

Searching Divide And Conquere

wijanarto

Searching

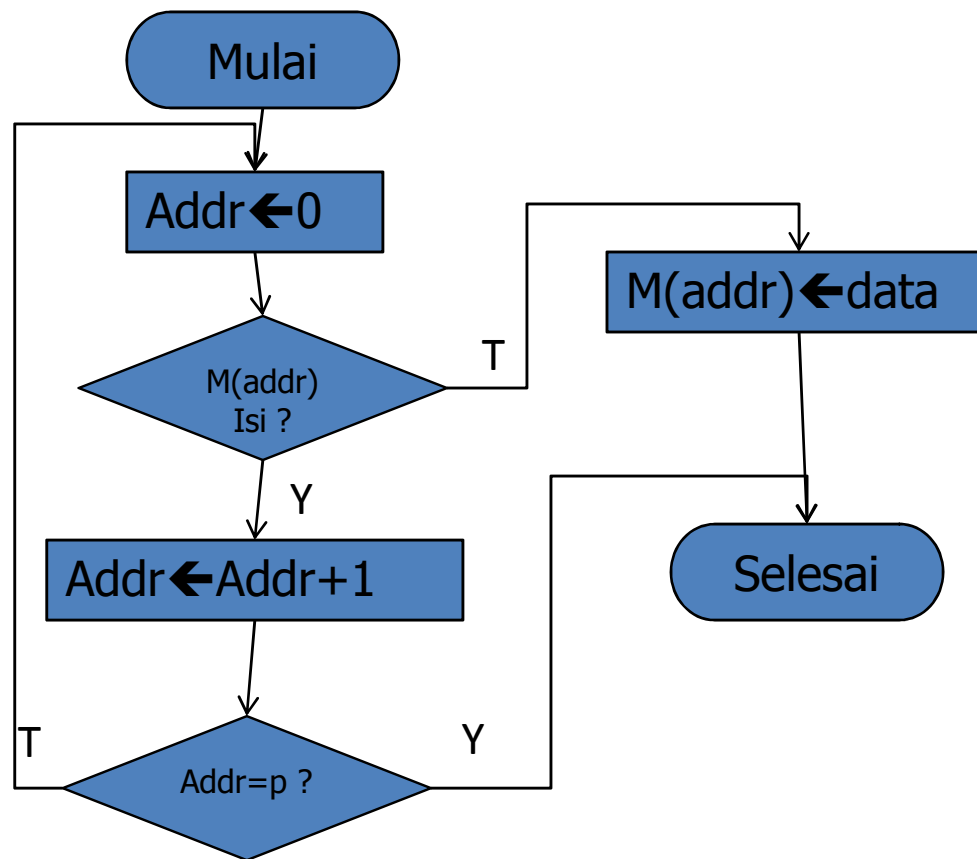
- Sekuential
- Binary Search
- Interpolation Search

Sequential Search

- Dikenal sebagai linear search
- Mencari key(info yang dicari) pada suatu data tak terurut hingga data di temukan atau data sudah mencapai akhir larik

