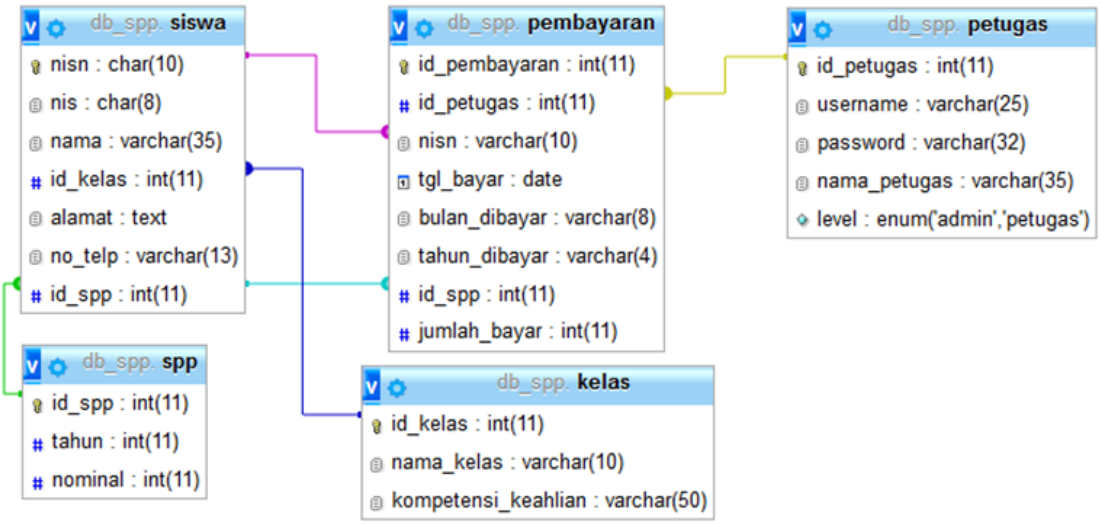
Membuat Database UKL SPP dengan Sequelize



Proses Instalasi

1. Siapkan folder project terserah bernama apapun, disini saya bernama “squelize_ukl_spp”
2. Dengan cmd gunakan direction dari folder project tersebut lalu ketik npm init --y

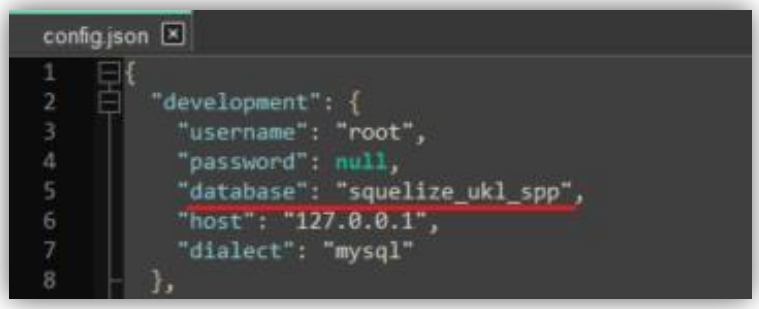
```
C:\WINDOWS\system32>cd D:\Desktop apps\Tugas_Materi\Kelas 11\Produktif XI RPL 1 - NodeJ
S + React\Squelize\ukl-spp

C:\WINDOWS\system32>D:

D:\Desktop apps\Tugas_Materi\Kelas 11\Produktif XI RPL 1 - NodeJS + React\Squelize\ukl-
spp>npm init --y
Wrote to D:\Desktop apps\Tugas_Materi\Kelas 11\Produktif XI RPL 1 - NodeJS + React\Sque
lize\ukl-spp\package.json:

{
  "name": "ukl-spp",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

3. Ketik npm install -g sequelize-cli
4. Ketik npm i --save sequelize mysql2
5. Jika **sistem belum memberi izin** kepada printah sequelize (di windows terminal bukan cmd) ketik Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy Unrestricted
6. Ketik sequelize init

Membuat Database Migration

1. Buat database terlebih dahulu dengan phpmyadmin atau Command

2. Pada folder config, terdapat file bernama config dan edit file tersebut, ubah nama database sesuai dengan database yang dibuat



3. Membuat tabel :

Membuat tabel spp
sequelize model:create --name spp --attributes id_spp:integer,tahun:integer,nominal:integer

Membuat tabel kelas
sequelize model:create --name kelas --attributes
id_kelas:integer,nama_kelas:string,kompetensi_keahlian:string

Membuat tabel petugas
sequelize model:create --name petugas --attributes
id_petugas:integer,username:string,password:string,nama_petugas:string,level:enum

Membuat tabel siswa
sequelize model:create --name siswa --attributes
nisn:string,nis:string,nama:string,id_kelas:integer,alamat:text,no_telp:string,id_spp:integer

Membuat tabel pembayaran
sequelize model:create --name pembayaran --attributes
id_pembayaran:integer,id_petugas:integer,nisn:string,tgl_bayar:date,bulan_dibayar:string,tahun_dibayar:string,id_spp:integer,jumlah_bayar:integer

*Info tambahan Tipedata char tidak ada di sequelize, check https://sequelize.org/v5/manual/data-types.html

*attribute belum dapat dispesifikan

4. Membuat detail tabel (Edit file yang terdapat pada folder migrations)

Membuat detail siswa

