MODUL IV

STACK

A. TUJUAN

1. Memahami terminologi yang terkait dengan struktur data stack
2. Memahami operasi-operasi yang ada dalam stack
3. Dapat mengidentifikasi permasalahan-permasalahan pemrograman yang harus diselesaikan dengan menggunakan stack, sekaligus menyelesaikannya

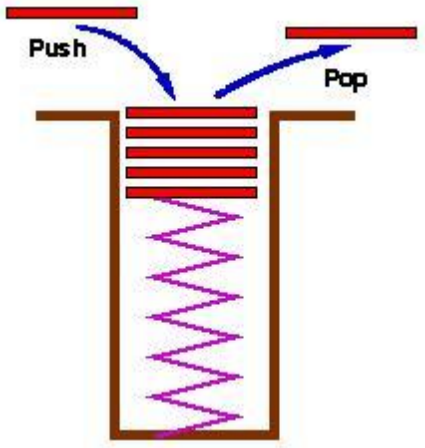
B. DASAR TEORI

Pengertian Stack

Stack adalah sebuah kumpulan data dimana data yang diletakkan di atas data yang lain. Dengan demikian stack adalah struktur data yang menggunakan konsep LIFO. Dengan demikian, elemen terakhir yang disimpan dalam stack menjadi elemen pertama yang diambil. Dalam proses komputasi, untuk meletakkan sebuah elemen pada bagian atas dari stack, maka kita melakukan push. Dan untuk memindahkan dari tempat yang atas tersebut, kita melakukan pop.

Ada beberapa cara untuk menyajikan sebuah stack tergantung pada permasalahan yang akan kita selesaikan. Dalam bab ini kita akan menggunakan cara yang paling sederhana, tipe data yang sudah kita kenal, yaitu array. Kita dapat menggunakan array untuk menyajikan sebuah stack, dengan anggapan bahwa banyaknya elemen maksimum dari stack tersebut tidak akan melebihi batas maksimum banyaknya elemen dalam array.

Pada saat ukuran stack, kalau kita teruskan menambah data lagi, akan terjadi overflow. Dengan demikian perlu data tambahan untuk mencatat posisi ujung stack. Dengan kebutuhan seperti ini, kita dapat menyajikan stack dengan menggunakan tipe data struktur (struct) yang terdiri dari dua field. Field pertama bertipe array untuk menyimpan elemen stack, medan kedua bertipe integer untuk mencatat posisi ujung stack.



Gambar 3.1 Ilustrasi Sebuah Stack

Operasi Pada Stack

Operasi-operasi Dasar pada stack adalah sebagai berikut:

1. Operasi data abstrak STACK

```
5   struct STACK {  // Membuat jenis data abstrak 'STACK'
6       int top;
7       float data[4];
8   };
9   float dta;
10  struct STACK stackbaru;
```

2. Fungsi yang melakukan pengecekan apakah stack dalam kondisi kosong

```
16  bool isempty() {                // Menanyakan : kosongkah ?
17      if(stackbaru.top==-1) return true;
18      else return false;
19  }
```

3. Fungsi yang melakukan pengecekan apakah stack dalam kondisi penuh

```
12  bool isfull() {             // Menanyakan : penuhkah?
13      if (stackbaru.top == maxstack) return true;
14      else return false;
15  }
```

4. Fungsi untuk menghapus seluruh stack

```
44  void clear () {
45      top = -1;
46  }
```

5. Fungsi untuk mencetak isi stack

```
38  void print() {      // 6. Mencetak stack
39      for (int i=0; i<=top; i++) {
40          cout <<stackbaru.data[i] <<"     ";
41      }
42  }
```

C. LATIHAN

Troubleshooting Program STACK

```
# include <iostream>
# define maxstack 5
using namespace std;

struct STACK {  // Membuat jenis data abstrak 'STACK'
    int top;
    float data[4];
};
float dta;
struct STACK stackbaru;

bool isfull() {             // Menanyakan : penuhkah?
    if (stackbaru.top == maxstack) return true;
    else return false;
}
bool isempty() {                // Menanyakan : kosongkah ?
    if(stackbaru.top==-1) return true;
    else return false;
}
void push(float dta) {      // Mengisi stack (menyimpan data di stack)
    if (isfull() == false) {
        puts ("Maaf, stack penuh");

    } else {
        stackbaru.top++;
        stackbaru.data[top]=dta;
    }
}
void pop() {                // Mengambil isi stack
    if (isempty() == false) {
        cout <<"Data telah kosong!";

    } else {
        cout <<"data yang terambil : " <<stackbaru.data[top] <<endl;
        stackbaru.top--;
    }
}
void print() {      // 6. Mencetak stack
    for (int i=0; i<=top; i++) {
        cout <<stackbaru.data[i] <<"    ";
    }
}

void clear () {
    top = -1;
}
```

```
49  void main()
50  {
51      inisialisasi();
52
53      char menu;
54      char uulang;
55
56  do {
57          system("cls");
58      printf("Menu:\n");
59      puts ("1. pop stack");
60      puts ("2. push stack");
61      puts ("3. Cetak");
62      puts ("4. Bersihkan stack");
63      puts ("5. Exit");
64
65      cout << "Menu pilihan Anda:  ";
66      cin >> menu;
67
68          if (menu == '1') {
69              pop();
70              uulang = 'y';
71
72          } else if (menu == '2') {
73              cout <<"data yang akan disimpan di stack:  ";
74              cin >>dta;
75              push(dta);
76              uulang = 'y';
77
78          } else if (menu == '3') {
79              print();
80              cout <<"Ulang? (y/t)";
81              cin >> uulang;
82
83          } else if (menu == '4') {
84              clear();
85              cout <<"Ulang? (y/t)";
86              cin >> uulang;
```

```
57          system("cls");
58      printf("Menu:\n");
59      puts ("1. pop stack");
60      puts ("2. push stack");
61      puts ("3. Cetak");
62      puts ("4. Bersihkan stack");
63      puts ("5. Exit");
64
65      cout << "Menu pilihan Anda:  ";
66      cin >> menu;
67
68          if (menu == '1') {
69              pop();
70              uulang = 'y';
71
72          } else if (menu == '2') {
73              cout <<"data yang akan disimpan di stack:  ";
74              cin >>dta;
75              push(dta);
76              uulang = 'y';
77
78          } else if (menu == '3') {
79              print();
80              cout <<"Ulang? (y/t)";
81              cin >> uulang;
82
83          } else if (menu == '4') {
84              clear();
85              cout <<"Ulang? (y/t)";
86              cin >> uulang;
87
88          } else if (menu == '5') {
89              exit(0);
90          }
91
92      } while( uulang == 'Y' || uulang == 'y');
93
94  }
```

E. TUGAS:

Buatlah program yang melakukan pembalikan kalimat dengan menggunakan stack

Contoh:

Kalimat : Struktur Data

Menjadi : ataD rutkurtS