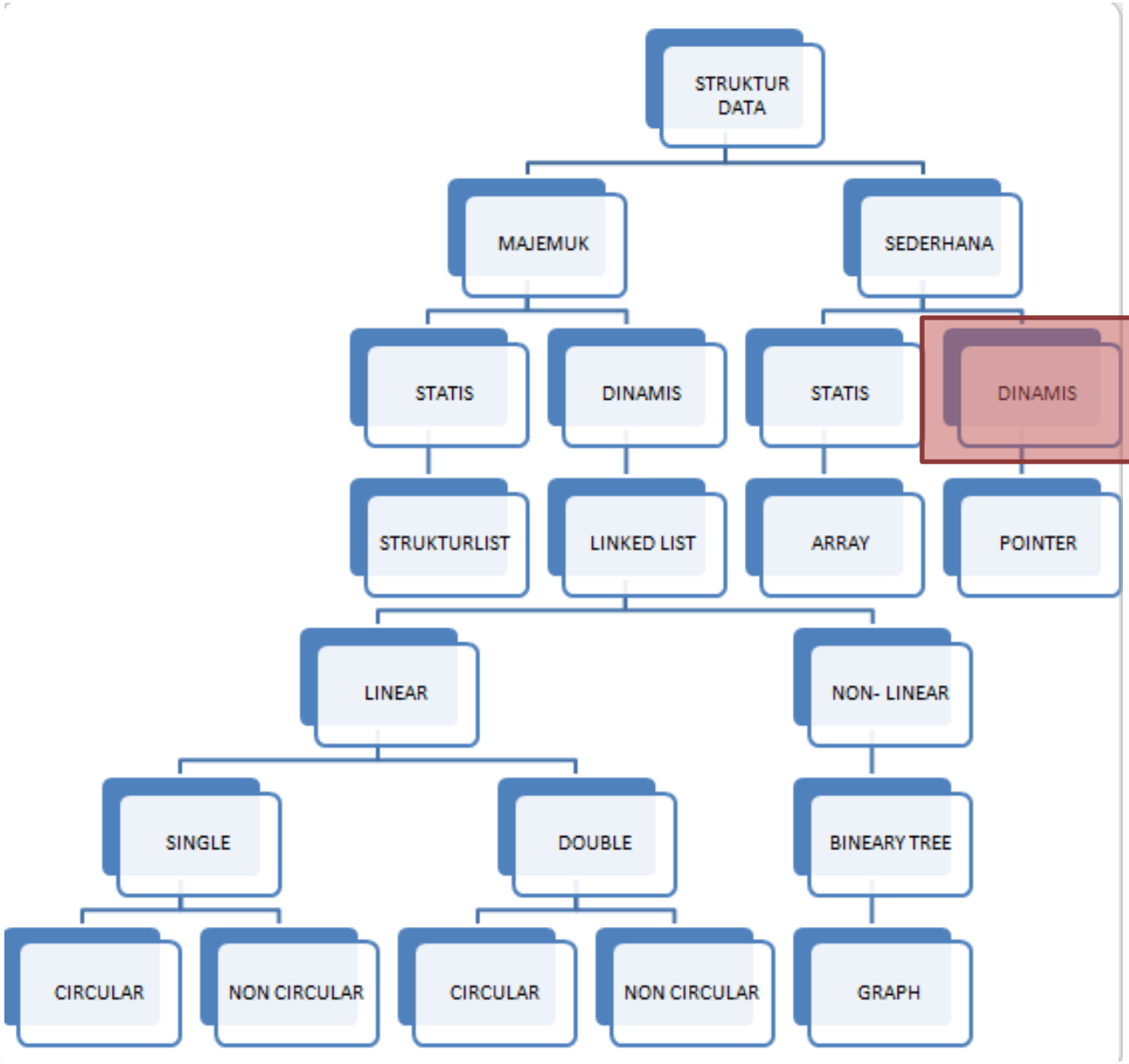
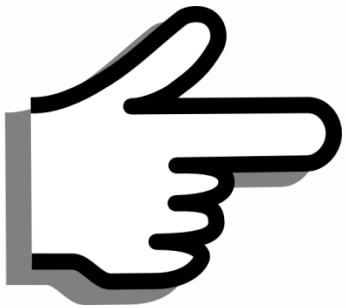
pointers

