|  |  |
| --- | --- |
| Mata Kuliah | **:** Pemrograman Web (PW) |
| Program Studi | **:** D4 – Teknik Informatika **/** D4 – Sistem Informasi Bisnis |
| Semester | **:** 3 (tiga) |
| Pertemuan ke- | **:** 13 (tiga belas) |

<https://github.com/safrizalrahman46/dasarweb.git>  
SAFRIZAL RAHMAN\_19\_SIB\_2G

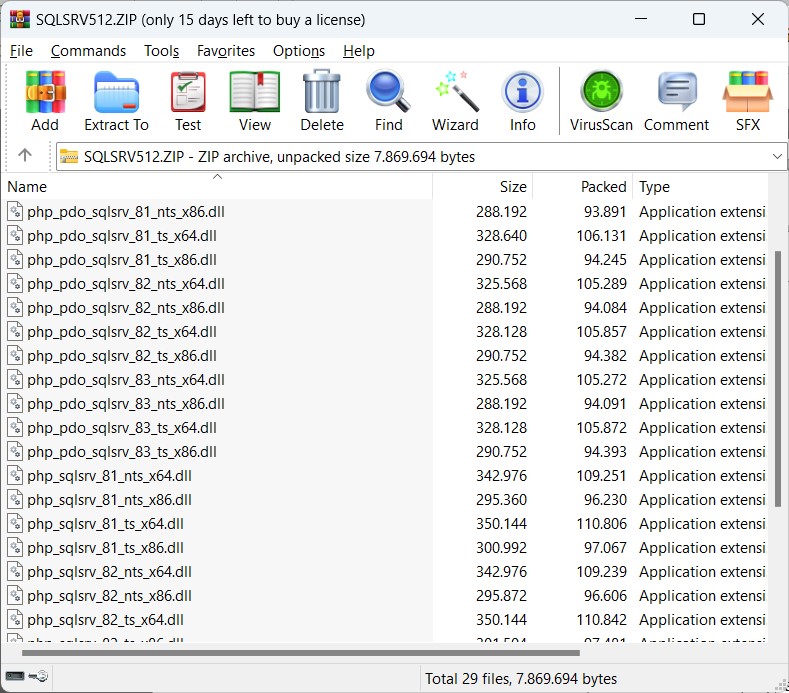
**JOBSHEET 13**

**Koneksi PHP ke SQL Server (Tambahan)**

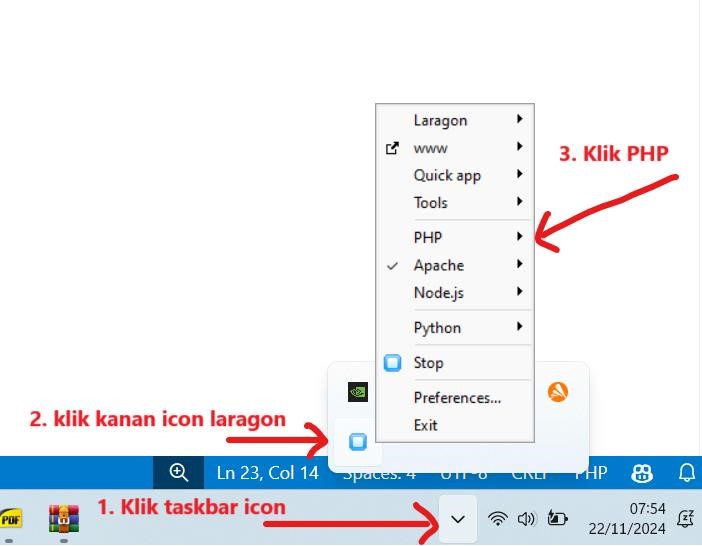
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **INFO**    Materi ini adalah materi tambahan dari Jobsheet 13 untuk menghubungkan PHP dengan Database SQL Server.    Jangan melakukan konversi materi PDF ini ke Docx, karena kode program hasil konversi ke Docx bisa tidak sama dengan kode program di materi PDF ini. | | | | |
|  | Untuk tabel contoh, kita bisa menggunakan DB dasar\_web.sql (MySQL) yang sudah | | |  |
|  | |  | | --- | | dikonversi ke SQL Server | |  |

**Praktikum 1 – Instalasi Ekstensi SQL Server pada PHP:**

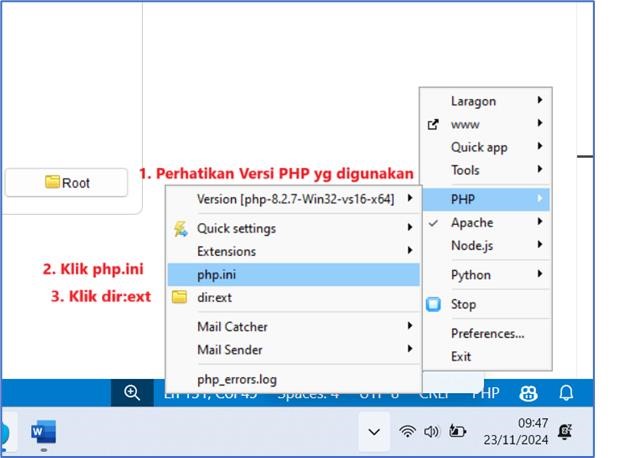
1. Kita download ekstensi **Microsoft Drivers for PHP for SQL Server** disini <https://go.microsoft.com/fwlink/?linkid=2258816>
2. Kemudian kita buka file SQLSRV512.zip yang telah kita download tadi



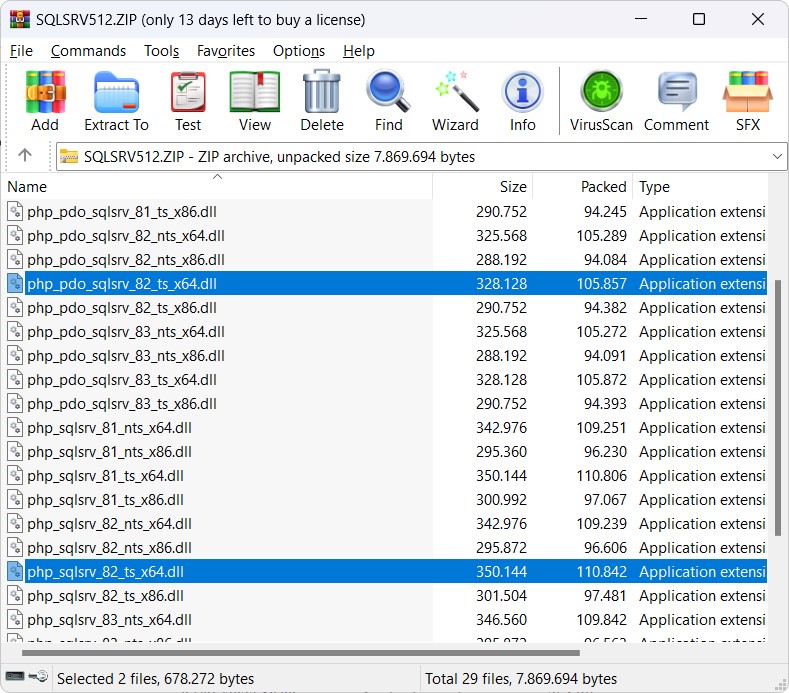
1. Setelah itu, kita cek dulu versi PHP yang kita gunakan di laragon



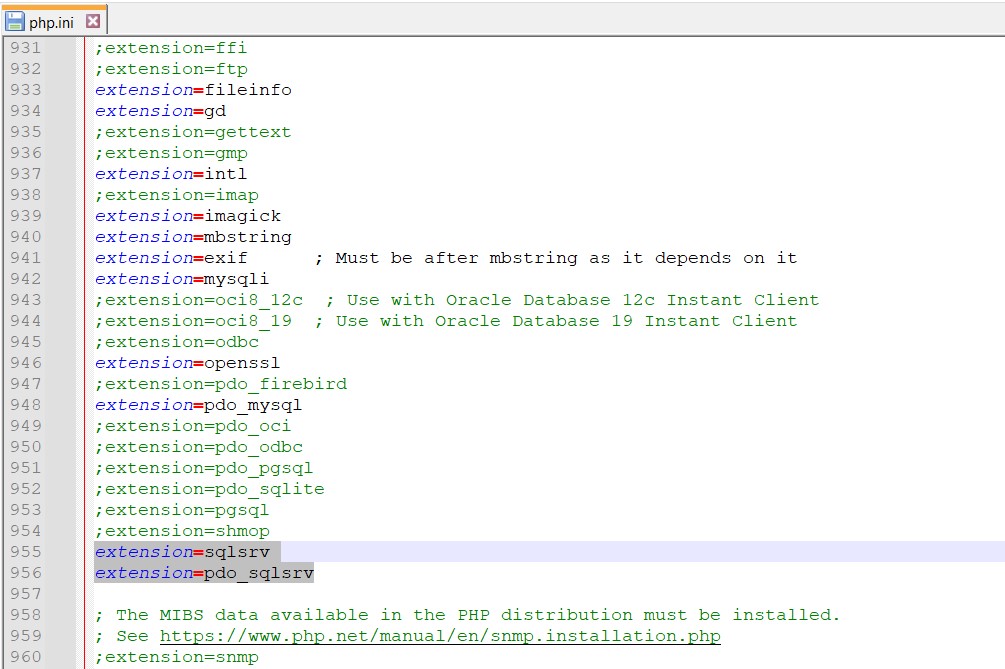
1. Kemudian akan muncul tampilan seperti berikut



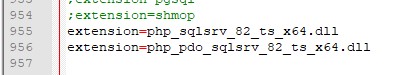
1. Dari gambar tahap 4 kita mengetahui bahwa versi PHP yang kita gunakan adalah versi PHP 8.2.7. Sekarang kita copy ekstensi PHP SQL Server sesuai versi PHP ke direktori ekstensi PHP di laragon di C:\laragon\bin\php\php-8.2.7-Win32-vs16-x64\ext (direktori bisa berbeda, sesuai dari masing-masing laptop dan versi PHP).
2. Kita pilih php\_sqlsrv\_82\_ts\_x64.dll dan php\_pdo\_sqlsrv\_82\_ts\_x64.dll yang akan kita masukkan ke ekstensi PHP.

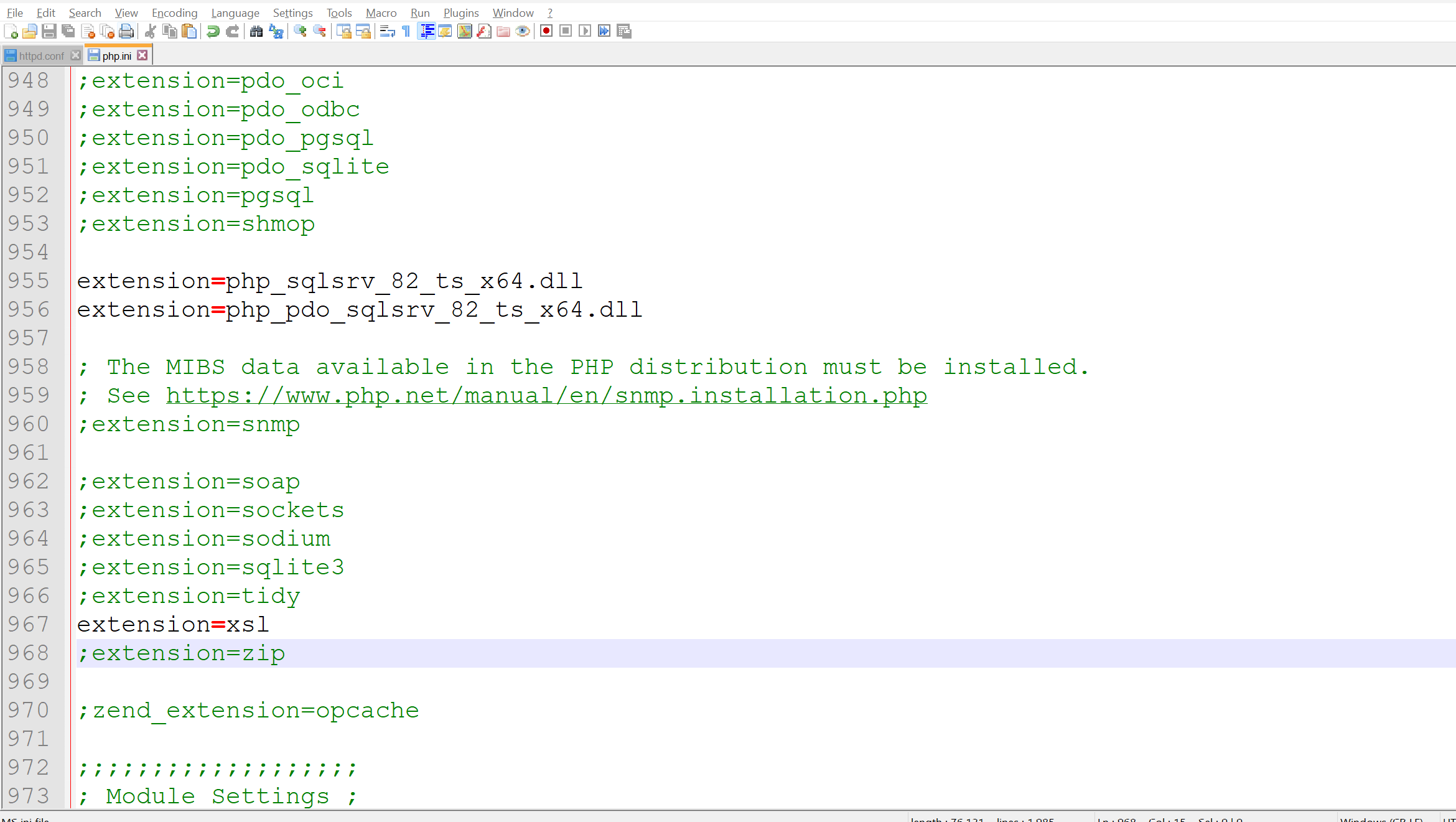


1. Kemudian kita ke file php.ini yang sudah kita buka pada tahap 4 tadi. Kemudian kita tambahkan load library ekstensi PHP SQL Server tadi pada baris 955 (atau pada group load ekstensi lain di php.ini)



\*catatan: kalau masih tidak bisa, bisa menuliskan nama file secara utuh seperti





1. Setelah itu, kita restart laragon kita. Selesai

**Praktikum 2 – Cek Koneksi ke SQL Server dengan PHP:**

* 1. Seleteh kita berhasil melakukan proses intalasi ekstensi PHP SQL Server, kita buat koneksi PHP ke SQL Server
  2. Kemudian kita modifiksi file Connection.php pada

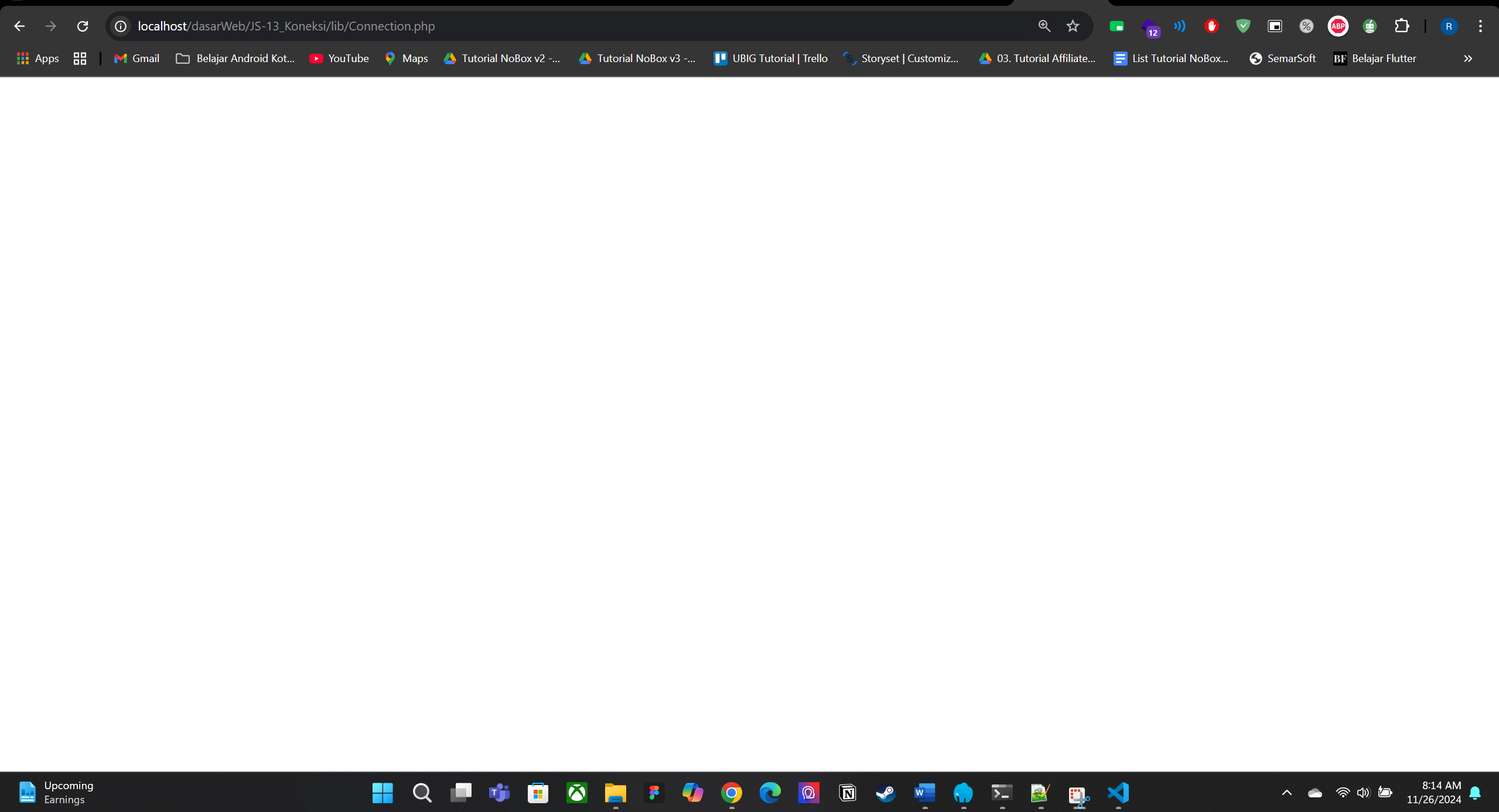
C:\laragon\www\dasarWeb\JS13\lib\Connection.php, seperti berikut

|  |
| --- |
| <?php  $use\_driver = 'sqlsrv'; // mysql atau sqlsrv    $host = 'localhost';  $username = ''; //'sa';  $password = '';  $database = 'master';  $db;    if($use\_driver == 'mysql'){ try{  $db = new mysqli('localhost', $username, $password, $database); if($db->connect\_error){  die('Connection DB failed: ' . $db->connect\_error);  }  }catch(Exception $e){ die($e->getMessage());  }  } else if($use\_driver == 'sqlsrv'){  $credential = [  'Database' => $database,  'UID' => $username,  'PWD' => $password  ]; try{  $db = sqlsrv\_connect($host, $credential);    if (!$db){  $msg = sqlsrv\_errors(); die($msg[0]['message']);  }  }catch(Exception $e){ die($e->getMessage());  }  } |

Kode di atas memungkinkan kita untuk mengubah koneksi ke database MySQL atau SQL Server, sesuai dengan konfigurasi.

* 1. Silahkan kalian sesuaikan dengan konfigurasi dengan SQL Server kemudian silahkan dicoba untuk mengakses file Connection.php dari browser

<http://localhost/dasarWeb/JS13/lib/Connection.php>

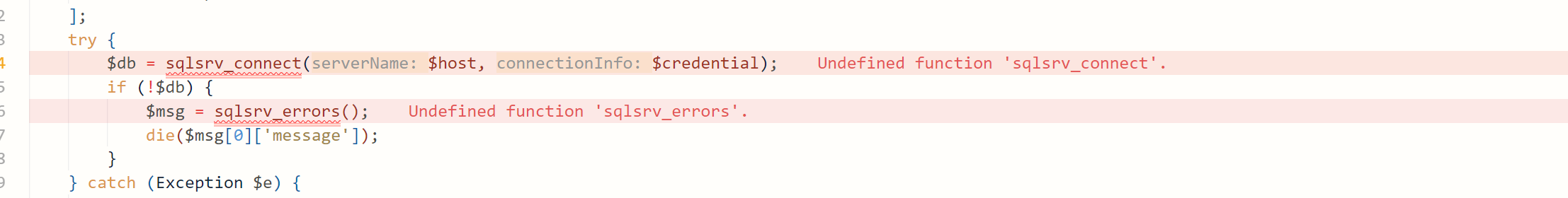


extension=sqlsrv

extension=pdo\_sqlsrv

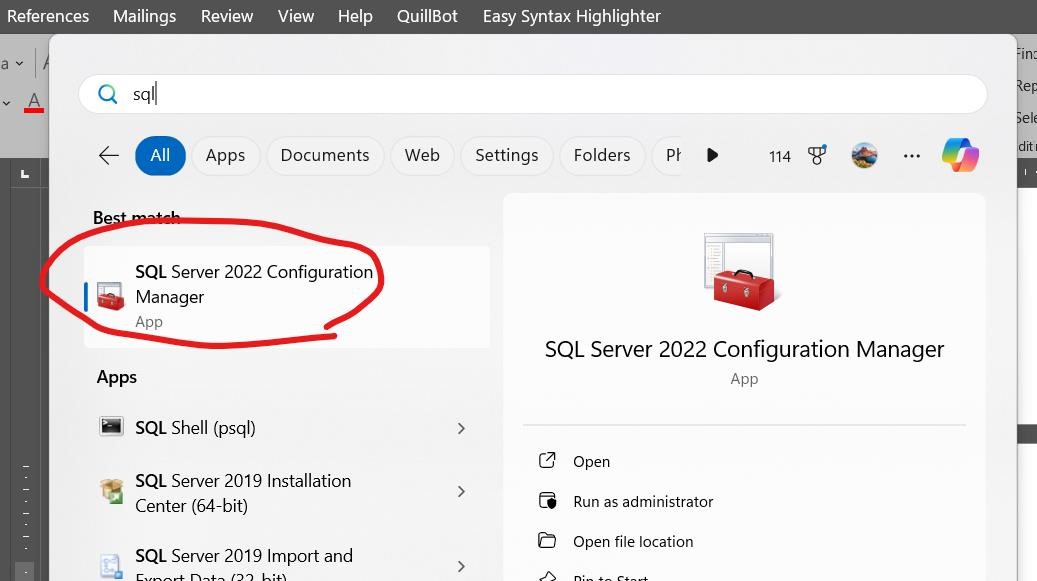
extension=php\_sqlsrv.dll

extension=php\_pdo\_sqlsrv.dll



jika tampilan browser adalah ***blank***, maka koneksi kita sudah sesuai dan **berhasil.** Praktikum 2 selesai dan lanjut ke praktikum 3**.**

* 1. Jika koneksi masih tidak berhasil, maka perlu kita ganti konfigurasinya.
  2. Kita buka SQL Server Configuration Manager pada Start Windows



* 1. Setelah terbuka, kita pilih sesuai dengan gambar

7.

Selanjutnya, kita pilih tab

Service

, dan kita lihat nilai dari

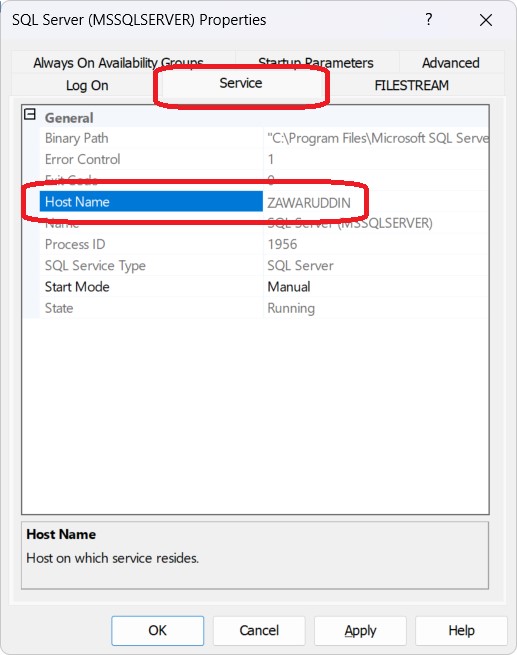
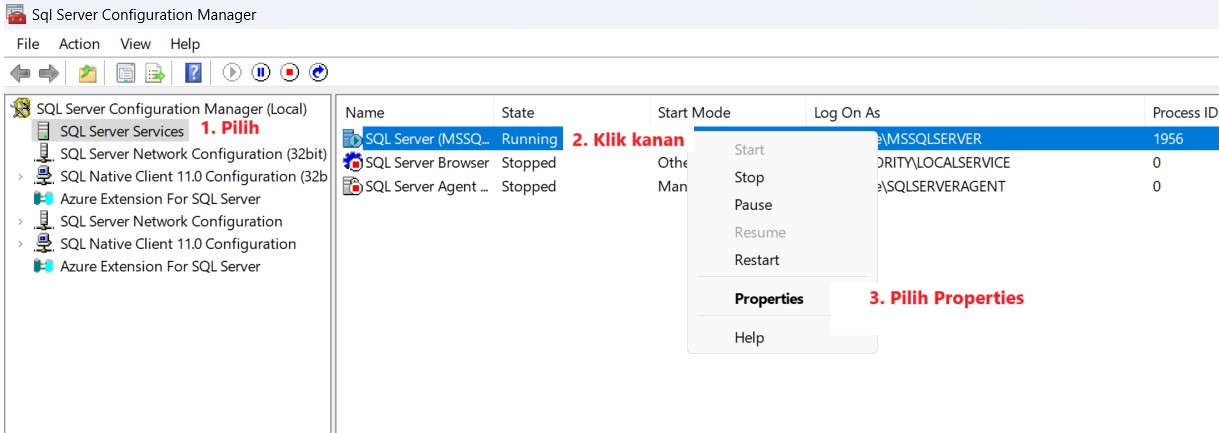
Host

