cover

Ach. Eko Darmawan

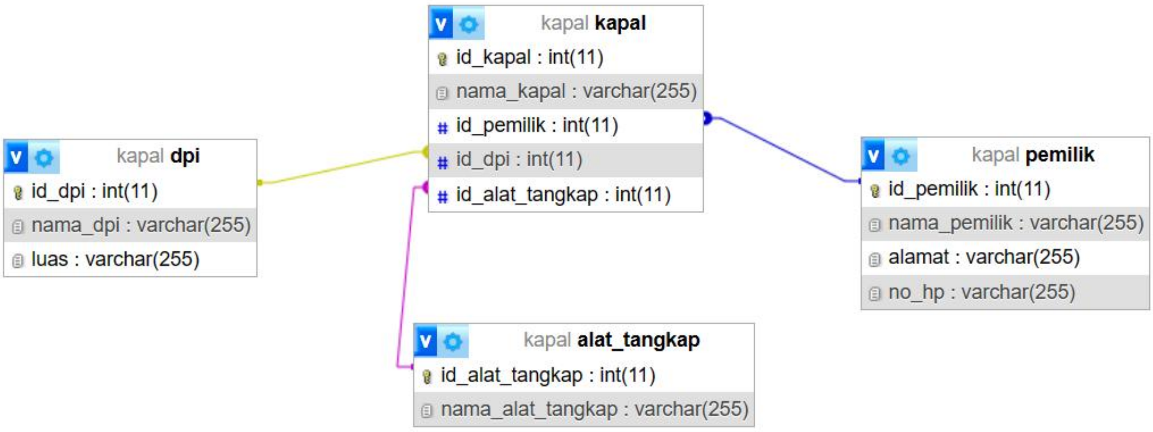
3122522019

TI D3 PSDKU Sumenep

PRODI D3 TEKNIK INFORMATIKA

DEPARTEMEN TEKNIK INFORMATIKA DAN KOMPUTER

PENS PSDKU SUMENEP



kapal **dpi**
- id_dpi : int(11)
- nama_dpi : varchar(255)
- luas : varchar(255)

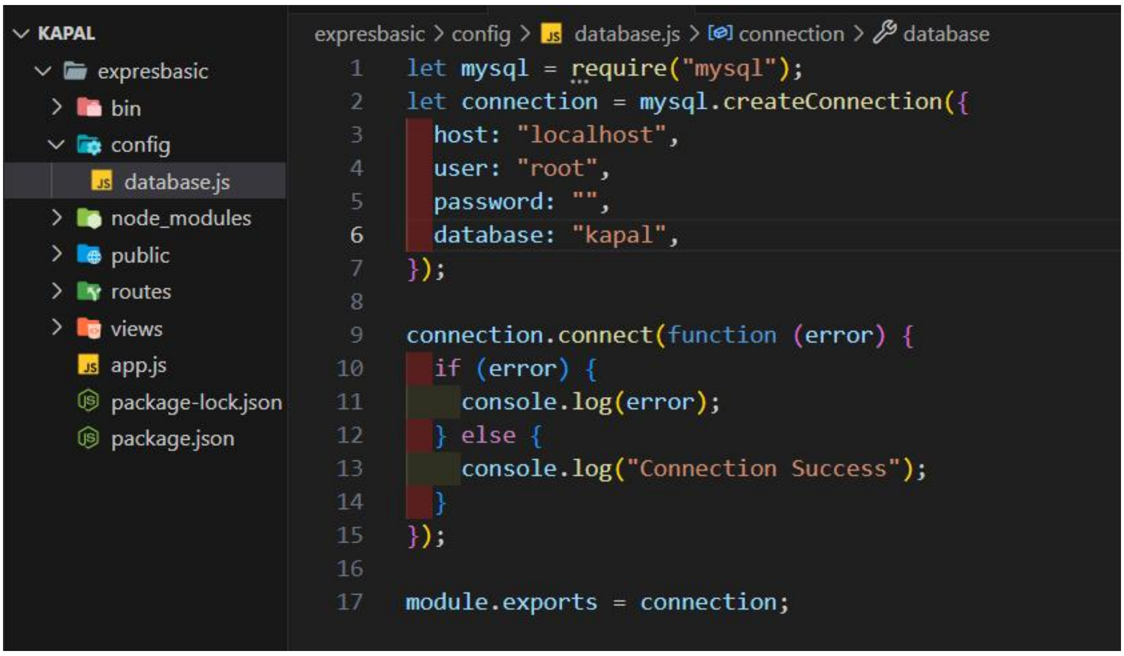
kapal **kapal**
- id_kapal : int(11)
- nama_kapal : varchar(255)
- id_pemilik : int(11)
- id_dpi : int(11)
- id_alat_tangkap : int(11)

kapal **pemilik**
- id_pemilik : int(11)
- nama_pemilik : varchar(255)
- alamat : varchar(255)
- no_hp : varchar(255)

kapal **alat_tangkap**
- id_alat_tangkap : int(11)
- nama_alat_tangkap : varchar(255)

Ini untuk tabelnya

Konfigurasi Koneksi Database



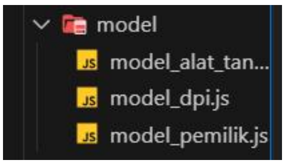
```
KAPAL
  expresbasic
    bin
    config
      database.js
    node_modules
    public
    routes
    views
    app.js
    package-lock.json
    package.json
```

expresbasic > config > database.js > connection > database

```
let mysql = require("mysql");
let connection = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "",
  database: "kapal",
});

connection.connect(function (error) {
  if (error) {
    console.log(error);
  } else {
    console.log("Connection Success");
  }
});

module.exports = connection;
```

