



MODUL WEB V1

Modul Praktikum Pemrograman Web PHP Versi 1
Basic Manipulasi Database dan CRUD PHP

ABSTRACT

Modul ini merupakan modul pelaksanaan praktikum web V1 (tingkat dasar). Berisi cara memanipulasi database dan membuat Create Read Update Delete (CRUD) dengan PHP dan MySQLi.

Fathul Hafidh

Pengajar Praktikum Web pada Fakultas Teknologi Informasi Universitas Islam Kalimantan Muhammad Arsyad Al Banjari Banjarmasin

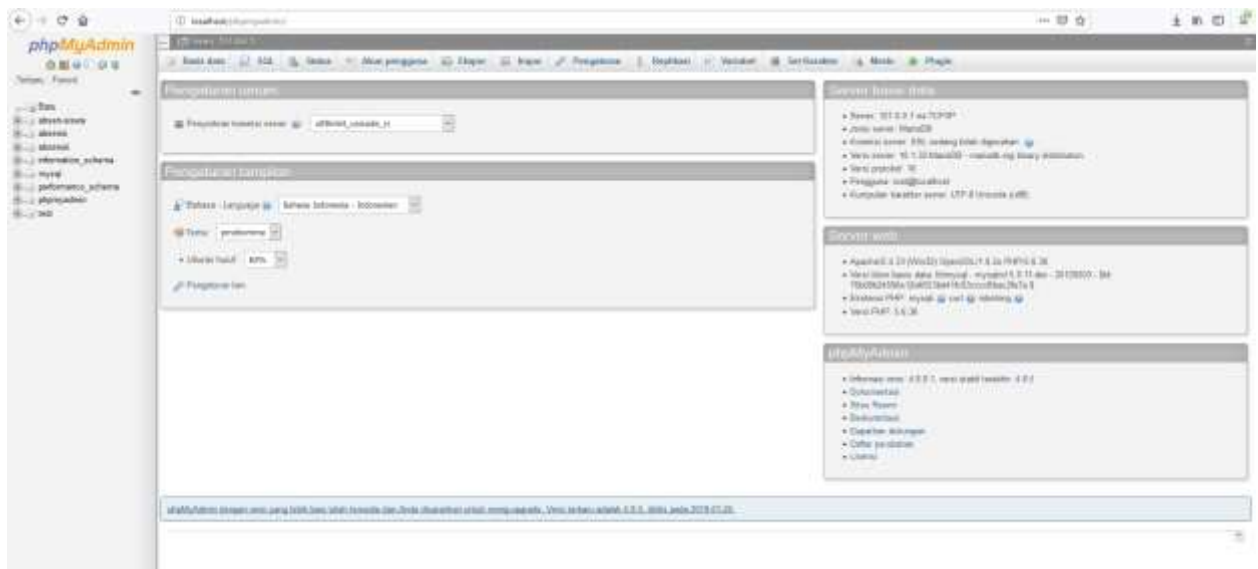
DAFTAR ISI

DAFTAR ISI	1
Manipulasi Database	2
1. Membuat dan Menghapus Database	2
A. Membuat Database	2
B. Menghapus Database	5
2. Membuat Table dan Tipe Data	7
3. Insert to Table	11
4. Update Data Table	15
5. Delete Data Table	19
6. Select dan Select Filter	23
A. SELECT	23
B. SELECT FILTER	25
CRUD PHP	30
1. Mengetahui MySQLi	30
2. Mengkoneksikan PHP dengan MySQL	32
3. Menampilkan Data Karyawan	35
4. Membuat Form Insert	38
5. Insert Data	41
6. Update Data	43
7. Delete Data	49
8. Search Data	52

MANIPULASI DATABASE

1. Membuat dan Menghapus Database

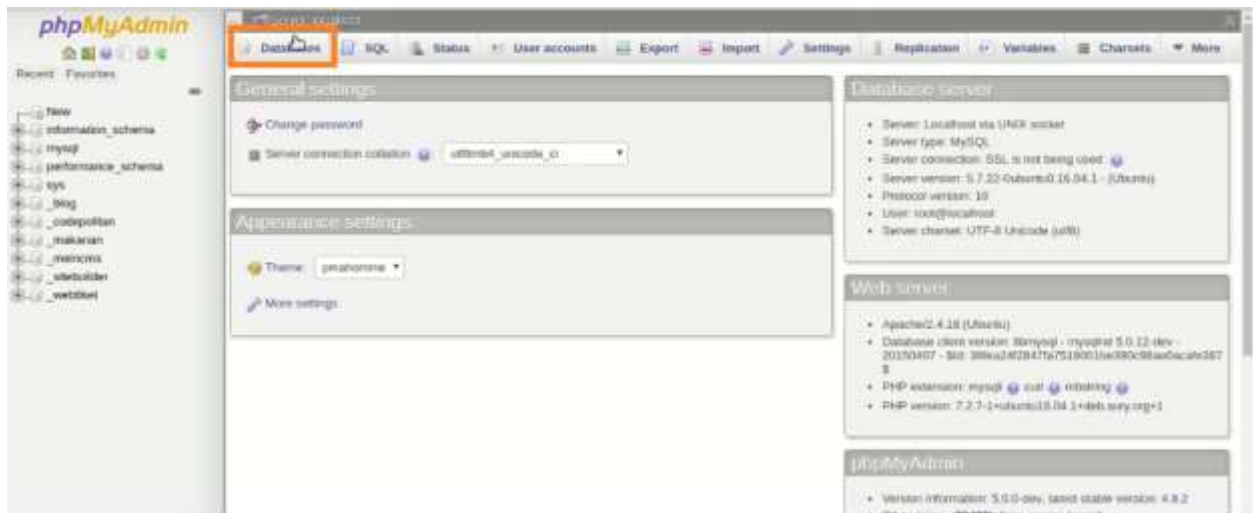
phpMyAdmin adalah salah satu aplikasi berbasis web yang berfungsi *manage* database, database yang di *support* adalah mysql dan mariadb. Ketika kita sudah menginstal xampp berarti kita sudah dapat mengakses phpMyAdmin, langsung saja buka localhost/phpmyadmin/ maka akan tampil seperti gambar dibawah ini.



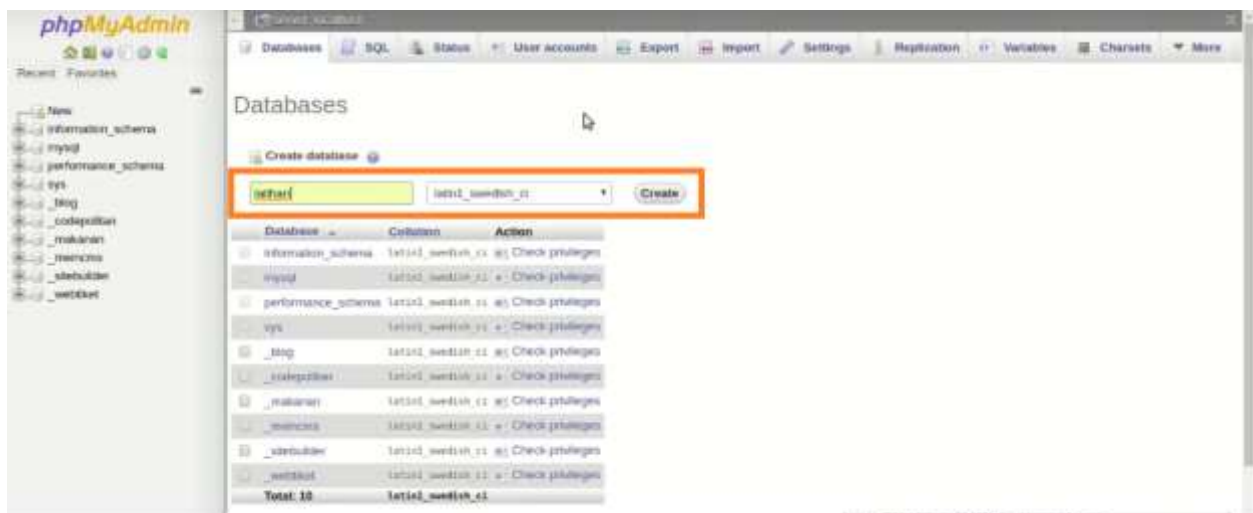
A. Membuat Database

Bagaimana cara membuat database di **phpMyAdmin**?

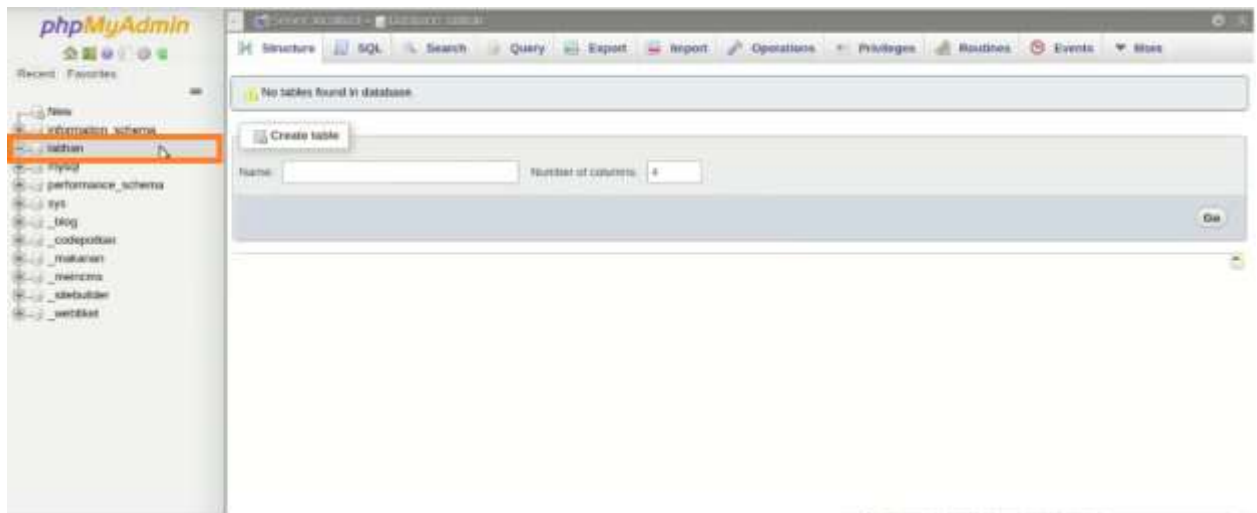
langsung saja klik tombol databases seperti dibawah ini.



Setelah itu masukan nama database sesuai kebutuhan, pada contoh kali ini kita beri nama *latihan*, kemudian klik tombol create.



Nah sekarang jika sudah berhasil, maka akan muncul database nama database yang kita buat di sebelah kiri seperti gambar dibawah ini :

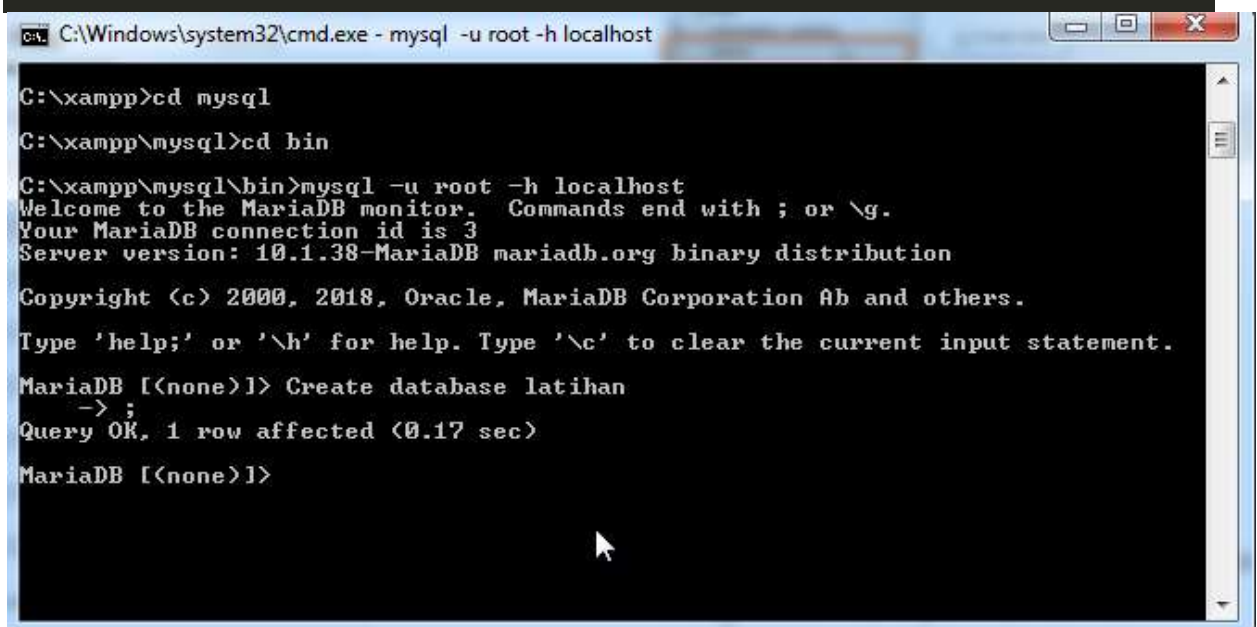


Selain membuat dengan phpmyadmin, kita juga bisa membuat database melalui console dengan mengetikkan baris perintah berikut ini.

```
CREATE database namaDatabase;
```

Contoh

```
CREATE database latihan;
```



Untuk membuka database latihan, kita bisa mengetikkan baris perintah berikut ini.

