

EXPRESS



Alif Aditya Ramadhani

3122522015

D3 PSDKU SUMENEP

PRODI D3 TEKNIK INFORMATIKA

DEPARTEMEN TEKNIK INFORMATIKA DAN KOMPUTER

PENS PSDKU SUMENEP

PemilikModel.js

```
const connection = require('../config/database');

class PemilikModel {
  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 create(data) {
    return new Promise((resolve, reject) => {
      connection.query('INSERT INTO pemilik SET ?', data, (err, result)
=> {
        if (err) {
          reject(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[0]);
        }
      });
    });
  }

  static async update(id, data) {
    return new Promise((resolve, reject) => {
      connection.query('UPDATE pemilik SET ? WHERE id_pemilik = ?',
[data, id], (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,
(err, result) => {
      if (err) {
        reject(err);
      } else {
        resolve(result);
      }
    });
  });
}
}

module.exports = PemilikModel;

```

Pemilik.js

```

const express = require('express');
const router = express.Router();
const PemilikModel = require('../model/PemilikModel');

router.get('/', async (req, res, next) => {
  try {
    let rows = await PemilikModel.getAll();
    res.render('pemilik/index', { data: rows, messages: req.flash() });
  } catch (error) {
    next(error);
  }
});

router.get('/create', (req, res) => {
  res.render('pemilik/create');
});

router.post('/store', async (req, res, next) => {
  try {
    let { nama_pemilik, alamat, no_hp } = req.body;
    await PemilikModel.create({ nama_pemilik, alamat, no_hp });
    req.flash('success', 'Berhasil menyimpan data pemilik');
  }

```

```

        res.redirect('/pemilik');
    } catch (error) {
        req.flash('error', 'Gagal menyimpan data pemilik');
        res.redirect('/pemilik');
    }
});

router.get('/edit/:id', async (req, res, next) => {
    try {
        let id = req.params.id;
        let pemilik = await PemilikModel.getById(id);
        res.render('pemilik/edit', {
            id: pemilik.id_pemilik,
            nama_pemilik: pemilik.nama_pemilik,
            alamat: pemilik.alamat,
            no_hp: pemilik.no_hp,
            messages: req.flash()
        });
    } catch (error) {
        next(error);
    }
});

router.post('/update/:id', async (req, res, next) => {
    try {
        let id = req.params.id;
        let { nama_pemilik, alamat, no_hp } = req.body;
        await PemilikModel.update(id, { nama_pemilik, alamat, no_hp });
        req.flash('success', 'Berhasil memperbarui data pemilik');
        res.redirect('/pemilik');
    } catch (error) {
        req.flash('error', 'Gagal memperbarui data pemilik');
        res.redirect('/pemilik');
    }
});

router.get('/delete/:id', async (req, res, next) => {
    try {
        let id = req.params.id;
        await PemilikModel.delete(id);
        req.flash('success', 'Berhasil menghapus data pemilik');
        res.redirect('/pemilik');
    } catch (error) {
        req.flash('error', 'Gagal menghapus data pemilik');
        res.redirect('/pemilik');
    }
});

```

```
module.exports = router;
```

create.ejs

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Create Pemilik</title>
</head>
<body>
  <h1>Create Pemilik</h1>
  <form action="/pemilik/store" method="POST">
    <label for="nama_pemilik">Nama Pemilik:</label><br>
    <input type="text" id="nama_pemilik" name="nama_pemilik"><br>
    <label for="alamat">Alamat:</label><br>
    <input type="text" id="alamat" name="alamat"><br>
    <label for="no_hp">No HP:</label><br>
    <input type="text" id="no_hp" name="no_hp"><br><br>
    <input type="submit" value="Submit">
  </form>
</body>
</html>
```

Edit.ejs

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Edit Pemilik</title>
</head>
<body>
  <h1>Edit Pemilik</h1>
  <form action="/pemilik/update/<%= data.id_pemilik %>" method="POST">
    <label for="nama_pemilik">Nama Pemilik:</label><br>
    <input type="text" id="nama_pemilik" name="nama_pemilik" value="<%=
data.nama_pemilik %>"><br>
    <label for="alamat">Alamat:</label><br>
    <input type="text" id="alamat" name="alamat" value="<%= data.alamat
%>"><br>
    <label for="no_hp">No HP:</label><br>
```

```

        <input type="text" id="no_hp" name="no_hp" value="<%= data.no_hp
%>"><br><br>
        <input type="submit" value="Submit">
    </form>
</body>
</html>

```

Pemilik List

[Add New Pemilik](#)

ID Nama Pemilik Alamat No HP Actions

Index.ejs

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Pemilik List</title>
</head>
<body>
    <h1>Pemilik List</h1>
    <a href="/pemilik/create">Add New Pemilik</a>
    <table>
        <thead>
            <tr>
                <th>ID</th>
                <th>Nama Pemilik</th>
                <th>Alamat</th>
                <th>No HP</th>
                <th>Actions</th>
            </tr>
        </thead>
        <tbody>
            <% data.forEach(function(pemilik) { %>
                <tr>
                    <td><%= pemilik.id_pemilik %></td>
                    <td><%= pemilik.nama_pemilik %></td>
                    <td><%= pemilik.alamat %></td>
                    <td><%= pemilik.no_hp %></td>

```

```

                <td>
                    <a href="/pemilik/edit/<%= pemilik.id_pemilik
%>">Edit</a>
                    <a href="/pemilik/delete/<%= pemilik.id_pemilik %>"
onclick="return confirm('Are you sure?')">Delete</a>
                </td>
            </tr>
        <% }>; %>
    </tbody>
</table>
</body>
</html>

```

App.js

```

const createError = require('http-errors');
const express = require('express');
const path = require('path');
const cookieParser = require('cookie-parser');
const logger = require('morgan');
const flash = require('express-flash');
const session = require('express-session');

const indexRouter = require('./routes/index');
const usersRouter = require('./routes/users');
const mahasiswaRouter = require('./routes/mahasiswa');
const keahliahRouter = require('./routes/keahliah');
const pendidikanRouter = require('./routes/pendidikan');
const pemilikRouter = require('./routes/pemilik');
const alatTangkapRouter = require('./routes/alat_tangkap');

const app = express();

// view engine setup
app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'ejs');

app.use(logger('dev'));
app.use(express.json());
app.use(express.urlencoded({ extended: false }));
app.use(cookieParser());
app.use(express.static(path.join(__dirname, 'public')));

app.use(
  session({
    secret: 'secret',
    resave: true,
    saveUninitialized: true,
    cookie: {

```

```

        maxAge: 60000,
      },
      store: new session.MemoryStore(),
    })
  );

app.use(flash());

app.use('/', indexRouter);
app.use('/users', usersRouter);
app.use('/mahasiswa', mahasiswaRouter);
app.use('/keahlian', keahlianRouter);
app.use('/pendidikan', pendidikanRouter);
app.use('/pemilik', pemilikRouter);
app.use('/alat_tangkap', alatTangkapRouter);

// catch 404 and forward to error handler
app.use(function (req, res, next) {
  next(createError(404));
});

// error handler
app.use(function (err, req, res, next) {
  // set locals, only providing error in development
  res.locals.message = err.message;
  res.locals.error = req.app.get('env') === 'development' ? err : {};

