

TIF1101 – Dasar-Dasar Pemrograman

HO 16 - Struct

Opim Salim Sitompul

Department of Information Technology
Universitas Sumatera Utara



Outline

- 1 Pendahuluan
- 2 Mendeklarasikan Struktur
- 3 Menginisialisasi Struktur
- 4 Mengakes Anggota Struktur
- 5 Array Struktur
 - Mengakes Anggota Struktur
 - Mengirim Struktur sebagai Argumen
- 6 Pointer ke Struktur
 - Membentuk struktur data linked-list
 - Operasi pada struktur data linked-list



Pendahuluan

- Struktur (**struct**) adalah jenis data C untuk mengumpulkan elemen-elemen data yang berbeda jenis di dalam satu kesatuan.
- Bentuk jenis data struktur ini dapat dipandang sebagai sebuah *rekord* yang masing-masing elemen datanya disebut *field*.
- Contoh: **Rekord Mahasiswa**
 - Nomor induk mahasiswa, berupa sebuah string yang terdiri dari 9 karakter.
 - Nama mahasiswa, berupa sebuah string yang terdiri dari 20 karakter.
 - Umur mahasiswa, berupa sebuah bilangan bulat positif tak bertanda.
 - Jenis kelamin, berupa sebuah karakter.



Mendeklarasikan Struktur

- Definisi struktur menggunakan kata kunci **struct**, diikuti elemen-elemen data yang membentuknya.

```
struct nama_struktur
{
    jenis_data elemen_data1;
    jenis_data elemen_data2;
    ...
    jenis_data elemen_dataN;
};
```

Mendeklarasikan Struktur

- Contoh:

```
struct dataMhs
```

```
{
```

```
    char nim[10];
```

```
    char nama[21];
```

```
    unsigned umur;
```

```
    char jenkel;
```

```
};
```

- Pendeklarasian tersebut merupakan pendeklarasian tempat penampungan (*place holder*) dari elemen-elemen data yang akan digunakan.
- Setelah pendeklarasian ini, perlu dideklarasikan variabel-variabel yang mempunyai jenis data struktur itu.



Mendeklarasikan Struktur

- Mendeklarasikan variabel berjenis struktur:
struct dataMhs mhsBaru;
 atau
struct dataMhs *mhsBaru;
- Struktur itu masih menggambarkan informasi untuk seorang mahasiswa.
- Untuk menyajikan informasi lebih dari satu orang mahasiswa, bentuk struktur dapat digabungkan dengan jenis-jenis data lain seperti array atau *linked list*.



Menginisialisasi Struktur

- Sebuah struktur dapat diberi harga awal pada saat mendeklarasikannya atau memberikan harga awal di dalam badan program.
- Contoh:
struct dataMhs mhsBaru = {"181402001", "Zhafar Salim", 18, 'L'};



Mengakses Anggota Struktur

- Ada dua jenis operator yang dapat digunakan, tergantung dari apakah variabelnya adalah sebuah struktur biasa atau sebuah pointer ke struktur.
- Jika variabelnya adalah sebuah variabel struktur biasa, maka pengaksesan terhadap anggota-anggota dapat dilakukan dengan menggunakan operator titik (.).
- Jika variabelnya adalah berupa pointer ke struktur, pengaksesan ke anggota adalah menggunakan anak panah (->).



Mengakes Anggota Struktur

- Contoh:

```
printf("Nomor induk mahasiswa: %s.\n", mhsBaru.nim);  
scanf("%s", &mhsBaru.nama);
```

atau

```
printf("Nomor induk mahasiswa: %s.\n",  
mhsBaru->nim);  
scanf("%s", mhsBaru->nama);
```

- Akan tetap perlu diperhatikan bahwa untuk variabel berjenis **pointer harus terlebih dahulu dialokasikan memori** yang sesuai.

