



**PRAKTEK**

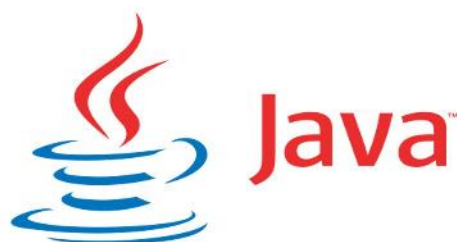
# **PEMROGRAMAN LANJUT**

Dibuat Oleh :

**YANCE SONATHA  
INDRI RAHMAYUNI  
DEFNI**



**Visual Studio®**



**PROGRAM STUDI MANAJEMEN INFORMATIKA  
JURUSAN TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI PADANG  
2016**

**LEMBARAN IDENTITAS DAN PENGESAHAN  
PENYUSUNAN MODUL PRAKTEK**

- |                             |   |   |
|-----------------------------|---|---|
| 1. Judul                    | : | Praktek Pemrograman Lanjut                              |
| 2. Ketua Pelaksana          | : |   |
| Nama/NIP                    | : | Yance Sonatha, S.Kom.,MT /<br>19801229 200604 2001      |
| Jurusan/Prodi               | : | Teknologi Informasi / Manajemen Informatika             |
| Matakuliah yang diampu      | : | Algoritma, Praktek Pemrograman                          |
| 3. Anggota Pelaksana 1      | : |   |
| Nama/NIP                    | : | Indri Rahmayuni,ST.,MT /<br>19830625 200801 2 004       |
| Jurusan/Prodi               | : | Teknologi Informasi / Teknik Komputer                   |
| Matakuliah yang diampu      | : | Algoritma, Praktek Pemrograman                          |
| Anggota Pelaksana 2         | : |   |
| Nama/NIP                    | : | Defni,S.Si.,M.Kom /<br>19811207 200812 2 001            |
| Jurusan/Prodi               | : | Teknologi Informasi / Manajemen Informatika             |
| Matakuliah yang diampu      | : | PTI, Praktek Pemrograman                                |
| 4. Lokasi Kegiatan          | : | Jurusan Teknologi Informasi<br>Politeknik Negeri Padang |
| 5. Biaya yang Dibutuhkan    | : |   |
| 6. Jangka Waktu Pelaksanaan | : | 6 bulan   |
| 7. Biaya                    | : | Rp 3.000.000,-  |

Mengetahui.  
Ketua Jurusan Teknologi Informasi

Padang, 30 November 2016  
Ketua Pelaksana

Ervan Asri, M.Kom  
NIP. 19780901 200812 1 001

Yance Sonatha, S.Kom.,MT  
NIP. 19801229 200604 2 001

Wakil Direktur I

Drs. Albar., M.Kom  
NIP. 19580917 198603 1 002

## KATA PENGANTAR

Puji syukur kita panjatkan kepada Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan pembuatan jobsheet mata kuliah Praktek Pemrograman Lanjut. Salawat dan salam juga kami untukkan kepada Rasulullah SAW yang telah membawa manusia dari zaman jahiliyah ke zaman yang beradab.

Bahan ajar ini dibuat untuk menuntun mahasiswa agar lebih mudah dalam melaksanakan perkuliahan mata kuliah Praktek Pemrograman Lanjut. Pada modul praktek ini terdapat teori untuk masing-masing bab yang berbeda pembahasannya. Setelah dipahami teori tersebut, ada kesimpulan yang diambil dan kemudian dikerjakan latihan pada masing-masing bab tersebut.

Penulis sadar bahwa bahan ajar ini masih ada kekurangannya, untuk itu diharapkan saran dan kritik pembaca agar bahan ajar ini menjadi lebih baik.

Penulis mengucapkan banyak terimakasih kepada :

1. Ir. Aidil Zamri, MT, selaku Direktur Politeknik Negeri Padang
2. Drs. Albar, M.Kom selaku Wakil Direktur I Politeknik Negeri Padang
3. Zulharbi, ST., MT selaku Ketua P3AI Politeknik Negeri Padang
4. Eryan Asri, M.Kom, selaku Ketua Jurusan Teknologi Informasi
5. Semua pihak yang tidak bisa disebutkan satu persatu

Akhir kata penulis berharap agar bahan ajar ini dapat bermanfaat bagi pembaca pada umumnya dan mahasiswa pada khususnya.

Padang, November 2016

Yance Sonatha, S.Kom., MT

## DESKRIPSI MODUL PRAKTEK

### a. Identitas Mata Kuliah

SSX	:	Praktek Pemrograman Lanjut
Kode Mata Kuliah	:	IS 4203
Jumlah SKS	:	1
Jumlah Jam	:	3
Semester / Tingkat	:	II / I
Status	:	Wajib
Pra Syarat	:	Algoritma dan Pemrograman, Praktek Pemrograman Dasar
Waktu Pertemuan	:	18 Kali Pertemuan/Semester (135 Menit/ Pertemuan)

### Deskripsi Ringkas

Mata Kuliah Praktikum Pemrograman Lanjut merupakan kelanjutan dari mata kuliah Praktikum Pemrograman Dasar yang diberikan pada semester 1. Pada mata kuliah ini, mahasiswa mempelajari pemrograman dalam bahasa C meliputi dasar bahasa C, array satu dan dua dimensi, tipe bentukan, fungsi dan prosedur, serta operasi pada file.

Mata Kuliah ini diberikan pada semester 2 dan merupakan mata kuliah wajib bagi mahasiswa Program Studi Manajemen Informatika Jurusan Teknologi Informasi di Politeknik Negeri Padang

Sebagai mata kuliah wajib beberapa materi pembelajaran diberikan dalam bentuk diskusi serta latihan pembuatan program agar mahasiswa lebih mengerti tentang materi yang diterangkan.

### Kriteria Penilaian

Proses penilaian pada mata kuliah Praktek Pemrograman Lanjut ini berasal dari empat komponen penilaian, yaitu :

#### 1. Kehadiran

Komponen ini memiliki poin sebesar 10% dari total pertemuan tatap muka sebanyak 18 kali. Kehadiran merupakan salah satu komponen penunjang

dalam melakukan proses penilaian dan keaktifan mahasiswa akan dijadikan nilai tambah secara objektif untuk penilaian akhir.

2. Tugas dan Laporan

Selama satu semester, mahasiswa diwajibkan mengerjakan semua latihan yang terdapat pada modul praktek yang dituliskan melalui Laporan. Latihan ini harus dikerjakan disetiap pertemuan praktek. Komponen keseluruhan tugas memiliki poin sebesar 30%.

3. UTS (Ujian Tengah semester)

UTS dilakukan pada pertemuan ke 9 dari keseluruhan total pertemuan melalui ujian tertulis, praktek atau mini project yang akan dipresentasikan. Materi yang diujikan adalah materi pertemuan 1 sampai dengan 8, dengan bobot yang diberikan sebesar 30%.

4. UAS (Ujian Akhir Semester)

UAS dilakukan pada saat jadwal ujian akhir yang ditetapkan oleh jadwal akademik di Jurusan Teknologi Informasi. UAS bisa dilaksanakan dalam bentuk ujian tertulis, praktek atau mini project yang akan dipresentasikan. Materi yang diujikan adalah materi setelah UTS yaitu materi pada pertemuan 10 sampai dengan 18, dengan bobot yang diberikan sebesar 30%.

Penilaian dilakukan berdasarkan semua komponen nilai yang ada. Nilai akhir yang diperoleh mahasiswa merupakan rata-rata dari perolehan tiap komponen dengan melibatkan bobot masing-masing. Nilai akhir merupakan gambaran kemampuan dan kualitas mahasiswa terhadap ilmu yang sudah diperoleh selama satu semester. Nilai akhir akan dikonversi ke dalam bentuk angka dan huruf sesuai dengan peraturan akademik yang ada di Jurusan Teknologi Informasi.

## DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBARAN IDENTITAS DAN PENGESAHAN.....	ii
KATA PENGANTAR.....	iii
DESKRIPSI MODUL PRAKTEK.....	iv
DAFTAR ISI.....	vi
DAFTAR TABEL.....	vii
DAFTAR GAMBAR.....	viii
PETUNJUK PENGGUNAAN MODUL PRAKTEK.....	ix
JOB 1     ARRAY SATU DIMENSI STATIK .....	1
JOB 2     ARRAY SATU DIMENSI DINAMIK.....	14
JOB 3     SORTING (METODE PENGURUTAN).....	23
JOB 4     SEARCHING(METODE PENCARIAN).....	36
JOB 5     ARRAY DUA DIMENSI.....	50
JOB 6     TIPE BENTUKAN.....	63
JOB 7     TIPE BENTUKAN DAN ARRAY.....	72
JOB 8     FUNGSI DENGAN NILAI BALIK.....	82
JOB 9     FUNGSI BERTIPE VOID.....	93
JOB 10    GABUNGAN FUNGSI DENGAN ARRAY.....	105
JOB 11    GABUNGAN FUNGSI DENGAN TIPE BENTUKAN.....	113
JOB 12    OPERASI FILE.....	122
LAMPIRAN	

## DAFTAR TABEL

Tabel 3.1	Proses Pengurutan dengan Metode Seleksi.....	27
Tabel 3.2	Proses Pengurutan dengan Metode Gelembung.....	29

## **DAFTAR GAMBAR**

Gambar 5.1	Pengaksesan Elemen Array Dua Dimensi.....	50
Gambar 8.1	Pemanggilan Fungsi dalam Bahasa C.....	83



## **PETUNJUK PENGGUNAAN MODUL PRAKTEK**

Bahan ajar ini dibuat sebagai bahan acuan materi dan pelaksanaan perkuliahan Praktek Pemrograman Lanjut untuk Program Studi Manajemen Informatika Jurusan Teknologi Informasi. Pada buku ini terdapat garis-garis besar materi praktek perkuliahan yang terdiri dari 18 bab, dimana 1 bab digunakan untuk 1 kali pertemuan. Masing-masing bab dilengkapi dengan landasan teori, langkah kerja dan latihan.

Sebagai upaya dalam meningkatkan efektivitas proses pembelajaran pada mata kuliah Praktek Pemrograman Lanjut, maka dibutuhkan modul praktek berupa buku sebagai pelengkap dalam aktivitas pembelajaran. Penulisan modul praktek mata kuliah Praktek Pemrograman Lanjut ini mengacu pada sasaran belajar dan pembelajaran untuk mencapai kompetensi sebagaimana yang tertuang dalam Rencana Pembelajaran Semester (RPS). Adapun materi yang jelaskan pada modul praktek mata kuliah Praktek Pemrograman Lanjut ini adalah Array Satu Dimensi Statik/ Dinamik, Proses Pengurutan (Bubble Sort dan Selection Sort), Proses Pencarian (Binary Search dan Sequential Search, Array Dua Dimensi, Tipe Bentuk, Fungsi dengan Tipe Data dan Void, Gabungan Fungsi dengan Array dan Tipe Bentuk serta Operasi File.


### **a. Petunjuk bagi Mahasiswa**

1. Baca dan pelajari modul praktek sebelum hari pelaksanaan praktek, pelajari dan pahami langkah-langkah percobaannya.
2. Jika ada langkah-langkah yang belum jelas, tanyakan kepada pengajar sebelum praktek dimulai.
3. Install terlebih dahulu program/software yang akan digunakan pada kegiatan praktek.
4. Bacalah informasi-informasi yang ada berkaitan dengan program/software guna kelancaran kegiatan praktek.

5. Ikutilah petunjuk-petunjuk yang ada berkaitan dengan penggunaan program/software tersebut.
6. Setiap mempelajari satu sub kompetensi, Anda harus mulai dari menguasai pengertian-pengertian dalam uraian materi dan kemudian kerjakan soal soal yang ada di bagian lembar latihan.

**b. Petunjuk bagi Dosen**

1. Jumlah keseluruhan materi percobaan ini ada 18 modul praktek yang dapat dipergunakan untuk satu semester (terdiri dari 18 x pertemuan kegiatan praktek yang masing-masing memerlukan waktu 3 X 45 menit).
2. Laporan praktek Mahasiswa diserahkan pada awal kegiatan praktek berikutnya, diserahkan kepada pengajar.
3. Materi untuk UTS dan UAS juga diserahkan kepada pengajar. Disesuaikan dengan jadwal akademik dan progres pelaksanaan praktek dari kelas yang bersangkutan.

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>1</b>
	<b>ARRAY SATU DIMENSI STATIK</b>	

### 1. Judul : Array Satu Dimensi Statik

### 2. Jumlah Jam/Pertemuan : 3 jam x 2 pertemuan = 6 jam praktek

### 3. Tujuan Praktek



- Mahasiswa mampu menjelaskan konsep array satu dimensi statik dan penggunaannya.
- Mahasiswa mampu mendefinisikan, mendeklarasikan, mengisi dan menuliskan nilai array satu dimensi statik menggunakan bahasa C.
- Mahasiswa mampu membuat program yang memproses nilai array satu dimensi statik.

### 4. Teori Dasar.

Sebenarnya, apa itu array/larik? Supaya mudah dipahami, perhatikanlah contoh kasus berikut: suatu universitas ingin mendata nilai mahasiswa di suatu kelas dengan banyak mahasiswa 10 orang. Setelah diinputkan, nilai-nilai tersebut ditampilkan ke layar. Untuk membuat program dengan ketentuan seperti diatas, ada beberapa cara untuk menyelesaikannya :

Program 1: tanpa menggunakan array

```
#include <stdio.h>
void main()
{
    int n1,n2,n3,n4,n5,n6,n7,n8,n9,n10;
    // Pembacaan semua nilai dari keyboard
    printf("Pemasukan data nilai mahasiswa : \n");
    printf("Nilai mahasiswa Ke-1 : ");scanf("%d",&n1);
    printf("Nilai mahasiswa Ke-2 : ");scanf("%d",&n2);
    printf("Nilai mahasiswa Ke-3 : ");scanf("%d",&n3);
    printf("Nilai mahasiswa Ke-4 : ");scanf("%d",&n4);
    printf("Nilai mahasiswa Ke-5 : ");scanf("%d",&n5);
    printf("Nilai mahasiswa Ke-6 : ");scanf("%d",&n6);
    printf("Nilai mahasiswa Ke-7 : ");scanf("%d",&n7);
    printf("Nilai mahasiswa Ke-8 : ");scanf("%d",&n8);
    printf("Nilai mahasiswa Ke-9 : ");scanf("%d",&n9);
```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>1</b>
	<b>ARRAY SATU DIMENSI STATIK</b>	



```
printf("Nilai mahasiswa Ke-10: ");scanf("%d",&n10);

// Menampilkan data nilai yang telah dimasukan
printf("Nilai mahasiswa Ke-1 : %d\n",n1);
printf("Nilai mahasiswa Ke-2 : %d\n",n2);
printf("Nilai mahasiswa Ke-3 : %d\n",n3);
printf("Nilai mahasiswa Ke-4 : %d\n",n4);
printf("Nilai mahasiswa Ke-5 : %d\n",n5);
printf("Nilai mahasiswa Ke-6 : %d\n",n6);
printf("Nilai mahasiswa Ke-7 : %d\n",n7);
printf("Nilai mahasiswa Ke-8 : %d\n",n8);
printf("Nilai mahasiswa Ke-9 : %d\n",n9);
printf("Nilai mahasiswa Ke-10: %d\n",n10);
}
```

Dengan menggunakan cara diatas, sebenarnya programnya telah mencukupi, tetapi jika nilai yang akan diolah menjadi lebih banyak, maka pendeklarasian variabel n harus dilakukan sebanyak yang diperlukan. Jadi kalau data yang akan diolah sebanyak 100 buah, maka pendeklarasian dan pembacaan datanya pun dilakukan sebanyak 100 kali. Dan perhitungannya juga. Rumus perhitungan total pun menjadi berubah. Pemrograman di atas sebenarnya sederhana tetapi bisa sangat merepotkan.

