



Teknik pencarian diperlukan untuk mencari informasi/data tertentu dari kumpulan data yang ada.



Array memungkinkan untuk menyimpan nilai yang bertipe sama. Operasi pencarian yang umum dalam array adalah Linear Search dan Binary search.

INTRODUCTION



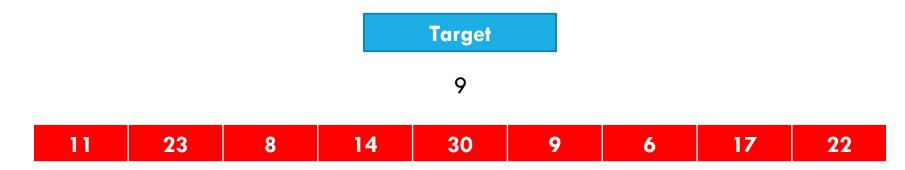
Perbedaan dari kedua teknik ini terletak pada mekanisme pencarian dan keadaan data.

LINEAR SEARCH

Pada Linear Search, Logika utama dari algoritmanya adalah melakukan pencarian data dengan melakukan perulangan pada seluruh baris data yang terdapat dari array dari elemen paling kiri ke kanan.

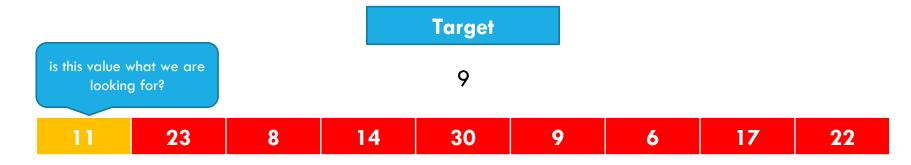
In Pseudocode:

- Jika elemen pertama merupakan data yang kita cari, pencarian selesai.
- Jika bukan, maka lanjutkan pencarian pada elemen selanjutnya.



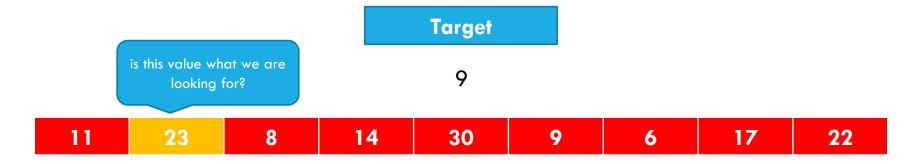
In Pseudocode:

- Jika elemen pertama merupakan data yang kita cari, pencarian selesai.
- Jika bukan, maka lanjutkan pencarian pada elemen selanjutnya.



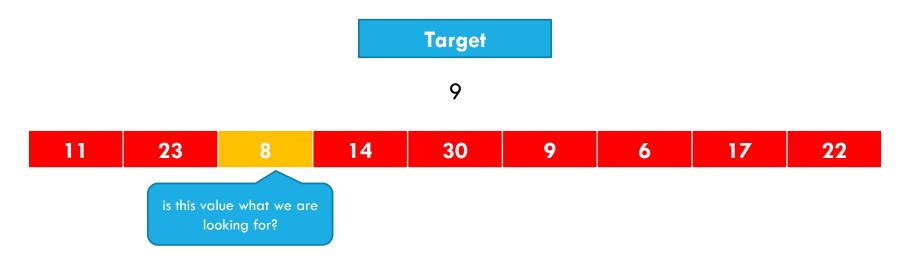
In Pseudocode:

- Jika elemen pertama merupakan data yang kita cari, pencarian selesai.
- Jika bukan, maka lanjutkan pencarian pada elemen selanjutnya.



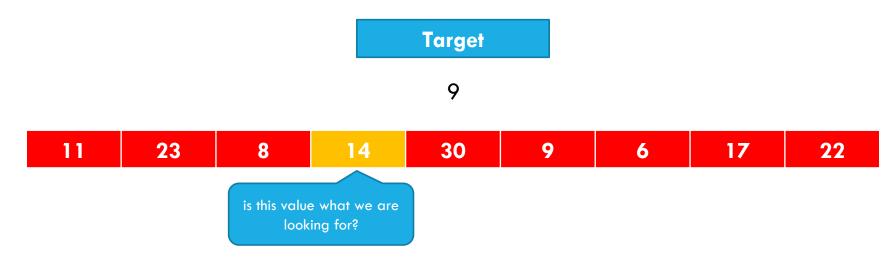
In Pseudocode:

- Jika elemen pertama merupakan data yang kita cari, pencarian selesai.
- Jika bukan, maka lanjutkan pencarian pada elemen selanjutnya.



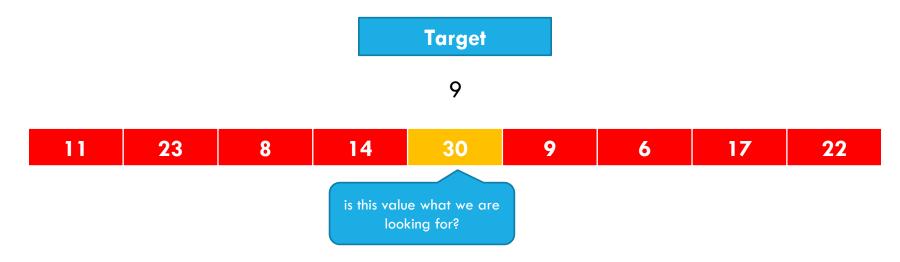
In Pseudocode:

- Jika elemen pertama merupakan data yang kita cari, pencarian selesai.
- Jika bukan, maka lanjutkan pencarian pada elemen selanjutnya.



In Pseudocode:

- Jika elemen pertama merupakan data yang kita cari, pencarian selesai.
- Jika bukan, maka lanjutkan pencarian pada elemen selanjutnya.



In Pseudocode:

- Jika elemen pertama merupakan data yang kita cari, pencarian selesai.
- Jika bukan, maka lanjutkan pencarian pada elemen selanjutnya.



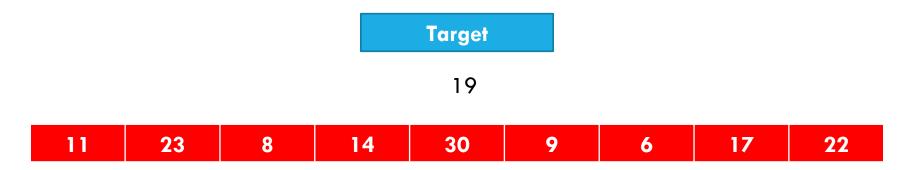
Yeah, it's the target!

In Pseudocode:

- Jika elemen pertama merupakan data yang kita cari, pencarian selesai.
- Jika bukan, maka lanjutkan pencarian pada elemen selanjutnya.

But, How if we search for the data which isn't available on array?

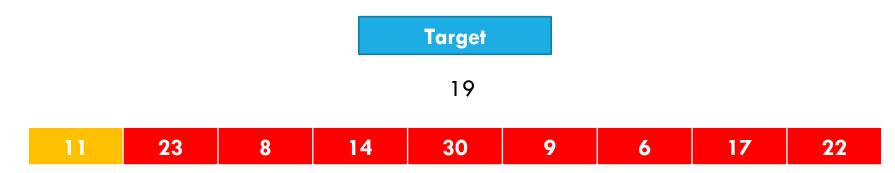




Lets Start Search!

In Pseudocode:

- Jika elemen pertama merupakan data yang kita cari, pencarian selesai.
- Jika bukan, maka lanjutkan pencarian pada elemen selanjutnya.



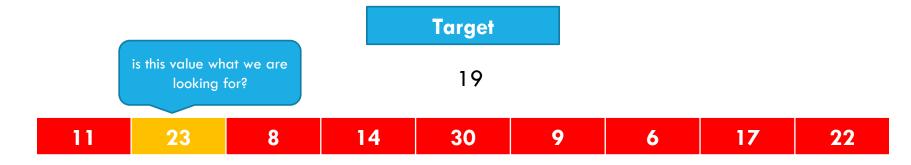
In Pseudocode:

- Jika elemen pertama merupakan data yang kita cari, pencarian selesai.
- Jika bukan, maka lanjutkan pencarian pada elemen selanjutnya.



In Pseudocode:

- Jika elemen pertama merupakan data yang kita cari, pencarian selesai.
- Jika bukan, maka lanjutkan pencarian pada elemen selanjutnya.



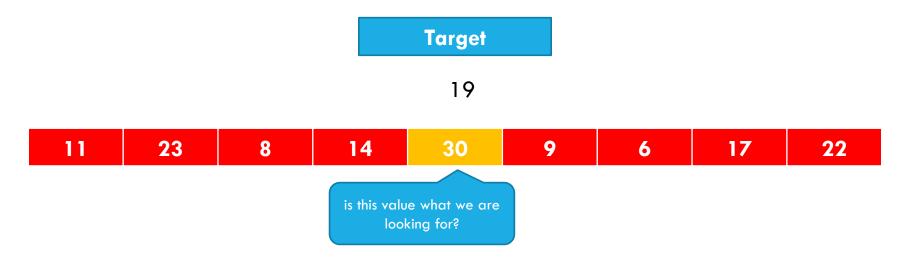
In Pseudocode:

- Jika elemen pertama merupakan data yang kita cari, pencarian selesai.
- Jika bukan, maka lanjutkan pencarian pada elemen selanjutnya.



