

MODUL BASIS DATA – DASAR BASIS DATA 3**A. MANIPULASI DAN RETRIEVE DATA LANJUTAN (DML)****1) UNION**

UNION merupakan operator atau perintah yang digunakan untuk *menggabungkan* hasil query atau isi data dari 2 (dua) tabel atau lebih dengan ketentuan jumlah, nama dan tipe data atau kolom dari masing – masing tabel yang akan ditampilkan datanya harus sama.

Perintah ini terdiri dari dua jenis yaitu UNION dan UNION ALL. Untuk menghasilkan suatu data set perintah UNION harus disisipkan diantara perintah SELECT. Untuk lebih jelasnya kita ambil contoh sederhana dari sekumpulan data sebagai berikut :

❖ Contoh UNION:

```
select iddiv from divisi union select iddiv from pegawai;
```

```
MariaDB [pegw]> select iddiv from divisi union select iddiv from pegawai;
+-----+
| iddiv |
+-----+
| HUM   |
| KEU   |
| PEG   |
+-----+
3 rows in set (0.25 sec)
```

❖ Contoh UNION ALL:

Berbeda dengan perintah UNION, perintah UNION ALL menggabungkan keseluruhan data walaupun data tersebut memiliki isi yang sama.

```
select iddiv from divisi union all select iddiv from pegawai;
```

```
MariaDB [pegw]> select iddiv from divisi union all select iddiv from pegawai;
+-----+
| iddiv |
+-----+
| HUM   |
| KEU   |
| PEG   |
| KEU   |
| KEU   |
| HUM   |
| PEG   |
+-----+
7 rows in set (0.00 sec)
```

```

MariaDB [pegw]> select npp,total_gaji * 1.5 AS "Salary up" from gaji;
+-----+-----+
| npp | Salary up |
+-----+-----+
| 1101 | 2250000.0 |
| 1102 | 2250000.0 |
| 1103 | 1950000.0 |
| 1105 | 2100000.0 |
+-----+-----+
4 rows in set (0.06 sec)

MariaDB [pegw]> select * from gaji;
+-----+-----+
| npp | total_gaji |
+-----+-----+
| 1101 | 1500000 |
| 1102 | 1500000 |
| 1103 | 1300000 |
| 1105 | 1400000 |
+-----+-----+
4 rows in set (0.00 sec)

```

2) GROUP BY

GROUP BY digunakan untuk menyeleksi himpunan yang dihitung berdasarkan fungsi spesifik atau digunakan untuk melakukan pengelompokan data. Fungsi spesifik yang dapat digunakan antara lain AVG, COUNT dan lainnya.

```
select iddiv, count(*) as jumlah from pegawai group by iddiv;
```

```

+-----+-----+-----+-----+-----+
| npp | nama_peg | alamat | iddiv | asal_kota |
+-----+-----+-----+-----+-----+
| 1101 | Alana | Jl. Bunga Matahari | KEU | Kendari |
| 1102 | Andini | Jl. Sao-sao | PEG | Bandung |
| 1104 | Aiman | Wua-wua | KEU | NULL |
+-----+-----+-----+-----+-----+
3 rows in set (0.08 sec)

MariaDB [pegw]> select iddiv, COUNT(*) AS Jumlah from pegawai group by iddiv;
+-----+-----+
| iddiv | Jumlah |
+-----+-----+
| KEU | 2 |
| PEG | 1 |
+-----+-----+
2 rows in set (0.00 sec)

```

3) GROUP BY....HAVING

HAVING berlaku untuk kelompok query group dan berfungsi seperti WHERE. Klausula HAVING digunakan untuk menentukan kondisi bagi klausula GROUP BY. Hanya kelompok atau group yang memenuhi kriteria HAVING saja yang akan diproses atau dihasilkan.

Contoh: perintah untuk menampilkan data hanya kolom iddiv yang dikelompokkan berdasarkan kolom iddiv, dimana jumlah divisi berdasarkan kelompoknya harus lebih besar dari satu pada tabel pegawai.

```
SELECT iddiv, COUNT(*) AS Jumlah FROM pegawai GROUP BY iddiv
HAVING COUNT(iddiv) >1;
```

```

MariaDB [pegw]> SELECT iddiv, COUNT(*) AS Jumlah FROM pegawai GROUP
-> BY iddiv HAVING COUNT(iddiv) >1;
+-----+-----+
| iddiv | Jumlah |
+-----+-----+
| KEU | 2 |
+-----+-----+
1 row in set (0.00 sec)

```

Berikut ini adalah hal-hal yang perlu diperhatikan saat menggunakan klause Where, Group By, dan Having:

- ❖ Where digunakan untuk menampilkan data yang difilter /baris – baris dari operasi yang dinyatakan oleh perintah From.
- ❖ Group By digunakan untuk mengelompokkan hasil dari klausa Where
- ❖ Having digunakan untuk memfilter baris – baris dari hasil pengelompokkan.

4) JOIN

Join merupakan operasi yang digunakan untuk menggabungkan dua tabel atau lebih dengan hasil merupakan gabungan dari kolom – kolom yang berasal dari tabel – tabel tersebut.

a) Inner Join

Digunakan untuk menampilkan gabungan data dari dua arah/dua tabel yang berisi data sesuai dengan syarat dibelakang on (tidak bernilai null), artinya semua data tabel di sebelah kiri mendapatkan pasangan data dari tabel sebelah kanan. Contoh : kita akan menggabungkan tabel pegawai dan tabel divisi dimana kita akan menampilkan daftar pegawai dan keterangan divisinya.

➤ Cara 1:

```
SELECT p.npp, p.nama_peg, d.iddiv, d.nama FROM pegawai p,
divisi d WHERE p.iddiv=d.iddiv
```

koding lainnya:

```
SELECT pegawai.npp, pegawai.nama_peg, divisi.iddiv, divisi.nama
FROM pegawai, divisi WHERE pegawai.iddiv=divisi.iddiv
```

```
MariaDB [kepeg]> SELECT p.npp, p.nama_peg,
-> d.iddiv, d.nama FROM pegawai p,
-> divisi d WHERE p.iddiv=d.iddiv
-> ;
+-----+-----+-----+-----+
| npp   | nama_peg | iddiv | nama      |
+-----+-----+-----+-----+
| 1101  | Lisa     | KEU   | Keuangan  |
| 1102  | Dimas    | PEG   | Kepegawaia |
| 1103  | Dian     | HUM   | Humas     |
| 1104  | Ahmad    | KEU   | Keuangan  |
| 1105  | Kamil    | PEG   | Kepegawaia |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

➤ Cara 2:

```
SELECT p.npp, p.nama_peg, d.iddiv, d.nama FROM pegawai p INNER
JOIN divisi d ON p.iddiv=d.iddiv
```

```

+-----+-----+-----+-----+
| npp   | nama_peg | iddiv  | nama   |
+-----+-----+-----+-----+
| 1101  | Lisa    | KEU    | Keuangan |
| 1102  | Dimas   | PEG    | Kepegawaia |
| 1103  | Dian    | HUM    | Humas   |
| 1104  | Ahmad   | KEU    | Keuangan |
| 1105  | Kamil   | PEG    | Kepegawaia |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

b) Outer Join

Digunakan untuk menggabungkan satu arah, sehingga memungkinkan ada data yang Null (kosong) di satu sisi. Contoh kita akan menggabungkan tabel pegawai dan divisi dimana kita akan menampilkan daftar pegawai beserta keterangan divisinya. Outer join terbagi menjadi dua yaitu Left Join dan Right Join.

```

SELECT p.npp, p.nama_peg, d.iddiv, d.nama FROM pegawai p
LEFT JOIN divisi d ON p.iddiv=d.iddiv;

```

```

MariaDB [kepeg]> SELECT p.npp, p.nama_peg,
-> d.iddiv, d.nama FROM pegawai p LEFT JOIN divisi d ON
-> p.iddiv=d.iddiv ;
+-----+-----+-----+-----+
| npp   | nama_peg | iddiv  | nama   |
+-----+-----+-----+-----+
| 1101  | Lisa    | KEU    | Keuangan |
| 1102  | Dimas   | PEG    | Kepegawaia |
| 1103  | Dian    | HUM    | Humas   |
| 1104  | Ahmad   | KEU    | Keuangan |
| 1105  | Kamil   | PEG    | Kepegawaia |
| 1106  | Mia     | NULL   | NULL    |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

