

LAPORAN PRAKTIKUM STRUKTUR DATA

LAPORAN AWAL PRAKTIKUM

Pertemuan ke-12

Stack Lanjut



Disusun Oleh:

Nama Lengkap : Nova Ardiansyah

NIM : 211011401309

Kelas : 04-TPLE008

TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PAMULANG

Jl. Raya Puspitek No. 11 Buaran, Serpong Telp. (021) 7412566, Fax. (021) 7412566
Tangerang Selatan - Banten

A. RANGKUMAN MATERI

Stack atau tumpukan adalah kumpulan elemen yang hanya dapat di tambah atau dihapus dari satu ujung (gerbang) yang sama. Hal ini menunjukkan bahwa seolah-olah suatu elemen diletakan di atas elemen yang lain. Yang memberi gambaran bahwa Stack mempunyai sifat LIFO (Last In First Out) yang berarti bahwa elemen yang terakhir masuk akan pertama keluar. Representasi Stack dapat dilakukan menggunakan Array atau Linked List. Kedua representasi mempunyai keunggulan dan kelemahan. Dengan Array, stack juga dapat disajikan dengan Single Stack dan Double Stack.

B. TUGAS PENDAHULUAN

1. Jelaskan perbedaan program Stack antara menggunakan Array dan Linked List!

Jawab :

Program stack menggunakan array.

Proses inisialisasi dimana proses ini untuk stack yang menggunakan array adalah dengan mengisi nilai field top dengan 0 (nol), jika elemen pertama diawali dengan nomor 1. Kalau elemen pertama array dimulai dengan 0 (contoh bahasa c), maka top di isi dengan nilai -1.

- Top yang menunjuk posisi data terakhir (top).
- Elemen yang berisi data yang ada dalam stack. Bagian ini lah yang berbentuk array.
- Maks elemen yaitu variabel yang menunjuk maksimal banyaknya elemen dalam stack

Program stack menggunakan linked list

Adapun stack yang menggunakan linked list, hanya memerlukan suatu pointer yang menunjuk ke data terakhir (perhatikan proses dihalaman sebelumnya), setiap elemen linked list mempunyai 2 field yaitu elemen datanya dan pointer bawah yang menunjuk posisi terakhir sebelum proses push

2. Jelaskan Aplikasi-Aplikasi Stack dalam dunia nyata!

Jawab :

Dalam dunia nyata bisa kita bayangkan seperti tumpukan buku, tumpukan kartu, atau tumpukan kursi yang tersusun secara menumpuk ke atas. Konsep stack yang utuh memiliki beberapa aturan atau batasan tersendiri yang membedakannya dengan struktur data lain, misalnya kita tidak bisa menambah data langsung ditengah-tengah tumpukan dengan cara diselipkan. Beberapa contoh aplikasi yang menerapkan stack, diantaranya adalah:

- Expression evaluation, baik ekspresi aritmatika, logik maupun boolean.
- Notasi infix, prefix, dan postfix, proses perhitungannya maupun konversi antar notasi tersebut
- Backtracking, contohnya history call pada browser (tombol back).
- Membantu penelusuran simpul pohon dengan algoritma DFS (Depth-First-Search).
- Manajemen memori dan alokasi memori, komputer modern saat ini menerapkan stack untuk memodelkan manajemen memori dari program yang sedang berjalan (running program).
- Permainan Tower of Hanoi.
- Konversi bilangan desimal ke biner. Sampai yang paling sederhana yaitu membalikkan urutan string

3. Tuliskan contoh program pada operasi Full!

Jawab :

```
int IsFull ()
{
    if (tumpuk.top == MAX_STACK-1)
        return 1;
    else
        return 0;
}
```

4. Tuliskan contoh program pada operasi Empty!

Jawab :

```
int IsEmpty ()
{
    if (tumpuk.top == -1)
        return 1;
    else
        return 0;
}
```