```
'use strict';
module.exports = {
  async up(queryInterface, Sequelize) {
    await queryInterface.createTable('siswa', {
      nisn: {
        allowNull: false,
        primaryKey: true,
        type: Sequelize.STRING(10),
        references: {
          model: "pembayaran",
          key: "nisn"
        }
      },
      nis: {
        type: Sequelize.STRING(8)
      },
      nama: {
        type: Sequelize.STRING(35)
      },
      id_kelas: {
        allowNull: false,
        type: Sequelize.INTEGER(11),
        references: {
          model: "kelas",
          key: "id_kelas"
        }
      },
      alamat: {
        type: Sequelize.TEXT
      },
      no_telp: {
        type: Sequelize.STRING(13)
      },
      id_spp: {
        allowNull: false,
        type: Sequelize.INTEGER(11),
        references: {
          model: "spp",
          key: "id_spp"
        }
      },
    });
  },
  async down(queryInterface, Sequelize) {
    await queryInterface.dropTable('siswa');
```

```
  }
};
```

Membuat detail spp

```
'use strict';
module.exports = {
  async up(queryInterface, Sequelize) {
    await queryInterface.createTable('spp', {
      id_spp: {
        allowNull: false,
        autoIncrement: true,
        primaryKey: true,
        type: Sequelize.INTEGER(11)
      },
      tahun: {
        type: Sequelize.INTEGER(11)
      },
      nominal: {
        type: Sequelize.INTEGER(11)
      },
    });
  },
  async down(queryInterface, Sequelize) {
    await queryInterface.dropTable('spp');
  }
};
```

Membuat detail pembayaran

```
'use strict';
module.exports = {
  async up(queryInterface, Sequelize) {
    await queryInterface.createTable('pembayaran', {
      id_pembayaran: {
        allowNull: false,
        autoIncrement: true,
        primaryKey: true,
        type: Sequelize.INTEGER(11)
      },
      id_petugas: {
        allowNull: false,
        type: Sequelize.INTEGER(11),
        references: {
          model: "petugas",
          key: "id_petugas"
        }
      },
      nisn: {
        allowNull: false,
        type: Sequelize.STRING(10),
        references: {
          model: "siswa",
          key: "nisn"
        }
      },
      tgl_bayar: {
        type: Sequelize.DATE
      },
      bulan_dibayar: {
        type: Sequelize.STRING(8)
      },
      tahun_dibayar: {
        type: Sequelize.STRING(4)
      },
      id_spp: {
        allowNull: false,
        type: Sequelize.INTEGER(11),
        references: {
          model: "siswa",
          key: "id_spp"
        }
      },
      jumlah_bayar: {
        type: Sequelize.INTEGER(11)
      },
```

```
    });
  },
  async down(queryInterface, Sequelize) {
    await queryInterface.dropTable('pembayaran');
  }
};
```

Membuat detail kelas

```
'use strict';
module.exports = {
  async up(queryInterface, Sequelize) {
    await queryInterface.createTable('kelas', {
      id_kelas: {
        allowNull: false,
        autoIncrement: true,
        primaryKey: true,
        type: Sequelize.INTEGER(11)
      },
      nama_kelas: {
        type: Sequelize.STRING(10)
      },
      kompetensi_keahlian: {
        type: Sequelize.STRING(50)
      },
    });
  },
  async down(queryInterface, Sequelize) {
    await queryInterface.dropTable('kelas');
  }
};
```

Membuat detail petugas

```
'use strict';
module.exports = {
  async up(queryInterface, Sequelize) {
    await queryInterface.createTable('petugas', {
      id_petugas: {
        allowNull: false,
        autoIncrement: true,
        primaryKey: true,
        type: Sequelize.INTEGER(11)
      },
      username: {
        type: Sequelize.STRING(25)
      },
      password: {
        type: Sequelize.STRING(32)
      },
      nama_petugas: {
        type: Sequelize.STRING(35)
      },
      level: {
        type: Sequelize.ENUM('admin', 'petugas')
      },
    });
  },
  async down(queryInterface, Sequelize) {
    await queryInterface.dropTable('petugas');
  }
};
```

1. Mengatur Relasi pada db (edit file-file pada folder models)

Index.js

