# Algoritma Manual & *Deep Code Tracing*

Fase 1 (Zero-Coding Analytical Track): *Greedy*, DP Tabel, Graf, *Bitwise*, & Pola Rekursi

Novalio Daratha
Bimbingan OSN Informatika — Universitas Bengkulu

2026

# Peta Materi Modul 3

1. Greedy Manual vs. Jebakan Counterexample
2. Dynamic Programming (DP) Tabel Manual
3. Graf & Pohon Manual (MST, Dijkstra, Euler)
4. Operasi Bitwise di Atas Kertas
5. Deep Code Tracing: Deduksi Pola Input Besar
6. Studi Kasus Soal Asli OSN-K (2025–2026)
7. Strategi Hari-H OSN-K

# *Greedy* di OSN-K: Cepat, Tapi Rawan Jebakan!

## Prinsip Algoritma *Greedy*

Memilih keputusan yang tampak **paling untung saat ini** di setiap langkah tanpa melihat ulang keputusan sebelumnya.

- **Kapan Benar?** Jika ada *greedy choice property* (contoh: *Activity Selection* urut **waktu selesai tercepat**, MST Kruskal, Dijkstra bobot $\geq 0$).
- **Kapan Gagal?** Jika pilihan lokal mengunci peluang global yang jauh lebih baik!

## Jebakan Klasik: Koin $\{1, 3, 4\}$, Target $= 6$

Cari jumlah koin **minimal** agar jumlahnya tepat 6:

- **Greedy (ambil terbesar):** $4 + 1 + 1 \implies$ 3 koin.
- **Optimal (DP):** $3 + 3 \implies$ 2 koin!

## Aturan Emas Hari-H

Sebelum percaya pada strategi *greedy*, luangkan 20 detik mencari **contoh tandingan (*counterexample*)** kecil! Jika ditemukan, segera pindah ke **Tabel DP**!

# DP Tabel Manual: Grid Koin Maksimum $3 \times 3$

## Soal

Mulai dari $(1,1)$ ke $(3,3)$, hanya gerak **Kanan** atau **Bawah**. Kumpulkan total koin **maksimal**!

**Peta Koin ($a[r][c]$)**

| | | |
|---|---|---|
| 2 | 5 | 1 |
| 4 | 1 | 6 |
| 3 | 7 | 2 |

**Tabel $dp[r][c]$**

| | | |
|---|---|---|
| 2 | 7 | 8 |
| 6 | 8 | 14 |
| 9 | 16 | 18 |

## Transisi DP (Isi Baris per Baris)

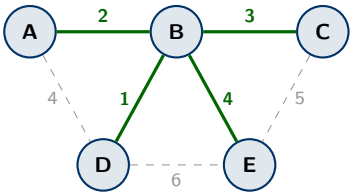
$dp[r][c] = a[r][c] + \max(dp[r-1][c],\ dp[r][c-1])$

- $dp[2][2] = 1 + \max(7, 6) = 8$
- $dp[3][2] = 7 + \max(8, 9) = 16$
- $dp[3][3] = 2 + \max(14, 16) = 18$!

## Bandingkan dengan *Greedy* Lokal!

Dari 2, *greedy* pilih kanan ($5 > 4$)
$\rightarrow 1 \rightarrow 7 \rightarrow 2 = 17 < 18$!

# Graf Manual OSN-K: MST Kruskal & Lintasan Euler




**Garis Hijau =**

**MST** ($V - 1 = 4$ sisi):
$(B, D) : 1,\ (A, B) : 2,\ (B, C) : 3,\ (B, E) : 4 \Rightarrow 10.$
Sisi $(A, D) : 4$ dilewati karena membentuk siklus $A$-$B$-$D$!