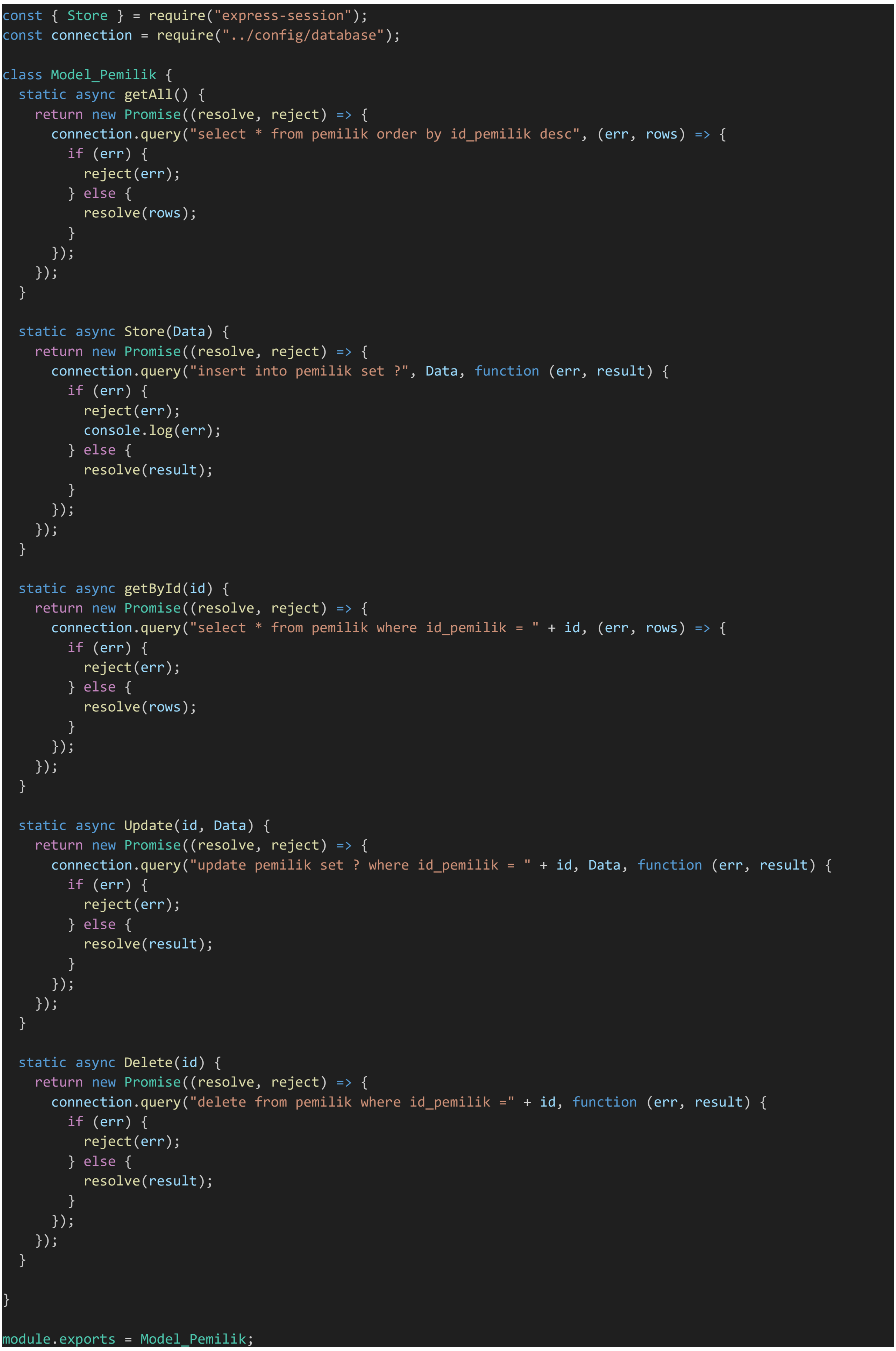
Membuat Direktori Model

model merupakan sebuah fungsi pada konsep pemrograman yang digunakan untuk mengatur data, serta berinteraksi langsung dengan tabel di dalam database. pada express kita bisa menggunakan model. Buat direktori dengan nama model letakkan di direktori utama



```
model
  model_alat_tan...
  model_dpi.js
  model_pemilik.js
```

Ini isi dari model_pemilik.js



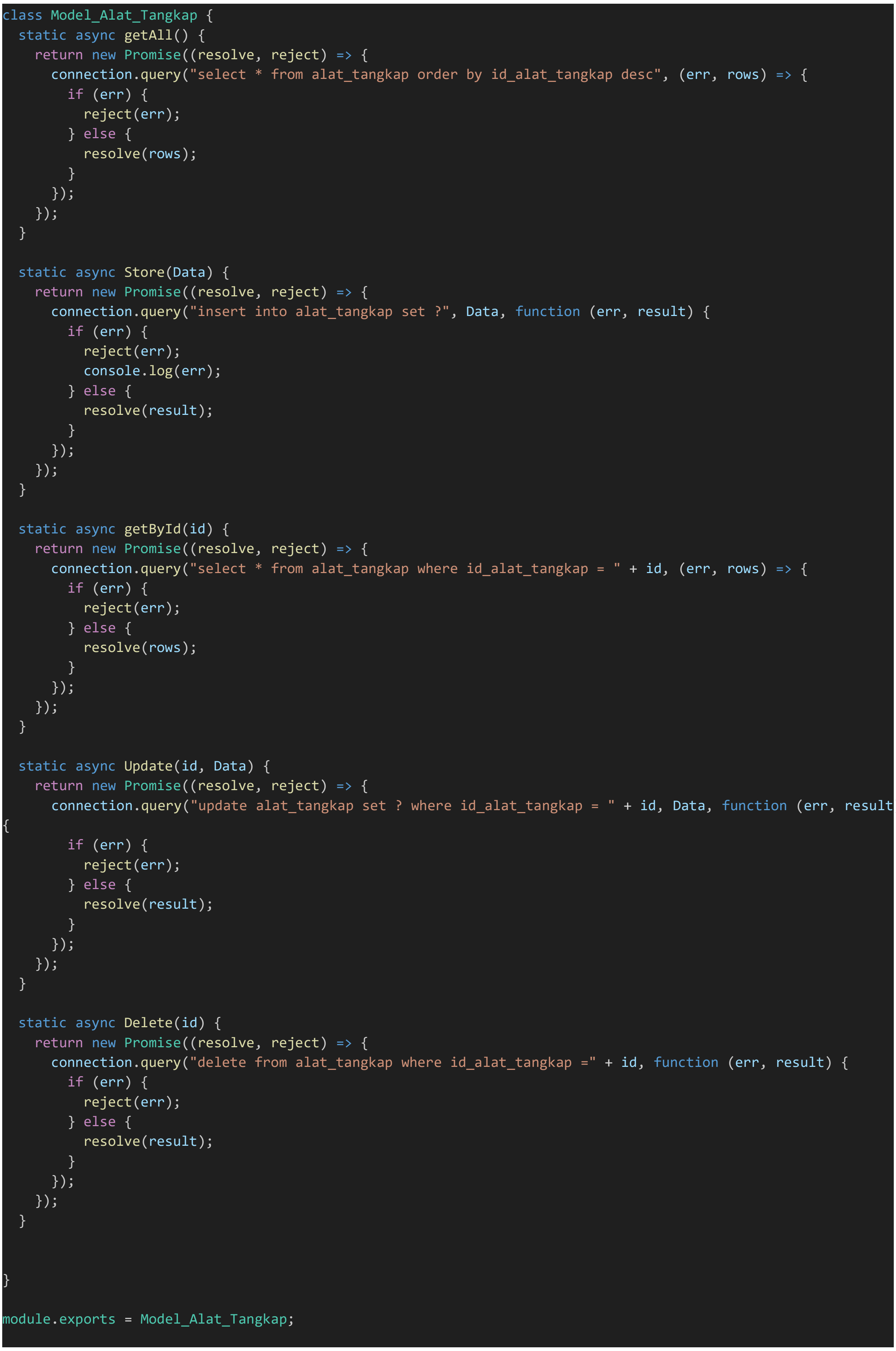
```
const { Store } = require("express-session");
const connection = require("../config/database");

class Model_Pemilik {
  static async getAll() {
    return new Promise((resolve, reject) => {
      connection.query("select * from pemilik order by id_pemilik desc", (err, rows) => {
        if (err) {
          reject(err);
        } else {
          resolve(rows);
        }
      });
    });
  }

  static async Store(Data) {
    return new Promise((resolve, reject) => {
      connection.query("insert into pemilik set ?", Data, function (err, result) {
        if (err) {
          reject(err);
          console.log(err);
        } else {
          resolve(result);
        }
      });
    });
  }

  static async getById(id) {
    return new Promise((resolve, reject) => {
      connection.query("select * from pemilik where id_pemilik = " + id, (err, rows) => {
        if (err) {
          reject(err);
        } else {
          resolve(rows);
        }
      });
    });
  }

  static async Update(id, Data) {
    return new Promise((resolve, reject) => {
      connection.query("update pemilik set ? where id_pemilik = " + id, Data, function (err, result) {
        if (err) {
          reject(err);
        } else {
          resolve(result);
        }
      });
    });
  }

  static async Delete(id) {
    return new Promise((resolve, reject) => {
      connection.query("delete from pemilik where id_pemilik =" + id, function (err, result) {
        if (err) {
          reject(err);
        } else {
          resolve(result);
        }
      });
    });
  }

}

module.exports = Model_Pemilik;
```

