

IK141 Struktur Data
Pendahuluan Struktur Data
Review Array dan Fungsi



Di Susun Oleh :
Wendi Kardan, 2100016

PROGRAM STUDI PENDIDIKAN ILMU KOMPUTER
FAKULTAS PENDIDIKAN MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS PENDIDIKAN INDONESIA
12 FEBRUARI 2022

1. Implementasi dan Hasil

Implementasi dan Hasil

a. Array 1 Dimensi

Buatlah sebuah array yang dapat menyimpan data nama bulan dari bulan Januari sampai Desember.

Source Code :

```
#include<stdio.h>
// Wendi Kardian (2100016) - PILKOM A

void main(){
    // Mendeklarasikan array 2 dimensi dengan tipe data char dimana array dimensi pertama digunakan
    // untuk menampung data nama-nama bulan dari Januari hingga Desember
    // lalu array pada dimensi ke-2 digunakan untuk menampung kumpulan character yang membentuk
    string bulan
    char month[12][12]={"Januari", "Februari", "Maret", "April", "Mei", "Juni", "Juli", "Agustus",
    "September", "Oktober", "November", "Desember"};
    // Mendeklarasikan tipe data integer untuk inputan user yang menampung urutan bulan
    int data;
    // Menerima inputan user yang akan dimasukkan ke variabel data
    scanf("%d",&data);
    // Mencetak array data berdasarkan inputan user yang dikurangi satu hingga menjadi index array
    bulan tersebut

    // Mengecek apakah data yang diinput user sudah benar (0<x<13) sesuai dengan jumlah bulan
    if(data >= 1 && data<= 12){
        printf("%s",month[data-1]);
    }else{
        printf("Tidak ada bulan %d", data);
    }
}
```

Input/Output :

```
PS D:\UPI\SEMESTER 2\STRUKTUR DATA\Praktikum 1\task> cd "d:\UPI\SEMESTER 2\STRUKTUR DATA\Praktikum 1\task
\" ; if ($?) { gcc 1.c -o 1 } ; if ($?) { .\1 }
1
Januari
PS D:\UPI\SEMESTER 2\STRUKTUR DATA\Praktikum 1\task> cd "d:\UPI\SEMESTER 2\STRUKTUR DATA\Praktikum 1\task
\" ; if ($?) { gcc 1.c -o 1 } ; if ($?) { .\1 }
2
Februari
PS D:\UPI\SEMESTER 2\STRUKTUR DATA\Praktikum 1\task> cd "d:\UPI\SEMESTER 2\STRUKTUR DATA\Praktikum 1\task
\" ; if ($?) { gcc 1.c -o 1 } ; if ($?) { .\1 }
12
Desember
PS D:\UPI\SEMESTER 2\STRUKTUR DATA\Praktikum 1\task> cd "d:\UPI\SEMESTER 2\STRUKTUR DATA\Praktikum 1\task
\" ; if ($?) { gcc 1.c -o 1 } ; if ($?) { .\1 }
0
Tidak ada bulan 0
PS D:\UPI\SEMESTER 2\STRUKTUR DATA\Praktikum 1\task> █
```

Deskripsi :

1. Dalam fungsi utama(*main*) deklarasikan array 2 dimensi yang bernama *month* dimana dimensi pertama array tersebut akan diisi dengan banyaknya bulan dari Januari sampai Desember yang berjumlah 12, sedangkan dimensi kedua dari array tersebut akan diisi dengan string yang merupakan kumpulan dari character yang memiliki batas maksimal 12 character

2. Setelah itu ada variabel data yang akan diisi oleh user, data ini akan diisi oleh user dengan urutan bulan.
3. Jika user memasukan 1 maka yang akan ditampilkan adalah bulan ke-1 yaitu januari, Ketika user memasukan 2 maka yang akan ditampilkan adalah bulan ke-2 yaitu february. Apabila input yang ditampilkan user berada di luar range bulan ($0 < x < 13$), maka akan menampilkan bahwa tidak ada bulan x.

b. Array 1 dimensi

Array 1 dimensi akan digunakan untuk menyimpan data integer yang dimasukan sebanyak n. Output yang akan ditampilkan adalah nilai awal, nilai tengah, dan nilai terakhir.

