

SPM2102

BAHASA PENGATURCARAAN 1

TATASUSUNAN

Disediakan oleh:

NORASYKIN MOHD ZAID

NORAH MD NOOR



TATASUSUNAN

- Pengisytiharaan tatasusunan satu dimensi
 - Mengumpukan nilai awal pada tatasusunan satu dimensi
 - Tatasusunan dan fungsi
 - Tatasusunan dua dimensi (multidimensi)
 - Tatasusunan jenis rentetan
-

TATASUSUNAN

- Satu **struktur data** yang boleh digunakan untuk menyimpan senarai nilai daripada jenis data tertentu
- Merupakan teknik yang amat sesuai untuk **mencetak atau membaca data yang lebih daripada satu**
- Sebagai contoh: Markah bagi 100 orang pelajar dapat dicetak dan dibaca dengan mudah dengan hanya menggunakan satu pembolehubah
- **Lebih cepat dan memudahkan kerja** berbanding dengan cara biasa di mana sesuatu nilai data disimpan menggunakan satu pembolehubah

TATASUSUNAN

- Tatasusunan (array) dalam C++ merujuk kepada suatu koleksi jujukan nilai yang:
 - Mempunyai **jenis data yang sama**
 - Menggunakan **nama pembolehubah yang sama**; yakni satu pembolehubah
 - Mempunyai **saiz yang tertentu**
 - Mempunyai **subskrip/indeks**
 - Subskrip merupakan lokasi ingatan yang diperuntukan oleh ingatan komputer untuk sesuatu data.
 - Subskrip bermula dengan **nilai 0**.

TATASUSUNAN SATU DIMENSI

- Pengisytiharaan tatasusunan satu dimensi menggunakan sintaks:

jenis_pembolehkan nama_pembolehkan[saiz]

- Jenis pembolehkan merupakan **jenis data asas** (char, int, float, double)
- Nama pembolehkan terdiri dari **sebarang nama** mengikut **peraturan pembinaan pembolehkan**
- Saiz merujuk kepada **bilangan ruangan ingatan** (sel ingatan) atau elemen yang diperuntukkan kepada pembolehkan
 - Terdiri daripada pemalar, pembolehkan atau ungkapan bernilai integer
 - Contoh:
 - Pemalar markah[5];
 - Pembolehkan gred[SAIZ];
 - Ungkapan result[MAKS-1];

TATASUSUNAN SATU DIMENSI

Perbandingan penggunaan pembolehubah dan tatasusunan

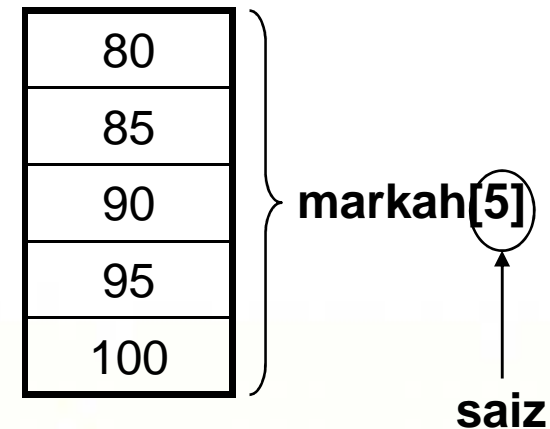
1. Pembolehubah

```
int markah1 = 80;
int markah2 = 85;
int markah3 = 90;
int markah4 = 95;
int markah5 = 100;
```

2. Tatasusunan

```
markah[0]
markah[1]
markah[2]
markah[3]
markah[4]
```

Subskrip



MEMPROSES TATASUSUNAN

- Perlu menggunakan indeks
- Indeks merupakan suatu pemalar, pembolehubah atau ungkapan yang bernilai integer yang menyatakan unsur manakah di dalam tatasusunan yang hendak dicapai
 - Pemalar `markah[1];`
 - Pembolehubah `markah[i];`
 - Ungkapan `markah[n-1];`

MEMPROSES TATASUSUNAN

Contoh untuk menyimpan lima markah ujian

```
1. float markah[5];
   for(int i=0; i < 4; i++)
       markah[i] = 20.0;
   markah[4] = 15.0;
```

20.0	markah[0]
20.0	markah[1]
20.0	markah[2]
20.0	markah[3]
15.0	markah[4]

Indeks berupa pembolehubah

NILAI AWAL TATASUSUNAN

- Terdapat dua cara untuk menilai awalkan tatasusunan
 - Semasa proses pengisytiharaan tatasusunan
 - Semasa di dalam badan aturcara
- Sintaks pengumpulan nilai awal tatasusunan semasa pengisytiharaan adalah:

jenis_data nama_pembolehkan[saiz] = { senarai data };