```

Dengan right join, tabel yang menjadi acuan adalah tabel sebelah kanan, jadi semua isi tabel pesan akan ditampilkan. Jika data dosen tidak ada di tabel mengajar, maka isi tabel mengajar tetap ditampilkan.

c) Join 3 Tabel

Untuk menggabungkan tiga atau lebih tabel, secara prinsip sama dengan penggabungan dengan dua buah tabel. Sebagai contoh misalnya kita akan menampilkan data pegawai beserta divisi dan gajinya.

```

SELECT p.npp, p.nama_peg,
d.iddiv, d.nama, g.total_gaji FROM pegawai p, divisi d, gaji g
WHERE p.iddiv=d.iddiv AND p.npp=g.npp

```

```

MariaDB [kepeg]> SELECT p.npp, p.nama_peg,
-> d.iddiv, d.nama, g.total_gaji FROM pegawai p, divisi d, gaji g
-> WHERE p.iddiv=d.iddiv AND p.npp=g.npp
-> ;
+-----+-----+-----+-----+-----+
| npp   | nama_peg | iddiv  | nama   | total_gaji |
+-----+-----+-----+-----+-----+
| 1103  | Dian    | HUM    | Humas   | 1200000 |
| 1101  | Lisa    | KEU    | Keuangan | 1500000 |
| 1104  | Ahmad   | KEU    | Keuangan | 1500000 |
| 1102  | Dimas   | PEG    | Kepegawaia | 1300000 |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

```

```

MariaDB [kepeg]> select * from gaji;
+-----+-----+
| npp | total_gaji |
+-----+-----+
| 1101 | 1500000 |
| 1102 | 1300000 |
| 1103 | 1200000 |
| 1104 | 1500000 |
+-----+-----+
4 rows in set (0.00 sec)

MariaDB [kepeg]> select * from pegawai;
+-----+-----+-----+-----+-----+
| npp | nama_peg | alamat | iddiv | asal_kota |
+-----+-----+-----+-----+-----+
| 1101 | Lisa | Jl. Bunga Matahari | KEU | Kendari |
| 1102 | Dimas | Jl. Lumba-lumba | PEG | Bandung |
| 1103 | Dian | Wua-wua | HUM | Kendari |
| 1104 | Ahmad | Kampus baru | KEU | Surabaya |
| 1105 | Kamil | Lepo-lepo | PEG | Aceh |
| 1106 | Mia | Mandonga | | Aceh |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

```

B. DATABASE RELASIONAL

Pada database relasional, data disimpan dalam bentuk relasi atau tabel dua dimensi, serta antar tabel yang satu dengan tabel lainnya terdapat hubungan atau relationship. Kemudian kumpulan dari data yang diorganisasikan sebagai tabel disimpan dalam bentuk data elektronik di dalam media penyimpanan komputer.

Primary key adalah suatu field atau kombinasi field yang secara unik mengidentifikasi setiap record dalam tabel. Foreign key adalah primary key pada suatu tabel yang dimasukkan pada tabel lain dan dijadikan salah satu key pada tabel tersebut. References menunjukkan bahwa foreign key pada suatu tabel merupakan key pada tabel lain. Pada perubahan suatu record baik update maupun delete, dapat diikuti dengan kata : cascade, restrict, atau set null sesuai dengan kebutuhan.

