

# Debugging Using Jest

## Virtual Internship Experience



Core  
Initiative

# Outline

- 1 Pengertian Debugging
- 2 Tujuan Debugging
- 3 Debugging Menggunakan Jest

# Pengertian *Debugging*



*Debugging* merupakan sebuah metode yang dilakukan oleh para pemrogram dan pengembang perangkat lunak untuk menganalisis alur kerja program, mencari dan mengurangi bug atau kerusakan di dalam sebuah program komputer atau perangkat keras sehingga dapat bekerja sesuai dengan harapan

**Mendeteksi Bug Lebih  
Awal**

**Memberikan Informasi  
Terstruktur**

## **Tujuan *Debugging***

**Mengurangi Resiko  
*Hacker***

**Menghindari Proses  
yang Rumit**

# *Debugging* Menggunakan Jest

**Visual Studio Code's  
Debug Configuration**

**Node's Native Debugger  
& Chrome Inspector**



# Node Native Debugger & Chrome Inspector



1. Membuat *Breakpoint* menggunakan *syntax "debugger"*

```
test('adding 5 + 7 should return 12', () => {  
  debugger  
  // Arrange and Act  
  const test1 = mathOperations.sum(5, 7);  
  
  // Assert  
  expect(test1).toBe(12);  
});
```

2. Eksekusi *syntax debugger* dengan cara memasukkan *syntax* berikut pada terminal / command prompt

```
node --inspect-brk ./node_modules/jest/bin/jest.js --runInBand
```

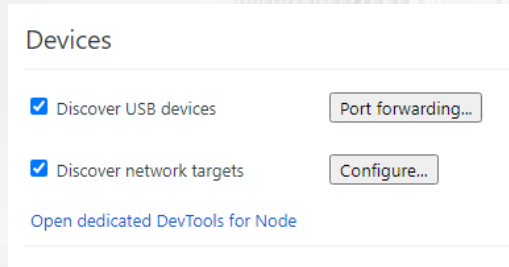
\* *Syntax* ini dapat anda temukan melalui <https://jestjs.io/docs/troubleshooting>

```
PS D:\Project\Vue\tutorial_project> node --inspect-brk ./node_modules/jest/bin/jest.js --runInBand  
Debugger listening on ws://127.0.0.1:9229/ff017b13-2fbb-4c3b-bf91-8375af46b095  
For help, see: https://nodejs.org/en/docs/inspector  
Debugger attached.  
|
```

# Node Native Debugger & Chrome Inspector

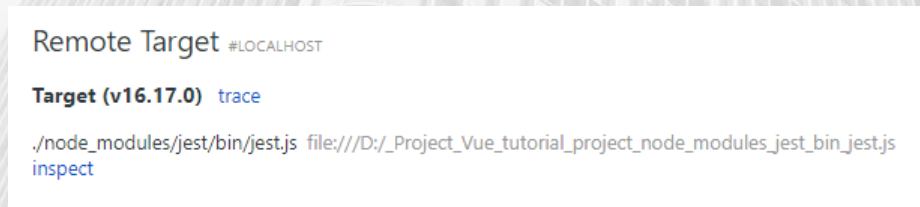


3. Lalu Buka Chrome dan pergi ke : `chrome://inspect/`



\*klik pada bagian *"Open dedicated DevTools for Node"*

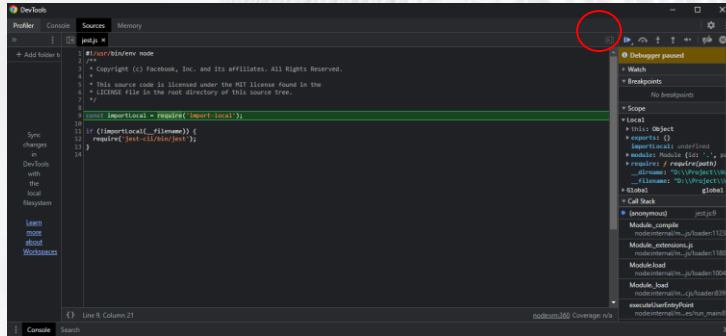
4. Akan tampil *Node Debugger*



\*klik pada bagian *"Inspect"*

# Node Native Debugger & Chrome Inspector

## 5. Akan tampil tab inspeksi



\*klik tombol berikut untuk menjalankan *debugging*

## 6. Breakpoint Akan ditampilkan

```
test('adding 5 + 7 should return 12', () => {  
  debugger  
  // Arrange and Act  
  const test1 = mathOperations.sum(5, 7);  
  
  // Assert  
  expect(test1).toBe(12);  
});
```



