# L

# A

# M

# P

# I

# R

# A

# N

**Lampiran 1 : Surat Pengantar Penelitian**

**UNIVERSITAS MUHAMMADIYAH PAREPARE**
**LEMBAGA PENELITIAN DAN PENGABDIAN KEPADA MASYARAKAT**
**(RESEARCH INSTITUTE AND COMMUNITY SERVICES)**
Alamat : Gedung F3.19 Kampus II UMPAR, Jl. Jend. Ahmad Yani KM. 6 Kota Parepare, Kode Pos 91113, e-mail : lppm@umpar.ac.id

بسم الله الرحمن الرحيم

Nomor : 0348/LPPM/II.3.AU/IP/2024
Lampiran : -
Perihal : **Permohonan Izin Penelitian**

Kepada Yth
**KEPALA DINAS PENANAMAN MODAL DAN PTSP PROV. SULSEL** di-

Makassar

*Assalamu Alaikum Warahmatullahi Wabarakatuh*

Ketua Lembaga Penelitian dan Pengabdian kepada Masyarakat Universitas Muhammadiyah Parepare, menerangkan bahwa:

| | |
|---|---|
| **Nama** | **: Dian Indriyani** |
| **NIM** | **: 220 280 059** |
| **Fakultas/Prodi** | **: Teknik/Teknik Informatika** |

Adalah Mahasiswa Universitas Muhammadiyah Parepare yang bermaksud untuk melakukan penelitian dengan judul "**Rancang Bangun Aplikasi Persediaan dan Pengadaan Material Pada Gudang PDAM TIRTA WAE SAI Barru**".
Sehubungan dengan hal tersebut, dimohon kepada Bapak/Ibu untuk memberikan izin penelitian kepada Mahasiswa tersebut selama 2 (dua) Bulan di **PDAM TIRTA WAE SAI Kabupaten Barru**.
Atas Perhatian dan kerjasamanya, diucapkan terima kasih. *Wassalamu Alaikum Warahmatullahi Wabarakatuh*

Parepare, 1 Agustus 2024
**Ketua,**

**Dr. Iradhatullah Rahim, M.P.**
NIDN: 0926117601

Tembusan:
1. Rektor IPH UMPAR
2. Wakil Rektor I UMPAR
3. Wakil Rektor III UMPAR
4. Dekan Fakultas Teknik UMPAR
5. Ybs. Dian Indriyani
6. Arsip

## Lampiran 2 : Surat Permohonan Izin Penelitian

**PEMERINTAH KABUPATEN BARRU**
**DINAS PENANAMAN MODAL DAN PELAYANAN TERPADU SATU PINTU**
Mal Pelayanan Publik Masiga Lt. 1-3 Jl. Iskandar Unru
http://izinonline.barrukab.go.id e-mail : barrudpmptsptk@gmail.com .Kode Pos 90711

Barru, 05 Agustus 2024

Nomor : 440/IP/DPMPTSP/VIII/2024
Lampiran : -
Perihal : Izin/Rekomendasi Penelitian

Kepada
Yth Direktur Perumda Tirta Waesai Kab. Barru
di-
Tempat

Berdasarkan Surat Kepala Dinas Penanaman Modal dan Pelayanan Terpadu Satu Pintu Provinsi SulSel Nomor 20629/S.01/PTSP/2024 tanggal, 2 Agustus 2024 perihal tersebut di atas, maka ***Mahasiswi*** di bawah ini

| | | |
|---|---|---|
| **Nama** | : | **Dian Indriyani** |
| **Nomor Pokok** | : | **220280059** |
| **Program Studi** | : | **Teknik Informatika** |
| **Perguruan Tinggi** | : | **Universitas Muhammadiyah Parepare** |
| **Pekerjaan/Lembaga** | : | **Mahasiswi** |
| **Alamat** | : | **Aroppoe Desa Tellumpanua Kec. Tanete Rilau Kab. Barru** |

