

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

Sesa Nugroho Satriva

NIM

23051430031

KELAS / ROMBEL

TI-A2 2026

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.

Karena nilai variabel "kode mesin" tidak berubah/tetap

SCREENSHOT OUTPUT CONSOLE

NAME	VALUE	TIME
Kode Mesin	MSN-001	Blalihun.js:17
Tahun Pembuatan	2022	Blalihun.js:18
Status Mesin	running	Blalihun.js:19
Kapasitas Produksi/Dlm	500 unit	Blalihun.js:20
Penggunaan Kapasitas Saat Ini	75%	Blalihun.js:21

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?

Karena pada logika perhitungan biayaPerUnit tidak lagi dibagi dengan jumlahProduksi, maka tidak ada pembagian dengan nol. Tidak muncul Infinity, dan tidak ada error matematis pada variabel tersebut. Yang berubah adalah totalBerat = $0,35 \times 0 = 0$ dan totalBiaya = $0 \times 15000 = 0$

SCREENSHOT OUTPUT CONSOLE

```
www.KALKULATOR_BIAYA_MATERIAL.***  
Biaya material per unit : Rp 5,250  
Total berat material : 77 kg  
Total biaya material hari ini : Rp 1,155,000
```

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.

Bisa berbeda. Karena `if–else if` berhenti di kondisi pertama yang terpenuhi. Kalau Anda cek $>= 3$ dulu lalu $>= 5$, maka nilai 6% akan masuk ke $>= 3$ dan tidak pernah sampai ke $>= 5$. Urutan menentukan hasil akhir. Jika salah urut, klasifikasinya bisa keliru.

SCREENSHOT OUTPUT CONSOLE — UJI MINIMAL 2 SKENARIO

```
Logbook Suplemen — Dasar | 127.0.0.1:5500/PermenaraO4/Supplementen_23051430031_SesaNugrohoSatriva/Blathan.html | Ses - Begin 8. Latihan Mencari | +
```

Sesa Nugroho Satriva
23051430031
Bagian 8. Latihan Mandiri Teststruktur

*** SISTEM KLAIFIAKAS REJECT ***
Total Produk : 1000 unit
Jumlah Reject : 10 unit
Reject Rate : 1.00%
Kategori : Acceptable
Timakar : Monitor lebih ketat

Screenshot · 28/2/2026

```
Logbook Suplemen — Dasar | 127.0.0.1:5500/PermenaraO4/Supplementen_23051430031_SesaNugrohoSatriva/Blathan.html | Ses - Begin 8. Latihan Mencari | +
```

Sesa Nugroho Satriva
23051430031
Bagian 8. Latihan Mandiri Teststruktur

*** SISTEM KLAIFIAKAS REJECT ***
Total Produk : 1000 unit
Jumlah Reject : 35 unit
Reject Rate : 3.50%
Kategori : Warnings
Timakar : Investigasi proses

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 per jam adalah $3.500.000 \div 173 = 20.231,21$, sehingga untuk 5 jam lembur dihitung 3 jam pertama $\times 1,5 \times 20.231,21$ dan 2 jam berikutnya $\times 2 \times 20.231,21$, totalnya sekitar Rp 171.965.

SCREENSHOT OUTPUT CONSOLE — UJI MINIMAL 2 SKENARIO

```
Logbook Suplemen — Dasar | 127.0.0.1:5500/PermenaraO4/Supplementen_23051430031_SesaNugrohoSatriva/Blathan.html | Ses - Begin 8. Latihan Mencari | +
```

Sesa Nugroho Satriva
23051430031
Bagian 8. Latihan Mandiri Teststruktur

*** KALKULATOR UPAH LEMBUR ***
Gaji Pokok : Rp 3,500,000
Upah per Jam : Rp 20231,21
Jam Lembur : 5 jam
Total Upah Lembur : Rp 171,964