C. TUGAS PRAKTIKUM

- Lat12_1

```
#include<iostream>
#include<stdlib.h>

#define true 1
#define false 0

using namespace std;

typedef struct node *simpul;

struct node {
    char Isi;
    simpul Next;
};

void Sisip_Belakang(simpul &L, char elemen);
void Hapus_Belakang(simpul &L);
void Cetak(simpul L);

int main() {
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    char huruf;
    simpul L = NULL;
    cout << "Operasi Single Linked List Pada Stack\n\n";

    cout << "Penyisipan Stack \n\n";

    cout << "Masukkan Elemen : ";
    cin >> huruf;
    Sisip_Belakang(L, huruf);
    cout << "Masukkan Elemen : ";
    cin >> huruf;
    Sisip_Belakang(L, huruf);
    cout << "Masukkan Elemen : ";
    cin >> huruf;
    Sisip_Belakang(L, huruf);
    cout << "Masukkan Elemen : ";
    cin >> huruf;
    Sisip_Belakang(L, huruf);
    cout << "Masukkan Elemen : ";
    cin >> huruf;
    Sisip_Belakang(L, huruf);
    cout << "Masukkan Elemen : ";
    cin >> huruf;
    Sisip_Belakang(L, huruf);

    Cetak(L);

    cout << "\n\nHapus Elemen \n";
    Hapus_Belakang(L);
    Cetak(L);

    cout << "\n\nHapus Elemen \n";
    Hapus_Belakang(L);
    Cetak(L);
}
```

```

cout << "\n\nHapus Elemen \n";
Hapus_Belakang(L);
Cetak(L);

cout << "\n\nHapus Elemen \n";
Hapus_Belakang(L);
Cetak(L);

return 0;
}

void Sisip_Belakang(simpul &L, char elemen) {
    simpul baru = (simpul) malloc(sizeof(node));
    baru->Isi = elemen;
    baru->Next = NULL;

    if (L == NULL) {
        L = baru;
    } else {
        simpul bantu = L;
        while (bantu->Next != NULL) {
            bantu = bantu->Next;
        }
        bantu->Next = baru;
    }
}

void Hapus_Belakang(simpul &L) {
    if (L == NULL) {
        cout << "List Kosong, Tidak ada yang dihapus\n";
    } else if (L->Next == NULL) {
        free(L);
        L = NULL;
    } else {
        simpul bantu = L;
        while (bantu->Next->Next != NULL) {
            bantu = bantu->Next;
        }
        simpul hapus = bantu->Next;
        bantu->Next = NULL;
        free(hapus);
    }
}

void Cetak(simpul L) {
    if (L == NULL) {
        cout << "List Kosong\n";
    } else {
        simpul bantu = L;
        cout << "\nIsi List : ";
        while (bantu->Next != NULL) {
            cout << bantu->Isi << "->";
            bantu = bantu->Next;
        }
        cout << bantu->Isi;
    }
}

```

```

Nama    : Nova Ardiansyah
NIM     : 211011401309
=====

Operasi Single Linked List Pada Stack

Penyisipan Stack

Masukkan Elemen : a
Masukkan Elemen : b
Masukkan Elemen : c
Masukkan Elemen : d
Masukkan Elemen : e
Masukkan Elemen : f

Isi List : a->b->c->d->e->f

Hapus Elemen

Isi List : a->b->c->d->e

Hapus Elemen

Isi List : a->b->c->d

Hapus Elemen

Isi List : a->b->c

Hapus Elemen

Isi List : a->b
=====
Process exited after 8.578 seconds with return value 0
Press any key to continue . . .

```

- Lat12_2

```

#include<iostream>
#include<stdlib.h>

using namespace std;

typedef struct node *simpul;

struct node {
    char Isi;
    simpul Next;
};

void Sisip_Belakang(simpul &L, char elemen);
void Hapus_Belakang(simpul &L);
void Cetak(simpul L);
void TampilkanMenu(simpul &L);

int main() {
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    char huruf;
    simpul L = NULL;
    cout << "Operasi Single Linked List Pada Stack\n\n";

    int pilihan;
    do {
        TampilkanMenu(L);
        cout << "Pilih operasi (1-3): ";
        cin >> pilihan;

        switch (pilihan) {

