

2311104025

Eduardo Bagus Prima .J

SE0701


Index.html

```
<> index.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8" />
5    <meta name="viewport" content="width=device-width, initial-scale=1" />
6    <title>GUI Cari Tanda Bilangan</title>
7  </head>
8  <body>
9    <h2>Cari Tanda Bilangan</h2>
10
11    <input type="text" id="inputNumber" placeholder="Masukkan angka" />
12    <button id="btnCheck">Cek Tanda</button>
13    <p>Hasil: <span id="outputLabel"></span></p>
14
15    <script>
16      // Fungsi sesuai instruksi
17      function CariTandaBilangan(a) {
18        if (a < 0) return "Negatif";
19        else if (a > 0) return "Positif";
20        else return "Nol";
21      }
22
23      // Event button click
24      document.getElementById('btnCheck').addEventListener('click', () => {
25        const inputText = document.getElementById('inputNumber').value.trim();
26        // Parse input ke integer
27        const inputNum = parseInt(inputText, 10);
28
29        // Validasi input
30        if (isNaN(inputNum)) {
31          document.getElementById('outputLabel').textContent = "Input bukan angka valid!";
32          return;
33        }
34
35        // Panggil fungsi dan tampilkan hasil
36        const hasil = CariTandaBilangan(inputNum);
37        document.getElementById('outputLabel').textContent = hasil;
38      });
39    </script>
```

Penjelasan:

Kode di atas adalah implementasi antarmuka pengguna (GUI) sederhana berbasis HTML dan JavaScript untuk menentukan tanda dari suatu bilangan. Tampilan ini terdiri dari satu elemen input teks untuk menerima angka dari pengguna, sebuah tombol untuk memicu proses pengecekan, dan sebuah label untuk menampilkan hasilnya. Ketika tombol "Cek Tanda" ditekan, JavaScript akan mengambil nilai dari input, memverifikasinya sebagai angka valid, lalu memanggil fungsi `CariTandaBilangan` yang akan mengembalikan string "Negatif", "Positif", atau "Nol" tergantung pada nilai input. Hasil dari fungsi ini kemudian ditampilkan ke pengguna melalui elemen span yang ada dalam paragraf hasil. Validasi input juga disertakan untuk memastikan bahwa pengguna tidak memasukkan nilai yang bukan angka, sehingga aplikasi tetap memberikan umpan balik yang informatif dan aman.

Cari tanda bilangan.js

```
JS cariTandaBilangan.js >  CariTandaBilangan
1  function CariTandaBilangan(a) {
2      if (a < 0) return "Negatif";
3      else if (a > 0) return "Positif";
4      else return "Nol";
5  }
6
7  module.exports = { CariTandaBilangan };
```

Penjelasan:

Fungsi `CariTandaBilangan` adalah sebuah fungsi yang menerima satu input berupa angka dan mengembalikan sebuah string yang menjelaskan tanda dari angka tersebut. Jika angka yang diberikan kurang dari nol, fungsi ini akan mengembalikan string "Negatif", menandakan bahwa angka tersebut berada di bawah nol. Jika angka lebih besar dari nol,

maka fungsi akan mengembalikan string "Positif", yang berarti angka tersebut bernilai positif. Sedangkan jika angka sama dengan nol, fungsi akan mengembalikan string "Nol". Fungsi ini bekerja dengan cara melakukan pengecekan kondisi secara berurutan menggunakan statement if-else untuk menentukan kategori tanda dari angka yang diberikan. Fungsi ini sangat berguna untuk mengklasifikasikan nilai angka dalam tiga kategori utama tersebut secara sederhana dan langsung.

File test

```
JS cariTandaBilangan.test.js > ...
1  const { CariTandaBilangan } = require('./cariTandaBilangan');
2
3  test('input negatif menghasilkan "Negatif"', () => {
4    expect(CariTandaBilangan(-5)).toBe("Negatif");
5  });
6
7  test('input positif menghasilkan "Positif"', () => {
8    expect(CariTandaBilangan(10)).toBe("Positif");
9  });
10
11 test('input nol menghasilkan "Nol"', () => {
12   expect(CariTandaBilangan(0)).toBe("Nol");
13 });
```

Penjelasan:

serangkaian unit test yang ditulis menggunakan framework Jest untuk menguji fungsi CariTandaBilangan. Fungsi ini diimpor dari file cariTandaBilangan.js dan diuji dalam tiga skenario utama: ketika input bernilai negatif, positif, dan nol. Pada pengujian pertama, nilai -5 diuji untuk memastikan hasilnya adalah "Negatif", sesuai dengan logika fungsi. Pengujian kedua menggunakan nilai 10 untuk memastikan fungsi mengembalikan "Positif", dan pengujian ketiga menggunakan nilai 0 untuk memastikan hasilnya adalah "Nol". Ketiga test case ini dirancang untuk memastikan bahwa fungsi CariTandaBilangan memiliki branch coverage yang baik dan mampu menangani seluruh kemungkinan kondisi input utama dengan benar.

Output:

Cari Tanda Bilangan

Hasil: Negatif

Cari Tanda Bilangan

Hasil: Nol

Cari Tanda Bilangan

Hasil: Positif