**LAPORAN PRAKTIKUM**

**ALGORITMA DAN STRUKTUR DATA**

**PROGRAM STACK DELIMITER**



**DISUSUN OLEH:**

**Filfimo Yulfiz Ahsanul Hulqi**

**(19102143)**

|  |  |
| --- | --- |
|  |  |

Dikumpulkan Tanggal :

Dosen Pengampu :

Ipam Fuaddina Adam, S.T., M.Kom.

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

### FAKULTAS INFORMATIKA

**INSTITUT TEKNOLOGI TELKOM PURWOKERTO 2020**

**Bab I**

**Penjelasan Unguided**

1. Program tugas stack untuk mencheck suatu inputan mempunyai [] dan () balance atau tidak

#include <iostream>

#include <string>

using namespace std;

class Stack{

private:

char Data[50]; **//Array untuk menampung data**

int MaxCapacity;

int top;

public:

Stack() **//Konstruktor untuk membuat oobjek stack**

{

MaxCapacity=50; **//Kapasitas array yang akan ditampung**

top=-1; **//Kondisi stack kosong**

}

//Kumpulan Fungsi/Aksi/Method

char getTop(); **//Mengambil top stack tanpa membuang**

char pop(); **//Return data dalam stack + buang dari top**

void push(char E); **//Meletakkan data baru di top**

int Empty(); **//Return 1 jika stack kosong**

int CurSize(); **//Return jumlah elemen dalam stack**

int IsFull(); **//Return 1 jika stack ful**l

};

**//Fungsi untuk mengecek stack kosong**

int Stack::Empty()

{

if(top==-1)

return 1;

else

return 0;

}

**//Fungsi untuk megambil data di top stack tanpa membuang**

char Stack::getTop()

{

if(!Empty())

return(Data[top]);

}

**//Fungsi untuk mengecek stack full**

int Stack::IsFull()

{

if(top==MaxCapacity-1)

return 1;

else

return 0;

}

**//Prosedur melakukan push**

void Stack::push(char E)

{

if(!IsFull())

Data[++top]=E;

}

**//Fungsi untuk mereturn jumlah elemen dalam stack**

int Stack::CurSize()

{

return(top+1);

}

int main()

{

Stack s; **//Program untuk mengecek operator**

string input;

string nama = "Filfimo Yulfiz Ahsanul Hulqi";

int nim = 19102143;

cout << "Nama :" << nama << endl;

cout << "NIM :" << nim << endl << endl << endl;

cout << "=====PROGRAM CEK DELIMITER=====" << endl << endl;

cout << "Masukkan string input: ";

cin >> input;

bool isBalance=true; **//Kondisi awal**

int i=0; **//Pencacah**

while (isBalance && i<input.length())

{

if(input[i]=='(') **//Jika program mendapat '(' akan mempush ke stack**

s.push('(');

else if(input[i]=='[') **//Jika program mendapat '[' akan mempush ke stack**

s.push('[');

else if(input[i]==']') **//Kondisi untuk pop**

1. Screenshot Program

{

if(!s.Empty()) //Pop '(' jika stack kosong

s.pop();

else **//Jika knndisi diatas tidak terpenuhi set isBalance menjadi false**

isBalance=false;

}

else if(input[i]==')')**//Kondisi untuk pop**

{

if(!s.Empty()) **//Mempop '(' jika stack kosong**

s.pop();

else **//Jika kondisi diatas tidak terpenuhi set isBalance1 jadi false**

isBalance=false;

}

i++; **//Inkremen i tiap kali literasi**

}

**//Kondisi untuk check imputan yang dimasukkan balance atau tidak**

if(isBalance && s.Empty())

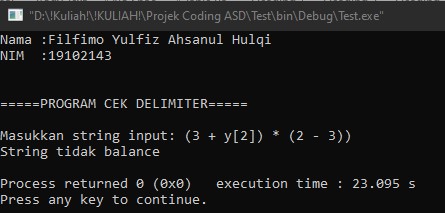
cout << "String balance" << endl;

else

cout << "String tidak balance" << endl;

return 0;

}



**Bab II**

**Kesimpulan**

Dari Praktik, yang telah dilakukan Praktikan dari sini dapat diambil sebuah kesimpulan. Parsing dapat digunakan untu melakukan penentuan persamaan yang diinputkan oleh user pada program, dan program akan melakukan check apakah persamaan tersebut balance atau tidak dengan melakukan pop dan push.