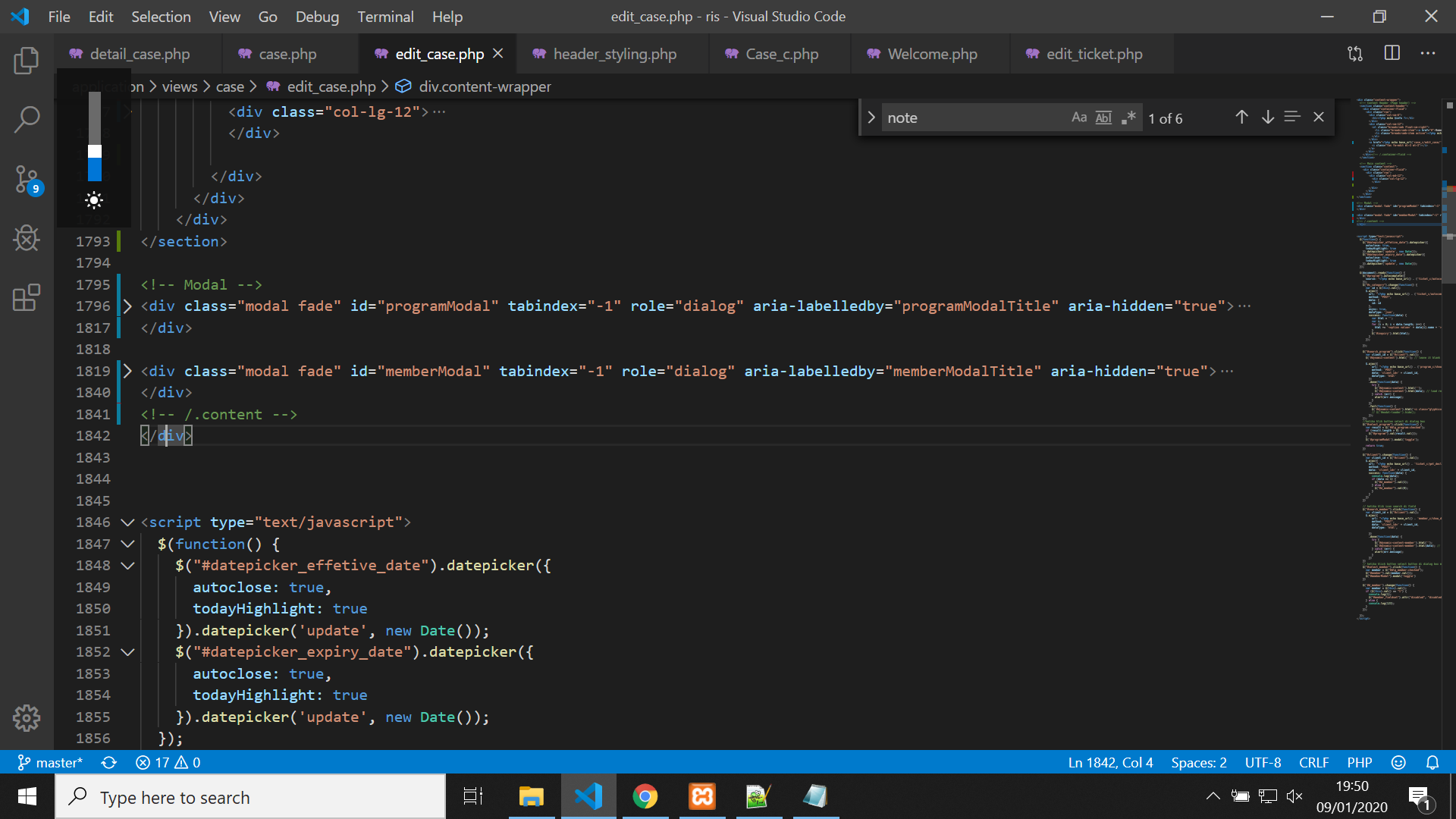
**Tutorial PHP**

***Dalam mempermudah mengecek line saat coding di vs studio code kita bisa collapase line.  
Ini cocok untuk cek tag pembuka dan tag penutup saat mengkoding html.***

***Contoh:***



Cara download File menggunakan SFTP

Kita anggap requirmentnya sudah ok (ssh2\_connect). Untuk extension ssh2\_connect bisa dipelajarin di tutorial campur-campur.

Pertama yang perlu kita pelajari:

1. Mengkoneksikan ke server:

Untuk konek ke server, kita menggunakan ssh2\_connect(ip\_server, port)

Contoh:

$server = "103.41.246.140";

$connection = ssh2\_connect($server, 22);

1. setelah ke server sudah terkoneksi/ tersambung, kini kita mencoba mengakses directory didalam server menggunakan ssh2\_sftp(koneksi).  
   contoh:

$stream = ssh2\_sftp($connection)

Seperti kita lihat parameternya, itu mengambil variable dari $connection yg digunakan di koneksinya (**ssh2\_connect($server, 22)**).

1. Setelah berhasil mengakses/membuka jalan ke server melalui **sftp,** kini kita bisa buka directory menggunakan fungsi native php (seperti **opendir**). Untuk formatnya ke dalam server lewat sftp adalah kita harus menambah ssh2.sftp://{akses jalan ke sftp yaitu $stream (seperti yang kita lihat di atas)}/nama\_folder yang dituju.

Contoh:

$dir = opendir("ssh2.sftp://{$stream}/{$remoteDir}")

**opendir** : kita gunakan untuk membuka satu directory/folder.  
**ssh2.sftp://** : ini adalah format yang sudah php punya, tidak boleh diganti-ganti dengan kata lain harus ada itu di dalam parameternya.  
**$stream** : itu adalah variable untuk mengakses ke sftp seperti yang kita lihat di atas (**ssh2\_sftp($connection)**).

Variable **$remoteDir** : adalah Path/folder yang kita tuju. Sebenarnya itu kita bisa langsung tulis dengan nama folder yang di server kita.   
Contoh:

$dir = opendir("ssh2.sftp://{$stream}/FOLDER\_AAI/old/test/")

Folder itu : **FOLDER\_AAI/old/test/** adalah folder yang ada di server ftp aai hanya sebagai contoh.

Dalam penulisan itu kita perhatikan tanda slashnya (/), sebab jika salah, itu membuat **opendir** tidak berjalan dengan benar, dan hasilnya jika salah adalah false. Kesalahan ini biasanya terjadi di sana kita pakai variable. Contoh kesalahan yang pernah saya alami :

$remoteDir = '/FOLDER\_AAI/old/test/uploaded/UAT\_Doc\_Claim/12/Cashless';

Diakhir dari nilai variable (setelah Cashless) itu tidak ada tanda slash (/). Karena itu, ketika kita ingin mebuka file itu dengan **opendir,** kita tidak menambah slash setelah variable **$remoteDir** kita panggil.

Contoh:

$remote = @fopen("ssh2.sftp://{$stream}/{$remoteDir}{$file}", 'r');

Seperti yang kita lihat di dalam {$remoteDir}{$file}, diataranya tidak ada tanda slash (/), itu akan salah yang mana kalau kita var\_dump $remote, itu hasilnya akan false. Maka yang benarnya kita harus menambah tanda slash diantara mereka ($remoteDir dan $file). Maka yang benarnya adalah

$remote = @fopen("ssh2.sftp://{$stream}/{$remoteDir}/{$file}", 'r');

namun jika di dalam variable$remoteDir, paling belakang ada tanda slash (/), maka kita tidak perlu menambah tanda slah diantar tanda itu, karena sudah ada di dalam variable $remoteDir. Seperti ini

$remoteDir = '/FOLDER\_AAI/old/test/uploaded/UAT\_Doc\_Claim/12/Cashless/';

Berikut adalah conoth lengkap (yang sudah berjalan) untuk mendowload file dari server menggunakan bahasa php dengan ssh2\_connect. Project ini ada di dalam module **download\_file** di dalam htdocs**.**

public function download()

  {

    include\_once(BASEURL . '/app/config/connection.php');

    $remoteDir = '/FOLDER\_AAI/old/test/uploaded/UAT\_Doc\_Claim/12/Cashless';

    $localDir = SITENAME . "/file\_b/";

