

### 13.3. Fibonacci Search.

Fibonacci Search adalah pencarian sebuah elemen dalam sebuah array satu dimensi dengan menggunakan angka Fibonacci sebagai titik-titik (index) elemen array yang isinya dibandingkan dengan nilai yang dicari (misal N).

Sama dengan Binary Search, Fibonacci Search menyatakan data sudahurut baik menaik maupun menurun. Dibandingkan dengan Binary Search, jumlah langkah pada Fibonacci Search dalam memeriksa suatu nilai adalah lebih banyak, hanya keuntungan Fibonacci Search, prosesnya hanya menggunakan operasi **tambah** dan **kurang** yang memerlukan waktu jauh lebih cepat dibandingkan dengan proses **pembagian** seperti yang digunakan pada Binary Search.

Contoh : Sudah ada array satu dimensi yang dibuat dengan int A[16].Sudah ada isinya **urut menaik**. A[0] tidak diisi, A[1] sampai dengan A[15] isininya sebagai berikut :

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	5	7	12	15	17	19	22	25	27	32	35	37	40	45	53

Susun algoritma untuk menginput sebuah nilai, (misal N) kemudian memeriksa dengan cara atau metoda **Fibonacci Search** apakah ada isi array yang sama dengan nilai N. Bila ada cetak perkataan “ADA”, sedangkan bila tidak ada maka cetak perkataan “TIDAK ADA”.

Contoh, untuk isi array yang diilustrasikan diatas.

Bila diinput untuk N = 25, maka tercetak “ADA”.

Bila diinput untuk N = 10, maka tercetak “TIDAK ADA”

**Jawab :**

Fibonacci Search, memanfaatkan nilai nilai yang ada dalam Deret Fibonacci (Fibonacci number). Deret Fibonacci adalah deret yang nilai suatu suku sama dengan jumlah nilai dua buah suku sebelumnya, atau :

$$F_n = F_{n-1} + F_{n-2} \quad \text{untuk } n \geq 2,$$

n adalah nomor suku dimulai dari nomor 0

dimana suku nomor 0 = 0, dan suku nomor 1 = 1

Nomor Suku (n) : 0 1 2 3 4 5 6 7 8 9 dan seterusnya

Deret Fibonacci : 0 1 1 2 3 5 8 13 21 34 dan seterusnya

Bagian ini yang dimanfaatkan dalam Fibonacci Search

Untuk contoh data diatas, maka nomor index yang dalam lingkaran adalah index yang nilainya sama dengan nilai deret Fibonacci.

Nomor-nomor inilah yang dipakai sebagai titik-titik utama pemeriksaan.

0	①	②	③	4	⑤	6	7	⑧	9	10	11	12	⑬	14	15
	5	7	12	15	17	19	22	25	27	32	35	37	40	45	53

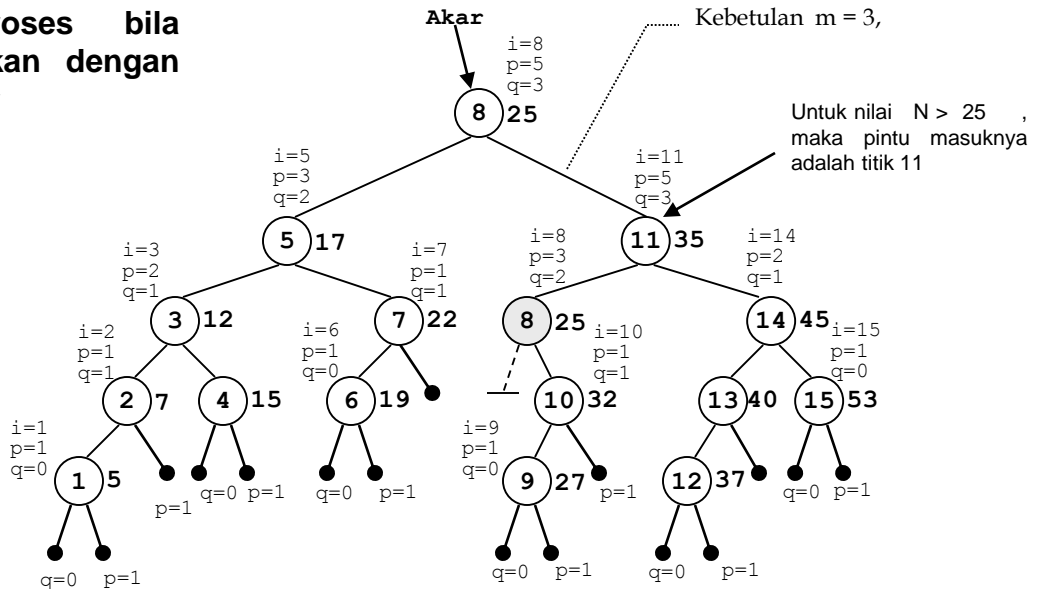
## Ilustrasi urutan proses

### I Untuk jumlah elemen $n = 15$

Data yang tersedia :

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	5	7	12	15	17	19	22	25	27	32	35	37	40	45	53

### Urutan proses bila diilustrasikan dengan pohon biner

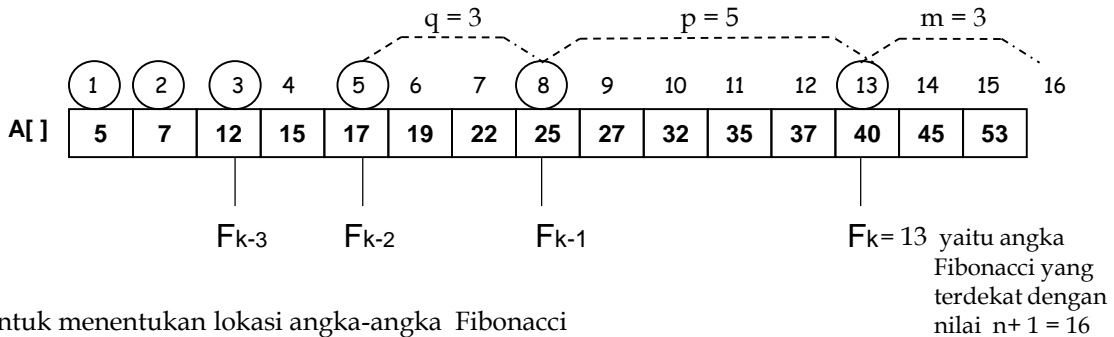


Konsep diatas, bila diterapkan dalam langkah-langkah algoritma dapat dibuat sebagai berikut :