Ini isi dari model_dpi.js

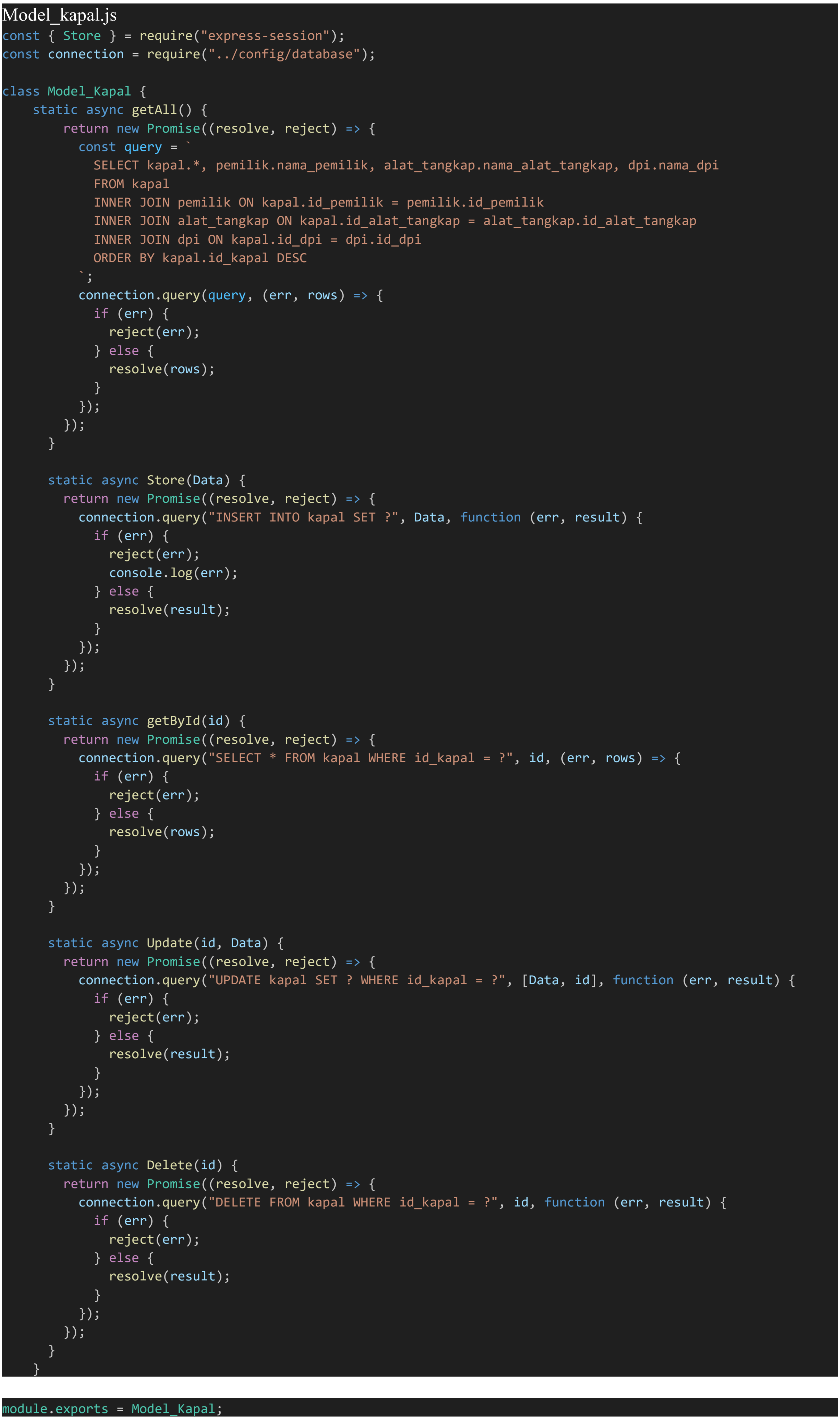
```
const { Store } = require("express-session");
const connection = require("../config/database");

class Model_Dpi {
  static async getAll() {
    return new Promise((resolve, reject) => {
      connection.query("select * from dpi order by id_dpi desc", (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 getById(id) {
    return new Promise((resolve, reject) => {
      connection.query("select * from dpi where id_dpi = " + id, (err, rows) => {
        if (err) {
          reject(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);
        } else {
          resolve(result);
        }
      });
    });
  }

  static async Delete(id) {
    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;
```

Dan ini dari model_alat_tangkap.js

```
const { Store } = require("express-session");
const connection = require("../config/database");
```



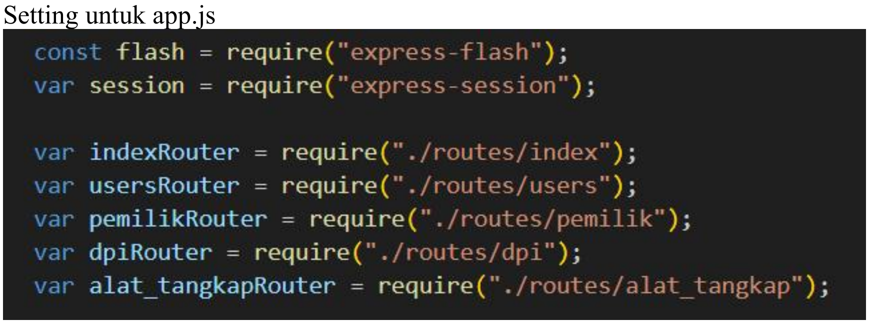
```
class Model_Alat_Tangkap {
  static async getAll() {
    return new Promise((resolve, reject) => {
      connection.query("select * from alat_tangkap order by id_alat_tangkap desc", (err, rows) => {
        if (err) {
          reject(err);
        } else {
          resolve(rows);
        }
      });
    });
  }

  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 getById(id) {
    return new Promise((resolve, reject) => {
      connection.query("select * from alat_tangkap where id_alat_tangkap = " + id, (err, rows) => {
        if (err) {
          reject(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);
        } else {
          resolve(result);
        }
      });
    });
  }

  static async Delete(id) {
    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;
```



