ARRAY / LARIK

Oleh : Agus Priyanto, M.Kom

INSTITUT TEKNOLOGI TELKOM
Smart, Trustworthy, And Teamwork

Tujuan

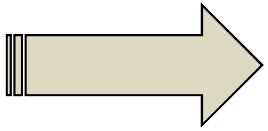
- Mahasiswa memahami makna dan kegunaan array (tabel)
- Mahasiswa dapat menggunakan notasi pendefinisian dan pengacuan array dengan benar hingga proses pencarian terhadap elemen array
- Mahasiswa dapat membuat program dengan menggunakan array

Ilustrasi

Mengolah 3 data

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

```
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;
}
```



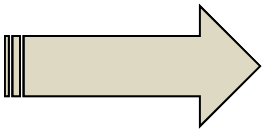
Ali
Budi
Ani
Ali – Budi
Ali – Ani
Budi - Ani

Mengolah 10 data

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

```
int main () {
// Kamus
string nama1, nama2, nama3, nama4, nama5 ;
string nama6, nama7, nama8, nama9, nama10 ;
//Algoritma
cin >> nama1;
cin >> nama2;
….// Dilanjutkan sendiri
cin >> nama10;

cout << nama1 " – " nama2 << endl;
cout << nama1 " – " nama3 << endl;
….// Dilanjutkan sendiri
cout << nama9 " – " nama10 << endl;
}
```



Nama 1 : Ali
Nama 2 : Budi
….
Nama 10 : Ina
Ali – Budi
Ali – Ani
…..
Ina- Jaja

Bagaimana kalau…

Anda diminta menampilkan semua kombinasi pasangan nama yang mungkin dari …

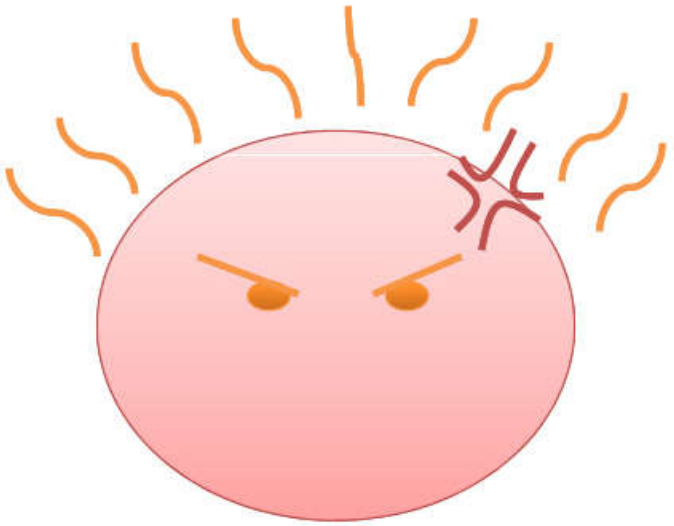
100 nama ???

1000 nama ???

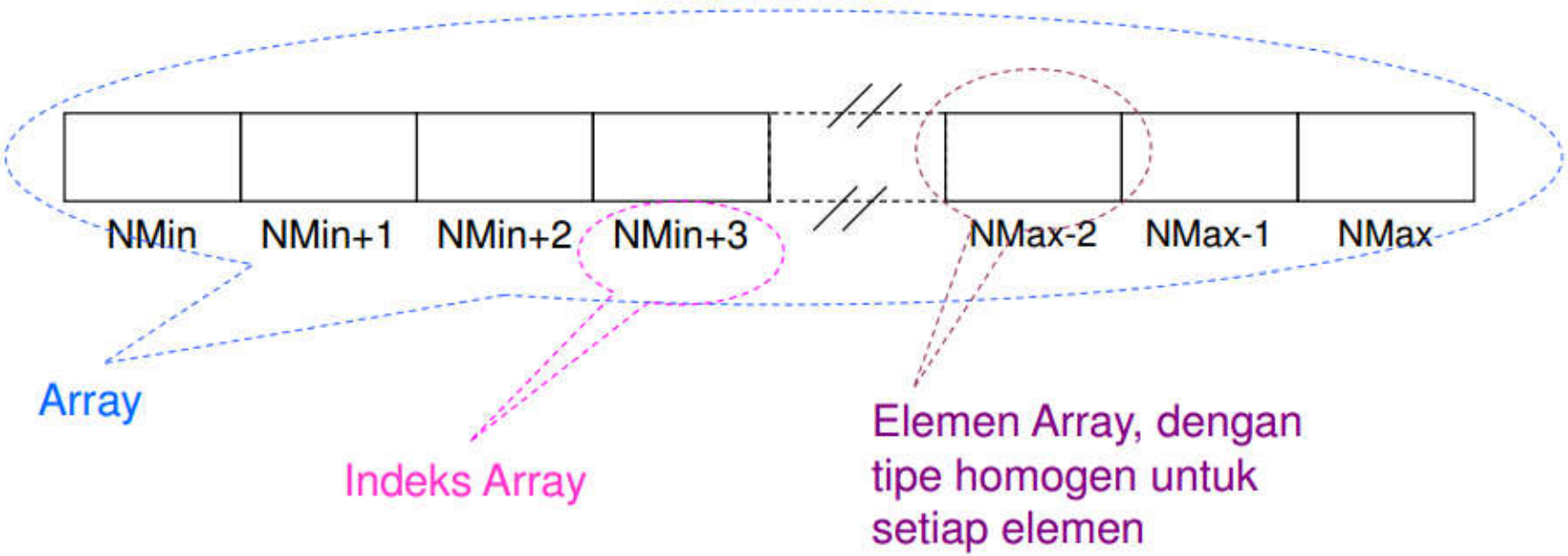
10000 nama ???