Use latihan;

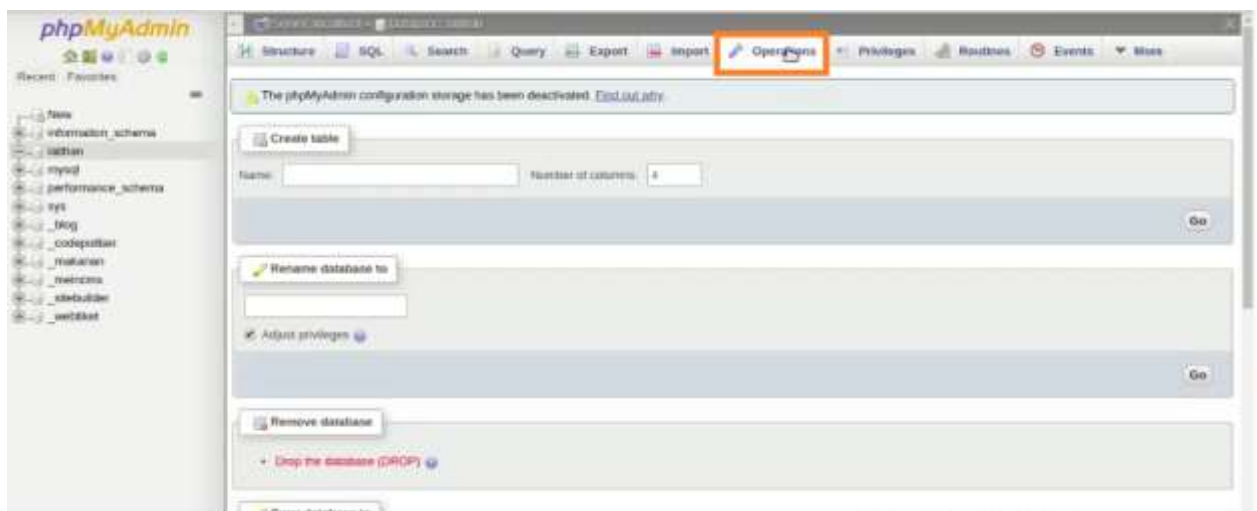
```
C:\Windows\system32\cmd.exe - mysql -u root -h localhost

C:\xampp>cd mysql
C:\xampp\mysql>cd bin
C:\xampp\mysql\bin>mysql -u root -h localhost
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 3
Server version: 10.1.38-MariaDB mariadb.org binary distribution
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
MariaDB [(none)]> Create database latihan
-> ;
Query OK, 1 row affected (0.17 sec)

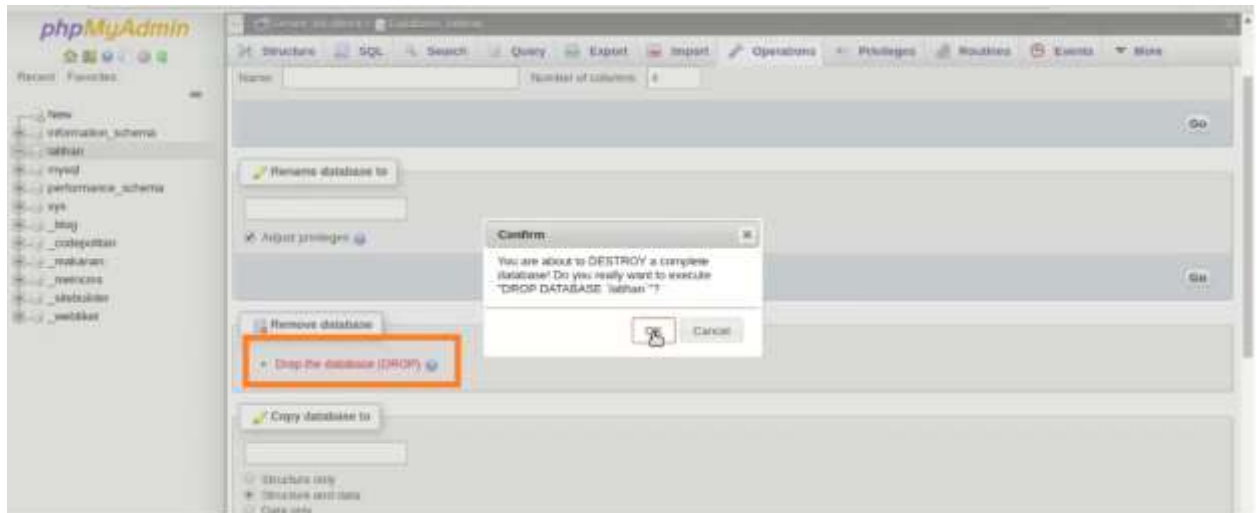
MariaDB [(none)]> use latihan;
Database changed
MariaDB [latihan]>
```

B. Menghapus Database

Untuk menghapus database di **phpMyAdmin** dapat kita lakukan dengan mudah, klik tombol **Operations** pada menu bar di bagian atas,



Setelah itu muncul seperti gambar dibawah ini, kemudian klik tombol **Drop the database (DROP)** kemudian akan muncul *popup* konfirmasi, kemudian pilih oke jika memang sudah yakin.

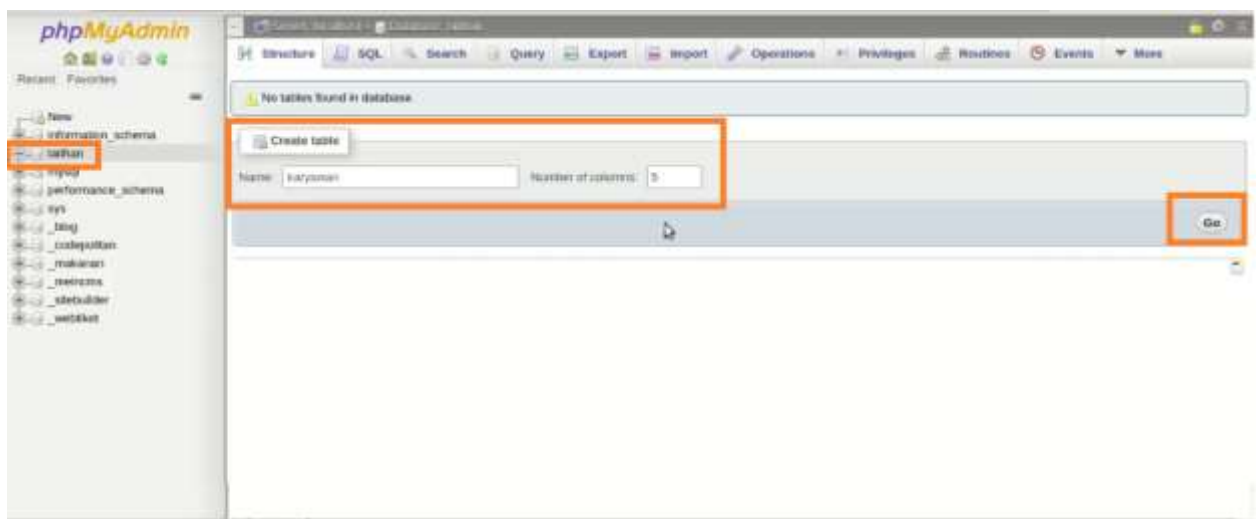


Apabila ingin menghapus database melalui console, ketikkan baris perintah berikut ini.

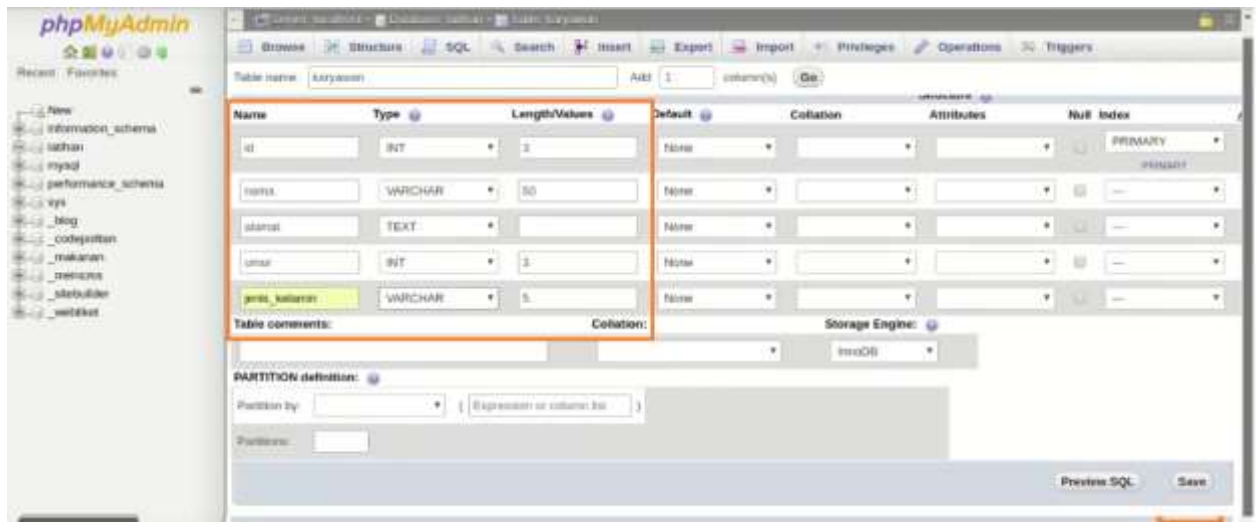
```
Drop database namaDatabase;
```

2. Membuat Table dan Tipe Data

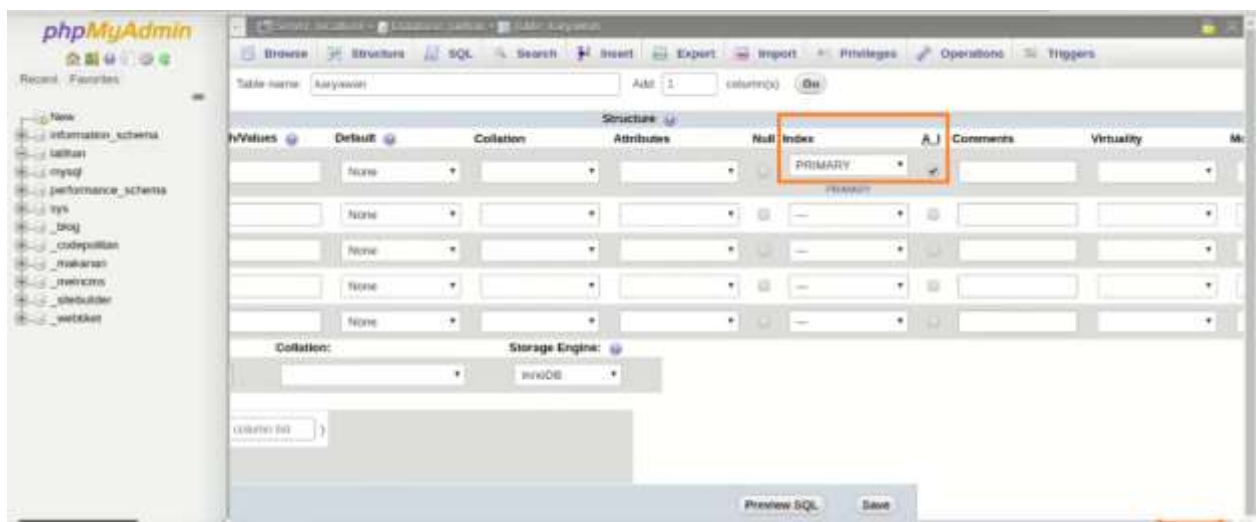
Untuk membuat tabel dan tipe data pada **phpMyAdmin** klik pada database *latihan* yang sudah kita buat sebelumnya. Karena pada tutorial ini kita ingin membuat manajemen data-data karyawan, maka nama yang relevan untuk tabelnya adalah *karyawan* dengan jumlah *field* 5, setelah itu klik tombol **Go**.



Kita sudah membuat tabel dengan nama *karyawan* dengan jumlah *field* 5, dimana *field-field* tersebut akan kita isi informasi terkait karyawan, yang berisi atribut seperti dibawah ini :

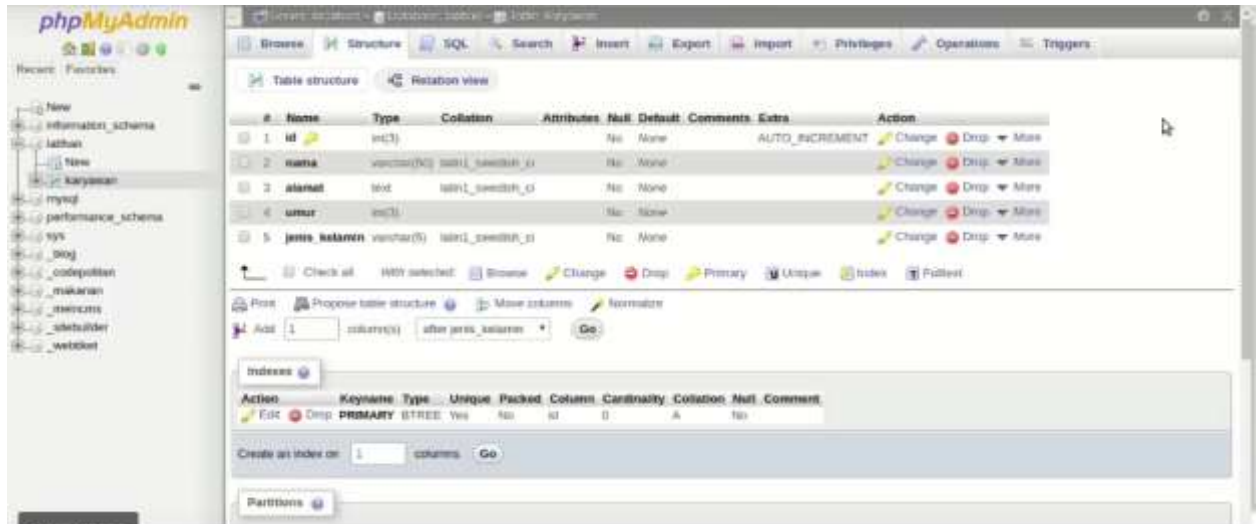


Jika kita lihat diatas terdapat **id** dimana **id** ini kita jadikan sebagai *field* unik atau perwakilan dari semua *record* yang ada maka kita perlu jadikan **id** sebagai *primary key* seperti dibawah ini.



Selain itu, jika kita lihat gambar diatas kita juga tetapkan **id** nya menjadi AI (*Auto Increment*) sehingga ketika menambah data baru kita tidak perlu menambahkan **id** nya secara manual tapi otomatis oleh sistem.

Setelah kita pastikan data yang diinput sudah benar semua, maka sekarang tekan tombol **Save** yang berada pada pojok kanan bawah, dan jika berhasil akan tampil tabel seperti dibawah ini :



Selain menggunakan fitur **phpmyadmin**, kita juga bisa membuat database melalui **console** dengan mengetikkan baris perintah berikut ini.

```
Create table [namatabel]
```

```
(
```

```
    field1 datatype(length),
```

```
    field2 datatype(length),
```

```
    field3 datatype(length),
```

```
    PRIMARY KEY(field_key)
```

```
);
```

Untuk beberapa tipe data tidak wajib menyebutkan panjang karakter (**length**) antara lain **int** dan **text**.

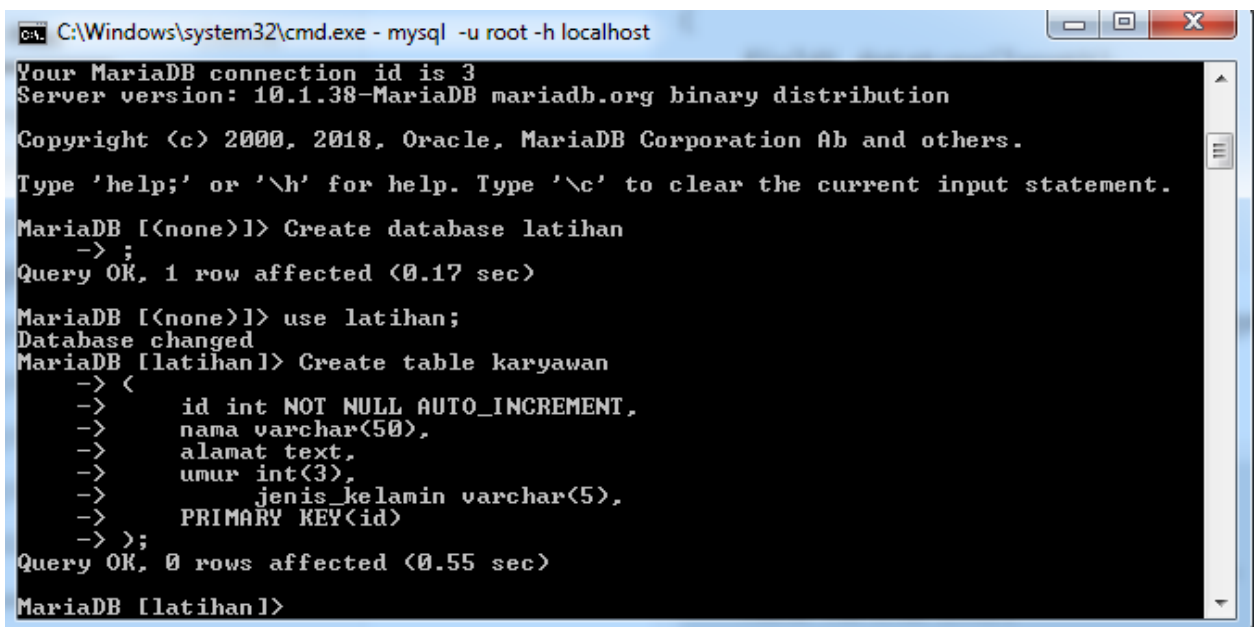
Contoh

```
Create table karyawan
```

```
(  
  
    id int NOT NULL AUTO_INCREMENT,  
  
    nama varchar(50),  
  
    alamat text,  
  
    umur int(3),  
  
    jenis_kelamin varchar(5),  
  
    PRIMARY KEY(id)  
  
);
```

Kamu bisa sesuaikan nama field, tipe data dan juga panjang karakter sesuai kebutuhan ya.

