

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

Ardita Natalia

NIM

23051430022

KELAS / ROMBEL

Teknik Industri – 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 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

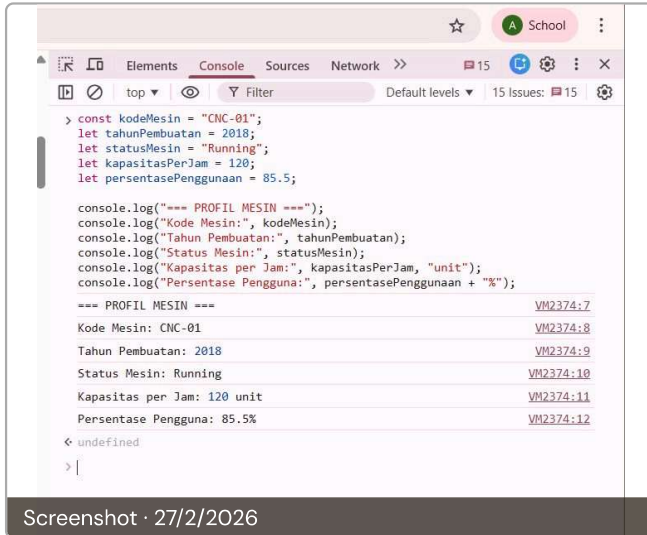
Membuat variabel dengan tipe tepat (let/const) dan menampilkan semua dengan console.log()

UJI PEMAHAMAN SINGKAT

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

Karena kode mesin itu identitas tetap, tidak berubah-ubah selama program jalan. Kalau pakai `const`, nilainya tidak bisa diganti lagi, jadi lebih aman dan tidak ketimpa tanpa sengaja. Kalau pakai `let`, nanti bisa diubah, padahal harusnya kode mesin itu tetap.

SCREENSHOT OUTPUT CONSOLE



```
> const kodeMesin = "CNC-01";
let tahunPembuatan = 2018;
let statusMesin = "Running";
let kapasitasPerJam = 120;
let persentasePenggunaan = 85.5;

console.log("=== PROFIL MESIN ===");
console.log("Kode Mesin:", kodeMesin);
console.log("Tahun Pembuatan:", tahunPembuatan);
console.log("Status Mesin:", statusMesin);
console.log("Kapasitas per Jam:", kapasitasPerJam, "unit");
console.log("Persentase Pengguna:", persentasePenggunaan + "%");

=== PROFIL MESIN ===
Kode Mesin: CNC-01
Tahun Pembuatan: 2018
Status Mesin: Running
Kapasitas per Jam: 120 unit
Persentase Pengguna: 85.5%
```

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?

Masalah matematis muncul kalau biaya per unit dihitung dengan cara $\text{totalBiaya} / \text{jumlahProduksi}$, karena itu berarti membagi dengan 0, dan pembagian dengan nol tidak bisa (hasilnya jadi Infinity atau tidak valid).

SCREENSHOT OUTPUT CONSOLE

```
> const hargaPerKg = 15000;
const kebutuhanPerUnit = 0.35;
const jumlahProduksi = 220;

let biayaMaterialPerUnit = hargaPerKg * kebutuhanPerUnit;
let totalBeratMaterial = kebutuhanPerUnit * jumlahProduksi;
let totalBiayaMaterial = totalBeratMaterial * hargaPerKg;

console.log("===== LAPORAN BIAYA MATERIAL =====");
console.log("Biaya material per unit: Rp", biayaMaterialPerUnit);
console.log("Total berat material:", totalBeratMaterial, "kg");
console.log("Total biaya material: Rp", totalBiayaMaterial);

===== LAPORAN BIAYA MATERIAL =====
Biaya material per unit: Rp 5250
Total berat material: 77 kg
Total biaya material: Rp 1155000
```

Screenshot · 27/2/2026

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.

Karena `else if` itu dicek dari atas ke bawah. Begitu ada yang benar (`true`), yang bawahnya tidak bakal dicek lagi. Jadi kalau urutannya salah, misalnya kondisi yang lebih umum ditaruh dulu, nanti bisa salah masuk kategori. Makanya urutan itu penting banget biar hasil klasifikasinya tidak keliru.

