**KONEKSI**

var mysql = require('mysql');

const con = mysql.createConnection({

    host : "localhost",

    user : "moshi",

    password : "kartini67",

    database : "nodeapi"

})

con.connect((err)=>{

    if(err)throw err

    else console.log("Connection Success");

});

module.exports = con;

**ROUTE**

const express = require('express');

var app = express.Router();

var myjson = require('./controller');

app.route('/').get(myjson.index);

app.route('/tampil').get(myjson.showData);

app.route('/tampil/:id').get(myjson.showDataId);

app.route('/mahasiswa').post(myjson.addData);

app.route('/mahasiswa/:id').put(myjson.editData);

app.route('/mahasiswa/delete/:id').delete(myjson.deleteData);

app.route('/shownested').get(myjson.nestedJSON);

module.exports = app;

**CONTROLLER**

var response = require('./res');

var con = require('./koneksi');

exports.index = (req,res)=>{

    response.ok("Aplication is working",res);

}

exports.showData = (req,res) => {

    con.query("SELECT \* FROM MAHASISWA",(err,rows,field)=> {

        if(err) throw err;

        else

            response.ok(rows,res);

    });

}

**RES**

exports.ok = function(value,res){

    var data = {

        'status' : 200,

        'value' : value

    }

     res.json(data);

     res.end();

}

**AUTHORIZATION & JWT**

**Secret key**

module.exports ={

    'secret' : 'mysecret'

}

**SIGN UP MODULE**

exports.registrasi = (req,res)=>{

    var pos = {

        username: req.body.username,

        email : req.body.email,

        password : md5(req.body.password),

        role : req.body.role,

        tanggal\_daftar : new Date()

    }

    var sqlstmt = "SELECT EMAIL FROM user WHERE EMAIL = ?";

    con.query(sqlstmt,[pos.email],(err,rows,field)=>{

        if (err) throw err

        else

        {

            if(rows.length==0)

            {

                sqlstmt = `INSERT INTO user (USERNAME,EMAIL,PASSWORD,ROLE,TANGGAL\_DAFTAR)

                VALUES(?,?,?,?,?)`;

                con.query(sqlstmt,[pos.username,pos.email,pos.password,pos.role,pos.tanggal\_daftar],(err,res1)=>{

                    if(err) throw err

                    else response.ok("Data Inserted",res);

                })

            }

            else{

                response.ok("Data Already Exist",res)

            }

        }

    })

}

**LOG IN & JWT GENERATE**

exports.login= (req,res) => {

    var post = {

        email : req.body.email,

        password : md5(req.body.password)

    }

    var sqlstmt = "SELECT \* FROM user WHERE EMAIL = ? AND PASSWORD = ?";

    con.query(sqlstmt,[post.email,post.password],(err,rows,field)=>{

        if(err) throw err

        else

        if(rows.length== 1)

        {

**var token = jwt.sign({rows},config.secret,{expiresIn : 1440})**

            id\_user = rows[0].id

            var data = {

                id\_user : id\_user,

                access\_token : token,

                ip\_address :ip.address()

            }

            sqlstmt = "INSERT INTO akses\_token(ID\_USER,ACCESS\_TOKEN,IP\_ADDRESS) VALUES(?,?,?)";

            con.query(sqlstmt,[data.id\_user,data.access\_token,data.ip\_address],(err1,res2)=>{

                if(err1) throw err1

                else{

                    res.json({

                        success : true,

                        message : "token has been generated",

                        token : token,

                        curruser : data.id\_user

                    })

                }

            })

        }

        else{

            res.json({

                success : "error",

                message : "Email / password Wrong",

            })

        }

    })

}

**AUTHENTICATION**

function verifikasi(role)

{

    return (req,res,next)=> {

**var tokenBearer = req.headers.authorization;**

        if(tokenBearer)

        {

**var token = tokenBearer.split(' ')[1];**

            console.log(token);

**jwt.verify(token,secret.secret,(err,decode)**=> {

                if(err){

                    return res.status(401).send({auth : false, message :"Not Authorized Token"});

                }

                else{

                    if(role == 2)

                    {

                        req.auth = decode;

                        next();

                    }

                    else{

                        return res.status(401).send({auth : false, message :"Not Authorized Roles"});

                    }

                }

            })

        }

        else

        {

            return res.status(401).send({auth : false, message :"You Have No Token"});

        }

    }

}

module.exports = verifikasi;