Source Code :

```

1  #include<stdio.h>
2  // Wendi Kardian (2100016) - PILKOM A
3
4  int main(){
5      // deklarasi variabel untuk menampung panjang array
6      int arrayLength;
7      // menerima input user untuk panjang array yang akan dibuat
8      scanf("%d", &arrayLength);
9      // Deklarasi array 1 dimensi yang panjangnya tergantung input dari user
10     int array[arrayLength];
11     // Deklarasi variabel untuk iterasi perulangan
12     int i;
13     for(i = 0; i < arrayLength; i++){
14         // Proses input data kedalam array hingga penuh sesuai input user
15         scanf("%d", &array[i]);
16     }
17     // Menentukan titik tengah pada array
18     int mid;
19     if(arrayLength%2 == 0){
20         // Apabila blangan itu genap maka nilai tengahnya ada pembagian bilangan bulat panjang array
21         mid = (arrayLength/ 2)-1;
22     }else{
23         // Menentukan nilai tengah apabila panjang array ganjil
24         mid = arrayLength/2;
25     }
26     // Mencetak nilai pertama pada array
27     printf("%d\n", array[0]);
28     // Mencetak nilai tengah pada array
29     printf("%d\n", array[mid]);
30     // Mencetak nilai akhir pada array
31     printf("%d\n", array[arrayLength-1]);
32 }

```

Input/Output :

```

PS D:\UPI\SEMESTER 2\STRUKTUR DATA\Praktikum 1\task> cd "d:\UPI\SEMESTER 2\STRUKTUR DATA\Praktikum 1\task"
\n ; if ($?) { gcc 2.c -o 2 } ; if ($?) { .\2 }
3
1
3
5
1
3
5

```

```

PS D:\UPI\SEMESTER 2\STRUKTUR DATA\Praktikum 1\task> cd "d:\UPI\SEMESTER 2\STRUKTUR DATA\Praktikum 1\task
\" ; if ($?) { gcc 2.c -o 2 } ; if ($?) { .\2 }
4
1
3
5
7
1
3
7
PS D:\UPI\SEMESTER 2\STRUKTUR DATA\Praktikum 1\task> 

```

Deskripsi :

1. Deklarasi array 1 dimensi yang akan berisikan angka yang nanti akan dimasukan oleh user sebanyak n kali berdasarkan input pertama user
2. Setelah input sudah dimasukan, maka program akan memeriksa dimana nilai pertamanya (index pertama pada array), lalu nilai tengah yang diperoleh dari panjang array tersebut dibagi dengan 2, lalu nilai terakhir yang diperoleh dari panjang array dikurangi dengan 1.

c. Array 2 Dimensi

Buatlah array dua dimensi yang dapat membentuk persegi panjang dengan ukuran $n \times n$ yang disusun dengan karakter yang dimasukan oleh user.

Source Code :

```

1  #include<stdio.h>
2  #include<string.h>
3  // Wendi Kardian (2100016) - PILKOM A
4
5  void main(){
6      // Variabel untuk menentukan panjang ordo dan simbol apa yang akan dituliskan
7      int row;
8      char symbols;
9      // Input untuk simbol yang akan ditampilkan
10     scanf("%d", &row);
11     scanf(" %c", &symbols);
12     // mendeklarasikan array berdasarkan ordo yang diinputkan oleh user
13     char matriks[row][row];
14
15     // Proses input char dari input user kedalam array
16     int i,j;
17     for(i = 0; i<row; i++){
18         for(j=0;j<row; j++){
19             matriks[i][j] = symbols;
20         }
21     }
22
23     // Proses menampilkan data dari array 2 dimensi yang sudah di isi
24     // Proses untuk melakukan looping untuk mencetak berdasarkan baris
25     for(i = 0; i<row; i++){
26         // Proses untuk melakukan looping untuk mencetak berdasarkan kolom
27         for(j=0;j<row; j++){
28             printf("%c ", matriks[i][j]);
29         }
30         printf("\n");
31     }
32 }

```

Input/Output :

```
PS D:\UPI\SEMESTER 2\STRUKTUR DATA\Praktikum 1\task> cd "d:\UPI\SEMESTER 2\STRUKTUR DATA\Praktikum 1\task\n" ; if ($?) { gcc 3.c -o 3 } ; if ($?) { .\3 }
2
*
* *
* *
-
PS D:\UPI\SEMESTER 2\STRUKTUR DATA\Praktikum 1\task> cd "d:\UPI\SEMESTER 2\STRUKTUR DATA\Praktikum 1\task\n" ; if ($?) { gcc 3.c -o 3 } ; if ($?) { .\3 }
1
#
#
-
PS D:\UPI\SEMESTER 2\STRUKTUR DATA\Praktikum 1\task> cd "d:\UPI\SEMESTER 2\STRUKTUR DATA\Praktikum 1\task\n" ; if ($?) { gcc 3.c -o 3 } ; if ($?) { .\3 }
3
%
% % %
% % %
% % %
```

Dekripsi :

1. Deklarasi variabel row dengan tipe data integer untuk panjang baris dan kolom array 2 dimensi tersebut, kemudian deklarasi variable char symbols untuk menyimpan character yang nantinya akan diisikan kedalam array 2 dimensi tersebut.
2. Setelah program sudah mendapatkan data row, selanjutnya program akan mendeklarasikan array 2 dimensi dengan panjang data row yang sudah user inputkan kedalam sistem.
3. Setelah data symbols character sudah dimasukan kedalam matriks, selanjutnya program akan menampilkan data yang ada didalam array 2 dimensi tersebut menggunakan perulangan for

d. Array 2 Dimensi

Array 2 dimensi. Buatlah array dua dimensi yang dapat membentuk persegi panjang dengan ukuran $n \times n$, dimana apabila baris semakin bawah maka kolom yang bintangnya dicetak akan semakin sedikit.

Source Code :

```
1 #include<stdio.h>
2 // Wendi Kardian (2100016) - PILKOM A
3
4 void main(){
5     // Deklarasi variabel untuk menampung panjangnya array 2 dimensi
6     // yang nantinya akan menjadi input yang akan di isi oleh user
7     int row;
8     scanf("%d", &row);
9     // Deklarasi array 2 dimensi yang panjangnya ditentukan oleh input user
10    char matriks[row][row];
11    // Proses pengisian array dimana semakin bawah baris maka semakin sedikit kolom yang akan diisikannya
12    int i,j;
13    for(i = 0; i<row; i++){
14        // Perlu dikurangi variabel i dalam iterasi tersebut agar semakin bawah barisnya
15        // Maka kolom yang akan terbuat akan semakin sedikit
16        for(j=0;j<row-i; j++){
17            matriks[i][j] = '*';
18        }
19    }
```

```

20
21 // Proses cetak array sesuai yang sudah diisi sebelumnya
22 for(i = 0; i < row; i++){
23     for(j=0; j < row-i; j++){
24         printf("%c ", matriks[i][j]);
25     }
26     printf("\n");
27 }
28 }

```

Input/Output :

```

PS D:\UPI\SEMESTER 2\STRUKTUR DATA\Praktikum 1\task> cd "d:\UPI\SEMESTER 2\STRUKTUR DATA\Praktikum 1\task
\" ; if ($?) { gcc 4.c -o 4 } ; if ($?) { .\4 }
2
* *
*
PS D:\UPI\SEMESTER 2\STRUKTUR DATA\Praktikum 1\task> cd "d:\UPI\SEMESTER 2\STRUKTUR DATA\Praktikum 1\task
\" ; if ($?) { gcc 4.c -o 4 } ; if ($?) { .\4 }
1
*
PS D:\UPI\SEMESTER 2\STRUKTUR DATA\Praktikum 1\task> cd "d:\UPI\SEMESTER 2\STRUKTUR DATA\Praktikum 1\task
\" ; if ($?) { gcc 4.c -o 4 } ; if ($?) { .\4 }
3
* * *
* *
*

```

Deskripsi :

1. Program yang ini memiliki logika yang mirip dari yang sebelumnya, dimana kita akan menerima input dari user untuk panjang array 2 dimensi yang akan kita buat.
2. Setelah menerima input panjang row dari user, deklarasi array 2 dimensi sesuai dengan yang sudah user masukan baik itu untuk baris dan kolomnya.
3. Pengisian array 2 dimensi dengan tipe data char yang akan diisi dengan char '*', dimana semakin tinggi nilai barisnya, maka panjang kolomnya akan semakin pendek karena nilai iterasi pada kolom akan dikurangi dengan index baris tersebut. Sehingga pola yang akan terbentuk adalah pola segitiga terbalik.

e. Array 3 Dimensi

Buatlah array yang dapat menyimpan menyusun data nama-nama siswa menjadi sebuah matriks n x n yang paling sesuai dengan jumlah siswa yang akan diinputkan. Urutan mahasiswa berdasarkan baris ke 1 sampai baris n.

