

Tugas Pendahuluan Modul 12

1. Index.html

```
public > index.html > ...
 1  <!DOCTYPE html>
 2  <html lang="en">
 3  <head>
 4    <meta charset="UTF-8">
 5    <title>TP Modul 12</title>
 6  </head>
 7  <body>
 8    <h1>TP Modul 12</h1>
 9    <input type="number" id="inputNumber" placeholder="Masukkan angka">
10    <button onclick="checkNumber()">Cek Tanda Bilangan</button>
11    <p id="resultLabel"></p>
12
13    <script src="script.js"></script>
14  </body>
15 </html>
16
```

2. App.js

```
index.html  script.js  test.js  package.json  app.js
public > script.js > ...
 1  function checkNumber() {
 2    const input = parseInt(document.getElementById("inputNumber").value);
 3    fetch(`/check/${input}`)
 4      .then(res => res.json())
 5      .then(data => {
 6        document.getElementById("resultLabel").innerText = data.result;
 7      })
 8      .catch(err => {
 9        console.error(err);
10      });
11  }
12
```

3. Test.js

```
index.html script.js test.js x package.json app.js
test > .js test.js > ...
1  const assert = require("assert");
2  const { CariTandaBilangan } = require("../app");
3
4  describe("CariTandaBilangan", () => {
5    it("should return 'Negatif' for negative numbers", () => {
6      assert.strictEqual(CariTandaBilangan(-10), "Negatif");
7    });
8
9    it("should return 'Positif' for positive numbers", () => {
10     assert.strictEqual(CariTandaBilangan(5), "Positif");
11   });
12
13   it("should return 'Nol' for zero", () => {
14     assert.strictEqual(CariTandaBilangan(0), "Nol");
15   });
16 });
17
```

4. App.js

```
index.html .js script.js .js test.js package.json .js app.js x
.js app.js > ...
1  const express = require("express");
2  const app = express();
3  const path = require("path");
4
5  function CariTandaBilangan(a) {
6    if (a < 0) return "Negatif";
7    if (a > 0) return "Positif";
8    return "Nol";
9  }
10
11  app.use(express.static(path.join(__dirname, "public")));
12
13  app.get("/check/:num", (req, res) => {
14    const num = parseInt(req.params.num);
15    const result = CariTandaBilangan(num);
16    res.json({ result });
17  });
18
19  app.listen(3000, () => {
20    console.log("Server running on http://localhost:3000");
21  });
22
23  module.exports = { CariTandaBilangan };
24
```

Aplikasi ini merupakan implementasi sederhana dari GUI berbasis web menggunakan HTML, JavaScript, dan Node.js untuk memenuhi tugas TP Modul 12. Tampilan web menampilkan input bertipe number, sebuah tombol, dan label untuk menampilkan hasil. Ketika tombol ditekan, JavaScript akan mengambil nilai input dan mengirimkannya ke backend menggunakan `fetch()` ke endpoint `/check/:num`. Di sisi backend, fungsi `CariTandaBilangan` akan menentukan apakah angka tersebut negatif, positif, atau nol, lalu mengembalikannya dalam bentuk JSON untuk ditampilkan di halaman. Aplikasi juga dilengkapi dengan unit test menggunakan Mocha untuk memastikan bahwa fungsi `CariTandaBilangan` bekerja dengan benar pada semua kemungkinan input.

TP Modul 12

<input type="text" value="-1000"/>	<input type="button" value="Cek Tanda Bilangan"/>
------------------------------------	---

Negatif

TP Modul 12

<input type="text" value="0"/>	<input type="button" value="Cek Tanda Bilangan"/>
--------------------------------	---

