

MODUL MATA KULIAH

ALGORITMA DAN STRUKTUR DATA 1

KP002 – 3 SKS



**FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BUDI LUHUR**

**JAKARTA
SEPTEMBER 2019**

TIM PENYUSUN

Painem, S.Kom, M.Kom
Reva Ragam, S.Kom., M.Kom
Ir. Moch. Sjukani



MODUL PERKULIAHAN #11

POINTER

Capaian Pembelajaran	:	Mahasiswa mampu memahami konsep pointer dan implementasi pointer dalam bahasa pemrograman
Sub Pokok Bahasan	:	1.1. Operator Pointer 1.2. Pointer Aritmatika 1.3. Pointer Perbandingan 1.4. Kontanta String dan Pointer 1.5. Array Pointer 1.6. Pointer dalam Pointer
Daftar Pustaka	:	1. Sjukani M, "Struktur data dengan C++(Algoritma dan Struktur Data 2 dengan C, C++)", Mitra Wacana Media, 2007 2. Kristanto Andri, "Algoritma dan Pemrograman dengan C++", Graha Ilmu, 2003 3. Darmawan Erico, "Pemrograman Dasar C-Java-C#", Informatika, 2012

POINTER

1.1. PENGERTIAN POINTER

Pointer adalah variabel yang berisi alamat memori suatu lokasi memori. Dengan kata lain, pointer berisi alamat dari variabel yang mempunyai nilai tertentu. Bentuk umum dari pernyataan variabel pointer adalah

Type *namavariabel;

- Tipe adalah tipe dasar pointer
- Nama variabel adalah nama variabel pointer
- * adalah operator memori yang fungsinya untuk mengembalikan nilai variabel pada alamat yang telah ditentukan oleh operand

1.2. OPERATOR POINTER

Ada dua operator yang bisa digunakan dalam pointer:

a. Operator *Deference* (&)

Operator ini biasanya disebut dengan *address of* atau operator alamat. Dengan menggunakan operator deference (&) ini, suatu variabel akan menghasilkan alamat memori. Alamat dalam hexadecimal (0-9,A-F)

Pada saat pendeklarasian variabel, user tidak diharuskan menentukan lokasi sesungguhnya pada memory, hal ini akan dilakukan secara otomatis oleh kompiler dan *operating sysem* pada saat *run-time*.

Contoh :

```
int A=45;  
printf ("%x", &A) ;
```

Pada program di atas, akan ditampilkan alamat memori dari variabel x, bukan nilai x.

Misal tampil A : 62fe4c

b. Operator *Reference* (*)

Operator ini biasanya disebut *value pointed by*. Dengan menggunakan operator



ini, kita dapat mengakses secara langsung nilai yang terdapat pada suatu alamat memori.

Contoh :

```
int A = 45;

int *P;

P = &A;

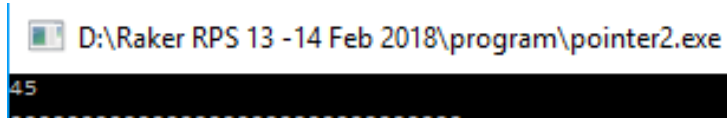
printf("%i", *P);
```

Pada program di atas, akan ditampilkan nilai dari alamat memori &P. Penggalan program diatas akan tampil 45

Contoh penggunaan pointer :

```
1  #include <stdio.h>
2  main()
3  {
4      int A=45;
5      int *P;
6      P=&A;
7      printf("%i",*P);
8
9  }
10
```

Output :



