ) {  
41.     try {  
42.         db = connect.createStatement ();  
43.         return db.executeUpdate ( sql );  
44.     } catch (SQLException e) {  
45.         return 0;  
46.     }  
47. }
```



Tugas Pertanyaan

47. 3

48.

49.

50. 3

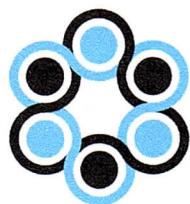
Keterangan :

Pada class ini kita mempunyai 3 fungsi yakni koneksi, GetData dan ManipulasiData. Koneksi sendiri berguna untuk membuat koneksi pada database Oracle. Fungsi ini hanya kita jalankan 1 kali pada 1 objek. Fungsi GetData adalah fungsi yang digunakan untuk mengambil data dari database, biasanya kita menggunakan Query Select untuk mengambil data dan menampungnya dalam sebuah objek ResultSet. Fungsi yang terakhir adalah Manipulasi Data yang berguna untuk mengubah data di database. Pada sana kita menggunakan Query Insert, Update dan Delete

Setelah selesai membuat koneksi kita akan membuat class Data yang akan menampung data dari database untuk diolah di lagu program Pada java. Masukan kode berikut pada setiap kelasnya :

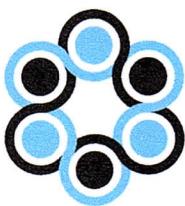
Pegawai.java

1. Package model;
- 2.
3. Import java.util.Date;
- 4.
5. Public Class Pegawai {
6. Private Integer Id_Pegawai;
7. Private String nama Depan;
8. Private String nama Belakang;
9. Private Date Tanggal Lahir;



Tugas Pertanyaan

10. Private String Alamat;
11. Private Integer kode_pos;
12. Private String No_Telp;
- 13.
14. Public Integer getId_Pegawai () {
15. Return Id_Pegawai;
16. }
- 17.
18. Public void setId_Pegawai (Integer Id_Pegawai) {
19. This.Id_Pegawai = Id_Pegawai;
20. }
- 21.
22. Public String getNama_Depan () {
23. Return Nama_Depan;
24. }
- 25.
26. Public void setNama_Depan (String Nama_Depan) {
27. This.Nama_Depan = Nama_Depan;
28. }
- 29.
30. Public String getNama_Belakang () {
31. Return Nama_Belakang;
32. }
- 33.
34. Public void setNama_Belakang (String Nama_Belakang)
35. This.Nama_Belakang = Nama_Belakang;
36. }
- 37.
38. Public Date getTanggal_Lahir () {
39. Return Tanggal_Lahir;
40. }



Tugas Pertanyaan

41.

42. Public void SetTanggal - Lahir (Date Tanggal - (lahir) {

43. this . tanggal - Lahir = Tanggal - Lahir ;

44. 3

45.

46. Public String get Alamat () {

47 Return Alamat ;

48 3

49.

50. Public void Set Alamat (String Alamat) {

51. this . Alamat = Alamat ;

52. 3

53.

54. Public Integer get Kode - pos () {

55. Return Kode - Pos ;

56. 3

57.

58. Public void Set Kode - Pos (Integer Kode - Pos) {

59. this . Kode - Pos = Kode - Pos ;

60. 3

61.

62. Public String get No - Telp () {

63. Return No - Telp ;

64. 3

65.

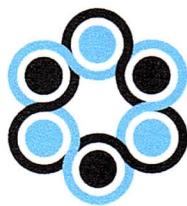
66. Public void Set No - Telp (String No - Telp) {

67. this . No - Telp = No - Telp ;

68. 3

69.

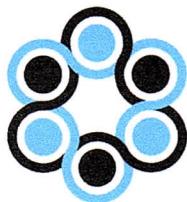
70. 3



Tugas Pertanyaan

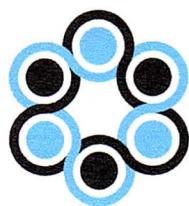
Pelanggan.java

```
1. package model;  
2.  
3. import java.util.Date;  
4.  
5. public class Pelanggan {  
6.     private Integer Id_Pelanggan;  
7.     private String Nama_Depan;  
8.     private String Nama_Belakang;  
9.     private Date Tanggal_Lahir;  
10.    private String Alamat;  
11.    private Integer Kode_Pos;  
12.    private String NO_Telp;  
13.  
14.    public Integer getId_Pelanggan () {  
15.        return Id_Pelanggan;  
16.    }  
17.  
18.    public void setId_Pelanggan (Integer Id_Pelanggan) {  
19.        this.Id_Pelanggan = Id_Pelanggan;  
20.    }  
21.  
22.    public String getNama_Depan () {  
23.        return Nama_Depan;  
24.    }  
25.  
26.    public void setNama_Depan (String Nama_Depan) {  
27.        this.Nama_Depan = Nama_Depan;  
28.    }  
29.}
```



Tugas Pertanyaan

```
30. Public String getNama - Belakang () {  
31.     return Nama - Belakang;  
32. }  
33.  
34. Public void setNama - Belakang ( String Nama - Belakang)  
35. {  
36.     this. Nama - Belakang = Nama - Belakang;  
37.  
38.     Public Date getTanggal - Lahir()  
39.     {  
40.         return Tanggal - Lahir;  
41.     }  
42.     Public void setTanggal - Lahir ( Date Tanggal - Lahir)  
43.     {  
44.         this. Tanggal - Lahir = Tanggal - Lahir;  
45.  
46.     Public String getAlamat()  
47.     {  
48.         return Alamat;  
49.  
50.     Public void setAlamat ( String Alamat) {  
51.         this. Alamat = Alamat;  
52.     }  
53.  
54.     Public Integer getCode - Pos () {  
55.         return kode - pos;  
56.     }  
57.  
58.     Public void setkode - Pos ( Integer kode - Pos) {  
59.         this. kode - pos = kode - pos;
```



Tugas Pertanyaan

60. 3

61.

62. Public String get No-Telp () {

63. return No-Telp;

64. 3

65.

66. Public void setNo_Telp (String No-Telp) {

67. this.No-Telp = No-Telp;

68. 3

69.

70.

71. 3

Kurr.java

1. Package model;

2.

3. Public class kurr {

4. Private Integer Id-kurr;

5. Private String Nama_Perusahaan;

6. Private String No-Telp;

7.

8. Public Integer get Id-kurr () {

9. Return Id-kurr;

10. 3

11.

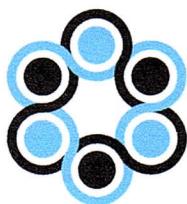
12. Public void set Id-kurr (Integer Id-kurr) {

13. this.Id-kurr = Id-kurr;

14. 3

15.

16. Public String get Nama-Perusahaan () {

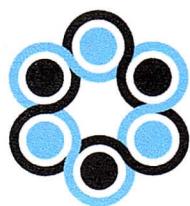


Tugas Pertanyaan

```
17.     return Nama_Perusahaan;  
18.     }  
19.  
20.     public void setNama_Perusahaan (String Nama-  
         Perusahaan) {  
21.         this.Nama_Perusahaan = Nama_Perusahaan;  
22.     }  
23.  
24.     public String getNo_Telp () {  
25.         return No_Telp;  
26.     }  
27.  
28.     public void setNo_Telp (String No_Telp) {  
29.         this.No_Telp = No_Telp;  
30.     }  
31.  
32. }
```

Kategori.java

```
1. package model;  
2.  
3. public class Kategori {  
4.     private Integer Id_Kategori;  
5.     private String Nama_Kategori;  
6.  
7.     public Integer getId_Kategori () {  
8.         return Id_Kategori;  
9.     }  
10.  
11.    public void setId_Kategori (Integer Id_Kategori) {
```

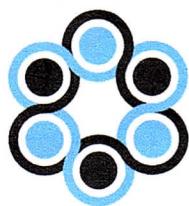


Tugas Pertanyaan

```
12. this. Id-Kategori = Id-Kategori;  
13. }  
14.  
15. Public String getNama-Kategori () {  
16.     return Nama-Kategori;  
17. }  
18.  
19. Public void setNama-Kategori (String Nama-  
20. kategori) {  
21.     this. Nama-Kategori = Nama-Kategori;  
22. }  
23.  
24. }
```

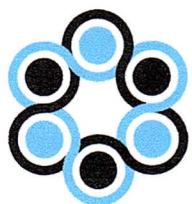
Pemasok.java

