

# MODUL 8

## SEARCHING

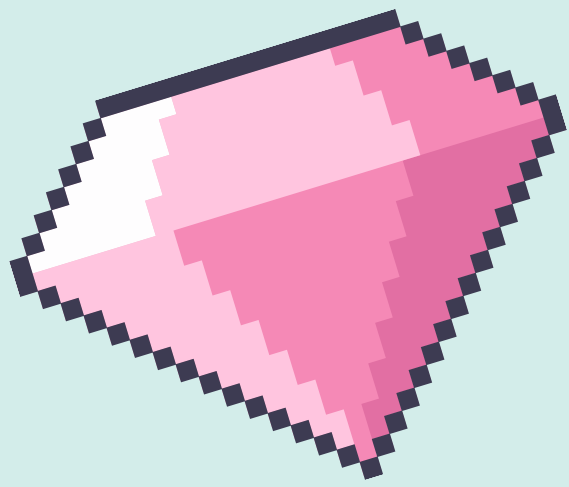


ALIZA NURFITRIAN MEIZAHRA – 1101223083

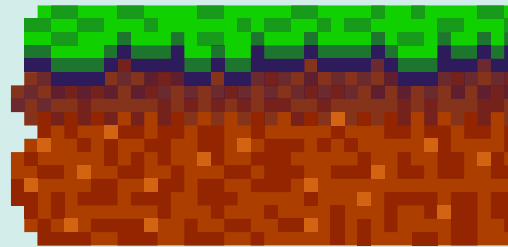
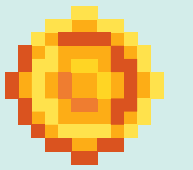
# POINT

- 1 Tujuan Praktikum
- 2 Apa itu searching ?
- 3 Searching dan sorting ?

- 4 Metode searching
- 5 Contoh kasus



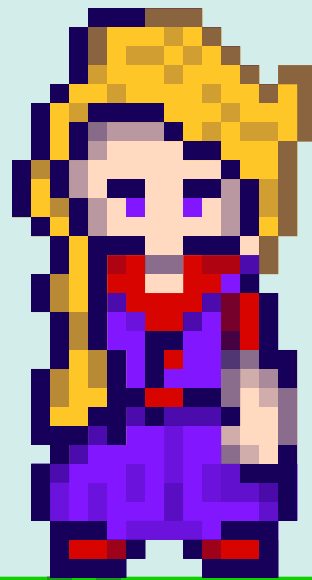
# TUJUAN PRAKTIKUM



Memahami dan mengerti logika pencarian dengan menggunakan metode sequential dan binary



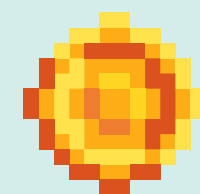
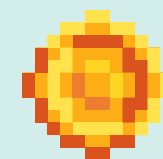
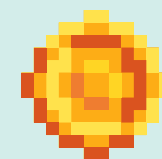
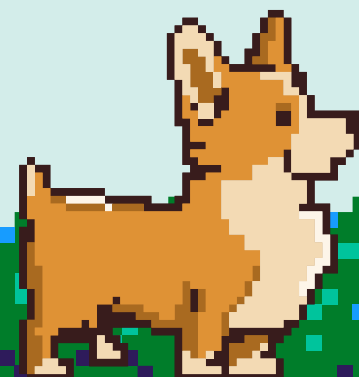
Mampu mengimplementasikan logika pencarian ke dalam program menggunakan bahasa c



# APA ITU SEARCHING ?

Searching adalah proses mencari elemen atau data tertentu dalam suatu kumpulan data.

searching digunakan untuk menemukan posisi atau keberadaan suatu nilai dalam struktur data seperti array.



# QUESTION

Apakah searching dan sorting  
adalah satu hal yang sama ?



Betul



Tidak



# RIGHT ANSWER

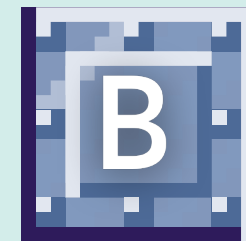


Tidak



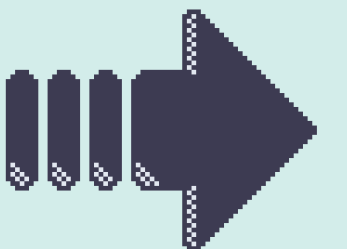
## Searching

- Mencari elemen/ nilai tertentu dalam koleksi data
- Tujuan untuk menemukan posisi elemen dalam urutan tertentu
- Metode Sequential dan binary search

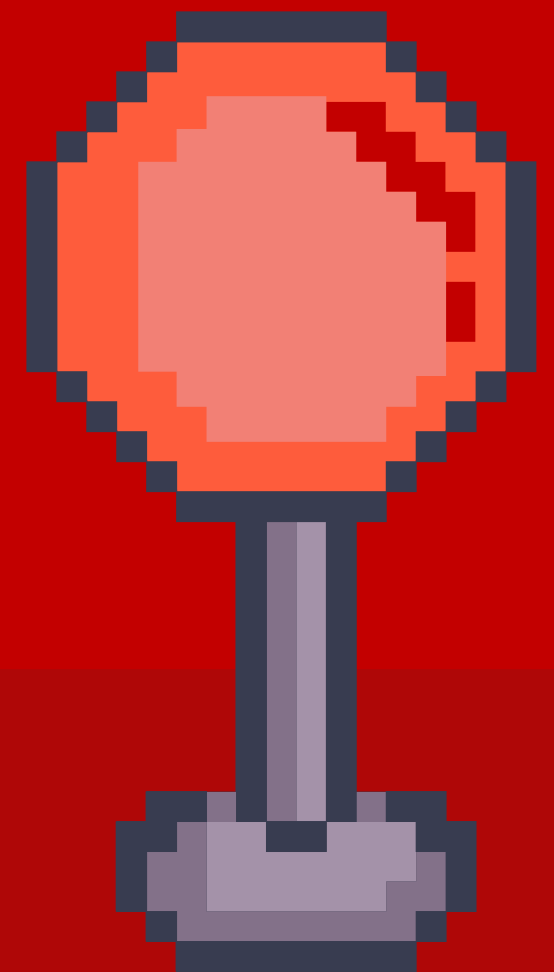
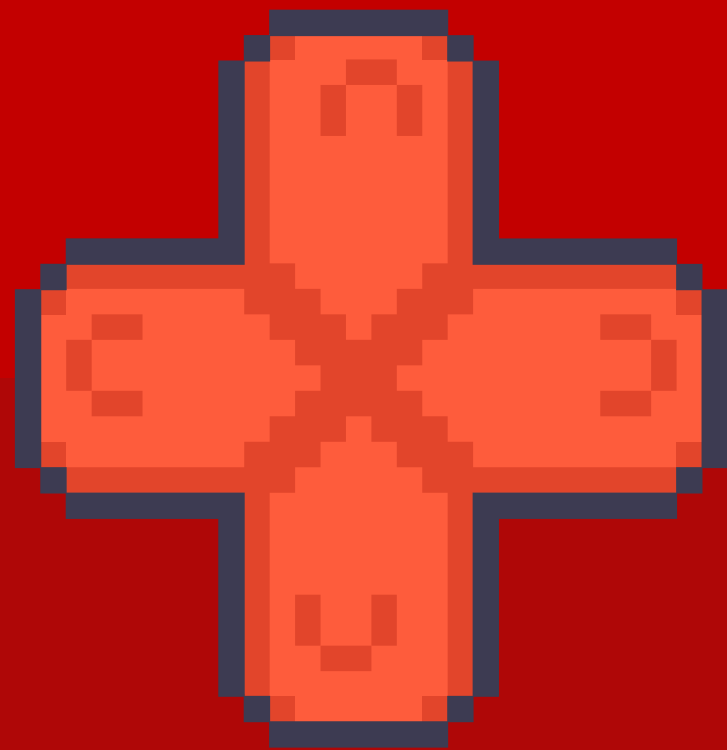


## Sorting

- Mengurutkan urutan nilai/ data dari tertinggi ke terendah atau sebaliknya
- Tujuan untuk mempermudah pemrosesan data
- Metode counting, selection, insertion dan binary sort



# METODE SEARCHING



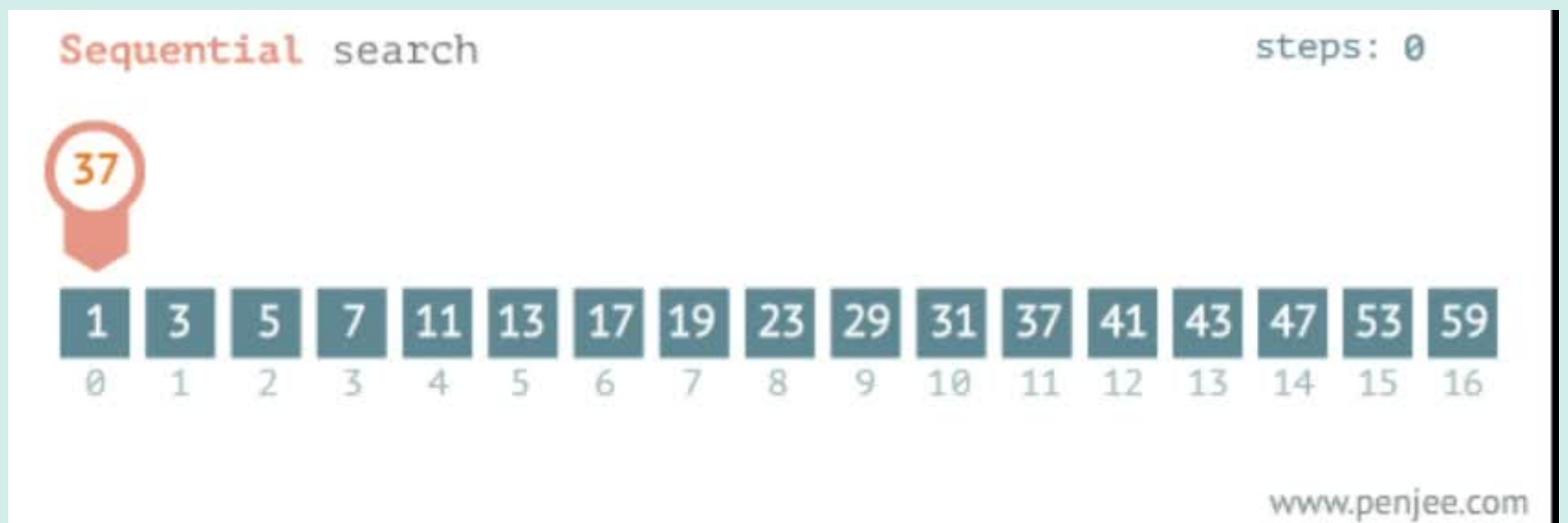
START



HARD LEVEL

# SEQUENTIAL SEARCH

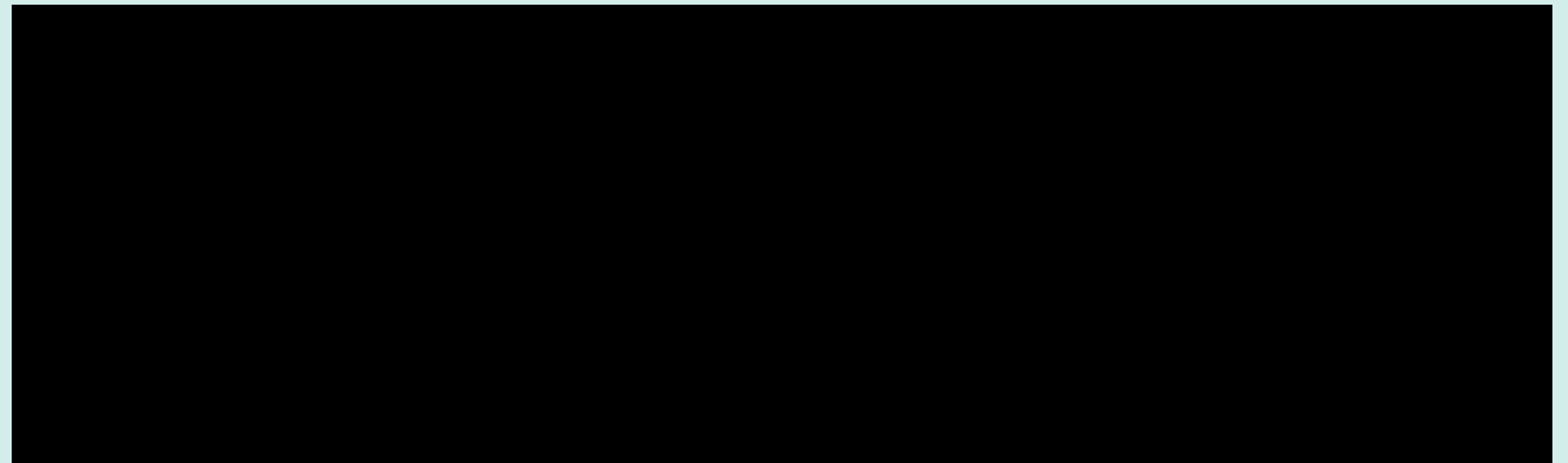
Metode pencarian sederhana yang dilakukan dengan membandingkan setiap elemen dalam koleksi data satu per satu.