- Jika jumlah mahasiswa adalah 1 maka ukuran matriks 2D adalah 1 x 1
- Jika jumlah mahasiswa adalah 2-4 maka ukuran matriks 2D yang sesuai adalah 2 x 2
- Jika jumlah mahasiswa adalah 4-9 maka ukuran matriks 2D yang sesuai adalah 3 x 3
- Dst

Output dari program adalah susunan mahasiswa dalam matriks 2D.

Source Code :

```

1  #include<stdio.h>
2  #include<string.h>
3  #include<math.h>
4  // Wendi Kardan (2100016), PILKOM A
5
6  void main(){
7      // Variabel total digunakan untuk data yang nanti akan user input
8      int total;
9      // Variabel row digunakan untuk data panjang baris dan kolom suatu array 3 dimensi
10     int row;
11     // Proses pengecekan panjang row yang nanti akan dialokasikan untuk membuat array
12     scanf("%d", &total);
13     // Proses menentukan row berdasarkan panjang array dari user input
14     int k;
15     for(k= 0; k<=total; k++){
16         // Proses pengecekan range untuk panjang array tertentu
17         // Hingga dapat ditentukan ordo dari array tersebut
18         if(k*k < total && (k+1)*(k+1) >= total){
19             row = k+1;
20             break;
21         }
22     }
23
24     // Deklarasi array 3 dimensi dimana :
25     // dimensi 1 untuk baris, dimensi 2 untuk kolom, dimensi 3 untuk menampung string
26     char matriks[row][row][10];
27     int i,j;
28     int count=0;
29     // Proses looping untuk menerima input dari user kedalam array 3 dimensi
30     for(i= 0; i<row; i++){
31         for(j=0;j<row; j++){
32             // Proses pengecekan apabila jumlah iterasi untuk input sudah memenuhi
33             if(count == total){
34                 break;
35             }
36             scanf("%s",&matriks[i][j]);
37             // Penambahan nilai count 1, apabila sudah memasukan input
38             count += 1;
39         }
40     }
41
42     // Mereset nilai count
43     count =0;
44     // Proses iterasi untuk menampilkan array 3 dimensi yang sudah di input oleh user
45     for(i= 0; i<row; i++){
46         for(j=0;j<row; j++){
47             // Proses pengecekan apabila jumlah iterasi untuk input sudah memenuhi
48             if(count == total){
49                 break;
50             }
51             printf("%s  ", matriks[i][j]);
52             count +=1;
53         }
54         printf("\n");
55     }
56 }
57

```

Output/Input :

```

PS D:\UPI\SEMESTER 2\STRUKTUR DATA\Praktikum 1\task> cd "d:\UPI\SEMESTER 2\STRUKTUR DATA\Praktikum 1\task
\" ; if ($?) { gcc 5.c -o 5 } ; if ($?) { .\5 }
2
Budi
Ani
Budi   Ani

```

```

PS D:\UPI\SEMESTER 2\STRUKTUR DATA\Praktikum 1\task> cd "d:\UPI\SEMESTER 2\STRUKTUR DATA\Praktikum 1\task
\" ; if ($?) { gcc 5.c -o 5 } ; if ($?) { .\5 }
3
Budi
Ani
Firman
Budi Ani
Firman
PS D:\UPI\SEMESTER 2\STRUKTUR DATA\Praktikum 1\task> cd "d:\UPI\SEMESTER 2\STRUKTUR DATA\Praktikum 1\task
\" ; if ($?) { gcc 5.c -o 5 } ; if ($?) { .\5 }
5
Budi
Ani
Firman
Isabela
Michael
Budi Ani Firman
Isabela Michael

```

Deskripsi :

1. Deklarasi variabel total untuk menampung jumlah data yang nantinya akan user inputkan dan diolah menjadi array 3 dimensi, dimana dimensi 1 merepresentasikan baris, dimensi 2 merepresentasikan kolom, sedangkan dimensi 3 merepresentasikan array char yang membentuk menjadi string nama.
2. Setelah program menerima banyaknya data yang akan user masukan, selanjutnya program akan menentukan alokasi pembagian matriks terbaik untuk panjang data tersebut.
3. Program akan menerima input dari user sebanyak panjang array yang sudah user inputkan, input user yang merupakan nama akan dimasukan kedalam sebuah array 3 dimensi sesuai dengan alokasi array yang ditentukan
4. Setelah semua nama sudah dimasukan kedalam array. Program akan menampilkan semua isi dari array tersebut.

f. Fungsi dan Prosedur

Di bawah ini adalah contoh penggunaan fungsi dan prosedur. Tunjukkan hasil output program tersebut (screenshot) dan tunjukkan yang mana fungsi dan prosedur.