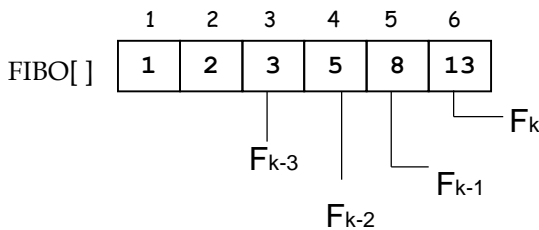
**TAHAP - I.** Menyiapkan parameter-parameter.

1. Cari angka Fibonacci (Fk) yang paling dekat dengan  $n + 1$  (misal  $n = 15$ )

Karena  $n = 16$ , maka 13 adalah angka Fibonacci yang paling mendekati  $n + 1$ .



Untuk menentukan lokasi angka-angka Fibonacci diatas, maka angka angka tersebut harus disimpan dalam array (misal FIBO[ ]), sehingga akan terlihat sebagai berikut :



2. Cari nilai  $F_{k-1}$  dan  $F_{k-2}$  dan  $F_{k-3}$  nyatakan nilai **i** dengan  $F_{k-1}$ ,  
**p** dengan nilai  $F_{k-2}$   
**q** dengan nilai  $F_{k-3}$   
sehingga didapat:

$$\begin{aligned} F_k &= 13 \\ F_{k-1} &= 8 \longrightarrow i = 8 \\ F_{k-2} &= 5 \longrightarrow p = 5 \\ F_{k-3} &= 3 \longrightarrow q = 3 \end{aligned}$$

3. Cari nilai increament (penambah) **m** dengan rumus :

$$\begin{aligned} m &= (n+1) - F_k \\ &= 15 + 1 - 13 \\ &= 3 \end{aligned}$$

Catatan :  $m \geq 0$

**m** ini digunakan hanya bila untuk pertama kali  $N > A[i]$

#### Menyiapkan data

```
int i, j, F0, F1, Fibo, n, m, N, Flag;
int FK, FK1, FK2, FK3, s, p, q, t;
int A[16] = { 0, 5, 7, 12, 15, 17,
              19, 22, 25, 27, 32,
              35, 37, 40, 45, 53 };
int FIBO[8];
```

#### Mengisi FIBO dan menghitung : **i, p, q, dan m**

```
clrscr();
n = 15;
F0 = 1; F1 = 1; Fibo = 1;
j = 1;
while(Fibo <= n+1)
{ FIBO[j] = Fibo;
  Fibo = F0+F1; F0=F1; F1=Fibo;
  j++; }
s = j - 1;

FK = FIBO[s];
FK1 = FIBO[s-1]; i = FK1;
FK2 = FIBO[s-2]; p = FK2;
FK3 = FIBO[s-3]; q = FK3;

m = (n+1) - FK;
```

**i** sebagai titik awal pencarian.

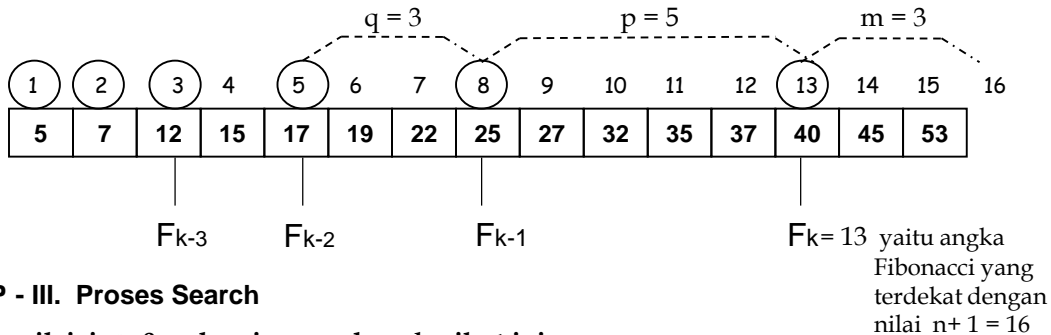
## TAHAP - II.

Menentukan nilai index  $i$  sebagai titik awal pemeriksaan (search)  
Sebelumnya sudah didapat nilai  $i = 8$ , sehingga  $A[i]$  adalah 25,  
Periksa nilai yang akan dicari (misal  $N$ )

1. Bila  $N$  lebih besar dari  $A[i]$ , maka ubah nilai  $i$  menjadi

$$\begin{aligned}i &= i + m \\ &= 8 + 3 = 11\end{aligned}$$

2. Bilai  $N$  tidak lebih besar dari  $A[i]$ , nilai  $i$  tetap = 8.



## TAHAP - III. Proses Search

Selama nilai  $i < 0$ , ulangi proses loop berikut ini :

Bila  $N == A[i]$

berarti  $N$  ditemukan pada langkah ini.  
dan proses dibuat selesai.

