

Nama : Muhammad Galuh Gumelar

NIM : J0403221017

4. jelaskan potongan kode yang kalian buat pada no 3

```
def vis (self):
    print ("")

    for i in range (self.kapasitas, 0, -1) :
        print("    [%2s]   "%(i), end="")
    print("")

    for i in range(self.kapasitas):
        | print("----- ", end="")
    print("")

    kosong = self.kapasitas - self.isi
    for i in range (kosong):
        | print(" %-8s >>"%( ""), end="")

    posisi = self.belakang
    for i in range (self.isi):
        print(" %-8s >>"%(self.data[posisi]), end="")
        posisi = (posisi - 1)
        if posisi < 0 :
            posisi = self.kapasitas-1
    print(" [DEPAN] ")
    for i in range (self.kapasitas):
        | print("----- ", end="")
    print("")
```

Kode yang diberikan adalah sebuah metode bernama 'vis' milik sebuah kelas, yang bertujuan untuk memvisualisasikan isi dari struktur data dengan kapasitas ukuran tetap, yang dikenal sebagai antrian melingkar.

Fungsi menampilkan struktur antrian melingkar di konsol/terminal dengan mencetak string yang diformat menggunakan iterasi loop dan pernyataan bersyarat.

Pertama, baris baru yang kosong (") dicetak untuk tujuan pemformatan, setelah itu kode membangun batas horizontal paling atas menggunakan perulangan for yang beralih dari kapasitas antrian "self.kapasitas" ke 1. Ia menggunakan penentu format untuk melapisi nilai indeks di kedua sisi.

Selanjutnya, garis horizontal lain dibentuk menggunakan for loop, menggantikan indeks entri dengan string dengan panjang yang sama di antaranya.

Kemudian, ia menghitung jumlah ruang kosong dalam antrean dengan mengurangi jumlah elemen yang terisi dari kapasitasnya, lalu menggunakan perulangan for lainnya untuk mencetak representasi panah kanan '>>' melalui padding.

Setelah itu, ia menetapkan 'posisi' variabel lokal ke indeks elemen paling belakang (yaitu, `self.belakang`), jangan sampai ada kesalahan underflow atau masalah penuh yang disebabkan oleh melebihi kapasitas lingkaran antrian.

Akhirnya, entri yang tersisa ditampilkan menggunakan perulangan `for` yang dimulai pada elemen terakhir 'posisi' dan diakhiri dengan elemen `pertama+1`:

Itu mencetak setiap elemen menggunakan penentu format hingga simbol penunjuk menuju ujung depan.

Jika penunjuk mencapai nol, iterasi berikutnya memutarkannya ke belakang.

Terakhir, ini mencetak batas horizontal terakhir yang identik dengan yang pertama.

Secara keseluruhan, metode 'vis' dari kelas `antrean` lingkaran menghasilkan visualisasi yang mudah dipahami dari kumpulan data yang dikelolanya sambil menunjukkan tata letak dan lokasi saat ini dari setiap elemen yang ada dalam antrean melingkar.