

LISTING PROGRAM

CI_Controllers

Home.php

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

class Home extends CI_Controller
{
    function __construct()
    {
        parent::__construct();
        //load library
        $this->load->library(['template',
'form_validation']);
        //load Model
        $this->load->model('app');

        header('Last-Modified:' .
gmdate('D, d M Y H:i:s') . 'GMT');
        header('Cache-Control: no-
cache, must-revalidate, max-age=0');
        header('Cache-Control: post-
check=0, pre-check=0, false);
        header('Pragma: no-cache');
    }

    public function index()
    {
        //cek apakah user sudah
melakukan login atau belum
        //jika sudah, maka redirect ke
halaman dashboard
        if ($this->session-
>userdata('level')) {
            redirect('dashboard');
        }

        //ketika button di klik, jalankan
proses login
        if ($this->input->post('submit',
TRUE) == 'submit') {
            //validasi data
            $this->form_validation-
>set_rules(
                'username',
                'Username',
```

```
                'required|min_length[3]',
                array(
                    'required' => 'Field
{field} wajib diisi',
                    'min_length' => '{field}
minimal 3 karakter'
                )
            );

            $this->form_validation-
>set_rules(
                'password',
                'Password',
                'required|min_length[3]',
                array(
                    'required' => 'Field
{field} wajib diisi',
                    'min_length' => '{field}
minimal 3 karakter'
                )
            );

            //jika validasi berhasil maka
lanjutkan ke proses pengecekan data
            if ($this->form_validation-
>run() == TRUE) {

                $username = $this-
>security->xss_clean($this->input-
>post('username', TRUE));
                $password = $this-
>security->xss_clean($this->input-
>post('password', TRUE));

                //ambil data user sesuai
username
                $get = $this->app-
>getData('tbl_user', ['username' =>
$username]);
                //validasi jumlah data
                if ($get->num_rows() != 1)
                {
                    $this->session-
>set_flashdata('alert', 'User tidak
dikenali...');
                } else {
```

```

        //fetch data
        $u = $get->row();
        //validasi password
        if
        (password_verify($password, $u-
        >password)) {
            //jika benar, maka
            simpan data ke dalam session
            //kemudian redirect ke
            halaman dashboard
            $this->session-
            >set_userdata(
                array(
                    'UserID' => $u-
            >id_user,
                    'User' => $u-
            >fullname,
                    'level' => $u-
            >level,
                    'foto' => $u-
            >foto
                )
            );
            redirect('dashboard');
        } else {
            //jika salah maka
            tampilkan pesan error
            $this->session-
            >set_flashdata('alert', 'Maaf,
            Password yang anda masukkan
            salah..');
        }
    }
}

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

public function dashboard()
{
    //cek login
    $this->is_login();

    $data = [

```

```

        'title' => 'Aplikasi Stok
        Barang'
    ];

    $this->template->kasir('home',
    $data);
}

public function profil_admin()
{
    if ($this->session-
    >userdata('level') != 'admin' || !$this-
    >session->userdata('level')) {
        redirect('dashboard');
    }

    //ambil data admin
    $getData = $this->app-
    >getData('tbl_user', ['id_user' =>
    $this->session->userdata('UserID'),
    'level' => 'admin']);

    if ($getData->num_rows() != 1)
    {
        redirect('dashboard');
    }

    //ketika button di klik
    if ($this->input->post('submit',
    TRUE) == 'submit') {

        $f = $getData->row();

        if ($this->security-
        >xss_clean($this->input-
        >post('username', TRUE)) == $f-
        >username) {

            $rules_username =
            'required|min_length[5]|max_length[
            20]|regex_match[/^[A-Za-z0-9]+$/]';
        } else {

            $rules_username =
            'required|min_length[5]|max_length[

```

```

20]|is_unique[tbl_user.username]|reg
ex_match[/^[A-Za-z0-9]+$]/';
    }

    $this->form_validation-
>set_rules(
    'username',
    'Username',
    $rules_username,
    array(
        'required' => '{field}
wajib diisi',
        'min_length' => '{field}
minimal 5 karakter',
        'max_length' => '{field}
maksimal 20 karakter',
        'is_unique' =>
'Username sudah digunakan,
silahkan pilih yang lain',
        'regex_match' =>
'{field} hanya boleh huruf dan angka'
    )
);

    $this->form_validation-
>set_rules(
    'fullname',
    'Fullname',
    "required|min_length[3]|m
ax_length[30]|regex_match[/^[A-Z
a-z.']+$/]",
    array(
        'required' => '{field}
wajib diisi',
        'min_length' => '{field}
minimal 3 karakter',
        'max_length' => '{field}
maksimal 30 karakter',
        'regex_match' =>
'{field} hanya boleh huruf, titik dan
kutip satu (\')'
    )
);

    $this->form_validation-
>set_rules(

```

```

    'password',
    'Password',
    "required|min_length[3]",
    array(
        'required' => '{field}
wajib diisi',
        'min_length' => '{field}
minimal 3 karakter'
    )
);

    //jika proses validasi berhasil,
    maka lanjutkan ke proses update data
    if ($this->form_validation-
>run() == TRUE) {
        //validasi password apakah
        benar atau salah
        if (!password_verify($this-
>input->post('password', TRUE), $f-
>password)) {
            $this->session-
>set_flashdata('error', 'Password
yang anda masukkan salah..');
            redirect('admin');
        }
        //simpan data foto lama
        $foto = $f->foto;
        //konfigurasi upload
        $config['upload_path']
= './assets/foto/';
        $config['allowed_types']
= 'jpg|png|jpeg';
        $config['max_size'] =
'2048';
        $config['file_name'] =
'foto' . time();

        $this->load-
>library('upload', $config);

        if ($this->upload-
>do_upload('foto')) {

            $b = $this->upload-
>data();
            $foto = $b['file_name'];

```

```

        //buat foto thumbnail
        untuk icon user
        $this->load-
        >library('image_lib');

        $config2['image_library'
    ] = 'gd2';
        $config2['source_image'
    ] = './assets/foto/' . $foto;
        $config2['new_image']
        = './assets/foto/thumb/' . $foto;
        $config2['maintain_ratio
    '] = FALSE;
        $imgSize2 = $this-
        >image_lib-
        >get_image_properties($config2['sou
        rce_image'], TRUE);

        if ($imgSize2['width'] >
        $imgSize2['height']) {
            $config2['width'] =
        $imgSize2['height'];
            $config2['height'] =
        $imgSize2['height'];
            $config2['x_axis'] =
        (($imgSize2['width'] / 2) -
        ($imgSize2['width'] / 2));
        } else {
            $config2['width'] =
        $imgSize2['width'];
            $config2['height'] =
        $imgSize2['width'];
            $config2['y_axis'] =
        (($imgSize2['height'] / 2) -
        ($imgSize2['height'] / 2));
        }

        $this->image_lib-
        >initialize($config2);

        if (!$this->image_lib-
        >crop()) {
            echo $this-
        >image_lib->display_errors();

```

```

        exit;
    }
    //hapus foto lama, jika
    foto bukan foto default
    if ($f->foto !=
    'default.jpg' && $f->foto != "") {
        //hapus foto utama
        unlink('./assets/foto/' .
        $f->foto);
        //hapus foto profil
        unlink('./assets/foto/th
        umb/' . $f->foto);
    }
}

//masukkan data ke
variable array
$update = [
    'username' => $this-
    >security->xss_clean($this->input-
    >post('username', TRUE)),
    'fullname' => $this-
    >security->xss_clean($this->input-
    >post('fullname', TRUE)),
    'foto' => $foto
];

$where = [
    'id_user' => $this-
    >session->userdata('UserID')
];
//proses update
$update = $this->app-
>update('tbl_user', $update, $where);
//jika proses update
berhasil, maka alihkan ke halaman
data pegawai dan tampilkan pesan
if ($update) {
    $this->session-
    >set_flashdata('success', 'Profil
    berhasil diperbarui..');
    //update session profil
    $this->session-
    >set_userdata(
        array(

```

```

        'User' => $this->security->xss_clean($this->input->post('fullname', TRUE)),
        'foto' => $foto
    )
);

    redirect('admin');
}
}

$data = [
    'title' => 'Profil ' . $this->session->userdata('User'),
    'data' => $getData->row()
];

$this->template->kasir('profil_admin', $data);
}

public function profil_pegawai()
{
    if ($this->session->userdata('level') != 'pegawai' || !$this->session->userdata('level')) {
        redirect('dashboard');
    }

    //ambil data pegawai
    $getData = $this->app->getData('tbl_user', ['id_user' => $this->session->userdata('UserID'), 'level' => 'pegawai', 'active' => 'Y']);

    if ($getData->num_rows() != 1)
    {
        redirect('dashboard');
    }

    //ketika button di klik
    if ($this->security->xss_clean($this->input->post('submit', TRUE)) == 'submit')
    {

```

```

        //fetch data lama pegawai
        $p = $getData->row();
        //cek apakah admin merubah
        username nya atau tidak
        if ($this->security->xss_clean($this->input->post('username', TRUE)) == $p->username) {
            $rules_username =
            'required|min_length[5]|max_length[20]|regex_match[/^[A-Za-z0-9]+$]/';
        } else {
            $rules_username =
            'required|min_length[5]|max_length[20]|is_unique[tbl_user.username]|regex_match[/^[A-Za-z0-9]+$]/';
        }

        $this->form_validation->set_rules(
            'username',
            'Username',
            $rules_username,
            array(
                'required' => '{field} wajib diisi',
                'min_length' => '{field} minimal 5 karakter',
                'max_length' => '{field} maksimal 20 karakter',
                'is_unique' => 'Username sudah digunakan, silahkan pilih yang lain',
                'regex_match' => '{field} hanya boleh huruf dan angka'
            )
        );

        $this->form_validation->set_rules(
            'fullname',
            'Fullname',
            "required|min_length[3]|max_length[30]|regex_match[/^[A-Za-z.']+$/]",
            array(

```

```

                'required' => '{field}'
wajib diisi',
                'min_length' => '{field}'
minimal 3 karakter',
                'max_length' => '{field}'
maksimal 30 karakter',
                'regex_match' =>
'{field} hanya boleh huruf, titik dan
kutip satu (\')'
            )
        );

```

```

//cek apakah form nomor hp
diisi atau tidak, jika diisi maka
lakukan validasi
if ($this->input->post('hp',
TRUE) != "") {
    $this->form_validation-
>set_rules(
        'hp',
        'Nomor HP',
        "required|min_length[8]|
max_length[15]|regex_match[/^[0-
9]+$]/",
        array(
            'required' => '{field}'
wajib diisi',
            'min_length' =>
'{field} minimal 8 karakter',
            'max_length' =>
'{field} maksimal 15 karakter',
            'regex_match' =>
'{field} hanya boleh angka'
        )
    );
}

```

```

    $this->form_validation-
>set_rules(
        'alamat',
        'Alamat',
        "required|min_length[10]|
max_length[255]|regex_match[/^[A-
Z a-z.0-9-,]+$]/",
        array(

```

```

                'required' => '{field}'
wajib diisi',
                'min_length' => '{field}'
minimal 5 karakter',
                'max_length' => '{field}'
maksimal 30 karakter',
                'regex_match' => 'Data
{field} yang anda masukkan tidak
valid'
            )
        );

```

```

    $this->form_validation-
>set_rules(
        'password',
        'Password',
        "required|min_length[3]",
        array(
            'required' => '{field}'
wajib diisi',
            'min_length' => '{field}'
minimal 3 karakter'
        )
    );

```

```

//jika proses validasi berhasil,
maka lanjutkan ke proses update data
if ($this->form_validation-
>run() == TRUE) {
    //validasi password apakah
benar atau salah
    if (!password_verify($this-
>input->post('password', TRUE), $p-
>password)) {
        $this->session-
>set_flashdata('error', 'Password
yang anda masukkan salah..');
        redirect('profil');
    }
    //simpan data foto lama
    $foto = $p->foto;
    //konfigurasi upload
    $config['upload_path']
= './assets/foto/';
    $config['allowed_types']
= 'jpg|png|jpeg';
}

```

```

        $config['max_size']      =
'2048';
        $config['file_name']    =
'foto' . time();

        $this->load-
>library('upload', $config);

        if ($this->upload-
>do_upload('foto')) {

            $b = $this->upload-
>data();
            $foto = $b['file_name'];

            //buat foto thumbnail
            untuk icon user
            $this->load-
>library('image_lib');

            $config2['image_library']
] = 'gd2';
            $config2['source_image']
] = './assets/foto/' . $foto;
            $config2['new_image']
= './assets/foto/thumb/' . $foto;
            $config2['maintain_ratio
'] = FALSE;
            $imgSize2 = $this-
>image_lib-
>get_image_properties($config2['sou
rce_image'], TRUE);

            if ($imgSize2['width'] >
$imgSize2['height']) {
                $config2['width'] =
$imgSize2['height'];
                $config2['height'] =
$imgSize2['height'];
                $config2['x_axis'] =
((($imgSize2['width'] / 2) -
($imgSize2['width'] / 2));
            } else {
                $config2['width'] =
$imgSize2['width'];

```

```

        $config2['height'] =
$imgSize2['width'];
        $config2['y_axis'] =
((($imgSize2['height'] / 2) -
($imgSize2['height'] / 2));
    }

    $this->image_lib-
>initialize($config2);

    if (!$this->image_lib-
>crop()) {
        echo $this-
>image_lib->display_errors();

        exit;
    }
    //hapus foto lama, jika
foto bukan foto default
    if ($p->foto !=
'default.jpg' && $p->foto != ") {
        //hapus foto utama
        unlink('./assets/foto/' .
$p->foto);
        //hapus foto profil
        unlink('./assets/foto/th
umb/' . $p->foto);
    }
}

    //masukkan data ke
variable array
    $update = [
        'username' => $this-
>security->xss_clean($this->input-
>post('username', TRUE)),
        'fullname' => $this-
>security->xss_clean($this->input-
>post('fullname', TRUE)),
        'hp' => $this->security-
>xss_clean($this->input->post('hp',
TRUE)),
        'alamat' => $this-
>security->xss_clean($this->input-
>post('alamat', TRUE)),
        'foto' => $foto

```

```

];

$where = [
    'id_user' => $this-
>security->xss_clean($p->id_user)
];
//proses update
$up = $this->app-
>update('tbl_user', $update, $where);
//jika proses update
berhasil, maka alihkan ke halaman
data pegawai dan tampilkan pesan
if ($up) {
    $this->session-
>set_flashdata('success', 'Data
pegawai berhasil diperbarui..');
    //update session profil
    $this->session-
>set_userdata(
        array(
            'User' => $this-
>security->xss_clean($this->input-
>post('fullname', TRUE)),
            'foto' => $foto
        )
    );

    redirect('profil');
}
}

}

$data = [
    'title' => 'Profil ' . $this-
>session->userdata('User'),
    'data' => $getData->row()
];

$this->template-
>kasir('profil_pegawai', $data);
}

public function ganti_password()
{
    $this->is_login();

```

```

//ketika user mengklik tombol
ubah password
if ($this->input->post('submit',
TRUE) == 'submit') {
    //lakukan validasi
    $this->form_validation-
>set_rules(
        'pwBaru',
        'Password Baru',
        "required|min_length[3]",
        array(
            'required' => '{field}
wajib diisi',
            'min_length' => '{field}
minimal 3 karakter'
        )
    );

    $this->form_validation-
>set_rules(
        'pwLama',
        'Password Lama',
        "required|min_length[3]",
        array(
            'required' => '{field}
wajib diisi',
            'min_length' => '{field}
minimal 3 karakter'
        )
    );

    if ($this->form_validation-
>run() == TRUE) {
        //ambil data user
        $getData = $this->app-
>getData('tbl_user', ['id_user' =>
$this->session->userdata('UserID')]);
        //fetch data
        $f = $getData->row();
        //tampung data dari form
        $pwBaru = $this-
>security->xss_clean($this->input-
>post('pwBaru', TRUE));
        $pwLama = $this-
>security->xss_clean($this->input-
>post('pwLama', TRUE));
    }
}

```