Bila  $N < A[i]$ ,

berarti harus mencari ke sisi kiri,  
cari nilai  $i$  yang baru dengan cara :  
- Bila  $q == 0$ , berarti nilai  $N$  tidak ditemukan dan  
proses dibuat selesai dengan cara  
membuat nilai  $i = 0$ ,  
- Bila  $q != 0$ , maka  $i = i - q$  (mundur sebanyak  $q$ )  
 $t = p$   
 $p = q$   
 $q = t - q$   
kemudian ulangi mengerjakan loop

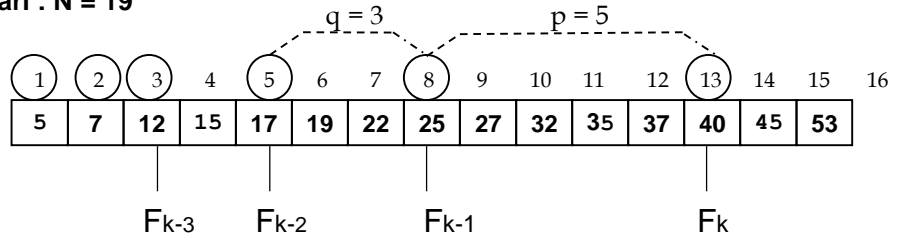
Bila  $N > A[i]$ ,

berarti harus mencari ke sisi kanan,  
cari nilai  $i$  yang baru dengan cara:  
- Bila  $p == 1$ , berarti  $N$  tidak ditemukan, dan  
proses dibuat selesai dengan cara  
membuat nilai  $i = 0$   
- Bila  $p != 1$ , maka  $i = i + q$  (maju sebanyak  $q$ )  
 $p = p - q$   
 $q = q - p$   
kemudian ulangi mengerjakan loop

```
Flag = 0;
while( i != 0 && Flag == 0)
{
    if(N == A[i]) Flag = 1;
    else
    {
        if(N < A[i])
        {
            if(q == 0) i = 0;
            else { i = i - q;
                    t = p;
                    p = q;
                    q = t - q;
                }
        }
        else
        {
            if(p == 1) i = 0;
            else
            {
                i = i + q;
                p = p - q;
                q = q - p;
            }
        }
    }
    if(Flag == 1)
        printf("\nFOUND");
    else printf("\nNOT FOUND");
}
```

## Contoh Proses : untuk jumlah elemen $n = 15$ .

1 Untuk nilai yang dicari :  $N = 19$



TAHAP - I didapat :

$$F_k = 13$$

$$F_{k-1} = 8 \rightarrow i = 8$$

$$F_{k-2} = 5 \rightarrow p = 5$$

$$F_{k-3} = 3 \rightarrow q = 3$$

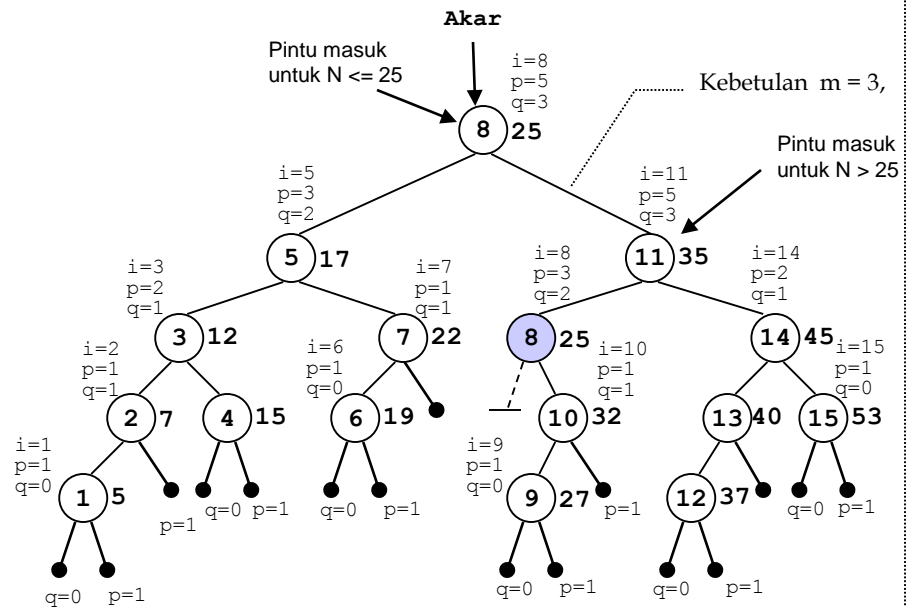
$$m = (15 + 1) - 13 = 3$$

TAHAP - II.

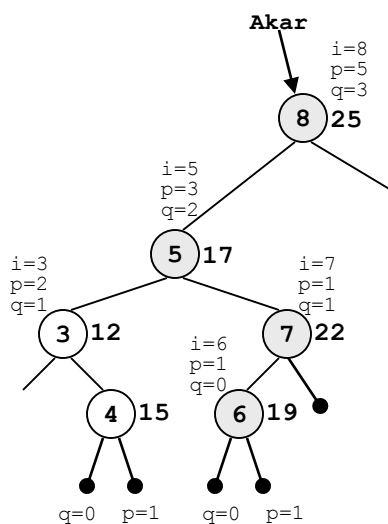
$$N = 19$$

$$A[i] = A[8] = 25$$

$$N < 25 \rightarrow i \text{ tetap} = 8$$



TAHAP - III Proses



$$i = 8, A[8] = 25$$

Langkah - 1.  $N < 25 \rightarrow$  cari ke kiri :

$$\begin{aligned} 19 \quad q \neq 0 &\rightarrow i = i - q \rightarrow i = 8 - 3 = 5 \\ t = p &\rightarrow t = 5 \\ p = q &\rightarrow p = 3 \\ q = t - q &\rightarrow q = 5 - 3 = 2 \end{aligned}$$

$$i = 5, A[5] = 17$$

Langkah - 2.  $N > 17$  cari ke kanan :

