

# Membuat Website Tematik Node.js

Membuat website tematik untuk proses membaca basis data

---

1. Pertama tama membuat project menggunakan npm init, isi data yang diperlukan dan setelah selesai akan muncul file **package.json**.
2. Install *library* yang diperlukan seperti **express, ejs, mysql, nodemon**.
  - a. Express digunakan untuk keperluan *back-end service*,
  - b. Ejs digunakan untuk *view engine*,
  - c. Mysql digunakan sebagai database,
  - d. Dan nodemon untuk *nodejs runner*.



```
package.json > ...
1  {
2    "name": "manajamen-sekolah",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "scripts": {
7      "test": "echo \"Error: no test specified\" && exit 1"
8    },
9    "author": "Ainul",
10   "license": "ISC",
11   "dependencies": {
12     "ejs": "^3.1.9",
13     "express": "^4.18.2",
14     "mysql": "^2.18.1",
15     "nodemon": "^3.0.2"
16   }
17 }
18
```

3. Pada **package.json** terdapat property main yang berisi "index.js" lalu buatlah file pada *root folder* dengan nama tersebut.

4. Import **express**, **ejs**, **mysql**, **nodemon** lalu inisiasi express dan koneksi ke database mysql dengan konfigurasi demikian

```
JS index.js > ...
1  const mysql = require('mysql')
2  const express = require('express')
3  let ejs = require('ejs')
4  const app = express()
5
6  const connection = mysql.createConnection({
7    host: 'localhost',
8    user: 'root',
9    password: '',
10   database: 'web_crud'
11 })
12
13 connection.connect()
```

5. Buat folder view lalu isi dengan file html yang telah dibuat pada, ganti file extension file dengan **.ejs** lalu konfigurasi pada index.js

```
JS index.js > ...
14
15 // configuration and middleware
16 app.set('view engine', 'ejs');
17 app.set('views', __dirname + '/view');
18 app.use(express.json());
19 app.use(express.urlencoded({ extended: true }));
20
21 // static assets
22 app.use('/assets', express.static('assets'))
23
```

6. Set *router* menggunakan **app.route** lalu panggil **mysql.connection** untuk mengambil data dari database & tabel yang telah dibuat.

```

JS index.js > ...
37 // routes
38 app.get('/', (_, res) => {
39   res.redirect('/home')
40 })
41
42 app.route('/home')
43   .get(_, res, next) => {
44     const state = { ...uiState, ...dataState }
45     connection.query("SELECT * FROM member", (error, results) => {
46       if (error) {
47         state.isError = true
48         res.render('add_siswa', state)
49         return
50       }
51
52       state.isError = false
53       dataState.siswa = results
54       state.siswa = results
55       res.render('add_siswa', state)
56       next()
57     })
58   })

```

7. Baca data yang telah *dipassing* dari backend menggunakan state pada file **ejs** menggunakan *syntax* berikut

```

<tbody>
  <% siswa.forEach(siswa => { %>
    <tr class="selectable-data">
      <td scope="row"><%= siswa.id_member %></td>
      <td><%= siswa.first_name %></td>
      <td><%= siswa.last_name %></td>
      <td><%= siswa.kota %></td>
      <td><%= siswa.provinsi %></td>
      <td><%= siswa.kode_pos %></td>
      <td><%= siswa.kelas %></td>
    </tr>
  <% }); %>
</tbody>

```

8. Hasil baca data dan konfigurasi yang telah dilakukan.

The screenshot shows a web browser at localhost:3000/home. The page has a header with "Manajemen Sekolah" and "Tambah Data Siswa". Below the header, there is a section titled "Add new data to the row". This section contains a form with the following fields: "First name" (containing "Mark"), "Last name" (containing "Otto"), "Kelas" (containing "1"), "Kota" (empty), "Provinsi" (empty), and "Kode Pos" (empty). A blue "Submit form" button is located below the form. To the right of the form is a table with the following data:

#	First Name	Last Name	City	State	Postal Code	Kelas
7	fira	kamila	surabaya	jatim	66112	1
13	David	Immanuel	Kota Surabaya	jawa timur	60128	1
23	Ainul	Yaqin	Kota Siodarjo	Jawa Timur	\$60128	4

The screenshot shows the same web browser at localhost:3000/home. The page has a header with "Manajemen Sekolah" and "Tambah Data Siswa". Below the header, there is a section titled "Update data from the row". This section contains a form with the following fields: "First name" (containing "David"), "Last name" (containing "Immanuel"), "Kelas" (containing "1"), "Kota" (containing "Kota Surabaya"), "Provinsi" (containing "jawa timur"), and "Kode Pos" (containing "60128"). Below the form are three buttons: a blue "Submit form" button, a red "Hapus" button, and a yellow "Clear" button. To the right of the form is a table with the following data:

#	First Name	Last Name	City	State	Postal Code	Kelas
7	fira	kamila	surabaya	jatim	66112	1
13	David	Immanuel	Kota Surabaya	jawa timur	60128	1
23	Ainul	Yaqin	Kota Siodarjo	Jawa Timur	\$60128	4