Diberikan izin untuk melakukan Penelitian/Pengambilan Data di Wilayah/Kantor Saudara yang berlangsung mulai tanggal **06 Agustus 2024 s/d 06 Oktober 2024,** dalam rangka penyusunan ***Tesis,*** dengan judul :

**RANCANG BANGUN APLIKASI PERSEDIAAN DAN PENGADAAN MATERIAL PADA GUDANG PDAM TIRTA WAESAI BARRU**

Sehubungan dengan hal tersebut diatas, pada prinsipnya kami menyetujui kegiatan dimaksud dengan ketentuan :

1. Sebelum dan sesudah melaksanakan kegiatan, kepada yang bersangkutan melapor kepada Kepala SKPD (Unit Kerja) Camat, apabila kegiatan dilaksanakan di SKPD (Unit Kerja) / Kecamatan setempat;
2. Penelitian tidak menyimpang dari izin yang diberikan;
3. Mentaati semua Peraturan Perundang Undangan yang berlaku dan mengindahkan adat istiadat setempat;
4. Menyerahkan 1(satu) eksampelar copy hasil penelitian kepada Bupati Barru Cq. Kepala Dinas Penanaman Modal Dan Pelayanan Terpadu Satu Pintu Kabupaten Barru;
5. Surat Izin akan dicabut kembali dan dinyatakan tidak berlaku apabila ternyata pemegang surat izin ini tidak mentaati ketentuan tersebut di atas

Untuk terlaksananya tugas penelitian tersebut degan baik dan lancar, diminta kepada Saudara (i) untuk memberikan bantuan fasilitas seperlunya.

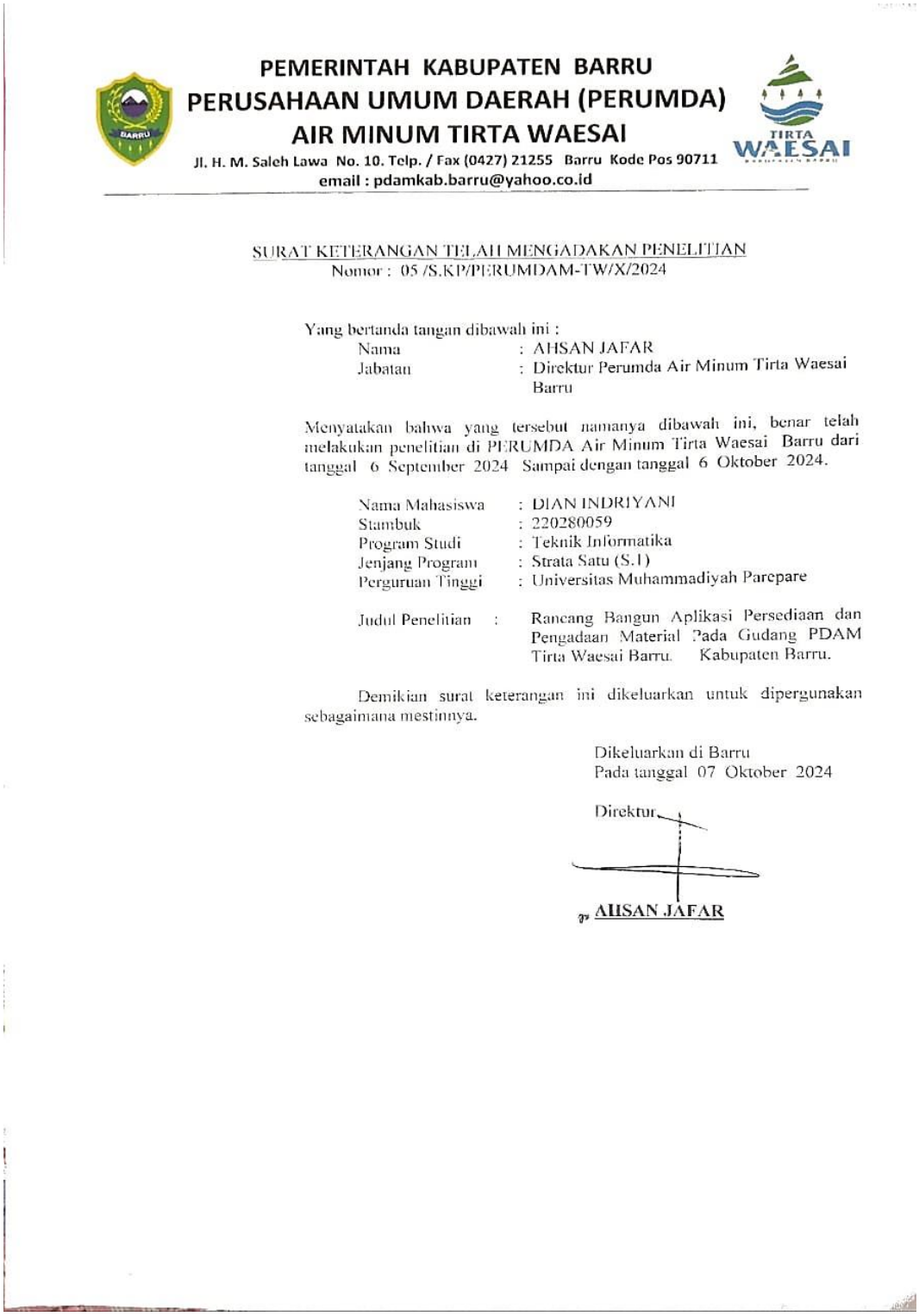
Demikian disampaikan untuk dimaklumi dan dipergunakan seperlunya.