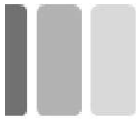
1000000 nama ???

….



Array / Tabel / Vektor / Larik





- **Array** adalah kumpulan data yang bertipe sama yang menggunakan nama yang sama.

- Dengan menggunakan array, sejumlah variabel dapat memakai nama yang sama.

- Antara satu variabel dengan variabel lain di dalam array dibedakan berdasarkan nomor elemen (subscript/indeks)

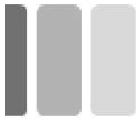
Array dalam C++ (1)

- Variabel dapat dideklarasikan ber-type array dari suatu type tertentu

- Setiap elemen array diakses dengan alamat berupa indeks yang bertype integer

Cara deklarasi :

```
<type> <namaArray>[<ukuran>]
```



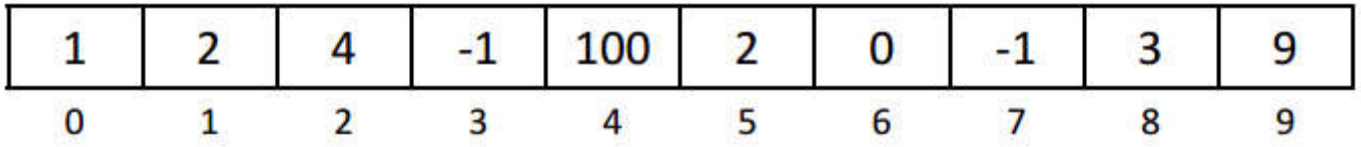
- **Contoh : int TabInt[10];**
  - ✓ Array bernama TabInt
  - ✓ Setiap elemen bertype integer,
  - ✓ Dengan ukuran 10 elemen,
  - ✓ Dengan alamat setiap elemen array (indeks) adalah dari indeks ke-0 s.d. 9

Array dalam C++ (2)

Cara akses elemen:

<namaArray>[<indeks>]

Contoh: int TabInt[10];




```
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

```
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
// Algoritma
.....
```

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

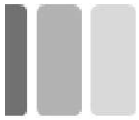
❑Mengisi Array

- 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 : **menampilkan semua isi array ke layar**

```
#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;
}
// Algoritma  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**

```
#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

```
#include <iostream>
using namespace std;
int main ()
{ // Kamus
int X, i; bool found;
int TabInt[10]
// Algoritma Pengisian data: asumsi array terisi
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;
```

➢Listing Program – 1

```
#include <iostream>
using namespace std;

int main()
{
   int data[5] = {4, 1, 0, -9, 8};
   int elemen;
   //tampilkandata
   for (elemen=0;elemen <= 4;elemen++)
   {
      cout << " Data ke - " << elemen << ": " << data[elemen];
   }
   return 0;
```

Data ke – 0 : 4 Data ke – 1 : 1 Data ke – 2 : 0
Data ke – 3 : -9 Data ke – 4 : 8

➢Listing Program – 2

```
#include <iostream>
using namespace std;
int main()
{
   int data[5]; //array dengan 5 elemen bertipe integer
   int elemen;
   //entri 5 data
   for (elemen=0;elemen <= 4;elemen++)
   {
      cout << "Data ke - " << elemen << ": ";
      cin >> data[elemen];
   } //tampilkandata setelah entri
      for (elemen=0;elemen <= 4;elemen++)
   {
      cout << " Data ke - " << elemen << ": " << data[elemen];
   }
   return 0;
}
```

➢Listing Program – 3

