Pretest Framework



Yusuf Rahmdhani A( 3122522030)

D3 Teknik Informatika

PSDKU Sumenep

PRODI D3 TEKNIK INFORMATIKA

DEPARTEMEN TEKNIK INFORMATIKA DAN KOMPUTER

PENS PSDKU SUMENEP

Alat Tangkap

Modul

var connection = require("../config/db");

class model\_alat\_tangkap{

    static async getALL(){

        return new Promise((*resolve*, *reject*) => {

            connection.query(

                "select \* from alat\_tangkap ORDER BY id\_alat\_tangkap DESC",

                function (*err*, *rows*) {

                  if (*err*) {

                    reject(*err*);

                  } else {

                    resolve(*rows*);

                  }

                }

              );

        });

    }

    static async Store(*Data*){

        return new Promise((*resolve*, *reject*) => {

            connection.query('insert into alat\_tangkap set ?', *Data*, function(*err*, *result*){

                if (*err*){

                    reject(*err*);

                    console.log(*err*);

                } else {

                    resolve(*result*);

                }

            });

        });

    }

    static async getId(*id*){

        return new Promise((*resolve*, *reject*) => {

            connection.query('select \* from alat\_tangkap where id\_alat\_tangkap = ' + *id* , (*err*, *rows*) => {

                if (*err*){

                    reject(*err*);

                    console.log(*err*);

                } else {

                    resolve(*rows*);

                }

            });

        });

    }

      static async update(*id*, *data*){

          return new Promise((*resolve*, *reject*) => {

              connection.query('update alat\_tangkap set ? where id\_alat\_tangkap =' + *id*, *data*, function(*err*, *result*){

                  if (*err*){

                      reject(*err*);

                      console.log(*err*);

                  } else {

                      resolve(*result*);

                  }

              });

          });

      }

      static async delete(*id*, *data*){

          return new Promise((*resolve*, *reject*) => {

              connection.query('delete from alat\_tangkap where id\_alat\_tangkap =' + *id*, function(*err*, *result*){

                  if (*err*){

                      reject(*err*);

                  } else {

                      resolve(result);

                  }

              });

          });

      }

}

module.exports = model\_alat\_tangkap;

Routes

var express = require("express");

var router = express.Router();

*// var connection = require("../config/db.js");*

const e = require("method-override");

const model\_alat\_tangkap = require("../model/model\_alat\_tangkap.js");

router.get("/", async function (*req*, *res*, *next*) {

  let rows = await model\_alat\_tangkap.getALL();

*res*.render("alat\_tangkap/index", {

    data: rows,

  });

});

router.get("/create", function (*req*, *res*, *next*) {

*res*.render("alat\_tangkap/create", {

    nama\_alat\_tangkap: "",

  });

});

router.post("/store", async function (*req*, *res*, *next*) {

  try {

    let { nama\_alat\_tangkap } = *req*.body;

    let Data = {

      nama\_alat\_tangkap,

    };

    await model\_alat\_tangkap.Store(Data);

*req*.flash("success", "Berhasil memperbarui data!");

*res*.redirect("/alat\_tangkap");

  } catch {

*res*.redirect("/alat\_tangkap");

  }

});

router.get("/edit/(:id)", async function (*req*, *res*, *next*) {

  let id = *req*.params.id;

  let rows = await model\_alat\_tangkap.getId(id);

*res*.render("alat\_tangkap/edit", {

    id: rows[0].id\_alat\_tangkap,

    nama\_alat\_tangkap: rows[0].nama\_alat\_tangkap,

  });

});

router.post("/update/:id", async function (*req*, *res*, *next*) {

  try {

    let id = *req*.params.id;

    let { nama\_alat\_tangkap } = *req*.body;

    let Data = {

      nama\_alat\_tangkap,

    };

    await model\_alat\_tangkap.update(id, Data);

*req*.flash("success", "Berhasil memperbarui data!");

*res*.redirect("/alat\_tangkap");

  } catch {

*res*.redirect("/alat\_tangkap");

  }

});

router.get("/delete/:id", async function (*req*, *res*) {

  let id = *req*.params.id;