**Kepala Dinas,**

**ANDI SYUKUR MAKKAWARU, S.STP,M.Si**
**Pangkat : Pembina Utama Muda, IV/c**
**NIP. 19770829 199612 1 001**

**TEMBUSAN** : disampaikan Kepada Yth.
1. Bapak Bupati (sebagai laporan);
2. Kepala Bappelitbangda Kab. Barru;
3. Ketua LPPM Univ. Muhammadiyah Parepare;
4. Mahasiswi yang bersangkutan.

**Lampiran 3 : Surat Sudah Penelitian**



PEMERINTAH KABUPATEN BARRU
PERUSAHAAN UMUM DAERAH (PERUMDA)
AIR MINUM TIRTA WAESAI
Jl. H. M. Saleh Lawa No. 10. Telp. / Fax (0427) 21255 Barru Kode Pos 90711
email : pdamkab.barru@yahoo.co.id

SURAT KETERANGAN TELAH MENGADAKAN PENELITIAN
Nomor : 05 /S.KP/PERUMDAM-TW/X/2024

Yang bertanda tangan dibawah ini :

| | | |
|---|---|---|
| Nama | : | AHSAN JAFAR |
| Jabatan | : | Direktur Perumda Air Minum Tirta Waesai Barru |

Menyatakan bahwa yang tersebut namanya dibawah ini, benar telah melakukan penelitian di PERUMDA Air Minum Tirta Waesai Barru dari tanggal 6 September 2024 Sampai dengan tanggal 6 Oktober 2024.

| | | |
|---|---|---|
| Nama Mahasiswa | : | DIAN INDRIYANI |
| Stambuk | : | 220280059 |
| Program Studi | : | Teknik Informatika |
| Jenjang Program | : | Strata Satu (S.1) |
| Perguruan Tinggi | : | Universitas Muhammadiyah Parepare |
| Judul Penelitian | : | Rancang Bangun Aplikasi Persediaan dan Pengadaan Material Pada Gudang PDAM Tirta Waesai Barru. Kabupaten Barru. |

Demikian surat keterangan ini dikeluarkan untuk dipergunakan sebagaimana mestinya.

