

Algoritma Searching

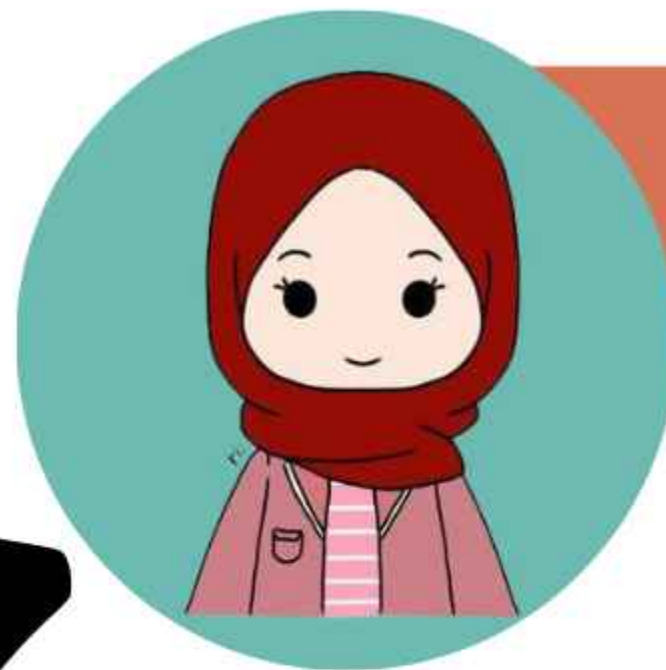
Definisi-Jenis-Implementasi



Annisa



Nazifa



Arina



Safira



Nama Anggota

Zulfa



Zahra



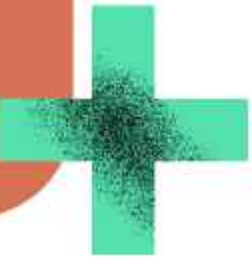
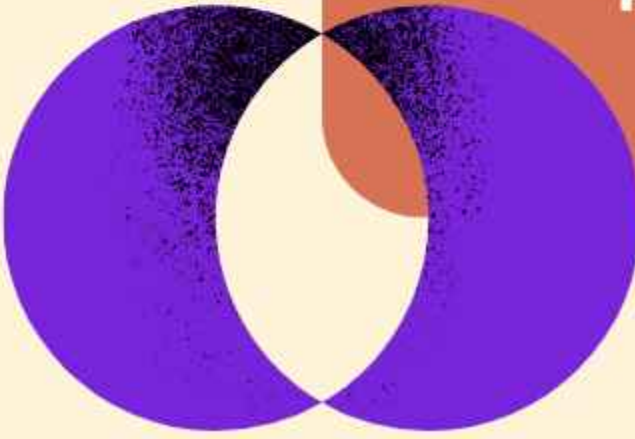
Quina

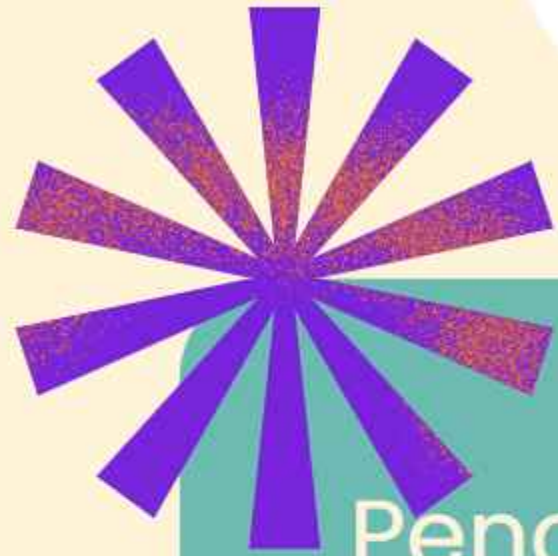




DEFINISI

Searching adalah sebuah metode pencarian guna menemukan data / informasi yang sedang dicari di dalam sebuah kumpulan data yang memiliki type data sama. Pencarian diperlukan untuk mendapatkan informasi/data dari kumpulan data yang belum diketahui.





