

Nama : Herlambang Rafli Wicaksono

NPM : 2019101609

WEB SEDERHANA MENGGUNAKAN NODE JS

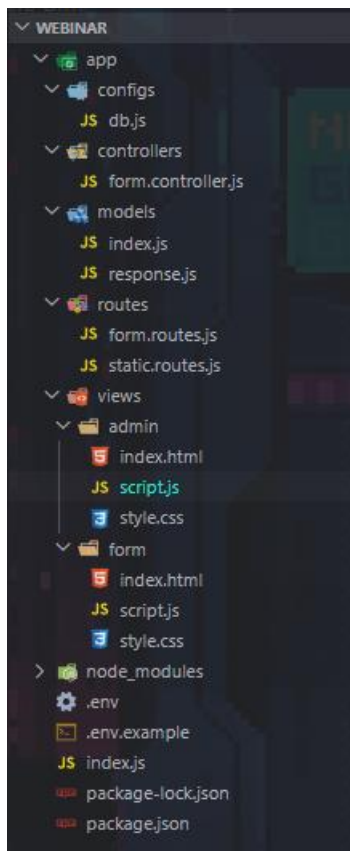
Untuk buat project baru, buat foldernya lalu jalankan npm init. Jangan lupa install nodejs terlebih dahulu

```
PS C:\Users\rafli\Documents\projects\webinar> npm init
```

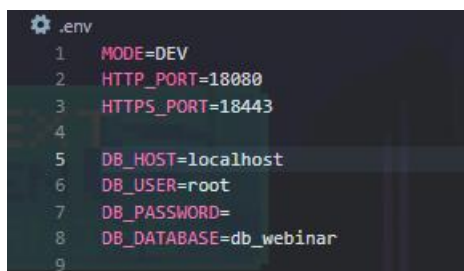
npm i untuk menginstall modul modul yang dibutuhkan

```
PS C:\Users\rafli\Documents\projects\webinar> npm i express mysql2 bcryptjs
```

Strukturnya kira-kira seperti ini



Atur settingan environment dengan membuat file .env



Buat konfigurasi untuk koneksi database

```
app > configs > JS dbjs > ...
1  require('dotenv').config()
2
3  module.exports = {
4    HOST: process.env.DB_HOST,
5    USER: process.env.DB_USER,
6    PASSWORD: process.env.DB_PASSWORD,
7    DB: process.env.DB_DATABASE,
8    collate: "latin1_swedish_ci",
9    charset: "latin1",
10   dialect: "mysql",
11   dialectOptions: {
12     dateStrings: true,
13     typeCast: true
14   },
15   timezone: '+07:00' //for writing to
16 };
```

Buat model untuk mendefinisikan table di database

```
app > models > JS responsejs > <unknown> > exports > Response > phone
1  module.exports = (sequelize, Sequelize) => {
2    const Response = sequelize.define("responses", {
3      name: {
4        type: Sequelize.STRING,
5        allowNull: false
6      },
7      email: {
8        type: Sequelize.STRING,
9        unique: true,
10       allowNull: false,
11       primaryKey: true
12     },
13     phone: {
14       type: Sequelize.STRING,
15       unique: true,
16       allowNull: false
17     }
18   });
19
20   return Response;
21 };
22
```

Buat index.js di models, disini export koneksi dan model. Jika ada relasi bisa dibuat disini

```
app > models > JS index.js > ...
1  const config = require("../configs/db.js");
2
3  const Sequelize = require("sequelize");
4  const sequelize = new Sequelize(
5    config.DB,
6    config.USER,
7    config.PASSWORD, {
8      host: config.HOST,
9      dialect: config.dialect,
10     dialectOptions: config.dialectOptions,
11     timezone: config.timezone,
12     logging: false,
13     define: {
14       collate: config.collate,
15       charset: config.charset,
16     }
17   });
18
19
20 const db = {};
21
22 db.Sequelize = Sequelize;
23 db.sequelize = sequelize;
24
25 db.responses = require("../response.js")(sequelize, Sequelize);
26
27 module.exports = db;
```

Disini otak dari aplikasi. Tulis logikanya di controller. Berikut form.controller.js

```
const db = require("../models");
const Response = db.responses;

const { validationResult } = require('express-validator');
const { Op } = require("sequelize");

exports.register = (req, res) => {
  const errors = validationResult(req)
  if (!errors.isEmpty()) {
    console.log(`[registration fail][${new Date()}]`, req.body,
    errors.array());
    return res.status(422).send({ message: "Invalid input", errors:
    errors.array() })
  }
  Response.findOne({
    where: {
      [Op.or]: {
        email: req.body.email,
        phone: req.body.phone,
      }
    }
  }).then(x => {
    if (x) {
```

```

        return res.status(422).send({ message: "Email or phone number
already registered" })
    }
    Response.create({
        name: req.body.name,
        email: req.body.email,
        phone: req.body.phone,
    }).then(user => {
        console.log(`[registration][${new Date()}] ${req.body.name} -
${req.body.email}`);
        res.status(200).send({ message: "Registered successfully!" });
    }).catch(err => {
        console.log(err);
        res.status(422).send({ message: err.message });
    });
}).catch(err => {
    console.log(err);
    res.status(500).send({ message: "Server error!" });
});
}

exports.all = (req, res) => {
    Response.findAll({}).then(x => {
        res.status(200).send({ data: x });
    }).catch(err => {
        console.log(err);
        res.status(500).send({ message: "Server error!" });
    });
}
}

```

Daftarkan fungsi di controller pada routes agar bisa diakses

```

app > routes > JS form.routes.js > <unknown> > exports
1  const controller = require("../controllers/form.controller");
2  const { body } = require('express-validator')
3
4  module.exports = function (app) {
5      app.use(function (req, res, next) {
6          res.header(
7              "Access-Control-Allow-Headers",
8              "Origin, Content-Type, Accept"
9          );
10         next();
11     });
12
13
14     app.post("/api/register", [
15         body('name').isLength({ min: 1 }),
16         body('email').trim().isEmail(),
17         body('phone').whitelist('+1234567890'),
18     ], controller.register);
19
20     app.get("/api/result", controller.all)
21
22 }

```

Buat rute untuk resource statis. Disini diarahkan ke folder views

```
app > routes > JS static.routes.js > <unknown> > exports
1  var path = require('path');
2  var serveIndex = require('serve-index');
3  const express = require("express");
4
5  var viewDir = path.join(__dirname, '../views');
6
7  function view(endpoint, app, dirname) {
8      let folderDir = path.join(viewDir, `/${dirname}`)
9      app.use(endpoint, express.static(folderDir))
10     app.use(endpoint, serveIndex(folderDir));
11 }
12
13 module.exports = function (app) {
14     view('/', app, 'form')
15     view('/admin', app, 'admin')
16 };

```

Berikut kodingan index.js untuk menjalankan semuanya

```
const express = require("express");
const bodyParser = require("body-parser");
const cors = require("cors");
var https = require('https');

const app = express();

require('dotenv').config()

app.use(cors()); // corsOptions));

// parse requests of content-type - application/json
app.use(bodyParser.json());

// parse requests of content-type - application/x-www-form-urlencoded
app.use(bodyParser.urlencoded({
    extended: true
}));

const db = require("./app/models");

db.sequelize.sync().then(() => {
    console.log('db synced');
}).catch(err => {
    console.log(err.message);
});

require('./app/routes/form.routes')(app);
require('./app/routes/static.routes')(app);

if (process.env.SSL_MODE=='ON') {
    var https_options = {