Algoritma Insert sekuensial dalam Flowchart

Data disimpan dari
Alamat awal ke alamat
berikutnya



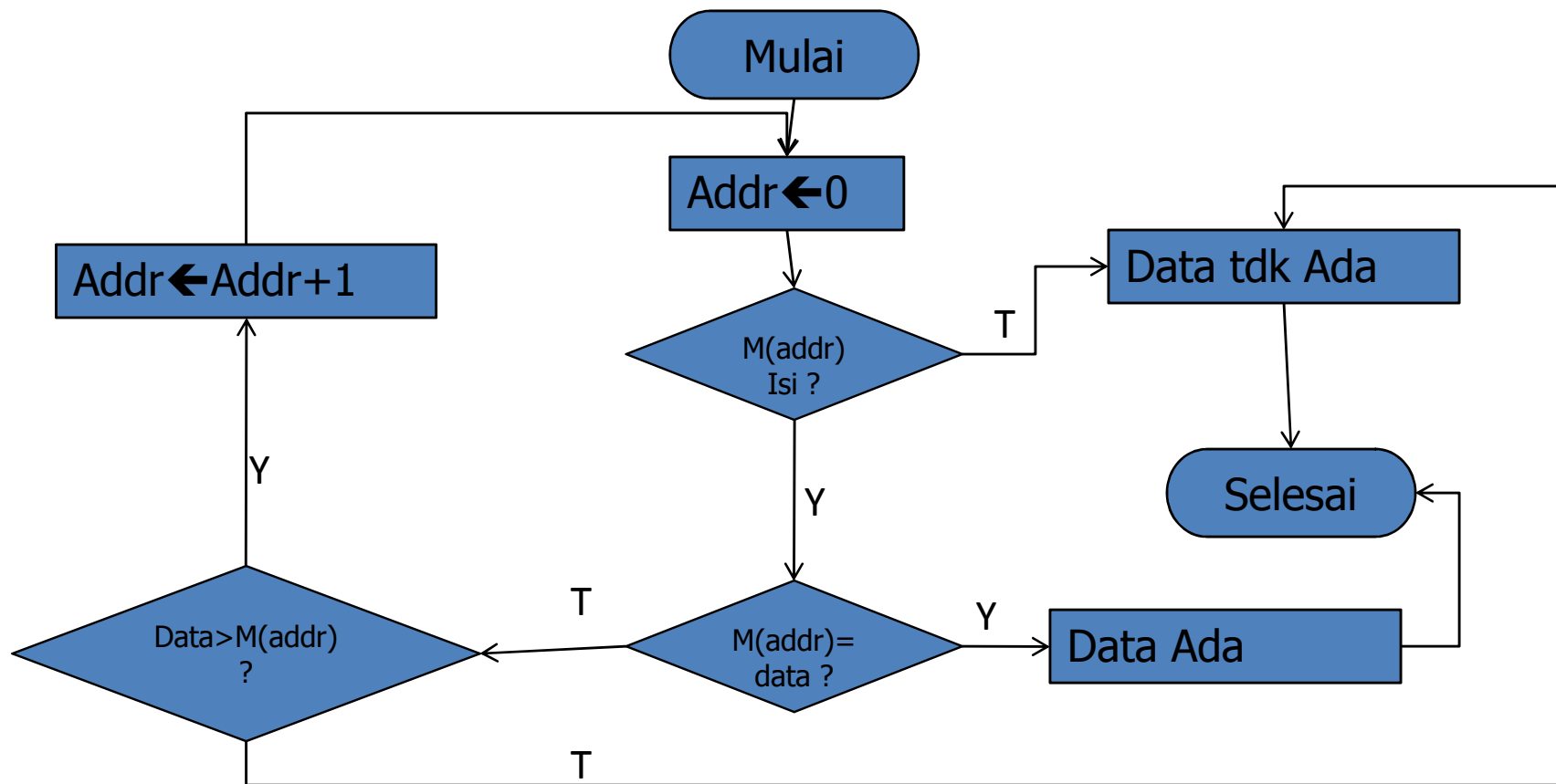
Searching Sekuensial terurut

- Data tersimpan dalam keadaan terurut
- Pencarian secara ascending atau descending
- Alamat terakhir($P1$) dari larik (P) adalah
 - $0 \leq P1 \leq P-1$

Analisa

- Dalam metode pencarian sekuensial, perbandingan yang dilakukan adalah sebanyak n kali dalam kasus worstcase, sehingga algoritma ini berorder $O(n)$
- Jika pada index i data ketemu maka dia berada pada $O(i)$
- Sehingga rata-rata pencariannya berada pada $O(n/2)$

Flowchart Searching Sekuensial



contoh

M	data
0	15
1	20
2	25
3	30
4	50
5	60
6	75
7	100

- Dicari data=50

1. Addr=0, m(addr) isi data ? \rightarrow Y
2. M(addr)=data ? \rightarrow T (15 \neq 50)
3. M(addr)>data ? \rightarrow Y (50>15)
4. Addr=addr+1 (addr=1)
5. Ulangi langkah 1

Jika di jalankan maka akan memerlukan 5 langkah untuk menemukan data 50, yaitu dr 15,20,25,30 dan 50

Binary Search

- $t[1..n]$ data tersortir menaik, $t[i] \leq t[j]$, dimana $1 \leq i \leq j \leq n$.
- X adalah elemen yang di cari dalam t , jika x tidak ada dalam t , maka kita dapat menyisipkannya ke t .
- Problem : mencari index i sedemikian rupa sehingga $1 \leq i \leq n$ dan $t[i-1] < x \leq t[i]$

Binary Search dengan DANDC

```
Binsearch (t[1..n],x) {  
    if n==0 and x> t[n] then return n+1  
    else return Binrecc(t[1..n],x)  
}
```

```
Binrecc (t[1..n],x) {  
    /*mencari x dalam array t*/  
    /*untuk t[i-1]<x ≤ t[j]*/  
    if i=j then return i  
    k ← (i+j) \ 2  
    if x ≤ t[k] then return Binrecc(t[1..k],x)  
    else return Binrecc(t[k+1..j],x)  
}
```