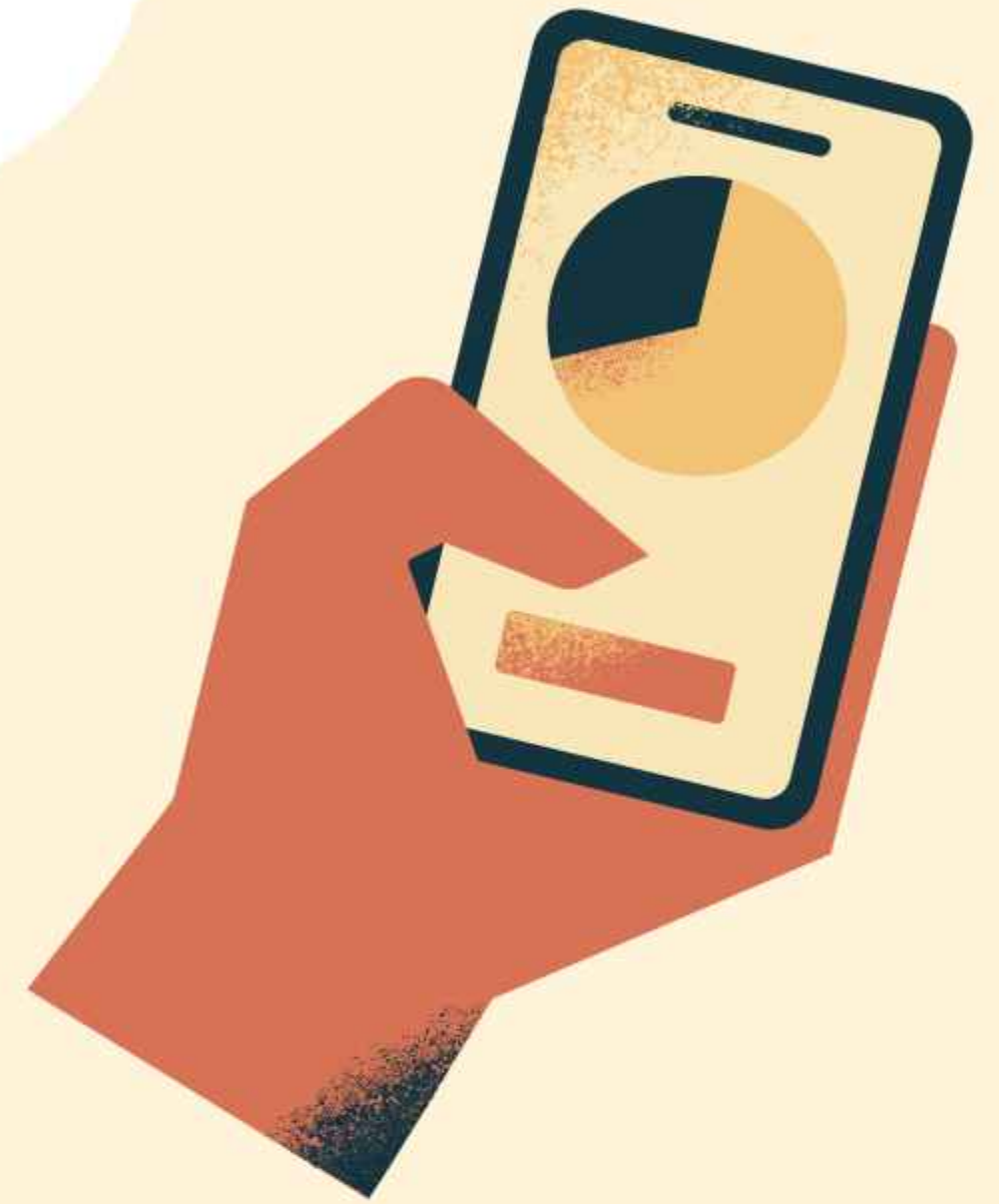
Definisi

Pencarian yang dilakukan terhadap data yang berada dalam komputer di kenal dengan pencarian internal sedangkan pencarian yang dilakukan pada media penyimpanan eksternal disebut pencarian eksternal.

Pencarian internal meliputi Pencarian Sekuensial (Sequential Search) dan Pencarian Biner (Binary Search).

