

MAKALAH DESAIN DAN ANALISIS ALGORITMA

"Algoritma Sorting - Selection Sort"

Tugas 2



Di Susun Oleh

Muhammad Firyanul Rizky (1708561006)

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS MIPA

UNIVERSITAS UDAYANA

TAHUN 2018

KATA PENGANTAR

Puji syukur saya panjatkan kehadirat Tuhan Yang Maha Esa yang telah memberikan segala rahmat dan karunia-Nya sehingga dapat menyelesaikan makalah ini untuk memenuhi syarat nilai tugas pada Mata Kuliah Desain dan Analisis Algoritma Prodi Teknik Informatika Universitas Udayana 2018/2019.

Dalam proses penulisan makalah ini, saya banyak menemui kesulitan baik yang disebabkan karena keterbatasan kemampuan, waktu maupun tenaga, karena itu saya sangat menghargai bantuan dari berbagai pihak yang telah memberi sumbangsihnya baik berupa dukungan semangat, buku-buku, serta bermacam-macam bahan penulisan sehingga makalah ini dapat terwujud.

Saya merasa bahwa dalam menyusun makalah ini bukanlah jerih payah sendiri, melainkan berkat bimbingan dari berbagai pihak. Untuk itu pada kesempatan kali ini perkenankanlah saya mengucapkan terima kasih kepada :

1. Tuhan Yang Maha Esa yang telah berkenan memberikan nikmat – Nya berupa kesehatan, Kedua orang tua saya yang telah memberikan dukungan, membimbing serta memberikan kesempatan untuk mengenyam pendidikan tinggi.
2. Dra. Luh Gede Astuti, M.Kom. selaku Dosen pengampu Mata Kuliah Desain dan Analisis Algoritma Prodi Teknik Informatika Universitas Udayana 2018/2019. yang telah memberikan motivasi dan semangat serta memberikan bekal ilmu dan pengetahuan.
3. Teman-teman seperjuangan selectiontan 2017 Prodi Teknik Informatika yang telah memberikan dukungan secara moral.
4. Semua pihak yang telah memberikan banyak bantuan berupa kritik dan saran.

Akhir kata, saya sadar bahwa makalah ini masih jauh dari kata sempurna. Karena itu saya sangat mengharapkan kritik dan saran dari para pembaca yang bersifat membangun makalah ini kedepan agar menjadi lebih baik. Dan saya harap semoga makalah ini bisa bermanfaat bagi kita semua.

Bukit-Jimbaran, 14
September 2018

Penyusun

PEMBAHASAN

1.1 Pengertian Pengurutan (Sorting)

Pengurutan data (sorting) didefinisikan sebagai suatu proses untuk menyusun kembali himpunan obyek menggunakan aturan tertentu. Menurut Microsoft Bookshelf, definisi algoritma pengurutan adalah algoritma untuk meletakkan kumpulan elemen data ke dalam urutan tertentu berdasarkan satu atau beberapa kunci dalam tiap-tiap elemen. Ada dua macam urutan yang biasa digunakan dalam proses pengurutan yaitu :

1. Urut naik (ascending) yaitu dari data yang mempunyai nilai paling kecil sampai paling besar.
2. Urut turun (descending) yaitu data yang mempunyai nilai paling besar sampai paling kecil.

Contoh : data bilangan 5, 2, 6 dan 4 dapat diurutkan naik menjadi 2, 4, 5, 6 atau diurutkan turun menjadi 6, 5, 4, 2. Pada data yang bertipe char, nilai data dikatakan lebih kecil atau lebih besar dari yang lain didasarkan pada urutan relatif (collating sequence).

Keuntungan dari data yang sudah dalam keadaan terurutkan antara lain :

1. Data mudah dicari (misalnya dalam buku telepon atau kamus bahasa), mudah untuk dibetulkan, dihapus, disisipi atau digabungkan. Dalam keadaan terurutkan, kita mudah melakukan pengecekan apakah ada data yang hilang.
2. Melakukan kompilasi program komputer jika tabel-tabel simbol harus dibentuk.
3. mempercepat proses pencarian data yang harus dilakukan berulang kali.

Data yang diurutkan sangat bervariasi, dalam hal jumlah data maupun jenis data yang akan diurutkan. Tidak ada algoritma terbaik untuk setiap situasi yang kita hadapi, bahkan cukup sulit untuk menentukan algoritma mana yang paling baik untuk situasi tertentu karena ada beberapa faktor yang mempengaruhi efektivitas algoritma pengurutan.

Beberapa faktor yang berpengaruh pada efektifitas suatu algoritma pengurutan antara lain:

1. Banyak data yang diurutkan.
2. Kapasitas pengingat apakah mampu menyimpan semua data yang kita miliki.
3. Tempat penyimpanan data, misalnya piringan, pita atau kartu, atau media penyimpan yang lain.

Pemilihan algoritma sangat ditentukan oleh struktur data yang digunakan. Metode pengurutan yang digunakan dapat diklasifikasikan menjadi dua kategori :

1. Pengurutan internal, yaitu pengurutan dengan menggunakan larik (array). Larik tersimpan dalam memori utama komputer
2. Pengurutan eksternal, yaitu pengurutan dengan menggunakan berkas (sequential access file). Berkas tersimpan dalam pengingat luar, misalnya cakram atau pita magnetis.

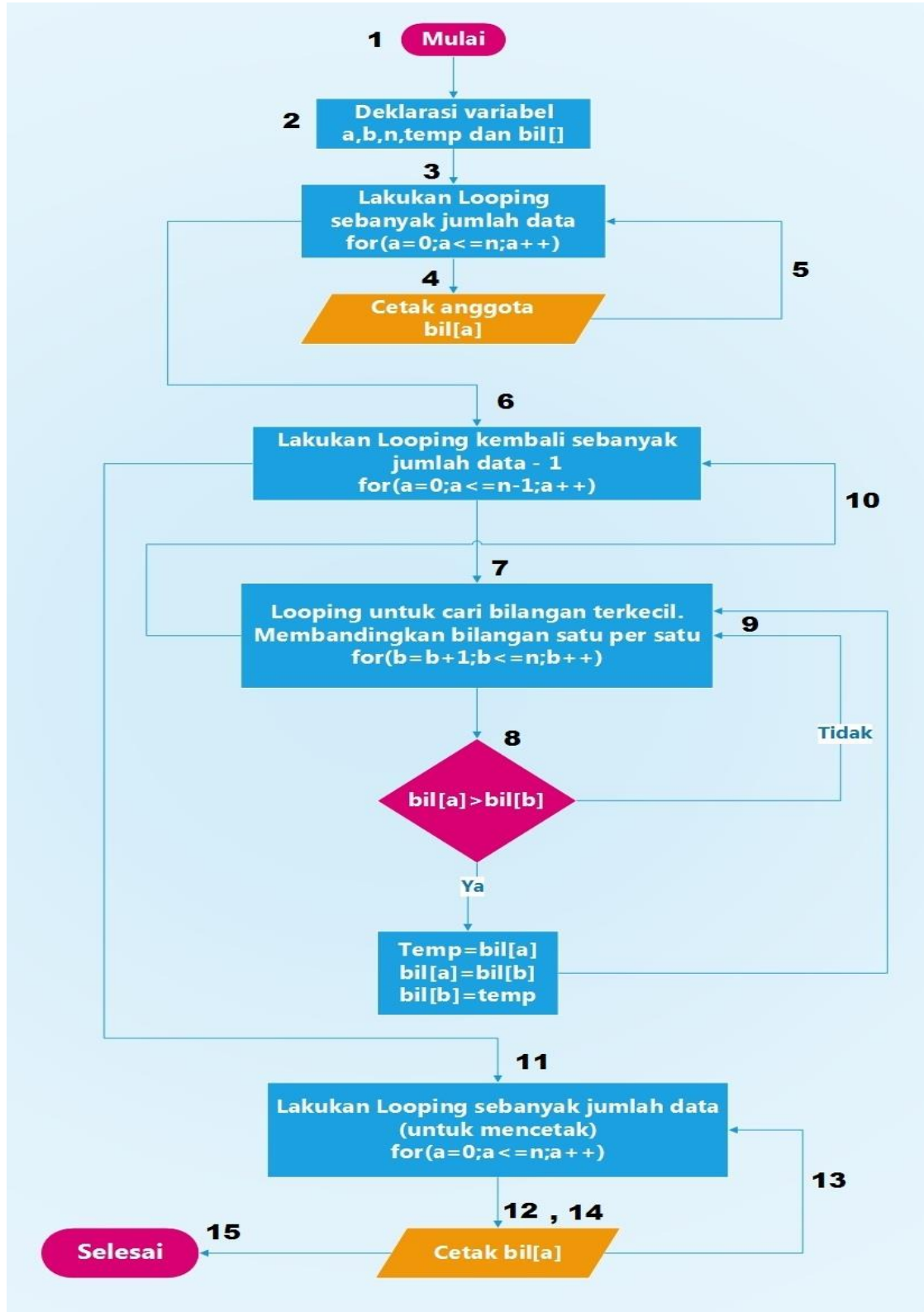
Untuk menggambarkan pengurutan dengan larik, bisa kita bayangkan semua kartu terletak di hadapan kita sehingga semua kartu terlihat dengan jelas nomornya. Pada penyusunan kartu sebagai sebuah berkas, kita bayangkan semua kartu kita tumpuk sehingga hanya kartu bagian atas saja yang bisa kita lihat nomornya.