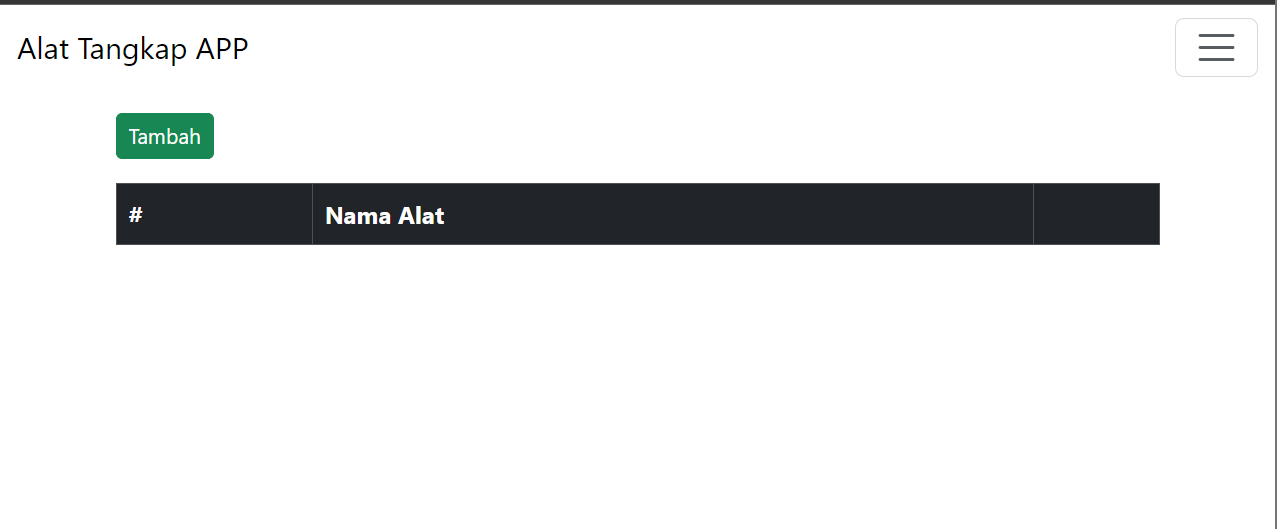
  await model\_alat\_tangkap.delete(id);

*res*.redirect("/alat\_tangkap");

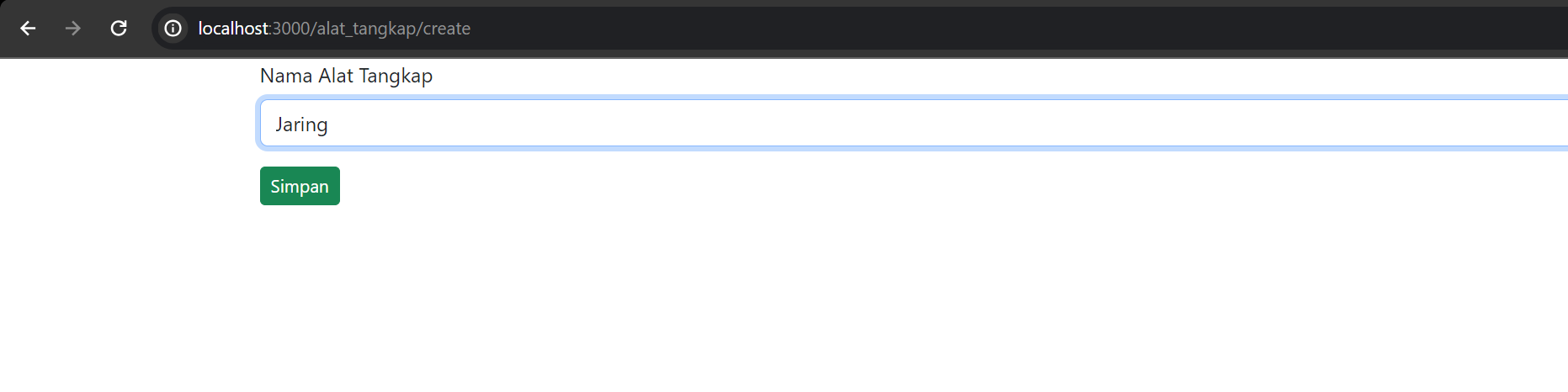
});

module.exports = router;

