

Nama : Dhimas Pramudya Tridharma
Nim : 21091060071
Kelas : A'21 Informatika
Tugas : Posttest_5

= POSTTEST 5 =

PERHATIAN !!!

1. Bersifat Individu dan Wajib Dikerjakan bagi setiap peserta praktikum
2. Kerjakan dengan sungguh-sungguh dan semaksimal mungkin
3. Percaya pada diri sendiri , Tidak perlu mencontek Teman kalian
4. Jika terindikasi meniru teman (hanya ganti tema, hanya ganti variabel, hanya ganti tipe data, hanya ganti tampilan), Keduanya mendapat nilai E.
5. Semangat Mengerjakan

INSTRUKSI

Ubahlah program kalian sebelumnya seperti berikut:

- Ubahlah penyimpanan data anda dari Single Linked List menjadi Double Linked List
- Bisa melakukan (CREATE, UPDATE, DELETE) data di posisi manapun
- Sort data Linked List (Menggunakan metode yang ada dimodul sebelumnya)
- Search data Linked List (Menggunakan metode yang ada dimodul sebelumnya)
- Menampilkan data secara :
 - Foward (Data dari head ke Tail)
 - Backward (Data dari Tail ke Head)

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;
    node *prev;
};

// Deklaris Variabel Global
```

```

Tentara Arr[50], temp;
int Last_Node = 0, pilih, nambah, up, urut;
bool nama_sort, ID_sort;
node *head = NULL, *tail = NULL, *VarDelete;

bool Kosong(node *head, node *tail, int type){
    if (type == 1){
        if (head == NULL){
            return true;
        }
        return false;
    }else if (type == 2){
        if (tail == NULL){
            return true;
        }
        return false;
    }else if (type == 3){
        if (head == NULL && tail == 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, node *tail, int type){
    cout << "===== " << endl;
    cout << "[DATA TENTARA REPUBLIK INDONESIA]" << endl;
    cout << "===== " << endl;
    if (type == 1){
        if (Kosong(head, tail, 1)){
            cout << "[Linked List Masih Kosong . . .]" << endl;
            return;
        }
        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;
        }
    }else if (type == 2){
        if (Kosong(head, tail, 2)){
            cout << "[Linked List Masih Kosong . . .]" << endl;
            return;
        }
        int i = 0; Last_Node = 0;
        node *temp = tail;
        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->prev;
    }
}
}

void Tambah_Data(node **head, node **tail, 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
            nodebaru->prev = NULL;
            nodebaru->next = NULL;
            if (Kosong(*head, *tail, 1)){// Jika Data Kosong
                *head = nodebaru;
                *tail = nodebaru;
            }else{ // Jika sudah ada data
                nodebaru->next = *head;
                (*head)->prev = nodebaru;
                *head = nodebaru;
            }
        }else if (type == 2){
            nodebaru->prev = *tail;
            nodebaru->next = NULL;
            if (Kosong(*head, *tail, 3)){
                *head = nodebaru;
                *tail = nodebaru;
            }else{
                (*tail)->next = nodebaru;
                *tail = nodebaru;
            }
        }else if (type == 3){
            if (posisi > Last_Node){
                cout << "ERROR, Posisi Terlalu Jauh . . .\n";
                system("pause");
                return;
            }
            node *temp = *head;
            if (posisi == 1){
                if(Kosong(*head, *tail, 1)){//Jika data masih kosong
                    *head = nodebaru;
                    *tail = nodebaru;
                }else{
                    nodebaru->prev = NULL;
                    nodebaru->next = *head;
                    (*head)->prev= nodebaru;
                    *head = nodebaru;
                }
            }else{
                for (int i = 1; i < posisi - 1; i++){
                    if (temp->next != NULL){
                        temp = temp->next;
                    }
                }
                nodebaru->next = temp->next;
                nodebaru->prev = temp->next->prev;
                temp->next = nodebaru;
                temp->next->prev = nodebaru;
            }
        }
    }
}

```

```

    }
    }Last_Node++;
    posisi++;
}cout << "\n[Data Berhasil Di Tambahkan . . .]" << endl;
system("pause");
}

