



# REPORT MANUAL OF MIDDLE EXAM

## REST SERVER & CLIENT – CODE IGNITER

### (Advanced Programming Web)

Name of Member	: Eko Setio Wijanarko	- 1841720043
	Revinda Amalia Saktyawati	- 1841720185
Class / NIM	: 2G / 1841720043 & 1841720185	
Major	: D-IV Informatics Engineering	

- REST SERVER (POSTMAN)

Aplikasi\_server/controllers/api/mozarella.php

```
<?php

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

require APPPATH . 'libraries/REST_Controller.php';
require APPPATH . 'libraries/Format.php';

class mozarella extends REST_Controller {

    public function __construct(){
        parent::__construct();
        $this->load->model('mozarella_model','barang');
    }
    public function index_get(){
        $id = $this->get('barang_id');
        if ($id === null){
            $barang = $this->barang->getMozarella();
        } else {
            $barang = $this->barang->getMozarella($id);
        }

        if($barang){
            $this->response([
                'status' => true,
                'data' => $barang
            ], REST_Controller::HTTP_OK);
        } else {
            $this->response([
                'status' => false,
                'message' => 'id not found'
            ], REST_Controller::HTTP_NOT_FOUND);
        }
    }
    public function index_delete(){
        $id=$this->delete('barang_id');

        if ($id === null){
            $this->response([
                'status' => false,
                'message' => 'provide an id!'
            ], REST_Controller::HTTP_BAD_REQUEST);
        } else {
            if ($this->barang->deleteMozarella($id) > 0){
                $this->response([
                    'status' => true,
                    'id' => $id,
                    'message' => 'deleted.'
                ], REST_Controller::HTTP_OK);
            } else {
                $this->response([
                    'status' => false,
                    'message' => 'id not found!'
                ], REST_Controller::HTTP_BAD_REQUEST);
            }
        }
    }
    public function index_post(){
        $data = [
            'barang_id' => $this->post('barang_id'),
            'kategori_id' => $this->post('kategori_id'),
            'nama_barang' => $this->post('nama_barang'),
            'harga' => $this->post('harga')
        ];

        if ($this->barang->createMozarella($data) > 0){
            $this -> response([
                'status' => true,
                'message' => 'new mozarella has been created'
            ], REST_Controller::HTTP_CREATED);
        } else {
            $this-> response([
                'status' => false,
                'message' => 'failed to create new data!'
            ], REST_Controller::HTTP_BAD_REQUEST);
        }
    }
    public function index_put(){
        $id = $this->put('barang_id');
        $data = [
            'barang_id' => $this->put('barang_id'),
            'kategori_id' => $this->put('kategori_id'),
            'nama_barang' => $this->put('nama_barang'),
            'harga' => $this->put('harga')
        ];

        if ($this->barang->updateMozarella($data, $id) > 0) {
            $this->response([
                'status' => true,
                'message' => 'data mozarella has been updated'
            ],REST_Controller::HTTP_OK);
        } else {
            $this->response([
                'status' => false,
                'message' => 'failed to update data!'
            ], REST_Controller::HTTP_BAD_REQUEST);
        }
    }
}
```

