

# **LAPORAN UAS**

## **FRONTEND DEVELOPMENT**

Disusun untuk Memenuhi Matakuliah Praktikum Fullstack

Dibimbing oleh Bapak M. Taufik M.Kom



Oleh:

**Chika Nabila Orisandi**

**1122102044**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**  
**SEKOLAH TINGGI ILMU KOMPUTER PGRI BANYUWANGI**  
**2024**

# DAFTAR ISI

## BAB I PENDAHULUAN

1.1	Latar Belakang .....	1
1.2	Tujuan .....	1
1.3	Manfaat .....	1

## BAB II ISI

2.1	Alat dan Bahan .....	2
2.2	Prosedur Kerja .....	2-9
KESIMPULAN.....		9

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Dalam era digital yang terus berkembang, tampilan antarmuka pengguna (UI) menjadi kunci utama dalam memberikan pengalaman yang memikat dan fungsional bagi pengguna aplikasi web. Front-end development, atau pengembangan sisi klien, memainkan peran krusial dalam membentuk bagaimana pengguna berinteraksi dengan suatu platform. Salah satu teknologi yang mendominasi ruang front-end development adalah React.js.

React.js, atau hanya React, adalah sebuah perpustakaan (library) JavaScript yang dikembangkan oleh Facebook. Dirilis pertama kali pada tahun 2013, React menjadi pilihan utama bagi banyak pengembang web karena kemampuannya dalam membangun antarmuka pengguna yang dinamis dan efisien. React menggunakan konsep komponen, memungkinkan pengembang untuk membuat UI yang terstruktur dan mudah dipelihara.

Antarmuka pengguna yang baik tidak hanya mengandalkan desain yang menarik, tetapi juga responsivitas, kecepatan, dan fungsionalitas yang optimal. Front-end development berkaitan erat dengan bagaimana elemen-elemen visual dan interaktivitas dirancang dan diimplementasikan untuk menciptakan pengalaman pengguna yang memuaskan.

Penggunaan React.js memberikan keunggulan tertentu dalam hal manajemen state aplikasi, rendering yang efisien, dan komponen yang dapat digunakan ulang. Dengan struktur kerja yang kuat, React memfasilitasi pengembangan aplikasi yang skalabel dan mudah diatur.

### **1.2 Tujuan**

Tujuan yang diharapkan dapat dicapai dari pemberian tugas ini antara lain:

1. Mahasiswa diharapkan dapat mempraktekkan tentang pembuatan proyek frontend
2. Mahasiswa diharapkan dapat mempraktekkan tentang menghubungkan backend dan frontend

### **1.3 Manfaat**

Manfaat yang diharapkan dapat dicapai dari pemberian tugas ini antara lain:

1. Mahasiswa dapat mempraktekkan tentang pembuatan proyek frontend
2. Mahasiswa dapat mempraktekkan tentang menghubungkan backend dan frontend

## BAB II

### ISI

#### 2.1 Alat dan bahan

Alat dan bahan yang dibutuhkan dalam pengerjaan tugas ini antara lain:

1. Laptop/pc
2. VS Code sebagai tool editor
3. MongoDB sebagai database open source untuk membuat database
4. Postman untuk mengecek proses CRUD
5. Webserver