## 3 Aturan Cepat Graf di OSN-K

1. **MST (Kruskal):** Urutkan sisi dari terkecil; ambil selama tidak membuat siklus sampai tercapai $V - 1$ sisi.
2. **Dijkstra (Bobot $\geq 0$):** Selalu permanenkan simpul dengan jarak sementara terkecil, lalu relaksasi tetangganya.
3. **Lintasan / Siklus Euler** (lewati tiap sisi tepat 1x):
   - 0 simpul berderajat ganjil $\Longrightarrow$ **Siklus Euler**.
   - 2 simpul berderajat ganjil $\Longrightarrow$ **Lintasan Euler**.
   - $> 2$ simpul ganjil $\Longrightarrow$ **Tidak ada**.

# Operasi *Bitwise* & 3 Idiom Favorit Soal OSN-K

## Operator Dasar (Per Bit Biner)

- `a & b` (**AND**): 1 jika keduanya 1.
- `a | b` (**OR**): 1 jika minimal satu 1.
- `a ^ b` (**XOR**): 1 jika kedua bit **berbeda**.
- `a << k`: geser kiri $\equiv a \times 2^k$.
- `a >> k`: geser kanan $\equiv \lfloor a/2^k \rfloor$.

## Konversi Cepat ke Biner

Hafalkan pangkat 2:
$1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024$.
Contoh: $25 = 16 + 8 + 1 = 11001_2$.

## 3 Trik *Bitwise* yang Wajib Dikenali Saat Tracing

1. **Pembatalan XOR:** `a ^ a = 0` dan `a ^ 0 = a`.
   Meng-XOR semua elemen array langsung menyisakan elemen tunggal yang muncul ganjil kali!
2. **Hapus Bit-1 Terkanan:** `n = n & (n - 1)`.
   - Cek pangkat 2: `n > 0 && (n & (n-1)) == 0`.
   - Loop `while (n>0) { n &= n-1; cnt++; }` menghitung **jumlah bit 1 (*popcount*)**!
3. **Nilai Bit-1 Terkanan (LSB):** `n & (-n)`.
   Contoh: $12 = 1100_2 \implies 12\,\&\,(-12) = 4\ (0100_2)$.

# *Deep Code Tracing*: Jangan Simulasikan Sampai 2026!

## Protokol 4 Langkah Menghadapi Kode Input Besar (mis. `f(2026)`)

**(1)** Uji nilai kecil ($n = 1, 2, 3, 4, 5, 6$) $\rightarrow$ **(2)** Buat tabel $(n, f(n))$ $\rightarrow$ **(3)** Kenali operasi matematika tersembunyi (`/2` $\rightarrow$ biner; `%b` $\rightarrow$ FPB) $\rightarrow$ **(4)** Cek rumus pada 1 nilai baru sebelum hitung 2026!

**Contoh 1: Rekursi Pembagian 2**

```
1 int f(int n) {
2     if (n == 0) return 0;
3     return (n % 2) + f(n / 2);
4 }
5 // Berapakah output f(2026)?
```

**Contoh 2: *Russian Peasant* Rekursif**

```
1 int g(int a, int b) {
2     if (b == 0) return 0;
3     if (b % 2 == 1) return a + g(a, b-1);
4     return g(a * 2, b / 2);
5 }
6 // Berapakah output g(37, 25)?
```

## Deduksi Pola

`n % 2` mengambil bit terakhir; `n / 2` menggeser biner ke kanan $\implies$ **jumlah bit 1 (*popcount*)**!
$2026 = 2047 - 21 = 11111101010_2 \implies 8$.

## Deduksi Invarian

Perhatikan hasil kali $a \times b$ tetap sama saat $(2a) \times (b/2)$! Jadi `g(a, b)` menghitung $a \times b$.
`g(37, 25)` $= 37 \times 25 = 925$!

