

PRAKTIKUM STRUKTUR DATA
“REKURSIF”



Disusun oleh :

Diana Eka Permata Sari

TRPL 2C

(362458302090)

PROGAM STUDI TEKNOLOGI REKAYASA PERANKAT LUNAK
JURUSAN BISNIS DAN INFORMATIKA

2025

Jalan Raya Jember No.KM13, Kawang, Labanasem, Kec. Kabat, Kabupaten
Banyuwangi, Jawa Timur 68461

Tujuan :

1. Memahami mengenai konsep rekursif
2. Mampu memecahkan permasalahan dengan konsep rekursif

Dasar Teori

Rekursif adalah suatu proses atau prosedur dari fungsi yang memanggil dirinya sendiri secara berulang-ulang. Karena proses dalam Rekursif ini terjadi secara berulang-ulang maka harus ada kondisi yang membatasi pengulangan tersebut, jika tidak maka proses tidak akan pernah berhenti sampai memori yang digunakan untuk menampung proses tersebut tidak dapat menampung lagi/penuh. Kelebihan Fungsi Rekursif adalah program menjadi lebih singkat. Pada beberapa kasus, lebih mudah menggunakan fungsi rekursif, contohnya: pangkat, factorial, dan fibonacci, dan beberapa proses deret lainnya. Fungsi rekursif lebih efisien dan cepat dibandingkan proses secara iteratif. Kekurangan Fungsi Rekursif adalah memakan memori lebih besar, karena setiap bagian dari dirinya dipanggil, akan membutuhkan sejumlah ruang memori untuk penyimpanan. Rekursif sering kali tidak bisa berhenti sehingga memori akan terpakai habis dan program bisa hang. Contoh penerapan Rekursif :

- a. Faktorial dari bilangan bulat positif n didefinisikan sebagai berikut. $n! = n \times (n-1)!$
Untuk $n > 1$ $0! = 1$ Untuk $n = 0$ atau $n = 1$ Secara pemrograman dapat ditulis sebagai:
Faktorial (0) = 1 (1) Faktorial (N) = N * Faktorial (N-1) (2) Persamaan (2) di atas adalah contoh hubungan rekurens (recurrence relation), yang berarti bahwa nilai suatu fungsi dengan argumen tertentu bisa dihitung dari fungsi yang sama dengan argumen yang lebih kecil. Persamaan (1) tidak bersifat rekursif, disebut nilai awal atau basis. Setiap fungsi rekursif paling sedikit mempunyai satu nilai awal, jika tidak fungsi tersebut tidak bisa dihitung secara eksplisit.
- b. Bilangan Fibonacci didefinisikan sebagai berikut 1 1 2 3 5 8 13 21 34 55 89 ...
dari barisan tersebut dapat dilihat bahwa bilangan ke-N ($N > 2$) dalam barisan dapat dicari dari dua bilangan sebelumnya yang terdekat dengan bilangan N, yaitu bilangan ke-(N-1) dan bilangan ke- (N-2), sehingga dapat dirumuskan sebagai : Fibonacci (1) = 1 (1) Fibonacci (2) = 1 (2) Fibonacci (N) = Fibonacci (N-1) + Fibonacci (N-2) (3) Dengan persamaan (1) dan (2) adalah basis dan persamaan (3) adalah rekurensnya. Dalam beberapa situasi, pemecahan secara rekursif maupun secara iteratif mempunyai keuntungan dan kekurangan masing-masing. Cukup sulit untuk menentukan mana yang paling sederhana, paling jelas, paling efisien dan paling mudah dibanding yang lain. Boleh dikatakan pemilihan cara iterative maupun rekursif merupakan kesenangan seorang programmer dan tergantung konteks permasalahan yang akan dipecahkan sesuai dengan kesanggupan yang bersangkutan. Dalam prakteknya, terdapat dua metode rekursif yaitu Tail recursive dan Nontail recursive.

Tugas Pendahuluan

Jawablah pertanyaan berikut !!:

1. Apa yang dimaksud dengan rekursif?

Jawab: Rekursif adalah suatu konsep dalam pemrograman dan matematika di mana suatu fungsi atau proses memanggil dirinya sendiri untuk menyelesaikan masalah. Proses ini biasanya dilakukan dengan cara membagi masalah besar menjadi sub-masalah yang lebih kecil dan lebih sederhana.