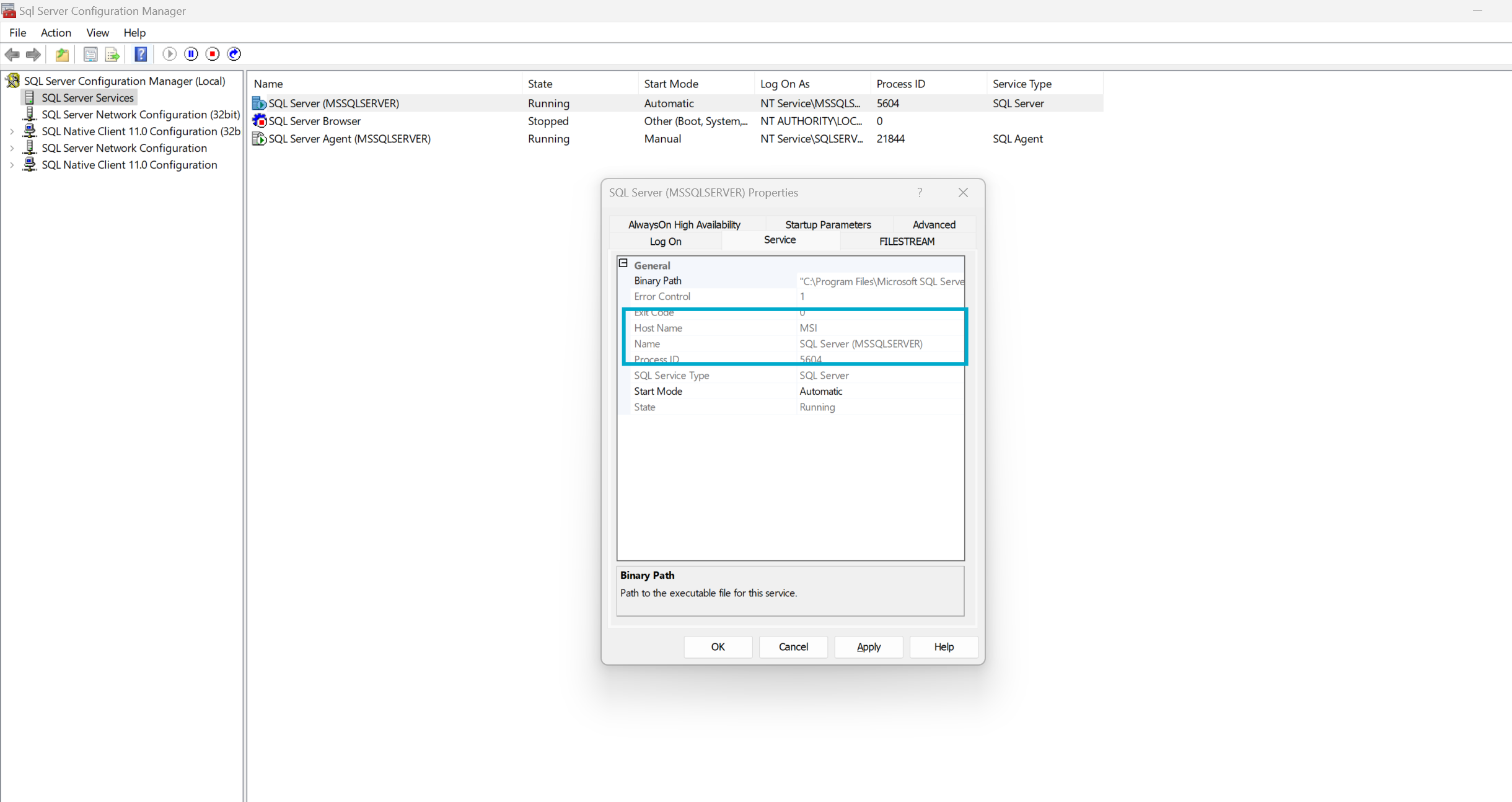
N

ame

-

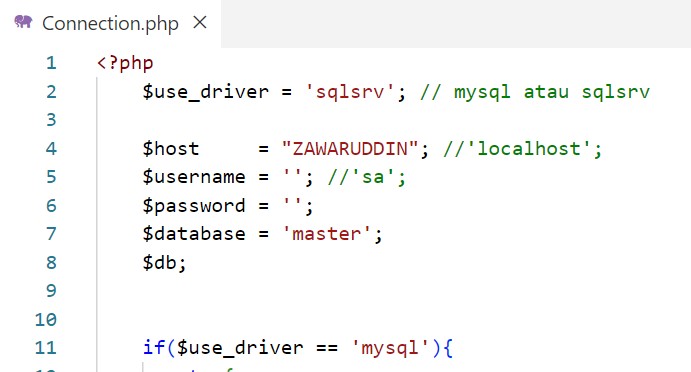
nya

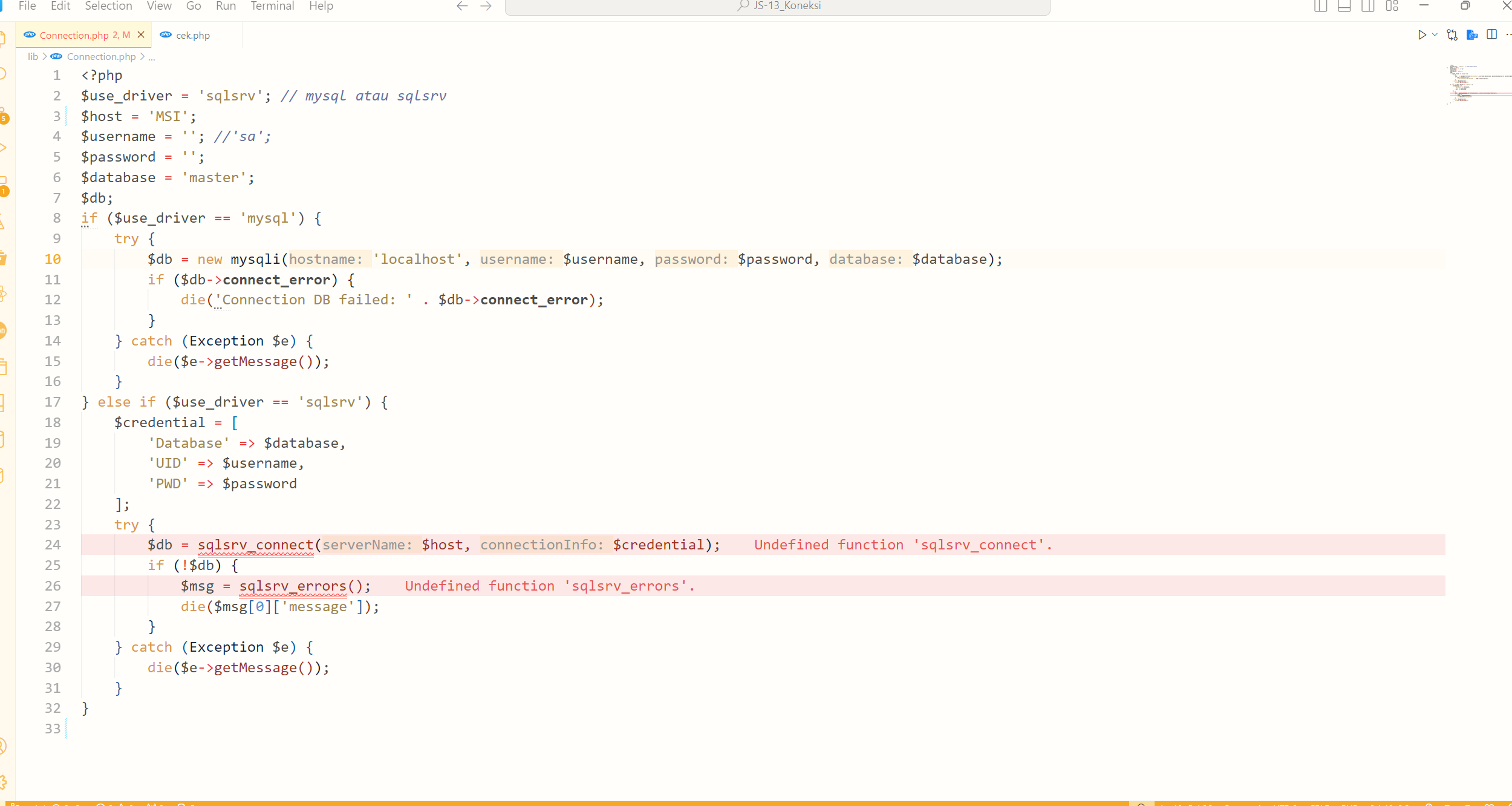




1. Host Name tersebut yang akan kita gunakan untuk konfigurasi $host pada file

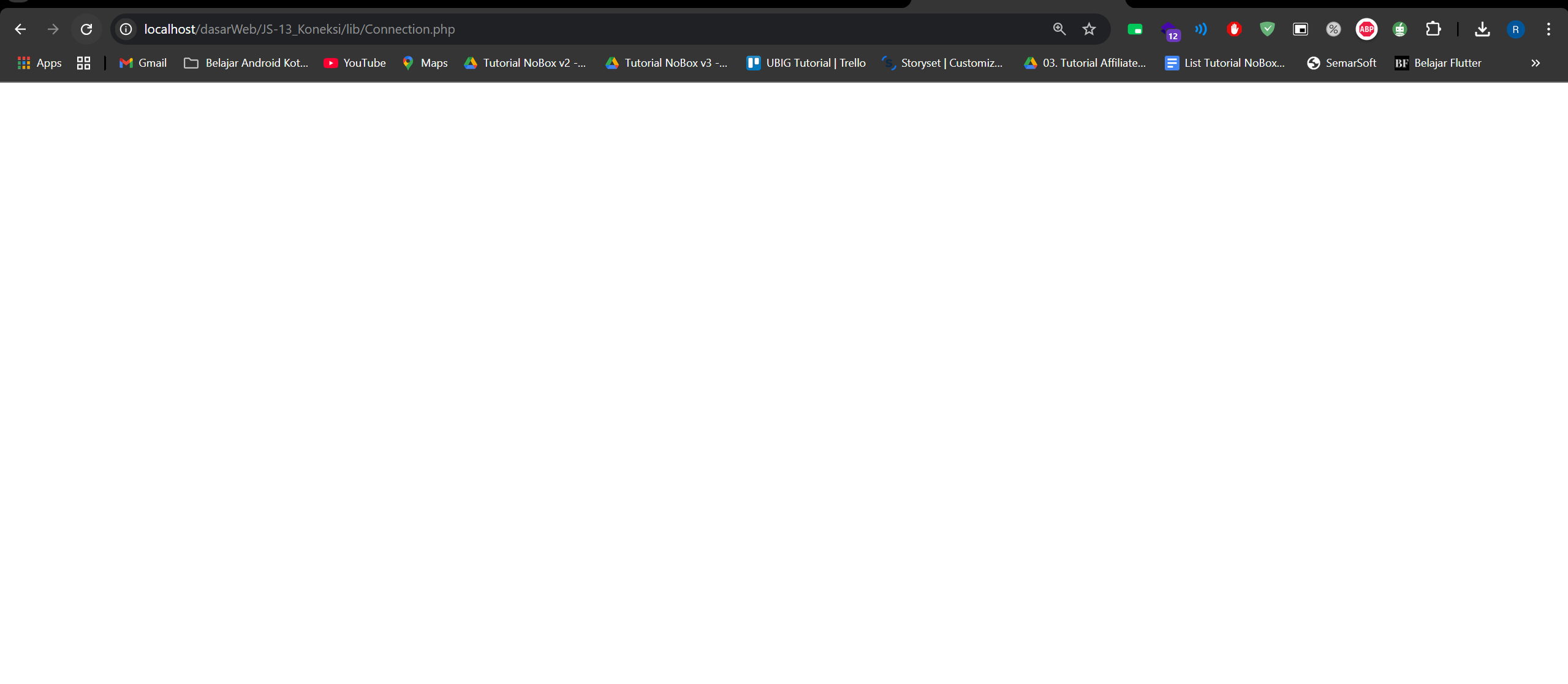
Connection.php kita





1. Sekarang kita coba lagi untuk cek koneksi ke DB SQL Server dengan mengakses

<http://localhost/dasarWeb/JS13/lib/Connection.php>



**Praktikum 3 – Modifikasi Source Code:**

1. Untuk bisa mencoba proses CRUD (Create, Read, Update, Delete) pada database SQL Server dengan PHP, kita perlu memodifikasi Source Code dari pertemuan 13 kemarin.
2. Pertama kita buat file baru di