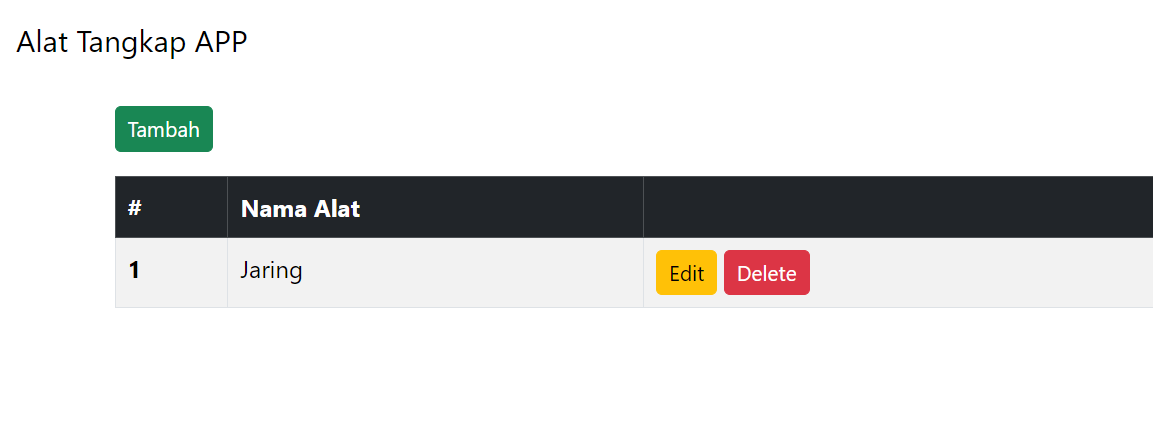
index



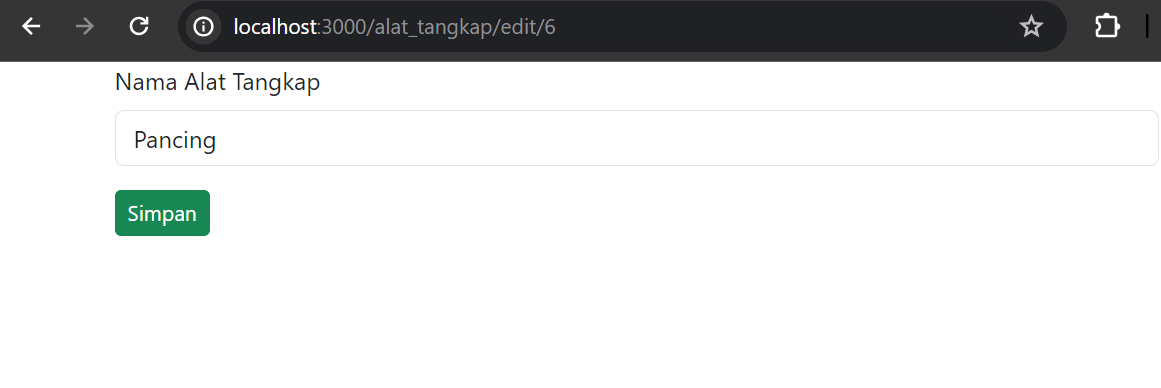
Create



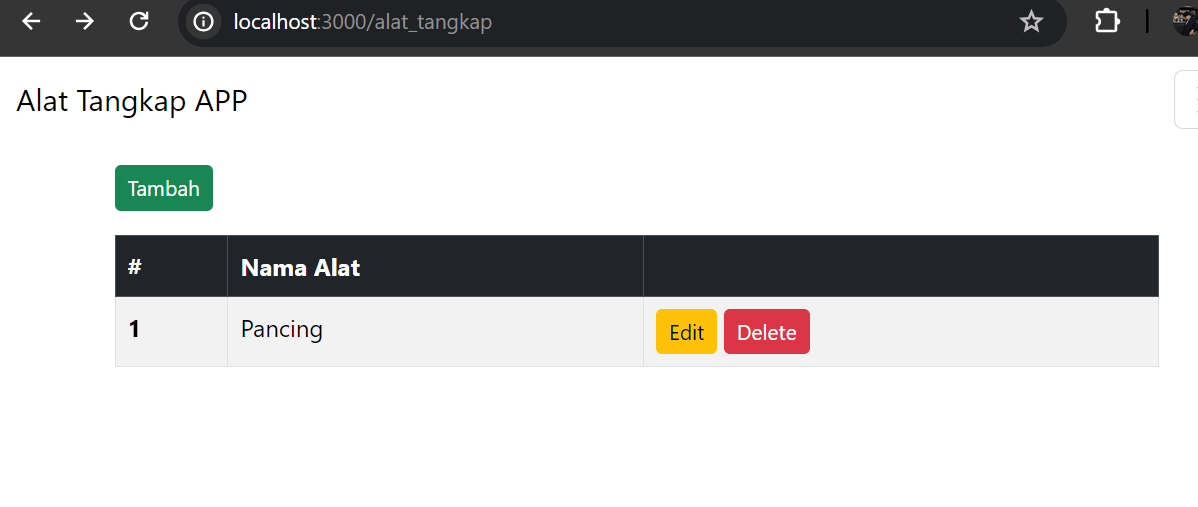
Data sesudah di tambah



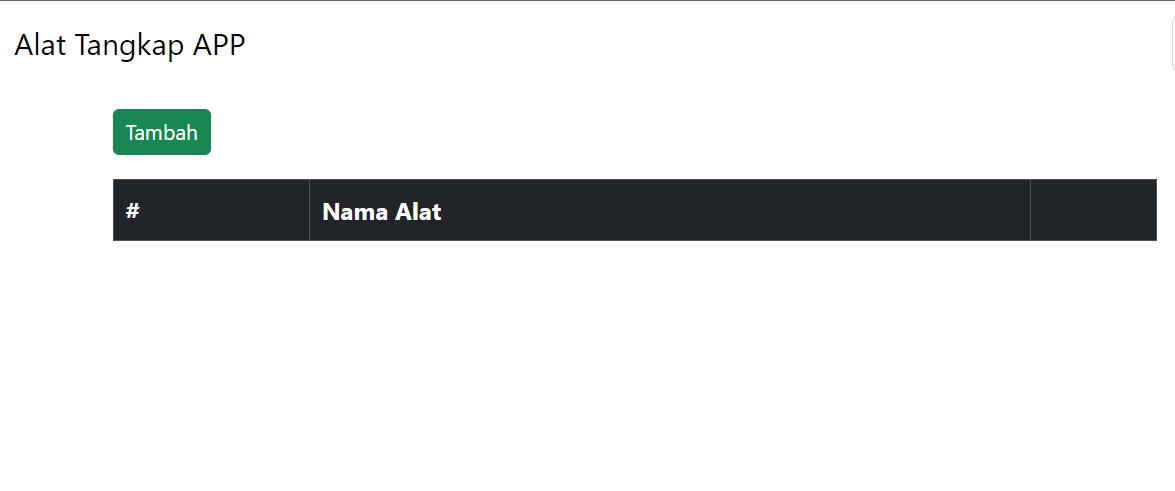
Edit



Data sesudah di edit



Data sesudah didelete

Dpi

Modul

var connection = require("../config/db");

class model\_dpi{

    static async getALL(){

        return new Promise((*resolve*, *reject*) => {

            connection.query(

                "select \* from dpi ORDER BY id\_dpi DESC",

                function (*err*, *rows*) {

                  if (*err*) {

                    reject(*err*);

                  } else {

                    resolve(*rows*);

                  }

                }

              );

        });

    }

    static async Store(*Data*){

        return new Promise((*resolve*, *reject*) => {

            connection.query('insert into dpi set ?', *Data*, function(*err*, *result*){

                if (*err*){

                    reject(*err*);

                    console.log(*err*);

                } else {

                    resolve(*result*);

                }

            });

        });

    }

    static async getId(*id*){

        return new Promise((*resolve*, *reject*) => {

            connection.query('select \* from dpi where id\_dpi = ' + *id* , (*err*, *rows*) => {

                if (*err*){

                    reject(*err*);

                    console.log(*err*);

                } else {

                    resolve(*rows*);

                }

            });

        });

    }

      static async update(*id*, *data*){

          return new Promise((*resolve*, *reject*) => {

              connection.query('update dpi set ? where id\_dpi =' + *id*, *data*, function(*err*, *result*){

                  if (*err*){

                      reject(*err*);

                      console.log(*err*);

                  } else {

                      resolve(*result*);

                  }

              });

          });

      }

      static async delete(*id*, *data*){

          return new Promise((*resolve*, *reject*) => {

              connection.query('delete from dpi where id\_dpi =' + *id*, function(*err*, *result*){

                  if (*err*){

                      reject(*err*);

                  } else {

                      resolve(*result*);

                  }

              });

          });

      }

}

module.exports = model\_dpi;

Routes

var express = require("express");

var router = express.Router();

*// var connection = require("../config/db.js");*

const e = require("method-override");

const model\_dpi = require("../model/model\_dpi.js");

router.get("/", async function (*req*, *res*, *next*) {

  let rows = await model\_dpi.getALL();

*res*.render("dpi/index", {

    data: rows,

  });

