

LAMPIRAN KODINGAN

USER.PHP

```

<?php
if (!defined('BASEPATH'))
    exit('No direct script access allowed');
class Users extends CI_Controller
{
    function __construct()
    {
        parent::__construct();
        check_not_login();
        check_admin();
        $this->load->model('Users_model');
        $this->load->library('form_validation');
        $this->load->model('user_level_model');
    }
    public function index()
    {
        $q = urldecode($this->input->get('q', TRUE));
        $start = intval($this->uri->segment(3));
        if ($q <> "") {
            $config['base_url'] = base_url() . '.php/c_url/index.html?q=' . urlencode($q);
            $config['first_url'] = base_url() . 'index.php/users/index.html?q=' .
urlencode($q);
        } else {
            $config['base_url'] = base_url() . 'index.php/users/index/';
            $config['first_url'] = base_url() . 'index.php/users/index/';
        }
        $config['per_page'] = 10;
        $config['page_query_string'] = FALSE;
        $config['total_rows'] = $this->Users_model->total_rows($q);
        $users = $this->Users_model->get_limit_data($config['per_page'], $start, $q);
        $config['full_tag_open'] = '<ul class="pagination pagination-sm no-margin
pull-right">';
        $config['full_tag_close'] = '</ul>';
        $this->load->library('pagination');
        $this->pagination->initialize($config);

        $data = array(
            'users_data' => $users,
            'q' => $q,
            'pagination' => $this->pagination->create_links(),

            'total_rows' => $config['total_rows'],

```

```

        'start' => $start,
    );
    $this->template->load('template', 'users/users_list', $data);
}

public function read($id)
{
    $row = $this->Users_model->get_by_id($id);
    if ($row) {
        $data = array(
            'id_username' => $row->id_username,
            'username' => $row->username,
            'nama' => $row->nama,
            'email' => $row->email,
            'alamat' => $row->alamat,
            'kota' => $row->kota,
            'provinsi' => $row->provinsi,
            'telepon' => $row->telepon,
            'id_level' => $row->id_level,
            'is_aktif' => $row->is_aktif,
            'create_date' => $row->create_date,
        );
        $this->template->load('template', 'users/users_read', $data);
    } else {
        $this->session->set_flashdata('message', 'Record Not Found');
        redirect(site_url('users'));
    }
}

public function create()
{
    $data = array(
        'button' => 'Create',
        'action' => site_url('users/create_action'),
        'id_username' => set_value('id_username'),
        'username' => set_value('username'),
        'password' => set_value('password'),
        'nama' => set_value('nama'),
        'email' => set_value('email'),
        'alamat' => set_value('alamat'),
        'kota' => set_value('kota'),
        'provinsi' => set_value('provinsi'),
        'telepon' => set_value('telepon'),
        'id_level' => set_value('id_level'),
        'is_aktif' => set_value('is_aktif'),
        'create_date' => set_value('create_date'),
    );
}

```

```

    );
    $data['coba'] = $this->user_level_model->tampil_level();
    $this->template->load('template', 'users/users_form', $data);
}

public function create_action()
{
    $this->_rules();

    if ($this->form_validation->run() == FALSE) {
        $this->create();
    } else {
        $plain_password = $this->input->post('password', TRUE); // Ambil
password asli
        $data = array(
            'username' => $this->input->post('username', TRUE),
            'password' => md5($plain_password), // Enkripsi password
            'password_plain' => $plain_password, // Simpan password asli
            'nama' => $this->input->post('nama', TRUE),
            'email' => $this->input->post('email', TRUE),
            'alamat' => $this->input->post('alamat', TRUE),
            'kota' => $this->input->post('kota', TRUE),
            'provinsi' => $this->input->post('provinsi', TRUE),
            'telepon' => $this->input->post('telepon', TRUE),
            'id_level' => $this->input->post('id_level', TRUE),
            'is_aktif' => $this->input->post('is_aktif', TRUE),
            'create_date' => date('y-m-d H:i:s')
        );

        $this->Users_model->insert($data);
        $this->session->set_flashdata('message', 'Create Record Success');
        redirect(site_url('users'));
    }
}

public function update($id)
{
    $row = $this->Users_model->get_by_id($id);

    if ($row) {
        $data = array(
            'button' => 'Update',
            'action' => site_url('users/update_action'),
            'id_username' => set_value('id_username', $row->id_username),
            'username' => set_value('username', $row->username),
            'password' => set_value('password', $row->password),

```

```

        'nama' => set_value('nama', $row->nama),
        'email' => set_value('email', $row->email),
        'alamat' => set_value('alamat', $row->alamat),
        'kota' => set_value('kota', $row->kota),
        'provinsi' => set_value('provinsi', $row->provinsi),
        'telepon' => set_value('telepon', $row->telepon),
        'id_level' => set_value('id_level', $row->id_level),
        'is_aktif' => set_value('is_aktif', $row->is_aktif),
        'create_date' => set_value('create_date', $row->create_date),
    );
    $data['coba'] = $this->user_level_model->tampil_level();
    $data['data_user'] = $this->Users_model->edit_data($id);
    $this->template->load('template', 'users/users_form', $data);
} else {
    $this->session->set_flashdata('message', 'Record Not Found');
    redirect(site_url('users'));
}
}
public function update_action()
{
    $this->_rules();

    if ($this->form_validation->run() == FALSE) {
        $this->update($this->input->post('id_username', TRUE));
    } else {
        $plain_password = $this->input->post('password', TRUE); // Ambil
password asli
        $data = array(
            'username' => $this->input->post('username', TRUE),
            'password' => md5($plain_password), // Enkripsi password
            'password_plain' => $plain_password, // Simpan password asli
            'nama' => $this->input->post('nama', TRUE),
            'email' => $this->input->post('email', TRUE),
            'alamat' => $this->input->post('alamat', TRUE),
            'kota' => $this->input->post('kota', TRUE),
            'provinsi' => $this->input->post('provinsi', TRUE),
            'telepon' => $this->input->post('telepon', TRUE),
            'id_level' => $this->input->post('id_level', TRUE),
            'is_aktif' => $this->input->post('is_aktif', TRUE),
        );

        $this->Users_model->update($this->input->post('id_username', TRUE),
$data);
        $this->session->set_flashdata('message', 'Update Record Success');
        redirect(site_url('users'));
    }
}

```

```

    }

    public function delete($id)
    {
        $row = $this->Users_model->get_by_id($id);

        if ($row) {
            $this->Users_model->delete($id);
            $this->session->set_flashdata('message', 'Delete Record Success');
            redirect(site_url('users'));
        } else {
            $this->session->set_flashdata('message', 'Record Not Found');
            redirect(site_url('users'));
        }
    }

    public function _rules()
    {
        $this->form_validation->set_rules('username', 'username', 'trim|required');
        if ($this->uri->segment(2) == 'create' || $this->uri->segment(2) ==
'create_action') {
            $this->form_validation->set_rules('password', 'password', 'trim|required');
        }

        $this->form_validation->set_rules('nama', 'nama', 'trim|required');
        $this->form_validation->set_rules('email', 'email', 'trim|required');
        // $this->form_validation->set_rules('alamat', 'alamat', 'trim|required');
        // $this->form_validation->set_rules('kota', 'kota', 'trim|required');
        // $this->form_validation->set_rules('provinsi', 'provinsi', 'trim|required');
        // $this->form_validation->set_rules('telepon', 'telepon', 'trim|required');
        $this->form_validation->set_rules('id_level', 'id level', 'trim|required');
        $this->form_validation->set_rules('is_aktif', 'is aktif', 'trim|required');

        $this->form_validation->set_rules('id_username', 'id_username', 'trim');
        $this->form_validation->set_error_delimiters('<span class="text-danger">',
'</span>');
    }

    public function excel()
    {
        $this->load->helper('exportexcel');
        $namaFile = "users.xls";
        $judul = "users";
        $tablehead = 0;
        $tablebody = 1;
        $nourut = 1;
    }

```

```

//penulisan header
header("Pragma: public");
header("Expires: 0");
header("Cache-Control: must-revalidate, post-check=0,pre-check=0");
header("Content-Type: application/force-download");
header("Content-Type: application/octet-stream");
header("Content-Type: application/download");
header("Content-Disposition: attachment;filename=" . $namaFile . "");
header("Content-Transfer-Encoding: binary ");

xlsBOF();

$kolomhead = 0;
xlsWriteLabel($tablehead, $kolomhead++, "No");
xlsWriteLabel($tablehead, $kolomhead++, "Username");
xlsWriteLabel($tablehead, $kolomhead++, "Password");
xlsWriteLabel($tablehead, $kolomhead++, "Nama");
xlsWriteLabel($tablehead, $kolomhead++, "Email");
xlsWriteLabel($tablehead, $kolomhead++, "Alamat");
xlsWriteLabel($tablehead, $kolomhead++, "Kota");
xlsWriteLabel($tablehead, $kolomhead++, "Provinsi");
xlsWriteLabel($tablehead, $kolomhead++, "Telepon");
xlsWriteLabel($tablehead, $kolomhead++, "Id Level");
xlsWriteLabel($tablehead, $kolomhead++, "Is Aktive");
xlsWriteLabel($tablehead, $kolomhead++, "Create Date");

foreach ($this->Users_model->get_all() as $data) {
    $kolombody = 0;

    //ubah xlsWriteLabel menjadi xlsWriteNumber untuk kolom numeric
    xlsWriteNumber($tablebody, $kolombody++, $nourut);
    xlsWriteLabel($tablebody, $kolombody++, $data->username);
    xlsWriteLabel($tablebody, $kolombody++, $data->password);
    xlsWriteLabel($tablebody, $kolombody++, $data->nama);
    xlsWriteLabel($tablebody, $kolombody++, $data->email);
    xlsWriteLabel($tablebody, $kolombody++, $data->alamat);
    xlsWriteLabel($tablebody, $kolombody++, $data->kota);
    xlsWriteLabel($tablebody, $kolombody++, $data->provinsi);
    xlsWriteLabel($tablebody, $kolombody++, $data->telepon);
    xlsWriteLabel($tablebody, $kolombody++, $data->id_level);
    xlsWriteLabel($tablebody, $kolombody++, $data->is_aktive);
    xlsWriteLabel($tablebody, $kolombody++, $data->create_date);

    $tablebody++;
    $nourut++;
}

```

```

        xlsEOF();
        exit();
    }

    public function word()
    {
        header("Content-type: application/vnd.ms-word");
        header("Content-Disposition: attachment;Filename=users.doc");

        $data = array(
            'users_data' => $this->Users_model->get_all(),
            'start' => 0
        );

        $this->load->view('users/users_doc', $data);
    }
}

```

LOGIN.PHP

```

<?php
defined('BASEPATH') or exit('No direct script access allowed');
class Login extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        $this->load->model('M_login');
        $this->load->model('M_user');
    }
    public function index()
    {
        check_already_login();
        $this->load->view('login');
    }
    public function loginsubmit()
    {
        $username = $this->input->post('username');
        $user = $this->M_user->get($username);
        $this->form_validation->set_rules('username', 'Username', 'required',
array('required' => 'Silahkan Masukkan Username Anda'));
        $this->form_validation->set_rules('password', 'Password', 'required',
array('required' => 'Silahkan Masukkan Password Anda'));
        if ($this->form_validation->run() == false) {

```

```

        $this->load->view('login');
    } else {
        if ($this->M_login->checklogin($_POST['username'], $_POST['password'])
> 0) {
            $this->session->set_userdata('is_aktif', $user->is_aktif);
            if ($this->session->userdata('is_aktif') == 1) {
                $this->session->set_userdata('username', $_POST['username']);
                $this->session->set_userdata('password', $_POST['password']);
                $this->session->set_userdata('nama', $user->nama);
                $this->session->set_userdata('level', $user->id_level);
                $this->session->set_userdata('id_username', $user->id_username);
                redirect('home');
            } else {
                echo "<script>
alert('Silahkan hubungi admin untuk aktivasi akun');
window.location="" . site_url('Login') . ""</script>";
            }
        } else {
            echo "<script>
alert('Login Gagal, Username atau password salah');
window.location="" . site_url('Login') . ""</script>";
        }
    }
}

public function logout()
{
    $this->session->sess_destroy();
    redirect('login');
}

public function simpan_user()
{
    $this->form_validation->set_rules('username', 'Username',
'trim|is_unique[users.username]');
    $this->form_validation->set_rules('email', 'Email', 'is_unique[users.email]');
    $data = array(
        'nama' => $this->input->post('nama', true),
        'username' => $this->input->post('username', true),
        'password' => md5($this->input->post('password', true)),
        'email' => $this->input->post('email', true),
        'password_plain' => $this->input->post('password', true),
        'create_date' => date('y-m-d H:i:s'),
        'id_level' => 2,
        'is_aktif' => 2,
    );
    if ($this->form_validation->run() === FALSE) {
        echo "<script>

```

```

alert('Username/Email ini sudah terpakai, Silahkan ganti');
window.location="" . site_url('Login#signup') . ""</script>";
} else {
    $soke = $this->M_login->tambahdata($data);
    echo "<script>
alert('Username berhasil dibuat');
window.location="" . site_url('Login') . ""</script>";
}
}
public function lupa_password()
{
    $this->form_validation->set_rules('email', 'Email', 'required');
    if ($this->form_validation->run() == false) {
        $this->load->view('login');
    } else {
        $email = $this->input->post('email');
        $user = $this->db->get_where('users', ['email' => $email, 'is_active' => 1])-
        >row_array();
        if ($user) {
            $token = base64_encode(random_bytes(32));
            $user_token = [
                'email' => $this->input->post('email', true),
                'token' => $token,
                'create_date' => time()
            ];
            $soke = $this->M_user->user_token($user_token);
            $this->_send_email($token, 'forgot');
            echo "<script>
alert('Silahkan cek email untuk reset password');
window.location="" . site_url('login') . ""</script>";
        } else {
            echo "<script>
alert('Email tidak terdaftar atau user belum aktive');
window.location="" . site_url('login') . ""</script>";
        }
    }
}
private function _send_email($token, $type)
{
    $config = [
        'protocol' => 'smtp',
        'smtp_host' => 'ssl://smtp.googlemail.com',
        'smtp_user' => 'asyudianggara61120@gmail.com',
        'smtp_pass' => 'asyudi@21620',
        'smtp_port' => 465,
        'mailtype' => 'html',
    ];

```

```

        'charset' => 'utf-8',
        'newline' => "\r\n"
    ];
    $this->load->library('email', $config);
    $this->email->from('asyudianggara61120@gmail.com', 'Admin WEB
Aplication');
    $this->email->to($this->input->post('email'));
    if ($type == 'verify') {
        $this->email->subject('Aktivasi Akun');
        $this->email->message('Click this link to verify your account : <a href=' .
base_url() . 'login/verify?email=' . $this->input->post('email') . '&token=' .
urlencode($token) . '&username=' . $this->input->post('username') .
"">Activate</a>');
    } else if ($type == 'forgot') {
        $this->email->subject('Reset Password');
        $this->email->message('Click this link to reset your password : <a href=' .
base_url() . 'login/reset_password?email=' . $this->input->post('email') . '&token='
. urlencode($token) . "">Reset Password</a>');
    }
    if ($this->email->send()) {
        return true;
    } else {
        echo $this->email->print_debugger();
        die;
    }
}
public function reset_password()
{
    $email = $this->input->get('email');
    $token = $this->input->get('token');
    $user = $this->db->get_where('users', ['email' => $email])->row_array();
    if ($user) {
        $user_token = $this->db->get_where('user_token', ['token' => $token])->
row_array();
        if ($user_token) {
            if (time() - $user_token['create_date'] < (60 * 60 * 24)) {
                $this->session->set_userdata('reset_email', $email);
                $this->rubah_password();
            } else {
                $this->db->delete('user_token', ['email' => $email]);
                echo "<script>
alert('Reset password gagal, Token Kadaluarsa');
window.location=' . site_url('Login') . ""</script>";
            }
        } else {
            echo "<script>

```

```

alert('Reset Password gagal, Token salah');
window.location="" . site_url('Login') . ""</script>";
    }
    } else {
        echo "<script>
alert('Reset Password gagal, Email salah');
window.location="" . site_url('Login') . ""</script>";
    }
}
public function rubah_password()
{ //reset password
    if (!$this->session->userdata('reset_email')) {
        redirect('login');
    }
    $this->form_validation->set_rules('password', 'password', 'required');
    $this->form_validation->set_rules('passcon', 'passcon', 'required');
    if ($this->form_validation->run() == false) {
        $this->load->view('rubah_password/v_rubah_password');
    } else {
        $password = md5($this->input->post('password', true));
        $email = $this->session->userdata('reset_email');
        $this->db->set('password', $password);
        $this->db->where('email', $email);
        $this->db->update('users');
        $this->db->delete('user_token', ['email' => $email]);
        $this->session->unset_userdata('reset_email');
        echo "<script>
alert('Password berhasil di rubah, Silahkan Login');
window.location="" . site_url('Login') . ""</script>";
    }
}
public function ganti_password($id_username)
{
    $this->template->load('template', 'rubah_password/v_ganti_password');
}
public function submit_ganti_password($id_username)
{
    $passwordlama = $this->input->post('lama'); // poSt dari inputan password
lama
    if ($this->session->userdata('level') != 1) {
        if ($this->session->userdata('password') == $passwordlama) { // cek kondisi
jika password lama sama degn password yg di database
            $data = array(
                //yang kiri sesuai filed table sedangkan yang kanan sesuai dari form
inputan//
                'password' => md5($this->input->post('password', true)),

```

```

    );
    $this->M_user->ubah_data($data, $id_username); // variable content dan
variable nama array harus sama
    redirect('login/logout');
    } else { //jika password lama beda dengan yang ada di database akan di
arahkan langsung ke HOME
        echo "<script>
alert('Password Lama Salah');
window.location="" . site_url('home') . ""</script>";
    }
    } else {

        $data = array(
            'password' => md5($this->input->post('password', true)), //yang kiri
sesuai filed table sedangkan yang kanan sesuai dari form inputan//
        );
        $this->M_user->ubah_data($data, $id_username); // variable content dan
variable nama array harus sama
        redirect('login/logout');
    }
}
public function edit_profile($id_username)
{
    $oke['data_user'] = $this->M_user->get_by_id($id_username)->row_array();
    $this->template->load('template', 'rubah_password/v_edit_profile', $oke);
}
public function update_edit_profile($id_username)
{
    $data = array(
        'nama' => $this->input->post('nama', TRUE),
        'email' => $this->input->post('email', TRUE),
        'alamat' => $this->input->post('alamat', TRUE),
        'kota' => $this->input->post('kota', TRUE),
        'provinsi' => $this->input->post('provinsi', TRUE),
        'telepon' => $this->input->post('telepon', TRUE),
    );
    $oke = $this->M_user->ubah_data($data, $id_username);
    redirect('home');
}
}

```

LAPORAN.PHP

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
date_default_timezone_set('Asia/Jakarta');
class Laporan extends CI_Controller {
    function __construct()
    {
        parent::__construct();
    }
    public function index() {
        $this->template->load('template', 'laporan/v_laporan');
    }
    public function kas_masjid_full() {
        $this->load->view('laporan/kas_masjid_full');
    }
    public function kas_masjid_per() {
        $this->load->view('laporan/kas_masjid_per');
    }
}

```

REKAP KAS MESJID.PHP

```

<?php
if (!defined('BASEPATH'))
    exit('No direct script access allowed');
class Rekap_kas_masjid extends CI_Controller
{
    function __construct()
    {
        parent::__construct();
        $this->load->model('Rekap_kas_masjid_model');
        $this->load->library('form_validation');
    }

    public function index()
    {
        $q = urldecode($this->input->get('q', TRUE));
        $start = intval($this->uri->segment(3));
        if ($q <> "") {
            $config['base_url'] = base_url() . '.php/c_url/index.html?q=' . urlencode($q);
            $config['first_url'] = base_url() .
            'index.php/rekap_kas_masjid/index.html?q=' . urlencode($q);
        } else {
            $config['base_url'] = base_url() . 'index.php/rekap_kas_masjid/index/';
        }
    }
}

```

```

        $config['first_url'] = base_url() . 'index.php/rekap_kas_masjid/index/';
    }
    $config['per_page'] = 10;
    $config['page_query_string'] = FALSE;
    $config['total_rows'] = $this->Rekap_kas_masjid_model->total_rows($q);
    $rekap_kas_masjid = $this->Rekap_kas_masjid_model->get_limit_data($config['per_page'], $start, $q);
    $config['full_tag_open'] = '<ul class="pagination pagination-sm no-margin pull-right">';
    $config['full_tag_close'] = '</ul>';
    $this->load->library('pagination');
    $this->pagination->initialize($config);

    $data = array(
        'rekap_kas_masjid_data' => $rekap_kas_masjid,
        'q' => $q,
        'pagination' => $this->pagination->create_links(),
        'total_rows' => $config['total_rows'],
        'start' => $start,
    );
    $this->template->load('template','rekap_kas_masjid/kas_masjid_list', $data);
}
public function read($id)
{
    $row = $this->Rekap_kas_masjid_model->get_by_id($id);
    if ($row) {
        $data = array(
            'id_km' => $row->id_km,
            'tgl_km' => $row->tgl_km,
            'uraian_km' => $row->uraian_km,
            'masuk' => $row->masuk,
            'keluar' => $row->keluar,
            'jenis' => $row->jenis,
        );
        $this->template->load('template','rekap_kas_masjid/kas_masjid_read',
$data);
    } else {
        $this->session->set_flashdata('message', 'Record Not Found');
        redirect(site_url('rekap_kas_masjid'));
    }
}
public function create()
{
    $data = array(
        'button' => 'Create',
        'action' => site_url('rekap_kas_masjid/create_action'),
    );
}

```