Dikeluarkan di Barru
Pada tanggal 07 Oktober 2024

Direktur

AHSAN JAFAR

**Coding Code Igniter**

/htdocs/inventaris/application/controllers/Auth.*PHP*

```
<?PHP
defined('BASEPATH') or exit('No direct script  access allowed');
class Auth extends CI_Controller
{
   public function __construct()
   {
      parent::__construct();
      $this->load->library('form_validation');
      $this->load->model('Auth_model', 'auth');
      $this->load->model('Admin_model', 'admin');
   }
   private function _has_login ()
   {
      if ($this->session->has_Userdata('login _session')) {
         redirect('dashboard');
      }
   }
   public function index()
   {
      $this->_has_login ();
      $this->form_validation->set_rules('Username ', 'Username ', 'required|trim');
      $this->form_validation->set_rules('password', 'Password', 'required|trim');
      if ($this->form_validation->run() == false) {
         $data['title'] = 'Login  Aplikasi';
         $this->template->load('templates/auth', 'auth/login ', $data);
      } else {
         $input = $this->input->post(null, true);
         $cek_Username  = $this->auth->cek_Username ($input['Username ']);
         if ($cek_Username  > 0) {
            $password = $this->auth->get_password($input['Username ']);
            if (password_verify($input['password'], $password)) {
               $User_db = $this->auth->Userdata($input['Username ']);
               if ($User_db['is_active'] != 1) {
                  set_pesan('akun anda belum aktif/dinonaktifkan. Silahkan hubungi
      admin.', false);
                  redirect('login ');
               } else {
                     $Userdata = [
                        'User'  => $User_db['id_User'],
                        'role'  => $User_db['role'],
                        'timestamp' => time()
                     ];
                  $this->session->set_Userdata('login _session', $Userdata);
                  redirect('dashboard');
```

```
            }
        } else {
            set_pesan('password salah', false);
            redirect('auth');
        }
    } else {
        set_pesan('Username  belum terdaftar', false);
        redirect('auth');
    }
  }
}
public function logout()
{
  $this->session->unset_Userdata('login _session');

  set_pesan('anda telah berhasil logout');
  redirect('auth');
}
public function register()
{
  $this->form_validation->set_rules('Username ', 'Username ',
    'required|trim|is_unique[User.Username ]|alpha_numeric');
  $this->form_validation->set_rules('password', 'Password',
    'required|min_length[3]|trim');
  $this->form_validation->set_rules('password2', 'Konfirmasi Password',
    'matches[password]|trim');
  $this->form_validation->set_rules('nama', 'Nama', 'required|trim');
  $this->form_validation->set_rules('email', 'Email',
    'required|trim|valid_email|is_unique[User.email]');
  $this->form_validation->set_rules('no_telp', 'Nomor Telepon', 'required|trim');
  if ($this->form_validation->run() == false) {
    $data['title'] = 'Buat Akun';
    $this->template->load('templates/auth', 'auth/register', $data);
  } else {
    $input = $this->input->post(null, true);
    unset($input['password2']);
    $input['password']  =  password_hash($input['password'],
    PASSWORD_DEFAULT);
    $input['role']  = 'gudang';
    $input['foto']  = 'User.png';
    $input['is_active']  = 0;
    $input['created_at']  = time();
    $query = $this->admin->insert('User', $input);
    if ($query) {
      set_pesan('daftar berhasil. Selanjutnya silahkan hubungi admin untuk
    mengaktifkan akun anda.');
```

```
            redirect('login ');
        } else {
            set_pesan('gagal menyimpan ke database', false);
            redirect('register');
        }
    }
  }
}
```

