REVIEW ALGORITMA PEMROGRAMAN

“Array dan Pointer”

Indra Hermawan, S.Kom, M.Kom

indrah13@gmail.com / indra@nurulfikri.ac.id

No. 085217987034

Tujuan

- Mahasiswa memahami makna dan kegunaan array
- Mahasiswa dapat menggunakan notasi pendefinisian dan pengacuan array dengan benar hingga proses pencarian terhadap elemen array
- Mahasiswa dapat membuat program dengan menggunakan array

Mengolah Data

- Tuliskan program yang menerima 3 nama, lalu menampilkan semua kombinasi pasangan nama.
- Contoh keluaran:

```
int main () {
 // KAMUS
 string nama1, nama2, nama3;
 //ALGORITMA
    cin >> nama1;
    cin >> nama2;
    cin >> nama3;
    cout << nama1 " - " nama2 << endl;
    cout << nama1 " - " nama3 << endl;
    cout << nama2 " - " nama3 << endl;
}
```

```
Ali
Budi
Caca
Ali - Budi
Ali - Caca
Budi - Caca
```

Mengolah 10 Data

- Tuliskan program yang menerima 10 nama, lalu menampilkan semua kombinasi pasangan nama.
- Contoh keluaran:

```
int main () {
  // KAMUS
  string nama1, nama2, nama3,nama 4, nama5;
  string nama6, nama7, nama8, nama9, nama10;
  //ALGORITMA
  cin >> nama1;
  cin >> nama2;
  … // lanjutkan sendiri!!
  cin >> nama10;

  cout << nama1 << " - " << nama2 << endl;
  cout << nama1 << " - " << nama3 << endl;
  … // lanjutkan sendiri!!
  cout << nama9 << " - " << nama10 << endl;

}
```

```
Nama-1: Ali
Nama-2: Budi
…
Nama-9: Ina
Nama-10: Jaja
Ali – Budi
Ali – Caca
…
Ina - Jaja
```

Mengolah N Data

- Anda diminta menampilkan semua kombinasi pasangan nama yang mungkin dari …

  100 nama ???

  Bagaimana kalau…

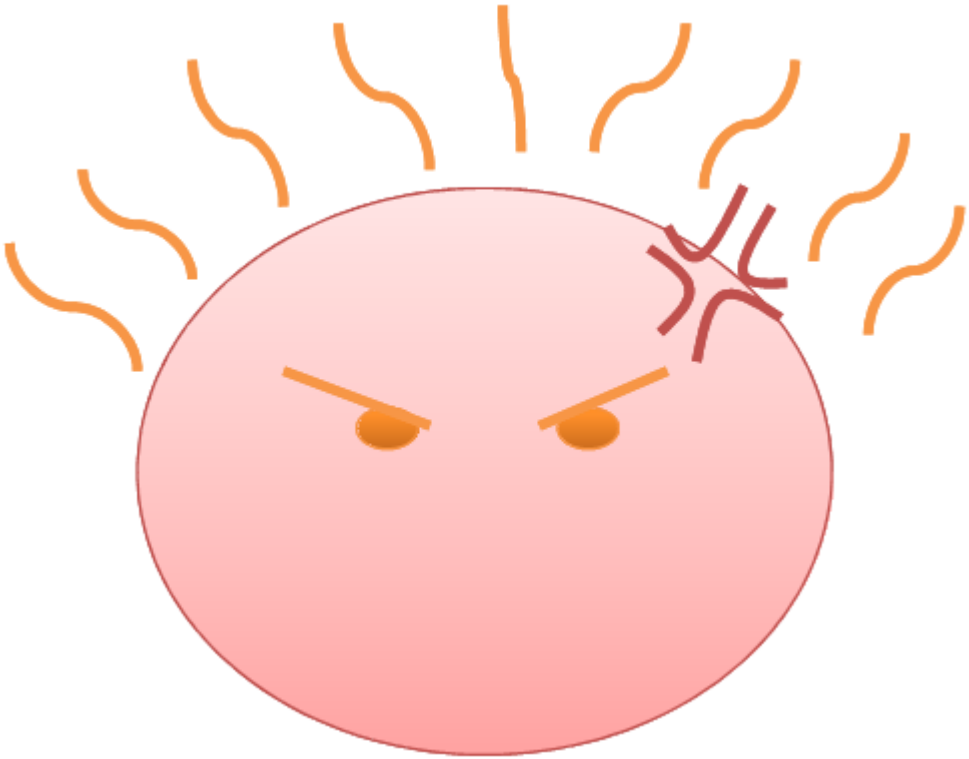
  100 nama ???