Tenia Wahyuningrum
www.st3telkom.ac.id


STRUKTUR DATA
MAJEMUK
SEDERHANA
STATIS
DINAMIS
STATIS
DINAMIS
STRUKTURLIST
LINKED LIST
ARRAY
POINTER
LINEAR
NON- LINEAR
SINGLE
DOUBLE
BINEARY TREE
CIRCULAR
NON CIRCULAR
CIRCULAR
NON CIRCULAR
GRAPH



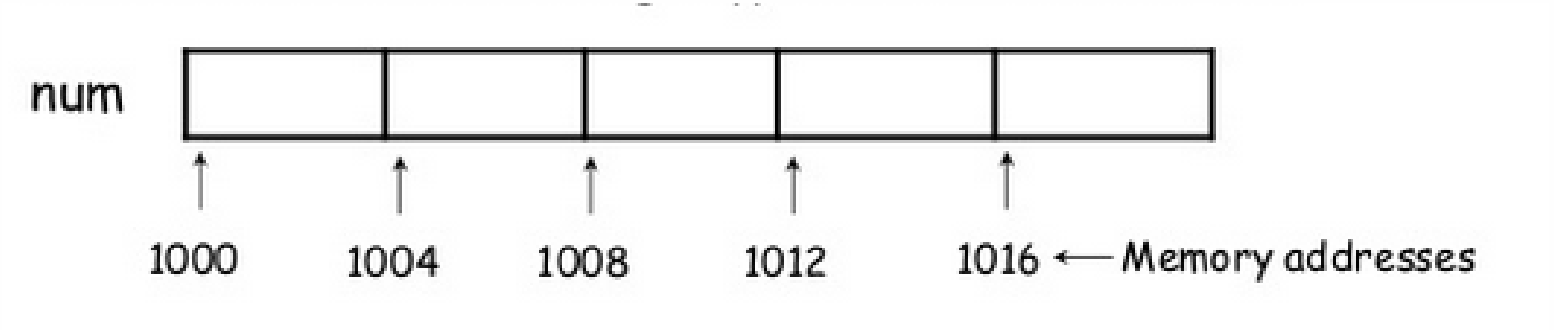
Mengapa pointer disebut dinamis?

Array

-Sebuah urutan variabel dengan nama dan tipe data yang sama

int num [5]

-5 buah urutan variabel dengan tipe integer dengan nama variabel num



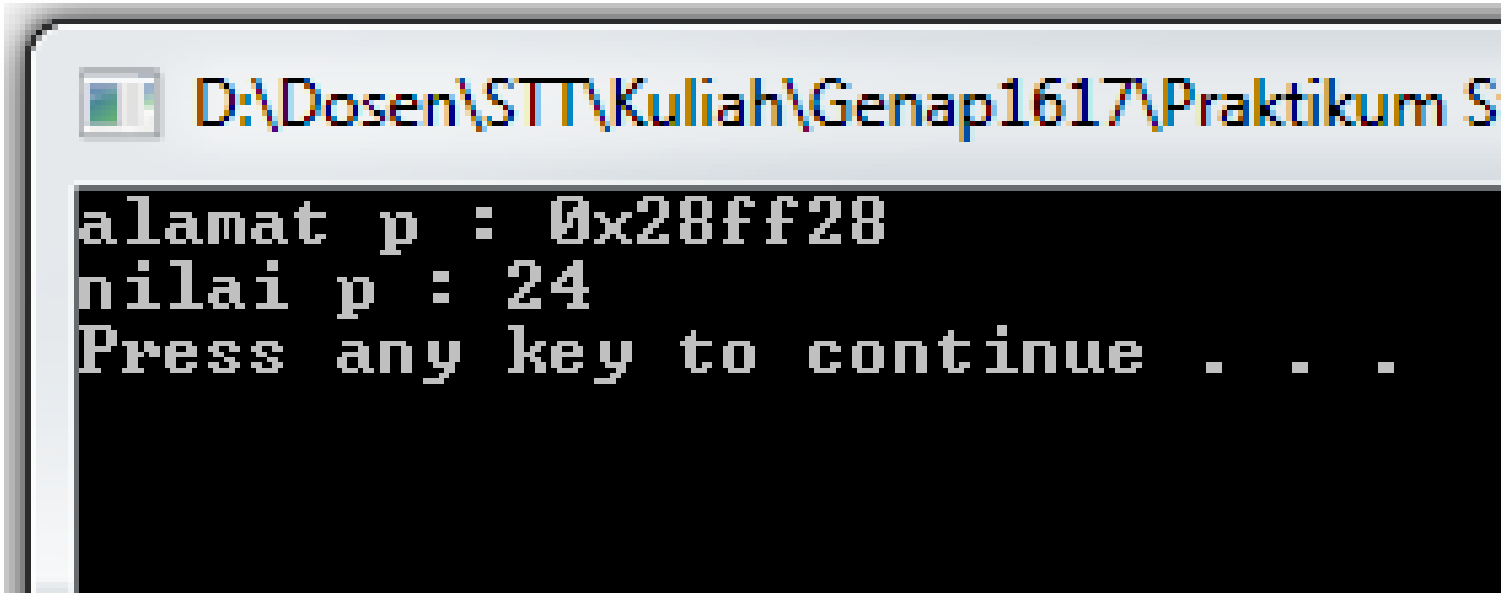
- bersifat statis (ukuran dan urutannya sudah pasti).
- ruang memori yang dipakai olehnya tidak dapat dihapus bila variabel bertipe array tersebut sudah tidak digunakan lagi pada saat program dijalankan.
- pointer bersifat dinamis, **variabel akan dialokasikan hanya pada saat dibutuhkan** dan sesudah tidak dibutuhkan dapat dihapus kembali.

