

Nama : Dhimas Pramudya Tridharma

Nim : 2109106071

Pratikum : Struktur Data

Tugas : Posttest3

## INSTRUKSI

Tambahkan ketentuan berikut dari program kalian sebelumnya :

- Fitur sorting dengan pointer
- Minimal 1 metode sorting (Metode hanya boleh yang ada dimodul)
- Menu pilihan disorting secara ascending atau descending
- Menu pilihan disorting berdasarkan pilihan kategori (Contoh : Berdasarkan Nama/NIM/IPK)
- Minimal 2 Kategori

## NILAI TAMBAH :

- Menggunakan semua metode sorting yang ada dimodul

## Source Code

```
#include <iostream>
#include <bits/stdc++.h>
using namespace std;

// Deklarasi Struct
struct Lahir{
    string Tempat;
    int tanggal;
    int bulan;
    int tahun;
};

struct Tentara{
    string Nama;
    float Tinggi;
    int ID_Tentara;
    string Divisi;
    Lahir TTL;
};

struct node{
```

```

    Tentara Data_TNI;
    node *next = NULL;
};

// Deklaris Variabel Global

Tentara Arr[50], temp;
int Last_Node, pilih, nambah, up, sudah_sorting;
node *head = NULL, *VarDelete;

bool Kosong(node *head){
    if (head == NULL){
        return true;
    }
    return false;
}

node *new_node(){
    // Membuat Node Baru
    node *nodebaru = new node;
    cout << "Nama\t\t : "; fflush(stdin);getline(cin, nodebaru->Data_TNI.Nama);
    cout << "Tinggi\t\t : "; cin >> nodebaru->Data_TNI.Tinggi;
    cout << "ID Tentara\t : "; cin >> nodebaru->Data_TNI.ID_Tentara;
    cout << "Divisi\t\t : "; fflush(stdin);getline(cin, nodebaru->Data_TNI.Divisi);
    cout << "Tempat Lahir\t : "; getline(cin, nodebaru->Data_TNI.TTL.Tempat);
    cout << "Tanggal Lahir\t : "; cin >> nodebaru->Data_TNI.TTL.tanggal;
    cout << "Bulan Lahir\t : "; cin >> nodebaru->Data_TNI.TTL.bulan;
    cout << "Tahun Lahir\t : "; cin >> nodebaru->Data_TNI.TTL.tahun;
    return nodebaru;
}

void Tampilkan(node *head){
    if (Kosong(head)){
        cout << "[Linked List Masih Kosong . . . ]" << endl;
        return;
    }
    cout << "======" << endl;
    cout << "[DATA TENTARA REPUBLIK INDONESIA]" << endl;
    cout << "======" << endl;
    int i = 0; Last_Node = 0;
    node *temp = head;
    while (temp != NULL){
        cout << "\nData Ke- " << i+1 << "." << endl;
        cout << "Nama\t\t : " << temp->Data_TNI.Nama << endl;
        cout << "Tinggi\t\t : " << temp->Data_TNI.Tinggi << endl;
        cout << "ID_Tentara\t : " << temp->Data_TNI.ID_Tentara << endl;
        cout << "Divisi\t\t : " << temp->Data_TNI.Divisi << endl;
        cout << "TTL\t\t : " << temp->Data_TNI.TTL.Tempat << ", " << temp->Data_TNI.TTL.tanggal << "-" << temp->Data_TNI.TTL.bulan << "-" << temp->Data_TNI.TTL.tahun << endl;
        Last_Node++; i++;
        temp = temp->next;
    }
}

```

```

void Tambah_Data(node **head, int type){
    int posisi;
    if (type == 3){
        cout << "Tambah Data pada No "; cin >> posisi;
    }
    cout << "Banyak Data yang di Input : "; cin >> nambah;
    for (int i = 0; i < nambah; i++){
        cout << "Data Ke- " << i+1 << endl;
        node *nodebaru = new_node();
        if (type == 1){ // ADD FIRST
            if (Kosong(*head)){// Jika Data Kosong
                *head = nodebaru;
            }else{ // Jika sudah ada data
                nodebaru->next = *head;
                *head = nodebaru;
            }
        }else if (type == 2){
            if (Kosong(*head)){
                *head = nodebaru;
            }else{
                node *temp = *head;
                while (temp->next != NULL){
                    temp = temp->next;
                }
                // menghubungkan node di akhir dengan node baru
                temp->next = nodebaru;
            }
        }else if (type == 3){
            if (posisi > Last_Node){
                cout << "ERROR, Posisi Terlali Jauh . . .\n";
                system("pause");
                return;
            }
            node *temp = *head;
            if (posisi == 1){
                if(Kosong(*head)){//Jika data masih kosong
                    *head = nodebaru;
                }else{
                    nodebaru->next = *head;
                    *head = nodebaru;
                }
            }else{
                for (int i = 2; i < posisi; i++){
                    if (temp->next != NULL){
                        temp = temp->next;
                    }
                }
                nodebaru->next = temp->next;
                temp->next = nodebaru;
            }
        }
        }Last_Node++;
        posisi++;
    }cout << "\n[Data Berhasil Di Tambahkan . . .]" << endl;
    system("pause");
}

```

```

void Hapus_Data(node **head, int type){
    if (Kosong(*head)){
        cout << "[Linked List Masih Kosong . . . ]" << endl;
        system("pause");
        return;
    }
    if (type == 1){ // Delete First
        VarDelete = *head;
        *head = (*head)->next;
        delete VarDelete;
    }else if (type == 2){ // Delete Last
        if ((*head)->next == NULL){
            *head = NULL;
            return;
        }
        node *temp = *head;
        while (temp->next->next != NULL){
            temp = temp->next;
        }
        VarDelete = temp->next;
        temp->next = NULL;
        delete VarDelete;
    }else if (type == 3){
        node *temp = *head;
        int posisi;
        cout << "Hapus Data Pada Posisi : "; cin >> posisi;
        if (posisi == 1){
            *head = temp->next;
        }else{
            if (posisi < 1 || posisi > Last_Node){
                cout << "[Data yang anda cari tidak di temukan . . .]" << endl;
                system("pause");
                return;
            }
            node *Hapus = temp;
            for (int i = 1; Hapus != NULL && i < posisi; i++){
                Hapus = Hapus->next;
            }
            for (int i = 1; temp != NULL && i < posisi-1; i++){
                temp = temp->next;
            }
            temp->next = temp->next->next;
            free(Hapus);
        }
    }
    Last_Node--;
    cout << "\n[Data Berhasil Di Hapus . . .]" << endl;
    system("pause");
}

void Update_Data(node **head){
    if(Kosong(*head)){
        cout << "[Linked List Masih Kosong . . . ]" << endl;
        system("pause");
    }
}

