

Referensi Cepat Jest

Test, matcher, mocking, async, dan snapshot

Setup

Instalasi

```
npm install --save-dev jest
# package.json: "scripts": { "test": "jest" }
npx jest # run all tests
npx jest --watch # re-run on changes
```

Penamaan File

***.test.js** File test (pola default)
***.spec.js** Pola test alternatif
__tests__/ Direktori test (ditemukan otomatis)

Menjalankan Test Tertentu

```
npx jest path/to/file.test.js
npx jest --testNamePattern="adds"
npx jest --verbose # detailed output
```

Test Dasar

Struktur Test

```
describe("Calculator", () => {
  test("adds 1 + 2 to equal 3", () => {
    expect(add(1, 2)).toBe(3);
  });
});
```

test vs it

```
test("works correctly", () => { /* ... */ });
it("should work correctly", () => { /* ... */ });
// Both are identical; "it" reads like English
```

Lewati & Fokus

test.skip() Lewati test ini
test.only() Jalankan hanya test ini
describe.skip() Lewati seluruh suite
describe.only() Jalankan hanya suite ini

Matcher

Kesamaan

.toBe(val) Kesamaan ketat (===)
.toEqual(val) Kesamaan mendalam (objek/array)
.toStrictEqual(val) Mendalam + tipe + props undefined
.not.toBe(val) Negasi matcher mana pun

Truthiness

.toBeTruthy() Nilai truthy
.toBeFalsy() Nilai falsy
.toBeNull() Tepat **null**
.toBeUndefined() Tepat **undefined**
.toBeDefined() Bukan **undefined**

Angka

.toBeGreaterThan(n) Lebih besar dari n
.toBeLessThanOrEqual(n) Kurang dari atau sama dengan
.toBeCloseTo(0.3, 5) Perbandingan float (5 digit)

String, Array, Objek

.toMatch(/regex/) String cocok dengan regex
.toContain(item) Array/iterable mengandung item
.toHaveLength(n) Panjang array/string
.toHaveProperty(key, val) Objek memiliki properti
.toMatchObject(obj) Objek mengandung subset

Test Async

async / await

```
test("fetches data", async () => {
  const data = await fetchData();
  expect(data).toEqual({ id: 1 });
});
```

Promise

```
test("resolves to data", () => {
  return expect(fetchData())
    .resolves.toEqual({ id: 1 });
});
```

Rejection

```
test("rejects with error", async () => {
  await expect(fetchBad())
    .rejects.toThrow("Not Found");
});
```

Exception

```
test("throws on invalid input", () => {
  expect(() => validate(null)).toThrow();
  expect(() => validate(null)).toThrow("invalid");
});
```

Mocking

Fungsi Mock

```
const fn = jest.fn();
fn("hello");
expect(fn).toHaveBeenCalledWith("hello");
expect(fn).toHaveBeenCalledTimes(1);
```

Nilai Kembalian Mock

```
const fn = jest.fn()
  .mockReturnValue(42)
  .mockReturnValueOnce(99);
fn(); // 99 (first call)
fn(); // 42 (subsequent)
```

Mocking Module

```
jest.mock("./api");
const { fetchUser } = require("./api");
fetchUser.mockResolvedValue({ name: "Alice" });
```

Matcher Mock

.toHaveBeenCalled() Dipanggil setidaknya sekali
.toHaveBeenCalledTimes(n) Dipanggil tepat n kali
.toHaveBeenCalledWith(args) Dipanggil dengan argumen tertentu
.toHaveBeenLastCalledWith(args) Panggilan terakhir dengan argumen ini

Spy

Memata-matai Method

```
const spy = jest.spyOn(Math, "random")
  .mockReturnValue(0.5);
expect(Math.random()).toBe(0.5);
spy.mockRestore(); // restore original
```

Memata-matai Method Objek

```
const obj = { greet: (n) => `Hi ${n}` };
const spy = jest.spyOn(obj, "greet");
obj.greet("Alice");
expect(spy).toHaveBeenCalledWith("Alice");
```

Snapshot

Snapshot Testing

```
test("renders correctly", () => {
  const tree = renderer.create(<App />).toJSON();
  expect(tree).toMatchSnapshot();
});
```

Inline Snapshot

```
test("formats name", () => {
  expect(formatName("alice"))
    .toMatchInlineSnapshot(`"Alice"`);
});
```

Memperbarui Snapshot

```
npx jest --updateSnapshot # update all
npx jest --updateSnapshot --testNamePattern="renders"
```

Setup & Teardown

Hook Siklus Hidup

```
beforeAll(() => { /* once before all tests */ });
afterAll(() => { /* once after all tests */ });
beforeEach(() => { /* before each test */ });
afterEach(() => { /* after each test */ });
```

Scoping

```
describe("Database", () => {
  beforeEach(() => db.connect());
  afterEach(() => db.disconnect());
  test("reads data", () => { /* ... */ });
});
```

Hook di dalam describe hanya berlaku untuk blok tersebut

Konfigurasi

jest.config.js

```
module.exports = {
  testEnvironment: "node",
  coverageThreshold: {
    global: { branches: 80, lines: 80 }
  },
};
```

Opsi Umum

testEnvironment "node" atau "jsdom" (DOM)
roots Direktori untuk mencari test
collectCoverage Aktifkan pelaporan coverage
coverageDirectory Direktori output untuk coverage
moduleNameMapper Alias path (mis. prefiks @/)
transform Transformasi file (Babel, TS, dll.)
setupFilesAfterFramework Jalankan setup sebelum setiap suite

Coverage

```
npx jest --coverage
npx jest --collectCoverageFrom="src/**/*.js"
```

Referensi Cepat Jest

Pola Umum

Testing Panggilan API

```
jest.mock("./api");
test("loads users", async () => {
  api.getUsers.mockResolvedValue([{id: 1}]);
  const users = await loadUsers();
  expect(users).toHaveLength(1);
});
```

Mock Timer

```
jest.useFakeTimers();
test("delays execution", () => {
  const cb = jest.fn();
  setTimeout(cb, 1000);
  jest.advanceTimersByTime(1000);
  expect(cb).toHaveBeenCalled();
});
```

Test Terparameterisasi

```
test.each([
  [1, 1, 2],
  [2, 3, 5],
])("add(%i, %i) = %i", (a, b, expected) => {
  expect(add(a, b)).toBe(expected);
});
```

Custom Matcher

```
expect.extend({
  toBeWithinRange(received, floor, ceil) {
    const pass = received >= floor
      && received <= ceil;
    return { pass, message: () =>
      `expected ${received} in [${floor},${ceil}]` };
  }
});
```