In Pseudocode:

- Jika elemen pertama merupakan data yang kita cari, pencarian selesai.
- Jika bukan, maka lanjutkan pencarian pada elemen selanjutnya.



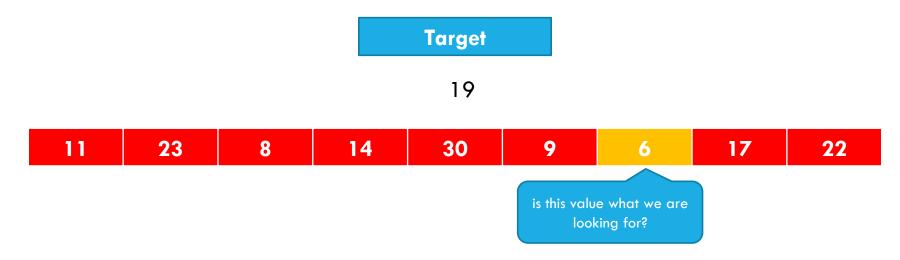
In Pseudocode:

- Jika elemen pertama merupakan data yang kita cari, pencarian selesai.
- Jika bukan, maka lanjutkan pencarian pada elemen selanjutnya.



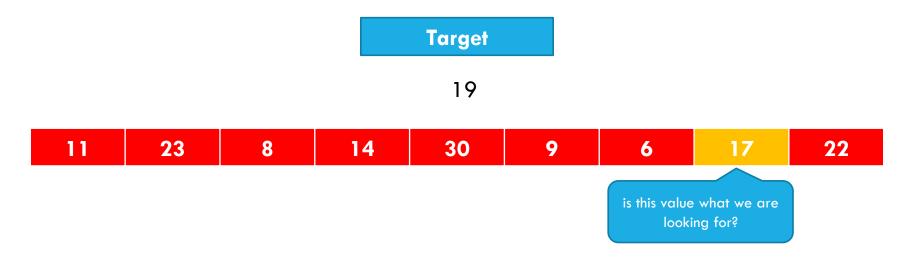
In Pseudocode:

- Jika elemen pertama merupakan data yang kita cari, pencarian selesai.
- Jika bukan, maka lanjutkan pencarian pada elemen selanjutnya.



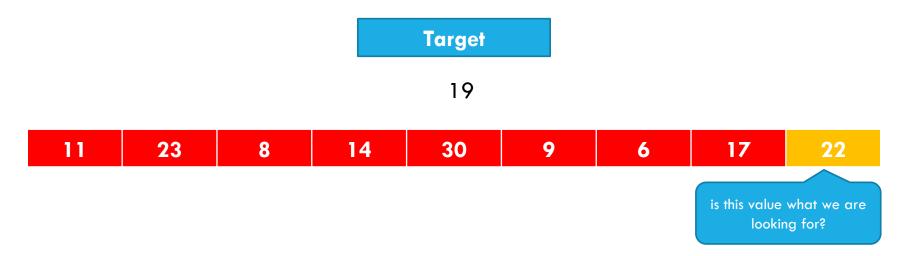
In Pseudocode:

- Jika elemen pertama merupakan data yang kita cari, pencarian selesai.
- Jika bukan, maka lanjutkan pencarian pada elemen selanjutnya.



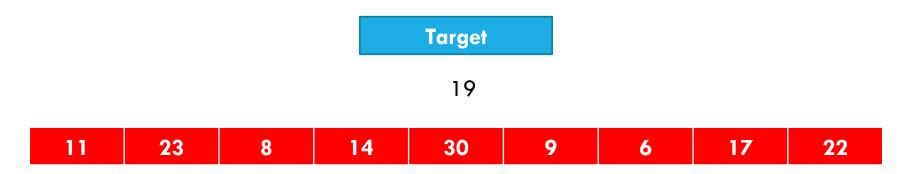
In Pseudocode:

- Jika elemen pertama merupakan data yang kita cari, pencarian selesai.
- Jika bukan, maka lanjutkan pencarian pada elemen selanjutnya.



In Pseudocode:

- Jika elemen pertama merupakan data yang kita cari, pencarian selesai.
- Jika bukan, maka lanjutkan pencarian pada elemen selanjutnya.



