

LAPORAN PRAKTIKUM STRUKTUR DATA

LAPORAN AWAL PRAKTIKUM

Pertemuan ke-11

Stack



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. Secara sederhana stack dimisalkan kita mempunyai 4 buah kotak (A,B,C, dan D) yang ditumpukkan. Kotak A diletakkan paling bawah, lalu diikuti kotak B, C, dan yang teratas atau terakhir adalah D. Maka untuk mengambil tiap kotak harus dilakukan berurutan dari kotak D, C, B kemudian A. Karena jika kita mengambil kotak B tanpa terlebih dahulu mengambil kotak di atasnya maka tumpukan akan roboh.

B. TUGAS PENDAHULUAN

1. Apa yang dimaksud dengan Stack!

Jawab :

Stack atau Tumpukan adalah suatu struktur data yang terbentuk dari barisan hingga yang terurut dari satuan data. Pada Stack, penambahan dan penghapusan elemennya hanya dapat dilakukan pada satu posisi, yaitu posisi akhir stack.

2. Bagaimana tahapan-tahapan proses operasi PUSH!

Jawab :

- Periksa apakah stack penuh (isfull), jika bernilai false/0 (tidak penuh) maka proses push dilaksanakan dan jika pemeriksaan ini bernilai true/1, maka proses push digagalkan.
- Proses push-nya sendiri adalah dengan menambahkan field top dengan 1, kemudian elemen pada posisi top di isi dengan elemen data baru

3. Bagaimana tahapan-tahapan proses operasi POP!

Jawab :

Operasi ini biasanya dibuat dalam bentuk function yang me-return-kan nilai sesuai data yang ada di top. Operasi pop pada stack yang menggunakan array adalah terlebih dahulu memeriksa apakah stack sedang keadaan kosong, jika tidak kosong maka data diambil pada posisi yang ditunjuk oleh posisi top, kemudian disimpan dalam variabel baru dengan nama “data”.

4. Jelaskan karakteristik-karakteristik dari Stack!

Jawab :

- Elemen stack yaitu item-item data di elemen stack.
- Top (elemen puncak dari stack)
- Jumlah elemen pada stack
- Status/kondisi stack Kondisi stack yang menjadi perhatian adalah [penuh/kosong]

C. TUGAS PRAKTIKUM

- Lat11_1

```
#include <iostream>
using namespace std;

#define MaxS 10

struct Stack {
    char Isi[MaxS];
    int Top;
};

void INITS(Stack& S);
void PUSH(Stack& S, char Data);
void CETAK(Stack S);
char POP(Stack& S, char& Hsl);

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

    char huruf;
    Stack S;
    INITS(S);

    cout << "Masukkan Karakter : ";
    cin >> huruf;
    PUSH(S, huruf);

    cout << "Masukkan Karakter : ";
    cin >> huruf;
    PUSH(S, huruf);

    cout << "Masukkan Karakter : ";
    cin >> huruf;
    PUSH(S, huruf);

    CETAK(S);
    POP(S, huruf);

    cout << "\nYang dihapus ... : " << huruf << endl;
    CETAK(S);

    cout << "\nMasukkan Karakter : ";
    cin >> huruf;
    PUSH(S, huruf);

    cout << "Masukkan Karakter : ";
    cin >> huruf;
    PUSH(S, huruf);

    cout << "Masukkan Karakter : ";
    cin >> huruf;
    PUSH(S, huruf);

    CETAK(S);
    POP(S, huruf);
```

```

    cout << "\nYang dihapus ... : " << huruf << endl;
    CETAK(S);

    return 0;
}

void INITS(Stack& S)
{
    S.Top = -1;
}

void PUSH(Stack& S, char Data)
{
    if (S.Top < MaxS - 1) {
        S.Top++;
        S.Isi[S.Top] = Data;
    }
    else {
        cout << "Stack Penuh";
    }
}

void CETAK(Stack S)
{
    int i;
    cout << "\nIsi Stack : ";

    if (S.Top != -1) {
        for (i = 0; i <= S.Top; i++) {
            cout << S.Isi[i] << " ";
        }
    }
    else {
        cout << "Stack Kosong";
    }
}

char POP(Stack& S, char& Hsl)
{
    if (S.Top != -1) {
        Hsl = S.Isi[S.Top];
        S.Top--;
    }
    else {
        cout << "Stack Kosong";
    }

    return Hsl;
}

