

Agnes Destiny Sava S

IF-03-01 / 1203230092

OTS STACK ASD

Source Code:

```
#include <stdio.h> // Header file yang berfungsi untuk input-output
#include <stdbool.h> // Header file yang berfungsi untuk mendapatkan tipe data
bool dan nilai konstan 'True' or 'False'

#define MAX_SIZE 100 // Definisi yang menjelaskan berapa ukuran maksimum dari
stack atau tumpukan

// Definisi struktur stack dengan menggunakan struct
typedef struct { // Berfungsi untuk mendeklarasi tipe data
    char data[MAX_SIZE]; // Berfungsi untuk menyimpan elemen-elemen tumpukan
    int top; // Berfungsi untuk menyimpan indeks dari elemen teratas tumpukan
} Stack;

// Fungsi untuk menginisialisasi stack
void initialize(Stack *stack) { // Sebagai pointer ke tipe data stack yang
bertujuan untuk menginisialisasi atau menyiapkan sebuah tumpukan (stack)
tersebut untuk digunakan
    stack->top = -1; // Menandakan bahwa tumpukan paling atas kosong
}

// Fungsi untuk menambahkan elemen ke dalam stack
void push(Stack *stack, char item) { // Sebagai pointer yang digunakan untuk
menambahkan elemen kedalam stack
    stack->data[++stack->top] = item; // Digunakan untuk memasukkan atau
menyimpan item yang akan dimasukkan kedalam tumpukan
}

// Fungsi untuk menghapus elemen dari stack
char pop(Stack *stack) { // Digunakan untuk menghapus dan mengembalikan elemen
teratas dari stack
    return stack->data[stack->top--]; // 'stack->top--' adalah indeks yang
menunjukkan posisi teratas dari stack, menunjukkan elemen teratas yang akan
dihapus sehingga menunjukkan elemen teratas terbaru setelah penghapusan.
Karena nilai tersebut diambil dari tumpukan yang telah dikurangi, maka nilai
tersebut tidak lagi menjadi tumpukan
}

// Fungsi untuk mengecek apakah tanda kurung berpasangan
bool isBalanced(const char *expression) { // Definisi fungsi 'isBalanced',
menerima variabel kurung yang merupakan string dan akan diperiksa apakah
```

urutan dari kurung tersebut seimbang atau tidak. Tipe data bool ini adalah kembalian yang dimana akan mengindikasikan apakah string tersebut seimbang atau tidak

```
Stack stack; // Variable ini adalah sebuah tumpukan yang digunakan untuk menyimpan kurung yang ditemukan pada string
```

```
initialize(&stack); // Untuk menginisialisasi tumpukan sebelum digunakan, fungsi ini akan mengatur indeks teratas menjadi -1 yang menandakan bahwa tumpukan kosong
```

```
for (int i = 0; expression[i] != '\0'; i++) { // Loop ini akan berlangsung selama karakter yang sedang diperiksa bukanlah karakter null terminator yang menandakan akhir dalam string
```

```
    if (expression[i] == '(' || expression[i] == '[' || expression[i] == '{') { // Digunakan untuk memeriksa jenis karakter saat ini yaitu salah satu dari tiga jenis kurung buka. Jika iya, maka karakter tersebut akan dimasukkan kedalam tumpukan menggunakan fungsi 'push'
```

```
        push(&stack, expression[i]); // Digunakan untuk memanggil fungsi 'push' untuk memasukkan karakter kurung buka kedalam tumpukan
```

```
    } else if (expression[i] == ')' || expression[i] == ']' || expression[i] == '}') { // Memeriksa apakah karakter saat ini adalah salah satu dari tiga jenis kurung tutup, jika iya akan dilakukan pengecekan apakah tumpukan tersebut kosong. Karena harus ada kurung buka yang berpasangan untuk kurung tutup tersebut
```

```
        if (stack.top == -1) { // Digunakan untuk memeriksa apakah tumpukan tersebut kosong, jika iya maka tidak ada kurung buka yang berpasangan dengan kurung tutup. Sehingga urutan kurung menjadi tidak seimbang
```

```
            return false; // Maka akan mengembalikan nilai else dalam if di fungsi main atau akan langsung print 'NO'
```

```
        }
```

```
        char current = pop(&stack); // Digunakan untuk memanggil fungsi 'pop' yang dapat menghapus dan mengambil karakter teratas dari tumpukan. Karakter ini akan menjadi kurung buka yang berpasangan dengan kurung tutup yang sedang di proses
```

```
        if ((expression[i] == ')' && current != '(') || // Jika variabel kurung memiliki inisialisasi nilai = karakter ')' dan pasangan != '('  
            (expression[i] == ']' && current != '[') || // Atau variabel kurung memiliki inisialisasi nilai = karakter ']' dan pasangan != '['  
            (expression[i] == '}' && current != '{')) { // Dan variabel kurung memiliki inisialisasi nilai = karakter '}' dan pasangan != '{'
```

```
            return false; // Maka akan mengembalikan nilai else dalam if di fungsi main atau akan langsung print 'NO'
```

```
        }
```

```
    }
```

```
}
```

```
return stack.top == -1; // Fungsi tersebut akan terus berjalan hingga nilai teratas dalam stack habis
```

```
}
```

```

int main() { // Merupakan deklarasi fungsi 'main' yang digunakan untuk titik
masuk utama program
    char expression[100]; // Deklarasi dari variabel yang merupakan array
karakter dengan ukuran 100
    printf("Masukkan urutan tanda kurung: "); // Digunakan untuk mencetak
'Masukkan urutan tanda kurung'
    scanf("%s", expression); // Digunakan untuk membaca inputan dan menyimpan
inputan tersebut kedalam variabel 'expression'

    if (isBalanced(expression)) { // Memanggil fungsi 'isBalanced' untuk
memeriksa apakah urutan tanda kurung yang dimasukkan seimbang atau tidak. Jika
iya, maka urutan seimbang. Maka blok kode dalam 'if' akan dijalankan
        printf("YES\n"); // Jika tanda kurung yang dimasukkan seimbang maka
program akan mencetak 'YES'
    } else { // Jika hasil dari 'isBalanced' adalah false artinya urutan kurung
tidak seimbang, maka blok kode dalam else akan di jalankan
        printf("NO\n"); // Jika tanda kurung yang dimasukkan kurang seimbang
atau artinya urutan tanda kurung tersebut kurang tepat maka program akan
mencetak 'NO'
    }

    return 0; // Memberi nilai kembali pada sistem operasi atau menandakan
program selesai dijalankan
}

```

Output:

```

PS C:\AlPro> cd "c:\AlPro\ASD\" ; if ($?) { gcc soalpraktikum.c -o soalpraktikum } ; if ($?) { .\soalpraktikum }
Masukkan urutan tanda kurung: {[()]}
YES
PS C:\AlPro\ASD> cd "c:\AlPro\ASD\" ; if ($?) { gcc soalpraktikum.c -o soalpraktikum } ; if ($?) { .\soalpraktikum }
Masukkan urutan tanda kurung: {[()]}
NO
PS C:\AlPro\ASD> cd "c:\AlPro\ASD\" ; if ($?) { gcc soalpraktikum.c -o soalpraktikum } ; if ($?) { .\soalpraktikum }
Masukkan urutan tanda kurung: {[[(())]]}
YES
PS C:\AlPro\ASD> cd "c:\AlPro\ASD\" ; if ($?) { gcc soalpraktikum.c -o soalpraktikum } ; if ($?) { .\soalpraktikum }
Masukkan urutan tanda kurung: {(())[ ]}()
YES
PS C:\AlPro\ASD> █

```