SCREENSHOT OUTPUT CONSOLE — UJI MINIMAL 2 SKENARIO

```

let skenario = [
  { totalProduksi: 1000, jumlahReject: 405 },
  { totalProduksi: 1000, jumlahReject: 1505 },
  { totalProduksi: 1000, jumlahReject: 2505 }
];

for (let i = 0; i < skenario.length; i++) {
  let total = skenario[i].totalProduksi;
  let reject = skenario[i].jumlahReject;
  let rate = (reject / total) * 100;

  let kategori, tindakan;

  if (rate < 3) {
    kategori = "EXCELLENT";
    tindakan = "Tindak aja Tindakan";
  } else if (rate >= 3 && rate < 5) {
    kategori = "ACCEPTABLE";
    tindakan = "Monitor Jodoh kejut";
  } else if (rate >= 5 && rate < 10) {
    kategori = "CRITICAL";
    tindakan = "Investigasi proses";
  } else {
    kategori = "CRITICAL";
    tindakan = "Monitor Jodoh kejut, Lakukan RCA";
  }

  console.log("Rate:", rate, "Kategori:", kategori, "Tindakan:", tindakan);
}

```

ScreenShot · 27/2/2026

```

let skenario = [
  { totalProduksi: 1000, jumlahReject: 405 },
  { totalProduksi: 1000, jumlahReject: 1505 },
  { totalProduksi: 1000, jumlahReject: 2505 }
];

for (let i = 0; i < skenario.length; i++) {
  let total = skenario[i].totalProduksi;
  let reject = skenario[i].jumlahReject;
  let rate = (reject / total) * 100;

  let kategori, tindakan;

  if (rate < 3) {
    kategori = "EXCELLENT";
    tindakan = "Tindak aja Tindakan";
  } else if (rate >= 3 && rate < 5) {
    kategori = "ACCEPTABLE";
    tindakan = "Monitor Jodoh kejut";
  } else if (rate >= 5 && rate < 10) {
    kategori = "CRITICAL";
    tindakan = "Investigasi proses";
  } else {
    kategori = "CRITICAL";
    tindakan = "Monitor Jodoh kejut, Lakukan RCA";
  }

  console.log("Rate:", rate, "Kategori:", kategori, "Tindakan:", tindakan);
}

```

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.

Upah / Jam = $3.500.000 / 173 = 20.231,21$

Jam 1-3 = $3 \times (1.5 \times 20.231,21)$
 = $3 \times 30.346,82$
 = 91.040,46

Jam 4-5 = $2 \times (2 \times 20.231,21)$
 = $2 \times 40.462,42$
 = 80.924,84

Total Lembur
 = $91.040,46 + 80.924,84$
 = 171.965,30
 = 171.965

SCREENSHOT OUTPUT CONSOLE — UJI MINIMAL 2 SKENARIO

```
> console.log("=== SKENARIO 1 ===");
let gajiPokok = 3500000;
let upahPerJam = gajiPokok / 173;
let n = 3;

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

console.log("Upah per Jam", upahPerJam.toFixed(2));
console.log("Jam Lembur", n);
console.log("Total Upah Lembur: Rp", totalLembur.toFixed(0));
```

