

Pembahasan Toko Online Esdea

Deksripsi Singkat

Diberikan T pelanggan dengan masing-masing memiliki N referral. Untuk tiap pelanggan ke- i , cari $1 + 2 + 3 + \dots + N_i$.

Ide 1: Looping

Untuk tiap pelanggan, loop dari 1 sampai N untuk menghitung sum. Gunakan *long* untuk menghindari *overflow*.

Berikut ini adalah potongan program untuk **Ide 1: Looping**

```
public static long sumToN(long N) {  
    long sum = 0;  
    for (long i = 1; i <= N; i++) {  
        sum += i;  
    }  
    return sum;  
}
```

Kompleksitas: $O(TN)$

Ide 2: Rumus Barisan

Gunakan rumus barisan $\frac{N(N+1)}{2}$ untuk menghitung sum.

Berikut ini adalah potongan program untuk **Ide 2: Rumus Barisan**

```
public static long sumToN(long N) {  
    return N * (N + 1) / 2;  
}
```

Kompleksitas: $O(T)$

Ide 3: Rumus Barisan (Optimized)

Menggunakan rumus pada **Ide 2** masih bisa menyebabkan overflow walaupun telah menggunakan *long*. Hal ini dapat terjadi pada perhitungan $N(N + 1)$ untuk nilai N yang besar. Untuk mengatasi kasus tersebut kita dapat membagi 2 terlebih dahulu untuk mengecilkan nilai N (atau $N+1$). Operator pembagian ($/$) akan membulatkan hasil pembagian ke bawah (*flooring*) jika hasil pembagian

tidak bulat. Oleh karena itu, kita perlu melakukan pengecekan ganjil/genap agar tidak terjadi *flooring*.

Berikut ini adalah potongan program untuk **Ide 3: Rumus Barisan (Optimized)**.

```
public static long sumToN(long N) {  
    if (N % 2 == 0)  
        return N / 2 * (N + 1);  
    else  
        return (N + 1) / 2 * N;  
}
```

Kompleksitas: $O(T)$