```

        //validasi password lama
        dengan password yang ada didalam
        database
        if
        (!password_verify($pwLama, $f-
        >password)) {
            $this->session-
            >set_flashdata('error', 'Password
            Lama yang anda masukkan salah..');
            redirect('password');
        }
        //cek apakah password
        lama dan baru sama, jika sama maka
        munculkan pesan
        if ($pwBaru == $pwLama)
        {
            $this->session-
            >set_flashdata('error', 'Password
            Baru dan Password Lama sama, Data
            tidak diubah...');
            redirect('password');
        }
        //encrypt & update
        password
        $hash_password =
        password_hash($pwBaru,
        PASSWORD_DEFAULT, ['cost' =>
        8]);

        $update = $this->app-
        >update('tbl_user', ['password' =>
        $hash_password], ['id_user' => $f-
        >id_user]);

        if ($update) {
            $this->session-
            >set_flashdata('success', 'Password
            berhasil diubah..');
        } else {
            $this->session-
            >set_flashdata('error', 'Password
            gagal diubah..');
        }

        redirect('password');
    }

```

```

    }

    $data = [
        'title' => 'Ganti Password'
    ];

    $this->template-
    >kasir('ganti_password', $data);
    }

    public function sign_out()
    {
        //cek login
        $this->is_login();
        //update last sign in
        $user = $this->session-
        >userdata('UserID');
        $last_login = date('Y-m-d
        H:i:s');

        $this->app->update('tbl_user',
        ['last_login' => $last_login],
        ['id_user' => $user]);
        //hapus session
        $this->session->sess_destroy();
        //redirect halaman login
        redirect(site_url());
    }

    private function is_login()
    {
        if (!$this->session-
        >userdata('level')) {
            redirect(site_url());
        }
    }
}

```

Data_barang.php

```

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

class Data_barang extends
CI_Controller
{

```

```

function __construct()
{
    parent::__construct();
    //load library
    $this->load->library(['template',
'form_validation']);
    //load model
    $this->load-
>model('m_barang');

    header('Last-Modified:' .
gmdate('D, d M Y H:i:s') . 'GMT');
    header('Cache-Control: no-
cache, must-revalidate, max-age=0');
    header('Cache-Control: post-
check=0, pre-check=0', false);
    header('Pragma: no-cache');
}

public function index()
{
    //cek apakah user yang login
adalah admin atau bukan
    //jika bukan maka alihkan ke
dashboard
    $this->is_admin();

    $data = [
        'title' => 'Data Barang'
    ];

    $this->template-
>kasir('data_barang/index', $data);
}

public function tambah_data()
{
    //cek apakah user yang login
adalah admin atau bukan
    //jika bukan maka alihkan ke
dashboard
    $this->is_admin();

    if ($this->input->post('submit',
TRUE) == 'submit') {
        //set rules form validasi

```

```

        $this->form_validation-
>set_rules(
            'kode',
            'Kode Barang',
            'required|min_length[3]|ma
x_length[7]|is_unique[tbl_barang.ko
de_barang]',
            array(
                'min_length' => '{field}
minimal 3 karakter',
                'max_length' => '{field}
maksimal 7 karakter',
                'is_unique' => 'Kode
sudah terdaftar'
            )
        );

        $this->form_validation-
>set_rules(
            'nama_barang',
            'Nama Barang',
            'required|min_length[3]|ma
x_length[255]',
            array(
                'required' => '{field}
wajib diisi',
                'min_length' => '{field}
minimal 3 karakter',
                'max_length' => '{field}
maksimal 255 karakter'
            )
        );

        $this->form_validation-
>set_rules(
            'brand',
            'Nama Brand',
            'required|min_length[2]',
            array(
                'required' => '{field}
wajib diisi',
                'min_length' => '{field}
minimal 2 karakter'
            )
        );

```

```

        $this->form_validation-
>set_rules(
        'harga',
        'Harga Jual',
        "required|regex_match[/^[0
-9.]+$/]",
        array(
            'required' => '{field}
wajib diisi',
            'regex_match' =>
            '{field} hanya boleh angka'
        )
    );

```

```

        //jika data sudah valid maka
        lakukan proses penyimpanan
        if ($this->form_validation-
>run() == TRUE) {
            //masukkan data ke
            variable array
            $simpan = array(
                'kode_barang' => $this-
>security->xss_clean($this->input-
>post('kode', TRUE)),
                'nama_barang' => $this-
>security->xss_clean($this->input-
>post('nama_barang', TRUE)),
                'brand' => $this-
>security->xss_clean($this->input-
>post('brand', TRUE)),
                'harga' => str_replace('.',
            ", $this->security->xss_clean($this-
>input->post('harga', TRUE)))
            );

```

```

        //simpan ke database
        $save = $this->m_barang-
>save('tbl_barang', $simpan);

        if ($save) {
            $this->session-
>set_flashdata('success', 'Data
Barang berhasil ditambah...');

            redirect('barang');
        }

```

```

        }
    }

    $data = [
        'title' => 'Tambah Data
Barang'
    ];

    $this->template-
>kasir('data_barang/form_tambah',
    $data);
}

    public function
    edit_data($kode_barang = "")
    {

        //cek apakah user yang login
        adalah admin atau bukan
        //jika bukan maka alihkan ke
        dashboard
        $this->is_admin();

        //cek uri
        if ($kode_barang == "") {
            $this->session-
>set_flashdata('error', 'Data tidak
valid...');

            redirect('barang');
        }

        //ambil data barang
        $barang = $this->m_barang-
>getData('tbl_barang', ['kode_barang'
=> $kode_barang]);

        //validasi jumlah data
        if ($barang->num_rows() != 1)
        {
            $this->session-
>set_flashdata('error', 'Data tidak
valid...');

            redirect('barang');
        }
    }

```

```

        //ketika button diklik
        if ($this->input->post('update',
TRUE) == 'Update') {
            //cek apakah user merubah
kode barang atau tidak
            $b = $barang->row();
            if ($b->kode_barang ==
$this->security->xss_clean($this-
>input->post('ID', TRUE))) {
                $rules_kode_barang =
'required|min_length[3]|max_length[
6]';
            } else {
                $rules_kode_barang =
'required|min_length[3]|max_length[
6]|is_unique[tbl_barang.kode_barang
]';
            }
            //set rules form validasi
            $this->form_validation-
>set_rules(
                'ID',
                'Kode Barang',
                $rules_kode_barang,
                array(
                    'required' => '{field}
wajib diisi',
                    'min_length' => '{field}
minimal 3 karakter',
                    'max_length' => '{field}
maksimal 6 karakter',
                    'is_unique' => 'Kode
sudah terdaftar'
                )
            );

            $this->form_validation-
>set_rules(
                'nama_barang',
                'Nama Barang',
                'required|min_length[3]|ma
x_length[255]',
                array(
                    'required' => '{field}
wajib diisi',

```

```

                    'min_length' => '{field}
minimal 3 karakter',
                    'max_length' => '{field}
maksimal 255 karakter'
                )
            );

            $this->form_validation-
>set_rules(
                'brand',
                'Nama Brand',
                'required|min_length[2]',
                array(
                    'required' => '{field}
wajib diisi',
                    'min_length' => '{field}
minimal 2 karakter'
                )
            );

            $this->form_validation-
>set_rules(
                'harga',
                'Harga Jual',
                "required|regex_match[/^[0
-9.]+$/]",
                array(
                    'required' => '{field}
wajib diisi',
                    'regex_match' =>
'{field} hanya boleh angka'
                )
            );

            $this->form_validation-
>set_rules(
                'status',
                'Status',
                "required|min_length[1]|m
ax_length[1]|regex_match[/^[YN]+$/
]",
                array(
                    'required' => '{field}
wajib diisi',
                    'min_length' => '{field}
hanya boleh 1 karakter',

```

```

                'max_length' => '{field}'
                hanya boleh 1 karakter',
                'regex_match' => 'Input
                {field} tidak valid'
            )
        );

        //jika validasi berhasil
        if ($this->form_validation-
        >run() == TRUE) {
            //masukkan data ke
            variable array
            $update = array(
                'nama_barang' => $this-
                >security->xss_clean($this->input-
                >post('nama_barang', TRUE)),
                'brand' => $this-
                >security->xss_clean($this->input-
                >post('brand', TRUE)),
                'harga' => str_replace('.',
                ", $this->security->xss_clean($this-
                >input->post('harga', TRUE))),
                'active' => $this-
                >security->xss_clean($this->input-
                >post('status', TRUE))
            );

            //simpan ke database
            $sup = $this->m_barang-
            >update('tbl_barang', $update,
            ['kode_barang' => $this->security-
            >xss_clean($this->input->post('ID',
            TRUE))]);

            if ($sup) {
                $this->session-
                >set_flashdata('success', 'Data
                Barang berhasil diperbarui...');

                redirect('barang');
            }
        }

        $data = [
            'title' => 'Edit Data Barang',

```

```

            'barang' => $barang->row()
        ];

        $this->template-
        >kasir('data_barang/form_edit',
        $data);
    }

    public function stok()
    {
        //cek pegawai
        if (!$this->session-
        >userdata('level') || $this->session-
        >userdata('level') != 'pegawai') {
            redirect('dashboard');
        }

        $data = [
            'title' => 'Data Stok Barang'
        ];

        $this->template-
        >kasir('data_barang/stok', $data);
    }

    public function ajax_barang()
    {
        $this->is_admin();
        //cek apakah request berupa ajax
        atau bukan, jika bukan maka redirect
        ke home
        if ($this->input-
        >is_ajax_request()) {
            //ambil list data
            $list = $this->m_barang-
            >get_datatables();
            //siapkan variabel array
            $data = array();
            $no = $_POST['start'];

            foreach ($list as $i) {

                $no++;
                $row = array();
                $row[] = $no;
                $row[] = $i->kode_barang;

```

```

        $row[] = $i-
>nama_barang;
        $row[] = $i->brand;
        $row[] = $i->stok;
        $row[] = '<span
class="float-left">Rp.</span><span
class="float-right">' .
number_format($i->harga, 0, ',', '.') .
',-</span>';
        $row[] = ($i->active ==
'Y') ? 'Aktif' : 'Tidak Aktif';
        $row[] = '<a href="' .
site_url('edit_barang/' . $i-
>kode_barang) . '" class="btn btn-
warning btn-sm text-
white">Edit</a>';

        $data[] = $row;
    }

    $output = array(
        "draw" =>
$_POST['draw'],
        "recordsTotal" => $this-
>m_barang->count_all(),
        "recordsFiltered" => $this-
>m_barang->count_filtered(),
        "data" => $data
    );
    //output to json format
    echo json_encode($output);
} else {
    redirect('dashboard');
}
}

public function ajax_stok_barang()
{
    //cek pegawai
    if (!$this->session-
>userdata('level') || $this->session-
>userdata('level') != 'pegawai') {
        redirect('dashboard');
    }
}

```

```

        //cek apakah request berupa ajax
atau bukan, jika bukan maka redirect
ke home
        if ($this->input-
>is_ajax_request()) {
            //ambil list data
            $list = $this->m_barang-
>get_datatables();
            //siapkan variabel array
            $data = array();
            $no = $_POST['start'];

            foreach ($list as $i) {

                $no++;
                $row = array();
                $row[] = $no;
                $row[] = $i->kode_barang;
                $row[] = $i-
>nama_barang;
                $row[] = $i->brand;
                $row[] = $i->stok;
                $row[] = '<span
class="float-left">Rp.</span><span
class="float-right">' .
number_format($i->harga, 0, ',', '.') .
',-</span>';

                $data[] = $row;
            }

            $output = array(
                "draw" =>
$_POST['draw'],
                "recordsTotal" => $this-
>m_barang->count_all(),
                "recordsFiltered" => $this-
>m_barang->count_filtered(),
                "data" => $data
            );
            //output to json format
            echo json_encode($output);
        } else {
            redirect('dashboard');
        }
    }
}

```

```

private function is_admin()
{
    if (!$this->session-
>userdata('level') || $this->session-
>userdata('level') != 'admin') {
        redirect('dashboard');
    }
}
}

```

Data_pegawai.php

```

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

class Data_pegawai extends
CI_Controller
{
    function __construct()
    {
        parent::__construct();
        //load library
        $this->load->library(['template',
'form_validation', 'upload']);
        //load model
        $this->load-
>model('m_pegawai');

        header('Last-Modified:' .
gmdate('D, d M Y H:i:s') . 'GMT');
        header('Cache-Control: no-
cache, must-revalidate, max-age=0');
        header('Cache-Control: post-
check=0, pre-check=0', false);
        header('Pragma: no-cache');
    }

    public function index()
    {
        //cek apakah user seorang admin
atau bukan
        //jika bukan admin maka
alihkan ke halaman dashboard
        $this->is_admin();
    }
}

```

```

$data = [
    'title' => 'Data Pegawai'
];

$this->template-
>kasir('pegawai/index', $data);
}

public function tambah_pegawai()
{
    //cek apakah user seorang admin
atau bukan
    //jika bukan admin maka
alihkan ke halaman dashboard
    $this->is_admin();

    //ketika button tambah data di
klik, lakukan proses validasi data
    if ($this->input->post('submit',
TRUE) == 'submit') {
        //validasi input data
        $this->form_validation-
>set_rules(
            'username',
            'Username',
            "required|min_length[5]|m
ax_length[20]|is_unique[tbl_user.use
rname]|regex_match[/^[A-Za-z0-
9]+$/]";
            array(
                'required' => '{field}
wajib diisi',
                'min_length' => '{field}
minimal 5 karakter',
                'max_length' => '{field}
maksimal 20 karakter',
                'is_unique' =>
'Username sudah digunakan,
silahkan pilih yang lain',
                'regex_match' =>
'{field} hanya boleh huruf dan angka'
            )
        );

        $this->form_validation-
>set_rules(

```

```

        'fullname',
        'Fullname',
        "required|min_length[3]|m
ax_length[30]|regex_match[/^[A-Z
a-z.']+$/]",
        array(
            'required' => '{field}
wajib diisi',
            'min_length' => '{field}
minimal 3 karakter',
            'max_length' => '{field}
maksimal 30 karakter',
            'regex_match' =>
            '{field} hanya boleh huruf, titik dan
kutip satu (\')'
        )
    );

```

```

    $this->form_validation-
>set_rules(
        'password',
        'Password',
        "required|min_length[3]|m
ax_length[30]",
        array(
            'required' => '{field}
wajib diisi',
            'min_length' => '{field}
minimal 3 karakter',
            'max_length' => '{field}
maksimal 30 karakter'
        )
    );

```

```

        //cek apakah form nomor hp
        diisi atau tidak, jika diisi maka
        lakukan validasi
        if ($this->input->post('hp',
        TRUE) != "") {
            $this->form_validation-
>set_rules(
                'hp',
                'Nomor HP',
                "required|min_length[8]|
max_length[15]|regex_match[/^[0-
9]+$/]",

```

```

        array(
            'required' => '{field}
wajib diisi',
            'min_length' =>
            '{field} minimal 8 karakter',
            'max_length' =>
            '{field} maksimal 15 karakter',
            'regex_match' =>
            '{field} hanya boleh angka'
        )
    );
}

```

```

    $this->form_validation-
>set_rules(
        'alamat',
        'Alamat',
        "required|min_length[10]|
max_length[255]|regex_match[/^[A-
Z a-z.0-9-,']+$/]",
        array(
            'required' => '{field}
wajib diisi',
            'min_length' => '{field}
minimal 5 karakter',
            'max_length' => '{field}
maksimal 30 karakter',
            'regex_match' => 'Data
{field} yang anda masukkan tidak
valid'
        )
    );

```

```

        //jika data sudah valid, maka
        jalankan proses penyimpanan data
        if ($this->form_validation-
>run() == TRUE) {

```

