

Laporan Praktek Database

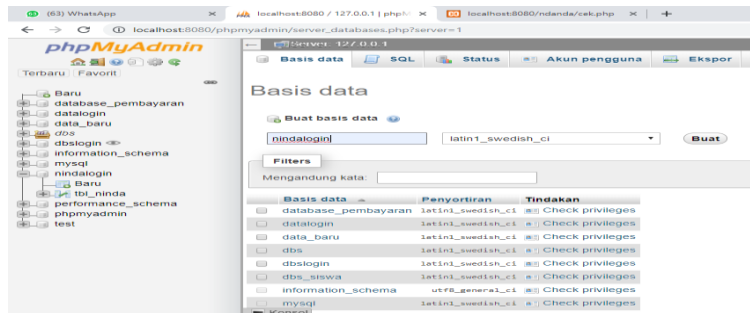


Nama : Daris

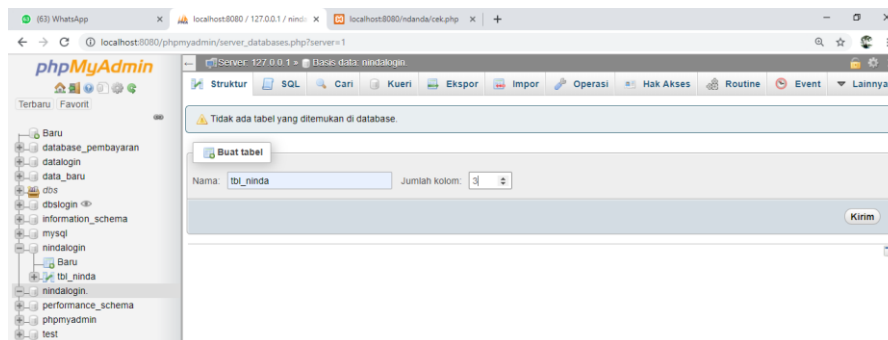
Kelas : XI-A

Langkah-langkah membuat Login!

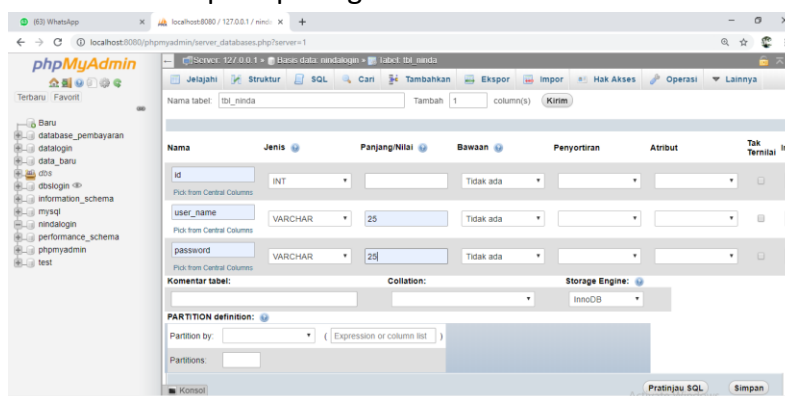
1. Pertama kita buka browser lalu ketikkan “localhost:8080/phpmyadmin”
2. Lalu kita klik baru, isikan nama database seperti gambar dibawah ini Lalu klik “Buat”.



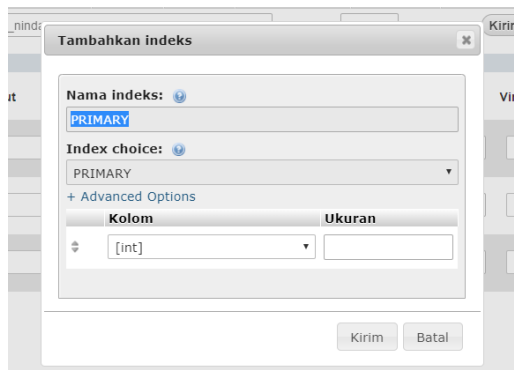
3. Untuk membuat tabel isikan nama tabel dan jumlah kolom yang di inginkan. Seperti pada gambar dibawah ini sebagai contoh, Lalu kita klik kirim.



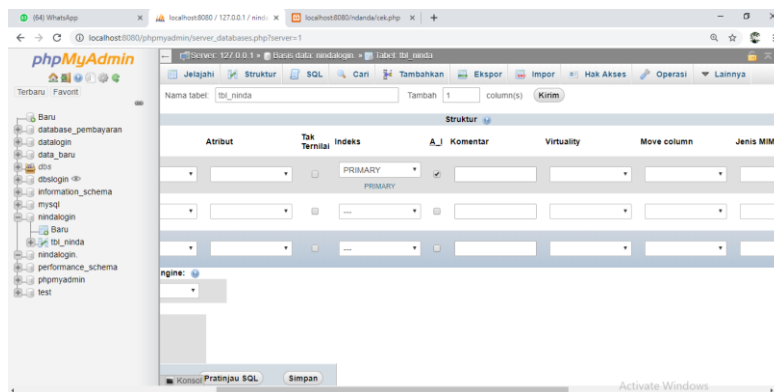
4. Isikan kolom seperti pada gambar di bawah ini!