Sebelum menjalankan baris perintah di atas, pastikan kita sudah membuka database yang dituju dengan mengetikkan `use namaDatabase`.

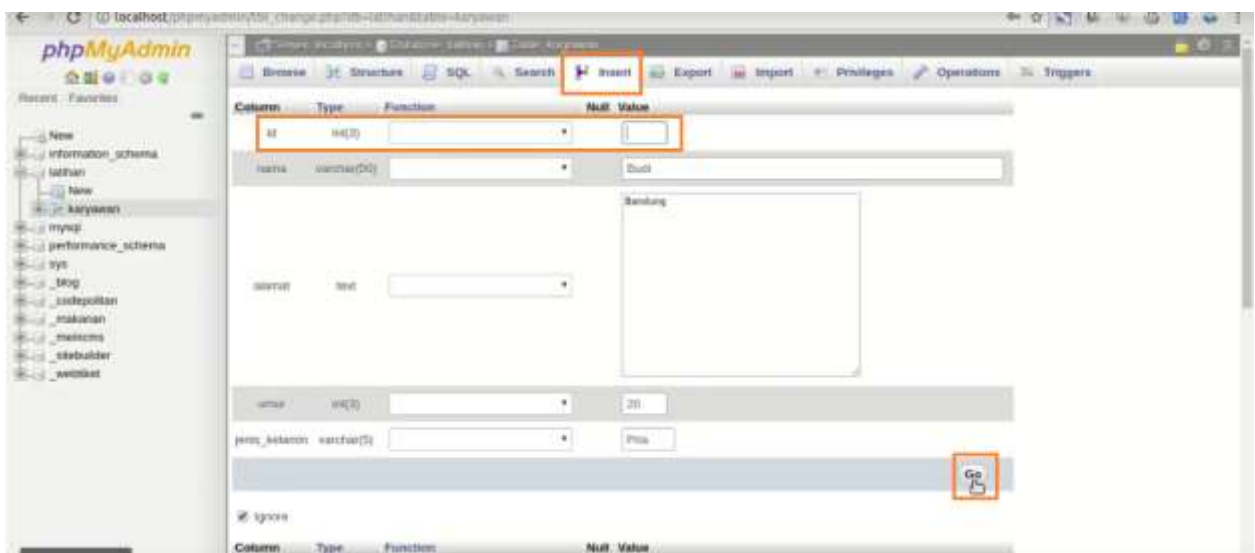


```
C:\Windows\system32\cmd.exe - mysql -u root -h localhost  
Your MariaDB connection id is 3  
Server version: 10.1.38-MariaDB mariadb.org binary distribution  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
MariaDB [(none)]> Create database latihan  
-> ;  
Query OK, 1 row affected (0.17 sec)  
MariaDB [(none)]> use latihan;  
Database changed  
MariaDB [latihan]> Create table karyawan  
-> <  
->     id int NOT NULL AUTO_INCREMENT,  
->     nama varchar(50),  
->     alamat text,  
->     umur int(3),  
->     jenis_kelamin varchar(5),  
->     PRIMARY KEY(id)  
-> >;  
Query OK, 0 rows affected (0.55 sec)  
MariaDB [latihan]>
```

3. Insert to Table

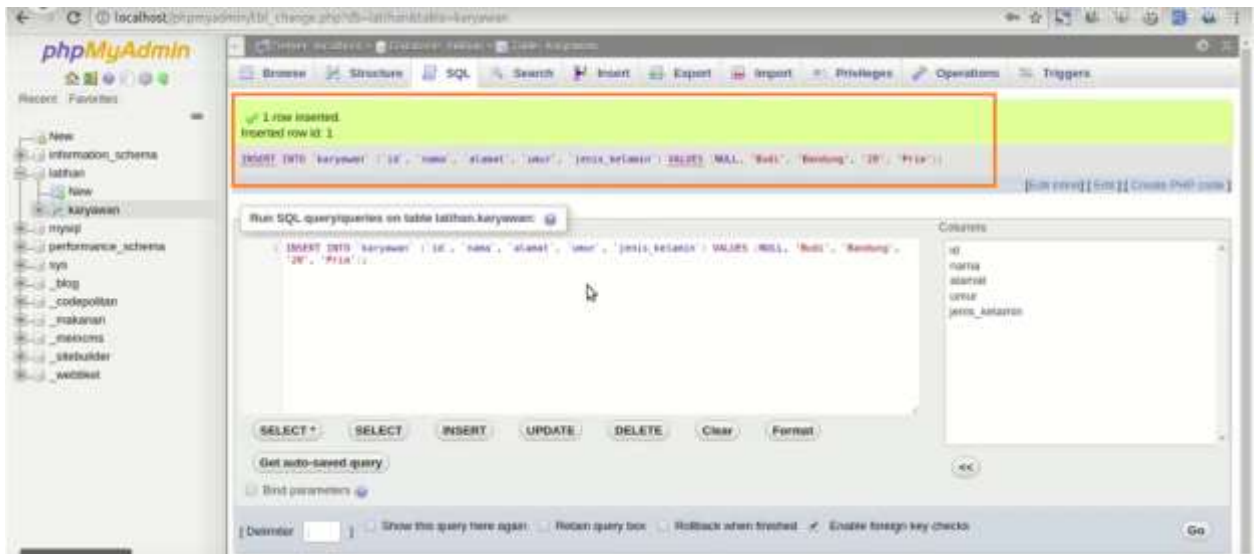
Setelah pada episode sebelumnya kita sudah membuat tabel karyawan, Selanjutnya kita akan insert/menambahkan data pada table yang sudah dibuat sebelumnya.

Sebenarnya di phpmyadmin sudah ada menu untuk menambahkan data seperti gambar dibawah ini :

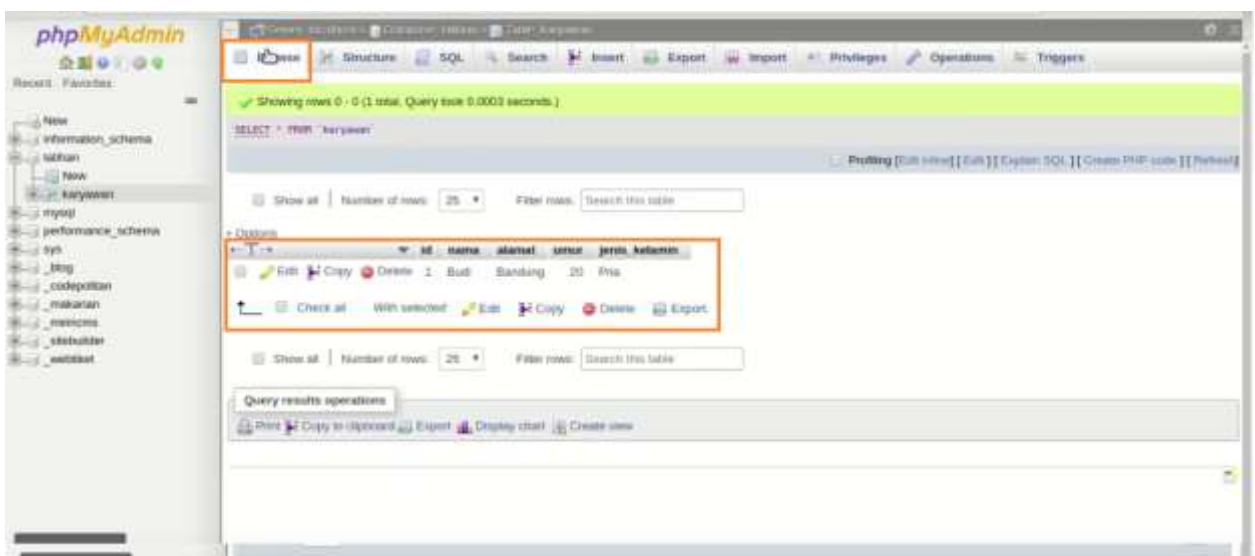


Jika kita perhatikan pada gambar diatas, pada field **id** valuenya di kosongkan karena pada materi sebelumnya **id** di set dengan auto increment yang artinya nilainya otomatis akan bertambah jika terjadi penambahan row pada table.

Setelah sudah yakin data yang dimasukan benar, kemudian klik tombol Go , jika sudah berhasil maka ada alert 1 row inserted seperti gambar dibawah ini.



Untuk memastikan apakah datanya sudah masuk pada tabel kita bisa melihatnya dengan klik tombol **Browse**.

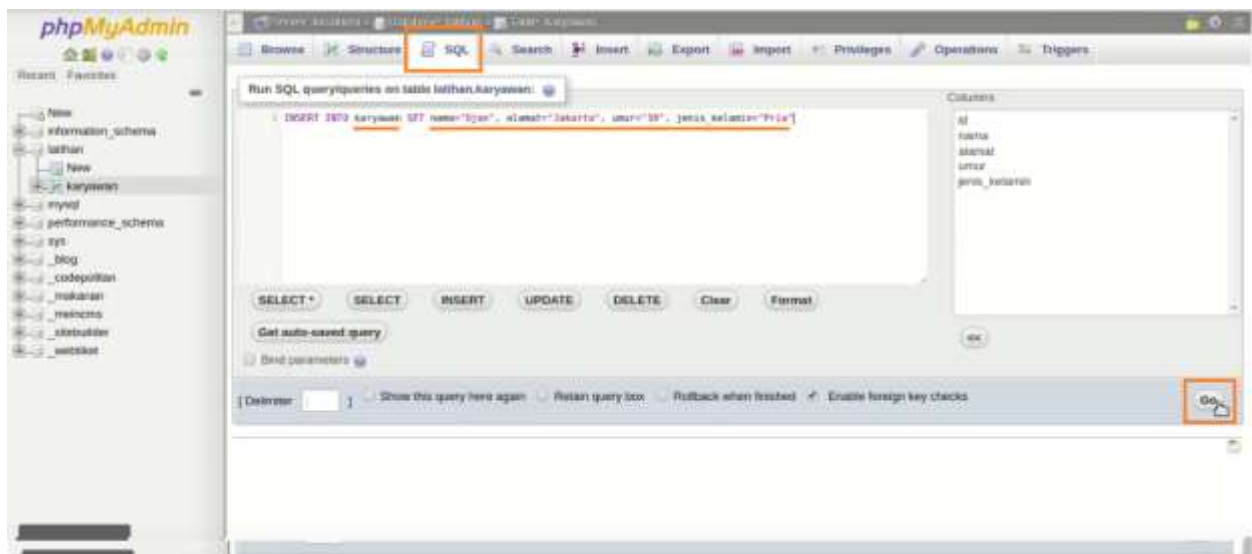


Jika kita lihat pada gambar diatas maka datanya sudah masuk ya.

Selain menggunakan tombol **Insert** untuk menambahkan data kita juga dapat menggunakan sql untuk insert data nya. Materi sql ini akan dipakai ketika kita membuat form sendiri di php, SQL atau Structure Query Language. SQL ini bahasa pemrograman yang dikhususkan untuk query

database, support di beberapa database seperti MySQL, Postgre, Oracle, Sql Server dan lain sebagainya.

Nah untuk insert data dengan sql kita dapat perhatikan gambar dibawah ini :



Jika kita lihat pada gambar diatas yang digaris bawah dimana **INSERT INTO** diikuti dengan nama tabel nya **karyawan** kemudian diikuti **set** dan kemudian nama **field** beserta valuenya, dimana antar **field** dipisahkan dengan koma.

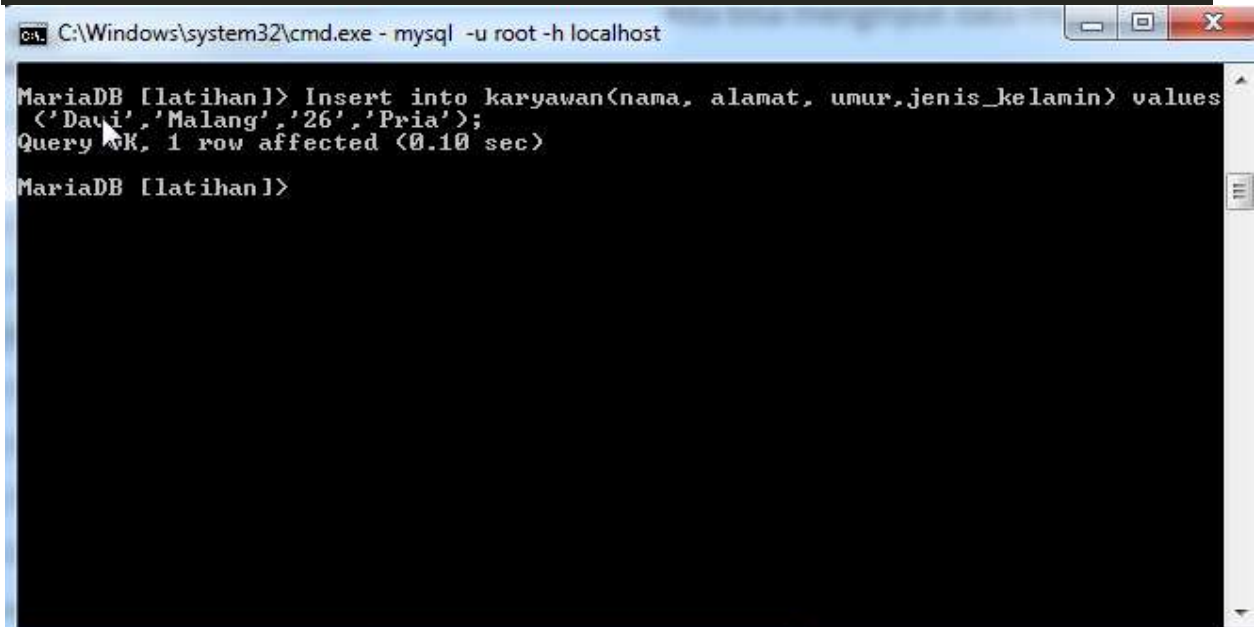
Nah jika sudah yakin dengan query diatas maka kemudian klik tombol Go dan datanya akan masuk ke dalam tabel kita sama persis seperti ketika menggunakan tombol Insert.