  // render the error page
  res.status(err.status || 500);
  res.render('error');
});

module.exports = app;

```

AlatTangkapModel.js

```

const connection = require('../config/database');

class AlatTangkapModel {

  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, (err,
result) => {
            if (err) {
                reject(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[0]);
            }
        });
    });
}

static async update(id, data) {
    return new Promise((resolve, reject) => {
        connection.query('UPDATE alat_tangkap SET ? WHERE id_alat_tangkap
= ?', [data, id], (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, (err, result) => {
            if (err) {
                reject(err);
            } else {
                resolve(result);
            }
        });
    });
}
}

module.exports = AlatTangkapModel;

```

alat_tangkap.js

```

const express = require('express');
const router = express.Router();
const AlatTangkapModel = require('../model/AlatTangkapModel');

router.get('/', async (req, res, next) => {
    try {
        let rows = await AlatTangkapModel.getAll();
        res.render('alat_tangkap/index', { data: rows });
    } catch (error) {
        next(error);
    }
});

router.get('/create', (req, res) => {
    res.render('alat_tangkap/create');
});

router.post('/store', async (req, res, next) => {
    try {
        const alatTangkapData = req.body;
        await AlatTangkapModel.store(alatTangkapData);
        req.flash('success', 'Berhasil menyimpan data Alat Tangkap');
        res.redirect('/alat_tangkap');
    } catch (error) {
        req.flash('error', 'Gagal menyimpan data Alat Tangkap');
        res.redirect('/alat_tangkap');
    }
});

router.get('/edit/:id', async (req, res, next) => {
    try {
        const id = req.params.id;

```

```

        let data = await AlatTangkapModel.getById(id);
        res.render('alat_tangkap/edit', { data: data });
    } catch (error) {
        next(error);
    }
});

router.post('/update/:id', async (req, res, next) => {
    try {
        const id = req.params.id;
        const alatTangkapData = req.body;
        await AlatTangkapModel.update(id, alatTangkapData);
        req.flash('success', 'Berhasil memperbarui data Alat Tangkap');
        res.redirect('/alat_tangkap');
    } catch (error) {
        req.flash('error', 'Gagal memperbarui data Alat Tangkap');
        res.redirect('/alat_tangkap');
    }
});

router.get('/delete/:id', async (req, res, next) => {
    try {
        const id = req.params.id;
        await AlatTangkapModel.delete(id);
        req.flash('success', 'Berhasil menghapus data Alat Tangkap');
        res.redirect('/alat_tangkap');
    } catch (error) {
        req.flash('error', 'Gagal menghapus data Alat Tangkap');
        res.redirect('/alat_tangkap');
    }
});

module.exports = router;

```

create.ejs

```

<!-- views/alat_tangkap/create.ejs -->
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Create New Alat Tangkap</title>
    <link
href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet"

```

```

        integrity="sha384-
QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMhY6hW+ALEwIH"
crossorigin="anonymous">
</head>

<body>
  <div class="container">
    <h2 class="mt-5">Create New Alat Tangkap</h2>
    <form action="/alat_tangkap/store" method="POST">
      <div class="mb-3">
        <label for="nama" class="form-label">Nama Alat Tangkap</label>
        <input type="text" class="form-control" id="nama"
name="nama_alat_tangkap" required>
      </div>
      <button type="submit" class="btn btn-primary">Submit</button>
    </form>
  </div>
</body>

</html>

```

Edit.ejs

```

<!-- views/alat_tangkap/edit.ejs -->
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Edit Alat Tangkap</title>
  <link
href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet"
  integrity="sha384-
QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMhY6hW+ALEwIH"
crossorigin="anonymous">
</head>

<body>
  <div class="container">
    <h2 class="mt-5">Edit Alat Tangkap</h2>
    <form action="/alat_tangkap/update/<%= id %>" method="POST">
      <div class="mb-3">
        <label for="nama" class="form-label">Nama Alat Tangkap</label>
        <input type="text" class="form-control" id="nama"
name="nama_alat_tangkap" value="<%= nama_alat_tangkap %>"

```

```

        required>
    </div>
    <button type="submit" class="btn btn-primary">Submit</button>
  </form>
</div>
</body>