```
1. Package model;  
2.  
3. Public class Pemasok {  
4.     Private Integer Id-Pemasok;  
5.     Private String Nama-Perusahaan;  
6.     Private String Alamat;  
7.     Private Integer Kode-Pos;  
8.     Private String No-telp;  
9.  
10.    Public Integer getId-Pemasok () {  
11.        return Id-Pemasok;  
12.    }  
13.  
14.    Public void setId-Pemasok (Integer Id-Pemasok) {
```



Tugas Pertanyaan

15. this.Id_Pemasok = Id_Pemasok;
16. }
- 17.
18. Public String getNama_Perusahaan () {
19. Return Nama_Perusahaan;
20. }
- 21.
22. Public void SetNama_Perusahaan (String Nama_Perusahaan) {
23. this.Nama_Perusahaan = Nama_Perusahaan;
24. }
- 25.
26. Public String getAlamat () {
27. Return Alamat;
28. }
- 29.
30. Public void SetAlamat (String Alamat) {
31. this.Alamat = Alamat;
32. }
- 33.
34. Public Integer getKode_Pos () {
35. Return Kode_Pos;
36. }
- 37.
38. Public void setKode_Pos (Integer kode_Pos) {
39. this.Kode_Pos = Kode_Pos;
40. }
- 41.
42. Public String getNo_Telp () {
43. Return No_Telp;
44. }



Tugas Pertanyaan

45.

46. Public void set Nu-Telp (String Nu-Telp) {

47. this.Nu-Telp = Nu-Telp;

48. }

49.

50.

Produk.java

1. Package model;

2.

3. Public class Produk {

4. Private Integer Id-Produk;

5. Private Pemasok Pemasok;

6. Private Kategori Kategori;

7. Private String Nama-Produk;

8. Private Integer Harga-Satuan;

9. Private Integer Stok-Produk;

10.

11. Public Integer get Id-Produk () {

12. return Id-Produk;

13. }

14.

15. Public void set Id-Produk (Integer Id-Produk) {

16. this.Id-Produk = Id-Produk;

17. }

18.

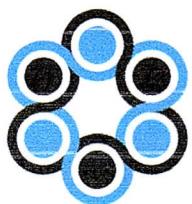
19. Public Pemasok get Pemasok () {

20. return Pemasok;

21. }

22.

23. Public void set Pemasok (Pemasok Pemasok) {



Tugas Pertanyaan

```
24. this.Pemasok = Pemasok;
25. }
26.
27. Public Kategori getKategori () {
28.     return kategori;
29. }
30.
31. Public void setKategori ( Kategori kategori ) {
32.     this.kategori = kategori;
33. }
34.
35. Public String getNama_Produk () {
36.     return Nama_Produk;
37. }
38.
39. Public void setNama_Produk (String Nama_Produk ) {
40.     this.Nama_Produk = Nama_Produk;
41. }
42.
43. Public Integer getHarga_Satuan () {
44.     return Harga_Satuan;
45. }
46.
47. Public void setHarga_Satuan ( Integer Harga_Satuan ) {
48.     this.Harga_Satuan = Harga_Satuan;
49. }
50.
51. Public Integer getStock_Produk () {
52.     return stock_Produk;
53. }
54.
```

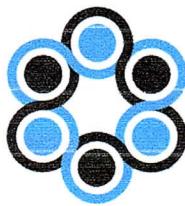


Tugas Pertanyaan

58. Public void set stock - produk (Integer stock - produk){
59. this. stock - produk = stock - produk ;
60. }
61. }
62. }

Pemesanan . Java

1. Package model;
- 2.
3. Import java. util. ArrayList;
4. Import java. util. Date;
- 5.
6. Public class Pemesanan {
7. Private Integer id - Pemesanan ;
8. Private Pelanggan Pelanggan ;
9. Private Pegawai Pegawai ;
10. Private kurir Kurir ;
11. Private Date Tanggal - Pemesanan ;
12. Private Date Tanggal - Pengiriman ;
13. Private Double Harga - Total ;
14. Private String Alamat - Pengiriman ;
15. Private ArrayList < Detail - Pemesanan > arr Detail - Pemesanan ;
- 16.
17. Public Integer getId - Pemesanan () {
18. Return id - Pemesanan ;
19. }



Tugas Pertanyaan

20.

21. Public void setId - Pemesanan (Integer Id - Pemesanan) {

22. this. Id - Pemesanan = Id - Pemesanan;

23. }

24.

25. Public Pelanggan getPelanggan () {

26. return Pelanggan;

27. }

28.

29. Public void setPelanggan (Pelanggan Pelanggan) {

30. this. Pelanggan = Pelanggan;

31. }

32.

33. Public Pegawai getPegawai () {

34. return Pegawai;

35. }

36.

37. Public void setPegawai (Pegawai Pegawai) {

38. this. Pegawai = Pegawai;

39. }

40.

41. Public Kurir getKurir () {

42. return Kurir;

43. }

44.

45. Public void setKurir (Kurir kurir) {

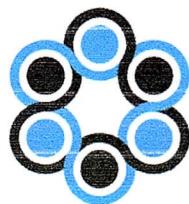
46. this. kurir = kurir;

47. }

48.

49. Public Date getTanggal - Pemesanan () {

50. return Tanggal - Pemesanan;



Tugas Pertanyaan

51. 3

52.

53. Public void setTanggal_Pemesanan (Date Tanggal_Pemesanan)
 {

54. this.Tanggal_Pemesanan = Tanggal_Pemesanan;

55. 3

56.

57. Public Date getTanggal_Pengiriman () {

58. return tanggal_Pengiriman;

59. 3

60.

61. Public void setTanggal_Pengiriman (Date Tanggal_Pengiriman) {

62. this.Tanggal_Pengiriman = Tanggal_Pengiriman;

63. 3

64.

65. Public String getAlamat_Pengiriman () {

66. return Alamat_Pengiriman;

67. 3

68.

69. Public void setAlamat_Pengiriman (String Alamat_Pengiriman)

{

70. this.Altamat_Pengiriman = Alamat_Pengiriman;

71. 3

72.

73. Public Double getHarga_Total () {

74. return Harga_Total;

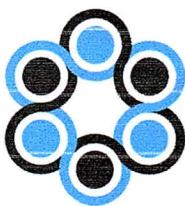
75. 3

76.

77. Public void setHarga_Total (Double Harga_Total) {

78. this.Harga_Total = Harga_Total;

79. 3



Tugas Pertanyaan

80.

81. Public void SetArrDetail_Pemesanan (ArrayList<Detail_Pemesanan> arr Detail_Pemesanan) {
arr Detail_Pemesanan = arr Detail_Pemesanan } ;

82. this. arr Detail_Pemesanan = arr Detail_Pemesanan } ;

83. } ;

84.

85. Public ArrayList<Detail_Pemesanan> getArrDetail_Pemesanan()
} ;

86. Return arr Detail_Pemesanan;

87. } ;

88.

89.

Detail_Pemesanan.java

1. Package model;

2.

3. Public class Detail_Pemesanan {

4. Private Produk produk;

5. Private Integer jumlah;

6. Private Double Diskon;

7.

8. Public Produk getProduk () {

9. Return produk;

10. } ;

11.

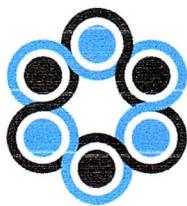
12. Public void setProduk (Produk produk) {

13. this. produk = produk;

14. } ;

15.

16. Public Integer getjumlah () {



Tugas Pertanyaan

17. return jumlah;

18. }
19.

20. Public void Set jumlah (Integer jumlah) {

21 this.jumlah = jumlah;

22. }
23.

24. Public Double get Diskon () {

25. return Diskon;

26. }
27.

28. Public void Set Diskon (Double Diskon) {

29. this.Diskon = Diskon;

30. }
31.

32. }

Setelah membuat class Data kita akan menulis Pada class transaksi.java Pada package Controller untuk memproses logika program. Silahkan tulis kode berikut :

1. Package Controller;

2.

3. Import model.Detail_Pemesanan;

4. Import model.Kategori;

5. Import model.Kurir;

6. Import model.Pegawai;

7. Import model.Pelanggan;

8. Import model.Pemasok;

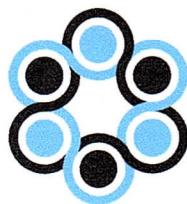
9. Import model.Pemesanan;

10. Import model.Produk;



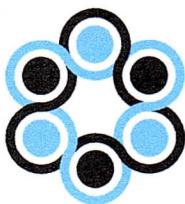
Tugas Pertanyaan

11. Import .database. Koneksi;
12. Import java.sql. ResultSet;
13. Import Java.sql. SQLException;
14. Import java.text. SimpleDateFormat;
15. Import java.util. ArrayList;
16. Import java.util. Date;
- 17.
18. Public Class Transaksi {
19. Koneksi koneksi;
20. ArrayList & Pelanggan > arr Pelanggan;
21. ArrayList & Pegawai > arr Pegawai;
22. ArrayList & kurir > arr kurir;
23. ArrayList & produk > arr Produk;
24. ArrayList & Pemesanan > arr Pemesanan;
- 25.
26. Public Transaksi () throws SQLException {
27. this. koneksi = new Koneksi();
28. this. arr Pelanggan = new ArrayList <>();
29. this. arr Pegawai = new ArrayList <>();
30. this. arr kurir = new ArrayList <>();
31. this. arr Produk = new ArrayList <>();
32. this. arr Pemesanan = new ArrayList <>();
33. }
- 34.
35. Public ArrayList & produk > getDataProduk () throws
 SQLException {
36. this. arr Produk. clear();
37. Result Set rs = this. koneksi. getdata ("SELECT * From
 Produk JOIN kategori ON Produk. ID_kategori = kategori.
 ID_kategori JOIN Pemasok ON Produk. ID_Pemasok =
 Pemasok. ID_Pemasok = Pemasok. ID_Pemasok");



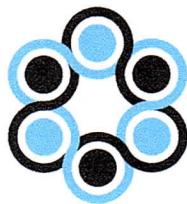
Tugas Pertanyaan