Kita bisa menginput data melalui console dengan baris perintah di atas atau bisa juga seperti berikut ini.

```
Insert      into      namaTabel(field1,      field2,      field3)      values
('value1','value2','value3');
```

Contoh

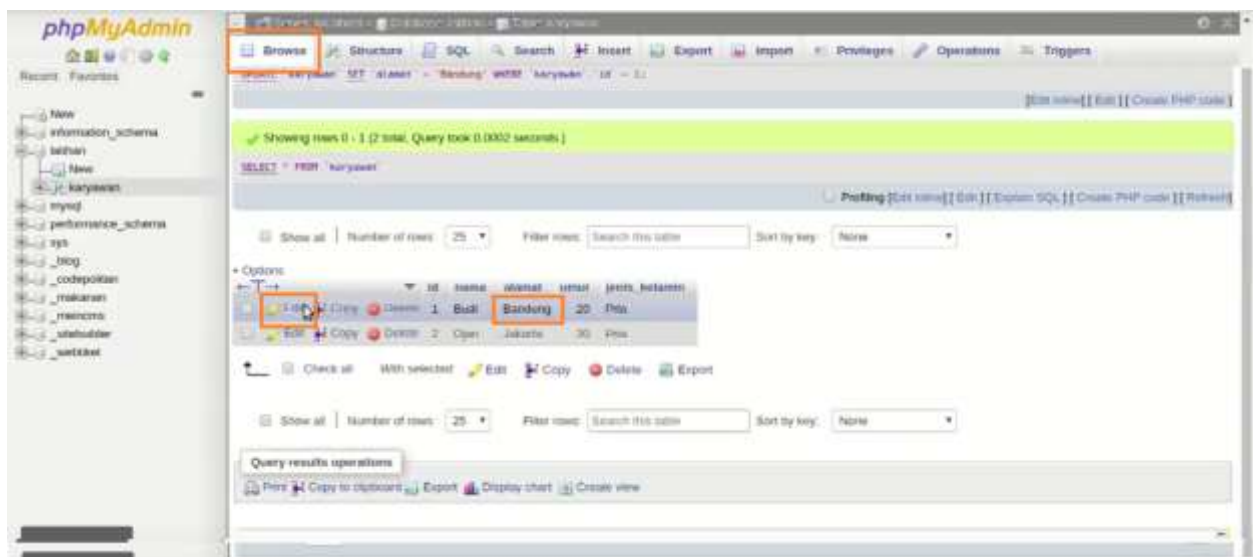
```
Insert into karyawan(nama, alamat, umur,jenis_kelamin) values  
( 'Davi', 'Malang', '26', 'Pria');
```



```
C:\Windows\system32\cmd.exe - mysql -u root -h localhost  
  
MariaDB [latihan]> Insert into karyawan(nama, alamat, umur,jenis_kelamin) values  
<'Davi','Malang','26','Pria');  
Query OK, 1 row affected (0.10 sec)  
  
MariaDB [latihan]>
```

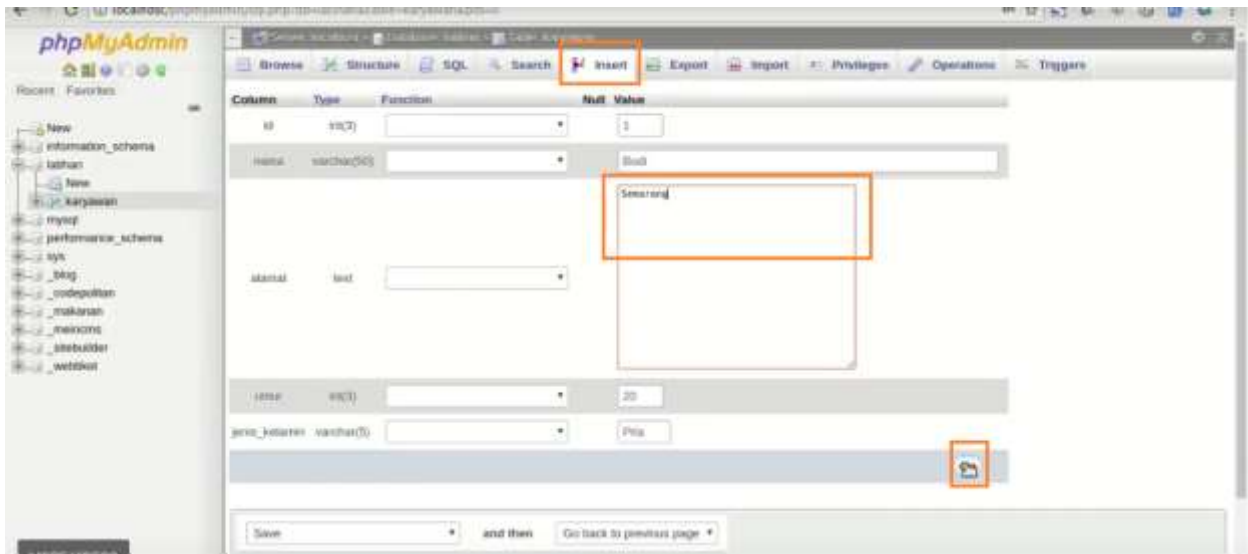
4. Update Data Table

Sebelumnya kita sudah berhasil *insert* data pada tabel *karyawan* dan mempunyai 2 *record*, lalu bagaimana jika kita ingin meng-*update* data yang sudah kita *insert* sebelumnya?

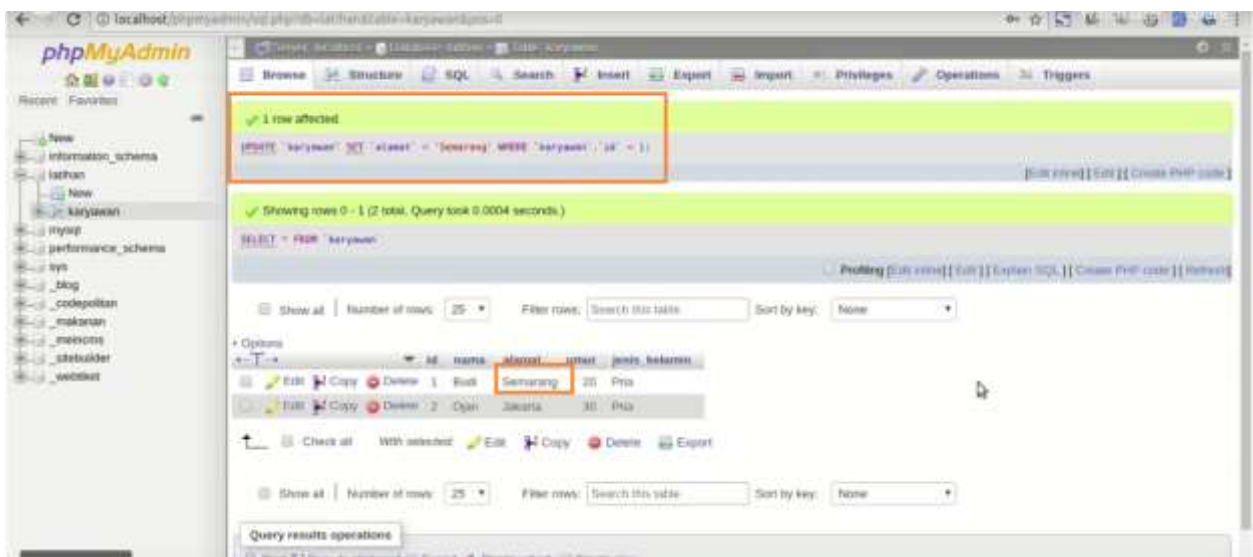


Jika kita lihat maka terdapat tombol edit , pada kasus ini kita akan coba *update* data *record* pertama dimana memiliki alamat *Bandung* dan akan kita ganti menjadi *semarang*.

Perhatikan gambar dibawah ini :



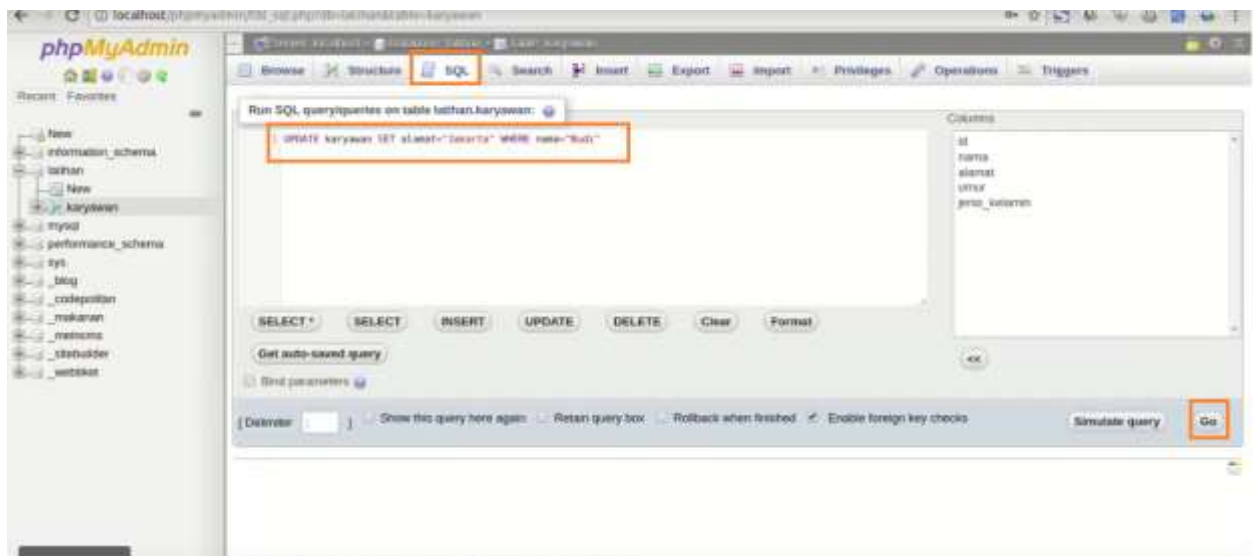
Perhatikan gambar diatas, dimana saya sudah mengganti value dari alamat yang tadinya *Bandung* saya *update* menjadi *Semarang*, setelah yakin semua datanya benar sekarang klik tombol **Go** .



Nah ketika kita sudah berhasil meng-*update* datanya maka akan muncul alerts seperti gambar di atas dan data alamat sudah berubah menjadi semarang.

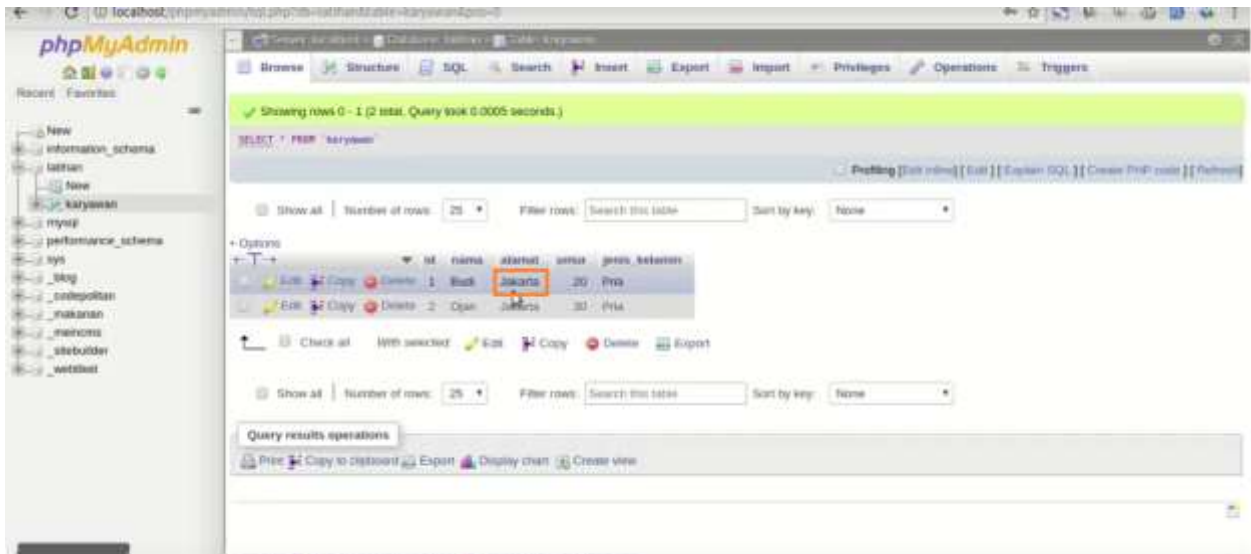
Lalu bagaimana cara *update* data menggunakan sql sintaks?

Perhatikan gambar dibawah ini :



Jika kita lihat pada gambar diatas untuk meng-*update* data dengan sql, maka kita cukup klik menu **SQL** kemudian tuliskan sintaks dikotak yang sudah disediakan seperti gambar diatas, sintaks **UPDATE** diikuti nama table **karyawan** kemudian set dan tuliskan nama *field* yang akan di *update* pada gambar diatas kita akan *update field* **alamat** dengan value 'Jakarta' , **where** digunakan sebagai pengecekan kondisi **nama** sama dengan **Budi**, jadi *update* tabel karyawan pada *field* alamat dimana nama nya memiliki value **Budi**.

Nah jika sudah yakin dengan data yang akan di *update* selanjutnya klik tombol **Go** maka jika berhasil, data tersebut akan ter-*update* seperti dibawah ini :

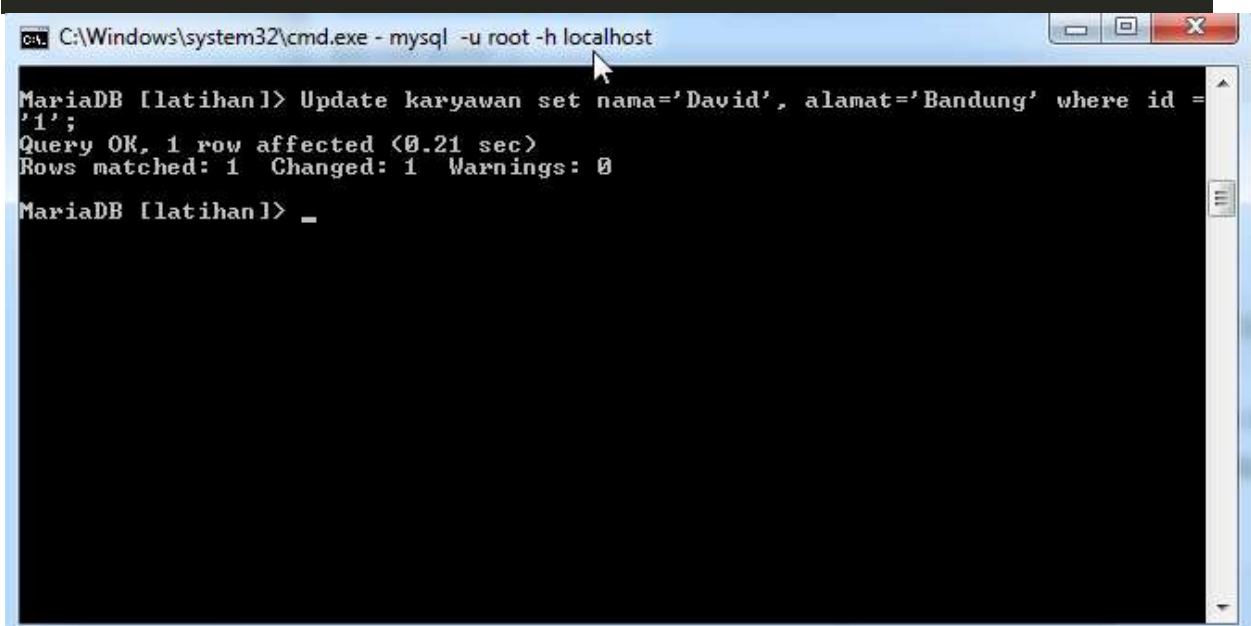


Kita juga bisa mengedit data melalui **console** seperti berikut ini.

```
Update namaTabel set field1='newValue1', field2='newValue2' where fieldFilter='value';
```

fieldFilter adalah nama field yang kita gunakan sebagai filter data apa yang hendak diupdate, kita harus sangat berhati-hati menentukan **fieldFilter** karena berpengaruh terhadap data-data yang hendak diupdate, umumnya **id** atau data unik yang dijadikan **fieldFilter**.