```
#include <iostream>
using namespace std;
int main()
{
   int data[10]= {4, 1, 0, -9, 8, 5,-1,2, 3, -7};
   int elemen, max;
   max = data[0];
   for(elemen=0;elemen<=9;elemen++)
   {
      if (data[elemen]>max) max = data[elemen];
      else max =max;
   }
   cout << "Nilai maksimum adalah :" << max;
   return 0;
}
```

Nilai maksimum adalah : 8

Array 2 Dimensi

- Array dapat dibuat berdimensi dua, dimana array ini mempunyai dua buah index / subscript.

- Sama seperti array dimensi satu, array dua dimensi juga merupakan kumpulan elemen-elemen yang bertipe data sama dengan satu nama variabel, tetapi terdiri dari dua index.

- **Contoh :**

```
int posisi [3][4]
```

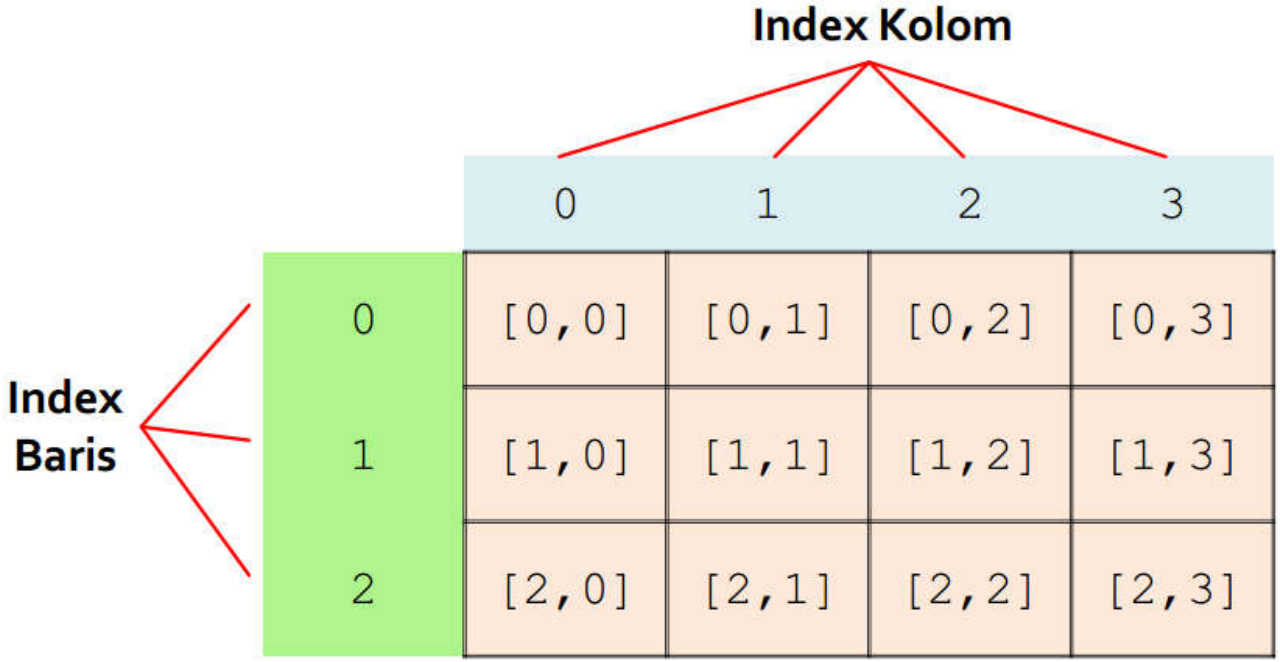
✓ Int → tipe data array

✓ posisi → variabel, nama array

✓ [3] → index (menyatakan jumlah baris)

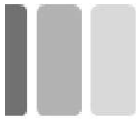
✓ [4] → index (menyatakan jumlah kolom)

- Dalam menganalogikan array 2 dimensi, sering digunakan istilah baris (x) dan kolom (y)
- Contoh : **int posisi [3][4];**
  - ✓ [3] → 3 buah baris (index 0 – 2)
  - ✓ [4] → 4 buah kolom (index 0 – 3)



- Representasi array dua dimensi di dalam memori secara berurutan per baris dengan elemen pertama adalah data berindex [0,0].

- Dari contoh maka representasinya di dalam memori sebagai berikut : [0,0]|[0,1]| [0,2]| [0,3]|[1,0]|[1,1]|[1,2]|[1,3]|[2,0]|[2,1]|[2,2]|[2,3]



- Array dua dimensi dapat digunakan untuk berbagai keperluan, antara lain :
  - ✓ **Matrik**
  - ✓ **Pemetaan** yang berhubungan dengan koordinat. Misalkan data jumlah pohon dalam koordinat tertentu, atau warna pixel di layar
  - ✓ Menampilkan **data multidimensi**

Listing Program ➔Inputan Matrik 2 Dimensi