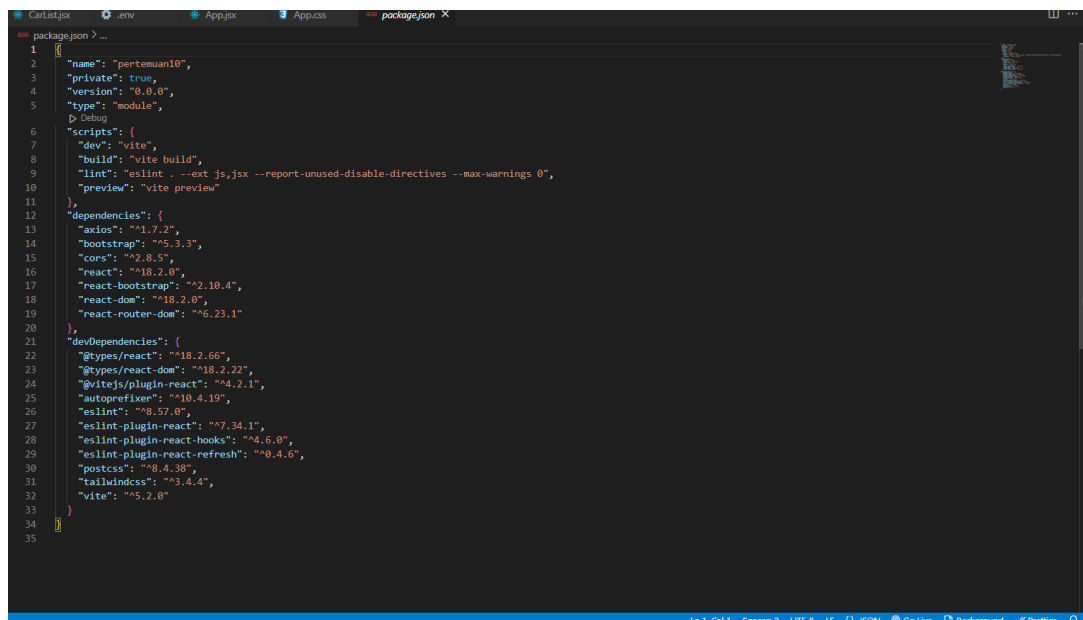
#### 2.2 Prosedur Kerja

<https://github.com/chikanabila/uas-back-sem-4.git>

<https://github.com/chikanabila/uas-front-sem4.git>

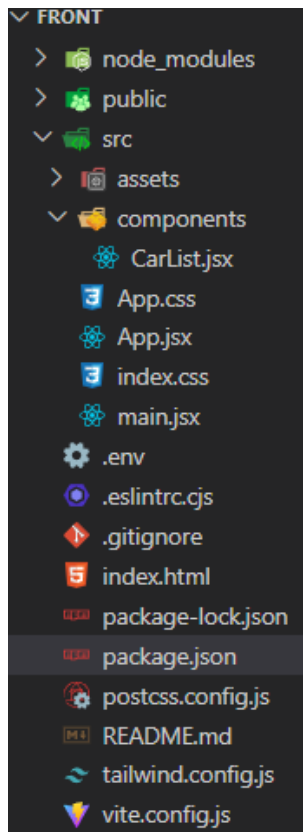
Untuk prosedur kerja yang dilakukan dalam pengerjaan proyek ini antara lain:

1. Melakukan instalasi React.js. Langkah-langkah dalam penginstalan React.js antara lain:
  - a. Melakukan instal react terlebih dahulu dengan mengetik `npx create-react-app`
  - b. Setelah instalasi react berhasil maka pada folder tadi akan terdapat file `package-lock.json` dan file Express akan ditambah ke dependencies ke file `package.json`



```
1 {
2   "name": "pertemuan10",
3   "private": true,
4   "version": "0.0.0",
5   "type": "module",
6   "scripts": {
7     "dev": "vite",
8     "build": "vite build",
9     "lint": "eslint . --ext js,jsx --report-unused-disable-directives --max-warnings 0",
10    "preview": "vite preview"
11  },
12  "dependencies": {
13    "axios": "^1.7.2",
14    "bootstrap": "^5.3.3",
15    "cors": "^2.8.5",
16    "react": "^18.2.0",
17    "react-bootstrap": "^2.10.4",
18    "react-dom": "^18.2.0",
19    "react-router-dom": "^6.23.1"
20  },
21  "devDependencies": {
22    "@types/react": "^18.2.66",
23    "@types/react-dom": "^18.2.22",
24    "@vitejs/plugin-react": "^4.2.1",
25    "autoprefixer": "^10.4.19",
26    "eslint": "^8.57.0",
27    "eslint-plugin-react": "^7.34.1",
28    "eslint-plugin-react-hooks": "^4.6.0",
29    "eslint-plugin-react-refresh": "^0.4.6",
30    "postcss": "^8.4.38",
31    "tailwindcss": "^3.4.4",
32    "vite": "^5.2.0"
33  }
34 }
```

2. Mulai membuat proyek menggunakan React. Langkah-langkahnya antara lain:
  - a. Tambahkan beberapa folder dan file baru hingga tampilan seperti ini:



- b. Masukkan syntax dibawah ini pada src → components → CarList.jsx

```
// src/components/CarList.jsx
import React, { useState, useEffect } from 'react';
import axios from 'axios';

const CarList = () => {
  const [cars, setCars] = useState([]);

  useEffect(() => {
    axios.get('http://localhost:3000/cars')
      .then(response => {
        console.log('Response from server:', response.data);
        const carsWithDate = response.data.map(car => ({
          ...car,
          tanggal_keluar: new Date(car.tanggal_keluar)
        }));
        setCars(carsWithDate);
      })
      .catch(error => {
        console.error('Error fetching cars:', error);
      });
  }, []);

  return (
    <div>
      <h2><b>Car List</b></h2>
      <table>
```

```

        <thead>
          <tr>
            <th>Name</th>
            <th>Model</th>
            <th>Release Date</th>
          </tr>
        </thead>
        <tbody>
          {cars.length > 0 ? cars.map(car => (
            <tr key={car._id}>
              <td>{car.nama}</td>
              <td>{car.model}</td>
              <td>{car.tahun_keluar}</td>
            </tr>
          )) : (
            <tr>
              <td colspan="3">No data available</td>
            </tr>
          )}
        </tbody>
      </table>
    </div>
  );
};

export default CarList;

```

- c. Masukkan syntax dibawah ini pada src → components → App.jsx

```

// src/App.jsx
import React from 'react';
import CarList from './components/CarList';
import './App.css';

function App() {
  return (
    <div className="App">
      <h1>Car List App</h1>
      <CarList />
    </div>
  );
}

export default App;

```

- d. Masukkan syntax dibawah ini pada src → components → App.css

```

/* src/App.css */

```

```

body {
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 0;
  background-color: #f0f0f0;
}

.container {
  max-width: 800px;
  margin: 20px auto;
  padding: 20px;
  background-color: #fff;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

h2 {
  text-align: center;
  margin-bottom: 20px;
  font-size: 32px;
  font-weight: bold;
}

table {
  width: 100%;
  border-collapse: collapse;
  margin-top: 20px;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1); /* Shadow di bawah tabel */
}

th, td {
  padding: 12px 15px;
  text-align: left;
  border-bottom: 1px solid #766969;
}

th {
  background-color: #ead3d3;
  font-weight: bold;
  text-transform: uppercase; /* Teks header huruf kapital */
}

tr:hover {
  background-color: #c99292;
}

.error-message {
  color: red;
  font-style: italic;
}

```

- e. Save file lalu jalankan perintah dengan mengetik `nodemon index.js` pada terminal. Jika berhasil maka akan terminal akan menampilkan seperti dibawah ini:

```
PS D:\Kuliah\Semester 4\Praktikum Fullstack\back> nodemon app.js
[nodemon] 3.0.2
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node app.js`
(node:25632) [MONGODB DRIVER] Warning: useUrlParser is a deprecated option: useUrlParser has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
(node:25632) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
Server started on port 3000
MongoDB Connected
```

- f. Save lalu ketikkan `npm run dev` pada terminal untuk memainkan frontend. Pastikan pada saat yang bersamaan mongodb telah ter-connect dan project backend telah di-run pula.

