# Binary Search

merupakan metode pencarian dimana data harus diurutkan terlebih dahulu sebelum dilakukan proses pencarian. Pada metode pencarian ini, data dibagi menjadi dua bagian untuk setiap tahap pencarian.

1. Pertama mengambil data pada posisi awal=1 dan akhir =n

2. mengambil posisi tengah dengan cara membagi dua dari jumlah awal dan akhir, misalnya data akhir adalah 9 maka data tengah adalah 9 + 1 = 10, 10/2 = 5

3. Setelah data tengah didapat, kemudian bandingkan dengan keyword atau kata kunci jika sama maka proses selesai. Jika lebih kecil, proses dilakukan kembali tetapi posisi akhir dianggap sama dengan posisi tengah -1, Jika lebih besar proses dilakukan kembali tetapi posisi awal dianggap sama dengan posisi tengah +1

4. Mengulai langkah ke dua hingga berakhir.

5. Proses akan berhenti jika data ditemukan. jika posisi awal sudah lebih besar dari posisi akhir berarti data tidak ditemukan.

misal terdapat data ={70,71,72,73,74,75,76,77,78}

mencari 75 yang terletak pada data ke 5

(1 + 9 ) / 2 = 5 data kelima nilainya 74 berarti tidak sama, maka akan dilakukan perncarian lagi. periksa apakah data ke 5 (74) < 75..?. ya . maka data tengah - 1.

(4+9)/2 = 6,5 dibulatkan menjadi 6 nilai data ke 6 adalah 75 dan data sama dengan nilai pencarian. Proses berhenti.

Contoh program :

```
#include <iostream>
using namespace std;
#include <conio.h>
#include <iomanip>

int data[7] = {1, 8, 2, 5, 4, 9, 7};
int cari;

void selection_sort()
{
	int temp, min, i, j;

	for(i=0; i<7;i++)  //pengulangan untuk membandingkan array data i

	{
		min = i; // mendeklarasikan awal bahwa i adalah min
		for(j = i+1; j<7; j++) //pengulangan data hingga data ke 7
		{
			if(data[j]<data[min])//pembandingkan nilai data i dngan
salah satuelemen array
			{
				min=j; // mendeklarasiakan bila data J lebih kecil
dari min (i) maka min berubah menjadi J bukan i
			}
		}
		temp = data[i]; // peroses penukaran dan data i yang dirubah
		data[i]  = data[min];// bila ditemukan data i lebih besar dari
		data[min] = temp; //element array lain, maka sortingan mencari
nilai terkecilkemudian menempatkannya dikiri
	}
}

void binarysearch()
{
	//searching
	int awal, akhir, tengah, b_flag = 0;
	awal = 0;
	akhir = 7;
	while (b_flag == 0 && awal<=akhir)
	{
		tengah = (awal + akhir)/2; //rumus untuk mencari nilai tengah
		if(data[tengah] == cari) //dan bila nilai tengah sudah didapat
maka akan dibandingkan dengan nilai pencarian
		{
			b_flag = 1; //bila didapat tengah didapat perogram akan
berhenti
			break;
		}
		else if(data[tengah]<cari) // bial tidak atau data pencarian
lebih besar dari data tengah, perogram akan kembali mencari dengan
membandingkan data ke awal dan akhir
			awal = tengah + 1; //rumus menari nilai awal dengan + 1
nilai tengah
		else
```

```
                  akhir = tengah -1; rumus mencari nilai akhir dengan
mengurai nilai tengah 1
      }

        if(b_flag == 1)
            cout<<"\nData ditemukan pada index ke-"<<tengah<<endl;
      else
            cout<<"\nData tidak ditemukan\n";
}

int main()
{
      cout<<"\t   'BINARY SEARCH'"<<endl;
      cout<<"\t====================="<<endl;
      cout<<"\nData           : ";
      //tampilkan data awal
      for(int x = 0; x<7; x++)
            cout<<setw(3)<<data[x];
      cout<<endl;

      cout<<"\nMasukkan data yang ingin Anda cari : ";
      cin>>cari;
      cout<<"\nData diurutkan : ";
      //urutkan data dengan selection sort
      selection_sort();
      //tampilkan data setelah diurutkan
      for(int x = 0; x<7;x++)
            cout<<setw(3)<<data[x];

      cout<<endl;

        binarysearch();

      _getche();
      return EXIT_SUCCESS;

 }
```