    ################

    ### buat jalan ke dalam sftp dengan code ssh2\_sftp dan simpan ke dalam $stream

    ################

    if (!$stream = ssh2\_sftp($connection))

      die('Unable to create a stream.');

    ################

    ### buka directory tujuan sftp kita sambungkan dengan var $stream yang di atas.

    ################

    if (!$dir = opendir("ssh2.sftp://{$stream}{$remoteDir}"))

      die('Could not open the directory');

    ################

    ### baca dir dan dan simpan nama-nama file nya ke dalam variable $files

    ################

    $files = array();

    while (false !== ($file = readdir($dir))) {

      if ($file == "." || $file == "..")

        continue;

      $files[] = $file;

    }

    ################

    ### loop nama-nama file yang ada di variable $files diatas

    ################

    foreach ($files as $file) {

      try {

        $remote = @fopen("ssh2.sftp://{$stream}/{$remoteDir}/{$file}", 'r');

        ### sengaja kita var\_dump untuk melihat hasi dari @fopen ssh2 di atas, apakah berjalan benar,

        ### jika hasilnya false, maka mungkin pathnya tidak benar, atau mungkin koneksinya. untuk perlu mengeceknya

        ### perlu cek satu-satu mulai dari koneksi kemudian pathnya, namu jika sudah berhasil, ini hapus atau comment aja var\_dumpnya

        var\_dump($remote);

        // throw new Exception('Division by zero.');

      } catch (exception $e) {

        echo "message: " . $e->getMessage();

      }

      ################

      ### tampilkan warning message jika, @fopen dir ke server via sftp salah

      ################

      if (!$remote) {

        echo "Unable to open remote file: $file\n";

        continue;

      }

      ################

      ### open local kita dengan @fopen, dnegan code seperti di bawah, jika filenya tidak ditemukan, maka nama file itu akan di create

      ### di local kita pertama kosong, dan nantinya akan di write di code berikutnya (fwrite).

      ################

      if (!$local = @fopen($localDir . $file, 'w')) {

        echo "Unable to create local file: $file\n";

        continue;

      }

      $read = 0;

      $filesize = filesize("ssh2.sftp://{$stream}/{$remoteDir}/{$file}");           # kita perlu check size file untuk di pakai pada fread, sebab itu adalah salah satu parameter requirednya

      ################check apakah file $read < lebih dari file size dan                                                                                  kemudian read file di remote dan simpan ke dalam                                                                                    variable $buffer

      while ($read < $filesize && ($buffer = fread($remote, $filesize - $read))) {

        $read += strlen($buffer);

        ################

        ### Write file local yang sudah di create di atas

        ################

        if (fwrite($local, $buffer) === FALSE) {

          echo "Unable to write to local file: $file\n";

          break;

        }

      }

      ################

      ### Close connection

      ################

      fclose($local);

      fclose($remote);

    }

  }

Penjelasan:

include\_once(BASEURL . '/app/config/connection.php');

Baris ini ada kita panggil file koneksi ke server.  
isi file connection.php:

<?php

$port = "22";

$server = "103.41.246.140";

$username = "user\_aai1";

$password = "T4usY7z`5s";

$connection = ssh2\_connect($server, 22);

if(!ssh2\_auth\_password($connection, $username, $password))

{

    throw new Exception('error 1');

}

Baris selanjutnya:

if (!$stream = ssh2\_sftp($connection))

   die('Unable to create a stream.');

Disini kita mengecek connection akses ke **sftp** ke dalam directory dan sekalian menyimpan hasil akses/koneksinya ke dalam variable $stream yang akan kita pakai di baris selanjutnya.

if (!$dir = opendir("ssh2.sftp://{$stream}{$remoteDir}"))

    die('Could not open the directory');

disini kita membuka folder di server sftp kita dan menyimpan ke dalam $dir, namun jika hasilnya adalah **false**, maka itu akan di die/di putus prosesnya dan tampilkan pesan could not…

untuk penjelasan selanjutnya sudah ada di dialam code di atas.

**File Handling**

File handling merupakan satu bagian yang penting didalam aplikasi web. Kita sering ingin membuka sebuah file dan memprosesnya untuk beberapa tujuan, misalnya mengedit, menghapus, dan lain sebagainya.

Berikut beberapa tujuan yang kita bisa lakukan dengan file handling:

1. Manipulasi file :

1.1 Membuat file baru

1.2 membaca file

1.3 mengupload

1.4 mengeditnya.

Peringatan:

Kita harus berhati-hati Ketika ingin memanipulasi file menggunakan code php ini. Karena jika salah, itu akan merusak file kita. Contoh: saat kita ingin membuka sebuah file dengan mode **write(w)**, maka itu akan menghapus semua dari isi file kita. Berikut contoh codenya : fopen(“document.txt”, “w”). Oleh sebah itu kita harus berhati-hati dalam memanipulasi file menggunakan code php.

* 1. Membuat file baru

Untuk membuat/create file baru kita menggunakan **fopen()**. Sedikit

membingungkan memang, tapi itu adalah memang fungsi php nya. Di parameternya kita harus menggunakan mode ‘w’.

Format: fopen(“nama\_file”, ‘w’);

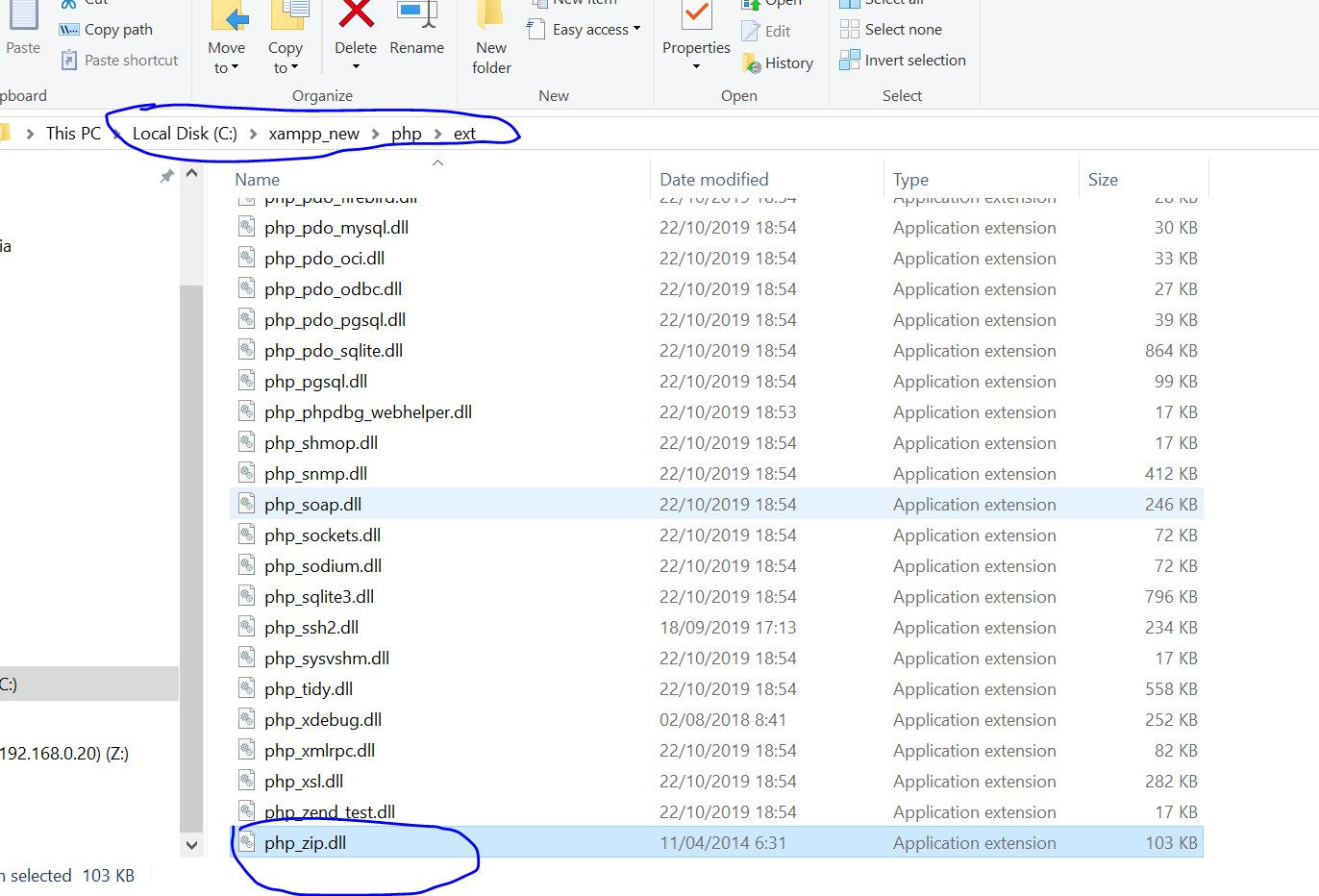
Contoh: fopen(“test.txt”, ‘w’);

Penjelasan:

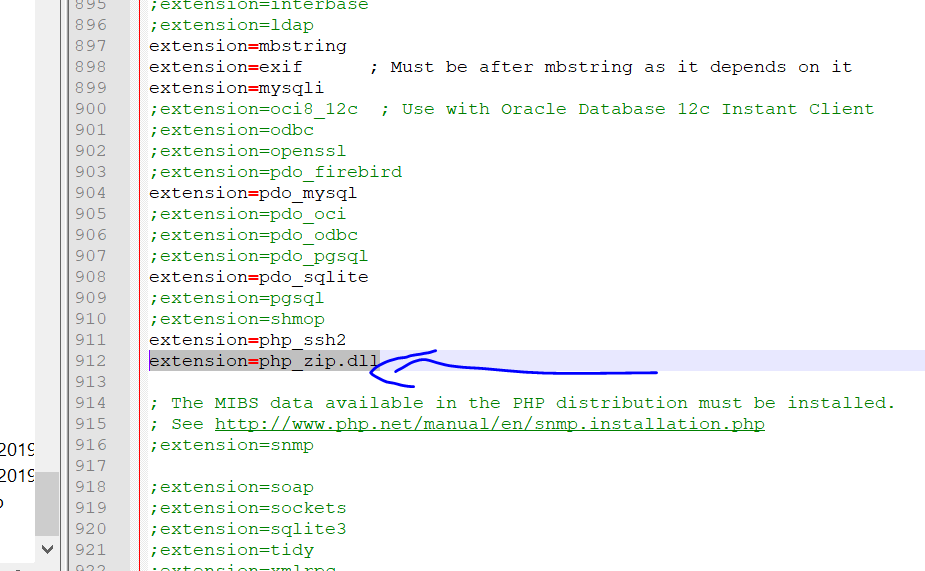
Code diatas akan membuat sebuah file dengan nama test.txt di dalam directory yang sama dengan code php kit aitu jika belum ada file dengan nama test.txt itu sebelumnya. Namun jika sudah ada itu akan mereplace file sebelumnya dan menghapus isi sebelumnya sehingga jika sebelumnya file test.txt ada isinya, maka setelah code diatas kita jalankan, isinya akan menjadi kosong. Untuk itulah kita harus hati-hati menggunakan file handling dari php.

**Men-zip file dengan php (projectnya ada di htdocs/zipped)**

Untuk menzip file, kita perlu menambah extension dari php. Yaitu kita menggunakan **php\_zip.dll** dan letakkan di folder php/ext seperti gambar di bawah.



Setelah itu buka file php.ini kemudian aktifkan extension dengan menambah kode ini : extension=php\_zip.dll



Untuk penambahan code di php.ini pastikan rapi. Contoh seperti di atas, untuk menambah extension kita menambahnya di bawah urutan extension yang udah ada dari core phpnya.

Berikut untuk menzip file:

1. Buat object baru (Instantiate)

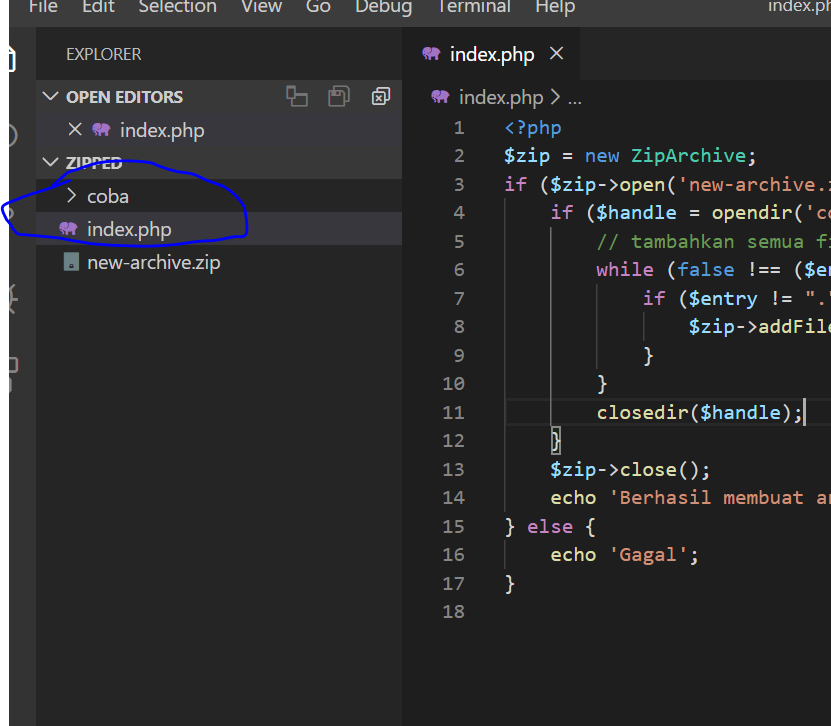
$zip = new ZipArchive;

1. Create baru zip file yang nanti kita write isi valuenya dengan file-file kita.

$zip->open('new-archive.zip', ZipArchive::CREATE) === TRUE

1. Open lokasi file yg akan kita zip;

$handle = opendir('coba')

**Coba** adalah folder module kita dimana lokasi sama dengan file index.kita  


1. Setelah kita open, sekarng kita read directorynya dan sekalian loop.

while (false !== ($entry = readdir($handle))) {

            if ($entry != "." && $entry != ".." && !is\_dir('coba/' . $entry)) {

                $zip->addFile('coba/' . $entry);

            }

        }

        closedir($handle);

1. Kemudian close object zip yang sudah kita open di atas.

$zip->close();

**Cara mengirim file dari local ke server menggunakan sftp (projectnya ada di htdocs/document\_upload)**

Ini adalah kebalikan dari kita download file dari server. Kali ini kita akan mengirim file dari local ke server menggunakan sftp.

Caranya hampir sama.

1. Koneksi

include\_once(BASEURL . '/app/config/connection.php');

1. Buat sumber dan tujuan folder

$srcFile = SITENAME. "/file/";

$dstFile = "/FOLDER\_AAI/old/test/";

1. Kita masuk ke dalam suber file yang akan kita kirim (dilocal)

chdir($srcFile);

1. kemudian kita loop:

foreach (glob("\*.\*") as $filename) {

    $sftpStream = @fopen('ssh2.sftp://' . $sftp . $dstFile . $filename, 'w');

Dari code di atas di sana sekalian membuat file baru di server dengan nama filenya adalah sama dengan nama file di local kita (karena itu adalah nama variable file yang kita loop dari local kita).