Solusi dari permasalahan di atas adalah dengan menggunakan array. Array adalah suatu variabel yang dapat menampung lebih dari satu data dengan tipe data yang sama dan dibedakan berdasarkan nomor indeksnya. Array memiliki dimensi (1, 2, n) yang menggambarkan keterkaitan data yang akan disimpan dalam array. Dalam pembuatannya, array dapat dibagi kepada dua jenis yaitu array statik dan array dinamik. Pada array statik, ukuran array telah ditetapkan saat pendeklarasian array dan tidak dapat diubah lagi (tetap). Sedangkan pada array dinamik, ukuran array dapat diubah-ubah sesuai dengan kebutuhan pengguna (dibahas pada topik berikutnya).

Karena konsep array adalah penyimpanan banyak data yang bertipe sama, maka penyimpanan setiap data dibedakan dengan nomer indeks. Ibaratkan sebuah array adalah sebuah hotel dengan kamar-kamar hotel merupakan lokasi penyimpanan data yang dibedakan dengan nomer kamar. Nomer kamar hotel inilah yang kita sebut indeks. Dalam bahasa C, indeks sebuah array dimulai dari 0 (nol).

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>1</b>
	<b>ARRAY SATU DIMENSI STATIK</b>	

Untuk praktek kali ini, kita akan fokus kepada array satu dimensi statik, yaitu sebuah array statik yang hanya terdiri dari satu baris tapi banyak kolom. Bentuk deklarasi array satu dimensi statik adalah :

```
tipe_data nama_array[ukuran_array];
```

Contoh :

```
int N[10];
```

Deklarasi diatas berarti pendeklarasian sebuah array bernama N yang mempunyai elemen sebanyak 10 buah dengan indeks dimulai dengan nomor 0 sampai 9. Dalam memori komputer, deklarasi tersebut dapat digambarkan seperti berikut:

N[0]	N[1]	N[2]	N[3]	N[4]	N[5]	N[6]	N[7]	N[8]	N[9]
------	------	------	------	------	------	------	------	------	------

Untuk memasukan suatu elemen data dalam array, perintah yang dilakukan ditulis seperti pembacaan data variabel biasa (**scanf()**), hanya perbedaannya pada nama variabel yaitu harus ditulis dengan jelas nama array-nya beserta nomer indeks. Contohnya:

```
scanf("%d",&N[3]);
```

Perintah diatas berarti pembacaan data dari keyboard untuk nilai bertipe integer ("%d") dan dimasukan ke sebuah array N pada indeks ke-3 (urutan ke-4).

Contoh-contoh lain pengisian nilai ke suatu elemen array:


```
i=5; // variabel i diisi dengan nilai 5
N[i] = 7; // data ke-i dari array N diisi dengan nilai 7
```

Karena nomor elemen dari array bisa diisi dengan sebuah variabel, berarti kita bisa melakukan pengulangan (*looping*) untuk melakukan penginputan data dari elemen pertama sampai elemen terakhir.

Untuk menuliskan data yang ada dalam array ke layar, perintah **printf()** tetap digunakan, dengan menambahkan nomer indeks array pada nama array yang akan dituliskan.

```
printf("Isi = %d",N[3]);
```

Perintah diatas berarti menuliskan data bertipe integer (%d) yang ada dalam array N indeks ke-3 (urutan ke-4) ke layar. Sama seperti penginputan nilai array, pengulangan

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>1</b>
	<b>ARRAY SATU DIMENSI STATIK</b>	

(*looping*) juga dapat digunakan dalam penulisan isi/data array.

## 5. Bahan/Peralatan yang digunakan

- Komputer
- Aplikasi CodeBlocks

## 6. Langkah Kerja

- Buat folder dengan nama folder: **NamaAnda + Kelas** di direktori selain C. Folder ini Anda gunakan untuk menyimpan semua program yang Anda buat selama praktikum semester ini.
- Jalankan program CodeBlocks.
- Buatlah beberapa program berikut di komputer anda. Compile dan jalankan (Run). Perhatikan hasil keluarannya. Jalankan program dan analisa hasilnya.
  - Program yang meminta user menginputkan 10 buah bilangan bulat, menghitung jumlah dan rata-ratanya, serta menampilkan isi array tersebut.

```

/* Nama File : array1.c
   Pembuat   : tuliskan nama dan no.bp anda
   Tgl pembuatan : tuliskan tanggal hari ini
   Deskripsi  : menghitung total dan rata-rata nilai
               10 orang mahasiswa yang disimpan dalam array      */



#include "stdio.h"

void main()
{
    int Nilai[10];
    int indeks;
    float total,ratarata;

    // Pembacaan data dari keyboard
    printf("Pembacaan data nilai \n");
    for (indeks=0;indeks<10;indeks++)
    {
        printf("Nilai mahasiswa ke-%d = ",indeks+1);
        scanf("%d",&Nilai[indeks]);
    }

    // Perhitungan total dan rata-rata
    total=0;
    for (indeks=0;indeks<10;indeks++)

```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>1</b>
	<b>ARRAY SATU DIMENSI STATIK</b>	

```

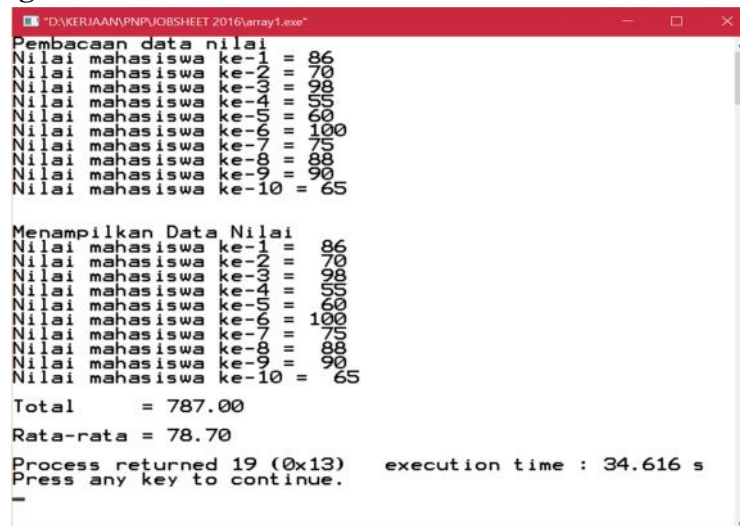
        total=total+Nilai[indeks];
    //endfor

    ratarata=total/10.0;

    // Menampilkan data yang telah dimasukan dan rata-rata.
    printf("\n\nMenampilkan Data Nilai \n");
    for (indeks=0;indeks<10;indeks++)
        printf("Nilai mahasiswa ke-%d = %3d\n",
            indeks+1,Nilai[indeks]);
    printf("\nTotal      = %0.2f\n",total);
    printf("\nRata-rata = %0.2f\n",ratarata);
}

```

### Output Program



```

Pembacaan data nilai
Nilai mahasiswa ke-1 = 86
Nilai mahasiswa ke-2 = 70
Nilai mahasiswa ke-3 = 98
Nilai mahasiswa ke-4 = 55
Nilai mahasiswa ke-5 = 60
Nilai mahasiswa ke-6 = 100
Nilai mahasiswa ke-7 = 75
Nilai mahasiswa ke-8 = 88
Nilai mahasiswa ke-9 = 90
Nilai mahasiswa ke-10 = 65

Menampilkan Data Nilai
Nilai mahasiswa ke-1 = 86
Nilai mahasiswa ke-2 = 70
Nilai mahasiswa ke-3 = 98
Nilai mahasiswa ke-4 = 55
Nilai mahasiswa ke-5 = 60
Nilai mahasiswa ke-6 = 100
Nilai mahasiswa ke-7 = 75
Nilai mahasiswa ke-8 = 88
Nilai mahasiswa ke-9 = 90
Nilai mahasiswa ke-10 = 65

Total      = 787.00
Rata-rata = 78.70
Process returned 19 (0x13)   execution time : 34.616 s
Press any key to continue.

```

- b) Program yang mengisi nilai sebuah array satu dimensi statik (A) dan menyalin isi array tersebut ke array lain (B) dengan ukuran dan tipe data yang sama.



```

/* Nama File : salin_array.c
   Pembuat   : tuliskan nama dan no.bp anda
   Tgl pembuatan : tuliskan tanggal hari ini
   Deskripsi : menyalin isi array A ke array B          */

#include "stdio.h"

void main()
{
    int A[100], B[100];
    int n,i;

```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>1</b>
	<b>ARRAY SATU DIMENSI STATIK</b>	

```

printf("Masukkan banyak data : "); scanf("%d",&n);

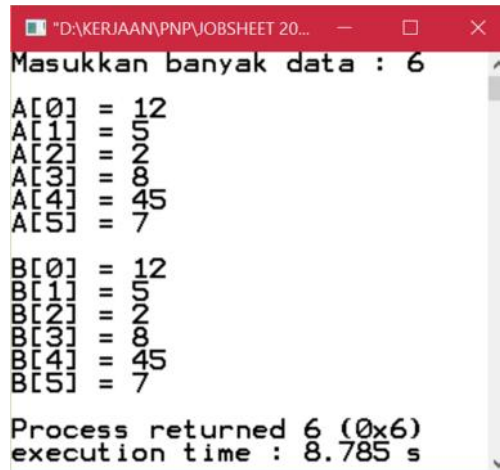
//Input isi array A sebanyak n buah
printf("\n");
for(i=0;i<n;i++)
{
    printf("A[%d] = ",i); scanf("%d",&A[i]);
}

//Menyalin isi array A ke array B
for(i=0;i<n;i++)
{
    B[i]=A[i];
}

//Menampilkan isi array B ke layar output
printf("\n");
for(i=0;i<n;i++)
{
    printf("B[%d] = %d\n",i,B[i]);
}
}

```

### Output Program



```

D:\KERJAAN\PNP\JOBSHEET 20...
Masukkan banyak data : 6



A[0] = 12
A[1] = 5
A[2] = 2
A[3] = 8
A[4] = 45
A[5] = 7

B[0] = 12
B[1] = 5
B[2] = 2
B[3] = 8
B[4] = 45
B[5] = 7

Process returned 6 (0x6)
execution time : 8.785 s

```



	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>1</b>
	<b>ARRAY SATU DIMENSI STATIK</b>	

c) Program yang melakukan konversi nilai array, hasil konversi disimpan ke array baru.

```



/* Nama File : nilai_mhs.c
   Pembuat   : tuliskan nama dan no.bp anda
   Tgl pembuatan : tuliskan tanggal hari ini
   Deskripsi  : konversi nilai, simpan di array baru*/

#include <stdio.h>

void main()
{
    int nilai[15];
    char huruf[15];
    int i;

    for (i=0;i<15;i++)
    {
        printf("Masukkan nilai mahasiswa %d:  ",i+1);
        scanf("%d",&nilai[i]);
    }
    for (i=0;i<15;i++)
    {
        if (nilai[i]>=81 && nilai[i]<=100)
            huruf[i]='A';
        else if (nilai[i]>=71 && nilai[i]<=80)
            huruf[i]='B';
        else if (nilai[i]>=56 && nilai[i]<=70)
            huruf[i]='C';
        else if (nilai[i]>=41 && nilai[i]<=55)
            huruf[i]='D';
        else
            huruf[i]='E';
        //endif
    }
    printf("\n\n");
    for(i=0;i<15;i++)
        printf("Nilai mahasiswa %d = %d = %c \n", i+1,
nilai[i],huruf[i]);
}

```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>1</b>
	<b>ARRAY SATU DIMENSI STATIK</b>	

### Output Program



```

Masukkan nilai mahasiswa 1: 86
Masukkan nilai mahasiswa 2: 75
Masukkan nilai mahasiswa 3: 90
Masukkan nilai mahasiswa 4: 100
Masukkan nilai mahasiswa 5: 40
Masukkan nilai mahasiswa 6: 51
Masukkan nilai mahasiswa 7: 45
Masukkan nilai mahasiswa 8: 67
Masukkan nilai mahasiswa 9: 78
Masukkan nilai mahasiswa 10: 89
Masukkan nilai mahasiswa 11: 90
Masukkan nilai mahasiswa 12: 60
Masukkan nilai mahasiswa 13: 56
Masukkan nilai mahasiswa 14: 50
Masukkan nilai mahasiswa 15: 99

Nilai mahasiswa 1 = 86 = A
Nilai mahasiswa 2 = 75 = B
Nilai mahasiswa 3 = 90 = A
Nilai mahasiswa 4 = 100 = A
Nilai mahasiswa 5 = 40 = E
Nilai mahasiswa 6 = 51 = D
Nilai mahasiswa 7 = 45 = D
Nilai mahasiswa 8 = 67 = C
Nilai mahasiswa 9 = 78 = B
Nilai mahasiswa 10 = 89 = A
Nilai mahasiswa 11 = 90 = A
Nilai mahasiswa 12 = 60 = C
Nilai mahasiswa 13 = 56 = C
Nilai mahasiswa 14 = 50 = D
Nilai mahasiswa 15 = 99 = A

```

- d) Program yang mengisi nilai dua buah array satu dimensi (A dan B) kemudian melakukan penjumlahan isi array pada indeks yang sama dan menyimpan hasilnya pada array ke tiga (C).

```

/* Nama File : jumlah_array.c
   Pembuat   : tuliskan nama dan no.bp anda
   Tgl pembuatan : tuliskan tanggal hari ini
   Deskripsi  : menjumlahkan isi array A + B = C */

#include "stdio.h"



void main()
{
    int A[100], B[100], C[100];
    int n,i;

    printf("Masukkan banyak data : "); scanf("%d",&n);

    //Input isi array A sebanyak n buah
    printf("\n");
    for(i=0;i<n;i++)
    {
        printf("A[%d] = ",i); scanf("%d",&A[i]);
    }

    //Input isi array B sebanyak n buah
    printf("\n");
    for(i=0;i<n;i++)
    {
        printf("B[%d] = ",i); scanf("%d",&B[i]);
    }
}

```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>1</b>
	<b>ARRAY SATU DIMENSI STATIK</b>	

```

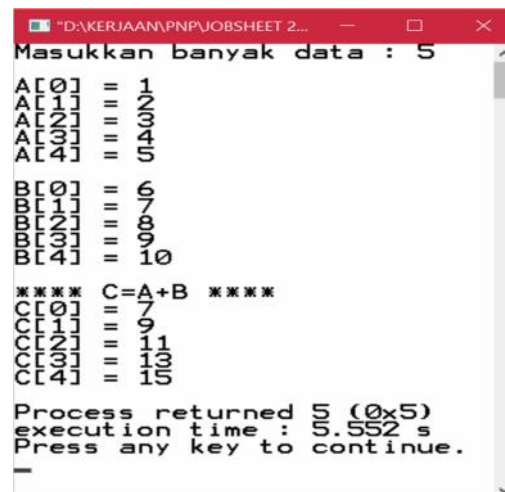
}

//Penjumlahan isi array A + B = C
for(i=0;i<n;i++)
{
    C[i]=A[i]+B[i];
}

//Menampilkan isi array C ke layar output
printf("\n**** C=A+B ****\n");
for(i=0;i<n;i++)
{
    printf("C[%d] = %d\n",i,C[i]);
}
}

```

Output Program



```

Masukkan banyak data : 5

A[0] = 1
A[1] = 2
A[2] = 3
A[3] = 4
A[4] = 5

B[0] = 6
B[1] = 7
B[2] = 8
B[3] = 9
B[4] = 10



**** C=A+B ****
C[0] = 7
C[1] = 9
C[2] = 11
C[3] = 13
C[4] = 15

Process returned 5 (0x5)
execution time : 5.552 s
Press any key to continue.