```

```

    case 1:
        cout << "Masukkan Elemen : ";
        cin >> huruf;
        Sisip_Belakang(L, huruf);
        break;
    case 2:
        Hapus_Belakang(L);
        break;
    case 3:
        Cetak(L);
        break;
    case 0:
        cout << "Terima kasih. Program selesai.\n";
        break;
    default:
        cout << "Pilihan tidak valid. Silakan coba lagi.\n";
        break;
}
} while (pilihan != 0);

return 0;
}

void Sisip_Belakang(simpul &L, char elemen) {
    simpul baru = (simpul) malloc(sizeof(node));
    baru->Isi = elemen;
    baru->Next = NULL;

    if (L == NULL) {
        L = baru;
    } else {
        simpul bantu = L;
        while (bantu->Next != NULL) {
            bantu = bantu->Next;
        }
        bantu->Next = baru;
    }
}

void Hapus_Belakang(simpul &L) {
    if (L == NULL) {
        cout << "List Kosong, Tidak ada yang dihapus\n";
    } else if (L->Next == NULL) {
        free(L);
        L = NULL;
    } else {
        simpul bantu = L;
        while (bantu->Next->Next != NULL) {
            bantu = bantu->Next;
        }
        simpul hapus = bantu->Next;
        bantu->Next = NULL;
        free(hapus);
    }
}

void Cetak(simpul L) {
    if (L == NULL) {
        cout << "List Kosong\n";
    } else {
        simpul bantu = L;
        cout << "\nIsi List : ";
        while (bantu->Next != NULL) {

```

```

        cout << bantu->Isi << "->";
        bantu = bantu->Next;
    }
    cout << bantu->Isi;
}
cout << endl;
}

void TampilkanMenu(simpul &L) {
    cout << "\nMenu:\n";
    cout << "1. Sisipkan Elemen\n";
    cout << "2. Hapus Elemen\n";
    cout << "3. Cetak List\n";
    cout << "4. Keluar\n";
}

```

```

Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

Operasi Single Linked List Pada Stack

Menu:
1. Sisipkan Elemen
2. Hapus Elemen
3. Cetak List
4. Keluar
Pilih operasi (1-3): 1
Masukkan Elemen : a

Menu:
1. Sisipkan Elemen
2. Hapus Elemen
3. Cetak List
4. Keluar
Pilih operasi (1-3): 1
Masukkan Elemen : b

Menu:
1. Sisipkan Elemen
2. Hapus Elemen
3. Cetak List
4. Keluar
Pilih operasi (1-3): 3

Isi List : a->b

```


LAPORAN PRAKTIKUM STRUKTUR DATA

LAPORAN AKHIR PRAKTIKUM

Pertemuan ke-12

Stack Lanjut



Disusun Oleh:

Nama Lengkap : Nova Ardiansyah

NIM : 211011401309

Kelas : 04-TPLE008

TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PAMULANG

Jl. Raya Puspitek No. 11 Buaran, Serpong Telp. (021) 7412566, Fax. (021) 7412566
Tangerang Selatan - Banten

A. TUGAS AKHIR

1. Buatlah program untuk mengetahui suatu kalimat adalah polindrom atau tidak!

Polindrom adalah suatu kata atau kalimat yang jika dibaca dari depan akan sama maknanya dengan jika dibaca dari belakang. Contoh: "KASUR NABABAN RUSAK" maka jika kalimat tersebut dibalik akan mempunyai makna yang sama yaitu: "KASUR NABABAN RUSAK"!

Jawab :

```
#include<iostream>
#include<string>

using namespace std;

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    string kalimat;
    cout << "Masukkan Kalimat : ";
    getline(cin, kalimat);

    int len = kalimat.length();
    bool flag = true;

    for(int i = 0, j = len - 1; i < len / 2; ++i, --j)
    {
        if(kalimat[j] != kalimat[i])
        {
            flag = false;
            break;
        }
    }

    cout << "\nKata Setelah Dibalik: ";
    for(int i = len - 1; i >= 0; --i)
    {
        cout << kalimat[i];
    }

    if(flag)
        cout << "\nKalimat Ini adalah Kalimat Palindrome";
    else
        cout << "\nKalimat Ini Bukan Kalimat Palindrome";

    return 0;
}
```

```
Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

Masukkan Kalimat : kasur nababan rusak

Kata Setelah Dibalik: kasur nababan rusak
Kalimat Ini adalah Kalimat Palindrome
=====
Process exited after 11.29 seconds with return value 0
Press any key to continue . . . █
```

2. KESIMPULAN

Stack adalah suatu tumpukan. Konsep utama dari stack adalah LIFO (Last In First Out), yaitu benda yang terakhir masuk ke dalam stack akan menjadi benda pertama yang dikeluarkan dari tumpukan. Dalam C++ ada dua cara penerapan stack. Sesuai dengan sifat stack, maka pengambilan/penghapusan elemen dalam stack harus dimulai dari elemen teratas. Deklarasi konstanta, tipe, dan variable yang akan dipakai dalam penjelasan operasi-operasi stack dengan array.