1. Kita kemudian ambil isi content yg di local kita, dan simpan kesatu variable:

data\_to\_send = @file\_get\_contents($filename);

1. Kemudian setelah kita berhasil masuk ke dalam directory dan membuat nama file baru, kemudian variable yang menampung isi dari content local kita tadi, kita buat ke file yang di server dengan kode @**fwrite**: codenya adalah @file\_get\_contents. Seperti di bawah ini:

@fwrite($sftpStream, $data\_to\_send)

Kode lengkapnya seperti berikut:

public function upload($column)

    {

        include\_once(BASEURL . '/app/config/connection.php');

        //include\_once(BASEURL . '/app/config/koneksi.php');

        $folder = 'C:\xampp/htdocs/document\_upload/excel\_file/';

        $file\_xlsx = 'file\_excel.xlsx';

        $Import = Import::open(path($folder) . $file\_xlsx);

        $Import->setMap($column);

        // $query = Database::on('dblocal\_dev\_aia')->builder();

        $datas = array();

        $memberIn = '';

        $Import->each(function ($row, $index) use (&$datas) {

            $datas[] = $row;

        }, true);

        foreach ($datas as $data) {

            $dir = $data['case'];

            //$data\_case = $this->getCase('`case`', $dir);

            //$name = $data\_case['upload\_original\_bill'] . ',' . $data['upload\_original\_bill'];

            //$query = "UPDATE `case` set upload\_original\_bill = '" . $name . "' where id = '" . $dir . "' ";

            //$koneksi->query($query);

            //if (!$koneksi) {

                //echo mysqli\_error($koneksi);

            //}

            $srcFile = SITENAME. "/file/$dir/";

            $dstFile = "/FOLDER\_AAI/old/test/$dir/";

            if (is\_dir($srcFile)) {

                $opdir = opendir($srcFile);

            } else {

                return 'Error Path';

            }

            chdir($srcFile);

            foreach (glob("\*.\*") as $filename) {

                $sftpStream = @fopen('ssh2.sftp://' . $sftp . $dstFile . $filename, 'w');

                try {

                    if (!$sftpStream) {

                        throw new Exception("Could not open remote file");

                    }

                    $data\_to\_send = @file\_get\_contents($filename);

                    if ($data\_to\_send === false) {

                        throw new Exception("Could not open local file: $srcFile.");

                    }

                    if (@fwrite($sftpStream, $data\_to\_send) === false) {

                        throw new Exception("Could not send data from file: $srcFile.");

                    }

                } catch (Exception $e) {

                    error\_log('Exception :' . $e->getMessage());

                }

                fclose($sftpStream);

            }

        }

        return true;

    }

Konsep kopi file adalah seperti berikut:

1. Buat 1 file kosong dan sekalian open. Contoh test.txt

Contoh :

$sftpStream = @fopen('ssh2.sftp://' . $sftp . $dstFile . 'test.txt', 'w');

2. sekarng filenya sudah tercreate dan terbuka (open), kemudian kita sekarng bisa mengisinya dengan content file yang akan kita duplicate/copy.

$data\_to\_send = @file\_get\_contents('test\_sumber.txt');

@fwrite($sftpStream, $data\_to\_send)

3. terakhir, kita perlu menglose file yang kita open dilangkah 1 di atas.

fclose($sftpStream);

cara memanggil function di dalam model dengan php atau menjalankan url dengan server.

<?php

file\_get\_contents("http://10.10.10.29:8007/intsys/copy\_file\_tmli");

**PDO**

**PDO** hanya merupakan interface

Konsepnya:

**PHP PDO –> Database Driver –> Database Server**

Cara melihat driver PDO yg aktif di php.

<?php

   print\_r(PDO::getAvailableDrivers());

?>

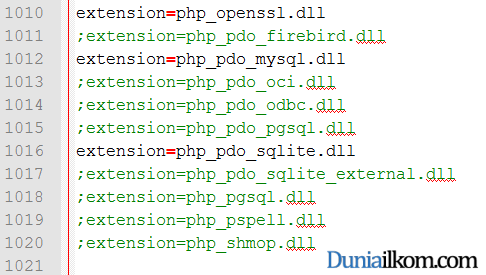
Contoh output:

Array ( [0] => mysql [1] => sqlite )

Dari tampilan diatas, dapat dilihat bahwa driver PDO bawaan PHP yang aktif (dan yang bisa digunakan) hanyalah **MySQL** dan **SQLite**.

**Sebenarnya masih banyak driver yang bisa diaktifkan selain yang default udah aktif (oracle, postgresql, dll). Kenapa php tidak mengaktifkan semua secara default? Alasannya karna untuk tidak mengurangi performance.**

Jadi bagaimana cara mengaktifkan driver untuk database lainnya? Caranya adalah mengubah setingan **php.ini.**



Seperti kita lihat di atas yang berawalan php\_pdo itu semua yang support untuk php pdo.

Contoh php\_pdo\_pgsql untuk postgresql.

Cara mengkoneksikan ke database:

Format:

$dbh = new PDO($dsn, $user, $passwd);

Itu hanya ada 3 parameter, yaitu $dsn (domain name server), $user, dan $passwd

$dsn itu terdiri dari 3 yaitu: type dbms, ip, nama database.

Maka untuk lengkapnya seperti berikut:

$dsn = "mysql:host=localhost;dbname=mydb";

$user = "root";

$passwd = "";

Kemudian kita panggil di formatnya

$dbh = new PDO($dsn, $user, $passwd);

- Cara menghapus data

$nrows = $pdo->exec("DELETE FROM countries WHERE id IN (1, 2, 3)");

$nrows kita gunakan untuk menampung berapa data yang dihapus sehingga nanti bisa kita tampilkan berapa data yang dihapus.

Contoh:

echo "The statement affected $nrows rows\n";

- Mendapatkan data dari database:

Ada 3 cara mendapatkan data dari database:

PDO::FETCH\_ASSOC : mendapatkan data dengan array dimana keynya adalah nama kolom

PDO::FETCH\_NUM : mendapatkan data dengan array dimana keynya adalah angka berurut mulai dari 0

PDO::FETCH\_BOTH : mendapatkan data dengan array dimana keynya adalah bisa kedua-duanya. Jadi pada saat memanggilnya nanti kita bisa menggunakan key nomor dan nama kolom

contoh:

<?php

$dsn = "mysql:host=localhost;dbname=mydb";

$user = "user12";

$passwd = "12user";

$dbh = new PDO($dsn, $user, $passwd);

$stm = $dbh->query("SELECT \* FROM countries");

$rows = $stm->fetchAll(PDO::FETCH\_NUM);

foreach($rows as $row) {

printf("$row[0] $row[1] $row[2]\n");

}

Penjelasan:

$stm = $dbh->query("SELECT \* FROM countries");

Kode diatas ini adalah kita menjalankan query untuk panggil ke database

$rows = $stm->fetchAll(PDO::FETCH\_NUM);

Kode inilah kita mengambil data dari query itu yg sudah ditampung didalam variable $stm.

Di sini kita menggunakan cara PDO::FETCH\_NUM, diman seperti yang kita lihat di dalam perulangannya dipanggil dengan cara index (printf("$row[0] $row[1] $row[2]\n");

).

Untuk penggunaan cara yang lain, tinggal ganti FECT\_NUM kedalam cara manggil kita yang baru.

Fungsi di dalam myqli ada fetchall, dll.

**PHP PDO parameter binding**

Maksud dari binding adalah dimana di query kita kasih variable untuk menampung input yang dibuat di client.

Contoh:

$id = 12;

$stm = $pdo->prepare("SELECT \* FROM countries WHERE id = ?");

$stm->bindValue(1, $id);

$stm->execute();

Seperti yang kita lihat di atas, WHERE id = ? , dalam hal ini, nilai dari id akan di isi di statement berikutnya dengan variable $id.

Cara menginput data kedalam database:

$pdo->exec("INSERT INTO words(word) VALUES ('pen')");

$pdo->exec("INSERT INTO words(word) VALUES ('bum')");

$pdo->exec("INSERT INTO words(word) VALUES ('hum')");

$pdo->exec("INSERT INTO words(word) VALUES ('den')");

Mengetahui data yang di insert terakhir:

$rowid = $pdo->lastInsertId();

Echo $rowid

Mencreate table di dalam database kita:

$sql = "CREATE TABLE words(id INT PRIMARY KEY AUTO\_INCREMENT,

word VARCHAR(255))";

$ret = $pdo->exec($sql);

Bagaimana kita tahu itu, kalau query itu akan mencreate table di database yang kita mau? Jawab: itu akan otomatis mengcreate di atas dengan koneksi database yang sudah kita pilih saat di configurasi koneksi.

## PHP PDO Transactions

Transactions sama fungsi dengan sql transaction dimana query akan dijalankan jika semua query tidak ada yang error. Contoh jika kita ingin mengimput/mengupdate data ke dalam 2 table atau 2 database yang berbeda. Maka kita perlu memastikan semua query berjalan dengan benar baru semua di proses. kalau misalnya tidak menggunakan transaction kemungkinan akan terjadi proses data sebagian tidak tersimpan database dan sebagian sudah terupdate. Hal ini akan membuat rusaknya data, dimana kita harus cek data2 supaya menjadi benar. Maka untuk mengatasi itu kita pakai transactions.

