Tugas Week 5 : Stack and Queue

Qornain Aji

21/481767/TK/53170

No.11 Syntax HTML checker.

//Qornain Aji

//21/481767/TK/53170

#include <iostream>

#include <fstream>

#include <stack>

using namespace std;

int main()

{

    ifstream myFile;

    ofstream writeFile;

    char tag;

    stack<string> syntax;

    myFile.open("html.html");

    if (myFile.is\_open())

    {

        while (!myFile.eof())

        {

            while (myFile >> tag)

            {

                string kata="";

                if (tag == '<')

                {

                    while (myFile >> tag)

                    {

                        if (tag == '>')

                        {

                            break;

                        }

                        kata = kata + tag;

                    }

                    if (kata[0] != '/')

                    {

                        syntax.push(kata);

                    }

                    else

                    {

                        if (kata.substr(1).compare(syntax.top()) == 0)

                        {

                            syntax.pop();

//ketika tidak menemukan perbedaan, maka akan di keluarkan karena sudah lengkap tag nya.

                        }

                        else

                        {

                            cout << "Syntax TIDAK lengkap";

                            return 0;

                        }

                    }

                }

            }

        }

    if (syntax.empty())

    {

        cout << "Syntax lengkap";

    }

    else

    {

        cout << "Sytax TIDAK lengkap";

    }

    }

    myFile.close();

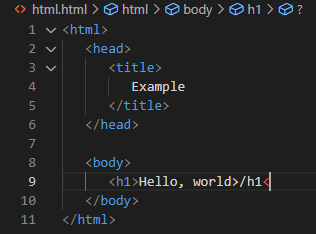
    return 0;

}

Untuk syntax yang benar :



Untuk syntax yang salah :



Kesimpulannya algorithma dengan struktur data stack berhasil dijalankan dengan akurat dan dapat mengenali syntax yang salah ataupun benar.

No. 12 Palindrome Checker

//Qornain Aji

//21/481767/TK/53170

#include <iostream>

#include <queue>

#include <string>

using namespace std;

int main()

{

    queue<char> palindrome1;

    queue<char> palindrome2;

    string kata;

    int kondisi = 1;

    cout<<"Masukkan huruf-huruf yang palindrome atau bukan palindrome : ";

    getline(cin,kata);

    for (int i=0; i<kata.length(); i++)

    {

        if (kata[i]!=' ')

        {

            palindrome1.push(kata[i]);

        }

    }

    for (int i=kata.length()-1; i>=0; i--)

    {

        if (kata[i]!=' ')

        {

            palindrome2.push(kata[i]);

        }

    }

    for (int i=0; i<kata.length(); i++)

    {

        if (palindrome1.front()!=palindrome2.front())

        {

            kondisi = 0;

        }

        palindrome1.pop();

        palindrome2.pop();

    }

    if (kondisi==1)

    {

        cout<<"sebuah PALINDROME";

    }else{

        cout<<"BUKAN PALINDROME";

    }

    return 0;

}

Palindrome:



Bukan Palindrome:



Kesimpulannya adalah algorithma metode struktur data queue berhasil mengenali kalimat yang palindrome walau ada spasi di antaranya.