Praktikum 10 (4/5)

POINTER

A. TUJUAN

1. Menjelaskan tentang Array of Pointer
2. Menjelaskan tentang Pointer to Pointer

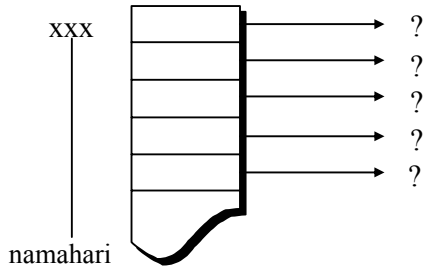
B. DASAR TEORI

Array dari Pointer (Array of Pointer)

- Suatu array bisa digunakan untuk menyimpan sejumlah pointer. Sebagai contoh:

```
char *namahari[10];
```

merupakan pernyataan untuk mendeklarasikan array pointer. Array **namahari** terdiri dari 10 elemen berupa pointer yang menunjuk ke data bertipe *char*.



Gambar 8.5 Array pointer

- Array pointer bisa diinisialisasi sewaktu pendeklarasian. Sebagai contoh:

```
static char *namahari[] =
{"Senin",
"Selasa",
"Rabu",
"Kamis",
"Jumat",
"Sabtu",
"Minggu"};
```

Pada contoh ini,

namahari[0] berisi alamat yang menunjuk ke string "Senin".

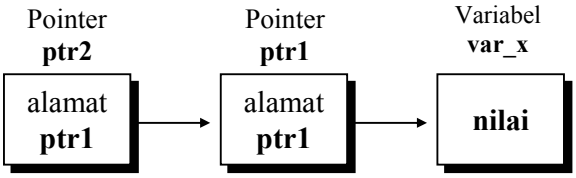
namahari[1] berisi alamat yang menunjuk ke string “Selasa”.

namahari[2] berisi alamat yang menunjuk ke string “Rabu”.

`dan sebagainya.`

Pointer menunjuk Pointer (Pointer to Pointer)

Suatu pointer bisa saja menunjuk ke pointer lain. Gambar berikut memberikan contoh mengenai pointer menunjuk pointer.



Gambar 8.6 Pointer yang menunjuk pointer

- Untuk membentuk rantai pointer seperti pada gambar di atas, pendeklarasian yang diperlukan berupa

```
int var_x;
int *ptr1;
int **ptr2;
```

Perhatikan pada deklarasi di depan:

- **var_x** adalah variabel bertipe int.
- **ptr1** adalah variabel pointer yang menunjuk ke data bertipe *int.*
- **ptr2** adalah variabel pointer yang menunjuk ke pointer *int.*
  (itulah sebabnya deklarasinya berupa *int **ptr2;*)
- Agar **ptr1** menunjuk ke variabel **var_x**, perintah yang diperlukan berupa

```
ptr1 = &var_x;
```

- Sedangkan supaya **ptr2** menunjuk ke **ptr1**, instruksi yang diperlukan adalah

```
ptr2 = &ptr1;
```

C. PERCOBAAN

Untuk setiap program di bawah ini,

- gambarkan ilustrasi alokasi memori dari setiap baris pernyataan yang diproses
- perkirakan hasil eksekusinya

```
1.#include <stdio.h>

  main()
  {
    static char *days[] = {"Sunday", "Monday", "Tuesday",
          "Wednesday","Thursday", "Friday", "Saturday"};

    int i;

    for( i = 0; i < 6; ++i )
      printf( "%s\n", days[i]);
  }
```

2. Pointer yang menunjuk ke pointer yang lain.

```
#include <stdio.h>

main()
{
  int a, *b, **c;

  a = 155;
  b = &a;
  c = &b;

  printf("Nilai a = %d atau %d atau %d\n", a, *b, **c);
  printf("b = %p = alamat a di memori\n", b);
  printf("c = %p = alamat b di memori\n", c);
  printf("alamat c di memori = %p\n", &c);
}
```

3. Pointer yang menunjuk ke pointer yang lain.

```
#include <stdio.h>

main()
{
  int var_x = 273;
  int *ptr1;
  int **ptr2;

  ptr1 = &var_x;
  ptr2 = &ptr1;
  printf("Nilai var_x = *ptr1  = %d\n", *ptr1);
  printf("Nilai var_x = **ptr2 = %d\n\n", **ptr2);
```

```
    printf("ptr1 = &var_x = %p\n", ptr1);
    printf("ptr2 = &ptr1  = %p\n", ptr2);
    printf("       &ptr2  = %p\n", &ptr2);
  }
```

4.
```
#include <stdio.h>
main()
{
   int a, *b, **c;

   a = 1975;
   b = &a;
   c = &b;

   printf("Nilai a = %d atau %d atau %d\n", a, *b, **c);
   printf("b = %p = alamat a di memori\n", b);
   printf("c = %p = alamat b di memori\n", c);
   printf("alamat c di memori = %p\n", &c);

}
```

5. Untuk potongan program di bawah ini, gambarkan ilustrasi alokasi memori dari setiap baris pernyataan yang diproses

```
int *i;
*i = 2;
i++;
int j, k;
*i = j;
j = *i;
i = &j;
k = *(&j);
int **l;
l = &i;
```

6. Tentukan setiap statemen di bawah ini benar atau salah. Jika salah sertakan alasannya.

Deklarasi :

```
int a[5] = {2,4,8,1,23};
int c = 5;
int *ptr1 = &c;
int *ptr2 = a;
```

Statement	Benar (beri tanda X)	Salah (beri tanda X)
`a = c;`		
`*c = 6;`		
`a[2] = c;`		
`*ptr2 = c;`		
`&ptr1 = c;`		
`*(ptr2 + 1) = *(a + 3);`		
`c = *(ptr2 + 1);`		
`c = &ptr1;`		
`c = a[3] + 2;`		
`*(ptr2 + 2) = *ptr1;`		

D. LAPORAN RESMI

Kumpulkan listing program, ilustrasi alokasi memorinya beserta hasil eksekusinya