Binary Search dengan DANDC

- Mencari $x=12$ dalam array t

```

Binrecc (t[1..n],x) {
  if i=j then return i
     $k \leftarrow (i+j) \setminus 2$ 
    if  $x \leq t[k]$  then return Binrecc(t[1..k],x)
    else return Binrecc(t[k+1..j],x)
}
    
```

	1	2	3	4	5	6	7	8	9	10	11	
	-5	-2	0	3	8	8	9	12	12	26	31	$x \leq t[k] ?$
	i					k					j	T
							i		k		j	Y
							i	k	j			Y
							ik	j				T
								i	j			i=j berhenti

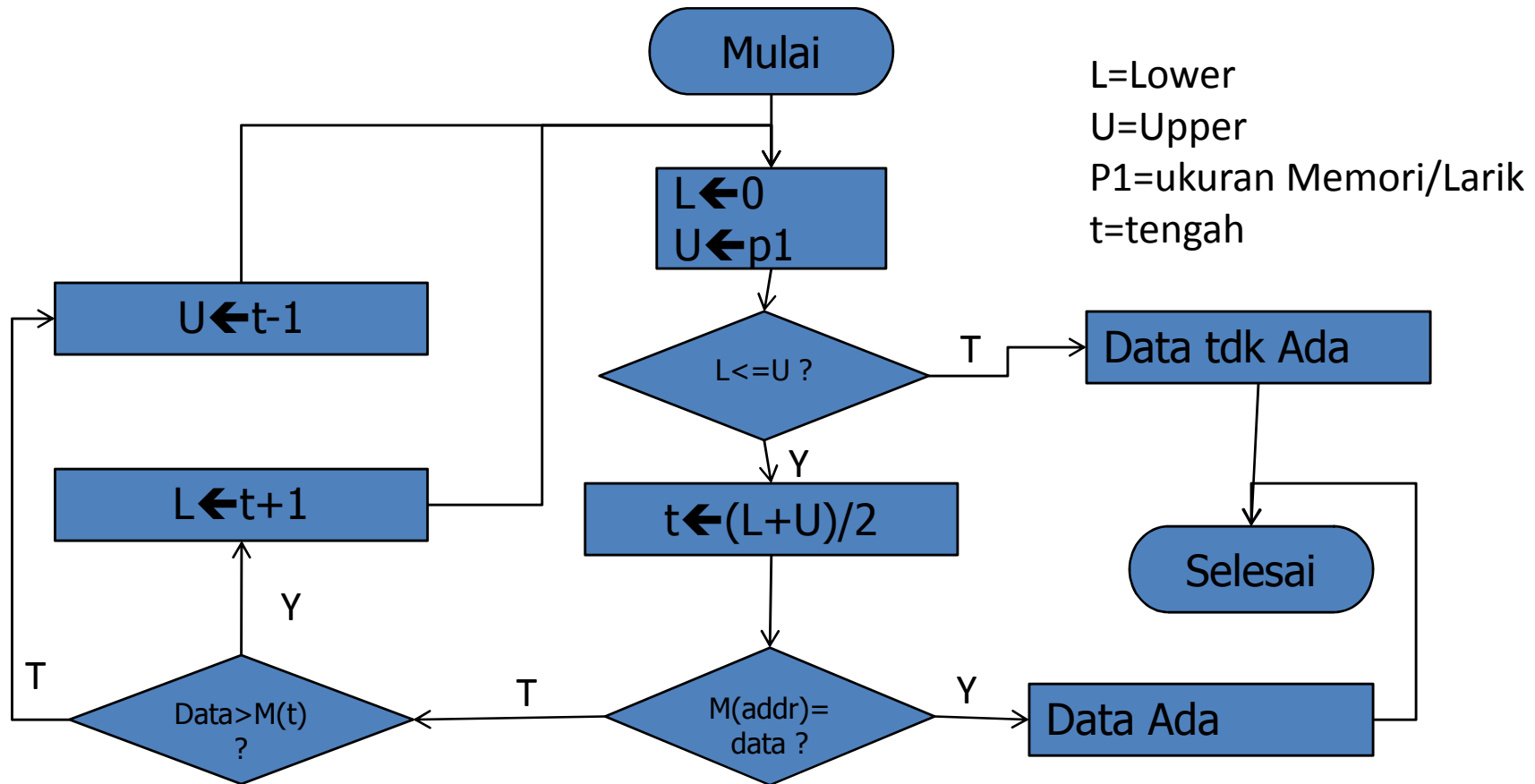
Analisa

- Dalam binary search, jumlah perbandingan yang di perlukan untuk mencari satu elemen dalam list dalah tidak lebih dari $\log_2 n$, dimana n adalah ukuran list .
- Dengan demikian algoritma binary search mempunyai kompleksitas waktu $O(n * (\log_2 n))$
- Ukuran efisiens :
 - $\max = \log_2 n$ mis $n=16$ maka 4 langkah
 - $\min = 1$
 - $\text{rata}^2 = (1+2+\dots+\log n) \log n = \frac{1}{2} \log^2 n$

Algoritma Binary Search Iterative

```
int BinSearch(int *arr, int val, int N)
{
    int low = 0; int high = N - 1; int mid;
    while ( low <= high ) {
        mid = ( low + high )/2;
        if (arr[mid]==val) {return 1; }
        else if (arr[mid] < val) {
            low = mid + 1;
        }
        else { high = mid - 1;}
    }
    return 0;
}
```

Flowchart Binary Search



contoh

x= 2 7 9 12 14 20 36 76 100 125

i= 1 2 3 4 5 6 7 8 9 10

ingin dicari a=36 dalam x

x= 2 7 9 12 14 20 36 76 100 125