# Soal Asli OSN-K 2025 (No. 26–28): Deduksi Rekursi TIGA(N)

```c
int TIGA(int N) {
  if (N <= 1) return 1;
  else if (N % 3 == 0)
    return TIGA(N - 1) + TIGA(N - 3);
  else if (N % 3 == 1)
    return TIGA(N - 2);
  else return TIGA(N - 3);
}
```

## Tabel Uji Kecil ($N = 1 \ldots 9$)

| $N$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| $T(N)$ | 1 | 1 | **2** | 1 | 1 | **3** | 1 | 1 | **4** |

## Pola Tertutup (Tanpa Tracing Panjang!)

- Jika $N \not\equiv 0 \pmod 3$: TIGA(N) selalu turun ke nilai $\leq 1 \implies 1$!
- Jika $N = 3k$: $T(3k) = 1 + T(3k-3) \implies k+1$!

## 3 Soal Resmi Selesai Sekaligus!

- **No. 26:** Terbesar di $\{2, 6, 14, 18, 25\}$ adalah TIGA(18) $= 6 + 1 = 7$ (Opsi D).
- **No. 27:** Banyak nilai 1 pada $1 \ldots 2025$: $2025 - \frac{2025}{3} = 1350$.
- **No. 28:** $\sum_{i=1}^{100} T(i) = 67(1) + \sum_{k=1}^{33}(k+1) = 67 + 594 = 661$!

# Soal Asli OSN-K 2026: Lipat Kode (No. 35) & MAI (No. 38)

**OSN-K 2026 No. 35 — Lipat Kode**

```
int LIPAT(int n) {
  if (n == 1) return 1;
  return 2 * LIPAT(n / 2);
}
int KODE(int n) {
  return 2 * (n - LIPAT(n)) + 1;
}
```

## Deduksi

`LIPAT(n)` = pangkat 2 terbesar $\leq n$ (Josephus!).
`KODE(41)` $= 2(41 - 32) + 1 = 19$.

**OSN-K 2026 No. 38 — Fungsi `MAI`**

```
int MAI(int x, int y, int z) {
  if (y == 0) return (x + z) % 67;
  return MAI(MAI(x,y-1,z), y-1, z);
}
// Hitung MAI(13, 666666, 37)!
```

## Deduksi + Fermat Modulo 67

Tiap $y$ naik 1, penambahan $z$ menjadi $2\times$ lipat:
$\texttt{MAI}(x, y, z) \equiv x + 2^y z \pmod{67}$.
Karena 67 prima: $2^{66} \equiv 1 \pmod{67}$ dan $66 \mid 666666$
$\implies (13 + 1 \times 37) \bmod 67 = 50$!

# Checklist Cepat Soal Algoritma & Tracing OSN-K

1. **Soal Optimasi (*Greedy* vs. DP):** Coba patahkan ide *greedy* dengan contoh kecil; jika patah, gambar **tabel DP kecil** (1D/2D) dan isi secara disiplin.
2. **Soal Graf:** Bedakan dengan tegas **MST** (total bobot seluruh pohon minimum) vs. **Shortest Path** (jarak dari $S$ ke $T$ minimum)—keduanya bisa memakai sisi berbeda!
3. **Soal *Bitwise*:** Selalu tulis angka dalam **kolom biner rata kanan** sebelum melakukan `&`, `|`, atau `^` agar tidak salah geser bit.
4. **Soal Rekursi / Loop Input Besar:** Buat tabel $n = 1 \ldots 8$ terlebih dahulu. Cari kaitan dengan **pangkat 2**, **jumlah pembagi**, **FPB**, atau **Fibonacci**.
5. **Hitung Jumlah Pemanggilan Rekursif:** Gambar pohon rekursi kecil ($n = 2, 3, 4$) untuk menemukan pola jumlah simpul (mis. $c(n) = 2T(n) - 1$).

## Target Latihan Modul 3

Selesaikan 12 soal pada **Lembar Kerja Modul 3 (C1–C6)** dan catat pola fungsi rekursi yang baru dikenali di *Error Log*!

# Terima Kasih

Sesi Diskusi & Latihan Lembar Kerja Modul 3

**Novalio Daratha**
Bimbingan OSN Informatika — Universitas Bengkulu