```

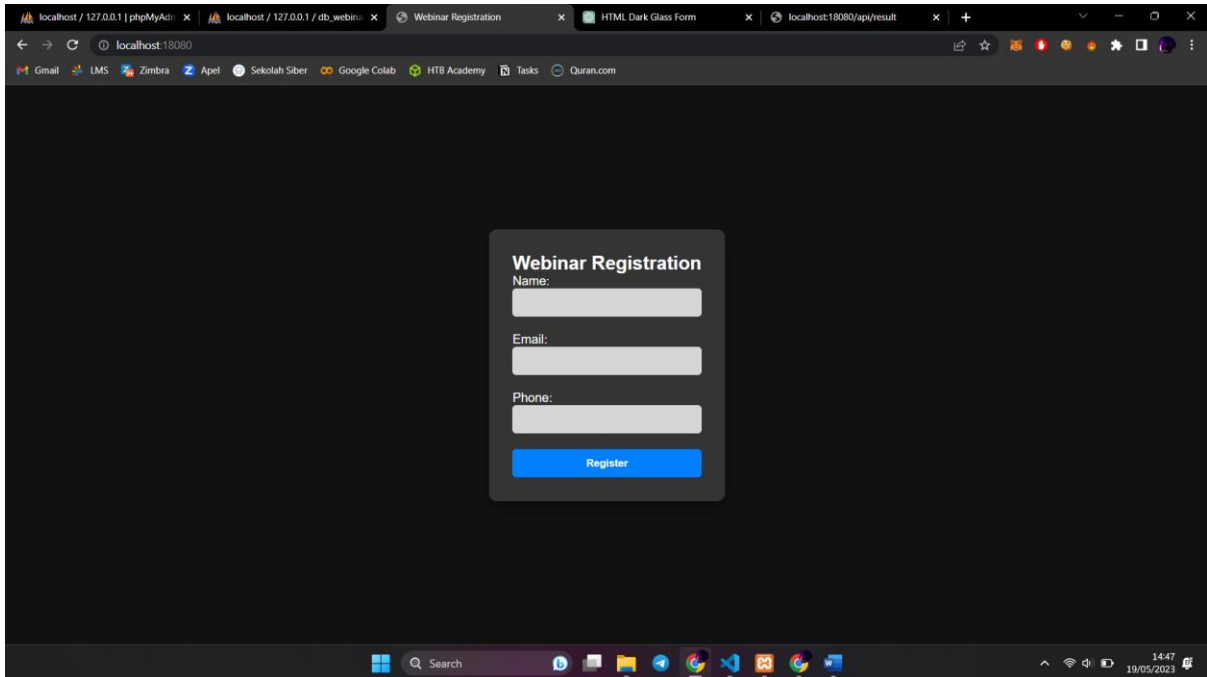
```
    key: fs.readFileSync(process.env.SSL_KEYPATH),
    cert: fs.readFileSync(process.env.SSL_CERTPATH),
    ca: [
      fs.readFileSync(process.env.SSL_CERTPATH),
      fs.readFileSync(process.env.SSL_CABUNDLEPATH)
    ]
  };
  const server = https.createServer(https_options,
app).listen(process.env.HTTPS_PORT);
}

const PORT = process.env.HTTP_PORT || 8080;
app.listen(PORT, () => {
  console.log(`app is running on ${PORT}.`);
});

process.on('SIGINT', () => { console.log("Bye bye!"); process.exit(); });
```