```

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

```
Masukkan Karakter : a
Masukkan Karakter : b
Masukkan Karakter : c
```

```
Isi Stack : a b c
Yang dihapus ... : c
```

```
Isi Stack : a b
Masukkan Karakter : d
Masukkan Karakter : e
Masukkan Karakter : f
```

```
Isi Stack : a b d e f
Yang dihapus ... : f
```

```
Isi Stack : a b d e
```

- Lat11_2

```
#include <iostream>
#include <stack>
using namespace std;

#define MaxS 10

struct Stack {
    char Isi[MaxS];
    int Top;
};

void INITS(Stack& S);
void PUSH(Stack& S, char Data);
void CETAK(Stack S);
char POP(Stack& S, char& Hsl);

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

    char huruf;
    Stack S;
    INITS(S);

    cout << "Masukkan Karakter : ";
    cin >> huruf;
    PUSH(S, huruf);

    cout << "Masukkan Karakter : ";
    cin >> huruf;
    PUSH(S, huruf);

    cout << "Masukkan Karakter : ";
    cin >> huruf;
    PUSH(S, huruf);

    CETAK(S);
    POP(S, huruf);

    cout << "\nYang dihapus ... : " << huruf << endl;
    CETAK(S);
}
```

```

cout << "\nMasukkan Karakter : ";
cin >> huruf;
PUSH(S, huruf);

cout << "Masukkan Karakter : ";
cin >> huruf;
PUSH(S, huruf);

cout << "Masukkan Karakter : ";
cin >> huruf;
PUSH(S, huruf);

CETAK(S);
POP(S, huruf);

cout << "\nYang dihapus ... : " << huruf << endl;
CETAK(S);

// Membalik karakter-karakter dalam stack
stack<char> charStack;
for (int i = 1; i <= S.Top; i++) {
    charStack.push(S.Isi[i]);
}

cout << "\nKarakter yang terbalik: ";
while (!charStack.empty()) {
    cout << charStack.top();
    charStack.pop();
}

return 0;
}

void INITS(Stack& S)
{
    S.Top = 0;
}

void PUSH(Stack& S, char Data)
{
    if (S.Top < MaxS) {
        S.Top++;
        S.Isi[S.Top] = Data;
    } else {
        cout << "Stack Penuh";
    }
}

void CETAK(Stack S)
{
    int i;
    cout << "\nIsi Stack : ";

    if (S.Top != 0) {
        for (i = 1; i <= S.Top; i++) {
            cout << S.Isi[i] << " ";
        }
    } else {
        cout << "Stack Kosong";
    }
}

```

```

char POP(Stack& S, char& Hsl)
{
    if (S.Top != 0) {
        Hsl = S.Isi[S.Top];
        S.Top--;
    } else {
        cout << "Stack Kosong";
    }

    return Hsl;
}

```

```

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

Masukkan Karakter : a
Masukkan Karakter : b
Masukkan Karakter : c

Isi Stack : a b c
Yang dihapus ... : c

Isi Stack : a b
Masukkan Karakter : d
Masukkan Karakter : e
Masukkan Karakter : f

Isi Stack : a b d e f
Yang dihapus ... : f

Isi Stack : a b d e
Karakter yang terbalik: edba

```


LAPORAN PRAKTIKUM STRUKTUR DATA

LAPORAN AKHIR PRAKTIKUM

Pertemuan ke-11

Stack



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 mengkonversi bilangan desimal menjadi bilangan biner dengan menggunakan Stack!

Jawab :

```
#include <iostream>
using namespace std;

const int MAX_SIZE = 100;

void DecimalToBinary(int decimal) {
    int binary[MAX_SIZE];
    int index = 0;

    while (decimal > 0) {
        int remainder = decimal % 2;
        binary[index++] = remainder;
        decimal /= 2;
    }

    cout << "Biner: ";
    for (int i = index - 1; i >= 0; i--) {
        cout << binary[i];
    }
    cout << endl;
}

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

    int decimal;

    cout << "Masukkan bilangan desimal: ";
    cin >> decimal;

    DecimalToBinary(decimal);

    return 0;
}
```

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

Masukkan bilangan desimal: 15
Biner: 1111

-----
Process exited after 4.582 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 di pakai dalam penjelasan operasi-operasi stack dengan array.