

**STRUKTUR DATA**  
**Tugas semua primitif list**



**NAMA: Bagas Diatama Wardoyo**  
**NPM: 140810230061**

**Dikumpulkan tanggal :**  
**25 April 2024**

**Universitas Padjadjaran**  
**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**  
**Program Studi S-1 Teknik Informatika**  
**2024**

### Source Code:

```
/* Nama program : soal_2
   Nama          : Bagas Diatama Wardoyo
   NPM           : 140810230061
   Tanggal buat  : 01/04/2024
   Deskripsi     : Tugas List Berkait
   *****/

#include <iostream>
#include <string>

// menyimpan informasi dosen (NIP, nama, golongan)
struct dosen
{
    std::string NIP;
    std::string nama;
    int golongan;
};

// struktur data untuk node dalam linked list
struct node
{
    dosen dataDosen;
    node *nextNode;
};

typedef node *pointerToNode; // pointer yang menunjuk ke
                             struktur node
typedef pointerToNode list; // alias untuk pointerToNode,
                             digunakan untuk menunjuk ke head dari linked list

// menginisialisasi head dari linked list menjadi NULL
void createHead(list &head)
{
    head = nullptr;
}
```

```

// membuat elemen baru (node) dan meminta input data dosen
(NIP, nama, golongan), lalu mengembalikan pointer ke elemen
baru tersebut
void createElement(pointerToNode &newNode)
{
    newNode = new node;
    std::cout << "Masukkan NIP      :  ";
    std::cin >> newNode->dataDosen.NIP;
    std::cout << "Masukkan Nama      :  ";
    std::cin.ignore();
    std::getline(std::cin, newNode->dataDosen.nama);
    std::cout << "Masukkan Golongan :  ";
    std::cin >> newNode->dataDosen.golongan;
    newNode->nextNode = nullptr;
}

// menyisipkan elemen baru (newNode) di depan linked list
(head)
void insertFirst(list &head, pointerToNode newNode)
{
    if (head == nullptr)
    {
        head = newNode;
    }
    else
    {
        newNode->nextNode = head;
        head = newNode;
    }
}

// Menyisipkan elemen baru sebelum elemen dengan kunci dalam
linked list (head)
void insertBefore(list &head, pointerToNode newNode,
std::string key)
{
    if (head == nullptr)
    {
        return;
    }
}

```

```

        if (head->dataDosen.NIP == key)
        {
            newNode->nextNode = head;
            head = newNode;
            return;
        }
        pointerToNode pHelp = head;
        while (pHelp->nextNode != nullptr &&
pHelp->nextNode->dataDosen.NIP != key)
        {
            pHelp = pHelp->nextNode;
        }
        if (pHelp->nextNode != nullptr)
        {
            newNode->nextNode = pHelp->nextNode;
            pHelp->nextNode = newNode;
        }
    }

// Menghapus elemen dalam linked list berdasarkan kunci NIP
void deleteAndSearch(list &head, std::string value)
{
    pointerToNode prev = nullptr;
    pointerToNode pHelp = head;
    while (pHelp != nullptr && pHelp->dataDosen.NIP != value)
    {
        prev = pHelp;
        pHelp = pHelp->nextNode;
    }
    if (pHelp == nullptr)
    {
        std::cout << "Elemen dengan nilai " << value << " tidak
ditemukan." << std::endl;
        return;
    }
    if (prev != nullptr)
    {
        prev->nextNode = pHelp->nextNode;
    }
    else

```

```

    {
        head = pHelp->nextNode;
    }
    std::cout << "Elemen dengan nilai " << value << " ditemukan
dan dihapus." << std::endl;
    delete pHelp;
}

// menghapus elemen pertama dari linked list (head) dan
mengembalikan pointer ke elemen selanjutnya
void deleteFirst(list &head)
{
    pointerToNode ptrHelp;
    ptrHelp = head;
    if (head == nullptr)
    {
        return;
    }
    else
    {
        head = ptrHelp->nextNode;
        ptrHelp = nullptr;
        delete ptrHelp;
    }
}

// mengembalikan string gaji berdasarkan golongan (if-else
untuk pengecekan)
std::string gajiGol(int gol)
{
    std::string gaji;
    if (gol == 1)
    {
        gaji = "2.5 juta";
    }
    else if (gol == 2)
    {
        gaji = "3.5 juta";
    }
    else if (gol == 3)

```

```

    {
        gaji = "5 juta";
    }
    else if (gol == 4)
    {
        gaji = "7.5 juta";
    }
    return gaji;
}

// menampilkan seluruh elemen dalam linked list (head) dengan
format tabel
void traversal(list head)
{
    pointerToNode pHelp;
    int no = 1;
    std::string gaji;
    if (head == nullptr)
    {
        return;
    }
    else
    {
        pHelp = head;
        std::cout << "NO\tNIP\tNama\t\t\tGol\tGaji\n";
        std::cout <<
        "-----\n";
        do
        {
            gaji = gajiGol(pHelp->dataDosen.golongan);
            std::cout << no << "\t" << pHelp->dataDosen.NIP <<
            "\t" << pHelp->dataDosen.nama << "\t\t" <<
            pHelp->dataDosen.golongan << "\t" << gaji << "\n";
            no++;
            pHelp = pHelp->nextNode;
        } while (pHelp != nullptr);
        std::cout <<
        "-----\n";
    }
}

