

VERSION 1.0  
SEPTEMBER 11, 2021



# ALGORIMA PEMOGRAMAN

RECURSION (QUICK SORT) | MODUL 3

DISUSUN OLEH:

- FARKHAN HAMZAH FIRDAUS
- GILLY HUGA ANARGYA

DIAUDIT OLEH:

- DIDIH RIZKI CHANDRANEGARA, S.KOM., M.KOM
- CHRISTIAN SRI KUSUMA, S.KOM, M.KOM

PRESENTED BY: TIM LAB-IT

UNIVERSITAS MUHAMMADIYAH MALANG

## ALGORIMA PEMOGRAMAN

### PERSIAPAN MATERI

#### 1. Recursion

Rekursif adalah method yang memanggil dirinya sendiri sebanyak satu atau beberapa kali untuk dieksekusi hingga berhenti saat memenuhi argument yang ditentukan.

Tipe-tipe method rekursif:

- Linear : Hanya satu kali memanggil methodnya sendiri dalam fungsi rekursif.
- Non-Linear : Memanggil methodnya sebanyak dua kali atau lebih dalam fungsi rekursif.

```

public static int recur (int n) {
    //base case
    if (n <=1) {
        Return 1;
    }
    else {
        //recursive case
        return 3 * recur (n - 1);
    }
}

```

Hasil eksekusi dari program rekursif diatas adalah seperti pada table dibawah.

n	Call	Results	Actual Value
4	recur(4)	3*recur(3)	$3*(3*(3*1)) = 27$
3	recur(3)	3*recur(2)	$3*(3*1) = 9$
2	recur(2)	3*recur(1)	$3*1 = 3$
1	recur(1)	1	1

#### 2. Quick Sort

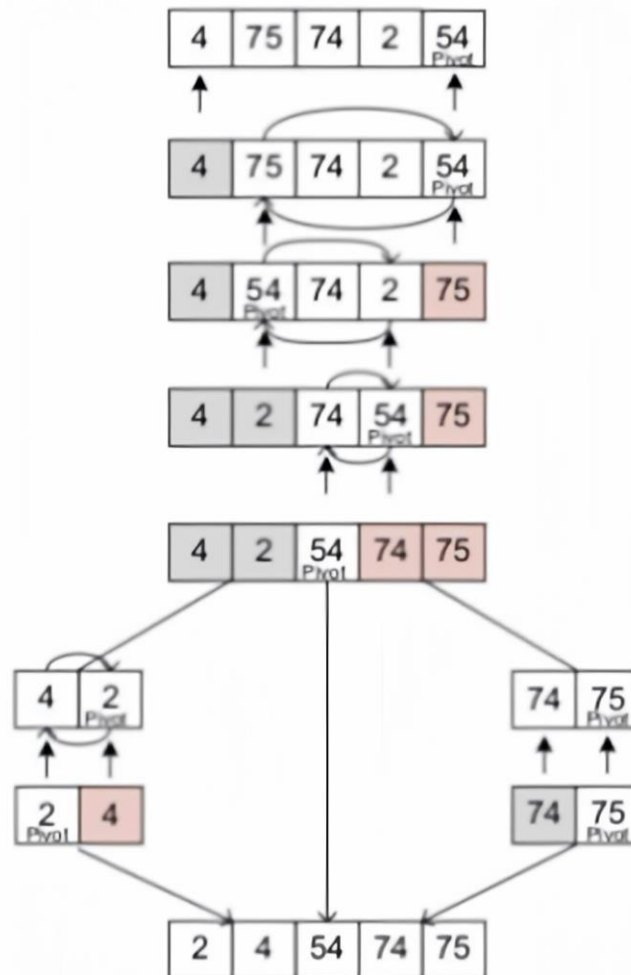
Quick sort merupakan salah satu algoritma pengurutan yang paling populer berdasarkan pendekatan divide-and-conquer, yaitu membagi masalah semula menjadi beberapa bagian submasalah sejenis yang lebih kecil dan menyelesaikannya. Dalam pendekatan ini ada tiga langkah yang lakukan :

1. Divide : Membagi masalah (problem) menjadi beberapa bagian (subproblems).
2. Conquer : Menaklukan (Menyelesaikan) subproblem tadi secara rekursif.
3. Combine : Mengkombinasikan penyelesaian subproblem tadi menjadi satu.

Quick sort melakukan pengurutan dalam perulangan secara rekursif. Pengulangan rekursif dilakukan jika jumlah data yang akan diurutkan lebih dari satu elemen.

Algoritma quick sort dimulai dengan memilih elemen yang biasa disebut dengan pivot, pemilihan elemen yang paling kiri dari array akan sering menjadi pilihan sebagai elemen pivot. Sayangnya, hal ini menyebabkan perilaku worst-case pada array yang telah diurut. Masalah ini dapat diselesaikan dengan memilih salah satu dari elemen acak untuk pivot, memilih elemen tengah dari array atau memilih median dari elemen awal, tengah, dan akhir untuk pivot. dan memindahkan semua elemen yang lebih kecil sebelum pivot, sementara semua elemen yang lebih besar diletakkan setelah pivot.