```

## 7. Latihan

- Terdapat sebuah array satu dimensi statik berukuran 10 dengan tipe data bilangan bulat. Pengisian array dilakukan melalui proses input. Buatlah sebuah program yang **menentukan nilai terbesar dan terkecil** dari nilai array tersebut.
- Terdapat dua buah array satu dimensi statik (A dan B) dengan ukuran dan tipe data yang sama (misal : integer). Buatlah sebuah program yang mengisi array A kemudian **menyalin isi array A ke array B dalam posisi terbalik** dimana nilai pertama pada array A merupakan nilai terakhir pada array B, begitu seterusnya.

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>1</b>
	<b>ARRAY SATU DIMENSI STATIK</b>	

Tampilkan isi array B ke layar output.



- c. Terdapat dua buah array satu dimensi statik (A dan B) bertipe data integer. Buatlah program yang mengisi array A kemudian **menyalin isi array A yang genap saja** ke array B. Tampilkan isi array B ke layar output.
- d. Terdapat tiga buah array satu dimensi statik (A,B,C) dengan ukuran dan tipe data yang sama. Buatlah sebuah program yang mengisi array A dan B, kemudian **bandingkan isi array A dan B pada indeks yang sama**. Salin nilai terbesar pada array C. Tampilkan isi array C ke layar output.

#### **8. Keselamatan dan Kesehatan Kerja (K3)**

- a. Periksaalah komputer yang anda gunakan, apakah terjadi kerusakan atau tidak. Jika terjadi kerusakan, segera melapor pada Dosen ataupun Teknisi Labor.
- b. Jika akan melakukan peng-copy-an data, pastikan komputer ataupun alat penyimpanan data Anda terbebas dari virus.
- c. Selalu matikan komputer setiap Anda selesai praktikum.
- d. Jangan meninggalkan sampah apapun di ruangan labor komputer.

#### **9. Hasil Pengujian dan Pembahasan**

Selesaikanlah soal latihan dan kemudian cocokan hasil output program Anda dengan hasil program berikut ini. Jalankan juga program Anda untuk nilai-nilai lainnya.

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>1</b>
	<b>ARRAY SATU DIMENSI STATIK</b>	

- a. Menentukan nilai terbesar dan terkecil isi array

```

"D:\KERJAAN\PNP\JOBSHEET 2016\maxmin_array.exe"
Input 10 buah bilangan bulat
Nilai ke-1 = 34
Nilai ke-2 = 7
Nilai ke-3 = 0
Nilai ke-4 = -10
Nilai ke-5 = 2
Nilai ke-6 = 45
Nilai ke-7 = -15
Nilai ke-8 = 3
Nilai ke-9 = 8
Nilai ke-10 = 4

Nilai terbesar = 45
Nilai terkecil = -15

```

- b. Menyalin isi array A ke array B dalam posisi terbalik



```

"D:\KERJAAN\PNP\JOBSHEET ..."
Masukkan banyak data : 6

A[0] = 7
A[1] = 3
A[2] = 9
A[3] = 12
A[4] = 4
A[5] = 8

B[0] = 8
B[1] = 4
B[2] = 12
B[3] = 9
B[4] = 3
B[5] = 7

```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>1</b>
	<b>ARRAY SATU DIMENSI STATIK</b>	

c. Menyalin isi array A yang genap saja ke array B

```

"D:\KERJAAN\PNP\JOBSHEET 2016\sali...
Masukkan banyak data : 10
A[0] = 1
A[1] = 2
A[2] = 3
A[3] = 4
A[4] = 5
A[5] = 6
A[6] = 7
A[7] = 8
A[8] = 9
A[9] = 10

B[0] = 2
B[1] = 4
B[2] = 6
B[3] = 8
B[4] = 10

```

d. Membandingkan isi array A dan array B serta menyalin yang terbesar ke array C

```

"D:\KERJAAN\PNP\JOBSHEET 2016\banding...
Masukkan banyak data : 5
A[0] = 12
A[1] = 6
A[2] = 3
A[3] = 9
A[4] = 4

B[0] = 1
B[1] = 7
B[2] = 10
B[3] = 3
B[4] = 2


C[0] = 12
C[1] = 7
C[2] = 10
C[3] = 9
C[4] = 4

```

## 10. Kesimpulan

Setelah menyelesaikan topik ini, mahasiswa mampu:

- Memahami konsep array dan penggunaannya.
- Mengisi dan menuliskan nilai array satu dimensi statik menggunakan bahasa C.
- Membuat program yang memproses isi array satu dimensi statik dalam bahasa

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>1</b>
	<b>ARRAY SATU DIMENSI STATIK</b>	

### 11. Lembar Penilaian

Item Penilaian	Bobot	Nilai	Nilai X Bobot
Persiapan	5%		
Proses, Contoh, Cara melakukan pekerjaan, sistematika langkah kerja, kemandirian, partisipasi,	25%		
Sikap dan keselamatan kerja, Contoh sikap duduk, sikap menggunakan alat kerja	10%		
Hasil/Data/Produk; Program, Logika program, Kerapian Program, Tampilan Output	50%		
Waktu	10%		
Total	100%		

### 12. Daftar Pustaka

Kelley, Al and Pohl, Ira. 2003. *C by Dissection: The Essentials of C Programming*. Addison-Weasley.

Khannedy, Eko Kurniawan. 2007. Diktat Pemrograman C. Unikom. Bandung.

Liem, Inggriani. 2003. Catatan Singkat Bahasa C. Jurusan Teknik Informatika. Institut Teknologi Bandung.

	<b>POLITEKNIK NEGERI PADANG</b>	
	<b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>2</b>
	<b>ARRAY SATU DIMENSI DINAMIK</b>	

### 1. Judul : Array Satu Dimensi Dinamik

### 2. Jumlah Jam/Pertemuan : 3 jam x 1 pertemuan = 3 jam praktek

### 3. Tujuan Praktek

- Mahasiswa mampu menjelaskan konsep array satu dimensi dinamik dan penggunaannya serta membedakan dengan array statik.
- Mahasiswa mampu mendefinisikan, mendeklarasikan, mengisi dan menuliskan nilai array satu dimensi dinamik menggunakan bahasa C.
- Mahasiswa mampu membuat program yang memproses nilai array satu dimensi dinamik menggunakan bahasa C.

### 4. Teori Dasar

Pada modul praktikum sebelumnya, kita telah mempelajari bagaimana membuat dan menggunakan array satu dimensi yang ukurannya telah didefinisikan diawal program atau disebut array satu dimensi statik. Namun, ada saatnya ukuran array tidak diketahui sebelumnya sehingga harus didefinisikan di tengah-tengah eksekusi program. Untuk mengatasi hal ini, maka digunakanlah sebuah konsep alokasi memori komputer secara dinamik (menggunakan perintah **malloc**) untuk mendefinisikan sebuah array. Array ini dikenal dengan array dinamik.

Sebuah array satu dimensi dialokasikan secara dinamik menggunakan dua tahap. Pertama-tama, sebuah pointer terhadap tipe data array dibuat saat mendefinisikan nama array pada kamus.

```
tipe_data* nama_array;
```

Kemudian memori dengan ukuran yang diinginkan dialokasikan menggunakan fungsi **malloc()**.

```
nama_array =(tipe_data*) malloc (ukuran_array*sizeof (int));
```

dimana ukuran\_array merupakan sebuah variable bertipe integer yang menunjukkan



	<b>POLITEKNIK NEGERI PADANG</b>	
	<b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>2</b>
	<b>ARRAY SATU DIMENSI DINAMIK</b>	

ukuran array yang ingin dialokasikan. Yang perlu diperhatikan saat menggunakan array dinamik adalah bahwa setelah selesai menggunakan array ini, memori yang digunakan harus dikosongkan sehingga dapat digunakan kembali. Pengosongan memori ini dilakukan dengan menggunakan fungsi free()

```
free ((void*) nama_array);
```

Fungsi malloc(), sizeof(), dan free() terdapat pada pustaka stdlib.h sehingga setiap program yang menggunakan array dinamik harus meng-include stdlib.h.

Secara garis besar, perbedaan antara array satu dimensi statik dan dinamik hanya terletak pada pendefinisian ukuran array, sedangkan penggunaannya dalam program relative sama. Contoh program berikut memperlihatkan bagaimana sebuah array satu dimensi dinamik dibuat dan digunakan untuk menerima input nilai serta menuliskan isinya ke layar.

```
/* Nama File : arraydinamik.c
   Pembuat   : tuliskan nama dan no.bp anda
   Tgl pembuatan : tuliskan tanggal hari ini
   Deskripsi  : pembuatan array satu dimensi dinamik */

#include <stdio.h>
#include <stdlib.h>

void main()
{
    int* nilai; //nilai merupakan array satu dimensi dinamik bertipe integer
    int ukuran, indeks;

    printf("Input ukuran array : "); scanf("%d",&ukuran);
    //alokasi memori sesuai ukuran array
    nilai = (int*)malloc( ukuran *sizeof(int));

    //mengisi array
    for(indeks=0;indeks<ukuran;indeks++)
    {
        printf("nilai[%d] = ",indeks);
        scanf("%d",&nilai[indeks]);
    }

    //menampilkan isi array
    printf("\nIsi Array Nilai Dengan Ukuran %d\n", ukuran);
    for(indeks=0;indeks<ukuran;indeks++)
    {
        printf("nilai[%d] = %d\n",indeks,nilai[indeks]);
    }
}
```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>2</b>
	<b>ARRAY SATU DIMENSI DINAMIK</b>	

```

}

//pembebasan memori
free((void*)nilai);
}

```

### Output Program



```

Input ukuran array : 6
nilai[0] = 5
nilai[1] = 10
nilai[2] = 12
nilai[3] = 8
nilai[4] = 4
nilai[5] = 2

Isi Array Nilai Dengan Ukuran 6
nilai[0] = 5
nilai[1] = 10
nilai[2] = 12
nilai[3] = 8
nilai[4] = 4
nilai[5] = 2

```

### 5. Bahan/Peralatan yang digunakan

- Komputer
- Aplikasi CodeBlocks

### 6. Langkah Kerja

- Jalankan aplikasi CodeBlocks.
- Buatlah beberapa program berikut di komputer anda. Compile dan jalankan (Run). Perhatikan hasil keluarannya. Jalankan program dan analisa hasilnya.
  - Program yang meminta user menginputkan sejumlah bilangan bulat yang disimpan pada sebuah array satu dimensi dinamik, menghitung jumlah dan rata-ratanya, serta menampilkan isi array tersebut.

**Bandingkan kode program ini dengan kode program pada modul sebelumnya. Perhatikan letak perbedaannya.**

```

/* Nama File : array1_dinamik.c
   Pembuat   : tuliskan nama dan no.bp anda
   Tgl pembuatan : tuliskan tanggal hari ini
   Deskripsi  : menghitung total dan rata-rata nilai

```

	<b>POLITEKNIK NEGERI PADANG</b>	
	<b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>2</b>
	<b>ARRAY SATU DIMENSI DINAMIK</b>	

```

10 orang mahasiswa yang disimpan dalam array */

#include "stdio.h"
#include "stdlib.h"

void main()
{
    int* Nilai;
    int ukuran, indeks;
    float total,ratarata;

    //Input ukuran array
    printf("Banyak data yang akan diinputkan = ");
    scanf("%d",&ukuran);

    //Alokasi memori array satu dimensi dinamik
    Nilai = (int*) malloc (ukuran *sizeof(int));
    // Pembacaan data dari keyboard
    printf("Pembacaan data nilai \n");
    for (indeks=0;indeks<ukuran;indeks++)
    {
        printf("Nilai mahasiswa ke-%d = ",indeks+1);
        scanf("%d",&Nilai[indeks]);
    }

    // Perhitungan total dan rata-rata
    total=0;
    for (indeks=0;indeks<ukuran;indeks++)
        total=total+Nilai[indeks];
    //endfor

    ratarata=total/10.0;

    // Menampilkan data yang telah dimasukan dan rata-rata.
    printf("\n\nMenampilkan Data Nilai \n");
    for (indeks=0;indeks<ukuran;indeks++)
        printf("Nilai mahasiswa ke-%d = %3d\n",
            indeks+1,Nilai[indeks]);
    //endfor

    printf("\nTotal      = %0.2f\n",total);
    printf("\nRata-rata = %0.2f\n",ratarata);

    //pembebasan memri
    free((void*) Nilai);
}

```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>2</b>
	<b>ARRAY SATU DIMENSI DINAMIK</b>	

### Output Program

```

D:\KERJAJAN\PNP\JOBSHEET 2016\Program\array1_dinamik.exe
Banyak data yang akan diinputkan = 10
Pembacaan data nilai
Nilai mahasiswa ke-1 = 67
Nilai mahasiswa ke-2 = 80
Nilai mahasiswa ke-3 = 95
Nilai mahasiswa ke-4 = 65
Nilai mahasiswa ke-5 = 50
Nilai mahasiswa ke-6 = 70
Nilai mahasiswa ke-7 = 100
Nilai mahasiswa ke-8 = 45
Nilai mahasiswa ke-9 = 98
Nilai mahasiswa ke-10 = 88

Menampilkan Data Nilai
Nilai mahasiswa ke-1 = 67
Nilai mahasiswa ke-2 = 80
Nilai mahasiswa ke-3 = 95
Nilai mahasiswa ke-4 = 65
Nilai mahasiswa ke-5 = 50
Nilai mahasiswa ke-6 = 70
Nilai mahasiswa ke-7 = 100
Nilai mahasiswa ke-8 = 45
Nilai mahasiswa ke-9 = 98
Nilai mahasiswa ke-10 = 88

Total      = 758.00
Rata-rata  = 75.80

```

- b) Program yang mengisi nilai dua buah array satu dimensi (A dan B) kemudian melakukan penjumlahan isi array pada indeks yang sama dan menyimpan hasilnya pada array ke tiga (C).

```

/* Nama File : jumlah_array.c
   Pembuat   : tuliskan nama dan no.bp anda
   Tgl pembuatan : tuliskan tanggal hari ini
   Deskripsi : menjumlahkan isi array A + B = C(dinamik) */

#include "stdio.h"
#include "stdlib.h"

void main()
{
    float* A;
    float* B;
    float* C;
    int n,i;

    printf("Masukkan banyak data : "); scanf("%d",&n);

    //alokasi memori array A, B, dan C
    A = (float*) malloc (n *sizeof(int));
    B = (float*) malloc (n *sizeof(int));
    C = (float*) malloc (n *sizeof(int));

```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>2</b>
	<b>ARRAY SATU DIMENSI DINAMIK</b>	

```

//Input isi array A sebanyak n buah
printf("\n");
for(i=0;i<n;i++)
{
    printf("A[%d] = ",i); scanf("%f",&A[i]);
}

//Input isi array B sebanyak n buah
printf("\n");
for(i=0;i<n;i++)
{
    printf("B[%d] = ",i); scanf("%f",&B[i]);
}

//Penjumlahan isi array A + B = C
for(i=0;i<n;i++)
{
    C[i]=A[i]+B[i];
}

//Menampilkan isi array C ke layar output
printf("\n**** C=A+B ****\n");
for(i=0;i<n;i++)
{
    printf("C[%d] = %.2f\n",i,C[i]);
}

//pembebasan memori
free((void*)A);
free((void*)B);
free((void*)C);
}

```

	<b>POLITEKNIK NEGERI PADANG</b>	
	<b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>2</b>
	<b>ARRAY SATU DIMENSI DINAMIK</b>	

### Output Program

```

D:\KERJAAN\PNP\JOBSHEET 2016\Program\jumlah_ar
Masukkan banyak data : 5
A[0] = 5.5
A[1] = 3
A[2] = 4.2
A[3] = 7.8
A[4] = 1.2

B[0] = 1
B[1] = 2
B[2] = 3
B[3] = 4
B[4] = 5

**** C=A+B ****
C[0] = 6.50
C[1] = 5.00
C[2] = 7.20
C[3] = 11.80
C[4] = 6.20

```

## 7. Latihan

- Buatlah **program untuk menyisipkan angka pada indeks yang diinginkan** dalam sebuah array 1 dimensi dinamis
- Buatlah **program untuk menghapus angka pada indeks yang diinginkan** dalam sebuah array dinamis yang isinya random
- Ubahlah program-program latihan yang terdapat pada modul Array Satu Dimensi Statik menggunakan konsep array satu dimensi dinamik.