```

```

    }
}

int main()
{
    /*Inisialisasi head (pointer ke head linked list) menjadi
NULL
    dan newNode (pointer untuk elemen baru)*/
    std::string key;
    pointerToNode head = nullptr;
    list newNode;

    int pilihan;
    bool menu = 1;

    // Membuat loop while untuk menampilkan menu program
    while (menu)
    {
        std::cout << "\nMenu:" << std::endl;
        std::cout << "1. Create List" << std::endl;
        std::cout << "2. Create Elemen" << std::endl;
        std::cout << "3. Insert First" << std::endl;
        std::cout << "4. Insert Before" << std::endl; //
Menampilkan menu dengan pilihan (Create List, Create Elemen,
Insert First, Delete First, Traversal, Keluar)
        std::cout << "5. Delete First" << std::endl;
        std::cout << "6. Delete and Search" << std::endl;
        std::cout << "7. Traversal" << std::endl;
        std::cout << "0. Keluar" << std::endl;

        std::cout << "Masukkan pilihan: ";
        std::cin >> pilihan;
        // Membaca input pilihan pengguna (pilihan)
        switch (pilihan)
        {
            case 1:
                createHead(head); // membuat head linked list menjadi
NULL (inisialisasi)
                std::cout << "List berhasil dibuat" << std::endl;
                break;

```

```

        case 2:
            createElement(newNode); // membuat elemen baru dosen,
            meminta input data, dan menyimpannya di newNode
            std::cout << "Elemen berhasil dibuat" << std::endl;
            break;
        case 3:
            insertFirst(head, newNode); // menyisipkan elemen baru
            (newNode) di depan linked list (head)
            std::cout << "Elemen berhasil disisipkan di depan" <<
            std::endl;
            break;
        case 4:
            std::cout << "Akan dimasukkan ke sebelum NIP ke: ";
            std::cin >> key;
            insertBefore(head, newNode, key);
            break;
        case 5:
            deleteFirst(head); // menghapus elemen pertama dari
            linked list (head)
            std::cout << "Elemen pertama berhasil dihapus" <<
            std::endl;
            break;
        case 6:
            std::cout << "Masukkan kunci yang akan dicari: ";
            std::cin >> key;
            deleteAndSearch(head, key);
            break;
        case 7:
            traversal(head); // menampilkan seluruh elemen dalam
            linked list (head) dengan format tabel
            break;
        default:
            menu = 0; // Pilihan lain (default): mengakhiri loop
            (menu menjadi 0)
        }
    }
}

```



Hasil Run (Dengan kondisi awal terdapat 3 node):

NO	NIP	Nama		Gol	Gaji
1	002	Diatama	2	3.5	juta
2	006	Wardoyo	2	3.5	juta
3	001	Bagas	2	3.5	juta

Menu:

1. Create List
2. Create Elemen
3. Insert First
4. Insert Before
5. Delete First
6. Delete and Search
7. Traversal
0. Keluar

Masukkan pilihan:

Menu:

1. Create List
2. Create Elemen
3. Insert First
4. Insert Before
5. Delete First
6. Delete and Search
7. Traversal
0. Keluar

Masukkan pilihan: 2

Masukkan NIP : 009

Masukkan Nama : Budi

Masukkan Golongan : 3

Elemen berhasil dibuat

Menu:

1. Create List
2. Create Elemen
3. Insert First
4. Insert Before
5. Delete First
6. Delete and Search
7. Traversal

0. Keluar

Masukkan pilihan: 4

Akan dimasukkan ke sebelum NIP ke: 006

Menu:

1. Create List
2. Create Elemen
3. Insert First
4. Insert Before
5. Delete First
6. Delete and Search
7. Traversal
0. Keluar

Masukkan pilihan: 7

NO	NIP	Nama	Gol	Gaji
1	002	Diatama	2	3.5 juta
2	009	Budi	3	5 juta
3	006	Wardoyo	2	3.5 juta
4	001	Bagas	2	3.5 juta

Menu:

1. Create List
2. Create Elemen
3. Insert First
4. Insert Before
5. Delete First
6. Delete and Search
7. Traversal
0. Keluar

Masukkan pilihan:

Menu:

1. Create List
2. Create Elemen
3. Insert First
4. Insert Before
5. Delete First
6. Delete and Search
7. Traversal
0. Keluar

Masukkan pilihan: 6

Masukkan kunci yang akan dicari: 002

Elemen dengan nilai 002 ditemukan dan dihapus.

Menu:

1. Create List
2. Create Elemen
3. Insert First
4. Insert Before
5. Delete First
6. Delete and Search
7. Traversal
0. Keluar

Masukkan pilihan: 7

NO	NIP	Nama		Gol	Gaji
1	009	Budi	3	5 juta	
2	006	Wardoyo	2	3.5 juta	
3	001	Bagas	2	3.5 juta	

Menu:

1. Create List
2. Create Elemen
3. Insert First
4. Insert Before
5. Delete First
6. Delete and Search
7. Traversal
0. Keluar

Masukkan pilihan: