

MODUL 13: PENGURUTAN

13.1. Deskripsi Singkat

Proses pengurutan adalah proses menyusun kembali data dengan aturan tertentu. Data dapat disusun kembali baik dengan urutan menaik (ascending) maupun urutan menurun (descending). Pada modul ini akan dibahas tentang pengurutan data pada data bertipe array menggunakan beberapa algoritma dasar.

13.2. Tujuan Praktikum

- 1) Mahasiswa mampu menerapkan pengurutan menggunakan metode insertion sort, selection sort, merge sort, bubble sort pada array menggunakan bahasa C
- 2) Mahasiswa mampu menerapkan pengurutan pada data yang lebih kompleks.

13.3. Material Praktikum

Kegiatan pada modul ini memerlukan material berupa software editor dan compiler (atau IDE) untuk bahasa pemrograman C.

13.4. Kegiatan Praktikum

A. Insertion Sort

Pada insertion sort, proses pengurutan dilakukan seperti mengurutkan kartu pada satu tangan. Untuk menemukan posisi yang banar, maka satu persatu kartu yang ada di tangan harus dibandingkan secara berurutan. Berikut adalah contoh listing program untuk melakukan selection sort. Anda dapat menyimpannya pada file **modul13a.c**

```
#include <stdio.h>
#define MAX 100 //ukuran maksimum array

void fill_data(int data[], int *size){ //mengisi data
    printf("Input ukuran array (max 100): ");
    scanf("%d", size);
    printf("Input data: ");
    for(int i=0;i<*size;i++){
        scanf("%d",&data[i]);
    }
}

void tampil_data(int data[], int size){
    for(int i=0;i<size;i++) printf("%d ",data[i]);
```

```

        printf("\n");
    }

void insertionSort(int arr[], int n)
{
    int i, key, j;
    for (i = 1; i < n; i++) {
        key = arr[i];
        j = i - 1;
        while (j >= 0 && arr[j] > key) {
            arr[j + 1] = arr[j];
            j = j - 1;
        }
        arr[j + 1] = key;
    }
}

void main(){
    int data[MAX];
    int size; //ukuran array yang dipakai
    fill_data(data,&size);
    insertionSort(data,size);
    printf("data setelah diurutkan:\n");
    tampil_data(data,size);
}

```

Berikut adalah tampilan ketika program dijalankan:

```

C:\Users\Panasonic\Documents\codeblocks\insertionsortarray.exe
Input ukuran array (max 100): 10
Input data: 12 14 11 90 45 23 56 78 79 20
data setelah diurutkan:
11 12 14 20 23 45 56 78 79 90

Process returned 10 (0xA)   execution time : 45.122 s
Press any key to continue.

```

B. Selection Sort

Pada selection sort, proses pengurutan dilakukan dengan cara mencari nilai data terkecil atau terbesar pada setiap perulangan dan menempatkannya pada posisi yang sesuai. Selection sort merupakan kombinasi dari searching dan sorting. Berikut adalah program untuk melakukan selection sort. Anda bisa menyimpannya pada file **modul13b.c**.

```
#include <stdio.h>
#define MAX 100 //ukuran maksimum array

void fill_data(int data[], int *size){ //mengisi data
    printf("Input ukuran array (max 100): ");
    scanf("%d", size);
    printf("Input data: ");
    for(int i=0;i<*size;i++){
        scanf("%d",&data[i]);
    }
}

void tampil_data(int data[], int size){
    for(int i=0;i<size;i++) printf("%d ",data[i]);
    printf("\n");
}

void swap(int *a, int *b){
    int temp = *a;
    *a = *b;
    *b = temp;
}

void selection_sort(int data[], int size){
    for(int step = 0; step < size - 1; step++){
        int min_idx = step;
        for(int i = step +1;i<size;i++)
            if(data[i]<data[min_idx])
                min_idx = i;
        swap(&data[min_idx],&data[step]);
    }
}
```

```

    }
}

void main(){
    int data[MAX];
    int size; //ukuran array yang dipakai
    fill_data(data,&size);
    selection_sort(data,size);
    printf("data setelah diurutkan:\n");
    tampil_data(data,size);
}

```

Berikut adalah tampilan ketika program dijalankan:

```

C:\Users\Panasonic\Documents\codeblocks\selectionsortarray.exe
Input ukuran array (max 100): 8
Input data: 1 6 3 9 8 2 5 4
data setelah diurutkan:
1 2 3 4 5 6 8 9

Process returned 10 (0xA)   execution time : 33.434 s
Press any key to continue.

