

TUGAS PENDAHULUAN
MODUL 7
“Stack“



Disusun Oleh:
Isabelle Putri Ardini - 2311104030
SE-07-01

Dosen:
Yudha Islami Sulistya

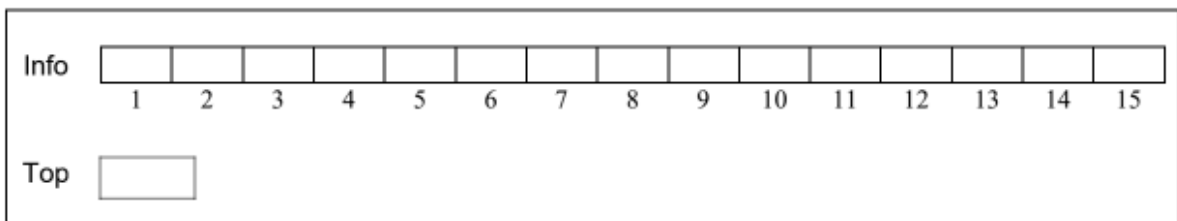
PROGRAM STUDI SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

SOAL TUGAS PENDAHULUAN

Anda diminta untuk membuat stack dengan representasi statis, yaitu menggunakan array. Untuk itu buatlah ADT stack pada file "stack.h" dengan struktur berikut:

Notasi Algoritmik	Bahasa C++
<pre>type infotype : char type stack : < info : array[1..15] of infotype top : integer ></pre>	<pre>typedef int infotype; struct stack { infotype info[15]; int Top; };</pre>

Ilustrasi untuk struktur di atas adalah:



Elemen info[1..15] akan digunakan untuk menampung data char. Sedangkan Top akan digunakan untuk menampung informasi mengenai indeks elemen info yang paling atas. Jadi, Top = 0 artinya stack dalam keadaan kosong. Top = 1, artinya stack berisi satu elemen.

Prosedur dan fungsi yang digunakan adalah

Notasi Algoritmik	Bahasa C++
<pre>procedure createStack(input/output S : stack) { I.S. sembarang F.S. terbentuk stack dengan Top = 0 }</pre>	<pre>void createStack(stack &S);</pre>
<pre>function isEmpty(S : stack): boolean { Mengembalikan nilai true jika stack kosong }</pre>	<pre>bool isEmpty(stack S);</pre>
<pre>function isFull(S : stack): boolean { Mengembalikan nilai true jika stack penuh }</pre>	<pre>bool isFull(stack S);</pre>

<u>procedure</u> push(<u>input/output</u> S : stack, <u>input</u> x : <u>infotype</u>) { I.S. mungkin kosong F.S. menambahkan elemen pada stack dengan nilai x, Top = Top + 1 }	<u>void</u> push(<u>stack</u> &S, <u>infotype</u> x);
<u>function</u> pop(<u>input/output</u> S : stack): <u>infotype</u> { Mengembalikan nilai pada indeks Top, Top = Top - 1 }	<u>int</u> pop(<u>stack</u> &S);
<u>procedure</u> printInfo(<u>input</u> S : stack) { I.S. stack mungkin kosong F.S. Jika stack tidak kosong, maka menampilkan semua info yang ada pada stack }	<u>void</u> printInfo(<u>stack</u> S);

Gunakan alias sebagai berikut:

Top(S) untuk (S).Top

info(S) untuk (S).info

Buatlah implementasi ADT stack pada file "stack.cpp".

Notasi Algoritmik	Bahasa C++
<u>procedure</u> createStack(<u>input/output</u> S : stack) { I.S. sembarang F.S. terbentuk stack dengan Top = 0 } Kamus: { Tidak ada kamus } Algoritma: Top(S) ← 0	<u>void</u> createStack(<u>stack</u> &S) { /* Lengkapi kodenya */ }

Notasi Algoritmik	Bahasa C++
<u>function</u> isEmpty(S : stack): <u>boolean</u> { Mengembalikan nilai true jika stackkosong } Kamus: { Tidak ada kamus } Algoritma: if Top(S) = 0 <u>then</u> → <u>true</u> <u>else</u> → <u>false</u> { <u>end if</u> }	<u>bool</u> isEmpty(<u>stack</u> S) { /* Lengkapi kodenya */ }

Notasi Algoritmik	Bahasa C++
<p><u>function</u> isFull(S : stack): <u>boolean</u> { Mengembalikan nilai true jika stack penuh } Kamus: { Tidak ada kamus } Algoritma: if Top(S) = 15 <u>then</u> → <u>true</u> <u>else</u> → <u>false</u> { end if }</p>	<pre>bool isFull(stack S) { /* Lengkapi kodenya */ }</pre>

Notasi Algoritmik	Bahasa C++
<p><u>procedure</u> push(input/output S : stack, input x : <u>infotype</u>) { I.S. mungkin kosong F.S. menambahkan elemen pada stack dengan nilai x, Top = Top + 1 } Kamus: { Tidak ada kamus } Algoritma: if isFull(S) = <u>false</u> <u>then</u> Top(S) ← Top(S) + 1 Info(S)[Top(S)] ← x { end if }</p>	<pre>void push(stack &S, infotype x) { /* Lengkapi kodenya */ }</pre>

Notasi Algoritmik	Bahasa C++
<p><u>function</u> pop(input/output S : stack): infotype { Mengembalikan nilai infotype yang ada pada indeks Top, Top = Top - 1 } Kamus: x : infotype Algoritma: x ← info(S)[Top(S)] Top(S) ← Top(S) - 1 → x</p>	<pre>int pop(stack &S) { /* Lengkapi kodenya */ }</pre>

Notasi Algoritmik	Bahasa C++
<pre> procedure printInfo(input S : stack) { I.S. stack mungkin kosong F.S. Jika stack tidak kosong, maka menampilkan semua info yang ada pada stack } Kamus: i : integer Algoritma: for i ← Top(S) downto 1 do output(info(S)[i]) { end for } </pre>	<pre> void printInfo(stack S) { /* Lengkapi kodenya */ } </pre>

Untuk menguji implementasi stack buatlah kode pada "main.cpp" sehingga menghasilkan output berikut ini dengan memanggil function/procedure yang sudah dipakai di atas:

- Digit terakhir NIM MOD 4 sisa 0 :

Output:

I F L A B J A Y A
J A Y A

- Digit terakhir NIM MOD 4 sisa 1 :

Output:

H A L O B A N D U N G
B A N D U N G

- Digit terakhir NIM MOD 4 sisa 2 :

Output:

P E R C A Y A D I R I
D I R I

- Digit terakhir NIM MOD 4 sisa 3 :

Output:

S T R U K T U R D A T A
D A T A

Note:

Output:

isi stack awal

isi stack sesudah pop

JAWABAN:

```

#include <iostream>
#include "stack.h"
using namespace std;

void createStack(stack &S)
{
    S.Top = 0;
}
bool isEmpty(stack S)
{
    return S.Top == 0;
}
bool isFull(stack S)
{
    return S.Top == 15;
}
void push(stack &S, infotype x)
{
    if (!isFull(S))
    {
        S.Top += 1;
        S.info[S.Top - 1] = x;
    }
    else
    {
        cout << "Stack penuh!" << endl;
    }
}

```

```

infotype pop(stack &S)
{
    if (!isEmpty(S))
    {
        infotype x = S.info[S.Top - 1];
        S.Top -= 1;
        return x;
    }
    else
    {
        cout << "Stack kosong!" << endl;
        return '\0';
    }
}
void printInfo(stack S)
{
    if (!isEmpty(S))
    {
        for (int i = S.Top - 1; i >= 0; i--)
        {
            cout << S.info[i] << " ";
        }
        cout << endl;
    }
    else
    {
        cout << "Stack kosong!" << endl;
    }
}

```

```

#include <iostream>
#include "stack.cpp" // pastikan file stack.cpp sudah disertakan
using namespace std;

int main() {
    stack S;
    createStack(S);

    // Mengisi stack dengan huruf "I R I D A Y A C R E P" (urutan terbalik)
    char kata[] = {'I', 'R', 'I', 'D', 'A', 'Y', 'A', 'C', 'R', 'E', 'P'};
    for (int i = 0; i < 11; i++) {
        push(S, kata[i]);
    }

    // Menampilkan isi stack awal
    cout << "Isi stack awal: ";
    printInfo(S);

    // Mengeluarkan 7 elemen terakhir agar hanya tersisa "D I R I"
    for (int i = 0; i < 7; i++) {
        pop(S);
    }

    // Menampilkan isi stack setelah operasi pop
    cout << "Isi stack setelah pop: ";
    printInfo(S);

    return 0;
}

```

1. Buatlah program untuk menentukan apakah kalimat tersebut yang diinputkan dalam program stack adalah palindrom/tidak. Palindrom kalimat yang dibaca dari depan dan belakang sama. Jelaskan bagaimana cara kerja programnya.

Contoh:

Kalimat : ini

Kalimat tersebut adalah polindrom

Kalimat : telkom

Kalimat tersebut adalah bukan polindrom

```
Masukan Kalimat : ini
Kalimat tersebut adalah : Palindrom
```

2. Buatlah program untuk melakukan pembalikan terhadap kalimat menggunakan stack dengan minimal 3 kata. Jelaskan output program dan source codenya beserta operasi/fungsi yang dibuat?

Contoh

Kalimat : Telkom Purwokerto

Hasil : otrekowruP mogleT

```
Masukkan Kata Telkom Purwokerto
Datastack Array :
Data : otrekowruP mogleT
```

JAWABAN:

1. Program:

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

```

bool isPalindrome(string kalimat) {
    stack<char> S;

    // Memasukkan setiap karakter kalimat ke dalam stack
    for (char ch : kalimat) {
        S.push(ch);
    }

    // Mengambil karakter dari stack untuk membentuk kalimat terbalik
    string reversedKalimat = "";
    while (!S.empty()) {
        reversedKalimat += S.top();
        S.pop();
    }

    // Membandingkan kalimat asli dengan kalimat terbalik
    return kalimat == reversedKalimat;
}

int main() {
    string kalimat;
    cout << "Masukkan Kalimat: ";
    cin >> kalimat;

    // Mengecek apakah kalimat tersebut adalah palindrom
    if (isPalindrome(kalimat)) {
        cout << "Kalimat tersebut adalah palindrom" << endl;
    } else {
        cout << "Kalimat tersebut adalah bukan palindrom" << endl;
    }

    return 0;
}

```

Cara kerja:

- **Mengisi Stack:** Setiap karakter dalam kalimat dimasukkan ke dalam stack satu per satu. Misalnya, jika kalimatnya adalah "ini", maka stack akan berisi:
- **Membalikkan Kalimat:** Saat kita mengeluarkan karakter dari stack, kita mendapatkan urutan terbalik dari kalimat yang asli. Misalnya, jika stack berisi "i", "n", "i", maka pengambilan dari stack akan menghasilkan "ini".
- **Membandingkan:** Setelah kalimat asli dan kalimat terbalik diperoleh, program membandingkan keduanya. Jika sama, maka kalimat tersebut adalah palindrom; jika tidak, maka bukan palindrom.

2. Program:


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

```
string reverseStringUsingStack(string kalimat) {
    stack<char> S;

    for (char ch : kalimat) {
        S.push(ch);
    }

    string reversedKalimat = "";
    while (!S.empty()) {
        reversedKalimat += S.top();
        S.pop();
    }

    return reversedKalimat;
}

int main() {
    string kalimat;
    cout << "Masukkan kata (minimal 3 kata): ";
    getline(cin, kalimat);

    string hasil = reverseStringUsingStack(kalimat);

    for (char ch : hasil) {
        cout << ch << " ";
    }
    cout << endl;

    cout << "Hasil : " << hasil << endl;

    return 0;
}
```

Cara kerja:

- **Input Kalimat:** Program akan meminta pengguna memasukkan kalimat dengan minimal 3 kata.
- **Mengisi Stack:** Setiap karakter dari kalimat dimasukkan ke dalam stack satu per satu.
- **Membalikkan Kalimat:** Setelah seluruh karakter berada dalam stack, kita mengeluarkan karakter satu per satu dari stack untuk mendapatkan kalimat yang dibalik.

- **Menampilkan Hasil:** Program menampilkan kalimat asli dan kalimat yang dibalik.