C:\laragon\www\dasarWeb\JS13\model\UserModel.php

|  |
| --- |
| <?php  include('Model.php');    class UserModel extends Model{ protected $db;  protected $table = 'm\_user'; protected $driver;    public function \_\_construct(){ include('../lib/Connection.php');  $this->db = $db;  $this->driver = $use\_driver;  }  public function insertData($data){ if($this->driver == 'mysql'){  // prepare statement untuk query insert  $query = $this->db->prepare("insert into {$this->table} (username, nama, level, password) values(?,?,?,?)");  // binding parameter ke query, "s" berarti string, "ss" berarti dua string $query->bind\_param('ssss', $data['username'], $data['nama'], $data['level'], password\_hash($data['password'], PASSWORD\_DEFAULT));    // eksekusi query untuk menyimpan ke database  $query->execute();  } else {  // eksekusi query untuk menyimpan ke database  sqlsrv\_query($this->db, "insert into {$this->table} (username, nama, level, password) values(?,?,?,?)", array($data['username'], $data['nama'], $data['level'], password\_hash($data['password'], PASSWORD\_DEFAULT)));  }  } |

|  |
| --- |
| public function getData(){ if($this->driver == 'mysql'){  // query untuk mengambil data dari tabel  return $this->db->query("select \* from {$this->table} ")-  >fetch\_all(MYSQLI\_ASSOC);  } else {  // query untuk mengambil data dari tabel  $query = sqlsrv\_query($this->db, "select \* from {$this->table}");    $data = [];  while($row = sqlsrv\_fetch\_array($query, SQLSRV\_FETCH\_ASSOC)){  $data[] = $row;  }    return $data;  }  } public function getDataById($id){ if($this->driver == 'mysql'){  // query untuk mengambil data berdasarkan id  $query = $this->db->prepare("select \* from {$this->table} where user\_id = ?");  // binding parameter ke query "i" berarti integer. Biar tidak kena SQL Injection $query->bind\_param('i', $id);    // eksekusi query  $query->execute();    // ambil hasil query  return $query->get\_result()->fetch\_assoc();  } else {  // query untuk mengambil data berdasarkan id  $query = sqlsrv\_query($this->db, "select \* from {$this->table} where user\_id =  ?", [$id]);    // ambil hasil query  return sqlsrv\_fetch\_array($query, SQLSRV\_FETCH\_ASSOC);  }  } public function updateData($id, $data){ if($this->driver == 'mysql'){ // query untuk update data  $query = $this->db->prepare("update {$this->table} set username = ?, nama = ?, level = ?, password = ? where user\_id = ?");    // binding parameter ke query  $query->bind\_param('ssssi', $data['username'], $data['nama'], $data['level'], password\_hash($data['password'], PASSWORD\_DEFAULT), $id);  // eksekusi query  $query->execute();  } else {  // query untuk update data  sqlsrv\_query($this->db, "update {$this->table} set username = ?, nama = ?, level = ?, password = ? where user\_id = ?", [$data['username'], $data['nama'], $data['level'], password\_hash($data['password'], PASSWORD\_DEFAULT), $id]);  }  }  public function deleteData($id){ if($this->driver == 'mysql'){ // query untuk delete data  $query = $this->db->prepare("delete from {$this->table} where user\_id = ?"); |
| // binding parameter ke query  $query->bind\_param('i', $id);    // eksekusi query  $query->execute();  } else {  // query untuk delete data  sqlsrv\_query($this->db, "delete from {$this->table} where user\_id = ?", [$id]);  }  }  public function getSingleDataByKeyword($column, $keyword){ if($this->driver == 'mysql'){  // query untuk mengambil data berdasarkan id  $query = $this->db->prepare("select \* from {$this->table} where {$column} = ?");    // binding parameter ke query "i" berarti integer. Biar tidak kena SQL Injection $query->bind\_param('s', $keyword);    // eksekusi query  $query->execute();    return $query->get\_result()->fetch\_assoc();  } else {  // query untuk mengambil data berdasarkan id  $query = sqlsrv\_query($this->db, "select \* from {$this->table} where {$column} =  ?", [$keyword]);    // ambil hasil query  return sqlsrv\_fetch\_array($query, SQLSRV\_FETCH\_ASSOC);  }  }  } |

1. Selanjutnya, kita modifikasi KategoriModel.php seperti berikut

|  |
| --- |
| <?php  include('Model.php');    class KategoriModel extends Model{ protected $db;  protected $table = 'm\_kategori'; protected $driver;    public function \_\_construct(){  include\_once('../lib/Connection.php');    $this->db = $db;  $this->driver = $use\_driver;  } public function insertData($data){ if($this->driver == 'mysql'){  // prepare statement untuk query insert  $query = $this->db->prepare("insert into {$this->table} (kategori\_kode, kategori\_nama) values(?,?)");  // binding parameter ke query, "s" berarti string, "ss" berarti dua string  $query->bind\_param('ss', $data['kategori\_kode'], $data['kategori\_nama']);  // eksekusi query untuk menyimpan ke database  $query->execute();  } else {  // eksekusi query untuk menyimpan ke database |

|  |
| --- |
| sqlsrv\_query($this->db, "insert into {$this->table} (kategori\_kode, kategori\_nama) values(?,?)", array($data['kategori\_kode'], $data['kategori\_nama']));  }  }  public function getData(){ if($this->driver == 'mysql'){  // query untuk mengambil data dari tabel  return $this->db->query("select \* from {$this->table} ")-  >fetch\_all(MYSQLI\_ASSOC);  } else {  // query untuk mengambil data dari tabel  $query = sqlsrv\_query($this->db, "select \* from {$this->table}");    $data = [];  while($row = sqlsrv\_fetch\_array($query, SQLSRV\_FETCH\_ASSOC)){  $data[] = $row;  }    return $data;  }  } public function getDataById($id){ if($this->driver == 'mysql'){  // query untuk mengambil data berdasarkan id  $query = $this->db->prepare("select \* from {$this->table} where kategori\_id = ?");    // binding parameter ke query "i" berarti integer. Biar tidak kena SQL Injection $query->bind\_param('i', $id);    // eksekusi query  $query->execute();    // ambil hasil query  return $query->get\_result()->fetch\_assoc();  } else {  // query untuk mengambil data berdasarkan id  $query = sqlsrv\_query($this->db, "select \* from {$this->table} where kategori\_id  = ?", [$id]);    // ambil hasil query  return sqlsrv\_fetch\_array($query, SQLSRV\_FETCH\_ASSOC);  }  }  public function updateData($id, $data){ if($this->driver == 'mysql'){ // query untuk update data  $query = $this->db->prepare("update {$this->table} set kategori\_kode = ?, kategori\_nama = ? where kategori\_id = ?");    // binding parameter ke query  $query->bind\_param('ssi', $data['kategori\_kode'], $data['kategori\_nama'], $id);    // eksekusi query  $query->execute();  } else {  // query untuk update data  sqlsrv\_query($this->db, "update {$this->table} set kategori\_kode = ?,  kategori\_nama = ? where kategori\_id = ?", [$data['kategori\_kode'], $data['kategori\_nama'],  $id]);  }  } |

|  |
| --- |
| public function deleteData($id){ if($this->driver == 'mysql'){ // query untuk delete data  $query = $this->db->prepare("delete from {$this->table} where kategori\_id = ?");  // binding parameter ke query  $query->bind\_param('i', $id);    // eksekusi query  $query->execute();  } else {  // query untuk delete data  sqlsrv\_query($this->db, "delete from {$this->table} where kategori\_id = ?",  [$id]);  }  }  } |

1. Kemudian kita modifikasi file C:\laragon\www\dasarWeb\JS13\action\auth.php kita seperti berikut

|  |
| --- |
| <?php  include('../lib/Session.php'); include('../lib/Connection.php');    $session = new Session();    $act = isset($\_GET['act'])? strtolower($\_GET['act']) : '';    if($act == 'login'){    $username = $\_POST['username'];  $password = $\_POST['password'];  include('../model/UserModel.php');  // digunakan untuk query user  $user = new UserModel();  $data = $user->getSingleDataByKeyword('username', $username);  // jika password sesuai  if(password\_verify($password, $data['password'])){  $session->set('is\_login', true);  $session->set('username', $data['username']);  $session->set('name', $data['nama']);  $session->set('level', $data['level']);  $session->commit();  header('Location: ../index.php', false); }else{  $session->setFlash('status', false);  $session->setFlash('message', 'Username dan password salah.');  $session->commit();  header('Location: ../login.php', false);  }  }else if($act == 'logout'){    $session->deleteAll();    header('Location: ../login.php', false);  } |

# Pertanyaan 1

Kenapa kita perlu memodifikasi file auth.php? apa perbedaan dengan yang sebelumnya?