Operator pointer

- & → menghasilkan **alamat**
- * → menghasilkan reference dari sebuah alamat (**nilai/value**)

```
int a[5];
int *p;
a[0]=24;
a[1]=32;
a[2]=81;
a[3]=44;
a[4]=23;

p=&a[0];
cout<<"alamat p : "<<p<<endl;
cout<<"nilai p : "<<*p<<endl;
```


D:\Dosen\STT\Kuliah\Genap1617\Praktikum S
alamat p : 0x28ff28
nilai p : 24
Press any key to continue . . .

Pointer Bertipe Void

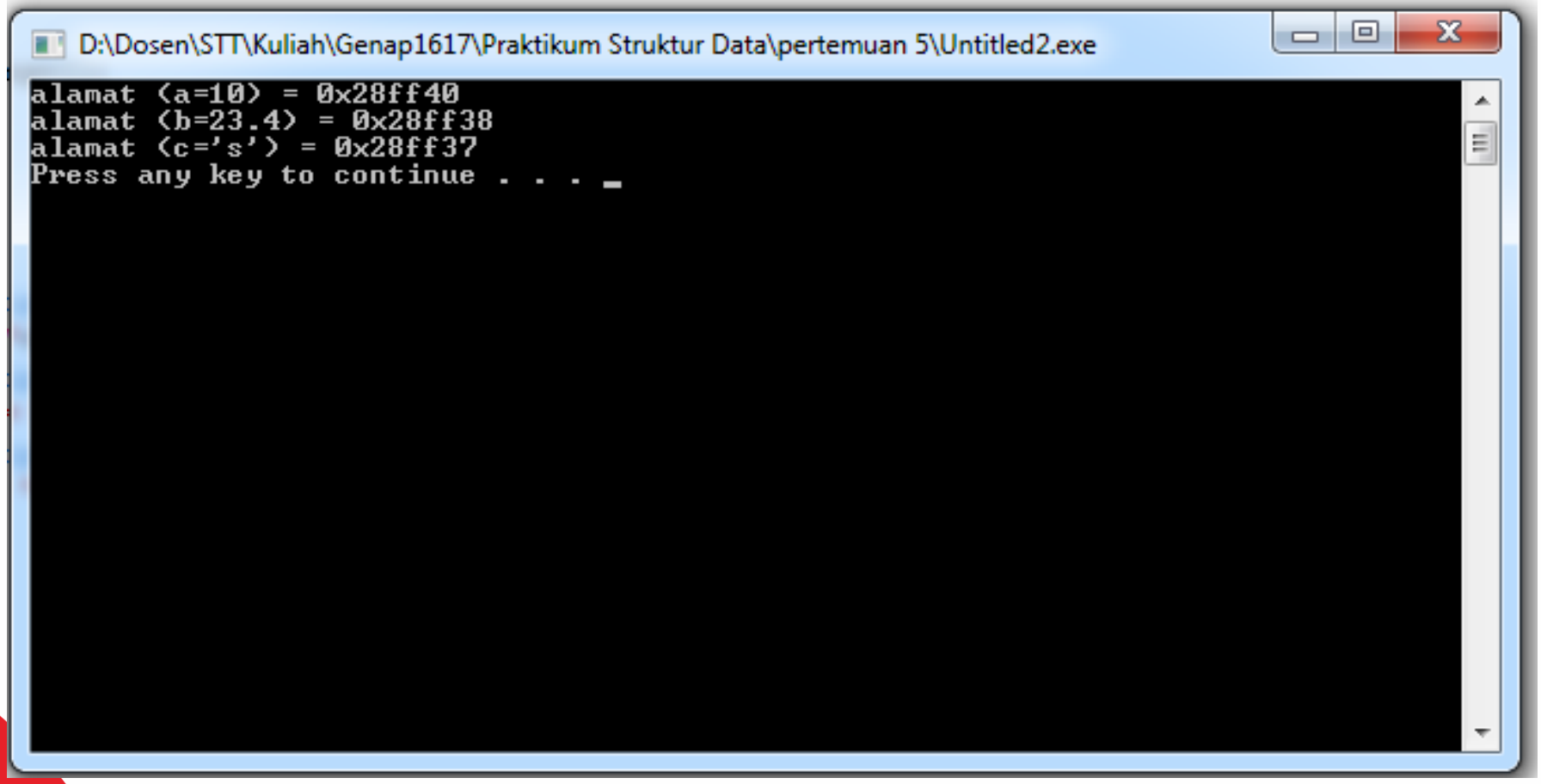
- Pada C++ terdapat pointer yang dapat menunjuk ke tipe data apapun, pointer semacam ini dideklarasikan dengan tipe **void** sehingga sering dikenal dengan istilah **Void Pointer**.

