

STRUKTUR DATA

STACK

ROSA ARIANI SUKAMTO

Blog: <http://hariiniadalahhadiah.wordpress.com>

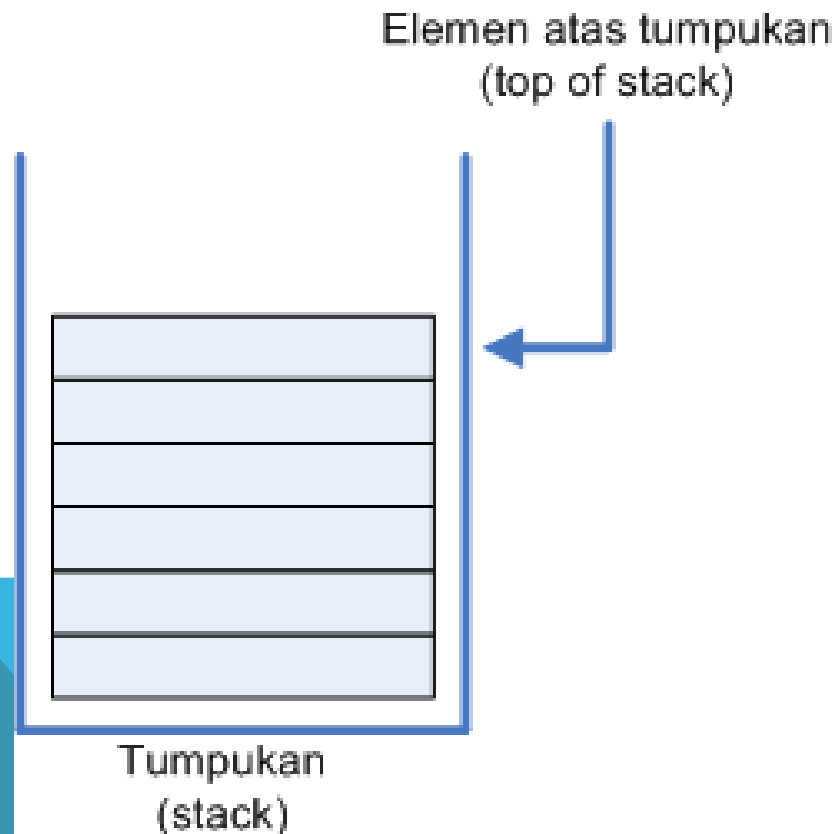
Facebook: <https://www.facebook.com/rosa.ariani.sukamto>

Email: rosa_if_itb_01@yahoo.com



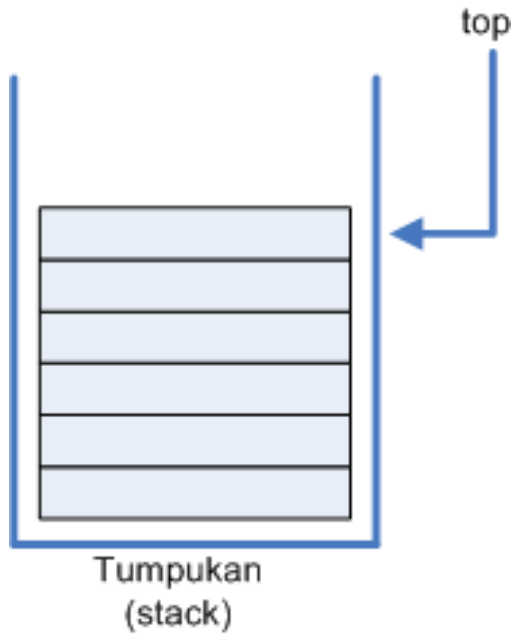
STACK

Tumpukan atau *stack* adalah salah satu konsep struktur data yang memiliki sistem kerja yang terakhir masuk adalah yang pertama keluar (LIFO = *Last In First Out*)