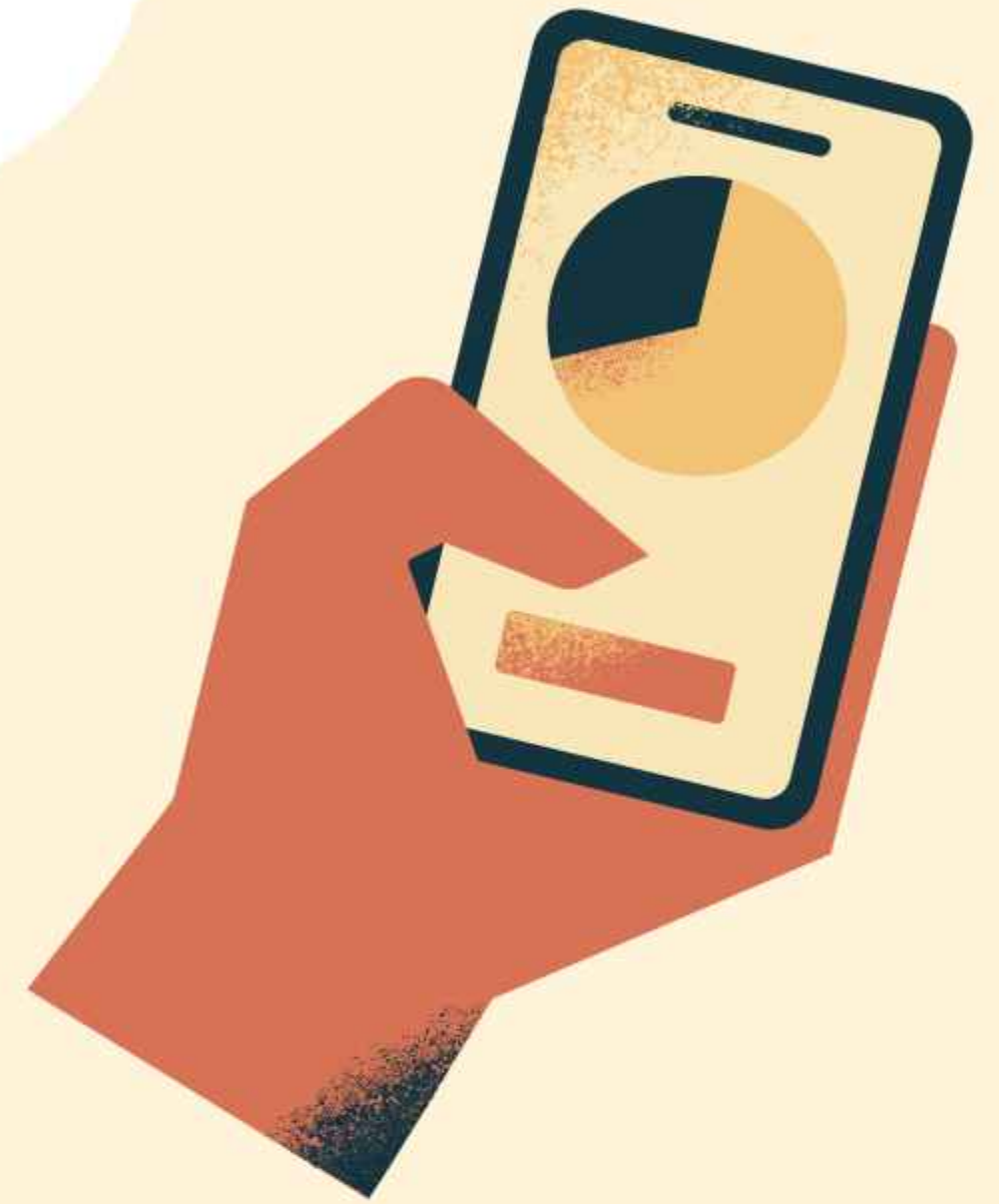
Pencarian Internal

1. Dilakukan dalam satu situs web/platform
2. Memberikan hasil yang hanya dapat ditemukan di satu situs web
3. Mempertahankan pengguna di situs web/platform