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

void Update_Data(node **head, node **tail){
    if(Kosong(*head, *tail, 3)){
        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){
    if (type == 1 and urut == 1){
        nama_sort = 1;
        ID_sort = 0;
    }
    else if (type == 2 and urut == 1){
        nama_sort = 0;
        ID_sort = 1;
    }
    else{
        nama_sort = 0;
        ID_sort = 0;
    }
    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){
                    // 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){
        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;
        if (pilih == 0)return;
        system("cls");
        cout << "===== " << endl;
        cout << "|      Sorting         |" << endl;
        cout << "| [1] Ascending       |" << endl;
        cout << "| [2] Descending      |" << endl;
        cout << "===== " << endl;
        cout << "Pilih >> "; cin >> urut;
        switch (pilih){
            case 1 :
                Shellsort(&Arr[0], Last_Node, 1, urut);break;
            case 2 :
                Shellsort(&Arr[0], Last_Node, 2, urut);break;
            case 3 :
                Shellsort(&Arr[0], Last_Node, 3, urut);break;
            case 4 :
                Shellsort(&Arr[0], Last_Node, 4, urut);break;
            default:
                cout << "[Kembali Ke Menu Utama . . .]" << endl;
                system ("pause");
        }
    }
}

```

```

        break;
    }
}

void jumpsearch(node *head,int ind, string Cari_Nama){
    int step = sqrt(ind);
    int prev;
    node *temp = head;
    for (int trv = 0; temp->next != NULL && trv < min(step, ind)-1; trv++){
        temp = temp->next;
    }
    while (temp->Data_TNI>Nama < Cari_Nama){
        prev = step;
        step += sqrt(ind);
        if (prev >= ind) {
            cout << "[Data tidak di temukan . . . ]" << endl;
            system("pause");
            return;
        }
        for(int trv = 0; temp->next != NULL && trv < min (step, ind)-1; trv++){
            temp = temp->next;
        }
    }
    temp = head;
    for (int trv = 0; temp->next != NULL && trv < prev; trv++){
        temp = temp->next;
    }
    while (temp->Data_TNI>Nama < Cari_Nama){
        prev++;
        if (prev == min(step, ind)) {
            cout << "[Data tidak di temukan . . . ]" << endl;
            system("pause");
            return;
        }
        for (int trv = 0; temp->next != NULL && trv < prev; trv++){
            temp = temp->next;
        }
    }
    if (temp->Data_TNI>Nama == Cari_Nama){
        cout << "\nData Ke- " << prev + 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;
        system("pause");
        return;
    }
}

void Fibonacci(node *head, int ind, int ID){
    int F0 = 0;
    int F1 = 1;
    int F = F0 + F1;
    while (F < ind){
        F0 = F1; // F0 = 1 // F0 = 1
        F1 = F; // F1 = 1 // F1 = 2
        F = F0 + F1; // F = 2 // F = 3
    }
    int offset = -1;

    while (F > 1){
        // Inisiasi awal
        node *temp = head;
        // min (-1+1=0 , 3-1=2) yang di ambil 0 iterasi 1 (i = 0)
        // min ( 0+1=1 , 3-1=2) yang di ambil 1 iterasi 2 (i = 1)
        int i = min(offset + F0, ind - 1);
    }
}

```

```

        for (int trv = 0; temp->next != NULL && trv < i; trv++){
            temp = temp->next;
        }
        if (temp->Data_TNI.ID_Tentara < ID){
            F = F1; // F = 2 // F = 1
            F1 = F0; // F1 = 1 // F1 = 1
            F0 = F - F1; // F0 = 1 // F0 = 0
            offset = i; // offset = 0, offset = 1
        }else if(temp->Data_TNI.ID_Tentara > ID){
            F = F0;
            F1 = F1 - F0;
            F0 = F - F1;
        }else{
            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;
            system("pause");
            return;
        }
    }
    node *temp2 = head;
    for (int trv = 0; temp2->next != NULL && trv < offset + 1; trv++){
        temp2 = temp2->next;
    }
    if (F1 && temp2->Data_TNI.ID_Tentara == ID){
        cout << "\nData Ke- " << ind << endl;
        cout << "Nama\t\t : " << temp2->Data_TNI>Nama << endl;
        cout << "Tinggi\t\t : " << temp2->Data_TNI.Tinggi << endl;
        cout << "ID_Tentara\t : " << temp2->Data_TNI.ID_Tentara << endl;
        cout << "Divisi\t\t : " << temp2->Data_TNI.Divisi << endl;
        cout << "TTL\t\t : " << temp2->Data_TNI.TTL.Tempat << ", " << temp2->Data_TNI.TTL.tanggal <<
        "-" << temp2->Data_TNI.TTL.bulan << "-" << temp2->Data_TNI.TTL.tahun << endl;
        system("pause");
        return;
    }else{
        cout << "[Data tidak di temukan . . .]" << endl;
        system("pause");
        return;
    }
}