Well, we didn't find the data on the array, then stop

In Pseudocode:

- Jika elemen pertama merupakan data yang kita cari, pencarian selesai.
- Jika bukan, maka lanjutkan pencarian pada elemen selanjutnya.

LINEAR SEARCH

Jika hingga akhir array data tidak ditemukan maka pencarian akan berakhir.

Apakah dapat dikatakan pencarian gagal?

Jawabannya adalah tidak, pencarian linear tetap sukses akan tetapi data yang kita cari tidak berada/tidak ditemukan didalam array tersebut.

LINEAR SEARCH

Worst-case scenario: Kita harus melakukan pengecekan ke seluruh elemen array hingga elemen terakhir untuk menemukan data yang kita cari.

Best-case scenario: Kita cukup beruntung karena kita menemukan data yang dicari pada elemen pertama.

```
#include <iostream>
using namespace std;
int linearsearch(int arr[], int n, int target)
    int i;
    for (i = 0; i < n; i++)
        if (arr[i] == x)
            return i;
    return -1;
int main()
    int arr[] = { 2, 3, 4, 10, 40 };
    int target = 10;
    int n = sizeof(arr) / sizeof(arr[0]);
    int result = linearsearch(arr, n, target);
\cdot \cdot \cdot (result == -1)? cout<<"Nilai yang dicari tidak terdapat pada Array"
                 : cout<<"Nilai terdapat pada index array Ke-" <<result;</p>
  return 0;
```

SOURCE

CODE

FYI

Pencarian dapat dilakukan tidak hanya dengan menggunakan operator equal "==", akan tetapi bisa juga menggunakan operator lebih besar ">" atau lebih kecil "<".

Pengaplikasiannya adalah pada pencarian bilangan terbesar dan terkecil dari sebuah kumpulan data pada array, termasuk mencari frekuensi kemunculan suatu bilangan.

PENCARIAN BILANGAN TERBESAR & TERKECIL

```
#include <iostream>
    using namespace std;
    int main() {
    int bil[10];
    int trbesar, trkecil, i, n;
    cout<<"Masukkan banyaknya bilangan : "; cin>>n;
    for(i=0; i<n; i++){
        cout<<"Masukkan bilangan : "; cin>>bil[i];
        if(i==1){
            trbesar=bil[i];
            trkecil=bil[i];
        else if(trkecil>bil[i]){
            trkecil=bil[i];
        else if (trbesar<bil[i]){</pre>
            trbesar=bil[i];
    cout<<"\nBilangan Terbesar = "<<trbesar;</pre>
    cout<<"\nBilangan Terkecil = "<<trkecil;</pre>
    return 0;
32
```

BINARY SEARCH

Dalam Binary Search, Logika utama dari algoritma ini adalah untuk melakukan pencarian data dengan cara membagi dua (mencari titik tengah) dan mengurangi area pencarian di setiap iterasi pencariannya.

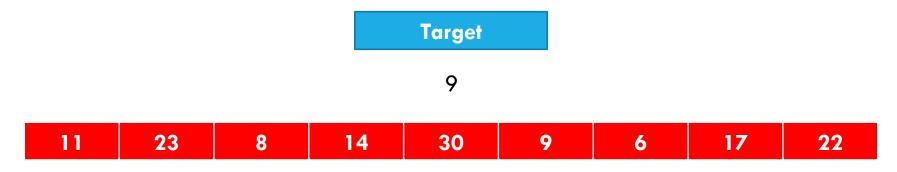
Namun, untuk memaksimalkan kinerja dari algoritma ini, kumpulan elemen array harus di urutkan terlebih dahulu sebelum pencarian dilakukan.

BINARY SEARCH

In Pseudocode:

Lakukan perulangan hingga ukuran dari (sub) array bernilai 0:

- Hitung lokasi titik tengah dari (sub) array saat ini.
- Jika data yang dicari berada pada titik tengah tersebut, maka hentikan pencarian.
- Jika bukan, jika data yang dicari nilainya lebih kecil dari nilai titik tengah, ulangi pencarian dengan mengubah titik akhir array tepat di sebelah kiri titik tengah sebelumnya.
- Jika bukan, jika data yang dicari nilainya lebih besar dari nilai titik tengah, ulangi pencarian dengan mengubah titik awal array tepat di sebelah kanan titik tengah sebelumnya.



Lets Start Search!

But don't forget to sort the array first



Cari titik tengahnya!





9 < 5, selanjutnya kita pindah titik akhir array



6 8 9 11

Ulangi, cari titik tengah!