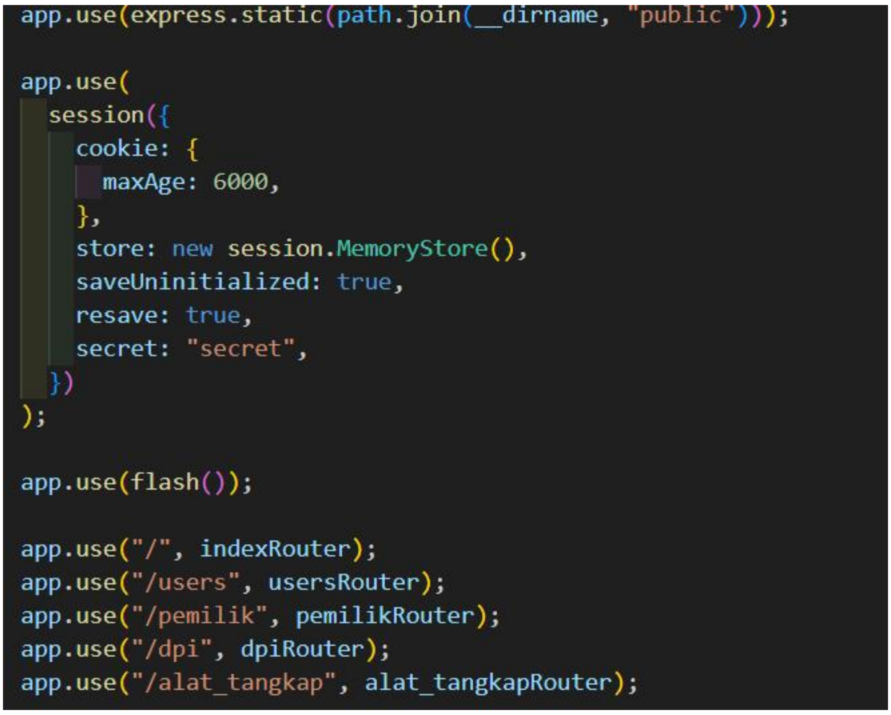
Model_kapal.js

```
const { Store } = require("express-session");
const connection = require("../config/database");

class Model_Kapal {
    static async getAll() {
        return new Promise((resolve, reject) => {
          const query = `
            SELECT kapal.*, pemilik.nama_pemilik, alat_tangkap.nama_alat_tangkap, dpi.nama_dpi
            FROM kapal
            INNER JOIN pemilik ON kapal.id_pemilik = pemilik.id_pemilik
            INNER JOIN alat_tangkap ON kapal.id_alat_tangkap = alat_tangkap.id_alat_tangkap
            INNER JOIN dpi ON kapal.id_dpi = dpi.id_dpi
            ORDER BY kapal.id_kapal DESC
          `;
          connection.query(query, (err, rows) => {
            if (err) {
              reject(err);
            } else {
              resolve(rows);
            }
          });
        });
      }

      static async Store(Data) {
        return new Promise((resolve, reject) => {
          connection.query("INSERT INTO kapal SET ?", Data, function (err, result) {
            if (err) {
              reject(err);
              console.log(err);
            } else {
              resolve(result);
            }
          });
        });
      }

      static async getById(id) {
        return new Promise((resolve, reject) => {
          connection.query("SELECT * FROM kapal WHERE id_kapal = ?", id, (err, rows) => {
            if (err) {
              reject(err);
            } else {
              resolve(rows);
            }
          });
        });
      }

      static async Update(id, Data) {
        return new Promise((resolve, reject) => {
          connection.query("UPDATE kapal SET ? WHERE id_kapal = ?", [Data, id], function (err, result) {
            if (err) {
              reject(err);
            } else {
              resolve(result);
            }
          });
        });
      }

      static async Delete(id) {
        return new Promise((resolve, reject) => {
          connection.query("DELETE FROM kapal WHERE id_kapal = ?", id, function (err, result) {
            if (err) {
              reject(err);
            } else {
              resolve(result);
            }
          });
        });
      }
    }

module.exports = Model_Kapal;
```

Setting untuk app.js



```
const flash = require("express-flash");
var session = require("express-session");

var indexRouter = require("./routes/index");
var usersRouter = require("./routes/users");
var pemilikRouter = require("./routes/pemilik");
var dpiRouter = require("./routes/dpi");
var alat_tangkapRouter = require("./routes/alat_tangkap");
```



```
app.use(express.static(path.join(__dirname, "public")));

app.use(
  session({
    cookie: {
      maxAge: 6000,
    },
    store: new session.MemoryStore(),
    saveUninitialized: true,
    resave: true,
    secret: "secret",
  })
);

app.use(flash());

app.use("/", indexRouter);
app.use("/users", usersRouter);
app.use("/pemilik", pemilikRouter);
app.use("/dpi", dpiRouter);
app.use("/alat_tangkap", alat_tangkapRouter);
```

