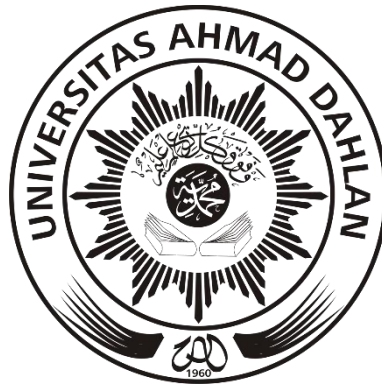


LAPORAN
ALGORITMA PEMORGRAMAN



DISUSUN OLEH
RIFAL FEBIYAN (2100018345)
SLOT SELASA 13.30 – KELAS G

PROGRAM STUDI INFORMATIKA FAKULTAS
TEKNOLOGI INDUSTRI
UNIVERSITAS AHMAD DAHLAN
TAHUN AJARAN 2021/2022

KEGIATAN PRAKTIKUM 7 : SORTING & SEARCHING

Untuk praktikum, kita menggunakan pengurutan menggunakan cara seleksi. Class Sorting yang menerapkan metode pengurutan seleksi adalah sebagai berikut :

Source Code:

```
include <iostream>
using namespace std;

class Sorting{
    friend istream& operator>>(istream& in, Sorting& a);
    friend ostream& operator<<(ostream& out,Sorting& a);
public:
    void selection_sort();
    int pencarianBiner();
private:
    void minimum(int,int,int&);
    void tukar(int*,int*);
    int data[10],n,low,high;
};

void Sorting::minimum(int dari, int n, int &tempat){
    int min = data[dari];
    tempat = dari;
    for(int i = dari+1; i<n; i++){
        if (data[i]<min){
            min = data[i];
            tempat=i;
        }
    }
}

void Sorting::tukar(int *a,int *b){
    int temp;
    temp =*a;
    *a = *b;
    *b = temp;
}

void Sorting::selection_sort(){
    int t;
    for (int i=0; i <n; i++){
        minimum(i,n,t);
        tukar(&data[i],&data[t]);
        cout<<data[i];
    }
}
```

```

}
int Sorting::pencarianBiner(){
int middle, kunciPencarian;
cout<<"\nMasukkan data yang dicari : ";
cin>>kunciPencarian;
low = 0;
high = (n-1);
middle = (low+high)/2;

while(low <= high){
    middle =(low+high)/2;
    // cetakBaris (low,middle,high)
    if(kunciPencarian == data[middle]){
        return middle;
    } else if(kunciPencarian < data[middle]){
        high =middle -1;
    } else {
        low = middle + 1;
    }
    if (low>high){
        cout<<"Data tidak ada";
    }
}
}

istream&operator >> (istream& in,Sorting& a){
    cout<<"Tentukan banyak data : ";
    cin>>a.n;
    for(int i=0; i<a.n; i++){
        cout<<"Masukkan data ke-"<<i<<" : ";
        cin>>a.data[i];
    }
    cout<<"Data setelah diurutkan : ";
    a.selection_sort();
    return in;
}

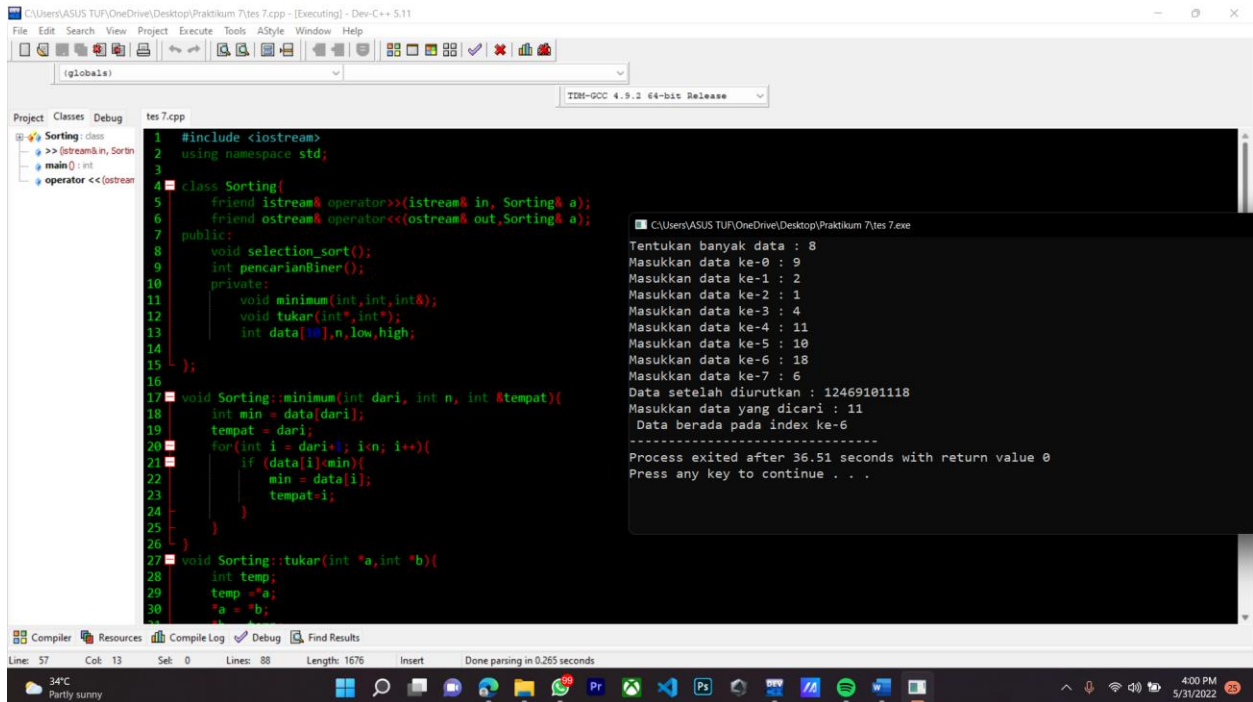
ostream& operator << (ostream& out, Sorting& a){
    out<<" Data berada pada index ke-"<<a.pencarianBiner();
    return out;
}

int main(){
    Sorting run;
    cin>>run;
    cout<<run;
}