2. Tuliskan fungsi untuk menghitung nilai faktorial

```
Jawab: int faktorial(int n) {  
    if (n == 0) {  
        return 1; // Basis kasus  
    } else {  
        return n * faktorial(n - 1); // Kasus rekursif  
    }  
}  
  
void main() {  
    int angka = 5;  
    print('Faktorial dari $angka adalah ${faktorial(angka)}');  
}
```

3. Tuliskan fungsi untuk menampilkan nilai fibonacci dari deret fibonacci

Jawab:

```
int fibonacci(int n) {  
    if (n <= 0) {  
        return 0; // Basis kasus untuk n = 0  
    } else if (n == 1) {  
        return 1; // Basis kasus untuk n = 1  
    } else {  
        return fibonacci(n - 1) + fibonacci(n - 2); // Kasus rekursif  
    }  
}  
  
void main() {  
    int angka = 10; // Misalnya, kita ingin menampilkan Fibonacci hingga angka ke-10  
    print('Deret Fibonacci hingga angka ke-$angka:');  
    for (int i = 0; i <= angka; i++) {  
        print(fibonacci(i));  
    }  
}
```

```
}
```

4. Apa yang dimaksud dengan rekursif tail?

Jawab: Rekursif tail (tail recursion) adalah bentuk rekursi di mana pemanggilan fungsi rekursif terjadi sebagai langkah terakhir dari fungsi tersebut. Dengan kata lain, tidak ada operasi tambahan yang dilakukan setelah pemanggilan fungsi rekursif. Ini memungkinkan beberapa bahasa pemrograman untuk mengoptimalkan penggunaan memori dan menghindari tumpukan panggilan (call stack) yang dalam, sehingga dapat mencegah terjadinya stack overflow.

5. Tuliskan fungsi untuk menghitung deret fibonacci menggunakan tail rekursif!

Jawab:

```
int fibonacciTail(int n, [int a = 0, int b = 1]) {
    if (n == 0) {
        return a; // Basis kasus, mengembalikan nilai Fibonacci ke-0
    } else if (n == 1) {
        return b; // Basis kasus, mengembalikan nilai Fibonacci ke-1
    } else {
        return fibonacciTail(n - 1, b, a + b); // Kasus rekursif tail
    }
}

void main() {
    int angka = 10; // Misalnya, kita ingin menampilkan Fibonacci hingga angka ke-10
    print('Deret Fibonacci hingga angka ke-$angka:');
    for (int i = 0; i <= angka; i++) {
        print(fibonacciTail(i));
    }
}
```

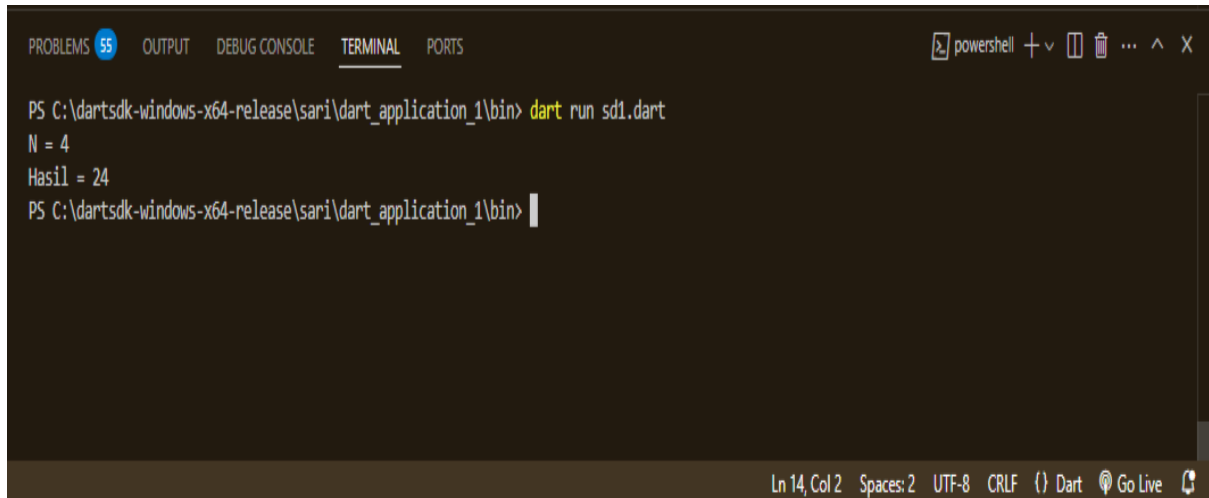
Percobaan

1. Fungsi rekursif untuk menghitung nilai factorial

```
2. import 'dart:io';
3.
4. int faktorial (int x) {
5.     if (x == 1) {
6.         return x;
7.     } else {
8.         return x * faktorial (x - 1);
9.     }
10.}
11.void main() {
12.    stdout.write("N = ");
```

```
13.  int n =int.parse(stdin.readLineSync());
14.  print("Hasil = ${faktorial(n)}");
15. }
```

