

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

#define MAX_STACK_SIZE 20 //maximum jumlah stack

//Mendefinisikan struktur stack
typedef struct {
    char data [MAX_STACK_SIZE];
    int top;
}stack;

//Menginisialisasi stack kosong
void createEmptyStack(stack *stack){
    stack->top = -1;
}

bool isEmpty(stack *stack){// memeriksa apakah stack
kosong
    return stack->top == -1;
}

bool isFull(stack *stack){ //untuk memeriksa apakah
stack penuh
    return stack->top == MAX_STACK_SIZE - 1;
}

char peek(stack *stack){ //untuk melihat elemen
teratas dari stack
    if (isEmpty(stack))
    {
        printf("stack is empty\n");
        return 0;
    }
}
```

```

        return stack->data[stack->top];
    }

void push(stack *stack, char data){ //untuk
menambahkan elemen ke dalam stack
    if (isFull(stack))
    {
        printf("stack penuh\n");
        return;
    }
    stack->top++;
    stack->data[stack->top] = data;
}

int pop(stack *stack){ //untuk menghapus elemen
teratas
    if (isEmpty(stack))
    {
        printf("stack kosong\n");
        return 0;
    }
    char data = stack->data[stack->top];
    stack->top--;
    return data;
}

char isBalanced(char *str){ //cek tanda kurung
berpasangan
    stack stack;
    createEmptyStack(&stack);

    for (int i = 0; str[i] != '\0'; i++) {
        char ch = str[i];
        if (ch == '(' || ch == '[' || ch == '{') {

```

```

        push(&stack, ch);
    } else if (ch == ')' || ch == ']' || ch ==
'}') {
        if (isEmpty(&stack)) {
            return 'N'; // Kurung penutup tanpa
pasangan
        }
        char openingBracket = peek(&stack);
        if ((ch == ')' && openingBracket == '(')
||
            (ch == ']' && openingBracket == '[')
||
            (ch == '}' && openingBracket == '{'))
        {
            pop(&stack); // Kurung pasangan, pop
dari stack
        } else {
            return 'N'; // Kurung penutup tidak
sesuai
        }
    }

    if (isEmpty(&stack)) {
        return 'Y'; // Tanda kurung berpasangan
    } else {
        return 'N'; // Tanda kurung tidak
berpasangan, masih ada yang tersisa di stack
    }
}

void printStack(stack *stack){ //tidak terpanggil
hanya coba"
    if (isEmpty(stack))

```

```

    {
        printf("stack kosong\n");
        return;
    }
    printf("stack: ");
    for (int i = stack->top; i >= 0; i--)
    {
        printf("%d ", stack->data[i]);
    }
}

int main(){ //fungsi main
    char str[MAX_STACK_SIZE];

    printf("Input string tanda kurung: "); //input
tanda kurung
    scanf("%s", str);

    char hasil = isBalanced(str); //panggil
isBalanced

    if (hasil == 'Y') { // jika Tanda kurung
berpasangan
        printf("YES\n");
    } else { // jika Tanda kurung tidak
berpasangan, masih ada yang tersisa di stack
        printf("NO\n");
    }

    return 0;
}

```

OUTPUT :

```
Masukkan string tanda kurung: {}[]()
```

YES

```
PS C:\Users\MSI GAMING\Documents\TUGAS IMAM PRASETYO\ASD\stack> cd "c:\U
```

```
Masukkan string tanda kurung: [)]}({][[]
```

NO

```
PS C:\Users\MSI GAMING\Documents\TUGAS IMAM PRASETYO\ASD\stack> cd "c:\U
```

```
Input string tanda kurung: }{ }{ }{ }{ }{ }{ }{ }{ }{ }{ }{ }{ }{ }
```

NO

```
PS C:\Users\MSI GAMING\Documents\TUGAS IMAM PRASETYO\ASD\stack> cd "c:\U
```

```
Input string tanda kurung: {}[]()
```

YES