

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

#define MAX_SIZE 100

// Struktur tumpukan
typedef struct {
    char items[MAX_SIZE];
    int top;
} Stack;

// Fungsi untuk inisialisasi tumpukan
void initialize(Stack *s) {
    s->top = -1;
}

// Fungsi untuk menambahkan elemen ke tumpukan
void push(Stack *s, char value) {
    if (s->top == MAX_SIZE - 1) {
        printf("Stack Overflow\n");
        exit(EXIT_FAILURE);
    }
    s->items[++(s->top)] = value;
}

// Fungsi untuk menghapus elemen dari tumpukan
char pop(Stack *s) {
    if (s->top == -1) {
        printf("Stack Underflow\n");
        exit(EXIT_FAILURE);
    }
    return s->items[(s->top)--];
}

// Fungsi untuk memeriksa apakah tumpukan kosong
int isEmpty(Stack *s) {
```

```

    return s->top == -1;
}

// Fungsi untuk memeriksa keseimbangan tanda kurung
char* isBalanced(char *s) {
    Stack stack;
    initialize(&stack);

    int i;
    for (i = 0; s[i] != '\0'; i++) {
        if (s[i] == '(' || s[i] == '[' || s[i] == '{') {
            push(&stack, s[i]);
        } else if (s[i] == ')' || s[i] == ']' || s[i] == '}') {
            if (isEmpty(&stack)) {
                return "NO";
            } else {
                char popped = pop(&stack);
                if ((s[i] == ')' && popped != '(') ||
                    (s[i] == ']' && popped != '[') ||
                    (s[i] == '}' && popped != '{')) {
                    return "NO";
                }
            }
        }
    }

    if (isEmpty(&stack)) {
        return "YES";
    } else {
        return "NO";
    }
}

int main() {
    char input[MAX_SIZE];
    printf("Masukkan urutan tanda kurung: ");
    fgets(input, MAX_SIZE, stdin);

```

```
input[strcspn(input, "\n")] = '\0'; // Menghapus karakter newline dari input

printf("Tanda kurung seimbang? %s\n", isBalanced(input));

return 0;
}
```