**/htdocs/inventaris/application/controllers/Barang.*PHP***

```
<?PHP
defined('BASEPATH') or exit('No direct script  access allowed');
class Barang extends CI_Controller
{
  public function __construct()
  {
    parent::__construct();
    cek_login ();

    $this->load->model('Admin_model', 'admin');
    $this->load->library('form_validation');
  }
  public function index()
  {
    $data['title'] = "Barang";
    $data['barang'] = $this->admin->getBarang();
    $this->template->load('templates/dashboard', 'barang/data', $data);
  }
  private function _validasi()
  {
    $this->form_validation->set_rules('nama_barang', 'Nama Barang',
      'required|trim');
    $this->form_validation->set_rules('jenis_id', 'Jenis Barang', 'required');
    $this->form_validation->set_rules('satuan_id', 'Satuan Barang', 'required');
  }
  public function add()
  {
    $this->_validasi();
    if ($this->form_validation->run() == false) {
      $data['title'] = "Barang";
      $data['jenis'] = $this->admin->get('jenis');
      $data['satuan'] = $this->admin->get('satuan');

      // Mengenerate ID Barang
      $kode_terakhir = $this->admin->getMax('barang', 'id_barang');
      $kode_tambah = substr($kode_terakhir, -6, 6);
```

```
        $kode_tambah++;
        $number = str_pad($kode_tambah, 6, '0', STR_PAD_LEFT);
        $data['id_barang'] = 'B' . $number;
        $this->template->load('templates/dashboard', 'barang/add', $data);
    } else {
        $input = $this->input->post(null, true);
        $insert = $this->admin->insert('barang', $input);

        if ($insert) {
            set_pesan('data berhasil disimpan');
            redirect('barang');
        } else {
            set_pesan('gagal menyimpan data');
            redirect('barang/add');
        }
    }
}
public function edit($getId)
{
    $id = encode_PHP_tags($getId);
    $this->_validasi();

    if ($this->form_validation->run() == false) {
        $data['title'] = "Barang";
        $data['jenis'] = $this->admin->get('jenis');
        $data['satuan'] = $this->admin->get('satuan');
        $data['barang'] = $this->admin->get('barang', ['id_barang' => $id]);
        $this->template->load('templates/dashboard', 'barang/edit', $data);
    } else {
        $input = $this->input->post(null, true);
        $update = $this->admin->update('barang', 'id_barang', $id, $input);
        if ($update) {
            set_pesan('data berhasil disimpan');
            redirect('barang');
        } else {
            set_pesan('gagal menyimpan data');
            redirect('barang/edit/' . $id);
        }
    }
}
public function delete($getId)
{
    $id = encode_PHP_tags($getId);
    if ($this->admin->delete('barang', 'id_barang', $id)) {
        set_pesan('data berhasil dihapus.');
    } else {
```

```
            set_pesan('data gagal dihapus.', false);
        }
        redirect('barang');
    }
    public function getstok($getId)
    {
        $id = encode_PHP_tags($getId);
        $query = $this->admin->cekStok($id);
        output_json($query);
    }
}
```

**/htdocs/inventaris/application/controllers/Barangkeluar.*PHP***

```
<?PHP
defined('BASEPATH') or exit('No direct script  access allowed');
class Barangkeluar extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        cek_login ();
        $this->load->model('Admin_model', 'admin');
        $this->load->library('form_validation');
    }
    public function index()
    {
        $data['title'] = "Barang Keluar";
        $data['barangkeluar'] = $this->admin->getBarangkeluar();
        $this->template->load('templates/dashboard', 'barang_keluar/data', $data);
    }
    private function _validasi()
    {
        // Aturan validasi untuk tanggal keluar
        $this->form_validation->set_rules('tanggal_keluar', 'Tanggal Keluar',
            'required|trim');
        $this->form_validation->set_rules('barang_id', 'Barang', 'required|trim');
        // Mendapatkan stok dari barang yang dipilih
        $input = $this->input->post('barang_id', true);
        $barang = $this->admin->get('barang', ['id_barang' => $input]);
        // Pastikan barang ada sebelum mengakses stok
        if ($barang) {
            $stok = $barang['stok'];
            $stok_valid = $stok + 1;
            // Aturan validasi untuk jUMLah keluar
            $this->form_validation->set_rules(
                'jUMLah_keluar',
```

```
            'JUMLah Keluar',
            "required|trim|numeric|greater_than[0]|less_than[{$stok_valid}]",
            [
                'less_than' => "JUMLah Keluar tidak boleh lebih dari {$stok}."
            ]
        );
    } else {
        $this->form_validation->set_message('barang_id', 'Barang tidak ditemukan.');
    }
}
public function add()
{
    $this->_validasi(); // Validasi input

    if ($this->form_validation->run() == false) {
        $data['title'] = "Tambah Barang Keluar";
        $data['barang'] = $this->admin->get('barang', null, ['stok >' => 0]);

        // Menggenerate kode transaksi barang keluar
        $kode = 'T-BK-' . date('ymd');
        $kode_terakhir = $this->admin->getMax('barang_keluar', 'id_barang_keluar', $kode);
        $kode_tambah = (int)substr($kode_terakhir, -5) + 1; // Tambahkan 1 pada kode terakhir
        $number = str_pad($kode_tambah, 5, '0', STR_PAD_LEFT);
        $data['id_barang_keluar'] = $kode . $number;
        $this->template->load('templates/dashboard', 'barang_keluar/add', $data);
    } else {
        $input = $this->input->post(null, true);
        $insert = $this->admin->insert('barang_keluar', $input);
        // Cek apakah insert berhasil
        if ($insert) {
            set_pesan('Data berhasil disimpan.');
            redirect('barangkeluar');
        } else {
            set_pesan('Opps ada kesalahan! Silakan coba lagi.', false);
            redirect('barangkeluar/add');
        }
    }
}
public function delete($getId)
{
    $id = encode_PHP_tags($getId); // Sanitasi ID yang diterima
    if ($this->admin->delete('barang_keluar', 'id_barang_keluar', $id)) {
        set_pesan('Data berhasil dihapus.');
```

```
        } else {
            set_pesan('Data gagal dihapus.', false);
        }
        redirect('barangkeluar');
    }
}
```

**/htdocs/inventaris/application/controllers/Barangmasuk.*PHP***

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

class Barangmasuk extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        cek_login ();

        $this->load->model('Admin_model', 'admin');
        $this->load->library('form_validation');
    }
    public function index()
    {
        $data['title'] = "Barang Masuk";
        $data['barangmasuk'] = $this->admin->getBarangMasuk();
        $this->template->load('templates/dashboard', 'barang_masuk/data', $data);
    }
    public function setujui($id_barang_masuk)
    {
        $this->admin->update('barang_masuk', 'id_barang_masuk',
            $id_barang_masuk, ['status' => 'disetujui']);
        set_pesan('Pembelian disetujui.');
        redirect('barangmasuk');
    }
    public function tolak($id_barang_masuk)
    {
        $this->admin->update('barang_masuk', 'id_barang_masuk',
            $id_barang_masuk, ['status' => 'ditolak']);
        set_pesan('Pembelian ditolak.');
        redirect('barangmasuk');
    }
    private function _validasi()
    {
        $this->form_validation->set_rules('tanggal_masuk', 'Tanggal Masuk',
            'required|trim');
        $this->form_validation->set_rules('supplier_id', 'Supplier', 'required');
        $this->form_validation->set_rules('barang_id', 'Barang', 'required');
```

```
    $this->form_validation->set_rules('jUMLah_masuk', 'JUMLah Masuk',
        'required|trim|numeric|greater_than[0]');
    $this->form_validation->set_rules('harga', 'Harga', 'required|numeric'); //
        Validasi untuk harga
}
public function add()
{
    $this->_validasi();
    if ($this->form_validation->run() == false) {
        $data['title'] = "Barang Masuk";
        $data['supplier'] = $this->admin->get('supplier');
        $data['barang'] = $this->admin->get('barang');
        // Mendapatkan dan men-generate kode transaksi barang masuk
        $kode = 'T-BM-' . date('ymd');
        $kode_terakhir = $this->admin->getMax('barang_masuk',
        'id_barang_masuk', $kode);
        $kode_tambah = substr($kode_terakhir, -5, 5);
        $kode_tambah++;
        $number = str_pad($kode_tambah, 5, '0', STR_PAD_LEFT);
        $data['id_barang_masuk'] = $kode . $number;
        $this->template->load('templates/dashboard', 'barang_masuk/add', $data);
    } else {
        $input = $this->input->post(null, true);
        // Menghitung total harga
        $harga_satuan = $input['harga'];
        $jUMLah_masuk = $input['jUMLah_masuk'];
        $total_harga = $harga_satuan * $jUMLah_masuk;
        // Menambahkan total_harga ke input data
        $input['total_harga'] = $total_harga;
        // Insert ke database
        $insert = $this->admin->insert('barang_masuk', $input);
        if ($insert) {
            set_pesan('data berhasil disimpan.');
            redirect('barangmasuk');
        } else {
            set_pesan('Opps ada kesalahan!');
            redirect('barangmasuk/add');
        }
    }
}
public function delete($getId)
{
    $id = encode_PHP_tags($getId);
    if ($this->admin->delete('barang_masuk', 'id_barang_masuk', $id)) {
        set_pesan('data berhasil dihapus.');
    } else {
```

