Dian

1 a Sequential Search

⇒ Langkah

(I) iterasi (i) 1 → 19 ≠ 10 → i++

(II) iterasi (i) 2 → 1 ≠ 10 → i++

(III) iterasi (i) 3 → 28 ≠ 10 → i++

(IV) iterasi (i) 4 → 5 ≠ 10 → i++

(V) iterasi (i) 5 → 20 ≠ 10 → i++

(VI) iterasi (i) 6 → 15 ≠ 10 → i++

(VII) iterasi (i) 7 → 51 ≠ 10 → i++

(VIII) iterasi (i) 8 → 13 ≠ 10 → i++

(IX) iterasi (i) 9 → 16 ≠ 10 → i++

(X) iterasi (i) 10 → 29 ≠ 10 → i++

(XI) iterasi (i) 11 → 71 ≠ 10 → i++

(XII) iterasi (i) 12 → 65 ≠ 10 → i++

(XIII) iterasi (i) 13 → 10 = 10 → data 10 ditemukan pada elemen ke-13

1b. Binary Search

Data terurut

↳ T = [1 5 5 7 10 11 13 15 16 18 19 20 28 29 35 51 65 71 87 90]

Langkah

(I) T = [1 5 5 7 10 11 13 15 16 18 19 20 28 29 35 51 65 71 87 90]

left = 1 mid = 10 right = 16

↳ 10 < 18

(II) T = [1 5 5 7 10 11 13 15 16 18 19 20 28 29 35 51 65 71 87 90]

left = 1 mid = 5 right = 10

↳ 10 = 10

2b State = Kaki maka punya 2 atribut, yaitu posisi pada koordinat kartesius dan rotasi/arah menghadap

State = Tangan maka punya 2 atribut, yaitu rotasi gerak tangan dan tangga mengepal / tidak

State = kepala maka punya 2 atribut yaitu rotasi kepala secara vertikal dan horizontal

3 N = 20

T = [19 1 28 5 20 15 51 13 16 29 71 65 10 18 87 5 90 35 7 11]

Langkah

a) (I) T = [19 1 28 5 20 15 51 13 16 29 71 65 10 18 87 5 90 35 7 11]

(II) T = [90 1 28 5 20 15 51 13 16 29 71 65 10 18 87 5 19 35 7 11]

(III) T = [90 87 28 5 20 15 51 13 16 29 71 65 10 18 1 5 19 35 7 11]

(IV) T = [90 87 71 5 20 15 51 13 16 29 28 65 10 18 1 5 19 35 7 11]

(V) T = [90 87 71 65 20 15 51 13 16 29 28 5 10 18 1 5 19 35 7 11]

(V) T = [90 87 71 65 51 15 20 13 16 29 28 5 10 18 1 5 19 35 7 11]

(VI) T = [90 87 71 65 51 35 20 13 16 29 28 5 10 18 1 5 19 15 7 11]

(VII) T = [90 87 71 65 51 35 29 13 16 20 28 5 10 18 1 5 19 15 7 11]

(VIII) T = [90 87 71 65 51 35 29 28 16 20 13 5 10 18 1 5 19 15 7 11]

(IX) T = [90 87 71 65 51 35 29 28 20 16 13 5 10 18 1 5 19 15 7 11]

(X) T = [90 87 71 65 51 35 29 28 20 19 13 5 10 18 1 5 16 15 7 11]

(XI) T = [90 87 71 65 51 35 29 28 20 19 18 5 10 13 1 5 16 15 7 11]

(XII) T = [90 87 71 65 51 35 29 28 20 19 18 16 10 13 1 5 5 15 7 11]

(XIII) T = [90 87 71 65 51 35 29 28 20 19 18 16 15 13 1 5 5 10 7 11]

(XIV) T = [90 87 71 65 51 35 29 28 20 19 18 16 15 13 1 5 5 10 7 11]

(XV) T = [90 87 71 65 51 35 29 28 20 19 18 16 15 13 11 5 5 10 7 1]

(XVI) T = [90 87 71 65 51 35 29 28 20 19 18 16 15 13 11 10 5 5 7 1]

(XVII) T = [90 87 71 65 51 35 29 28 20 19 18 16 15 13 11 10 7 5 5 1]

b. Seleetion Sort Descending

4 N = 20

T = [19 1 28 5 20 15 51 13 16 29 71 65 10 18 87 5 90 35 7 11]

a) Langkah

(I) T = [19 1 28 5 20 15 51 13 16 29 71 65 10 18 87 5 90 35 7 11]

(II) T = [1 19 5 20 15 28 13 16 29 51 65 10 18 71 5 87 35 7 11 90]

(III) T = [1 5 19 15 20 13 16 28 29 51 10 18 65 5 71 35 7 11 87 90]

(IV) T = [1 5 15 19 13 16 20 28 29 10 18 51 5 65 35 7 11 71 87 90]

(V) T = [1 5 15 13 16 19 20 28 10 18 29 5 51 35 7 11 65 71 87 90]

(VI) T = [1 5 13 15 16 19 20 10 18 28 5 29 35 7 11 51 65 71 87 90]

(VII) T = [1 5 13 15 16 19 10 18 20 5 28 29 7 11 35 51 65 71 87 90]

(VIII) T = [1 5 13 15 16 10 18 19 5 20 28 7 11 29 35 51 65 71 87 90]

(IX) T = [1 5 13 15 10 16 18 5 19 20 7 11 28 29 35 51 65 71 87 90]

(X) T = [1 5 13 10 15 16 5 18 19 7 11 20 28 29 35 51 65 71 87 90]