Application/models/mozzarella\_model.php

```
<?php

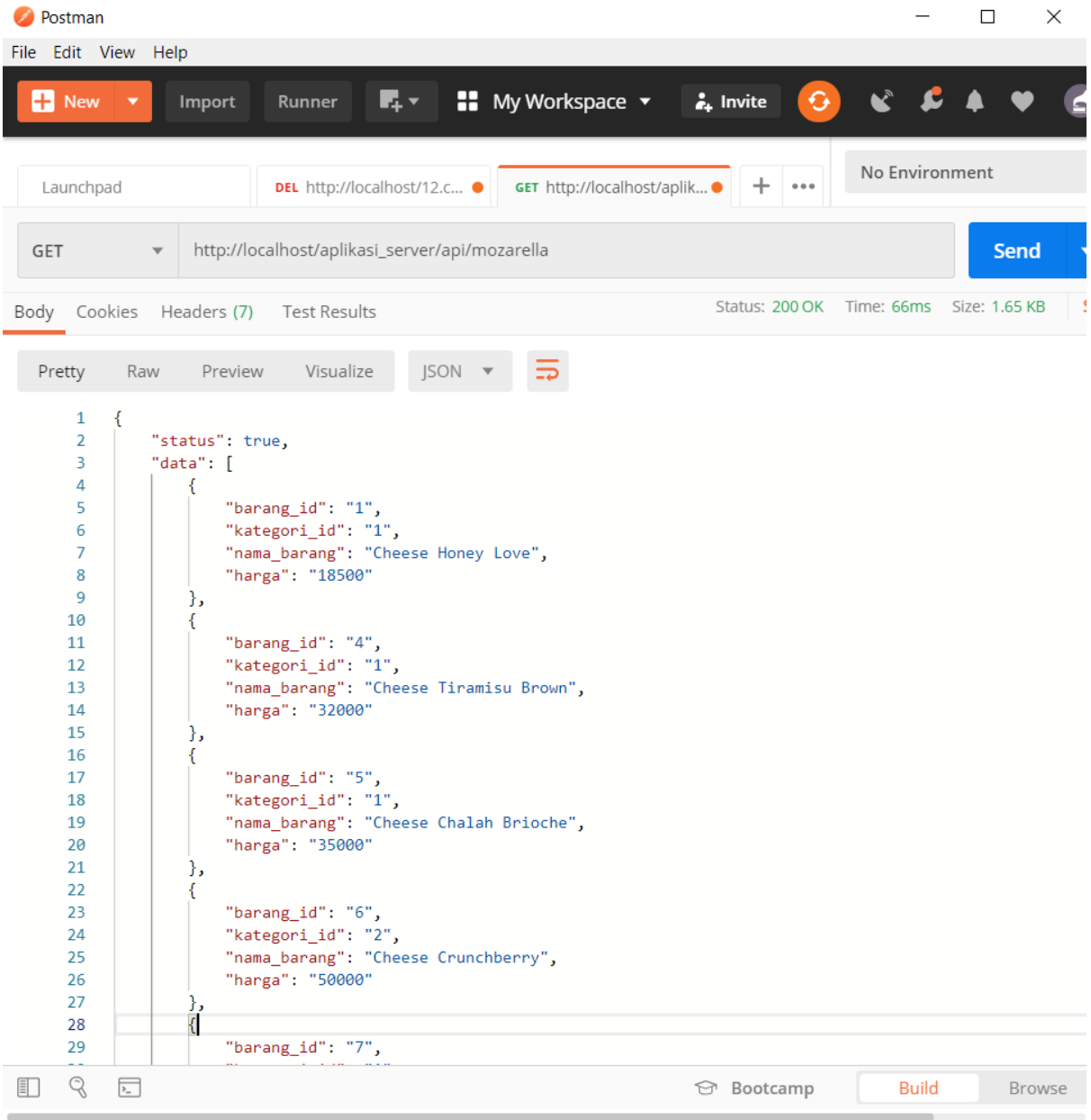
class mozzarella_model extends CI_Model {

    public function getMozarella($id = null) {
        if ($id === null){

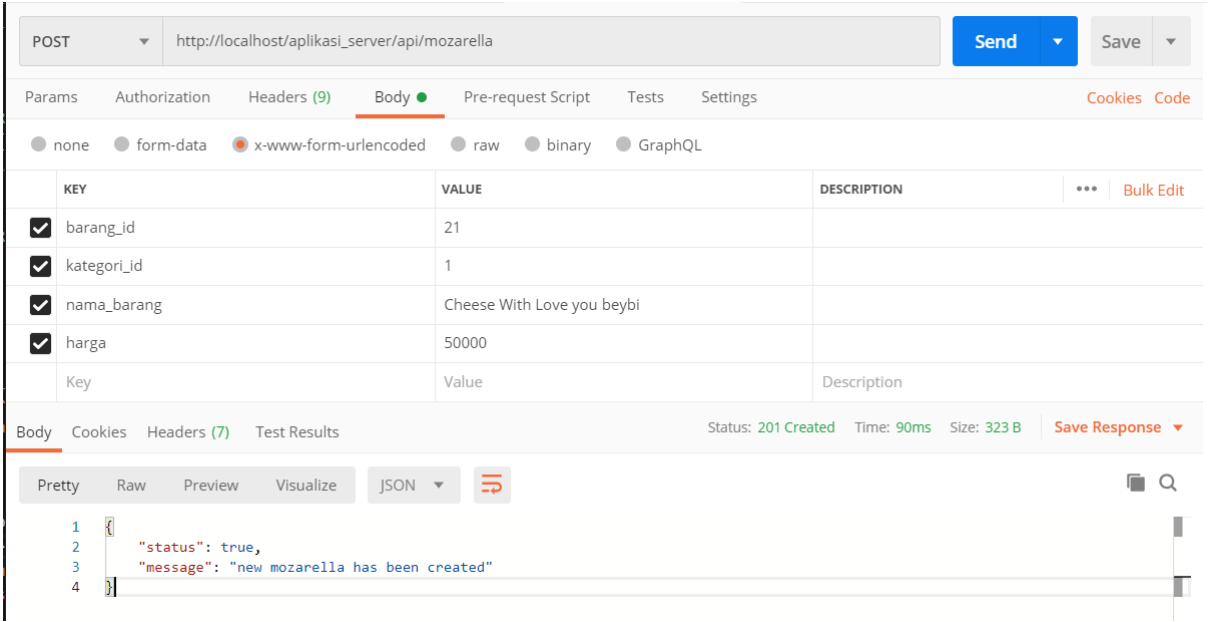
            return $this->db->get('barang')->result_array();
        } else{
            return $this->db->get_where('barang', ['barang_id' => $id])->result_array();
        }
    }
    public function deleteMozarella($id)
    {
        $this->db->delete('barang',['barang_id' => $id]);
        return $this->db->affected_rows();
    }
    public function createMozarella($data)
    {
        $this->db->insert('barang', $data);
        return $this->db->affected_rows();
    }
    public function updateMozarella($data, $id){
        $this->db->update('barang', $data, ['barang_id' => $id]);
        return $this->db->affected_rows();
    }
}

?>
```

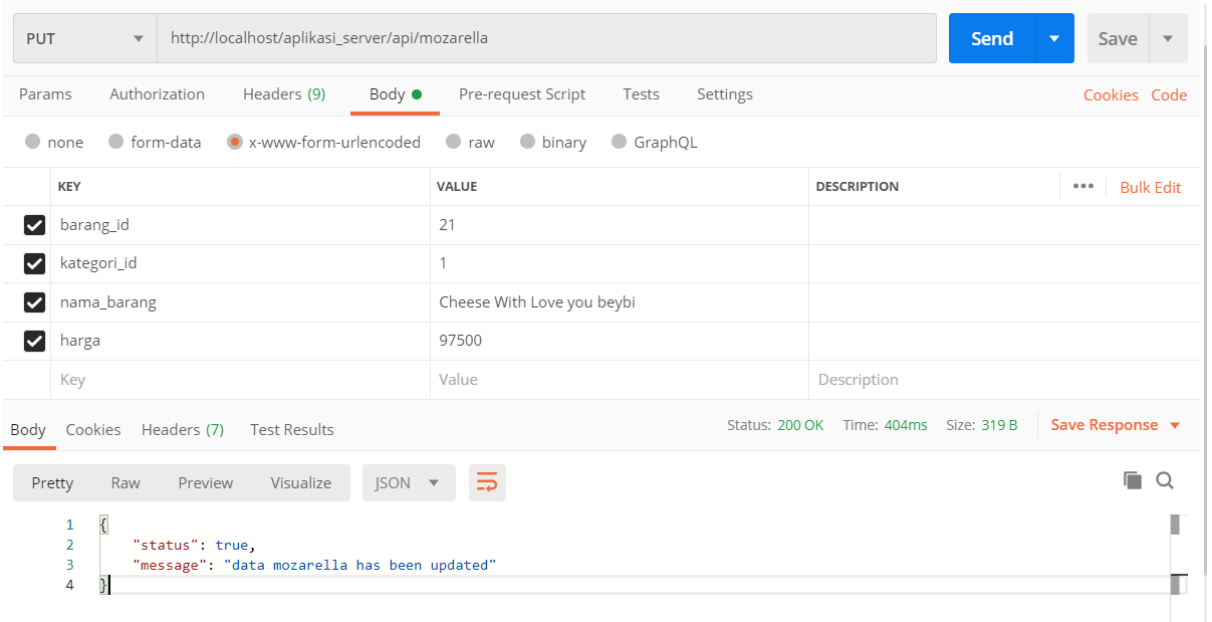
GET API BY POSTMAN



CREATE DATA BY POSTMAN



UPDATE DATA BY POSTMAN



DELETE DATA BY POSTMAN

DELETE

http://localhost/aplikasi\_server/api/mozarella

Send

Save

ParamsAuthorizationHeaders (9)BodyPre-request ScriptTestsSettingsCookiesCode

noneform-datax-www-form-urlencodedrawbinaryGraphQL

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	barang_id	21			
<input checked="" type="checkbox"/>	kategori_id	1			
<input checked="" type="checkbox"/>	nama_barang	Cheese With Love you beybi			
<input checked="" type="checkbox"/>	harga	97500			
	Key	Value	Description		

BodyCookiesHeaders (7)Test ResultsStatus: 200 OKTime: 84msSize: 306 BSave Response

PrettyRawPreviewVisualizeJSON

```
1 {
2   "status": true,
3   "id": "21",
4   "message": "deleted."
5 }
```

CLIENT APPLICATION INTERFACE

LOGIN PAGE as Index

Mozarella Malang

localhost/aplikasi\_pos/auth/login

Login Form | Mozarella Malang

Username

Password

LOGIN

Forgot Password?

Website : Mozarella Malang

DASHBOARD PAGE as Index after Login

Mozarella Malang

localhost/aplikasi\_pos/dashboard

MOZARELLA MALANG

Dashboard

Kategori Mozarella

Data Mozarella

Operator Sistem

Transaksi

Laporan

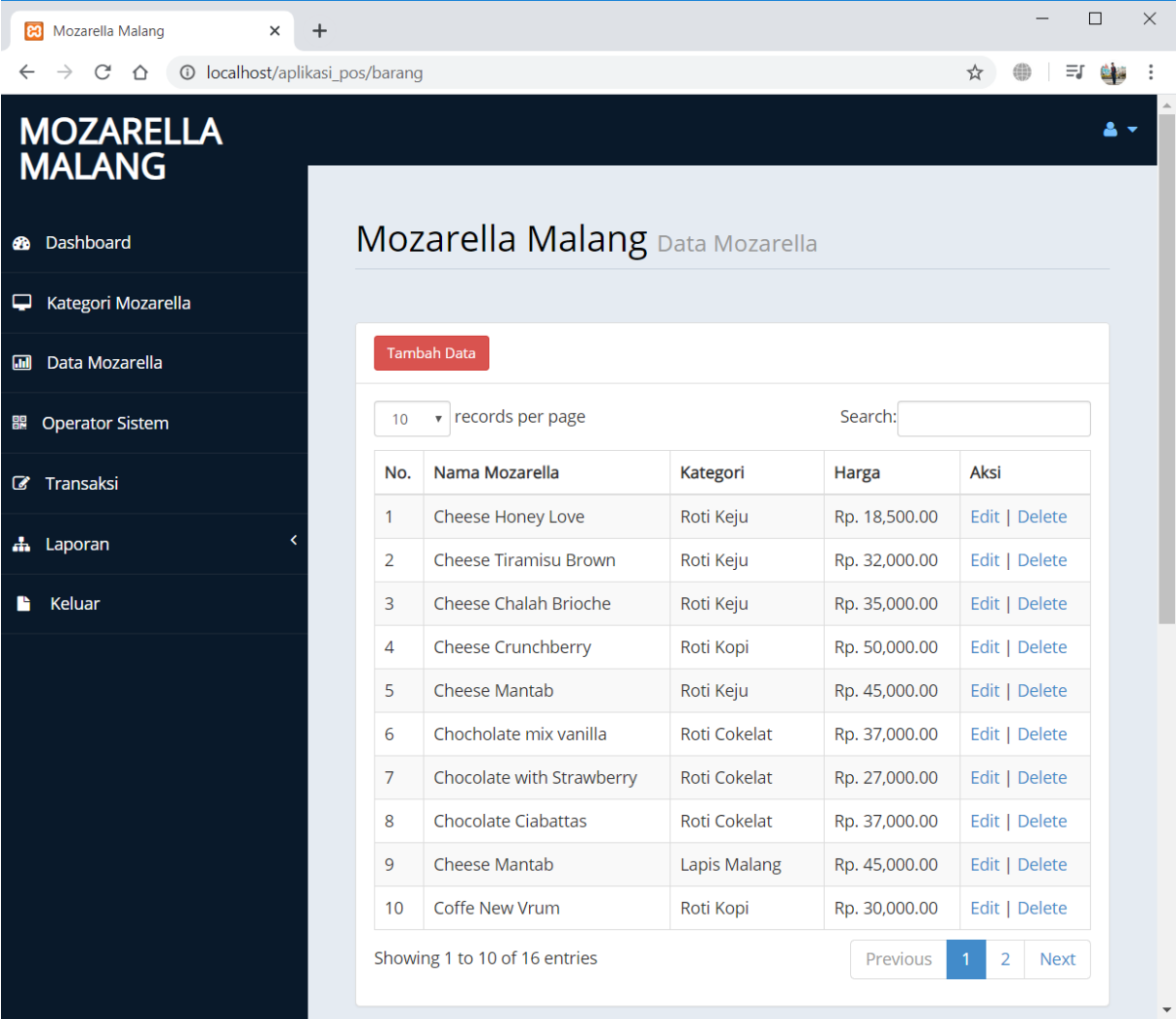
Keluar

Dashboard

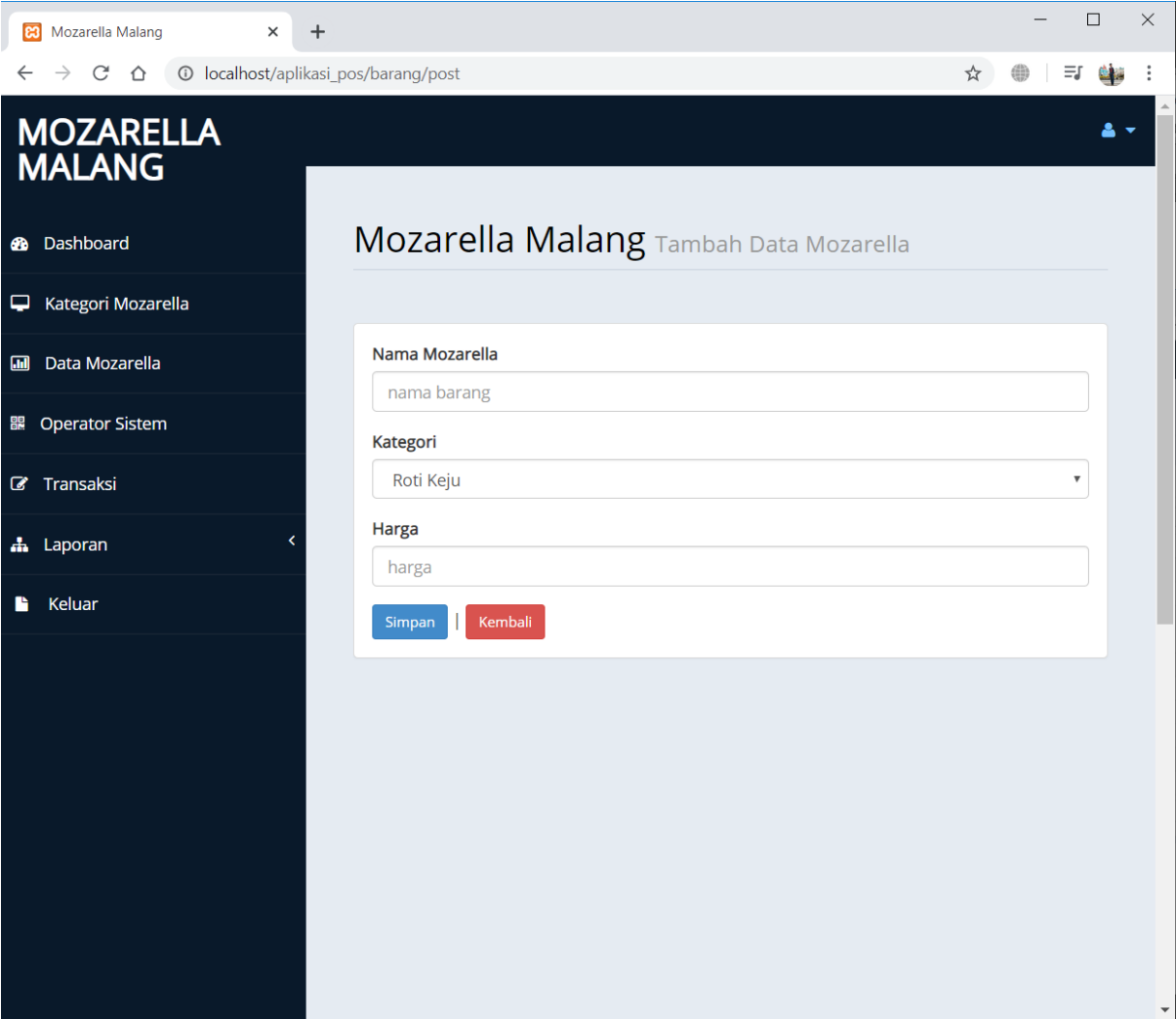
Aplikasi Penjualan Mozarella.

Aplikasi ini dikembangkan Oleh Eko Setio Wijanarko & Revinda Amalia Saktyawati

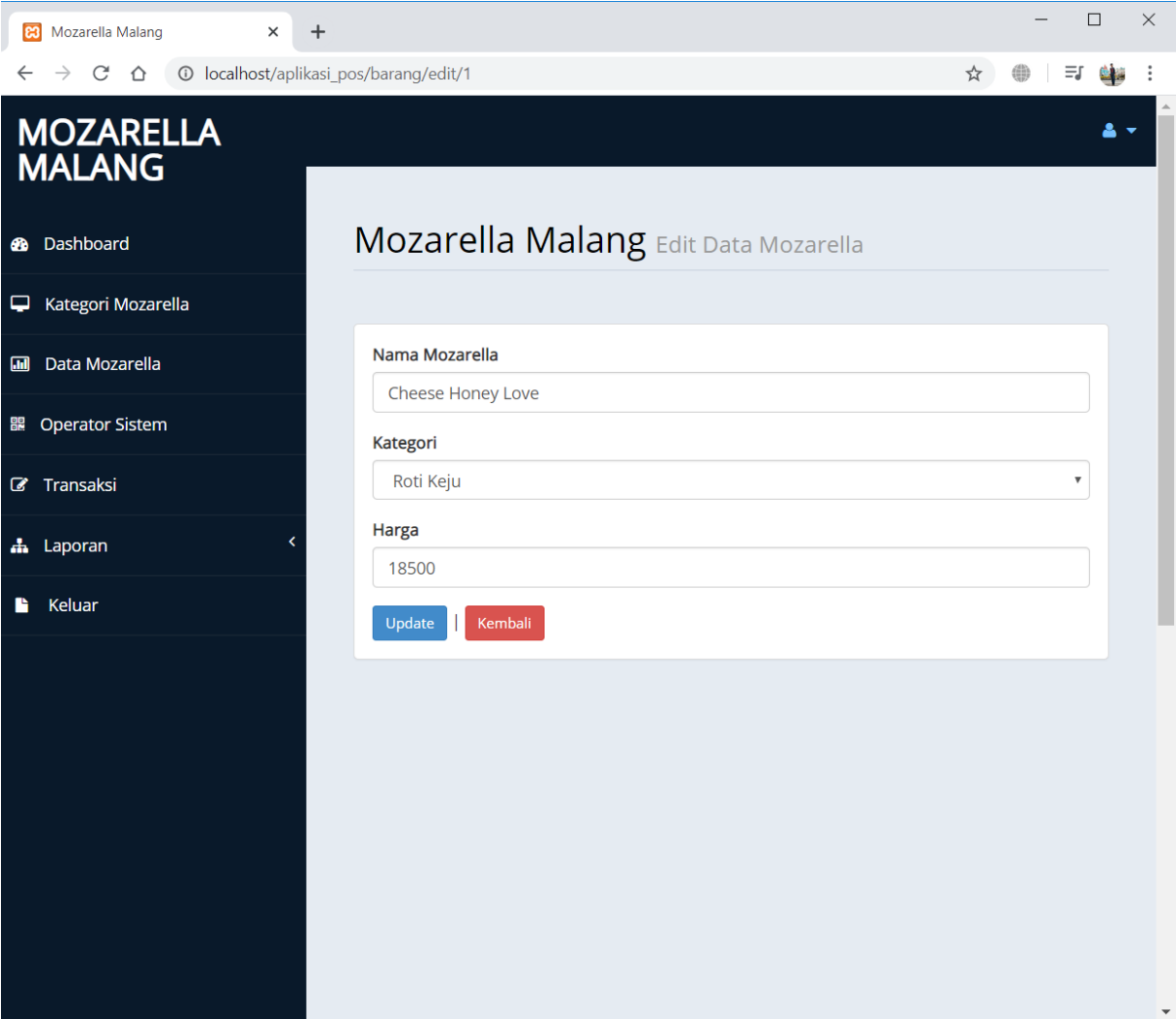
DATA MOZARELLA PAGE (MANAGE EDIT AND DELETE BUTTON EACH DATA)