# Pertanyaan 2

Kenapa kita perlu menambahkan file baru UserModel.php? apa fungsinya?

1. Yang terakhir, kita modifikasi file

C:\laragon\www\dasarWeb\JS13\action\kategoriAction.php seperti berikut

|  |
| --- |
| <?php  include('../lib/Session.php');    $session = new Session();  if($session->get('is\_login') !== true){ header('Location: login.php');  }  include\_once('../model/KategoriModel.php'); include\_once('../lib/Secure.php');    $act = isset($\_GET['act'])? strtolower($\_GET['act']) : '';    if($act == 'load'){  $kategori = new KategoriModel();  $data = $kategori->getData();  $result = []; $i = 1;  foreach($data as $row){ $result['data'][] = [  $i,  $row['kategori\_kode'],  $row['kategori\_nama'],  '<button class="btn btn-sm btn-warning"  onclick="editData('.$row['kategori\_id'].')"><i class="fa fa-edit"></i></button>  <button class="btn btn-sm btn-danger"  onclick="deleteData('.$row['kategori\_id'].')"><i class="fa fa-trash"></i></button>'  ];  $i++;  }  echo json\_encode($result);  }  if($act == 'get'){  $id = (isset($\_GET['id']) && ctype\_digit($\_GET['id']))? $\_GET['id'] : 0;  $kategori = new KategoriModel(); $data = $kategori->getDataById($id); echo json\_encode($data);  } if($act == 'save'){  $data = [  'kategori\_kode' => antiSqlInjection($\_POST['kategori\_kode']),  'kategori\_nama' => antiSqlInjection($\_POST['kategori\_nama']) ]; |
| $kategori = new KategoriModel();  $kategori->insertData($data);  echo json\_encode([ 'status' => true,  'message' => 'Data berhasil disimpan.'  ]);  }  if($act == 'update'){  $id = (isset($\_GET['id']) && ctype\_digit($\_GET['id']))? $\_GET['id'] : 0; $data = [  'kategori\_kode' => antiSqlInjection($\_POST['kategori\_kode']),  'kategori\_nama' => antiSqlInjection($\_POST['kategori\_nama']) ];    $kategori = new KategoriModel();  $kategori->updateData($id, $data);  echo json\_encode([ 'status' => true,  'message' => 'Data berhasil diupdate.'  ]);  }  if($act == 'delete'){  $id = (isset($\_GET['id']) && ctype\_digit($\_GET['id']))? $\_GET['id'] : 0;  $kategori = new KategoriModel();  $kategori->deleteData($id);  echo json\_encode([ 'status' => true,  'message' => 'Data berhasil dihapus.'  ]);  } |

1. Selesai, sekarang kita coba testing aplikasi yang sudah kita modifikasi di http://localhost/dasarWeb/JS13

# Pertanyaan 3

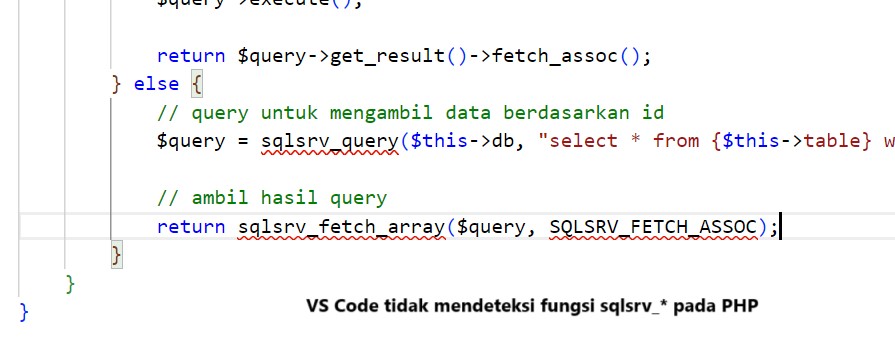
Apa perbedaan dengan kode program sebelumnya untuk KategoriModel.php dan kategoriAction.php? Jelaskan.

# Pertanyaan 4

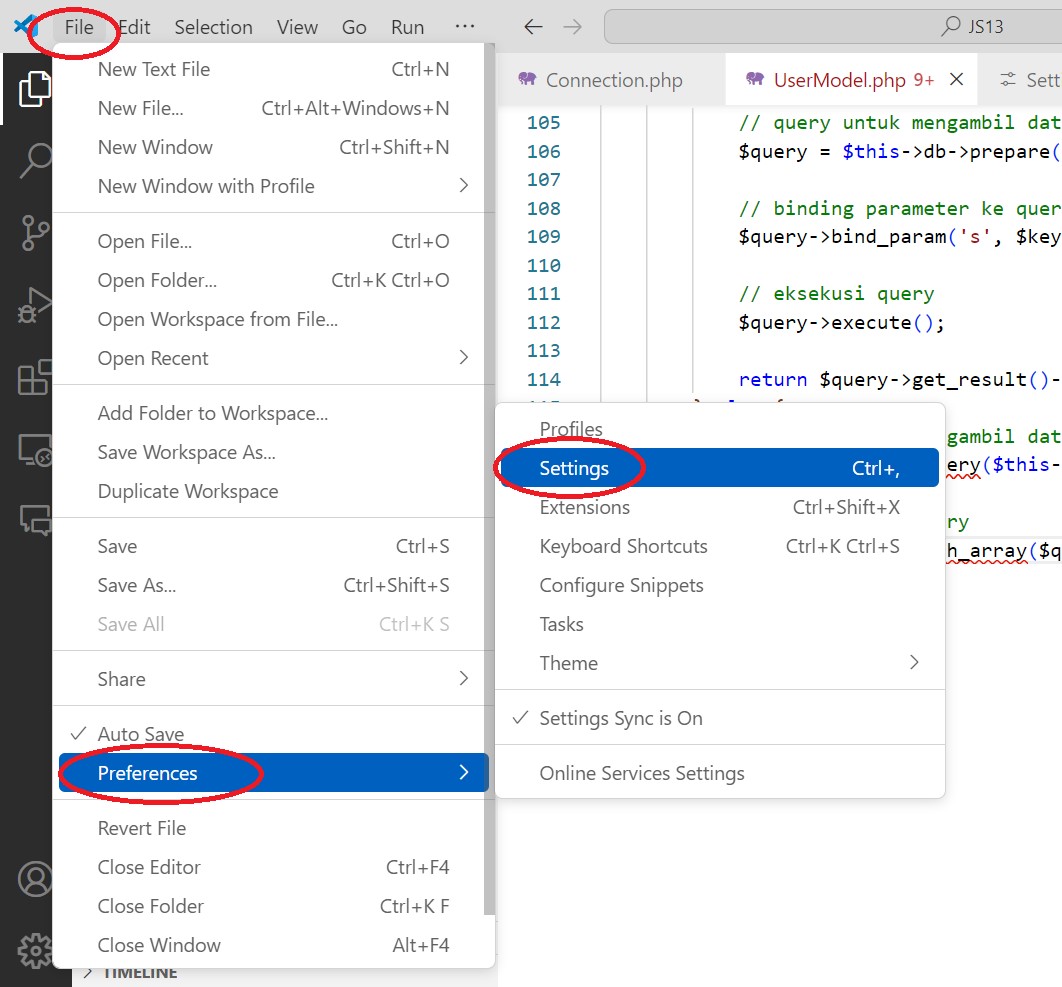
Silahkan implementasikan juga untuk menu Buku.