Pencarian Eksternal

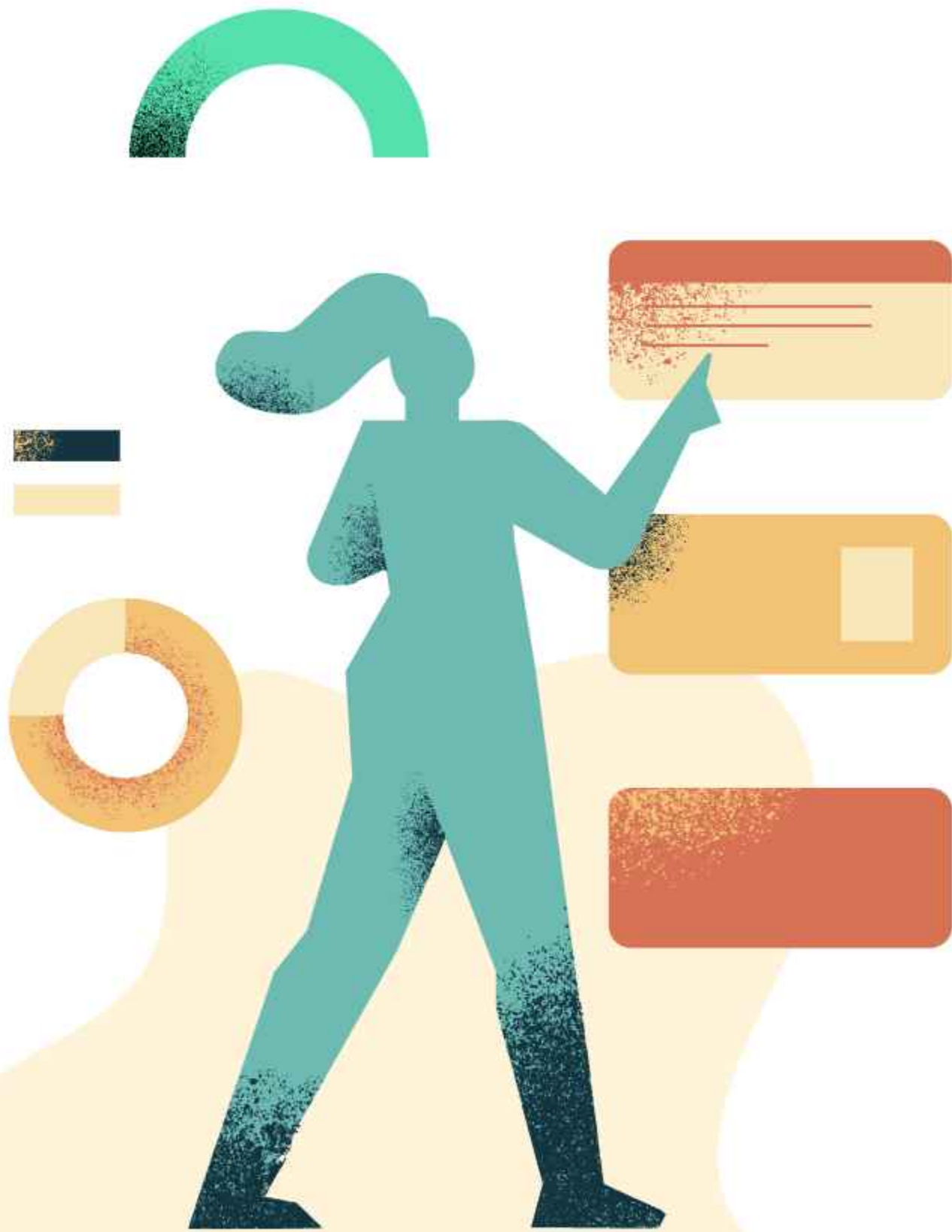
1. Dilakukan melalui mesin pencari
2. Memberikan hasil dari beberapa situs web
3. Menavigasi pengguna keluar dari situs web/platform



A. Sequential Search

Adalah metode pencarian sebuah data dari suatu kumpulan data dimana data dicari dari depan ke belakang atau dari awal sampai akhir data tanpa harus data tersebut terurut.

Konsepnya yaitu dengan melakukan perbandingan data satu-persatu secara berurutan sampai data tersebut ditemukan ataupun tidak di temukan.





B. Binary Search

Adalah metode pencarian sebuah data dari suatu kumpulan data, dimana kumpulan data tersebut harus berurutan dengan benar agar proses pencarian data bisa dilakukan.

Dalam proses pencarian data pada metode ini data akan dibagi menjadi dua bagian untuk setiap tahap pencariannya.





