

PERTEMUAN XII

STACK (Lanjut)

TUJUAN PRAKTIKUM

- a) Mahasiswa dapat menjelaskan pengertian dan pembuatan Stack dengan C++.
- b) Mahasiswa dapat melakukan operasi penyisipan dan penghapusan elemen dalam Stack dengan C++.
- c) Mahasiswa dapat mengimplementasikan Stack dengan C++.

TEORI DASAR

a) Pendahuluan

Stack atau tumpukan adalah kumpulan elemen yang hanya dapat di tambah atau dihapus dari satu ujung (gerbang) yang sama. Hal ini menunjukkan bahwa seolaholah 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

TUGAS PENDAHULUAN

1. Jelaskan perbedaan program Stack antara menggunakan Array dan Linked List!
2. Jelaskan Aplikasi-Aplikasi Stack dalam dunia nyata!
3. Tuliskan contoh program pada operasi Full!
4. Tuliskan contoh program pada operasi Empty!

JAWABAN

1. 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. 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. Begitu juga ketika kita mengambilnya, melainkan setiap operasi harus dikerjakan pada data yang berada dipaling atas tumpukan.

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. Contoh program pada operasi full.

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

4. Contoh program pada operasi empty.

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

TUGAS PRAKTIKUM

- a) Buatlah program lengkap dari suatu Stack dengan menggunakan Array (simpan dengan nama lat11_1.cpp)

The image displays two screenshots of a C++ program named `lat11_1.cpp` and its execution. The program implements a stack using an array.

Source Code (lat11_1.cpp):

```
1 #include<iostream>
2 #include<conio.h>
3 #define maxs 10
4 using namespace std;
5 struct Stack
6 {
7     char Isi [maxs];
8     unsigned int Top;
9 };
10 void INITS (Stack &S) ;
11 void PUSH (Stack &S, char Data) ;
12 void CETAK (Stack &S) ;
13 void POP (Stack &S, char &Hsl) ;
14 main ( )
15 {
16     char huruf ;
17     Stack S;
18     INITS (S) ;
19     cout<<"Masukan Karakter :";
20     cin>>huruf ;
21     PUSH(S, huruf);
22     cout<<"Masukan Karakter :";
23     cin>>huruf ;
24     PUSH(S, huruf);
25     cout<<"Masukan Karakter :";
26     cin>>huruf ;
27     PUSH(S, huruf);
28     CETAK (S);
29     POP (S, huruf);
30     cout<<endl<<"Yang Dihapus .....<churuf;
31     CETAK (S) ;
32     cout<<endl<<"Masukan Karakter :";
33     cin>>huruf ;
34     PUSH(S, huruf);
35     cout<<"Masukan karakter :";
36     cin>>huruf;
37     PUSH(S, huruf);
38     cout<<"Masukan karakter :";
39     cin>>huruf;
40     PUSH(S, huruf);
41     CETAK (S) ;
42     POP (S, huruf);
43     cout<<endl<<"Yang Dihapus .....<churuf;
44     CETAK (S) ;
45     getch ( ) ;
46 }
47 void INITS (Stack &S)
48 {
49     S.Top = 0;
50 }
51 void PUSH (Stack &S, char Data)
52 {
53     if (S.Top < maxs)
54     {
55         S.Top++;
56         S.Isi [S.Top] = Data;
57     }
58     else
59         cout<<"Stack penuh.....";
60 }
61 void CETAK (Stack S)
62 {
63     int i;
64     cout<<endl<<"Isi Stack : ";
65     if (S.Top != 0)
66     {
67         for(i=1;i<=S.Top;i++){
68             cout<<S.Isi [i];
69         }
70     }
71     else
72         cout<<"Stack Kosong ....";
73 }
74 void POP (Stack &S, char &Hsl)
75 {
76     if (S.Top !=0)
77     {
78         Hsl = S.Isi [S.Top];
79         S.Top--;
80     }
81     else
82         cout<<"Stack Kosong...";
83 }
84 //=====endif of file=====
```

Execution Output:

```
Masukan Karakter :r
Masukan Karakter :o
Masukan Karakter :n

Isi Stack : no
Yang Dihapus .....n
Isi Stack : r
Masukan Karakter :y
Masukan Karakter :q
Masukan Karakter :e

Isi Stack : royq
Yang Dihapus .....e
Isi Stack : roy
.....
Process exited after 44.27 seconds with return value 0
Press any key to continue . . .
```

