

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

Ahmad Habib Ridho

NIM

23051430024

KELAS / ROMBEL

TI-A1 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 contoh kodenya 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**Bagian 6 — Control Flow: if, else if, switch, ternary**

Praktik

Saya memahami urutan else if dan kenapa break diperlukan di switch**Bagian 7 — Studi Kasus Kalkulator OEE (membaca dan memahami seluruh kode)**

Analisis

Saya dapat menjelaskan setiap variabel dan setiap blok if dalam kode OEE tersebut**Bagian 9 — Membaca seluruh daftar Kesalahan Umum Pemula**

Review

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 menggunakan **const** karena nilainya bersifat tetap dan tidak berubah selama program berjalan. Penggunaan **const** mencegah perubahan nilai secara tidak sengaja sehingga menjaga konsistensi dan keamanan data dalam sistem.

SCREENSHOT OUTPUT CONSOLE

```
1 const kodeMesin = "CNC-01";
2 const tipeMesin = "CNC Milling";
3 const tahunProduksi = 2022;
4 let [jumlahProduksi, biayaPerUnit] = [100, 150000];
5 console.log("Kode Mesin:", kodeMesin);
6 console.log("Tipe Mesin:", tipeMesin);
7 console.log("Tahun Produksi:", tahunProduksi);
8 console.log("Jumlah Produksi:", jumlahProduksi);
9 console.log("Biaya per Unit:", biayaPerUnit);
10 console.log("Status Mesin:", statusMesin);
11
12 // Kode Mesin: "CNC-01"
13 // Tipe Mesin: "CNC Milling"
14 // Tahun Produksi: 2022
15 // Jumlah Produksi: 100
16 // Biaya per Unit: 150000
17 // Status Mesin: "Aktif"
18 console.log()
```

Screenshot · 28/2/2026



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?

Jika jumlahProduksi diubah menjadi 0, maka perhitungan biayaPerUnit akan menghasilkan Infinity karena terjadi pembagian dengan nol. Secara matematis, pembagian dengan nol tidak terdefinisi dan dapat menyebabkan kesalahan logika dalam program.

SCREENSHOT OUTPUT CONSOLE

```
1 // Kode Mesin: "CNC-01"
2 // Tipe Mesin: "CNC Milling"
3 // Tahun Produksi: 2022
4 // Jumlah Produksi: 100
5 // Biaya per Unit: 150000
6 // Status Mesin: "Aktif"
7 console.log()
8
9 // Menghitung biaya per unit
10 let totalBiayaPerUnit = 150000;
11 let jumlahProduksi = 100;
12 let [jumlahProduksi, biayaPerUnit] = [100, 150000];
13 let totalBiaya = biayaPerUnit * jumlahProduksi;
14 let totalBerat = jumlahProduksi * 1000000;
15 console.log("Total Biaya Material:", totalBiaya, "Rp");
16 console.log("Total Berat Material:", totalBerat, "kg");
17 console.log("Biaya per Unit:", biayaPerUnit, "Rp");
18 console.log("Berat per Unit:", 1000000, "kg");
19
20 // Total Biaya Material: 15000000
21 // Total Berat Material: 100000000
22 // Biaya per Unit: 150000
23 // Berat per Unit: 1000000
24 console.log()
```

Screenshot · 28/2/2026

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.

Urutan kondisi `else if` sangat penting karena JavaScript akan mengecek kondisi dari atas ke bawah dan berhenti pada kondisi pertama yang bernilai `true`. Jika urutan dibalik, hasil kategori bisa menjadi salah karena kondisi tertentu mungkin sudah terpenuhi lebih dulu.

SCREENSHOT OUTPUT CONSOLE — UJI MINIMAL 2 SKENARIO

```
// SKENARIO 1 - EXCELLENT
let reject = 1;
let total = 200;

let rate = (reject / total) * 100;

console.log("Reject:", reject);
console.log("Total:", total);
console.log("Reject Rate:", rate, "%");

if (rate < 1) {
  console.log("Kategori: Excellent");
} else if (rate < 3) {
  console.log("Kategori: Acceptable");
} else if (rate < 5) {
  console.log("Kategori: Warning");
} else {
  console.log("Kategori: Critical");
}
```

Reject: 1
Total: 200
Reject Rate: 0.5 - %
Kategori: Excellent

Screenshot · 28/2/2026

```
// SKENARIO 2 - WARNING
let reject = 8;
let total = 200;

let rate = (reject / total) * 100;

console.log("Reject:", reject);
console.log("Total:", total);
console.log("Reject Rate:", rate, "%");

if (rate < 1) {
  console.log("Kategori: Excellent");
} else if (rate < 3) {
  console.log("Kategori: Acceptable");
} else if (rate < 5) {
  console.log("Kategori: Warning");
} else {
  console.log("Kategori: Critical");
}
```