- Contoh:

```
mhsBaru->nama = (char *) malloc(sizeof(char)*21);
```



Array Struktur

- Informasi lebih lengkap dapat diberikan apabila struktur dinyatakan dalam bentuk sebuah daftar.
- Salah satu cara untuk mengumpulkan beberapa buah struktur adalah dengan menggunakan array struktur.
- Bentuk umum pendeklarasian array struktur:

```
struct nama_struktur
{
    jenis_data elemen_data1;
    jenis_data elemen_data2;
    ...
    jenis_data elemen_dataN;
};
```

```
struct nama_struktur variabel_struktur [ukuran_array];
```

Array Struktur

- Contoh:

```
struct dataMhs
```

```
{
```

```
    char nim[10];
```

```
    char nama[21];
```

```
    unsigned umur;
```

```
    char jenkel;
```

```
};
```

```
struct dataMhs mhs2122[150];
```



Mengakses Anggota Struktur

- variabel *mhs2122* adalah sebuah array satu-dimensi berukuran 150, yang masing-masing elemennya merupakan sebuah struktur.
- Ada dua acuan untuk mengakses anggota struktur:
 - 1 mengakses elemen ke-*i* dari array
 - 2 mengakses anggota struktur.
- Misalkan kita ingin memberikan data nama seorang mahasiswa yang terletak pada elemen array ke-10.

```
strcpy(mhs2122[9].nama, "Rommy");
```



Mengakes Anggota Struktur

```
1 /* Program 16.1
2 Nama file: datamhs.c
3 Struct data mahasiswa */
4 #include <stdio.h>
5 #include <stdlib.h>
6 #include <string.h>
7 #define MAXMHS 60
8 struct dataMhs
9 {
10     char *nim;
11     char *nama;
12     char jenkel;
13     unsigned umur;
14 };
15 struct dataMhs mhs2112[MAXMHS];
```

Mengakes Anggota Struktur

```
16 void cetakDataMhs(struct dataMhs[]);
17
18 int main()
19 {
20     int i;
21
22     for(i = 0; i < MAXMHS; i++)
23     {
24         mhs2122[i].nim = (char *) malloc (10);
25         printf("Nim           : ");
26         gets(mhs2122[i].nim);
27         printf("Nama           : ");
28         mhs2122[i].nama = (char *) malloc (21);
29         gets(mhs2122[i].nama);
30         printf("Jenis Kelamin : ");
31         scanf("%c", &mhs2122[i].jenkel);
```

Mengakes Anggota Struktur

```
32     printf("Umur          : ");
33     scanf("%u", &mhs2122[i].umur);
34     printf("\n");
35     fflush(stdin);
36     getchar();
37 }
38 cetakDataMhs(mhs2122);
39
40 return 0;
41 }
```

Mengakes Anggota Struktur

```

42 void cetakDataMhs(struct dataMhs mhs2122[])
43 {
44     int i;
45     printf("Data Pribadi Mahasiswa.\n");
46     printf("-----\n");
47     printf("%-10s%-21s%-5s%-4s\n", "NIM", "NAMA",
48         "UMUR", "L/P");
49     printf("-----\n");
50     for(i = 0; i < MAXMHS; i++)
51     {
52         printf("%-10s", mhs2122[i].nim);
53         printf("%-22s", mhs2122[i].nama);
54         printf("%c", mhs2122[i].jenkel);
55         printf("%2d\n", mhs2122[i].umur);
56     }
57     printf("-----\n");

```


Mengirim Struktur sebagai Argumen

- Mengirimkan struktur sebagai argumen fungsi dapat dilakukan seperti pada pengiriman argumen fungsi lainnya.
- Untuk memudahkan akses jenis data struct, digunakan *typedef*.
 - yakni pada bagian argumen formal fungsi yang menerima, struktur tersebut dideklarasikan menggunakan jenis data baru *typedef* tersebut.
- Hal ini karena telah dibuat jenis data baru bernama *dataMhs* dari struktur *rekordDataMhs* menggunakan *typedef*.
- Dengan cara ini, sebuah struktur dapat dikirim hanya dengan menuliskan namanya pada argumen aktual ataupun pada argumen formal.



Mengakes Anggota Struktur

```

1  /* Program 16.2
2  Nama file: fdatamhs.c
3  Program untuk mencatat data mahasiswa
   menggunakan fungsi */
4  #include <stdio.h>
5  #include <stdlib.h>
6  #define MAXMHS 2
7
8  struct rekordDataMhs
9  {
10     char nim[9];
11     char nama[21];
12     char jenkel;
13     unsigned umur;
14 };
15 typedef struct rekordDataMhs dataMhs;
    
```