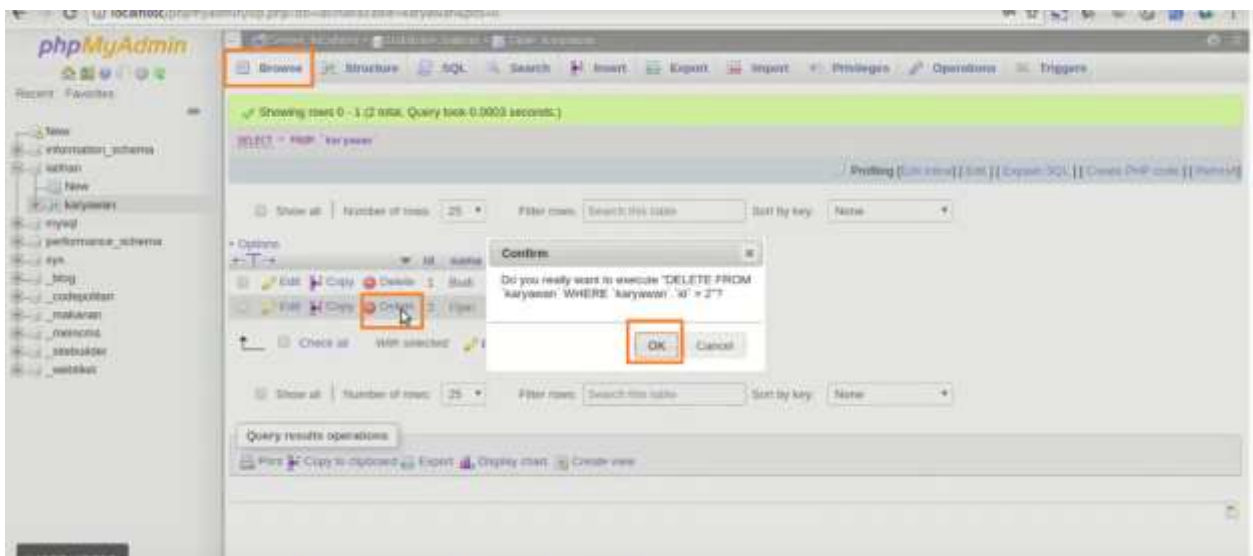
```
Update karyawan set nama='David', alamat='Bandung' where id ='1';
```



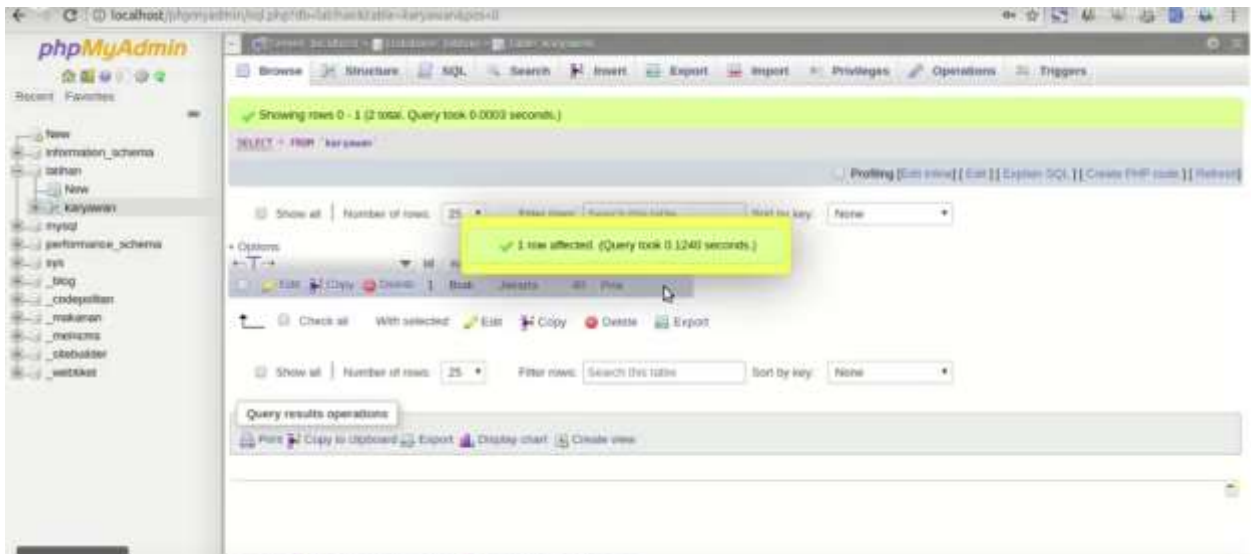
5. Delete Data Table

Setelah pada materi sebelumnya kita sudah berhasil melakukan *update* data pada tabel *karyawan* nah sekarang bagaimana jika kita ingin menghapus data pada tabel.

Perhatikan gambar di bawah ini :

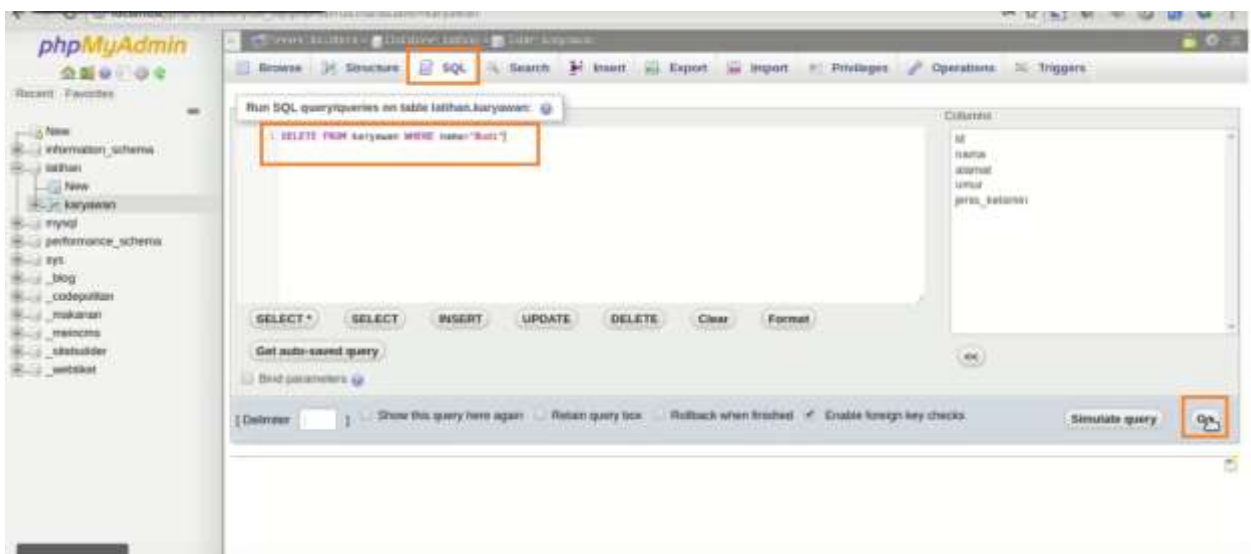


Jika kita lihat gambar diatas, maka untuk menghapus data kita cukup klik pada tombol **Delete** pada data yang akan kita hapus, kemudian akan muncul konfirmasi penghapusan data, kemudian pilih **OK** jika sudah yakin. Setelah berhasil melakukan penghapusan data maka akan muncul *alert* seperti dibawah ini:



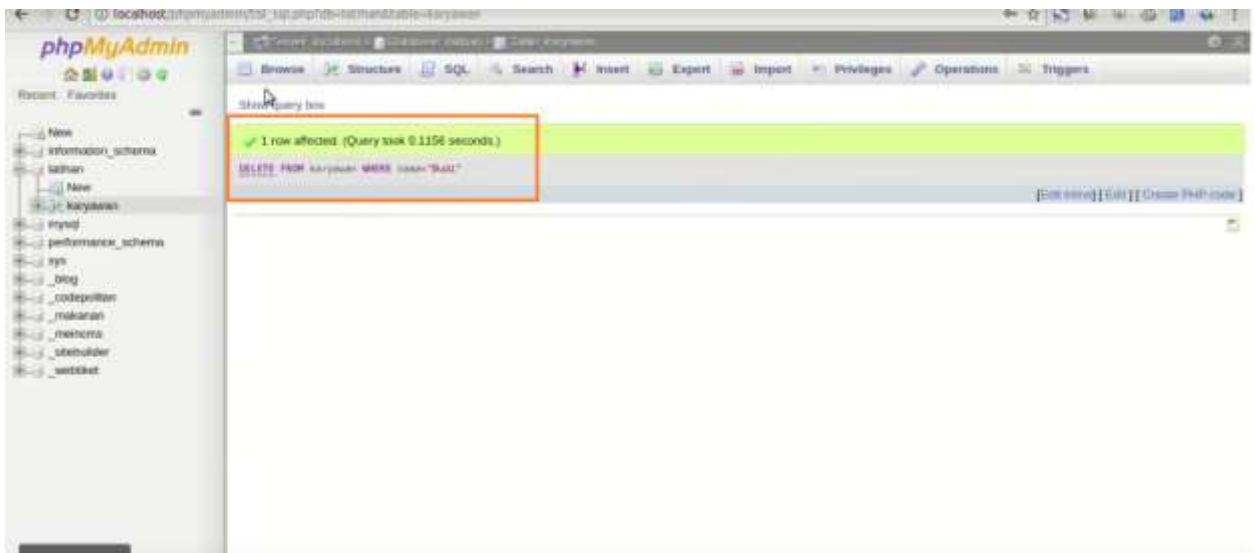
Dan jika kita lihat pada gambar diatas maka tersisa satu *record* data saja.

Lalu bagaimana jika kita ingin menghapus data menggunakan sintaks SQL ?



Jika kita lihat pada sintaks diatas maka tertulis **DELETE FROM** kemudian diikuti nama tabel nya **karyawan** jika sampai disitu saja maka semua *record* yang ada ada tabel akan dihapus, untuk menghapus data spesifik maka kita dapat menggunakan **WHERE**

Kemudian diikuti dengan data spesifiknya, contoh diatas kita akan menghapus data dengan data yang memiliki `nama="Budi"`. Jika sudah yakin dengan sintaks tersebut maka kemudian tekan tombol **Go** dan akan muncul *alerts* seperti dibawah ini.



Dan jika dilihat pada **Browse** maka data pada tabel sudah kosong.



Kita bisa menghapus melalui **console** seperti berikut ini.

```
delete from namaTabel where fieldFilter='value';
```

Contoh

```
delete from karyawan where id ='2';
```

Misalnya pada baris perintah di atas, kita akan menghapus data karyawan dengan **id** sama dengan 2.



```
C:\Windows\system32\cmd.exe - mysql -u root -h localhost

MariaDB [latihan]> delete from karyawan where id ='2';
Query OK, 1 row affected (0.15 sec)

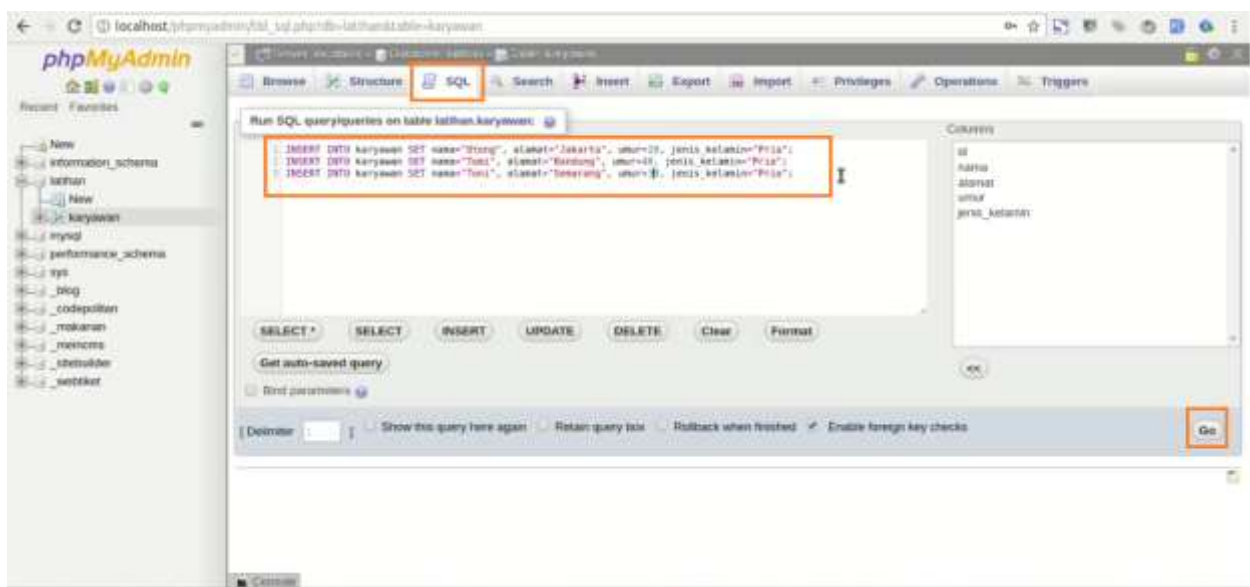
MariaDB [latihan]> _
```


6. Select dan Select Filter

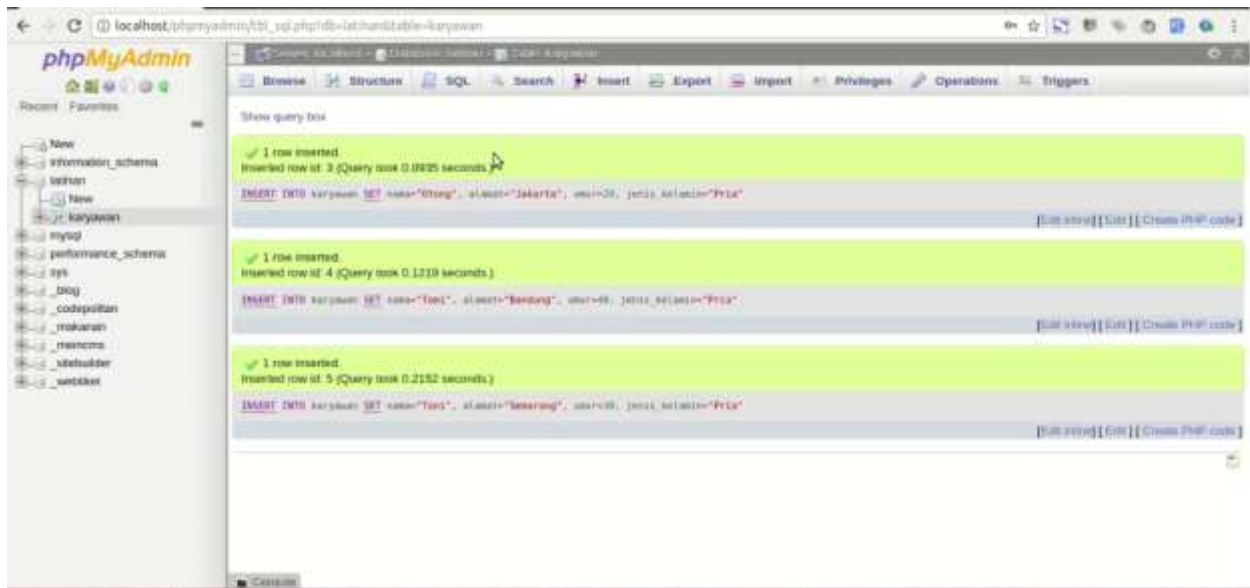
A. SELECT

Sebelum melakukan select data kita, coba insert data baru terlebih dahulu, karena data sebelumnya sudah kita hapus.

Kita coba insert data seperti gambar dibawah ini.

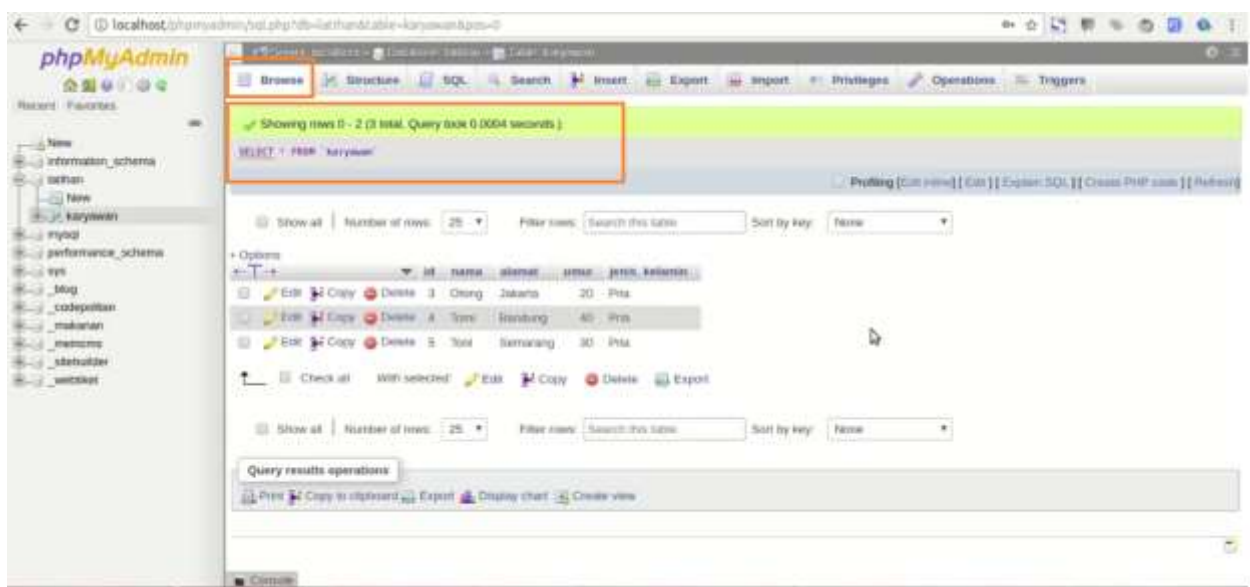


Jika kita lihat pada gambar diatas, kita mencoba insert langsung tiga data sekaligus.



Jika sudah tampil seperti gambar diatas, artinya kita sudah berhasil meng-insert data baru.

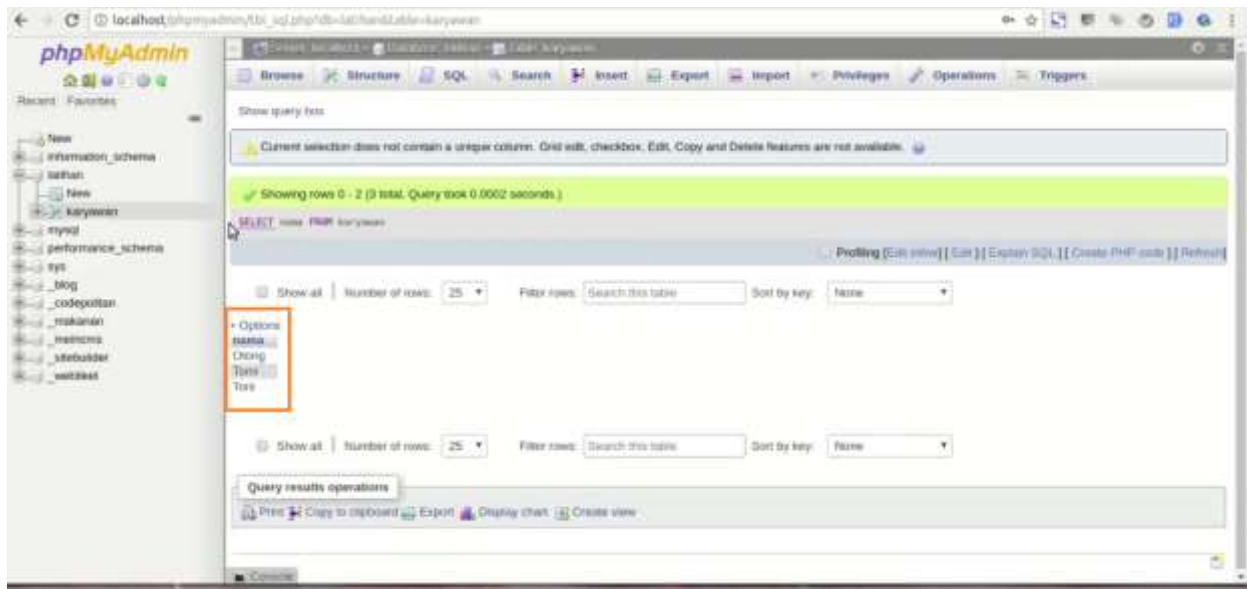
Jika kita perhatikan, ketika kita klik tombol Browse seperti gambar dibawah, sebenarnya mesin menjalankan sql select, dimana phpmyadmin sebenarnya adalah hanya alat bantu, karena dia dibelakang layar menjalankan script sql untuk berkomunikasi dengan mesin mysql, databasenya yaitu mysql.



Sekarang kita akan coba jalankan script select data berdasarkan namanya, klik menu SQL masukan script dibawah ini, kemudian jalankan.

```
SELECT nama from karyawan
```

Nah jika kita menjalankan script diatas dan datanya ditemukan, maka akan ditampilkan seperti gambar di bawah ini.



B. SELECT FILTER

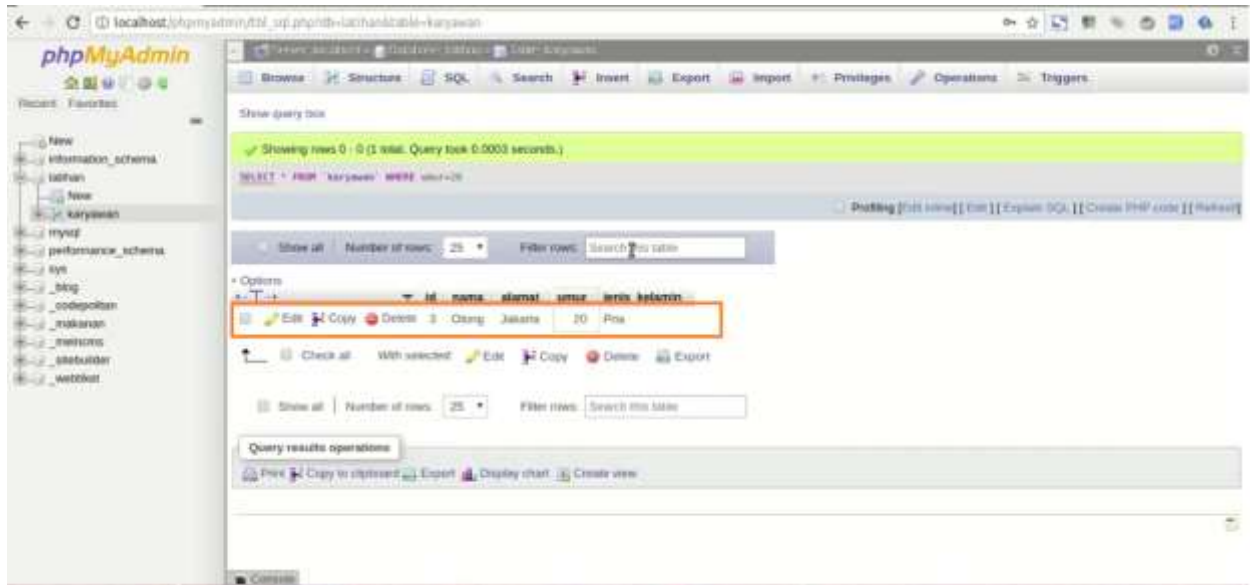
Sql select ini berguna untuk teknik reporting, ketika datanya sudah banyak ribuan atau bahkan jutaan kita menggunakan select filter ini.

Bagaimana jika kita ingin mengambil data spesifik yang hanya memiliki umur 20 saja ? nah coba jalankan script dibawah ini.

```
SELECT * FROM 'karyawan' WHERE umur=20
```

Jika kita perhatikan script diatas, penulisan nama tabel menggunakan tanda kutip, sebenarnya penulisan nama tabel bisa menggunakan tanda kutip atau tanpa tanda kutip seperti script diatas.

Jika datanya ditemukan yaitu data yang memiliki umur 20, maka akan ditampilkan data seperti pada gambar dibawah ini.

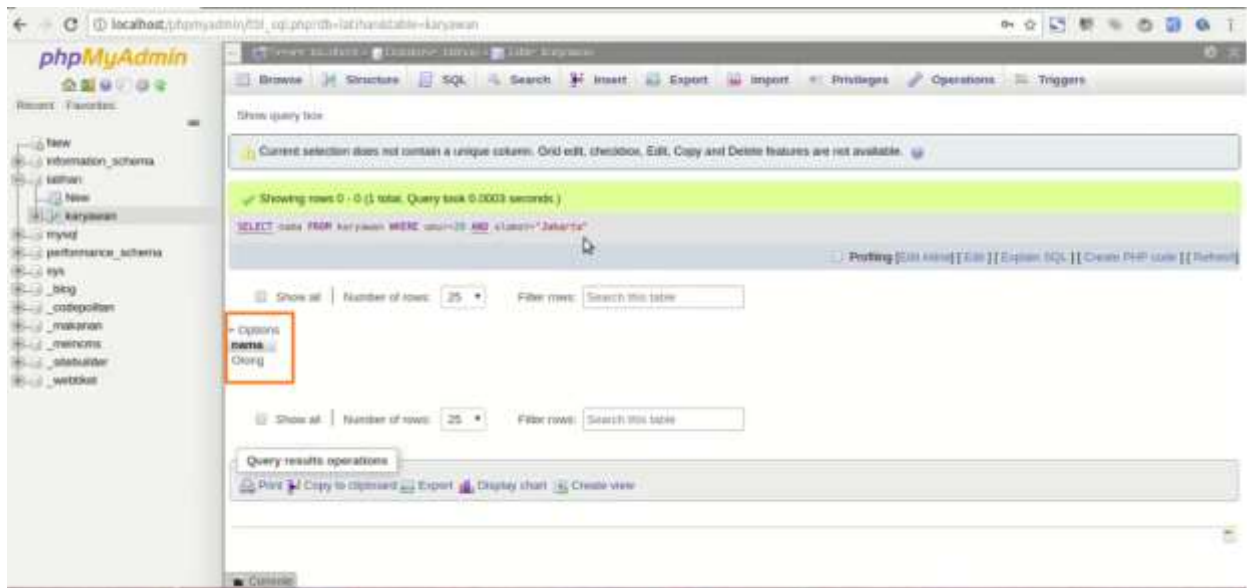


Kita juga dapat melakukan select lebih dari satu field, dimana kita bisa menggunakan keyword **AND**, kita coba select filter berdasarkan field umur dan kelamin.

Jalankan script dibawah ini.

```
SELECT nama FROM 'karyawan' WHERE umur=20 AND alamat="Jakarta"
```

Jika datanya ditemukan dimana data tersebut memiliki umur 20 dan alamat Jakarta, maka akan ditampilkan data seperti gambar dibawah ini.



Kita juga dapat menjalankan perintah select di console seperti berikut ini.

```
select * from namaTabel;
```

atau

```
select field1, field2 from namaTabel Where fieldFilter='value';
```

Contoh

```
select * from karyawan where umur >= 17;
```

Misalkan kita akan menampilkan semua data karyawan yang memiliki umur lebih dari atau sama dengan 17 tahun.

```
C:\Windows\system32\cmd.exe - mysql -u root -h localhost

MariaDB [latihan]> select * from karyawan where umur >= 17;
+----+-----+-----+-----+-----+
| id | nama  | alamat | umur | jenis_kelamin |
+----+-----+-----+-----+-----+
| 1  | David | Bandung | 26  | Pria           |
+----+-----+-----+-----+-----+
1 row in set (0.00 sec)

MariaDB [latihan]>
```

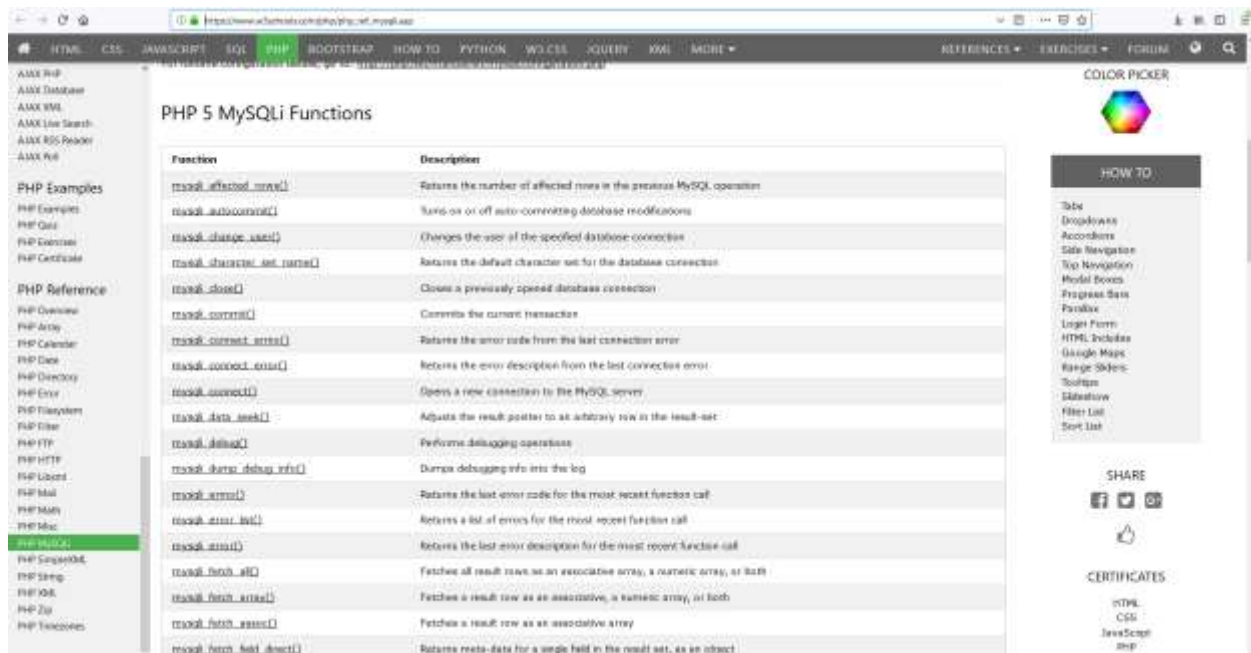

CRUD PHP

1. Mengenal MySQLi

Pada episode kali ini kita akan membuat aplikasi sederhana untuk *manage* data karyawan. Pada episode sebelumnya kita sudah belajar tentang sql dasar dan sudah membuat database karyawan.

Kita akan mengkoneksikan database karyawan tersebut dengan PHP. Kita akan menggunakan beberapa *function* internal berkategori MySQLi, sehingga diharapkan Anda sudah belajar dasar-dasar PHP sampai dengan materi *function*.

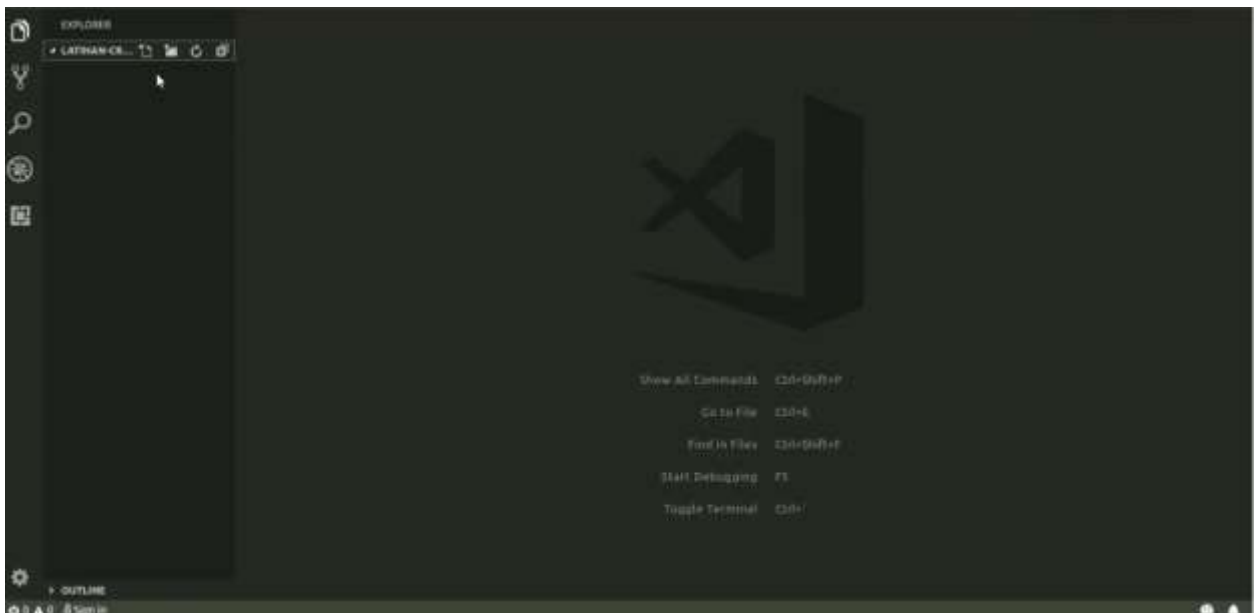
Peran dari *function* MySQLi adalah sebagai jembatan komunikasi antara PHP dan mesin MySQL, PHP sudah memudahkan sekali untuk kita mengoperasikan MySQL pada sintaks- sintaks PHP. Kita dapat melihat beberapa *function* yang siap kita pakai di https://www.w3schools.com/php/php_ref_mysqli.asp.