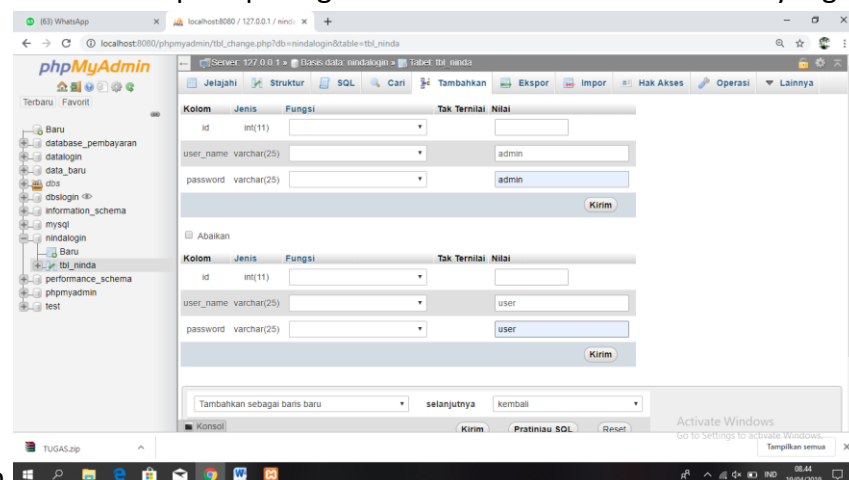
5. Pilih Primary pada indeks. Setelah muncul tampilan pada gambar di bawah ini, klik kirim.



6. Lalu centang kolom seperti pada gambar dibawah ini!

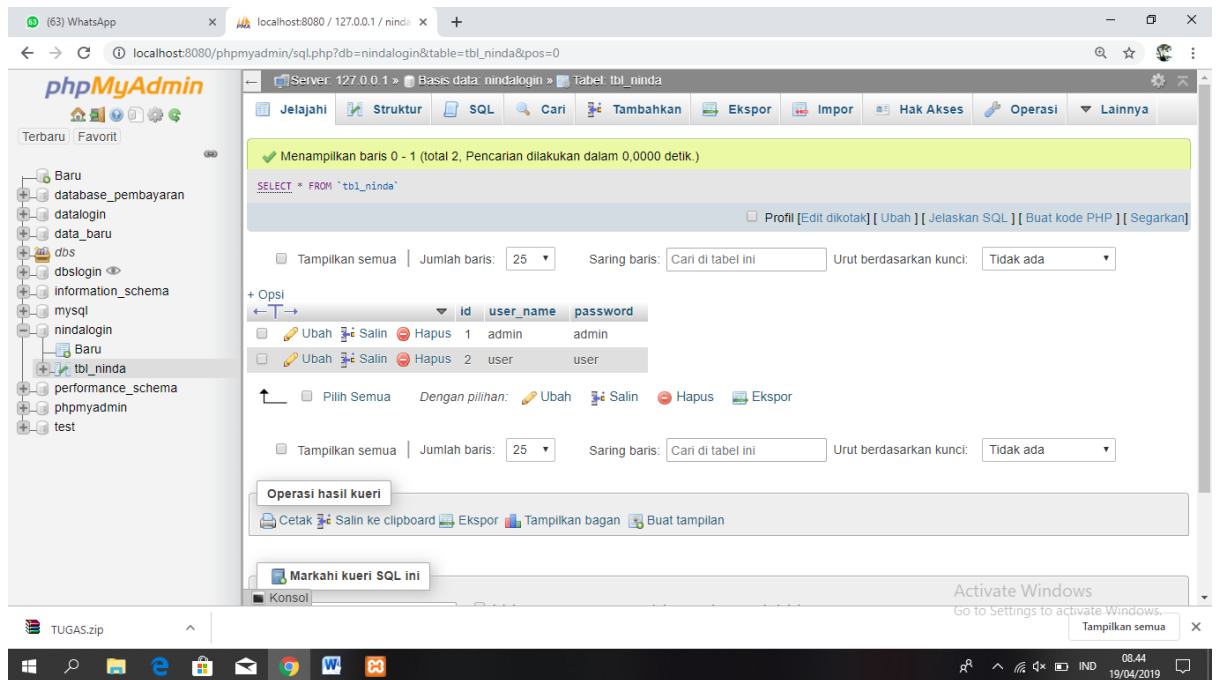


7. Setelah itu isikan kolom seperti pada gambar dibawah ini! Lalu klik kirim yang berada

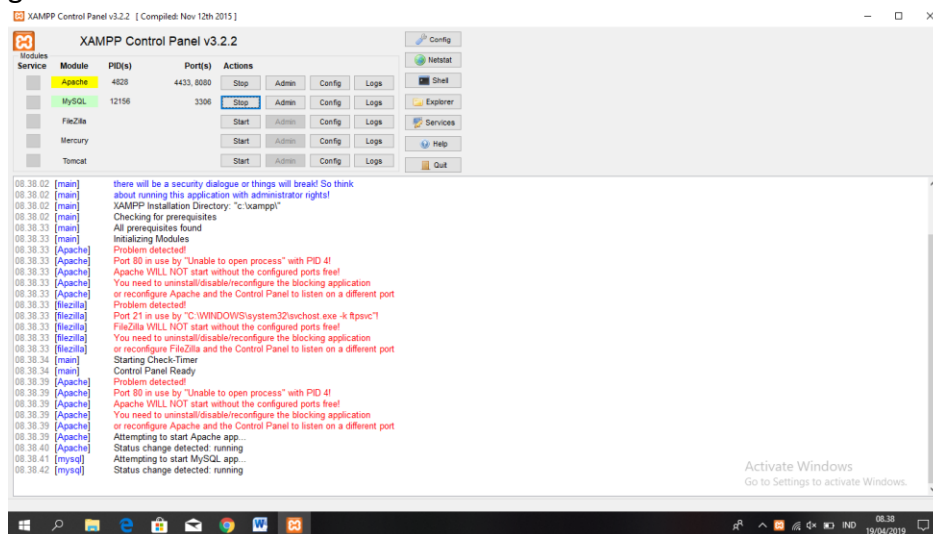


paling bawah.