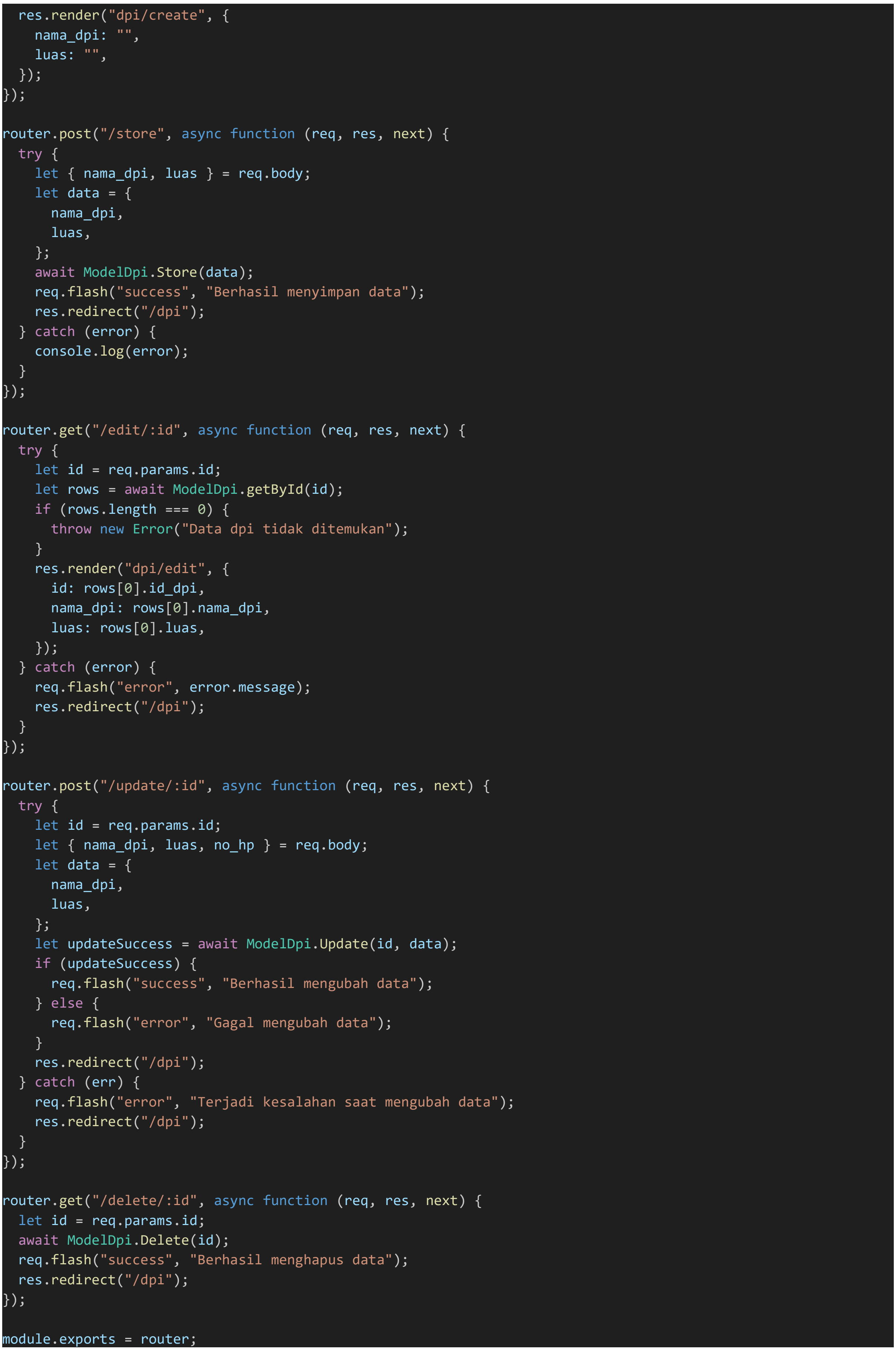
Ini untuk routesnya dari semua

Pemilik.ejs

```
var express = require("express");
var router = express.Router();
var connection = require("../config/database.js");
const ModelPemilik = require("../model/model_pemilik.js");

router.get("/", async function (req, res, next) {
  try {
    let rows = await ModelPemilik.getAll();
    res.render("pemilik/index", {
      data: rows,
    });
  } catch (error) {
    req.flash("error", error.message);
    res.redirect("/pemilik");
  }
});

router.get("/create", function (req, res, next) {
  res.render("pemilik/create", {
    nama_pemilik: "",
    alamat: "",
    no_hp: "",
  });
});

router.post("/store", async function (req, res, next) {
  try {
    let { nama_pemilik, alamat, no_hp } = req.body;
    let data = {
      nama_pemilik,
      alamat,
      no_hp,
    };
    await ModelPemilik.Store(data);
    req.flash("success", "Berhasil menyimpan data");
    res.redirect("/pemilik");
  } catch (error) {
    console.log(error);
  }
});

router.get("/edit/:id", async function (req, res, next) {
  try {
    let id = req.params.id;
    let rows = await ModelPemilik.getById(id);
    res.render("pemilik/edit", {
      id: rows[0].id_pemilik,
      nama_pemilik: rows[0].nama_pemilik,
      alamat: rows[0].alamat,
      no_hp: rows[0].no_hp,
    });
  } catch (error) {
    req.flash("error", error.message);
    console.log(error);
    res.redirect("/pemilik");
  }
});

router.post("/update/:id", async function (req, res, next) {
  try {
    let id = req.params.id;
    let { nama_pemilik, alamat, no_hp } = req.body;
    let data = {
      nama_pemilik,
      alamat,
      no_hp,
    };
    let updateSuccess = await ModelPemilik.Update(id, data);
    if (updateSuccess) {
      req.flash("success", "Berhasil mengubah data");
    } else {
      req.flash("error", "Gagal mengubah data");
    }
    res.redirect("/pemilik");
  } catch (err) {
    req.flash("error", "Terjadi kesalahan saat mengubah data");
    res.redirect("/pemilik");
  }
});

router.get("/delete/:id", async function (req, res, next) {
  let id = req.params.id;
  await ModelPemilik.Delete(id);
  req.flash("success", "Berhasil menghapus data");
  res.redirect("/pemilik");
});

module.exports = router;
```

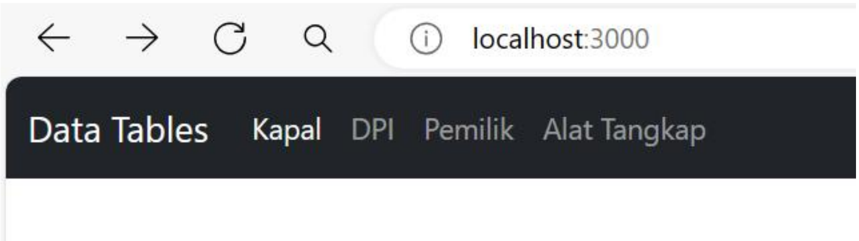
Dpi.ejs

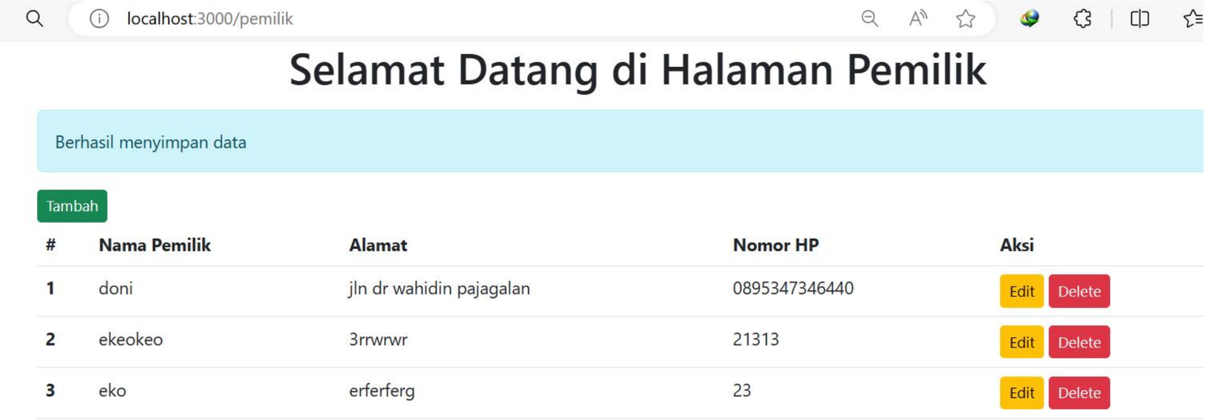
```
var express = require("express");
var router = express.Router();
var connection = require("../config/database.js");
const ModelDpi = require("../model/model_dpi.js");

router.get("/", async function (req, res, next) {
  try {
    let rows = await ModelDpi.getAll();
    res.render("dpi/index", {
      data: rows,
    });
  } catch (error) {
    req.flash("error", error.message);
    res.redirect("/dpi");
  }
});

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



```
  res.render("dpi/create", {
    nama_dpi: "",
    luas: "",
  });
});

router.post("/store", async function (req, res, next) {
  try {
    let { nama_dpi, luas } = req.body;
    let data = {
      nama_dpi,
      luas,
    };
    await ModelDpi.Store(data);
    req.flash("success", "Berhasil menyimpan data");
    res.redirect("/dpi");
  } catch (error) {
    console.log(error);
  }
});

router.get("/edit/:id", async function (req, res, next) {
  try {
    let id = req.params.id;
    let rows = await ModelDpi.getById(id);
    if (rows.length === 0) {
      throw new Error("Data dpi tidak ditemukan");
    }
    res.render("dpi/edit", {
      id: rows[0].id_dpi,
      nama_dpi: rows[0].nama_dpi,
      luas: rows[0].luas,
    });
  } catch (error) {
    req.flash("error", error.message);
    res.redirect("/dpi");
  }
});

router.post("/update/:id", async function (req, res, next) {
  try {
    let id = req.params.id;
    let { nama_dpi, luas, no_hp } = req.body;
    let data = {
      nama_dpi,
      luas,
    };
    let updateSuccess = await ModelDpi.Update(id, data);
    if (updateSuccess) {
      req.flash("success", "Berhasil mengubah data");
    } else {
      req.flash("error", "Gagal mengubah data");
    }
    res.redirect("/dpi");
  } catch (err) {
    req.flash("error", "Terjadi kesalahan saat mengubah data");
    res.redirect("/dpi");
  }
});

router.get("/delete/:id", async function (req, res, next) {
  let id = req.params.id;
  await ModelDpi.Delete(id);
  req.flash("success", "Berhasil menghapus data");
  res.redirect("/dpi");
});