i= 1 2 3 4 5 6 7 8 9 10

lower=1

a=36

upper=10

t=(lower+upper) div 2 = 5 dengan x[5]=**14**

t=5

Dicari a=36

x= 2 7 9 12 14 20 36 76 100 125

i= 1 2 3 4 5 6 7 8 9 10

a>x[t] maka lower = 6

a=36

upper=10

Lower=t+1

t=16/2=8 dengan x[8]=76

t=16/2=8

x= 2 7 9 12 14 20 36 76 100 125

i= 1 2 3 4 5 6 7 8 9 10

karena $a < x[8]$, lower=6

a=36

upper=t-1=7

upper=t-1

t=6 dengan $x[6]=20$

t=13/2=6

x= 2 7 9 12 14 20 36 76 100 125

i= 1 2 3 4 5 6 7 8 9 10

karena $a > x[6]$,

a=36

lower=t+1=6+1=7

Upper=t+1

upper=7

t=7 dengan $x[7]=36$

t=7

x= 2 7 9 12 14 20 36 76 100 125

i= 1 2 3 4 5 6 7 8 9 10

a=36

karena $a=x[7]$,

Data Di temukan

Interpolation Search

- Dikenal sebagai estimated entry search
- Data sudah terurut
- Mirip Binary Search tetapi lebih canggih sedikit (lihat flowchart)
- Pada tiap langkah pencarian algoritma membuat suatu tebakan/perkiraan (Interpolasi)

Interpolasi Search

$$k \leftarrow i + \frac{t_size - t(i)}{t(j) - t(i)} * (j - i)$$

Perbandingan Analisa

- Ketika n besar, binary search membutuhkan waktu lebih sedikit, dalam worst case, dibanding linear search. Hal ini karena di membuat perbandingan $\lg n$ dari pada n .
- Langkah yang di tunjukan dalam binary search lebih kompleks dan membutuhkan waktu yang lebih dari pada langkah dalam linear search.
- Dengan demikian untuk kasus n kecil, linear search lebih cepat.