Hasil run:



```
PS C:\dartsdk-windows-x64-release\sari\dart_application_1\bin> dart run sd1.dart
N = 4
Hasil = 24
PS C:\dartsdk-windows-x64-release\sari\dart_application_1\bin>
```

2. Fungsi rekursif untuk menampilkan deret Fibonacci

```
int fibbon(int x) {
  if (x <= 0 || x <= 1) {
    return x;
  } else {
    return fibbon(x - 2) + fibbon(x - 1);
  }
}

void main() {
  int n = 10;
  for (int i = 0; i < n; i++) {
    print("f$i = ${fibbon(i)}");
  }
}
```

Hasil run :



```
f0 = 0
f1 = 1
f2 = 1
f3 = 2
f4 = 3
f5 = 5
f6 = 8
f7 = 13
f8 = 21
f9 = 34
PS C:\dartsdk-windows-x64-release\sari\dart_application_1\bin>
```

3. Fungsi rekursif untuk menentukan bilangan prima atau bukan prima

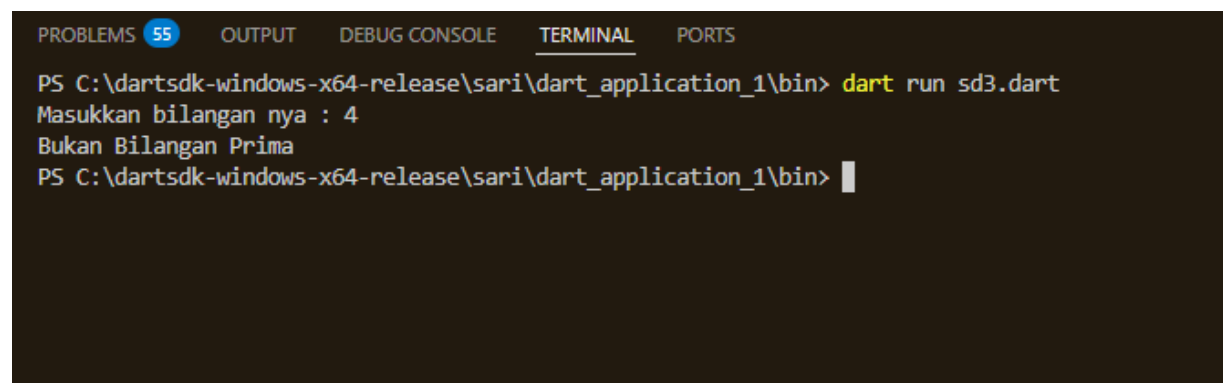
```
import 'dart:io';

int ambilNilaiRekursif(int number, int index) {
  if (index == 1) {
    return 1;
  } else if (number % index == 0) {
    return 1 + ambilNilaiRekursif(number, --index);
  } else {
    return 0 + ambilNilaiRekursif(number, --index);
  }
}

bool cekBilanganPrima(int num) {
  if (num > 1) {
    return (ambilNilaiRekursif(num, num) == 2);
  } else {
    return false;
  }
}

void main() {
  stdout.write("Masukkan bilangan nya : ");
  int num = int.parse(stdin.readLineSync());
  if (cekBilanganPrima(num)) {
    print("Bilangan Prima");
  } else {
    print("Bukan Bilangan Prima");
  }
}
```

Hasil run :



```
PROBLEMS 55 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\dartsdk-windows-x64-release\sari\dart_application_1\bin> dart run sd3.dart
Masukkan bilangan nya : 4
Bukan Bilangan Prima
PS C:\dartsdk-windows-x64-release\sari\dart_application_1\bin> 
```