Menyiapkan variabel A dengan tipe int, dan isinya dibuat = 25

```
int A=45;
```

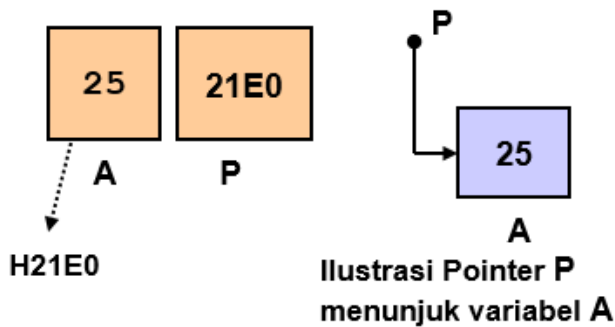
Menyiapkan variabel P dengan tipe Pointer to integer (biasa ditulis : " int * ")

```
int *P;
```

Mengisi Pointer P dengan alamat variabel A. Disebut pointer P menunjuk variabel A. Bila alamat variabel A = H21E0, maka isi Pointer P atau nilai Pointer P = H21E0

```
P=&A;
```





Maksudnya mencetak isi variabel yang alamatnya ditunjuk oleh pointer P. Sedangkan variabel yang alamatnya sedang ditunjuk oleh pointer P adalah variabel A. Jadi yang dicetak adalah isi variabel A sehingga tercetak : 25

***P Maksudnya variabel yang ditunjuk oleh pointer P**

```
printf("%d", *P);
```

a) Mencetak isi Pointer P sebagai alamat variabel A

Contoh :

```

1  #include <stdio.h>
2  main()
3  {
4      int A=25;
5      int *P;
6      P = &A;
7      printf("%p", P);
8
9  }
```

Output :



b) Mencetak isi Pointer P sebagai alamat variabel A dalam bentuk desimal

Contoh :

```
1  #include <stdio.h>
2  main()
3  {
4      int A=25;
5      int *P;
6      P = &A;
7      printf("%i", P);
8  }
9
```

Dicetak dengan format integer (%i)
maka isi variabel P (alamat variabel A)
dicetak dalam bentuk decimal

Output :

```
D:\Raker RPS 13 -14 Feb 2018\program\pointerdes.exe
6487620
-----
```

c) Mengisi variabel dengan nilai yang ditunjuk oleh sebuah Pointer.

Contoh :

```
1  #include <stdio.h>
2  main()
3  {
4      int *P;
5      int A, B;
6      A = 25;
7      P = &A;
8      B = *P;
9      printf("%i",A);
10     printf("\n%i",B);
11 }
```

Output :

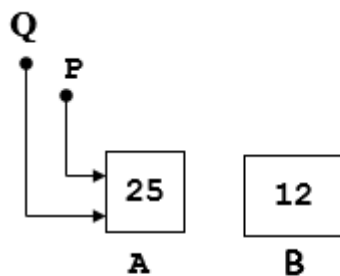
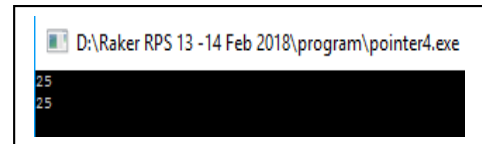
```
D:\Raker RPS 13 -14 Feb 2018\program\isipoiter.exe
25
25
-----
```



d) Sebuah alamat dapat ditunjuk oleh lebih dari satu pointer.

Contoh :

```
1  #include <stdio.h>
2  main()
3  {
4      int A = 25, B = 12;
5      int *P, *Q;
6      P = &A;
7      Q = P;
8      printf("%i\n", *P);
9      printf("%i\n", *Q);
10 }
```



Pointer P dan Q
keduanya menunjuk variabel A

Keduanya mencetak isi variabel A

1.3. POINTER MENUNJUK ALAMAT ELEMEN ARRAY SATU DIMENSI

a) Pointer menunjuk alamat elemen array

Secara internal array juga menyatakan alamat, yaitu pengenal array sama dengan alamat pada elemen pertama, pada array.