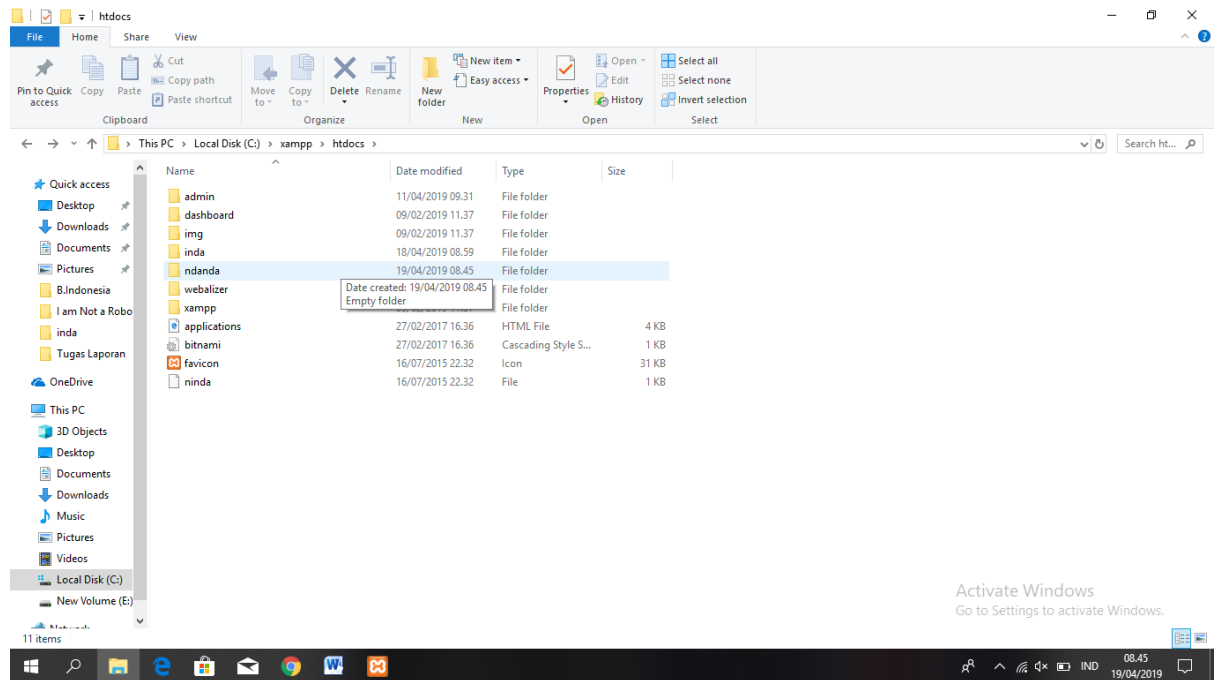
8. Setelah muncul klik kirim, akan muncul tampilan seperti pada gambar dibawah ini!



9. Setelah itu buka aplikasi XAMPP, lalu klik start pada Apache dan Mysql seperti pada gambar dibawah ini!

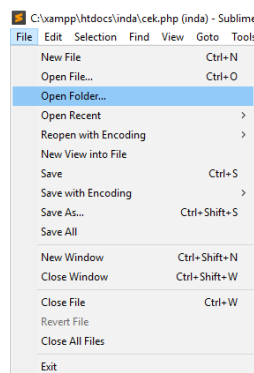


10. Lalu buat folder baru pada Htdocs, dengan cara buka Local Disk C, pilih Xampp, dan pilih folder Htdocs. Lalu buat folder baru seperti pada gambar dibawah ini!

