

Kasus Awal:

Buatlah program untuk menghitung sejumlah inputan **jumlah** dan pengkali berdasarkan inputan **nilai**

```
Masukan Jumlah Baris : 5
Masukan Penjumlah : 2
2 + 1 = 3
2 + 2 = 4
2 + 3 = 5
2 + 4 = 6
2 + 5 = 7
```



Perulangan 2

Minggu 10

Poin Penting:

- Mahasiswa memahami logika Algoritma Perulangan bersarang
- Mahasiswa dapat menggambarkan konstruksi Flowchart Perulangan bersarang
- Mahasiswa dapat menggunakan konstruksi perulangan bersarang dalam code

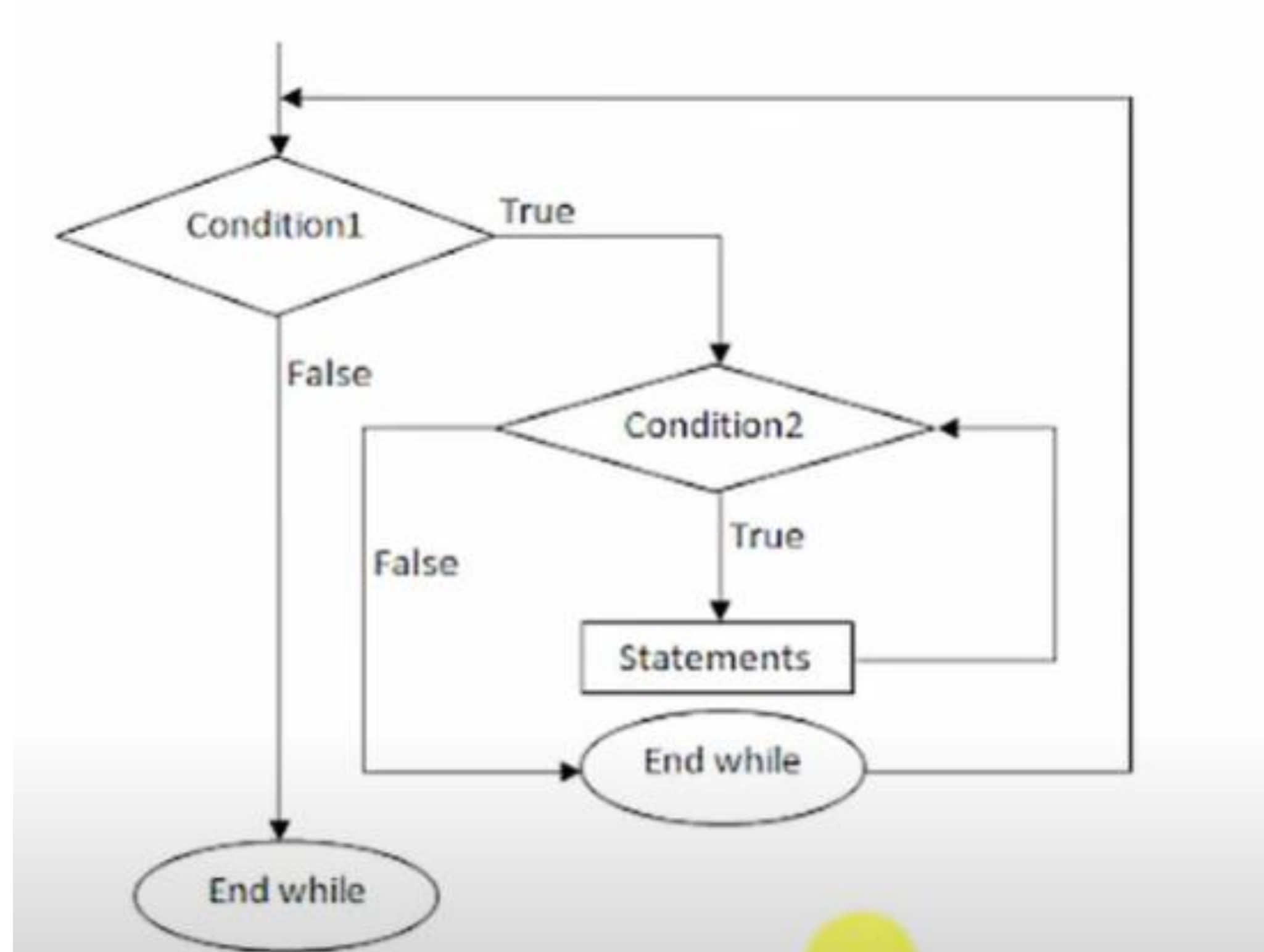


Perulangan Bersarang (Nested Loop):

- Nested loop adalah sebutan perulangan di dalam perulangan
- Konsep Nested Loop digunakan untuk penyelesaian algoritma yang cukup kompleks
- Semua jenis perulangan dapat digunakan untuk Nested Loop



Perulangan Bersarang (Nested Loop):



Struktur Kontrol Nested Loop:

while ekspresi1:

 while ekspresi2 :

 statementA (s)

 statementB (s)

for indeks1 in tipe_koleksi :

 for indeks2 in tipe_koleksi :

 statementA (s)

 statementB (s)



Contoh Struktur Kontrol Nested Loop:

```
Masukan maksimal pengkali : 3
```

```
1 * 1 = 1
```

```
1 * 2 = 2
```

```
1 * 3 = 3
```

```
2 * 1 = 2
```

```
2 * 2 = 4
```

```
2 * 3 = 6
```

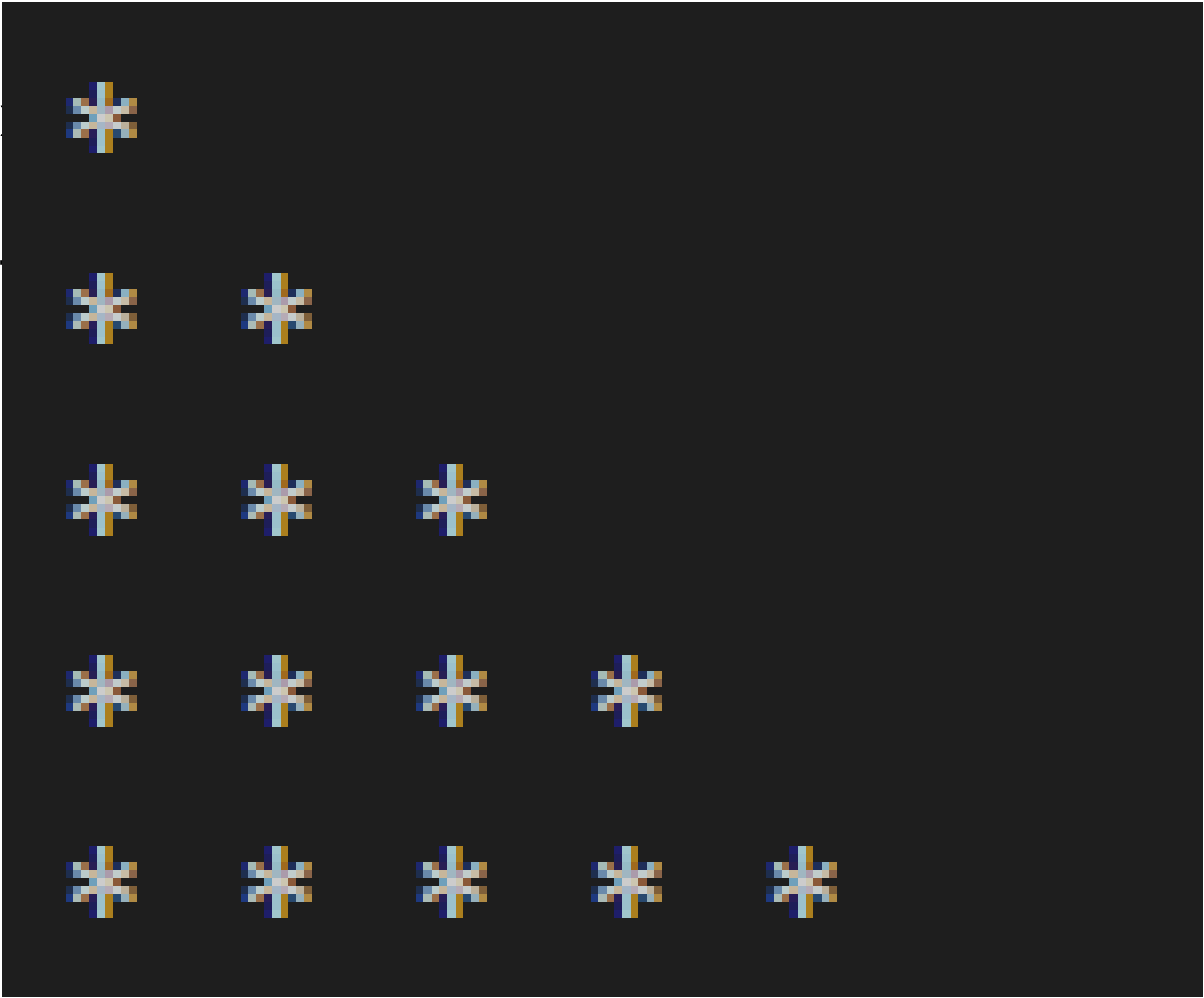
```
3 * 1 = 3
```

```
3 * 2 = 6
```

```
3 * 3 = 9
```