```

            //konfigurasi upload
            $config['upload_path']
            = './assets/foto/';
            $config['allowed_types']
            = 'jpg|png|jpeg';
            $config['max_size']      =
            '2048';

```

```

        $config['file_name']    =
'foto' . time();

        $this->upload-
>initialize($config);

        if ($this->upload-
>do_upload('foto')) {

            $b = $this->upload-
>data();
            $foto = $b['file_name'];

            //buat foto thumbnail
            untuk icon user
            $this->load-
>library('image_lib');

            $config2['image_library'
] = 'gd2';
            $config2['source_image'
] = './assets/foto/' . $foto;
            $config2['new_image']
= './assets/foto/thumb/' . $foto;
            $config2['maintain_ratio
'] = FALSE;
            $imgSize2 = $this-
>image_lib-
>get_image_properties($config2['sou
rce_image'], TRUE);

            if ($imgSize2['width'] >
$imgSize2['height']) {
                $config2['width'] =
$imgSize2['height'];
                $config2['height'] =
$imgSize2['height'];
                $config2['x_axis'] =
(($imgSize2['width'] / 2) -
($imgSize2['width'] / 2));
            } else {
                $config2['width'] =
$imgSize2['width'];
                $config2['height'] =
$imgSize2['width'];

```

```

        $config2['y_axis'] =
(($imgSize2['height'] / 2) -
($imgSize2['height'] / 2));
    }

    $this->image_lib-
>initialize($config2);

    if (!$this->image_lib-
>crop()) {
        echo $this-
>image_lib->display_errors();

        exit;
    }
} else {
    $foto = 'default.jpg';
}

$password = $this-
>security->xss_clean($this->input-
>post('password', TRUE));

$simpan = [
    'username' => $this-
>security->xss_clean($this->input-
>post('username', TRUE)),
    'fullname' => $this-
>security->xss_clean($this->input-
>post('fullname', TRUE)),
    'password' =>
password_hash($password,
PASSWORD_DEFAULT, ['cost' =>
10]),
    'hp' => $this->security-
>xss_clean($this->input->post('hp',
TRUE)),
    'alamat' => $this-
>security->xss_clean($this->input-
>post('alamat', TRUE)),
    'foto' => $foto,
    'level' => 'pegawai'
];

//simpan data

```

```

        $save = $this->m_pegawai->save('tbl_user',
$simpan);
        //jika data berhasil
disimpan, maka alihkan kehalaman
data pegawai
        if ($save) {
            $this->session->set_flashdata('success', 'Data
pegawai berhasil ditambahkan..');

            redirect('pegawai');
        }
    }

    $data = [
        'title' => 'Tambah Pegawai'
    ];

    $this->template->kasir('pegawai/form_input', $data);
}

public function
detail_pegawai($user = "")
{
    if ($user == "") {
        redirect('pegawai');
    }

    //validasi admin
    $this->is_admin();
    //ambil data pegawai
    $pegawai = $this->m_pegawai->getData('tbl_user', ['username' =>
$user, 'level' => 'pegawai']);
    //validasi jumlah data
    if ($pegawai->num_rows() != 1)
    {
        redirect('pegawai');
    }

    $data = [
        'title' => 'Detail Pegawai',
        'data' => $pegawai->row()

```

```

];

    $this->template->kasir('pegawai/detail', $data);
}

public function edit_data($user =
"")
{
    //validasi admin
    $this->is_admin();

    if ($user == "") {
        redirect('pegawai');
    }

    //ambil data pegawai
    $pegawai = $this->m_pegawai->getData('tbl_user', ['username' =>
$user, 'level' => 'pegawai']);
    //validasi jumlah data
    if ($pegawai->num_rows() != 1)
    {
        redirect('pegawai');
    }

    //ketika button di klik
    if ($this->security->xss_clean($this->input->post('submit', TRUE)) == 'submit')
    {
        //fetch data lama pegawai
        $p = $pegawai->row();
        //cek apakah admin merubah
username nya atau tidak
        if ($this->security->xss_clean($this->input->post('username', TRUE)) == $p->username) {
            $rules_username =
'required|min_length[5]|max_length[
20]|regex_match[/^[A-Za-z0-9]+$/]';
        } else {
            $rules_username =
'required|min_length[5]|max_length[
20]|is_unique[tbl_user.username]|reg
ex_match[/^[A-Za-z0-9]+$/]';

```

```

    }

    $this->form_validation-
>set_rules(
    'username',
    'Username',
    $rules_username,
    array(
        'required' => '{field}
wajib diisi',
        'min_length' => '{field}
minimal 5 karakter',
        'max_length' => '{field}
maksimal 20 karakter',
        'is_unique' =>
'Username sudah digunakan,
silahkan pilih yang lain',
        'regex_match' =>
'{field} hanya boleh huruf dan angka'
    )
);

```

```

    $this->form_validation-
>set_rules(
    'fullname',
    'Fullname',
    "required|min_length[3]|m
ax_length[30]|regex_match[/^[A-Z
a-z.']+$/]",
    array(
        'required' => '{field}
wajib diisi',
        'min_length' => '{field}
minimal 3 karakter',
        'max_length' => '{field}
maksimal 30 karakter',
        'regex_match' =>
'{field} hanya boleh huruf, titik dan
kutip satu (\')'
    )
);

```

//cek apakah form nomor hp
diisi atau tidak, jika diisi maka
lakukan validasi

```

    if ($this->input->post('hp',
TRUE) != "") {
        $this->form_validation-
>set_rules(
            'hp',
            'Nomor HP',
            "required|min_length[8]|
max_length[15]|regex_match[/^[0-
9]+$]/",
            array(
                'required' => '{field}
wajib diisi',
                'min_length' =>
'{field} minimal 8 karakter',
                'max_length' =>
'{field} maksimal 15 karakter',
                'regex_match' =>
'{field} hanya boleh angka'
            )
        );
    }

```

```

    $this->form_validation-
>set_rules(
    'alamat',
    'Alamat',
    "required|min_length[10]|
max_length[255]|regex_match[/^[A-
Z a-z.0-9-,']+$]/",
    array(
        'required' => '{field}
wajib diisi',
        'min_length' => '{field}
minimal 5 karakter',
        'max_length' => '{field}
maksimal 30 karakter',
        'regex_match' => 'Data
{field} yang anda masukkan tidak
valid'
    )
);

```

```

    $this->form_validation-
>set_rules(
    'status',
    'Status',

```

```

                'required|min_length[1]|max_
ax_length[1]|regex_match[/^[YN]+$/'
                ],
                array(
                    'required' => '{field}
wajib diisi',
                    'min_length' => '{field}
hanya boleh 1 karakter',
                    'max_length' => '{field}
hanya boleh 1 karakter',
                    'regex_match' => 'Input
{field} tidak valid'
                )
            );

            //jika proses validasi berhasil,
            maka lanjutkan ke proses update data
            if ($this->form_validation-
            >run() == TRUE) {
                //simpan data foto lama
                $foto = $p->foto;
                //konfigurasi upload
                $config['upload_path']
                = './assets/foto/';
                $config['allowed_types']
                = 'jpg|png|jpeg';
                $config['max_size'] =
                '2048';
                $config['file_name'] =
                'foto' . time();

                $this->upload-
                >initialize($config);

                if ($this->upload-
                >do_upload('foto')) {

                    $b = $this->upload-
                    >data();
                    $foto = $b['file_name'];

                    //buat foto thumbnail
                    untuk icon user
                    $this->load-
                    >library('image_lib');

```

```

                $config2['image_library'
                ] = 'gd2';
                $config2['source_image'
                ] = './assets/foto/' . $foto;
                $config2['new_image']
                = './assets/foto/thumb/' . $foto;
                $config2['maintain_ratio
                '] = FALSE;
                $imgSize2 = $this-
                >image_lib-
                >get_image_properties($config2['sou
                rce_image'], TRUE);

                if ($imgSize2['width'] >
                $imgSize2['height']) {
                    $config2['width'] =
                    $imgSize2['height'];
                    $config2['height'] =
                    $imgSize2['height'];
                    $config2['x_axis'] =
                    (($imgSize2['width'] / 2) -
                    ($imgSize2['width'] / 2));
                } else {
                    $config2['width'] =
                    $imgSize2['width'];
                    $config2['height'] =
                    $imgSize2['width'];
                    $config2['y_axis'] =
                    (($imgSize2['height'] / 2) -
                    ($imgSize2['height'] / 2));
                }

                $this->image_lib-
                >initialize($config2);

                if (!$this->image_lib-
                >crop()) {
                    echo $this-
                    >image_lib->display_errors();

                    exit;
                }
                //hapus foto lama, jika
                foto bukan foto default
                if ($p->foto !=
                'default.jpg' && $p->foto != ") {

```

```

        //hapus foto utama
        unlink('./assets/foto/' .
$P->foto);
        //hapus foto profil
        unlink('./assets/foto/th
umb/' . $P->foto);
    }
}

//masukkan data ke
variable array
$update = [
    'username' => $this-
>security->xss_clean($this->input-
>post('username', TRUE)),
    'fullname' => $this-
>security->xss_clean($this->input-
>post('fullname', TRUE)),
    'hp' => $this->security-
>xss_clean($this->input->post('hp',
TRUE)),
    'alamat' => $this-
>security->xss_clean($this->input-
>post('alamat', TRUE)),
    'foto' => $foto,
    'active' => $this-
>security->xss_clean($this->input-
>post('status', TRUE))
];

$where = [
    'id_user' => $this-
>security->xss_clean($this->input-
>post('Id', TRUE))
];
//proses update
$up = $this->m_pegawai-
>update('tbl_user', $update, $where);
//jika proses update
berhasil, maka alihkan ke halaman
data pegawai dan tampilkan pesan
if ($up) {
    $this->session-
>set_flashdata('success', 'Data
pegawai berhasil diperbarui..');

```

```

        redirect('pegawai');
    }
}

}

$data = [
    'title' => 'Edit Pegawai',
    'pegawai' => $pegawai-
>row()
];

$this->template-
>kasir('pegawai/form_edit', $data);
}

public function ganti_password()
{
    $this->is_admin();

    if ($this->input->post('submit',
TRUE) == 'Submit') {
        //validasi
        $this->form_validation-
>set_rules(
            'pwUser',
            'Password Baru Pegawai',
            "required|min_length[3]|m
ax_length[30]",
            array(
                'required' => '{field}
wajib diisi',
                'min_length' => '{field}
minimal 3 karakter',
                'max_length' => '{field}
maksimal 30 karakter'
            )
        );

        $this->form_validation-
>set_rules(
            'pwAdmin',
            'Password Admin',
            "required|min_length[3]|m
ax_length[30]",
            array(

```

```

                'required' => '{field}'
wajib diisi',
                'min_length' => '{field}'
minimal 3 karakter',
                'max_length' => '{field}'
maksimal 30 karakter'
            )
        );

        $this->form_validation-
>set_rules(
            'IdU',
            'ID User',
            "required|regex_match[/^[0
-9]+$/]\"",
            array(
                'required' => 'Data tidak
valid',
                'regex_match' => 'Data
tidak valid'
            )
        );

        if ($this->form_validation-
>run() == TRUE) {
            //validasi password admin
            $whereAdmin = [
                'id_user' => $this-
>session->userdata('UserID'),
                'level' => 'admin'
            ];

            $getDataAdmin = $this-
>m_pegawai->getData('tbl_user',
            $whereAdmin);
            //cek jumlah data
            if ($getDataAdmin-
>num_rows() != 1) {
                redirect('dashboard');
            }
            //fetch data admin
            $a = $getDataAdmin-
>row();

```

```

            $passwordUser = $this-
>security->xss_clean($this->input-
>post('pwUser', TRUE));
            $passwordAdmin = $this-
>security->xss_clean($this->input-
>post('pwAdmin', TRUE));
            //validasi password
            if
            (password_verify($passwordAdmin,
            $a->password)) {
                //lakukan update data
                user
                $hashPW =
                password_hash($passwordUser,
                PASSWORD_DEFAULT, ['cost' =>
                10]);
                $idUser = $this-
>security->xss_clean($this->input-
>post('IdU', TRUE));

                $updated = $this-
>m_pegawai->update('tbl_user',
                ['password' => $hashPW], ['id_user'
                => $idUser, 'level' => 'pegawai']);

                if ($updated) {
                    $this->session-
>set_flashdata('success', 'Password
pegawai berhasil diganti...');

                    redirect('pegawai');
                } else {
                    $this->session-
>set_flashdata('error', 'Password
pegawai gagal diganti...');

                    redirect('pegawai');
                }
                } else {
                    $this->session-
>set_flashdata('error', 'Password
Admin anda salah, Password
Pegawai gagal diperbarui..');

                    redirect('pegawai');
                }
            }

```

```

        } else {
            $this->session-
>set_flashdata('error',
validation_errors('<p class="mb-0
mt-0"><i class="fa fa-caret-
right"></i> ', '</p>'));

            redirect('pegawai');
        }
    } else {
        redirect('dashboard');
    }
}

public function ajax_pegawai()
{
    $this->is_admin();
    //cek apakah request berupa ajax
atau bukan, jika bukan maka redirect
ke home
    if ($this->input-
>is_ajax_request()) {
        //ambil list data
        $list = $this->m_pegawai-
>get_datatables();
        //siapkan variabel array
        $data = array();
        $no = $_POST['start'];

        foreach ($list as $i) {

            $no++;
            $row = array();
            $row[] = $no;
            $row[] = $i->username;
            $row[] = $i->fullname;
            $row[] = ($i->active ==
'Y') ? 'Aktif' : 'Tidak Aktif';
            $row[] = ($i->last_login !=
'') ? date('d/m/Y - H:i:s', strtotime($i-
>last_login)) : '';
            $row[] = '<a href="' .
site_url('pegawai/' . $i->username) .
'" class="btn btn-success btn-
sm">Detail</a>

```

```

        <a href="' .
site_url('edit_pegawai/' . $i-
>username) . '" class="btn btn-
warning btn-sm text-
white">Edit</a>
        <button type="button"
class="btn btn-info btn-sm text-
white" data-toggle="modal" data-
target="#modal-password"
onclick="GantiPW(' . $i->id_user .
')">Ubah Password</button>;

        $data[] = $row;
    }

    $output = array(
        "draw" =>
$_POST['draw'],
        "recordsTotal" => $this-
>m_pegawai->count_all(),
        "recordsFiltered" => $this-
>m_pegawai->count_filtered(),
        "data" => $data
    );
    //output to json format
    echo json_encode($output);
} else {
    redirect('dashboard');
}
}

private function is_admin()
{
    if (!$this->session-
>userdata('level') || $this->session-
>userdata('level') != 'admin') {
        redirect('dashboard');
    }
}
}

```

Data_supplier.php

```

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

```

```

class Data_supplier extends
CI_Controller
{
    function __construct()
    {
        parent::__construct();
        //load library
        $this->load->library(['template',
'form_validation']);
        //load Model
        $this->load-
>model('m_supplier');

        header('Last-Modified:' .
gmdate('D, d M Y H:i:s') . 'GMT');
        header('Cache-Control: no-
cache, must-revalidate, max-age=0');
        header('Cache-Control: post-
check=0, pre-check=0', false);
        header('Pragma: no-cache');
    }

    public function index()
    {
        $this->is_admin();

        $data = [
            'title' => 'Data Supplier'
        ];

        $this->template-
>kasir('supplier/index', $data);
    }

    public function tambah_supplier()
    {
        $this->is_admin();
        //ketika user mengklik submit
        if ($this->input->post('submit',
TRUE) == 'submit') {
            //validasi form
            $this->form_validation-
>set_rules(
                'nama_supplier',
                'Nama Supplier',

```