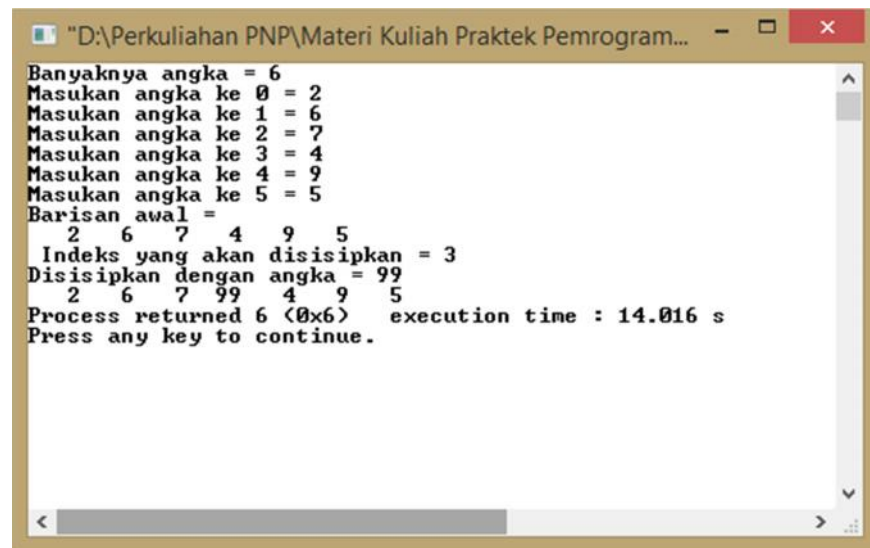
## 8. Keselamatan dan Kesehatan Kerja (K3)

- Periksalah komputer yang anda gunakan, apakah terjadi kerusakan atau tidak. Jika terjadi kerusakan, segera melapor pada Dosen ataupun Teknisi Labor.
- Jika akan melakukan peng-copy-an data, pastikan komputer ataupun alat penyimpanan data Anda terbebas dari virus.
- Selalu matikan komputer setiap Anda selesai praktikum.
- Jangan meninggalkan sampah apapun di ruangan labor komputer.

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>2</b>
	<b>ARRAY SATU DIMENSI DINAMIK</b>	

## 9. Hasil Pengujian dan Pembahasan

- a. Program untuk menyisipkan angka pada indeks yang diinginkan

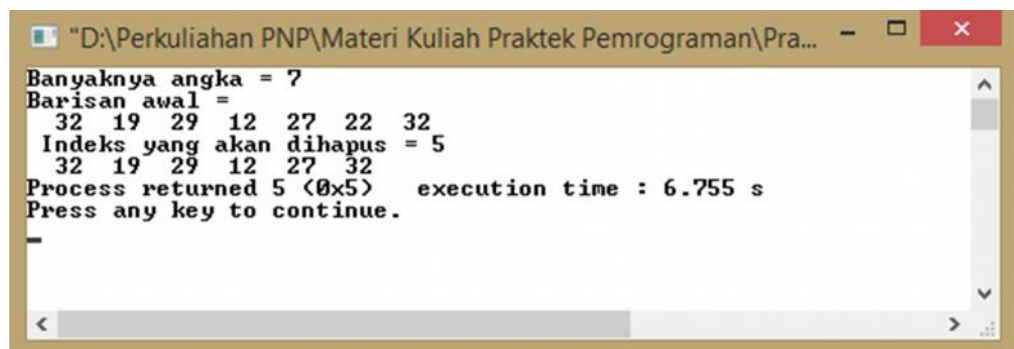


```

"D:\Perkuliahan PNP\Materi Kuliah Praktek Pemrograman...
Banyaknya angka = 6
Masukan angka ke 0 = 2
Masukan angka ke 1 = 6
Masukan angka ke 2 = 7
Masukan angka ke 3 = 4
Masukan angka ke 4 = 9
Masukan angka ke 5 = 5
Barisan awal =
2 6 7 4 9 5
Indeks yang akan disisipkan = 3
Disisipkan dengan angka = 99
2 6 7 99 4 9 5
Process returned 6 (0x6)   execution time : 14.016 s
Press any key to continue.

```

- b. Buatlah program untuk menghapus angka pada indeks yang diinginkan



```

"D:\Perkuliahan PNP\Materi Kuliah Praktek Pemrograman\Pra...
Banyaknya angka = 7
Barisan awal =
32 19 29 12 27 22 32
Indeks yang akan dihapus = 5
32 19 29 12 27 32
Process returned 5 (0x5)   execution time : 6.755 s
Press any key to continue.

```

- c. Bandingkan kode program latihan array satu dimensi statik dengan program yang menggunakan konsep array satu dimensi dinamik. Perhatikan perbedaan dan persamaannya.

	<b>POLITEKNIK NEGERI PADANG</b>	
	<b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>2</b>
	<b>ARRAY SATU DIMENSI DINAMIK</b>	

## 10. Kesimpulan

Setelah menyelesaikan topik ini, mahasiswa mampu:

- Memahami konsep array satu dimensi dinamik, penggunaannya serta perbedaannya dengan array satu dimensi statik.
- Mengisi dan menuliskan array satu dimensi dinamik menggunakan bahasa C.
- Membuat program yang memproses isi array satu dimensi dinamik dalam C.

## 11. Lembar Penilaian

Item Penilaian	Bobot	Nilai	Nilai X Bobot
Persiapan	5%		
Proses, Contoh, Cara melakukan pekerjaan, sistematika langkah kerja, kemandirian, partisipasi,	25%		
Sikap dan keselamatan kerja, Contoh sikap duduk, sikap menggunakan alat kerja	10%		
Hasil/Data/Produk; Program, Logika program, Kerapian Program, Tampilan Output	50%		
Waktu	10%		
Total	100%		

## 12. Daftar Pustaka



Kelley, Al and Pohl, Ira. 2003. *C by Dissection: The Essentials of C Programming*. Addison-Weasley.

Khannedy, Eko Kurniawan. 2007. Diktat Pemrograman C. Unikom. Bandung.

Liem, Inggriani. 2003. Catatan Singkat Bahasa C. Jurusan Teknik Informatika. Institut Teknologi Bandung.

<http://www.phy225.dept.shuf.ac.uk/mediawiki/index.php/>. Arrays, Dynamic Array Allocation - Programming in C. Tanggal akses 17 April 2014



	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>3</b>
	<b>SORTING (METODE PENGURUTAN)</b>	

### 1. Judul : Sorting (Metode Pengurutan)

2. Jumlah Jam/Pertemuan : 3 jam x 1 pertemuan = 3 jam praktek

### 3. Tujuan Praktek



- Mahasiswa mampu mengenal dan menjelaskan konsep pengurutan data dalam Bahasa C
- Mahasiswa mampu mendefinisikan dan mendeklarasikan metode selection sort pada array satu dimensi.
- Mahasiswa mampu mendefinisikan dan mendeklarasikan metode bubble sort pada array satu dimensi.
- Mahasiswa mampu membuat program menggunakan metode pengurutan.

### 4. Teori Dasar.

Pengurutan data (*sorting*) didefinisikan sebagai suatu proses untuk menyusun kembali humpunan obyek menggunakan aturan tertentu. Menurut Microsoft Bookshelf, definisi algoritma pengurutan adalah algoritma untuk meletakkan kumpulan elemen data ke dalam urutan tertentu berdasarkan satu atau beberapa kunci dalam tiap-tiap elemen.

Ada dua macam urutan yang biasa digunakan dalam proses pengurutan yaituurut naik (*ascending*) yaitu dari data yang mempunyai nilai paling kecil sampai paling besar urut turun (*descending*) yaitu data yang mempunyai nilai paling besar sampai paling kecil.

Contoh : data bilangan 5, 2, 6 dan 4 dapat diurutkan naik menjadi 2, 4, 5, 6 atau diurutkan turun menjadi 6, 5, 4, 2. Pada data yang bertipe char, nilai data dikatakan lebih kecil atau lebih besar dari yang lain didasarkan pada urutan relatif (*collating sequence*) seperti dinyatakan dalam tabel ASCII.

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>3</b>
	<b>SORTING (METODE PENGURUTAN)</b>	

Keuntungan dari data yang sudah dalam keadaan terurutkan antara lain :



- Data mudah dicari (misalnya dalam buku telepon atau kamus bahasa), mudah untuk dibetulkan, dihapus, disisipi atau digabungkan. Dalam keadaan terurutkan, kita mudah melakukan pengecekan apakah ada data yang hilang
- Melakukan kompilasi program komputer jika tabel-tabel simbol harus dibentuk
- Mempercepat proses pencarian data yang harus dilakukan berulang kali.

Data yang diurutkan sangat bervariasi, dalam hal jumlah data maupun jenis data yang akan diurutkan. Tidak ada algoritma terbaik untuk setiap situasi yang kita hadapi, bahkan cukup sulit untuk menentukan algoritma mana yang paling baik untuk situasi tertentu karena ada beberapa faktor yang mempengaruhi efektifitas algoritma pengurutan. Beberapa faktor yang berpengaruh pada efektifitas suatu algoritma pengurutan antara lain :

- Banyak data yang diurutkan
- Kapasitas pengingat apakah mampu menyimpan semua data yang kita miliki
- Tempat penyimpanan data, misalnya piringan, pita atau kartu, atau media penyimpan yang lain.

Pemilihan algoritma sangat ditentukan oleh struktur data yang digunakan. Metode pengurutan yang digunakan dapat diklasifikasikan menjadi dua katagori yaitu :

- Pengurutan internal, yaitu pengurutan dengan menggunakan larik (*array*). Larik tersimpan dalam memori utama komputer
- Pengurutan eksternal, yaitu pengurutan dengan menggunakan berkas

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>3</b>
	<b>SORTING (METODE PENGURUTAN)</b>	

(*sequential access file*). Berkas tersimpan dalam penganget luar, misalnya cakram atau pita magnetis.

Untuk menggambarkan pengurutan dengan larik, bisa kita bayangkan semua kartu terletak di hadapan kita sehingga semua kartu terlihat dengan jelas nomornya. Pada penyusunan kartu sebagai sebuah berkas, kita bayangkan semua kartu kita tumpuk sehingga hanya kartu bagian atas saja yang bisa kita lihat nomornya. Berikut ini akan dijelaskan beberapa metode pengurutan beserta analisisnya.

### Deklarasi Larik

Pada pengurutan larik, ada beberapa aspek yang perlu dipertimbangkan, antara lain aspek menyangkut kapasitas penganget yang ada dan aspek waktu, yaitu waktu yang diperlukan untuk melakukan permutasi sehingga semua elemen akhirnya menjadi terurutkan.



Deklarasi larik yang digunakan adalah larik dimensi satu (*vektor*) dengan elemennya bertipe integer.

```
#define Max 100;
int Data[Max];
```

Pada deklarasi diatas, **Max** adalah banyaknya elemen vektor. Anda bisa mengubah nilai konstanta **Max** sesuai kebutuhan. Indeks larik dimulai dari 0. Data yang sebenarnya disimpan mulai dari indeks 0.

Selain deklarasi diatas, proses yang juga selalu digunakan pada algoritma pengurutan adalah proses menukarkan elemen. Di bawah ini satu prosedur sederhana untuk menukarkan nilai dua buah elemen A dan B.

```
void Tukar (int *a, int *b)
{
    int temp;
    temp = *a;
    *a = *b;
    *b = temp;
}
```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>3</b>
	<b>SORTING (METODE PENGURUTAN)</b>	

Prosedur tukar diatas akan digunakan pada beberapa prosedur pengurutan yang akan dijelaskan dibawah ini.

### **Metode Seleksi (*Selection Sort*)**



Metode seleksi melakukan pengurutan dengan cara mencari data yang terkecil kemudian menukarkannya dengan data yang digunakan sebagai acuan atau sering dinamakan pivot.

Proses pengurutan dengan metode seleksi dapat dijelaskan sebagai berikut : langkah pertama dicari data terkecil dari data pertama sampai data terakhir. Kemudian data terkecil ditukar dengan data pertama. Dengan demikian, data pertama sekarang mempunyai nilai paling kecil dibanding data yang lain. Langkah kedua, data terkecil kita cari mulai dari data kedua sampai terakhir. Data terkecil yang kita peroleh ditukar dengan data kedua dan demikian seterusnya sampai semua elemen dalam keadaan terurutkan.

Algoritma seleksi dapat dituliskan sebagai berikut :

- $i \leftarrow 0$
- selama  $(i < N-1)$  kerjakan baris 3 sampai dengan 9
- $k \leftarrow i$
- $j \leftarrow i + 1$
- Selama  $(j < N)$  kerjakan baris 6 dan 7
- Jika  $(Data[k] > Data[j])$  maka  $k \leftarrow j$
- $j \leftarrow j + 1$
- Tukar  $Data[i]$  dengan  $Data[k]$
- $i \leftarrow i + 1$

Untuk lebih memperjelas langkah-langkah algoritma seleksi dapat dilihat pada tabel 4.2. Proses pengurutan pada tabel 4.2 dapat dijelaskan sebagai berikut:

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>3</b>
	<b>SORTING (METODE PENGURUTAN)</b>	

- ✓ Pada saat  $i=0$ , data terkecil antara data ke-1 s/d ke-9 adalah data ke-4, yaitu 3, maka data ke-0 yaitu 12 ditukar dengan data ke-4 yaitu 3.
- ✓ Pada saat  $i=1$ , data terkecil antara data ke-2 s/d ke-9 adalah data ke-2, yaitu 9, maka data ke-1 yaitu 35 ditukar dengan data ke-2 yaitu 9.
- ✓ Pada saat  $i=2$ , data terkecil antara data ke-3 s/d ke-9 adalah data ke-3, yaitu 11, maka data ke-2 yaitu 35 ditukar dengan data ke-3 yaitu 11.
- ✓ Pada saat  $i=3$ , data terkecil antara data ke-4 s/d ke-9 adalah data ke-4, yaitu 12, maka data ke-3 yaitu 35 ditukar dengan data ke-4 yaitu 12.
- ✓ Dan seterusnya.