void search(){
    system("cls");
    cout << "=====\n";
    cout << "Pilih Data yang mau di cari" << endl;
    cout << " [1] Nama" << endl;
    cout << " [2] ID Tentara" << endl;
    cout << "=====\n";
    cout << " >> "; cin >> pilih;
    if (pilih == 1){
        if (nama_sort != 1){
            konversi(&Arr[0], head, 1);
            Shellsort(&Arr[0], Last_Node, 1, 1);
            konversi(&Arr[0], head, 2);
        }
        string Cari_Nama;
        cin.ignore();
        cout << "[Note: Perhatikan Huruf kapital pada Nama yang dicari, inputan harus sesuai pada
        Nama yang ada!]\n" << endl;
        cout << "Masukkan Nama yang ingin dicari : "; getline(cin, Cari_Nama);
        jumpsearch(head, Last_Node, Cari_Nama);
    }else if (pilih == 2){
        if (ID_sort != 1){
            konversi(&Arr[0], head, 1);
            Shellsort(&Arr[0], Last_Node, 2, 1);
        }
    }
}

```



```

        konversi(&Arr[0], head, 2);
    }
    int ID = 0;
    cout << "\n[Note: Perhatikan Detail ID yang Anda Cari]" << endl;
    cout << "Masukkan ID yang ingin dicari : "; cin >> ID;
    Fibonacci(head, Last_Node, ID);
}
}

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 Dari Awal" << endl;
        cout << "[2] Lihat Data Tentara Dari Akhir" << endl;
        cout << "[3] Tambah Data Tentara Awal" << endl;
        cout << "[4] Tambah Data Tentara Akhir" << endl;
        cout << "[5] Tambah Data Tentara Spesifik" << endl;
        cout << "[6] Hapus Data Tentara Awal" << endl;
        cout << "[7] Hapus Data Tentara Akhir" << endl;
        cout << "[8] Hapus Data Tentara Spesifik" << endl;
        cout << "[9] Edit Data Tentara" << endl;
        cout << "[10] Sorting Data" << endl;
        cout << "[11] Searching" << 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, tail, 1);system("Pause");break;
            case 2:
                Tampilkan(head, tail, 2);system("Pause");break;
            case 3:
                Tampilkan(head, tail, 1);
                Tambah_Data(&head, &tail, 1);break;
            case 4:
                Tampilkan(head, tail, 1);
                Tambah_Data(&head, &tail, 2);break;
            case 5:
                Tampilkan(head, tail, 1);
                Tambah_Data(&head, &tail, 3);break;
            case 6:
                Tampilkan(head, tail, 1);
                Hapus_Data(&head, &tail, 1);break;
            case 7:
                Tampilkan(head, tail, 1);
                Hapus_Data(&head, &tail, 2);break;
            case 8:
                Tampilkan(head, tail, 1);
                Hapus_Data(&head, &tail, 3);break;
            case 9:
                Tampilkan(head, tail, 1);
                Update_Data(&head, &tail); break;
            case 10:
                sort();
                konversi(&Arr[0], head, 2);break;
            case 11:
                search();break;
            case 0:cout << "[Terima Kasih Telah Mendaftar semoga beruntung . . .]" << endl;break;
            default:
                cout << "Pilihan Anda tidak Tersedia . . ." << endl;
                system("Pause");
                break;
        }
    }
}

```

