

LOGBOOK SUPLEMEN PRAKTIKUM — DASAR PEMROGRAMAN JAVASCRIPT

Mata Kuliah: Aplikasi Web dan Mobile | Program Studi Teknik Industri | Universitas Negeri Yogyakarta
Persiapan Pertemuan 4 | Semester Genap 2025/2026

SUPLEMEN PRAKTIKUM · TEKNIK INDUSTRI UNY

Logbook Dasar Pemrograman JavaScript

Identitas Mahasiswa

NAMA LENGKAP

Firza Oktavian Rihantoro

NIM

23051430018

KELAS / ROMBEL

TI-A-2023

TANGGAL PENGERJAAN

27/02/2026



DOSEN PENGAMPU

Dr. Eng. Ir. Aji Ery Burhandenny, S.T., M.AIT.

BAGIAN A

Checklist Materi Suplemen

Centang setiap topik setelah Anda membaca, memahami, dan mencoba contohnya di Browser Console.



Bagian 1 & 2 — Mengapa JavaScript & cara kerja di browser

Konsep

Saya memahami peran JS dalam tiga pilar web dan dapat membuka Browser Console (F12)



Bagian 3 — Variabel: let, const, dan aturan penamaan

Praktik

Saya sudah mencoba mendeklarasikan variabel dan memahami kapan memakai let vs const



Bagian 4 — Tipe Data: Number, String, Boolean

Praktik

Saya memahami jebakan String + Number dan cara konversi dengan Number()



Bagian 5 — Operator: Aritmatika, Perbandingan (==), Logika (&&, ||)

Praktik

Saya sudah mencoba minimal satu perhitungan dan satu ekspresi perbandingan



Praktik

Bagian 6 — Control Flow: if, else if, switch, ternary*Saya memahami urutan else if dan kenapa break diperlukan di switch*

Analisis

Bagian 7 — Studi Kasus Kalkulator OEE (membaca dan memahami seluruh kode)*Saya dapat menjelaskan setiap variabel dan setiap blok if dalam kode OEE tersebut*

Review

Bagian 9 — Membaca seluruh daftar Kesalahan Umum Pemula*Saya sudah mengidentifikasi minimal dua kesalahan yang pernah atau mungkin saya lakukan***BAGIAN A2**

Bukti Pengerjaan Latihan Mandiri

Untuk setiap latihan: (1) centang setelah selesai, (2) jawab pertanyaan uji pemahaman singkat, dan (3) **unggah screenshot output Console** sebagai bukti pengerjaan yang akan tercetak di PDF.

Level 1 — Latihan Dasar

**Latihan 1.1 — Variabel Profil Mesin**

Level 1

Membuat variabel dengan tipe tepat (let/const) dan menampilkan semua dengan console.log()

UJI PEMAHAMAN SINGKAT

Mengapa variabel "kode mesin" sebaiknya dideklarasikan dengan `const` bukan `let`? Jelaskan singkat.

Variabel `"kode mesin"` sebaiknya dideklarasikan dengan `const` karena nilainya tidak dirancang untuk berubah setelah ditetapkan. Dalam konteks sistem produksi, kode mesin biasanya bersifat tetap (misalnya *CNC-01* atau *MILL-02*) dan berfungsi sebagai identitas unik. Dengan menggunakan `const`, kita mencegah perubahan nilai secara tidak sengaja selama program berjalan, sehingga meningkatkan keamanan data dan konsistensi logika program. Selain itu, penggunaan `const` juga membantu memperjelas niat penulisan kode bahwa variabel tersebut bersifat tetap (immutable reference), sedangkan `let` lebih tepat digunakan untuk variabel yang nilainya dapat berubah.

SCREENSHOT OUTPUT CONSOLE

```
// JavaScript
const kodeMesin = "CNC-01";

console.log("Kode mesin yang digunakan:", kodeMesin);

// Jika mencoba mengubah nilainya
kodeMesin = "CNC-02"; // Ini akan menghasilkan error
```

Screenshot · 27/2/2026

Latihan 1.2 — Kalkulator Biaya Material

Level 1

Menghitung biaya per unit, total biaya, dan berat total material; tampilkan di console



UJI PEMAHAMAN SINGKAT

Jika jumlahProduksi diubah menjadi 0, apa yang terjadi pada biayaPerUnit? Apa masalah matematis yang muncul?

Nilai biaya per unit menjadi tidak valid

Secara logika bisnis, tidak mungkin menghitung biaya per unit jika tidak ada produksi

Dalam sistem industri nyata, ini dapat menyebabkan error perhitungan laporan biaya

Karena itu, diperlukan validasi input untuk mencegah nilai 0 sebelum melakukan perhitungan, seperti yang sudah ditambahkan pada kode di atas.

SCREENSHOT OUTPUT CONSOLE



Level 2 — Latihan Menengah



Latihan 2.1 — Sistem Klasifikasi Reject

Level 2

Logika if/else if/else menentukan kategori (Excellent/Acceptable/Warning/Critical) dari reject rate



UJI PEMAHAMAN SINGKAT

Jika Anda membalik urutan kondisi (cek ≥ 5 dulu, baru ≥ 3), apakah hasilnya berbeda? Jelaskan mengapa urutan kondisi else if sangat penting.

Jika urutan kondisi dibalik dengan mengecek ` ≥ 3 ` terlebih dahulu baru kemudian ` ≥ 5 `, maka hasilnya akan berbeda dan menjadi tidak akurat karena struktur `if–else if` bekerja dari atas ke bawah dan akan langsung menjalankan kondisi pertama yang bernilai true lalu menghentikan pengecekan berikutnya. Sebagai contoh, jika jumlah reject adalah 6, seharusnya masuk kategori "KRITIS" (karena ≥ 5), tetapi jika kondisi ` ≥ 3 ` diletakkan lebih dulu, maka 6 sudah memenuhi syarat tersebut sehingga program langsung mengklasifikasikannya sebagai "WARNING" dan tidak pernah mencapai kondisi ` ≥ 5 `. Hal ini menunjukkan bahwa urutan `else if` sangat penting karena menentukan prioritas logika keputusan; kondisi yang lebih tinggi atau lebih spesifik harus ditempatkan lebih dulu agar sistem klasifikasi berjalan sesuai dengan aturan yang diinginkan dan tidak menimbulkan kesalahan dalam pengambilan keputusan.

SCREENSHOT OUTPUT CONSOLE — UJI MINIMAL 2 SKENARIO

Sistem Klasifikasi Reject Produksi

Jumlah Produk Reject: Cek Status

Status: MINOR - Masih dalam batas toleransi

Screenshot · 27/2/2026

Sistem Klasifikasi Reject Produksi

Jumlah Produk Reject: Cek Status

Status: KRITIS - Perlu investigasi dan tindakan korektif segera

Screenshot · 27/2/2026



Latihan 2.2 — Kalkulator Lembur

Level 2

Menghitung total upah lembur berdasarkan jam lembur, dengan tarif 1.5x dan 2x



UJI PEMAHAMAN SINGKAT

Berapa total lembur (Rp) untuk operator dengan gaji pokok Rp 3.500.000 yang lembur 5 jam? Tulis perhitungan manual Anda.

30347/jam

SCREENSHOT OUTPUT CONSOLE — UJI MINIMAL 2 SKENARIO**Kalkulator Upah Lembur**

Gaji Pokok Bulanan (Rp):

Jumlah Jam Lembur:

Total Upah Lembur: Rp 30347

Screenshot · 27/2/2026

Kalkulator Upah Lembur

Gaji Pokok Bulanan (Rp):

Jumlah Jam Lembur:

Total Upah Lembur: Rp 151734

Screenshot · 27/2/2026

BAGIAN B

Uji Pemahaman Kode

Prediksi output kode berikut *tanpa menjalankannya* terlebih dahulu, lalu klik "Periksa".

SOAL B-1 · OPERATOR & TIPE DATA

```
let a = 10;
let b = "5";
let c = a + Number(b);
let d = a + b;
console.log(c);           // Jawaban 1
console.log(d);           // Jawaban 2
console.log(typeof c);   // Jawaban 3
```

JAWABAN 1 — CONSOLE.LOG(C)

15

JAWABAN 2 — CONSOLE.LOG(D)

105

JAWABAN 3 — TYPEOF C

number

SOAL B-2 · CONTROL FLOW

```
let reject = 8;
let total = 200;
let rate = (reject / total) * 100;
if (rate < 1) {
  console.log("Excellent");
} else if (rate < 3) {
  console.log("Acceptable");
} else if (rate < 5) {
  console.log("Warning");
} else {
  console.log("Critical");
}
```

NILAI RATE (%)

4

OUTPUT DI CONSOLE

Warning

SOAL B-3 · SWITCH & LOGIKA

```
let shift = 2;
let isWeekend = true;
let bonus = 0;
switch (shift) {
  case 3: bonus = 50000; break;
  case 2: bonus = 25000; break;
  default: bonus = 0;
}
if (isWeekend && shift === 2) {
  bonus = bonus * 2;
}
console.log(bonus);
```

OUTPUT — CONSOLE.LOG(BONUS)

50000

BAGIAN C Refleksi Per Topik

Tuliskan refleksi jujur untuk setiap topik. Minimal 40 karakter per jawaban.

C-1

Jelaskan dengan kata-kata Anda sendiri: apa perbedaan `let` dan `const`? Berikan satu contoh nyata dari konteks industri untuk masing-masing.

