

## TUGAS OTS (STACK)

NAMA : AWRA SANK RAMA

NIM : 1203230014

KELAS : IF.03.01

### INPUT

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX 100

// Struktur data untuk stack
typedef struct {
    char data[MAX];
    int top;
} Stack;

// Inisialisasi stack
void initStack(Stack *s) {
    s->top = -1;
}

// Memeriksa apakah stack kosong
int isEmpty(Stack *s) {
    return (s->top == -1);
}

// Memeriksa apakah stack penuh
int isFull(Stack *s) {
    return (s->top == MAX - 1);
}

// Memasukkan elemen ke dalam stack
void push(Stack *s, char c) {
    if (!isFull(s)) {
        s->data[++(s->top)] = c;
    }
}

// Menghapus elemen dari stack
char pop(Stack *s) {
    if (!isEmpty(s)) {
        return s->data[(s->top)--];
    }
}
```

```

    return '\0';
}

// Fungsi untuk memeriksa keselarasan tanda kurung
char* isBalanced(char *str) {
    Stack s;
    initStack(&s);
    int i;

    for (i = 0; str[i] != '\0'; i++) {
        if (str[i] == '(' || str[i] == '[' || str[i] == '{') {
            push(&s, str[i]);
        } else if (str[i] == ')' || str[i] == ']' || str[i] == '}') {
            if (isEmpty(&s)) {
                return "NO"; // Kurung penutup tanpa kurung pembuka
            }
            char opening = pop(&s);
            if ((opening == '(' && str[i] != ')') || (opening == '[' && str[i] !=
                ']') || (opening == '{' && str[i] != '}')) {
                return "NO"; // Tidak ada pasangan yang sesuai
            }
        }
    }

    if (isEmpty(&s)) {
        return "YES"; // Semua pasangan tanda kurung seimbang
    } else {
        return "NO"; // Kurung pembuka tanpa kurung penutup
    }
}

int main() {
    char expression[MAX];
    fgets(expression, sizeof(expression), stdin); // Membaca input string

    // Menghapus karakter newline jika ada
    if (expression[strlen(expression) - 1] == '\n') {
        expression[strlen(expression) - 1] = '\0';
    }

    // Memeriksa keselarasan tanda kurung dan mencetak hasilnya
    printf("%s\n", isBalanced(expression));

    return 0;
}

```

OUTPUT :

```
PS D:\CODING> cd "d:\CODING\" ; if ($?) { gcc praktikumlg.c -o praktikumlg } ; if ($?) { .\praktikumlg }  
{[()]}  
YES  
PS D:\CODING> cd "d:\CODING\" ; if ($?) { gcc praktikumlg.c -o praktikumlg } ; if ($?) { .\praktikumlg }  
{[()]}  
NO  
PS D:\CODING> cd "d:\CODING\" ; if ($?) { gcc praktikumlg.c -o praktikumlg } ; if ($?) { .\praktikumlg }  
{{[[((()]]}}  
YES  
PS D:\CODING> cd "d:\CODING\" ; if ($?) { gcc praktikumlg.c -o praktikumlg } ; if ($?) { .\praktikumlg }  
{()[]}()  
YES  
PS D:\CODING> █
```