```

```

        return;
    }
    cout << "Pilih Data yang ingin di Update : "; cin >> pilih;
    node *temp = *head;
    if (pilih > 0 && pilih <= Last_Node){
        for (int i =1; i < pilih; i++){
            temp = temp->next;
        }
        cout << "Nama\t\t : "; fflush(stdin);getline(cin, temp->Data_TNI>Nama);
        cout << "Tinggi\t\t : "; cin >> temp->Data_TNI.Tinggi;
        cout << "ID Tentara\t : "; cin >> temp->Data_TNI.ID_Tentara;
        cout << "Divisi\t\t : "; fflush(stdin); getline(cin, temp->Data_TNI.Divisi);
        cout << "Tempat Lahir\t : "; getline(cin, temp->Data_TNI.TTL.Tempat);
        cout << "Tanggal Lahir\t : "; cin >> temp->Data_TNI.TTL.tanggal;
        cout << "Bulan Lahir\t : "; cin >> temp->Data_TNI.TTL.bulan;
        cout << "Tahun Lahir\t : "; cin >> temp->Data_TNI.TTL.tahun;
    }
    else{
        cout << "[Data Yang Anda Cari Tidak di Temukan . . .]" << endl;
    }
    cout << "\n[Data Berhasil Di Update . . .]" << endl;
    system("pause");
}

void konversi(Tentara *arr, node *head, int type){
    if (type == 1){
        for (int i = 0; i < Last_Node && head != NULL; i++){
            arr[i].Nama = head->Data_TNI>Nama;
            arr[i].Tinggi = head->Data_TNI.Tinggi;
            arr[i].ID_Tentara = head->Data_TNI.ID_Tentara;
            arr[i].Divisi = head->Data_TNI.Divisi;
            arr[i].TTL.Tempat = head->Data_TNI.TTL.Tempat;
            arr[i].TTL.tanggal = head->Data_TNI.TTL.tanggal;
            arr[i].TTL.bulan = head->Data_TNI.TTL.bulan;
            arr[i].TTL.tahun = head->Data_TNI.TTL.tahun;
            head = head->next;
        }
    }
    else if (type == 2){
        for (int i = 0; i < Last_Node && head != NULL; i++){
            head->Data_TNI>Nama = arr[i].Nama;
            head->Data_TNI.Tinggi = arr[i].Tinggi;
            head->Data_TNI.ID_Tentara = arr[i].ID_Tentara;
            head->Data_TNI.Divisi = arr[i].Divisi;
            head->Data_TNI.TTL.Tempat = arr[i].TTL.Tempat;
            head->Data_TNI.TTL.tanggal = arr[i].TTL.tanggal;
            head->Data_TNI.TTL.bulan = arr[i].TTL.bulan;
            head->Data_TNI.TTL.tahun = arr[i].TTL.tahun;
            head = head->next;
        }
    }
}

void Shellsort(Tentara *arr, int size, int type){
    int urut;

```

```

cout << "======" << endl;
cout << "|      Sorting      |" << endl;
cout << "| [1] Ascending    |" << endl;
cout << "| [2] Descending    |" << endl;
cout << "======" << endl;
cout << "Pilih >> "; cin >> urut;
for (int gap = size/2; gap > 0; gap /= 2) {
    for (int i = 0; i < size; i += 1){
        temp = arr[i];
        int j;
        if (type == 1){
            if (urut == 1){
                sudah_sorting = 1;
                // membandingkan data nama
                for(j=i; j >= gap && arr[j-gap].Nama > temp.Nama; j -= gap)
                // lalu menukarkan data secara keseluruhan
                    arr[j] = arr[j-gap];
            }else if (urut == 2){
                for(j=i; j >= gap && arr[j-gap].Nama < temp.Nama; j -= gap)
                    arr[j] = arr[j-gap];
            }
        }else if (type == 2){
            if (urut == 1){
                for(j=i; j >= gap && arr[j-gap].ID_Tentara > temp.ID_Tentara; j
-= gap)
                    arr[j] = arr[j-gap];
            }else if (urut == 2){
                for(j=i; j >= gap && arr[j-gap].ID_Tentara < temp.ID_Tentara; j
-= gap)
                    arr[j] = arr[j-gap];
            }
        }else if (type == 3){
            if (urut == 1){
                for(j=i; j >= gap && arr[j-gap].Divisi > temp.Divisi; j -= gap)
                    arr[j] = arr[j-gap];
            }else if (urut == 2){
                for(j=i; j >= gap && arr[j-gap].Divisi < temp.Divisi; j -= gap)
                    arr[j] = arr[j-gap];
            }
        }else if (type == 4){
            if (urut == 1){
                sudah_sorting = 1;
                for(j=i; j >= gap && arr[j-gap].Tinggi > temp.Tinggi; j -= gap)
                    arr[j] = arr[j-gap];
            }else if (urut == 2){
                for(j=i; j >= gap && arr[j-gap].Tinggi < temp.Tinggi; j -= gap)
                    arr[j] = arr[j-gap];
            }
        }
        arr[j] = temp;
    }
}cout << "[Data Telah Berhasil di Sorting . . .]" << endl;
system("pause");
}

void sort(){