```

                "required|min_length[2]|m
ax_length[100]|regex_match[/^[A-Z
a-z.0-9']+$/]",
                array(
                    'required' => '{field}
wajib diisi',
                    'min_length' => '{field}
minimal 2 karakter',
                    'max_length' => '{field}
maksimal 100 karakter',
                    'regex_match' =>
                    '{field} tidak valid'
                )
            );

            if ($this->input->post('hp',
TRUE) != "") {
                $this->form_validation-
>set_rules(
                    'hp',
                    'Nomor Telp.',
                    "required|min_length[8]|
max_length[15]|regex_match[/^[0-
9']+$/]",
                    array(
                        'required' => '{field}
wajib diisi',
                        'min_length' =>
                        '{field} minimal 8 karakter',
                        'max_length' =>
                        '{field} maksimal 15 karakter',
                        'regex_match' =>
                        '{field} hanya boleh angka'
                    )
                );

                $this->form_validation-
>set_rules(
                    'alamat',
                    'Alamat',
                    "required|min_length[10]|
max_length[255]|regex_match[/^[A-
Z a-z.0-9-,']+$/]",
                    array(

```

```

                'required' => '{field}'
wajib diisi',
                'min_length' => '{field}'
minimal 5 karakter',
                'max_length' => '{field}'
maksimal 30 karakter',
                'regex_match' => 'Data
{field} yang anda masukkan tidak
valid'
            )
        );

```

```

        //jika validasi berhasil maka
        lakukan proses penyimpanan
        if ($this->form_validation-
        >run() == TRUE) {
            //tampung data ke variabel
            $id = 'ID' . time();
            $nama = $this->security-
            >xss_clean($this->input-
            >post('nama_supplier', TRUE));
            $telp = $this->security-
            >xss_clean($this->input->post('hp',
            TRUE));
            $alamat = $this->security-
            >xss_clean($this->input-
            >post('alamat', TRUE));

```

```

            $data_simpan = [
                'id_supplier' => $id,
                'nama_supplier' =>
$nama,
                'alamat' => $alamat,
                'telp' => $telp
            ];

```

```

            $simpan = $this-
            >m_supplier->save('tbl_supplier',
            $data_simpan);

            if ($simpan) {
                $this->session-
                >set_flashdata('success', 'Data
                Supplier berhasil ditambahkan..');

                redirect('supplier');

```

```

            }
        }
    }

    $data = [
        'title' => 'Tambah Supplier'
    ];

    $this->template-
    >kasir('supplier/form_input', $data);
}

```

```

    public function edit_supplier($id)
    {
        $this->is_admin();

        //ketika user mengklik submit
        if ($this->input->post('submit',
        TRUE) == 'submit') {
            //validasi form
            $this->form_validation-
            >set_rules(
                'idSupplier',
                'ID Supplier',
                'required|min_length[10]',
                array(
                    'required' => '{field}'
wajib diisi',
                    'min_length' => '{field}'
tidak valid'
                )
            );

```

```

            $this->form_validation-
            >set_rules(
                'nama_supplier',
                'Nama Supplier',
                "required|min_length[2]|m
                ax_length[100]|regex_match[/^[A-Z
                a-z.0-9']+$/] ",
                array(
                    'required' => '{field}'
wajib diisi',
                    'min_length' => '{field}'
minimal 2 karakter',

```

```

                'max_length' => '{field}
maksimal 100 karakter',
                'regex_match' =>
'{field} tidak valid'
            )
        );

        if ($this->input->post('hp',
TRUE) != "") {
            $this->form_validation-
>set_rules(
                'hp',
                'Nomor Telp.',
                "required|min_length[8]|
max_length[15]|regex_match[/^[0-
9]+$]/",
                array(
                    'required' => '{field}
wajib diisi',
                    'min_length' =>
'{field} minimal 8 karakter',
                    'max_length' =>
'{field} maksimal 15 karakter',
                    'regex_match' =>
'{field} hanya boleh angka'
                )
            );
        }

        $this->form_validation-
>set_rules(
            'alamat',
            'Alamat',
            "required|min_length[10]|
max_length[255]|regex_match[/^[A-
Z a-z.0-9-,']+$]/",
            array(
                'required' => '{field}
wajib diisi',
                'min_length' => '{field}
minimal 5 karakter',
                'max_length' => '{field}
maksimal 30 karakter',
                'regex_match' => 'Data
{field} yang anda masukkan tidak
valid'
            )
        );

```

```

        )
    );

    //jika validasi berhasil maka
    lakukan proses penyimpanan
    if ($this->form_validation-
>run() == TRUE) {
        //tampung data ke variabel
        $idSupplier = $this-
>security->xss_clean($this->input-
>post('idSupplier', TRUE));
        $nama = $this->security-
>xss_clean($this->input-
>post('nama_supplier', TRUE));
        $telp = $this->security-
>xss_clean($this->input->post('hp',
TRUE));
        $alamat = $this->security-
>xss_clean($this->input-
>post('alamat', TRUE));

        $data_update = [
            'nama_supplier' =>
$nama,
            'alamat' => $alamat,
            'telp' => $telp
        ];

        $sup = $this->m_supplier-
>update('tbl_supplier', $data_update,
['id_supplier' => $idSupplier]);

        if ($sup) {
            $this->session-
>set_flashdata('success', 'Data
Supplier berhasil diperbarui.');
```

```

            redirect('supplier');
        }
    }

    //ambil data
    $where = [
        'id_supplier' => $this-
>security->xss_clean($id)
    ];

```

```

];
$getData = $this->m_supplier-
>getData('tbl_supplier', $where);
//cek jumlah data
if ($getData->num_rows() != 1)
{
    redirect('supplier');
}

$data = [
    'title' => 'Edit Supplier',
    'data' => $getData->row()
];

$this->template-
>kasir('supplier/form_edit', $data);
}

public function hapus_data()
{
    //cek login
    $this->is_admin();
    //validasi request ajax
    if ($this->input-
>is_ajax_request()) {
        //validasi
        $this->form_validation-
>set_rules(
            'id',
            'ID Supplier',
            "required|min_length[10]",
            array(
                'required' => '{field}
tidak valid',
                'min_length' => 'Isi
{field} tidak valid'
            )
        );

        if ($this->form_validation-
>run() == TRUE) {
            //tangkap rowid
            $id = $this->security-
>xss_clean($this->input->post('id',
TRUE));

```

```

        $hapus = $this-
>m_supplier->delete('tbl_supplier',
['id_supplier' => $id]);

        if ($hapus) {
            echo
            json_encode(['message' =>
'success']);
        } else {
            echo
            json_encode(['message' => 'failed']);
        }
        } else {
            echo
            json_encode(['message' => 'failed']);
        }
        } else {
            redirect('dashboard');
        }
    }

    public function ajax_supplier()
    {
        $this->is_admin();
        //cek apakah request berupa ajax
        atau bukan, jika bukan maka redirect
        ke home
        if ($this->input-
>is_ajax_request()) {
            //ambil list data
            $list = $this->m_supplier-
>get_datatables();
            //siapkan variabel array
            $data = array();
            $no = $_POST['start'];

            foreach ($list as $i) {

                $no++;
                $row = array();
                $row[] = $no;
                $row[] = $i->id_supplier;
                $row[] = $i-
>nama_supplier;
                $row[] = $i->alamat;

```

```

                $row[] = ($i->telp != "") ?
$i->telp : '-';
                $row[] = '<a href=' .
site_url('supplier/' . $i->id_supplier) .
'" class="btn btn-warning btn-sm
text-white">Edit</a>
                <button type="button"
class="btn btn-danger btn-
sm"onclick="hapus_supplier(\' . $i-
>id_supplier . '\")">Hapus</button>;

                $data[] = $row;
            }

            $output = array(
                "draw" =>
$_POST['draw'],
                "recordsTotal" => $this-
>m_supplier->count_all(),
                "recordsFiltered" => $this-
>m_supplier->count_filtered(),
                "data" => $data
            );
            //output to json format
            echo json_encode($output);
        } else {
            redirect('dashboard');
        }
    }

    private function is_admin()
    {
        if (!$this->session-
>userdata('level') || $this->session-
>userdata('level') != 'admin') {
            redirect('dashboard');
        }
    }
}

```

Pembelian.php

```

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

```

```

class Pembelian extends
CI_Controller
{
    function __construct()
    {
        parent::__construct();
        //load library
        $this->load->library(['template',
'form_validation', 'cart']);
        //load model
        $this->load-
>model('m_pembelian');

        header('Last-Modified:' .
gmdate('D, d M Y H:i:s') . 'GMT');
        header('Cache-Control: no-
cache, must-revalidate, max-age=0');
        header('Cache-Control: post-
check=0, pre-check=0', false);
        header('Pragma: no-cache');
    }

    public function index()
    {
        //cek login
        $this->is_login();
        //kosongkan cart
        $this->cart->destroy();

        $data = [
            'title' => 'Data Pembelian
Barang'
        ];

        $this->template-
>kasir('pembelian/index', $data);
    }

    public function tambah_data()
    {
        $this->is_login();

        //ketika button simpan di klik
        maka lakukan proses validasi dan
        penyimpanan data
    }
}

```

```

        if ($this->input->post('submit',
TRUE) == 'Submit') {
            //cek apakah user sudah
memilih barang atau belum, jika
belum maka munculkan pesan
kesalahan
            if (!$this->cart->contents()) {
                $this->session-
>set_flashdata('alert', 'Anda belum
memilih barang...');

                redirect('tambah_pembelia
n', 'refresh');
            }
            //validasi input data tanggal
            $this->form_validation-
>set_rules(
                'tanggal',
                'Tanggal Pembelian',
                'required|callback_checkDa
teFormat',
                array(
                    'required' => '{field}
wajib diisi',
                    'checkDateFormat' =>
'{field} tidak valid'
                )
            );

            $this->form_validation-
>set_rules(
                'supplier',
                'Supplier',
                'required|min_length[10]',
                array(
                    'required' => '{field}
wajib dipilih',
                    'min_length' => '{field}
tidak valid'
                )
            );

            if ($this->form_validation-
>run() == TRUE) {

                $id = 'ID' . time();

```

```

                $tgl = date('Y-m-d',
strtotime(str_replace('/', '-', $this-
>security->xss_clean($this->input-
>post('tanggal', TRUE))));
                $sup = $this->security-
>xss_clean($this->input-
>post('supplier', TRUE));
                $user = $this->session-
>userdata('UserID');

                $data_pembelian = [
                    'id_pembelian' => $id,
                    'tgl_pembelian' => $tgl,
                    'id_supplier' => $sup,
                    'id_user' => $user
                ];
                //baca cart dan
memasukkannya dalam array untuk
disimpan
                $cart = array();

                foreach ($this->cart-
>contents() as $c) {
                    $item = [
                        'id_pembelian' => $id,
                        'id_barang' => $c['id'],
                        'qty' => $c['qty'],
                        'harga' => $c['price']
                    ];

                    //push ke array cart
                    array_push($cart,
                    $item);
                }
                //simpan data pembelian
                $simpan = $this-
>m_pembelian-
>save('tbl_pembelian',
                $data_pembelian);

                if ($simpan) {
                    //simpan data detail
pembelian
                    $this->m_pembelian-
>multiSave('tbl_detail_pembelian',
                    $cart);

```

```

        //kosongkan cart
        $this->cart->destroy();
        //buat notifikasi
        penyimpanan berhasil
        $this->session-
        >set_flashdata('success', 'Data
        pembelian berhasil ditambahkan...');

        redirect('data_pembelian
        ');
    }
}

$data = [
    'title' => 'Tambah Data
    Pembelian',
    'data' => $this-
    >m_pembelian-
    >getData('tbl_barang', ['active' =>
    'Y']),
    'supplier' => $this-
    >m_pembelian-
    >getAllData('tbl_supplier'),
    'table' => $this->read_cart()
];

$this->template-
>kasir('pembelian/form_input',
$data);
}

public function hapus_data()
{
    //cek login
    $this->is_login();
    //validasi request ajax
    if ($this->input-
    >is_ajax_request()) {
        //validasi
        $this->form_validation-
        >set_rules(
            'id',
            'ID Pembelian',
            "required|min_length[10]",
            array(

```

```

            'required' => '{field}
            tidak valid',
            'min_length' => 'Isi
            {field} tidak valid'
        )
    );

    if ($this->form_validation-
    >run() == TRUE) {
        //tangkap rowid
        $id = $this->security-
        >xss_clean($this->input->post('id',
        TRUE));

        $hapus = $this-
        >m_pembelian-
        >delete(['tbl_pembelian',
        'tbl_detail_pembelian'],
        ['id_pembelian' => $id]);

        if ($hapus) {
            echo
            json_encode(['message' =>
            'success']);
        } else {
            echo
            json_encode(['message' => 'failed']);
        }
        } else {
            echo
            json_encode(['message' => 'failed']);
        }
        } else {
            redirect('dashboard');
        }
    }

    public function
    detail_pembelian($id = null)
    {
        if ($id == null) {
            redirect('data_pembelian');
        }

        //cek login
        $this->is_login();

```

```

        //ambil data
        $getData = $this->m_pembelian->getDataPembelian($this->security->xss_clean($id));

        if ($getData->num_rows() < 1)
        {
            redirect('dashboard');
        }

        $data = [
            'title' => 'Detail Pembelian ' . $id,
            'data' => $getData
        ];

        $this->template->kasir('pembelian/detail', $data);
    }

    public function edit_pembelian($id = null)
    {
        if ($id == null) {
            redirect('data_pembelian');
        }
        //ambil data pembelian
        $getData = $this->m_pembelian->getDataPembelian($this->security->xss_clean($id));
        //hitung data
        if ($getData->num_rows() < 1)
        {
            redirect('data_pembelian');
        }
        //ketika button diklik
        if ($this->input->post('submit', TRUE) == 'Update') {
            //cek apakah user sudah memilih barang atau belum, jika belum maka munculkan pesan kesalahan
            if (!$this->cart->contents()) {

```

```

                $this->session->set_flashdata('alert', 'Anda belum memilih barang...');

                redirect('edit_pembelian/' . $id, 'refresh');
            }
            //validasi input data tanggal
            $this->form_validation->set_rules(
                'tanggal',
                'Tanggal Pembelian',
                'required|callback_checkDateFormat',
                array(
                    'required' => '{field} wajib diisi',
                    'checkDateFormat' => '{field} tidak valid'
                )
            );

            $this->form_validation->set_rules(
                'supplier',
                'Supplier',
                'required|min_length[10]',
                array(
                    'required' => '{field} wajib dipilih',
                    'min_length' => '{field} tidak valid'
                )
            );

            $this->form_validation->set_rules(
                'idP',
                'ID Pembelian',
                'required|min_length[10]',
                array(
                    'required' => '{field} wajib diisi',
                    'min_length' => '{field} tidak valid'
                )
            );

```

```

    );

    if ($this->form_validation-
>run() == TRUE) {

        $idP = $this->security-
>xss_clean($this->input->post('idP',
TRUE));

        $tgl = date('Y-m-d',
strtotime(str_replace('/', '-', $this-
>security->xss_clean($this->input-
>post('tanggal', TRUE))));
        $sup = $this->security-
>xss_clean($this->input-
>post('supplier', TRUE));

        $data_pembelian = [
            'tgl_pembelian' => $tgl,
            'id_supplier' => $sup
        ];

        //baca cart dan
memasukkannya dalam array untuk
disimpan
        $cart = array();

        foreach ($this->cart-
>contents() as $c) {
            $item = [
                'id_pembelian' =>
$idP,
                'id_barang' => $c['id'],
                'qty' => $c['qty'],
                'harga' => $c['price']
            ];

            //push ke array cart
            array_push($cart,
$item);
        }
        //simpan data pembelian
        $update = $this-
>m_pembelian-
>update('tbl_pembelian',
$data_pembelian, ['id_pembelian' =>
$idP]);

```

```

        if ($update) {
            //hapus detail pembelian
            $this->m_pembelian-
>delete('tbl_detail_pembelian',
['id_pembelian' => $idP]);
            //simpan data detail
pembelian
            $this->m_pembelian-
>multiSave('tbl_detail_pembelian',
$cart);
            //kosongkan cart
            $this->cart->destroy();
            //buat notifikasi
penyimpanan berhasil
            $this->session-
>set_flashdata('success', 'Data
pembelian berhasil diperbarui...');

            redirect('data_pembelian
');
        }
    }
    //cek apakah yang akses adalah
admin / user yang menginputkan,
jika bukan keduanya maka redirect
ke halaman data pembelian
    $fData = $getData->row();

    if ($this->session-
>userdata('level') != 'admin' &&
$this->session->userdata('UserID')
!= $fData->id_user) {
        redirect('data_pembelian');
    }
    //masukkan detail pembelian ke
cart
    if (!$this->cart->contents()) {
        $dataCart = [];

        foreach ($getData->result() as
$c) {
            $keranjang = array(
                'id' => $c-
>kode_barang,

```

```

        'qty' => $c->qty,
        'price' => $c->harga,
        'name' => $c->nama_barang);

        array_push($dataCart, $keranjang);
    }

    $this->cart->insert($dataCart);
}

$data = [
    'title' => 'Edit Data Pembelian',
    'fdata' => $fData,
    'data' => $this->m_pembelian->getData('tbl_barang', ['active' => 'Y']),
    'supplier' => $this->m_pembelian->getAllData('tbl_supplier', ['table' => $this->read_cart()]);
];

$this->template->kasir('pembelian/form_edit', $data);

public function tambah_cart()
{
    //cek login
    $this->is_login();
    //validasi request ajax
    if ($this->input->is_ajax_request()) {
        //validasi data
        $this->form_validation->set_rules(
            'barangx',
            'Barang',
            'required|min_length[3]|max_length[6]',

```

```

            array(
                'required' => '{field} wajib dipilih',
                'min_length' => 'Isi {field} tidak valid',
                'max_length' => 'Isi {field} tidak valid',
            )
        );

        $this->form_validation->set_rules(
            'jumlah',
            'Jumlah',
            "required|min_length[1]|regex_match[/^[0-9]+$/]|greater_than[0]",
            array(
                'required' => '{field} wajib diisi',
                'min_length' => 'Isi {field} tidak valid',
                'regex_match' => '{field} hanya boleh angka',
                'greater_than' => '{field} harus lebih dari nol'
            )
        );

        $this->form_validation->set_rules(
            'harga',
            'Harga Satuan',
            "required|min_length[2]|regex_match[/^[0-9.]+$/]|greater_than[0]",
            array(
                'required' => '{field} wajib diisi',
                'min_length' => 'Isi {field} tidak valid',
                'regex_match' => '{field} hanya boleh angka',
                'greater_than' => '{field} harus lebih dari nol'
            )
        );
    }
}

```

```

    );

    if ($this->form_validation-
>run() == TRUE) {
        //ambil barang sesuai kode
        $get_barang = $this-
>m_pembelian-
>getData('tbl_barang', ['kode_barang'
=> $this->security->xss_clean($this-
>input->post('barangx', TRUE)),
'active' => 'Y']);

        if ($get_barang-
>num_rows() == 1) {
            //fetch data barang dan
            masukkan kedalam cart
            $b = $get_barang-
>row();

            $keranjang = array(
                'id' => $b-
>kode_barang,
                'qty' => $this-
>security->xss_clean($this->input-
>post('jumlah', TRUE)),
                'price' => $this-
>security->xss_clean(str_replace('.',
'', $this->input->post('harga',
TRUE))),
                'name' => $b-
>nama_barang
            );

            $this->cart-
>insert($keranjang);

            $table = $this-
>read_cart();

            $alert = '<div
class="alert alert-success"
role="alert">Data berhasil
ditambahkan ke daftar</div>';

```

```

        $arr = array('table' =>
$table, 'alert' => $alert, 'status' =>
'success');

        echo json_encode($arr);
    } else {
        $table = $this-
>read_cart();

        $alert = '<div
class="alert alert-danger"
role="alert">Data barang tidak
valid</div>';

        $arr = array('table' =>
$table, 'alert' => $alert, 'status' =>
'gagal');

        echo json_encode($arr);
    }
} else {
    $table = $this->read_cart();

    $alert = '<div class="alert
alert-danger" role="alert">' .
validation_errors('<p class="mb-0
mt-0"><i class="fa fa-caret-
right"></i> ', '</p>') . '</div>';

    $arr = array('table' =>
$table, 'alert' => $alert, 'status' =>
'gagal');

    echo json_encode($arr);
}
}

public function get_item()
{
    //cek login
    $this->is_login();
    //validasi request ajax

```

```

        if ($this->input-
>is_ajax_request()) {
            //tangkap rowid
            $rowid = $this->security-
>xss_clean($this->input-
>post('rowid', TRUE));

            $get_item = $this->cart-
>get_item($rowid);

            if ($get_item) {
                $arr = [
                    'barang' =>
$get_item['id'],
                    'qty' => $get_item['qty'],
                    'harga' =>
number_format($get_item['price'], 0,
'', ''),
                    'rowid' => '<input
type="hidden" id="rowid" value="' .
$get_item['rowid'] . '" />',
                    'table' => $this-
>read_cart(),
                    'status' => 'true'
                ];
            } else {
                $arr = [
                    'barang' => "",
                    'qty' => "",
                    'harga' => "",
                    'rowid' => "",
                    'table' => $this-
>read_cart(),
                    'status' => 'false'
                ];
            }

            echo json_encode($arr);
        } else {
            redirect('dashboard');
        }
    }

    public function update_cart()
    {
        //cek login

```

```

        $this->is_login();
        //validasi request ajax
        if ($this->input-
>is_ajax_request()) {
            //validasi data
            $this->form_validation-
>set_rules(
                'jumlah',
                'Jumlah',
                "required|min_length[1]|re
gex_match[/^[0-
9]+$|greater_than[0]",
                array(
                    'required' => '{field}
wajib diisi',
                    'min_length' => 'Isi
{field} tidak valid',
                    'regex_match' =>
'{field} hanya boleh angka',
                    'greater_than' => '{field}
harus lebih dari nol'
                )
            );

            $this->form_validation-
>set_rules(
                'harga',
                'Harga Satuan',
                "required|min_length[2]|re
gex_match[/^[0-
9.]+$|greater_than[0]",
                array(
                    'required' => '{field}
wajib diisi',
                    'min_length' => 'Isi
{field} tidak valid',
                    'regex_match' =>
'{field} hanya boleh angka',
                    'greater_than' => '{field}
harus lebih dari nol'
                )
            );

            $this->form_validation-
>set_rules(
                'rowid',

```

```

        'Row ID',
        "required|min_length[10]",
        array(
            'required' => '{field}
tidak valid',
            'min_length' => 'Isi
{field} tidak valid'
        )
    );

    if ($this->form_validation-
>run() == TRUE) {

        $keranjang = array(
            'rowid' => $this-
>security->xss_clean($this->input-
>post('rowid', TRUE)),
            'qty' => $this->security-
>xss_clean($this->input-
>post('jumlah', TRUE)),
            'price' => $this-
>security->xss_clean(str_replace('.',
'', $this->input->post('harga',
TRUE)))
        );

        $this->cart-
>update($keranjang);

        $table = $this->read_cart();

        $alert = '<div class="alert
alert-success" role="alert">Data
barang berhasil diubah</div>';

        $arr = array('table' =>
$table, 'alert' => $alert, 'status' =>
'success');

        echo json_encode($arr);
    } else {
        $table = $this->read_cart();

        $alert = '<div class="alert
alert-danger" role="alert">'.
validation_errors('<p class="mb-0

```

```

mt-0"><i class="fa fa-caret-
right"></i> ', '</p>') . '</div>';

        $arr = array('table' =>
$table, 'alert' => $alert, 'status' =>
'gagal');

        echo json_encode($arr);
    }
    } else {
        redirect('dashboard');
    }
}

public function remove_item()
{
    //cek login
    $this->is_login();
    //validasi request ajax
    if ($this->input-
>is_ajax_request()) {
        //validasi
        $this->form_validation-
>set_rules(
            'rowid',
            'Row ID',
            "required|min_length[10]",
            array(
                'required' => '{field}
tidak valid',
                'min_length' => 'Isi
{field} tidak valid'
            )
        );

        if ($this->form_validation-
>run() == TRUE) {
            //tangkap rowid
            $rowid = $this->security-
>xss_clean($this->input-
>post('rowid', TRUE));

            $this->cart-
>remove($rowid);

```

```

        $alert = '<div class="alert
alert-success" role="alert">Data
barang berhasil dihapus</div>';

```

```

        $message = 'success';
    } else {
        $alert = '<div class="alert
alert-danger" role="alert">' .
validation_errors('<p class="mb-0
mt-0"><i class="fa fa-caret-
right"></i> ', '</p>') . '</div>';

```

```

        $message = 'failed';
    }

```

```

    $table = $this->read_cart();

```

```

    $arr = array('table' => $table,
'alert' => $alert, 'message' =>
$message);

```

```

        echo json_encode($arr);
    } else {
        redirect('dashboard');
    }
}

```

```

function checkDateFormat($date)
{
    if (preg_match("/^(0[1-9]|[1-
2][0-9]|3[0-1])\.(0[1-9]|1[0-2])\.[0-
9]{4}$/", $date)) {
        if (checkdate(substr($date, 3,
2), substr($date, 0, 2), substr($date,
6, 4))) {
            return true;
        } else {
            return false;
        }
    } else {
        return false;
    }
}

```

```

public function ajax_pembelian()
{

```

```

        $this->is_login();
        //cek apakah request berupa ajax
atau bukan, jika bukan maka redirect
ke home

```

```

        if ($this->input-
>is_ajax_request()) {
            //ambil list data
            $list = $this->m_pembelian-
>get_datatables();
            //siapkan variabel array
            $data = array();
            $no = $_POST['start'];

```

```

        foreach ($list as $i) {

```

```

            $button = "";
            if ($this->session-
>userdata('level') == 'admin' || $this-
>session->userdata('UserID') == $i-
>id_user) :

```

```

                $button .= '<a href="' .
site_url('edit_pembelian/' . $i-
>id_pembelian) . '" class="btn btn-
warning btn-sm text-
white">Edit</a>
                <button type="button"
class="btn btn-danger btn-
sm" onclick="hapus_pembelian(\' .
$i->id_pembelian .
'\")">Hapus</button>';

```

```

            endif;

```

```

            $no++;
            $row = array();
            $row[] = $no;
            $row[] = $i-
>id_pembelian;
            $row[] = $this-
>tanggal_indo($i->tgl_pembelian);
            $row[] = $i-
>nama_supplier;
            $row[] = $i->jumlah;

```

```

        $row[] = '<span class="pr-3">' . number_format($i->total, 0, ',', '.') . ',</span>';
        $row[] = $i->fullname;
        $row[] = '<a href="' . site_url('data_pembelian/' . $i->id_pembelian) . '" class="btn btn-sm btn-success">Detail</a>' . $button;

        $data[] = $row;
    }

    $output = array(
        "draw" =>
        $_POST['draw'],
        "recordsTotal" => $this->m_pembelian->count_all(),
        "recordsFiltered" => $this->m_pembelian->count_filtered(),
        "data" => $data
    );
    //output to json format
    echo json_encode($output);
} else {
    redirect('dashboard');
}
}

private function read_cart()
{
    if ($this->cart->contents()) {

        $table = "";
        $i = 1;
        foreach ($this->cart->contents() as $c) {
            $table .= '<tr><td>' . $i++ . '</td>';
            $table .= '<td>' . $c['name'] . '</td>';
            $table .= '<td class="text-center">' . $c['qty'] . '</td>';
            $table .= '<td class="text-right">' . number_format($c['price'], 0, ',', '.') . '</td>';

```

```

        $table .= '<td class="text-right">' .
        number_format($c['subtotal'], 0, ',', '.') . '</td>';
        $table .= '<td class="text-center">
            <button
            type="button" class="btn btn-warning btn-sm text-white"
            onclick="get_item(\' . $c['rowid'] . '\')>Edit</button>
            <button
            type="button" class="btn btn-danger btn-sm text-white"
            onclick="remove_item(\' . $c['rowid'] . '\')>Hapus</button>
            </td></tr>';
        }
    } else {
        $table = '<tr>
            <td scope="col"
            colspan="6" class="text-center"><i>Belum ada data</i></td>
            </tr>';
    }

    return $table;
}

private function tanggal_indo($tgl)
{
    $bulan = [1 => 'Januari',
    'Februari', 'Maret', 'April', 'Mei',
    'Juni', 'Juli', 'Agustus', 'September',
    'Oktober', 'November', 'Desember'];

    $exp = explode('-', date('d-m-Y', strtotime($tgl)));

    return $exp[0] . '-' . $bulan[(int)
    $exp[1]] . '-' . $exp[2];
}

private function is_login()
{

```

```

        if (!$this->session-
>userdata('level')) {
            redirect('login');
        }
    }
}

```

Penjualan.php

```

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

class Penjualan extends
CI_Controller
{
    function __construct()
    {
        parent::__construct();
        //load library
        $this->load->library(['template',
'form_validation', 'cart']);
        //load model
        $this->load-
>model('m_penjualan');

        header('Last-Modified:' .
gmdate('D, d M Y H:i:s') . 'GMT');
        header('Cache-Control: no-
cache, must-revalidate, max-age=0');
        header('Cache-Control: post-
check=0, pre-check=0', false);
        header('Pragma: no-cache');
    }

    public function index()
    {
        //cek login
        $this->is_login();
        //kosongkan cart
        $this->cart->destroy();

        $data = [
            'title' => 'Data penjualan
Barang'
        ];
    }
}

```

```

        $this->template-
>kasir('penjualan/index', $data);
    }

    public function tambah_data()
    {
        //cek login
        $this->is_login();
        // ketika button submit di klik
        if ($this->input->post('submit',
TRUE) == 'Submit') {
            //cek apakah cart ada isinya
            atau kosong
            if (!$this->cart->contents()) {
                $this->session-
>set_flashdata('alert', 'Anda belum
memilih barang...');

                redirect('tambah_penjualan
', 'refresh');
            }
            //validasi input data
            $this->form_validation-
>set_rules(
                'tanggal',
                'Tanggal Pembelian',
                'required|callback_checkDa
teFormat',
                array(
                    'required' => '{field}
wajib diisi',
                    'checkDateFormat' =>
'{field} tidak valid'
                )
            );

            $this->form_validation-
>set_rules(
                'pembeli',
                'Nama Pembeli',
                "required|min_length[3]|m
ax_length[30]|regex_match[/^[A-Z
a-z.']+$/]",
                array(
                    'required' => '{field}
wajib diisi',

```

```

        'min_length' => '{field}
minimal 3 karakter',
        'max_length' => '{field}
maksimal 30 karakter',
        'regex_match' =>
'{field} hanya boleh huruf, titik dan
kutip satu (\')
    )
);

if ($this->form_validation-
>run() == TRUE) {

    $id = 'ID' . time();
    $tgl = date('Y-m-d',
    strtotime(str_replace('/', '-', $this-
>security->xss_clean($this->input-
>post('tanggal', TRUE))));
    $pembeli = $this-
>security->xss_clean($this->input-
>post('pembeli', TRUE));
    $user = $this->session-
>userdata('UserID');

    $data_penjualan = [
        'id_penjualan' => $id,
        'tgl_penjualan' => $tgl,
        'nama_pembeli' =>
$pembeli,
        'id_user' => $user
    ];
    //baca cart dan
memasukkannya dalam array untuk
disimpan
    $cart = array();

    foreach ($this->cart-
>contents() as $c) {
        $item = [
            'id_penjualan' => $id,
            'id_barang' => $c['id'],
            'qty' => $c['qty'],
            'harga' => $c['price']
        ];