Target	Start	End	Middle	
9	0	3	2	

6 8 9 11

is this value what we are looking for?



Yeah, it's the target!, then stop.

How if we search for the data which isn't available on array?

BINARY SEARCH

Pada akhirnya pencarian akan terus dilakukan hingga titik awal lebih besar dari titik akhir array.

Sama seperti Linear Search, Apakah dapat dikatakan **pencarian gagal** jika data tidak ditemukan ?

Jawabannya adalah tidak, pencarian tetap sukses akan tetapi data yang kita cari tidak berada/tidak ditemukan didalam array tersebut.

BINARY SEARCH

Worst-case scenario: Kita harus membagi kumpulan n-array menjadi dua secara berulang kali untuk menemukan data yang dicari, baik ketika data ditemukan atau tidak.

Best-case scenario: data yang kita cari tepat berada pada titik tengah pertama dari kumpulan elemen n-array yang kita miliki, sehingga pencarian bias langsung dihentikan.

SOURCE CODE

```
#include <iostream>
using namespace std;
int binarySearch(int arr[], int start, int end, int target)
    if (end >= start) {
                                                //selama end point > dari start point
        int mid = start + (end - start) / 2;
        // jika data yang dicari berada ditengah pada pencarian pertama
        if (arr[mid] == target)
            return mid;
        // jika data yang dicari lebih kecil nilainya dari mid
        // maka akan melakukan pencarian di kiri array. dan end point akan berubah mid-1, da
        if (arr[mid] > target)
            return binarySearch(arr, start, mid - 1, target);
        // jika data yang dicari lebih besar nilainya dari mid
        // maka akan melakukan pencarian di kanan array. dan start point akan berubah mid+1,
        return binarySearch(arr, mid + 1, end, target);
    // jika data tidak ditemukan
    return -1;
int main()
    int arr[] = { 2, 3, 4, 10, 40 };
    int n = sizeof(arr) / sizeof(arr[0]); //fungsi untuk menghitung panjang array; 1 elemen
    int target = 3;
    int result = binarySearch(arr, 0, n - 1, target);
    (result == −1) ? cout<<"Nilai yang dicari tidak terdapat pada Array"
                : cout<<"Nilai terdapat pada index array Ke-" <<result;
    return 0;
```

SEARCH ON RECORD

```
int i, k;
    #include<iostream>
                                                        bool found;
                                                        cout<<"Nilai yang dicari= ";</pre>
    using namespace std;
                                                        cin >>k:
                                                        found =false;
    struct mahasiswa
                                                        i=0;
                                                        while ((i<10)&(!found))
    int nim;
                                                            if (mhs[i].nim==k)
    char nama[25];
                                                            found=true;
    char alamat[25];
                                                            else
    };
                                                            i=i+1;
    mahasiswa mhs[10];
                                                        if(found){
    int main()
                                                        cout<<"\nnim: "<<k<<" Ditemukan ";</pre>
                                                        cout<<"\n\n";
                                                        cout<<"Detil Data Mahasiswa :";</pre>
    for (int i=0; i<2; i++)
                                                        cout<<"\n";
                                                        cout<<"Nim : "<<mhs[i].nim;</pre>
         cout<<"mahasiswa ke : "<<i+1;</pre>
                                                        cout<<"\n";
         cout<<" \n":
                                                        cout<<"Nama : "<<mhs[i].nama;</pre>
         cout<<"Nomor Induk : ";</pre>
                                                        cout<<"\n";
         cin>>mhs[i].nim;
                                                        cout<<"Alamat : "<<mhs[i].alamat;</pre>
         cout<<"Nama : ";</pre>
22
         cin>>mhs[i].nama;
23
                                                        else
         cout<<"alamat : ";</pre>
                                                        cout<<"nim "<<k<<" tidak ada dalam data";</pre>
         cin>>mhs[i].alamat;
                                                        return 0;
```



TERIMA KASIH

TUGAS MINGGU KE 12

Buatlah algoritma dan program yang dapat mengecek apakah sebuah karakter ada dalam kata yang telah input ?

Buatlah algoritma dan progam yang dapat mengecek apakah sebuah kata ada dalam kalimat yang telah input ?

Buatlah algoritma dan program yang dapat menghitung banyakntya huruf vokal dan konsonan dalam sebuah kalaimat

Modifikasi program soal nomor 3, selain menghitung banyaknya huruf vokal dan konsonan dalam sebuah kalimat, juga dapat menghitung banyaknya angka dalam kalimat tersebut.