  1000 nama ???

  10000 nama ???

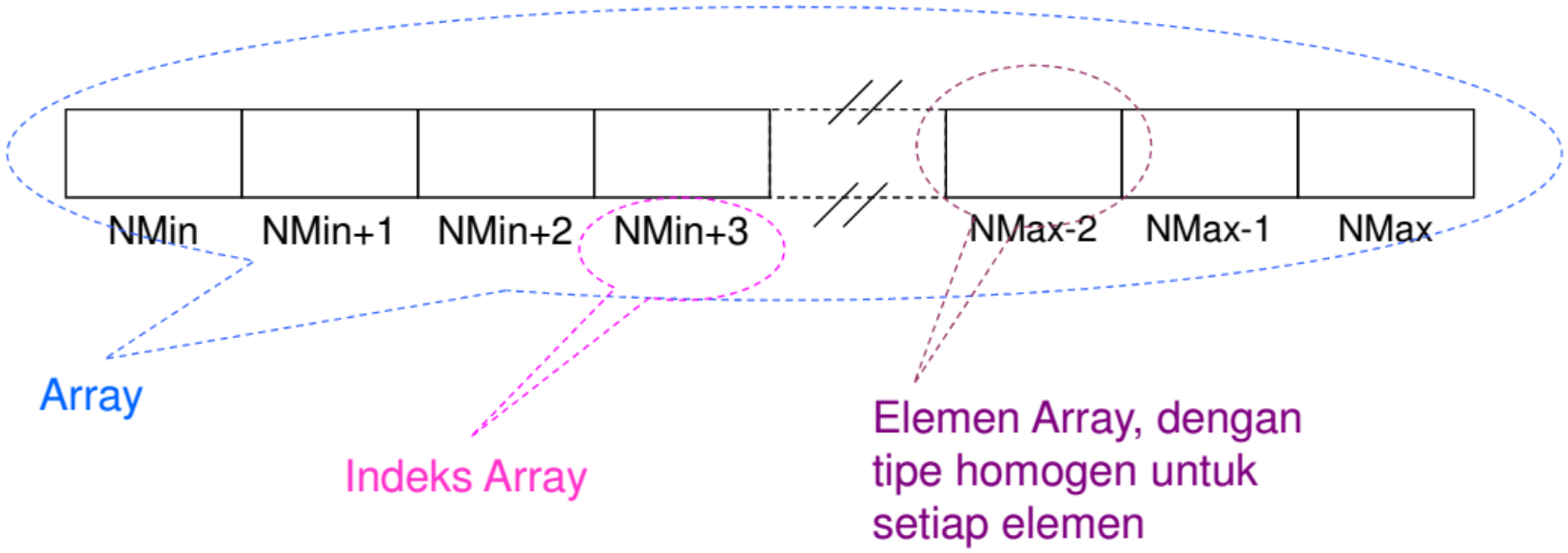
  1000000 nama ???

  ….



Array / Tabel / Vektor / Larik

- Type array adalah type yang mengacu kepada sebuah atau sekumpulan elemen melalui indeks
- Merepresentasikan sekumpulan informasi yang bertypesama dan disimpan dengan urutan yang sesuai dengan definisi indeks



Array dalam C/C++ (1)

- Variabel dapat dideklarasikan ber-type array dari suatu type tertentu
- Setiap elemen array diakses dengan alamat berupa indeks yang bertype integer
- Cara deklarasi: <type> <namatabel>[<ukuran>];
- Contoh: int TabInt[10];
- Array bernama TabInt dengan setiap elemen bertype integer, dengan ukuran 10 elemen, dengan alamat setiap elemen array (indeks) adalah dari indeks ke-0 s.d. 9

Array dalam C/C++ (2)

- Cara akses elemen: <namatabel>[<indeks>]
- Contoh: int TabInt[10];

1	2	4	-1	100	2	0	-1	3	9
0	1	2	3	4	5	6	7	8	9

  - cout << TabInt[4]; // akan tercetak: 100
  - int x = TabInt[0] + TabInt[5]; // x bernilai 3
  - TabInt[9] = 8; // Elemen array indeks 9 menjadi 8
  - TabInt[10] ??? // Berada di luar range, tidak terdefinisi!!

Contoh Deklarasi Array yang lain

- Elemen dari array dapat diakses langsung jika dan hanya jika indeks terdefinisi
- Cara mengacu sebuah elemen:
  - TabInt[2]
  - TabInt[i] jika i terdefinisi

```
int main() {
    // KAMUS
    int TabJumlahHari[12]; // indeks 0..11
    float TabNilai[15];    // indeks 0..14
    char TabHuruf[100];    // indeks 0..99
    string TabKata[100];   // indeks 0..99
    Point TabTitik[20];    // indeks 0..19, Point terdefinisi
    // ALGORITMA
    …
}
```

Mengisi Array

- Mengisi array merupakan aktifitas memberi nilai elemen array
  - – Pemberian nilai satu elemen, contoh: TabInt[0]=31;
  - – Pemberian nilai beberapa elemen, contoh:
    - for (i=0;i<10;i++) {
    - TabInt[i]=i*10;
    - }
- Hati-hati!
  - Jangan mengakses elemen yang indeks-nya berada di luar definisi, misalnya TabInt[10] index ke-10 tidak terdefinisi untuk TabInt
  - Jangan membaca elemen yang belum diisi nilainya

Mengisi dan membaca isi array

- Elemen array yang telah diberi nilai dapat diakses kembali
- Contoh berikut menampilkan semua isi array ke layar

```
// File: isibacaarray.cpp
// Mengisi array dan menampilkan
// seluruh elemen pada array
#include <iostream>
using namespace std;
int main ()
{  // KAMUS
   int TabInt[10]; int i;

   // Algoritma

   // mengisi array
   for (i=0; i<10; i++) {
       TabInt[i]=i*10;
   }
   // membaca dan menuliskan isi
   // array ke layar
   for (i=0; i<10; i++) {
       cout << TabInt[i] << endl;
   }
   return 0;
}
```

Memproses Array

- Pemrosesan koleksi data pada array dilakukan secara sekuensial
- Asumsi : seluruh elemen array terdefinisi
- Contoh: menjumlahkan data dan menghitung rata-rata

```
// File: sumArray.cpp
// menghitung jumlah seluruh elemen pada array
#include <iostream>
using namespace std;
int main ()
{   // KAMUS
    int sum, i;
    int TabInt[10];
    // ALGORITMA
    // mengisi data nilai dari input user
    cout << "Isilah 10 data nilai dalam range 0-100:"
<< endl;
    for (i=0; i<10; i++) {
        cin >> TabNilai[i];
    }
    // menjumlahkan nilai dan menghitung rata-rata
    cout << "Data input:" << endl;
    sum=0;
    for (i=0; i<10; i++) {
        cout << TabInt[i] << endl;
        sum = sum + TabInt[i];
    }
    cout << "Rata-rata: ";
    cout << (float)sum/10.0 << endl;
    return 0;
}
```

Mencari Indeks Suatu Nilai (searching)

- Dengan asumsi semua elemen array terdefinisi, dapat dilakukan pencarian indeks suatu nilai ditemukan pertama kali dalam array

```
// File: searchArray.cpp
// mencari indeks di mana nilai ditemukan
#include <iostream>
using namespace std;
int main ()
{  // KAMUS
   int X, i; bool found;
   int TabInt[10];
   // ALGORITMA
   // Pengisian data: asumsi array terisi

   // mencari suatu nilai, yaitu X
   cin >> X;
   i = 0; found = false;
   while ((i < 10) && (!found)) {
      if (TabInt[i]==X) {
         found = true;
      } else {
         i++;
      }
   } // i = 10 atau found
   if (found) { // X ada di
      cout << X << " ada di indeks " << i;
   } else {
      cout << X << " tidak ditemukan";
   }
   return 0;
}
```

Mencari Nilai Ekstrim Array

- Dengan asumsi array tidak kosong, dapat dilakukan pencarian elemen array bernilai ekstrim
- Contohnya: mencari nilai maksimum

```
// File: maxArray.cpp
// mencari nilai maksimum pada array
#include <iostream>
using namespace std;
int main ()
{  // KAMUS
   int max, i;
   int TabInt[10];
   // Algoritma
   // Pengisian data: Buat sebagai
   // latihan
   // mencari nilai maksimum
   max=TabInt[0]; //inisialisasi
                  //max dgn elemen ke-0
   for (i=1; i<10; i++) {
       //ganti max kalau ada nilai elemen
       //array yang lebih besar
       if (TabInt[i]>max) {
           max=TabInt[i];
       }
   }
   cout << "Nilai maksimum: “ << max <<
endl;
   return 0;
}
```

Latihan

*POINTER

Pengantar mengenai memori komputer

- Sebelum berbicara mengenai pointer, diperlukan pengetahuan mengenai memori komputer
- Apakah fungsi memori komputer?
- Bagaimana hubungan program, variabel, dan memori?

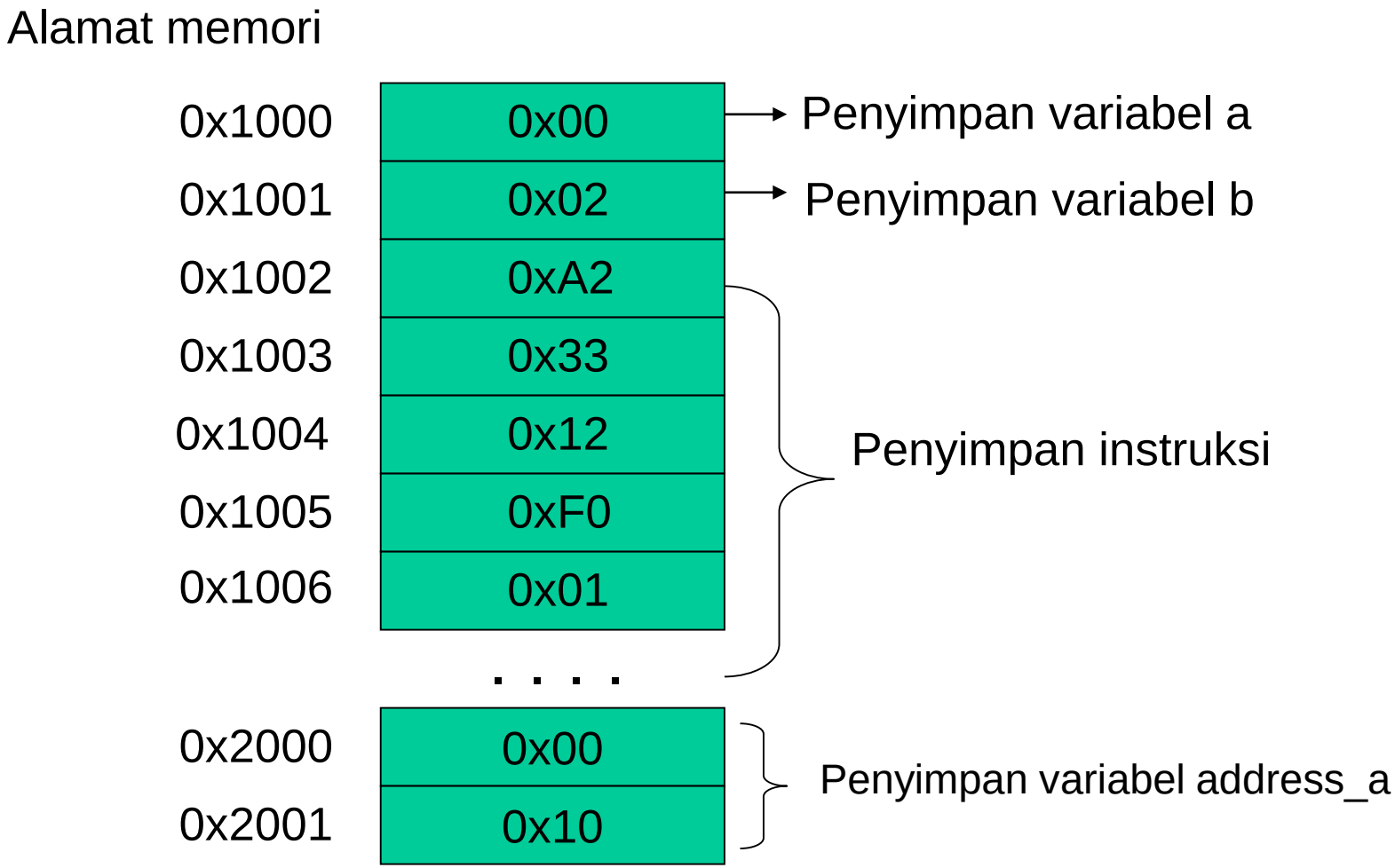
Memori komputer

- RAM (Random access memory)
  - Istilah untuk memory yang bersifat volatile (isi memori akan hilang jika catu daya dicabut)
  - RAM bisa dibaca dan ditulisi data secara cepat
  - Contoh: DDRAM, SDRAM, cache memory
- ROM (Read Only Memory)
  - Bersifat non volatile (data masih eksis di memori walaupun cata daya dicabut)
  - ROM bisa dibaca secara cepat, tapi untuk menulisi ROM butuh sekuens/urutan tertentu
  - Contoh: ROM BIOS
- Pada mata kuliah ini, memori yang akan dibahas adalah RAM

RAM

- Fungsi RAM:
  - Menyimpan instruksi-instruksi hasil kompilasi program
  - Menyimpan data-data variabel
- Ketika suatu program di-compile, beberapa byte memori dari RAM akan dialokasikan untuk menyimpan instruksi dan menyimpan data variabel
- Alamat memori yang dialokasikan ditentukan secara otomatis oleh compiler

