

Penjelasan Metode Rekursif mystery()

java

```
public static int mystery(int n) {  
    if (n < 10) {  
        return n;  
    } else {  
        int a = n / 10;  
        int b = n % 10;  
        return mystery(a + b);  
    }  
}
```

Analisis Metode `mystery(int n)`

Metode ini bekerja dengan cara:

1. Jika $n < 10$ (angka satu digit), langsung kembalikan nilai n
2. Jika $n \geq 10$ (angka multi-digit):
 - Bagi n dengan 10 untuk mendapatkan a (semua digit kecuali digit terakhir)
 - Ambil sisa pembagian n dengan 10 untuk mendapatkan b (digit terakhir)
 - Panggil kembali `mystery(a + b)` (jumlah dari digit-digit)

Pada dasarnya, metode ini berulang kali menjumlahkan semua digit sampai hasilnya menjadi angka satu digit.

1. `mystery(678)`

Pelacakan rekursi langkah demi langkah:

```
mystery(678)
  n = 678, lebih besar dari 10
  a = 678 / 10 = 67
  b = 678 % 10 = 8
  Return mystery(67 + 8) = mystery(75)
```

```
mystery(75)
  n = 75, lebih besar dari 10
  a = 75 / 10 = 7
  b = 75 % 10 = 5
  Return mystery(7 + 5) = mystery(12)
```

```
mystery(12)
  n = 12, lebih besar dari 10
  a = 12 / 10 = 1
  b = 12 % 10 = 2
  Return mystery(1 + 2) = mystery(3)
```

```
mystery(3)
  n = 3, kurang dari 10
  Return 3
```

Jadi, `mystery(678)` menghasilkan output `3`.

2. `mystery(555)`

Pelacakan rekursi langkah demi langkah:

```
mystery(555)
  n = 555, lebih besar dari 10
  a = 555 / 10 = 55
  b = 555 % 10 = 5
  Return mystery(55 + 5) = mystery(60)
```

```
mystery(60)
  n = 60, lebih besar dari 10
  a = 60 / 10 = 6
  b = 60 % 10 = 0
  Return mystery(6 + 0) = mystery(6)
```

```
mystery(6)
  n = 6, kurang dari 10
  Return 6
```

Jadi, `mystery(555)` menghasilkan output `6`.

Kesimpulan

Metode `mystery()` ini sebenarnya menghitung apa yang dalam matematika dikenal sebagai "digital root" atau "akar digital" - jumlah berulang dari semua digit suatu bilangan hingga mendapatkan satu digit.

- Bilangan 678 memiliki digital root 3 ($6+7+8=21$, $2+1=3$)
- Bilangan