CREATE DATA MOZARELLA PAGE



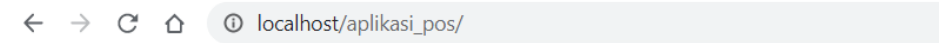
EDIT/UPDATE DATA MOZARELLA PAGE



Explanation of Web Client

In this occasion we introduce our application project is about manage Selling of Malang Cake Mozarella, so we want make to who can use the application can manage the data of Cake, all the name, categories, and price. In the interface of client, we can access that via local server. Using xampp as server and use php as a models.

First, we as we know we must access the application base url and show the main index.



Before we can access the main interface of application, we must login to make a verification user.

Login Form | Mozarella Malang

Username

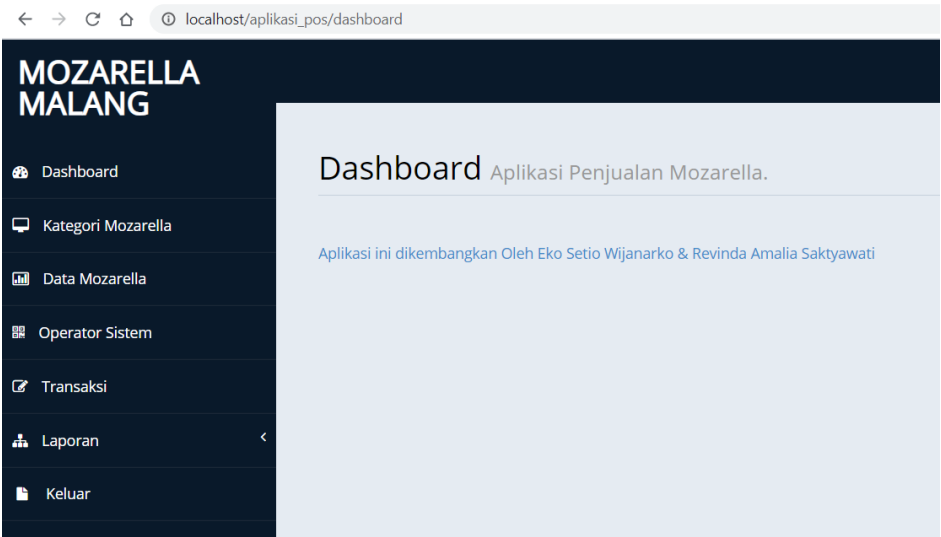
Password

LOGIN

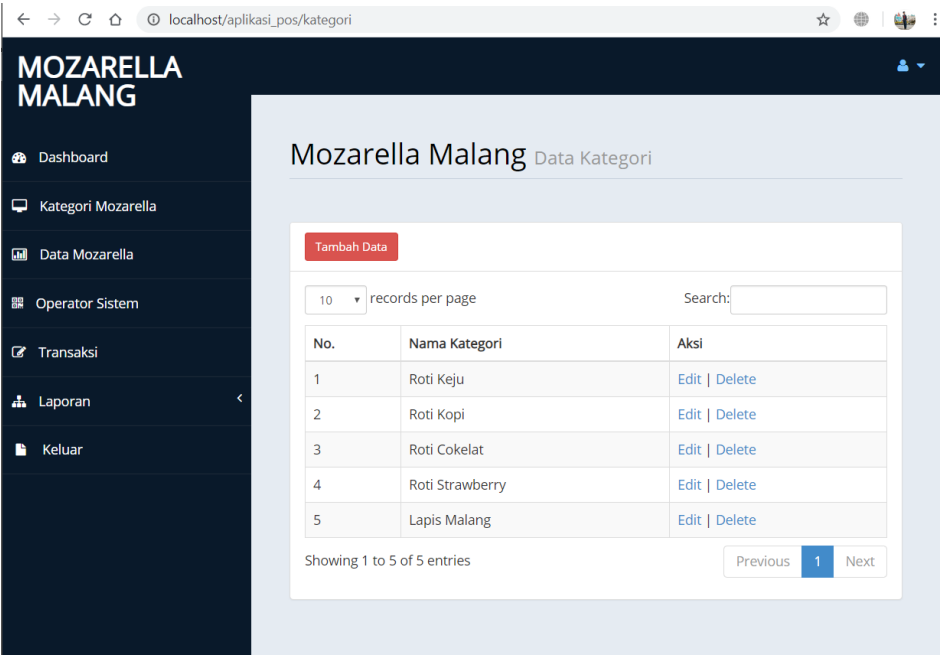
Forgot Password?

We use our username to login. We can use **eko** as username and **eko** as password, or **revinda** as username and **revinda** as password.

