

Konsep_1

Stack adalah salah satu cara penyimpanan data dalam komputer yang bekerja berdasarkan prinsip LIFO (Last In First Out), yang berarti data yang terakhir dimasukkan ke dalam stack akan menjadi data pertama yang keluar dari stack.

Operasi-operasi yang terdapat pada Stack meliputi:

- Push: Menambahkan data ke dalam stack
- Pop: Mengeluarkan data paling atas dari stack
- Peek: Melihat data paling atas pada stack tanpa menghapusnya
- Size: Mengetahui jumlah data pada stack
- isEmpty: Mengetahui apakah stack kosong atau tidak

```
In [ ]: def stack():  
        return []  
def push(stack, data):  
    stack.append(data)  
def pop(stack):  
    return stack.pop()  
def peek(stack):  
    return stack[-1]  
def isEmpty(stack):  
    return stack == []  
def size(stack):  
    return len(stack)
```

Implementasi_2.1

```
In [47]: def dec2bin(num):  
        temp=stack()  
        while num>0:  
            sisa = num%2  
            print(num, 'div 2 = ', end='')  
            push(temp,sisa)  
            num=num//2  
            print(num, 'sisa = ', sisa, 'push(stack,',sisa,')')  
            [print('|', st, "|") for st in temp[::-1]]  
            print('-----')  
  
        tempStr=''  
        for i in range(size(temp)):  
            tempStr=tempStr+str(pop(temp))  
        return 'Biner = '+tempStr  
print(dec2bin(19))
```

```
19 div 2 = 9 sisa = 1 push(stack, 1 )
| 1 |
-----
9 div 2 = 4 sisa = 1 push(stack, 1 )
| 1 |
| 1 |
-----
4 div 2 = 2 sisa = 0 push(stack, 0 )
| 0 |
| 1 |
| 1 |
-----
2 div 2 = 1 sisa = 0 push(stack, 0 )
| 0 |
| 0 |
| 1 |
| 1 |
-----
1 div 2 = 0 sisa = 1 push(stack, 1 )
| 1 |
| 0 |
| 0 |
| 1 |
| 1 |
-----
Biner = 10011
```

Implementasi_3.1

```

In [52]: def isValidTagHTML(html):
    st = stack()
    err = ['Error : ']
    flag = True
    for tag in html.split('\n'):
        if tag.startswith('</'):
            if isEmpty(st):
                err.append("Kelebihan Close Tag : "+tag)
                flag = False
            # kosong
            elif tag[2:] != peek(st)[1:]:
                err.append("Open Tag "+tag+" --> "+peek(st))
                pop(st)
                flag = False
            # tidak sama
            else:
                pop(st)
            elif tag.startswith('<'):
                push(st, tag)
        if not isEmpty(st):
            [err.append(f"Kelebihan Open Tag : "+ i) for i in st[::-1]]
            flag = False
        if flag:
            return True
    return flag, [f"{i}. {err[i]}" if i != 0 else err[i] for i in range(1, len(err))]

html = """
<html>
<head>
<body>
<title>
praktiku struktur data
<h1>
Stack
</h1>
</body>
</html>
"""
isValidTagHTML(html)

```

```

Out[52]: (False,
['Error : ',
'1. Open Tag </body> --> <title>',
'2. Open Tag </html> --> <body>',
'3. Kelebihan Open Tag : <head>',
'4. Kelebihan Open Tag : <html>'])

```

In []: