

# Pembukaan

▼ Mata Kuliah	Struktur Data
▼ Kelas	C
▼ Tahun Ajaran	2021/2022
▼ Praktikum Modul	1

Oleh: Luthfiyyah Hanifah Amari (5025201090)

## Description

Selamat datang di kelas struktur data.

Dalam rangka penyambutan, pak imron menulis buku pemrograman. Buku itu telah dicetak sejumlah n buah. Sebelum buku-buku itu dijual, buku perlu ditandatangani oleh pak imron, kemudian disampuli plastik.

Dalam proses tersebut, pak imron dan timnya ingin menaruh buku-buku di 3 tumpukan buku.

tumpukan 1 adalah tumpukan buku yang baru dicetak.

tumpukan 2 adalah tumpukan buku yang sudah ditandatangani.

tumpukan 3 adalah tumpukan buku yang sudah disampul.

setiap buku mempunyai kode, yaitu sesuai dengan urutan cetaknya (dimulai dari 1).

Pada soal ini, kamu diberikan operasi query, yang dapat berupa CETAK, TTD, SAMPUL.

CETAK -> ada x buku yang baru tercetak. letakkan di tumpukan 1

TTD -> y buku diambil dari tumpukan 1, lalu ditandatangani oleh pak imron. lalu, letakkan di tumpukan 2

SAMPUL -> z buku diambil dari tumpukan 2, lalu disampul. lalu, letakkan di tumpukan 3

karena ini tumpukan, maka buku yang diambil ialah buku paling atas. setelah dilakukan seluruh operasi, Cetaklah seluruh kode buku dalam setiap tumpukan.

## Input Format

- baris 1 adalah Q, yakni jumlah Query
- Q baris selanjutnya ialah query yang dapat berupa
  - `CETAK x`, di mana x ialah jumlah buku yang dicetak
  - `TTD y`, di mana y ialah jumlah buku yang ditandatangani
  - `SAMPUL z`, di mana z ialah jumlah buku yang disampul

## Constraints

- $1 \leq N \leq 20$
- $1 \leq X, Y, Z \leq 20$

## Output Format

cetak dengan format

tumpukan 1:

tumpukan 2:

tumpukan 3:

setelah ":" diisi dengan kode-kode buku yang ada di masing-masing tumpukan

## Sample Input 0

```
5
CETAK 5
TTD 3
TTD 2
```

CETAK 3  
SAMPUL 3

## Sample Output 0

tumpukan 1: 6 7 8  
tumpukan 2: 5 4  
tumpukan 3: 1 2 3

## Explanation 0

CETAK 5

5 buku dicetak. masing-masing buku auto mempunyai kode buku, yakni 1 2 3 4 5  
maka isi dari

tumpukan 1: 1 2 3 4 5

tumpukan 2:

tumpukan 3:

TTD 3

3 buku diambil dari tumpukan 1 (ambil dari atas). dipindahkan ke tumpukan 2

tumpukan 1: 1 2

tumpukan 2: 5 4 3

tumpukan 3:

TTD 2

2 buku diambil dari tumpukan 1 (ambil dari atas). dipindahkan ke tumpukan 2

tumpukan 1:

tumpukan 2: 5 4 3 2 1

tumpukan 3:

CETAK 3

3 buku dicetak. masing-masing auto mempunyai kode buku, yakni 6 7 8. (melanjutkan  
kode buku yang dicetak sebelumnya)

tumpukan 1: 6 7 8

tumpukan 2: 5 4 3 2 1

tumpukan 3:

**SAMPUL 3**

tumpukan 1: 6 7 8

tumpukan 2: 5 4

tumpukan 3: 1 2 3

seluruh operasi sudah selesai dijalankan. maka, cetak kondisi terakhir dari setiap tumpukan

## Pembahasan

kita dapat memakai 3 stack. kemudian dioperasikan

\*pada code solusi, saya memakai 1 stack tambahan untuk mempermudah dalam mencetak output secara LIFO.

selain memakai stack, kita dapat juga memakai deque.

## Source Code

```
/**
 * Implementasi ADT Stack (Stack menggunakan Linked List)
 *
 * Dibuat dan ditulis oleh Bayu Laksana
 * -- tanggal 22 Januari 2019
 * Struktur Data 2020
 * Implementasi untuk bahasa C
 *
 * !!NOTE!!
 * cara menggunakan lihat pada fungsi main()
 *
 * Dimodifikasi oleh luthfiyyah hanifah amari
 * -- tanggal 6 maret 2022
 * untuk penyelesaian problem "Pembukaan" pada praktikum modul 1
 * kelas struktur data C 2022
 */

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

```

#include <stdio.h>

/* Struktur Node */

typedef struct stackNode_t {
    int data;
    struct stackNode_t *next;
} StackNode;

/* Struktur ADT Stack */

typedef struct stack_t {
    StackNode *_top;
    unsigned _size;
} Stack;

/* Function prototype */

void stack_init(Stack *stack);
bool stack_isEmpty(Stack *stack);
void stack_push(Stack *stack, int value);
void stack_pop(Stack *stack);
int stack_top(Stack *stack);
unsigned stack_size(Stack *stack);

/* Function definition below */

void stack_init(Stack *stack)
{
    stack->_size = 0;
    stack->_top = NULL;
}

bool stack_isEmpty(Stack *stack) {
    return (stack->_top == NULL);
}

void stack_push(Stack *stack, int value)
{
    StackNode *newNode = (StackNode*) malloc(sizeof(StackNode));
    if (newNode) {
        stack->_size++;
        newNode->data = value;

        if (stack_isEmpty(stack)) newNode->next = NULL;
        else newNode->next = stack->_top;

        stack->_top = newNode;
    }
}

void stack_pop(Stack *stack)
{
    if (!stack_isEmpty(stack)) {

```

```

        StackNode *temp = stack->_top;
        stack->_top = stack->_top->next;
        free(temp);
        stack->_size--;
    }
}

int stack_top(Stack *stack)
{
    if (!stack_isEmpty(stack))
        return stack->_top->data;
    return 0;
}

unsigned stack_size(Stack *stack) {
    return stack->_size;
}

int main(int argc, char const *argv[])
{
    // Buat objek Stack
    Stack tumpukan1;
    Stack tumpukan2;
    Stack tumpukan3;

    // PENTING! Jangan lupa diinisialisasi
    stack_init(&tumpukan1);
    stack_init(&tumpukan2);
    stack_init(&tumpukan3);

    int Q, i, jumlah, j;
    char perintah[20];
    int b=1;
    scanf("%d", &Q);
    for(i=0; i<Q; i++)
    {
        scanf("%s", &perintah);
        scanf("%d", &jumlah);
        if(strcmp(perintah, "CETAK") == 0)
        {
            for(j=0; j<jumlah; j++)
            {
                stack_push(&tumpukan1, b++);
            }
        }
        else if(strcmp(perintah, "TTD") == 0)
        {
            for(j=0; j<jumlah; j++)
            {
                stack_push(&tumpukan2, stack_top(&tumpukan1));
                stack_pop(&tumpukan1);
            }
        }
        else if(strcmp(perintah, "SAMPUL") == 0)
    }
}

```

```

        {
            for(j=0; j<jumlah; j++)
            {
                stack_push(&tumpukan3, stack_top(&tumpukan2));
                stack_pop(&tumpukan2);
            }
        }
    }

    Stack tumpukanTemp;
    stack_init(&tumpukanTemp);
    // cetak isi stack
    printf("tumpukan 1: ");
    while (!stack_isEmpty(&tumpukan1)) {
        stack_push(&tumpukanTemp, stack_top(&tumpukan1));
        stack_pop(&tumpukan1);
    }
    while (!stack_isEmpty(&tumpukanTemp)) {
        printf("%d ", stack_top(&tumpukanTemp));
        stack_pop(&tumpukanTemp);
    }
    printf("\n");

    printf("tumpukan 2: ");
    while (!stack_isEmpty(&tumpukan2)) {
        stack_push(&tumpukanTemp, stack_top(&tumpukan2));
        stack_pop(&tumpukan2);
    }
    while (!stack_isEmpty(&tumpukanTemp)) {
        printf("%d ", stack_top(&tumpukanTemp));
        stack_pop(&tumpukanTemp);
    }
    printf("\n");

    printf("tumpukan 3: ");
    while (!stack_isEmpty(&tumpukan3)) {
        stack_push(&tumpukanTemp, stack_top(&tumpukan3));
        stack_pop(&tumpukan3);
    }
    while (!stack_isEmpty(&tumpukanTemp)) {
        printf("%d ", stack_top(&tumpukanTemp));
        stack_pop(&tumpukanTemp);
    }
    printf("\n");

    return 0;
}

```