

Back-End Development

Express & MongoDB

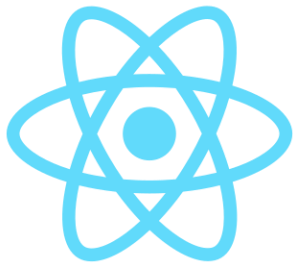
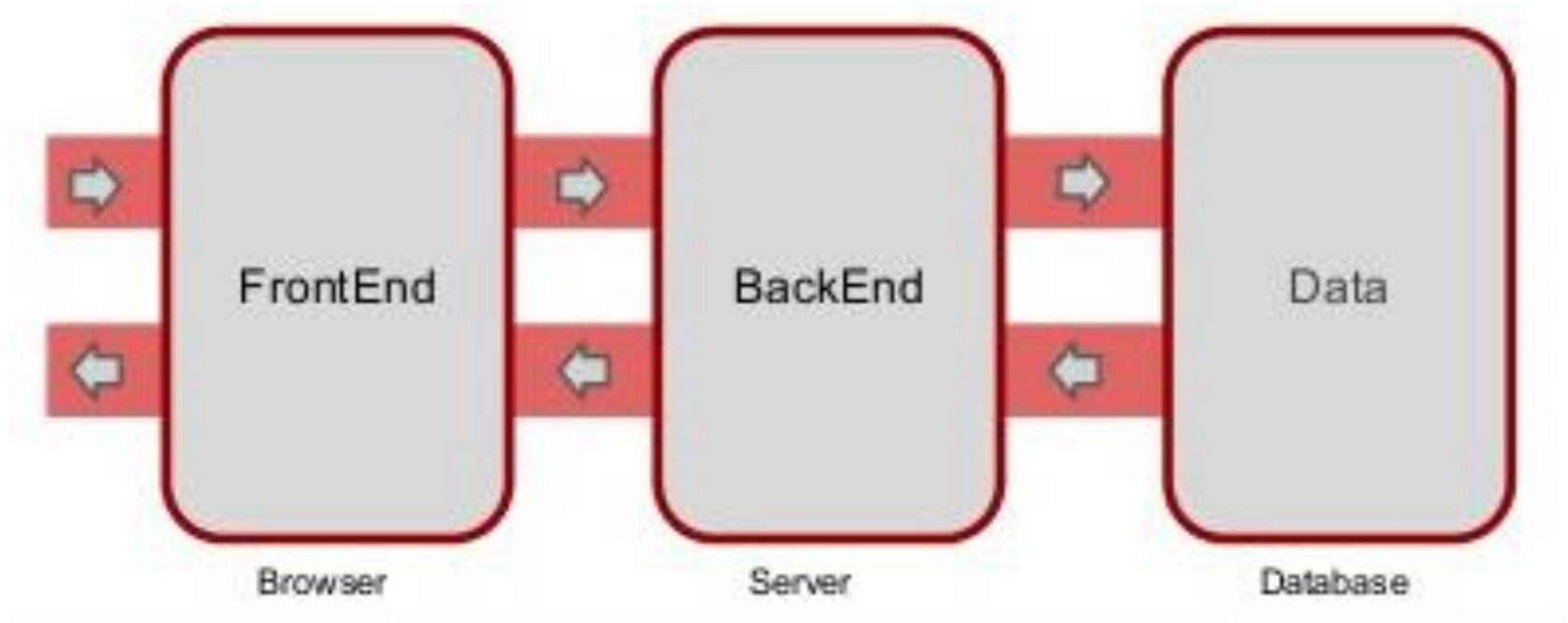
#13c Using Mongoose

express



mongoDB

Web App Architecture



#1 Activate MongoDB Server

- Open terminal:

```
$ cd C:\Program Files\MongoDB\Server\3.6\bin
```

```
$ mongod -dbpath C:\data\db //jika belum//
```

```
$ mongod
```

- If OK, it will show “Waiting for connections on port 27017”. Then open new terminal:

```
$ cd C:\Program Files\MongoDB\Server\3.6\bin
```

```
$ mongo
```

#2 Create Db & User

- Create db toko:

```
$ use toko
```

- Create db user:

```
$ db.createUser({user:"lintang",  
pwd:"1234", roles:["readWrite",  
"dbAdmin"]});
```

- You don't have to create a collection first, we will create it through Mongoose later!



mongoose

- **Mongoose** is an elegant MongoDB object modeling for Node.js framework.
- More info: mongoosejs.com
- On project dir, install Mongoose package:

```
$ npm install mongoose --save
```

#3 Setup Mongoose

Using Mongoose Package

#1 Connect to MongoDB

***a. Create Model & Schema**

// models/user.js

```
const mongoose = require('mongoose');  
const Schema = mongoose.Schema;
```

```
const userSchema = new Schema({  
  nama: String,  
  usia: Number  
});
```

```
const User = mongoose.model('user', userSchema);  
// It will create a 'Users' collection!
```

```
module.exports = User;
```

Using Mongoose Package

#1 Connect to MongoDB

***a. Create Model & Schema**

Model

Represents a collection's records

```
{  
  nama: 'Andi',  
  usia: 24  
}
```

```
{  
  nama: 'Budi',  
  usia: 25  
}
```

```
{  
  nama: 'Caca',  
  usia: 26  
}
```

