**Data Manipulation Language**

Saya menggunakan model yang saya buat di project

1. **User**
   1. **Menambah User**

public function add($input)

    {

        if (!$this->db->insert($this->\_table, $input)) {

            return false;

        } else {

            return true;

        }

    }

Berfungsi untuk menambah user

Insert into user {nama fields} values {value};

* 1. **Mengubah User**

public function edit($input, $id)

    {

        $this->db->where('id', $id);

        $this->db->update($this->\_table, $input);

    }

Berfungsi untuk mengubah profil user

update user set {field} = {value} where {condition};

field, value, dan where bisa dikondisikan sejumlah keinginan

* 1. **Get List User**

public function getList($type)

    {

        $sql = "  SELECT \*

        from " . $this->\_table . "

        where type<>'administrator' and type='".$type."'

        order by nama";

        $list = $this->db->query($sql)->result();

        return $list;

    }

Berfungsi untuk mengambil list user kecuali administrator

Select \* from user where type<>’administrator’ and type={type}

* 1. **Get Record By Id**

public function getListById($id)

    {

        $sql = "  SELECT \*

        from " . $this->\_table . "

        where id='".$id."'

        order by nama";

        $list = $this->db->query($sql)->result();

        return $list;

    }

Berfungsi untuk mengambil 1 record user sesuai dengan id mereka

Select \* from user where id={id} order by nama

1. **Kategori**
   1. **Menambah Kategori**

public function add($input)

    {

        if (!$this->db->insert($this->\_table, $input)) {

            return false;

        } else {

            return true;

        }

    }

Berfungsi untuk menambah kategori

Insert into kategori {nama fields} values {value};

* 1. **Mengubah Kategori**

public function edit($input, $id)

    {

        $this->db->where('id', $id);

        $this->db->update($this->\_table, $input);

    }

Berfungsi untuk mengubah kategori

update kategori set {field} = {value} where {condition};

field, value, dan where bisa dikondisikan sejumlah keinginan

* 1. **Get List Kategori**

public function getList()

    {

        $sql = "  SELECT \*

        from " . $this->\_table . "

        order by nama";

        $list = $this->db->query($sql)->result();

        return $list;

    }

Berfungsi untuk mengambil list kategori

Select \* from kategori order by nama

* 1. **Get Record By Id**

public function getListById($id)

    {

        $sql = "  SELECT \*

        from " . $this->\_table . "

        where id='".$id."'

        order by nama";

        $list = $this->db->query($sql)->result();

        return $list;

    }

Berfungsi untuk mengambil single record kategori by id

Select \* from kategori where id={id} order by nama

1. **Produk**
   1. **Menambah Produk**

public function add($input)

    {

        if (!$this->db->insert($this->\_table, $input)) {

            return false;

        } else {

            return true;

        }

    }

Berfungsi untuk menambah kategori

Insert into kategori {nama fields} values {value};

* 1. **Mengubah Produk**

public function edit($input, $id)

    {

        $this->db->where('id', $id);

        $this->db->update($this->\_table, $input);

    }

Berfungsi untuk mengubah kategori user

update kategori set {field} = {value} where {condition};

field, value, dan where bisa dikondisikan sejumlah keinginan

* 1. **Get List**

public function getList($aktif='')

    {

        $sql = "  SELECT k.id as id\_kategori,k.nama as nama\_kategori,p.id,p.nama as nama\_produk,p.harga\_beli,p.harga\_jual,p.deskripsi,gambar,aktif,

        ( (select sum(jumlah) from pembelian pe left outer join pembelian\_detail pd on pe.id=pd.id\_pembelian where pe.batal=0 and pd.id\_produk=p.id) -

        (select sum(jumlah) from penjualan pj left outer join penjualan\_detail pd on pj.id=pd.id\_penjualan where pd.id\_produk=p.id) ) as stok

        from " . $this->\_table . " p left outer join

        kategori k on p.id\_kategori=k.id ";

        if($aktif<>'')

            $sql .= "where aktif='".$aktif."' ";

        $sql .= "order by p.nama";

        $list = $this->db->query($sql)->result();

        return $list;

    }

Berfungsi untuk menampilkan list produk beserta stoknya

* 1. **Get List By Id**

public function getListById($id)

    {

        $sql = "  SELECT k.id as id\_kategori,k.nama as nama\_kategori,p.id,p.nama as nama\_produk,p.harga\_beli,p.harga\_jual,p.deskripsi,gambar,aktif,

        ( (select sum(jumlah) from pembelian pe left outer join pembelian\_detail pd on pe.id=pd.id\_pembelian where pe.batal=0 and pd.id\_produk=p.id) -

        (select sum(jumlah) from penjualan pj left outer join penjualan\_detail pd on pj.id=pd.id\_penjualan where pd.id\_produk=p.id) ) as stok

        from " . $this->\_table . " p left outer join

        kategori k on p.id\_kategori=k.id

        where p.id='".$id."'

        order by p.nama";

        $list = $this->db->query($sql)->result();

        return $list;

    }

Berfungsi untuk mengambil single record dari produk

* 1. **Check Id Exist**

public function checkExist($id)

    {

        $this->db->where('id', $id);

        $query = $this->db->get($this->\_table);

        if ($query->num\_rows() > 0) {

            return true;

        } else {

            return false;

        }

    }

Berfungsi untuk mengecek apakah id sudah exist atau tidak

Select \* from produk where id={id}

1. **Pembelian**
   1. **Menambah Pembelian**

public function add($table, $input)