(XI) T = [1 5 10 13 15 5 16 18 7 11 19 20 28 29 35 51 65 71 87 90]

(XII) T = [1 5 10 13 15 5 16 18 7 11 19 20 28 29 35 51 65 71 87 90]

(XIII) T = [1 5 10 5 13 15 7 11 16 18 19 20 28 29 35 51 65 71 87 90]

(XIV) T = [1 5 5 10 13 7 11 15 16 18 19 20 28 29 35 51 65 71 87 90]

(XV) T = [1 5 5 10 7 11 13 15 16 18 19 20 28 29 35 51 65 71 87 90]

(XVI) T = [1 5 5 7 10 11 13 15 16 18 19 20 28 29 35 51 65 71 87 90]

b) Bubble Sort Ascending

5a Metode Selection sort ascending

T = [19 1 28 5 20 15 51 13 16 29 71 65 10 18 87 5 90 35 7 11]

Langkah

1 T = [19 1 28 5 20 15 51 13 16 29 71 65 10 18 87 5 90 35 4 11]

2 T = [1 19 28 5 20 15 51 13 16 29 7 1 65 10 18 87 5 90 35 7 11]

3 T = [1 5 28 19 20 15 51 13 18 2 0 71 65 10 18 87 5 90 35 7 11]

4 T = [1 5 5 19 20 15 51 13 16 29 71 65 10 18 87 28 90 35 7 11]

5 T = [1 5 5 7 20 15 51 13 16 29 71 65 10 18 87 28 90 35 19 11]

6 T = [1 5 5 7 10 15 51 13 16 29 71 65 20 18 87 28 90 35 19 11]

7 T = [1 5 5 7 10 11 51 13 16 29 71 65 20 18 87 28 90 35 19 15]

8 T = [1 5 5 7 10 11 13 51 16 29 71 65 20 18 87 28 90 35 19 15]

9 T = [1 5 5 7 10 11 13 15 16 29 71 65 20 18 87 28 90 35 19 51]

10 T = [1 5 5 7 10 11 13 15 16 29 71 65 20 18 87 28 90 35 19 51]

11 T = [1 5 5 7 10 11 13 15 16 18 71 65 20 29 87 28 90 35 19 51]

12 T = [1 5 5 7 10 11 13 15 16 18 19 65 20 29 87 28 90 35 71 51]

13 T = [1 5 5 7 10 11 13 15 16 18 19 20 65 29 87 28 90 35 71 51]

14 T = [1 5 5 7 10 11 13 15 16 18 19 20 28 29 87 65 90 35 71 51]

15 T = [1 5 5 7 10 11 13 15 16 18 19 20 28 29 35 65 90 87 71 51]

16 T = [1 5 5 7 10 11 13 15 16 18 19 20 28 29 35 51 90 87 71 65]

17 T = [1 5 5 7 10 11 13 15 16 18 19 20 28 29 35 51 65 90 87 71]

18 T = [1 5 5 7 10 11 13 15 16 18 19 20 28 29 35 51 65 71 81 90]

↳ Hasil selection sort

5b Metode Insertion Sort

T = [19 1 28 5 20 15 5 13 10 29 71 65 10 18 87 5 90 35 7 11]

Langkah

1) tukar data 1 ke 19
2) data 28 stay
3) tukar data 5 ke 19
4) tukar data 20 ke 28
5) tukar data 15 ke 19
6) data 51 stay
7) tukar data 13 ke 15
8) tukar data 16 ke 19
9) tukar data 29 ke 51
10) data 71 stay
11) tukar data 65 ke 71
12) tukar data 10 ke 13
13) tukar data 18 ke 19
14) data 87 stay
15) tukar data 5 ke 10
16) data 90 stay
17) tukar data 35 ke 51
18) tukar data 7 ke 10
19) tukar data 11 ke 13

Hasil Sorting

T = [1 5 5 7 10 11 13 15 16 18 19 20 28 29 35 51 65 71 87 90]

6 KAMUS

```
totMK = array [1..6] of integer
MK1 = array [1..3] of integer
i, k, maks, maksI = integer
type record = <hari : integer, hasMK : array [1..6] of integer>
result_survey : SEQFILE of
        (*) Reksurvey : rekaman
        (1) <EOF>
```

ALGORITMA

```
OPEN ( result_survey, Reksurvey )
repeat 5 times
        i traversal [1..6]
                totMK [i] ← totMK [i] + Reksurvey.rekaman [i]
{ mendapat hasil total survey }
i traversal [1..3]
    maks = totMK [1]
    j traversal [1..6]
        if totMK [j] > maks [j]
                maks ← totMK [j]
                maksI ← j
    if maks > 25 then
        MK1 ← maksI
    totMK [maksI] ← 0
{ mendapat array berisi kode MK yang terpilih }
```

KAMUS

```
type NIM : <nim : integer, nama : string, kdwali : string, namawali : string>
type KDMK : < kdmk : integer, namaMK : string, sks : integer >
type Nilai : <nilai : integer, nilai huruf : char >
```

ALGORITMA

input (tahun, nim, kodemk, nilai)
kodewali ← nim → kdwali
namawali ← nim → namawali
namaMK ← KDMK → namamk
nama ← nim → nama

nilaihuruf ← nilai → nilaihoruf

output (
DPNA SP TAHUN {tahun}
KDWALI : {kodewali} KODEMK : {kodemk}
NAMAWALI : {namawali} NAMAMK : {namaMK}

NO	NAMA	NIM	NILAIANGKA	NILAIHURUF
1	{nama}	{nim}	{nilai}	{nilaihoruf}