Format:

PDO beginTransaction() : membuka transaction

$pdo->commit(); : menjalankan semua query, jika tidak ada error

$pdo->rollback(); : menampilkan hasil error jika terjadi error di querynya

Contoh:

<?php

$dsn = "mysql:host=localhost;dbname=mydb";

$user = "user12";

$passwd = "12user";

$pdo = new PDO($dsn, $user, $passwd);

try {

$pdo->beginTransaction();

$stm = $pdo->exec("INSERT INTO countries(name, population) VALUES ('Iraq', 38274000)");

$stm = $pdo->exec("INSERT INTO countries(name, population) VALUES ('Uganda', 37673800)");

$pdo->commit();

} catch(Exception $e) {

$pdo->rollback();

throw $e;

}

Mendapatkan meta data.

Salah meta data adalah jumlah rows dari hasil sebuah select:

Contoh:

$stm = $pdo->query("SELECT name, population FROM countries WHERE id=1");

$ncols = $stm->columnCount();

Cara filter dari url untuk memberikan parameter:

key dan parameternya harus diberikan setelah semua linknya kemudian ditambahkan tanda tanya, kemudian key(name) arraynya dan valuenya (Pakaian), tanpa tanda kutip

contoh:

<http://localhost/toko/api/categories/list.php?name=Pakaian>

Mensetting time zone:

**date\_default\_timezone\_set** ( string $timezone\_identifier ) : bool

contoh: date\_default\_timezone\_set('America/Los\_Angeles');

mempersingkat tulisan untuk mmengecek array di key.

Contoh:

$limit = isset($\_GET['limit']) ? (int) $\_GET['limit'] : 0;

Cara panjangnya kita menggunakan if.

Contoh :

if (isset($\_GET['limit']))

Namespace

Contoh:

client\Supplier

Beberapa manfaat namespace:

Supaya saat require/memanggil kita tidak perlu menuliskan lokasi filenya.

Untuk menghindari jika ada beberapa nama file yang sama di dalam module/aplikasi

Menghapus isi dari array:

array array\_splice($array1, $start\_point, $range, $array2)

$array1 = sumber array yg akan dihapus.

$start\_point = penghapusan dimulai dari index keberapa

$range = jumlah data yang akan hapus dimulai dari $start\_point di atas.

contoh:

$array1 = array("10"=>"raghav", "20"=>"ram",

    "30"=>"laxman","40"=>"aakash","50"=>"ravi");

print\_r array\_splice($array, 1, 2).

Output:

$array1 = array("10"=>"Raghav,

"40"=>"aakash","50"=>"ravi");

Merubah stdclass jadi array

get\_object\_vars(stdClass)

Ternary:

Ternary adalah sebuah pengkondisian, fungsinya sama dengan fungsi **if else.**

Contoh:

$var = 5;

$var\_is\_greater\_than\_two = ($var > 2 ? true : false);

echo $var\_is\_greater\_than\_two // returns true

jika di dalam if, maka itu sama dengan:

if ($tickets[0]->status == '1') {

    echo false;

    }

else {

     echo true;

     }

Untuk ternary yang bertingkat, atau if elseif else.

<?php echo ($tickets[0]->status == '1' ? 'pending': ($tickets[0]->status == '2' ? 'Proceed to case': 'Cancel')); ?>

maka code di atas dalam if sama dengan:

if ($tickets[0]->status == '1') {

   echo 'Pending';

   }

elseif($tickets[0]->status == '2')

   {

   echo 'Proceed to case';

   }

else {

   echo 'Cancel';

   }

Membuat block untuk membuat perulang html element, kita bisa lakukan seperti berikut:

<?php foreach ($clients as $client) : ?>

      <tr>

        <!-- <td><a type="submit" href=""><i class="far fa-check-square"></i></a></td> -->

        <td><?php echo $client->id;?></td>

        <td><input class="btn" style="color: blue" id='client\_idb' type="button" id="myBtn" value="<?php echo $client->full\_name; ?>"></td>

      </tr>

    <?php endforeach; ?>

Cara mempermudah untuk mapping field-field dari form ke dalam table database.

Dalam hal ini, kita akan membuatnya dengan format codeighniter.

Contoh:

Untuk mengambil data dari form html, maka kita harus mengetikkan seperti berikut:

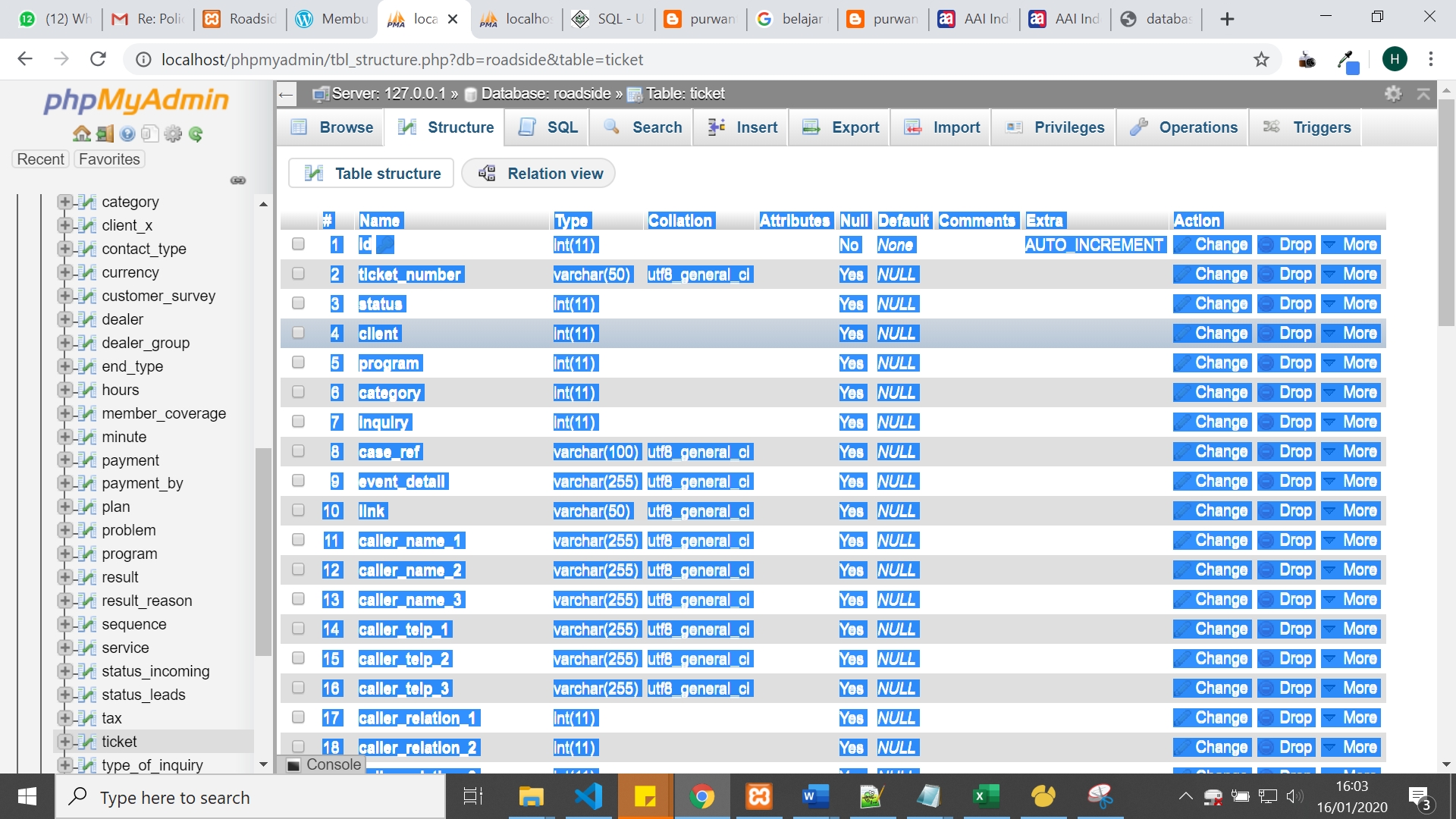
$id = $this->input->post(‘id’)

Jika ada 50 field, maka kita akan mengetikkan itu 1 demi 1. Dengan demikian kita akan memakan waktu banyak. Untuk mengatasi itu kita bisa pakai bantuan excel.