Contoh :

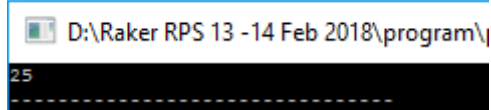


```

1  #include <stdio.h>
2  main()
3  {
4
5      int A[5] = {25,12,17,10,15};
6      int *P;
7      P = A;
8      printf("%i", *P );
9  }

```

Output :



Deklarasi array satu dimensi

```
int A[5] = {25,12,17,10,15}
```

Deklarasi variabel pointer

```
int *P;
```

Maksudnya Pointer P menyimpan alamat array A. Untuk sebuah array, alamatnya diwakili oleh alamat elemen pertama, untuk contoh diatas yaitu A[0])

Perhatikan : **P = A;** bukan **P = &A;** karena dalam bahasa C, bila menyebut nama array satu dimensi berarti menyebut atau menunjuk elemen pertama, dalam contoh diatas adalah A[0].

Sehingga : **P = A;**

dapat ditulis menjadi : **P = &A[0];** dan kalau ingin menunjuk langsung ke **A[3]**

dapat ditulis dengan : **P = &A[3]** atau: **P = A + 3**

```
P=A;
```

Mencetak isi pointer P

```
printf("%i", P);
```



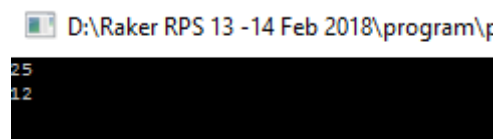
b) Menggerakkan Pointer yang menunjuk array

Nilai sebuah pointer dapat dikenai operasi arithmetic seperti ditambah atau dikurang.

Contoh :

```
1  #include <stdio.h>
2  main()
3  {
4      int A[5] = {25,12,17,10,15};
5      int *P;
6      P = A;
7      printf("%i\n", *P );
8      P++;
9      printf("%i\n", *P );
10
11 }
```

Output :

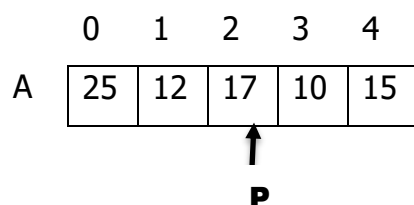


```
D:\Raker RPS 13 -14 Feb 2018\program\p
25
12
```

Dengan P++, atau P+=1, atau P=P+1; maka pointer P maju satu 'langkah'.

c) Memajukan atau memundurkan pointer

Sudah ada array yang dibuat dengan int A[5]. Sudah ada isinya, dan pointer P sedang menunjuk elemen A[2] dengan ilustrasi berikut ini :



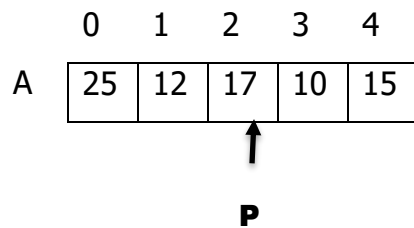
```
printf("%i", *P );
```

Perintah diatas akan tercetak :

17



Mencetak isi elemen yang sedang ditunjuk oleh P, sehingga tercetak : 17 (isi A[2]) Pointer P tetap menunjuk A[2] sebagai berikut:



```
printf("%i", *P++ );
```

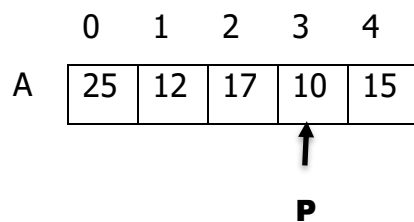
Perintah diatas akan tercetak :

17

Instruksi diatas sebenarnya dua instruksi yang digabung menjadi satu

```
printf("%i", *P);  
P++;
```

Mencetak isi elemen yang sedang ditunjuk oleh P, sehingga tercetak : 17 (isi A[2]). Setelah itu Pointer P bergerak kekanan satu (1) elemen sehingga menunjuk A[3] sebagai berikut:



```
printf("%i", *++P );
```

Output :

