TUGAS MENJELASKAN PROGRAM

Tugas Ini Dibuat Guna Memenuhi Mata Kuliah Struktur Data

Dosen pengampu:

Adam bachtiar, s.kom, M.MT

Disusun Oleh :

Nama : Dini putri buana

NIM : 24241064

Kelas : PTI B

PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
FAKULTAS SAINS, TEKNIK DAN TERAPAN
UNIVERSITAS PENDIDIKAN MANDALIKA MATARAM

2025

❖ Membuat node

```python
# Node sebagai elemen linked list
def buat_node(data):
    return {"data": data, "next": None}

```

Penjelasan:

1. Fungsi ini digunakan untuk membuat satu node dalam linked list.
2. Node diwakili dengan dictionary yang memiliki dua elemen:
   - "data" → untuk menyimpan nilai elemen stack.
   - "next" → untuk menunjuk ke node berikutnya (default None).

❖ Menambah elemen ke stack

```python
# Fungsi-fungsi stack
def push(stack, data, kapasitas):
    if ukuran(stack) >= kapasitas:
        print("Stack sudah penuh!")
        return stack
    node_baru = buat_node(data)
    node_baru["next"] = stack
    stack = node_baru
    tampilkan_stack(stack)
    return stack

```

Penjelasan:

1. Mengecek apakah ukuran stack sudah mencapai kapasitas.
2. Jika sudah penuh, maka elemen tidak ditambahkan.
3. Membuat node baru.
4. Node baru akan menunjuk ke stack lama (elemen sebelumnya).
5. Puncak stack sekarang adalah node baru tersebut.
6. Menampilkan isi stack setelah ditambahkan, lalu mengembalikannya.

❖ Menghapus elemen dari stack

```python
def pop(stack):
    if stack is None:
        print("Stack kosong!")
        return stack
    print(f"Elemen {stack['data']} dihapus dari stack.")
    stack = stack["next"]
    tampilkan_stack(stack)
    return stack

```

Penjelasan:

1. Mengecek apakah stack kosong.
2. Jika kosong, tampilkan pesan dan kembalikan stack.
3. Menampilkan elemen yang dihapus.
4. Pindahkan puncak stack ke elemen berikutnya (next).
5. Menampilkan stack setelah di-pop dan mengembalikan stack.

❖ Menghitung jumlah elemen dalam stack

```python
def ukuran(stack):
    count = 0
    while stack is not None:
        count += 1
        stack = stack["next"]
    return count
```

Penjelasan:

1. Mengiterasi setiap node dari puncak ke bawah.
2. Menambahkan 1 setiap kali ada node hingga mencapai None.
3. Mengembalikan total jumlah elemen dalam stack.

❖ Menampilkan data paling atas

```python
def puncak(stack):
    if stack is None:
        print("Stack kosong!")
    else:
        print(f"Elemen puncak stack: {stack['data']}")
```

Penjelasan:

Menampilkan data yang ada di puncak stack (jika tidak kosong).

❖ Fungsi cek penuh

```python
def cek_penuh(stack, kapasitas):
    if ukuran(stack) >= kapasitas:
        print("Stack penuh!")
    else:
        print("Stack belum penuh.")
```

Penjelasan:

Mengecek apakah jumlah elemen dalam stack sudah mencapai kapasitas.

❖ Menampilkan Isi Stack

```python
def tampilkan_stack(stack):
    elemen = []
    while stack is not None:
        elemen.append(stack["data"])
        stack = stack["next"]
    elemen.reverse()
    print("Stack :", elemen)
```

Penjelasan:

1. Mengambil semua elemen dari puncak ke dasar dan memasukkannya ke dalam list elemen.
2. Kemudian dibalik (reverse) agar ditampilkan dari bawah ke atas (seperti tampilan stack sebenarnya).
3. Misal, jika puncak 30, lalu 20, lalu 10, maka ditampilkan [10, 20, 30].

❖ Fungsi utama

```python
# Program utama
def main():
    kapasitas = int(input("Tentukan berapa kapasitas stack : "))
    stack = None

    while True:
        print("\nPilih menu berikut ini : ")
        print("1. Menambah isi stack")
        print("2. Menghapus isi stack")
        print("3. Cek Ukuran Stack saat ini")
        print("4. Cek Puncak Stack")
        print("5. Cek Stack Full")
        print("6. Keluar")
        pilihan = input("\nMasukkan pilihan anda : ")

        if pilihan == '1':
            while True:
                data = input("\nMasukkan isi stack : ")
                stack = push(stack, data, kapasitas)
                lanjut = input("\nMenambah isi Stack Pilih [Ya/Tidak] : ")
                if lanjut.lower() != 'ya':
                    break
```

```python
        elif pilihan == '2':
            stack = pop(stack)

        elif pilihan == '3':
            print(f"Ukuran stack saat ini: {ukuran(stack)}")

        elif pilihan == '4':
            puncak(stack)

        elif pilihan == '5':
            cek_penuh(stack, kapasitas)

        elif pilihan == '6':
            print("Keluar dari program.")
            break

        else:
            print("Pilihan tidak valid!")
```

Penjelasan:

1. Meminta pengguna menentukan kapasitas maksimum stack.
2. Stack diinisialisasi dengan None (kosong).
3. Perulangan menu menampilkan menu ke pengguna dan menunggu input pilihan.
4. Push, Pengguna bisa menambahkan lebih dari satu data secara berurutan, hingga menjawab "Tidak".
5. Pop, Menghapus elemen paling atas dari stack.
6. Ukuran, Menampilkan jumlah elemen dalam stack.
7. Puncak, Menampilkan data di puncak stack.
8. Cek penuh, Mengecek apakah stack sudah penuh.
9. Keluar dari program.
10. Menangani input yang tidak sesuai pilihan menu.

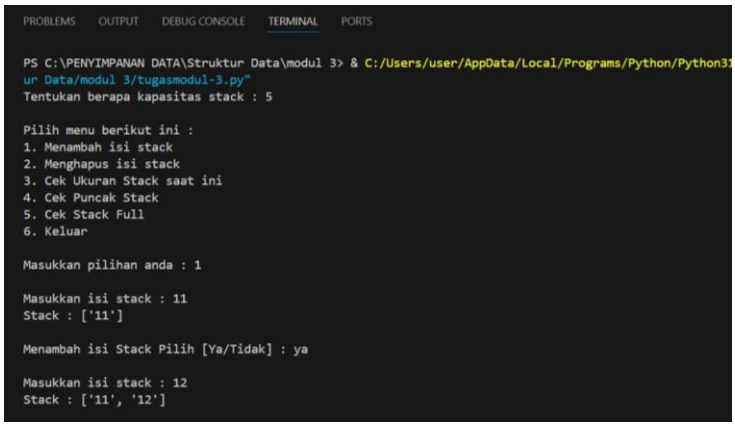
❖ Menjalankan Program

```python
# Jalankan program
if __name__ == "__main__":
    main()
```

Penjelasan:

Menjalankan Program

Output:


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
PS C:\PENYIMPANAN DATA\Struktur Data\modul 3> & C:/Users/user/AppData/Local/Programs/Python/Python3
ur Data/modul 3/tugasmodul-3.py"
Tentukan berapa kapasitas stack : 5
Pilih menu berikut ini :
1. Menambah isi stack
2. Menghapus isi stack
3. Cek Ukuran Stack saat ini
4. Cek Puncak Stack
5. Cek Stack Full
6. Keluar
Masukkan pilihan anda : 1
Masukkan isi stack : 11
Stack : ['11']
Menambah isi Stack Pilih [Ya/Tidak] : ya
Masukkan isi stack : 12
Stack : ['11', '12']