```

'id_km' => set_value('id_km'),
'tgl_km' => set_value('tgl_km'),
'uraian_km' => set_value('uraian_km'),
'masuk' => set_value('masuk'),
'keluar' => set_value('keluar'),
'jenis' => set_value('jenis'),
);
$this->template->load('template','rekap_kas_masjid/kas_masjid_form', $data);
}
public function create_action()
{
    $this->_rules();
    if ($this->form_validation->run() == FALSE) {
        $this->create();
    } else {
        $data = array(
            'tgl_km' => $this->input->post('tgl_km',TRUE),
            'uraian_km' => $this->input->post('uraian_km',TRUE),
            'masuk' => $this->input->post('masuk',TRUE),
            'keluar' => $this->input->post('keluar',TRUE),
            'jenis' => $this->input->post('jenis',TRUE),
        );
        $this->Rekap_kas_masjid_model->insert($data);
        $this->session->set_flashdata('message', 'Create Record Success 2');
        redirect(site_url('rekap_kas_masjid'));
    }
}
public function update($id)
{
    $row = $this->Rekap_kas_masjid_model->get_by_id($id);
    if ($row) {
        $data = array(
            'button' => 'Update',
            'action' => site_url('rekap_kas_masjid/update_action'),
            'id_km' => set_value('id_km', $row->id_km),
            'tgl_km' => set_value('tgl_km', $row->tgl_km),
            'uraian_km' => set_value('uraian_km', $row->uraian_km),
            'masuk' => set_value('masuk', $row->masuk),
            'keluar' => set_value('keluar', $row->keluar),
            'jenis' => set_value('jenis', $row->jenis),
        );
        $this->template->load('template','rekap_kas_masjid/kas_masjid_form',
$data);
    } else {
        $this->session->set_flashdata('message', 'Record Not Found');
        redirect(site_url('rekap_kas_masjid'));
    }
}

```

```

    }
}
public function update_action()
{
    $this->_rules();
    if ($this->form_validation->run() == FALSE) {
        $this->update($this->input->post('id_km', TRUE));
    } else {
        $data = array(
            'tgl_km' => $this->input->post('tgl_km',TRUE),
            'uraian_km' => $this->input->post('uraian_km',TRUE),
            'masuk' => $this->input->post('masuk',TRUE),
            'keluar' => $this->input->post('keluar',TRUE),
            'jenis' => $this->input->post('jenis',TRUE),
        );
        $this->Rekap_kas_masjid_model->update($this->input->post('id_km',
TRUE), $data);
        $this->session->set_flashdata('message', 'Update Record Success');
        redirect(site_url('rekap_kas_masjid'));
    }
}
public function delete($id)
{
    $row = $this->Rekap_kas_masjid_model->get_by_id($id);
    if ($row) {
        $this->Rekap_kas_masjid_model->delete($id);
        $this->session->set_flashdata('message', 'Delete Record Success');
        redirect(site_url('rekap_kas_masjid'));
    } else {
        $this->session->set_flashdata('message', 'Record Not Found');
        redirect(site_url('rekap_kas_masjid'));
    }
}

public function _rules()
{
    $this->form_validation->set_rules('tgl_km', 'tgl km', 'trim|required');
    $this->form_validation->set_rules('uraian_km', 'uraian km', 'trim|required');
    $this->form_validation->set_rules('masuk', 'masuk', 'trim|required');
    $this->form_validation->set_rules('keluar', 'keluar', 'trim|required');
    $this->form_validation->set_rules('jenis', 'jenis', 'trim|required');
    $this->form_validation->set_rules('id_km', 'id_km', 'trim');
    $this->form_validation->set_error_delimiters('<span class="text-danger">',
'</span>');
}
public function excel()

```

```

{
    $this->load->helper('exportexcel');
    $namaFile = "kas_masjid.xls";
    $judul = "kas_masjid";
    $tablehead = 0;
    $tablebody = 1;
    $nourut = 1;
    //penulisan header
    header("Pragma: public");
    header("Expires: 0");
    header("Cache-Control: must-revalidate, post-check=0,pre-check=0");
    header("Content-Type: application/force-download");
    header("Content-Type: application/octet-stream");
    header("Content-Type: application/download");
    header("Content-Disposition: attachment;filename=" . $namaFile . "");
    header("Content-Transfer-Encoding: binary ");
    xlsBOF();
    $kolomhead = 0;
    xlsWriteLabel($tablehead, $kolomhead++, "No");
    xlsWriteLabel($tablehead, $kolomhead++, "Tgl Km");
    xlsWriteLabel($tablehead, $kolomhead++, "Uraian Km");
    xlsWriteLabel($tablehead, $kolomhead++, "Masuk");
    xlsWriteLabel($tablehead, $kolomhead++, "Keluar");
    xlsWriteLabel($tablehead, $kolomhead++, "Jenis");
    foreach ($this->Rekap_kas_masjid_model->get_all() as $data) {
        $kolombody = 0;
        //ubah xlsWriteLabel menjadi xlsWriteNumber untuk kolom numeric
        xlsWriteNumber($tablebody, $kolombody++, $nourut);
        xlsWriteLabel($tablebody, $kolombody++, $data->tgl_km);
        xlsWriteLabel($tablebody, $kolombody++, $data->uraian_km);
        xlsWriteNumber($tablebody, $kolombody++, $data->masuk);
        xlsWriteNumber($tablebody, $kolombody++, $data->keluar);
        xlsWriteLabel($tablebody, $kolombody++, $data->jenis);

        $tablebody++;
        $nourut++;
    }
    xlsEOF();
    exit();
}
public function word()
{
    header("Content-type: application/vnd.ms-word");
    header("Content-Disposition: attachment;Filename=kas_masjid.doc");
    $data = array(
        'kas_masjid_data' => $this->Rekap_kas_masjid_model->get_all(),

```

```

        'start' => 0
    );
    $this->load->view('rekap_kas_masjid/kas_masjid_doc',$data);
}
}

```

USERS MODEL.PHP

```

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

class Users_model extends CI_Model
{
    public $table = 'users';
    public $id = 'id_username';
    public $order = 'DESC';

    function __construct()
    {
        parent::__construct();
    }

    // get all
    function get_all()
    {
        $this->db->order_by($this->id, $this->order);
        return $this->db->get($this->table)->result();
    }

    // get data by id
    function get_by_id($id)
    {
        $this->db->where($this->id, $id);
        return $this->db->get($this->table)->row();
    }

    // get total rows
    function total_rows($q = NULL)
    {
        $this->db->like('id_username', $q);
        $this->db->or_like('username', $q);
        $this->db->or_like('password', $q);
        $this->db->or_like('password_plain', $q);
        $this->db->or_like('nama', $q);
        $this->db->or_like('email', $q);
    }
}

```

```

    $this->db->or_like('alamat', $q);
    $this->db->or_like('kota', $q);
    $this->db->or_like('provinsi', $q);
    $this->db->or_like('telepon', $q);
    $this->db->or_like('id_level', $q);
    $this->db->or_like('is_aktif', $q);
    $this->db->or_like('create_date', $q);
    $this->db->from($this->table);
    return $this->db->count_all_results();
}

```

// get data with limit and search

```
function get_limit_data($limit, $start = 0, $q = NULL)
```

```

{
    $this->db->order_by($this->id, $this->order);
    $this->db->like('id_username', $q);
    $this->db->or_like('username', $q);
    $this->db->or_like('password', $q);
    $this->db->or_like('password_plain', $q);
    $this->db->or_like('nama', $q);
    $this->db->or_like('email', $q);
    $this->db->or_like('alamat', $q);
    $this->db->or_like('kota', $q);
    $this->db->or_like('provinsi', $q);
    $this->db->or_like('telepon', $q);
    $this->db->or_like('id_level', $q);
    $this->db->or_like('is_aktif', $q);
    $this->db->or_like('create_date', $q);
    $this->db->limit($limit, $start);
    return $this->db->get($this->table)->result();
}
// insert data
function insert($data)
{
    $this->db->insert($this->table, $data);
}

// update data
function update($id, $data)
{
    $this->db->where($this->id, $id);
    $this->db->update($this->table, $data);
}

```

```
// delete data
function delete($id)
{
    $this->db->where($this->id, $id);
    $this->db->delete($this->table);
}
public function edit_data($id)
{
    $this->db->from('users');
    $this->db->join('user_level', 'user_level.id_level = users.id_level');
    $this->db->where('id_username', $id);
    return $this->db->get()->row_array();
}
}
```