

# LAMPIRAN

## 1. Connection Database

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

$active_group = 'default';
$query_builder = TRUE;

$db['default'] = array(
    'dsn' => "",
    'hostname' => 'localhost',
    'username' => 'root',
    'port' => '3306',
    'password' => "",
    'database' => 'showroom',
    'dbdriver' => 'mysqli',
    'dbprefix' => "",
    'pconnect' => FALSE,
    'db_debug' => (ENVIRONMENT !== 'production'),
    'cache_on' => FALSE,
    'cachedir' => "",
    'char_set' => 'utf8',
    'dbcollat' => 'utf8_general_ci',
    'swap_pre' => "",
    'encrypt' => FALSE,
    'compress' => FALSE,
    'stricton' => FALSE,
    'failover' => array(),
    'save_queries' => TRUE,
);
```

## 2. Class Admin

```
<?php

/**
 *
 */

class Admin extends CI_Controller {

    function __construct() {
        parent::__construct();
        $this->load->helper('tglindo');
        $this->load->helper('rupiah');
        if ($this->session->userdata('status') != "login") {
            redirect(base_url("login"));
        }
    }

    function index() {
        $this->load->view('crud-admin/index');
    }

    function dashboard() {
        $this->load->view('crud-admin/index');
    }

    function sort($start, $end) {
        header('Content-Type: application/json');

        $data['list'] = $this->db
            ->from('Pesan')
            ->where('created_at','>=', $start)
```

```
        ->where('created_at', '<=', $end)
        ->get()->result_array();
$this->load->view('crud-admin/pesan', $data);

    echo json_encode($data);
}

function printToPdf()
{
    $tahun = $this->input->post('tahun');

    if (!empty($tahun)) {

        ob_start();

        $this->db->select('*');
        $this->db->from('pesan');
        $this->db->where("YEAR(created_at)", $tahun);
        $query['pesan'] = $this->db->get()->result();

        $this->load->view('print', $query);
        $html = ob_get_contents();
        ob_end_clean();

        require_once('./assets/html2pdf/html2pdf.class.php');
        $pdf = new HTML2PDF('P', 'A4', 'en');
        $pdf->WriteHTML($html);
        $pdf->Output('Data LJM.pdf', 'D');
    }
}
```

```

function pesan() {
    $data['list'] = $this->db
        ->from('Pesan')
        ->order_by('created_at', 'DESC')
        ->get()->result_array();
    $this->load->view('crud-admin/pesan', $data);
}

function view_by_date($date){
    $this->db->where('created_at(tgl)', $date); // Tambahkan where tanggal nya

    return $this->db->get('transaksi')->result(); // Tampilkan data transaksi sesuai tanggal yang
    diinput oleh user pada filter
}

function view_by_month($month, $year){
    $this->db->where('MONTH(tgl)', $month); // Tambahkan where bulan
    $this->db->where('YEAR(tgl)', $year); // Tambahkan where tahun

    return $this->db->get('transaksi')->result(); // Tampilkan data transaksi sesuai bulan dan tahun
    yang diinput oleh user pada filter
}

function view_by_year($year){
    $this->db->where('YEAR(tgl)', $year); // Tambahkan where tahun

    return $this->db->get('transaksi')->result(); // Tampilkan data transaksi sesuai tahun yang
    diinput oleh user pada filter
}

function view_all(){
    return $this->db->get('transaksi')->result(); // Tampilkan semua data transaksi
}

```

```
function option_tahun(){
    $this->db->select('YEAR(tgl) AS tahun'); // Ambil Tahun dari field tgl
    $this->db->from('transaksi'); // select ke tabel transaksi
    $this->db->order_by('YEAR(tgl)'); // Urutkan berdasarkan tahun secara Ascending (ASC)
    $this->db->group_by('YEAR(tgl)'); // Group berdasarkan tahun pada field tgl

    return $this->db->get()->result(); // Ambil data pada tabel transaksi sesuai kondisi diatas
}

```