- b) Buatlah program untuk membalik karakter-karakter dalam suatu kalimat (karakter depan menjadi belakang dan karakter belakang menjadi karakter depan) dengan menggunakan Stack (simpan dengan nama lat11_2.cpp)

```
1 #include <iostream>
2 #include <stdlib.h>
3 #include <conio.h>
4 #include <windows.h>
5
6 using namespace std;
7
8 struct Tumpukan
9 {
10     char isi[25];
11     int atas;
12 }
13
14 void push (char x){
15     if (T.atas==25){
16         cout<<"Tumpukan sudah penuh";
17         getch();
18     }else{
19         T.atas=T.atas+1;
20         T.isi[T.atas]=x;
21     }
22 }
23
24 char pop(){
25     char hasil;
26     if (T.atas == 0){
27         cout<<"Tumpukan sudah kosong";
28         hasil = ' ';
29     }else{
30         hasil = T.isi[T.atas];
31         T.atas = T.atas - 1;
32     }
33     return hasil;
34 }
35
36 int main()
37 {
38     int i;
39     char kalimat[25];
40     system("CLS");
41     T.atas=0;
42     cout<<"masukan kalimat : ";cin.getline(kalimat,sizeof(kalimat));
43     system("CLS");
44     cout<<"kalimat asli : "<<kalimat;
45     for(i=0; i<strlen(kalimat); i++)
46     {
47         push(kalimat[i]);
48     }
49     cout<<"\nkalimat Setelah di Balik : ";
50     for(i=0; i<strlen(kalimat); i++)
51     {
52         cout<<pop();
53     }
54     cout<<endl;
55     return 0;
56 }
```

kalimat asli : roni
kalimat Setelah di Balik : inor

Process exited after 3.655 seconds with return value 0
Press any key to continue . . .

TUGAS AKHIR

The image shows a C++ IDE with two windows. The top window displays the source code for a stack implementation in `RoniSefia lat11_2.cpp`. The code includes standard headers, defines a stack structure with an array of integers and a count, and implements functions for initialization, checking if empty or full, pushing, and popping elements. The bottom window shows the execution of the program, which prompts the user to input a decimal number and then displays its binary representation.

```
1 #include<stdio.h>
2 #include<conio.h>
3 using namespace std;
4 int MAXSTACK; typedef int itemType;
5
6 typedef struct
7 {
8     itemType item[300]; int count;
9 }stack;
10
11 void initializestack(stack *s)
12 {
13     s->count = 0;
14 }
15
16 int empty(stack *s)
17 {
18     return (s->count == 0);
19 }
20
21 int full(stack *s)
22 {
23     return (s->count == MAXSTACK);
24 }
25
26 void push(itemType x, stack *s)
27 {
28     if(full(s))
29         printf("stack penuh !\n");
30     else
31     {
32         s->item[s->count]=x; ++(s->count);
33     }
34 }
35
36 int pop(stack *s)
37 {
38     if(empty(s))
39         printf("stack kosong\n");
40     else
41     {
42         --(s->count);
43     }
44 }
45
46 main()
47 {
48     int i, n, m, l, z; int input;
49     stack tumpukan;
50
51     printf("Program Pengkonversi Desimal ke Biner\n\n");
52     initializestack(&tumpukan);
53     printf("Masukkan bilangan desimal = ");
54     scanf("%d", &input);
55
56     for(z=1,n=1;n>0;n=n/2, z++)
57     {
58         MAXSTACK=z;
59     }
60     m=0;
61     for(n=1;n>0;n=n/2)
62     {
63         l=n%2;
64         push(l,&tumpukan);
65         ++m;
66     }
67
68     printf("Masukkan bilangan biner = ");
69     for(i=MAXSTACK;i>0;i--)
70     {
71         printf("%d", pop(&tumpukan));
72     }
73
74     getch();
75     return 0;
76 }
```

Program Pengkonversi Desimal ke Biner

Masukkan bilangan desimal = 12
Masukkan bilangan biner = 1100

Process exited after 5.46 seconds with return value 0
Press any key to continue . . .

Kesimpulan

Stack merupakan kumpulan elemen yang hanya dapat di tambahatau dihapus dari satu ujung (gerbang) yang sama. Stack memiliki sifat LIFO(Last In First Out). Stack juga memiliki 2 operasi yaitu POP Dan PUSH .

LAPORAN AWAL

PRAKTIKUM

STRUKTUR DATA

LAPORAN KE - 11



Disusun Oleh :

Nama : Roni Sefia
NIM : 201011401617
Kelas : 04TPLP016

TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PAMULANG

Jl. Surya Kencana No.1, Pamulang Bar., Kec. Pamulang, Kota Tangerang
Selatan, Banten 15417

LAPORAN AKHIR

PRAKTIKUM

STRUKTUR DATA

LAPORAN KE - 10



Disusun Oleh :

Nama : Roni Sefia
NIM : 201011401617
Kelas : 04TPLP016

TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PAMULANG

Jl. Surya Kencana No.1, Pamulang Bar., Kec. Pamulang, Kota Tangerang
Selatan, Banten 15417