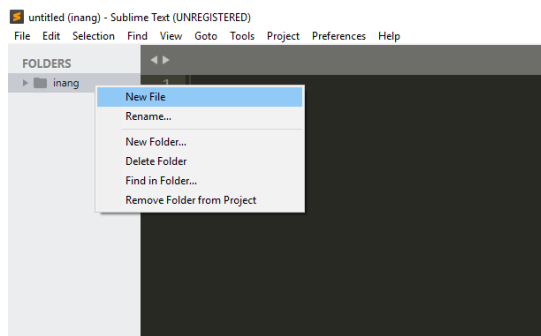


11. Lalu buka aplikasi Sublime Text.

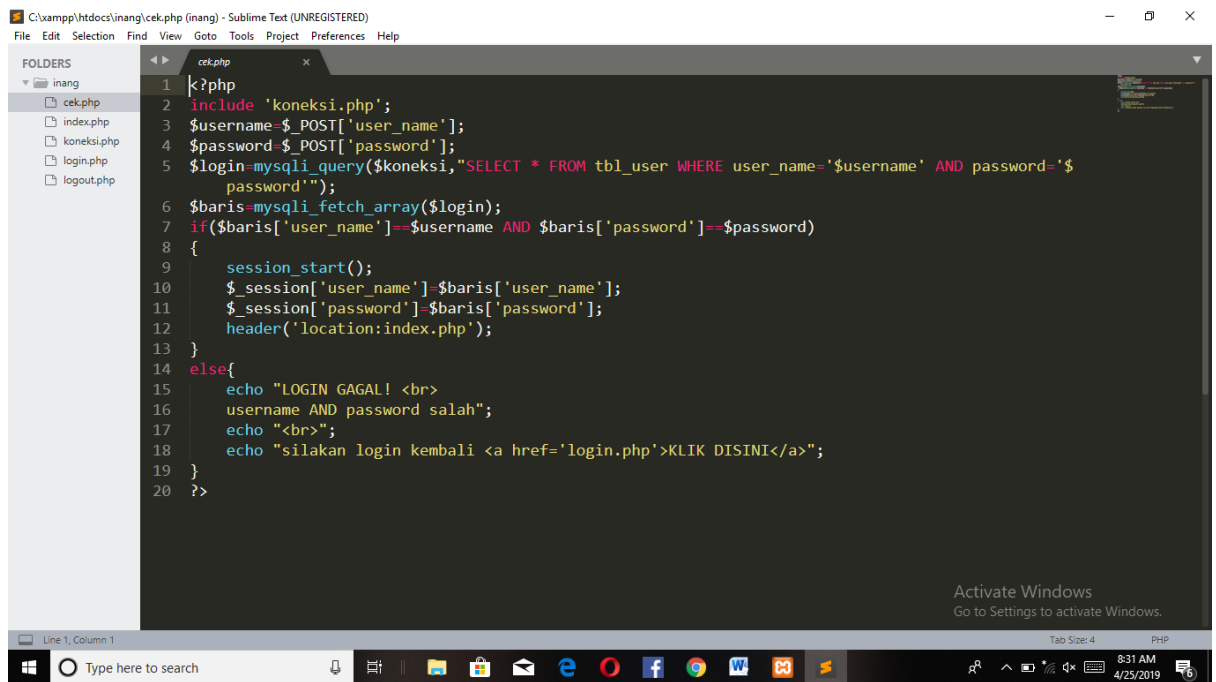
12. Lalu buka folder kosong yang telah dibuat tadi dengan cara pilih File>Open Folder. Seperti pada gambar di bawah ini!



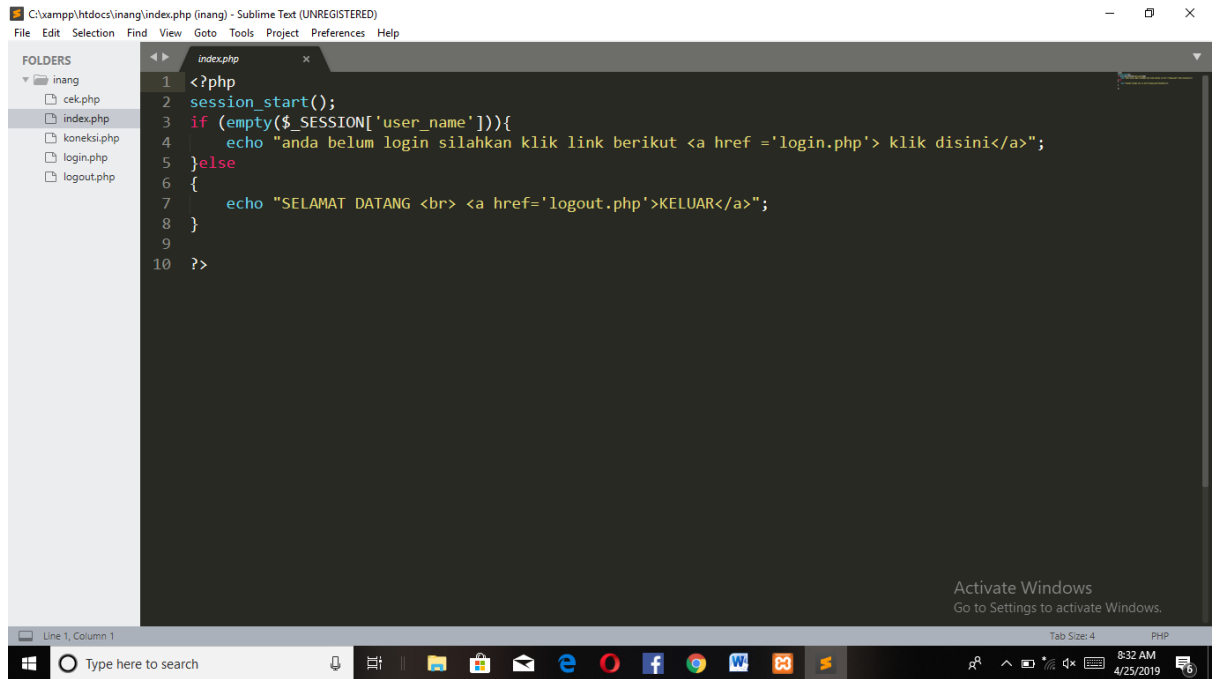
13. Setelah memilih folder kosong yang tadi kita buat, buat file baru (index.php, login.php, logout.php, cek.php, koneksi.php) dengan cara klik kanan dan pilih New File seperti pada gambar dibawah ini!