NILAI AWAL TATASUSUNAN

Contoh pengumpulan nilai awal tatasusunan semasa pengisytiharaan:

- int markah[5]={20, 10, 15, 18, 25};
- char gred[3]={'a', 'b', 'c'};
- Saiz tatasusunan dapat diabaikan apabila pengumpulan nilai berlaku semasa proses pengisytiharaan
 - char bangsa[]={'m', 'c', 'i', 'L'};

NILAI AWAL TATASUSUNAN

- Pengumpulan nilai awal tatasusunan juga boleh berlaku untuk sebahagian data sahaja
 - Contoh:
 - `int umur[5]={18, 19, 20};`
 - `char huruf[3]='a';`
- Bagi unsur yang tiada umpukan nilai, pengkompil akan mengumpulkan nilai 0 untuk jenis data int dan float, sementara null untuk char

NILAI AWAL TATASUSUNAN

- Contoh
- `int umur[5]={16, 17, 18};`

- umur[0]

16

- umur[1]

17

- umur[2]

18

- umur[3]

0

- umur[4]

0

NILAI AWAL TATASUSUNAN

- Sintaks pengumpulan nilai awal tatasusunan dalam badan aturcara pula adalah:

nama_pembolehkan[subskrip]=nilai;

- Pengumpulan nilai berlaku secara satu-persatu untuk semua elemen tatasusunan
- Contoh:

Markah[0]=20;

Markah[1]=15;

Markah[2]=25;

Markah[3]=10;

Markah[4]=5;

UMPUKAN NILAI

- Pengumpulan nilai boleh berlaku dalam tiga bentuk:
 - Pemalar contoh: `markah[0]=20;`
 - Pembolehubah contoh: `markah[1]=skor_rendah;`
 - Ungkapan contoh: `markah[2]=skor_tinggi-1;`
- Pengumpulan nilai daripada satu tatasusunan ke satu tatasusunan juga boleh berlaku dengan syarat ianya mesti dilakukan satu persatu.

UMPUKAN NILAI

- Contoh:
- `int markah_ujian1[5]; markah_ujian2[5];`
`markah_ujian1= markah_ujian2 //tidak sah`
- `for(int i=0; i<=5; i++)`
`markah_ujian1[i]=markah_ujian2[i];//sah`

TATASUSUNAN DUA DIMENSI

- Tatasusunan satu dimensi hanya membenarkan suatu nilai disimpan dalam satu satah
- Tatasusunan dua dimensi membenarkan nilai disimpan dalam dua satah (baris dan lajur)
 - Cth vektor 3x3

1 x 1	1 x 2	1 x 3
2 x 1	2 x 2	2 x 3
3 x 1	3 x 2	3 x 3

TATASUSUNAN DUA DIMENSI

- Pengisytiharan tatasusunan dua dimensi

jenis_pembolehubah nama_pembolehubah[saiz baris][saiz lajur]

Contoh:

```
int vektor[3][3];
```

```
float markah[x][y]
```

TATASUSUNAN DUA DIMENSI

- Contoh:

```
const int x = 3;
```

```
const int y = 3;
```

```
float harga[x][y];
```

[0][0]	[0][1]	[0][2]
[1][0]	[1][1]	[1][2]
[2][0]	[2][1]	[2][2]

- Tatasusunan harga akan menyediakan sembilan lokasi (sel ingatan) bagi menyimpan nilai

MENILAI AWALKAN TATASUSUNAN

- Sintaks pengumpulan nilai awal tatasusunan semasa pengisytiharaan adalah:

jenis_data nama_pembolehubah[saiz baris] [saiz lajur]= {senarai data};

- *Contoh:*
- *int number[4][5]={{1,2,3,4,5}, {2,3,4,5,6}, {7,8,9}, {2}}*
- *Int result [3][2] = {{45,55}, {90, 85}, {75, 69}}*

TATASUSUNAN JENIS RENTETAN

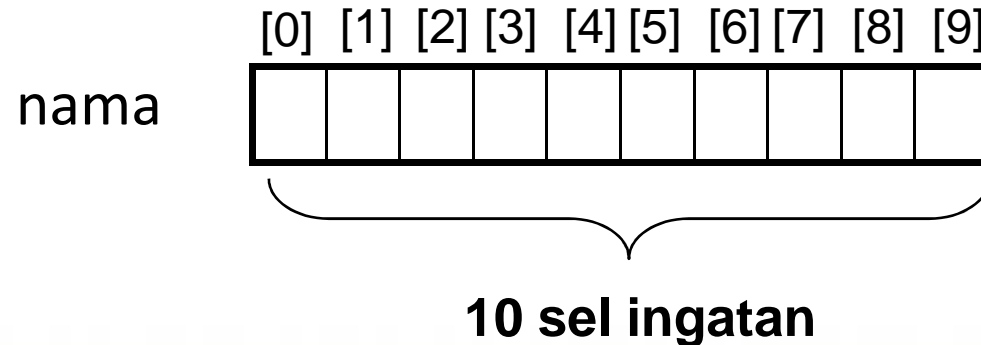
- Rentetan adalah suatu tatasusunan khas
- Terdiri daripada jenis data char dan diakhiri dengan aksara null '\0'
- Pembolehkan rentetan dapat diisytiharkan seperti pengisytiharaan tatasusunan
- Contoh:
 - *char nama_pembolehkan[saiz]*