Contoh Implementasi Sequential Search

Contoh Implementasi Sequential Search

```
#include <stdio.h>
#include <conio.h>
#include <process.h>
main()
{ int A[10],index[10], i,j=0,k;
  //proses penginputan data
  for(i=0;i<10;i++)
  {
    printf("Data ke-%d:",i+1);
    scanf("%d",&A[i]);
  }
  //memasukkan data yang akan dicari ke dalam K
  printf("Masukkan data yang akan anda cari:");
  scanf("%d",&k);
```

```
//proses pencarian data
j=0;
for (i=0;i<10;i++)
{
  if(A[i]==k)
  {
    index[j]=i;
    j++;
  }
}

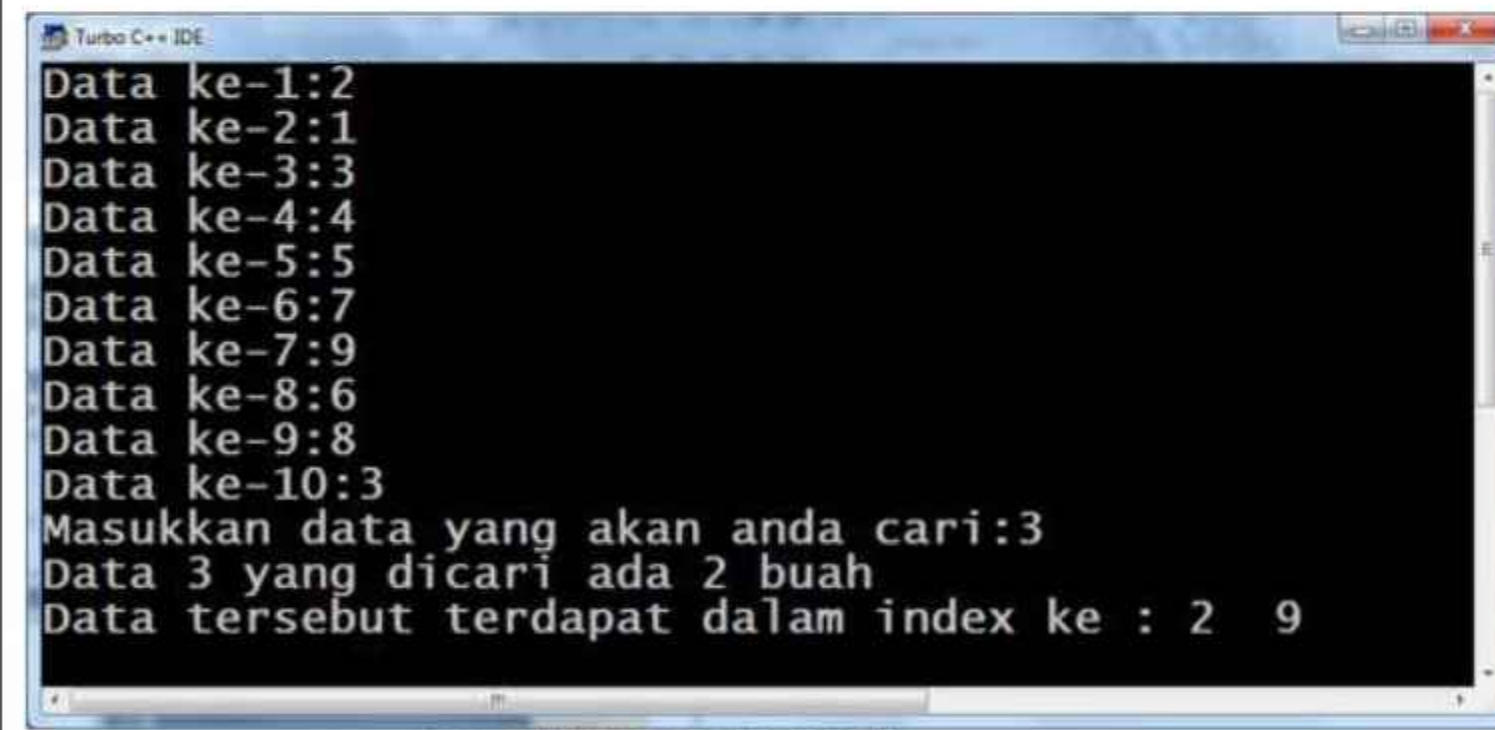
//jika data ditemukan dalam array
if (j>0)
{
  printf("Data %d yang dicari ada %d buah\n",k,j);
  printf("Data tersebut terdapat dalam index ke :");
  for(i=0;i<j;i++)
    printf(" %d ",index[i]);
  printf("\n");
}
```