```
            set_pesan('data gagal dihapus.', false);
        }
        redirect('barangmasuk');
    }
}
```

**/htdocs/inventaris/application/controllers/Dashboard.*PHP***

```
<?PHP
defined('BASEPATH') or exit('No direct script  access allowed');
class Dashboard extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        cek_login ();
        $this->load->model('Admin_model', 'admin');
    }
    public function index()
    {
        $data['title'] = "Dashboard";
        $data['barang'] = $this->admin->count('barang');
        $data['barang_masuk'] = $this->admin->count('barang_masuk');
        $data['barang_keluar'] = $this->admin->count('barang_keluar');
        $data['supplier'] = $this->admin->count('supplier');
        $data['User'] = $this->admin->count('User');
        $data['stok'] = $this->admin->sum('barang', 'stok');
        $data['barang_min'] = $this->admin->min('barang', 'stok', 10);
        $data['transaksi'] = [
            'barang_masuk' => $this->admin->getBarangMasuk(5),
            'barang_keluar' => $this->admin->getBarangKeluar(5)
        ];
        // Line Chart
        $bln = ['01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12'];
        $data['cbm'] = [];
        $data['cbk'] = [];

        foreach ($bln as $b) {
            $data['cbm'][] = $this->admin->chartBarangMasuk($b);
            $data['cbk'][] = $this->admin->chartBarangKeluar($b);
        }
        $this->template->load('templates/dashboard', 'dashboard', $data);
    }
}
```

**/htdocs/inventaris/application/models/Admin_model.*PHP***