Nol

TP Modul 12

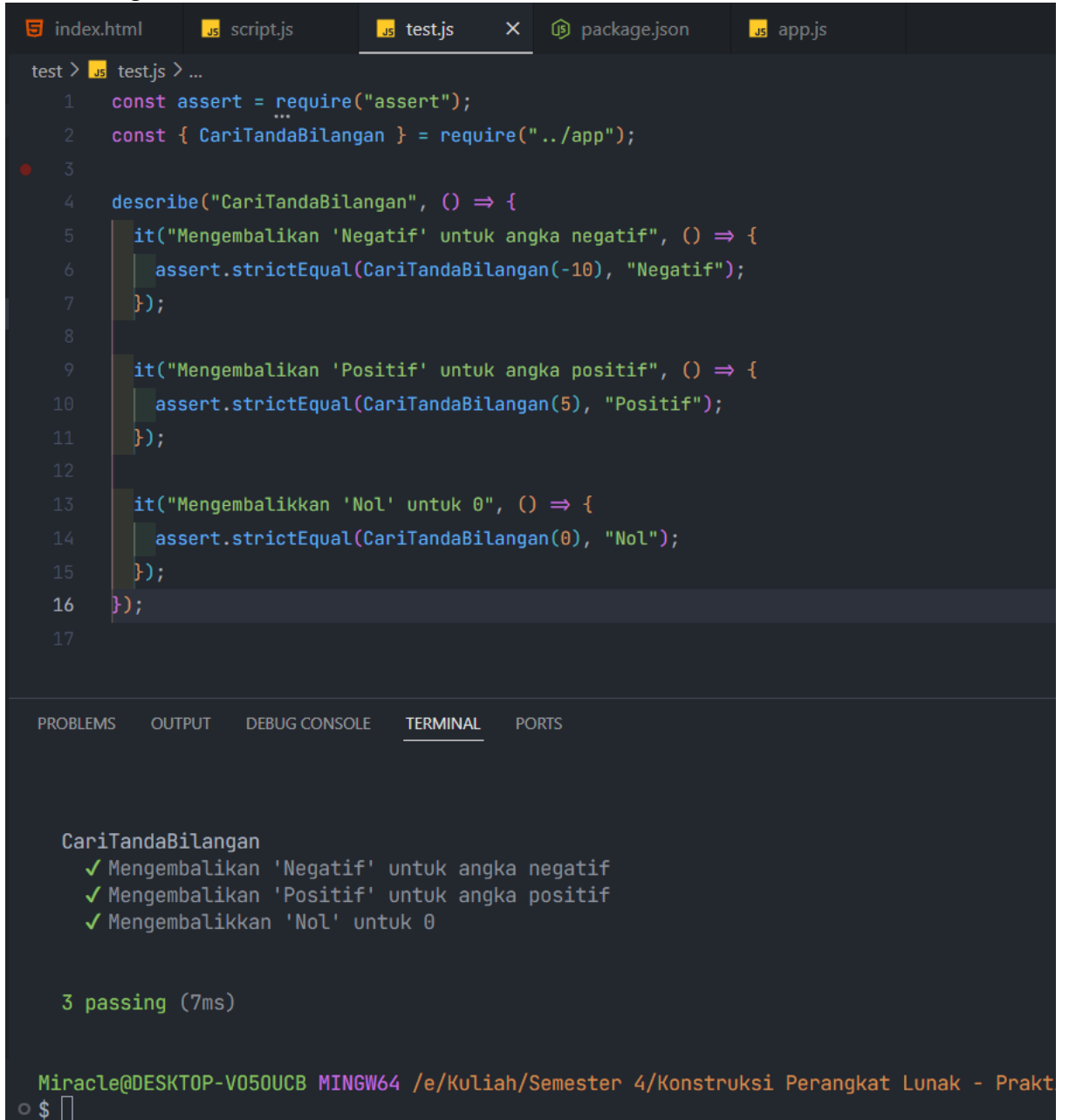
Cek Tanda Bilangan

Positif

5. Software Profilling

Skenario	CPU Usage	Memory Usage
Saat program idle (tidak ada input)	0.1%	8 KB
Setelah input angka dan klik tombol	0.3%	10 KB

6. Unit Testing



The image shows a VS Code editor with a file named `test.js` open. The code defines a function `CariTandaBilangan` and tests it using Jest. The tests check for negative, positive, and zero inputs, expecting 'Negatif', 'Positif', and 'No!' respectively. The terminal output shows that all three tests passed.

```
test > test.js > ...
1  const assert = require("assert");
2  const { CariTandaBilangan } = require("../app");
3
4  describe("CariTandaBilangan", () => {
5    it("Mengembalikan 'Negatif' untuk angka negatif", () => {
6      assert.strictEqual(CariTandaBilangan(-10), "Negatif");
7    });
8
9    it("Mengembalikan 'Positif' untuk angka positif", () => {
10     assert.strictEqual(CariTandaBilangan(5), "Positif");
11   });
12
13   it("Mengembalikan 'No!' untuk 0", () => {
14     assert.strictEqual(CariTandaBilangan(0), "No!");
15   });
16 });
17
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

CariTandaBilangan

- ✓ Mengembalikan 'Negatif' untuk angka negatif
- ✓ Mengembalikan 'Positif' untuk angka positif
- ✓ Mengembalikan 'No!' untuk 0

3 passing (7ms)

Miracle@DESKTOP-V050UCB MINGW64 /e/Kuliah/Semester 4/Konstruksi Perangkat Lunak - Prakt
o \$