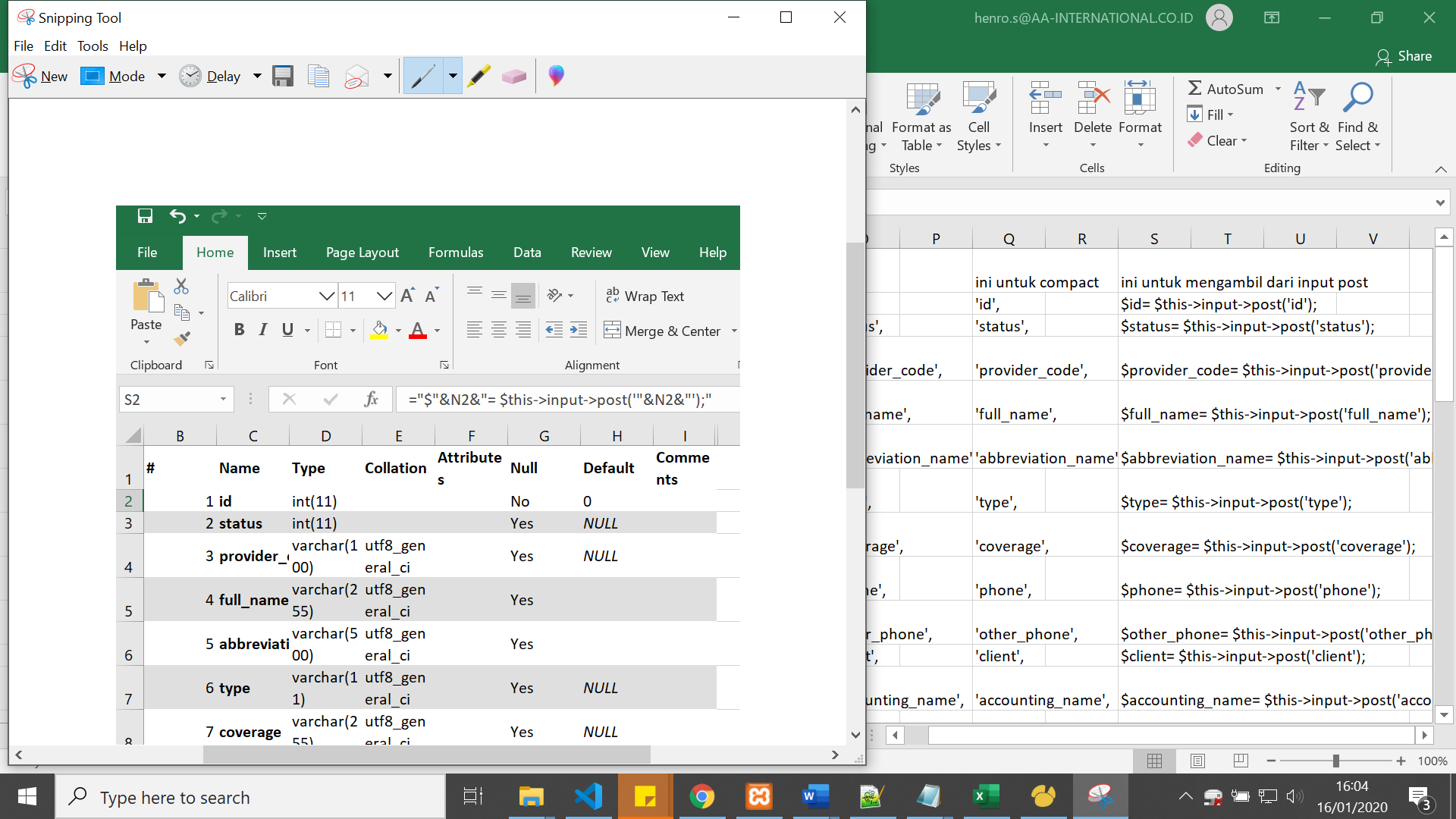
Langkahnya:

1. Copy field dari table di database kemudian copy ke excel:

- pilih tablenya -> structure -> kemudian select manual dan copy.



Copy ke excel



Kemudian kita buat rumus untuk menambahkan character dan semua seperti format untuk mengambil data valuenya sesuai codeighniter, seperti berikut:

="$"&C2&"= $this->input->post('"&C2&"');"

Penjelasan:

C2 adalah nama kolom dari name dari database. (pastikan name field di form selalu sama dengan nama field di database supaya lebih gampang untuk memprosesnya)

Nanti hasilnya akan seperti ini:

$id = $this->input->post('id');

$id= $this->input->post('id');

Setelah udah dapat 1 hasilnya, kita tinggal Tarik ke bawah Seperti fungsi excel biasanya.

Cara mengatasi jika di database ada nama field angka duluan yang dimana itu tidak akan bisa diakses ketika di php jika kita tampilkan resultnya ke dalam object. Untuk kita harus merubahnya ke array assosiative dulu biar bisa diambil nilainya.

Contoh: nama fieldnya adalah : 24\_hours. Jika kita ingin mengaksesnya dengan dengan object (ini saya alami di codeighniter), $this->provider->24\_hours, itu ada error. Untuk itulah kita harus mengkonvertnya ke array assosiative dulu.

$provid = (array) $provider

Mengecek sebuah array kosong:

empty($empty\_array)

cara cek sebuah nilai di array:

**in\_array**

contoh:

<?php  
$os = array("Mac", "NT", "Irix", "Linux");  
if (in\_array("Irix", $os)) {  
    echo "Got Irix";  
}  
if (in\_array("mac", $os)) {  
    echo "Got mac";  
}  
?>

Tapi dengan hanya code di atas, masih bisa terjadi bugs. Perhatikan contoh berikut ini:

$array = array(  
    'egg' => true,  
    'cheese' => false,  
    'hair' => 765,  
    'goblins' => null,  
    'ogres' => 'no ogres allowed in this array'  
);

Ketika kita cek dengan :

in\_array(null, $array); hasilnya true  
in\_array(false, $array); hasilnya true

in\_array('egg', $array); hasilnya true  
in\_array('hhh', $array); hasilnya true

yang 2 baris pertama masih masuk akal, tapi baris ke 3 dan ke 4 sudha tidak seperti yang kita harapkan.

Maka untuk solusinya adalah dengan menambahkan true diparameter ke 3 nya.

in\_array(null, $array, true); // true  
in\_array(false, $array, true); // true

in\_array(763, $array, true); // false  
in\_array('egg', $array, true); // false

cara cek keys di array

contoh:

<?php  
$search\_array = array('first' => 1, 'second' => 4);  
if (array\_key\_exists('first', $search\_array)) {  
    echo "The 'first' element is in the array";  
}  
?>

Mendapatkan key pertama di array:

<?php  
$array = ['a' => 1, 'b' => 2, 'c' => 3];  
  
$firstKey = array\_key\_first($array);  
  
var\_dump($firstKey);  
?>

Result:

string(1) "a"

mendapatkan semua key dari array:

**array\_keys** ( array $array ) : array

contoh:

<?php  
$array = array(0 => 100, "color" => "red");  
print\_r(array\_keys($array));

Hasil :

Array

(

[0] => 0

[1] => color

)

$array = array("blue", "red", "green", "blue", "blue");  
print\_r(array\_keys($array, "blue"));

Hasil:

Array

(

[0] => 0

[1] => 3

[2] => 4

)

Memanggil/menjalankan function langsung di array

**array\_map** ( [callable](https://www.php.net/manual/en/language.types.callable.php) $callback , array $array1 [, array $... ] ) : array

contoh:

<?php  
function cube($n)  
{  
    return ($n \* $n \* $n);  
}  
  
$a = [1, 2, 3, 4, 5];  
$b = array\_map('cube', $a);  
print\_r($b);  
?>

Hasilnya:

Array

(

[0] => 1

[1] => 8

[2] => 27

[3] => 64

[4] => 125

)

Kalau kita lihat diatas, itu menjalan pangkat 3 dari bilangan array.