Untitled2.cpp

```cpp
#include <iostream>
#include <stdlib.h>
#include <string>
#include <conio.h>
#include <stdio.h>

int main(int argc, char *argv[])
{
    void *p;
    int a=10;
    double b=23.4;
    char c='s';
    p=&a; //p menunjuk ke tipe data int
    cout<<"alamat (a=10) = "<<p<<endl;
    p=&b; //p menunjuk ke tipe data double
    cout<<"alamat (b=23.4) = "<<p<<endl;
    p=&c; //p menunjuk ke tipe data char
    cout<<"alamat (c='s') = "<<p<<endl;


  system("PAUSE");
  return 0;
}
```

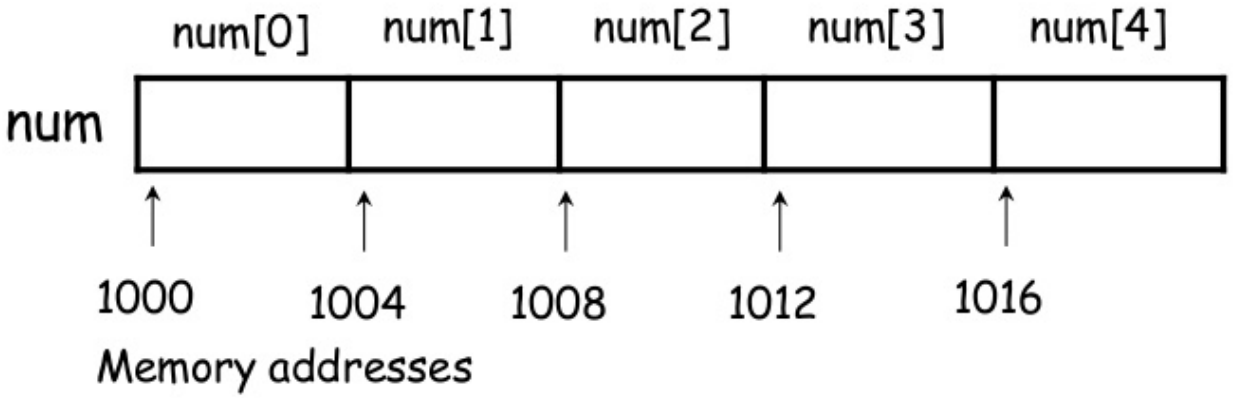

D:\Dosen\STT\Kuliah\Genap1617\Praktikum Struktur Data\pertemuan 5\Untitled2.exe
alamat (a=10) = 0x28ff40
alamat (b=23.4) = 0x28ff38
alamat (c='s') = 0x28ff37
Press any key to continue . . .

Hubungan array dan pointer

Manakah alamat masing-masing elemen?

```
int num[5] ;
```

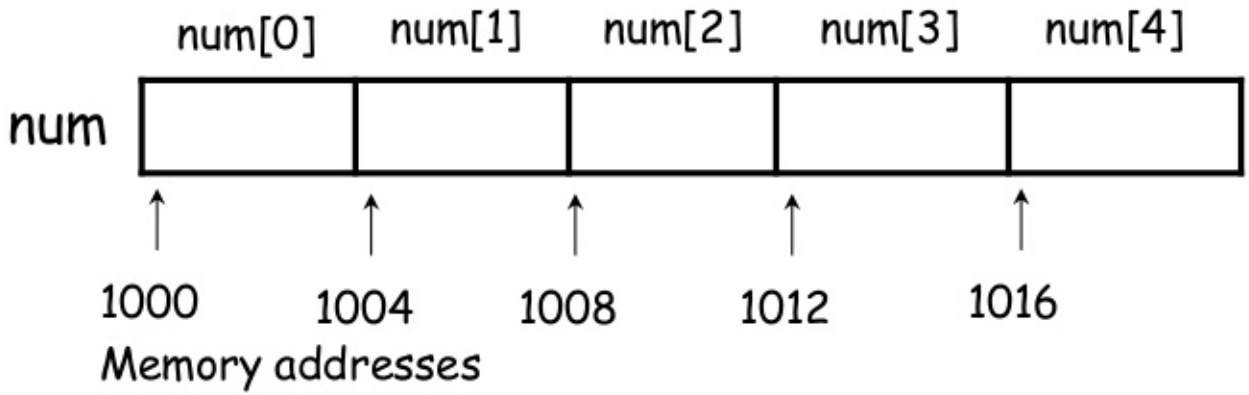
```
&num[0] == 1000
&num[1] == 1004
&num[2] == 1008
&num[3] == 1012
&num[4] == 1016
```



Hubungan array dan pointer

Apa itu num?

```
int num[5] ;
```



- *num* is the constant pointer of which value is the start address of the array.

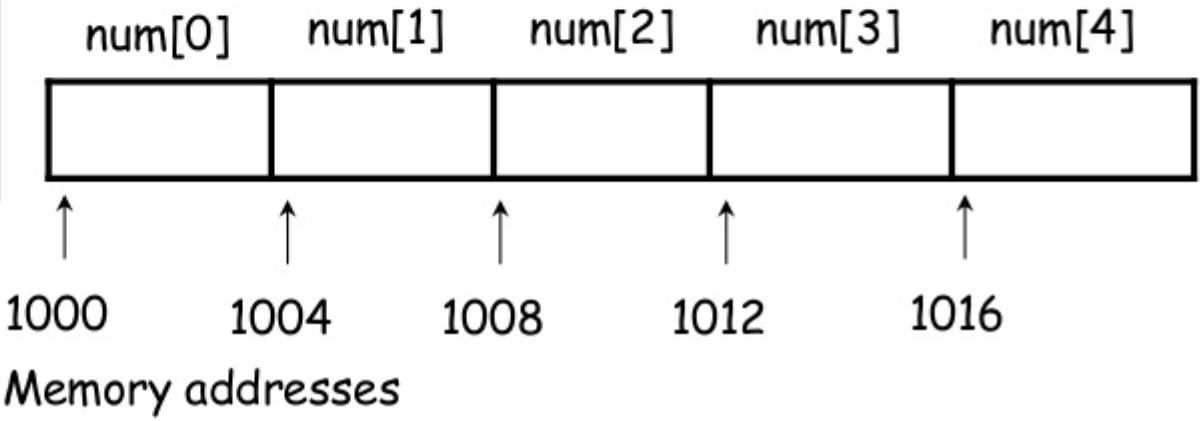
```
num == &num[0] == 1000
```

- **Example : Arithmetic of pointers**
  - "*pointer* + 1" does not mean increasing *pointer* by 1.
  - "*pointer* + 1" is "the address of the next element".
  - "*pointer* – 1" is "the address of the prior element".

```
num == &num[0] == 1000

(num+0) == &num[0]
(num+1) == &num[1]
(num+2) == &num[2]
(num+3) == &num[3]
(num+4) == &num[4]
```

```
int num[5] ;
```