Ilustrasi penggunaan RAM



Reference Operator (&)

- Alamat memori tempat suatu variabel disimpan disebut *reference* dari variabel tersebut
- *Reference* suatu variabel diakses melalui suatu *reference operator* (&)
- Secara bahasa, operator '&' bisa diartikan menjadi 'alamat dari'

Contoh

- Suatu variabel:
  - unsigned char a=0x00;
- Variabel a mempunyai 2 parameter :
  - nilai dari variabel a → 0x00
  - Alamat memori tempat variabel a disimpan: dibaca dengan perintah &a
- Misalnya: menggunakan susunan memori pada ilustrasi slide 5
- `unsigned char a=0x00;`
- `address_a=&a; //variabel address_a akan berisi 0x1000`
- Alamat memori ini tidak bisa diprediksi nilainya, karena nilai alamat memori ditentukan oleh compiler
- *address_a* disebut sebagai variabel yang menyimpan reference(alamat memori) untuk variabel a
- Variabel *address_a* bertipe apa??

Pengertian Pointer

- Variabel yang menyimpan reference (alamat memori) untuk variabel lain disebut pointer
- Dalam C dan C++ pointer adalah suatu tipe data tersendiri
- Tipe data pointer digunakan untuk menyimpan alamat memori suatu variabel atau bisa dikatakan ‘menunjuk suatu variabel’

Deklarasi tipe data pointer

- Contoh :
  - *unsigned int *my_pointer;*

- Deklarasi di atas menyatakan deklarasi variabel *my_pointer* yang bertipe pointer untuk variabel tipe unsigned int (pointer to unsigned int)
- Variable *my_pointer* digunakan untuk menyimpan suatu alamat memori suatu variabel bertipe unsigned int
- Deklarasi suatu pointer diawali dengan tanda '*' sebelum nama variabelnya

Dereference operator

- Selain untuk deklarasi pointer, tanda '*' juga berfungsi untuk dereference operator
- Dereference adalah kebalikan reference
- Reference → operasi melihat alamat suatu variabel
- Dereference → operasi untuk melihat isi dari suatu alamat memori
- Contoh : a = *my_pointer;
- Artinya: my_pointer berisi suatu alamat memori, dan variabel a akan berisi data yg tersimpan di alamat memori tsb.

Contoh penggunaan pointer

```
unsigned int a, *pointer_a;


a = 10; //variabel a diisi nilai 10
pointer_a = &a; //pointer_a berisi alamat memori dari a
//setelah perintah di atas, nilai *pointer_a akan berisi //sama
   dengan a
cout << “nilai a : “ << a;
cout  << “nilai *pointer_a” << *pointer_a;
```

Mengubah nilai variabel

- Mengubah nilai variabel secara langsung:
  - `a = 250;`
- Mengubah nilai variabel by-reference (tidak langsung)
  - `unsigned int a,*pointer_a;`
  - `pointer_a=&a;`
  - `//variabel pointer_a berisi alamat a`
  - `*pointer_a=250;`
  - `//alamat yang ditunjuk pointer_a berisi 250`
  - `//dengan kata lain,a akan berisi 250`

Array dan pointer

- Alamat memori suatu array:
  - Untuk mengetahui alamat memori suatu array tidak digunakan operator ‘&’. Alamat suatu array ditunjukkan oleh nama array itu sendiri. Misalnya:
  - unsigned char *hello[5]=“halo”;*
  - alamatnya adalah *hello* atau *&hello[0]* yaitu alamat data ke-0 dari suatu array

Array dan pointer

- **Pointer yang menunjuk ke suatu array**
  - Untuk membuat pointer menunjuk ke suatu array yang telah dibuat, variabel pointer diisi dengan alamat array tsb

  - `unsigned char hello[5]=“halo”;`
  - `unsigned char *p_hello;`
  - `p_hello = hello;// atau p_hello=&hello[0]`

Array dan pointer

- Pointer juga bisa digunakan untuk mengakses suatu array sbb:
  - **p_hello* adalah sama dengan *hello[0]*
  - **(p_hello+1)* sama dengan *hello[1]*
  - ….
  - **(p_hello+n)* sama dengan *hello[n]*

Array dan pointer

- Demikian juga dengan alamat masing elemen-elemen array tsb, bisa diakses lewat pointer sbb:
  - *p_hello* sama dengan *hello* atau *&hello[0]*
  - *p_hello+1* sama dengan *&hello[1]*
  - …. dst

Latihan