14. Setelah itu ketikkan coding pada file cek.php seperti pada gambar dibawah ini!

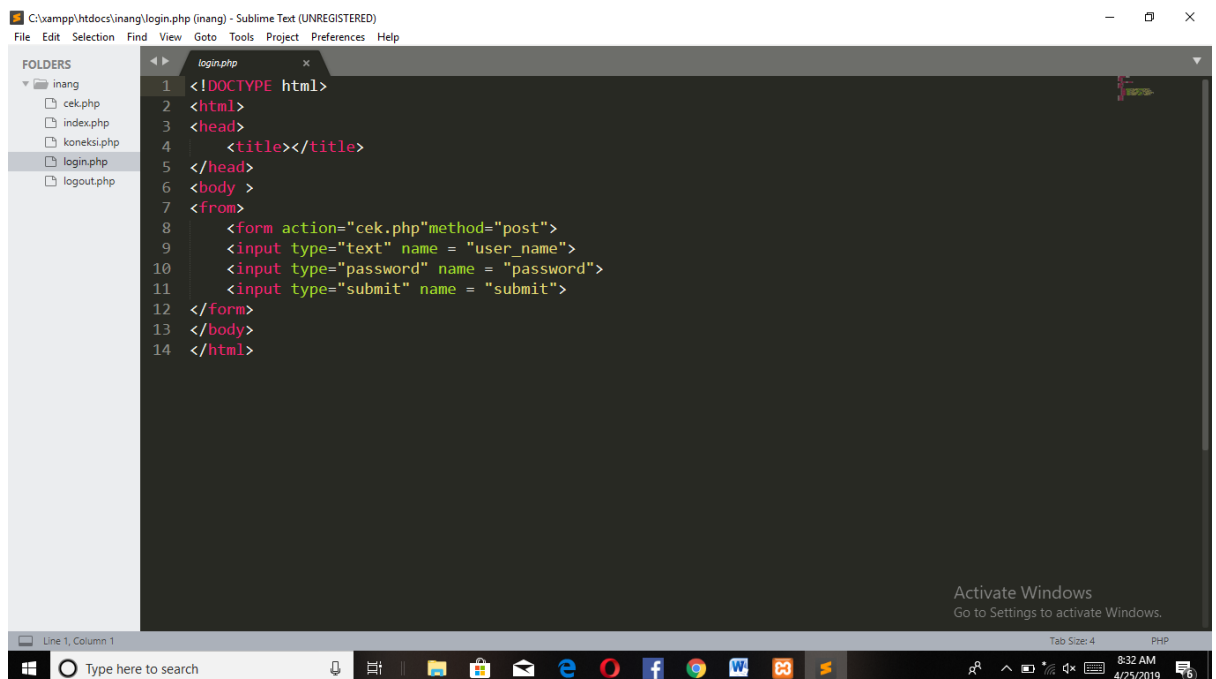


15. Lalu ketikkan coding pada file index.php seperti pada gambar dibawah ini



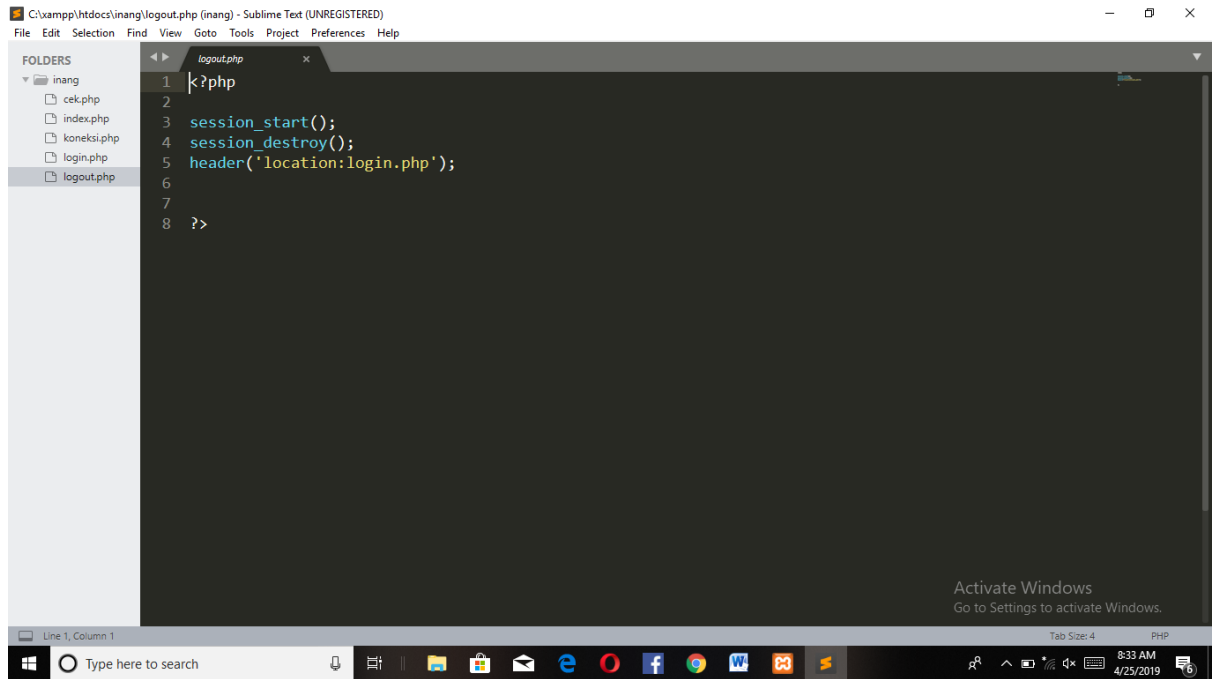
```
1 <?php
2 session_start();
3 if (empty($_SESSION['user_name'])) {
4     echo "anda belum login silahkan klik link berikut <a href='login.php'> klik disini</a>";
5 } else {
6     echo "SELAMAT DATANG <br> <a href='logout.php'>KELUAR</a>";
7 }
8 }
9
10 ?>
```