Contoh Struktur Kontrol Nested Loop:



: "))



Searching:

Pencarian (Searching) merupakan tindakan untuk mendapatkan suatu data dalam kumpulan data

“Algoritma yang menerima argument dan mencoba menemukan rekaman berdasarkan kunci”



Metode Searching:

- sequential
- index sequential
- binary
- Fibonacci



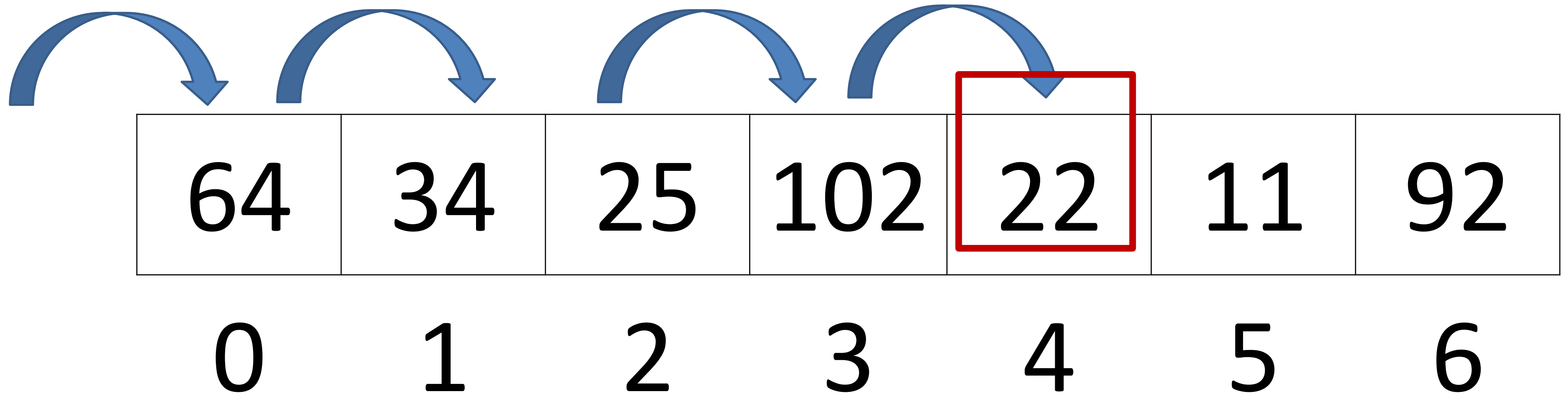
Sequential:

Sequential Searching merupakan algoritma pencarian beruntun yang prosesnya membandingkan setiap elemen satu persatu secara beruntun, metode ini tidak memerlukan data terurut



Sequential:

KEY = 22



Sequential:

```
def cari(a,b):  
    n=len(a)  
    for i in range(n):  
        if b==a[i]:  
            print('Data Ditemukan berada pada index ke ',i)
```

```
a=[3,4,5,6,9]  
print('Nilai A : ',a)  
cari(a,5)
```



Binary:

Binary Search merupakan sebuah teknik pencarian data dengan cara berulang kali membagi separuh dari jumlah data yang dicari sampai sehingga memperkecil lokasi pencarian menjadi satu data, metode ini memerlukan data terurut



Sequential:

KEY = 22

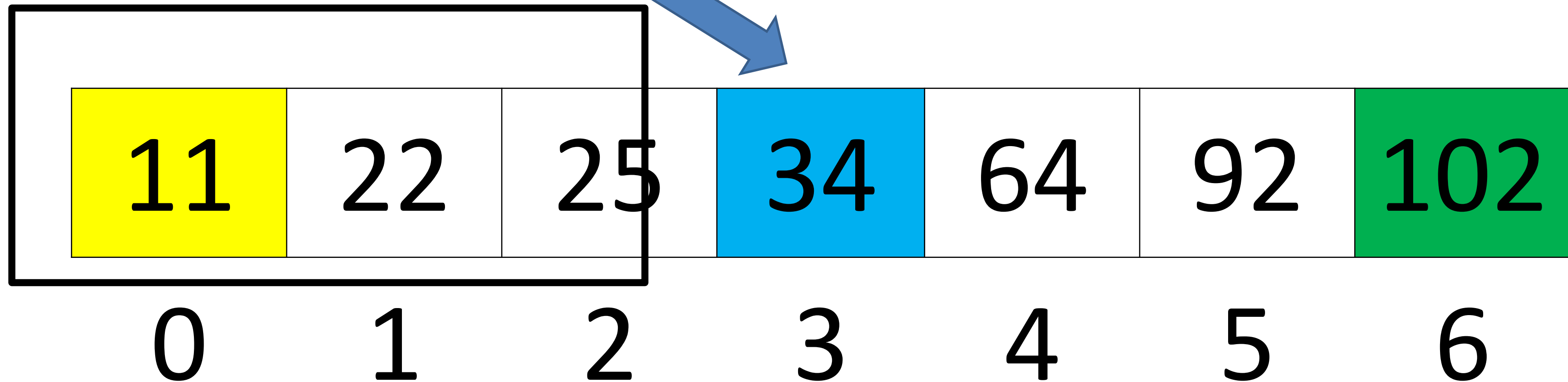
64	34	25	102	22	11	92
0	1	2	3	4	5	6



Sequential:

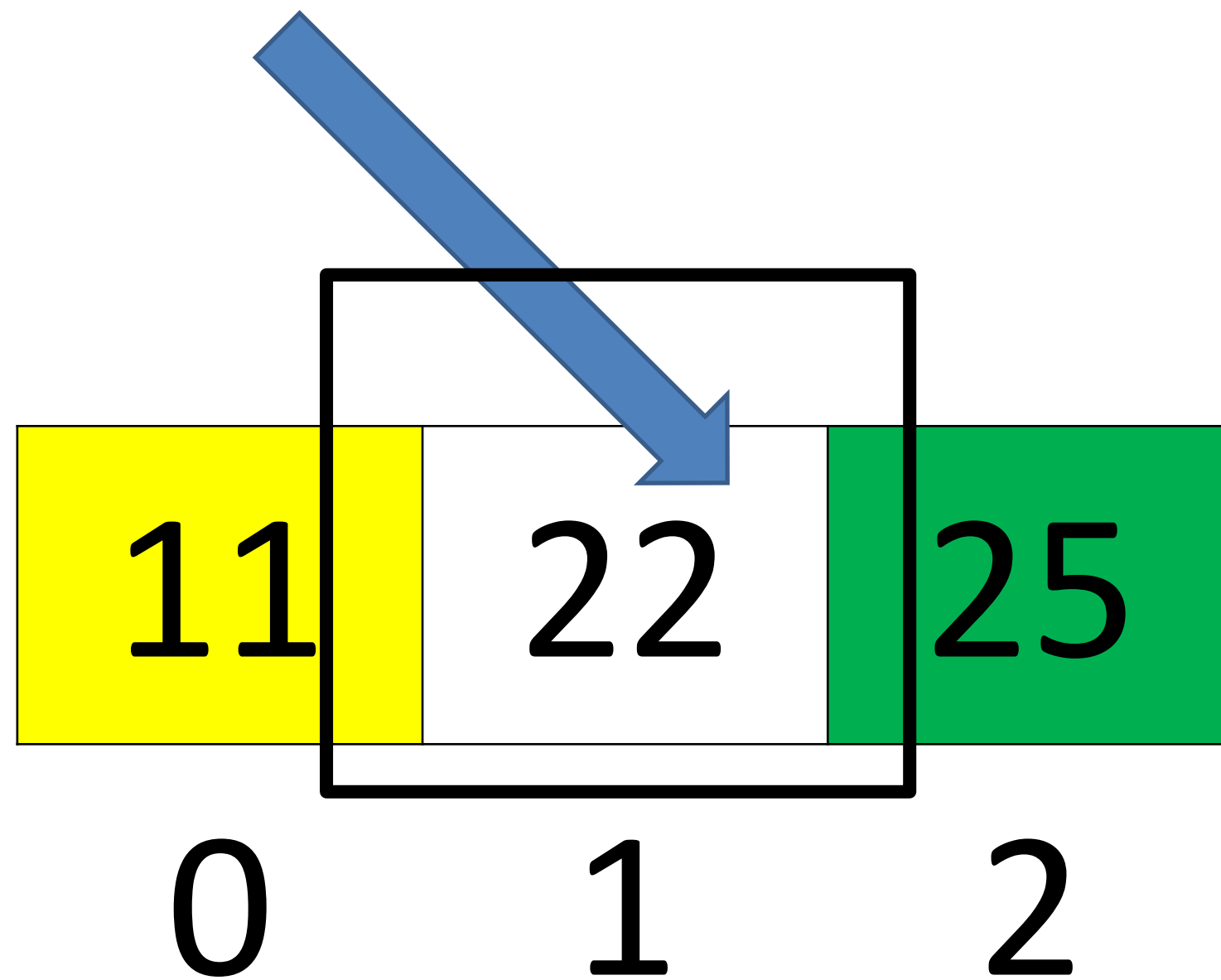
KEY = 22

KEY < TENGGAH



Sequential:

KEY = 22



Binary:

```
def cari(a,b):  
    awal=0  
    akhir=len(a)  
  
    for i in range(awal,akhir):  
        tengah=int((awal+akhir)/2)  
        if (b==a[tengah]):  
            print('Data Ditemukan pada indeks ke', tengah)  
            break  
        elif (b<a[tengah]):  
            akhir=tengah-1  
        else:  
            awal=tengah+1  
  
a=[1,2,3,4,5]  
print(a)  
cari(a,4)
```



Kasus:

Buatlah sebuah program yang bertugas untuk memasukkan nilai ke dalam list sejumlah inputan. Selanjutnya membuat pencarian posisi berdasarkan nilai yang dicari.

```
Masukan Jumlah : 4
Masukan Nilai : 4
Masukan Nilai : 2
Masukan Nilai : 1
Masukan Nilai : 3
Hasil Inputan : [4, 2, 1, 3]
Nilai yang dicari : 2
data : 2 Ditemukan di posisi : [2]
```





Terima Kasih