Contoh Implementasi Sequential Search

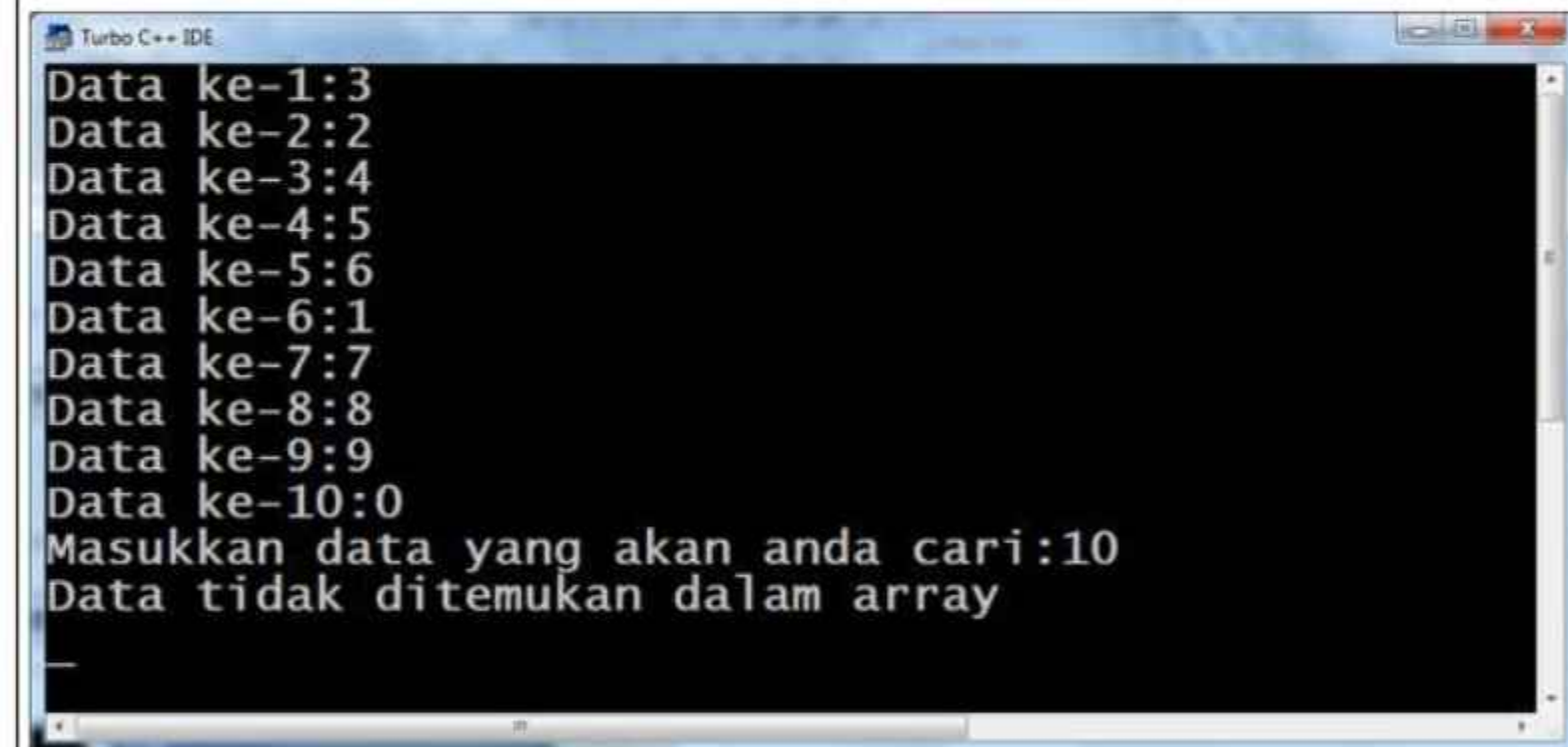
```
//jika tidak ditemukan
else
{
    printf("Data tidak ditemukan dalam array\n");
}
getch();
} //akhir program
```

Contoh Output 1:



```
Turbo C++ IDE
Data ke-1:2
Data ke-2:1
Data ke-3:3
Data ke-4:4
Data ke-5:5
Data ke-6:7
Data ke-7:9
Data ke-8:6
Data ke-9:8
Data ke-10:3
Masukkan data yang akan anda cari:3
Data 3 yang dicari ada 2 buah
Data tersebut terdapat dalam index ke : 2 9
```

