Stack (Tumpukan)

By
Entin Martiana, S.Kom.
Stack

- Merupakan Struktur Data dengan prinsip Last In First Out
Penyajian Stack dengan Array

#define MAXSTACK 100
/* Definisi struktur stack */
struct Stack{
    int item[MAXSTACK];    /* Array yang berisi data tumpukan */
    int Count;    /* menunjukkan indeks data paling atas dari stack */
};
Operasi pada Stack

• Inisialisasi
• Apakah stack masih kosong?
• Apakah stack sudah penuh?
• Menyisipkan sebuah elemen ke dalam stack (Push)
• Mengambil elemen terakhir dari stack (Pop)
Inisialisasi

void InitializeStack(Stack *S)
{
    S->Count = 0;
}

Cek Apakah Stack Kosong?

```c
int Empty(Stack *S)
{
    return (S->Count == 0);
}
```
Cek Apakah Stack Penuh?

```c
int Full(Stack *S)
{
    return (S->Count == MAXSTACK);
}
```
void Push(int x, Stack *S) 
{
    if (Full(S))
        printf("Stack penuh! Data tidak dapat masuk!");
    else
    {
        S->Item[S->Count]=x;
        ++(S->Count);
    }
}
void Pop(Stack *S, int *x)
{
    if (Empty(S))// stack kosong
        printf("Stack masih kosong!");
    else
    {
        --(S->Count);
        *x = S->Item[S->Count];
        S->Item[S->Count]=0;
    
    }
}
main()
{
    int in=0;
    //push
    pop(&S,&in);
    printf("%d",in);
}
Pop

int Pop(Stack *S)
{
    if (Empty(S))//stack kosong
        printf("Stack masih kosong!");
    else
    {
        --(S->Count);
        return ( S->Item[S->Count]);
        S->Item[S->Count]=0;
    }
}
main()
{
    int in;
    //push
    in=pop(&S);
    printf("%d",in);
}
Kondisi awal

MAX = 4

count 0
Push : 5

MAX = 4

count → 1
Push : 3

MAX = 4

count → 2

0 1 3 2

3 5
Push : 8

MAX = 4

3
2
1
0

count 3

3
Push : 1

MAX = 4

count → 4
Push : 9

Stack penuh! Data tidak dapat masuk!

MAX = 4

count → 4
Pop

MAX = 4

count \rightarrow 3

1

3

8

2

3

1

5

0
Pop

MAX = 4

count  2

8

3
2
3
1
5
0
MAX = 4

Pop

count → 1

3
2
1
0
5
Pop

MAX = 4

3
2
1
0

count → 0

5
Pop

Stack masih kosong!

MAX = 4

count ➔ 0
Penyajian Stack dengan Linked List

- Single Linked List
- Double Linked List