

1. Pencarian Binary Search

- Metode pencarian yang diterapkan pada sekumpulan data yang sudah terurut (menaik maupun menurun)
- Metode ini digunakan untuk melakukan pencarian secara cepat dengan data yang sudah terurut.

Konsep Pencarian Binary/Bagi Dua

- Pilih Indeks Kiri (Low) dan Indeks Kanan (High)
- Langkah 1:
Bagi dua elemen larik pada elemen tengah.
Elemen tengah adalah elemen dengan indeks $middle = (low + high) \div 2$.
Elemen tengah (middle), akan membagi array menjadi 2 bagian yaitu:
 - ⇒ Bagian kiri, dengan indeks **LARIK[Low .. middle-1]**
 - ⇒ Bagian Kanan, dengan indeks **LARIK[middle+1..High]**
- Langkah 2:
 - ⇒ Periksa apakah $LARIK[middle] = X$, pencarian akan dihentikan sebab X sudah ditemukan
 - ⇒ Jika $LARIK[middle] < X$, maka kita tentukan pencarian akan dilakukan disebelah kiri atau kanan.
 - Jika $LARIK[middle] < X$, maka pencarian dilakukan dibagian kiri LARIK
 - Jika $LARIK[middle] > X$, maka pencarian dilakukan di bagian kanan LARIK
- Langkah 3:
Ulangi langkah 1 sampai dengan X ditemukan, atau $low > high$ (menentukan ukuran larik sudah 0).

Contoh Program yang menggunakan Binnary search:

```
Program Binnary_Search;
uses crt;

VAR
  i,N:integer;
  max,min : integer;
  A : array[1..10] of integer;
begin
  clrscr;
  writeln('**PROGRAM BINNARY SEARCH**');writeln;
  write(' Banyaknya data ada: '); readln(N);

  for i:=1 to N do
    begin
      write('  Data ke-',i,'=');
      readln(A[i]);
    end;
  max := A[1]; min := A[1];
```

```
for i:=2 to N do
```

```
  if A[i] > max then max:=A[i]
```

```
  else if A[i] < min then min := A[i];
```

```
writeln(' ');
```

```
writeln(' Nilai MAX MIN= ',max:3, min:5);
```

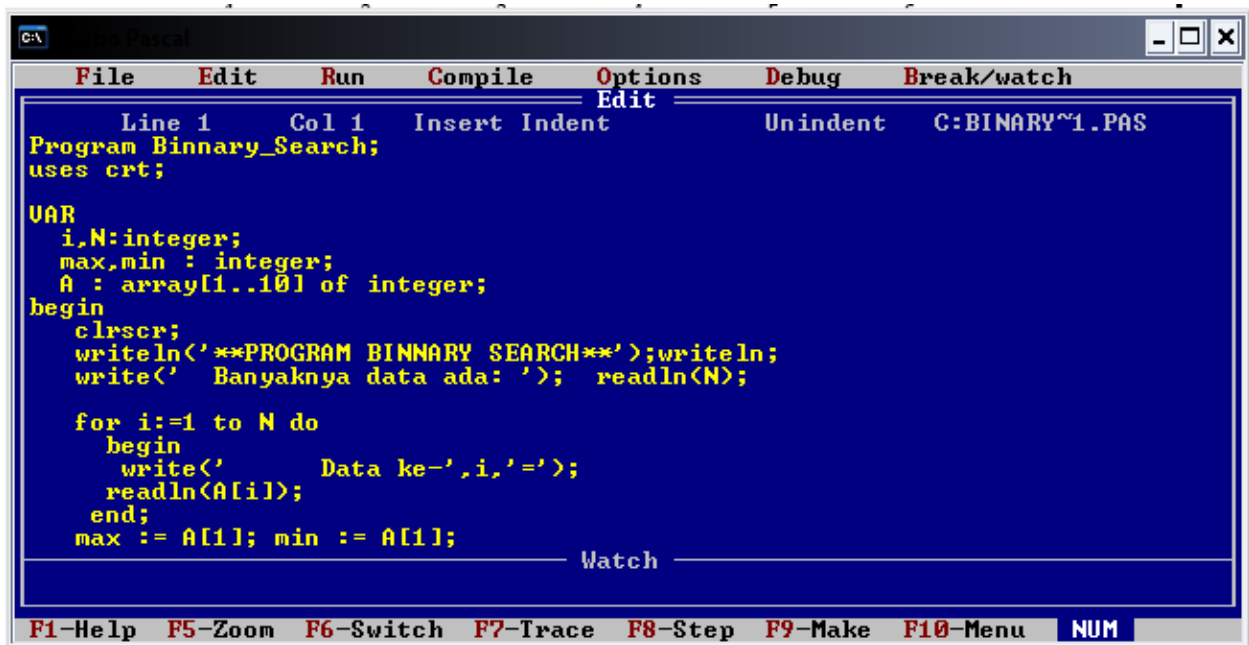
```
writeln('=====');
```

```
writeln('-----TERIMAKASIH-----');
```

```
writeln('=====');
```

```
readln;
```

```
end.
```



```
File Edit Run Compile Options Debug Break/watch
Edit
Line 1 Col 1 Insert Indent Unindent C:\BINARY~1.PAS
Program Binnary_Search;
uses crt;
UAR
i,N:integer;
max,min : integer;
A : array[1..10] of integer;
begin
clrscr;
writeln(' **PROGRAM BINNARY SEARCH**');writeln;
write(' Banyaknya data ada: '); readln(N);

for i:=1 to N do
begin
write(' Data ke-',i,'=' );
readln(A[i]);
end;
max := A[1]; min := A[1];
Watch
F1-Help F5-Zoom F6-Switch F7-Trace F8-Step F9-Make F10-Menu NUM
```

Kode Program

```
C:\
**PROGRAM BINNARY SEARCH**

Banyaknya data ada: 5
Data ke-1=90
Data ke-2=34
Data ke-3=67
Data ke-4=51
Data ke-5=23

Nilai MAX MIN= 90 23
=====
-----TERIMAKASIH-----
=====
```

Program saat dijalankan

2. Metode Bubble Sort (Gelembung)

Teknik yang diinspirasi oleh gelembung sabun yang berada dipermukaan air. Karena berat jenis gelembung lebih ringan dari pada air, maka gelembung akan naik keatas. (benda yang berat akan terbenam, benda ringan terapung).



Elemen data yang paling kecil diapungkan “diangkat keatas” melalui proses pertukaran.

Bubble Sort mengurutkan data dengan cara membandingkan elemen sekarang dengan elemen berikutnya

Algoritma Bubble Sort

1. Pengecekan mulai dari data ke-1 sampai data ke-n
2. Bandingkan data ke-n dengan data sebelumnya (n-1)
3. Jika lebih kecil maka pindahkan bilangan tersebut dengan bilangan yg ada didepannya (sebelumnya) satu persatu (n-1,n-2,n-3,....dst)
4. Jika lebih besar maka tidak terjadi pemindahan
5. Ulangi langkah 2 dan 3 s/d sort optimal.

```

Program Bubble_Sort_Asc;
uses crt;

VAR
Data : array[1..6] of integer;
i,k,n,bantu : integer;

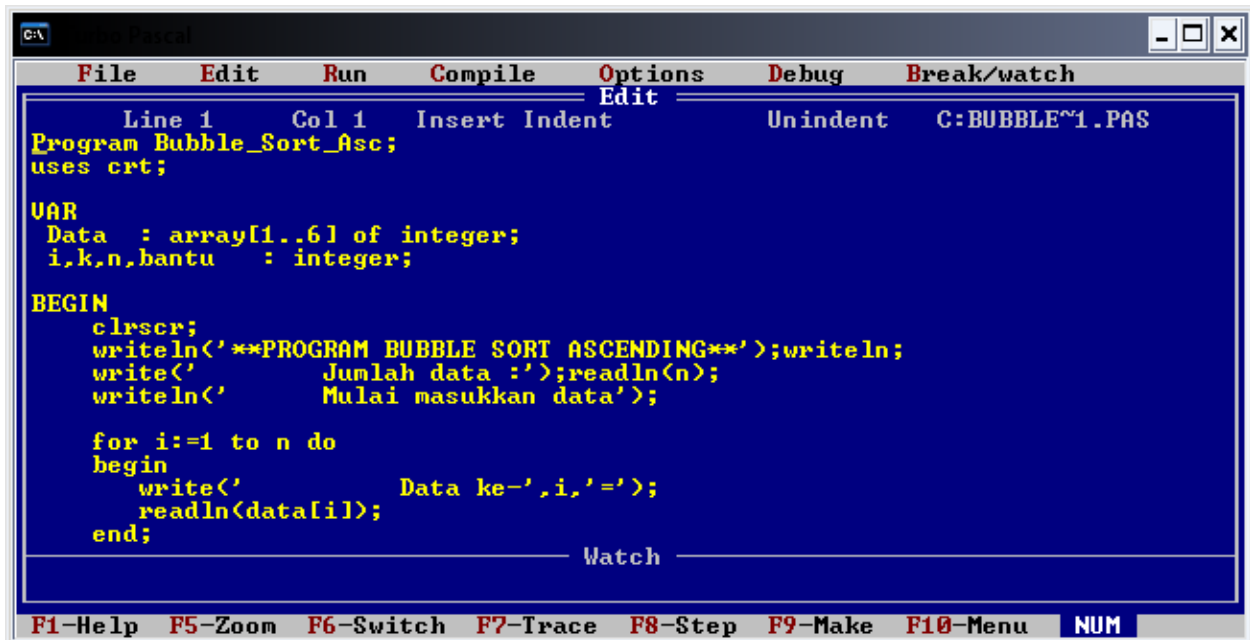
BEGIN
  clrscr;
  writeln('**PROGRAM BUBBLE SORT ASCENDING**');writeln;
  write('  Jumlah data :');readln(n);
  writeln('  Mulai masukkan data');

  for i:=1 to n do
  begin
    write('  Data ke-',i, '=');
    readln(data[i]);
  end;

  for i:=1 to N-1 do
  for k:=i+1 to N do
  begin
    if data[i] > data [k] then
    begin
      Bantu := data [i];
      data[i] := data [k];
      data [k] := Bantu;
    end;
  end;

  for i:=1 to n do
  write('(',data[i],'), '); writeln(' '); writeln(' ');
  writeln('=====');
  writeln('  TERIMAKASIH!');
  writeln('=====');
  readln;
end.