**Praktikum 4 – Integrasi PHP SQLSRV pada VSCode:**

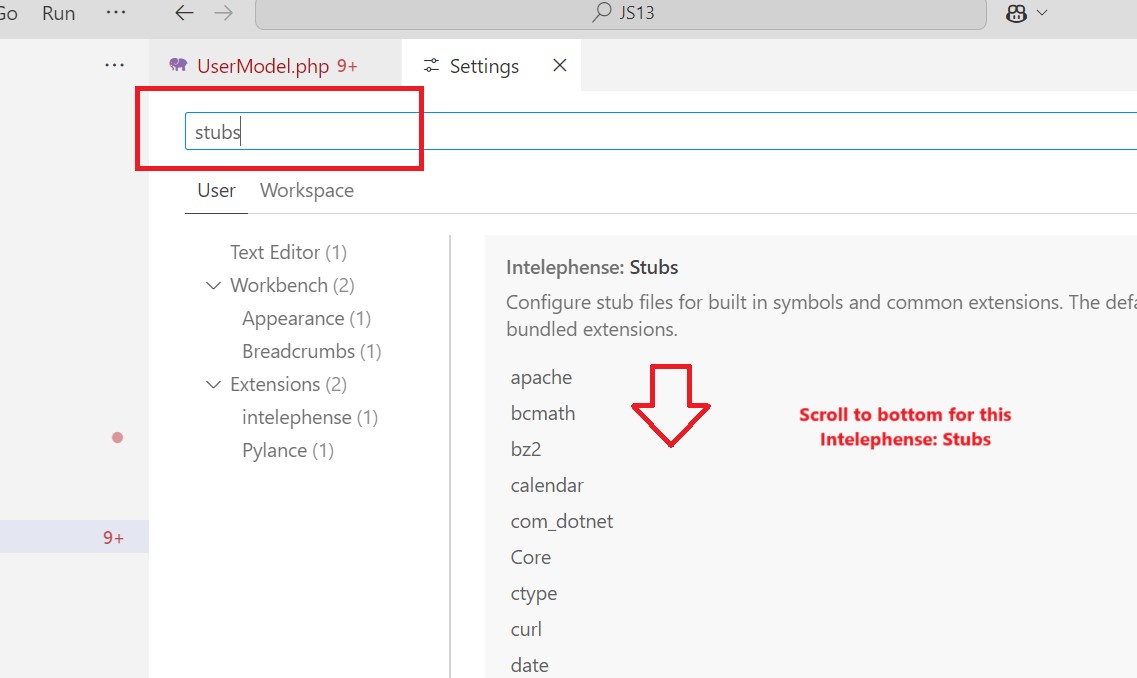
1. Terkadang teks editor kita akan menampilkan error ketika kita coding menggunakan ekstensi PHP SQLSRV. Hal ini terjadi karena secara default, ekstensi PHP SQLSRV tidak dipakai oleh PHP. Jadi perlu kita aktifkan



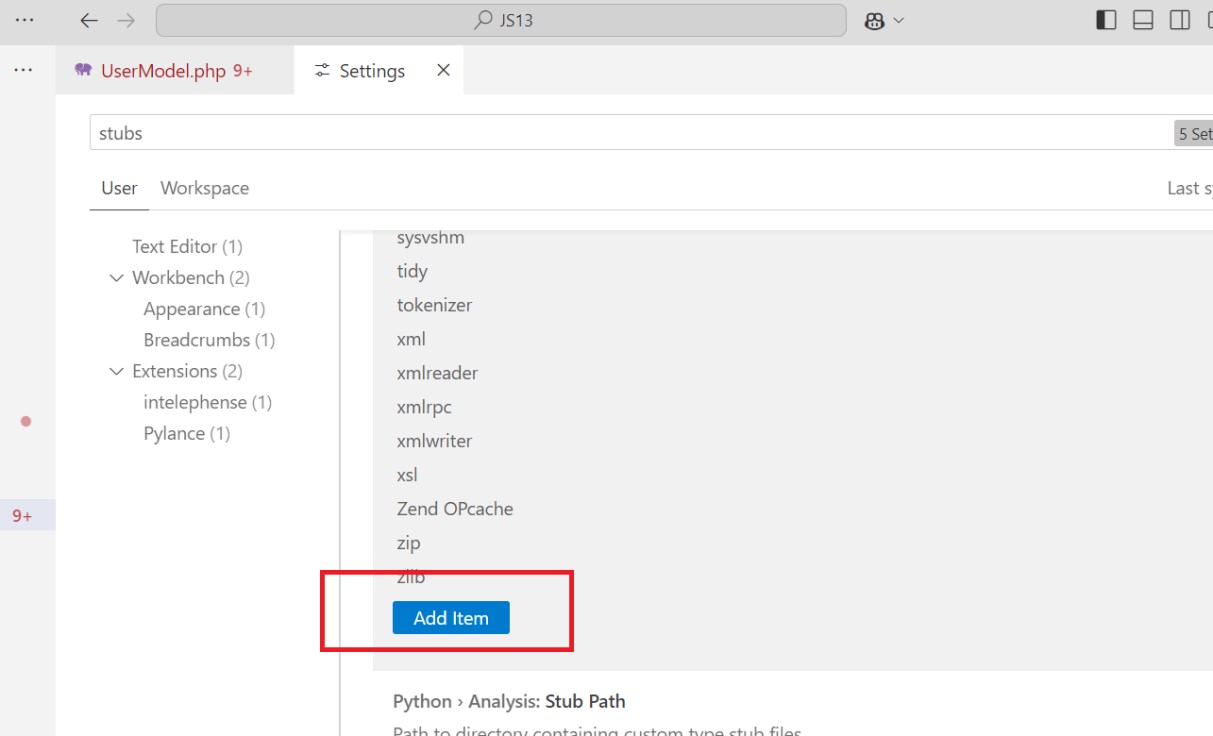
1. Unutk itu, kita pilih menu File > preferences > settings pada VSCode



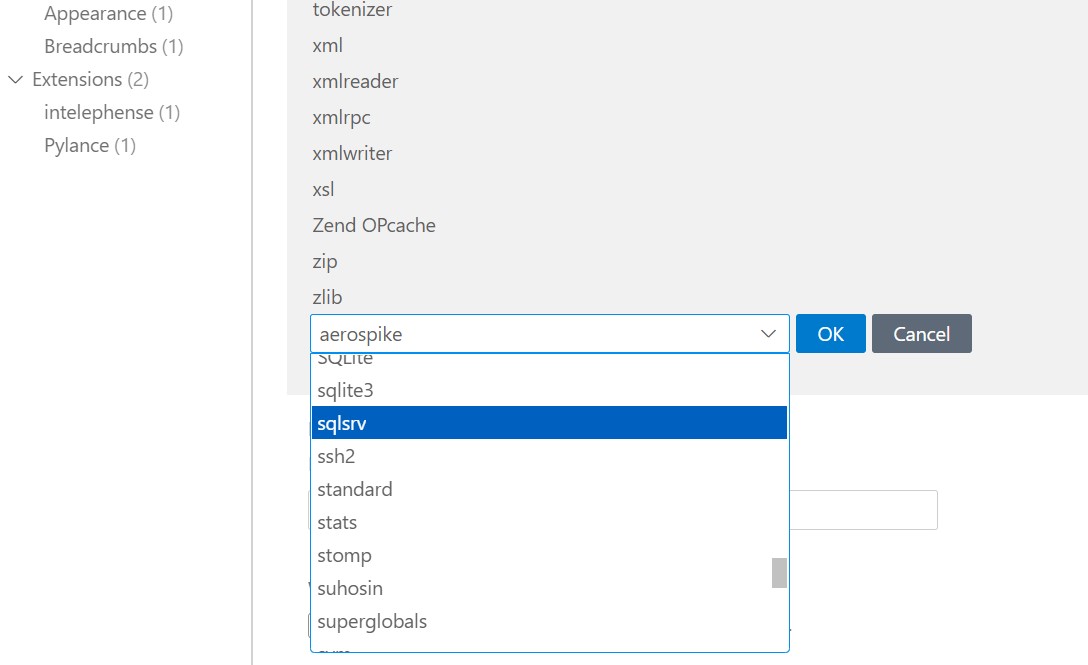
1. Kemudian kita ketikkan ***“stubs”*** pada kolom pencarian setting



1. Kemudian scroll pada bagian bawah ***“Intelephense:Stubs”***, kemudian klik ***“add item”***



1. Kemudian scroll pilihan, pilih “sqlsrv”, dan klik OK.



1. Selesai, VSCode akan mendeteksi fungsi yang ada di PHP SQLSRV.

*\*\*\* Sekian, dan selamat belajar \*\*\**