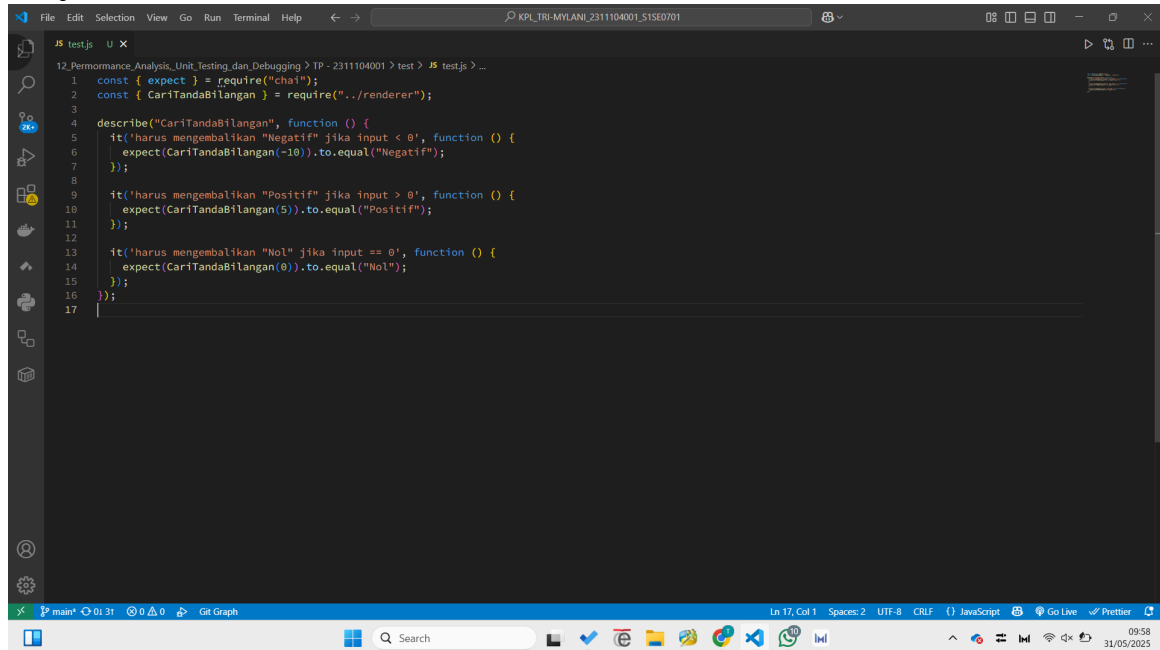


test.js



```

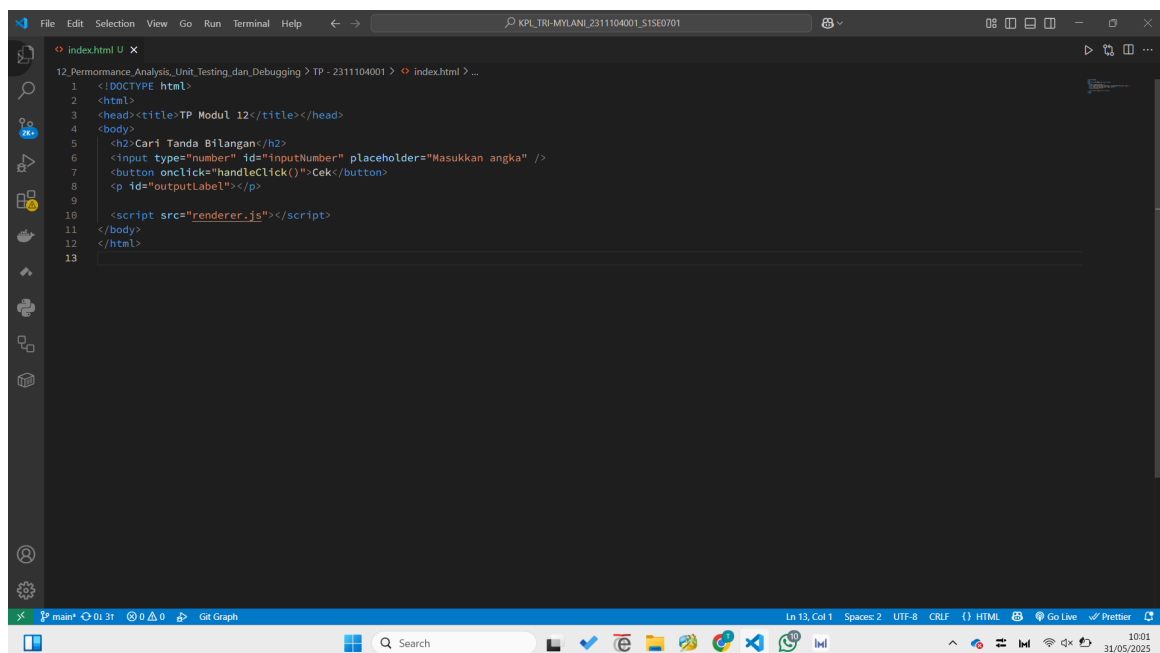
12_Performance_Analysis_Unit_Testing_dan_Debugging > TP - 2311104001 > test > .js test.js > ...
1  const { expect } = require("chai");
2  const { CariTandaBilangan } = require("../renderer");
3
4  describe("CariTandaBilangan", function () {
5    it('harus mengembalikan "Negatif" jika input < 0', function () {
6      expect(CariTandaBilangan(-10)).to.equal("Negatif");
7    });
8
9    it('harus mengembalikan "Positif" jika input > 0', function () {
10     expect(CariTandaBilangan(5)).to.equal("Positif");
11   });
12
13   it('harus mengembalikan "Nol" jika input == 0', function () {
14     expect(CariTandaBilangan(0)).to.equal("Nol");
15   });
16 });
17

```

Penjelasan :

- Kode tersebut merupakan unit test yang dibuat menggunakan framework Mocha dan library Chai untuk menguji fungsi bernama CariTandaBilangan. Fungsi ini diimpor dari file renderer.js, dan pengujiannya dilakukan dengan pendekatan expect dari Chai, yang memungkinkan hasil dari fungsi yang sesuai.
- Dalam test suite yang diberi nama "CariTandaBilangan", terdapat tiga pengujian (test case). Pertama, fungsi diuji dengan input angka negatif (misalnya -10), dan diharapkan mengembalikan string "Negatif". Kedua, fungsi diuji dengan angka positif (misalnya 5), dan hasil yang diharapkan adalah "Positif". Ketiga, fungsi diuji dengan nilai 0 dan harus mengembalikan "Nol".

index.html



```

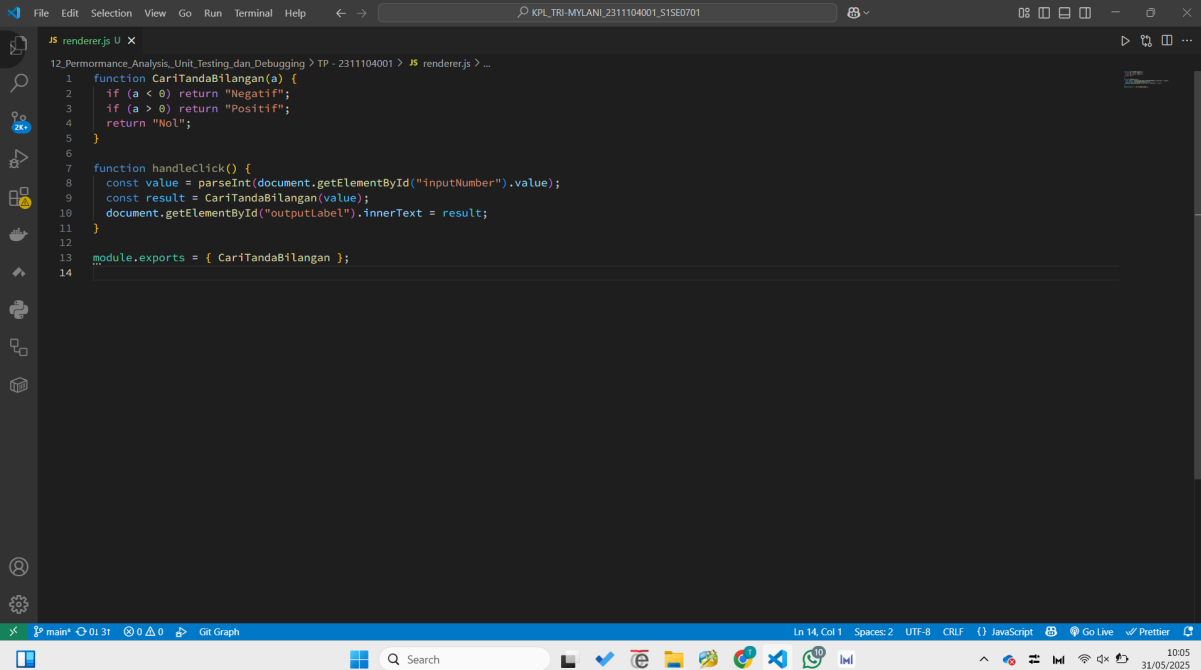
12_Performance_Analysis_Unit_Testing_dan_Debugging > TP - 2311104001 > index.html > ...
1  <!DOCTYPE html>
2  <html>
3  <head><title>TP Modul 12</title></head>
4  <body>
5    <h2>Cari Tanda Bilangan</h2>
6    <input type="number" id="inputNumber" placeholder="Masukkan angka" />
7    <button onclick="handleClick()">Cek</button>
8    <p id="outputLabel"></p>
9
10   <script src="renderer.js"></script>
11 </body>
12 </html>
13

```

Penjelasan :

- Kode HTML tersebut adalah GUI sederhana untuk aplikasi "Cari Tanda Bilangan" yang dibuat menggunakan HTML dan JavaScript.
- Di dalam halaman ini, terdapat elemen-elemen dasar seperti judul (<h2>), sebuah input bertipe number yang memungkinkan pengguna memasukkan sebuah angka, sebuah tombol (<button>) yang ketika diklik akan menjalankan fungsi handleClick(), serta sebuah paragraf (<p>) dengan ID outputLabel yang digunakan untuk menampilkan hasil dari proses pengecekan tanda bilangan.
- Saat pengguna memasukkan angka dan menekan tombol "Cek", skrip JavaScript yang terdapat dalam file renderer.js akan dipanggil untuk memproses input untuk menentukan apakah angka itu positif, negatif, atau nol, lalu menampilkan hasilnya di elemen paragraf

renderer.js



```

1 function CariTandaBilangan(a) {
2   if (a < 0) return "Negatif";
3   if (a > 0) return "Positif";
4   return "Nol";
5 }
6
7 function handleClick() {
8   const value = parseInt(document.getElementById("inputNumber").value);
9   const result = CariTandaBilangan(value);
10  document.getElementById("outputLabel").innerText = result;
11 }
12
13 module.exports = { CariTandaBilangan };
14

```

Penjelasan :

- Kode JavaScript tersebut berisi dua fungsi utama yang digunakan dalam aplikasi "Cari Tanda Bilangan". yaitu CariTandaBilangan(a) untuk menentukan apakah nilai a yang diberikan merupakan bilangan negatif, positif, atau nol. Jika nilai a kurang dari nol, fungsi akan mengembalikan string "Negatif", jika lebih dari nol akan mengembalikan "Positif", dan jika sama dengan nol akan mengembalikan "Nol".
- Fungsi kedua bernama handleClick() adalah fungsi yang dijalankan ketika tombol pada halaman HTML ditekan. Fungsi ini mengambil nilai yang dimasukkan pengguna dari elemen input dengan ID inputNumber, mengubahnya menjadi tipe integer menggunakan parseInt(), lalu memanggil fungsi CariTandaBilangan dengan nilai tersebut.
- Hasil dari fungsi itu kemudian ditampilkan ke pengguna dengan mengubah teks dalam elemen paragraf yang memiliki ID outputLabel.
- module.exports = { CariTandaBilangan }; yang digunakan untuk mengekspor fungsi

CariTandaBilangan

npm init -y

Membuat file konfigurasi package.json secara otomatis untuk proyek Node.js

npm install electron --save-dev

Perintah `npm install electron --save-dev` digunakan untuk menginstal Electron sebagai development dependency dalam proyek Node.js. Setelah perintah ini dijalankan, akan muncul folder baru bernama `node_modules/` yang berisi semua pustaka yang dibutuhkan proyek. Selain itu, file `package-lock.json` akan dibuat atau diperbarui untuk mencatat versi dan struktur dependensi secara rinci, memastikan konsistensi instalasi. File `package.json` juga akan diperbarui, khususnya pada bagian `devDependencies`, yang akan mencantumkan Electron beserta versinya

npx electron main.js