Contoh Output 2 :



```
Turbo C++ IDE
Data ke-1:3
Data ke-2:2
Data ke-3:4
Data ke-4:5
Data ke-5:6
Data ke-6:1
Data ke-7:7
Data ke-8:8
Data ke-9:9
Data ke-10:0
Masukkan data yang akan anda cari:10
Data tidak ditemukan dalam array
_
```


Contoh Implementasi Binary Search

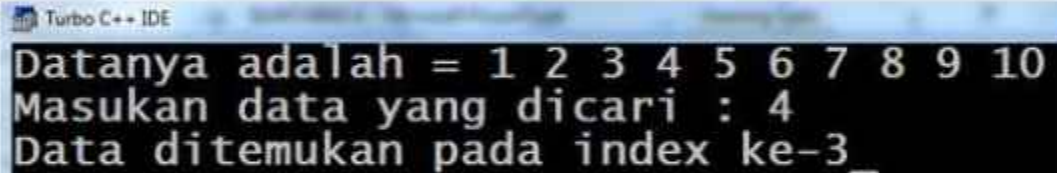
Contoh Implementasi Binary Search

```
#include <stdio.h>
#include <conio.h>
main()
{ int Ar[10] = {1,2,3,4,5,6,7,8,9,10}; // untuk proses ascending
  int i,tar, awal=0, akhir=10, tengah;
  clrscr(); printf("Datanya adalah = ");
  for(i=0;i<10;i++) printf("%d ",Ar[i]);
  printf("\nMasukan data yang dicari : "); scanf("%d",&tar);

  while (awal <= akhir)
  { tengah = (awal + akhir)/2;
    if (tar > Ar[tengah] )      // descending ubah tanda > menjadi <
    { awal = tengah + 1; }
    else if (tar < Ar[tengah]) // descending ubah tanda < menjadi >
    {akhir= tengah - 1;}
    else
    {awal = akhir +1;}
  }
```

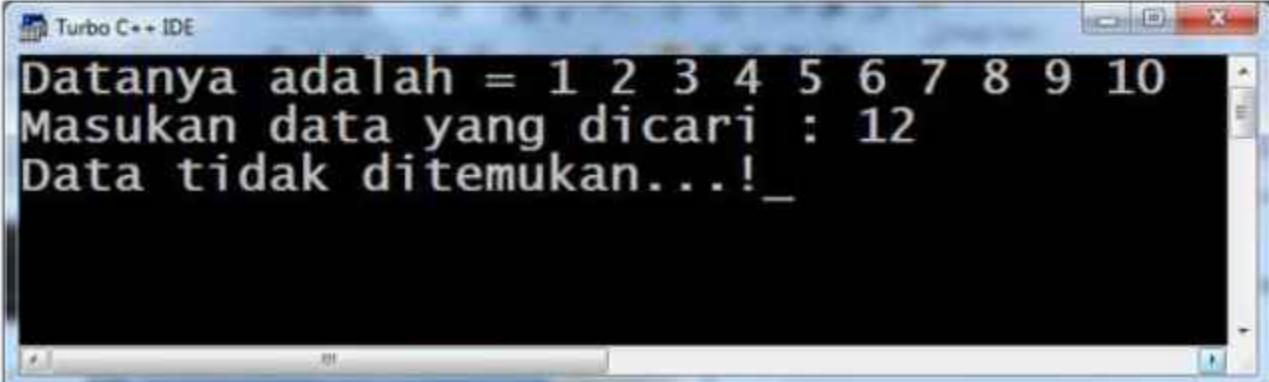
```
if (tar == Ar[tengah])
{
  printf(" Data ditemukan pada index ke-%d",tengah);
}
else
{
  printf("Data tidak ditemukan...! ");
}
getch();
}
```

Contoh Output 1 :



```
Turbo C++ IDE
Datanya adalah = 1 2 3 4 5 6 7 8 9 10
Masukan data yang dicari : 4
Data ditemukan pada index ke-3_
```

Contoh Output 2 :



```
Turbo C++ IDE
Datanya adalah = 1 2 3 4 5 6 7 8 9 10
Masukan data yang dicari : 12
Data tidak ditemukan...!_
```




**TERIMA
KASIH**