Source Code :

```

1  #include<stdio.h>
2
3  int hasilPangkat(int a){
4      return a*a;
5  }
6
7  void jenisBilangan(int a){
8      if(a%2==0){
9          printf("bilangan genap \n");
10         }else{
11             printf("bilangan ganjil \n");
12         }
13     }
14
15     int main(){
16         int a,pangkat;
17         scanf("%d",&a);
18         jenisBilangan(a);
19         pangkat = hasilPangkat(a);
20         printf("Hasil Pangkatnya adalah %d", pangkat);
21         return 0;
22     }
23

```


Input/Output :

```
\\" ; if ($?) { gcc 7.c -o 7 } ; if ($?) { .\\7 }
1
bilangan ganjil
Hasil Pangkatnya adalah 1
PS D:\\UPI\\SEMESTER 2\\STRUKTUR DATA\\Praktikum 1\\task> cd "d:\\UPI\\SEMESTER 2\\STRUKTUR DATA\\Praktikum 1\\task
\\" ; if ($?) { gcc 7.c -o 7 } ; if ($?) { .\\7 }
4
bilangan genap
Hasil Pangkatnya adalah 16
PS D:\\UPI\\SEMESTER 2\\STRUKTUR DATA\\Praktikum 1\\task> sn
```

Deskripsi :

1. Program diatas merupakan program untuk menentukan jenis bilangan dan pangkat dari bilangan tersebut yang dipisah kedalam fungsi dan prosedur.
2. Yang merupakan **FUNGSI** adalah **hasilPangkat** karena memiliki nilai Kembali dalam bentuk tipe data integer yang mengembalikan hasil pangkat 2 dari bilangan tersebut yang nantinya hasil Kembali dari fungsi tersebut disimpan kedalam variabel pangkat.
3. Yang merupakan **PROSEDUR** adalah **jenisBilangan** karena tidak memiliki nilai Kembali (void) dan didalam prosedur itu hanya menampilkan string kedalam layer untuk menentukan apakah bilangan tersebut ganjil atau genap.

g. Fungsi dan Prosedur

Buatlah program sederhana terdiri dari program utama dan prosedur untuk melakukan menentukan suatu kata polindrom atau bukan.

Source Code :

```
1 #include<stdio.h>
2 #include<string.h>
3 // Wendi Kardan (2100016), PILKOM - A
4
5 // Fungsi polindrom yang menerima 2 parameter
6 // Parameter 1 : array word yang bertipe data char yang benisikan kata untuk dicek polindrom atau
  bukan
7 // Parameter 2 : panjang array char yang akan dicek polindrom atau bukan
8 int polindrom(char word[], int length){
9     // Variabel start digunakan untuk menghitung jumlah huruf yang sama dari yang awal dengan yang
  akhir
10    // Variabel i digunakan untuk iterasi
11    int start = 0, i;
12    // Proses perulangan untuk mengecek apakah kata tersebut polindrom atau bukans
13    // Proses pengecekan digunakan dari index ke 0 hingga pertengahan array tersebut
14    for(i = 0; i<length/2;i++){
15        // Proses pengecekan disamakan dari paling kanan dibandingkan ke paling kiri, hingga index
  array berada di tengah-tengah
16        if(word[i] == word[length - i - 1]){
17            // Ketika array yang berbalikan (kanan-kiri) itu sama berarti variabel start akan
  bertambah, yang berarti char itu sama
18            start++;
19        }
20    }
```

```

21 // Variabel result digunakan untuk nilai kembali
22 int result;
23 // Apabila jumlah start sama dengan index tengah array tersebut maka kata tersebut polindrom
24 if(start == i){
25     result = 1;
26 }
27
28 // Apabila jumlah start tidak sama dengan index tengah array tersebut maka kata tersebut tidak
29 // polindrom
29 else{
30     result = 0;
31 }
32 // Jika result = 1 berarti kata tersebut polindrom, jika result = 0 kata tersebut bukan
33 // polindrom
34 // Nilai result akan dikembalikan ke prosedur utama
34 return result;
35 }
36
37 void main(){
38     // Deklarasi array 1 dimensi untuk menampung data kata yang akan diinputkan oleh user
39     char word[20];
40     int length;
41     int start = 0;
42
43     gets(word);
44     // variabel length merupakan variabel yang berisikan panjang array yang diinputkan user
45     length = strlen(word);
46
47     // Memanggil fungsi polindrom dimana nilai kembalinya disimpan di variabel result
48     int result = polindrom(word, length);
49
50     // Jika result = 1 berarti kata tersebut polindrom, jika result = 0 kata tersebut bukan
51     // polindrom
51     if(result == 1){
52         printf("polindrom");
53     }else{
54         printf("Bukan polindrom");
55     }
56 }

```

Input/Output :

```

PS D:\UPI\SEMESTER 2\STRUKTUR DATA\Praktikum 1\task> cd "d:\UPI\SEMESTER 2\STRUKTUR DATA\Praktikum 1\task
\" ; if ($?) { gcc 6.c -o 6 } ; if ($?) { .\6 }
katak
polindrom
PS D:\UPI\SEMESTER 2\STRUKTUR DATA\Praktikum 1\task> cd "d:\UPI\SEMESTER 2\STRUKTUR DATA\Praktikum 1\task
\" ; if ($?) { gcc 6.c -o 6 } ; if ($?) { .\6 }
hujan
Bukan polindrom
PS D:\UPI\SEMESTER 2\STRUKTUR DATA\Praktikum 1\task> cd "d:\UPI\SEMESTER 2\STRUKTUR DATA\Praktikum 1\task
\" ; if ($?) { gcc 6.c -o 6 } ; if ($?) { .\6 }
oppo
polindrom
PS D:\UPI\SEMESTER 2\STRUKTUR DATA\Praktikum 1\task>

```

Deskripsi :

1. Untuk mengetahui suatu kata itu polindrom atau bukan adalah dengan membalikan kata tersebut dan cek apakah kata yang sudah dibalik sama dengan kata yang sebelum dibalik, contoh :
Hujan != najuh
Katak == katak
Aha == aha
2. Didalam fungsi atau prosedur utama (*main*) deklarasi variabel untuk menampung array char yang berisikan string yang nantinya akan dicek apakah kata tersebut polidrom atau tidak.

3. Setelah user memberikan input kata yang kemudian disimpan kedalam array word yang merupakan array 1 dimensi. Selanjutnya array tersebut dikirimkan ke fungsi Bernama polindrom yang menerima 2 parameter yaitu array 1 dimensi dan panjang array tersebut.
4. Didalam fungsi polindrom, akan dicek menggunakan perulangan for apakah character yang paling depan sama dengan character paling belakang, kemudian perulangan akan terus dilakukan sampai pertengahan dimana dicek apakah antara pasangan kiri dan kanan itu sama atau tidak.
5. Jika semua character antara kanan dan kiri sama semua, maka berarti kata tersebut adalah polindrom sehingga fungsi itu akan mengembalikan nilai kembali berupa integer 1 (polindrom). Jika tidak, fungsi itu akan mengembalikan integer 0 (tidak polindrom).
6. Nilai kembalian dari fungsi polindrom kemudian disimpan kedalam variabel result, lalu kemudian program akan mengecek apabila nilai variabel result == 1 akan dicetak string 'polindrom', selain itu akan dicetak string 'Bukan polindrom'.

7. Kesimpulan

Kesimpulan

Dari hasil implementasi praktikum ini mengenai array dan fungsi prosedur bisa ditarik kesimpulan bahwa array dapat menyimpan data lebih dari 1 dan untuk mengakses data dalam array menggunakan index. Array dapat digolongkan menjadi array 1 dimensi, array 2 dimensi, dan array multidimensi sesuai dengan kebutuhan program yang ingin dibuat. Dalam praktikum ini mengimplementasikan bagaimana cara untuk mendeklarasikan sebuah array, mengisikan array menggunakan input manual maupun input dari user hingga dapat mencetak array tersebut kedalam console. Dalam praktikum mengimplementasikan fungsi dan prosedur, terdapat perbedaan antara fungsi dan prosedur. Fungsi memiliki nilai kembali (*return value*) sedangkan prosedur tidak memiliki nilai kembali (*void*).