Schema

Represents the data structure of the records

```
{  
  nama: String,  
  usia: Number  
}
```

Using Mongoose Package

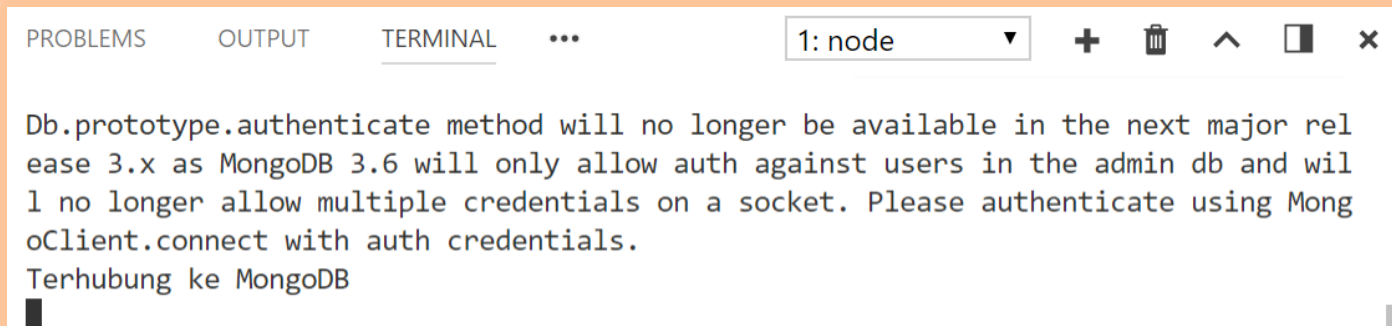
#1 Connect to MongoDB

***b. Make a connection**

// app.js

```
var mongoose = require('mongoose');
var User = require('./models/user');

mongoose.connect(
  'mongodb://Lintang:1234@localhost/toko', ()=>{
    console.log('Terhubung ke MongoDB');
  }
);
```



The screenshot shows a terminal window with the following content:

```
PROBLEMS OUTPUT TERMINAL ... 1: node + - ^ ▢ x

Db.prototype.authenticate method will no longer be available in the next major release 3.x as MongoDB 3.6 will only allow auth against users in the admin db and will no longer allow multiple credentials on a socket. Please authenticate using MongoClient.connect with auth credentials.
Terhubung ke MongoDB
```


Using Mongoose Package

#2 Input Data

// app.js

```
var mongoose = require('mongoose');
var User = require('./models/user');

mongoose.connect(
  'mongodb://lintang:1234@localhost/toko', ()=>{
    console.log('Terhubung ke MongoDB');
  }
);

new User({
  nama: 'Andi',
  usia: 21
}).save().then((newUser)=>{
  console.log('Data masuk: '+newUser);
});
```

Using Mongoose Package

#3 Show All Data

// app.js

```
var mongoose = require('mongoose');
var User = require('./models/user');

mongoose.connect(
  'mongodb://Lintang:1234@localhost/toko', ()=>{
    console.log('Terhubung ke MongoDB');
  }
);

User.find((err, users)=>{
  console.log(users);
});
```

Using Mongoose Package

#4 Show Specific Data

// app.js

```
var mongoose = require('mongoose');
var User = require('./models/user');

mongoose.connect(
  'mongodb://Lintang:1234@localhost/toko', ()=>{
    console.log('Terhubung ke MongoDB');
  }
);

// show data with nama = 'Andi'
User.find({nama: 'Andi'}, (err, users)=>{
  console.log(users);
});
```

Using Mongoose Package

#5 Update Specific Data

// app.js

```
var mongoose = require('mongoose');
var User = require('./models/user');

mongoose.connect(
  'mongodb://Lintang:1234@localhost/toko', ()=>{
    console.log('Terhubung ke MongoDB');
  }
);

//
```

Using Mongoose Package

#6 Remove Specific Data

// app.js

```
var mongoose = require('mongoose');
var User = require('./models/user');

mongoose.connect(
  'mongodb://Lintang:1234@localhost/toko', ()=>{
    console.log('Terhubung ke MongoDB');
  }
);

//
```

Express & MongoDB

express



mongoDB

Express & Mongoose

#1 Connect

```
var express = require('express');
var app = express();

var mongoose = require('mongoose');
var User = require('./models/user');
var url = 'mongodb://lintang:1234@localhost/db';

mongoose.connect(url, ()=>{
  console.log('Terhubung ke MongoDB');
});

### create route here ###

app.listen(3000);
console.log('Running on port 3000!');
```

Express & Mongoose

#2 GET to get data

```
app.get('/get', function(req, res){  
  User.find((err, users)=>{  
    res.send(users);  
  })  
});
```

```
app.get('/get/:nama', function(req, res){  
  User.find({nama: req.params.nama},  
    (err, users)=>{  
      res.send(users);  
    })  
});
```


Express & Mongoose

#3 POST to post data

```
app.post('/post', function(req, res){  
  new User({  
    nama: 'Yaya',  
    usia: 31  
  }).save().then((newUser)=>{  
    console.log('Data masuk: '+newUser);  
    res.send('Data masuk:<br>'+newUser);  
  })  
});
```

Express & mLab

express



mLab

Back-End Development

Express & MongoDB

#13c Using Mongoose

express



mongoDB