        //push ke array cart

```

```

        array_push($cart,
        $item);
    }
    //simpan data penjualan
    $simpan = $this-
>m_penjualan->save('tbl_penjualan',
    $data_penjualan);

    if ($simpan) {
        //simpan data detail
        penjualan
            $this->m_penjualan-
>multiSave('tbl_detail_penjualan',
            $cart);
            //kosongkan cart
            $this->cart->destroy();
            //buat notifikasi
            penyimpanan berhasil
            $this->session-
>set_flashdata('success', 'Data
            penjualan berhasil ditambahkan...');

            redirect('data_penjualan'
            );
        }
    }

    $data = [
        'title' => 'Tambah Data
        penjualan Barang',
        'data' => $this->m_penjualan-
>getData('tbl_barang', ['active' =>
        'Y']),
        'table' => $this->read_cart()
    ];

    $this->template-
>kasir('penjualan/form_input',
    $data);
    }

    public function
    detail_penjualan($id_penjualan =
    null)
    {

```

```

        if ($Sid_penjualan == null) {
            redirect('data_penjualan');
        }

        $this->is_login();
        //ambil data
        $getData = $this->m_penjualan-
>getDataPenjualan($this->security-
>xss_clean($id_penjualan));

        if ($getData->num_rows() < 1)
        {
            redirect('dashboard');
        }

        $data = [
            'title' => 'Detail Penjualan ' .
            $id_penjualan,
            'data' => $getData
        ];

        $this->template-
>kasir('penjualan/detail', $data);
    }

    public function hapus_penjualan()
    {
        //cek login
        $this->is_login();
        //validasi request ajax
        if ($this->input-
>is_ajax_request()) {
            //validasi
            $this->form_validation-
>set_rules(
                'id',
                'ID Penjualan',
                "required|min_length[10]",
                array(
                    'required' => '{field}
tidak valid',
                    'min_length' => 'Isi
{field} tidak valid'
                )
            );
        }
    }

```

```

        if ($this->form_validation-
>run() == TRUE) {
            //tangkap id
            $id = $this->security-
>xss_clean($this->input->post('id',
TRUE));

            $hapus = $this-
>m_penjualan-
>delete(['tbl_penjualan',
'tbl_detail_penjualan'],
['id_penjualan' => $id]);

            if ($hapus) {
                echo
                json_encode(['message' =>
'success']);
            } else {
                echo
                json_encode(['message' => 'failed']);
            }
            } else {
                echo
                json_encode(['message' => 'failed']);
            }
            } else {
                redirect('dashboard');
            }
        }

        public function edit_penjualan($id
= null)
        {
            if ($id == null) {
                redirect('data_penjualan');
            }

            $this->is_login();

            //ketika user menyimpan data
            if ($this->input->post('submit',
TRUE) == 'Submit') {
                //cek apakah user sudah
                memilih barang atau belum, jika
                belum maka munculkan pesan
                kesalahan
            }
        }
    }

```

```

        if (!$this->cart->contents()) {
            $this->session->set_flashdata('alert', 'Anda belum
memilih barang...');

            redirect('edit_penjualan/' .
$id, 'refresh');
        }
        //validasi input data tanggal
        $this->form_validation->set_rules(
            'tanggal',
            'Tanggal Pembelian',
            'required|callback_checkDateFormat',
            array(
                'required' => '{field}
wajib diisi',
                'checkDateFormat' =>
'{field} tidak valid'
            )
        );

        $this->form_validation->set_rules(
            'pembeli',
            'Nama Pembeli',
            "required|min_length[3]|max_length[30]|regex_match[/^[A-Z
a-z.']+$/]",
            array(
                'required' => '{field}
wajib diisi',
                'min_length' => '{field}
minimal 3 karakter',
                'max_length' => '{field}
maksimal 30 karakter',
                'regex_match' =>
'{field} hanya boleh huruf, titik dan
kutip satu (\')'
            )
        );

        $this->form_validation->set_rules(
            'idP',

```

```

            'ID Penjualan',
            'required|min_length[10]',
            array(
                'required' => '{field}
wajib diisi',
                'min_length' => '{field}
tidak valid'
            )
        );

        if ($this->form_validation->run() == TRUE) {

            $idP = $this->security->xss_clean($this->input->post('idP',
TRUE));
            $tgl = date('Y-m-d',
strtotime(str_replace('/', '-', $this->input->post('tanggal', TRUE))));
            $pembeli = $this->security->xss_clean($this->input->post('pembeli', TRUE));

            $data_penjualan = [
                'tgl_penjualan' => $tgl,
                'nama_pembeli' =>
$pembeli
            ];

            //baca cart dan
memasukkannya dalam array untuk
disimpan
            $cart = array();

            foreach ($this->cart->contents() as $c) {
                $item = [
                    'id_penjualan' =>
$idP,
                    'id_barang' => $c['id'],
                    'qty' => $c['qty'],
                    'harga' => $c['price']
                ];
                //hapus session
                $this->session->unset_userdata($c['id']);
            }
        }
    }
}

```

```

        //push ke array cart
        array_push($cart,
$item);
    }

    //simpan perubahan data
    penjualan
    $update = $this-
>m_penjualan-
>update('tbl_penjualan',
$data_penjualan, ['id_penjualan' =>
$idP]);

    if ($update) {
        //hapus detail penjualan
        $this->m_penjualan-
>delete('tbl_detail_penjualan',
['id_penjualan' => $idP]);
        //simpan data detail
        penjualan
        $this->m_penjualan-
>multiSave('tbl_detail_penjualan',
$cart);
        //kosongkan cart
        $this->cart->destroy();
        //buat notifikasi
        penyimpanan berhasil
        $this->session-
>set_flashdata('success', 'Data
pembelian berhasil diperbarui...');

        redirect('data_penjualan'
);
    }
}

//ambil data penjualan
$getdata = $this->m_penjualan-
>getdataPenjualan($this->security-
>xss_clean($id));

//hitung data
if ($getdata->num_rows() < 1)
{
    redirect('data_penjualan');
}

```

```

    }

    //cek apakah yang akses adalah
    admin / user yang menginputkan,
    jika bukan keduanya maka redirect
    ke halaman data pembelian
    $fdata = $getdata->row();

    if ($this->session-
>userdata('level') != 'admin' &&
$this->session->userdata('UserID')
!= $fdata->id_user) {
        redirect('data_pembelian');
    }

    //masukkan detail penjualan ke
    cart
    if (!$this->cart->contents()) {
        $dataCart = [];

        foreach ($getdata->result() as
$c) {
            $keranjang = array(
                'id' => $c-
>kode_barang,
                'qty' => $c->qty,
                'price' => $c->harga,
                'name' => $c-
>nama_barang
            );

            array_push($dataCart,
$keranjang);
        }

        $this->cart-
>insert($dataCart);
    }

    $data = [
        'title' => 'Edit Data
Penjualan',
        'fdata' => $fdata,
        'data' => $this->m_penjualan-
>getdata('tbl_barang', ['active' =>
'Y']),
    ]
}

```

```

        'table' => $this->read_cart()
    ];

    $this->template-
>kasir('penjualan/form_edit', $data);
    }

    public function
    cari_barang_penjualan()
    {
        $this->is_login();
        //cek apakah request berupa ajax
        atau bukan, jika bukan maka redirect
        ke home
        if ($this->input-
>is_ajax_request()) {
            //validasi data
            $this->form_validation-
>set_rules(
                'id',
                'Barang',
                'required|min_length[5]|ma
x_length[6]',
                array(
                    'required' => '{field}
wajib dipilih',
                    'min_length' => '{field}
tidak valid',
                    'max_length' => '{field}
tidak valid'
                )
            );

            if ($this->form_validation-
>run() == TRUE) {
                //ambil data
                $where = [
                    'kode_barang' => $this-
>security->xss_clean($this->input-
>post('id', TRUE)), 'active' => 'Y'
                ];
                $getBarang = $this-
>m_penjualan->getData('tbl_barang',
                $where);
                //cek jumlah data

```

```

        if ($getBarang-
>num_rows() == 1) {
            $barang = $getBarang-
>row();
            $stok = $barang->stok;
            //cari item di dalam cart
            foreach ($this->cart-
>contents() as $c) {
                if ($c['id'] ==
                $barang->kode_barang) {
                    $stok = $stok -
                    $c['qty'];
                }
            }

            $table = $this-
>read_cart();

            $array = ['table' =>
            $table, 'sisat' => $stok, 'status' =>
            'success'];

            echo
            json_encode($array);
        } else {
            $table = $this-
>read_cart();
            $alert = '<div
            class="alert alert-danger"
            role="alert">Data barang tidak
            ditemukan</div>';

            $array = ['table' =>
            $table, 'alert' => $alert, 'status' =>
            'failed'];

            echo
            json_encode($array);
        }
    } else {
        $table = $this->read_cart();
        $alert = '<div class="alert
        alert-danger" role="alert">Data
        barang tidak ditemukan</div>';
    }
}

```

```

        $array = ['table' => $table,
'alert' => $alert, 'status' => 'failed'];

        echo json_encode($array);
    }
    } else {
        redirect('data_penjualan');
    }
}

public function
tambah_cart_penjualan()
{
    //cek login
    $this->is_login();
    //validasi request ajax
    if ($this->input-
>is_ajax_request()) {
        //validasi data
        $this->form_validation-
>set_rules(
            'id',
            'Barang',
            'required|min_length[4]|ma
x_length[6]',
            array(
                'required' => '{field}
wajib dipilih',
                'min_length' => 'Isi
{field} tidak valid',
                'max_length' => 'Isi
{field} tidak valid',
            )
        );

        $this->form_validation-
>set_rules(
            'qty',
            'Jumlah',
            "required|min_length[1]|re
gex_match[/^[0-
9]+$]/|greater_than[0]",
            array(
                'required' => '{field}
wajib diisi',

```

```

                'min_length' => 'Isi
{field} tidak valid',
                'regex_match' =>
'"{field}" hanya boleh angka',
                'greater_than' => '{field}
harus lebih dari nol'
            )
        );

        if ($this->form_validation-
>run() == TRUE) {
            //ambil barang sesuai kode
            $where = [
                'kode_barang' => $this-
>security->xss_clean($this->input-
>post('id', TRUE)), 'active' => 'Y'
            ];

            $get_barang = $this-
>m_penjualan->getData('tbl_barang',
$where);

            if ($get_barang-
>num_rows() == 1) {
                //fetch data barang dan
                masukkan kedalam cart
                $b = $get_barang-
>row();
                $stok = $b->stok;
                //cari item di dalam cart
                foreach ($this->cart-
>contents() as $c) {
                    if ($c['id'] == $b-
>kode_barang) {
                        $stok = $stok -
                        $c['qty'];
                    }
                }

                if ($this->security-
>xss_clean($this->input->post('qty',
TRUE)) > $stok) {

                    $alert = '<div
class="alert alert-danger"

```

```

role="alert">Jumlah beli melebihi
stok yang ada</div>';

        $status = 'gagal';
    } else {
        $keranjang = array(
            'id' => $b-
>kode_barang,
            'qty' => $this-
>security->xss_clean($this->input-
>post('qty', TRUE)),
            'price' => $b-
>harga,
            'name' => $b-
>nama_barang
        );

        $this->cart-
>insert($keranjang);

        $alert = '<div
class="alert alert-success"
role="alert">Data berhasil
ditambahkan ke daftar</div>';

        $status = 'success';
    }

    $table = $this-
>read_cart();

    $sarr = array('table' =>
$table, 'alert' => $alert, 'status' =>
$status);
    } else {
        $table = $this-
>read_cart();

        $alert = '<div
class="alert alert-danger"
role="alert">Barang tidak
ditemukan</div>';

        $sarr = array('table' =>
$table, 'alert' => $alert, 'status' =>
'gagal');

```

```

    }

    echo json_encode($sarr);
} else {
    $table = $this->read_cart();

    $alert = '<div class="alert
alert-danger" role="alert">' .
validation_errors('<p class="mb-0
mt-0"><i class="fa fa-caret-
right"></i> ', '</p>') . '</div>';

    $sarr = array('table' =>
$table, 'alert' => $alert, 'status' =>
'gagal');

    echo json_encode($sarr);
}
} else {
    redirect('dashboard');
}
}

public function
hapus_cart_penjualan()
{
    //cek login
    $this->is_login();
    //validasi request ajax
    if ($this->input-
>is_ajax_request()) {
        //validasi
        $this->form_validation-
>set_rules(
            'rowid',
            'Row ID',
            "required|min_length[10]",
            array(
                'required' => '{field}
tidak valid',
                'min_length' => 'Isi
{field} tidak valid'
            )
        );
    }
}

```

```

        if ($this->form_validation-
>run() == TRUE) {
            //tangkap rowid
            $rowid = $this->security-
>xss_clean($this->input-
>post('rowid', TRUE));
            //ambil data barang dalam
            cart
            $get_item = $this->cart-
>get_item($rowid);
            //hapus session
            $this->session-
>unset_userdata($get_item['id']);
            //hapus cart
            $this->cart-
>remove($rowid);

            $alert = '<div class="alert
alert-success" role="alert">Data
barang berhasil dihapus</div>';

            $message = 'success';
        } else {
            $alert = '<div class="alert
alert-danger" role="alert">' .
validation_errors('<p class="mb-0
mt-0"><i class="fa fa-caret-
right"></i> ', '</p>') . '</div>';

            $message = 'failed';
        }

        $table = $this->read_cart();

        $arr = array('table' => $table,
'alert' => $alert, 'message' =>
$message);

        echo json_encode($arr);
    } else {
        redirect('dashboard');
    }
}

public function
get_item_penjualan($uri = null)

```

```

    {
        $this->is_login();

        if ($this->input-
>is_ajax_request()) {
            //tangkap rowid
            $rowid = $this->security-
>xss_clean($this->input-
>post('rowid', TRUE));

            $get_item = $this->cart-
>get_item($rowid);

            if ($get_item) {
                //cek item dalam database
                untuk mengambil stok terakhir dan
                ditambah stok barang yang akan
                dijual

                $getBarang = $this-
>m_penjualan->getData('tbl_barang',
['kode_barang' => $get_item['id']]);

                if ($getBarang-
>num_rows() != 1) {
                    $arr = [
                        'barang' => ",
                        'qty' => ",
                        'stok' => ",
                        'rowid' => ",
                        'table' => $this-
>read_cart(),
                        'status' => 'false'
                    ];
                } else {
                    $b = $getBarang-
>row();

                    if (strtolower($uri) ==
'edit_penjualan') {

                        if ($this->session-
>userdata($get_item['id'])) {
                            $stok = $this-
>session->userdata($get_item['id']);
                        } else {

```

```

        //masukkan stok ke
session
        $stoknya =
[$get_item['id'] => ($b->stok +
$get_item['qty'])];
        $this->session-
>set_userdata($stoknya);

        $stok = ($b->stok +
$get_item['qty']);
    }
    } else {
        $stok = $b->stok;
    }
    $arr = [
        'barang' =>
$get_item['id'],
        'qty' =>
$get_item['qty'],
        'stok' => $stok,
        'rowid' => '<input
type="hidden" id="rowid" value="' .
$get_item['rowid'] . '" /><input
type="hidden" id="lastQty" value="'
. $stok . '">',
        'table' => $this-
>read_cart(),
        'status' => 'true'
    ];
    }
    } else {
        $arr = [
            'barang' => "",
            'qty' => "",
            'stok' => "",
            'rowid' => "",
            'table' => $this-
>read_cart(),
            'status' => 'false'
        ];
    }

    echo json_encode($arr);
    } else {
        redirect('dashboard');
    }
}

```

```

    }

    public function
update_cart_penjualan()
    {
        //cek login
        $this->is_login();
        //validasi request ajax
        if ($this->input-
>is_ajax_request()) {
            //validasi data
            $this->form_validation-
>set_rules(
                'id',
                'Barang',
                'required|min_length[4]|ma
x_length[6]',
                array(
                    'required' => '{field}
wajib dipilih',
                    'min_length' => 'Isi
{field} tidak valid',
                    'max_length' => 'Isi
{field} tidak valid',
                )
            );