$$\begin{aligned} 19 \quad p \neq 1 &\rightarrow i = i + q \rightarrow i = 5 + 2 = 7 \\ p = p - q &\rightarrow p = 3 - 2 = 1 \\ q = q - p &\rightarrow q = 2 - 1 = 1 \end{aligned}$$

$$i = 7, A[7] = 22$$

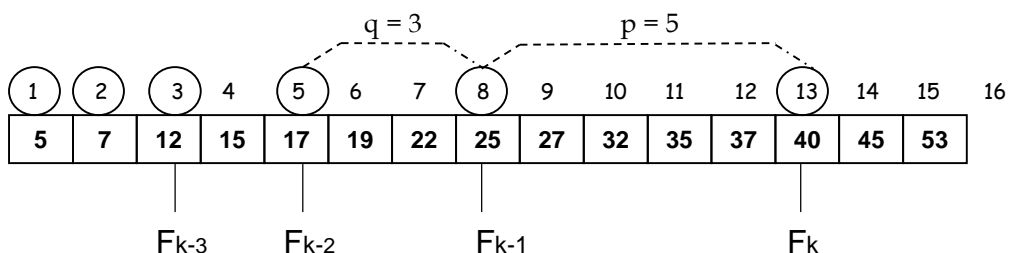
Langkah - 3.  $N < 22 \rightarrow$  cari ke kiri :

$$\begin{aligned} 19 \quad q \neq 0 &\rightarrow i = i - q \rightarrow i = 7 - 1 = 6 \\ t = p &\rightarrow t = 1 \\ p = q &\rightarrow p = 1 \\ q = t - q &\rightarrow q = 1 - 1 = 0 \end{aligned}$$

$$i = 6, A[6] = 19$$

Langkah - 4.  $N = 19 \rightarrow$  Ditemukan nilai yang = 19 (N) pada langkah ke - 4.

## 2 Untuk nilai yang dicari : $N = 20$



TAHAP - I didapat :

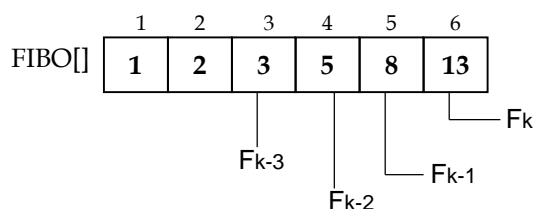
$$F_k = 13$$

$$F_{k-1} = 8 \rightarrow i = 8$$

$$F_{k-2} = 5 \rightarrow p = 5$$

$$F_{k-3} = 3 \rightarrow q = 3$$

$$m = (15 + 1) - 13 = 3$$



TAHAP - II.  $N = 20$

$$A[i] = A[8] = 25$$

$$N < 25 \rightarrow i \text{ tetap} = 8$$

TAHAP - III Proses mencari nilai 20

$$i = 8, A[8] = 25$$

Langkah - 1.  $N < 25 \rightarrow$  cari ke kiri :

$$\begin{aligned} 20 & q \neq 0 \rightarrow i = i - q \rightarrow i = 8 - 3 = 5 \\ & t = p \rightarrow t = 5 \\ & p = q \rightarrow p = 3 \\ & q = t - q \rightarrow q = 5 - 3 = 2 \end{aligned}$$

$$i = 5, A[5] = 17$$

Langkah - 2.  $N > 17 \rightarrow$  cari ke kanan :

$$\begin{aligned} 20 & p \neq 1 \rightarrow i = i + q \rightarrow i = 5 + 2 = 7 \\ & p = p - q \rightarrow p = 3 - 2 = 1 \\ & q = q - p \rightarrow q = 2 - 1 = 1 \end{aligned}$$

$$i = 7, A[7] = 22$$

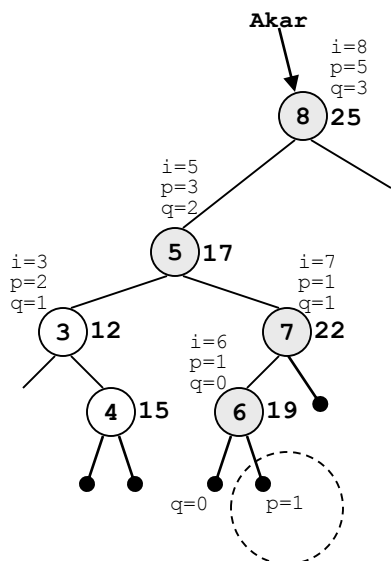
Langkah - 3.  $N < 22 \rightarrow$  cari ke kiri :

$$\begin{aligned} 20 & q \neq 0 \rightarrow i = i - q \rightarrow i = 7 - 1 = 6 \\ & t = p \rightarrow t = 1 \\ & p = q \rightarrow p = 1 \\ & q = t - q \rightarrow q = 1 - 1 = 0 \end{aligned}$$

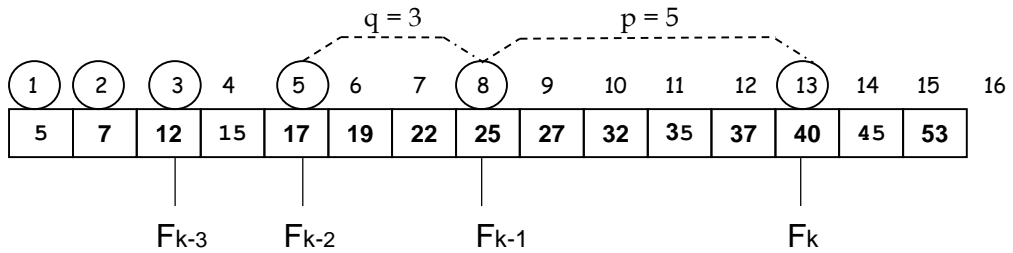
$$i = 6, A[6] = 19$$

Langkah - 4.  $N > 19 \rightarrow$  cari ke kanan :

$$\begin{aligned} 20 & p = 1 \rightarrow 20 \text{ Tidak ditemukan} \\ & \text{sampai dengan langkah} \\ & \text{terakhir (pada langkah ke-4)} \\ & \text{dan proses selesai.} \end{aligned}$$