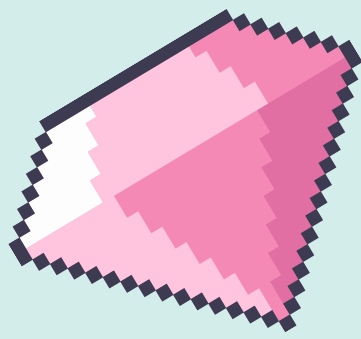




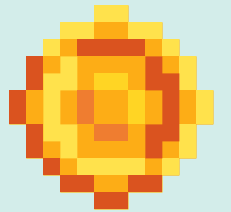
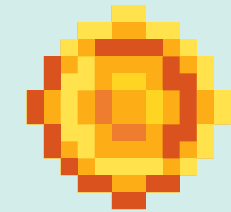
# BINARY SEARCH

Metode pencarian yang efisien digunakan pada koleksi data yang telah diurutkan, baik secara naik maupun turun.



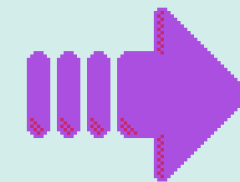


# SEQUENTIAL SEARCH

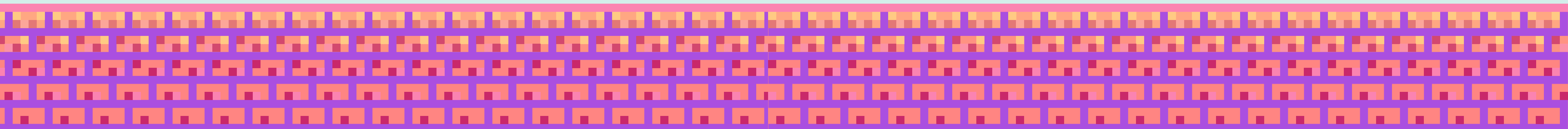


```
#include<stdio.h>
int a[100];
int i,n,cari;
int main () {
    printf("Banyak data : "); scanf("%d",&n);
    for (i=0;i<=n-1;i++){
        printf("Data ke-%d :",i+1); scanf("%d",&a[i]);
    }
    printf("Data yang ingin dicari : "); scanf("%d",&cari);

    //proses sequential search
    i=0;
    while ((i<=n-1) && (a[i]!=cari))
        i=i+1;
    if (a[i]==cari)
        printf("Data ditemukan pada urutan : %d",i+1);
    else printf("Data tidak ditemukan");
}
```



```
> cd C:\Use
Banyak data : 5
Data ke-1 :23
Data ke-2 :47
Data ke-3 :1
Data ke-4 :89
Data ke-5 :2
Data yang ingin dicari : 1
Data ditemukan pada urutan : 3
```



# BINARY SEARCH

```
#include <stdio.h>

int a[100];
int i, n, cari, idx, low, high, mid;

// Fungsi untuk melakukan bubble sort
void bubbleSort(int a[], int n) {
    int i, j;
    for (i = 0; i < n - 1; i++) {
        for (j = 0; j < n - i - 1; j++) {
            if (a[j] > a[j + 1]) {
                int temp = a[j];
                a[j] = a[j + 1];
                a[j + 1] = temp;
            }
        }
    }
}
```

```
int main() {
    printf("Banyak data: ");
    scanf("%d", &n);

    for (i = 0; i < n; i++) {
        printf("Data ke-%d: ", i + 1);
        scanf("%d", &a[i]);
    }

    // Urutkan data menggunakan bubble sort
    bubbleSort(a, n);

    printf("Data yang dicari: ");
    scanf("%d", &cari);

    // Proses binary search
    low = 0;
    high = n - 1;
    idx = -1;
```

```
while ((low <= high) && (idx == -1)) {
    mid = (low + high) / 2;

    if (a[mid] == cari)
        idx = mid;
    else if (a[mid] < cari)
        low = mid + 1;
    else
        high = mid - 1;
}

if (idx != -1)
    printf("Data ditemukan pada indeks: %d", idx);
else
    printf("Data tidak ditemukan");

return 0;
}
```

```
Banyak data: 5
Data ke-1: 23
Data ke-2: 1
Data ke-3: 100
Data ke-4: 2
Data ke-5: 78
Data yang dicari: 100
Data ditemukan pada indeks: 4
```

# FINISH LEVEL

Game telah berakhir, silahkan  
klik "Finish" untuk mengakhiri  
permainan



Finish



Repeat



# TERIMA KASIH



Sumber: Modul Praktikum Algoritma dan Pemograman, FTE