\*klik tombol berikut untuk berpindah *step (up/down)*



\*klik tombol berikut untuk melanjutkan *debugging*



# Node Native Debugger & Chrome Inspector

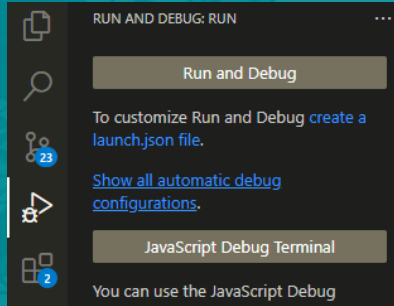


## 7. Debugging Selesai

```
Test Suites: 1 passed, 1 total
Tests:      3 passed, 3 total
Snapshots:  0 total
Time:       95.934 s
Ran all test suites.
```

# Visual Studio Code's Debug Configuration

## 1. Klik Tombol *Run/Debug* pada *sidebar*



\*klik pada bagian "create a launch.json file"

## 2. Masukkan kode berikut pada file launch.json

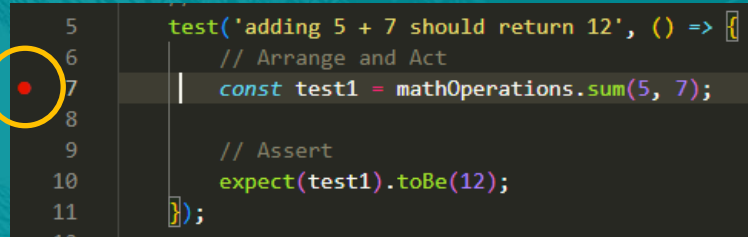
\*anda dapat menemukan kode tersebut pada <https://jestjs.io/docs/troubleshooting>

```
launch.json x
.vscode > { launch.json > ...
1  {
2    "version": "0.2.0",
3    "configurations": [
4      {
5        "name": "Debug Jest Tests",
6        "type": "node",
7        "request": "launch",
8        "runtimeArgs": [
9          "--inspect-brk",
10         "${workspaceRoot}/node_modules/jest/bin/jest.js",
11         "--runInBand"
12       ],
13       "console": "integratedTerminal",
14       "internalConsoleOptions": "neverOpen"
15     ]
16   ]
17 }
```

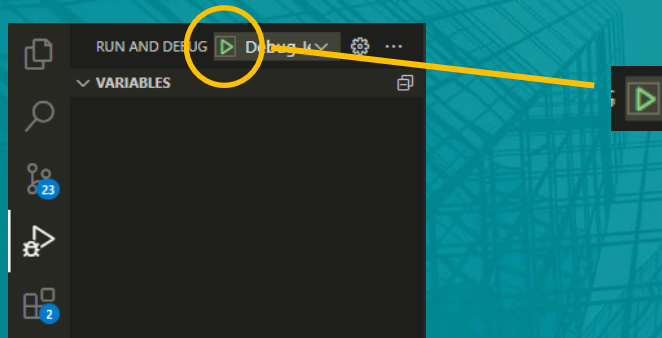
# Visual Studio Code's Debug Configuration

3. Buat *Breakpoint* dengan menekan sebelah kiri dari nomor baris kode hingga seperti berikut

Breakpoint



4. Kembali ke tab *Run/Debug* dan klik tombol berikut untuk menjalankan *debugging*



# Visual Studio Code's Debug Configuration

5. VS Code akan menghighlight kode dengan *breakpoint*

```
5     test('adding 5 + 7 should return 12', () => {  
6         // Arrange and Act  
7         const test1 = mathOperations.sum(5, 7);  
8     }  
9     // Assert  
10    expect(test1).toBe(12);  
11    });
```

6. Control Debugging Flow



Continuous  
Step Over  
Step In  
Step Up  
Restart

7. Debugging Selesai

```
Test Suites: 1 passed, 1 total  
Tests:      3 passed, 3 total  
Snapshots:  0 total  
Time:       136.161 s  
Ran all test suites.
```



**How is it implemented in Core Initiative?**





# Thank You!

