



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

JOBSHEET 13

Koneksi PHP ke SQL Server (Tambahan)

INFO

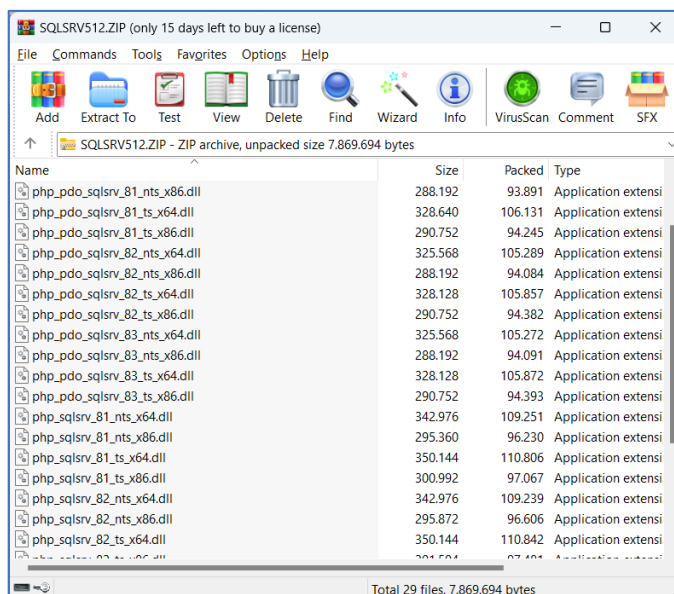
Materi ini adalah materi tambahan dari Jobsheet 13 untuk menghubungkan PHP dengan Database SQL Server.

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.

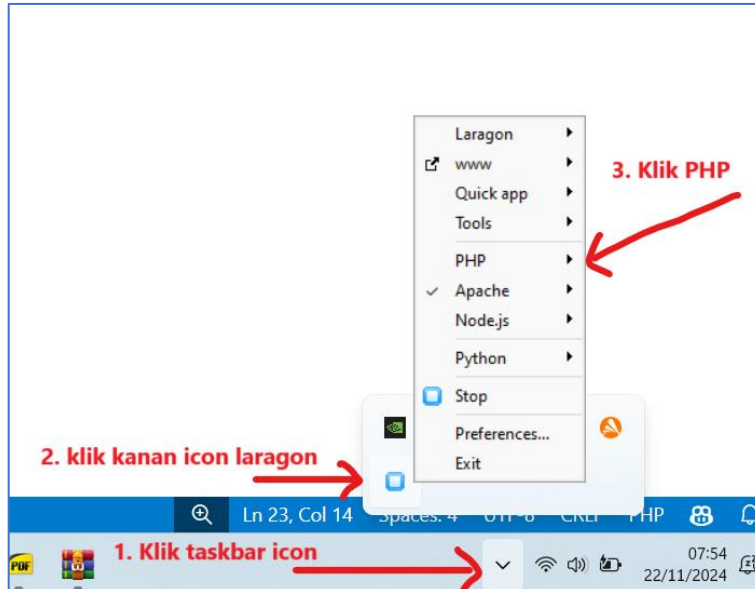
Untuk tabel contoh, kita bisa menggunakan DB dasar_web.sql (MySQL) yang sudah dikonversi ke SQL Server

Praktikum 1 – Instalasi Ekstensi SQL Server pada PHP:

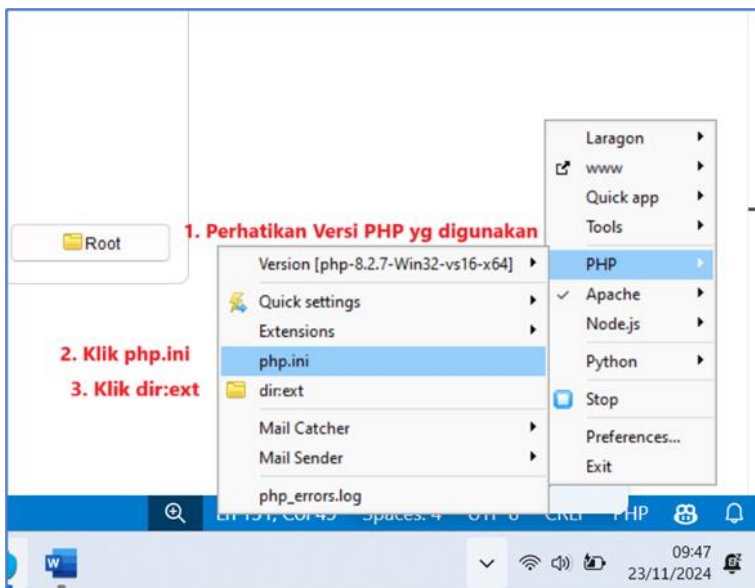
1. Kita download ekstensi **Microsoft Drivers for PHP for SQL Server** disini <https://go.microsoft.com/fwlink/?linkid=2258816>
2. Kemudian kita buka file **SQLSRV512.zip** yang telah kita download tadi



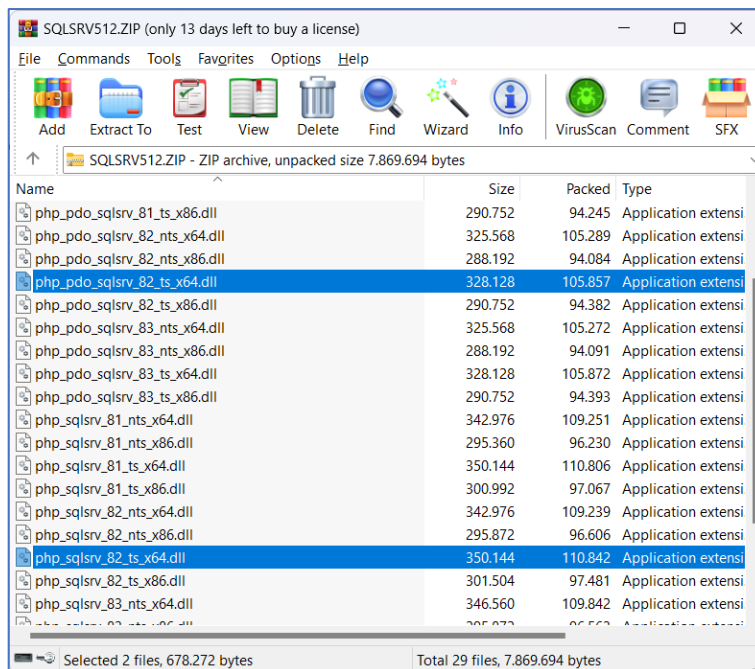
3. Setelah itu, kita cek dulu versi PHP yang kita gunakan di laragon



4. Kemudian akan muncul tampilan seperti berikut



5. 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).
6. Kita pilih `php_sqlsrv_82_ts_x64.dll` dan `php_pdo_sqlsrv_82_ts_x64.dll` yang akan kita masukkan ke ekstensi PHP.



7. 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**)

```
php.ini
931 ;extension=ffi
932 ;extension=ftp
933 extension=fileinfo
934 extension=gd
935 ;extension=gettext
936 ;extension=gmp
937 extension=intl
938 ;extension=imap
939 extension=imagick
940 extension=mbstring
941 extension=exif ; Must be after mbstring as it depends on it
942 extension=mysqli
943 ;extension=oci8_12c ; Use with Oracle Database 12c Instant Client
944 ;extension=oci8_19 ; Use with Oracle Database 19 Instant Client
945 ;extension=odbc
946 extension=openssl
947 ;extension=pdo_firebird
948 extension=pdo_mysql
949 ;extension=pdo_oci
950 ;extension=pdo_odbc
951 ;extension=pdo_pgsql
952 ;extension=pdo_sqlite
953 extension=pgsql
954 ;extension=shmop
955 extension=sqlsrv
956 extension=pdo_sqlsrv
957
958 ; The MIBS data available in the PHP distribution must be installed.
959 ; See https://www.php.net/manual/en/snmp.installation.php
960 extension=snmp
```

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

```
954 ;extension=shmop
955 extension=php_sqlsrv_82_ts_x64.dll
956 extension=php_pdo_sqlsrv_82_ts_x64.dll
957
```



8. Setelah itu, kita restart laragon kita. Selesai

Praktikum 2 – Cek Koneksi ke SQL Server dengan PHP:

1. Setelah kita berhasil melakukan proses instalasi ekstensi PHP SQL Server, kita buat koneksi PHP ke SQL Server
2. Kemudian kita modifikasi 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.

3. 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>

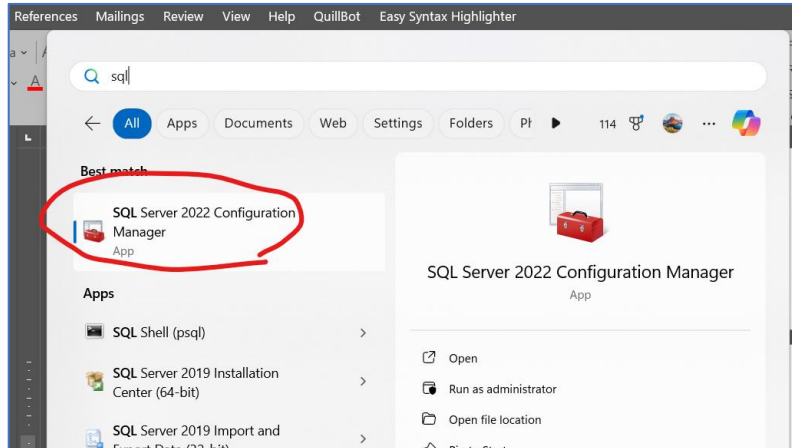
jika tampilan browser adalah **blank**, maka koneksi kita sudah sesuai dan **berhasil**.

Praktikum 2 selesai dan lanjut ke praktikum 3.

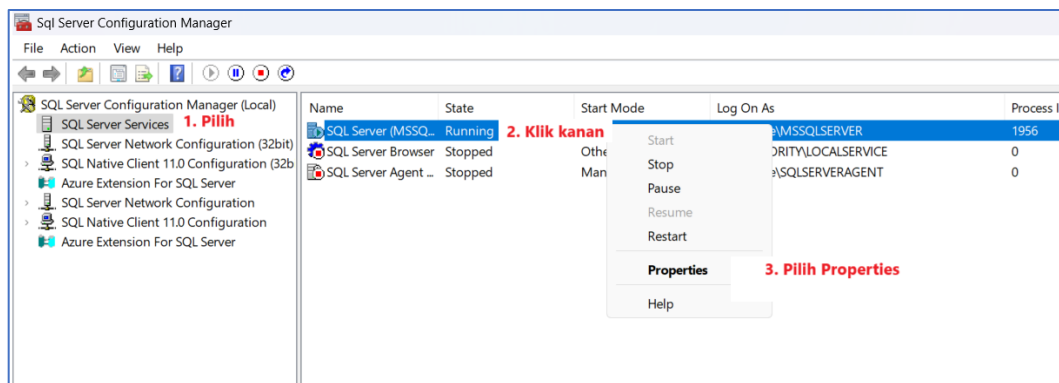
4. Jika koneksi masih tidak berhasil, maka perlu kita ganti konfigurasinya.



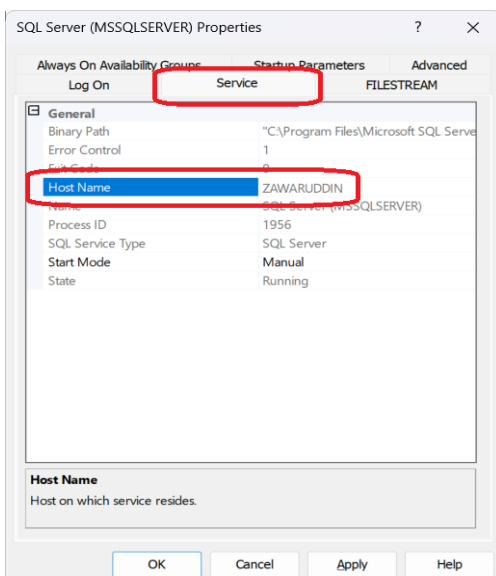
5. Kita buka SQL Server Configuration Manager pada Start Windows



6. Setelah terbuka, kita pilih sesuai dengan gambar



7. Selanjutnya, kita pilih tab **Service**, dan kita lihat nilai dari **Host Name**-nya



8. Host Name tersebut yang akan kita gunakan untuk konfigurasi **\$host** pada file **Connection.php** kita



```
Connection.php X
1  <?php
2      $use_driver = 'sqlsrv'; // mysql atau sqlsrv
3
4      $host      = "ZAWARUDDIN"; //'localhost';
5      $username  = ''; //'sa';
6      $password  = '';
7      $database  = 'master';
8      $db;
9
10
11     if($use_driver == 'mysql'){
```

9. 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);
    }
}
}
```

3. 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
        $query->execute();
    } else {
        // query untuk delete data
        sqsdrv_query($this->db, "delete from {$this->table} where kategori_id = ?",
[$id]);
    }
}
```

4. 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.');
```



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?

5. 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.'
    ]);
}
```

6. 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.

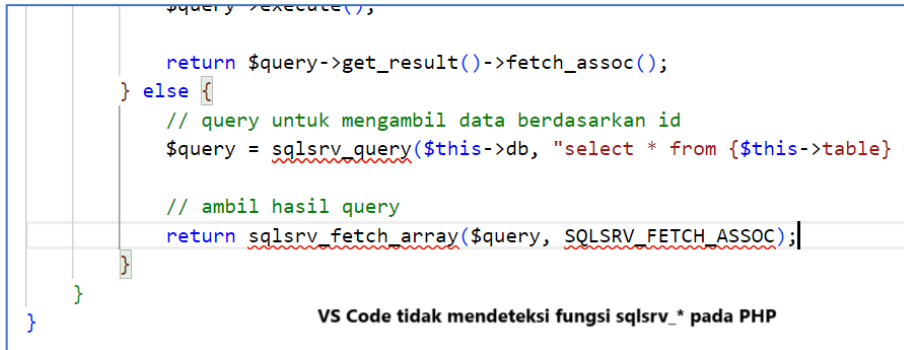
Pertanyaan 4

Silahkan implementasikan juga untuk menu Buku.



Praktikum 4 – Integrasi PHP SQLSRV pada VSCode:

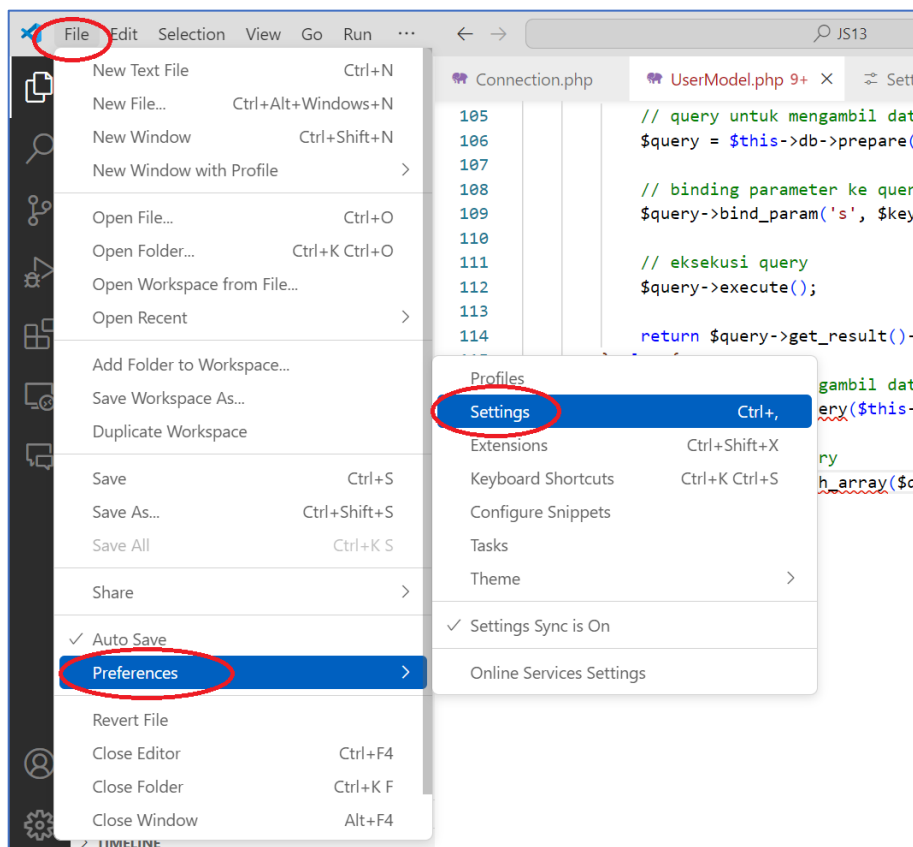
1. Terkadang teks editor kita akan menampilkan error ketika kita coding menggunakan ekstensi PHP SQLSRV. Hal ini terjadi karena secara default, ekstensi PHP SQLSRV tidak dipakai oleh PHP. Jadi perlu kita aktifkan