**Tabel 3.1 Proses Pengurutan dengan Metode Seleksi**

Iterasi	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]	Data[8]
Awal	12	35	9	11	3	17	23	15	31
$i=0$	<b>12</b>	35	9	11	<b>3</b>	17	23	15	31
$i=1$	3	<b>35</b>	<b>9</b>	11	12	17	23	15	31
$i=2$	3	9	<b>35</b>	<b>11</b>	12	17	23	15	31
$i=3$	3	9	11	<b>35</b>	<b>12</b>	17	23	15	31
$i=4$	3	9	11	12	<b>35</b>	17	23	<b>15</b>	31
$i=5$	3	9	11	12	15	<b>17</b>	23	35	31
$i=6$	3	9	11	12	15	17	<b>23</b>	35	31
$i=7$	3	9	11	12	15	17	20	<b>35</b>	31
$i=8$	3	9	11	12	15	17	20	23	<b>31</b>
Akhir	3	9	11	12	15	17	20	23	31

Di bawah ini merupakan prosedur yang menggunakan metode seleksi.

```
void SelectionSort()
{
    int i, j, k;

    for(i=0; i<Max-1; i++)
    {
        k = i;
        for (j=i+1; j<Max; j++)
```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>3</b>
	<b>SORTING (METODE PENGURUTAN)</b>	

```

        if(Data[k] > Data[j])
            k = j;
        Tukar(&Data[i], &Data[k]);
    }
}

```

Dari algoritma dan prosedur diatas, jumlah perbandingan ( $=C$ ) metode seleksi adalah :

$$C = N(N - 1) / 2$$

Jumlah penukaran ( $=M$ ) pada metode seleksi tergantung keadaan datanya. Penukaran minimum terjadi bila data sudah dalam keadaan urut, sebaliknya jumlah penukaran maksimum terjadi bila data dalam keadaan urut terbalik. Jumlah penukaran minimum dan maksimum dapat dirumuskan sebagai berikut :

$$M_{min} = 3(N - 1) M$$



$$M_{max} = [N^2 / 4] + 3(N - 1)$$

### **Metode Gelembung (*Bubble Sort*)**

Algoritma Bubble Sort ini merupakan proses pengurutan yang secara berangsur-angsur berpindah ke posisi yang tepat karena itulah dinamakan Bubble yang artinya gelembung. Algoritma ini akan mengurutkan data dari yang terbesar ke yang terkecil (*ascending*) atau sebaliknya (*descending*).

Secara sederhana, bisa didefinisikan algoritma Bubble Sort adalah pengurutan dengan cara pertukaran data dengan data disebelahnya secara terus menerus sampai dalam satu iterasi tertentu tidak ada lagi perubahan.

Metode Bubble Sort merupakan metode pengurutan yang mengambil proses dengan menggunakan bubble atau exchange. Perbandingan data dilakukan dari posisi pertama atau posisi terakhir bergeser satu persatu sampai semua data dibandingkan. Jika terdapat N data dan data terkoleksi dari urutan 0 sampai dengan N-1 maka algoritma pengurutan dengan metode bubble sort adalah sebagai berikut:

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>3</b>
	<b>SORTING (METODE PENGURUTAN)</b>	



1. Bandingkan posisi data  $i = 0$  dan  $j = 1$ . 2. Jika data diposisi  $i$  lebih besar daripada data diposisi  $j$ , maka data diposisi  $i$  ditukar dengan data diposisi  $j$  (swap). Jika tidak penukaran posisi tidak dilakukan. 3. Kemudian, lakukan perbandingan data diposisi  $i = 1$  dan data diposisi  $j = 2$ . Lakukan langkah 2, begitu juga untuk data berikutnya hingga  $i = N-2$  dan  $j = N-1$ . 4. Ulangi langkah 1, 2 dan 3 untuk data diposisi 0 sampai dengan data diposisi  $N-2$ , karena data di posisi  $N-1$  adalah data dengan nilai terbesar. Untuk tahap selanjutnya data yang dibandingkan akan semakin berkurang sebab data dengan nilai yang lebih besar akan terposisi dibagian sebelah kanan data. Tabel berikut menjelaskan langkah per langkah bagaimana metode bubble sort ini bekerja.

**Tabel 3.2 Proses Pengurutan dengan metode Gelembung**

44	55	12	42	94	18	06	67	<b>Data Awal</b>
44	12	42	55	18	06	67	94	Bandingkan data ke 1 dengan data ke 2, bila data ke 1 lebih besar tukar, seterusnya sampai data yang terbesar terletak di data ke 8
12	42	44	18	06	55	67	94	Bandingkan data ke 1 dengan data ke 2, bila data ke 1 lebih besar tukar, seterusnya sampai data yang terbesar terletak di data ke 7
12	42	18	06	44	55	67	94	Bandingkan data ke 1 dengan data ke 2, bila data ke 1 lebih besar tukar, seterusnya sampai data yang terbesar terletak di data ke 6
12	18	06	42	44	55	67	94	Bandingkan data ke 1 dengan data ke 2, bila data ke 1 lebih besar tukar, seterusnya sampai data yang terbesar terletak di data ke 5
12	06	18	42	44	55	67	94	Bandingkan data ke 1 dengan data ke 2, bila data ke 1 lebih besar tukar, seterusnya sampai data yang terbesar terletak di data ke 4
06	12	18	42	44	55	67	94	Bandingkan data ke 1 dengan data ke 2, bila data ke 1 lebih besar tukar, seterusnya sampai data yang terbesar terletak di data ke 3
06	12	18	42	44	55	67	94	Bandingkan data ke 1 dengan data ke 2, bila data ke 1 lebih besar tukar, seterusnya sampai data yang terbesar terletak di data ke 2
06	12	18	42	44	55	67	94	<b>Data setelah terurut</b>

## 5. Bahan/Peralatan yang digunakan

- Komputer

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>3</b>
	<b>SORTING (METODE PENGURUTAN)</b>	

b. Aplikasi CodeBlocks

## 6. Langkah Kerja

- 1) Jalankan program CodeBlocks.
- 2) Buatlah beberapa program berikut di komputer anda. Compile dan jalankan (Run). Perhatikan hasil keluarannya. Jalankan program dan analisa hasilnya.
  - a) Program bubble sort untuk mengurutkan nilai dari kecil ke besar

```

/* Nama File : bubble_sort.c
   Pembuat   : tuliskan nama dan no.bp anda
   Tgl pembuatan : tuliskan tanggal hari ini
   Deskripsi  : mengurutkan isi tabel dari kecil ke besar
                menggunakan algoritma bubble sort */

#include "stdio.h"

void main()
{
    int A[100], n, i, d, swap;
    printf("Masukkan banyak isi tabel : ");
    scanf("%d", &n);



    printf("Masukkan %d buah angka\n", n);
    for ( i = 0 ; i<n ; i++ )
        scanf("%d", &A[i]);

    for ( i = 1 ; i <= ( n - 1 ) ; i++ )
    {
        for ( d = n-1 ; d>=i ; d-- )
        {
            if ( A[d] < A[d-1] )
            {
                swap = A[d];
                A[d] = A[d-1];
                A[d-1] = swap;
            }
        }
    }

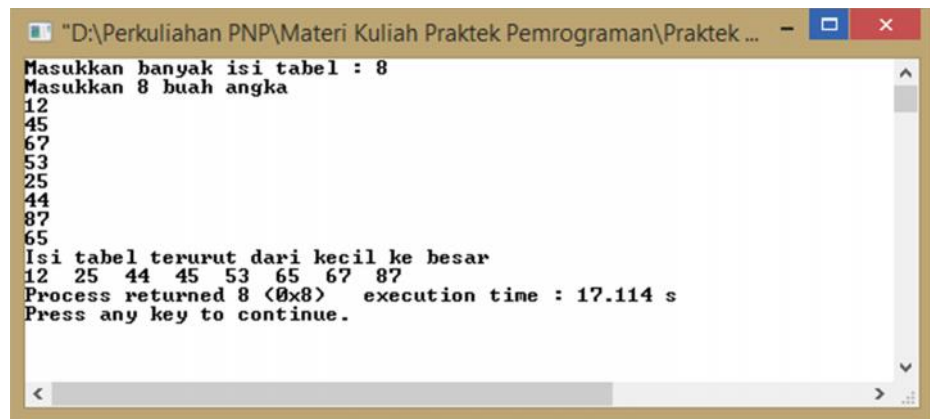
    printf("Isi tabel terurut dari kecil ke besar\n");
    for ( i = 0 ; i < n ; i++ )
        printf("%d  ", A[i]);
}

```



	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>3</b>
	<b>SORTING (METODE PENGURUTAN)</b>	

### Output Program



```

"D:\Perkuliahan PNP\Materi Kuliah Praktek Pemrograman\Praktek ... - [X]
Masukkan banyak isi tabel : 8
Masukkan 8 buah angka
12
45
67
53
25
44
87
65
Isi tabel terurut dari kecil ke besar
12 25 44 45 53 65 67 87
Process returned 8 (0x8)   execution time : 17.114 s
Press any key to continue.

```

- b) Program selection sort nilai terbesar ke nilai terkecil (max sort), dengan ukuran array ditentukan user (dinamis) dan isi random.

```

/* Nama File : max_sort.c
   Pembuat   : tuliskan nama dan no.bp anda
   Tgl pembuatan : tuliskan tanggal hari ini
   Deskripsi : mengurutkan isi tabel dari besar ke kecil
               menggunakan algoritma selection sort (max
               sort) */

#include "stdio.h"
#include "time.h"



void main()
{
    int *A;
    int n, i, j, k, swap, imax;

    printf("Masukkan banyak isi tabel : ");
    scanf("%d", &n);
    A=(int*)malloc(n*sizeof(int));

    srand(time(NULL));

    for (i=0;i<n;i++)
    {
        A[i]=rand()/1000;
    }
    printf("Isi Array Sebelum Terurut \n");
    for ( i = 0 ; i < n ; i++ )

```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>3</b>
	<b>SORTING (METODE PENGURUTAN)</b>	

```

printf("%d  ", A[i]);

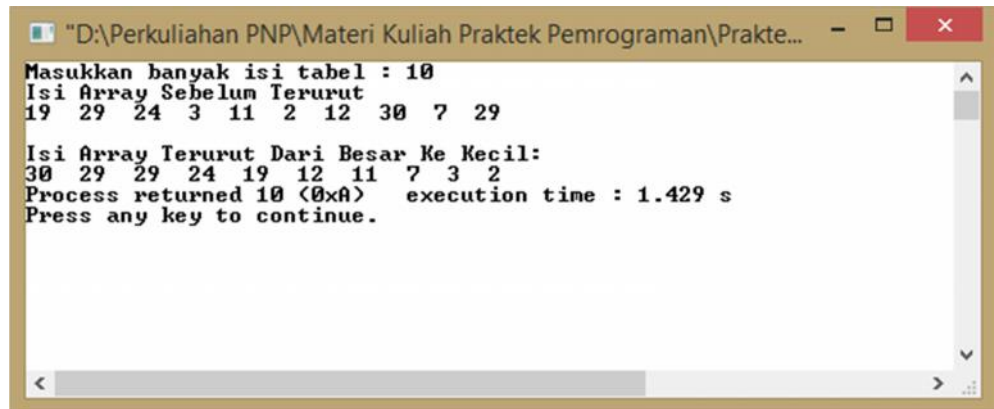
for ( i = 0 ; i<n-1 ; i++ )
{
    imax=i;
    for ( j = i+1 ; j<=n-1 ; j++ )
    {
        if(A[imax]<A[j])//cari indeks tempat nilai max
        imax=j;
    }

    //letakkan nilai max di indeks awal array
    swap = A[imax];
    A[imax] = A[i];
    A[i] = swap;
}

printf("\n\nIsi Array Terurut Dari Besar Ke Kecil:\n");
for ( i = 0 ; i < n ; i++ )
    printf("%d  ", A[i]);
}

```

### Output Program



```



"D:\Perkuliahan PNP\Materi Kuliah Praktek Pemrograman\Prakte...
Masukkan banyak isi tabel : 10
Isi Array Sebelum Terurut
19 29 24 3 11 2 12 30 7 29

Isi Array Terurut Dari Besar Ke Kecil:
30 29 24 19 12 11 7 3 2
Process returned 10 (0xA)   execution time : 1.429 s
Press any key to continue.

```

## 7. Latihan

- Buatlah program yang meminta user menginputkan ukuran array dan array diisi secara random. Tampilkan proses **selection sort dari nilai kecil ke besar (minsort)**. Tampilkan kondisi array sebelum dan sesudah proses pengurutan.

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>3</b>
	<b>SORTING (METODE PENGURUTAN)</b>	

- b. Buatlah **program yang menampilkan setiap fase bubble sort** dari kondisi awal ke kondisi array yang sudah terurut. Ukuran array dinamis dan isi random.
- c. Buatlah **program yang menampilkan setiap fase selection sort (max sort)** dari kondisi awal ke kondisi array yang sudah terurut. Ukuran array dinamis dan isi random.

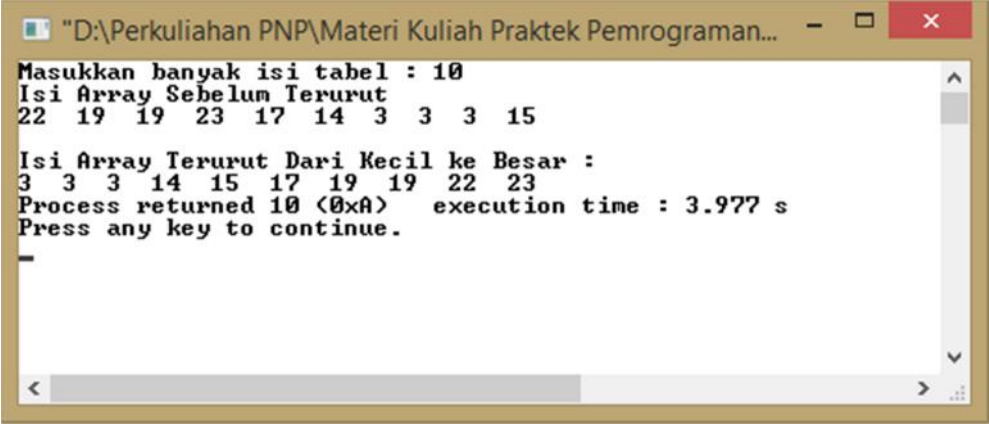
## 8. Keselamatan dan Kesehatan Kerja (K3)

- a. Periksa komputer yang anda gunakan, apakah terjadi kerusakan atau tidak. Jika terjadi kerusakan, segera melapor pada Dosen ataupun Teknisi Labor.
- b. Jika akan melakukan peng-copy-an data, pastikan komputer ataupun alat penyimpanan data Anda terbebas dari virus.
- c. Selalu matikan komputer setiap Anda selesai praktikum.
- d. Jangan meninggalkan sampah apapun di ruangan labor komputer.

## 9. Hasil Pengujian dan Pembahasan

Selesaikanlah soal latihan dan kemudian cocokkan hasil output program Anda dengan hasil program berikut ini. Jalankan juga program Anda untuk nilai-nilai lainnya.

- a. Program selection sort dari nilai terkecil ke nilai terbesar





```

"D:\Perkuliahan PNP\Materi Kuliah Praktek Pemrograman...
Masukkan banyak isi tabel : 10
Isi Array Sebelum Terurut
22 19 19 23 17 14 3 3 3 15

Isi Array Terurut Dari Kecil ke Besar :
3 3 3 14 15 17 19 19 22 23
Process returned 10 (0xA)   execution time : 3.977 s
Press any key to continue.

```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>3</b>
	<b>SORTING (METODE PENGURUTAN)</b>	



b. Program yang menampilkan setiap fase bubble sort

```

D:\Perkuliahan PNP\Materi Kuliah Praktek Pemrograman\Prakt...
Ukuran array = 6
          TAHAPAN BUBBLE SORT
FASE KE 1 =
21      32      10      27      28      26
21      32      10      27      26      28
21      32      10      26      27      28
21      32      10      26      27      28
21      10      32      26      27      28
10      21      32      26      27      28
FASE KE 2 =
10      21      32      26      27      28
10      21      32      26      27      28
10      21      32      26      27      28
10      21      26      32      27      28
10      21      26      32      27      28
FASE KE 3 =
10      21      26      32      27      28
10      21      26      32      27      28
10      21      26      27      32      28
10      21      26      27      32      28
FASE KE 4 =
10      21      26      27      32      28
10      21      26      27      28      32
10      21      26      27      28      32
FASE KE 5 =
10      21      26      27      28      32
10      21      26      27      28      32
Process returned 1 (0x1)   execution time : 1.287 s
Press any key to continue.

```

c. Program yang menampilkan setiap fase selection sort (max sort)

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>3</b>
	<b>SORTING (METODE PENGURUTAN)</b>	

```

D:\Perkuliahan PNP\Materi Kuliah Praktek Pemrograman\Pra...
Banyaknya angka = 7
22      27      14      8      29      11      16
TAHAPAN SELECTION SORT <MAX SORT>

Fase ke 0
22      27      14      8      29      11      16
29      27      14      8      22      11      16

Fase ke 1
29      27      14      8      22      11      16
29      27      14      8      22      11      16

Fase ke 2
29      27      14      8      22      11      16
29      27      22      8      14      11      16

Fase ke 3
29      27      22      8      14      11      16
29      27      22      16      14      11      8

Fase ke 4
29      27      22      16      14      11      8
29      27      22      16      14      11      8

Fase ke 5
29      27      22      16      14      11      8
29      27      22      16      14      11      8

Fase ke 6
29      27      22      16      14      11      8
29      27      22      16      14      11      8

Sesudah terurut
29      27      22      16      14      11      8
Process returned 1 (0x1)   execution time : 2.125 s
Press any key to continue.

```



## 10. Kesimpulan

Setelah menyelesaikan topik ini, mahasiswa mampu:

- Memahami konsep dan kegunaan metode pengurutan (sorting)
- Mendefinisikan dan mendeklarasikan metode selection sort (baik min\_sort maupun max\_sort) maupun Bubble Sort untuk pengurutan array 1 dimensi
- Membuat program yang memproses pengurutan(sorting)

## 11. Lembar Penilaian

Item Penilaian	Bobot	Nilai	Nilai X Bobot
Persiapan	5%		
Proses, Contoh,Cara melakukan pekerjaan, sistimatika langkah kerja, kemandirian, partisipasi,	25%		
Sikap dan keselamatan kerja, Contoh sikap duduk, sikap menggunakan	10%		

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>3</b>
	<b>SORTING (METODE PENGURUTAN)</b>	

alat kerja			
Hasil/Data/Produk; Program, Logika program, Kerapian Program, Tampilan Output	50%		
Waktu	10%		
Total	100%		



## 12. Daftar Pustaka

Hartono, J.2001. *Konsep Dasar Bahasa Pemrograman C*. Andi Offset Yogyakarta

Kelley, Al and Pohl, Ira. 2003. *C by Dissection: The Essentials of C Programming*. Addison-Weasley.

Khannedy, Eko Kurniawan. 2007. Diktat Pemrograman C. Unikom. Bandung.

Liem, Inggriani. 2003. Catatan Singkat Bahasa C. Jurusan Teknik Informatika. Institut Teknologi Bandung.

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>4</b>
	<b>SEARCHING (METODE PENCARIAN)</b>	

### 1. +Judul : Searching (Metode Pencarian)

2. Jumlah Jam/Pertemuan : 3 jam x 2 pertemuan = 6 jam praktek

### 3. Tujuan Praktek



- Mahasiswa mampu mengenal dan menjelaskan konsep pencarian data dalam Bahasa C
- Mahasiswa mampu mendefinisikan dan mendeklarasikan metode sequential search pada array satu dimensi.
- Mahasiswa mampu mendefinisikan dan mendeklarasikan metode binary search pada array satu dimensi.
- Mahasiswa mampu membuat program menggunakan metode pencarian.

### 4. Teori Dasar.

Pencarian data sering juga disebut *table look-up* atau *storage and retrieval information* adalah suatu proses untuk mengumpulkan sejumlah informasi di dalam pengingat komputer dan kemudian mencari kembali informasi yang diperlukan secepat mungkin.

Algoritma pencarian (*searching algorithm*) adalah algoritma yang menerima sebuah argumen kunci dan dengan langkah-langkah tertentu akan mencari rekaman dengan kunci tersebut. Setelah proses pencarian dilaksanakan, akan diperoleh salah satu dari dua kemungkinan, yaitu data yang dicari ditemukan (*successful*) atau tidak ditemukan (*unsuccessful*).

Metode pencarian data dapat dilakukan dengan dua cara yaitu **pencarian internal** (*internal searching*) dan **pencarian eksternal** (*external searching*). Pada pencarian internal, semua rekaman yang diketahui berada dalam pengingat komputer

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>4</b>
	<b>SEARCHING (METODE PENCARIAN)</b>	

sedangkan pada pencarian eksternal, tidak semua rekaman yang diketahui berada dalam pengingat komputer, tetapi ada sejumlah rekaman yang tersimpan dalam penyimpanan luar misalnya pita atau cakram magnetis.

Selain itu metode pencarian data juga dapat dikelompokkan menjadi **pencarian statis** (*static searching*) dan **pencarian dinamis** (*dynamic searching*). Pada pencarian statis, banyaknya rekaman yang diketahui dianggap tetap, pada pencarian dinamis, banyaknya rekaman yang diketahui bisa berubah-ubah yang disebabkan oleh penambahan atau penghapusan suatu rekaman.

Ada dua macam teknik pencarian yaitu pencarian sekuensial dan pencarian biner. Perbedaan dari dua teknik ini terletak pada keadaan data. Pencarian sekuensial digunakan apabila data dalam keadaan acak atau tidak teratur. Sebaliknya, pencarian biner digunakan pada data yang sudah dalam keadaan urut.

### **Pencarian Berurutan (*Sequential Searching*)**



Pencarian berurutan sering disebut pencarian linear merupakan metode pencarian yang paling sederhana. Pencarian berurutan menggunakan prinsip sebagai berikut : data yang ada dibandingkan satu per satu secara berurutan dengan yang dicari sampai data tersebut ditemukan atau tidak ditemukan.

Pada dasarnya, pencarian ini hanya melakukan pengulangan dari 1 sampai dengan jumlah data. Pada setiap pengulangan, dibandingkan data ke-i dengan yang dicari. Apabila sama, berarti data telah ditemukan. Sebaliknya apabila sampai akhir pengulangan tidak ada data yang sama, berarti data tidak ada. Pada kasus yang paling buruk, untuk N elemen data harus dilakukan pencarian sebanyak N kali pula.

Algoritma pencarian berurutan dapat dituliskan sebagai berikut :

- $i \leftarrow 0$
- ketemu     false



	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>4</b>
	<b>SEARCHING (METODE PENCARIAN)</b>	

- Selama (tidak ketemu) dan  $(i \leq N)$  kerjakan baris 4
- Jika  $(Data[i] = x)$  maka ketemu true, jika tidak  $i = i + 1$
- Jika (ketemu) maka i adalah indeks dari data yang dicari, jika tidak data tidak ditemukan

Di bawah ini merupakan fungsi untuk mencari data menggunakan pencarian sekuensial.



```
int SequentialSearch(int x)
{
    int i = 0;
    bool ketemu = false;

    while ((!ketemu) && (i < Max))
    {
        if(Data[i] == x)
            ketemu = true;
        else
            i++;
    }
    if(ketemu)
        return i;
    else
        return -1;
}
```

Fungsi diatas akan mengembalikan indeks dari data yang dicari. Apabila data tidak ditemukan maka fungsi diatas akan mengembalikan nilai -1.

### **Pencarian Biner (Binary Search)**

Salah satu syarat agar pencarian biner dapat dilakukan adalah data sudah dalam keadaan urut. Dengan kata lain, apabila data belum dalam keadaan urut, pencarian biner tidak dapat dilakukan. Dalam kehidupan sehari-hari, sebenarnya kita juga sering menggunakan pencarian biner. Misalnya saat ingin mencari suatu kata dalam kamus

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>4</b>
	<b>SEARCHING (METODE PENCARIAN)</b>	

Prinsip dari pencarian biner dapat dijelaskan sebagai berikut : mula-mula diambil posisi awal 0 dan posisi akhir =  $N - 1$ , kemudian dicari posisi data tengah dengan rumus  $(\text{posisi awal} + \text{posisi akhir}) / 2$ . Kemudian data yang dicari dibandingkan dengan data tengah. Jika lebih kecil, proses dilakukan kembali tetapi posisi akhir dianggap sama dengan posisi tengah  $-1$ . Jika lebih besar, proses dilakukan kembali tetapi posisi awal dianggap sama dengan posisi tengah  $+1$ . Demikian seterusnya sampai data tengah sama dengan yang dicari.

Untuk lebih jelasnya perhatikan contoh berikut. Misalnya ingin mencari data 17 pada sekumpulan data berikut :

3	9	11	12	15	17	20	23	31	35
awal		tengah				akhir			



Mula-mula dicari data tengah, dengan rumus  $(0 + 9) / 2 = 4$ . Berarti data tengah adalah data ke-4, yaitu 15. Data yang dicari, yaitu 17, dibandingkan dengan data tengah ini. Karena  $17 > 15$ , berarti proses dilanjutkan tetapi kali ini posisi awal dianggap sama dengan posisi tengah  $+1$  atau 5.

3	9	11	12	15	17	20	23	31	35
awal					tengah			akhir	

Data tengah yang baru didapat dengan rumus  $(5 + 9) / 2 = 7$ . Berarti data tengah yang baru adalah data ke-7, yaitu 23. Data yang dicari yaitu 17 dibandingkan dengan data tengah ini. Karena  $17 < 23$ , berarti proses dilanjutkan tetapi kali ini posisi akhir dianggap sama dengan posisi tengah  $-1$  atau 6.

3	9	11	12	15	17	20	23	31	35
awal=tengah					akhir				

Data tengah yang baru didapat dengan rumus  $(5 + 6) / 2 = 5$ . Berarti data tengah

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>4</b>
	<b>SEARCHING (METODE PENCARIAN)</b>	

yang baru adalah data ke-5, yaitu 17. data yang dicari dibandingkan dengan data tengah ini dan ternyata sama. Jadi data ditemukan pada indeks ke-5.

Pencarian biner ini akan berakhir jika data ditemukan atau posisi awal lebih besar daripada posisi akhir. Jika posisi sudah lebih besar daripada posisi akhir berarti data tidak ditemukan.

Untuk lebih jelasnya perhatikan contoh pencarian data 16 pada data diatas. Prosesnya hampir sama dengan pencarian data 17. Tetapi setelah posisi awal 5 dan posisi akhir 6, data tidak ditemukan dan  $16 < 17$ , maka posisi akhir menjadi posisi tengah - 1 atau = 4 sedangkan posisi awal = 5.

3	9	11	12	15	17	20	23	31	35
				akhir	awal				



Disini dapat dilihat bahwa posisi awal lebih besar daripada posisi akhir, yang artinya data tidak ditemukan.

Algoritma pencarian biner dapat dituliskan sebagai berikut :

- L = 0
- R = N - 1
- ketemu = false
- Selama (L <= R) dan (tidak ketemu) kerjakan baris 5 sampai dengan 8
- m = (L + R) / 2
- Jika (Data[m] = x) maka ketemu = true
- Jika (x < Data[m]) maka R = m - 1
- Jika (x > Data[m]) maka L = m + 1
- Jika (ketemu) maka m adalah indeks dari data yang dicari, jika tidak data tidak ditemukan

Di bawah ini merupakan fungsi untuk mencari data menggunakan pencarian biner.

```
int BinarySearch(int x)
```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>4</b>
	<b>SEARCHING (METODE PENCARIAN)</b>	

```

{
    int L = 0, R = Max-1, m;
    bool ketemu = false;

    while((L <= R) && (!ketemu))
    {
        m = (L + R) / 2;
        if(Data[m] == x)
            ketemu = true;
        else if (x < data[m])
            R = m - 1;
        else
            L = m + 1;
    }
    if(ketemu)
        return m;
    else
        return -1;
}

```

Fungsi diatas akan mengembalikan indeks dari data yang dicari. Apabila data tidak ditemukan maka fungsi diatas akan mengembalikan nilai -1.