16. Ketikkan juga coding pada login.php seperti pada gambar dibawah ini!



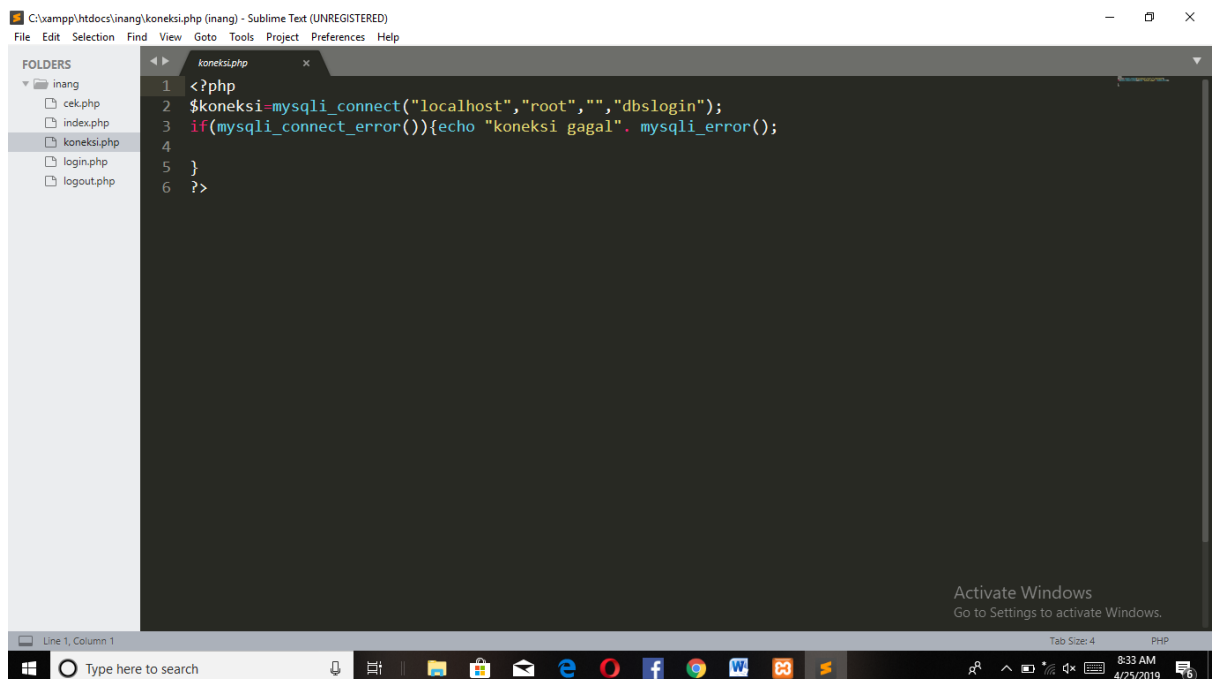
```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title></title>
5 </head>
6 <body>
7 <form>
8     <form action="cek.php" method="post">
9         <input type="text" name = "user_name">
10        <input type="password" name = "password">
11        <input type="submit" name = "submit">
12    </form>
13 </body>
14 </html>
```

17. Ketikkan coding pada file logout.php seperti pada gambar dibawah ini!



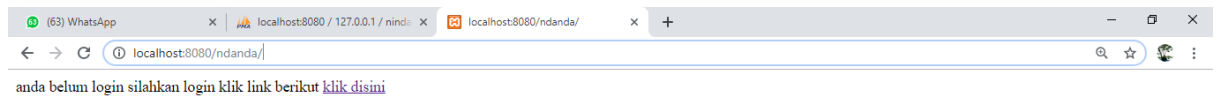
```
1 <?php
2
3 session_start();
4 session_destroy();
5 header('location:login.php');
6
7
8 ?>
```

18. Ketikkan juga coding pada file koneksi.php seperti pada gmabar dibawah ini!

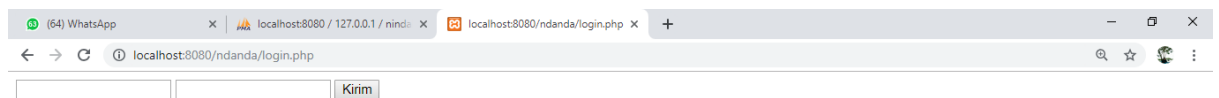


```
1 <?php
2 $koneksi=mysqli_connect("localhost","root","","dbslogin");
3 if(mysqli_connect_error()){echo "koneksi gagal". mysqli_error();
4
5 }
6 ?>
```

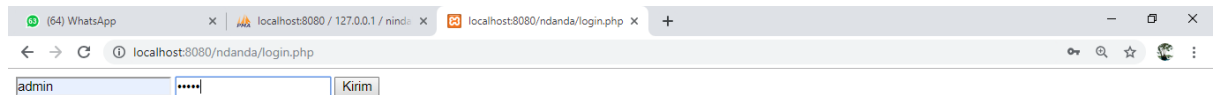
19. Setelah mengetikkan semua coding, buka browser dan ketikkan "localhost:8080/ndanda(nama folder yang telah dibuat). Setelah di enter, akan muncul tampilan seperti pada gambar di bawah ini! Setelah muncul tampilan di bawah ini, berarti coding sudah berhasil.



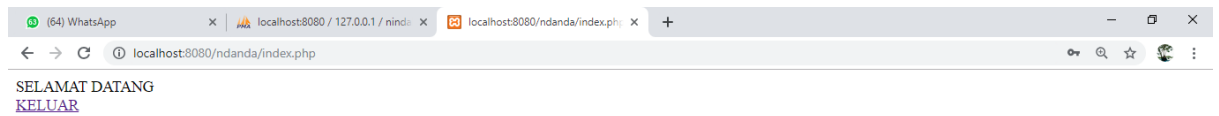
20. Setelah mengklik “klik disini”, akan muncul tampilan seperti pada gambar dibawah ini!