```

C. Merge Sort

Merge sort dirancang untuk memenuhi kebutuhan pengurutan data yang tidak memungkinkan untuk ditampung dalam memori komputer yang terbatas karena ukuran data sangat besar. Merge sort menggunakan prinsip divide and conquer yaitu dengan memecah data menjadi 2 bagian, mengurutkan setiap bagian, kemudian menggabungkan kembali hasil pengurutan kedua bagian (merge). Berikut adalah contoh listing program merge sort pada array. Anda bisa menyimpannya pada file **modul13c.c**.

```

#include <stdio.h>
#define MAX 100 //ukuran maksimum array

void fill_data(int data[], int *size){ //mengisi data
    printf("Input ukuran array (max 100): ");
    scanf("%d", size);
    printf("Input data: ");
    for(int i=0;i<*size;i++){
        scanf("%d",&data[i]);
    }
}

```

```

    }
}

void tampil_data(int data[], int size){
    for(int i=0;i<size;i++) printf("%d ",data[i]);
    printf("\n");
}

void merge(int arr[], int l, int m, int r){
    int i, j, k;
    int n1 = m - l + 1;
    int n2 = r - m;
    int L[n1], R[n2];
    for (i = 0; i < n1; i++)
        L[i] = arr[l + i];
    for (j = 0; j < n2; j++)
        R[j] = arr[m + 1 + j];
    i = 0;
    j = 0;
    k = l;
    while (i < n1 && j < n2){
        if (L[i] <= R[j]) {
            arr[k] = L[i];
            i++;
        }
        else{
            arr[k] = R[j];
            j++;
        }
        k++;
    }
    while (i < n1){
        arr[k] = L[i];
        i++;
        k++;
    }
}

```

```

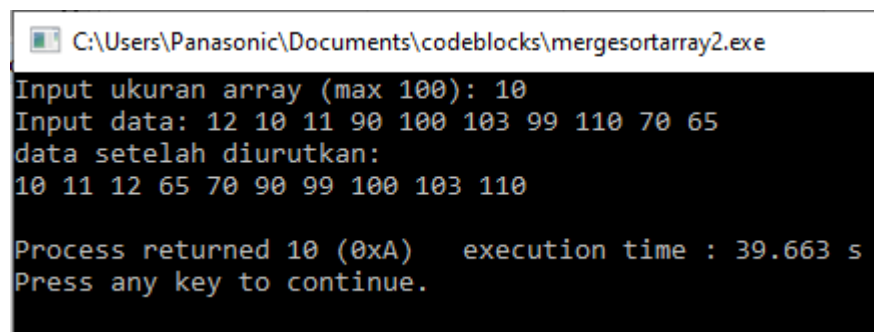
    }
    while (j < n2){
        arr[k] = R[j];
        j++;
        k++;
    }
}

void mergeSort(int arr[], int l, int r){
    if (l < r) {
        int m = l + (r - l) / 2;
        mergeSort(arr, l, m);
        mergeSort(arr, m + 1, r);
        merge(arr, l, m, r);
    }
}

void main(){
    int data[MAX];
    int size; //ukuran array yang dipakai
    fill_data(data,&size);
    mergeSort(data,0,size-1);
    printf("data setelah diurutkan:\n");
    tampil_data(data,size);
}

```

Berikut adalah tampilan ketika program dijalankan:



```

C:\Users\Panasonic\Documents\codeblocks\mergesortarray2.exe
Input ukuran array (max 100): 10
Input data: 12 10 11 90 100 103 99 110 70 65
data setelah diurutkan:
10 11 12 65 70 90 99 100 103 110

Process returned 10 (0xA)   execution time : 39.663 s
Press any key to continue.