Lambda di php:

<?php  
$func = **function($value) {**    return $value \* 2;  
};

Lambda itu sebenarnya sama dengan function aninous di javascript

Array map dengan lambda function:

<?php  
$func = function($value) {  
    return $value \* 2;  
};

print\_r(array\_map($func, range(1, 5)));

hasil:

Array

(

[0] => 2

[1] => 4

[2] => 6

[3] => 8

[4] => 10

)

Array\_map menggunakan lebih dari 1 array:

<?php  
function show\_Spanish($n, $m)  
{  
    return "The number {$n} is called {$m} in Spanish";  
}  
  
function map\_Spanish($n, $m)  
{  
    return [$n => $m];  
}  
  
$a = [1, 2, 3, 4, 5];  
$b = ['uno', 'dos', 'tres', 'cuatro', 'cinco'];  
  
$c = array\_map('show\_Spanish', $a, $b);  
print\_r($c);  
  
$d = array\_map('map\_Spanish', $a , $b);  
print\_r($d);  
?>

Hasil:

Array

(

[0] => The number 1 is called uno in Spanish

[1] => The number 2 is called dos in Spanish

[2] => The number 3 is called tres in Spanish

[3] => The number 4 is called cuatro in Spanish

[4] => The number 5 is called cinco in Spanish

)

Array

(

[0] => Array

(

[1] => uno

)

[1] => Array

(

[2] => dos

)

[2] => Array

(

[3] => tres

)

[3] => Array

(

[4] => cuatro

)

[4] => Array

(

[5] => cinco

)

)

Dari array di atas, kita bisa buat array dalam array dengan fungsi yang terakhir. Yaitu:

function map\_Spanish($n, $m)  
{  
    return [$n => $m];  
}  
  
$a = [1, 2, 3, 4, 5];  
$b = ['uno', 'dos', 'tres', 'cuatro', 'cinco'];

dari 2 array di atas kita bisa joinkan ke dalam array.

Array

(

[0] => Array

(

[1] => uno

)

[1] => Array

(

[2] => dos

)

[2] => Array

(

[3] => tres

)

[3] => Array

(

[4] => cuatro

)

[4] => Array

(

[5] => cinco

)

)

Menggabungkan 3 array menjadi sebuah array berdasarkan urutan:

<?php  
$a = [1, 2, 3, 4, 5];  
$b = ['one', 'two', 'three', 'four', 'five'];  
$c = ['uno', 'dos', 'tres', 'cuatro', 'cinco'];  
  
$d = array\_map(null, $a, $b, $c);  
print\_r($d);  
?>

Hasil:

Array

(

[0] => Array

(

[0] => 1

[1] => one

[2] => uno

)

[1] => Array

(

[0] => 2

[1] => two

[2] => dos

)

[2] => Array

(

[0] => 3

[1] => three

[2] => tres

)

[3] => Array

(

[0] => 4

[1] => four

[2] => cuatro

)

[4] => Array

(

[0] => 5

[1] => five

[2] => cinco

)

)

Mendapatkan IP address

$ip\_server = $\_SERVER['SERVER\_ADDR'];

**file\_exixsts**

file\_exists kita gunakan untuk mengecek suatu file ada atau tidak di dalam satu directory.

Contoh:

file\_exists(‘D\nama\_file.txt’);

jika file ada maka hasilnya akan 1 ketika di print ke layar.

<?php

$file = file\_exists('file\test.txt');

echo $file;

**Cara Mengirim Email Dengan XAMPP dari Localhost.**

1. Aktifkan **php\_openssl** di php.ini .

2. Cari fungsi email di php ini. Temukan code berikut:

*[mail function]*

*; XAMPP: Comment out this if you want to work with an SMTP Server like Mercury*

*; SMTP = localhost*

*; smtp\_port = 587*

*; For Win32 only.*

*; http://php.net/sendmail-from*

*;sendmail\_from = postmaster@localhost*

*;sendmail\_path = "\"C:\xampp\sendmail\sendmail.exe\" -t"*

Kemudian aktifkan settingan untuk: **SMTP, smtp\_port, sendmail\_from, sendmail\_path** dengan cara meng-uncomment mereka. Setelah itu ganti valuenya sesuai dengan value settingan kita. Jadi tampilan seperti berikut:

*; For Win32 only.*

*; http://php.net/smtp*

*[mail function]*

*; XAMPP: Comment out this if you want to work with an SMTP Server like Mercury*

SMTP = smtp.gmail.com

smtp\_port = 587

*; For Win32 only.*

*; http://php.net/sendmail-from*

sendmail\_from = henrotanjung@gmail.com

sendmail\_path = "\"C:\xampp\sendmail\sendmail.exe\" -t"

seperti yang kita bisa lihat diatas, saya memakai account gmail pribadi saya. Karena saya memakai account gmail, maka saya memakai smtp miliknya google mail(**smtp.gmail.com**). Berikut detail smtp google mail:

Outgoing Mail (SMTP) Server: smtp.gmail.com

Use Authentication: Yes

Username: akun Gmail Anda (emailanda@gmail.com)

Password: password akun Gmail Anda

Port (TLS): 587

Port (SSL): 465

Note:

User Secure Connection: Yes (ini bisa menjadi TLS atau SSL, tergantung dengan mail client yang Anda gunakan).

3. kemudian buka sendmail.ini yang ada di C:\xampp\sendmail\sendmail.ini

[mail function]

smtp\_server=smtp.gmail.com

smtp\_port=587

auth\_username=henrotanjung@gmail.com

auth\_password=XXXXXXX

force\_sender= [henrotanjung@gmail.com](mailto:henrotanjung@gmail.com)

4. Langkah Terakhir adalah, Restart webserver apache(XAMPP) kita.

Sampai langkah diatas kita sudah bisa test email kita sudah bisa test.

Buat sebuah file php dengan nama test­\_email.php letakkan di **htdocs/test\_email.php** . kemudian isikan filenya dengan code berikut:

<?php

mail("henrotanjung@gmail.com","Success","Send mail from localhost using PHP");

Kemudian jalankan kode tersebut dari browser kita.

<http://localhost/test_email.php>

Penyebab email tidak masuk ke penerima:

Mungkin email settingan pengirim membuat login dengan 2 authentacation atau mungkin juga karena less secure belum di aktifkan jika di gmail.

Untuk lihat penyebab errornya bisa di lihat di error\_log : C:/XAMPP/sendmail/error.log

**Namespace**

namespace kita gunakan ketika kita memiliki 2 kelas yang memiliki nama sama dalam 1 project. Kenapa kita bisa memiliki nama yang sama dalam 1 project? Jawabannya adalah:

1. Dalam 1 aplikasi kita bisa memiliki banyak module yang dimana di beberpa modulenya kita memerlukan kelas yang sama.

Contoh:

Sebuat aplikasi penjualan. Di aplikasi itu kita membuat module Sale dan Purchase. Disetiap module misalnya kita memiliki class user. Supaya tidak error (karena class yang sama), maka kita perlu yang Namanya **namespace.**

1. Didalam project yang dikerjain Bersama, misalnya project di git, orang lain tidak tahu kelas yang kita buat. Jadi kemungkinan mereka membuat kelas dengan nama yang sama dengan yang kita buat sebelumnya. Yang dimana seperti yang dijelaskan pada point no 1 diatas, itu akan membuat error. Untuk mengatasi itu, semua yang ikut dalam pemrograman tersebut harus menggunakan namespace mereka masing2 untuk setiap kelas yang di buat.

Cara penggunaan:

1. Declare
2. Pemanggilan

1. Declare

Ini merupakan dimana kita mendeklarisakan sebuah namespace untuk kelas kita. Aturan dalam pendeclarasian adalah:

1. Harus ada di line pertama (kecuali code pembuka <?php)

Contoh file1.php

<?php namespace Foo\Bar\subnamespace;

const FOO = 1;

function foo() {}

class foo

{

    static function staticmethod() {}

}

Atau

<?php

namespace Foo\Bar\subnamespace;

const FOO = 1;

function foo() {}

class foo

{

    static function staticmethod() {}

}

Kalau kita lihat ke dua snippet code diatas itu adalah sama, yang berbeda hanya baris penulisan declare namespacenya. Snippet code yang pertama ada satu baris dengan tag pembuka php, namun di snippet code yang kedua barisnya ada di line dibawah tag pembuka php.

2. Pemanggilan

Ada 3 type pemanggilan dari namespace yang sudah kita declare diatas. 3 cara yaitu:

1. Unqualified name (tidak memenuhi aturan) : ini adalah cara pemanggilan yang kurang memenuhi syarat dari namespace. Ini biasa kita gunakan jika pemanggiannya di file yang sama .