```

```

int pilih = -1;
while (pilih != 0){
    system("cls");
    cout << "===== " << endl;
    cout << "Pilih Data" << endl;
    cout << "[1] Nama" << endl;
    cout << "[2] ID Tentara" << endl;
    cout << "[3] Divisi" << endl;
    cout << "[4] Tinggi" << endl;
    cout << "[0] Menu Utama" << endl;
    cout << "===== " << endl;
    cout << "Pilih >> "; cin >> pilih;
    switch (pilih){
    case 1 :
        Shellsort(&Arr[0], Last_Node, 1);break;
    case 2 :
        Shellsort(&Arr[0], Last_Node, 2);break;
    case 3 :
        Shellsort(&Arr[0], Last_Node, 3);break;
    case 4 :
        Shellsort(&Arr[0], Last_Node, 4);break;
    default:
        cout << "[Kembali Ke Menu Utama . . .]" << endl;
        system ("pause");
        break;
    }
}
}

int main(){
    int pilih = -1;
    while (pilih != 0){
        system("cls");
        konversi(&Arr[0], head, 1);
        cout << "=====\n";
        cout << "[Menu Pendataan Tentara]" << endl;
        cout << "[1] Lihat Data Tentara" << endl;
        cout << "[2] Tambah Data Tentara di Awal" << endl;
        cout << "[3] Tambah Data Tentara di Akhir" << endl;
        cout << "[4] Tambah Data Tentara Spesifik" << endl;
        cout << "[5] Hapus Data Tentara di Awal" << endl;
        cout << "[6] Hapus Data Tentara di Akhir" << endl;
        cout << "[7] Hapus Data Tentara Spesifik" << endl;
        cout << "[8] Edit Data Tentara" << endl;
        cout << "[9] Sorting Data" << endl;
        cout << "[0] Exit Program" << endl;
        cout << "=====\n";
        cout << " >> ";
        cin >> pilih;
        switch (pilih){
        // fungsi menampilkan ada di setiap pilihan agar menampilkan data yang udah
        // ada sebelum nya
        case 1:
            Tampilkan(head);system("Pause");break;
        case 2:

```

```

        Tampilkan(head);
        Tambah_Data(&head, 1);break;
    case 3:
        Tampilkan(head);
        Tambah_Data(&head, 2);break;
    case 4:
        Tampilkan(head);
        Tambah_Data(&head, 3);break;
    case 5:
        Tampilkan(head);
        Hapus_Data(&head, 1);break;
    case 6:
        Tampilkan(head);
        Hapus_Data(&head, 2);break;
    case 7:
        Tampilkan(head);
        Hapus_Data(&head, 3);break;
    case 8:
        Tampilkan(head);
        Update_Data(&head); break;
    case 9:
        sort();
        konversi(&Arr[0], head, 2);break;
    case 0:cout << "[Terima Kasih Telah Mendaftar semoga beruntung . . .]" <<
endl;break;
    default:
        cout << "Pilihan Anda tidak Tersedia . . ." << endl;
        break;
    }
}
}

```

#### A. OUTPUT MENU

```

=====
[Menu Pendataan Tentara]
[1] Lihat Data Tentara
[2] Tambah Data Tentara di Awal
[3] Tambah Data Tentara di Akhir
[4] Tambah Data Tentara Spesifik
[5] Hapus Data Tentara di Awal
[6] Hapus Data Tentara di Akhir
[7] Hapus Data Tentara Spesifik
[8] Edit Data Tentara
[9] Sorting Data
[0] Exit Program
=====