==== SKENARIO 1 ===

Upah per Jam: 20231.21

Jam Lembur: 3

Total Upah Lembur: Rp 91040

==== SKENARIO 2 ===

Upah per Jam: 20231.21

Jam Lembur: 5

Total Upah Lembur: Rp 171965

==== SKENARIO 3 ===

Upah per Jam: 20231.21

Jam Lembur: 7

Total Upah Lembur: Rp 252767

==== SKENARIO 4 ===

Upah per Jam: 20231.21

Jam Lembur: 9

Total Upah Lembur: Rp 333569

==== SKENARIO 5 ===

Upah per Jam: 20231.21

Jam Lembur: 11

Total Upah Lembur: Rp 414371

==== SKENARIO 6 ===

Upah per Jam: 20231.21

Jam Lembur: 13

Total Upah Lembur: Rp 495173

==== SKENARIO 7 ===

Upah per Jam: 20231.21

Jam Lembur: 15

Total Upah Lembur: Rp 575975

==== SKENARIO 8 ===

Upah per Jam: 20231.21

Jam Lembur: 17

Total Upah Lembur: Rp 656777

==== SKENARIO 9 ===

Upah per Jam: 20231.21

Jam Lembur: 19

Total Upah Lembur: Rp 737579

==== SKENARIO 10 ===

Upah per Jam: 20231.21

Jam Lembur: 21

Total Upah Lembur: Rp 818381

```
> console.log("=== SKENARIO 2 ===");
let gajiPokok = 3500000;
let upahPerJam = gajiPokok / 173;
let n = 5;

let totalLembur;

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

console.log("Upah per Jam", upahPerJam.toFixed(2));
console.log("Jam Lembur", n);
console.log("Total Upah Lembur: Rp", totalLembur.toFixed(0));
```

==== SKENARIO 2 ===

Upah per Jam: 20231.21

Jam Lembur: 5

Total Upah Lembur: Rp 171965

==== SKENARIO 3 ===

Upah per Jam: 20231.21

Jam Lembur: 7

Total Upah Lembur: Rp 252767

==== SKENARIO 4 ===

Upah per Jam: 20231.21

Jam Lembur: 9

Total Upah Lembur: Rp 333569

==== SKENARIO 5 ===

Upah per Jam: 20231.21

Jam Lembur: 11

Total Upah Lembur: Rp 414371

==== SKENARIO 6 ===

Upah per Jam: 20231.21

Jam Lembur: 13

Total Upah Lembur: Rp 495173

==== SKENARIO 7 ===

Upah per Jam: 20231.21

Jam Lembur: 15

Total Upah Lembur: Rp 575975

==== SKENARIO 8 ===

Upah per Jam: 20231.21

Jam Lembur: 17

Total Upah Lembur: Rp 656777

==== SKENARIO 9 ===

Upah per Jam: 20231.21

Jam Lembur: 19

Total Upah Lembur: Rp 737579

==== SKENARIO 10 ===

Upah per Jam: 20231.21

Jam Lembur: 21

Total Upah Lembur: Rp 818381

==== SKENARIO 11 ===

Upah per Jam: 20231.21

Jam Lembur: 23

Total Upah Lembur: Rp 899183

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)

JAWABAN 2 — CONSOLE.LOG(D)

JAWABAN 3 — TYPEOF C

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 (%)

OUTPUT DI CONSOLE

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.

Menurut saya, perbedaan let dan const itu ada di apakah nilainya bisa berubah atau tidak. let dipakai kalau nilainya masih bisa diganti selama program berjalan, sedangkan const dipakai kalau nilainya sudah tetap dan tidak boleh diubah lagi. Misalnya di konteks industri, kapasitas maksimal mesin itu biasanya tetap sesuai spesifikasi pabrik, jadi cocok pakai const. Tapi kalau jumlah produksi hari ini, itu bisa berubah-ubah tergantung target atau kondisi produksi, jadi lebih cocok pakai let. Intinya, kalau datanya tetap pakai const, kalau bisa berubah pakai let.

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.

Menurut saya, menggunakan === itu lebih aman daripada == karena === mengecek nilai dan tipe datanya sekaligus. Sedangkan == cuma melihat nilainya saja dan bisa otomatis mengubah tipe data tanpa kita sadar. Itu yang kadang bikin hasilnya mengecoh. Misalnya angka 5 dibandingkan dengan string "5", kalau pakai == hasilnya bisa true karena JavaScript mengubah teks jadi angka dulu. Tapi kalau pakai ===, hasilnya false karena satu tipe number dan satu lagi string. Jadi menurut saya lebih baik pakai === supaya lebih aman dan tidak terjadi kesalahan yang tidak kita sadari.

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.