Berikut gambaran yang lebih jelasnya dengan memilih last elemen sebagai pivot.



Bisa dilihat pada gambar, pivot diletakkan di elemen terakhir, kemudian elemen yang lebih kecil dari pivot akan dipindah ke sebelah kiri pivot, dan elemen yang lebih besar dari pivot dipindah ke sebelah kanan pivot. Kemudian jika sebelah kanan pivot sudah benar elemen yang lebih besar dari pivot dan sebelah kiri pivot adalah elemen yang lebih kecil dari pivot, maka array akan di belah menjadi dua bagian (subarray) untuk dilakukan pivotting lagi. Pivotting tetap diletakkan pada index terakhir (index terakhir dari subarray). Kemudian dilakukan pemindahan elemen hingga sebelah kanan pivot sudah lebih besar dari pivot dan sebelah kiri pivot sudah lebih kecil dari pivot. Langkah ini dilakukan hingga bagian terkecil dari array.

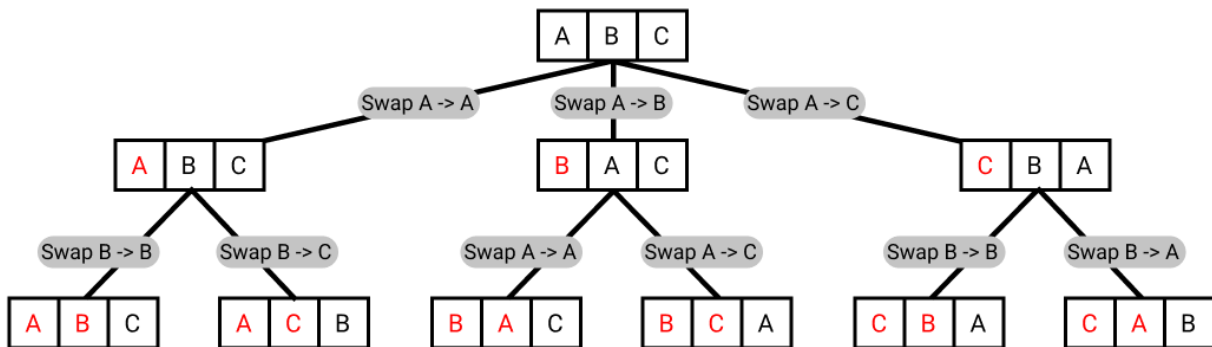
Kita juga dapat membuat visualisasi sendiri proses pengurutan algoritma quick sort pada link berikut : <https://visualgo.net/en/sorting>

### 3. Permutasi

Permutasi adalah penyusunan kembali suatu kumpulan objek dalam urutan yang berbeda dari urutan yang semula.

$${}_nP_r = \frac{n!}{(n-r)!}$$

Jika terdapat suatu objek abjad “abc”, maka objek itu dapat dituliskan kembali dengan urutan yang berbeda: acb, acb, bac, bca, cab, cba. Sehingga ada 6 cara menuliskan ketiga huruf tersebut dalam urutan yang berbeda satu sama lain seperti gambar berikut :



---

## TUJUAN

1. Mahasiswa mampu menjelaskan dan menerapkan algoritma untuk menyelesaikan masalah yang dihadapi.
2. Mahasiswa mampu menganalisa kompleksitas algoritma dalam menyelesaikan masalah.

---

## TARGET MODUL

1. Mahasiswa mampu membuat method rekursif.
2. Mahasiswa mampu membandingkan kelebihan dan kekurangan rekursif.
3. Mahasiswa mampu menjelaskan algoritma quick sort.

---

## PERSIAPAN SOFTWARE/APLIKASI

Eclipse/ Netbeans/IntelliJ IDEA/ dsb (IDE Bahasa Java)

---

## TUGAS

### TUGAS 1

Buatlah program untuk melakukan permutasian **menggunakan rekursi** dan **tanpa rekursi**. Sebagai contoh, jika input string adalah "XYZ" maka output yang dihasilkan adalah XYZ, XZY, YXZ, YZX, ZXY, ZYX. Kemudian analisa perbedaan kompleksitas waktu antara dua program tersebut serta jelaskan kelebihan dan kekurangan menggunakan rekursi.

### TUGAS 2

Buatlah program yang menyimpan array nama hewan sebagai berikut :

["gajah", "kuda", "ikan", "ayam", "bebek"]

Kemudian urutkan menggunakan algoritma quick sort dengan menggunakan pivot bebas. Jelaskan alur dari algoritma quick sort kepada asisten (bukan membaca source code).

---

## DETAIL PENILAIAN TUGAS

1. Menjelaskan konsep **rekursif dan quick sort**. (40 poin)
2. Menyelesaikan tugas praktikum sesuai dengan ketentuan. (30 poin)
3. Menjelaskan alur dan maksud tugas praktikum yang dikerjakan. (30 poin)