3 Untuk nilai yang dicari :  $N = 45$



TAHAP - I sama dengan proses sebelumnya didapat :

$$\begin{aligned} i &= 8 \\ p &= 5 \\ q &= 3 \\ m &= 3 \end{aligned}$$

TAHAP - II.  $N = 45$

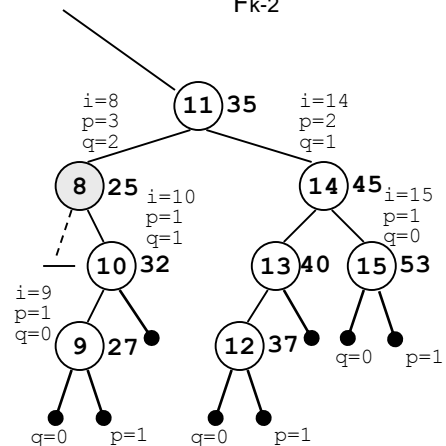
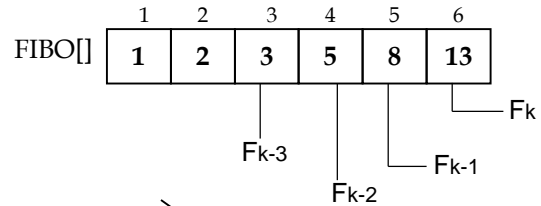
$$A[i] = A[8] = 25$$

$N > 25 \rightarrow i$ , dibuat menjadi :

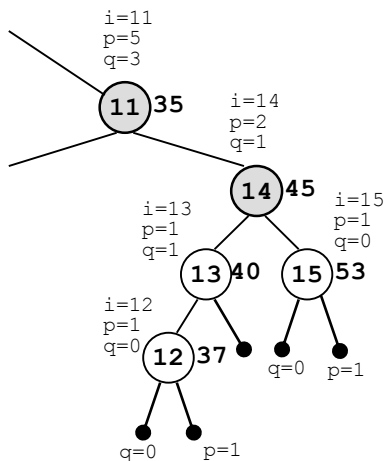
$$i = i + m = 8 + 3 = 11$$

Titik pemeriksaan awal = titik 11

yaitu yang nilainya = 35



TAHAP - III Proses mencari nilai 45



$$i = 11, A[11] = 35$$

Langkah - 1.  $N > 35 \rightarrow$  cari ke kanan :

45

$$p \neq 1 \rightarrow i = i + q \rightarrow i = 11 + 3 = 14$$

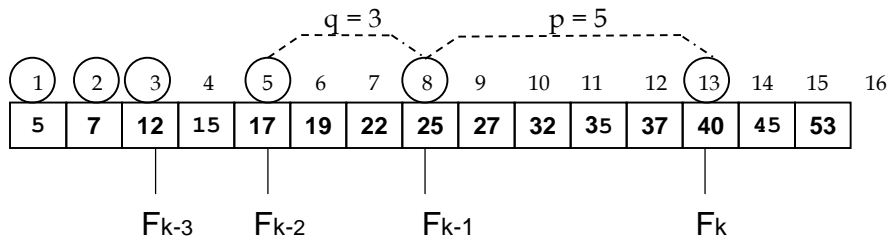
$$p = p - q \rightarrow p = 5 - 3 = 2$$

$$q = q - p \rightarrow q = 3 - 2 = 1$$

$$i = 14, A[14] = 45$$

Langkah - 2.  $N = 45 \rightarrow$  45 Ditemukan pada langkah-2

4 Untuk nilai yang dicari :  $N = 27$



TAHAP - I sama dengan proses sebelumnya didapat :

$i = 8$   
 $p = 5$   
 $q = 3$   
 $m = 3$

TAHAP - II.  $N = 27$

$A[i] = A[8] = 25$

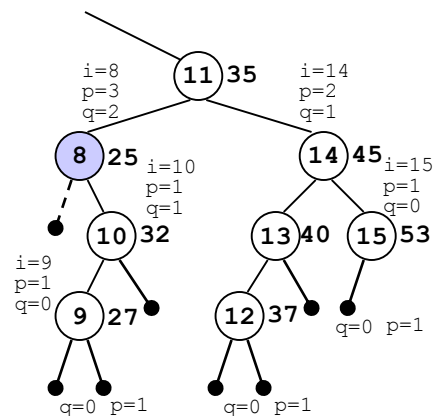
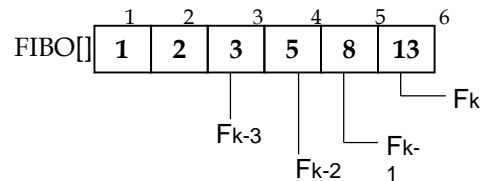
$N > 25 \rightarrow i$

dibuat menjadi :

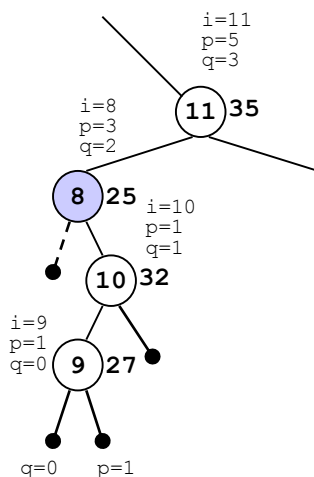
$i + m = 8 + 3 = 11$

Titik pemeriksaan awal = titik 11

Yaitu yang nilainya = 35



TAHAP - III Proses mencari nilai 27



$i = 11, A[11] = 35$

Langkah - 1.  $N < 35 \rightarrow$  cari ke kiri :