```

Setelah dijalankan

Percobaan 1, mengurutkan 9 2 1 4 11 10 18 6 menjadi 1 2 4 6 9 10 11 18



```
#include <iostream>
using namespace std;

class Sorting{
public:
    void selection_sort();
    int pencarianBiner();
private:
    void minimum(int,int,int&);
    void tukar(int*,int*);
    int data[10],n,low,high;
};

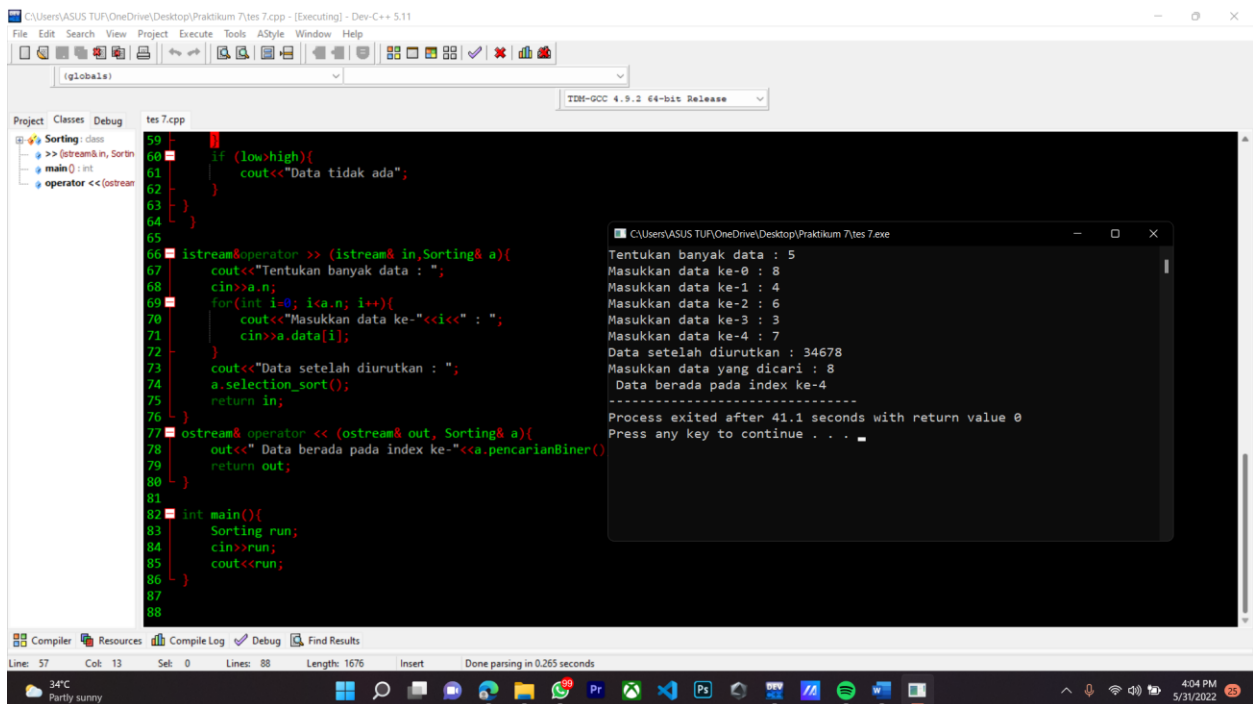
void Sorting::minimum(int dari, int n, int &tempat){
    int min = data[dari];
    tempat = dari;
    for(int i = dari+1; i<n; i++){
        if (data[i]<min){
            min = data[i];
            tempat=i;
        }
    }
}

void Sorting::tukar(int *a,int *b){
    int temp;
    temp =*a;
    *a = *b;
    *b = temp;
}
```

Tentukan banyak data : 8
Masukkan data ke-0 : 9
Masukkan data ke-1 : 2
Masukkan data ke-2 : 1
Masukkan data ke-3 : 4
Masukkan data ke-4 : 11
Masukkan data ke-5 : 10
Masukkan data ke-6 : 18
Masukkan data ke-7 : 6
Data setelah diurutkan : 12469101118
Masukkan data yang dicari : 11
Data berada pada index ke-6

Process exited after 36.51 seconds with return value 0
Press any key to continue . . .