```



The screenshot shows the Turbo Pascal IDE with the following code:

```
Line 1   Col 1   Insert Indent   Unindent   C: BUBBLE~1.PAS
Program Bubble_Sort_Asc;
uses crt;

UAR
Data : array[1..6] of integer;
i,k,n,bantu : integer;

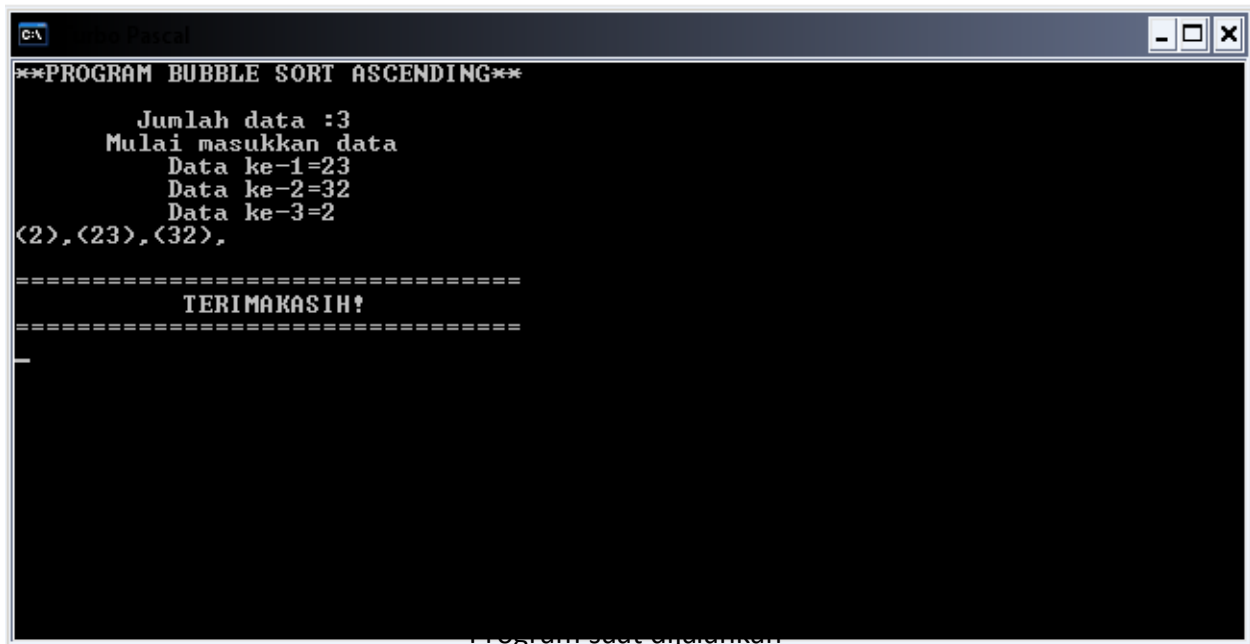
BEGIN
  clrscr;
  writeln('***PROGRAM BUBBLE SORT ASCENDING***');writeln;
  write('          Jumlah data :');readln(n);
  writeln('          Mulai masukkan data');

  for i:=1 to n do
  begin
    write('          Data ke-',i,'=');
    readln(data[i]);
  end;

  Watch
```

At the bottom of the IDE, the following function key shortcuts are listed: F1-Help, F5-Zoom, F6-Switch, F7-Trace, F8-Step, F9-Make, F10-Menu, NUM.

Kode Program



The screenshot shows the execution output of the program:

```
***PROGRAM BUBBLE SORT ASCENDING***

  Jumlah data :3
  Mulai masukkan data
    Data ke-1=23
    Data ke-2=32
    Data ke-3=2
<2>,<23>,<32>,

=====
  TERIMAKASIH!
=====
```

At the bottom of the IDE, the text "Program sudah dijalankan" is visible.