});

router.get("/create", function (*req*, *res*, *next*) {

*res*.render("dpi/create", {

    nama\_dpi: "",

  });

});

router.post("/store", async function (*req*, *res*, *next*) {

  try {

    let { nama\_dpi } = *req*.body;

    let { luas } = *req*.body;

    let Data = {

      nama\_dpi,

      luas,

    };

    await model\_dpi.Store(Data);

*req*.flash("success", "Berhasil memperbarui data!");

*res*.redirect("/dpi");

  } catch {

*res*.redirect("/dpi");

  }

});

router.get("/edit/(:id)", async function (*req*, *res*, *next*) {

  let id = *req*.params.id;

  let rows = await model\_dpi.getId(id);

*res*.render("dpi/edit", {

    id: rows[0].id\_dpi,

    nama\_dpi: rows[0].nama\_dpi,

    luas: rows[0].luas,

  });

});

router.post("/update/:id", async function (*req*, *res*, *next*) {

  try {

    let id = *req*.params.id;

    let { nama\_dpi } = *req*.body;

    let { luas } = *req*.body;

    let Data = {

      nama\_dpi,

      luas,

    };

    await model\_dpi.update(id, Data);

*req*.flash("success", "Berhasil memperbarui data!");

*res*.redirect("/dpi");

  } catch {

*res*.redirect("/dpi");

  }

});

router.get("/delete/:id", async function (*req*, *res*) {

  let id = *req*.params.id;

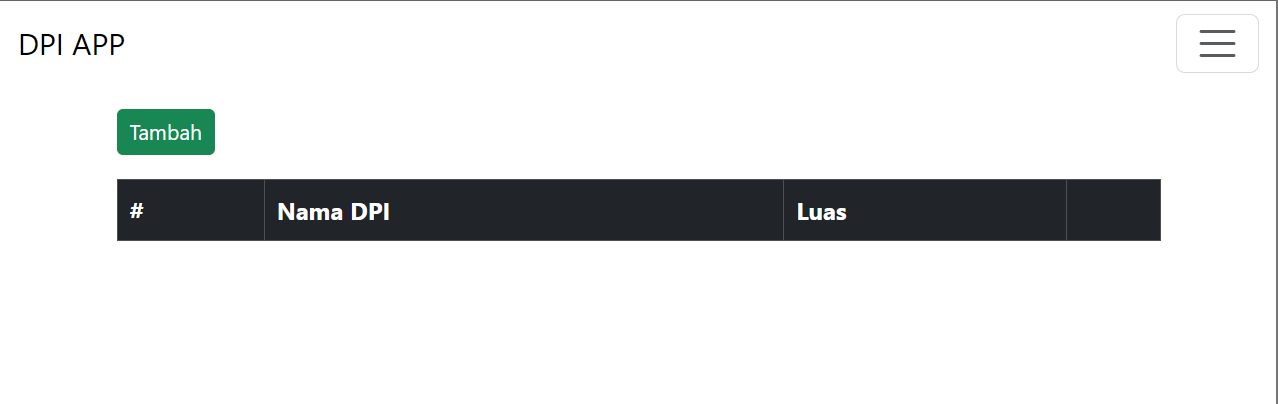
  await model\_dpi.delete(id);

*res*.redirect("/dpi");

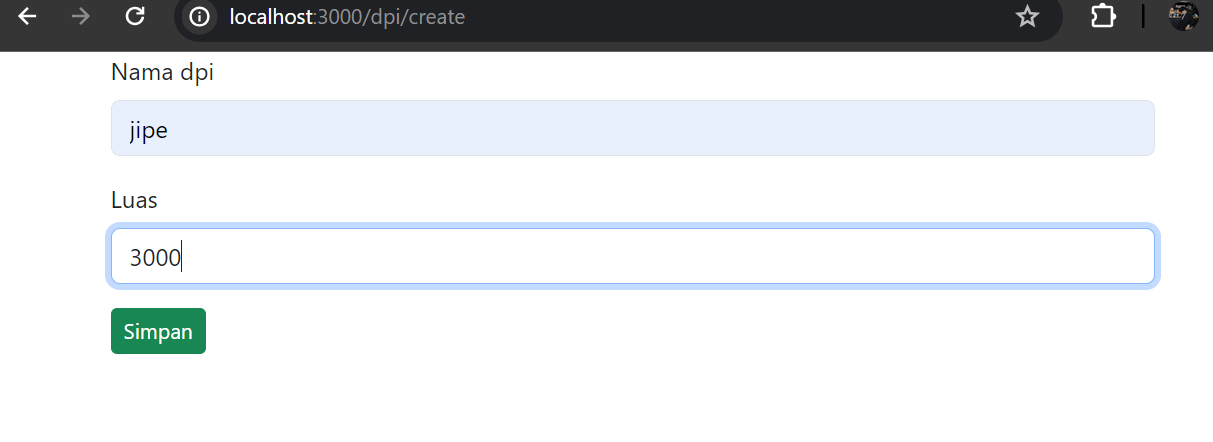
});