Dari seluruh materi suplemen, konsep yang paling sulit saya pahami adalah percabangan if...else if...else, terutama saat ada banyak kondisi dan batas angka. Yang bikin sulit itu karena harus benar-benar teliti melihat tanda seperti <, <=, atau >=, dan urutannya juga harus tepat. Kalau salah sedikit saja, hasilnya bisa beda dan program masuk ke kondisi yang tidak seharusnya. Kadang saya juga bingung kenapa kondisi yang di bawah tidak jalan, ternyata karena kondisi di atasnya sudah lebih dulu terpenuhi.

Untuk mengatasinya, saya coba hitung dan analisis dulu secara manual sebelum menjalankan kodenya. Saya juga sering mengganti-ganti angka untuk melihat bagaimana output-nya berubah. Dengan sering mencoba dan membaca lagi alurnya dari atas ke bawah, saya jadi lebih paham bagaimana cara kerja logika percabangan itu sebenarnya.

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.

```
console.log("=== SKENARIO 1 ===");

let skenario1 = [
  { totalProduksi: 1000, jumlahReject: 5 },
  { totalProduksi: 1000, jumlahReject: 20 }
];

for (let i = 0; i < skenario1.length; i++) {

  let total = skenario1[i].totalProduksi;
  let reject = skenario1[i].jumlahReject;
  let rate = (reject / total) * 100;

  let kategori, tindakan;

  if (rate < 1) {
    kategori = "Excellent";
    tindakan = "Tidak ada tindakan";
  }
  else if (rate >= 1 && rate < 3) {
    kategori = "Acceptable";
    tindakan = "Monitor lebih ketat";
  }
  else if (rate >= 3 && rate <= 5) {
    kategori = "Warning";
    tindakan = "Investigasi proses";
  }
  else {
    kategori = "Critical";
    tindakan = "Hentikan produksi, lakukan RCA";
  }

  console.log("Reject Rate:", rate.toFixed(2) + "%");
  console.log("Kategori:", kategori);
  console.log("Tindakan:", tindakan);
  console.log("-----");
}
```

Pada kode ini saya membuat dua data skenario dalam bentuk array, lalu menggunakan perulangan for untuk mengecek setiap data satu per satu. Di dalam perulangan, saya mengambil nilai total produksi dan jumlah reject, kemudian menghitung persentase reject dengan rumus $(\text{reject} / \text{total}) \times 100$. Setelah mendapatkan nilainya, saya menggunakan if...else if...else untuk menentukan kategori dan tindakan berdasarkan batas persentase yang sudah ditentukan. Saya menyusun kondisinya secara berurutan supaya tidak saling tumpang tindih, karena program membaca dari atas ke bawah dan akan berhenti di kondisi pertama yang terpenuhi. Terakhir, hasil reject rate, kategori, dan tindakan ditampilkan di console untuk setiap skenario.

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?

Yang paling berkesan dari suplemen ini membuat saya adalah bagaimana materi dasarnya terlihat sederhana, tapi ternyata butuh ketelitian yang tinggi. Hal-hal kecil yang ternyata sangat berpengaruh ke hasil akhir. Saya jadi sadar kalau pemrograman itu bukan cuma soal bisa menulis kode, tapi soal berpikir logis, runtut, dan sangat teliti. Sedikit saja kurang hati-hati, hasilnya bisa langsung berbeda dan itu bisa berdampak besar kalau diterapkan di konteks industri.

Sebelum Pertemuan 4, saya akan benar-benar memahami semua materi dengan lebih pelan dan teliti. Saya ingin lebih hati-hati dalam membaca soal, memahami logika, dan mengecek kembali setiap kondisi yang saya tulis. Ke depannya saya harus lebih teliti, lebih hati-hati, dan lebih cermat dalam melihat detail kecil supaya tidak melakukan kesalahan yang sebenarnya bisa dihindari. Saya juga akan lebih banyak latihan supaya terbiasa berpikir sistematis dan tidak terburu-buru saat mengerjakan soal.

Ardita Natalia
23051430022

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