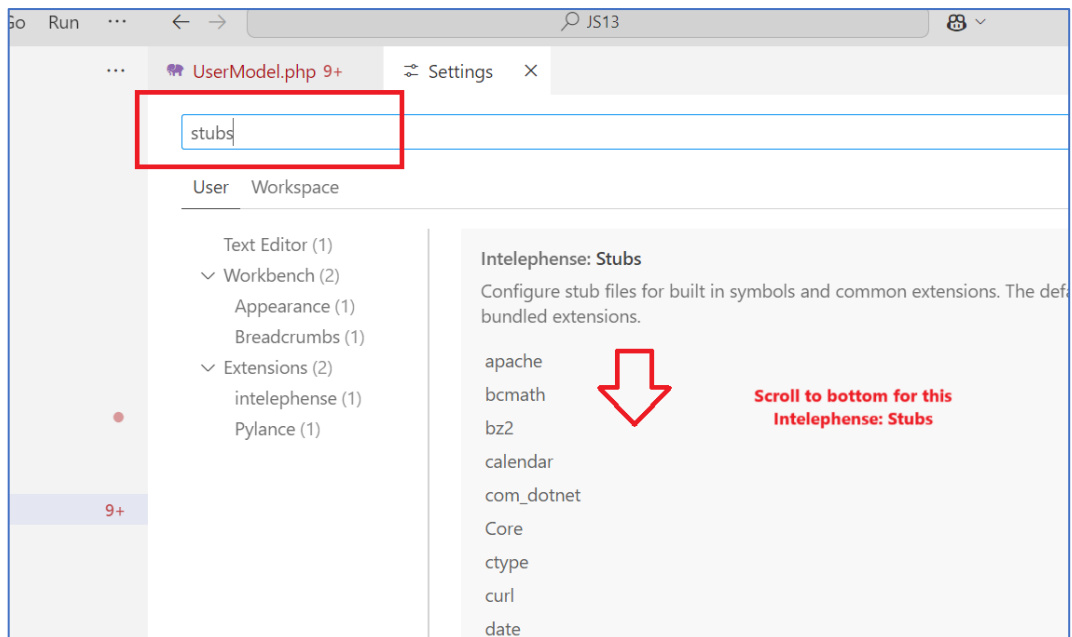
```
    $query->execute();  
  
    return $query->get_result()->fetch_assoc();  
} else {  
    // query untuk mengambil data berdasarkan id  
    $query = sqlsrv_query($this->db, "select * from {$this->table} w  
  
    // ambil hasil query  
    return sqlsrv_fetch_array($query, SQLSRV_FETCH_ASSOC);  
}
```