            $this->form_validation-
>set_rules(
                'jumlah',
                'Jumlah',
                "required|min_length[1]|re
gex_match[/^[0-
9]+$/]|greater_than[0]",
                array(
                    'required' => '{field}
wajib diisi',
                    'min_length' => 'Isi
{field} tidak valid',
                    'regex_match' =>
'{field} hanya boleh angka',
                    'greater_than' => '{field}
harus lebih dari nol'
                )
            );
        }
    }
}

```

```

        $this->form_validation-
>set_rules(
            'rowid',
            'Row ID',
            "required|min_length[10]",
            array(
                'required' => '{field}
tidak valid',
                'min_length' => 'Isi
{field} tidak valid'
            )
        );

        if ($this->form_validation-
>run() == TRUE) {

            //ambil barang sesuai kode
            $where = [
                'kode_barang' => $this-
>security->xss_clean($this->input-
>post('id', TRUE))
            ];

            $get_barang = $this-
>m_penjualan->getData('tbl_barang',
            $where);

            if ($get_barang-
>num_rows() == 1) {
                //fetch data barang dan
                masukkan kedalam cart
                $b = $get_barang-
>row();
                $stok = $b->stok;
                //cari item di dalam cart
                foreach ($this->cart-
>contents() as $c) {
                    if ($c['id'] == $b-
>kode_barang) {
                        $stok = $stok +
                        $c['qty'];
                    }
                }
            }

```

```

        if ($this->security-
>xss_clean($this->input->post('qty',
        TRUE)) > $stok) {

            $alert = '<div
            class="alert alert-danger"
            role="alert">Jumlah beli melebihi
            stok yang ada</div>';

            $status = 'gagal';
        } else {
            $skanjang = array(
                'rowid' => $this-
>security->xss_clean($this->input-
>post('rowid', TRUE)),
                'qty' => $this-
>security->xss_clean($this->input-
>post('jumlah', TRUE))
            );

            $this->cart-
>update($skanjang);

            $stable = $this-
>read_cart();

            $alert = '<div
            class="alert alert-success"
            role="alert">Data barang berhasil
            diubah</div>';

            $status = 'success';
        }

            $stable = $this-
>read_cart();

            $sarr = array('table' =>
            $stable, 'alert' => $alert, 'status' =>
            $status);
        } else {
            $stable = $this-
>read_cart();

            $alert = '<div
            class="alert alert-danger"

```

```

role="alert">Barang tidak
ditemukan</div>';

        $arr = array('table' =>
$table, 'alert' => $alert, 'status' =>
'gagal');
    }

    echo json_encode($arr);
} else {
    $table = $this->read_cart();

    $alert = '<div class="alert
alert-danger" role="alert">' .
validation_errors('<p class="mb-0
mt-0"><i class="fa fa-caret-
right"></i> ', '</p>') . '</div>';

    $arr = array('table' =>
$table, 'alert' => $alert, 'status' =>
'gagal');

    echo json_encode($arr);
}
} else {
    redirect('dashboard');
}

function checkDateFormat($date)
{
    if (preg_match("/^(0[1-9]|[1-
2][0-9]|3[0-1])\.(0[1-9]|1[0-2])\.[0-
9]{4}$/", $date)) {
        if (checkdate(substr($date, 3,
2), substr($date, 0, 2), substr($date,
6, 4))) {
            return true;
        } else {
            return false;
        }
    } else {
        return false;
    }
}
}

```

```

public function ajax_penjualan()
{
    $this->is_login();
    //cek apakah request berupa ajax
atau bukan, jika bukan maka redirect
ke home
    if ($this->input-
>is_ajax_request()) {
        //ambil list data
        $list = $this->m_penjualan-
>get_datatables();
        //siapkan variabel array
        $data = array();
        $no = $_POST['start'];

        foreach ($list as $i) {

            $button = "<button type='button'
class='btn btn-danger btn-
sm' onclick='hapus_penjualan(\" . $i-
>id_penjualan .
'\')">Hapus</button>";

            $button .= '<a href="' .
site_url('edit_penjualan/' . $i-
>id_penjualan) . '" class="btn btn-
warning btn-sm text-
white">Edit</a>';

            $button .= '<button type="button"
class="btn btn-danger btn-
sm" onclick="hapus_penjualan(\' . $i-
>id_penjualan .
'\')">Hapus</button>';

            $button .= '</button>';

            $no++;
            $row = array();
            $row[] = $no;
            $row[] = $i->id_penjualan;
            $row[] = $this-
>tanggal_indo($i->tgl_penjualan);
            $row[] = $i-
>nama_pembeli;
            $row[] = $i->jumlah;
        }
    }
}

```

```

        $row[] = '<span class="pr-3">' . number_format($i->total, 0, ',', '.') . ',</span>';
        $row[] = $i->fullname;
        $row[] = '<a href="' . site_url('data_penjualan/' . $i->id_penjualan) . '" class="btn btn-sm btn-success">Detail</a>' . $button;

        $data[] = $row;
    }

    $output = array(
        "draw" =>
        $_POST['draw'],
        "recordsTotal" => $this->m_penjualan->count_all(),
        "recordsFiltered" => $this->m_penjualan->count_filtered(),
        "data" => $data
    );
    //output to json format
    echo json_encode($output);
} else {
    redirect('dashboard');
}

private function read_cart()
{
    if ($this->cart->contents()) {

        $table = "";
        $i = 1;
        foreach ($this->cart->contents() as $c) {
            $table .= '<tr><td>' . $i++ .
            '</td>';
            $table .= '<td>' . $c['name']
            . '</td>';
            $table .= '<td class="text-center">' . $c['qty'] . '</td>';
            $table .= '<td class="text-right">' . number_format($c['price'], 0, ',', '.') . '</td>';

```

```

        $table .= '<td class="text-right">' .
        number_format($c['subtotal'], 0, ',', '.') . '</td>';
        $table .= '<td class="text-center">
            <button
            type="button" class="btn btn-warning btn-sm text-white"
            onclick="get_item_penjualan(\' . $c['rowid'] . \')">Edit</button>
            <button
            type="button" class="btn btn-danger btn-sm text-white"
            onclick="hapus_item_penjualan(\' . $c['rowid'] . \')">Hapus</button>
            </td></tr>';
        }
        $table .= '<tr>
            <td scope="col"
            colspan="4" class="text-center"><b><i>Total
            Harga</i></b></td>
            <td scope="col"
            class="text-right"><b>' .
            number_format($this->cart->total(), 0, ',', '.') . '</b></td>
            <td
            scope="col"></td>
            </tr>';
        } else {
            $table = '<tr>
                <td scope="col"
                colspan="6" class="text-center"><i>Belum ada data</i></td>
                </tr>';
        }

        return $table;
    }

    private function tanggal_indo($tgl)
    {
        $bulan = [1 => 'Januari',
        'Februari', 'Maret', 'April', 'Mei',

```

```
'Juni', 'Juli', 'Agustus', 'September',  
'Oktober', 'November', 'Desember'];
```