1.1 Pengertian Selection Sort

Metode selection sort merupakan perbaikan dari metode bubble sort dengan mengurangi jumlah perbandingan. Selection sort merupakan metode pengurutan dengan mencari nilai data terkecil (Taufik Fuadi Abidin, M.Tech.,Ph.D, 2011).

Metode selection sort melakukan pengurutan dengan cara mencari data yang terkecil kemudian menukarkannya dengan data yang digunakan sebagai acuan atau sering dinamakan pivot.

1.2 Flowchart Selection Sort



1.3. Pseudocode Selection Sort ditinjau berdasarkan Flowchart

No	LSELECTIONH	FUNGSI	PSEUDOCODE
1.	Pertama	Permulaan Program	Program Selection Sort Baru dimulai
2.	Kedua	Proses Pengolahan Data	Proses Deklarasi Variabel a = data sebelum terurut b = data setelah terurut n = jumlah data diinput user temp = pointer bil[] = array sebagai pemuatan data
3.	Ketiga	Proses Perhitungan	Akan dilakukan looping sebanyak jumlah data.
4.	Keempat	Output Data	Dari proses 3 (looping) akan dihasilkan output cetak selection dalam array bil[a].
5.	Kelima	Flow Line (Garis Alir)	Proses Cetak selection satu demi satu akan terus berlangsung sampai keseluruhan batas loop habis.
6.	Keenam	Proses Perhitungan	Proses looping kembali dengan jumlah data - 1 (posisi awal sebagai pembandingan) untuk mencari data/posisi selection yang terkecil, caranya adalah membandingkan selection satu per satu dengan posisi selection awal.
7.	Ketujuh	Proses Perhitungan	Sama seperti proses looping lselectionh keenam, hanya saja pada tahap ini cara membandingkannya adalah dengan data selection yang sudah diurutkan dan tersimpan pada array baru bil[b].
8.	Kedelapam	Decision (Keadaan)	Dari proses sebelumnya didapatlah 2 array baru sebagai perbandingan apakah data array sebelum diurutkan bil[a] > data array setelah diurutkan bil[b]. Ada 2 keadaan yang menyatakan ya dan tidak pada tahap ini.
9.	Kesembilan	Flow Line (Garis Alir)	1. Keadaan YA, jika bil[a]>bil[b] maka pointer yang semula menunjuk ke array bil[a] (data selection sebelum terurut) akan berpindah ke array bil[b] (data selection setelah terurut) untuk

			<p>kemudian memproses looping pengurutan array bil[b] seperti lselectionh ketujuh.</p> <p>2. Keadaan TIDAK, jika $bil[a] < bil[b]$ tidak ada pemindahan pointer dan proses looping tetap bekerja pada array bil[a] yang prosesnya sama seperti lselectionh ketujuh</p>
10.	Kesepuluh	Flow Line (Garis Alir)	Dilselectionh ini proses kembali ke lselectionh keenam, menandakan proses pengurutan masih berlangsung sampai menemukan data selection yang terurut menaik.
11.	Kesebelas	Proses Perhitungan	Dalam tahap ini, data selection terurut sudah didapatkan, tetapi proses pengurutan masih berlangsung untuk memastikan semua posisi sudah sesuai urutan yang pas.
12.	Keduabelas	Output Data	Hasil data selection yang terurut pada array bil[b] akan dikembalikan dan disimpan ke array bil[a] yang kemudian akan dicetak.
13.	Ketigabelas	Flow Line (Diagram Alir)	Looping akan dilakukan untuk mencetak keseluruhan hasil data selection terurut pada array bil[a].
14.	Keempatbelas	Output Data	Keseluruhan Data Selection Terurut yang tersimpan pada array bil[a] telah tercetak.
15.	Kelimabelas	Akhir Program	Program Selection Sort Selesai.

