

Nama: Arina Qurrata Aini

NIM: 1203230053

Kelas: If-03-01

1. Code Program

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

typedef struct stack {
    char data;
    struct stack* next;
} Stack;

void push(Stack** top, char data) {
    Stack* new_node = (Stack*) malloc(sizeof(Stack));
    if (!new_node) return;
    new_node->data = data;
    new_node->next = (*top);
    (*top) = new_node;
}

int isEmpty(Stack* top) {
    return top == NULL;
}

char pop(Stack** top) {
    char popped;
    Stack* temp;

    if (isEmpty(*top))
        return '\0';
    else {
        temp = *top;
        popped = temp->data;
        *top = temp->next;
        free(temp);
        return popped;
    }
}

char peek(Stack* top) {
    if (!isEmpty(top))
```

```

        return top->data;
    else
        return '\0';
}

int isMatchingPair(char character1, char character2) {
    if (character1 == '(' && character2 == ')')
        return 1;
    else if (character1 == '{' && character2 == '}')
        return 1;
    else if (character1 == '[' && character2 == ']')
        return 1;
    else
        return 0;
}

int isBalanced(char exp[]) {
    int i = 0;
    Stack* stack = NULL;

    while (exp[i]) {
        if (exp[i] == '{' || exp[i] == '(' || exp[i] == '[')
            push(&stack, exp[i]);
        if (exp[i] == '}' || exp[i] == ')' || exp[i] == ']') {
            if (stack == NULL)
                return 0;
            else if (!isMatchingPair(pop(&stack), exp[i]))
                return 0;
        }
        i++;
    }

    if (stack == NULL)
        return 1;
    else
        return 0;
}

int main() {
    char exp[100];
    printf("Masukkan urutan tanda kurung: ");
    scanf("%s", exp);

    if (isBalanced(exp))
        printf("YES\n");
}

```

```

else
    printf("NO\n");

return 0;
}

```

2. Hasil output:

```

PS D:\Kuliah\Smester 2\Tugas Kuliah\Algoritma Dan Struktur Data\Code C> & 'c:\Users\MSI GAMING\.vscode\extensions\ms-vscode.cpptools-1.19.9\bin\DebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-e0obn5p0.0d2' '--stdout=Microsoft-MIEngine-Out-iqm1mmyw.nph' '--stderr=Microsoft-MIEngine-Out-iqm1mmyw.nph' --dbgExe=C:\CodeBlocks\MinGW\bin\gdb.exe --interpreter=mi'
Masukkan urutan tanda kurung: []{}()
YES
PS D:\Kuliah\Smester 2\Tugas Kuliah\Algoritma Dan Struktur Data\Code C> & 'c:\Users\MSI GAMING\.vscode\extensions\ms-vscode.cpptools-1.19.9\bin\DebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-qbcwxiwr.tp4' '--stdout=Microsoft-MIEngine-Out-nnxmohul.krb' '--stderr=Microsoft-MIEngine-Out-nnxmohul.krb' --dbgExe=C:\CodeBlocks\MinGW\bin\gdb.exe --interpreter=mi'
Masukkan urutan tanda kurung: {{{}}}{]()
NO
PS D:\Kuliah\Smester 2\Tugas Kuliah\Algoritma Dan Struktur Data\Code C> & 'c:\Users\MSI GAMING\.vscode\extensions\ms-vscode.cpptools-1.19.9\bin\DebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-chuyjwvc.1lo' '--stdout=Microsoft-MIEngine-Out-4hxx1z31.z3i' '--stderr=Microsoft-MIEngine-Out-4hxx1z31.z3i' --dbgExe=C:\CodeBlocks\MinGW\bin\gdb.exe --interpreter=mi'
Masukkan urutan tanda kurung: {{{}}}{]()
YES

```

3. Cara kerja program:

1. **Definisi Struktur Stack:** Kode ini mendefinisikan struktur stack yang terdiri dari data (yang berisi karakter tanda kurung) dan pointer ke node stack berikutnya.
2. **Fungsi Push:** Fungsi ini digunakan untuk menambahkan elemen ke puncak stack. Fungsi ini membuat node baru, mengatur datanya menjadi karakter yang diberikan, dan menempatkannya di puncak stack.
3. **Fungsi isEmpty:** Fungsi ini memeriksa apakah stack kosong atau tidak. Jika puncak stack adalah NULL, maka stack dianggap kosong.
4. **Fungsi Pop:** Fungsi ini digunakan untuk menghapus elemen dari puncak stack. Fungsi ini menghapus node di puncak stack dan mengembalikan datanya.
5. **Fungsi Peek:** Fungsi ini digunakan untuk melihat elemen di puncak stack tanpa menghapusnya.
6. **Fungsi isMatchingPair:** Fungsi ini memeriksa apakah dua karakter tanda kurung membentuk pasangan atau tidak.
7. **Fungsi isBalanced:** Fungsi ini adalah fungsi utama yang memeriksa apakah tanda kurung dalam string berpasangan atau tidak. Fungsi ini melakukan iterasi melalui setiap karakter dalam string. Jika karakter adalah tanda kurung pembuka, fungsi ini menambahkannya ke stack. Jika karakter adalah tanda kurung penutup, fungsi ini memeriksa apakah tanda kurung tersebut berpasangan dengan tanda kurung di puncak stack. Jika ya, fungsi ini menghapus tanda kurung dari stack. Jika tidak, fungsi ini mengembalikan 0, yang berarti string tidak seimbang. Jika semua tanda kurung dalam

string telah diperiksa dan stack kosong, fungsi ini mengembalikan 1, yang berarti string seimbang.

8. **Fungsi Main:** Fungsi ini meminta pengguna untuk memasukkan string tanda kurung, memeriksa apakah string tersebut seimbang dengan fungsi `isBalanced`, dan mencetak hasilnya. Jika string seimbang, fungsi ini mencetak "YES". Jika tidak, fungsi ini mencetak "NO".