```
    $exp = explode('-', date('d-m-  
Y', strtotime($tgl)));
```

```
    return $exp[0] . ' ' . $bulan[(int)  
$exp[1]] . ' ' . $exp[2];  
}
```

```
private function is_login()  
{  
    if (!$this->session-  
>userdata('UserID')) {  
        redirect('dashboard');  
    }  
}  
}
```



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI

POLITEKNIK NEGERI SRIWIJAYA
JURUSAN MANAJEMEN INFORMATIKA

Jalan Srijaya Negara Bukit Besar - Palembang 30139 Telepon (0711) 353414
Laman : <http://polsri.ac.id>. Pos El : info@polsri.ac.id

KESEPAKATAN BIMBINGAN LAPORAN AKHIR (LA)

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Raden Ayu Yasmin Nadira
NIM : 062130801809
Jurusan : Manajemen Informatika
Program Studi : DIII Manajemen Informatika

Pihak Kedua

Nama : Dr. Indri Ariyanti, SE., M.Si.
NIP : 197306032008012008
Jurusan : Manajemen Informatika
Program Studi : DIII Manajemen Informatika

Pada hari ini Selasa tanggal 19 Maret 2024 telah sepakat untuk melakukan konsultasi bimbingan Laporan Akhir. Konsultasi Bimbingan sekurang-kurangnya 1 (Satu) kali dalam satu minggu. Pelaksanaan bimbingan pada setiap hari Kamis pukul 11.00 WIB yang bertempat di Politeknik Negeri Sriwijaya.

Demikianlah kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Laporan Akhir.

Pihak Pertama,

Raden Ayu Yasmin Nadira
NIM 062130801809

Palembang, 19 Maret 2024

Pihak Kedua,

Dr. Indri Ariyanti, SE., M.Si.
NIP 197306032008012008

Mengetahui,
a.n Ketua Jurusan Manajemen Informatika
Sekretaris Jurusan Manajemen Informatika

Meivi Kusnandar, S.Kom., M.Kom.
NIP 197407052002121014



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI

POLITEKNIK NEGERI SRIWIJAYA
JURUSAN MANAJEMEN INFORMATIKA

Jalan Srijaya Negara Bukit Besar - Palembang 30139 Telepon (0711) 353414
Laman : <http://polsri.ac.id>. Pos El : info@polsri.ac.id

KESEPAKATAN BIMBINGAN LAPORAN AKHIR (LA)

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Raden Ayu Yasmin Nadira
NIM : 062130801809
Jurusan : Manajemen Informatika
Program Studi : DIII Manajemen Informatika

Pihak Kedua

Nama : Henny Madora, S.Kom., M.M.
NIP : 197709272005012001
Jurusan : Manajemen Informatika
Program Studi : DIII Manajemen Informatika

Pada hari ini Jum'at tanggal 22 Maret 2024 telah sepakat untuk melakukan konsultasi bimbingan Laporan Akhir. Konsultasi Bimbingan sekurang-kurangnya 1 (Satu) kali dalam satu minggu. Pelaksanaan bimbingan pada setiap hari Senin yang bertempat di Politeknik Negeri Sriwijaya.

Demikianlah kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Laporan Akhir.

Pihak Pertama,

Raden Ayu Yasmin Nadira
NIM 062130801809

Palembang, 22 Maret 2024

Pihak Kedua,

Henny Madora, S.Kom., M.M.
NIP 197709272005012001

Mengetahui,
Ketua Jurusan Manajemen Informatika

Dr. Indri Ariyanti, SE., M.Si.
NIP 197306032008012008





KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI

POLITEKNIK NEGERI SRIWIJAYA
JURUSAN MANAJEMEN INFORMATIKA

Jalan Srijaya Negara Bukit Besar - Palembang 30139 Telepon (0711) 353414
Laman : <http://polsri.ac.id> Pos El : info@polsri.ac.id

PENGAJUAN JUDUL LAPORAN AKHIR

Kepada Yth,
Dosen Pembimbing I
Di
Politeknik Negeri Sriwijaya

Dengan Hormat,

Dalam rangka memenuhi syarat penyusunan Laporan Akhir, maka saya:

Nama : Raden Ayu Yasmin Nadira

NIM : 062130801809

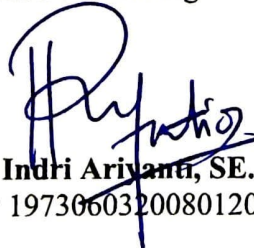
Jurusan / Program Studi : Manajemen Informatika / DIII Manajemen Informatika

Bermaksud mengajukan judul Laporan Akhir. Adapun judul Laporan Akhir yang saya ajukan yaitu:

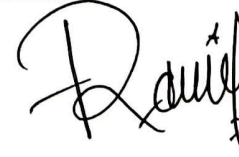
1. Aplikasi *Database Human Resource* (HR) Dan Persediaan Penjualan Stok
ace Barang Pada CV Kerabat Indo Jaya Berbasis *Website*
2. Sistem Informasi Pengaduan Pelanggan Pada CV Kerabat Indo Jaya Berbasis
Website
3. Aplikasi Pengolahan Data Kepegawaian Pada CV Kerabat Indo Jaya Berbasis
Website

Demikian surat pengajuan judul Laporan Akhir ini dibuat, sekiranya judul diatas dapat disetujui. Atas persetujuan Ibu kami ucapkan terima kasih.

Menyetujui,
Dosen Pembimbing I


Dr. Indri Ariyanti, SE., M.Si.
NIP 197306032008012008

Palembang, 19 Maret 2024
Hormat Saya,
Mahasiswa


Raden Ayu Yasmin Nadira
NIM 062130801809



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI

POLITEKNIK NEGERI SRIWIJAYA
JURUSAN MANAJEMEN INFORMATIKA

Jalan Srijaya Negara Bukit Besar - Palembang 30139 Telepon (0711) 353414
Laman : <http://polsri.ac.id>, Pos El : info@polsri.ac.id

PENGAJUAN JUDUL LAPORAN AKHIR

Kepada Yth,
Dosen Pembimbing II
Di
Politeknik Negeri Sriwijaya

Dengan Hormat,

Dalam rangka memenuhi syarat penyusunan Laporan Akhir, maka saya:

Nama : Raden Ayu Yasmin Nadira

NIM : 062130801809

Jurusan / Program Studi : Manajemen Informatika / DIII Manajemen Informatika

Bermaksud mengajukan judul Laporan Akhir. Adapun judul Laporan Akhir yang saya ajukan yaitu:

1. Aplikasi *Database Human Resource* (HR) Dan Persediaan Penjualan Stok Barang Pada CV Kerabat Indo Jaya Berbasis *Website*
2. Sistem Informasi Pengaduan Pelanggan Pada CV Kerabat Indo Jaya Berbasis *Website*
3. Aplikasi Pengolahan Data Kepegawaian Pada CV Kerabat Indo Jaya Berbasis *Website*

Demikian surat pengajuan judul Laporan Akhir ini dibuat, sekiranya judul diatas dapat disetujui. Atas persetujuan Ibu kami ucapkan terima kasih.

Menyetujui,
Dosen Pembimbing II

Henny Madora, S.Kom., M.M.
NIP 197709272005012001

Palembang, 22 Maret 2024
Hormat Saya,
Mahasiswa

Raden Ayu Yasmin Nadira
NIM 062130801809





KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI

POLITEKNIK NEGERI SRIWIJAYA
JURUSAN MANAJEMEN INFORMATIKA

Jalan Srijaya Negara Bukit Besar - Palembang 30139 Telepon (0711) 353414
Laman : <http://polsri.ac.id> Pos El : info@polsri.ac.id

LEMBAR PENGESAHAN JUDUL LAPORAN AKHIR (LA)

Nama : Raden Ayu Yasmin Nadira
NIM : 062130801809
Jurusan / Program Studi : Manajemen Informatika/ DIII Manajemen Informatika
Judul Laporan Akhir : Aplikasi *Database Human Resource* (HR) Dan Persediaan
Penjualan Stok Barang Pada CV Kerabat Indo Jaya
Berbasis *Website*

Palembang, Maret 2024

Tim Pembimbing:

Pembimbing I

Dr. Indri Ariyanti, SE., M.Si.
NIP 197306032008012008

Pembimbing II

Henny Madora, S.Kom., M.M.
NIP 197709272005012001

Mengetahui,
a.n Ketua Jurusan Manajemen Informatika
Sekretaris Jurusan Manajemen Informatika

Meivi Kusnandar, S.Kom., M.Kom.
NIP 197407052002121014



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI

POLITEKNIK NEGERI SRIWIJAYA
JURUSAN MANAJEMEN INFORMATIKA

Jalan Sriwijaya Negara Bukit Besar - Palembang 30139 Telepon (0711) 353414
Laman : <http://polsri.ac.id> Pos El : info@polsri.ac.id

321 22/3/24

PERMOHONAN SURAT PENGANTAR PENGAMBILAN DATA

Hal : Permohonan Surat Pengantar
Pengambilan Data

Palembang, 19 Maret 2024

Kepada Yth.
Wakil Direktur I
Politeknik Negeri Sriwijaya
di
Tempat

Dengan Hormat,

Sehubungan dengan adanya penulisan Laporan Akhir yang merupakan salah satu mata kuliah Semester VI dan sebagai syarat untuk menyelesaikan pendidikan Diploma III Jurusan Manajemen Informatika Politeknik Negeri Sriwijaya, untuk itu saya mohon kepada Bapak agar dibuatkan surat pengantar yang ditujukan kepada :

Yth. Kepala CV. Kerabat Indo Jaya

Jl. Angkatan 66 Ujung No. 1503, Sukajaya, Kec. Kemuning, Kota Palembang, Sumatera Selatan 30127

Untuk pengambilan data di CV Kerabat Indo Jaya Palembang dalam pembuatan aplikasi yang berjudul "**Aplikasi Database Human Resource (HR) Dan Persediaan Penjualan Stok Barang Pada CV Kerabat Indo Jaya Berbasis Website**".

Pengambilan data ini akan saya mulai pada tanggal 01 April 2024 s.d. 31 Juli 2024. Saya mahasiswa yang akan melakukan pengambilan data adalah :

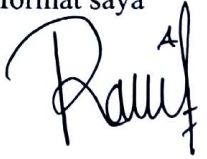
No	NAMA	NIM	KELAS	JURUSAN
1.	Raden Ayu Yasmin Nadira	062130801809	6IA	Manajemen Informatika

Demikianlah surat permohonan ini saya buat, atas bantuan dan kerjasama Bapak saya ucapkan terima kasih.

Mengetahui,
Ketua Jurusan Manajemen Informatika


Dr. Indri Ariyanti, SE., M.Si
NIP 197306032008012008

Hormat saya


Raden Ayu Yasmin Nadira
NIM 062130801809





KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI SRIWIJAYA

Jalan Srijaya Negara Bukit Besar – Palembang 30139

Telepon 0711-353414 Faksimili 0711-355918

Laman : <http://polsri.ac.id>, Pos El : info@polsri.ac.id

Nomor : 3011/PL6.3.1/SP/2024

Hal : Izin Pengambilan Data

22 Maret 2024

Yth. Kepala
CV. Kerabat Indo Jaya
Jalan Angkatan 66 Ujung No. 1503
Kecamatan Kemuning
Palembang

Dengan hormat,

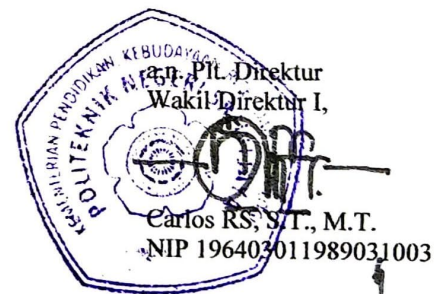
Sesuai dengan kurikulum Jurusan Manajemen Informatika pada Politeknik Negeri Sriwijaya, Laporan Akhir merupakan mata kuliah wajib pada semester VI (enam). Untuk itu mahasiswa kami memerlukan data untuk penyusunan Laporan Akhir tersebut.

Sehubungan dengan hal tersebut di atas, kami mohon Bapak/Ibu dapat memberikan izin dan membantu mahasiswa kami ini untuk mengumpulkan data.

Mahasiswa kami yang akan mengumpulkan data tersebut adalah :

No	Nama	NPM	Kelas	Jurusan/ Program Studi
1	Raden Ayu Yasmin Nadira	0621 3080 1809	6 IA	Manajemen Informatika

Atas perhatian dan bantuannya diucapkan terima kasih.



Tembusan:

1. Plt. Direktur
2. Ketua Jurusan Manajemen Informatika
3. Yang bersangkutan
4. Arsip

F:MS.Word/D/TM





CV. KERABAT INDO JAYA

Jl. Angkatan 66 No. 1503 Kel. Pipa Reja Kec. Kemiuning Kota Palembang
0711 – 5614256 | 0821 8095 6030 | dezainla.official@gmail.com

www.dezainla.com

Palembang, 04 Mei 2024

Nomor : 11/022/KIJ/V/2024
Perihal : Konfirmasi Izin Pengambilan Data

Kepada Yth,
Politeknik Negeri Sriwijaya
Di -

Tempat

Dengan Hormat,

Berdasarkan Surat Nomor : 3011/PL6.3.1/SP/2024 tanggal 22 Maret 2024. Perihal Permohonan Izin Pengambilan Data Mahasiswa Jurusan Manajemen Informatika Politeknik Sriwijaya yang bernama :

Nama : Raden Ayu Yasmin Nadira
NPM : 062130801809

Bersama ini saya menyampaikan bahwa kami mengkonfirmasi memberikan Izin Pengambilan Data untuk Mahasiswa tersebut di perusahaan CV. Kerabat Indo Jaya.

Demikianlah hal ini saya sampaikan, atas perhatian dan kerjasamanya sama ucapkan Terima Kasih.

Hormat Kami,
CV. Kerabat Indo Jaya

M Rizki Saputra, S.SI



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI

POLITEKNIK NEGERI SRIWIJAYA
JURUSAN MANAJEMEN INFORMATIKA

Jalan Srijaya Negara Bukit Besar - Palembang 30139 Telepon (0711) 353414
Laman : <http://polsri.ac.id> Pos El : info@polsri.ac.id

LEMBAR BIMBINGAN LAPORAN AKHIR

Lembar : 1

Nama : Raden Ayu Yasmin Nadira
NIM : 062130801809
Jurusan / Program Studi : Manajemen Informatika / DIII Manajemen Informatika
Judul Laporan Akhir : Aplikasi *Database Human Resource* (HR) Dan Persediaan
Penjualan Stok Barang Pada CV Kerabat Indo Jaya Berbasis
Website
Pembimbing I : Dr. Indri Ariyanti, SE., M.Si.

No.	Tanggal	Uraian Bimbingan	Tanda Tangan Pembimbing
1.	19 / 03 / 2024	Pengajuan Judul	
2.	19 / 03 / 2024.	ACC Judul	
3.	22 / 03 / 2024	Pengajuan BAB I	
4.	22 / 03 / 2024	Revisi BAB I	
5.	22 / 03 / 2024	ACC BAB I	
6.	22 / 03 / 2024	Pengajuan BAB II	
7.	22 / 03 / 2024	Revisi BAB II	
8.	22 / 03 / 2024	ACC BAB II	



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI

POLITEKNIK NEGERI SRIWIJAYA
JURUSAN MANAJEMEN INFORMATIKA

Jalan Srijaya Negara Bukit Besar - Palembang 30139 Telepon (0711) 353414
Laman : <http://polsri.ac.id>. Pos El : info@polsri.ac.id

Lembar : 2

9.	28/03/2024	Pengajuan BAB II	J
10.	28/03/2024	Revisi BAB II	J
11.	01/04/2024	ACC BAB II	J
12.	07/06/2024	Pengajuan BAB I	J
13.	11/06/2024	ACC BAB IV	J
14.	12/06/2024	Pengajuan BAB V	J
15.	14/06/2024	Revisi BAB V	J
16.	21/06/2024	ACC BAB V	J
17.			
18.			
19.			

a.n Ketua Jurusan Manajemen Informatika
Sekretaris Jurusan Manajemen Informatika

Meivi Kusnandar, S.Kom., M.Kom.
NIP 197407052002121014

Palembang,
Dosen Pembimbing I

2024

Dr. Indri Ariyanti, S.E., M.Si.
NIP 197306032008012008





KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI

POLITEKNIK NEGERI SRIWIJAYA
JURUSAN MANAJEMEN INFORMATIKA

Jalan Srijaya Negara Bukit Besar - Palembang 30139 Telepon (0711) 353414
Laman : <http://polsri.ac.id>, Pos El : info@polsri.ac.id

LEMBAR BIMBINGAN LAPORAN AKHIR

Lembar : 1

Nama : Raden Ayu Yasmin Nadira
NIM : 062130801809
Jurusan / Program Studi : Manajemen Informatika / DIII Manajemen Informatika
Judul Laporan Akhir : Aplikasi *Database Human Resource* (HR) Dan Persediaan
Penjualan Stok Barang Pada CV Kerabat Indo Jaya Berbasis
Website
Pembimbing II : Henny Madora, S.Kom., M.M.

No.	Tanggal	Uraian Bimbingan	Tanda Tangan Pembimbing
1.	20-03-2024	Acc awal	
2.	25-03-2024	Revisi bab I	
3.	1 Apr 2024	Revisi bab I	
4.	3 April 2024	Revisi bab I	
5.	22 April 2024	Acc bab I	
6.	29 April 2024	Revisi bab I	
7.	6 Mei 2024	Acc bab I + revisi bab II	
8.	13 Mei 2024	Acc bab II	



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI

POLITEKNIK NEGERI SRIWIJAYA
JURUSAN MANAJEMEN INFORMATIKA

Jalan Srijaya Negara Bukit Besar - Palembang 30139 Telepon (0711) 353414
Laman : <http://polsri.ac.id>. Pos El : info@polsri.ac.id

Lembar : 2

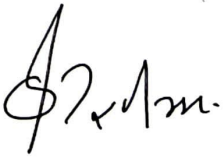
9.	3 Juni 2024	Revisi bab IV	§
10.	8. Juni 2024	Acc bab IV + Revisi bab IV	§
11.	8 Juni 2024	Program	
12.	10/ Juni 2024	Acc bab IV	§
13.			
14.			
15.			
16.			
17.			
18.			
19.			
20.			
21.			
22.			

Ketua Jurusan Manajemen Informatika

Palembang,
Dosen Pembimbing II

2024


Dr. Indri Ariyanti, S.E., M.Si.
NIP 197306032008012008


Henny Madora, S.Kom., M.M.
NIP 197709272005012001

Catatan :

Ketua Jurusan harus memeriksa jumlah pelaksanaan bimbingan sesuai yang dipersyaratkan dalam pedoman Laporan Akhir sebelum menandatangani lembar bimbingan ini.

Lembar pembimbingan Laporan Akhir ini harus dilampirkan dalam Laporan Akhir





KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI

POLITEKNIK NEGERI SRIWIJAYA
JURUSAN MANAJEMEN INFORMATIKA

Jalan Srijaya Negara Bukit Besar - Palembang 30139 Telepon (0711) 353414
Laman : <http://polsri.ac.id> Pos El : info@polsri.ac.id

REKOMENDASI UJIAN LAPORAN AKHIR

Pembimbing Laporan Akhir memberikan rekomendasi kepada,

Nama : Raden Ayu Yasmin Nadira

NIM : 062130801809

Jurusan / Program Studi : Manajemen Informatika / DIII Manajemen Informatika

Judul Laporan Akhir : Aplikasi *Database Human Resource* (HR) dan Persediaan
Penjualan Stok Barang Pada CV Kerabat Indo Jaya Berbasis
Website

Mahasiswa tersebut telah memenuhi persyaratan dan dapat mengikuti Ujian Laporan Akhir
(LA) pada Tahun Akademik 2023/2024.

Palembang, Juni 2024

Tim Pembimbing:

Pembimbing I

Dr. Indri Ariyanti, SE., M.Si.
NIP 197306032008012008

Pembimbing II

Henny Madora, S.Kom., M.M.
NIP 197709272005012001

Mengetahui,
a.n Ketua Jurusan Manajemen Informatika
Sekretaris Jurusan Manajemen Informatika

Meivi Kusnandar, S.Kom., M.Kom.
NIP 197407052002121014





KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI

POLITEKNIK NEGERI SRIWIJAYA

Jalan Srijaya Negara Bukit Besar - Palembang 30139 Telepon (0711) 353414
Laman : <http://polsri.ac.id>, Pos El : info@polsri.ac.id

REVISI UJIAN LAPORAN AKHIR (LA)

Ruang : Ruang 2
Dosen Penguji : Meivi Kusnandar, S.Kom., M.Kom.
Nama Mahasiswa : Raden Ayu Yasmin Nadira (062130801809)
Jurusan/Program Studi : Manajemen Informatika / DIII Manajemen Informatika
Judul Laporan Akhir : Aplikasi *Database Human Resource* (HR) dan Persediaan
Penjualan Stok Barang Pada CV Kerabat Indo Jata Berbasis
Website

No	Uraian Revisi	Paraf

Palembang, 15 Juli 2024
Dosen Penguji,

(.....)





KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI

POLITEKNIK NEGERI SRIWIJAYA

Jalan Srijaya Negara Bukit Besar - Palembang 30139 Telepon (0711) 353414
Laman : <http://polsri.ac.id>, Pos El : info@polsri.ac.id

REVISI UJIAN LAPORAN AKHIR (LA)

Ruang : Ruang 2
Dosen Penguji : Surahmat, M.Kom.
Nama Mahasiswa : Raden Ayu Yasmin Nadira (062130801809)
Jurusan/Program Studi : Manajemen Informatika / DIII Manajemen Informatika
Judul Laporan Akhir : Aplikasi *Database Human Resource* (HR) dan Persediaan
Penjualan Stok Barang Pada CV Kerabat Indo Jata Berbasis
Website

No	Uraian Revisi	Paraf
1.	Aplikasi disesuaikan.	
2.	pendaftaran. diperbaiki	
3.	Sistem yg berjalan - BAB 3 diperbaiki	
4.	perbaiki BAB 4	

Palembang, 15 Juli 2024
Dosen Penguji,


(.....
Surahmat, M. Kom.)





KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI

POLITEKNIK NEGERI SRIWIJAYA

Jalan Srijaya Negara Bukit Besar - Palembang 30139 Telepon (0711) 353414
Laman : <http://polsri.ac.id>, Pos El : info@polsri.ac.id

REVISI UJIAN LAPORAN AKHIR (LA)

Ruang : Ruang 2
Dosen Penguji : Ida Wahyuningrum, S.E., M.Si.
Nama Mahasiswa : Raden Ayu Yasmin Nadira (062130801809)
Jurusan/Program Studi : Manajemen Informatika / DIII Manajemen Informatika
Judul Laporan Akhir : Aplikasi *Database Human Resource* (HR) dan Persediaan
Penjualan Stok Barang Pada CV Kerabat Indo Jata Berbasis
Website

No	Uraian Revisi	Paraf

Palembang, 15 Juli 2024
Dosen Penguji,


(.....Ida. Wahyuningrum.....)





KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI

POLITEKNIK NEGERI SRIWIJAYA

Jalan Srijaya Negara Bukit Besar - Palembang 30139 Telepon (0711) 353414

Laman : <http://polsri.ac.id> Pos El : info@polsri.ac.id

LEMBAR PELAKSANAAN REVISI LAPORAN AKHIR

Mahasiswa berikut,

Nama : Raden Ayu Yasmin Nadira
 NPM : 062130801809
 Jurusan/Program Studi : Manajemen Informatika/D-III Manajemen Informatika
 Judul Laporan Akhir : Aplikasi *Database Human Resource* (HR) dan Persediaan Penjualan Stok Barang Pada CV Kerabat Indo Jaya Berbasis *Website*

Telah melaksanakan revisi terhadap Laporan Akhir yang diujikan pada hari Senin tanggal 15 bulan Juli tahun 2024. Pelaksanaan revisi terhadap Laporan Akhir tersebut telah disetujui oleh Dosen Penguji yang memberikan revisi :

No.	Komentar	Nama Dosen Penguji	Tanggal	Tanda Tangan
1	- Perbaiki <i>Class Diagram</i>	Meivi Kusnandar, S.Kom., M.Kom.	20/7/2024	
2	- Aplikasi sesuaikan - Pendahuluan di perbaiki - Sistem yang berjalan pada bab 3 - Perbaiki bab 4	Surahmat, S.Kom., M.Kom.	30/7-2024	
3	-	Ida Wahyuningrum, S.E., M.Si.	31/7/2024	

Palembang, Juli 2024

Ketua Penguji,

Meivi Kusnandar, S.Kom., M.Kom.
 NIP. 197407052002121014