```
<?PHP
```

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

class Admin_model extends CI_Model
{
   public function get($table, $data = null, $where = null)
   {
      if ($data != null) {
         return $this->db->get_where($table, $data)->row_array();
      } else {
         return $this->db->get_where($table, $where)->result_array();
      }
   }
   public function update($table, $pk, $id, $data)
   {
      $this->db->where($pk, $id);
      return $this->db->update($table, $data);
   }
   public function insert($table, $data, $batch = false)
   {
      return $batch ? $this->db->insert_batch($table, $data) : $this->db-
         >insert($table, $data);
   }
   public function delete($table, $pk, $id)
   {
      return $this->db->delete($table, [$pk => $id]);
   }
  public function getUsers($id)
   {
      /**
       * ID disini adalah untuk data yang tidak ingin ditampilkan.
       * Maksud saya disini adalah
       * tidak ingin menampilkan data User yang digunakan,
       * pada managemen data User
       */
      $this->db->where('id_User !=', $id);
      return $this->db->get('User')->result_array();
   }
   public function getBarang()
   {
      $this->db->join('jenis j', 'b.jenis_id = j.id_jenis');
      $this->db->join('satuan s', 'b.satuan_id = s.id_satuan');
      $this->db->order_by('id_barang');
      return $this->db->get('barang b')->result_array();
   }
   public function getBarangMasuk($limit = null, $id_barang = null, $range = null)
   {
```

```
    $this->db->select('bm.*, u.nama, sp.nama_supplier, sp.no_telp, sp.alamat,
      b.nama_barang, s.nama_satuan, bm.harga, bm.total_harga, bm.status'); //
      tambahkan kolom status
    $this->db->join('User u', 'bm.User_id = u.id_User');
    $this->db->join('supplier sp', 'bm.supplier_id = sp.id_supplier');
    $this->db->join('barang b', 'bm.barang_id = b.id_barang');
    $this->db->join('satuan s', 'b.satuan_id = s.id_satuan');

    if ($limit != null) {
      $this->db->limit($limit);
    }

    if ($id_barang != null) {
      $this->db->where('id_barang', $id_barang);
    }
    if ($range != null) {
      $this->db->where('tanggal_masuk >=', $range['mulai']);
      $this->db->where('tanggal_masuk <=', $range['akhir']);
    }
    $this->db->order_by('id_barang_masuk', 'DESC');
    return $this->db->get('barang_masuk bm')->result_array();
  }
  public function getBarangKeluar($limit = null, $id_barang = null, $range = null)
  {
    $this->db->select('*');
    $this->db->join('User u', 'bk.User_id = u.id_User');
    $this->db->join('barang b', 'bk.barang_id = b.id_barang');
    $this->db->join('satuan s', 'b.satuan_id = s.id_satuan');
    if ($limit != null) {
      $this->db->limit($limit);
    }
    if ($id_barang != null) {
      $this->db->where('id_barang', $id_barang);
    }
    if ($range != null) {
      $this->db->where('tanggal_keluar' . ' >=', $range['mulai']);
      $this->db->where('tanggal_keluar' . ' <=', $range['akhir']);
    }
    $this->db->order_by('id_barang_keluar', 'DESC');
    return $this->db->get('barang_keluar bk')->result_array();
  }
  public function getMax($table, $field, $kode = null)
  {
    $this->db->select_max($field);
    if ($kode != null) {
      $this->db->like($field, $kode, 'after');
```

```
        }
        return $this->db->get($table)->row_array()[$field];
    }
    public function count($table)
    {
        return $this->db->count_all($table);
    }
   public function sum($table, $field)
    {
        $this->db->select_sum($field);
        return $this->db->get($table)->row_array()[$field];
    }
    public function min($table, $field, $min)
    {
        $field = $field . ' <=';
        $this->db->where($field, $min);
        return $this->db->get($table)->result_array();
    }
    public function chartBarangMasuk($bulan)
    {
        $like = 'T-BM-' . date('y') . $bulan;
        $this->db->like('id_barang_masuk', $like, 'after');
        return count($this->db->get('barang_masuk')->result_array());
    }
   public function chartBarangKeluar($bulan)
    {
        $like = 'T-BK-' . date('y') . $bulan;
        $this->db->like('id_barang_keluar', $like, 'after');
        return count($this->db->get('barang_keluar')->result_array());
    }
    public function laporan($table, $mulai, $akhir)
    {
        $tgl = $table == 'barang_masuk' ? 'tanggal_masuk' : 'tanggal_keluar';
        $this->db->where($tgl . ' >=', $mulai);
        $this->db->where($tgl . ' <=', $akhir);
        return $this->db->get($table)->result_array();
    }

    public function cekStok($id)
    {
        $this->db->join('satuan s', 'b.satuan_id=s.id_satuan');
        return $this->db->get_where('barang b', ['id_barang' => $id])->row_array();
    }
}
```

**/htdocs/inventaris/application/models/Auth_model.*PHP***

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

class Auth_model extends CI_Model
{
 public function cek_Username ($Username )
   {
      $query = $this->db->get_where('User', ['Username ' => $Username ]);
      return $query->num_rows();
   }

   public function get_password($Username )
   {
      $data = $this->db->get_where('User', ['Username ' => $Username ])-
         >row_array();
      return $data['password'];
   }

   public function Userdata($Username )
   {
      return $this->db->get_where('User', ['Username ' => $Username ])-
         >row_array();
   }
}
```