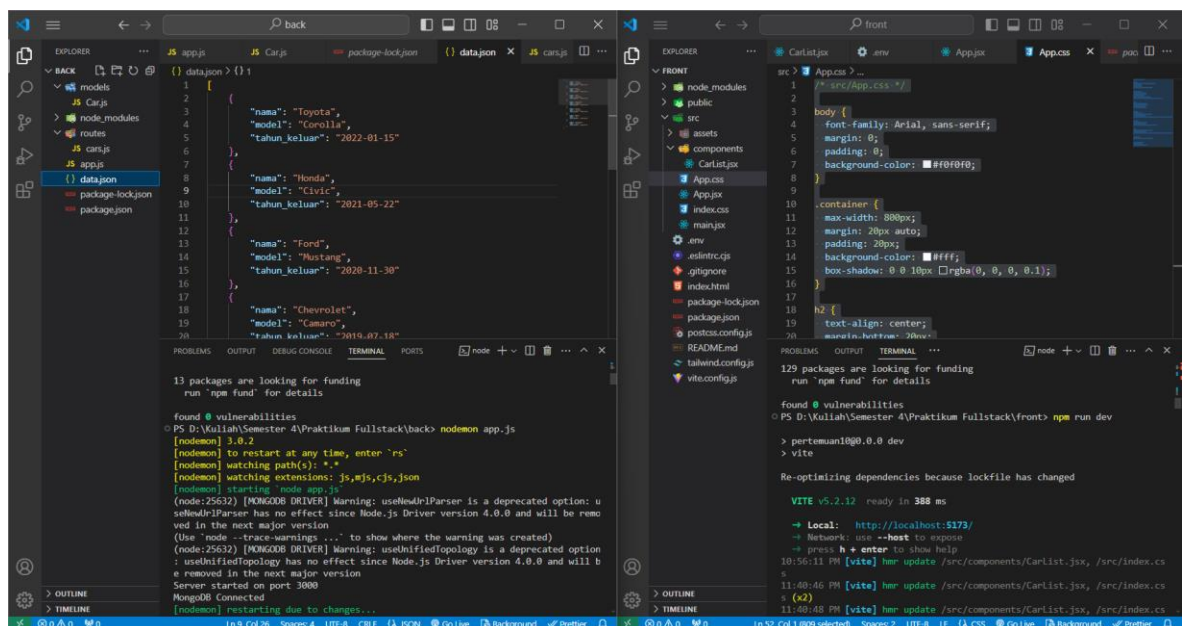
```
found 0 vulnerabilities
PS D:\Kuliah\Semester 4\Praktikum Fullstack\front> npm run dev

> pertemuan10@0.0.0 dev
> vite

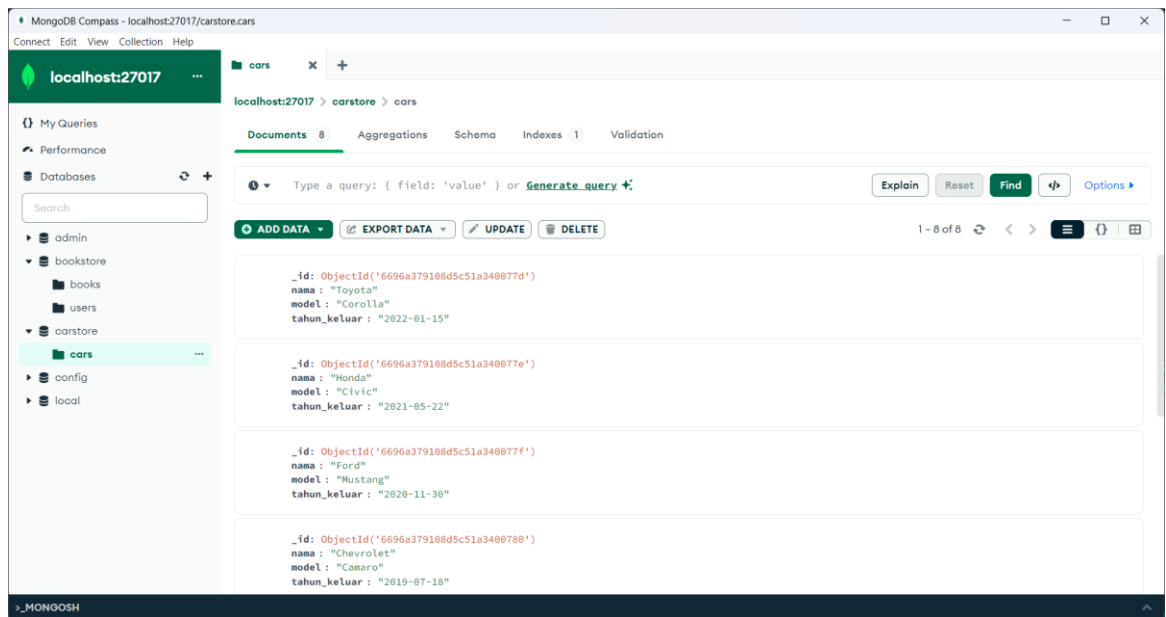
Re-optimizing dependencies because lockfile has changed

VITE v5.2.12 ready in 388 ms

→ Local: http://localhost:5173/
→ Network: use --host to expose
→ press h + enter to show help
```







- g. Klik <http://localhost:5173> untuk melihat hasil implementasi frontend dan backend. Jika berhasil maka akan menampilkan output seperti dibawah ini:

Car List		
NAME	MODEL	RELEASE DATE
Toyota	Corolla	2022-01-15
Honda	Civic	2021-05-22
Ford	Mustang	2020-11-30
Chevrolet	Camaro	2019-07-18
Nissan	Altima	2023-03-10
Mazda	3	2021-09-25
Volkswagen	Golf	2020-04-12
Audi	A4	2021-08-09