Contoh: Foo() ;

1. Qualified name : pemanggilan dengan memenuhi syarat dari namespace.

Contoh: Subnamespace\Foo() ;

1. Fully qualified name: dengna menyertakan lenkap sesua declare namespace.

Contoh: \Foo\Bar\foo() ;

Contoh:

Di file1.php.

<?php namespace Foo\Bar\subnamespace;

const FOO = 1;

function foo() {}

class foo

{

    static function staticmethod() {}

}

Kemudian kita buat file2.php untuk memanggilnya kelas dan attribute yang ada di kelas file1.php

<?php

namespace Foo\Bar;

include 'namespace1.php';

const FOO = 2;

function foo() {}

class foo

{

    static function staticmethod() {

        echo 'this is function on this class file';

    }

}

foo();

foo::staticmethod();

echo "<pre>";

echo FOO;

subnamespace\foo();

subnamespace\foo::staticmethod();

echo "<pre>";

echo Foo\Bar\subnamespace\FOO;

Penjelasan dari file2.php:

namespace Foo\Bar;

itu kita membuat declarasi juga mengenai class kita di file2, namun dengan demikian kita sudah bisa memanggil class lain tanpa mengikutkan kata Foo\Bar , jika nama namespace awalnya sama.

Contoh, kita bisa tidak menambahkan namespace itu di atas,

Tapi langsung panggil di bawah. Kalau kita lihat pada pemanggilna namespace\foo(); itu tidak usah dipanggil Foo\Bar. Itu karena diatas kita sudah buat namespace \Foo\Bar. Jika kita tidak buat itu di atas, dibawah (pada saat pemanggilan), kita harus buat completely pemanggilan seperti ini:

Foo\Bar\subnamespace\foo();

**Penggunaan use**

use kita gunakan untuk memanggil namespace yang sudah kita buat tadi. Kelebihan didalam use ini adalah kita bisa membuat alias. Dengan alias code kita bisa lebih singkat.

Contoh:

use Foo\Bar\subnamespace as f;

kemudian kita gunakan aliasnya (f) nya

echo f\FOO;

FOO adalah sebuah function di dalam namespace : Foo\Bar\subnamespace di atas.

Sebenarnya kita bisa menggunakan namespace untuk memanggil satu fungsi declarasi namespace.

Contoh:

File1.php memiliki namespace Foo\Bar\subnamespace, kemudian kita ingin memanggil ini dari file2.php. maka pertama buat declare namespacenya dimana namespace ada kesamaan dengan yang kita buat tadi, kemudian kita include dulu: include ‘file1.php’.

namespace Foo\Bar;

include 'namespace1.php';

kemudian kita bisa menaggil property ataupun methode:

subnamespace\foo();

dari pemaggilannya kita tidak perlu harus menambahkan Foo\Bar seperti di atas, karena sudah kita declare di atas. Jadi kenapa kita harus menggunakan use?

Jawabnya:

1. use kita bisa gunakan di baris berapa saja, sedangkan namespace tidak
2. namespace akan membuat 1 pointer untuk file dimana dia dibuat, sedangkan use hanya akan memanggil namespace yg Sudah dibuat.
3. Use bisa dialias yg dimana akan membuat code lebih singkat

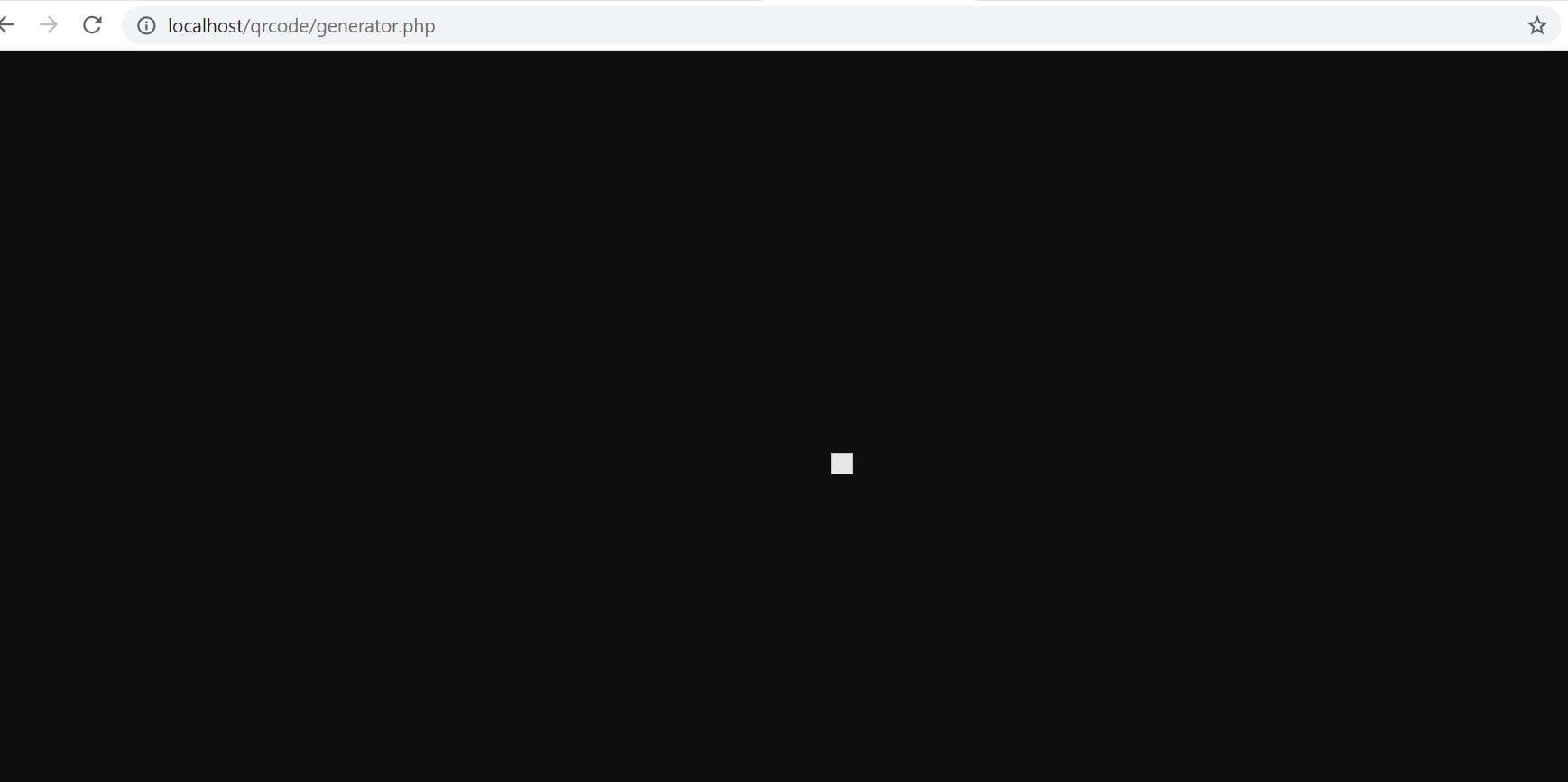
**Filemtime()**

filemtime kita gunakan untuk mendapatkan waktu terakhir modifikasi sebuah file.

Jika kita ingin membuat qrcode dengan library php, pastikan sebelum perintah:

QRcode::png($codeContents);

Pastikan code sebelum baris itu, tidak boleh ada perintah output. Seperti echo, dll, termasuk di line script yang di include (cth: include “koneksi.php”). Sebab kalau itu ada, maka tampilan barcode nya tidak akan benar.  
bisa jadi seperti ini:



Jika diperhatikan di scriptnya, karena ada perintah echo sebelum perintah generate qrcodenya (line 4):



error tidak bisa jalankan server mysql xampp

https://stackoverflow.com/questions/56767200/cant-run-xampp-mysql

incorrect format parameter

https://stackoverflow.com/questions/50690076/phpmyadmin-error-incorrect-format-parameter

php - Unknown: Failed opening required on line 0

di Codeigniter ini muncul ketika kita menjalankan server dari terminal dan mengakse website dari browser http://localhost:8080

ini penyebabnya karena server.php kita terhapus, mungkin karena anti virus

https://stackoverflow.com/questions/49574906/php-unknown-failed-opening-required-on-line-0-laravel-5-6

solusinya:

kita lebih bagus install ulang xampp