```
38. while (rs.next ()) {  
39.  
40. Pemasok pemasok = new Pemasok ();  
41. pemasok .set Id - Pemasok (rs .getInt (" ID - PEMASOK "));  
42. pemasok .set Nama - Perusahaan (rs .getString (" Nama -  
PERUSAHAAN "));  
43. pemasok .set Alamat (rs .getString (" ALAMAT "));  
44. pemasok .set Kode - Pos (rs .getInt (" KODE - POS "));  
45. pemasok .set No - Telp (rs .getString (" NO - TELP "));  
46.  
47. kategori kategori = new kategori ();  
48. kategori .set Id - kategori (rs .getInt (" ID - KATEGORI "));  
49. kategori .set Nama - kategori (rs .getString (" Nama -  
KATEGORI "));  
50.  
51. Produk produk = new Produk ();  
52. produk .set Id - Produk (rs .getInt (" ID - PRODUK "));  
53. produk .set Pemasok (pemasok );  
54. produk .set kategori (kategori );  
55. produk .set Nama - Produk (rs .getString (" NAMA - PRODUK "  
));  
56. produk .set Harga - Satuan (rs .getInt (" HARGA - SATUAN "));  
57. produk .set Stok - Produk (rs .getInt (" STOK - PRODUK "));  
58.  
59. this .arr Produk .add (produk );  
60. }  
61. return this .arr Produk ;  
62. }  
63.  
64. public ArrayList < Pelanggan > getDutaPelanggan ()  
throws SQLException {
```



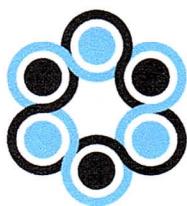
Tugas Pertanyaan

```
65. this.arr.Pelanggan.clear();
66. ResultSet rs = this.koneksi.GetData("SELECT * FROM
    PELANGGAN");
67. while (rs.next()) {
68.
69.     Pelanggan Pelanggan = new Pelanggan();
70.     Pelanggan.SetId_Pelanggan(rs.getInt("ID-PELANGGAN"));
71.     Pelanggan.SetNama_Depan(rs.getString("NAMA-DEPAN"));
72.     Pelanggan.SetNama_Belakang(rs.getString("NAMA-
    BELAKANG"));
73.     Pelanggan.setTanggal_lahir(new Date(rs.getString(
    "TANGGAL-LAHIR")));
74.     Pelanggan.SetAlamat(rs.getString("ALAMAT"));
75.     Pelanggan.setkode_Pos(rs.getInt("KODE-POS"));
76.     Pelanggan.setNo_Telp(rs.getString("NO-TELP"));
77.
78.     this.arr.Pelanggan.add(Pelanggan);
79. }
80. return this.arr.Pelanggan;
81. }
82.
83. Public ArrayList<Pegawai> getDataPegawai () throws
    SQLException {
84.     this.arr.Pegawai.clear();
85.     this.koneksi.GetData("SELECT * FROM PEGAWAI");
86.     while (rs.next()) {
87.
88.         Pegawai Pegawai = new Pegawai();
89.         Pegawai.SetId_Pegawai(rs.getInt("ID-PEGAWAI"));
90.         Pegawai.SetNama_Depan(rs.getString("NAMA-DEPAN"));
91.         Pegawai.SetNama_Belakang(rs.getString("NAMA-BELAKANG"));
92.
```



Tugas Pertanyaan

92. Pegawai. SetTanggal_Lahir (new Date (rs.getString ("Tanggal_Lahir")));
93. Pegawai. SetAlamat (rs.getString ("ALAMAT"));
94. Pegawai. SetKode_Pos (rs.getInt ("KODE_POS"));
95. Pegawai. SetNo_Telp (rs.getString ("NO_TELP"));
- 96.
97. this. arr Pegawai. add (Pegawai);
98. 3
99. Return this. arr Pegawai;
100. 3
- 101.
102. Public ArrayList & kurir > getData_kurir () throws SQL Exception {
103. this. arr kurir. Clear ();
104. ResultSet rs = this. koneksi. Getdata ("SELECT * FROM KURIR");
105. While (rs. next ()) {
- 106.
107. Kurir kurir = new Kurir ();
108. kurir. SetId_kurir (rs. getInt ("ID_KURIR"));
109. kurir. setNama_Perusahaan (rs. getString ("NAMA_PERUSAHAAN"));
110. kurir. setNo_Telp (rs. getString ("NO_TELP"));
- 111.
112. this. arr kurir. add (kurir);
113. 3
114. Return this. arr kurir;
115. 3
- 116.
117. Public ArrayList & Pesanan > getData_Pesanan () throws SQL Exception {



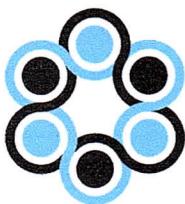
Tugas Pertanyaan

118. `this.setPemesanan().clear();`
119. `ResultSet rs = this.koneksi.GetData ("SELECT PELANGGAN.ID_PELANGGAN, PELANGGAN.NAMA_DEPAN AS NAMA_DEPAN_PELANGGAN, PELANGGAN.NAMA_BELAKANG AS NAMA_BELAKANG_PELANGGAN, PELANGGAN.TANGAL_LAHIR AS TANGAL_LAHIR_PELANGGAN, PELANGGAN.ALAMAT AS ALAMAT_PELANGGAN, PELANGGAN.KODE_POS AS KODE_POS_PELANGGAN, PELANGGAN.NO_TELP AS NO_TELP_PELANGGAN, PEGAWAI.ID_PEGAWAI, PEGAWAI.NAMA_DEPAN AS NAMA_DEPAN_PEGAWAI, PEGAWAI.NAMA_BELAKANG AS NAMA_BELAKANG_PEGAWAI, PEGAWAI.NAMA_TAHGGAL_LAHIR AS TAHGGAL_LAHIR_PEGAWAI, PEGAWAI.ALAMAT AS ALAMAT_PEGAWAI, PEGAWAI.KODE_POS AS KODE_POS_PEGAWAI, PEGAWAI.NO_TELP AS NO_TELP_PEGAWAI, PEMESANAN.*, KURIR.* FROM PEMESANAN JOIN PELANGGAN ON PEMESANAN.ID_PELANGGAN = PELANGGAN.ID_PELANGGAN JOIN KURIR ON PEMESANAN.ID_KURIR = KURIR.ID_KURIR JOIN PEGAWAI ON PEMESANAN.ID_PEGAWAI = PEGAWAI.ID_PEGAWAI ORDER BY ID_PEMESANAN DESC");`
120. `while(rs.next()) {`
121. `Pelanggan Pelanggan = new Pelanggan();`
122. `Pelanggan.setId_Pelanggan(rs.getInt("ID_PELANGGAN"));`
123. `Pelanggan.setName_Depan(rs.getString("NAMA_DEPAN_PELANGGAN"));`
124. `Pelanggan.setName_Belakang(rs.getString("NAMA_BELAKANG_PELANGGAN"));`
125. `Pelanggan.setDate_Lahir(new Date(rs.getString("TANGAL_LAHIR_PELANGGAN")));`
126. `Pelanggan.setAlamat(rs.getString("ALAMAT_PELANGGAN"));`



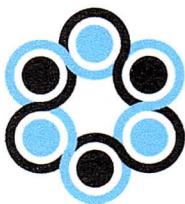
Tugas Pertanyaan

127. Pelanggan . SetKode_Pos (rs.getInt ("KODE_POS_PELANGGAN"));
128. Pelanggan . SetNo_Telp (rs.getString ("NO_TELP_PELANGGAN"));
- 129.
130. Pegawai Pegawai = new Pegawai ();
131. Pegawai . SetId_Pegawai (rs.getInt ("ID_PEGAWAI"));
132. Pegawai . SetNama_Belakang (rs.getString ("NAMA_BELAKANG_PEGAWAI"));
133. Pegawai . setNama_Depan (rs.getString ("NAMA_DEPAN_PEGAWAI"));
134. Pegawai . setTanggal_Lahir (new Date (rs.getString ("TANGGAL_LAHIR_PEGAWAI")));
135. Pegawai . SetAlamat (rs.getString ("ALAMAT_PEGAWAI"));
136. Pegawai . setKode_Pos (rs.getInt ("KODE_POS_PEGAWAI"));
137. Pegawai . SetNo_Telp (rs.getString ("NO_TELP_PEGAWAI"));
- 138.
139. kurir kurir = new kurir ();
140. kurir . SetId_Kurir (rs.getInt ("ID_KURIR"));
141. kurir . SetNama_Perusahaan (rs.getString ("NAMA_PERUSAHAAN"));
142. kurir . SetNo_Telp (rs.getString ("NO_TELP"));
- 143.
144. Pemesanan Pemesanan = new Pemesanan ();
145. Pemesanan . SetId_Pemesanan (rs.getInt ("ID_PEMESANAN"));
146. Pemesanan . SetPelanggan (Pelanggan);
147. Pemesanan . SetPegawai (Pegawai);
148. Pemesanan . SetKurir (kurir);
149. Pemesanan . setTanggal_Pemesanan (new Date (rs.getString ("TANGGAL_PEMESANAN")));
150. Pemesanan . SetTanggal_Pengiriman (new Date (rs.getString ("TANGGAL_PENGIRIMAN")));



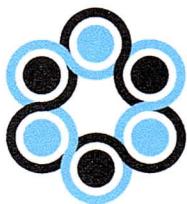
Tugas Pertanyaan