```
function mobil() {
    $data['list'] = $this->db->
        select(
            '
Mobil.id as id,
Mobil.judul as judul,
Merk.nama as merk,
Model.nama as model,
Varian.nama as varian,
Transmisi.nama as transmisi,
Mobil.tahun as tahun,
Mobil.cakupan_mesin as cakupan_mesin,
Mobil.deskripsi as deskripsi,
Mobil.rekomendasi as rekomendasi,
Tipe.nama as tipe,
Kilometer.nama as km,
Mobil.kondisi as kondisi,
Mobil.harga as harga
'
        )
    ->from('Mobil')
}

```

```

->join('Merk', 'Merk.id = Mobil.merk')
->join('Model', 'Model.id = Mobil.model')
->join('Varian', 'Varian.id = Mobil.varian')
->join('Transmisi', 'Transmisi.id = Mobil.transmisi')
->join('Kilometer', 'Kilometer.id = Mobil.km')
->join('Tipe', 'Tipe.id = Mobil.tipe')
->order_by('Mobil.created_at', 'DESC')
->get()->result_array();
$this->load->view('crud-admin/mobil', $data);
}

function create_mobil() {

    $this->form_validation->set_rules('judul', 'Judul harus diisi', 'required');
    $this->form_validation->set_rules('tahun', 'Tahun harus diisi', 'required');
    $this->form_validation->set_rules('cakupan_mesin', 'Cakupan Mesin harus diisi',
required');
    $this->form_validation->set_rules('harga', 'Harga harus diisi', 'required');
    $this->form_validation->set_rules('deskripsi', 'Deskripsi harus diisi', 'required');
    $this->form_validation->set_rules('merk', 'Merk harus diisi', 'required');
    $this->form_validation->set_rules('model', 'Model harus diisi', 'required');
    $this->form_validation->set_rules('varian', 'Varian harus diisi', 'required');
    $this->form_validation->set_rules('tipe', 'Tipe harus diisi', 'required');
    $this->form_validation->set_rules('transmisi', 'Transmisi harus diisi', 'required');
    $this->form_validation->set_rules('km', 'Kilometer harus diisi', 'required');

    if ($this->form_validation->run() == FALSE) {
        $data['merk'] = $this->db->from('Merk')->get()->result();
        $data['tipe'] = $this->db->from('Tipe')->get()->result();
        $data['transmisi'] = $this->db->from('Transmisi')->get()->result();
        $data['km'] = $this->db->from('Kilometer')->get()->result();
        $this->load->view('crud-admin/tambah-mobil', $data);
    }
}

```

```

} else {
    // Ambil nilai form
    $data['judul'] = $this->input->post('judul');
    $data['tahun'] = $this->input->post('tahun');
    $data['cakupan_mesin'] = $this->input->post('cakupan_mesin');
    $data['harga'] = $this->input->post('harga');
    $data['deskripsi'] = $this->input->post('deskripsi');
    $data['merk'] = $this->input->post('merk');
    $data['model'] = $this->input->post('model');
    $data['varian'] = $this->input->post('varian');
    $data['tipe'] = $this->input->post('tipe');
    $data['transmisi'] = $this->input->post('transmisi');
    $data['km'] = $this->input->post('km');
    $data['kondisi'] = $this->input->post('kondisi');
    $data['rekomendasi'] = $this->input->post('rekomendasi');
    $data['video'] = $this->input->post('video');
    $this->db->insert('Mobil', $data);
    $id = $this->db->insert_id();
    if ($id) {
        $files = $_FILES;
        $cpt = count($_FILES['files']['name']);
        for ($i = 0; $i < $cpt; $i++) {
            $_FILES['file']['name'] = $_FILES['files']['name'][$i];
            $_FILES['file']['type'] = $_FILES['files']['type'][$i];
            $_FILES['file']['tmp_name'] =
$_FILES['files']['tmp_name'][$i];

            $_FILES['file']['error'] = $_FILES['files']['error'][$i];
            $_FILES['file']['size'] = $_FILES['files']['size'][$i];