Reject: 8
Total: 200
Reject Rate: 4 - %
Kategori: Warning

Screenshot · 28/2/2026



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.

upah/jam : $3.500.000 / 173 = \text{Rp } 20.231$ perjam

jam 1-3 = $3 \times 1,5 \times 20.231 = \text{Rp } 91.039$

jam 4-5 = $2 \times 2 \times 20.231 = 80.924$

Total = $91.039 + 80.924$

= Rp 171.963

SCREENSHOT OUTPUT CONSOLE — UJI MINIMAL 2 SKENARIO

```
> // SKENARIO 1 - LEMBUR ≤ 3 JAM
let gajiBulanan = 3500000;
let jamLembur = 2;

// asumsi 173 jam kerja per bulan
let upahPerJam = gajiBulanan / 173;

let totalLembur = 0;

if (jamLembur <= 3) {
  totalLembur = jamLembur * 1.5 * upahPerJam;
} else {
  let lemburPertama = 3 * 1.5 * upahPerJam;
  let lemburSisanya = (jamLembur - 3) * 2 * upahPerJam;
  totalLembur = lemburPertama + lemburSisanya;
}

console.log("Gaji Bulanan: Rp", gajiBulanan);
console.log("Jam Lembur:", jamLembur);
console.log("Upah per Jam: Rp", Math.round(upahPerJam));
console.log("Total Lembur: Rp", Math.round(totalLembur));

Gaji Bulanan: Rp - 3500000
Jam Lembur: - 2
Upah per Jam: Rp - 20231
Total Lembur: Rp - 60694
< console.log
```

Screenshot · 28/2/2026

```
Successfully preconnected to https://fonts.googleapis.com/
> // SKENARIO 2 - LEMBUR > 3 JAM
let gajiBulanan = 3500000;
let jamLembur = 5;

// asumsi 173 jam kerja per bulan
let upahPerJam = gajiBulanan / 173;

let totalLembur = 0;

if (jamLembur <= 3) {
  totalLembur = jamLembur * 1.5 * upahPerJam;
} else {
  let lemburPertama = 3 * 1.5 * upahPerJam;
  let lemburSisanya = (jamLembur - 3) * 2 * upahPerJam;
  totalLembur = lemburPertama + lemburSisanya;
}

console.log("Gaji Bulanan: Rp", gajiBulanan);
console.log("Jam Lembur:", jamLembur);
console.log("Upah per Jam: Rp", Math.round(upahPerJam));
console.log("Total Lembur: Rp", Math.round(totalLembur));

Gaji Bulanan: Rp - 3500000
Jam Lembur: - 5
Upah per Jam: Rp - 20231
Total Lembur: Rp - 171965
< console.log
```

Screenshot · 28/2/2026

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.

let digunakan untuk variabel yang nilainya dapat berubah selama program berjalan, sedangkan const digunakan untuk variabel yang nilainya tetap dan tidak boleh diubah. Dalam konteks industri, jumlah produksi harian cocok menggunakan let karena nilainya berubah setiap hari. Sedangkan kode mesin atau ID produk sebaiknya menggunakan const karena bersifat tetap dan tidak berubah dalam sistem produksi.

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 `===` membandingkan nilai sekaligus tipe data, sedangkan `==` hanya membandingkan nilai dan dapat melakukan konversi tipe data secara otomatis. Hal ini dapat menyebabkan hasil yang tidak terduga. Oleh karena itu, penggunaan `===` lebih aman karena menghindari kesalahan akibat konversi tipe data otomatis.

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 menantang bagi saya adalah memahami alur control flow dengan beberapa kondisi else if. Kesulitannya terletak pada memahami urutan evaluasi kondisi dari atas ke bawah dan bagaimana program berhenti pada kondisi pertama yang terpenuhi. Untuk mengatasinya, saya mencoba mengubah nilai variabel di Console dan mengamati perubahan output agar lebih memahami logika program.

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("Acceptable");  
} else if (rate < 5) {  
  console.log("Warning");  
} else {  
  console.log("Critical");  
}
```

Logika program dimulai dengan menghitung persentase reject. Setelah itu program mengevaluasi kondisi secara berurutan dari batas terendah. Program akan menampilkan kategori berdasarkan kondisi pertama yang terpenuhi. Dengan reject 8 dari total 200, diperoleh 4%, sehingga masuk kategori "Warning".

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?

Hal yang paling berkesan dari suplemen ini adalah saya mulai memahami bagaimana logika pemrograman digunakan dalam konteks industri seperti perhitungan lembur dan klasifikasi kualitas produk. Saya menyadari bahwa kesalahan kecil dalam tipe data atau urutan kondisi dapat menghasilkan output yang berbeda.

Ahmad Habib Ridho
23051430024

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