$27$   $q \neq 0 \rightarrow i = i - q \rightarrow i = 11 - 3 = 8$   
 $t = p \rightarrow t = 5$   
 $p = q \rightarrow p = 3$   
 $q = t - q \rightarrow q = 5 - 3 = 2$

$i = 8, A[8] = 25$

Langkah - 2.  $N > 25 \rightarrow$  cari ke kanan :

$27$   $p \neq 1 \rightarrow i = i + q \rightarrow i = 8 + 2 = 10$   
 $p = p - q \rightarrow p = 3 - 2 = 1$   
 $q = q - p \rightarrow q = 2 - 1 = 1$

$i = 10, A[10] = 32$

Langkah - 3.  $N < 32 \rightarrow$  cari ke kiri :

$27$   $q \neq 0 \rightarrow i = i - q \rightarrow i = 10 - 1 = 9$   
 $t = p \rightarrow t = 1$   
 $p = q \rightarrow p = 1$   
 $q = t - q \rightarrow q = 1 - 1 = 0$

$i = 9, A[9] = 27$

Langkah - 4.  $N == 27 \rightarrow 27$  Ditemukan pada langkah ke-4

Terlihat titik 8 dikunjungi sekali lagi, tapi tentu saja nilainya tidak sama dengan nilai N



```

// FiboPrg1
#include<stdio.h>
#include<conio.h>
void main()
{ int i, j, F0, F1, Fibo, n, m, N, Flag;
  int FK, FK1, FK2, FK3, s, p, q, t;
  int A[16] = { 0, 5, 7, 12, 15, 17,
               19, 22, 25, 27, 32,
               35, 37, 40, 45, 53 };

  int FIBO[8];

  clrscr();
  n = 15;
  F0 = 1; F1 = 1; Fibo = 1;
  j = 1;
  while(Fibo <= n+1)
  { FIBO[j] = Fibo;
    Fibo = F0+F1; F0=F1; F1=Fibo;
    j++; }
  s = j - 1;

  FK = FIBO[s];
  FK1 = FIBO[s-1]; i = FK1;
  FK2 = FIBO[s-2]; p = FK2;
  FK3 = FIBO[s-3]; q = FK3;

  m = (n+1) - FK;

  printf("\n Inputkan sebuah nilai : " );
  scanf("%d", &N);

  if(N > A[i]) i = i + m;

  Flag = 0;
  while( i != 0 && Flag == 0)
  { if(N == A[i]) Flag = 1;
    else
      if(N < A[i])
      { if(q == 0) i = 0;
        else { i = i - q;
              t = p;
              p = q; q = t - q;
            }
      }
    else
      { if(p==1) i = 0;
        else
          { i = i + q;
            p = p - q; q = q - p;
          }
      }
  }
  if(Flag == 1)
    printf("\nFOUND");
  else
    printf("\nNOT FOUND" );
}

```

## Program Fibonacci Search dalam Bahasa C

Menyiapkan array A[0..16]  
beserta isinya, dan yang akan disearch  
isinya adalah A[1] s.d. A[15]

Mengisi array FIBO[1..j]  
yang menyimpan deret fibonacci  
1,2,3,5.. dan seterusnya sampai  
mendekati nilai n+1  
Untuk contoh n = 15, maka angka Fibonacci  
yang paling mendekati n+1 adalah 13,  
yaitu pada FIBO[6],  
dan pada saat keluar dari loop nilai j = 7.

Keluar dari loop nilai J = 7  
sedangkan angka Fibonacci = 13  
ada di A[6]

Menyiapkan nilai i, p, q, dan m

Menginput nilai N

Untuk pertama kali,  
bila  $N > A[i]$ , maka i harus ditambah  
dengan nilai m

Flag dibuat = 1  
bila ditemui  $A[i] = N$

i dibuat = 0  
agar keluar  
dari loop, tanpa  
menemukan nilai N

Kurangkan  
nilai i untuk  
menelusuri  
ke kiri

i dibuat = 0  
agar keluar  
dari loop, tanpa  
menemukan nilai N

Tambahkan  
nilai i untuk  
menelusuri  
ke kanan

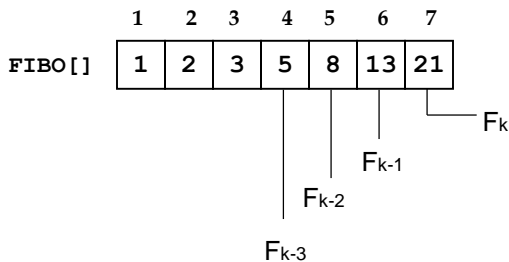
## Ilustrasi urutan proses

### II Untuk jumlah elemen $n = 20$

Misal data yang tersedia dalam array sebagai berikut :

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	5	7	12	15	17	19	22	25	27	32	35	37	40	45	53	57	62	66	69	73

Untuk  $n = 20$



$$\begin{aligned}
 m &= n+1 - F_k & i &= 13 \\
 &= 21 - 21 & p &= 8 \\
 &= 0 & q &= 5 \\
 & & m &= 0
 \end{aligned}$$