```
( "TANGGAL - PENGIRIMAN" )));  
151. Pemesanan . SetAlamat_Pengiriman (rs.getString (" ALAMAT -  
PENGIRIMAN "));  
152. Pemesanan . SetHarga_Total ( rs.getDouble (" HARGA - TOTAL "));  
153.  
154. ResultSet rsDetail_Pemesanan = fhs . koneksi . Get Data  
(" SELECT * FROM DETAIL - PEMESANAN JOIN PRODUK ON  
DETAL - PEMESANAN . ID - PRODUK = PRODUK . ID - PRODUK  
JOIN PEMASOK ON PRODUK . ID - PEMASOK = PEMASOK .  
ID - PEMASOK JOIN KATEGORI ON PRODUK . ID - KATEGORI =  
KATEGORI . ID - KATEGORI WHERE DETAL - PEMESANAN .  
ID - PEMESANAN = " + rs.getString (" ID - PEMESANAN "));  
155. ArrayList < Detail_Pemesanan > dp = new ArrayList < >();  
156.  
157. while (rs Detail - Pemesanan . next (1) {  
158. Pemasok Pemasok = new Pemasok ();  
159. Pemasok . SetId_Pemasok (rs Detail - Pemesanan . getInt (  
" ID - PEMASOK "));  
160. Pemasok . SetNama_Perusahaan (rs Detail - Pemesanan . getString  
(" NAMA PERUSAHAAN "));  
161. Pemasok . SetAlamat (rs Detail - Pemesanan . getString  
(" ALAMAT "));  
162. Pemasok . SetKode_Pos (rs Detail - Pemesanan . getInt  
(" KODE - POS "));  
163. Pemasok . SetNo_Telp (rs Detail - Pemesanan . getString  
(" NO - TELP "));  
164.  
165. kategori Kategori = new Kategori ();  
166. Kategori . setId_Kategori (rs Detail - Pemesanan . getInt  
(" ID - KATEGORI "));  
167. Kategori . SetNama_Kategori (rs Detail - Pemesanan .
```



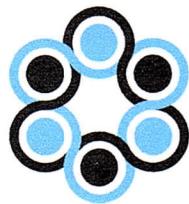
Tugas Pertanyaan

```
get String (" NAMA_KATEGORI "));  
168.  
169. Produk Produk = new Produk();  
170. Produk .Set Id - Produk (rs Detail - Pemesanan .get Int  
(" ID-PRODUK "));  
171. Produk .Set Pemasok (Pemasok);  
172. Produk .Set Kategori (Kategori);  
173. Produk .Set Nama - Produk (rs Detail - Pemesanan  
get String (" NAMA-PRODUK "));  
174. Produk .Set Harga - Satuan (rs Detail - Pemesanan .get Int  
(" HARGA-SATUAN "));  
175. Produk .Set Stok - Produk (rs Detail - Pemesanan .get Int  
(" STOK-PRODUK "));  
176.  
177. Detail - Pemesanan detail_Pemesanan = new Detail -  
Pemesanan ();  
178. detail_Pemesanan .set Produk (Produk);  
179. detail_Pemesanan .set Jumlah (rs Detail - Pemesanan  
get Int (" JUMLAH "));  
180. detail_Pemesanan .set Diskon (rs Detail - Pemesanan  
get Double (" DISKON "));  
181.  
182. dp.aaa (detail - Pemesanan );  
183. }  
184. Pemesanan .set Attr Detail - Pemesanan (dp);  
185.  
186. this. arr Pemesanan .aaa (Pemesanan );  
187. }  
188. return this. arr Pemesanan ;  
189. }  
190. }
```



Tugas Pertanyaan

191. public void insertTransaksi (Pemesanan Pemesanan) {
192. try {
193. String datePemesanan = new SimpleDateFormat ("dd/MM/yyyy").format (Pemesanan.getTanggal - Pemesanan());
194. String datePengiriman = new SimpleDateFormat ("dd/MM/yyyy").format (Pemesanan.getTanggal - Pengiriman());
195. this.koneksi.ManipulasiData ("INSERT INTO PEMESANAN VALUES (ID-PEMESANAN, NEXTVAL, " + Pemesanan.getId - Pelanggan () + "," + Pemesanan.getId - Pegawai () + "," + Pemesanan.getKurir () .getId - Kurir () + ", TO-DATE ('" + datePengiriman + "', 'dd/MM/yyyy'), TO-DATE ('" + datePengiriman + "', 'dd/MM/yyyy'), '" + Pemesanan.getAlamat - Pengiriman () + "', " + Pemesanan.getHarga - Total () .toString () + ")");
196. ResultSet rs = this.koneksi.GetData ("SELECT ID-PEMESANAN - CURRVAL FROM DUAL");
197. rs.next();
198. int Id - Pemesanan = rs.getInt ("CURRVAL");
199. for (Detail - Pemesanan p : Pemesanan.getArrDetail - Pemesanan ()) {
200. this.koneksi.ManipulasiData ("INSERT INTO DETAIL-PEMESANAN VALUES (" + p.getIdProduk () .getId - Produk () + "," + Id - Pemesanan + "," + p.getIdProduk () .getJumlah () + "," + p.getIdProduk () + ")");
201. this.koneksi.ManipulasiData ("UPDATE PRODUK SET STOK - PRODUK = STOK - PRODUK - " + p.getIdProduk () .getJumlah () + " WHERE ID-PRODUK = " + p.getIdProduk () .getId - Produk ());
202. }
203. } catch (SQLException e) {



Tugas Pertanyaan

205. System.out.println (ex);

206. ?

207. ?

208. ?

Setelah selesai mengerjakan tugas menyelesaikan tampilan yang sudah dibuat pada class TransaksiView.java dengan logika program pada Transaksi.java
Pada class TransaksiView.java buat Objek sebagai berikut:

1. Private Transaksi transaksi;

2. Private ArrayList<Detail_Pemesanan> arrkerangang;

3. Private Double hargatal;

Pada constructor tentukan juga berikut:

1. this.hargatal = 0.0;

2. this.transaksi = new Transaksi();

3. this.arrkerangang = new ArrayList<>();

4. this.showTable_Pemesanan();

5. this.showCombo_Box_Pelanggan();

6. this.showCombo_Box_Pegawai();

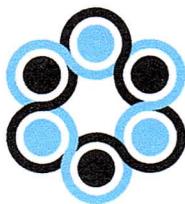
7. this.showCombo_Box_Kontr();

8. this.showCombo_Box_Prodak();

9. this.showTable_Kerangang();

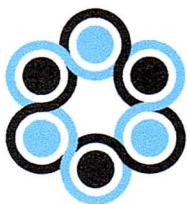
10. this.showHargaTotal();

Buat fungsi sebagai berikut:



Tugas Pertanyaan

1. Public void ShowTableKeranjang () {
2. DefaultTableModel dtmKeranjang = new DefaultTableModel()
(new String [] {"Nama Produk", "Harga Satuan", "Jumlah",
"Diskon"}, 0);
3. dtmKeranjang . SetRowCount (0);
- 4.
5. For (Detail_Pemesanan dp : this . arrKeranjang) {
6. dtmKeranjang . addRow (new String [] {dp . getProduk () .
getNama_Produk (), dp . getProduk () . getHarga_Satuan () .
toString (), dp . getJumlah () . toString (), dp . getDiskon () .
toString ()});
7. }
- 8.
- 9.
10. this . HblKeranjang . SetModel (dtmKeranjang);
11. }
- 12.
13. Public void ShowComboBoxPelanggan () throws SQLException
{
14. DefaultComboBoxModel dcblPelanggan = new DefaultComboBoxModel ();
- 15.
16. For (Pelanggan P : this . transaksi . getDataPelanggan ()) {
17. dcblPelanggan . addElement (P . getNama_Depan () + " " + P . get
Nama_Belakang ());
 18. }
- 19.
20. this . cbPelanggan . SetModel (dcblPelanggan);
21. }
- 22.
23. Public void ShowComboBoxPegawai () throws SQLException {



Tugas Pertanyaan

24. DefaultComboBoxModel dcblm Pegawai = new DefaultComboBoxModel();

25.

26. for (Pegawai p : this.transactions.getDataPegawai ()) {

27. dcblm Pegawai. addElement (p.getNamaDepan () + " " +
p.getNamaBelakang ());

28. }

29.

30. this.cb Pegawai. setModel (dcblm Pegawai);

31. }

32.

33. Public void showComboBoxKurir () throws SQLException {

34. DefaultComboBoxModel dcblm Kurir = new DefaultComboBoxModel();

35.

36. for (Kurir k : this.transactions.getDataKurir ()) {

37. dcblm.Kurir. addElement (k.getNamaPerusahaan ());

38. }

39.

40. this.cb Kurir. setModel (dcblm.Kurir);

41. }

42.

43. Public void showComboBoxProduk () throws SQLException {

44. DefaultComboBoxModel dcblm Produk = new DefaultComboBoxModel();

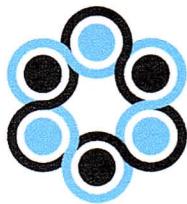
45.

46.

47. for (Produk p : this.transactions.getDataProduk ()) {

48. dcblm Produk. addElement (p.getNamaProduk () + " ("
Stok " + p.getStockProduk () + ")");

49.



Tugas Pertanyaan

50. 3

51. this. cb Produk. Remove All Items();

52. this. cb Produk. Set Model (dcbm Produk);

53. 3

54.

55. Public void showHargaTotal () {

56. this. tf total Harga. Set Text (this. hargaTotal, to String());

57. 3

Pada Komponen btn Simpan tambahkan Action Performer

sebagai berikut:

1. Private void btnSimpanAction Performer (Java.awt.event.ActionEvent e) {

2. Pemesanan = new Pemesanan();

3. try {

4. Pemesanan. Set Pelanggan (this. transaksi. getData Pelanggan());
get (this. cb Pelanggan. get SelectedIndex ());

5. Pemesanan. Set Pegawai (this. transaksi. getData Pegawai());
get (this. cb Kurr. get SelectedIndex ());

6. Pemesanan. Set kurir (this. transaksi. getDataKurir());
get (this. cb kurir. get SelectedIndex ());

7. Pemesanan. Set tanggal_Pemesanan (new Date());

8. Pemesanan. Set tanggal_Pengiriman (new SimpleDateFormat()
"dd/MM/yyyy"). Parse (this. tf TglPengiriman. get Text());

9. Pemesanan. Set Alamat_Pengiriman (this. tf Alamat. get Text());

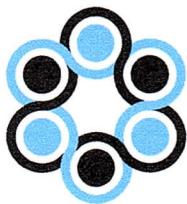
10. Pemesanan. Set Harga_Total (this. hargaTotal);

11. Pemesanan. Set ArrDetail_Pemesanan (this. arrkerangang);

12. this. transaksi. Insert Transaksi (Pemesanan);

13. this. showTable Pemesanan();

14. this. arrkerangang = new ArrayList<Y>();

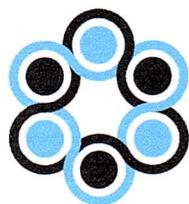


Tugas Pertanyaan

```
15. this.showTableKeranginan();  
16. this.hargaTotal = 0.0;  
17. this.showHargaTotal();  
18. this.showComboBoxProduct();  
19. } catch (SQLException | ParseException err) {  
20. System.out.println(err);  
21. }  
22.  
23. }
```

Pada btr tambah tambahkan kode berikut:

```
1. Private void btrTambahActionPerformed (java.awt.event.ActionEvent evt) {  
2. try {  
3. Detai_Pemesanan dP = new Detai_Pemesanan ();  
4. dP.setJumlah (Integer.parseInt (tfJumlah.getText ()));  
5. dP.setProduct (this.transaksi.getDataProduct () .get (cbProduct  
.getSelectedIndex ()));  
6.  
7. If (this.tfDiskon.getText () .equals ("")) {  
8. dP.setDiskon (Double.parseDouble ("0"));  
9. this.transaksi.getDataProduct () .get (cbProduct.get.SelectedIndex  
()); get Harga_Satuan () * Integer.parseInt (tfJumlah  
.getText ()) ;  
10. } else {  
11. Double diskon = Double.parseDouble (this.tfDiskon.getText  
());  
12. Integer hargaTotalProduct = this.transaksi.getDataProduct  
().get (cbProduct.get.SelectedIndex ()) .get Harga_Satuan  
() * Integer.parseInt (tfJumlah.getText ()) ;
```

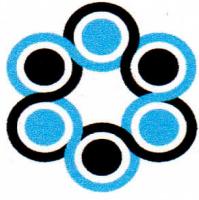


Tugas Pertanyaan

13. Double hargaTotalDiskon = hargaTotalProduk * diskon;
14. dp. setDiskon (diskon);
15. this. hargaTotal + hargaTotalProduk = hargaTotalDiskon;
16. }
- 17.
18. this. showHargaTotal ();
19. this. arrKerangjang.setData (dp);
20. this. showTableKerangjang ();
21. this. tfJumlah. setText ("");
22. this. tpDiskon. setText ("");
23. } catch (SQLException ex) {
24. Logger.getLogger (transaksiView.class.getName ()) .log (Level.SEVERE, null, ex);
25. }
26. }

Pada btnDetail tambahkan kode berikut!

1. Private void btnDetailAction (ActionEvent e) {
2. try {
3. Pemesanan pemesanan = this.transaksi. getDataPemesanan () .get (this.tblPesanan .getSelectedRow ());
4. String teks = "
5. " + "======" + "====" + "PELANGGAN" + "====" + "====" + "
6. + "Nama Depan : " + pemesanan.getPelanggan () .getNamaDepan () + "\n"
7. + "Nama Belakang : " + pemesanan.getPelanggan () .getNamaBelakang () + "\n"
8. + "Tanggal Lahir : " + new SimpleDateFormat ("dd/MM/yyyy")



Tugas Pertanyaan

- format(Pemesanan.getPelanggan()), getTanggal_lahir() +
"\\n"
 - 9. + "Alamat : " + Pemesanan.getPelanggan(), getAlamat() +
"\\n"
 - 10. + "Kode Pos : " + Pemesanan.getPelanggan(), getKode_pos() +
"\\n"
 - 11. + "No. telp : " + Pemesanan.getPelanggan(), getNo_Telp() + "\\n"
 - 12. + "====PEGAWAI====" + "\\n"
 - 13. + "Nama Depan : " + Pemesanan.getPegawai(), getNama_Depan()
+ "\\n"
 - 14. Pemesanan.getPegawai(), getName_Belakang() + "\\n"
 - 15. + "tanggal Lahir : " + new SimpleDateFormat("dd/MM/yyyy").
format(Pemesanan.getPegawai(), getTanggal_lahir()) + "\\n"
 - 16. + "Alamat : " + Pemesanan.getPegawai(), getAlamat() + "\\n"
 - 17. + "Kode Pos : " + Pemesanan.getPegawai(), getKode_Pos()
+ "\\n"
 - 18. + "No. Telp : " + Pemesanan.getPegawai(), getNo_Telp() + "\\n"
 - 19. + "=====KURIR===== "+ "\\n"
 - 20. + "Nama Perusahaan : " + Pemesanan.getKurir(), getNama_Perusahaan()
+ "\\n"
 - 21. + "No. Telp : " + Pemesanan.getKurir(), getNo_Telp() + "\\n"
 - 22. + "=====PESANAN===== "+ "\\n"
 - 23. + new SimpleDateFormat("dd/MM/yyyy").format(
(Pemesanan.getTanggal_Pemesanan())) + "\\n"
 - 24. + "Tanggal Pemesanan : " + new SimpleDateFormat(
"dd/MM/yyyy").format(Pemesanan.getTanggal_
Pengiriman()) + "\\n"
 - 25. + "Alamat Pengiriman : " + Pemesanan.getAlamat_
Pengiriman() + "\\n"
 - 26. + "Harga total : " + Pemesanan.getHarga_total()



Tugas Pertanyaan

- + "ln"
27. + " Produk : " + "ln";
28. ArrayList<Detail_Pemesanan> dp = Pemesanan.getArr
Detail_Pemesanan();
29. for(int i=0; i < dp.size(); i++) {
30. teks += "lnln" + (i+1) + ". "
31. + " Perusahaan Pemasok : " + dp.get(i).getProduk().get
Pemasok().getNama_Perusahaan() + "ln"
32. + " Alamat Pemasok : " + dp.get(i).getProduk().
getPemasok().getAlamat() + "ln"
33. + " Kode Pos Pemasok : " + dp.get(i).getProduk().
getPemasok().getKode_Pos() + "ln"
34. + " No. Telp Pemasok : " + dp.get(i).getProduk().get
Pemasok().getNo_Telp() + "ln"
35. + " kategori Produk : " + dp.get(i).getProduk().get
kategori().getNama_kategori() + "ln"
36. + " Nama Produk : " + dp.get(i).getProduk().
getNama_Prodak() + "ln"
37. + " Harga Produk : " + dp.get(i).getProduk().getHarga_
Satuan() + "ln"
38. + " Stok Produk : " + dp.get(i).getProduk().getStok_
Produk() + "ln"
39. + " jumlah Beli Produk : " + dp.get(i).getJumlah() + "ln"
40. + " Diskon Produk : " + dp.get(i).getDiskon();
- 41.
42. }
43. this.taDetail_Pesanan.setText(teks);
44. } catch (SQLException ex) {
45. System.out.println(ex);
46. }
- 47.



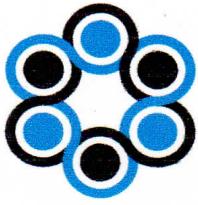
Tugas Pertanyaan

457 3

Setelah semuanya selesai tinggal kita panggil krew tersebut
ke dalam class main yakni oracle.java.java pada package
Oracle java

1. Package oracle.java;
- 2.
3. Import Krew . transaksiKrew;
4. Import java. sql. SQLException;
5. Import java. text. ParseException;
- 6.
7. Public class Oracle Java {
- 8.
9. /* *
10. * @ Param args the command line arguments
11. */
12. Public static void main (String [] args) throws SQLException,
ParseException
13. new TransaksiKrew(). show();
14. }
15. .
16. }

Hasil program adalah sebagai berikut:



Tugas Pertanyaan

Pelanggan :

Pegawai : (V)

Kunr : 17

Alamat :

tgl Pengiriman :

Pranale : IV

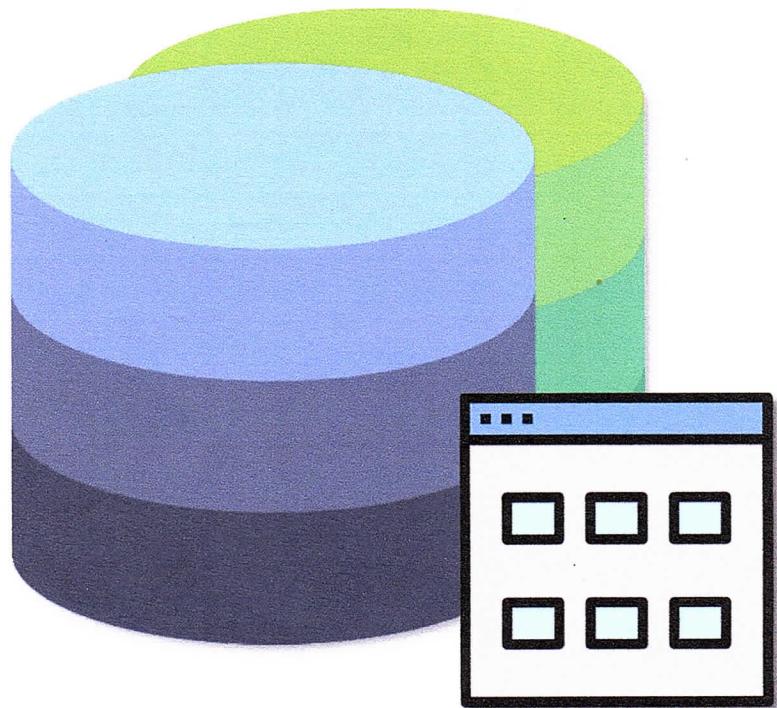
Jumlah :

Diseño:

tambak

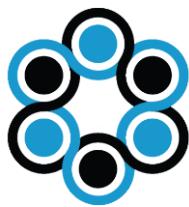
Total Harga : 0,0

LAPORAN PRAKTIKUM



PRAKTIKUM BASIS DATA PERIODE V - Tahun 2019/2020

NAMA : Machamad Khartul Huda
NPM : 06.2018.1.06890
MODUL : ✓



Tugas Praktikum

Soal Praktikum

1. Buat koneksi database Oracle dengan Java/Web berdasarkan project masing-masing.
2. Buatlah CRUD pada salah satu tabel (boleh menggunakan GUI, Console atau Web).
3. Terapkan salah satu konsep JOIN dan tampilkan pada project Anda.
4. Terapkan salah satu konsep VIEW dan tampilkan pada project Anda.

Langkah Pertama

Buat file untuk konfigurasi koneksi dari database. Lalu kita import beberapa package dari java.sql. Untuk konfigurasinya seperti dibawah ini

Query

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class Koneksi {
    private Connection connect;
    private Statement db;
    private String database = "oraclejava";

    public Koneksi() {
        try {

Class.forName("oracle.jdbc.driver.OracleDriver");
            System.out.println("Class Driver Ditemukan");
            try {
                connect =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","khoirul_huda_06990","kh
oirulhuda");
                System.out.println("Koneksi DataBase Ditemukan");
            }
        }
    }
}
```

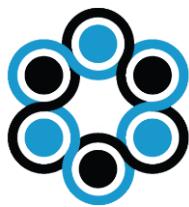


Tugas Praktikum

```
 }catch (SQLException se) {
        System.out.println("Koneksi
Database Gagal : " + se);
    }
} catch (ClassNotFoundException err)
{
    System.out.println("Class Driver
Tidak Ditemukan, Terjadi Kesalahan Pada : " +
err);
}
}

public ResultSet GetData(String sql){
    try {
        db =
connect.createStatement	ResultSet.TYPE_SCROLL
_INSENSITIVE, ResultSet.CONCUR_UPDATABLE);
        return db.executeQuery(sql);
    } catch (SQLException e) {
        return null;
    }
}

public int ManipulasiData(String sql) {
    try {
        db = connect.createStatement();
        return db.executeUpdate(sql);
    } catch(SQLException e) {
        return 0;
    }
}
}
```



Tugas Praktikum

Screenshot

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class Koneksi {
    private Connection connect;
    private Statement db;
    private String database = "oraclejava";

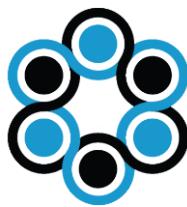
    public Koneksi() {
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            System.out.println("Class Driver Ditemukan");
            try {
                connect = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","Khoirul_huda_06990","Khoirulhuda");
                System.out.println("Koneksi DataBase Ditemukan");
            }catch (SQLException se) {
                System.out.println("Koneksi DataBase Gagal : " + se);
            }
        } catch (ClassNotFoundException err) {
            System.out.println("Class Driver Tidak Ditemukan, Terjadi Kesalahan Pada : " + err);
        }
    }

    public ResultSet GetData(String sql){
        try {
            db = connect.createStatement	ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_UPDATABLE);
            return db.executeQuery(sql);
        } catch (SQLException e) {
            return null;
        }
    }

    public int ManipulasiData(String sql) {
        try {
            db = connect.createStatement();
            return db.executeUpdate(sql);
        } catch(SQLException e) {
            return 0;
        }
    }
}
```

Analisa

Tahap pertama kita akan mengimport beberapa package yang ada dalam `java.sql`. Lalu kita akan mengkonfigurasikan sebuah DB dengan salah satu package. Untuk `resultset` adalah sebuah method yang beguna untuk mengatur menampilkan sebuah data. `ManipulasiData` adalah sebuah methoda yang mengatur sebuah peritah query.



Tugas Praktikum

Langkah Kedua

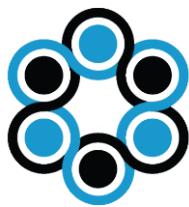
Disini kita akan membuat sebuah method yang akan mengatur sebuah query
Create, Read, Update, Delete

Query