Percobaan 2, mengurutkan 8 4 6 3 7 menjadi 3 4 6 7 8



```
#include <iostream>
using namespace std;

class Sorting{
public:
    void selection_sort();
    int pencarianBiner();
private:
    void minimum(int,int,int&);
    void tukar(int*,int*);
    int data[10],n,low,high;
};

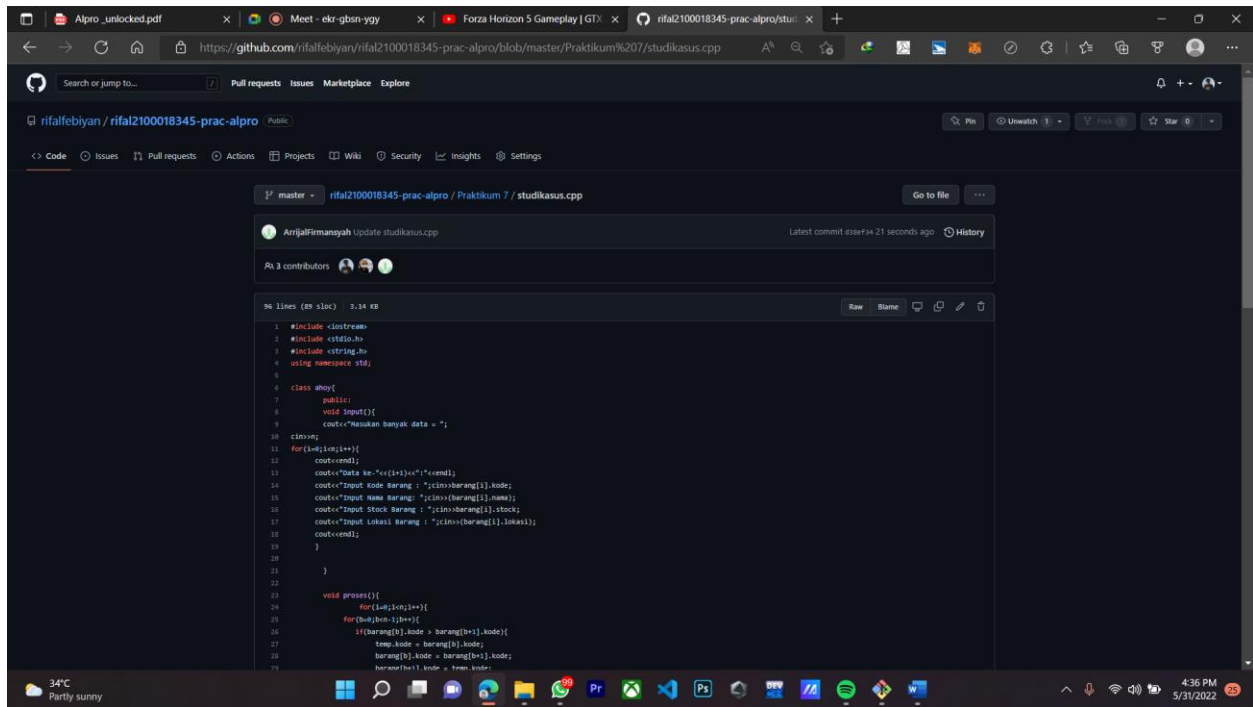
void Sorting::minimum(int dari, int n, int &tempat){
    int min = data[dari];
    tempat = dari;
    for(int i = dari+1; i<n; i++){
        if (data[i]<min){
            min = data[i];
            tempat=i;
        }
    }
}

void Sorting::tukar(int *a,int *b){
    int temp;
    temp =*a;
    *a = *b;
    *b = temp;
}
```

Tentukan banyak data : 5
Masukkan data ke-0 : 8
Masukkan data ke-1 : 4
Masukkan data ke-2 : 6
Masukkan data ke-3 : 3
Masukkan data ke-4 : 7
Data setelah diurutkan : 34678
Masukkan data yang dicari : 8
Data berada pada index ke-4

Process exited after 41.1 seconds with return value 0
Press any key to continue . . .

Screenshot dan Link Repository



The screenshot shows a web browser displaying a GitHub repository page. The repository is named 'rifal2100018345-prac-alpro' and is located at the path 'Praktikum%207/studikasuk.cpp'. The page shows the file 'studikasuk.cpp' with 96 lines of code. The code is written in C++ and includes headers for `<iostream>`, `<string.h>`, and `<string>`. It defines a `barang` struct with fields `kode`, `nama`, `stock`, and `tanggal`. The `main` function prompts the user to enter data for multiple items and then displays the data in a table format.

```
1 #include <iostream>
2 #include <string.h>
3 #include <string>
4 using namespace std;
5
6 struct barang {
7     int kode;
8     string nama;
9     int stock;
10    int tanggal;
11 };
12
13 void input() {
14     cout << "Masukkan banyak data : ";
15     int n;
16     for (int i = 1; i <= n; i++) {
17         cout << "Data ke-" << i << " : ";
18         barang[i].kode = 0;
19         barang[i].nama = "";
20         barang[i].stock = 0;
21         barang[i].tanggal = 0;
22     }
23 }
24
25 void proses() {
26     for (int i = 1; i <= n; i++) {
27         for (int j = 1; j <= n; j++) {
28             if (barang[i].kode > barang[j].kode) {
29                 temp_kode = barang[i].kode;
30                 barang[i].kode = barang[j].kode;
31                 barang[j].kode = temp_kode;
32             }
33         }
34     }
35 }
```

<https://github.com/rifalfebiyan/rifal2100018345-prac-alpro/blob/e4cd9043fdef042d252e5300e598403591fe1315/Praktikum%207/studikasuk.cpp>