    {

        if (!$this->db->insert($table, $input)) {

            return false;

        } else {

            return true;

        }

    }

Berfungsi untuk menambah pembelian

Insert into pembelian {nama fields} values {value};

* 1. **Mengubah Pembelian**

public function edit($input, $id)

    {

        $this->db->where('id', $id);

        $this->db->update($this->\_table, $input);

    }

Berfungsi untuk mengubah kategori lebih khususnya untuk membatalkan pembelian dengan mengupdate field batal menjadi 1

update pembelian set {field} = {value} where {condition};

* 1. **Commit Pembelian**

public function commit($input\_master, $input\_detail)

    {

        $this->db->trans\_start();

        $this->add($this->\_table,$input\_master);

        $this->add($this->\_table\_detail,$input\_detail);

        $this->db->trans\_complete();

        if ($this->db->trans\_status() === FALSE) {

            $this->db->trans\_rollback();

            return false;

        } else {

            $this->db->trans\_commit();

            return true;

        }

    }

Berfungsi untuk commit pembelian, dikarenakan insert lebih dari 1 tabel maka sebaiknya menggunakan trans commit

* 1. **Get List Pembelian**

public function getList($id='')

    {

        $sql = "  SELECT p.id as id,p.id\_supplier,p.created\_on,pd.id\_produk,pd.harga,pd.jumlah,u.nama as nama\_supplier,pr.nama as nama\_produk,batal

        from " . $this->\_table . " p left outer join

        " . $this->\_table\_detail . "  pd on p.id=pd.id\_pembelian left outer join

        user u on p.id\_supplier=u.id left outer join

        produk pr on pd.id\_produk=pr.id ";

        if($id<>'')

            $sql .= "where pr.id='".$id."'";

        $sql .= " order by p.created\_on desc";

        $list = $this->db->query($sql)->result();

        return $list;

    }

Berfungsi untuk menampilkan list pembelian.

* 1. **Get Pembelian Pre Produk**

public function getListPerProduk()

    {

        $sql = "SELECT pr.id as id\_produk,pr.nama as nama\_produk,count(p.id) as jumlah,harga as harga\_satuan,harga\*count(p.id) as harga\_total

        from " . $this->\_table . " p left outer join

        " . $this->\_table\_detail . "  pd on p.id=pd.id\_pembelian left outer join

        produk pr on pd.id\_produk=pr.id

        group by pd.id\_produk

        order by pr.nama;";

        $list = $this->db->query($sql)->result();

        return $list;

    }

1. **Penjualan**
   1. **Menambah Penjualan**

public function add($table, $input)

    {

        if (!$this->db->insert($table, $input)) {

            return false;

        } else {

            return true;

        }

    }

Berfungsi untuk menambah kategori

Insert into penjualan {nama fields} values {value};

* 1. **Mengubah Penjualan**

public function edit($input, $id)

    {

        $this->db->where('id', $id);

        $this->db->update($this->\_table, $input);

    }

Berfungsi untuk mengubah penjualan lebih khususnya untuk membatalkan penjualan dengan mengupdate field batal menjadi 1

update penjualan set {field} = {value} where {condition};

* 1. **Commit Penjualan**

public function commit($input\_master, $input\_detail)

    {

        $this->db->trans\_start();

        $this->add($this->\_table,$input\_master);

        $this->add($this->\_table\_detail,$input\_detail);

        $this->db->trans\_complete();

        if ($this->db->trans\_status() === FALSE) {

            $this->db->trans\_rollback();

            return false;

        } else {

            $this->db->trans\_commit();

            return true;

        }

    }

Berfungsi untuk commit penjualan, dikarenakan insert lebih dari 1 tabel maka sebaiknya menggunakan trans commit

* 1. **Get List Penjualan**

public function getList($id\_pelanggan='', $id\_produk='')

    {

        $sql = "  SELECT p.id as id,p.id\_pelanggan,p.created\_on,pd.id\_produk,pd.harga,pd.jumlah,u.nama as nama\_pelanggan,pr.nama as nama\_produk

        from " . $this->\_table . " p left outer join

        " . $this->\_table\_detail . "  pd on p.id=pd.id\_penjualan left outer join

        user u on p.id\_pelanggan=u.id left outer join

        produk pr on pd.id\_produk=pr.id ";

        if($id\_pelanggan<>'')

            $sql .= "where id\_pelanggan='".$id\_pelanggan."'";

        else if($id\_produk<>'')

            $sql .= "where pr.id='".$id\_produk."'";

        $sql .= " order by p.created\_on desc";

        $list = $this->db->query($sql)->result();

        return $list;

    }

Berfungsi untuk menampilkan list penjualan.

* 1. **Get List Penjualan Per Produk**

public function getListPerProduk()

    {

        $sql = "SELECT pr.id as id\_produk,pr.nama as nama\_produk,count(p.id) as jumlah,harga as harga\_satuan,harga\*count(p.id) as harga\_total

        from " . $this->\_table . " p left outer join

        " . $this->\_table\_detail . "  pd on p.id=pd.id\_penjualan left outer join

        produk pr on pd.id\_produk=pr.id

        group by pd.id\_produk

        order by pr.nama;";

        $list = $this->db->query($sql)->result();

        return $list;

    }

Berfungsi untuk menampilkan list penjualan per produk.