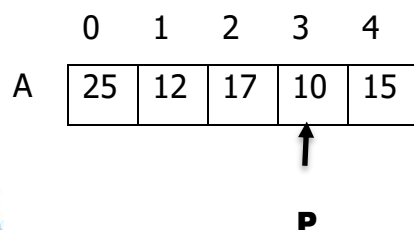
10

Instruksi diatas sebenarnya dua instruksi yang gabung menjadi satu:

P++;

printf("%i", *P);

Pointer **P** bergerak dulu satu elemen ke kanan, ke **A[3]**, kemudian mencetak isi **A[3]**, sehingga tercetak : **10**. Ilustrasi sebagai berikut :



```
printf("%i", ++*P );
```

Output :

10

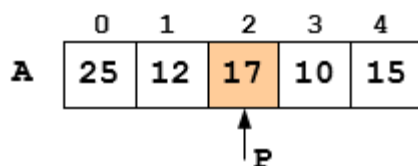
Instruksi diatas sebenarnya dua instruksi yang digabung jadi satu:

```
++*P;
```

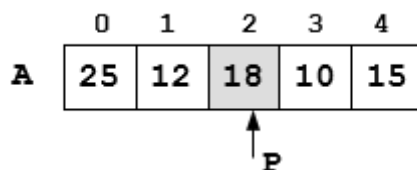
```
printf("%i", *P);
```

Sama juga dengan : `A[2]++;`

```
printf("%i", A[2]);
```



Elemen yang ditunjuk oleh Pointer **P**, isinya ditambah 1 lebih dulu. Kemudian baru dicetak.

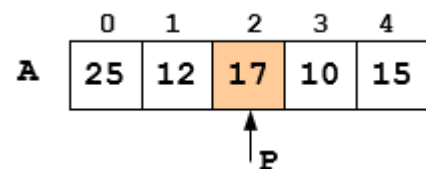


```
printf("%i", ++*P++ );
```

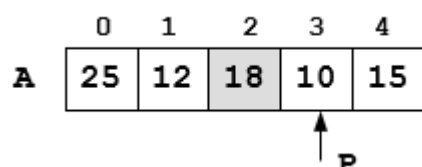
Output :

18

Instruksi diatas sebenarnya tiga instruksi dibawah ini digabung menjadi satu:



Elemen yang ditunjuk oleh Pointer **P**, isinya ditambah 1 lebih dulu. Kemudian baru dicetak. Setelah mencetak Pointer **P** bergerak satu langkah ke kanan

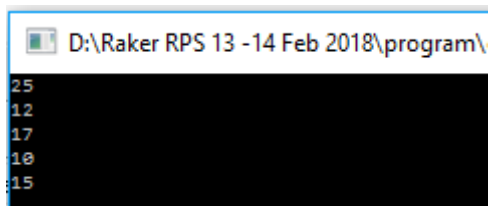


1.4. MENCETAK ISI ARRAY SATU DIMENSI DENGAN POINTER

Contoh :

```
1  #include <stdio.h>
2  main()
3  {
4      int I,*P,A[5] = {25,12,17,10,15};
5      P = A;
6      for(I=1; I <= 5; I++ )
7      { printf("%i\n", *P);
8        P++;
9      }
10
11 }
```

Output :



```
D:\Raker RPS 13 -14 Feb 2018\program\
25
12
17
10
15
```

I digunakan untuk mengatur loop 5 kali *P yang menunjuk data yang akan dicetak

1.5. POINTER MENUNJUK ALAMAT ELEMEN ARRAY DUA DIMENSI

Contoh :

```
1  #include <stdio.h>
2  main()
3  { char A[3][5] = { 'A','B','C','D','E',
4                    'F','G','H','I','J',
5                    'K','L','M','N','O' };
6      char *P;
7      P = &A[0][0];
8      printf("%c", *P );
9  }
```

Ouput :

A



Sebenarnya oleh Bahasa C, array diatas, strukturnya dalam memory dibuat tersusun dengan ilustrasi sebagai berikut :

```
char A[3][5];
```