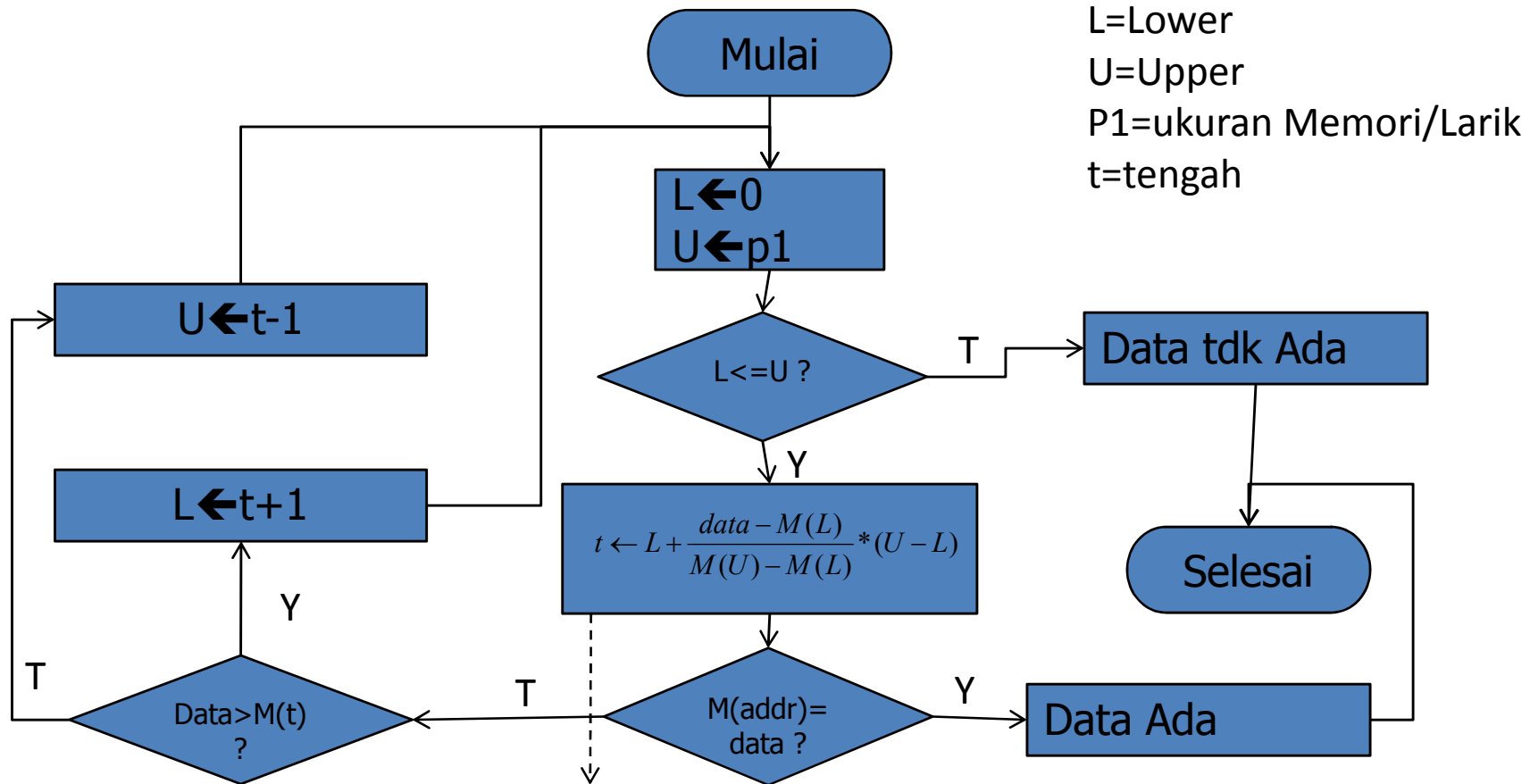
Perbandingan Analisa

- Mirip dengan hal sebelumnya, interpolation perlu melakukan langkah kompleks dalam loop , walaupun dia memerlukan lebih sedikit waktu di banding binary search.
- Pada n yang besar maka interpolasi lebih cepat di banding binary, karena pada saat melakukan interpolasi terhadap penentuan median lebih presisi, walupun langkahnya lebih kompleks

Interpolation Sequential Search

- Kombinasi Interpolasi dan sequential
- Algoritma akan melakukan pencarian secara interpolasi, jika tidak ketemu Algoritma akan mencari data secara sequen, kedepan atau belakang , tergantung arah yang di berikan

Flowchart Interpolation



Selama $L \leq U$ dan $M(L) \leq \text{data} \leq M(U)$

T merupakan integer yg di round-up atau yang terdekat (ceiling)

Algoritma

```
int isearch(typekey key, dataarray r[]){
int high, j, low ;
low = 1; high = n;
while (r[high]>= key) && (key > r[low]) do{
    j= trunc((key- r[low])/(r[high]- r{low})*(high-
        low))+low;
    if (key>r[j]) low = j+1;
    else if (key < r[j]) high := j-1
    else low =j;
} if (r[low]== key) return(low) /*** ketemu(r[low])
    ***/
else return(-1); /*** tidakketemu ***/
}
```