Jumlah perbandingan minimum pada pencarian biner adalah 1 kali, yaitu apabila data yang dicari tepat berada di tengah-tengah. Jumlah perbandingan maksimum yang dilakukan dengan pencarian biner dapat dicari menggunakan rumus logaritma, yaitu :



$$C = {}^2\log(N)$$

## 5. Bahan/Peralatan yang digunakan

- Komputer
- Aplikasi CodeBlocks

## 6. Langkah Kerja

- Jalankan program CodeBlocks.
- Buatlah beberapa program berikut di komputer anda. Compile dan jalankan

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>4</b>
	<b>SEARCHING (METODE PENCARIAN)</b>	

(Run). Perhatikan hasil keluarannya. Jalankan program dan analisa hasilnya.

a) Program sequential search tanpa penanda (flag)

```

/* Nama File : searching1.c
   Pembuat   : tuliskan nama dan no.bp anda
   Tgl pembuatan : tuliskan tanggal hari ini
   Deskripsi  : isi array manual, cari nilai*/

#include "stdio.h"

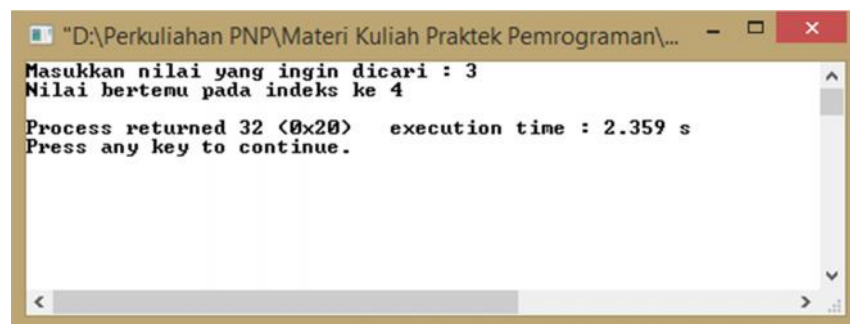
void main()
{
    int N[5]={8,6,5,4,3};
    int i,x;

    printf("Masukkan nilai yang ingin dicari : ");
    scanf("%d",&x);
    i=0;
    while (i<=5 && N[i]!=x)
    {
        i=i+1;
    }

    if (N[i]==x)
        printf("Nilai bertemu pada indeks ke %d \n",i);
    else
        printf("Nilai tidak ditemukan");
}

```

**Output Program**





b) Program sequential search dengan penanda (flag)

```

/* Nama File : searching2.c
   Pembuat   : tuliskan nama dan no.bp anda
   Tgl pembuatan : tuliskan tanggal hari ini
   Deskripsi  : isi array manual, cari nilai*/

```



	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>4</b>
	<b>SEARCHING (METODE PENCARIAN)</b>	

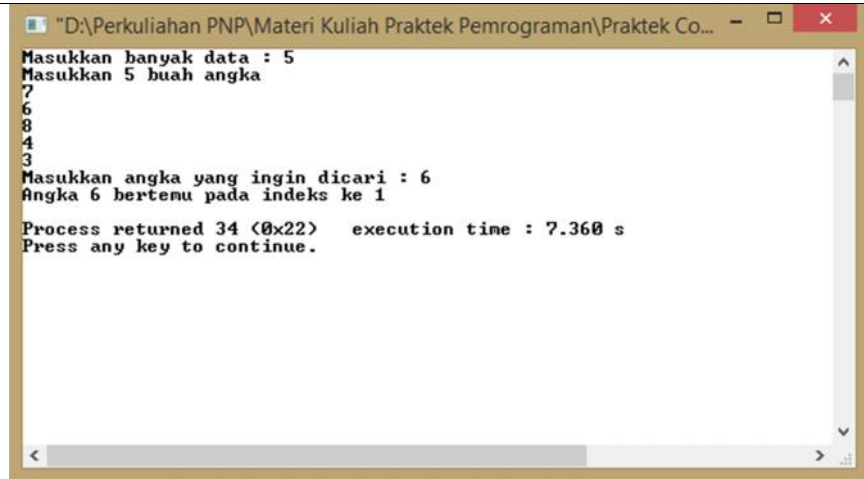
```
#include "stdio.h"

void main()
{
    int N[100];
    int i,x,idx,num_data,flag;
    printf("Masukkan banyak data : "); scanf("%d",&num_data);
    printf("Masukkan %d buah angka \n", num_data);
    for(i=0;i<num_data;i++)
    {
        scanf("%d",&N[i]);
    }
    printf("Masukkan angka yang ingin dicari : ");
    scanf("%d",&x);
    flag=0;
    for(i=0;i<num_data;i++)
    {
        if(N[i]==x)
        {
            flag=1;
            idx=i;
        }
    }

    if (flag==1)
        printf("Angka %d bertemu pada indeks ke %d \n",x,idx);
    else
        printf("Angka tidak ditemukan");
}
```

**Output Program**

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>4</b>
	<b>SEARCHING (METODE PENCARIAN)</b>	



c) Program sequential search untuk lebih dari 1 angka yang ditemukan

```

/* Nama File : search_n.c
Pembuat : tuliskan nama dan no.bp anda
Tgl pembuatan : tuliskan tanggal hari ini
Deskripsi : searching dengan banyak nilai yang sama*/

#include "stdio.h"



void main()
{
    int n[100],i,x,num,idx,jum;

    printf("Masukkan banyak data yang akan di-isi : ");
    scanf("%d",&num);

    for(i=0;i<num;i++)
    {
        printf("Data ke-%i = ",i+1); scanf("%i",&n[i]);
    }
    printf("Masukkan data yang akan dicari : ");
    scanf("%d",&x);
    jum=0;

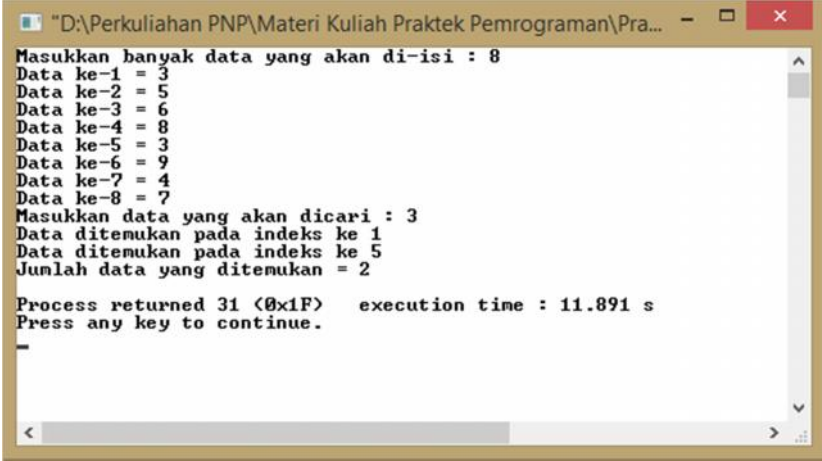
    for(i=0;i<num;i++)
    {
        if(n[i]==x)
        {
            printf("Data ditemukan pada indeks ke %i\n",i+1);
            jum=jum+1;
        }
    }
}

```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>4</b>
	<b>SEARCHING (METODE PENCARIAN)</b>	

```
printf("Jumlah data yang ditemukan = %d\n",jum);
if (jum==0)
    printf("Data yang dicari tidak ditemukan!!");
}
```

### Output Program



### d) Program pencarian nilai dengan konsep binary search

```
/* Nama File : binary_search.c
Pembuat : tuliskan nama dan no.bp anda
Tgl pembuatan : tuliskan tanggal hari ini
Deskripsi : melakukan pencarian menggunakan algoritma
binary*/

#include "stdio.h"



void main()
{
    int nilai[100];
    int i,x,n,found,ix;
    int atas, bawah, tengah;
    printf("\n---Binary Search---\n");
    printf("\n---Data Harus Terurut Dari Kecil Ke Besar---\n");

    printf("Masukkan jumlah data = ");scanf("%d",&n);

    for (i=0;i<=n-1;i++)
        {printf("Nilai[%d]= ",i); scanf("%d",&nilai[i]); }
    printf("\n\n");
    printf("Masukkan nilai yang dicari = ");scanf("%d",&x);

    atas = 0; bawah=n-1;
    found=0; i=0;
```



	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>4</b>
	<b>SEARCHING (METODE PENCARIAN)</b>	

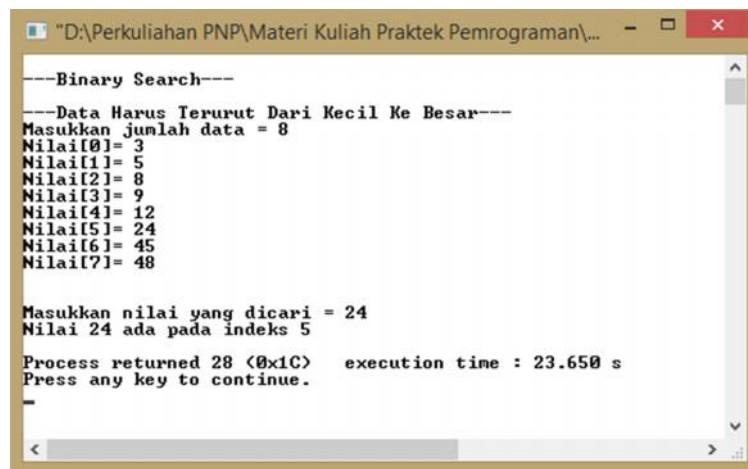
```

while ((atas<=bawah) && (found==0))
{
    tengah=(atas+bawah)/2;
    if(x==nilai[tengah])
    {found=1;ix=tengah;}
    else if(x<nilai[tengah])
        bawah=tengah-1;
    else //x>nilai[tengah]
        atas=tengah+1;
    i++;
}

if(found==1)
    printf("Nilai %d ada pada indeks %d \n",x,ix);
else
    printf("Data tidak ada\n");
}

```

### Output Program



```

---Binary Search---
---Data Harus Terurut Dari Kecil Ke Besar---
Masukkan jumlah data = 8
Nilai[0]= 3
Nilai[1]= 5
Nilai[2]= 8
Nilai[3]= 9
Nilai[4]= 12
Nilai[5]= 24
Nilai[6]= 45
Nilai[7]= 48



Masukkan nilai yang dicari = 24
Nilai 24 ada pada indeks 5

Process returned 28 (0x1C) execution time : 23.650 s
Press any key to continue.

```

## 7. Latihan

- Buatlah program binary search dengan ukuran array dinamis dan isi random. Lakukan pengurutan data array sebelumnya dari nilai terbesar ke nilai terkecil.
- Buatlah **program yang mencari banyaknya dan posisi data yang diinginkan** dari sebuah array dinamis yang isinya random

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>4</b>
	<b>SEARCHING (METODE PENCARIAN)</b>	

## 8. Keselamatan dan Kesehatan Kerja (K3)

- Periksalah komputer yang anda gunakan, apakah terjadi kerusakan atau tidak. Jika terjadi kerusakan, segera melapor pada Dosen ataupun Teknisi Labor.
- Jika akan melakukan peng-copy-an data, pastikan komputer ataupun alat penyimpanan data Anda terbebas dari virus.
- Selalu matikan komputer setiap Anda selesai praktikum.
- Jangan meninggalkan sampah apapun di ruangan labor komputer.

## 9. Hasil Pengujian dan Pembahasan



Selesaikanlah soal latihan dan kemudian cocokan hasil output program Anda dengan hasil program berikut ini. Jalankan juga program Anda untuk nilai-nilai lainnya.

- Program binary search dengan ukuran array dinamis dan isi random

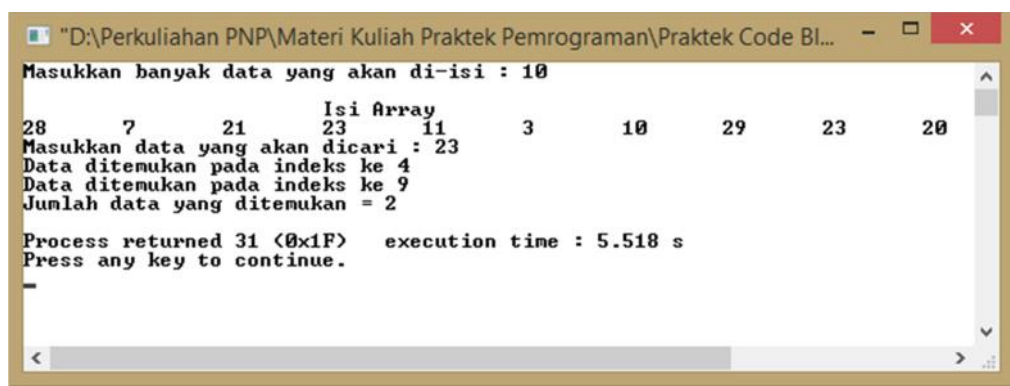
```

"D:\Perkuliahan PNP\Materi Kuliah Praktek Pemrograman\Praktek Code Block...
Banyaknya angka = 10
25 4 26 5 20 2 20 24 11 31
Setelah terurut
31 26 25 24 20 20 11 5 4 2
Masukkan angka yang dicari = 24
ketemu di indeks 3
Process returned 18 (0x12) execution time : 7.016 s
Press any key to continue.

```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>4</b>
	<b>SEARCHING (METODE PENCARIAN)</b>	

- b. Program yang mencari banyaknya dan posisi data yang diinginkan



```

"D:\Perkuliahan PNP\Materi Kuliah Praktek Pemrograman\Praktek Code Bl...
Masukkan banyak data yang akan di-isi : 10
Isi Array
28      7      21      23      11      3      10      29      23      20
Masukkan data yang akan dicari : 23
Data ditemukan pada indeks ke 4
Data ditemukan pada indeks ke 9
Jumlah data yang ditemukan = 2

Process returned 31 (0x1F)   execution time : 5.518 s
Press any key to continue.

```



## 10. Kesimpulan

Setelah menyelesaikan topik ini, mahasiswa mampu:

- Memahami konsep dan kegunaan fungsi bertipe data (dengan nilai kembalian)
- Mendefinisikan dan mendeklarasikan fungsi bertipe data (dengan nilai kembalian) serta melakukan pemanggilannya.
- Membuat program yang memproses fungsi bertipe data (dengan nilai kembalian)

## 11. Lembar Penilaian

Item Penilaian	Bobot	Nilai	Nilai X Bobot
Persiapan	5%		
Proses, Contoh, Cara melakukan pekerjaan, sistematika langkah kerja, kemandirian, partisipasi,	25%		
Sikap dan keselamatan kerja,	10%		

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>4</b>
	<b>SEARCHING (METODE PENCARIAN)</b>	

Contoh sikap duduk, sikap menggunakan alat kerja			
Hasil/Data/Produk; Program, Logika program, Kerapian Program, Tampilan Output	50%		
Waktu	10%		
Total	100%		



**12. Daftar Pustaka**

Hartono, J.2001. *Konsep Dasar Bahasa Pemrograman C*. Andi Offset Yogyakarta

Kelley, Al and Pohl, Ira. 2003. *C by Dissection: The Essentials of C Programming*. Addison-Weasley.

Khannedy, Eko Kurniawan. 2007. Diktat Pemrograman C. Unikom. Bandung.

Liem, Inggriani. 2003. Catatan Singkat Bahasa C. Jurusan Teknik Informatika. Institut Teknologi Bandung.

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>5</b>
	<b>ARRAY DUA DIMENSI</b>	

### 1. Judul : Array Dua Dimensi

### 2. Jumlah Jam/Pertemuan : 3 jam x 2 pertemuan = 6 jam praktek

### 3. Tujuan Praktek

- Mahasiswa mampu menjelaskan konsep array dua dimensi dan penggunaannya.
- Mahasiswa mampu mendefinisikan, mendeklarasikan, mengisi dan menuliskan nilai array dua dimensi menggunakan bahasa C.
- Mahasiswa mampu membuat program yang memproses nilai array dua dimensi

### 4. Teori Dasar.

Array dua dimensi adalah array yang memiliki dua buah elemen, yaitu baris dan kolom, sering juga disebut dengan bentuk matriks. Untuk mendeklarasikan array dua dimensi, digunakan bentuk umum sebagai berikut :



```
Tipedata nama_array [banyakbaris][banyakkolom];
```

Sebagai contoh apabila akan mendeklarasikan sebuah matrik berordo 5 x 5 dengan nama\_array adalah M dan masing-masing elemen matrik bertipe bilangan bulat/integer, maka dideklarasikan sebagai berikut :

```
int M[5][5];
```

Perintah di atas akan membuat sebuah array 2 dimensi yang kalau digambarkan adalah sebagai berikut :

index	0	1	2	3	4
0	0,0	0,1	0,2	0,3	0,4
1	1,0	1,1	1,2	1,3	1,4
2	2,0	2,1	2,2	2,3	2,4
3	3,0	3,1	3,2	3,3	3,4
4	4,0	4,1	4,2	4,3	4,4

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>5</b>
	<b>ARRAY DUA DIMENSI</b>	

### Gambar 5.1 Pengaksesan Elemen Array Dua Dimensi

Cara pengaksesan elemen array 2 dimensi dapat dilihat pada contoh di bawah ini:

```
mat[0][0]=7;
printf("Masukan data : ");scanf("%d",&mat[2][1]);
printf("Data yang dimasukan : %d\n",mat[2][1]);
```

Keterangan :

- ✓ Baris pertama adalah mengisi nilai 7 ke array mat pada baris 0 kolom 0.
- ✓ Baris kedua adalah perintah untuk membaca data elemen matrik pada baris 2 kolom ke 1.
- ✓ Baris ketiga adalah perintah untuk menampilkan data elemen matrik/array pada baris 2 dan kolom ke-1.