            // Set preference
            $config['upload_path'] = 'img/';

```

```

$config['allowed_types'] = 'jpg|jpeg|png';
$config['max_size'] = '5000'; // max_size in kb
$config['file_name'] = $_FILES['files']['name'][$i];

//Load upload library
$this->load->library('upload', $config);
if ($this->upload->do_upload('file')) {
    // Get data about the file
    $uploadData = $this->upload->data();
    $filename = $uploadData['file_name'];

    // Initialize array
    $insertgambar['mobil'] = $id;
    $insertgambar['image'] = $filename;
    $this->db->insert('Gambar', $insertgambar);
}

}

redirect(base_url('admin/mobil'), 'refresh');

} else {
    $data['merk'] = $this->db->from('Merk')->get()->result();
    $data['tipe'] = $this->db->from('Tipe')->get()->result();
    $data['transmisi'] = $this->db->from('Transmisi')->get()-
>result();

    $data['km'] = $this->db->from('Kilometer')->get()->result();
    $error = array('error' => 'judul yang anda masukkan sudah
ada.');
```

```

    $this->load->view('crud-admin/tambah-mobil', $error);
}
}

```



```

    }

function tambahmobil() {
    $data['merk'] = $this->db->from('Merk')->get()->result();
    $data['tipe'] = $this->db->from('Tipe')->get()->result();
    $data['transmisi'] = $this->db->from('Transmisi')->get()->result();
    $data['km'] = $this->db->from('Kilometer')->get()->result();
    $this->load->view('crud-admin/tambah-mobil', $data);
}

// Hapus Berita
function hapusmobil($id = NULL) {
    $this->db->where('id', $id);
    $this->db->delete('Mobil');

    redirect(base_url('admin/mobil'), 'refresh');
}

function jualmobil($id = NULL) {
    $this->db->set('kondisi', 'Terjual');
    $this->db->where('id', $id);
    $this->db->update('Mobil');

    redirect(base_url('admin/mobil'), 'refresh');
}
}

```

### 3. *Class Home*

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Home extends CI_Controller {

```

```
function __construct() {
    parent::__construct();
    $this->load->helper('rupiah');
}

public function contact() {
    $this->load->view('contact');
}

public function about() {
    $this->load->view('about');
}

public function create_pesanan() {
    $data['nama'] = $this->input->post('nama');
    $data['email'] = $this->input->post('email');
    $data['pesan'] = $this->input->post('pesan');
    $data['phone'] = $this->input->post('phone');
    $this->db->insert('Pesanan', $data);
    $this->load->view('contact', $data);
}

public function index() {
    $data['rekomendasi'] = $this->db->
        select(
            '
Mobil.id as id,
Mobil.kondisi as kondisi,
Mobil.tahun as tahun,
```

```

Mobil.judul as judul,
Mobil.cakupan_mesin as cakupan_mesin,
Mobil.deskripsi as deskripsi,
Mobil.harga as harga,
Mobil.created_at as created_at,
Mobil.rekomendasi as rekomendasi,
Merk.nama as merk,
Model.nama as model,
Varian.nama as varian,
Transmisi.nama as transmisi,
Tipe.nama as tipe,
Kilometer.nama as km

```

```

,
```

```

)
```

```

->from('Mobil')
->join('Merk', 'Merk.id = Mobil.merk')
->join('Model', 'Model.id = Mobil.model')
->join('Varian', 'Varian.id = Mobil.varian')
->join('Transmisi', 'Transmisi.id = Mobil.transmisi')
->join('Kilometer', 'Kilometer.id = Mobil.km')
->join('Tipe', 'Tipe.id = Mobil.tipe')
->where('Mobil.rekomendasi', 1)
->where('Mobil.kondisi !=', 'Terjual')
->order_by('Mobil.created_at', 'DESC')
->limit(9)
->get()->result_array();

```

```

$data['terbaru'] = $this->db->

```

```

select(

```

```

,
```

```

Mobil.id as id,

```

```

Mobil.kondisi as kondisi,
Mobil.tahun as tahun,
Mobil.judul as judul,
Mobil.cakupan_mesin as cakupan_mesin,
Mobil.deskripsi as deskripsi,
Mobil.harga as harga,
Mobil.created_at as created_at,
Mobil.rekomendasi as rekomendasi,
Merk.nama as merk,
Model.nama as model,
Varian.nama as varian,
Transmisi.nama as transmisi,
Tipe.nama as tipe,
Kilometer.nama as km
,
)

->from('Mobil')
->join('Merk', 'Merk.id = Mobil.merk')
->join('Model', 'Model.id = Mobil.model')
->join('Varian', 'Varian.id = Mobil.varian')
->join('Transmisi', 'Transmisi.id = Mobil.transmisi')
->join('Kilometer', 'Kilometer.id = Mobil.km')
->join('Tipe', 'Tipe.id = Mobil.tipe')
->where('Mobil.kondisi !=', 'Terjual')
->order_by('Mobil.created_at', 'DESC')
->limit(9)
->get()->result_array();

$data['alltipe'] = $this->db->from('Tipe')->get()->result_array();
$this->load->view('beranda', $data);
}

```

```

public function rekomendasi() {
    $data['list'] = $this->db->
        select(
            '
Mobil.id as id,
Mobil.kondisi as kondisi,
Mobil.tahun as tahun,
Mobil.judul as judul,
Mobil.cakupan_mesin as cakupan_mesin,
Mobil.deskripsi as deskripsi,
Mobil.harga as harga,
Mobil.created_at as created_at,
Mobil.rekomendasi as rekomendasi,
Merk.nama as merk,
Model.nama as model,
Varian.nama as varian,
Transmisi.nama as transmisi,
Tipe.nama as tipe,
Kilometer.nama as km
            '
        )
    ->from('Mobil')
    ->join('Merk', 'Merk.id = Mobil.merk')
    ->join('Model', 'Model.id = Mobil.model')
    ->join('Varian', 'Varian.id = Mobil.varian')
    ->join('Transmisi', 'Transmisi.id = Mobil.transmisi')
    ->join('Kilometer', 'Kilometer.id = Mobil.km')
    ->join('Tipe', 'Tipe.id = Mobil.tipe')
    ->where('Mobil.rekomendasi', 1)
    ->where('Mobil.kondisi !=', 'Terjual')
}

```

```

->order_by('Mobil.created_at', 'DESC')
->limit(9)
->get()->result_array();

$data['page_title'] = 'Rekomendasi';
$data['alltipe'] = $this->db->from('Tipe')->get()->result_array();
$this->load->view('general_view', $data);
}

public function terbaru() {
    $data['list'] = $this->db->
        select(
            '
Mobil.id as id,
Mobil.kondisi as kondisi,
Mobil.tahun as tahun,
Mobil.judul as judul,
Mobil.cakupan_mesin as cakupan_mesin,
Mobil.deskripsi as deskripsi,
Mobil.harga as harga,
Mobil.created_at as created_at,
Mobil.rekomendasi as rekomendasi,
Merk.nama as merk,
Model.nama as model,
Varian.nama as varian,
Transmisi.nama as transmisi,
Tipe.nama as tipe,
Kilometer.nama as km
            '
        )
    ->from('Mobil')

```

```

->join('Merk', 'Merk.id = Mobil.merk')
->join('Model', 'Model.id = Mobil.model')
->join('Varian', 'Varian.id = Mobil.varian')
->join('Transmisi', 'Transmisi.id = Mobil.transmisi')
->join('Kilometer', 'Kilometer.id = Mobil.km')
->join('Tipe', 'Tipe.id = Mobil.tipe')
->where('Mobil.kondisi !=', 'Terjual')
->order_by('Mobil.created_at', 'DESC')
->get()->result_array();

```

```

$data['page_title'] = 'Mobil Terbaru';
$data['alltipe'] = $this->db->from('Tipe')->get()->result_array();
$this->load->view('general_view', $data);