1.4 Listing Code Dalam C++

```
#include<iostream>
using namespace std;
#include<conio.h>

int main()
{
    int size, bil[50], a, b, temp;
    cout << "Masukkan Jumlah Selection : ";
    cin >> size;
    cout << "Masukkan Elements Selection : \n";
    for (a = 0; a<size; a++)
    {
        cout << "Data selection ke - "<<a+1<<" :
";cin>>bil[a];
    }
    cout << "Sorting bilangan dengan selection
sort...\n";
    for (a = 0; a<size; a++)
    {
        for (b = a + 1; b<size; b++)
        {
            if (bil[a]>bil[b])
            {
                temp = bil[a];
                bil[a] = bil[b];
                bil[b] = temp;
            }
        }
    }
    cout << "Hasil Sorting Adalah :\n";
    for (a = 0; a<size; a++)
    {
        cout << bil[a] << " ";
    }
    getch();
}
```


1.5 Listing Code Bahasa C++ dengan operasi file, pointer dan linked list

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>

typedef struct sort{
    int selection;
    struct sort *next;
}node;

node * head = NULL;
void list()
{
    printf ("\n1. Menambahkan Data\n");
    printf ("2. Sort Data\n");
    printf ("3. Keluar");
}
node* aktif() {

    node* result = (node *)malloc(sizeof(node));
    result->next = NULL;
    return result;
}
void tambah( int nilai)
{
    node *data1,*data2;
    int a=0,b=0;

    data1 = aktif();
    data1->selection = nilai;

    if(head == NULL)
    {
        head = data1;
    }
    else{
        data2 = head;
        while (data2->next!=NULL)
        {
            data2=data2->next;
        }
        data2->next = data1;
    }
}
```

```

}
void tampil()
{
    node *data2;
    if (head == NULL)
    {
        printf ("\nData Kosong");
    }
    else {
        printf ("Daftar Angka : ");
        for ( data2 = head; data2!=NULL ; data2=data2-
>next)
        {
            printf (" {%d} ",data2->selection);
        }
    }
}

void sorting()
{
    node *h = head, *i, *j;
    for(i = h; i!=NULL && i->next!=NULL; i=i->next)
    {
        node *min;
        min = i;
        for(j = i->next; j!=NULL ; j=j->next)
        {
            if(j->selection < min->selection)
                min=j;
        }
        if(min!=i)
        {
            int temp;
            temp = min->selection;
            min->selection = i->selection;
            i->selection = temp;
        }
    }
    head = h;
}

int main()
{
    int a=0;
    int nilai;
    char filename[20] = {"Input Angka.txt"};

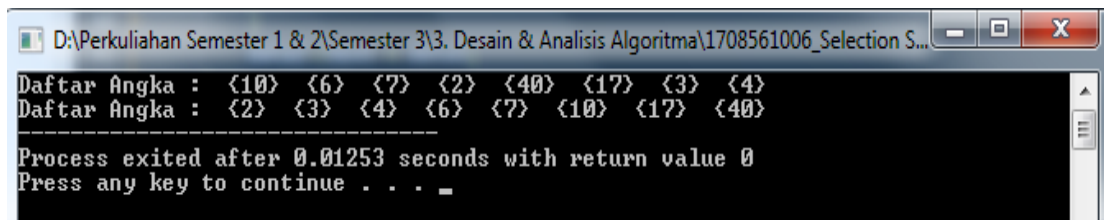
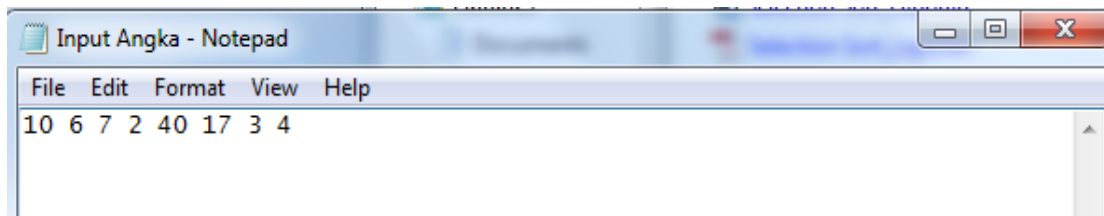
```

```

FILE *cek;
cek=fopen(filename,"r");
while (fscanf(cek,"%d",&nilai)!=EOF)
{
    tambah(nilai);
}
fclose(cek);
tampil();
printf ("\n");
sorting();
tampil();
}

```

Tampilan :



DAFTAR PUSTAKA

1. Setyo, Danang dkk. 2008. Analisis Algoritma Selection Sort. Institut Teknologi Telkom. Bandung.
2. B. Tjaru, Setia Negara 2009," Kompleksitas Algoritma Pengurutan Selection Sort dan Insertion Sort", *Makalah Program Studi Teknik Informatika ITB*. 135-144.
3. Wikipedia, <http://wikipedia.org/selection%sort>. Diakses tanggal 05 September 2018.
4. Liem, Inggriani. 2007. Draft Diktat Kuliah Dasar Pemrograman (Bagian Pemrograman Prosedural). Program Studi Teknik Informatika, Institut Teknologi Bandung. hlm. 141-142.
5. Heriyanto, Gito Rolis. 2014. Pemrograman Dasar untuk SMK Kelas XI. Yuhistira. Bogor.