- h. Jika ingin mengedit databse, maka editlah database pada postman. Setiap perubahan yang terjadi akan langsung terlihat hasilnya pada laman <http://localhost:5173> tadi, dan juga akan ada notifikasi yang muncul pad atiap perubahan.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Kuliah\Semester 4\Praktikum Fullstack\back> nodemon app.js
[nodemon] 3.0.2
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting 'node app.js'
(node:25632) [MONGODB DRIVER] Warning: useNewUriParser is a deprecated option: useNewUriParser has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
(Use `node --trace-warnings ...` to show where the warning was created)
(node:25632) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
Server started on port 3000
MongoDB Connected
[nodemon] restarting due to changes...
[nodemon] starting 'node app.js'
(node:28108) [MONGODB DRIVER] Warning: useNewUriParser is a deprecated option: useNewUriParser has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
(Use `node --trace-warnings ...` to show where the warning was created)
(node:28108) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
Server started on port 3000
MongoDB Connected
[nodemon] restarting due to changes...
[nodemon] starting 'node app.js'
(node:526) [MONGODB DRIVER] Warning: useNewUriParser is a deprecated option: useNewUriParser has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
(Use `node --trace-warnings ...` to show where the warning was created)
```

## Notifikasi pada folder project backend

```
VITE v5.2.12 ready in 388 ms

→ Local: http://localhost:5173/
→ Network: use --host to expose
→ press h + enter to show help

10:56:11 PM [vite] hmr update /src/components/CarList.jsx, /src/index.css
11:40:46 PM [vite] hmr update /src/components/CarList.jsx, /src/index.css (x2)
11:40:48 PM [vite] hmr update /src/components/CarList.jsx, /src/index.css (x3)
11:40:56 PM [vite] hmr update /src/components/CarList.jsx, /src/index.css (x4)
11:41:58 PM [vite] hmr update /src/components/CarList.jsx, /src/index.css (x5)
11:41:59 PM [vite] hmr update /src/components/CarList.jsx, /src/index.css (x6)
11:42:07 PM [vite] hmr update /src/components/CarList.jsx, /src/index.css (x7)
11:42:17 PM [vite] hmr update /src/components/CarList.jsx, /src/index.css (x8)
11:42:41 PM [vite] hmr update /src/components/CarList.jsx, /src/index.css (x9)
11:42:46 PM [vite] hmr update /src/components/CarList.jsx, /src/index.css (x10)
11:42:49 PM [vite] hmr update /src/components/CarList.jsx, /src/index.css (x11)
11:42:51 PM [vite] hmr update /src/components/CarList.jsx, /src/index.css (x12)
11:48:47 PM [vite] hmr update /src/components/CarList.jsx, /src/index.css (x13)
11:48:42 PM [vite] hmr update /src/components/CarList.jsx, /src/index.css (x14)
11:50:50 PM [vite] hmr update /src/components/CarList.jsx, /src/index.css (x15)
11:50:56 PM [vite] hmr update /src/components/CarList.jsx, /src/index.css (x16)
```

## Notifikasi pada folder project frontend

## KESIMPULAN

### Kesimpulan untuk Proyek Frontend

1. Penggunaan React: Penggunaan React sebagai library JavaScript untuk membangun antarmuka pengguna (UI) aplikasi. React memudahkan pengelolaan state dan rendering dinamis komponen UI.
2. Komponen CarList: Pembuatan komponen `CarList` untuk mengambil dan menampilkan data mobil dari backend. Komponen ini menggunakan `axios` untuk melakukan permintaan HTTP GET ke endpoint backend dan menyimpan data yang diterima dalam state komponen.
3. Penggunaan CSS untuk Styling: Pembuatan file `App.css` untuk menambahkan gaya pada aplikasi frontend, termasuk menambahkan gaya pada tabel untuk membuatnya lebih menarik dan mudah dibaca.

### Kesimpulan untuk Proyek Fullstack

1. Integrasi Backend dan Frontend: Proyek fullstack ini berhasil mengintegrasikan backend Node.js/Express dengan frontend React, memungkinkan aplikasi untuk mengambil dan menampilkan data mobil dari basis data MongoDB.
2. Operasi CRUD: Implementasi operasi CRUD pada backend memungkinkan pengguna untuk membuat, membaca, memperbarui, dan menghapus data mobil. Frontend mengambil data ini dan menampilkannya dalam bentuk tabel.
3. Penggunaan Teknologi Modern: Penggunaan teknologi modern seperti MongoDB untuk basis data NoSQL, Mongoose sebagai ODM, Express untuk membuat API RESTful, dan React untuk membangun antarmuka pengguna yang dinamis dan responsif.