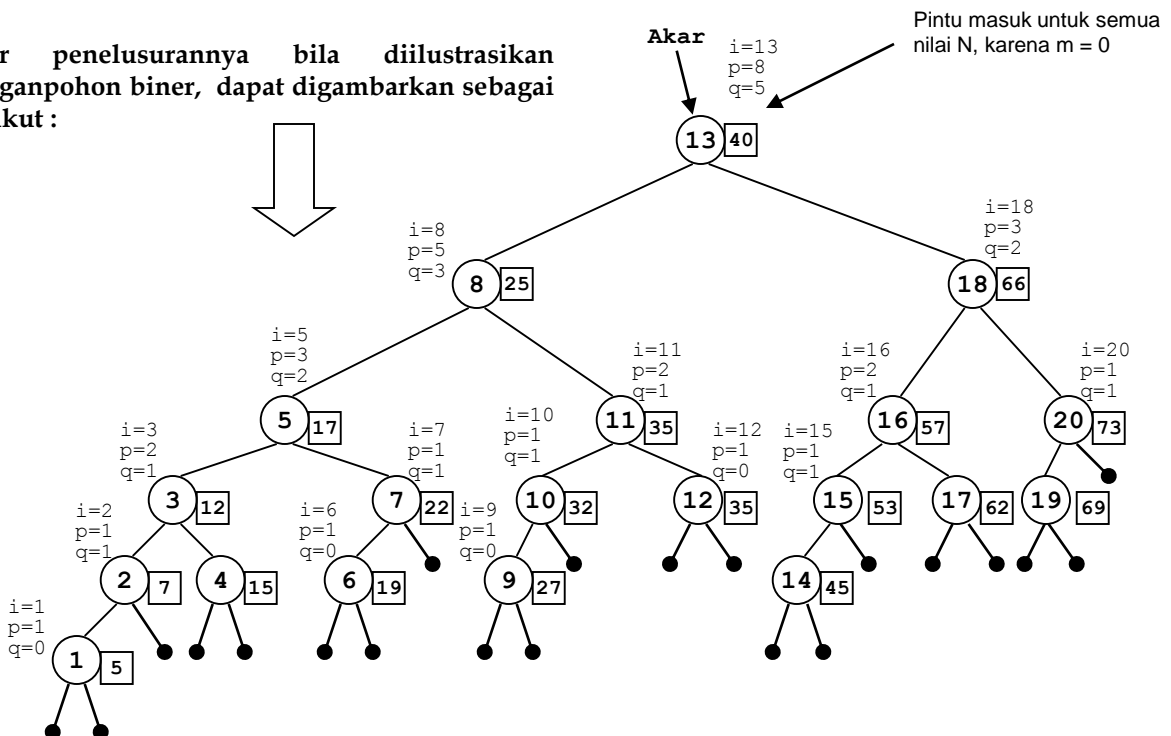
```

clrscr();
n = 15;
F0 = 1; F1 = 1; Fibo = 1;
j = 1;
while(Fibo <= n+1)
{ FIBO[j] = Fibo;
  Fibo = F0+F1; F0=F1; F1=Fibo;
  j++; }
s = j - 1;

FK = FIBO[s];
FK1 = FIBO[s-1]; i = FK1;
FK2 = FIBO[s-2]; p = FK2;
FK3 = FIBO[s-3]; q = FK3;

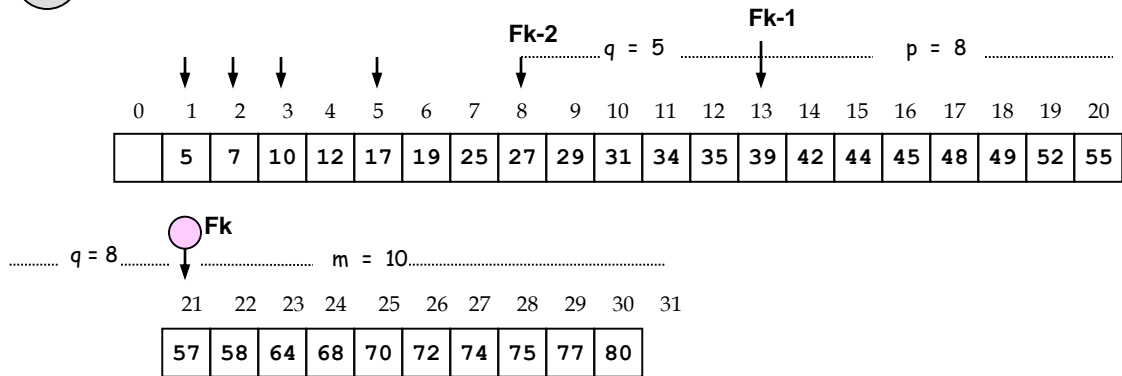
m = (n+1) - FK;
    
```

Alur penelusurannya bila diilustrasikan dengan pohon biner, dapat digambarkan sebagai berikut :



