Nama : Dhimas Pramudya Tridharma

Nim : 2109106071

Pratikum : Struktur Data

Tugas : Postest4

**= POSTTEST 4 =**

**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**
Tambahkan ketentuan berikut dari program kalian sebelumnya :

- Fitur Searching menggunakan metode ( Salah satu saja ):
  - Fibonacci Search
  - Jump Search

**NILAI TAMBAH :**
Menggunakan semua metode searching yang ada dimodul
**CATATAN !**
Tema tidak boleh sama dengan teman sekelas (Tidak boleh menggunakan tema : Mahasiswa)
Yang ingin ganti tema silahkan, tapi tetap pastikan tema kamu berbeda dengan yang lain

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 = 0, pilih, nambah, up, urut;
bool nama_sort, ID_sort;
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 Terlalu 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 = 1; i < posisi - 1; 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){
   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 Fibonaci(node *head, int ind, int ID){
    int F0 = 0;
    int F1 = 1;
    int F = F0 + F1;
    while (F < ind){
        F0 = F1;
        F1 = F;
        F = F0 + F1;
    }
    int offset = -1;

    while (F > 1){
        // Inisiasi awal
        node *temp = head;
        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;
            F1 = F0;
```

```
            F0 = F - F1;
            offset = i;
        }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 Nama yang ingin dicari : "; cin >> ID;
      Fibonaci(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" << 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 << "[10] 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);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 10:
        search();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
[10] Searching
[0] Exit Program
========================
 >> _
```

B. DATA YANG DITAMBAHKAN

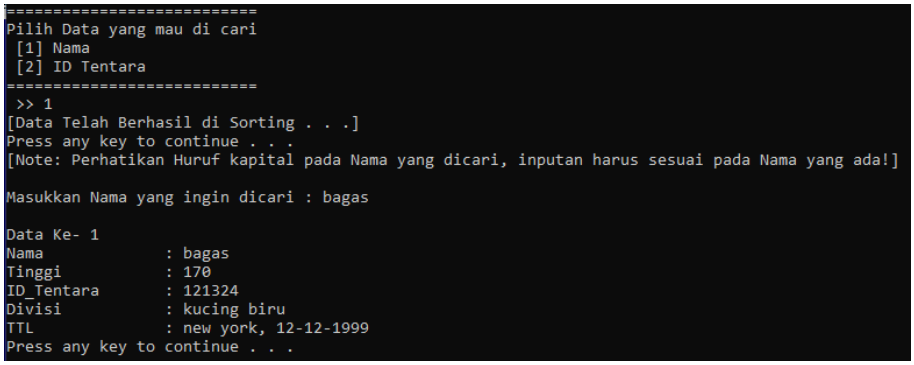
```
 >> 2
[Linked List Masih Kosong . . . ]
Banyak Data yang di Input : 2
Data Ke- 1
Nama              : bagas
Tinggi            : 170
ID Tentara        : 121324
Divisi            : kucing biru
Tempat Lahir      : new york
Tanggal Lahir     : 12
Bulan Lahir       : 12
Tahun Lahir       : 1999
Data Ke- 2
Nama              : dimas
Tinggi            : 172
ID Tentara        : 12345
Divisi            : landak kuning
Tempat Lahir      : sedney
Tanggal Lahir     : 26
Bulan Lahir       : 10
Tahun Lahir       : 2003

[Data Berhasil Di Tambahkan . . .]
Press any key to continue . . .
```

C. SEARCHING (MENU ==10)

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

Pada menu searching sebelum searching akan di sorting sepuya jika data nya lebih dari 1 bisa atau dapat di searcing dan tak terjadi eror.

- NAMA -> JUMPSEARCH




- ID_TENTARA -> FIBONACI