Quiz 2

Workshop Pemrograman Framework 2024

Albi Nur Rosif

3122522010

D3 PSDKU Sumenep

PRODI D3 TEKNIK INFORMATIKA

DEPARTEMEN TEKNIK INFORMATIKA DAN KOMPUTER

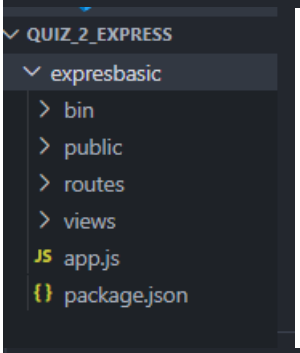
PENS PSDKU SUMENEP

Link github:

https://github.com/albinurrosif/quiz_2_express.git

Membuat Project Express JS

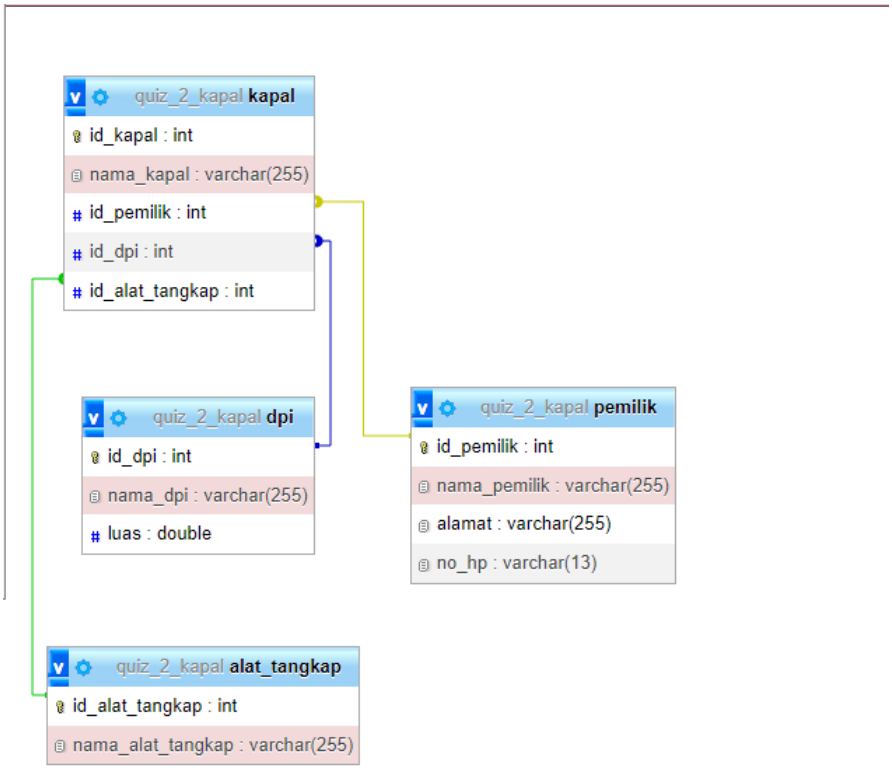
```
TERMINAL   OUTPUT   DEBUG CONSOLE   PORTS   COMMENTS   PROBLEMS

PS C:\semester 4\Workshop Pemrograman Framework\Quiz_2_express> express --view=ejs expresbasic

   create : expresbasic\
   create : expresbasic\public\
   create : expresbasic\public\javascripts\
   create : expresbasic\public\images\
   create : expresbasic\public\stylesheets\
   create : expresbasic\public\stylesheets\style.css
   create : expresbasic\routes\
   create : expresbasic\routes\index.js
   create : expresbasic\routes\users.js
   create : expresbasic\views\
   create : expresbasic\views\error.ejs
   create : expresbasic\views\index.ejs
   create : expresbasic\app.js
   create : expresbasic\package.json
   create : expresbasic\bin\
   create : expresbasic\bin\www

   change directory:
     > cd expresbasic

   install dependencies:
     > npm install

   run the app:
     > SET DEBUG=expresbasic:* & npm start

PS C:\semester 4\Workshop Pemrograman Framework\Quiz_2_express>
```

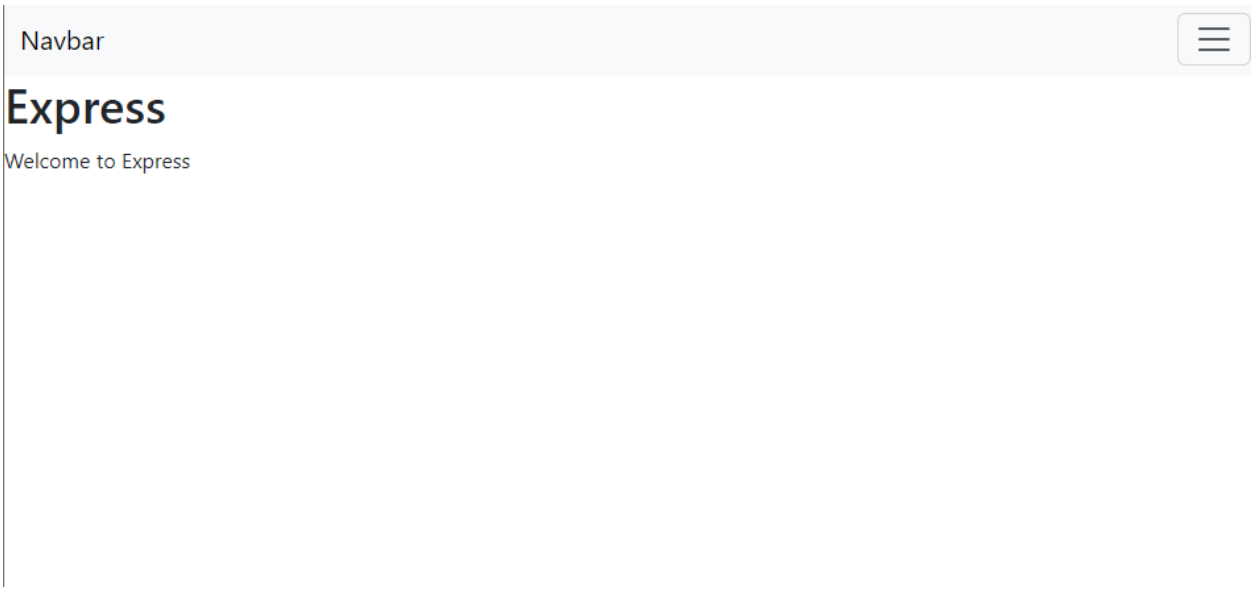



```
PS C:\semester 4\Workshop Pemrograman Framework\Quiz_2_express> cd expresbasic
PS C:\semester 4\Workshop Pemrograman Framework\Quiz_2_express\expresbasic> npm install

added 54 packages, and audited 55 packages in 11s

4 vulnerabilities (3 high, 1 critical)

To address all issues (including breaking changes), run:
expresbasic:* & npm start
At line:1 char:25
+ SET DEBUG=expresbasic:* & npm start
+                         ~
The ampersand (&) character is not allowed. The & operator is reserved for future use; wrap an ampersan
    + CategoryInfo          : ParserError: (:) [], ParentContainsErrorRecordException
    + FullyQualifiedErrorId : AmpersandNotAllowed

PS C:\semester 4\Workshop Pemrograman Framework\Quiz_2_express\expresbasic> SET DEBUG=expresbasic:*
PS C:\semester 4\Workshop Pemrograman Framework\Quiz_2_express\expresbasic> npm start

> expresbasic@0.0.0 start
> node ./bin/www

GET / 200 22.003 ms - 207
GET /stylesheets/style.css 200 7.857 ms - 111
GET / 304 4.697 ms - -
GET /stylesheets/style.css 304 2.392 ms - -
GET /favicon.ico 404 3.316 ms - 1413
```

```
PS C:\semester 4\Workshop Pemrograman Framework\Quiz_2_express\expresbasic> npm install method-override

added 2 packages, and audited 57 packages in 2s

4 vulnerabilities (3 high, 1 critical)

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
PS C:\semester 4\Workshop Pemrograman Framework\Quiz_2_express\expresbasic> npm install mysql

added 9 packages, and audited 66 packages in 2s

4 vulnerabilities (3 high, 1 critical)

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
PS C:\semester 4\Workshop Pemrograman Framework\Quiz_2_express\expresbasic> npm install express-flash

added 2 packages, and audited 68 packages in 2s

4 vulnerabilities (3 high, 1 critical)

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
PS C:\semester 4\Workshop Pemrograman Framework\Quiz_2_express\expresbasic> npm instal express-session

added 7 packages, and audited 75 packages in 2s
```

Database


quiz_2_kapal kapal
id_kapal : int
nama_kapal : varchar(255)
id_pemilik : int
id_dpi : int
id_alat_tangkap : int
quiz_2_kapal dpi
id_dpi : int
nama_dpi : varchar(255)
luas : double
quiz_2_kapal pemilik
id_pemilik : int
nama_pemilik : varchar(255)
alamat : varchar(255)
no_hp : varchar(13)
quiz_2_kapal alat_tangkap
id_alat_tangkap : int
nama_alat_tangkap : varchar(255)


Navbar
Express
Welcome to Express

```
/*jshint esversion: 11 */
const connection = require('../config/database.js');

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);
        } 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);
        } 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;
```

```
/*jshint esversion: 11 */
const connection = require('../config/database.js');

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);
        } 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);
        } 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;
```

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

class Model_kapal {
  static async getAll() {
    return new Promise((resolve, reject) => {
      connection.query(
        'SELECT kapal.*, pemilik.nama_pemilik AS nama_pemilik, dpi.nama AS
nama_dpi, alat_tangkap.nama_alat_tangkap AS nama_alat_tangkap FROM kapal JOIN
pemilik ON kapal.id_pemilik = pemilik.id_pemilik JOIN dpi ON kapal.id_dpi =
dpi.id_dpi JOIN alat_tangkap ON kapal.id_alat_tangkap =
alat_tangkap.id_alat_tangkap',
        (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);
        } else {
          resolve(result);
        }
      });
    });
  }

  static async getId(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 = ' + 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 kapal where id_kapal = ' + id, function (err,
result) {
        if (err) {
          reject(err);
        } else {
          resolve(result);
        }
      });
    });
  }
}

module.exports = Model_kapal;
```

```
/*jshint esversion: 11 */
const connection = require('../config/database.js');
```

```
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);
        } else {
          resolve(result);
        }
      });
    });
  }

  static async getId(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;
```