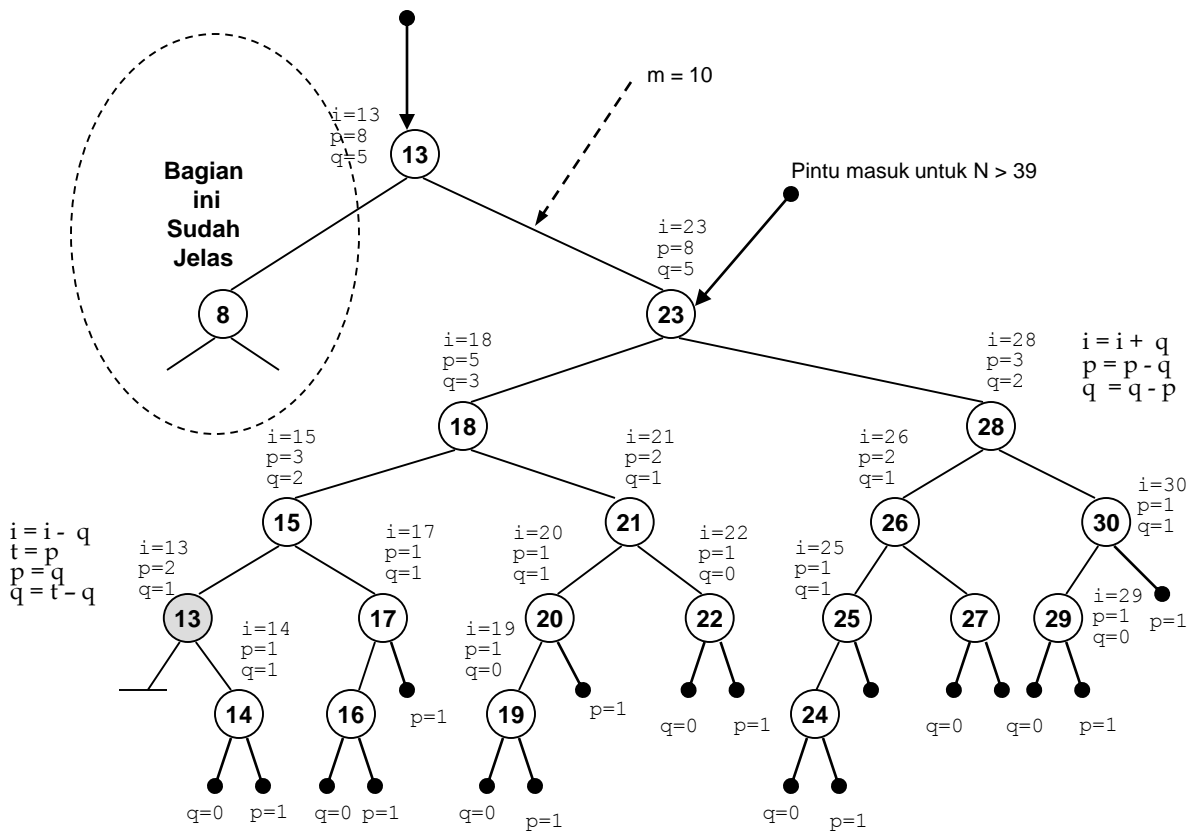
## Ilustrasi urutan proses

### III Untuk jumlah elemen $n = 30$



$n = 50$   
 $Fk = 34$

$Fk-1 = 13 \rightarrow i = 13$   
 $Fk-2 = 8 \rightarrow p = 8$   
 $Fk-3 = 5 \rightarrow q = 5$   
 $m = (30 + 1) - 21 = 10$



### Contoh Soal -1

Dalam array yang dibuat dengan `int A[51]` kolom A[1] sampai dengan A[50] sudah ada isinya nilai numerik yang urut menaik dari nilai terkecil sampai dengan nilai terbesar sebagai berikut :

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	5	7	10	12	17	19	25	27	29	31	34	35	39	42	44	45	48	49	52	55

21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
57	58	64	68	70	72	74	75	77	80	81	82	85	89	90	95	98	100	101	105

41	42	43	44	45	46	47	48	49	50
107	112	115	117	120	122	125	128	133	137

**Catatan :** Indeks yang ditandai dengan panah, (yaitu : 1,2,3,5,8,13,21,34) adalah nilai-nilai deret Fibonacci yang dipakai sebagai key.

**Soal :** Susun tabel yang memuat langkah-langkah untuk memeriksa suatu nilai N secara Fibonacci Search, apakah ditemukan atau tidak ditemukan isi array yang nilainya sama dengan nilai N.

bila nilai N sama dengan :

a. 5	d. 57	g. 100
b. 9	e. 85	h. 150
c. 17	f. 89	i. 2

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	5	7	10	12	17	19	25	27	29	31	34	35	39	42	44	45	48	49	52	55

21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
57	58	64	68	70	72	74	75	77	80	81	82	85	89	90	95	98	100	101	105

41	42	43	44	45	46	47	48	49	50
107	112	115	117	120	122	125	128	133	137

$n$	=	50	$F_{k-1}$	=	21	→	$i$	=	21
$F_k$	=	34	$F_{k-2}$	=	13	→	$p$	=	13
			$F_{k-3}$	=	8	→	$q$	=	8
			$m$	=	$(50 + 1) - 34$	=	17		

**Jawab :** Untuk data diatas, A[1] sampai dengan A[50], didapat:

$$n = 50$$

$$F_k = 34$$

$$F_{k-1} = 21 \rightarrow i = 21$$

$$F_{k-2} = 13 \rightarrow p = 13$$

$$F_{k-3} = 8 \rightarrow q = 8$$

$$m = (50 + 1) - 34 = 17$$

Catatan untuk menghitung  $i$ ,  $p$ , dan  $q$  serta kondisi "Tidak Ditemukan"

$$(1) i = i - q$$

$$(2) t = p$$

$$(3) p = q \quad q \leq 0 \rightarrow \text{tidak ditemukan}$$

$$(4) q = t - q$$

$$(5) i = i + q$$

$$(6) p = p - q \quad p \leq 1 \rightarrow \text{tidak ditemukan}$$

$$(7) q = q - p$$

a.

Untuk :  $N = 5$

$$A[i] = 57$$

$N = 5$ , ---->  $N$  Tidak Lebih besar dari  $A[i]$ , sehingga  $i$  tidak diroboh.

$q = 8$

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	5	7	10	12	17	19	25	27	29	31	34	35	39	42	44	45	48	49	52	55

$q = 8$

$p = 13$

21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
57	58	64	68	70	72	74	75	77	80	81	82	85	89	90	95	98	100	101	105

							Ke kiri					Ke kanan						
Langkah ke-	i	p	q	A[i]	Kondisi N & A[I]	Cari ke kiri / kanan	Kondisi q	i (1)	t (2)	p (3)	q (4)	Kondisi p	i (5)	p (6)	q (7)			
I	21	13	8	57	5 < 57	Ke kiri	q != 0	13	13	8	5							
II	13	8	5	39	5 < 39	ke kiri	q != 0	8	8	5	3							
III	8	5	3	27	5 < 27	ke kiri	q != 0	5	5	3	3							
IV	5	3	2	17	5 < 17	ke kiri	q != 0	3	3	2	1							
V	3	2	1	10	5 < 10	ke kiri	q != 0	2	2	1	1							
VI	2	1	1	7	5 < 7	ke kiri	q != 0	1	1	1	0							
VII	1	1	0	5	5 = 5	Pada langkah ke-VII												