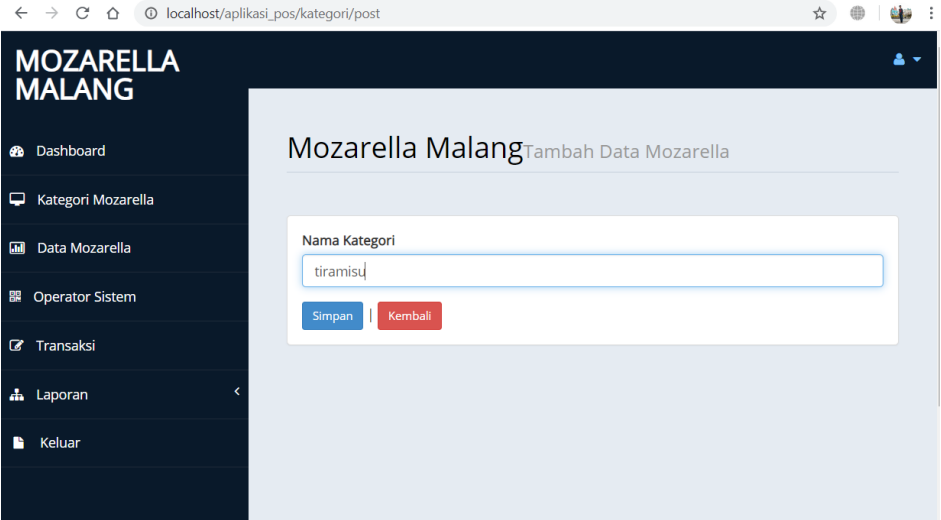
After login successfull we directly will go to the dashboard page is the main index after we can login.



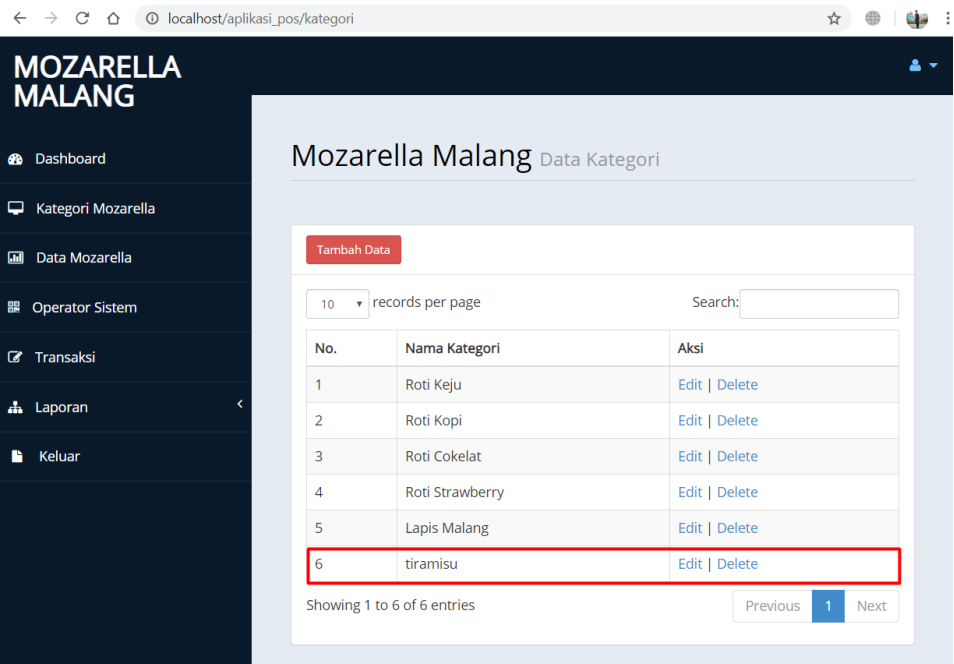
As we can see in the left side, there are sidebar to manage the operation of page look structurally and we can access each operation by click that sidebar.



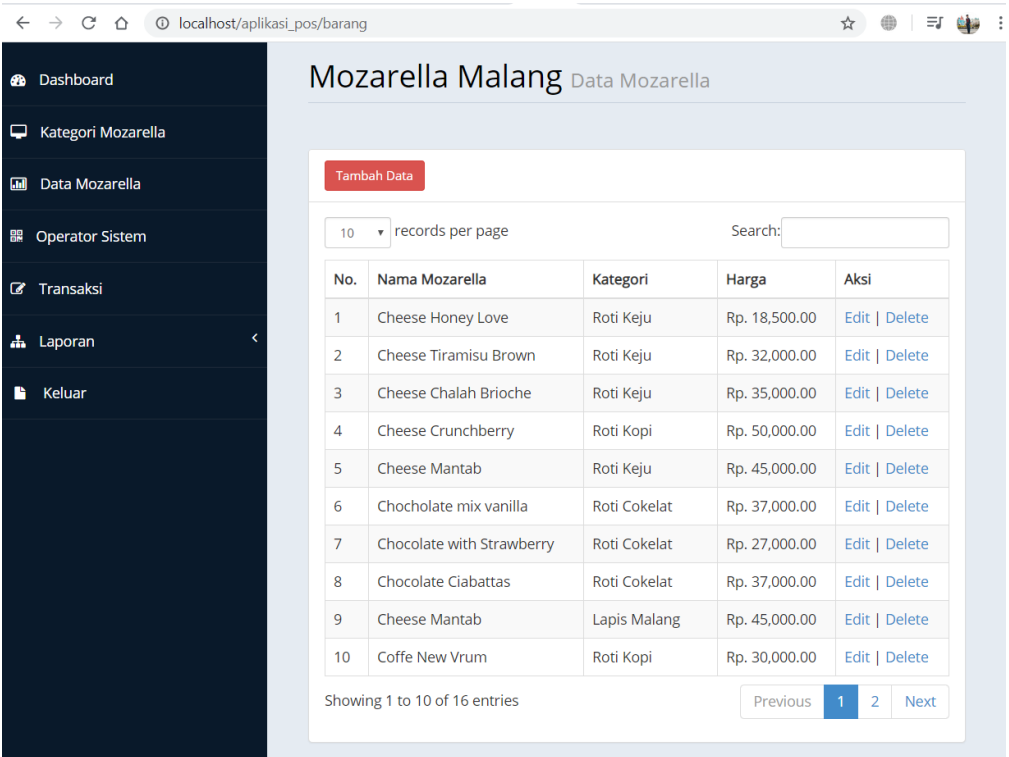
This is page of category, that we can view and manage the data of categories, we can also insert the new categories by clicking the **Tambah Data** button.