```
public ArrayList<Marketing> getMarketing()
() throws SQLException {
    this.marketing.clear();
    ResultSet rs =
this.koneksi.GetData("SELECT * FROM
MARKETING");
    while (rs.next()) {
        Marketing m = new Marketing();

        m.setId(rs.getInt("ID_MARKETING"));

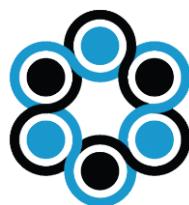
        m.setNama(rs.getString("NAMA_MARKETING"));
        m.setTelp(rs.getInt("NO_TELP"));
        this.marketing.add(m);
    }
    return this.marketing;
}
public void InsertAMP(String table,String
nama_sequence, String nama, String WT){
    String sql = "INSERT INTO "+table+
VALUES('"+nama_sequence+"','"+nama+"','"+WT+
's')";
    this.koneksi.ManipulasiData(sql);
}
public void UpdateAMP(String table,int
id, String update){
    String sql = null;
    if(null != table)switch (table) {
        case "arsitek":{
            sql = "UPDATE "+table+" set
telp_kantor = "+update+" where id_arsitek =
"+id;
            break;
        }
        case "marketing":{
            sql = "UPDATE "+table+" set
no_telp = "+update+" where id_marketing =
"+id;
        }
    }
}
```



Tugas Praktikum

```
        break;
    }
    case "profil_aluminium":{
        sql = "UPDATE "+table+" set
warna = "+update+" where id_profil = "+id;
        break;
    }
    default:
        break;
}
this.koneksi.ManipulasiData(sql);
}

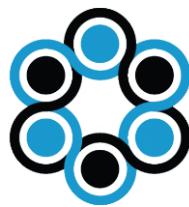
public void DeleteAMP(String table, int
id){
    String sql = null;
    switch (table) {
        case "arsitek":
            sql = "DELETE FROM "+table+
WHERE id_arsitek = "+id;
            break;
        case "marketing":
            sql = "DELETE FROM "+table+
WHERE id_marketing = "+id;
            break;
        case "profil_aluminium":
            sql = "DELETE FROM "+table+
WHERE id_profil = "+id;
            break;
        default:
            break;
    }
    this.koneksi.ManipulasiData(sql);
}
```



Tugas Praktikum

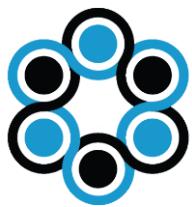
Screenshot

```
===== Selamat Datang =====
===== Menu =====
1 : Create
2 : Delete
3 : Update
4 : Select
5 : Select Join
6 : Select View
Masukan menu yang anda inginkan : 4
===== Select =====
ID      Nama    NO. TELP
=====
1       rahmad  123456789
2       artani   234567891
3       hendro   345678912
4       france   456789123
5       yudha   567891234
===== Ulang =====
Anda mau mengulangi (y/n)y
===== Selamat Datang =====
===== Menu =====
1 : Create
2 : Delete
3 : Update
4 : Select
5 : Select Join
6 : Select View
Masukan menu yang anda inginkan : 1
===== Create =====
Masukan nama : budi
Masukan NO. Telp : 987654321
===== Ulang =====
Anda mau mengulangi (y/n)y
===== Menu =====
1 : Create
2 : Delete
3 : Update
4 : Select
5 : Select Join
6 : Select View
Masukan menu yang anda inginkan : 4
===== Select =====
ID      Nama    NO. TELP
=====
1       rahmad  123456789
2       artani   234567891
3       hendro   345678912
4       france   456789123
5       yudha   567891234
21      budi     987654321
===== Ulang =====
Anda mau mengulangi (y/n)y
```



Tugas Praktikum

```
===== Menu =====
1 : Create
2 : Delete
3 : Update
4 : Select
5 : Select Join
6 : Select View
Masukan menu yang anda inginkan : 3
===== Update =====
Masukan ID Marketing : 21
Masukan No. Telp baru : 999999999
===== Ulang =====
Anda mau mengulangi (y/n)y
===== Menu =====
1 : Create
2 : Delete
3 : Update
4 : Select
5 : Select Join
6 : Select View
Masukan menu yang anda inginkan : 4
===== Select =====
ID      Nama    NO. TELP
=====
1      rahmad  123456789
2      artani   234567891
3      hendro   345678912
4      france   456789123
5      yudha    567891234
21     budi     999999999
===== Ulang =====
Anda mau mengulangi (y/n)|
===== Menu =====
1 : Create
2 : Delete
3 : Update
4 : Select
5 : Select Join
6 : Select View
Masukan menu yang anda inginkan : 2
===== Delete =====
Masukan ID Marketing : 21
===== Ulang =====
Anda mau mengulangi (y/n)y
===== Menu =====
1 : Create
2 : Delete
3 : Update
4 : Select
5 : Select Join
6 : Select View
Masukan menu yang anda inginkan : 4
===== Select =====
ID      Nama    NO. TELP
=====
1      rahmad  123456789
2      artani   234567891
3      hendro   345678912
4      france   456789123
5      yudha    567891234
===== Ulang =====
Anda mau mengulangi (y/n)|
```



Tugas Praktikum

Analisa :

Disini kita sudah membuat method yang akan dipergunakan create, read, update, dan delete. Dengan bantuan resultset yang ada pada java.



Tugas Praktikum

Langkah Ketiga

Kita akan membuat sebuah query join pada program kita

Query

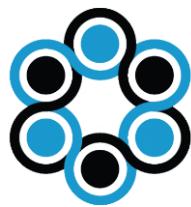
```
public ArrayList<Pengguna_Profil> getPp()
throws SQLException {
    this.pp.clear();
    ResultSet rs =
this.koneksi.GetData("SELECT
marketing.nama_marketing,
profil_aluminium.nama_profil,count(marketing.
nama_marketing) AS JUMLAH from proyek right
join profil_aluminium on
profil_aluminium.id_profil = proyek.id_profil
right join marketing on
marketing.id_marketing = proyek.id_marketing
where profil_aluminium.nama_profil =
'standard' and profil_aluminium.warna = 'iron
grey' group by marketing.nama_marketing,
profil_aluminium.nama_profil");
    while(rs.next()){
        Pengguna_Profil pProfil = new
Pengguna_Profil();

pProfil.setMarketing(rs.getString("NAMA_MARKE
TING"));

pProfil.setProfil(rs.getString("NAMA_PROFIL"))
;

pProfil.setJumlah(rs.getInt("JUMLAH"));

        this.pp.add(pProfil);
    }
    return this.pp;
}
```



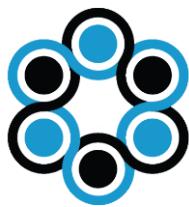
Tugas Praktikum

Screenshot