Pointer arithmetic

- **Example : Arithmetic of pointers**

```
int num[5] ;

num[0] =10 ;
num[1] = 20 ;
num[2] = 30 ;
num[3] = 40 ;
num[4] = 50 ;
```

```
int num[5] ;

*num = 10 ;
*(num+1) = 20 ;
*(num+2) = 30 ;
*(num+3) = 40 ;
*(num+4) = 50 ;
```

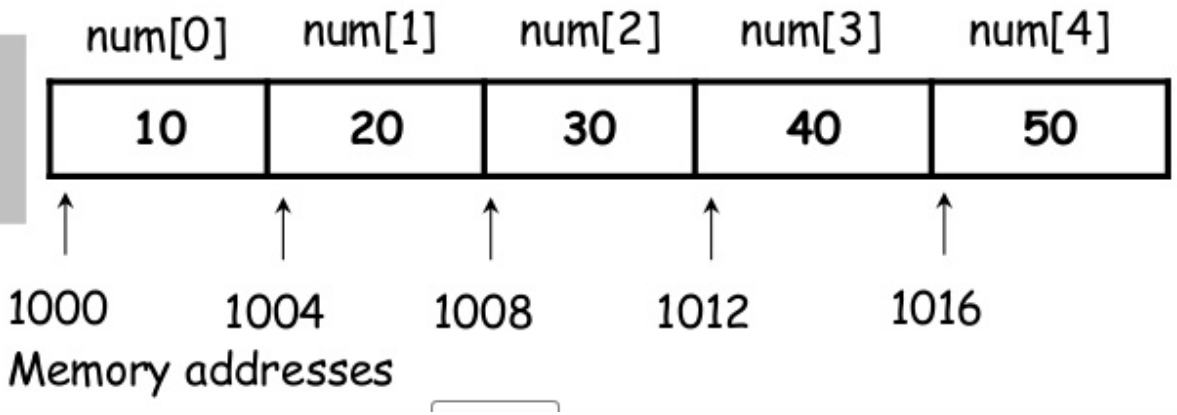
```
int num[5], *p = num ;

*p = 10 ;
*(p+1) = 20 ;
*(p+2) = 30 ;
*(p+3) = 40 ;
*(p+4) = 50 ;
```

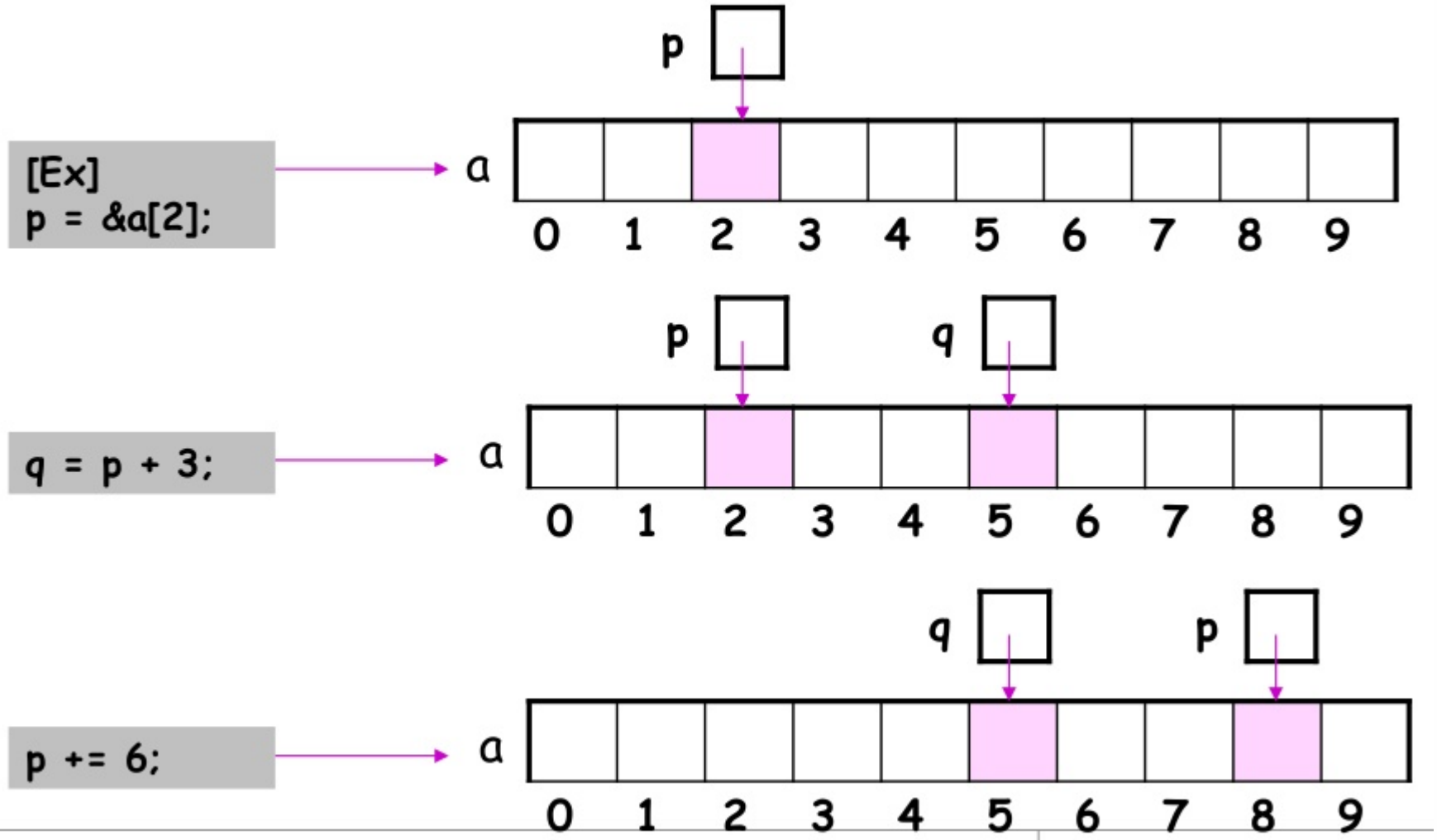
```
int num[5], *p = num ;

p[0] = 10 ;
p[1] = 20 ;
p[2] = 30 ;
p[3] = 40 ;
p[4] = 50 ;
```

```
int num[5] ;
```