Dengan instruksi diatas maka terbentuk sebuah array dua dimensi A[3][5] yang biasa diilustrasikan dengan gambar sebagai berikut :

	0	1	2	3	4
0	A	B	C	D	E
1	F	G	H	I	J
2	K	L	M	N	O

Sebenarnya oleh Bahasa C, array diatas, strukturnya dalam memory dibuat tersusun dengan ilustrasi sebagai berikut :

	0,0	0,1	0,2	0,3	0,4	1,0	1,1	1,2	1,3	1,4	2,0	2,1	2,2	2,3	2,4
A	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	Baris 1					Baris 2					Baris 3				

Tersusun sedemikian rupa sehingga secara fisik alamat A[1][0] merupakan sambungan dari alamat A[0][4] dan alamat A[2][0] merupakan sambungan alamat A[1][4]

```
P = &A[0][0];
```

Perintah diatas pointer P menunjuk elemen pertama dari array yang dapat diilustrasikan sebagai berikut :

	0,0	0,1	0,2	0,3	0,4	1,0	1,1	1,2	1,3	1,4	2,0	2,1	2,2	2,3	2,4
A	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	↑														
	P														



```
printf("\n%c", *P );
```

Perintah diatas akan mencetak isi elemen yang sedang ditunjuk oleh pointer P yaitu A[0][0], sehingga tercetak karakter 'A'.

SOAL LATIHAN

Soal-1

Susun program (penggalan program) untuk menyiapkan array satu dimensi tipe integer sebanyak 5 elemen dengan nama A, kemudian mengisi array tersebut sehingga isinya menjadi sebagai berikut :

	0	1	2	3	4
A	1	2	3	4	5

Soal -2

Apa yang tercetak jika program berikut ini dijalankan !

```
#include<stdio.h>

void main()
{ int A, B, T,  *P, *Q, *R ;
  A = 25; B = 12;
  P = &A; Q = &B; R = &T;
  *R = *P + *Q;
  printf("%i", *R );
}
```

Soal -3

Apa yang tercetak jika program berikut ini dijalankan

```
#include<stdio.h>

void main()
{ int I,*P;
  int A[5]={15,12,25,17,10};
  P = &A[2];
  printf("%i", *--P);
  printf("\n%i", *P);
}
```



Soal-4

Sudah dibuat array dua dimensi `int A[3][5]` lengkap dengan isinya, dengan ilustrasi sebagai berikut :

	0	1	2	3	4
0	15	12	25	17	10
1	21	30	32	23	35
2	16	8	7	40	11

Apa yang tercetak jika ada penggalan program sebagai berikut :

a.

```
int I,J;*P;
P = &A[0][0];
I=1;
while(I <= 3)
{ J = 1;
  while (J <= 5 )
  { printf("%3i", *P);
    P++;
    J++;
  }
  printf("\n");
  I++;
}
```

b.

```
int I,J;*P;
P = &A[0][0];
I=1;
while(I <= 3)
{ printf("%3i", *++P);
  P+=3;
  I++;
}
```

KESIMPULAN

1. Pointer adalah variabel yang berisi alamat memori suatu lokasi memori. Dengan kata lain, pointer berisi alamat dari variabel yang mempunyai nilai tertentu.
2. Ada dua operator yang bisa digunakan dalam pointer:

a. Operator *Deference* (&)

Operator ini biasanya disebut dengan *address of* atau operator alamat. Dengan menggunakan operator deference (&) ini, suatu variabel akan menghasilkan alamat memori



b. Operator Reference (*)

Operator ini biasanya disebut *value pointed by*. Dengan menggunakan operator ini, kita dapat mengakses secara langsung nilai yang terdapat pada suatu alamat memori.





FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BUDI LUHUR

Jl. Raya Ciledug, Petukangan Utara, Pesanggrahan

Jakarta Selatan, 12260

Telp: 021-5853753 Fax : 021-5853752

<http://fti.budiluhur.ac.id>