```

```

}

```

```

public function detail($id = NULL) {
    $data['mobil'] = (array) $this->db->
        select(
            '
Mobil.id as id,
Mobil.kondisi as kondisi,
Mobil.tahun as tahun,
Mobil.judul as judul,
Mobil.cakupan_mesin as cakupan_mesin,
Mobil.deskripsi as deskripsi,
Mobil.harga as harga,
Mobil.created_at as created_at,
Mobil.rekomendasi as rekomendasi,
Mobil.video as video,
Merk.nama as merk,
Model.nama as model,

```

```

        Varian.nama as varian,
        Transmisi.nama as transmisi,
        Tipe.nama as tipe,
        Kilometer.nama as km
    ,
    )

->from('Mobil')
->join('Merk', 'Merk.id = Mobil.merk')
->join('Model', 'Model.id = Mobil.model')
->join('Varian', 'Varian.id = Mobil.varian')
->join('Transmisi', 'Transmisi.id = Mobil.transmisi')
->join('Kilometer', 'Kilometer.id = Mobil.km')
->join('Tipe', 'Tipe.id = Mobil.tipe')
->where('Mobil.id', $id)
->get()->row();

    $data['mobil']['images'] = $this->db->from('Gambar')->where('mobil', $id)-
>get()->result_array();

    $this->load->view('detail', $data);
}

public function tipe($id = NULL) {
    $data['list'] = (array) $this->db->
        select(
            ,

Mobil.id as id,
Mobil.kondisi as kondisi,
Mobil.tahun as tahun,
Mobil.judul as judul,
Mobil.cakupan_mesin as cakupan_mesin,
Mobil.deskripsi as deskripsi,
Mobil.harga as harga,
Mobil.created_at as created_at,

```



```

Mobil.rekomendasi as rekomendasi,
Mobil.video as video,
Merk.nama as merk,
Model.nama as model,
Varian.nama as varian,
Transmisi.nama as transmisi,
Tipe.nama as tipe,
Kilometer.nama as km
,
    )

->from('Mobil')
->join('Merk', 'Merk.id = Mobil.merk')
->join('Model', 'Model.id = Mobil.model')
->join('Varian', 'Varian.id = Mobil.varian')
->join('Transmisi', 'Transmisi.id = Mobil.transmisi')
->join('Kilometer', 'Kilometer.id = Mobil.km')
->join('Tipe', 'Tipe.id = Mobil.tipe')
->where('Mobil.kondisi !=', 'Terjual')
->where('Tipe.id', $id)
->get()->result_array();

$data['page_title'] = $this->db->from('Tipe')->where('id', $id)->get()->row()-
>nama;

$data['alltipe'] = $this->db->from('Tipe')->get()->result_array();
$this->load->view('general_view', $data);
}
}

```

#### 4. Class Login

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

```

```
class Login extends CI_Controller {

    public function __construct() {

        parent::__construct();
        $this->load->model('admin');
        $this->load->model('M_user');
    }

    // List all your items
    public function index() {
        $this->load->view('crud-admin/login');
    }

    public function login_admin() {
        $username = $this->input->post('username');
        $password = $this->input->post('password');

        $where = array(
            'username' => $username,
        );

        $cek = $this->admin->cek_login("Admin", $where)->row();

        if (!empty($cek)) {
            if (password_verify($password, $cek->password)) {
                $data_session = array(
                    'nama' => $cek->nama,
                    'status' => "login",
                );
            }
        }
    }
}
```

```

        $this->session->set_userdata($data_session);
        redirect(base_url('admin/dashboard'));
    } else {
        $this->session->set_flashdata('error', 'Username dan password
salah');

        $this->load->view('crud-admin/login');
    }

    } else {
        $this->session->set_flashdata('error', 'Username dan password salah');
        $this->load->view('crud-admin/login');
    }
}

function logout() {
    $this->session->sess_destroy();
    redirect(base_url('admin'));
}
}

```

## 5. *Class Master*