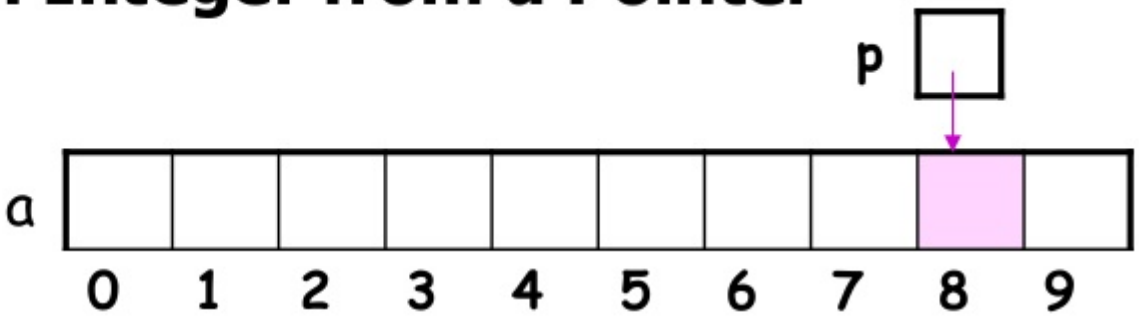


- **Adding an Integer to a Pointer**

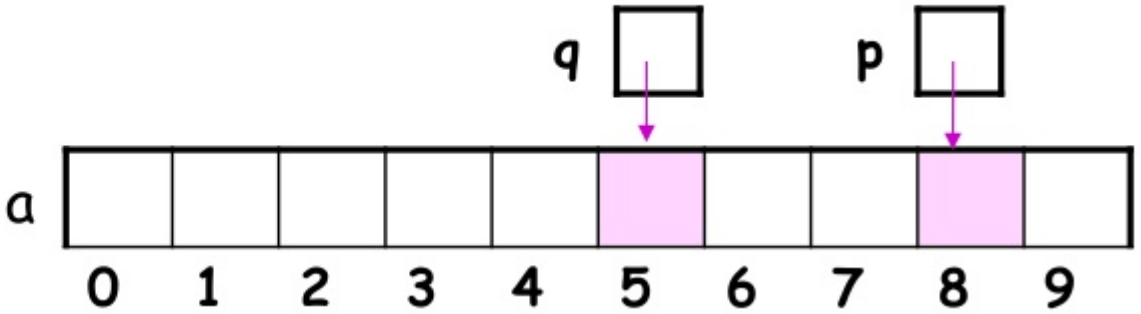


- **Subtracting an Integer from a Pointer**

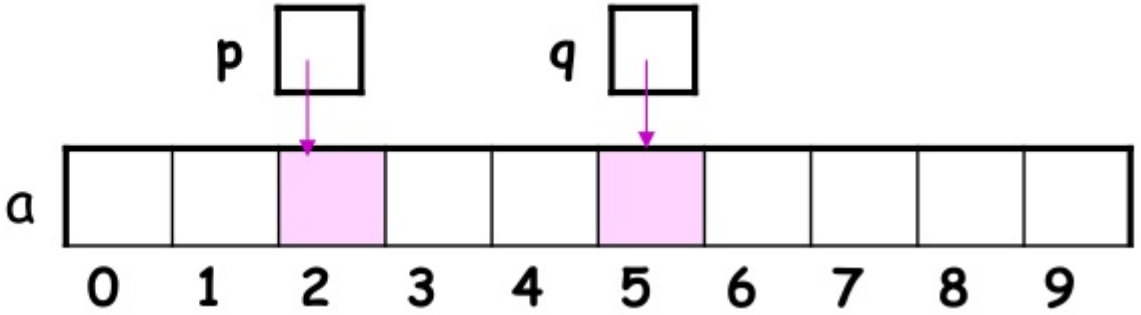
[Ex]
p = &a[8];



q = p - 3;



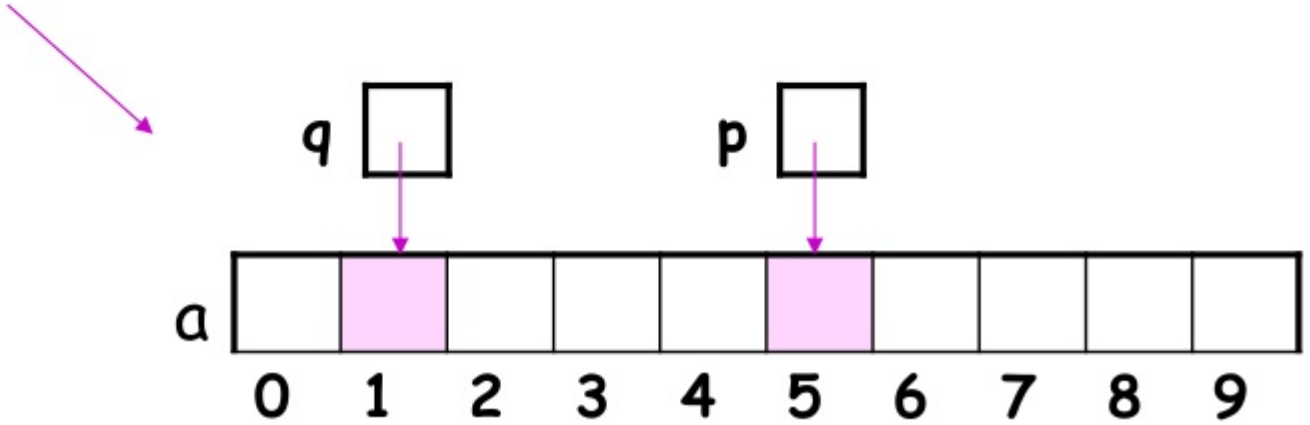
p -= 6;



Subtracting Pointers

```
[Ex]
p = &a[5];
q = &a[1];

i = p - q;          /* i == 4  */
i = q - p;          /* i == -4 */
```



- **Comparing Pointers**
  - Relational operators (<, <=, >, >=) can be applied
  - Equality operators (==, !=) can be applied

```
[Ex]
p = &a[5];
q = &a[1];

p <= q;    /* result is 0 */
p >= q;    /* result is 1 */
```

- **Example: Pointer Operation**

```
int a[ ] = { 5,15,25,43,12,1,7,89,32,11}
int *p = &a[1], *q = &a[5] ;

1. *(p + 3) ?

2. *(q - 2) ?

3. q - p ?

4. if ( p > q ) ?

5. if ( *p > *q )?
```

bila menggunakan **pointer** dengan cara yang salah maka akan menyebabkan **sistem operasi menjadi rusak.** **Jadi, berhati-hatilah…..**

Latihan (Lagi)

- Mengapa pointer disebut struktur data dinamis?
- Apa perbedaan perintah * dan & pada pointer?
- Tuliskan perintah untuk menampilkan alamat dari variabel berikut ini :

  int a=10;

Thank you!