```
}  
}
```

- OUTPUT PROGRAM

```
=====
[Menu Pendataan Tentara]
[1] Lihat Data Tentara Dari Awal
[2] Lihat Data Tentara Dari Akhir
[3] Tambah Data Tentara Awal
[4] Tambah Data Tentara Akhir
[5] Tambah Data Tentara Spesifik
[6] Hapus Data Tentara Awal
[7] Hapus Data Tentara Akhir
[8] Hapus Data Tentara Spesifik
[9] Edit Data Tentara
[10] Sorting Data
[11] Searching
[0] Exit Program
=====
>>
```

- SORTING (MENU == 10)

```
=====
|      Pilih Data      |
| [1] Nama             |
| [2] ID Tentara       |
| [3] Divisi           |
| [4] Tinggi           |
| [0] Menu Utama       |
=====
Pilih >> 1
```

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

Maka data akan ditampilkan sebagai berikut dengan menu 1 atau 2

Bila menggunakan menu 1 maka data akan terlihat descending namun bila menu 2 maka akan ascending kenapa karena menu 1 ia menampilkan berdasarkan head (Forward) dan menu 2 menampilkan dari tail (Backward) seperti berikut :

```
>> 1
=====
[DATA TENTARA REPUBLIK INDONESIA]
=====

Data Ke- 1.
Nama      : Pramudya
Tinggi    : 170
ID_Tentara : 170060
Divisi    : katak
TTL       : Longkali, 24-11-2000

Data Ke- 2.
Nama      : Dani
Tinggi    : 177
ID_Tentara : 177090
Divisi    : Merah
TTL       : Bandung, 12-12-1999

Data Ke- 3.
Nama      : Bagas
Tinggi    : 169
ID_Tentara : 169080
Divisi    : baret biru
TTL       : Tanah Grogot, 1-6-1998
Press any key to continue . . .
```

```
>> 2
=====
[DATA TENTARA REPUBLIK INDONESIA]
=====

Data Ke- 1.
Nama      : Bagas
Tinggi    : 169
ID_Tentara : 169080
Divisi    : baret biru
TTL       : Tanah Grogot, 1-6-1998

Data Ke- 2.
Nama      : Dani
Tinggi    : 177
ID_Tentara : 177090
Divisi    : Merah
TTL       : Bandung, 12-12-1999

Data Ke- 3.
Nama      : Pramudya
Tinggi    : 170
ID_Tentara : 170060
Divisi    : katak
TTL       : Longkali, 24-11-2000
Press any key to continue . . .
```

- **SEARCHING (MENU == 11)**

```
=====
Pilih Data yang mau di cari
[1] Nama
[2] ID Tentara
=====

>> 2
[Data Telah Berhasil di Sorting . . .]
Press any key to continue . . .

[Note: Perhatikan Detail ID yang Anda Cari]
Masukkan ID yang ingin dicari : 170060

Data Ke- 2
Nama      : Pramudya
Tinggi    : 170
ID_Tentara : 170060
Divisi    : katak
TTL       : Longkali, 24-11-2000
Press any key to continue . . .
```

- TAMPILKAN DATA Foward (Data dari head ke Tail)

```
>> 1
=====
[DATA TENTARA REPUBLIK INDONESIA]
=====

Data Ke- 1.
Nama      : Pramudya
Tinggi    : 170
ID_Tentara : 170060
Divisi    : katak
TTL       : Longkali, 24-11-2000

Data Ke- 2.
Nama      : Bagas
Tinggi    : 169
ID_Tentara : 169080
Divisi    : baret biru
TTL       : Tanah Grogot, 1-6-1998

Data Ke- 3.
Nama      : Dani
Tinggi    : 177
ID_Tentara : 177090
Divisi    : Merah
TTL       : Bandung, 12-12-1999
Press any key to continue . . .
```

- TAMPILKAN DATA Backward (Data dari Tail ke Head)

```
>> 2
=====
[DATA TENTARA REPUBLIK INDONESIA]
=====

Data Ke- 1.
Nama      : Dani
Tinggi    : 177
ID_Tentara : 177090
Divisi    : Merah
TTL       : Bandung, 12-12-1999

Data Ke- 2.
Nama      : Bagas
Tinggi    : 169
ID_Tentara : 169080
Divisi    : baret biru
TTL       : Tanah Grogot, 1-6-1998

Data Ke- 3.
Nama      : Pramudya
Tinggi    : 170
ID_Tentara : 170060
Divisi    : katak
TTL       : Longkali, 24-11-2000
Press any key to continue . . .
```