4. Fungsi rekursi untuk menampilkan kombinasi 2 karakter

```
import 'dart:io';

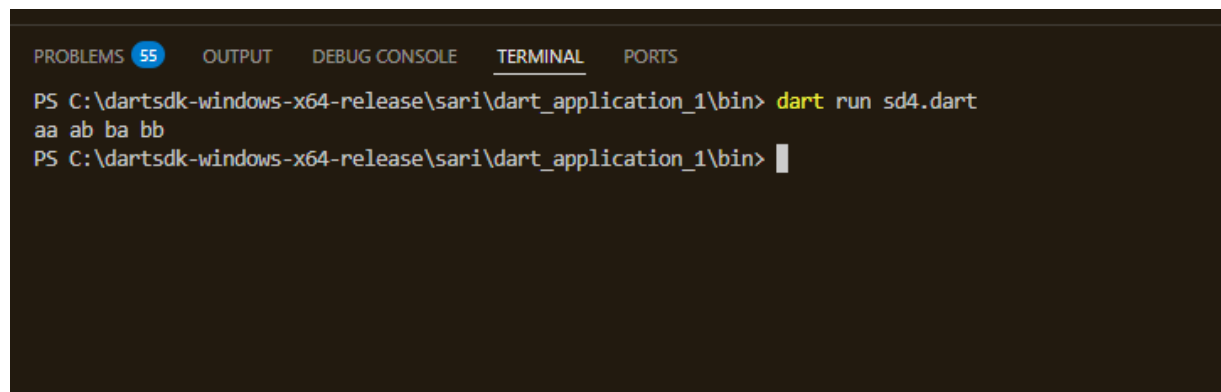
void charCombination(String a, int n) {
  if (n == 0) {
```

```

    stdout.write('$a ');
} else {
    for (int i = 97; i < 99; i++) {
        charCombination(a + String.fromCharCode(i), n - 1);
    }
}
}
void main() {
    charCombination("", 2);
}

```

Hasil run :



```

PROBLEMS 55 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\dartsdk-windows-x64-release\sari\dart_application_1\bin> dart run sd4.dart
aa ab ba bb
PS C:\dartsdk-windows-x64-release\sari\dart_application_1\bin>

```

5. Fungsi rekursi untuk menghitung pangkat

```

import 'dart:io';
int pangkatrekursif(int x, int y) {
    if (y == 0) {
        return 1;
    } else {
        return x * pangkatrekursif(x, y - 1);
    }
}
void main() {
    stdout.write("Bilangan x pangkat y : \n");
    stdout.write("Bilangan x : ");
    int x = int.parse(stdin.readLineSync());
    stdout.write("Bilangan y : ");
    int y = int.parse(stdin.readLineSync());
    print('$x dipangkatkan $y = ${pangkatrekursif(x, y)}');
}

```

Hasil run :

```
PROBLEMS 55 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\dartsdk-windows-x64-release\sari\dart_application_1\bin> dart run sd6.dart
Bilangan x pangkat y :
Bilangan x : 4
Bilangan y : 2
4 dipangkatkan 2 = 16
PS C:\dartsdk-windows-x64-release\sari\dart_application_1\bin> 
```

Fungsi tail rekursif untuk menampilkan i

```
import 'dart:io';

void tail(int i) {
  if (i > 0) {
    stdout.write('$i ');
    tail(i - 1);
  }
}
```

6. Fungsi tail rekursif untuk menghitung faktorial

```
int factAux(int n, int result) {
  if (n == 1) {
    return result;
  }
  return factAux(n - 1, n * result);
}

int fact(int n) {
  return factAux(n, 1);
}

void main() {
  int result = fact(5);
  print('Faktorial: $result');
}
```


Hasil run :

```
b2 C:/q9Lr2qK-MJnQOM2-Xe4-L6J6926/29Lr/q9Lr_9bbJrC9rrou_T/pTn> █  
E9KfOLr9J: JS8  
b2 C:/q9Lr2qK-MJnQOM2-Xe4-L6J6926/29Lr/q9Lr_9bbJrC9rrou_T/pTn> q9Lr Lnu 2qV.q9Lr  
bKOBGEM2 22  OULbUL  DEBNC CONSOLE  TERMINAL  bOKI2
```

7. Fungsi tail rekursif untuk menghitung fibonacci

```
int fibAux(int n, int next, int result) {  
    if (n == 0) {  
        return result;  
    }  
    return fibAux(n - 1, next + result, next);  
}  
int fib(int n) {  
    return fibAux(n, 1, 0);  
}  
void main() {  
    int result = fib(5);  
    print('Deret Fibonacci: $result');  
}
```

Hasil run :

```
PROBLEMS 55  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  
PS C:\dartsdk-windows-x64-release\sari\dart_application_1\bin> dart run sd8.dart  
Deret Fibonacci: 5  
PS C:\dartsdk-windows-x64-release\sari\dart_application_1\bin> █
```

Latihan :

1. Buat program BinarySearch dengan Rekursif ! (data tentukan sendiri): Data :
2,5,8,10,14,32, 35, 41, 67, 88, 90, 101, 109 Data yang dicari : 10 Data 10 berada pada
indek ke – 3
Jawab:

```
void main() {  
    List<int> data = [1, 2, 3, 4, 14, 32, 38, 49, 83, 88, 90, 101, 109];  
    int target = 6;  
  
    int result = binarySearch(data, target, 0, data.length - 1);  
  
    if (result != -1) {  
        print('Data $target berada pada indeks ke - $result');  
    } else {
```

```

    print('Data $target tidak ditemukan');
  }
}

int binarySearch(List<int> arr, int target, int left, int right) {
  if (left > right) {
    return -1; // Target tidak ditemukan
  }

  int mid = left + (right - left) ~/ 2;

  if (arr[mid] == target) {
    return mid; // Target ditemukan
  } else if (arr[mid] > target) {
    return binarySearch(arr, target, left, mid - 1); // Cari di sebelah kiri
  } else {
    return binarySearch(arr, target, mid + 1, right); // Cari di sebelah kanan
  }
}

```

Hasil run :

```

PROBLEMS 55 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\dartsdk-windows-x64-release\sari\dart_application_1\bin>
PS C:\dartsdk-windows-x64-release\sari\dart_application_1\bin> dart run soal3.dart
Data 41 berada pada indeks ke - 7
PS C:\dartsdk-windows-x64-release\sari\dart_application_1\bin> dart run soal3.dart
Data 3 berada pada indeks ke - 2
PS C:\dartsdk-windows-x64-release\sari\dart_application_1\bin> dart run soal3.dart
Data 6 tidak ditemukan
PS C:\dartsdk-windows-x64-release\sari\dart_application_1\bin> 

```

Dikarenakan saya menginput angka 6 yang tidak ada di data jadi hasilnya tidak ditemukan.

Pembahasan : fungsi 'main' itu mendefinisikan array 'data' dan nilai 'target' yang ingin dicari, lalu memanggil fungsi 'binaryserch' dengan parameter yang sesuai. fungsi 'binaryserch' ini menerima array yang dicari ('target') dan batas kiri ('left') serta batas kanan ('right').

2. Buatlah program secara rekursif, masukkan jumlah N karakter dan cetak dalam semua kombinasi ! Jumlah karakter = 3 aaa aab aac aba abb abc aca acb acc baa bab bac bba bbb bbc bca bcb bcc caa cab cac cba cbb cbc cca ccb ccc BUILD SUCCESSFUL (total time: 1 second)

Jawab:

```

void main() {
    int n = 3; // Jumlah karakter
    List<String> characters = ['a', 'b', 'c'];

    // Memanggil fungsi rekursif untuk mencetak kombinasi
    generateCombinations(characters, '', n);
}

void generateCombinations(List<String> characters, String current, int n) {
    // Basis kasus: jika panjang kombinasi saat ini sama dengan n
    if (current.length == n) {
        print(current); // Cetak kombinasi
        return;
    }

    // Rekursi: tambahkan setiap karakter ke kombinasi saat ini
    for (String char in characters) {
        generateCombinations(characters, current + char, n);
    }
}

```

Hasil run :

```

PROBLEMS 55 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\dartsdk-windows-x64-release\sari\dart_application_1\bin> dart run soal2.dart
aaa
aab
aac
aba
abb
abc
aca
acb
acc
baa
bab
bac
bba
bbb
bbc
bca
bcb
bcc
caa
cab
cac
cba
cbb
cbc
cca
ccb
ccc
PS C:\dartsdk-windows-x64-release\sari\dart_application_1\bin> 

```

Pembahasan :

- fungsi 'main' yaitu mendefinisikan jumlah karakter 'n' yang diinginkan (dalam hal ini, 3), lalu fungsi 'main' juga mendefinisikan jumlah karakter yang akan digunakan (a,b,c), memanggil fungsi 'generatecombinations' dengan parameter karakter, string kosong sebagai kombinasi saat ini, dan Panjang 'n'
- fungsi 'generatecombinations' yaitu basis kasus : jika Panjang dari 'current' sama dengan 'n' kita mencetak kombinasi saat ini, rekursi : jika Panjang 'current' belum mencapai 'n' kita melakukan iterasi melalui setiap karakter dalam daftar 'characters' dan memanggil 'generatecombinations' secara rekursif dengan menambahkan karakter saat ini ke 'current'

Kesimpulan : dari praktikum diatas bahwa mengetahui apa itu rekursif dan fungsinya dan juga penyelesaian masalah yang kompleks dengan cara yang terstruktur. Namun, penting untuk memahami kapan dan bagaimana menggunakan rekursi dengan efektif

Referensi : <https://sko.dev/wiki/rekursif> dan <https://www.blackbox.ai/chat/POZXX4P>