Modul 5: Induksi Matematika & Rekursi

Teknik Pembuktian Formal dan Analisis Algoritma Berulang

Kusuma Web

June 18, 2026

Prinsip Induksi Matematika

Definisi Dasar

Induksi Matematika adalah teknik pembuktian formal yang sangat penting untuk membuktikan kebenaran suatu pernyataan matematika $P(n)$ yang berlaku untuk semua bilangan bulat positif n.

Dua Tahap Utama Pembuktian Induksi:

1. **Basis Induksi:** Buktikan bahwa pernyataan $P(n)$ benar untuk kasus terkecil yang valid (biasanya $n = 1$ atau $n = 0$).
2. **Langkah Induksi:**
   - Asumsikan bahwa pernyataan $P(k)$ benar untuk suatu bilangan bulat $k \geq 1$ (**Hipotesis Induksi**).
   - Buktikan bahwa dengan asumsi tersebut, pernyataan untuk kasus berikutnya, yaitu $P(k+1)$, juga terbukti benar.

Analogi Efek Domino

Jika domino pertama dijatuhkan (Basis Induksi), dan jatuhnya domino ke-k selalu

Langkah Nyata Pembuktian Induksi

Kasus Soal

Buktikan bahwa jumlah n bilangan bulat positif pertama adalah:

$$1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2}$$

Penyelesaian Langkah Demi Langkah:

1. **Basis Induksi ($n = 1$):**
   Sisi kiri: 1. Sisi kanan: $\frac{1(1+1)}{2} = \frac{2}{2} = 1$. (Terbukti Benar).
2. **Langkah Induksi:**
   Asumsikan untuk $n = k$ benar: $1 + 2 + 3 + \cdots + k = \frac{k(k+1)}{2}$.
   Buktikan untuk $n = k + 1$ juga benar:

$$(1 + 2 + \cdots + k) + (k+1) = \frac{(k+1)((k+1)+1)}{2} = \frac{(k+1)(k+2)}{2}$$

   Substitusikan asumsi hipotesis ke persamaan di atas:

Definisi Rekursi & Relasi Rekurensi

Konsep Rekursi

Rekursi adalah konsep komputasi di mana sebuah fungsi memanggil dirinya sendiri dari dalam badan fungsi tersebut dengan ukuran parameter yang semakin mengecil untuk menyelesaikan masalah.

Struktur Fungsi Rekursif

Setiap implementasi rekursi wajib memiliki:

- **Base Case (Kasus Dasar):** Kondisi terminasi agar fungsi berhenti memanggil dirinya sendiri (mencegah *infinite loop* / Stack Overflow).
- **Recursive Step:** Memanggil fungsi itu sendiri dengan argumen yang didekatkan menuju Base Case.

Relasi Rekurensi

Persamaan matematika yang mendefinisikan barisan secara rekursif. Contoh:

Studi Kasus Algoritma Komputasi

1. Perhitungan Deret Fibonacci

Didefinisikan secara rekurensi: $F(0) = 0, F(1) = 1$, dan $F(n) = F(n-1) + F(n-2)$ untuk $n \geq 2$.

- Rekursif biasa memiliki kompleksitas buruk $\mathcal{O}(2^n)$.
- Dengan teknik pemrograman dinamis (*Memoization*), dapat dioptimasi menjadi $\mathcal{O}(n)$.

2. Divide and Conquer (Bagi dan Taklukkan)

Paradigma algoritma populer seperti **Merge Sort** atau **Quick Sort** memecah masalah besar menjadi sub-masalah rekursif berukuran setengahnya, menyelesaikannya secara paralel, lalu menggabungkannya kembali secara efisien.