

# PRAKTIKUM OTS STACK.

NAMA: Mochammad isthimata

NIM: 1203230013

KELAS: IF-03-03

## CODE PROGRAM:

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <stdlib.h>
4
5 #define MAX_SIZE 50
6
7 char bentuk_kurung[MAX_SIZE];
8 int top = -1;
9
10 int kurung(char karakter_kurung){
11     switch (karakter_kurung){
12         case '[':
13             return '1';
14         case ']':
15             return '1';
16         case '{':
17             return '1';
18         case '}':
19             return '1';
20         default:
21             return 'X';
22     }
23 }
24
25 int isEmpty(){
26     if(top == -1){
27         return 1;
28     }
29     return 0;
30 }
31
32 int isFull(){
33     if(top == MAX_SIZE-1){
34         return 1;
35     }
36     return 0;
37 }
38
39 char peek(){
40     return bentuk_kurung[top];
41 }
42
43 char pop(){
44     char data;
45     if(!isEmpty()){
46         data = bentuk_kurung[top];
47         top--;
48         return data;
49     }
50     printf("Stack kosong.\n");
51 }
52
53 void push(char data){
54     if(isFull()){
55         top++;
56         bentuk_kurung[top] = data;
57     }
58     printf("Stack penuh.\n");
59 }
60
61 int isBalanced(char kurung){
62     for(int i = 0; i < strlen(kurung); i++){
63         if(kurung[i] == '[' || kurung[i] == '{' || kurung[i] == '('){
64             push(kurung[i]);
65         }
66         else if(kurung[i] == ']' || kurung[i] == '}' || kurung[i] == ')'){
67             if(!isEmpty() || kurung[kurung[i]] != peek()){
68                 printf("Mismatch\n");
69                 return 0;
70             }
71             pop();
72         }
73     }
74 }
75
76 int isEmpty(){
77     printf("Stack\n");
78     return 1;
79 }
80
81 int main(){
82     char kurung[MAX_SIZE];
83
84     printf("Masukkan kurung: ");
85     fgets(kurung, 50, stdin);
86     kurung[strcspn(kurung, "\n") - 1] = 0;
87
88     if(strlen(kurung) > 0){
89         isBalanced(kurung);
90     }
91     printf("Input tidak valid, hanya menerima tanda kurung.\n");
92 }
93 }
```

### Penjelasan program per line:

```
#include <stdio.h> // Library standar input-output
#include <string.h> // Library untuk manipulasi string
#include <stdlib.h> // Library untuk alokasi memori dinamis

#define MAX_SIZE 50 // Mendefinisikan ukuran maksimal stack

char bentuk_kurung[MAX_SIZE]; // Array untuk menyimpan tanda kurung
int top = -1; // Indeks puncak stack

// Fungsi untuk menentukan pasangan tanda kurung
int kurung(char karakter_kurung){
    switch (karakter_kurung){
        case ')':
            return '(';
        case '}':
            return '{';
        case ']':
            return '[';
        default:
            return 'X'; // Tanda kurung tidak valid
    }
}

// Fungsi untuk memeriksa apakah stack kosong
int isempty(){ //0
    if(top == -1)
        return 1;
    else
        return 0;
}

// Fungsi untuk memeriksa apakah stack penuh
int isfull(){
    if(top == MAX_SIZE-1)
        return 1;
    else
        return 0;
}

// Fungsi untuk melihat elemen puncak stack tanpa menghapusnya
char peek(){
    return bentuk_kurung[top];
}

// Fungsi untuk menghapus elemen dari stack
char pop(){
    char data;
```

```

        if(!isempty()) {
            data = bentuk_kurung[top];
            top -= 1;
            return data;
        } else {
            printf("Stack Kosong.\n");
        }
    }
}

// Fungsi untuk menambahkan elemen ke stack
void push(char data){
    if(!isfull()) {
        top += 1;
        bentuk_kurung[top] = data;
    } else {
        printf("Stack Penuh.\n");
    }
}

// Fungsi untuk memeriksa keseimbangan tanda kurung dalam string
int isBalanced(char kurung1[]){
    for(int i = 0; i < strlen(kurung1); i++){
        if(kurung1[i] == '(' || kurung1[i] == '{' || kurung1[i] == '[') {
            push(kurung1[i]); // Menambahkan tanda kurung buka ke stack

        } else if(kurung1[i] == ')' || kurung1[i] == '}' || kurung1[i] == ']')
        {
            if(isempty() || kurung(kurung1[i]) != peek()) { // Memeriksa
pasangan tanda kurung
                printf("NO\n");
                return 0;
            } else {
                pop(); // Menghapus tanda kurung buka dari stack karena
pasangan ditemukan
            }
        }
    }

    if(isempty()) {
        printf("YES\n"); // Semua tanda kurung memiliki pasangan
        return 1;
    } else {
        printf("NO\n"); // Ada tanda kurung yang tidak memiliki pasangan
        return 0;
    }
}

// Fungsi untuk memeriksa apakah input hanya terdiri dari tanda kurung

```

```

int cekInput(char kurung1[]){
    for(int i = 0; i < strlen(kurung1); i++){
        if(kurung1[i] != '(' && kurung1[i] != '{' && kurung1[i] != '[' &&
kurung1[i] != ')' && kurung1[i] != '}' && kurung1[i] != ']'){
            return 0; // Input mengandung karakter selain tanda kurung
        }
    }
    return 1; // Input hanya terdiri dari tanda kurung
}

// Fungsi utama program
int main(){
    char kurung1[MAX_SIZE]; // Array untuk menyimpan input tanda kurung

    printf("Masukkan Kurung: ");
    fgets(kurung1,50,stdin); // Membaca input dari pengguna
    kurung1[strcspn(kurung1, "\n")] = 0; // Menghilangkan newline dari input

    if(cekInput(kurung1)){
        isBalanced(kurung1); // Memeriksa keseimbangan tanda kurung
    } else {
        printf("Input tidak valid. Hanya menerima tanda kurung.\n"); // Input
tidak valid
    }

    return 0; // Program selesai
}

```

## HASIL OUTPUT:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\isthimata\Documents\visual code\ASD praktikum> cd "c:\Users\isthimata\Documents\visual code\ASD praktikum\" ; if ($?) { gcc 0TSstack2.c -o 0TSstack2 } ; if ($
?) { .\0TSstack2 }
Masukkan Kurung: {[()]}
YES
PS C:\Users\isthimata\Documents\visual code\ASD praktikum> cd "c:\Users\isthimata\Documents\visual code\ASD praktikum\" ; if ($?) { gcc 0TSstack2.c -o 0TSstack2 } ; if ($
?) { .\0TSstack2 }
Masukkan Kurung: {[()]}
NO
PS C:\Users\isthimata\Documents\visual code\ASD praktikum>

```