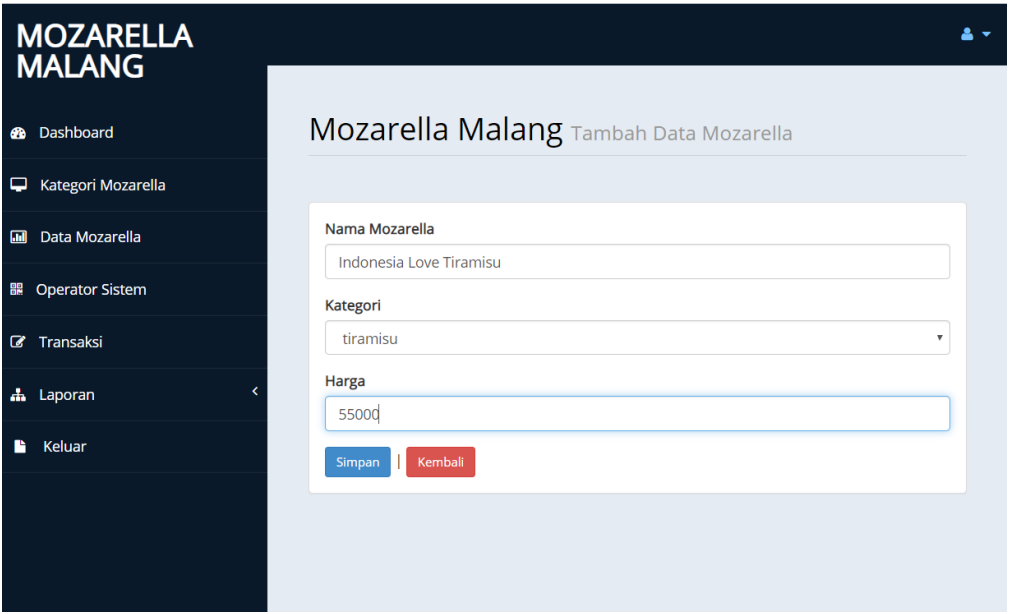
If click **Tambah Data** button for categories, we can add new categories by input the name of category and click Save. It will be shown in the Category page after we save it.



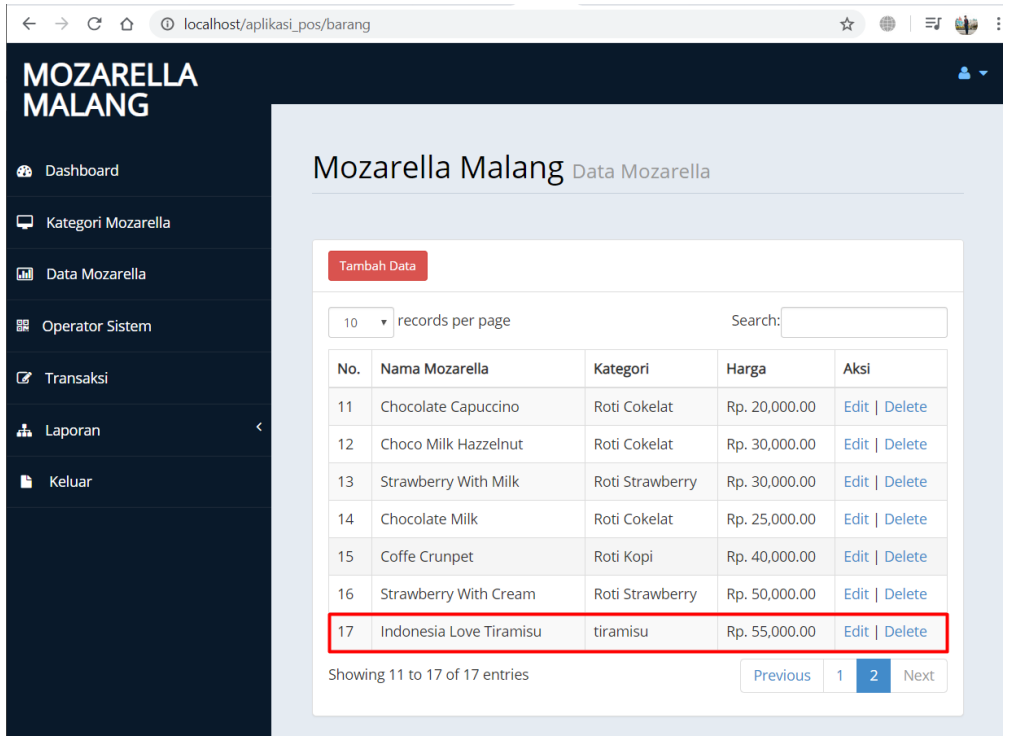
Then move to data mozzarella page, we can see the main data of Cake Mozarela contains Name, Category that we input before and Price.



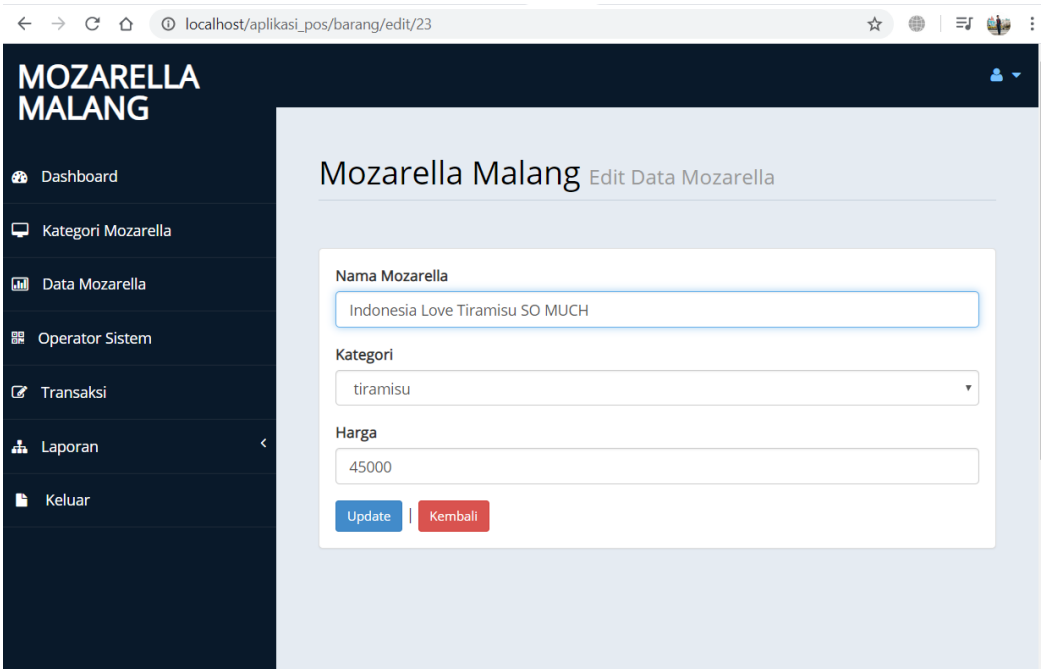
In this page we can manage the data of Mozarella, we can edit or update, delete and make a new data. If click ‘**Tambah Data**’ it will go to new data of Mozarella