Petunjuk: pikirkan data apa yang berubah vs data apa yang tetap dalam sistem produksi.

Perbedaan utama `let` dan `const` adalah pada kemungkinan perubahan nilainya. `let` digunakan untuk variabel yang nilainya bisa berubah selama program berjalan, sedangkan `const` digunakan untuk variabel yang nilainya tetap dan tidak boleh diubah setelah dideklarasikan. Contoh dalam konteks industri: jumlah produksi harian sebaiknya menggunakan `let` karena nilainya bisa berubah setiap shift, sedangkan kode mesin atau nomor batch produksi sebaiknya menggunakan `const` karena merupakan identitas tetap yang tidak boleh berubah selama proses berjalan.

C - 2

Mengapa menggunakan `==` lebih aman daripada `=?` Tuliskan contoh kode singkat yang menunjukkan perbedaan perilaku keduanya.

Petunjuk: coba bandingkan angka 0 dengan boolean false menggunakan keduanya di Console.

Operator `==` lebih aman daripada `=?` karena `==` membandingkan nilai sekaligus tipe data tanpa melakukan konversi otomatis (type coercion), sedangkan `=?` hanya membandingkan nilai dan akan mengubah tipe data terlebih dahulu jika berbeda, yang bisa menimbulkan hasil tidak terduga. Contohnya, `'0 == false'` menghasilkan `true` karena `false` dikonversi menjadi `'0`, sedangkan `'0 ==? false'` menghasilkan `false` karena tipe datanya berbeda (`number` dan `boolean`). Oleh karena itu, penggunaan `==` lebih direkomendasikan untuk menghindari bug dalam logika program.

C - 3

Dari seluruh materi suplemen, konsep mana yang paling sulit Anda pahami? Jelaskan apa yang membuat konsep tersebut sulit dan bagaimana Anda mencoba mengatasinya.

Konsep yang paling sulit saya pahami adalah alur logika pada struktur `if–else if` yang bertingkat, awal dan hanya menjalankan kondisi pertama yang bernilai true, sehingga jika urutannya salah, hasilnya bisa berbeda dari yang diharapkan meskipun secara logika terlihat benar. Untuk mengatasinya, saya mencoba menelusuri kode secara manual (dry run) dengan memasukkan contoh nilai, menghitung langkah demi langkah, serta menguji langsung di console agar bisa melihat bagaimana setiap kondisi dievaluasi dan memahami pentingnya urutan dalam pengambilan keputusan.

TINGKAT KESULITAN MATERI (PILIH SATU)

 Mudah dipahami

 Butuh usaha

 Cukup menantang

 Sangat sulit

C - 4

Dari latihan mandiri Bagian 8, pilih satu soal yang sudah Anda kerjakan. Tulis ulang kode solusi Anda dan jelaskan logika yang Anda gunakan.

Petunjuk: salin kode dari VS Code / Console Anda ke sini, lalu jelaskan baris-baris kuncinya.

```
let reject = 8;  
let total = 200;  
  
let rate = (reject / total) * 100;  
  
if (rate < 1) {  
    console.log("Excellent");  
} else if (rate < 3) {  
    console.log("Good");  
} else if (rate < 5) {  
    console.log("Warning");  
} else {  
    console.log("Critical");  
}  
  
console.log("Rate:", rate + "%");
```

Pertama, saya menghitung persentase reject menggunakan rumus $(\text{reject} / \text{total}) \times 100$ untuk mendapatkan nilai dalam bentuk persen. Setelah itu, saya menggunakan struktur if–else if untuk mengklasifikasikan hasil berdasarkan batas tertentu. Logika berjalan dari atas ke bawah, sehingga kondisi dengan batas terkecil dicek terlebih dahulu. Jika nilai rate kurang dari 1%, maka status "Excellent", jika kurang dari 3% maka "Good", jika kurang dari 5% maka "Warning", dan selain itu dianggap "Critical". Dengan contoh nilai 8 dari 200, diperoleh rate 4%, sehingga masuk kategori "Warning" karena 4 tidak kurang dari 1 atau 3, tetapi kurang dari 5.

Bagian D — Refleksi Akhir & Rencana Belajar

Tulis secara jujur: apa yang paling berkesan dari suplemen ini, dan apa yang akan Anda lakukan sebelum Pertemuan 4 untuk memastikan diri Anda siap?

saya akan mempelajari lebih dalam

Firza Oktavian Rihantoro
23051430018

Diperiksa oleh Dosen Pengampu
Dr. Eng. Ir. Aji Ery Burhandenny, S.T., M.AIT.
27 Februari 2026

Dokumen ini dicetak dari Logbook Digital Suplemen Praktikum — Aplikasi Web dan Mobile, Program Studi Teknik Industri, Universitas Negeri Yogyakarta