TATASUSUNAN JENIS RENTETAN

- Contoh:
- `char nama[30];`
- `char abjad[10];`
- Dengan pengisytiharaan ini, sebanyak 30 dan 10 sel ingatan diperuntukkan kepada pembolehubah nama dan abjad

TATASUSUNAN JENIS RENTETAN

- Contoh:
- `char nama[10];`



TATASUSUNAN JENIS RENTETAN

- Pemalar rentetan, pada kebiasaannya dinilai awalkan seperti di bawah:
- Contoh:
- `char simbol [10] = {'a', '*', '1'};`

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
a	*	1							

- Ianya juga boleh dinilai awal menggunakan tanda “ dan diakhiri dengan ”
- Contoh:
- `char simbol [10] = "a*1";`

TATASUSUNAN JENIS RENTETAN

- Pengolahan atau pengubahsuaian tatasusunan rentetan dapat dilakukan dengan menggunakan fail kepala string.h
- Antara fungsi sedia ada yang boleh digunakan untuk tujuan pengubahsuaian ini adalah seperti:
 - strcpy()
 - strcmp()
 - strlen()
 - strcat()

TATASUSUNAN JENIS RENTETAN

- `strcpy(string1, string2)`
 - Fungsi ini digunakan untuk menyalin satu rentetan kepada pembolehubah rentetan yang lain

```
char rentetan[10];  
strcpy(rentetan, "Hello");
```

- Rentetan "Hello" akan disalin ke pembolehubah rentetan

TATASUSUNAN JENIS RENTETAN

- `strcmp(string1, string2)`
 - Fungsi ini digunakan untuk membuat perbandingan rentetan yang disimpan dalam dua pembolehubah
 - Nilai sifar akan dipulangkan jikalau `string1` sama dengan `string2` (palsu)
 - Nilai positif akan dipulangkan jikalau `string1 > string2` (besar daripada 0, benar)
 - Nilai negatif akan dipulangkan jikalau `string1 < string2` (kurang daripada sifar, benar)

TATASUSUNAN JENIS RENTETAN

Contoh (*cuba kod ini, lihat hasilnya*):

```
#include<iostream.h>
#include<string.h>
void main()
{
    char nama1[10] = "ALFA" , nama2[10]="ROMEO";

    //pengujian string
    if(strcmp(nama1, nama2))
        //benar
        cout<<"nama berbeza\n";
    else
        //palsu
        cout<<"nama sama \n";
}
```

TATASUSUNAN JENIS RENTETAN

- `strlen(string1)`
 - Fungsi ini digunakan untuk mengira panjang bagi suatu rentetan yang dibaca

```
char rentetan[10];
```

```
cout<<"Masukkan nama anda: ";
```

```
cin>>rentetan;
```

```
cout<<"Panjang nama adalah "<<strlen(rentetan);
```

TATASUSUNAN JENIS RENTETAN

- `strcat(string1, string2)`
 - Fungsi ini digunakan untuk menambahkan rentetan dari pembolehubah `string2` kepada `string1` (di akhir `string1`) dan kandungan `string2` tidak berubah

```
char rentetan1[10], rentetan2[10];  
strcpy(rentetan1, "Hello");  
strcpy(rentetan2, "Semua");  
strcat(rentetan1, rentetan2);
```

Rujukan

- Jabatan Multimedia Pendidikan (2004), *Pengenalan kepada Bahasa Pengaturcaraan dalam Konteks Bahasa C++* , Faculti Pendidikan, UTM (unpublished)
- Abdullah Zawawi Talib, Ahamad Tajudin Khader, Maziani Sabudin, Wahidah Husain (2002), *Prinsip-Prinsip Pengaturcaraan Menggunakan C++ : Edisi kedua*, Pearson Malaysia, ISBN 983-9236-87-3
- Bakar, M. A et al (2002). Pengaturcaraan C. Malaysia. Prentice Hall
- Deitel, P. J., & Deitel, H. M. (2005). *C++ How to Program (5th Edition)*, Prentice Hall