VS Code tidak mendeteksi fungsi sqlsrv_* pada PHP

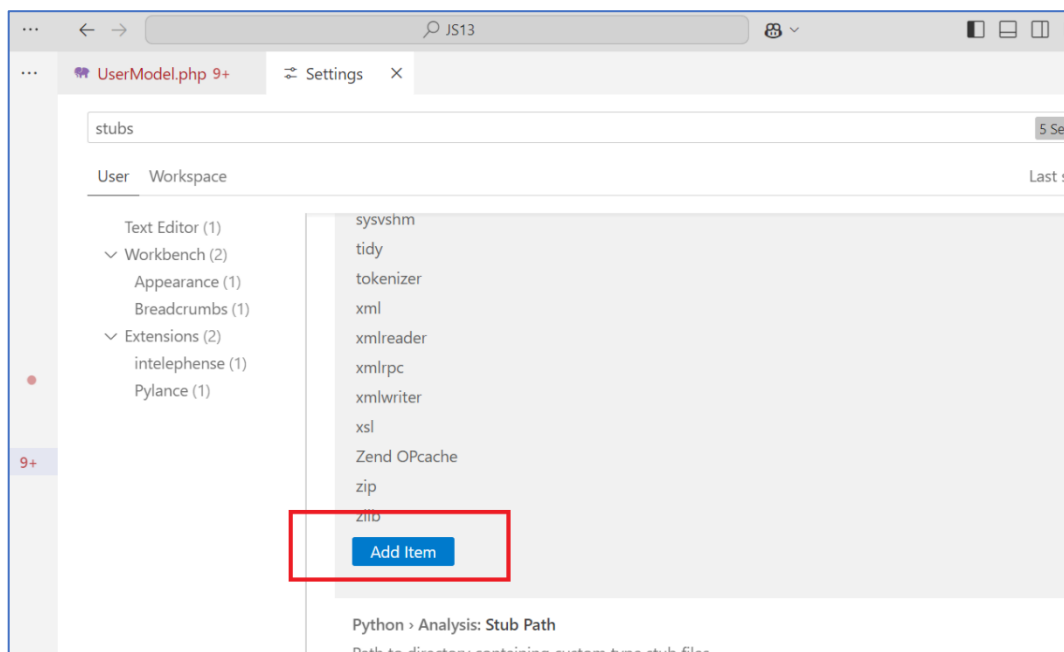
2. Untuk itu, kita pilih menu **File > preferences > settings** pada VSCode



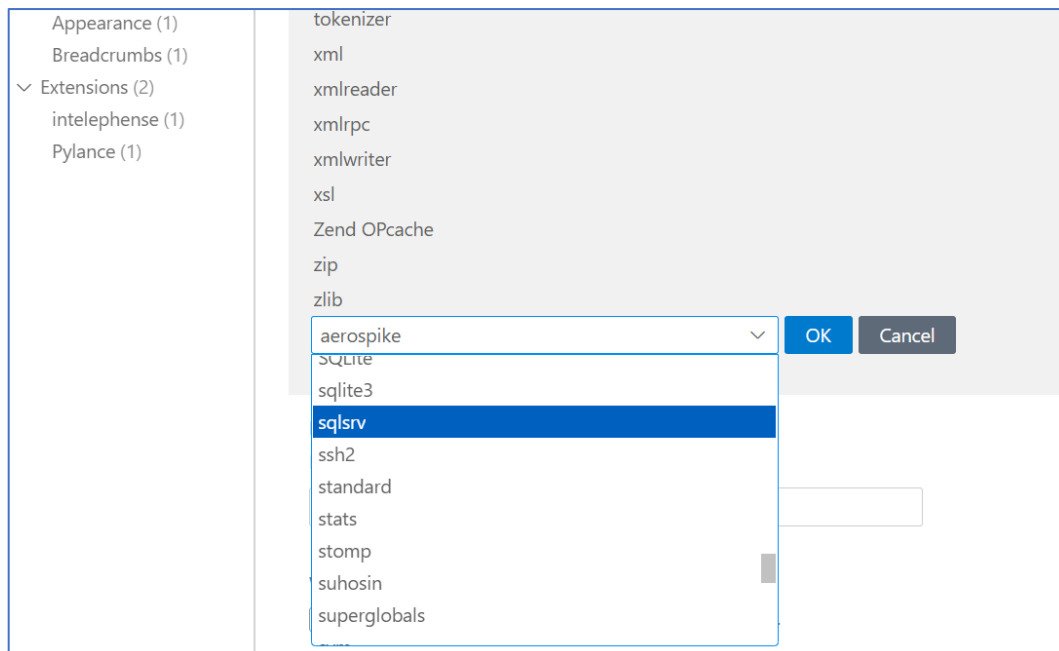
3. Kemudian kita ketikkan **“stubs”** pada kolom pencarian setting



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



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



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

*** Sekian, dan selamat belajar ***