</html>

```

Index.ejs

```

<!-- views/alat_tangkap/index.ejs -->
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Alat Tangkap List</title>
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet"
integrity="sha384-
QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMhJY6hW+ALEwIH"
crossorigin="anonymous">
</head>

<body>
  <div class="container">
    <% if (messages.success) { %>
    <div class="alert alert-info" role="alert">
      <%= messages.success %>
    </div>
    <% } %>

    <div class="row">
      <div class="col">
        <a href="/alat_tangkap/create" class="btn btn-sm btn-
success">Tambah</a>
        <table class="table">
          <thead>
            <tr>
              <th scope="col">#</th>
              <th scope="col">Nama Alat Tangkap</th>
              <th colspan="2" scope="col">Aksi</th>
            </tr>
          </thead>

```

```

        <tbody>
            <% for (var i = 0; i < data.length; i++) { %>
                <tr>
                    <th scope="row"><%= (i + 1) %></th>
                    <td><%= data[i].nama_alat_tangkap %></td>
                    <td>
                        <a href="/alat_tangkap/edit/<%=
data[i].id_alat_tangkap %>"
                            class="btn btn-sm btn-warning">Edit</a>
                    </td>
                    <td>
                        <a href="/alat_tangkap/delete/<%=
data[i].id_alat_tangkap %>"
                            class="btn btn-sm btn-danger">Delete</a>
                    </td>
                </tr>
            <% } %>
        </tbody>
    </table>
</div>
</div>
</div>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min
.js"
    integrity="sha384-
YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDz0xhy9GkcIdslK1eN7N6jIeHz"
    crossorigin="anonymous"></script>
</body>

</html>

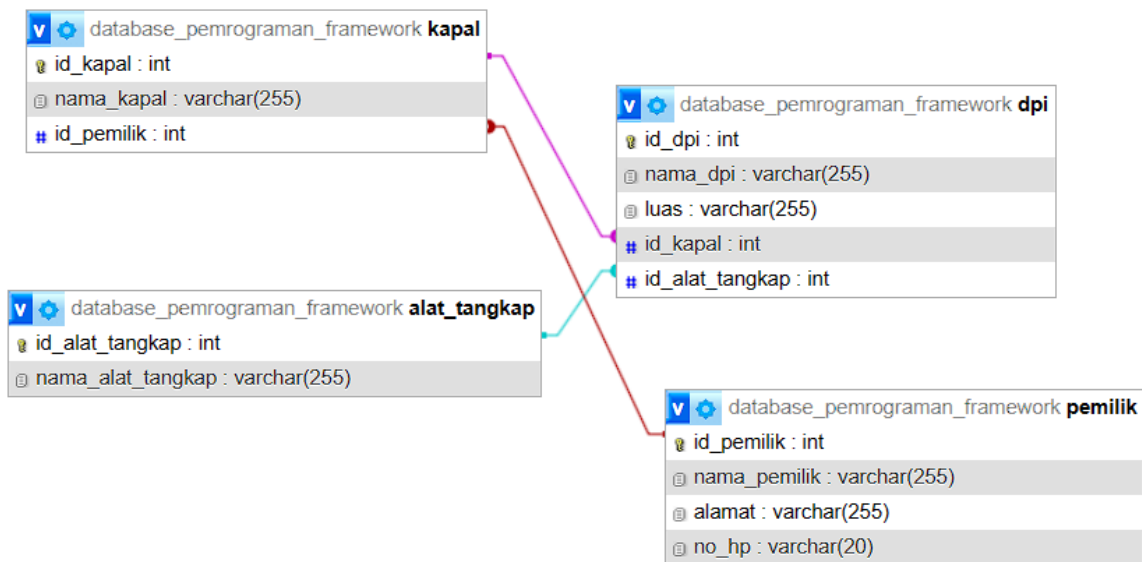
```

Tambah

#

Nama Alat Tangkap

Aksi



https://github.com/alifaditya0/alif_express