```
'use strict';

const fs = require('fs');
const path = require('path');
const Sequelize = require('sequelize');
const basename = path.basename(__filename);
const env = process.env.NODE_ENV || 'development';
const config = require(__dirname + '/../config/config.json')[env];
const db = {};

let sequelize;
if (config.use_env_variable) {
  sequelize = new Sequelize(process.env[config.use_env_variable], config);
} else {
  sequelize = new Sequelize(config.database, config.username, config.password,
config);
}

fs
  .readdirSync(__dirname)
  .filter(file => {
    return (file.indexOf('.') !== 0) && (file !== basename) && (file.slice(-3)
=== '.js');
  })
  .forEach(file => {
    const model = require(path.join(__dirname, file))(sequelize,
Sequelize.DataTypes);
    db[model.name] = model;
  });

Object.keys(db).forEach(modelName => {
  if (db[modelName].associate) {
    db[modelName].associate(db);
  }
});

db.sequelize = sequelize;
db.Sequelize = Sequelize;

module.exports = db;
```

Kelas.js

```
"use strict";
const { Model } = require("sequelize");
module.exports = (sequelize, DataTypes) => {
  class kelas extends Model {
    /**
     * Helper method for defining associations.
     * This method is not a part of Sequelize lifecycle.
     * The `models/index` file will call this method automatically.
     */
    static associate(models) {
      // define association here
      this.hasMany(models.siswa, {
        foreignKey: "id_kelas",
        as: "siswa"
      });
    }
  }

  kelas.init(
    {
      id_kelas: {
        type: DataTypes.INTEGER,
        primaryKey: true,
        autoIncrement: true
      },
      nama_kelas: DataTypes.STRING,
      kompetensi_keahlian: DataTypes.STRING,
    },
    {
      sequelize,
      modelName: "kelas",
      tableName: "kelas"
    }
  );
  return kelas;
};
```

Pembayaran.js

```
"use strict";
const { Model } = require("sequelize");
module.exports = (sequelize, DataTypes) => {
  class pembayaran extends Model {
    /**
     * Helper method for defining associations.
     * This method is not a part of Sequelize lifecycle.
     * The `models/index` file will call this method automatically.
     */
    static associate(models) {
      // define association here
      this.belongsTo(models.siswa, {
        foreignKey: "nisn",
        as: "siswa"
      })
      this.belongsTo(models.petugas, {
        foreignKey: "id_petugas",
        as: "petugas"
      })
      this.belongsTo(models.siswa, {
        foreignKey: "id_spp",
        as: "siswa"
      })
    }
  };

  pembayaran.init(
    {
      id_pembayaran: {
        type: DataTypes.INTEGER,
        primaryKey: true,
        autoIncrement: true
      },
      id_petugas: DataTypes.INTEGER,
      nisn: DataTypes.STRING,
      tgl_bayar: DataTypes.date,
      bulan_dibayar: DataTypes.STRING,
      tahun_dibayar: DataTypes.STRING,
      id_spp: DataTypes.INTEGER,
      jumlah_bayar: DataTypes.INTEGER
    },
    {
      sequelize,
      modelName: "pembayaran",
      tableName: "pembayaran"
```

```
    }
  );
  return pembayaran;
};
```

Petugas.js

```
"use strict";
const { Model } = require("sequelize");
module.exports = (sequelize, DataTypes) => {
  class petugas extends Model {
    /**
     * Helper method for defining associations.
     * This method is not a part of Sequelize lifecycle.
     * The `models/index` file will call this method automatically.
     */
    static associate(models) {
      // define association here
      this.hasMany(models.pembayaran, {
        foreignKey: "id_petugas",
        as: "pembayaran"
      })
    }
  };

  petugas.init(
    {
      id_petugas: {
        type: DataTypes.INTEGER,
        primaryKey: true,
        autoIncrement: true
      },
      username: DataTypes.STRING,
      password: DataTypes.STRING,
      nama_petugas: DataTypes.STRING,
      level: DataTypes.ENUM
    },
    {
      sequelize,
      modelName: "petugas",
      tableName: "petugas"
    }
  );
  return petugas;
};
```

Siswa.js