- * **Cascade** = apabila id pada table A dihapus maka table B juga ikut terhapus
- * **SET NULL** = apabila id pada table A dihapus maka table B di set null
- * **No action** = apabila id pada table A dihapus maka tidak dilakukan apa-apa
- * **Restrict** = apabila id pada table A ingin dihapus maka tidak diperbolehkan jika di table B ditemukan ID yang berelasi

Contoh :

-on update CASCADE on delete RESTRICT

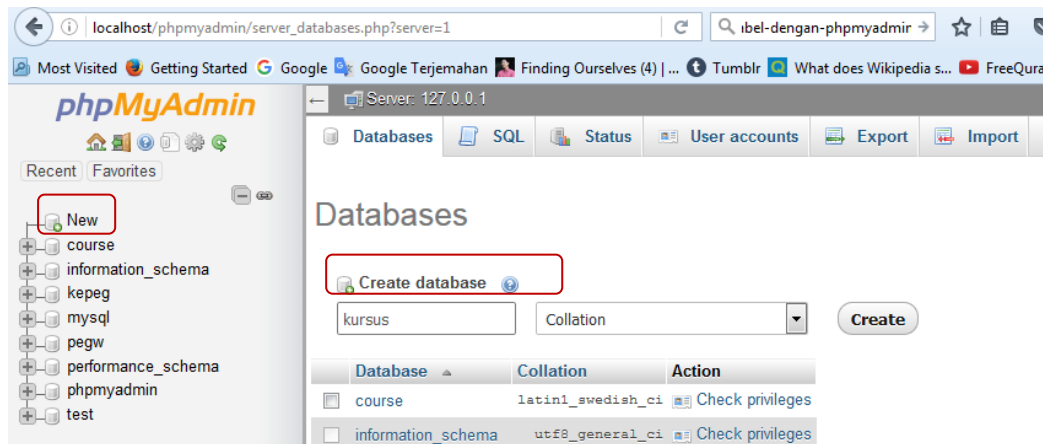
-on update RESTRICT on delete SET NULL

Alur:

- Buat sebuah database dengan nama '**kursus**'

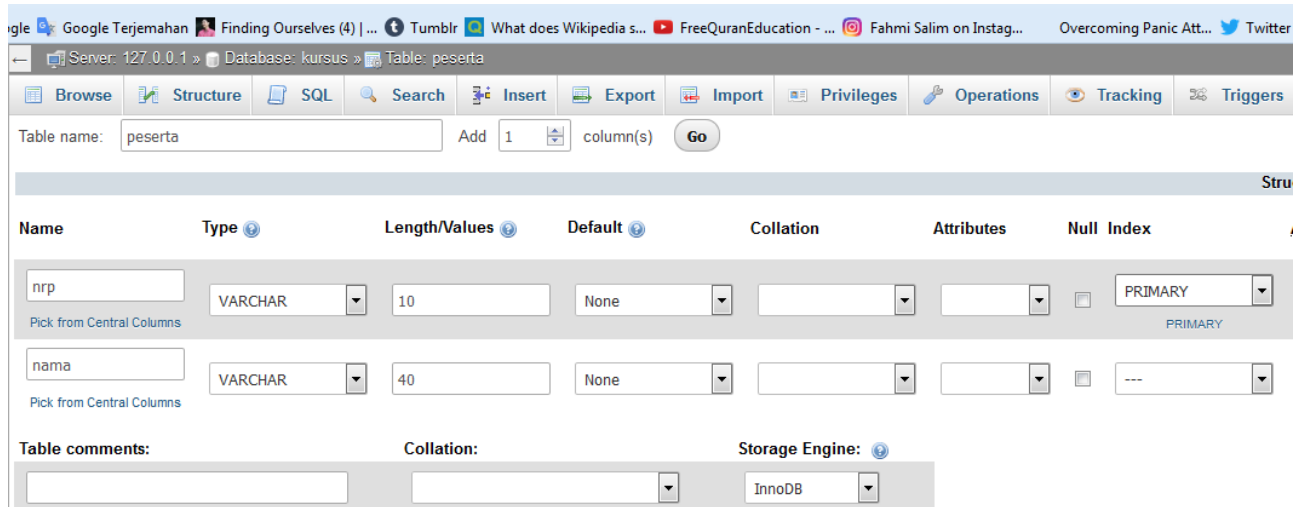
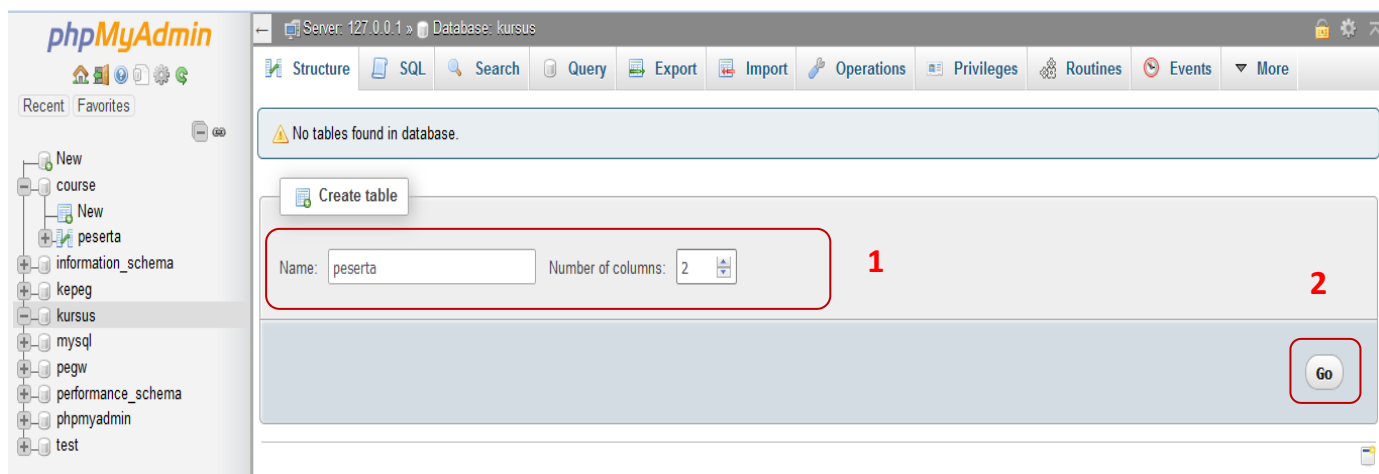
```
Create database kursus;
```