Mengakes Anggota Struktur

```
16 dataMhs inputDataMhs(dataMhs);
17 void cetakDataMhs(dataMhs []);
18
19 int main()
20 {
21
22     dataMhs mhsRek, mhs2122[MAXMHS];
23     int i;
24
25     /* Input data mahasiswa */
26     for(i=0; i<MAXMHS; i++)
27         mhs2122[i] = inputDataMhs(mhsRek);
28     /* Output data mahasiswa */
29     cetakDataMhs(mhs2122);
30     return 0;
31 }
```

```
32 dataMhs inputDataMhs(dataMhs rekMhs)
33 {
34     printf("Nomor induk mahasiswa: ");
35     gets(rekMhs.nim);
36     printf("Nama mahasiswa: ");
37     gets(rekMhs.nama);
38     printf("Jenis kelamin: ");
39     rekMhs.jenkel = getchar();
40     printf("Umur : ");
41     scanf("%u", &rekMhs.umur);
42     printf("\n");
43     fflush(stdin);
44
45     return rekMhs;
46 }
```

Mengakes Anggota Struktur

```

47 void cetakDataMhs (dataMhs mhs2122[])
48 {
49     int i;
50     printf("Data Pribadi Mahasiswa.\n");
51     printf("-----\n");
52     printf("%-10s%-21s%-5s%-4s\n", "NIM", "NAMA",
53         "UMUR", "L/P");
54     printf("-----\n");
55     for(i = 0; i < MAXMHS; i++)
56     {
57         printf("%-10s", mhs2122[i].nim);
58         printf("%-21s", mhs2122[i].nama);
59         printf("%c", mhs2122[i].jenkel);
60         printf("%2d\n", mhs2122[i].umur);
61     }
62     printf("-----\n");

```

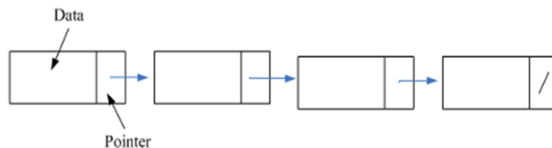
Mengakes Anggota Struktur

- Beberapa struktur dapat disatukan dengan menggunakan *pointer* dan melaukan alokasi memori secara dinamik.
- Salah satu implementasi teknik pengalokasian memori secara dinamik ini adalah pada pembentukan struktur data yang disebut *linked-list* (daftar berkait).



Membentuk struktur data linked-list

- Struktur data *linked-list* adalah sebuah untaian data yang antara satu elemen data dan elemen data berikutnya dihubungkan dengan sebuah *pointer*.



Gambar 1: Struktur data *linked-list*

- Elemen-elemen data pada *linked-list* digambarkan sebagai sebuah simpul (*node*) yang terdiri dari dua bagian.
 - bagian informasi atau bagian data, berupa anggota-anggota dari sebuah jenis data struktur.
 - sebuah pointer yang menunjuk ke jenis data struktur berikutnya.
- Pada akhir *linked-list*, bagian keduanya berupa sebuah pointer yang tidak menunjuk ke jenis data struktur lain, disimbolkan dengan karakter garis miring (/) menyatakan NULL.



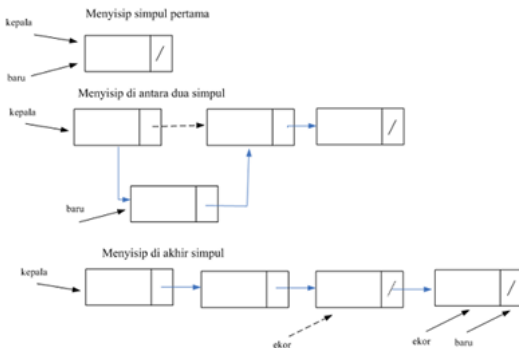
Membentuk struktur data linked-list

- Mendeklarasikan struktur data *linked-list*:

```
typedef struct titikSimpul titikKoordinat;  
struct titikSimpul  
{  
    int absis;  
    int ordinat;  
    titikKoordinat * next;  
};
```

- Ada dua operasi dasar pada struktur data

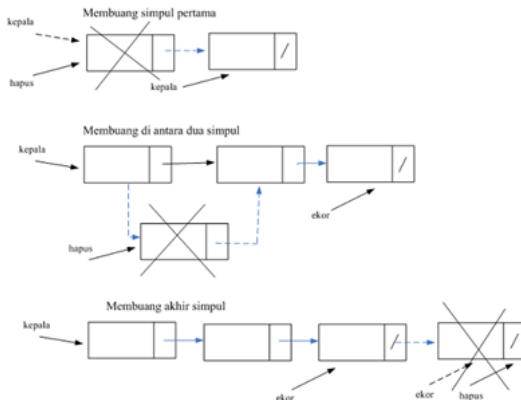
- 1 Menyisip simpul baru.