Jika kita lihat gambar diatas, *function* internal PHP yang berkategori MySQLi cukup banyak, tetapi dalam praktiknya kita hanya menggunakan beberapa saja.

2. Mengkoneksikan PHP dengan MySQL

Untuk mengkoneksikan PHP dan MySQL maka yang pertama kali kita lakukan adalah membuat folder **project** baru di dalam folder XAMPP untuk pengguna windows. Kemudian masuk ke folder htdocs dan membuat folder dengan nama **latihan-crud**, setelah itu buka folder tersebut dengan menggunakan teks editor, disini saya menggunakan visual studio code seperti dibawah ini.

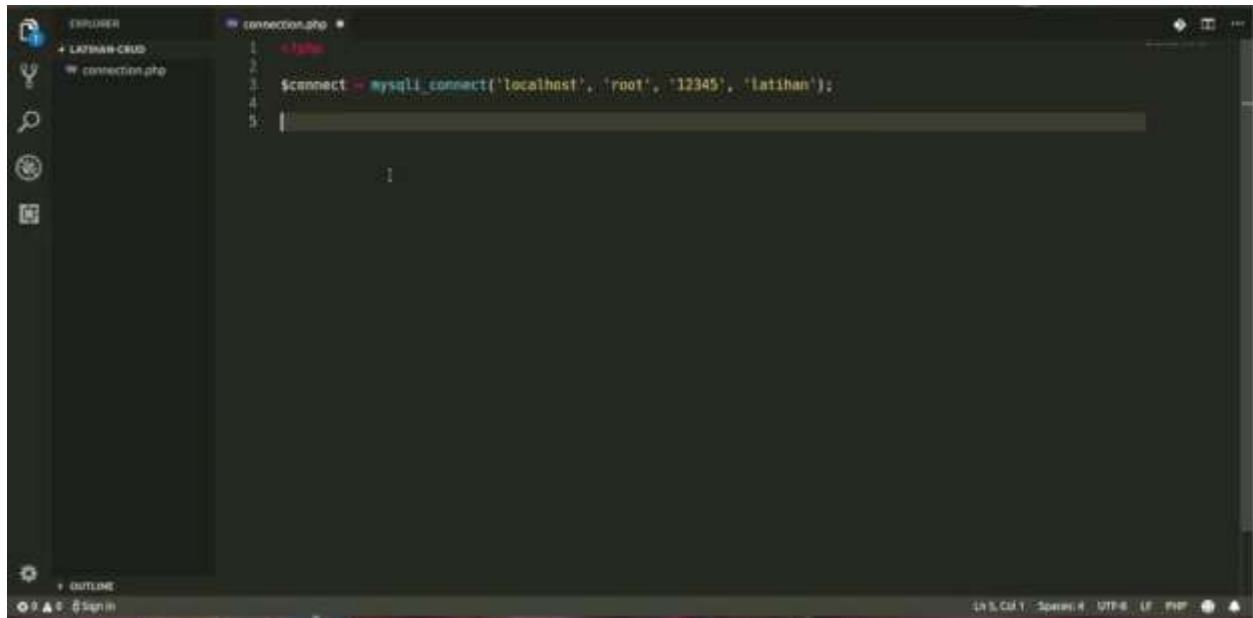


Kemudian kita tambahkan file dengan nama **connection.php** pada folder latihan-crud tersebut, lalu tambahkan kode dibawah ini :

```
<?php
    $connect = mysqli_connect('localhost','root','12345','latihan');
?>
```

Jika kita lihat pada potongan kode diatas, **mysqli_connect** menerima empat parameter, dimana yang pertama adalah tempat dimana mysql diinstal,

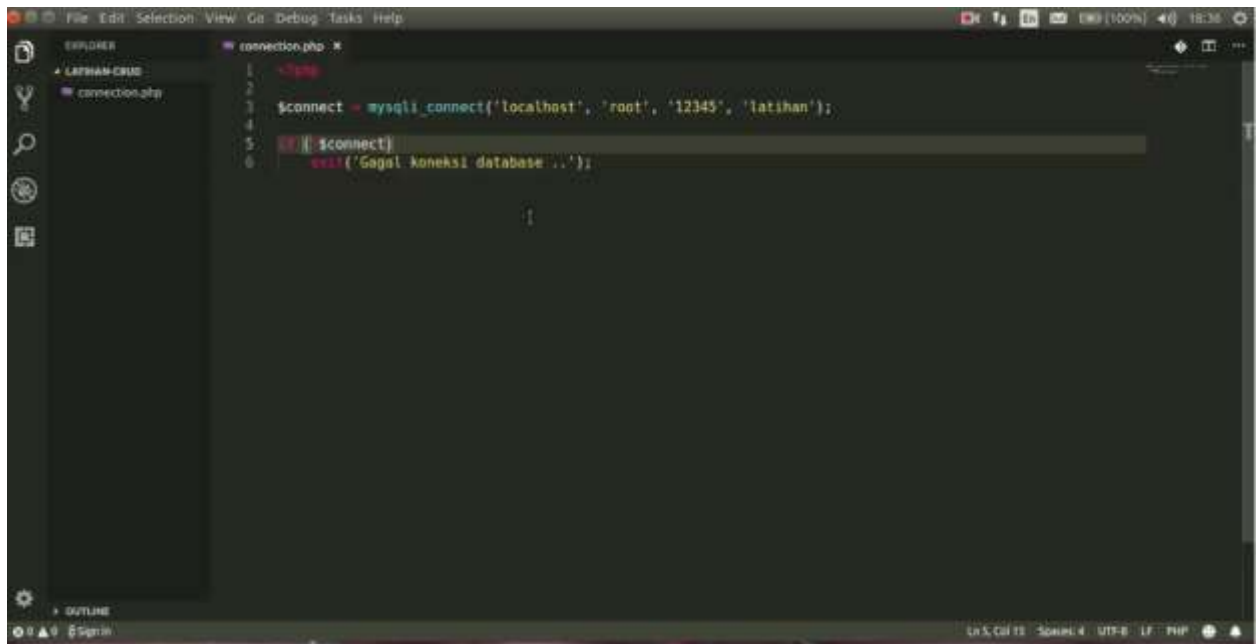
parameter kedua adalah username mysql, parameter ketiga adalah passwordnya, dan yang terakhir adalah nama database yang ingin kita koneksikan.



Jika kita lihat pada gambar diatas, fungsi `mysqli_connect` akan mengembalikan nilai `true` jika koneksinya berhasil atau `false` jika koneksinya gagal, sehingga kita dapat menambahkan kondisi jika koneksi gagal maka seluruh program kita di interrupt supaya digagalkan, dengan menggunakan kode dibawah ini :

```
if(!connect)
    exit('Gagal koneksi databse..');
```

Sehingga kode nya akan menjadi seperti di bawah ini :



The screenshot shows a code editor window with a dark theme. The Explorer panel on the left shows a project named 'LAPRAN-CRUD' with a file named 'connection.php'. The main editor area displays the following PHP code:

```
1 <?php
2
3 $connect = mysqli_connect('localhost', 'root', '12345', 'latihan');
4
5 if (!$connect)
6     exit('Gagal koneksi database ..');
```

The status bar at the bottom indicates the file is 'connection.php' and the encoding is 'UTF-8'.

Nah sekarang kita sudah selesai mengkoneksikan PHP dan MySQL.

3. Menampilkan Data Karyawan

Untuk menampilkan data karyawan kita perlu membuat file baru dengan nama `list.php`, setelah itu tambahkan kode dibawah ini kedalam file tersebut.

```
<?php

include ('connection.php');

$query = mysqli_query($connect,"SELECT * FROM karyawan");
$results = mysqli_fetch_all($query, MYSQLI_ASSOC);
?>

<html>
<body>
    <table border="1">
        <tr>
            <th>Nama</th>
            <th>Alama</th>
            <th>Umur</th>
            <th>Jenis Kelamin</th>
        </tr>
        <?php foreach($results as $result) : ?>
            <tr>
                <td><?php echo $result['nama']?></td>
                <td><?php echo $result['alamat']?></td>
                <td><?php echo $result['umur']?></td>
                <td><?php echo $result['jenis_kelamin']?></td>
            </tr>
        <?php endforeach ?>
    </table>
</body>
</html>
```

Jika kita lihat kode diatas, terdapat fungsi php `mysqli_query` yang digunakan untuk mengquerikan data didalam database yang sudah kita pilih

di `connection`. Terdapat dua parameter, dimana parameter pertama menerima variabel koneksinya, parameter kedua menerima sql sintaks yang ingin kita jalankan.

Kemudian terdapat juga fungsi `mysqli_fetch_all` yang digunakan untuk mengambil data dari mysql query dimana parameter pertama adalah variabel `$query` yang ada diatasnya, parameter kedua diisi dengan konstanta `MYSQLI_ASSOC` yang sudah di *defind* oleh php dimana akan mengembalikan nilai asosiasi.

Fungsi `foreach` diatas akan melakukan perulangan sebanyak array yang ada didalam variabel `$results`.

Setelah itu untuk melihat datanya sekarang kita buka url `localhost/latihan-crud/list.php`.

Jika kita sudah mengikuti perintah diatas, maka kita sudah berhasil menampilkan data karyawan, dan akan tampil seperti gambar dibawah ini pada halaman browser kita.

← → ↻ localhost/latihan-crud/lat.php

Nama	Alamat	Umur	Jenis Kelamin
Orang	Jakarta	20	Pria
Toni	Bandung	40	Pria
Toni	Semarang	30	Pria

4. Membuat Form Insert

Untuk membuat form insert, kita membutuhkan sebuah link untuk menuju ke halaman form insert.

```
<a href="add.php">Tambah Data</a>
```

Tambahkan kode di atas ke dalam file `list.php`, sehingga kodenya menjadi seperti dibawah ini :

```
<?php

include ('connection.php');

$query = mysqli_query($connect,"SELECT * FROM karyawan");
$results = mysqli_fetch_all($query, MYSQLI_ASSOC);
?>

<html>
<body>

    <a href="add.php">Tambah Data</a>

    <br/><br/>

    <table border="1">
        <tr>
            <th>Nama</th>
            <th>Alama</th>
            <th>Umur</th>
            <th>Jenis Kelamin</th>
        </tr>
        <?php foreach($results as $result) : ?>
            <tr>
                <td><?php echo $result['nama']?></td>
                <td><?php echo $result['alamat']?></td>
                <td><?php echo $result['umur']?></td>
                <td><?php echo $result['jenis_kelamin']?></td>
            </tr>
```

```
</table>
</body>
</html>
```

Sekarang kita coba buka `localhost/latihan-crud/list.php` maka akan tampil seperti dibawah ini :



Nama	Alamat	Umur	Jenis Kelamin
Oseng	Jakarta	20	Pria
Toni	Bandung	40	Pria
Toni	Semarang	30	Pria

Setelah membuat link untuk tambah data seperti gambar diatas, sekarang kita buat file baru dengan nama `add.php`, kemudian masukan kode di bawah ini untuk membuat formnya.

```
<html>
  <form action="insert.php" method="post">
    <label>Nama</label><br/>
    <input type="text" name="nama"/>
    <br/><br/>

    <label>Alamat</label><br/>
    <textarea name="alamat" cols="30" rows="10"/></textarea>
    <br/><br/>

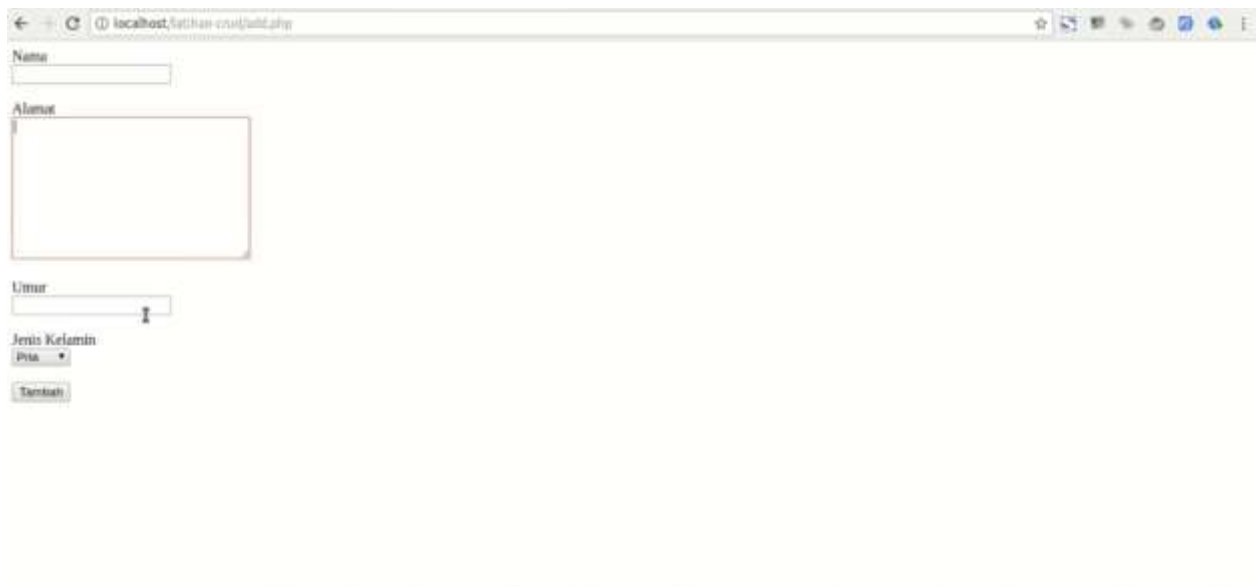
    <label>Umur</label><br/>
    <input type="text" name="umur"/>
    <br/><br/>
```



```
<label>Jenis Kelamin</label><br/>
<select name="jenis_kelamin">
  <option value="Pria">Pria</option>
  <option value="Wanita">Wanita</option>
</select>

</button type="submit">Tambah</button>
</form>
</html>
```

Sekarang kita coba buka url <localhost/latihan-crud/add.php>.



The screenshot shows a web browser window with the address bar displaying "localhost/latihan-crud/add.php". The form contains the following elements:

- A text input field labeled "Nama".
- A text area labeled "Alamat".
- A text input field labeled "Umur".
- A dropdown menu labeled "Jenis Kelamin" with "Pria" selected.
- A button labeled "Tambah".

Jika tampil seperti gambar diatas berarti kita sudah berhasil membuat form insert.