Pada browser, ketikkan localhost/phpmyadmin. Lalu akan muncul tampilan phpmyadmin. Klik menu New, pada tampilan, lihat menu create database, isikan nama database (isikan kursus) lalu klik tombol create.



➤ Buat tabel 'peserta', tipenya InnoDB

```
Create table peserta (
  nrp varchar(10) not null,
  nama varchar(40),
  primary key(nrp));
```



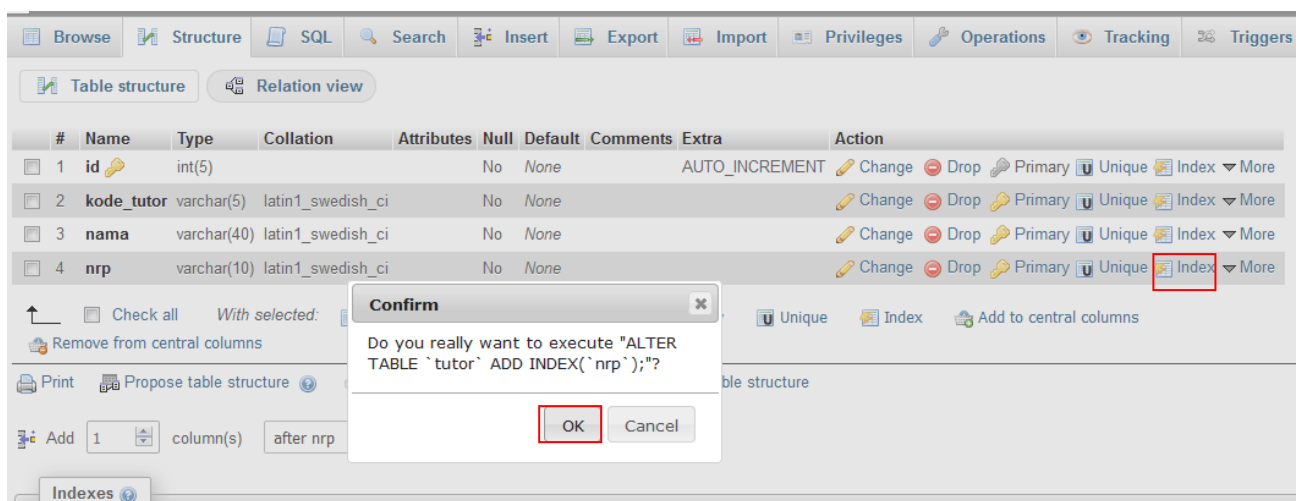
- Buat tabel **'tutor'**, tipenya InnoDB dan relasikan ke tabel mahasiswa dengan mengambil key NRP dari tabel peserta_kursus sebagai foreign key di tabel tutor

```
create table tutor (id int(5) not null auto_increment,
kode_tutor varchar(5),
nama varchar(40),
nrp varchar(10),
primary key(id),
index(nrp),
foreign key(nrp) references peserta(nrp) on update
cascade on delete cascade)
engine innodb;
```

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index
id	INT	5	None			<input type="checkbox"/>	PRIMARY <input type="checkbox"/>
kode_tutor	VARCHAR	5	None			<input type="checkbox"/>	--- <input type="checkbox"/>
nama	VARCHAR	40	None			<input type="checkbox"/>	--- <input type="checkbox"/>
nrp	VARCHAR	10	None			<input type="checkbox"/>	--- <input type="checkbox"/>

Table comments: Collation: Storage Engine:

Sebelum merelasikan antara tabel peserta dan tutor, terlebih dahulu jadikan **"nrp"** yang ada pada tabel **"peserta"** sebagai **"INDEX"** seperti gambar dibawah ini.



Hasilnya:

MySQL returned an empty result set (i.e. zero rows). (Query took 0.1248 seconds.)

```
ALTER TABLE `tutor` ADD INDEX(`nrp`);
```

[Edit inline] [Edit] [Create PHP code]

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(5)			No	None		AUTO_INCREMENT	Change Drop Primary Unique Index More
2	kode_tutor	varchar(5)	latin1_swedish_ci		No	None			Change Drop Primary Unique Index More
3	nama	varchar(40)	latin1_swedish_ci		No	None			Change Drop Primary Unique Index More
4	nrp	varchar(10)	latin1_swedish_ci		No	None			Change Drop Primary Unique Index More

Setelah itu kembali ke database klik **More > Designer**

Server: 127.0.0.1 » Database: kursus

Structure SQL Search Query Export Import Operations Privileges Routines Events More

Filters

Containing the word:

Table	Action	Rows	Type	Collation	Size	Overhead
peserta	Star Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	16 KiB	-
tutor	Star Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	16 KiB	-
2 tables	Sum	0	InnoDB	latin1_swedish_ci	32 KiB	0 B