Gambar 2: Menyisip simpul baru

Operasi pada struktur data linked-list

2 Membuang simpul.



Gambar 3: Membuang simpul



Operasi pada struktur data linked-list

```
1 /* Program 16.3
2 Nama file: koord.c
3 Program untuk menyusun titik koordinat dalam
   linked list */
4
5 #include <stdio.h>
6 #include <stdlib.h>
7
8 typedef struct titikSimpul titikKoord;
9 struct titikSimpul
10 {
11     int absis;
12     int ordinat;
13     titikKoord *berikut;
14 };
```

Operasi pada struktur data linked-list

```

15 titikKoord *tambahTitik(titikKoord *);
16 titikKoord *buangTitik(titikKoord *);
17 void cetakDaftar(titikKoord *);
18
19 int main()
20 {
21     titikKoord *kepala = NULL;
22     char pilih = '0';
23     do
24     {
25         system("cls");
26         printf("Menu:\n");
27         printf("1. Menambah titik.\n");
28         printf("2. Membuang titik.\n");
29         printf("3. Mencetak daftar.\n");
30         printf("4. Selesai.\n");

```

Operasi pada struktur data linked-list

```
31     printf("Pilihan: ");
32     pilih = getchar();
33     switch(pilih)
34     {
35         case '1': kepala = tambahTitik(kepala);
36             break;
37         case '2': kepala = buangTitik(kepala);
38             break;
39         case '3': cetakDaftar(kepala);
40             break;
41     }
42     fflush(stdin);
43 }
44 while (pilih < '4');
45 return 0;
46 }
```

Operasi pada struktur data linked-list

```
47 titikKoord *tambahTitik(titikKoord *kepala)
48 {
49     titikKoord *baru, *ekor;
50
51     baru = (titikKoord *) malloc(sizeof(
        titikKoord));
52
53     printf("Berikan absis   : ");
54     scanf("%d", &baru->absis);
55     printf("Berikan ordinat: ");
56     scanf("%d", &baru->ordinat);
57     baru->berikut = NULL;
58
59     if(kepala == NULL)
60         kepala = baru;
```

Operasi pada struktur data linked-list

```
61  else
62  {
63      ekor = kepala;
64      while (ekor->berikut != NULL)
65          ekor = ekor->berikut;
66      ekor->berikut = baru;
67  }
68  return kepala;
69 }
70
71 titikKoord *buangTitik(titikKoord *kepala)
72 {
73     int x, y, ada = 0;
74     titikKoord *cari, *hapus;
```



Operasi pada struktur data linked-list

```
75 printf("Titik mana?\n");
76 printf("Berikan absis dan ordinat: ");
77 scanf("%d %d", &x, &y);
78 cari = kepala;
79 while((cari != NULL) && !ada)
80 {
81     if((cari->absis == x) && (cari->ordinat ==
82         y))
83         ada = 1;
84     else
85         cari = cari->berikut;
86 }
87 if(ada)
88 {
89     hapus = kepala;
```

Operasi pada struktur data linked-list

```
89     if(hapus == cari)
90     {
91         kepala = kepala->berikut;
92         cari->berikut = NULL;
93         free(cari);
94     }
95     else
96     {
97         while ((hapus->berikut != NULL) &&\
98             (hapus->berikut !=\cari))
99         {
100             hapus = hapus->berikut;
101             hapus->berikut = cari->berikut;
102             cari->berikut = NULL;
103             free(cari);
104         }
```

Operasi pada struktur data linked-list

```

105     else
106     {
107         printf("Titik (%d,%d) tidak ada dalam
           daftar.\n", x, y);
108         printf("Tekan <ENTER>\n");
109         fflush(stdin);
110         getchar();
111     }
112     return kepala;
113 }
114
115 void cetakDaftar(titikKoord *kepala)
116 {
117     titikKoord *cetak;
118
119     printf("Daftar titik koordinat: \n");
    
```

Operasi pada struktur data linked-list

```
120     cetak = kepala;
121     while(cetak != NULL)
122     {
123         printf("(%d,%d) ", cetak->absis, cetak->
            ordinat);
124         cetak = cetak->berikut;
125     }
126     printf("\nTekan <enter>...\n");
127     fflush(stdin);
128     getchar();
129 }
```