```
===== Menu =====
1 : Create
2 : Delete
3 : Update
4 : Select
5 : Select Join
6 : Select View
Masukan menu yang anda inginkan : 5
===== Join =====
Marketing      Profil      Jumlah
=====
rahmad        standart      2
artani        standart      1
```

Analisa :

Disini kita membuat join dari beberapa table dan kita langsung menampilkan isinya



Tugas Praktikum

Langkah Keempat

Pada langkah keempat ini kita sudah pernah membuat sebuah view pada database saat modul 4. Disini kita akan memanggil view tersebut

Query

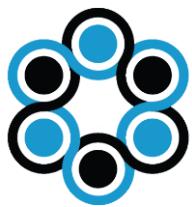
```
public ArrayList<Kepemilikan_Proyek>
getkp() throws SQLException {
    this.kp.clear();
    ResultSet rs =
this.koneksi.GetData("SELECT * FROM
KEPEMILIKAN_PROYEK");
    while (rs.next()) {
        Kepemilikan_Proyek kProyek = new
Kepemilikan_Proyek();

kProyek.setMarketing(rs.getString("NAMA_MARKE
TING"));

kProyek.setOwner(rs.getString("NAMA_OWNER"));
        this.kp.add(kProyek);
    }
    return this.kp;
}
```

Screenshot

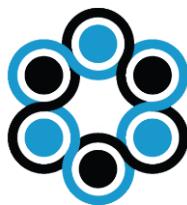
```
===== Menu =====
1 : Create
2 : Delete
3 : Update
4 : Select
5 : Select Join
6 : Select View
Masukan menu yang anda inginkan : 6
===== View =====
Marketing          Owner
=====
artani            kardono
rahmad           sudarmono
rahmad           yunus
artani            alex
artani            yos tanoyo
```



Tugas Praktikum

Analisa :

Disini kita hanya menampilkan bagian dari view yang sudah kita buat pada modul 4



Laporan Sementara

Soal Praktikum

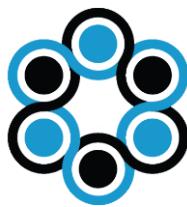
1. Buat koneksi database Oracle dengan Java/Web berdasarkan project masing-masing.
2. Buatlah CRUD pada salah satu tabel (boleh menggunakan GUI, Console atau Web).
3. Terapkan salah satu konsep JOIN dan tampilkan pada project Anda.
4. Terapkan salah satu konsep VIEW dan tampilkan pada project Anda.

Langkah Pertama

(Ketik penjelasan langkah – langkah disini)

Query

```
try {  
    Class.forName("oracle.jdbc.driver.OracleDriver");  
    System.out.println("Class Driver Ditemukan");  
    try {  
        connect =  
        DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","khoirul_huda_06990","kh  
oirulhuda");  
        System.out.println("Koneksi DataBase Ditemukan");  
    }catch (SQLException se) {  
        System.out.println("Koneksi Database Gagal : " + se);  
    }  
} catch (ClassNotFoundException err)  
{  
    System.out.println("Class Driver Tidak Ditemukan, Terjadi Kesalahan Pada : " +  
err);  
}
```



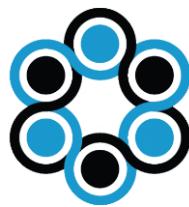
Laporan Sementara

Screenshot

```
public Koneksi() {
    try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        System.out.println("Class Driver Ditemukan");
        try {
            connect = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","khoirul_huda_06990","khoirulhuda");
            System.out.println("Koneksi DataBase Ditemukan");
        }catch (SQLException se) {
            System.out.println("Koneksi Database Gagal : " + se);
        }
    } catch (ClassNotFoundException err) {
        System.out.println("Class Driver Tidak Ditemukan, Terjadi Kesalahan Pada : " + err);
    }
}
```

Analisa

(Ketik hasil analisa disini)



Laporan Sementara

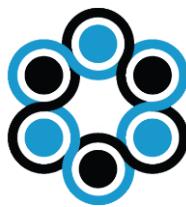
Langkah Kedua

(Ketik penjelasan langkah – langkah disini)

Query

```
public void InsertAMP(String table,String nama_sequence, String nama, String WT){
    String sql = "INSERT INTO "+table+
VALUES("+nama_sequence+","+nama+","+WT+");
    this.koneksi.ManipulasiData(sql);
}

public void UpdateAMP(String table,int id,
String update){
    String sql = null;
    if(null != table)switch (table) {
        case "arsitek":{
            sql = "UPDATE "+table+ set
telp_kantor = "+update+" where id_arsitek =
"+id;
            break;
        }
        case "marketing":{
            sql = "UPDATE "+table+ set
no_telp = "+update+" where id_marketing =
"+id;
            break;
        }
        case "profil_aluminium":{
            sql = "UPDATE "+table+ set
warna = "+update+" where id_profil = "+id;
            break;
        }
        default:
            break;
    }
    this.koneksi.ManipulasiData(sql);
}
```



Laporan Sementara

```
public void DeleteAMP(String table, int id){
    String sql = null;
    switch (table) {
        case "arsitek":
            sql = "DELETE FROM "+table+
WHERE id_arsitek = "+id;
            break;
        case "marketing":
            sql = "DELETE FROM "+table+
WHERE id_marketing = "+id;
            break;
        case "profil_aluminium":
            sql = "DELETE FROM "+table+
WHERE id_profil = "+id;
            break;
        default:
            break;
    }
    this.koneksi.ManipulasiData(sql);
}
```

Screenshot

```
public void InsertAMP(String table, String nama_sequence, String nama, String WT){
    String sql = "INSERT INTO "+table+" VALUES("+nama_sequence+","+nama+","+WT+)";
    this.koneksi.ManipulasiData(sql);
}

public void UpdateAMP(String table, int id, String update){
    String sql = null;
    if(null != table)switch (table) {
        case "arsitek":{
            sql = "UPDATE "+table+" set telp_kantor = "+update+" where id_arsitek = "+id;
            break;
        }
        case "marketing":{
            sql = "UPDATE "+table+" set no_telp = "+update+" where id_marketing = "+id;
            break;
        }
        case "profil_aluminium":{
            sql = "UPDATE "+table+" set warna = "+update+" where id_profil = "+id;
            break;
        }
        default:
            break;
    }
    this.koneksi.ManipulasiData(sql);
}
```



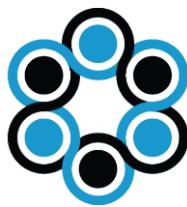
Laporan Sementara

Screenshot

```
public void DeleteAMP(String table, int id){  
    String sql = null;  
    switch (table) {  
        case "arsitek":  
            sql = "DELETE FROM "+table+" WHERE id_arsitek = "+id;  
            break;  
        case "marketing":  
            sql = "DELETE FROM "+table+" WHERE id_marketing = "+id;  
            break;  
        case "profil_aluminium":  
            sql = "DELETE FROM "+table+" WHERE id_profil = "+id;  
            break;  
        default:  
            break;  
    }  
    this.koneksi.ManipulasiData(sql);  
}
```

Analisa :

(Ketik hasil analisa disini)



Laporan Sementara

Langkah Ketiga

(Ketik penjelasan langkah – langkah disini)

Query

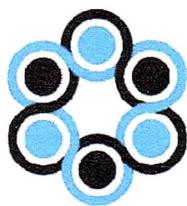
Masukkan query disini (Consolas 12 pt)

Screenshot

Masukkan query disini (Consolas 12 pt)

Analisa :

(Ketik hasil analisa disini)

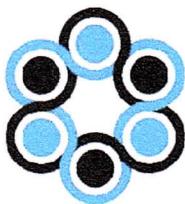


Kritik & Saran

Laboratorium Rekayasa Perangkat Lunak

Kritik : untuk fasilitas lab yang mungkin kurang adalah
PC

Saran : mungkin untuk fasilitas yang kurang banyak
diperbaiki



Kritik & Saran

* March Angga V. W.

- Kritik : menurut Saya tidak ada yang bisa di kritik

- Saran : Pertahankan mas. karena asistensinya mempermudah waktu untuk praktikan malam

* Aldi Wachid Anan

- Kritik : Untuk saat ini saya tidak bisa mengkritik karena tidak ada yang kurang baik mohon diperbaiki

- Saran : jika ada yang kurang baik mohon diperbaiki

* Khusby Al Ghopuri

- Kritik : Untuk saat ini saya tidak bisa mengkritik karena adanya pandemi ini yang menghalangi praktikum langsung.

- Saran : jika ada yang kurang dari anda mohon di perbaiki

* Rizqullah Nadhir Zabirky

- Kritik : Untuk saat ini saya tidak bisa mengkritik karena adanya pandemi ini yang menghalangi praktikum langsung

- Saran : jika ada yang kurang dari anda mohon diperbaiki

* Ura Shanta B.

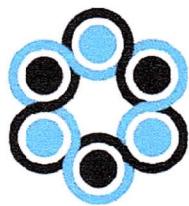
- Kritik : Untuk saat ini saya tidak bisa mengkritik karena adanya pandemi ini yang menghalangi praktikum secara langsung

- Saran : jika ada yang kurang baik dari anda mohon di perbaiki

* Hilmmy Maulana Ilmi

- Kritik : Untuk saat ini saya tidak bisa mengkritik karena adanya pandemi ini yang menghalangi praktikum secara langsung

- Saran : jika ada yang kurang baik dari anda mohon perbaiki



Kritik & Saran

* Taqiyudin Nursy

- Kritik : Untuk saat ini saran tidak bisa menyertakan

karena adanya Pandemi ini yang menghalangi
pelatihan secara langsung

- Saran : jika ada yang kurang baik dari anda mohon
di perbaiki