Contoh searching SS,BS,IS

- Data = 5,10,20,28,40,42,50,60,80,100,1000
- Cari 100,800,13 dan 40 dengan Sekuensial, Binary dan Interpolation

i	0	1	2	3	4	5	6	7	8	9	10
N	5	10	20	28	40	42	50	60	80	100	1000

- $N=11, P>N \rightarrow 0 \leq p1 \leq P-1$

Sekuensial Search

100,800,13,40

0	5
1	10
2	20
3	28
4	40
5	42
6	50
7	60
8	80
9	100
10	1000

step	addr	M(addr)	Data:M(Addr)				Jmlh step
			100	800	13	40	
1	0	5	>	>	>	>	
2	1	10	>	>	>	>	
3	2	20	>	>	<	>	3
4	3	28	>	>		>	
5	4	40	>	>		=	5
6	5	42	>	>			
7	6	50	>	>			
8	7	60	>	>			
9	8	80	>	>			
10	9	100	=	>			10
11	10	1000		<			11

Binary Search

(1)100,(2)800,(3)13,(4)40

0	5
1	10
2	20
3	28
4	40
5	42
6	50
7	60
8	80
9	100
10	1000

Data	step	L	U	t	M(t)	Data:M(t)
100	1	0	10	5	42	>
	2	5+1=6	10	8	80	>
	3	8+1=9	10	9	100	=
800	1	0	10	5	42	>
	2	5+1=6	10	8	80	>
	3	8+1=9	10	9	100	>
	4	9+1=10	10	10	1000	<
	5	10	10-1=9	9	sudah	stop

Data>M(addr) maka L=t+1 dan U tetap

Data<M(addr) maka L tetap dan U=t-1

BS Lanjutan

0	5
1	10
2	20
3	28
4	40
5	42
6	50
7	60
8	80
9	100
10	1000

Data	step	L	U	t	M(t)	Data:M(t)
13	1	0	10	5	42	<
	2	0	5-1=4	2	80	<
	3	0	2-1=1			stop
40	1	0	10	5	42	<
	2	0	5-1=4	2	20	>
	3	2+1=3	4	4	40	=

Data > M(addr) maka L=t+1 dan U tetap

Data < M(addr) maka L tetap dan U=t-1

Interpolasi Search

0	5
1	10
2	20
3	28
4	40
5	42
6	50
7	60
8	80
9	100
10	1000

Data	step	L	U	$t \leftarrow L + \frac{data - M(L)}{M(U) - M(L)} * (U - L)$	M(t)	Data:M(t)
100	1	0	10	1	10	>
	2	2	10	3	28	>
	3	4	10	5	42	>
	4	6	10	7	60	>
	5	8	10	9	100	=
800	1	0	10	8	80	>
	2	9	10	10	1000	<
	3	9	9	10		stop

Data > M(addr) maka L=t+1 dan U tetap

Data < M(addr) maka L tetap dan U=t-1

$$\begin{aligned}
 &= 0 + ((100 - 5) / (1000 - 5)) * (10 - 0) \\
 &= 0 + ((95) / (995)) * 10 \\
 &= 0 + 0.095 * 10 \\
 &= 0 + 0.95 \\
 &= 0.95 \rightarrow 1
 \end{aligned}$$

Interpolasi Search

0	5
1	10
2	20
3	28
4	40
5	42
6	50
7	60
8	80
9	100
10	1000

Data	step	L	U	$t \leftarrow L + \frac{data - M(L)}{M(U) - M(L)} * (U - L)$	M(t)	Data:M(t)
40	1	0	10	1	10	>
	2	2	10			stop
13	1	0	10	1	10	>
	2	2	10	3	28	>
	3	4	10	4	40	=

Data > M(addr) maka L=t+1 dan U tetap

Data < M(addr) maka L tetap dan U=t-1

Perbandingan Metode

Data Cari	Langkah		
	SS	BS	IS
100	10	3	5
800	11	5	3
13	3	3	2
40	5	3	3
Rata-rata = \sum/N	29/4	14/4	13/4

Jadi metode yang terbaik untuk kasus diatas adalah IS

[source](#)

[simulasi](#)

Soal

- $X=1,3,6,10,15,21,28,36,45,55,75,150,750,1500,3000$
- Cari data dengan 3 metode seq, binary dan interpolasi untuk data: 25, 21, 750 dan 1250. Hitung berapa langkah utk masing-masing data dan cari rata-rata pencariannya.
- [Solusi](#)

Soal Lagi ?

Data

27,18,29,28,39,13,16,42,17

Ditanya:

Dicari 39,50,42,20

Gunakan SS,BS,IS

Berapa rata-rata masing-masing metode