```
"use strict";
const { Model } = require("sequelize");
module.exports = (sequelize, DataTypes) => {
  class siswa extends Model {
    /**
     * Helper method for defining associations.
     * This method is not a part of Sequelize lifecycle.
     * The `models/index` file will call this method automatically.
     */
    static associate(models) {
      // define association here
      this.hasMany(models.pembayaran, {
        foreignKey: "nisn",
        as: "pembayaran"
      })
      this.hasMany(models.pembayaran, {
        foreignKey: "id_spp",
        as: "pembayaran"
      })
      this.belongsTo(models.spp, {
        foreignKey: "id_spp",
        as: "spp"
      })
      this.belongsTo(models.kelas, {
        foreignKey: "id_spp",
        as: "kelas"
      })
    }
  };

  siswa.init(
    {
      nisn: {
        type: DataTypes.STRING,
        primaryKey: true
      },
      nis: DataTypes.STRING,
      nama: DataTypes.STRING,
      id_kelas: DataTypes.INTEGER,
      alamat: DataTypes.text,
      no_telp: DataTypes.STRING,
      id_spp: DataTypes.INTEGER
    },
    {
      sequelize,
```

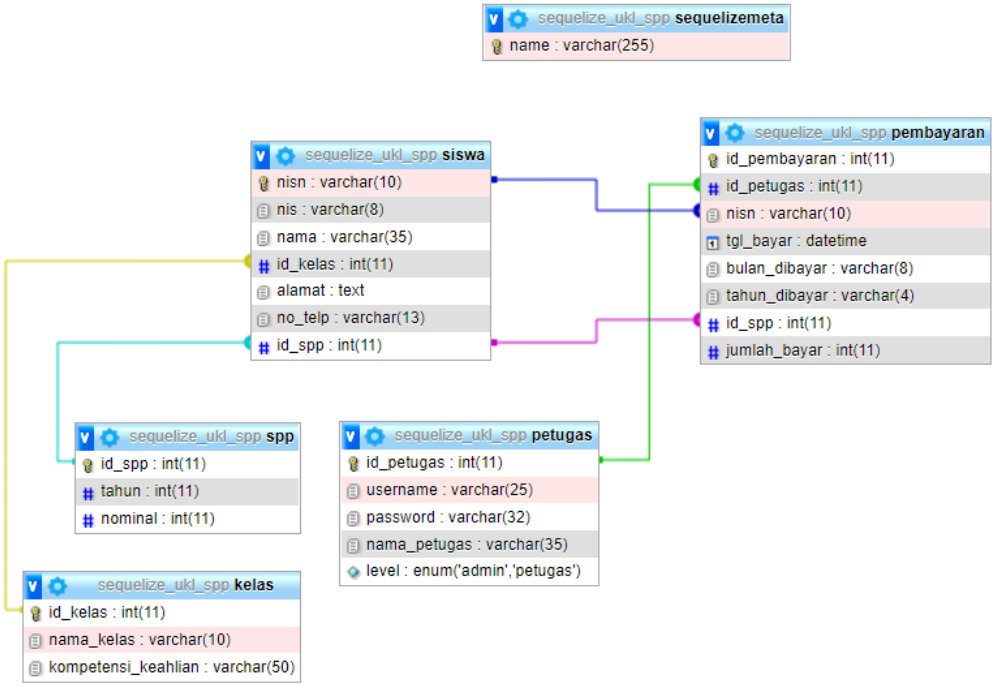
```
      modelName: "siswa",
      tableName: "siswa"
    }
  );
  return siswa;
};
```

Spp.js

```
"use strict";
const { Model } = require("sequelize");
module.exports = (sequelize, DataTypes) => {
  class spp extends Model {
    /**
     * Helper method for defining associations.
     * This method is not a part of Sequelize lifecycle.
     * The `models/index` file will call this method automatically.
     */
    static associate(models) {
      // define association here
      this.hasMany(models.siswa, {
        foreignKey: "id_spp",
        as: "siswa"
      })
    }
  };
  spp.init(
    {
      id_spp: {
        type: DataTypes.INTEGER,
        primaryKey: true,
        autoIncrement: true
      },
      tahun: DataTypes.INTEGER,
      nominal: DataTypes.INTEGER
    },
    {
      sequelize,
      modelName: "spp",
      tableName: "spp"
    }
  );
  return spp;
};
```

Setelah itu ketikkan sequelize db:migrate pada cmd

Untuk Hasil dari Sequelize tersebut adalah



*Jika terdapat kesalahan atau miss saya mohon maaf sebesar-besarnya

*** Note jika ingin merelasikan tabel a ke tabel b, maka tabel b harus ada telebih dahulu (jadi jika terdapat error seperti "can't create table errno: 150" terdapat beberapa kemungkinan**

- Nama dari rujukan model atau penulisan terdapat kesalahan

- Eksekusi tabel A yang ingin relasi ke B terlebih dahulu di migrate sedangkan tabel B belum ada

*** Note tambahan, ketika ingin mengurutkan saya**

1. menaruh file yang detail / yang ada tulisan create-spp, petugas, kelas (yang tidak merujuk kemanapun) lalu sequelize db:migrate

2. menaruh file siswa lalu sequelize db:migrate

3. menaruh file pembayaran lalu sequelize db:migrate

Jika masih terdapat error check pada note merah diatas