5. Insert Data

Setelah pada episode sebelumnya kita sudah membuat form insert, sekarang kita akan membuat handler untuk form insert tersebut. Selanjutnya kita buat file dengan nama `insert.php` sesuai dengan `action` yang ada di dalam form `list.php`, kemudian masukan kode dibawah ini kedalam file `insert.php` :

```
<?php

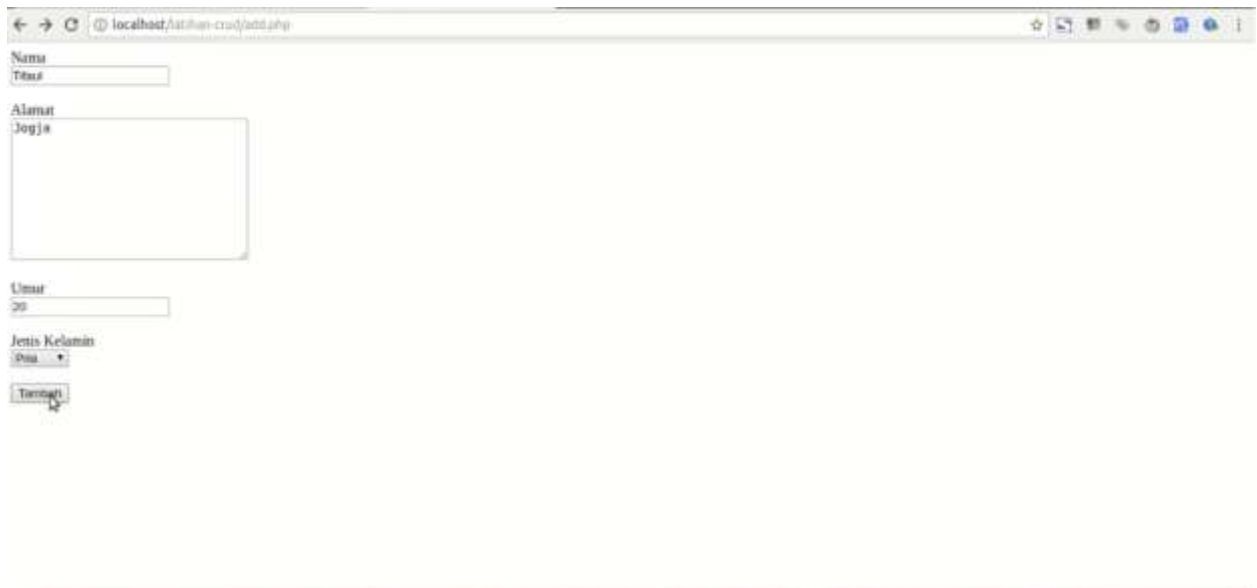
include('connection.php');

//karena form menggunakan method post kita gunakan $_POST
$nama = $_POST['nama']; //index didalamnya sesuai dengan input name yang ada di
form
$alamat = $_POST['alamat'];
$umur = $_POST['umur'];
$jenis_kelamin = $_POST['jenis_kelamin'];

$insert = mysqli_query($connect,"INSERT INTO karyawan SET nama='$nama',
alamat='$alamat', umur='$umur', jenis_kelamin='$jenis_kelamin' ");

if($insert)
    header('Location:list.php'); //Jika berhasil akan di arahkan ke halaman
list.php
else
    echo 'Input data gagal'; //jika gagal akan keluar pesan tersebut
```

Sekarang kita coba buka `localhost/latihan-crud/add.php` kemudian inputkan data seperti gambar dibawah.



localhost/latihan-crud/add.php

Nama
Tika

Alamat
Jogja

Umur
20

Jenis Kelamin
Pria

Tambah

Jika berhasil maka data akan diarahkan ke halaman [list.php](#) dan data yang di inputkan sudah masuk ke dalam tabel karyawan seperti gambar dibawah ini.



localhost/latihan-crud/list.php

Tambah Data

Nama	Alamat	Umur	Jenis Kelamin
Otono	Jakarta	20	Pria
Feni	Bandung	40	Pria
Feni	Semarang	30	Pria
Udin	Tangerang	40	Pria
Jaka	Surabaya	20	Pria
Tika	Jogja	20	Pria

6. Update Data

Untuk melakukan *update* data sebenarnya *logic* nya sama seperti yang sudah kita pelajari sebelumnya untuk melakukan edit data pada phpmyadmin, dimana kita dapat mengedit data satu persatu sesuai dengan *id* datanya.

Yang perlu kita buat pertama adalah *link* untuk edit data, kita buka file *list.php* kemudian kita tambahkan judul baru pada tabel dengan nama **pilihan** dimana didalamnya terdapat *link* edit.

```
<td>
    <a href="edit.php?id=?php echo $result['id']?>">Edit</a>
</td>
```

Kode diatas digunakan untuk membuat *link* edit, dimana ketika di klik akan di arahkan ke *edit.php* yang dimana setiap *link* tersebut menyimpan data spesifik *id* dari setiap data yang nantinya dapat kita olah pada file *edit.php*.

Kemudian tambahkan kode diatas kedalam file *list.php* .

```
<?php

include ('connection.php');

$query = mysqli_query($connect,"SELECT * FROM karyawan");
$results = mysqli_fetch_all($query, MYSQLI_ASSOC);
?>

<html>
<body>
    <a href="add.php">Tambah Data</a>

    <br/><br/>
```

```
<table border="1">
    <tr>
        <th>Nama</th>
        <th>Alamat</th>
        <th>Umur</th>
        <th>Jenis Kelamin</th>
        <th>Pilihan</th>
    </tr>
    <?php foreach($results as $result) : ?>
        <tr>
            <td><?php echo $result['nama']?></td>
            <td><?php echo $result['alamat']?></td>
            <td><?php echo $result['umur']?></td>
            <td><?php echo $result['jenis_kelamin']?></td>
            <td>
                <a href="edit.php?id=<?php echo $result['id']?>">Edit</a>
            </td>
        </tr>
    </table>
</body>
</html>
```

Maka akan tampil halaman seperti dibawah ini :

localhost/latihan-crud/edit.php

Tambah Data

Nama	Alamat	Umur	Jenis Kelamin	Pilihan
Otong	Jakarta	20	Pria	Edit
Foni	Bandung	40	Pria	Edit
Toni	Semarang	30	Pria	Edit
Udin	Tangerang	40	Pria	Edit
Jaka	Surabaya	20	Pria	Edit
Tihoni	Jogja	20	Pria	Edit

localhost/latihan-crud/edit.php?id=3

Jika kita lihat pada gambar diatas maka sekarang sudah terdapat *link* edit dimana dimana setiap *link* tersebut menyimpan *id* sesuai dengan datanya, seperti dapat kita lihat pada bagian pojok kiri bawah pada gambar diatas.

Selanjutnya kita buat file dengan nama *edit.php* dimana nantinya data yang akan kita edit di tampilkan dalam form edit ini .

```
<?php

include('connection.php');

$id = $_GET['id']; // Untuk mengambil id yang dilempar dari form list.php

$query = mysqli_query($connect,"SELECT * FROM karyawan WHERE id='$id' LIMIT 1");
//mengambil data sesuai dengan id nya
$result = mysqli_fetch_all($query, MYSQLI_ASSOC);
?>

<html>
    <form action="insert.php" method="post">

        <input type="hidden" name="id" value="<?php echo $result[0]['id']?>" <!--
untuk menyimpan id tanpa menampilkan data id pada form-->

        <label>Nama</label><br/>
        <input type="text" name="nama" value="<?php echo $result[0]['nama']?>"/>
<!--menampilkan data sesuai dr variabel $result diatas-->
        <br/><br/>

        <label>Alamat</label><br/>
        <textarea name="alamat" cols="30" rows="10"/>value="<?php echo
$result[0]['alamat']?>"</textarea>
        <br/><br/>

        <label>Umur</label><br/>
        <input type="text" name="umur"/>
        <br/><br/>
```

```

<label>Jenis Kelamin</label><br/>
<select name="jenis_kelamin">
    <option value="Pria"><?php echo ($result[0]['jenis_kelamin'] ==
'Pria') ? 'selected' : '';?> >Pria</option>
    <option value="Wanita" "><?php echo ($result[0]['jenis_kelamin'] ==
'Wanita') ? 'selected' : '';?> >Wanita</option>
</select>

</button type="submit">Perbaharui</button>
</form>
</html>

```

Sekarang jika kita coba klik *link* edit data pada salah satu data yang terdapat pada table tersebut, maka data lama akan di tampilkan pada form edit tersebut.

Nah sekarang kita perlu membuat handler untuk menyimpan file yang sudah diisikan pada form edit kedalam database, kita buat file **update.php**, kemudian masukan kode seperti dibawah ini.

```

<?php

include('connection.php'); // Mengkoneksikan dengan database

$id = $_POST['id'];
// Karena form menggunakan method post kita gunakan $_POST
$nama = $_POST['nama']; // Index didalamnya sesuai dengan input name yang ada di
form
$alamat = $_POST['alamat'];
$umur = $_POST['umur'];
$jenis_kelamin = $_POST['jenis_kelamin'];

$update = mysqli_query($connect,"UPDATE karyawan SET nama='$nama',
alamat='$alamat', umur='$umur', jenis_kelamin='$jenis_kelamin' WHERE id='$id' ");
//menggunakan kondisi where untuk menyimpan dengan data spesifik

```

```
if($update)
    header('Location:list.php'); // Jika berhasil akan di arahkan ke halaman
list.php
else
    echo 'Input data gagal'; // Jika gagal akan keluar pesan tersebut
?>
```

Sekarang kita coba update data dengan nama *Toni* dimana alamatnya menjadi *Medan*.



Tambah Data

Nama	Alamat	Umur	Jenis Kelamin	Pilihan
Otong	Jakarta	20	Pria	Edit
Timi	Bancong	40	Pria	Edit
Toni	Medan	30	Pria	Edit
Udin	Tanggulang	40	Pria	Edit
Jaka	Surabaya	20	Pria	Edit
Tifani	Jogja	20	Pria	Edit

Maka jika kita berhasil mengupdate datanya kita langsung diarahkan ke halaman `list.php` dan datanya sudah berubah seperti gambar dibawah ini.

← localboot:latihan-crud/fst.php

Tambah Data

Nama	Alamat	Umur	Jenis Kelamin	Pilihan
Otong	Jakarta	20	Pria	Edit
Timi	Bardong	40	Pria	Edit
Idan Tambah	Tambah	30	Pria	Edit
Udin	Tangerang	40	Pria	Edit
Jaka	Surabaya	20	Pria	Edit
Tilani	Jogja	20	Pria	Edit

7. Delete Data

Setelah kita sudah berhasil menambahkan fitur update sekarang kita akan menambahkan fitur delete , dimana sebenarnya konsepnya tidak jauh berbeda dengan update data karena kita perlu mengirim data spesifik dari data yang akan kita hapus.

Langsung saja, yang perlu kita buat adalah menambahkan link delete pada file `list.php`.

```
<a href="edit.php?id=<?php echo $result['id']?>">Delete</a>
```

Tambahkan kode diatas pada file `list.php` seperti dibawah ini.

```
<?php

include ('connection.php');

$query = mysqli_query($connect,"SELECT * FROM karyawan");
$results = mysqli_fetch_all($query, MYSQLI_ASSOC);
?>

<html>
<body>
    <a href="add.php">Tambah Data</a>

    <br/><br/>

    <table border="1">
        <tr>
            <th>Nama</th>
            <th>Alama</th>
            <th>Umur</th>
            <th>Jenis Kelamin</th>
            <th>Pilihan</th>
        </tr>
```

```
<?php foreach($results as $result) : ?>
    <tr>
        <td><?php echo $result['nama']?></td>
        <td><?php echo $result['alamat']?></td>
        <td><?php echo $result['umur']?></td>
        <td><?php echo $result['jenis_kelamin']?></td>
        <td>
            <a href="edit.php?id=<?php echo $result['id']?>">Edit</a>
            <a href="delete.php?id=<?php echo $result['id']?>">Delete</a>
        </td>
    </tr>
<?php endforeach;?>
</table>
</body>
</html>
```

Jika sudah menambahkan link delete, sekarang kita perlu membuat handler untuk menghapus data dari database, kita buat file dengan nama `delete.php` masukan kode dibawah ini.

```
<?php

include('connection.php');//mengkoneksikan database

$id = $_GET('id'); //mengambil id yang di parsing dari halaman list.php

$delete = mysqli_query($connect, "DELETE FROM karyawan WHERE id='$id'");
//menghapus data spesifik

if($delete)
    header('Location : list.php');
else
    echo 'Delete data gagal';
```

Nah sekarang kita coba hapus data dengan nama Jaka seperti gambar dibawah ini.



Jika kita lihat pada gambar diatas, maka ketika tombol delete diklik akan mengirimkan **id** dari data tersebut yang kemudian ditangkap **id** tersebut pada file **delete.php** yang kemudian dilakukan query untuk menghapus data.

Jika sudah berhasil maka datanya akan terhapus seperti gambar dibawah ini.



8. Search Data

Sekarang kita akan menambahkan fitur *search* , kita perlu membuat form untuk menginput *keyword*-nya pada file `list.php` seperti dibawah ini.

```
<form action="search.php" method="GET">
    <input type="text" name="keyword" placeholder="Keyword .." value="<?php echo
$_GET['keyword']?>" />
    <button type="submit">Search</button>
</form>
```

Jika kita lihat pada kode diatas kita menggunakan *method* `GET` karena kita tidak perlu menyembunyikan data yang akan kita cari.

Sekarang masukan kode diatas kedalam file `list.php` diatas tabel *list* data.

```
<?php

include('connection.php');

$query = mysqli_query($connect, "SELECT * FROM karyawan");
$results = mysqli_fetch_all($query, MYSQLI_ASSOC);
?>

<html>
<body>
    <a href="add.php">Tambah Data</a>

    <br/><br/>

    <form action="search.php" method="GET">
        <input type="text" name="keyword" placeholder="Keyword .." />
        <button type="submit">Search</button>
    </form>
    <table border="1">
        <tr>
            <th>Nama</th>
```

```

        <th>Alama</th>
        <th>Umur</th>
        <th>Jenis Kelamin</th>
        <th>Pilihan</th>
    </tr>
    <?php foreach($results as $result) : ?>
        <tr>
            <td><?php echo $result['nama']?></td>
            <td><?php echo $result['alamat']?></td>
            <td><?php echo $result['umur']?></td>
            <td><?php echo $result['jenis_kelamin']?></td>
            <td>
                <a href="edit.php?id=<?php echo $result['id']?>">Edit</a>
                <a href="edit.php?id=<?php echo $result['id']?>">Delete</a>
            </td>
        </tr>
    </table>
</body>
</html>

```

Kemudian kita perlu membuat file dengan nama `search.php` untuk menampilkan data berdasarkan *keyword* yang dikirim dari form list.

Masukan kode dibawah ini pada file `search.php`.

```

<?php

include('connection.php'); // Koneksi database

$keyword = $_GET['keyword']; // Mengambil keyword yang dikirim dari form

$query = mysqli_query($connect,"SELECT * FROM karyawan WHERE nama='$keyword'"); //
Mengambil data pada database berdasarkan nama sesuai keyword yang ditampung pada
variabel $keyword.

$results = mysqli_fetch_all($query, MYSQLI_ASSOC);
?>

<html>

```

```

<body>
  <a href="add.php">Tambah Data</a>

  <form action="search.php" method="GET">
    <input type="text" name="keyword" placeholder="Keyword .." value="<?php
echo $_GET['keyword']?>" />
    <button type="submit">Search</button>
  </form>
  <table border="1">
    <tr>
      <th>Nama</th>
      <th>Alama</th>
      <th>Umur</th>
      <th>Jenis Kelamin</th>
      <th>Pilihan</th>
    </tr>
    <?php foreach($results as $result) : ?>
      <tr>
        <td><?php echo $result['nama']?></td>
        <td><?php echo $result['alamat']?></td>
        <td><?php echo $result['umur']?></td>
        <td><?php echo $result['jenis_kelamin']?></td>
        <td>
          <a href="edit.php?id=<?php echo $result['id']?>">Edit</a>
          <a href="edit.php?id=<?php echo $result['id']?>">Delete</a>
        </td>
      </tr>
    </table>
</body>
</html>

```

Sekarang kita coba *search* data seperti gambar dibawah ini.



Jika tampil seperti gambar diatas artinya kita sudah berhasil membuat fitur *search* data.