

Nama : Josefania Tirsa Putri Immauely
Kelas : IF 03-02
NIM : 1203230012

Soal Latihan Stack Struct

1. Asisten Sherlock Holmes a) Source code

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

struct Node {
    char* alphabet; //berfungsi menyimpan huruf pada batu
    struct Node* link; //pointer ke node berikutnya
};

int main() {
    // Deklarasi node-node
    struct Node l1, l2, l3, l4, l5, l6, l7, l8, l9; //membuat sembilan
    objek node baru yang dapat digunakan dalam program
    struct Node *link, *l3ptr; //untuk menunjuk ke node-node yang dibuat
    sebelumnya atau mungkin node lainnya yang akan dibuat atau diakses dalam
    program

    // Inisialisasi node-node dengan menggunakan potongan kode soal
    l1.link = NULL;
    l1.alphabet = "F";

    l2.link = NULL;
    l2.alphabet = "M";

    l3.link = NULL;
    l3.alphabet = "A";

    l4.link = NULL;
    l4.alphabet = "I";

    l5.link = NULL;
    l5.alphabet = "K";

    l6.link = NULL;
    l6.alphabet = "T";

    l7.link = NULL;
```

```

17.alphabet = "N";

18.link = NULL;
18.alphabet = "O";

19.link = NULL;
19.alphabet = "R";

// Mengatur koneksi antar node sesuai dengan urutan yang diinginkan
17.link = &l1;// Menyambungkan ke l1
11.link = &l8;// Menyambungkan ke l1
18.link = &l2;// Menyambungkan ke l1
12.link = &l5;// Menyambungkan ke l1
15.link = &l3;// Menyambungkan ke l1
13.link = &l6;// Menyambungkan ke l1
16.link = &l9;
19.link = &l4;
14.link = &l7;

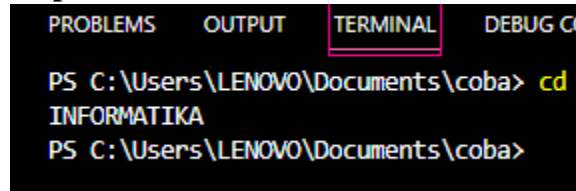
// Starting point
l3ptr = &l7;

// Akses data menggunakan printf
printf("%s", l3.link->link->link->alphabet);// Menampilkan huruf I
printf("%s", l3.link->link->link->link->alphabet);// Menampilkan huruf
N
printf("%s", l3.link->link->link->link->link->alphabet);// Menampilkan
huruf F
printf("%s", l3.link->link->link->link->link->link->alphabet);//
Menampilkan huruf O
printf("%s", l3.link->link->alphabet);// Menampilkan huruf R
printf("%s", l3.link->link->link->link->link->link->link->alphabet);//
Menampilkan huruf M
printf("%s", l3.alphabet);// Menampilkan huruf A
printf("%s", l3.link->alphabet);// Menampilkan huruf T
printf("%s", l3.link->link->link->alphabet);// Menampilkan huruf I
printf("%s", l3.link->link->link->link->link->link->link->link-
>alphabet);// Menampilkan huruf K
printf("%s", l3.alphabet);// Menampilkan huruf A

return 0;
}

```

b) Ouput



```
PROBLEMS OUTPUT TERMINAL DEBUG CO
PS C:\Users\LENOVO\Documents\coba> cd
INFORMATIKA
PS C:\Users\LENOVO\Documents\coba>
```

2. Game of Two Stacks

a) Source Code

```
#include <stdio.h>

int twoStacks(int maxSum, int a[], int n, int b[], int m) { //fungsi ini
mengembalikan sebuah integer.
    int sum = 0, count = 0, temp = 0, i = 0, j = 0; //inisialisasi
variabel lokal dengan 0

    while (i < n && sum + a[i] <= maxSum) { // menambahkan elemen dari
stack pertama ke 'sum' sampai jumlah 'sum' melebihi atau sama dengan
'maxSum' atau sampai semua elemenn dalam stack pertama habis
        sum += a[i++]; //elemen dari stack pertama yang memenuhi syarat
ditambahkan ke 'sum', dan indeks i ditingkatkan setelah penambahan elemen
ke 'sum'
    }
    count = i; //untuk menyimpan jumlah elemen yang telah ditambahkan ke
'sum'

    while (j < m && i >= 0) { //loop while yang berjalan selama indeks 'j'
masih kurang dari jumlah elemen dalam stack kedua '(m)' dan indeks 'i'
tidak negatif
        sum += b[j++]; //elemen-elemen dari stack kedua '(b)' ditambahkan
ke dalam 'sum' dan indeks 'j' ditingkatkan setelah nilai 'b[j]'
ditambahkan ke 'sum'
        while (sum > maxSum && i > 0) { //penambahan elemen dari stack
kedua menyebabkan 'sum' melebihi 'maxSum', maka loop 'while' bersarang
akan dijalankan
            sum -= a[--i]; //elemen-elemen dari stack pertama '(a)' akan
dihapus dari 'sum' sampai 'sum' menjadi kurang dari atau sama dengan
'maxSum'
        }
        if (sum <= maxSum && i + j > count) { //mengecek apakah 'sum'
tidak melebihi 'maxSum'
            count = i + j; //jumlah elemen yang telah dipilih dari kedua
stack '(i + j)' dibandingkan dengan 'count', dan jika jumlah ini lebih
besar dari 'count', maka 'count' diperbarui
        }
    }
}
```

```

    }
}
return count; //nilai 'count' dikembalikan sebagai hasil dari fungsi
'twoStacks'
}

int main() {
    int g; //menyimpan jumlah kasus uji yang akan dievaluasi
    scanf("%d", &g); //nilai 'g' dibaca dari input standar menggunakan
    fungsi 'scanf'
    while (g--) { //loop ini akan dijalankan sebanyak 'g' kali, yang
    menandakan jumlah kasus uji yang akan dievaluasi dan setiap kali iterasi
    'g' akan dikurangi satu
        int n, m, maxSum; //masing-masing akan menyimpan jumlah elemen
        dalam stack 'n', 'm', 'maxSum'
        scanf("%d%d%d", &n, &m, &maxSum); //nilai dari setiap elemen
        sebelumnya akan dicetak
        int a[n], b[m]; //mendefinisikan array 'a' dan 'b' dengan ukuran
        yang sesuai, yaitu 'n' dan 'm', untuk menyimpan elemen-elemen dari kedua
        stack
        for (int i = 0; i < n; i++) { //membaca elemen-elemen dari stack
        pertama
            scanf("%d", &a[i]); //membaca loop for atas dengan dicetak
            dibaris ini
        }
        for (int i = 0; i < m; i++) { //membaca elemen-elemen dari stack
        kedua
            scanf("%d", &b[i]); //membaca loop for atas dengan dicetak
            dibaris ini
        }
        printf("%d\n", twoStacks(maxSum, a, n, b, m)); //memanggil fungsi
        'twoStacks' dengan argumen yang sesuai
    }
    return 0;
}

```

b) Ouput

```

PS C:\Users\LENOVO\Documents\coba>
1
5 4 11
4 5 2 1 1
3 1 1 2
5
PS C:\Users\LENOVO\Documents\coba>

```