module.exports = router;
```

Hasilnya



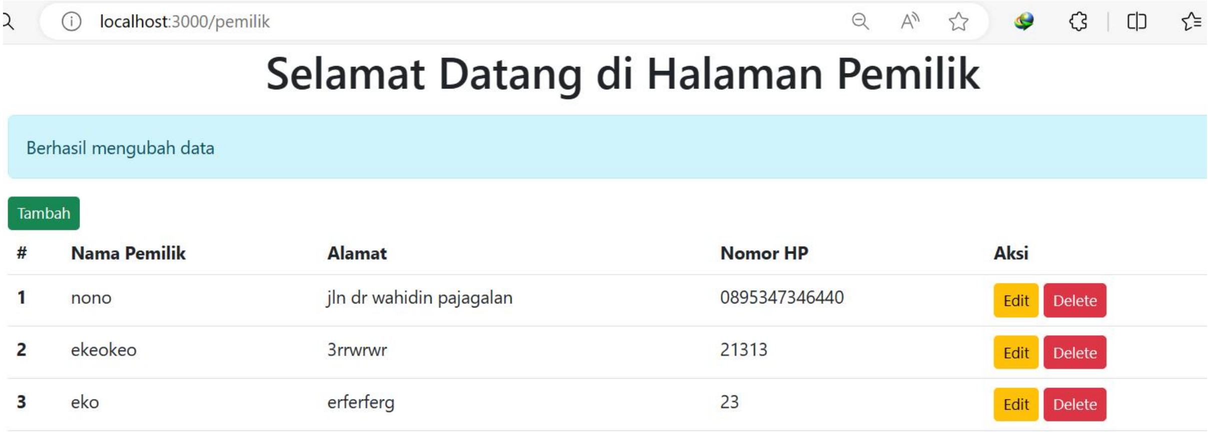


Selamat Datang di Halaman Pemilik

Berhasil menyimpan data

Tambah

#	Nama Pemilik	Alamat	Nomor HP	Aksi
1	doni	jln dr wahidin pajagalan	0895347346440	Edit Delete
2	ekeokeo	3rrwrwr	21313	Edit Delete
3	eko	erferferg	23	Edit Delete

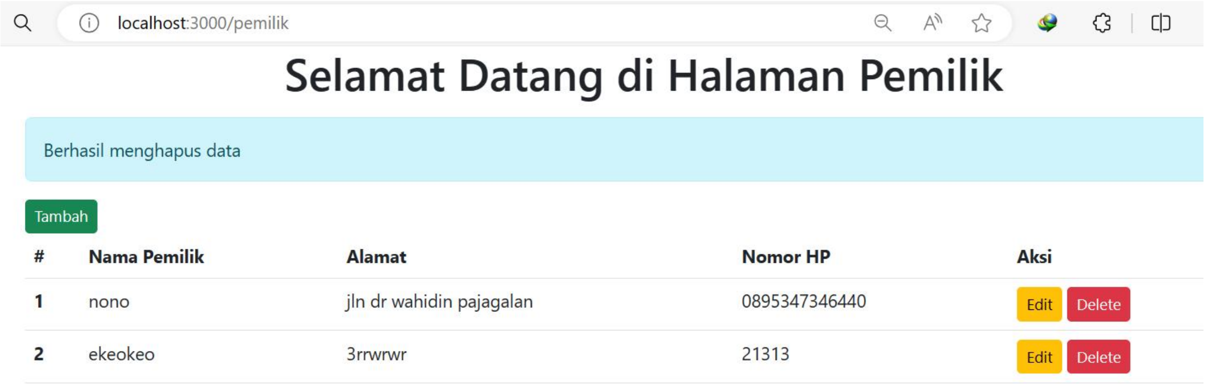


Selamat Datang di Halaman Pemilik

Berhasil mengubah data

Tambah

#	Nama Pemilik	Alamat	Nomor HP	Aksi
1	nono	jln dr wahidin pajagalan	0895347346440	Edit Delete
2	ekeokeo	3rrwrwr	21313	Edit Delete
3	eko	erferferg	23	Edit Delete

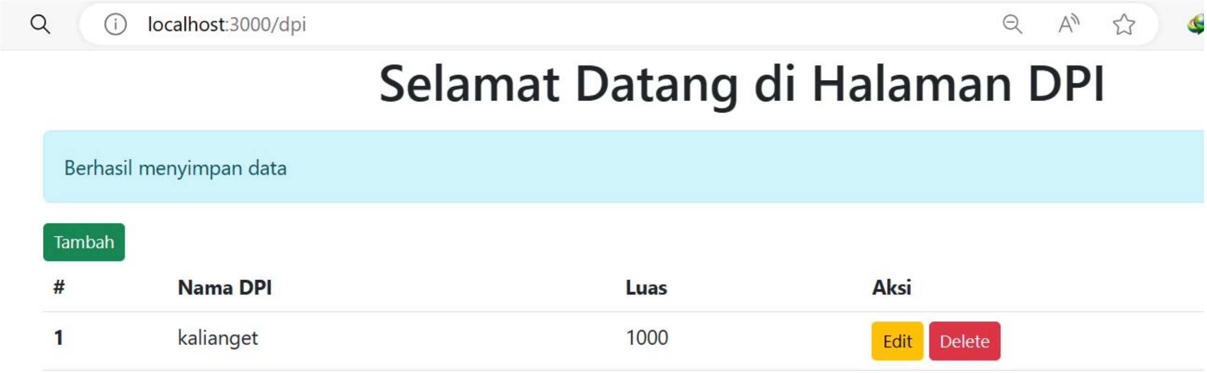


Selamat Datang di Halaman Pemilik

Berhasil menghapus data

Tambah

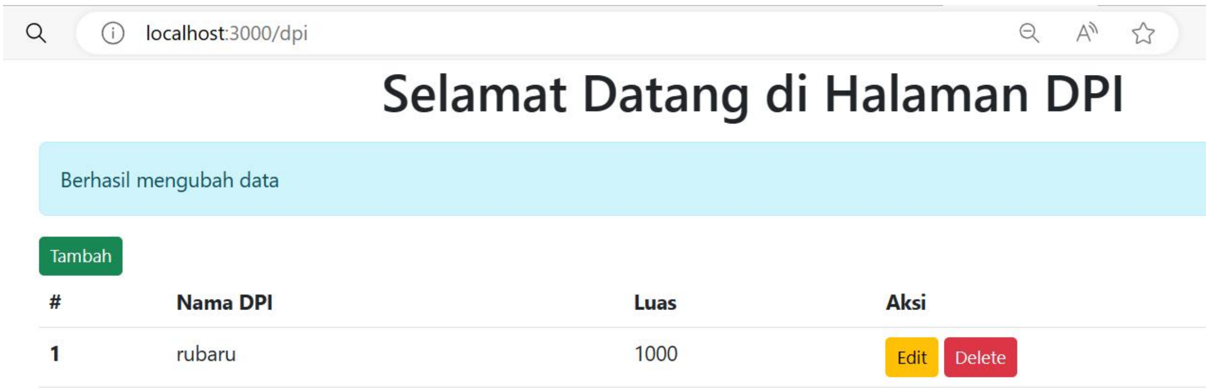
#	Nama Pemilik	Alamat	Nomor HP	Aksi
1	nono	jln dr wahidin pajagalan	0895347346440	Edit Delete
2	ekeokeo	3rrwrwr	21313	Edit Delete



Selamat Datang di Halaman DPI

Berhasil menyimpan data

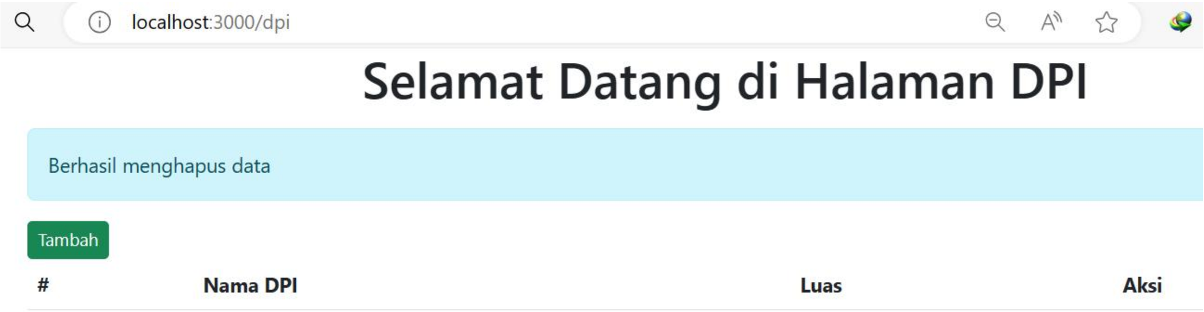
#	Nama DPI	Luas	Aksi
1	kalianget	1000	Edit Delete



Selamat Datang di Halaman DPI

Berhasil mengubah data

#	Nama DPI	Luas	Aksi
1	rubaru	1000	Edit Delete