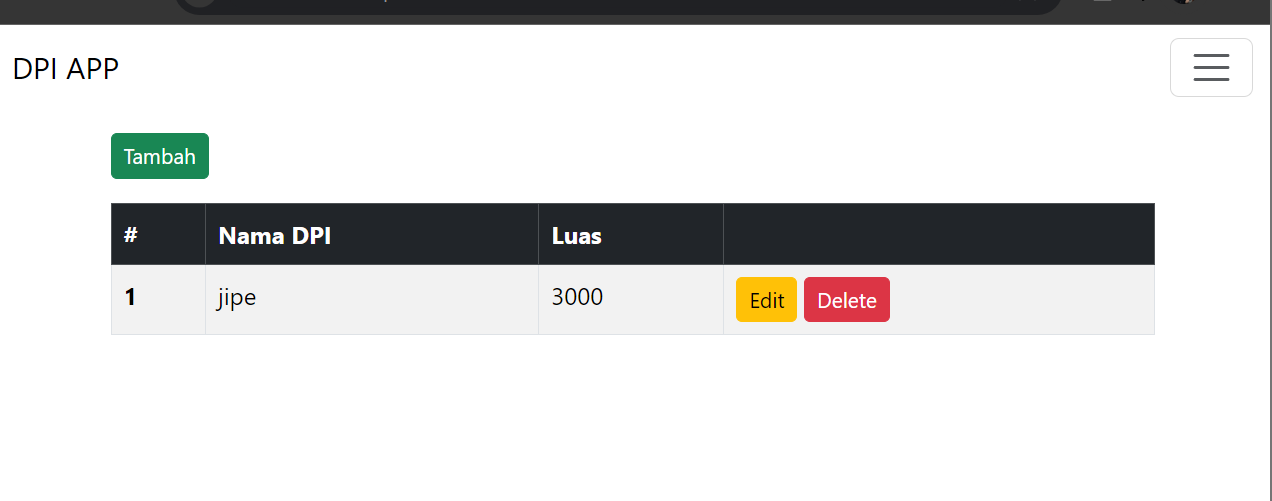
module.exports = router;

Index

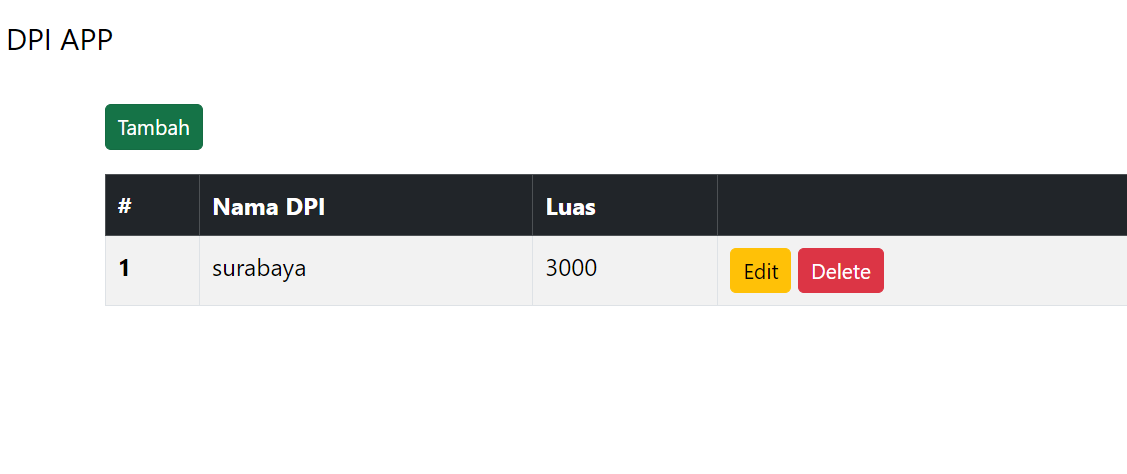


Create





Edit



Delete