Screenshot · 28/2/2026

```
Logbook Suplemen — Dasar | 127.0.0.1:5500/PermenaraO4/Supplementen_23051430031_SesaNugrohoSatriva/Blathan.html | Ses - Begin 8. Latihan Mencari | +
```

Sesa Nugroho Satriva
23051430031
Bagian 8. Latihan Mandiri Teststruktur

*** KALKULATOR UPAH LEMBUR ***
Gaji Pokok : Rp 3,500,000
Upah per Jam : Rp 20231,21
Jam Lembur : 5 jam
Total Upah Lembur : Rp 171,965

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 bisa berubah selama program berjalan, sedangkan const digunakan untuk nilai yang tidak boleh diubah setelah didefinisikan. Contoh konteks industri untuk let: jumlah produksi harian di lini manufaktur, karena nilainya bisa bertambah seiring update data, misalnya let totalProduksiHarian = 1200; lalu diperbarui ketika ada input tambahan. Contoh untuk const: tarif upah per jam yang sudah ditetapkan perusahaan dalam periode tertentu, misalnya const upahPerJam = 20231; karena nilainya tetap dan tidak seharusnya berubah di dalam program.

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.

=== lebih aman karena membandingkan nilai sekaligus tipe datanya tanpa melakukan konversi otomatis, sedangkan == melakukan type coercion sehingga bisa menghasilkan perbandingan yang tidak diharapkan. Pada ==, JavaScript mengubah tipe data terlebih dahulu sebelum membandingkan, sehingga hasilnya bisa terlihat "benar" padahal tipenya berbeda. Pada ===, nilai dan tipe harus sama persis agar hasilnya true.

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.

Di bagian switch, tapi sebenarnya saya kesulitan memahami karena basic saya logic menggunakan python jadi hanya sulit untuk penyesuaianya

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.

Latihan 3.2: Mini Program Pemilihan Prioritas Pekerjaan

```
const mesinA = {
    nama: "Mesin A",
    jamOperasional: 2500,
    breakdown: 7
};

const mesinB = {
    nama: "Mesin B",
    jamOperasional: 1800,
    breakdown: 6
};

const mesinC = {
    nama: "Mesin C",
    jamOperasional: 1200,
    breakdown: 2
};

function tentukanPrioritas(mesin) {
    let prioritas = "";

    if (mesin.jamOperasional > 2000 && mesin.breakdown > 5) {
        prioritas = "PRIORITAS TINGGI";
    } else if (mesin.jamOperasional > 2000 || mesin.breakdown > 5) {
        prioritas = "PRIORITAS SEDANG";
    } else {
        prioritas = "PRIORITAS RENDAH";
    }

    console.log("== " + mesin.nama + " ==");
    console.log("Jam Operasional : ", mesin.jamOperasional, "jam");
    console.log("Frekuensi Breakdown : ", mesin.breakdown, "kali");
    console.log("Hasil : ", prioritas);
    console.log("-----");
}

tentukanPrioritas(mesinA);
tentukanPrioritas(mesinB);
tentukanPrioritas(mesinC);
```

Kode ini menentukan prioritas perawatan mesin dari dua hal yaitu jam operasional dan jumlah breakdown.

Jika jam lebih dari 2000 dan breakdown lebih dari 5, maka dianggap berisiko tinggi sehingga masuk PRIORITAS TINGGI. Jika hanya salah satu yang melewati batas, maka PRIORITAS SEDANG. Jika keduanya masih di bawah batas, maka PRIORITAS RENDAH.

Mesin A masuk tinggi karena jam dan breakdown sama-sama tinggi. Mesin B masuk sedang karena hanya breakdown yang tinggi. Mesin C masuk rendah karena keduanya masih rendah.

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?

Bagian latihan nya karena menantang dan menguji pemahaman saya terkait materi2 javascript didalam suplemen dengan case uji yang berbeda tapi tetap menggunakan dasar nya



Sesa Nugroho Satriva
23051430031

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