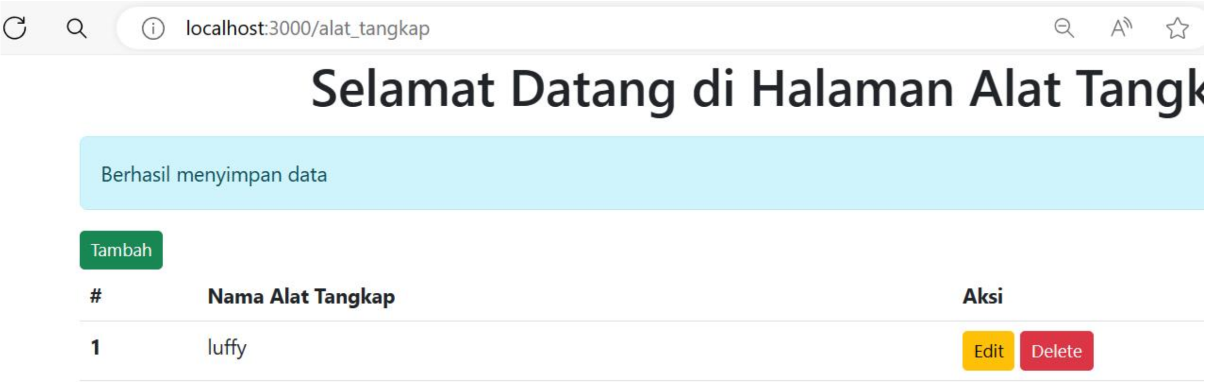


Selamat Datang di Halaman DPI

Berhasil menghapus data

#	Nama DPI	Luas	Aksi

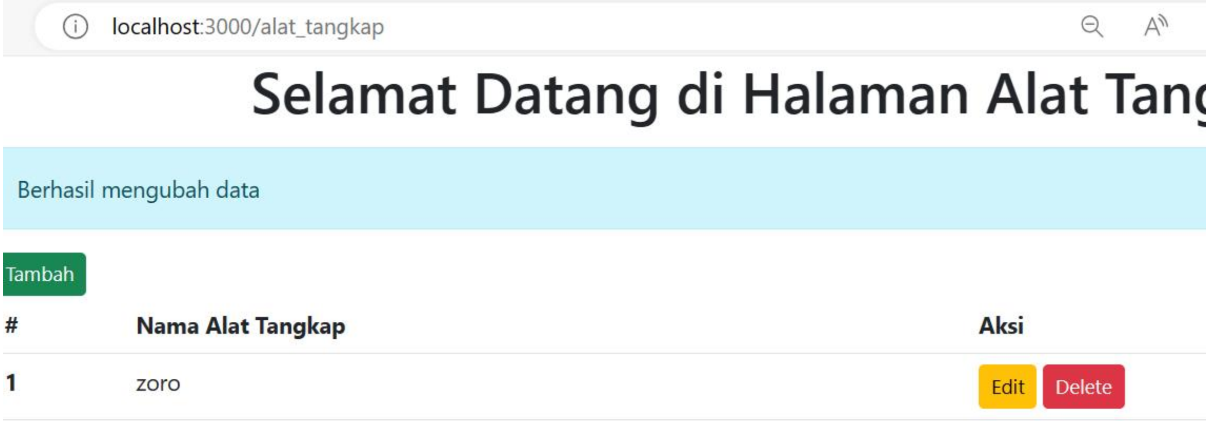
Alat_tangkap



Selamat Datang di Halaman Alat Tangk

Berhasil menyimpan data

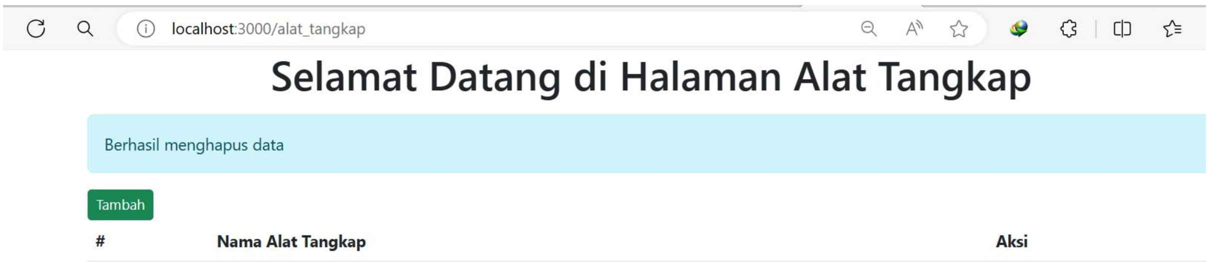
#	Nama Alat Tangkap	Aksi
1	luffy	Edit Delete



Selamat Datang di Halaman Alat Tang

Berhasil mengubah data

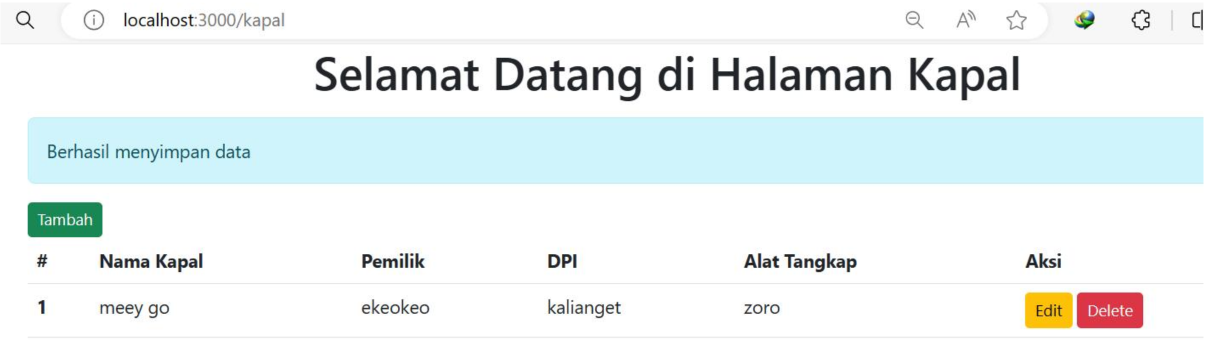
#	Nama Alat Tangkap	Aksi
1	zoro	Edit Delete



Selamat Datang di Halaman Alat Tangkap

Berhasil menghapus data

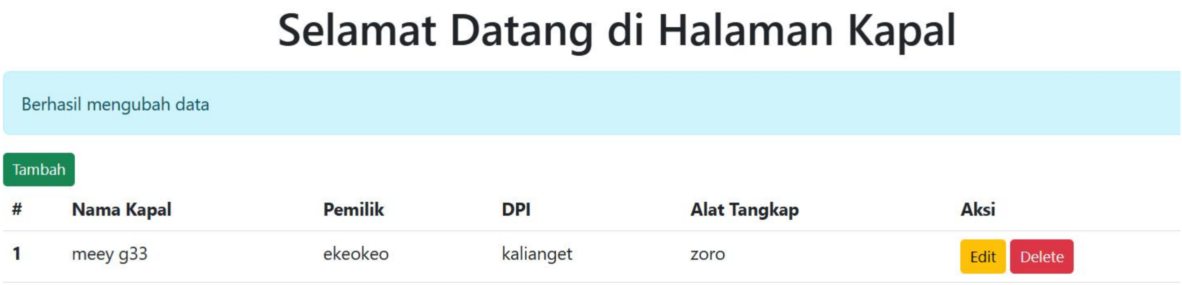
#	Nama Alat Tangkap	Aksi



Selamat Datang di Halaman Kapal

Berhasil menyimpan data

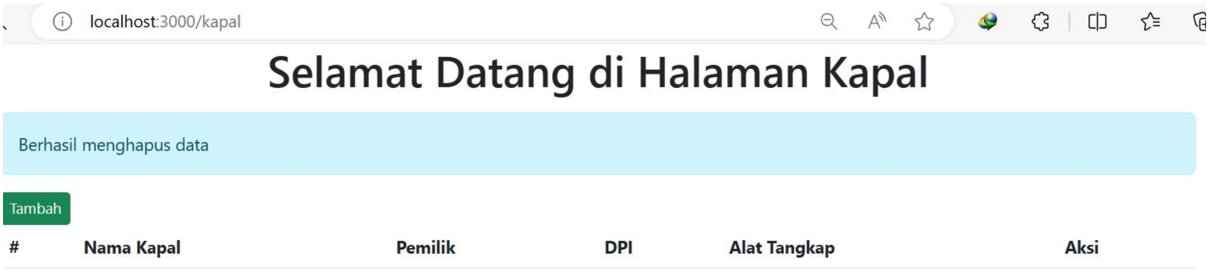
#	Nama Kapal	Pemilik	DPI	Alat Tangkap	Aksi
1	meey go	ekeokeo	kalianget	zoro	Edit Delete



Selamat Datang di Halaman Kapal

Berhasil mengubah data

#	Nama Kapal	Pemilik	DPI	Alat Tangkap	Aksi
1	meey g33	ekeokeo	kalianget	zoro	Edit Delete



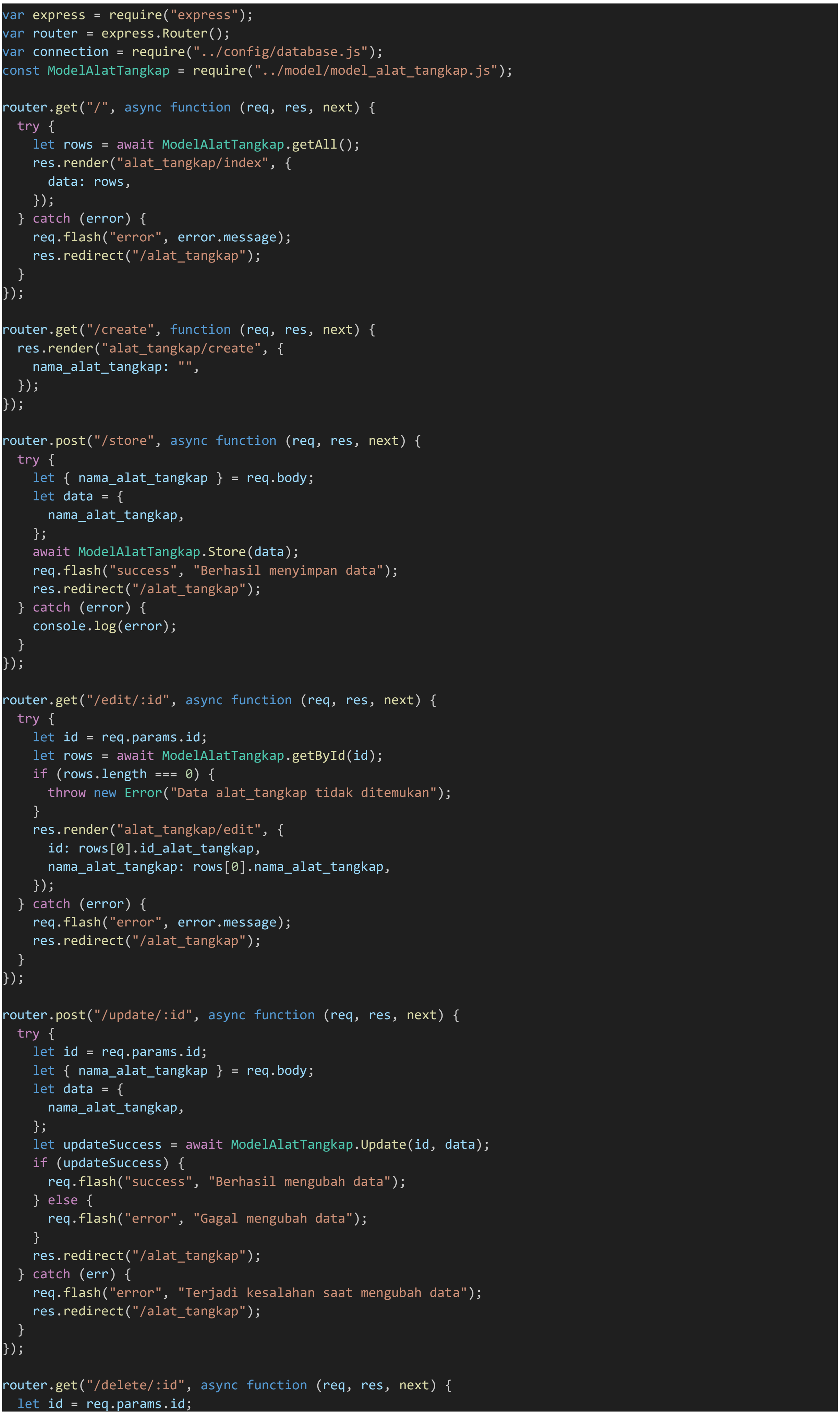
Selamat Datang di Halaman Kapal

Berhasil menghapus data

#	Nama Kapal	Pemilik	DPI	Alat Tangkap	Aksi

Link github : ekosarr/kapal (github.com)

Alat_tangkap.ejs