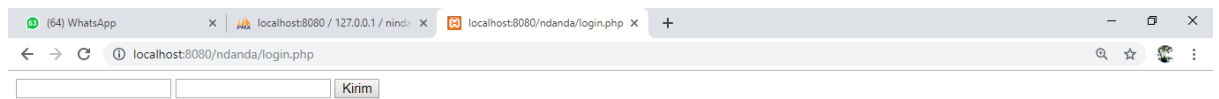
21. Lalu ketikkan username dan password yang telah dibuat sebelumnya. Seperti pada gambar dibawah ini!



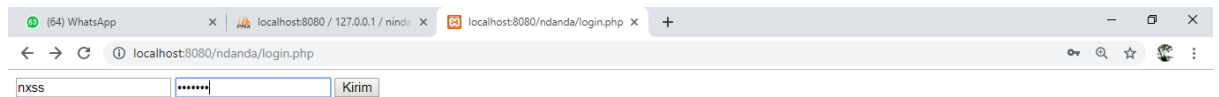
22. Setelah login, akan muncul tampilan seperti pada gambar di bawah ini! Ini berarti username dan password yang dimasukkan sudah benar.



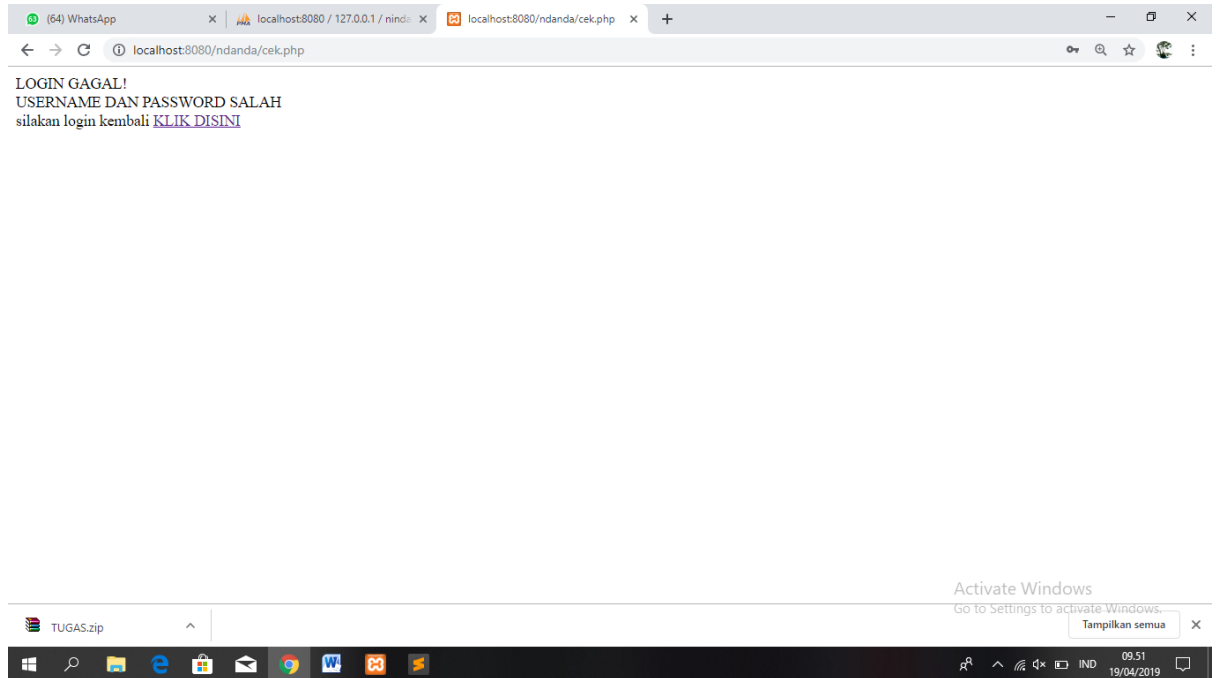
23. Apabila di klik “keluar” maka akan muncul tampilan seperti pada gambar dibawah ini!



24. Sekarang kita akan mencoba memasukkan username dan password yang salah. Seperti pada gambar dibawah ini!



25. Apabila username dan password yang dimasukkan salah, akan muncul tampilan seperti pada gambar dibawah ini!



Penjelasan coding dan fungsinya!

Penjelasan PHP

PHP singkatan dari Hypertext Processor yang merupakan bahasa pemrograman yang didesain khusus untuk web development atau pengembangan web.

Fungsi PHP Dalam Pemrograman Web

Untuk membuat halaman web, sebenarnya PHP bukanlah bahasa pemrograman yang wajib digunakan. Kita bisa saja membuat website hanya menggunakan HTML saja. Web yang dihasilkan dengan HTML (dan CSS) ini dikenal dengan website statis, dimana konten dan halaman web bersifat tetap.

Sebagai perbandingan, website dinamis yang bisa dibuat menggunakan PHP adalah situs web yang bisa menyesuaikan tampilan konten tergantung situasi. Website dinamis juga bisa menyimpan data ke dalam database, membuat halaman yang berubah-ubah sesuai input dari *user*, memproses form, dll.

Untuk pembuatan web, kode **PHP** biasanya di sisipkan kedalam dokumen HTML. Karena fitur inilah PHP disebut juga sebagai **Scripting Language** atau bahasa pemrograman **script**. Berikut adalah aturan dasar penulisan kode PHP:

1. Case Sensitivity (perbedaan huruf besar dan kecil) dalam PHP

PHP tidak membedakan huruf besar dan kecil (**case insensitive**) untuk **penamaan fungsi (function)**, nama **class**, maupun **keyword** bawaan PHP seperti **echo**, **while**, dan **class**.

Akan tetapi, PHP membedakan huruf besar dan huruf kecil (**case sensitive**) untuk **penamaan variabel**, sehingga **\$nama**, **\$Nama** dan **\$NAMA** akan dianggap sebagai 3 variabel yang berbeda. Sering kali *error* terjadi dikarenakan salah menuliskan nama variabel, yang seharusnya menggunakan huruf kecil, ditulis dengan huruf besar.

```
1  <?php
2  $andi="Andi";
3  echo $Andi; // Notice: Undefined variable: Andi
4  ?>
```

Untuk mengatasi perbedaan ini, disarankan menggunakan huruf kecil untuk seluruh kode PHP, termasuk **variabel**, **fungsi** maupun **class**. Jika membutuhkan nama variabel yang terdiri dari 2 kata, karakter spasi bisa digantikan dengan **underscore** (**_**)

2. Penulisan Baris Perintah dalam PHP

Statement (baris perintah) di dalam PHP adalah *kumpulan perintah PHP yang menginstruksikan PHP untuk melakukan sesuatu*. Baris perintah ini bisa terdiri dari satu baris singkat (seperti perintah **echo** untuk menampilkan text di layar) atau bisa sesuatu yang lebih rumit dan terdiri dari beberapa baris, seperti **kondisi if**, atau kode perulangan (**loop**). Berikut adalah contoh beberapa baris perintah dalam PHP:

```

1    <?php
2        echo "Hello, world";
3        sebuah_fungsi(21, "duniaikom");
4        $a = 1;
5        $nama = "duniaikom";
6        $b = $a / 25.0;
7        if ($y == $z) {
8            echo "Tampilkan Tabel";
9        }
10    ?>

```

Terlihat dari beberapa contoh baris perintah diatas, PHP menggunakan tanda **semicolon** (titik koma) “;” sebagai tanda akhir baris perintah.

Kumpulan baris perintah yang menggunakan tanda kurung kurawal seperti **kodisi IF** atau **perulangan (loop)** tidak membutuhkan tanda titik koma setelah kurung penutup.

```

1    <?php
2        if (true) {
3            echo "Perintah dijalankan"; // tanda titik koma harus ditulis
4        } // tidak diperlukan tanda titik koma setelah tanda kurung kurawal
5    ?>

```

3. Karakter Spasi dan Tab dalam PHP

Secara umum, karakter **spasi** dan **tab** diabaikan di dalam eksekusi program PHP. Anda boleh memecah sebuah *statement* menjadi beberapa baris, atau menyatukan beberapa *statement* dalam sebuah baris yang panjang. Seperti contoh berikut:

```

1    <?php
2        echo "Ini kalimat pertama"; echo "Ini kalimat kedua"; $nama="duniaikom";
3    ?>

```

Baris perintah itu sama artinya dengan

```
1  <?php
2      echo "Ini kalimat pertama";
3      echo "Ini kalimat kedua";
4      $nama = "duniaikom";
5  ?>
```

Fungsi echo()

Fungsi echo() adalah fungsi untuk menampilkan teks ke layar. Fungsi ini dapat digunakan dengan tanda kurung maupun tanpa tanda kurung.

Fungsi echo() tidak akan mengembalikan apa-apa setelah dieksekusi. Dia hanya bertugas menampilkan teks saja.

Fungsi **mysql_fetch_array()** merupakan salah satu fungsi yang banyak digunakan dalam proses pengambilan data MySQL. Fungsi ini akan menangkap data dari hasil perintah query dan membentuknya ke dalam array asosiatif dan array numerik. Jadi, fungsi mysql_fetch_array() menawarkan kemudahan dan fleksibilitas dalam mengakses data. Namun demikian, penggunaan fungsi ini dianggap kurang efektif dari sisi performa karena terlalu “boros” penyimpanan.

Session_start();

Kegunaan dari fungsi session_start(); adalah untuk memulai eksekusi session pada server dan kemudian menyimpannya pada browser. Dengan intruksi ini menunjukkan pada saat session di mulai, semua session akan diaftarkan selanjutnya akan disimpan. Penyimpanan ini terjadi sampai jangka waktu tertentu. Dan untuk membaut session berdasarkan waktu bisa Anda setting sendiri. Mudah-mudahan nanti saya buat tutorialnya dengan batas waktu session. Selain dengan batas waktu, session bisa hilang jika cookies browser Anda hilangkan. Dengan dihilangkan, maka secara otomatis session juga hilang dari browser Anda.

\$_SESSION[username]

Jika pada no adalah mendaftarkan sebuah session. Berbeda dengan yang ketiga ini. Variabel ini adalah sebuah inisialisasi dari session.

Session_destroy();

Pada bagian ini menjelaskan bahwa semua session yang telah di buat dan di inisialisasi akan di destroy atau di hancurkan. Nah pada tahanan inilah yang sering digunakan untuk script logout pada web atau aplikasi berbasis web lainnya.

Untuk menampilkan pesan kesalahan dari MySQL, kita harus memanggilnya menggunakan fungsi **mysql_errno()** dan **mysql_error()**.