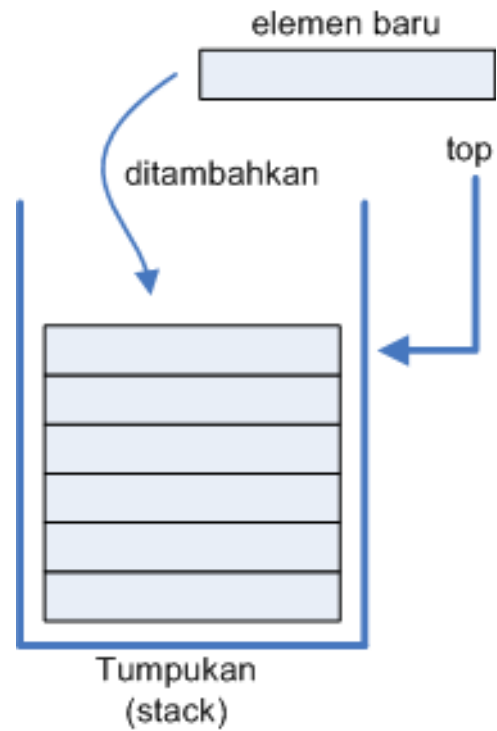


PROSES PUSH

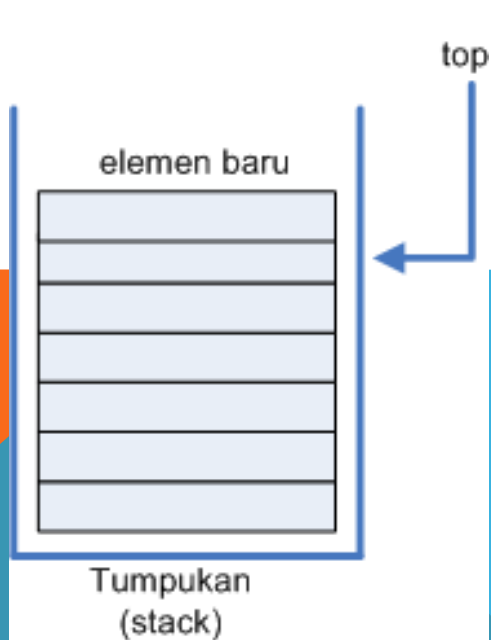
1



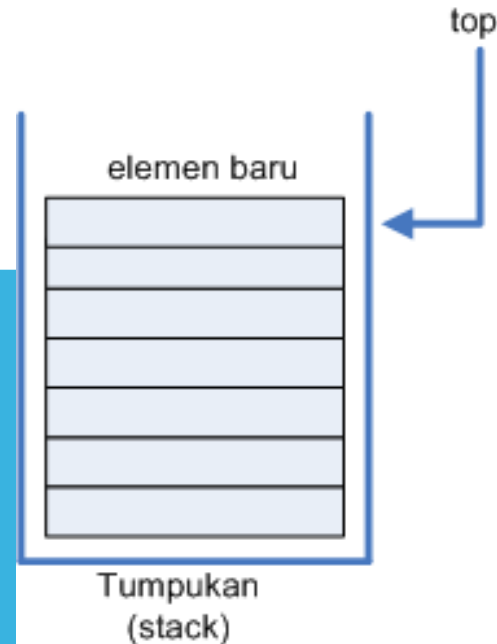
2



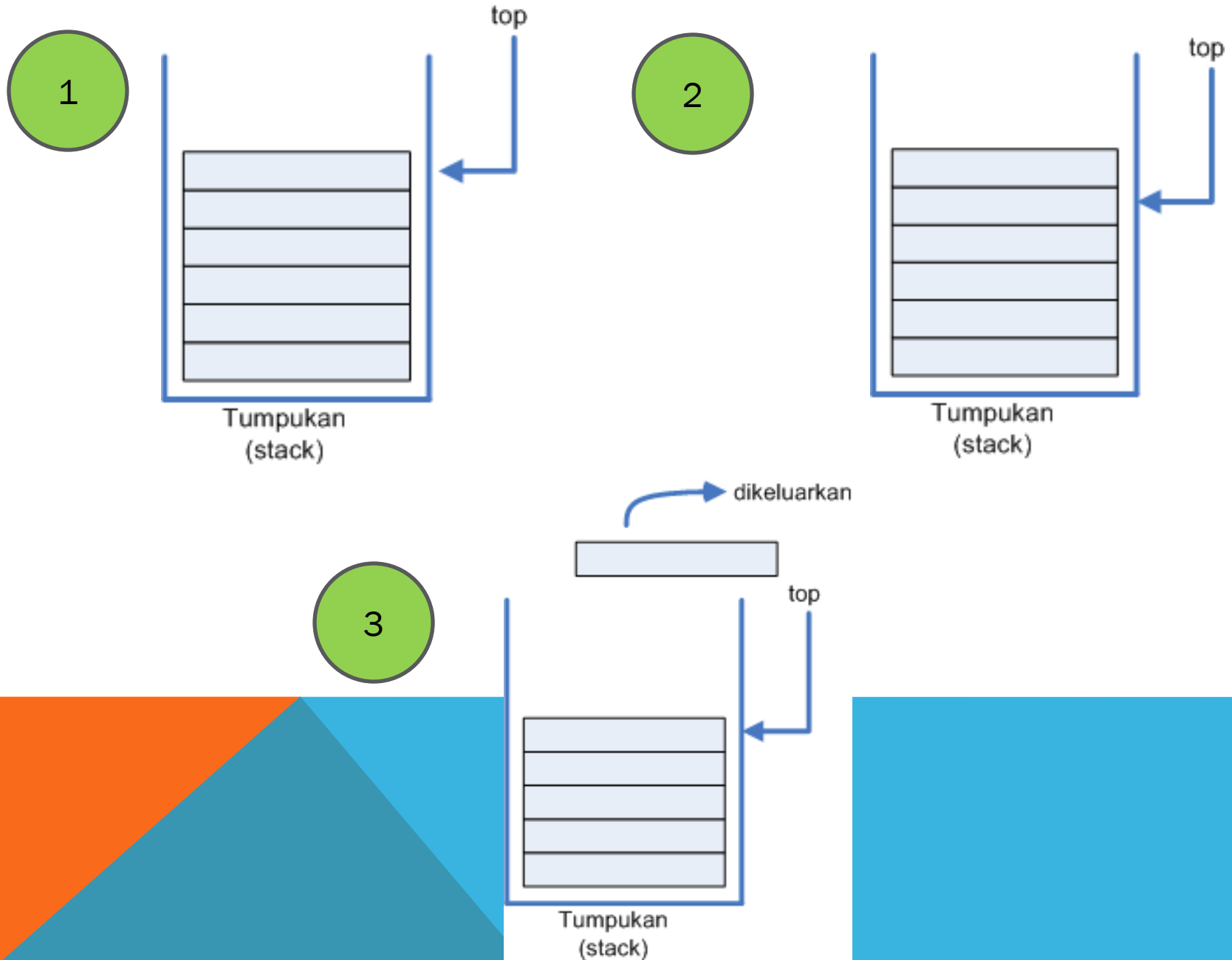
3



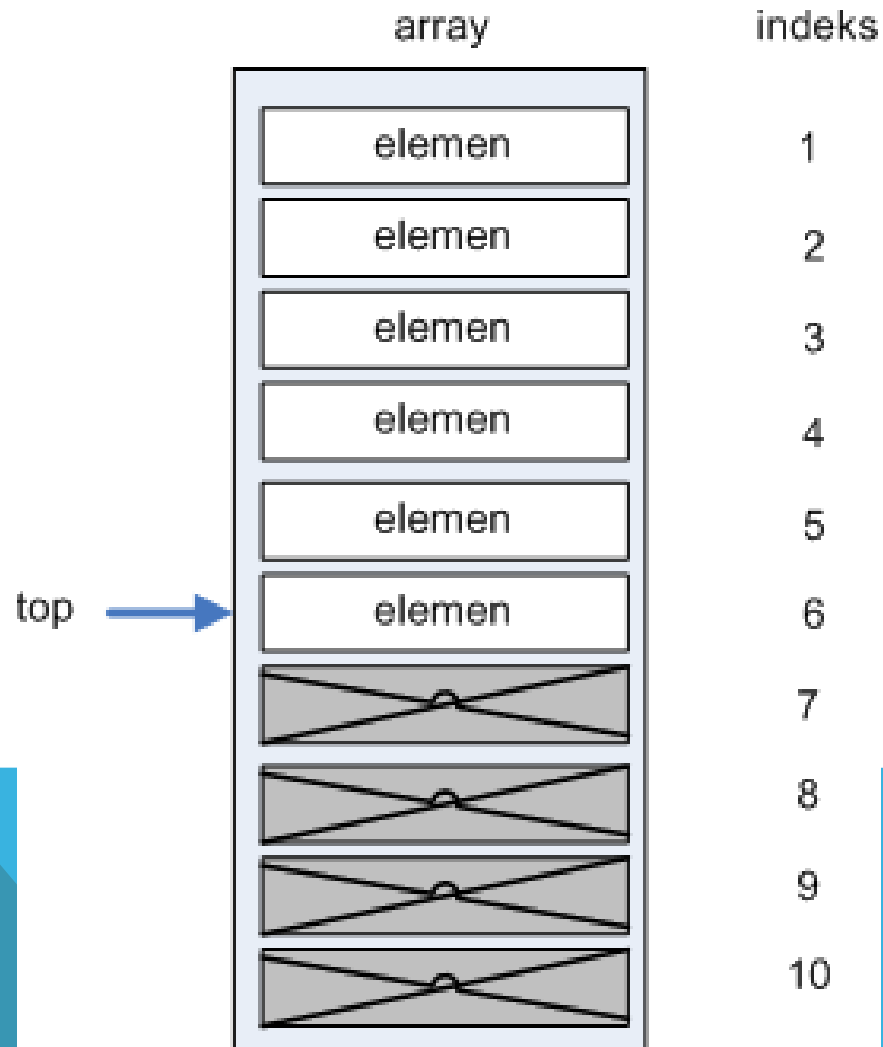
4



PROSES POP



STACK REPRESENTASI STATIS



DEKLARASI ELEMEN DAN INISIALISASI

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

typedef struct{
    char nim[10];
    char nama[50];
    float nilai;
}nilaiMatKul;

typedef struct{
    int top;
    nilaiMatKul data[10];
}stack;

void createEmpty(stack *S){
    (*S).top = -1;
}
```

```
int isEmpty(stack S){
    int hasil = 0;
    if(S.top == -1){
        hasil = 1;
    }
    return hasil;
}

int isFull(stack S){
    int hasil = 0;
    if(S.top == 9){
        hasil = 1;
    }
    return hasil;
}
```

PUSH

```
void push(char nim[], char
    nama[], float nilai, stack *S
){

    if(isFull(*S) == 1){
        /*jika stack penuh*/
        printf("stack penuh\n");
    }
    else{
        if(isEmpty(*S) == 1){
            /* jika stack kosong */
            (*S).top = 0;

            strcpy((*S).data[0].nim,
nim);

            strcpy((*S).data[0].nama,
nama);

            (*S).data[0].nilai = nilai;
        }
    }
}
```

```
        else{
            /* jika stack tidak kosong
            */
            (*S).top = (*S).top + 1;

            strcpy((*S).data[(*S).top].nim
, nim);

            strcpy((*S).data[(*S).top].nam
a, nama);

            (*S).data[(*S).top].nilai =
nilai;
        }
    }
}
```


POP

```
void pop(stack *S){

    if((*S).top == 0){
        /*jika stack berisi satu elemen*/
        (*S).top = -1;
    }
    else{
        if((*S).top != -1){
            /*jika stack tidak kosong*/
            (*S).top = (*S).top - 1;
        }
    }
}
```

PRINT STACK DAN MAIN

```
void printStack(stack S){

    if(S.top != -1){
        printf("-----isi stack-----\n");
        int i;
        for(i=S.top;i>=0;i--){
            printf("=====\n");
            printf("elemen ke : %d\n", i);
            printf("nim : %s\n",
                S.data[i].nim);
            printf("nama : %s\n",
                S.data[i].nama);
            printf("nilai : %f\n",
                S.data[i].nilai);
        }
        printf("-----\n");
    }
    else{
        /* proses jika stack kosong */
        printf("stack kosong\n");
    }

}
```

```
int main(){

    stack S;

    createEmpty(&S);
    printStack(S);

    printf("=====\n");

    push("13507701", "Nana", 64.75, &S);
    push("13507702", "Rudi", 75.11, &S);
    push("13507703", "Dea", 84.63, &S);
    printStack(S);

    printf("=====\n");

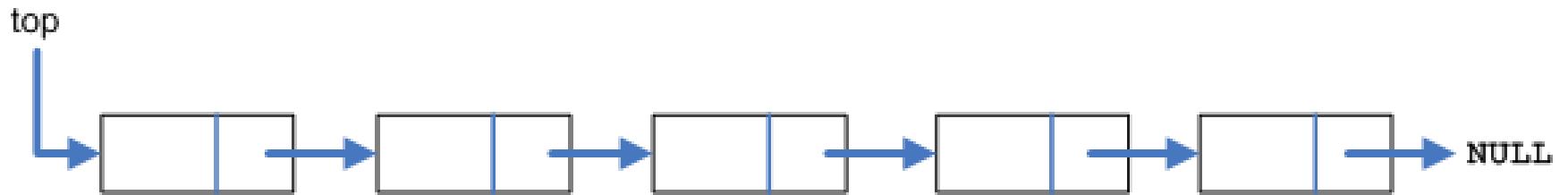
    pop(&S);
    pop(&S);
    printStack(S);

    printf("=====\n");

    return 0;

}
```

STACK REPRESENTASI DINAMIS



DEKLARASI ELEMEN DAN INISIALISASI

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

typedef struct{
    char nim[10];
    char nama[50];
    float nilai;
}nilaiMatKul;

typedef struct elmt *alamatelmt;
typedef struct elmt{
    nilaiMatKul elmt;
    alamatelmt next;
}elemen;

typedef struct {
    elemen* top;
}stack;
```

```
void createEmpty(stack *S){
    (*S).top = NULL;
}

int isEmpty(stack S){
    int hasil = 0;
    if(S.top == NULL){
        hasil = 1;
    }
    return hasil;
}
```

COUNTELEMENT

```
int countElement(stack S){

    int hasil = 0;

    if(S.top != NULL){
        /* stack tidak kosong */

        elemen *elmt;

        /* inisialisasi */
        elmt = S.top;
```

```
        while(elmt != NULL){
            /* proses */
            hasil= hasil + 1;

            /* iterasi */
            elmt = elmt->next;
        }

    }

    return hasil;

}
```

PUSH

```
void push(char nim[], char nama[], float nilai,  
    stack *S ){  
  
    elemen *elmt;  
    elmt = (elemen *) malloc (sizeof (elemen));  
    strcpy(elmt->elmt.nim, nim);  
    strcpy(elmt->elmt.nama, nama);  
    elmt->elmt.nilai = nilai;  
    elmt->next = (*S).top;  
    (*S).top = elmt;  
    elmt = NULL;  
  
}
```

POP

```
void pop(stack *S) {  
  
    if ((*S).top != NULL) {  
        /* jika stack bukan stack kosong */  
  
        elemen *elmt = (*S).top;  
        (*S).top = (*S).top->next;  
        elmt->next = NULL;  
        free(elmt);  
    }  
  
}
```

PRINTSTACK

```
void printStack(stack S){

    if(S.top != NULL){
        printf("-----isi stack-----\n");

        elemen *elmt = S.top;

        int i = 1;

        while(elmt != NULL){

            printf("=====\n");
            ;
            printf("elemen ke : %d\n", i);
            printf("nim : %s\n",
                elmt->elmt.nim);
            printf("nama : %s\n",
                elmt->elmt.nama);
            printf("nilai : %f\n",
                elmt->elmt.nilai);
```

```
        /* iterasi */
        elmt = elmt->next;
        i = i + 1;

    }

    printf("-----\n");
}

else{

    /* proses jika stack kosong */

    printf("stack kosong\n");
}

}
```


MAIN

```
int main() {  
    stack S;  
  
    createEmpty(&S);  
    printStack(S);  
  
    printf("=====\n");  
  
    push("13507701", "Nana", 64.75, &S);  
    push("13507702", "Rudi", 75.11, &S);  
    push("13507703", "Dea", 84.63, &S);  
    printStack(S);  
  
    printf("=====\n");  
  
    pop(&S);  
    pop(&S);  
    printStack(S);  
  
    printf("=====\n");  
  
    return 0;  
}
```

DAFTAR PUSTAKA

S, Rosa A. dan M. Shalahuddin. 2010. Modul Pembelajaran: Struktur Data. Modula: Bandung.

