




MODUL **ALPRO 2**

 Bubble & Quick Sort
Tim Asisten Pemrograman Angkatan 6

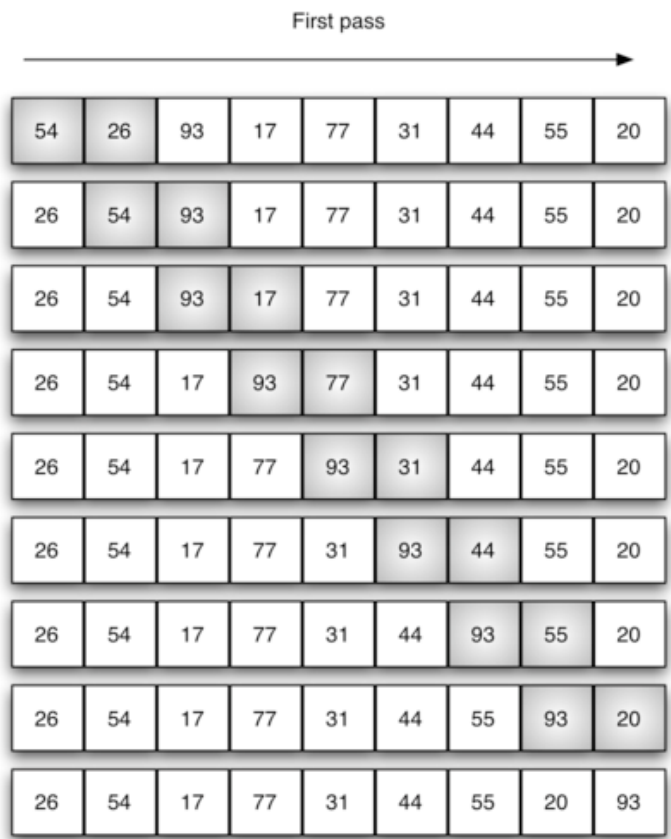
Modul ALPRO 2

Kecepatan Algoritma Sorting

Untuk yang penasaran dengan berbagai algoritma pengurutan dan kecepatannya saat proses, bisa di cek di <http://i.imgur.com/fq0A8hx.gif> (GIF Image).

Bubble Sort

- Membandingkan dua indeks (n & $n+1$)
- Menukarkan nilai antara dua indeks yang dibandingkan jika sesuai syarat pengurutan
- Setiap langkah membutuhkan pemeriksaan sebanyak $n-1$
- Relatif lambat jika data berurutan secara terbalik dari syarat, namun relatif cepat jika data sudah hampir berurutan



Satu langkah penyaringan dalam *Bubble Sort*

Modul ALPRO 2

Contoh Program BS – Pengurutan Naik (Dengan Bungkusan)

- Head.h

```
#include <stdio.h>
#include <string.h>

//Deklarasi bungkusan
typedef struct{
    char hp[12];
    char nama[255];
}kontak;

//Var global banyaknya kontak
int total;
//Deklarasi prosedur bubble
void bubble(kontak daftar[]);
```

- Main.c

```
int main(){
    //Penghitung
    int i;

    //Pemberian nilai [total] oleh user
    printf("Masukkan banyaknya kontak : ");
    scanf("%d", &total);

    //Inisialisasi deretan bungkusan sebanyak [total]
    kontak daftar[total];

    //Pemberian daftar kontak oleh user
    printf("Masukkan daftar kontak : \n");
    for(i = 0; i < total; i++){
        scanf("%s %s", &daftar[i].hp, &daftar[i].nama);
    }

    //Memanggil proses bubble
    bubble(daftar);

    //Mencetak daftar yang sudah diurutkan secara meningkat
    printf("-----Daftar Kontak Anda-----\n");
    for(i = 0; i < total; i++){
        printf("%s %s\n", daftar[i].hp, daftar[i].nama);
    }

    return 0;
}
```

Modul ALPRO 2

- Func.c

```
//Prosedur bubble sort
void bubble(kontak daftar[]){
    int i;          //Variabel counter
    int tukar;      //Status penukaran
    kontak temp;    //Inisialisasi bungkusannya sementara
    do{
        //Inisialisasi status tukar di set false
        tukar = 0;
        //Pengulangan dan pemeriksaan apakah ada pertukaran
        for(i = 0; i < (total - 1); i++){
            //Jika ada nilai yang ditukar
            if(strcmp(daftar[i].hp, daftar[i+1].hp) == 1){
                //Menukar posisi
                temp = daftar[i];
                daftar[i] = daftar[i+1];
                daftar[i+1] = temp;
                tukar = 1;
            }
        }
    }while(tukar == 1); //Mengulang selama ada pertukaran
}
```

Modul ALPRO 2

Quick Sort

- Menggunakan satu indeks sebagai dasar perbandingan (pivot)
- Salah satu algoritma pengurutan tercepat, karena membagi data menjadi sub-array yang lebih kecil
- Menggunakan rekursif

Contoh Program QS Pengurutan Turun (Dengan Bungkusan)

- Head.h

```
#include <stdio.h>
#include <string.h>

//Deklarasi bungkusan
typedef struct{
    int usia;
    char nama[255];
}data;

//Var global banyaknya data
int total;
//Deklarasi prosedur quick
void quick(int, int, data daftar[]);
```

- Main.c

```
int main(){
    //Penghitung
    int i;

    //Pemberian nilai [total] oleh user
    printf("Masukkan banyaknya data : ");
    scanf("%d", &total);

    //Inisialisasi deretan bungkusan sebanyak [total]
    data daftar[total];

    //Pemberian data oleh user
    printf("Masukkan data : \n");
    for(i = 0; i < total; i++){
        scanf("%d %s", &daftar[i].usia, &daftar[i].nama);
    }

    //Memanggil proses quick
    //Bagian kiri dari indeks ke [0]
    //Bagian kanan dari indeks terakhir [total-1]
    //Array atau data yang mau di sort
    quick(0, total-1, daftar);

    //Mencetak daftar yang sudah diurutkan secara menurun
    printf("-----Daftar Data Usia-----\n");
    for(i = 0; i < total; i++){
        printf("%d %s\n", daftar[i].usia, daftar[i].nama);
    }

    return 0;
}
```

Modul ALPRO 2

- Func.c

```
void quick(int l, int r, data daftar[]){
    //Deklarasi counter kiri dan kanan
    int i, j;    //Penghitung kiri (i) dan kanan (j)
    //deklarasi bungkusan pivot untuk perbandingan dan bungkusan sementara
    data temp;  //Bungkusan penyimpan sementara
    data pivot; //Bungkusan untuk pembanding

    //pengisian variabel counter kiri , kanan, dan bungkusan pivot (data yang ditengah)
    i = l; //Penghitung kiri (i) dimulai dari pembatas kiri (l)
    j = r; //Penghitung kanan (j) dimulai dari pembatas kanan (r)
    pivot = daftar[(l + r) / 2];    //Pivot diambil dari indeks tengah

    //Proses quick sort, lakukan paling tidak 1x (Do While)
    do{
        //Proses pemeriksaan nilai dari sebelah kiri
        //Jika usia yang diperiksa lebih besar dari usia yang ada di pivot
        //Maka lewatkan indeks tsb ke kanan(i bertambah)
        while((daftar[i].usia > pivot.usia) && (i <= j)){
            i = i + 1;
        }
        //Proses pemeriksaan nilai dari sebelah kanan
        //Jika usia yang diperiksa lebih kecil dari usia yang ada di pivot
        //Maka lewatkan indeks tsb ke kiri (j bertambah)
        while((daftar[j].usia < pivot.usia) && (i <= j)){
            j = j - 1;
        }

        //Proses pertukaran nilai setelah pemeriksaan selesai
        //Indeks (i) & (j) yang bisa ditukar sesuai syarat
        if(i < j){
            //Tukar isi bungkusan
            temp = daftar[i];
            daftar[i] = daftar[j];
            daftar[j] = temp;
            //Tambahkan penghitung kiri dan kurangi penghitung kanan
            i = i + 1;
            j = j - 1;
        }
    }while(i < j);    //Ulangi selama penghitung kiri lebih kecil dari kanan

    //Rekursif pembagian sub-array sebelah kiri
    if((l < j) && (j < r)){
        quick(l, j, daftar);
    }
    //Rekursif pembagian sub-array sebelah kanan
    if((i < r) && (i > l)){
        quick(i, r, daftar);
    }
}
```

Modul **ALPRO** 2

Referensi

Rosa Ariani Sukanto. *Modul Sorting Algoritma & Pemograman 2*.

interactivepython.org. *The Bubble Sort*.

<http://interactivepython.org/runestone/static/pythonds/SortSearch/TheBubbleSort.html>

Wikipedia. *Quick Sort*. <https://en.wikipedia.org/wiki/Quicksort>

Wikipedia. *Bubble Sort*. https://en.wikipedia.org/wiki/Bubble_sort