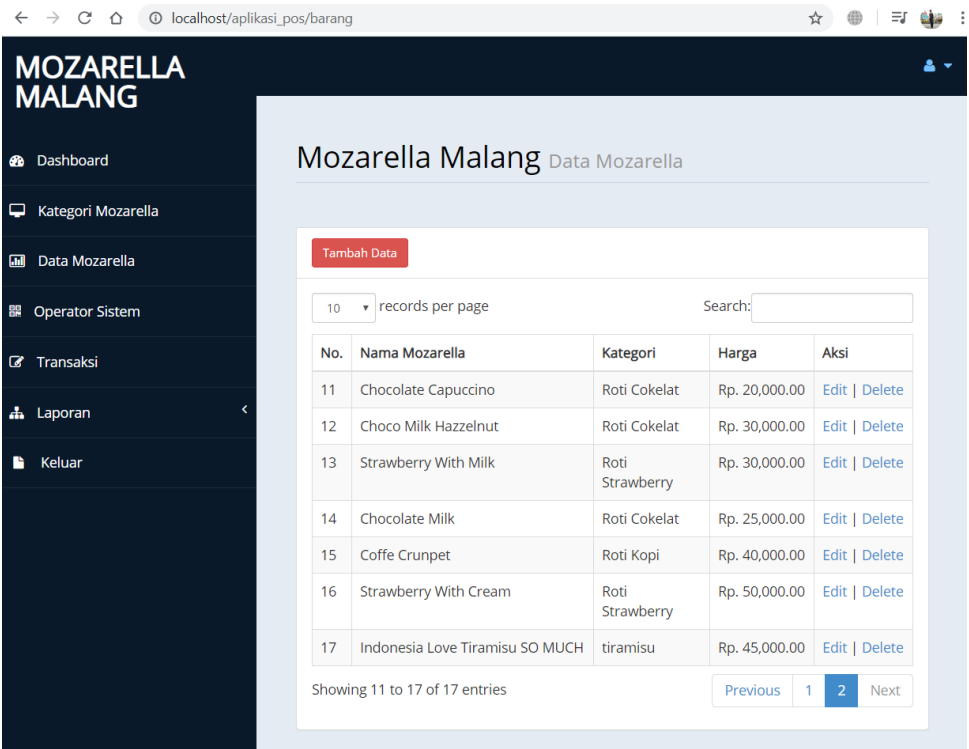
We input the new data of mozzarella by input the name, select the category that we have been input before, and then input the price, after that click save we will see the data that we have been created to Data Mozarella, the application will automatically direct to Data mozzarella again if we have save the new data.



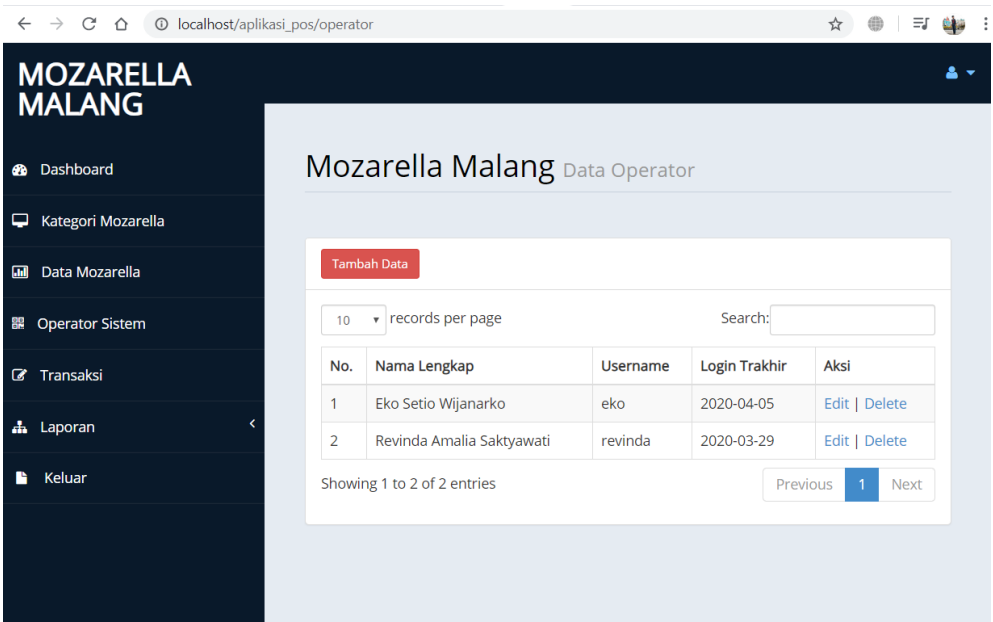
If we want to edit the data of mozzarella we can click edit in a action column of a data.



Also like new entry data, we can change the name, category, and also price. For example we update the name of data and price. It will automatically again direct to Data Mozarella page after we save the edit or update changes.



Then we go to the operator of system page, this is to manage the data of operator that have access to this web application.



We go to transaction page, it also manage the transaction of selling Mozarella Cake, we record the name of Mozarella cake then input the quantity of Cake that have been sell, system will automatically count the total price of each new entry.