```



## B. Sorting Data Menu (MENU 9)

### a. Nama

```
Pilih >> 1
=====
|      Sorting      |
| [1] Ascending    |
| [2] Descending    |
|                   |
=====
Pilih >> 1
[Data Telah Berhasil di Sorting . . .]
Press any key to continue . . .
```

#### ASCENDING

```
=====
[DATA TENTARA REPUBLIK INDONESIA]
=====

Data Ke- 1.
Nama       : alex
Tinggi     : 169
ID_Tentara : 2002981
Divisi     : katak
TTL        : bandung, 2-2-2000

Data Ke- 2.
Nama       : dian
Tinggi     : 188
ID_Tentara : 2002110
Divisi     : harimau
TTL        : batam, 12-10-1998

Data Ke- 3.
Nama       : dimas
Tinggi     : 177
ID_Tentara : 2002191
Divisi     : Gagak
TTL        : Grogot, 26-12-1999
Press any key to continue . . .
```

### b. ID Tentara

```
Pilih >> 2
=====
|      Sorting      |
| [1] Ascending    |
| [2] Descending    |
|                   |
=====
Pilih >> 1
[Data Telah Berhasil di Sorting . . .]
Press any key to continue . . .
```

#### ASCENDING

```

=====
[DATA TENTARA REPUBLIK INDONESIA]
=====

Data Ke- 1.
Nama       : dian
Tinggi     : 188
ID_Tentara : 2002110
Divisi     : harimau
TTL        : batam, 12-10-1998

Data Ke- 2.
Nama       : dimas
Tinggi     : 177
ID_Tentara : 2002191
Divisi     : Gagak
TTL        : Grogot, 26-12-1999

Data Ke- 3.
Nama       : alex
Tinggi     : 169
ID_Tentara : 2002981
Divisi     : katak
TTL        : bandung, 2-2-2000
Press any key to continue . . . █

```

c. Devisi

```

Pilih >> 3
=====
|      Sorting      |
| [1] Ascending     |
| [2] Descending    |
|                   |
=====
Pilih >> 2
[Data Telah Berhasil di Sorting . . .]
Press any key to continue . . . █

```

DESCENDING

```

=====
[DATA TENTARA REPUBLIK INDONESIA]
=====

Data Ke- 1.
Nama       : alex
Tinggi     : 169
ID_Tentara : 2002981
Divisi     : katak
TTL        : bandung, 2-2-2000

Data Ke- 2.
Nama       : dian
Tinggi     : 188
ID_Tentara : 2002110
Divisi     : harimau
TTL        : batam, 12-10-1998

Data Ke- 3.
Nama       : dimas
Tinggi     : 177
ID_Tentara : 2002191
Divisi     : Gagak
TTL        : Grogot, 26-12-1999
Press any key to continue . . . █

```

d. Tinggi

```
Pilih >> 4
=====
|      Sorting      |
| [1] Ascending     |
| [2] Descending    |
|                   |
=====
Pilih >> 2
[Data Telah Berhasil di Sorting . . .]
Press any key to continue . . .
```

DESCENDING

```
=====
[DATA TENTARA REPUBLIK INDONESIA]
=====

Data Ke- 1.
Nama       : dian
Tinggi     : 188
ID_Tentara : 2002110
Divisi     : harimau
TTL        : batam, 12-10-1998

Data Ke- 2.
Nama       : dimas
Tinggi     : 177
ID_Tentara : 2002191
Divisi     : Gagak
TTL        : Grogot, 26-12-1999

Data Ke- 3.
Nama       : alex
Tinggi     : 169
ID_Tentara : 2002981
Divisi     : katak
TTL        : bandung, 2-2-2000
Press any key to continue . . .
```

e. Menu Utama

```
=====
|      Pilih Data    |
| [1] Nama           |
| [2] ID Tentara     |
| [3] Divisi         |
| [4] Tinggi         |
| [0] Menu Utama     |
|                   |
=====
Pilih >> 0
[Kembali Ke Menu Utama . . .]
Press any key to continue . . .
```