```

D. Bubble Sort

Pada bubble sort, proses pengurutan dilakukan dengan cara membandingkan satu data dengan data berikutnya, jika lebih kecil maka akan ditukar urutannya. Berikut adalah fungsi pengurutan dengan bubble sort. Anda bisa menyimpannya pada file **modul13d.c**.

```
#include <stdio.h>
#define MAX 100 //ukuran maksimum array

void fill_data(int data[], int *size){ //mengisi data
    printf("Input ukuran array (max 100): ");
    scanf("%d", size);
    printf("Input data: ");
    for(int i=0;i<*size;i++){
        scanf("%d",&data[i]);
    }
}

void tampil_data(int data[], int size){
    for(int i=0;i<size;i++) printf("%d ",data[i]);
    printf("\n");
}

void swap(int *a, int *b){
    int temp = *a;
    *a = *b;
    *b = temp;
}

void bubbleSort(int arr[], int n){
    int i, j;
    for (i = 0; i < n-1; i++)
        for (j = 0; j < n-i-1; j++)
            if (arr[j] > arr[j+1])
                swap(&arr[j], &arr[j+1]);
}
```

```

void main() {
    int data[MAX];
    int size; //ukuran array yang dipakai
    fill_data(data,&size);
    bubbleSort(data,size);
    printf("data setelah diurutkan:\n");
    tampil_data(data,size);
}

```

Berikut adalah tampilan ketika program dijalankan:

```

C:\Users\Panasonic\Documents\codeblocks\bubblesortarray.exe
Input ukuran array (max 100): 6
Input data: 12 15 1 43 90 77
data setelah diurutkan:
1 12 15 43 77 90

Process returned 10 (0xA)   execution time : 18.448 s
Press any key to continue.

```

13.5. Penugasan

1. Modifikasilah file **modul13a.c**, **modul13b.c**, **modul13c.c**, dan **modul13d.c** supaya dapat juga mengakomodasi pengurutan data menurun (descending).
2. Gabungkan keempat metode sorting tersebut ke dalam satu program. Buatlah sebuah menu sehingga pengguna dapat memilih metode pengurutan yang diinginkan dan juga memilih urutan menaik atau menurun. Simpan dengan nama **Praktikum13A_kelas_nim.c**

Kira-kira tampilan menu sebagai berikut:

```

###PROGRAM SORTING DATA###
Input Jumlah data = 10
Metode Sorting yang tersedia
1. Insertion Sort
2. Selection Sort
3. Merge Sort
4. Bubble Sort
Pilih Metode Sorting (1/2/3/4): 4
Pilih pengurutan Naik/Turun(N/T): T
Input Data Anda: 1 3 2 4 5 7 6 8 10 9
Menjalankan sorting dengan metode Bubble Sort

```


Pilihan pengurutan Turun

Data setelah diurutkan: 10 9 8 7 6 5 4 3 2 1

3. Buat program untuk input, pengurutan, dan pencarian data mahasiswa berisi **int nim**, **char nama[50]**, **int nilai** dengan struktur array. Pengurutan dilakukan berdasarkan nim atau nama, terserah pada pilihan pengguna. Pencarian juga bisa dilakukan berdasarkan nim atau berdasarkan nama. Anda boleh menggunakan algoritma/metode pengurutan dan pencarian yang mana saja. Simpan program dengan **Praktikum13B_kelas_nim.c**

Unggah file **Praktikum13A_kelas_nim.c** dan **Praktikum13B_kelas_nim.c** di Google Classroom sesuai dengan batas waktu yang telah ditetapkan. Keterlambatan pengumpulan dikenakan sanksi pemotongan nilai sebesar 10 poin per jam.