```
{
  int matriks[3][3];
  int i,j;
  char opr;
   //Input nilai matriks dari baris dan kolom
  for(i=1;i<=2;i++){
      for(j=1;j<=2;j++){
          cout<<"Masukkan  nilai baris ke-"<<i<<" kolom ke-"<<j<<" : ";cin>>matrixa[i][j];
      }
  cout<<endl;
  }
   //Output matriks
  for(i=1;i<=2;i++){
       for(j=1;j<=2;j++){
           cout<<matriks[i][j]<<"\t";
       }
  cout<<endl;
  }
   return 0;
}
```

Listing Program ➔ data multidimensi
jumlah mahasiswa yang lulus di suatu universitas

```
int main()
{
   string jurusan[3];
   jurusan[0] = "S1 Teknik Informatika";
   jurusan[1] = "S1 Teknik Telekomunikasi";
   jurusan[2] = "D3 Teknik Telekomunikasi";
   int tahun[4] = {2011, 2012, 2013, 2014};
   int kelulusan[3][4] = {
      {35,45,80,120},
      {100,110,70,101},
      {10,15,20,17}
   };
   cout<<"-------------------------------------------------"<<endl;
   cout<<" Jurusan | ";
   for (int i=0; i<=3; i++) cout<<tahun[i]<<" "; cout<<endl;
   cout<<"-------------------------------------------------"<<endl;
   for (int x=0; x<=2; x++) {
      cout<<setw(22)<<jurusan[x];
      for (int y=0; y<=3; y++) {
         cout<<setw(7)<<kelulusan[x][y];
      }
      cout<<endl;
   }
   cout<<"-------------------------------------------------";
   return 0;
}
```

Listing Program ➔Inputan Matrik dengan Ordo 3 x 3

```
#include <iostream>

using namespace std;

int main() {

 // Melakukan inisialisasi nilai
 // ke dalam elemen-elemen array dua dimensi
 int A[3][3] = { {1,2,3}, {4,5,6}, {7,8,9} };

 // Menampilkan nilai yang tersimpan  dalam elemen array
 for (int j=0; j<3; j++) {
  for (int k=0; k<3; k++) {
   cout<<"A["<<j<<"]["<<k<<"] = "
       <<A[j][k]<<endl;
  }
  cout<<endl;
 }
 return 0;
}
```

```
A [0][0] = 1
A [0][1] = 2
A [0][2] = 3

A [1][0] = 4
A [1][1] = 5
A [1][2] = 6

A [2][0] = 7
A [2][1] = 8
A [2][2] = 9
```

```
1   2   3

4   5   6

7   8   9
```

Just In Minute (1)

```
#include <iostream>

using namespace std;

int main() {

      float angka[3] ;
      angka[0] = 1.5 ; angka[1] = 2.75 ; angka[2] = 3.25;
      char nama[5] = { 'u', 'd', 'i', 'n', '\0'} ;
      int koordinat[2] [3] = { { 1, 2, 3 } , { 4, 5, 6 } } ;
      cout << "angka[0]: "<<angka[0] << endl ;
      cout << "angka[1]: "<< angka[1] << endl ;
      cout << "angka[2]: "<< angka[2] << endl ;
      cout << "nama[0]: "<< nama[0] << endl ;
      cout << "Text string: "<<nama << endl ;
      cout << "koordinat[0][2]: "<<koordinat[0][2] << endl ;
      cout << "koordinat[1][2]: "<<koordinat[1][2] << endl ;
       return 0;;
}
```

Just In Minute (2)

```
#include <iostream>;
using namespace std;
int main()
{
   char negara [5][2][15] = {{ "Indonesia", "Jakarta"},
                             { "Philipina", "Manila"},
                             { "Austria", "Wina"},
                             { "India", "New Delhi"},
                             { "Iran", "Taheran"}};
   for (int baris = 0; baris <5; baris ++)
      if (negara[baris][0][0]=='I')
         cout << negara [baris][0] << " - "
              << negara [baris] [1] << "\n";
   return 0;
}
```

Indonesia -Jakarta
India - New Delhi
Iran - Taheran


Backspace
Thank You
Page Up
Home
Page Down
Insert
Shift