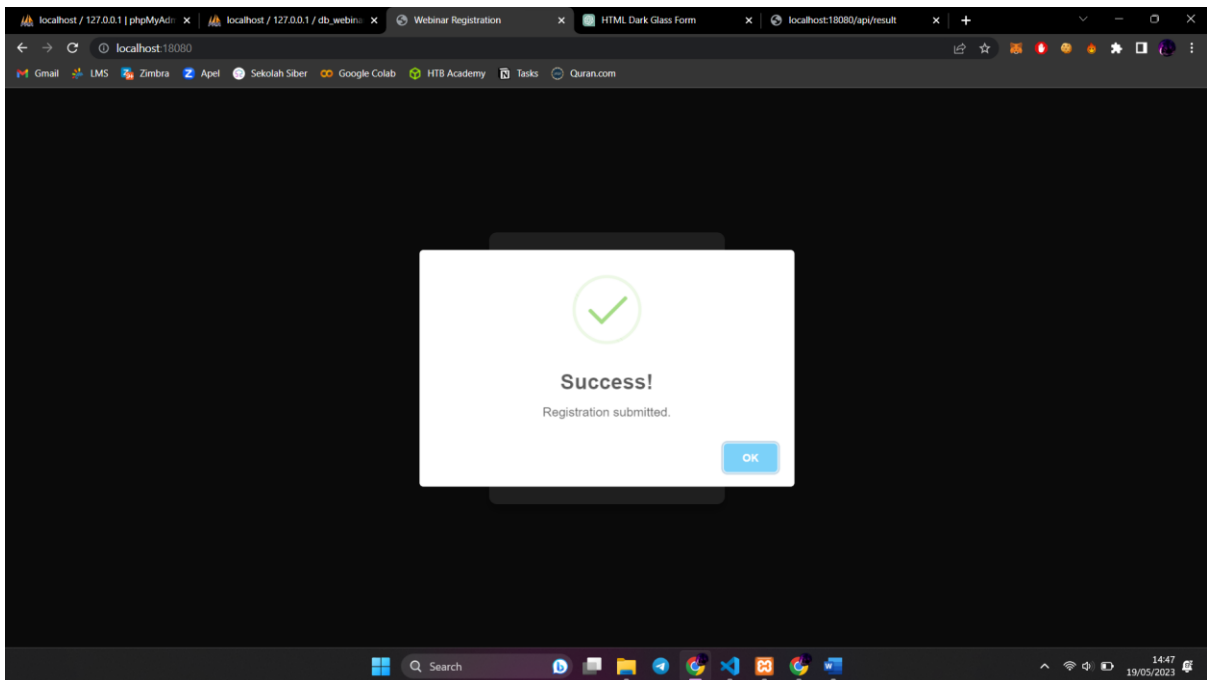
Hasil

Laman pendaftaran



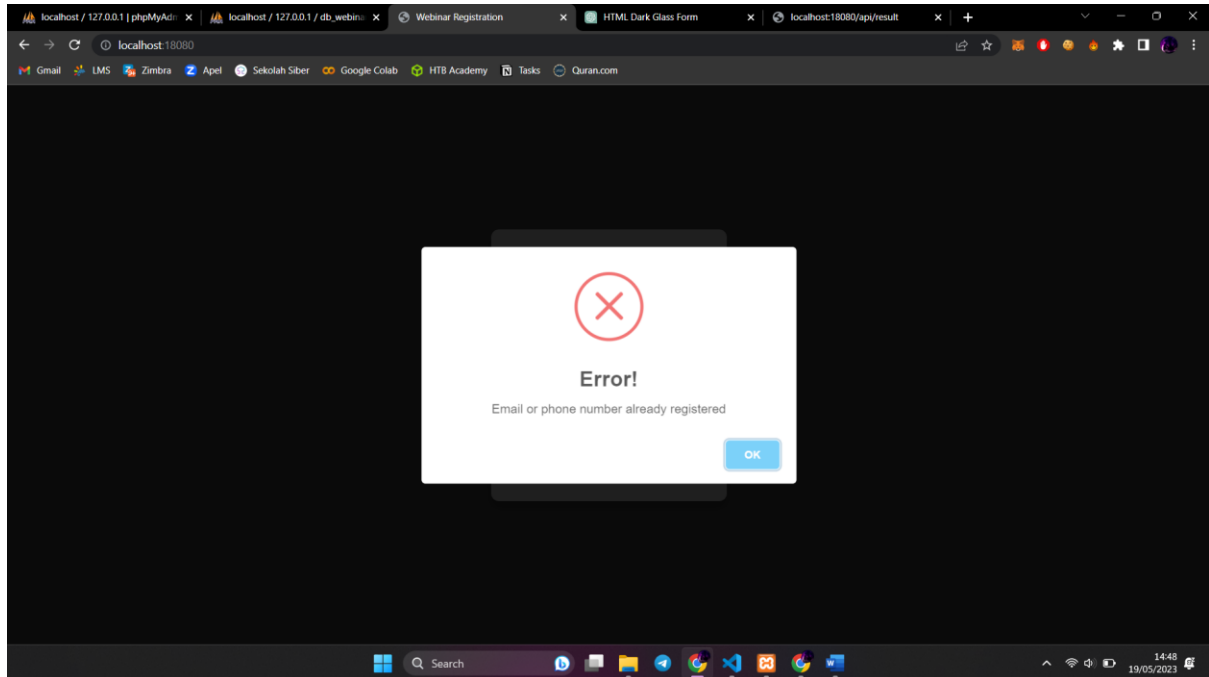
The screenshot shows a web browser window with the address bar displaying 'localhost:18080'. The browser's tab bar includes several tabs: 'localhost / 127.0.0.1 / phpMyAdmin', 'localhost / 127.0.0.1 / db_webin...', 'Webinar Registration', 'HTML Dark Glass Form', and 'localhost:18080/api/result'. The browser's address bar shows 'localhost:18080'. The browser's toolbar includes icons for back, forward, refresh, and search. The browser's bookmarks bar includes links to 'Gmail', 'LMS', 'Zimbra', 'Apel', 'Sekolah Siber', 'Google Colab', 'HTB Academy', 'Tasks', and 'Quran.com'. The main content area of the browser displays a 'Webinar Registration' form. The form has a title 'Webinar Registration' and three input fields labeled 'Name:', 'Email:', and 'Phone:'. Below the input fields is a blue button labeled 'Register'. The browser's status bar at the bottom shows the Windows taskbar with a search bar and several application icons. The system clock in the bottom right corner shows '14:47' and '19/05/2023'.

Tampilan ketika memasukkan input



The screenshot shows the same web browser window as the previous image. The main content area of the browser displays a 'Success!' message dialog box. The dialog box has a green checkmark icon and the text 'Success! Registration submitted.' Below the text is a blue button labeled 'OK'. The browser's status bar at the bottom shows the Windows taskbar with a search bar and several application icons. The system clock in the bottom right corner shows '14:47' and '19/05/2023'.

Tampilan jika input gagal



Laman untuk melihat hasilnya