```
var express = require("express");
var router = express.Router();
var connection = require("../config/database.js");
const ModelAlatTangkap = require("../model/model_alat_tangkap.js");

router.get("/", async function (req, res, next) {
  try {
    let rows = await ModelAlatTangkap.getAll();
    res.render("alat_tangkap/index", {
      data: rows,
    });
  } catch (error) {
    req.flash("error", error.message);
    res.redirect("/alat_tangkap");
  }
});

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 ModelAlatTangkap.Store(data);
    req.flash("success", "Berhasil menyimpan data");
    res.redirect("/alat_tangkap");
  } catch (error) {
    console.log(error);
  }
});

router.get("/edit/:id", async function (req, res, next) {
  try {
    let id = req.params.id;
    let rows = await ModelAlatTangkap.getById(id);
    if (rows.length === 0) {
      throw new Error("Data alat_tangkap tidak ditemukan");
    }
    res.render("alat_tangkap/edit", {
      id: rows[0].id_alat_tangkap,
      nama_alat_tangkap: rows[0].nama_alat_tangkap,
    });
  } catch (error) {
    req.flash("error", error.message);
    res.redirect("/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,
    };
    let updateSuccess = await ModelAlatTangkap.Update(id, data);
    if (updateSuccess) {
      req.flash("success", "Berhasil mengubah data");
    } else {
      req.flash("error", "Gagal mengubah data");
    }
    res.redirect("/alat_tangkap");
  } catch (err) {
    req.flash("error", "Terjadi kesalahan saat mengubah data");
    res.redirect("/alat_tangkap");
  }
});

router.get("/delete/:id", async function (req, res, next) {
  let id = req.params.id;
  await ModelAlatTangkap.Delete(id);
  req.flash("success", "Berhasil menghapus data");
  res.redirect("/alat_tangkap");
});

module.exports = router;
```

Untuk mahasiswa