Triggers Tracking **Designer** Central columns

Klik **Create relation**

phpMyAdmin

Recent Favorites

Server: 127.0.0.1 » Database: kursus

Structure SQL Search Query Export Import

nrp pada peserta sebagai referenced key

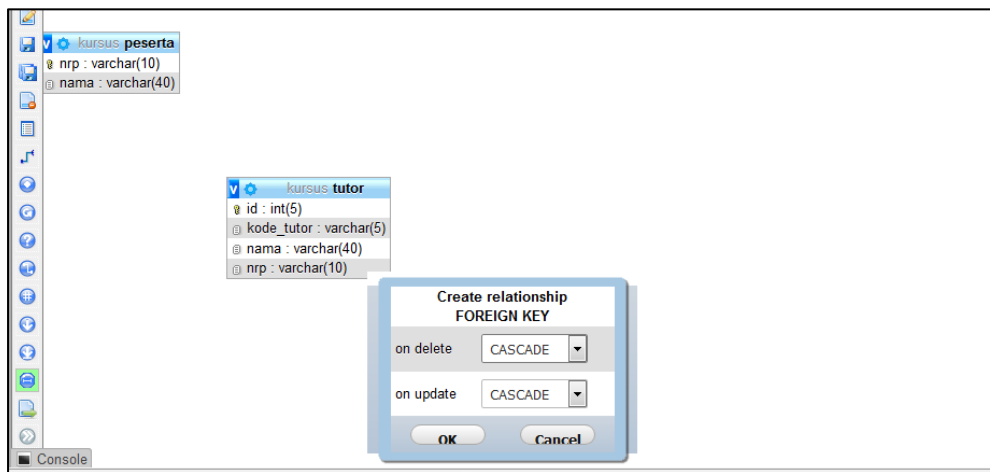
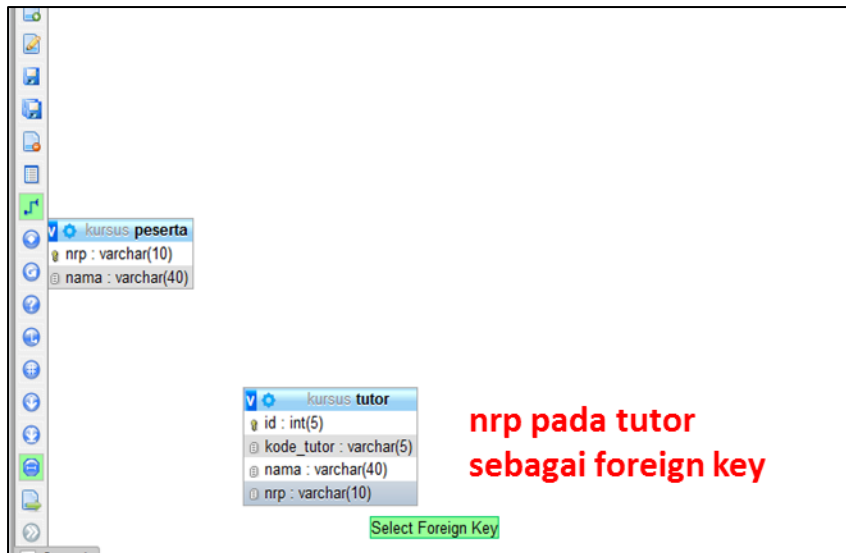
Select referenced key

kursus peserta

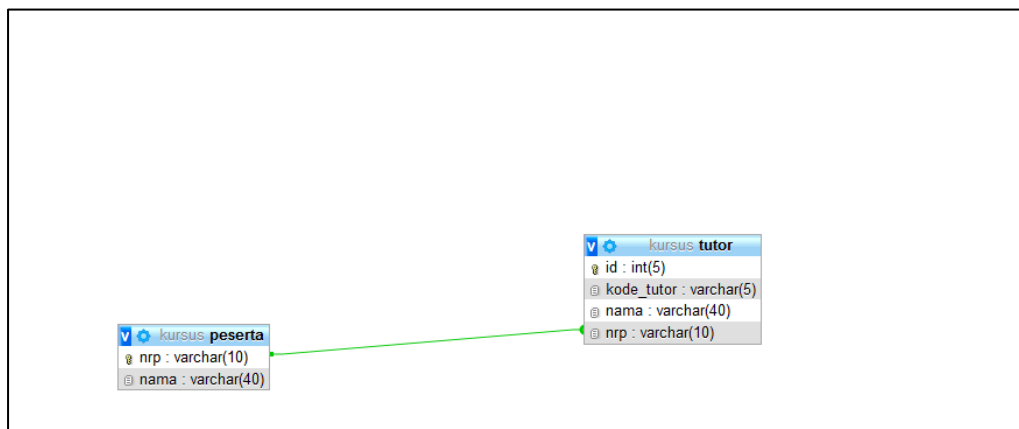
- nrp : varchar(10)
- nama : varchar(40)

kursus tutor

- id : int(5)
- kode_tutor : varchar(5)
- nama : varchar(40)
- nrp : varchar(10)



Hasil relasi:



➤ Isikan data sebagai berikut

nrp	nama	id	kode_tutor	nama	nrp
001	Anton Wahyu Nugroho	1	A001	Beni S	001
002	Budi Surya	2	A002	Wahyu	002
003	Cici Paramida	3	A001	Beni S	003
004	Denny Dargo	4	A002	Wahyu	004

dan

Untuk tabel peserta

Browse
Structure
SQL
Search
Insert
Export
Import
Priv

Column	Type	Function	Null	Value
nrp	varchar(10)			001
nama	varchar(40)			Anton

Go

☐ Ignore

Column	Type	Function	Null	Value
nrp	varchar(10)			002
nama	varchar(40)			Budi

Go

Insert as new row
and then
Go back to previous page

Go
Preview SQL
Reset

+ Options

nrp
nama

☐ Edit
☐ Copy
☐ Delete
001
Anton

☐ Edit
☐ Copy
☐ Delete
002
Budi

Untuk tabel tutor

Browse
Structure
SQL
Search
Insert
Export
Import
P

Column	Type	Function	Null	Value
id	int(5)			1
kode_tutor	varchar(5)			A001
nama	varchar(40)			Beni
nrp	varchar(10)			001 - 001

Go

☐ Ignore

Column	Type	Function	Null	Value
id	int(5)			2
kode_tutor	varchar(5)			A002
nama	varchar(40)			Wahyu
nrp	varchar(10)			002 - 002

Go

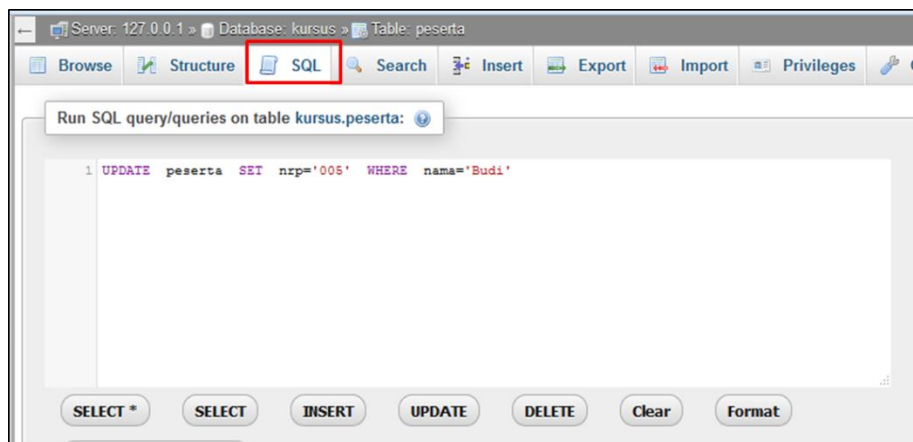
+ Options
← T →

	id	kode_tutor	nama	nrp
<input type="checkbox"/> Edit Copy Delete	1	A001	Beni	001
<input type="checkbox"/> Edit Copy Delete	2	A002	Wahyu	002

☐ Check all
 With selected:
 Edit
 Copy
 Delete
 Export










- Update satu record pada tabel peserta:

```
UPDATE peserta SET nrp='005' WHERE nama='Budi'
```



- Lihat perubahannya







+ Options				
← T →				
			nrp	nama
<input type="checkbox"/>	Edit	Copy	Delete	001 Anton
<input type="checkbox"/>	Edit	Copy	Delete	003 Cici
<input type="checkbox"/>	Edit	Copy	Delete	005 Budi

+ Options							
← T →				id	kode_tutor	nama	nrp
<input type="checkbox"/>	 Edit	 Copy	 Delete	1	A001	Beni	001
<input type="checkbox"/>	 Edit	 Copy	 Delete	2	A002	Wahyu	005
<input type="checkbox"/>	 Edit	 Copy	 Delete	3	A001	Beni	003

- Hapus sebuah record pada tabel peserta :

```
delete from peserta where nrp='005'
```

- Lihat perubahan pada tabel tutor

+ Options							
← T →				id	kode_tutor	nama	nrp
<input type="checkbox"/>	 Edit	 Copy	 Delete	1	A001	Beni	001
<input type="checkbox"/>	 Edit	 Copy	 Delete	3	A001	Beni	003

Daftar Pustaka

MODUL PRAKTIKUM BASIS DATA STT TELKOM, 2014, Didi Supriyadi, ST., M.Kom