```

<?php

/**
 *
 */

class Master extends CI_Controller {

    function __construct() {
        parent::__construct();
        $this->load->helper('tglindo');
        $this->load->helper('rupiah');
        if ($this->session->userdata('status') != "login") {

```

```

        redirect(base_url("login"));
    }
}

function modeloption($id) {

    $data = $this->db->from('Model')->where('merk', $id)->get()->result();
    $output = '<option value="">Pilih Model</option>';
    foreach ($data as $row) {
        $output .= '<option value="" . $row->id . ">' . $row->nama .
'</option>';
    }
    echo $output;
}

function varianoption($id) {

    $data = $this->db->from('Varian')->where('model', $id)->get()->result();
    $output = '<option value="">Pilih Varian</option>';
    foreach ($data as $row) {
        $output .= '<option value="" . $row->id . ">' . $row->nama .
'</option>';
    }
    echo $output;
}

function merk() {
    $data['list'] = $this->db
        ->from('Merk')
        ->get()->result_array();
    $this->load->view('crud-admin/merk', $data);
}

function model() {
    $data['list'] = $this->db->select('Model.id as id,Model.nama as nama,Merk.nama
as merk')
        ->from('Model')

```

```

        ->join('Merk', 'Merk.id = Model.merk')
        ->get()->result_array();
    $this->load->view('crud-admin/model', $data);
}
function varian() {
    $data['list'] = $this->db->select('Varian.id as id,Varian.nama as
nama,Model.nama as model')
        ->from('Varian')
        ->join('Model', 'Model.id = Varian.model')
        ->get()->result_array();
    $this->load->view('crud-admin/varian', $data);
}
function tipemobil() {
    $data['list'] = $this->db
        ->from('Tipe')
        ->get()->result_array();
    $this->load->view('crud-admin/tipemobil', $data);
}
function hapusmerk($id = NULL) {
    $this->db->where('id', $id);
    $this->db->delete('Merk');

    redirect(base_url('master/merk'), 'refresh');
}
function hapusmodel($id = NULL) {
    $this->db->where('id', $id);
    $this->db->delete('Model');

    redirect(base_url('master/model'), 'refresh');
}
function hapusvarian($id = NULL) {
    $this->db->where('id', $id);

```

```

        $this->db->delete('Varian');

        redirect(base_url('master/varian'), 'refresh');
    }
    function hapustipemobil($id = NULL) {
        $this->db->where('id', $id);
        $this->db->delete('Tipe');

        redirect(base_url('master/tipemobil'), 'refresh');
    }
    public function create_merk() {
        $data['nama'] = $this->input->post('nama');
        $this->db->insert('Merk', $data);
        redirect(base_url('master/merk'), 'refresh');
    }
    public function create_model() {
        $data['nama'] = $this->input->post('nama');
        $data['merk'] = $this->input->post('merk');
        $this->db->insert('Model', $data);
        redirect(base_url('master/model'), 'refresh');
    }
    public function create_varian() {
        $data['nama'] = $this->input->post('nama');
        $data['model'] = $this->input->post('model');
        $this->db->insert('Varian', $data);
        redirect(base_url('master/varian'), 'refresh');
    }
    public function create_tipemobil() {
        $data['nama'] = $this->input->post('nama');
        $this->db->insert('Tipe', $data);
        redirect(base_url('master/tipemobil'), 'refresh');
    }

```

```
}  
public function tambahmerk() {  
    $this->load->view('crud-admin/tambah-merk');  
}  
public function tambahmodel() {  
    $data['merk'] = $this->db->from('Merk')->get()->result();  
    $this->load->view('crud-admin/tambah-model', $data);  
}  
public function tambahvarian() {  
    $data['merk'] = $this->db->from('Merk')->get()->result();  
    $this->load->view('crud-admin/tambah-varian', $data);  
}  
public function tambahtipemobil() {  
    $this->load->view('crud-admin/tambah-tipemobil');  
}  
}
```