Pendeklarasian array dua dimensi juga bisa sekaligus dengan memberikan nilai-nilai elemennya. Misalkan akan dideklarasikan sebuah array dua dimensi bernama M dengan ukuran baris 2 dan kolom 3, dengan isi array / matriks sebagai berikut :

$$M = \begin{matrix} 2 & 1 & 3 \\ 5 & 6 & 7 \end{matrix}$$

Maka penulisan deklarasi array sebagai berikut :

```
int M[2][3]={2,1,3,5,6,7};
```

atau



```
int M[2][3]={ {2,1,3}, {5,6,7} };
```

#### 5. Bahan/Peralatan yang digunakan

- a. Komputer
- b. Aplikasi CodeBlocks

#### 6. Langkah Kerja

- 1) Jalankan program CodeBlocks.
- 2) Buatlah beberapa program berikut di komputer anda. Compile dan jalankan (Run). Perhatikan hasil keluarannya. Jalankan program dan analisa hasilnya.
  - a) Program untuk mengisi dan menampilkan array 2 dimensi, dimana ukuran da

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>5</b>
	<b>ARRAY DUA DIMENSI</b>	

ni array ditentukan oleh user

Cara 1 :

```
/* Nama File : matriks1.c
   Pembuat   : tuliskan nama dan no.bp anda
   Tgl pembuatan : tuliskan tanggal hari ini
   Deskripsi  : menuliskan isi matriks */

#include "stdio.h"

int main()
{
    int A[3][3];
    int i,j;//i=indeks baris, j=indeks kolom
    A[0][0] = 3; A[0][1] = 7; A[0][2] = 5;
    A[1][0] = 4; A[1][1] = 2; A[1][2] = 1;
    A[2][0] = 8; A[2][1] = 3; A[2][2] = 1;
    printf("Isi Matriks A = \n");
    for (i=0;i<=2;i++)
    {
        for(j=0;j<=2;j++)
            printf("%d    ",A[i][j]);
        printf("\n");
    }
    return 0;
}
```



Cara 2 :

```
/* Nama File : matriks1.c
   Pembuat   : tuliskan nama dan no.bp anda
   Tgl pembuatan : tuliskan tanggal hari ini
   Deskripsi  : menuliskan isi matriks */

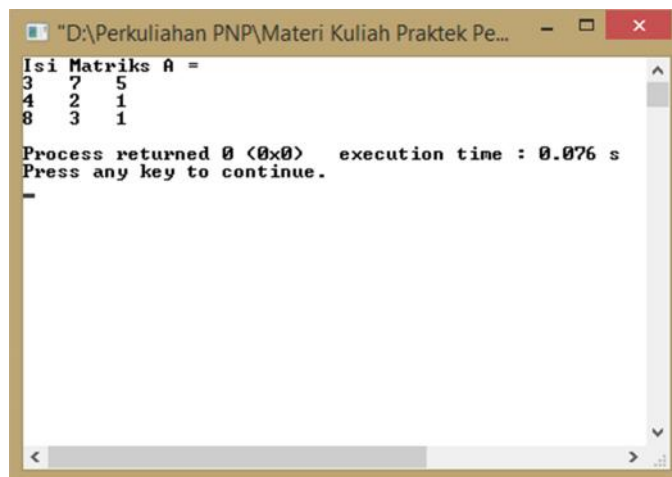
#include "stdio.h"

int main()
{
    int A[3][3] = {3,7,5,4,8,1,8,3,1};
    int i,j;//i=indeks baris, j=indeks kolom

    printf("Isi Matriks A = \n");
    for (i=0;i<=2;i++)
    {
        for(j=0;j<=2;j++)
            printf("%d    ",A[i][j]);
        printf("\n");
    }
    return 0;
}
```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>5</b>
	<b>ARRAY DUA DIMENSI</b>	

### Output Program



- b) Program untuk menjumlahkan isi elemen matriks setiap barisnya, untuk matriks ukuran 3 x 3 dengan isi diinputkan user.

```

/* Nama File : matriks2.c
   Pembuat   : tuliskan nama dan no.bp anda
   Tgl pembuatan : tuliskan tanggal hari ini
   Deskripsi  : menjumlahkan isi matriks tiap baris*/



#include "stdio.h"

int main()
{
    int A[3][3], i, j, jum;

    printf("Masukkan angka untuk matriks ukuran 3 x 3 \n");
    for (i=0; i<3; i++)
    {
        for(j=0; j<3; j++)
        {
            printf("Matriks baris%i kolom %i = ", i+1, j+1);
            scanf("%d", &A[i][j]);
        }
    }
}

```



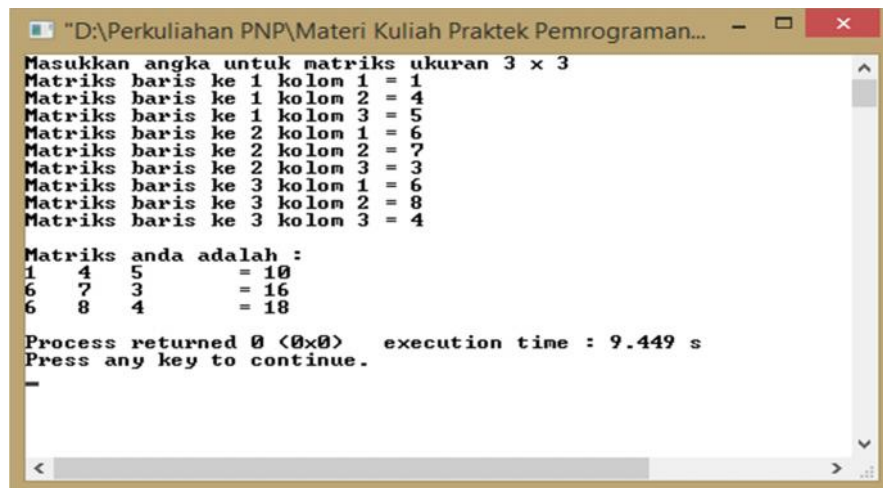
	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>5</b>
	<b>ARRAY DUA DIMENSI</b>	

```

printf("\nMatriks anda adalah : \n");
for (i=0;i<=2;i++)
{
    jum=0;
    for(j=0;j<=2;j++)
    {
        jum=jum+A[i][j];
        printf("%d    ",A[i][j]);
    }
    printf("\t= %d",jum);
    printf("\n");
}
return 0;
}

```

### Output Program



```

Masukkan angka untuk matriks ukuran 3 x 3
Matriks baris ke 1 kolom 1 = 1
Matriks baris ke 1 kolom 2 = 4
Matriks baris ke 1 kolom 3 = 5
Matriks baris ke 2 kolom 1 = 6
Matriks baris ke 2 kolom 2 = 7
Matriks baris ke 2 kolom 3 = 3
Matriks baris ke 3 kolom 1 = 6
Matriks baris ke 3 kolom 2 = 8
Matriks baris ke 3 kolom 3 = 4

Matriks anda adalah :
1    4    5    = 10
6    7    3    = 16
6    8    4    = 18

Process returned 0 (0x0)   execution time : 9.449 s
Press any key to continue.

```



- c) Program yang menguji apakah sebuah matriks yang menampilkan matriks identitas dengan jumlah baris dan kolom diinputkan user. Matriks identitas adalah matriks dengan ukuran baris dan kolom sama, dengan elemen utama bernilai 1 dan elemen lainnya bernilai 0.

```

/* Nama File : matriks3.c
   Pembuat   : tuliskan nama dan no.bp anda
   Tgl pembuatan : tuliskan tanggal hari ini
   Deskripsi  : menjumlahkan isi matriks tiap baris*/

#include "stdio.h"
#define maks 100

```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>5</b>
	<b>ARRAY DUA DIMENSI</b>	

```


int main()
{
    int A[maks][maks],brs,kol;
    int i,j,sama,jum;

    printf("====Matriks A====\n");
    printf("Masukkan ukuran baris matriks A : ");
    scanf("%d",&brs);
    printf("Masukkan ukuran kolom matriks A : ");
    scanf("%d",&kol);

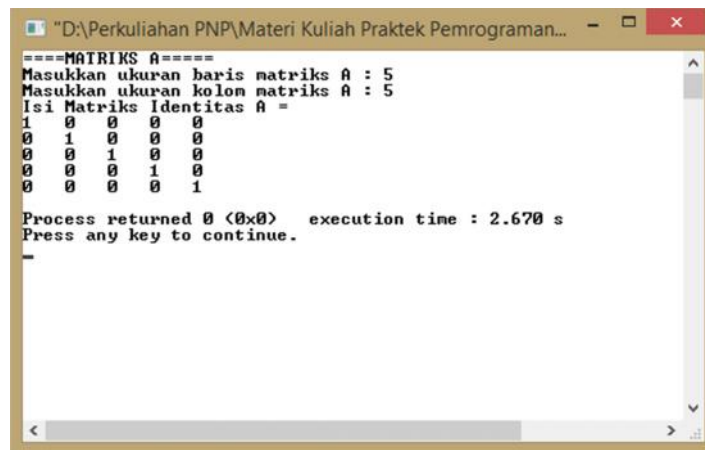
    if(brs==kol)
    {
        for (i=0;i<brs;i++)
        {
            for(j=0;j<kol;j++)
            {
                if(i==j)
                    A[i][j]=1;
                else
                    A[i][j]=0;
            }
        }
        printf("Isi Matriks Identitas A = \n");
        for (i=0;i<brs;i++)
        {
            for(j=0;j<kol;j++)
            {
                printf("%d  ",A[i][j]);
            }
            printf("\n");
        }
    }
    else
        printf("Bukan identitas karena brs dan klm beda\n");

    return 0;
}

```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>5</b>
	<b>ARRAY DUA DIMENSI</b>	

### Output Program



```

====Matriks A====
Masukkan ukuran baris matriks A : 5
Masukkan ukuran kolom matriks A : 5
Isi Matriks Identitas A =
1 0 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 1

Process returned 0 (0x0) execution time : 2.670 s
Press any key to continue.

```

- d) Program yang menguji apakah dua buah matriks yang diinputkan user identik satu sama lain atau bukan. Matriks dikatakan identik/ sama apabila matriks yang dibandingkan memiliki ukuran maupun isi yang sama.

```


/* Nama File : matriks4.c
   Pembuat   : tuliskan nama dan no.bp anda
   Tgl pembuatan : tuliskan tanggal hari ini
   Deskripsi  : menjumlahkan isi matriks tiap baris*/

#include "stdio.h"
#define maks 100



int main()
{
    int A[maks][maks],brsA,kolA;
    int B[maks][maks],brsB,kolB;
    int i,j,sama,jum;

    printf("====Matriks A====\n");
    printf("Masukkan ukuran baris matriks A : ");
    scanf("%d",&brsA);
    printf("Masukkan ukuran kolom matriks A : ");
    scanf("%d",&kolA);
    printf("Masukkan isi matriks A ukuran %dx%d \n",brsA,kolA);
    for (i=0;i<brsA;i++)
    {
        for(j=0;j<kolA;j++)
        {

```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>5</b>
	<b>ARRAY DUA DIMENSI</b>	

<pre>         printf("A[%d][%d] = ",i,j);scanf("%d",&amp;A[i][j]);     } }  printf("====Matriks B====\n"); printf("Masukkan ukuran baris matriks B : "); scanf("%d",&amp;brsB); printf("Masukkan ukuran kolom matriks B : "); scanf("%d",&amp;kolB); printf("Masukkan isi matriks B ukuran %dx%d \n",brsB,kolB); for (i=0;i&lt;brsB;i++) {     for(j=0;j&lt;kolB;j++)     {         printf("B[%d][%d] = ",i,j);scanf("%d",&amp;B[i][j]);     } }  printf("\nMatriks A anda adalah : \n"); for (i=0;i&lt;brsA;i++) {     for(j=0;j&lt;kolA;j++)     {         printf("%d    ",A[i][j]);     }     printf("\n"); }  printf("\nMatriks B anda adalah : \n"); for (i=0;i&lt;brsB;i++) {     for(j=0;j&lt;kolB;j++)     {         printf("%d    ",B[i][j]);     }     printf("\n"); }  //periksa kesamaan matriks sama=0; if((brsA==brsB)&amp;&amp;(kolA==kolB)) {     jum=brsA*kolA;     for (i=0;i&lt;brsA;i++)     {         for(j=0;j&lt;kolA;j++)         {             if (A[i][j]==B[i][j])                 sama=sama + 1;         }     } } </pre>
---

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>5</b>
	<b>ARRAY DUA DIMENSI</b>	

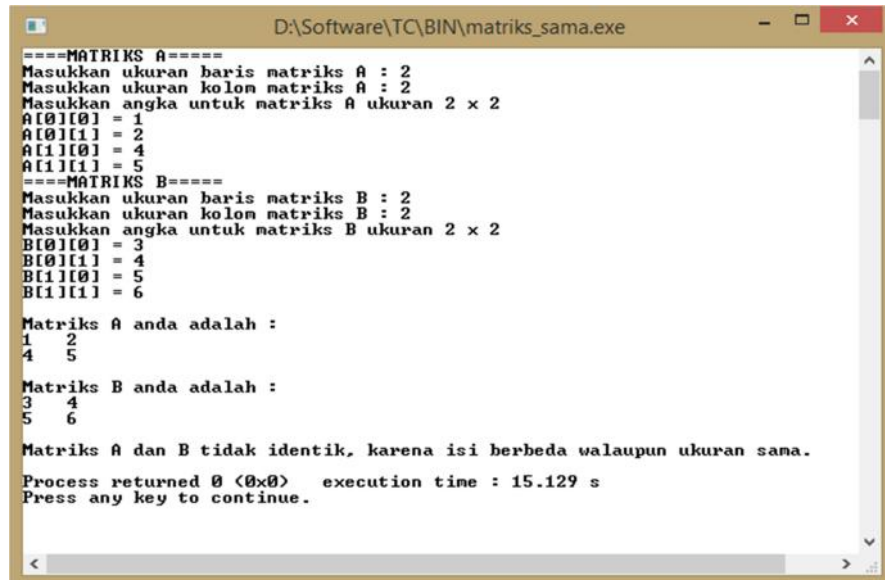
```

if (sama==jum)
    printf("\nMatriks A dan B identik. \n");
else
    printf("\nMatriks A dan B tidak identik, isi berbeda.\n");
}
else
    printf("\nMatriks A dan B tidak identik, ukuran berbeda.\n");
;

return 0;
}

```

### Output Program



```

D:\Software\TC\BIN\matriks_sama.exe
====MATRIKS A====
Masukkan ukuran baris matriks A : 2
Masukkan ukuran kolom matriks A : 2
Masukkan angka untuk matriks A ukuran 2 x 2
A[0][0] = 1
A[0][1] = 2
A[1][0] = 4
A[1][1] = 5
====MATRIKS B====
Masukkan ukuran baris matriks B : 2
Masukkan ukuran kolom matriks B : 2
Masukkan angka untuk matriks B ukuran 2 x 2
B[0][0] = 3
B[0][1] = 4
B[1][0] = 5
B[1][1] = 6

Matriks A anda adalah :
1 2
4 5



Matriks B anda adalah :
3 4
5 6

Matriks A dan B tidak identik, karena isi berbeda walaupun ukuran sama.
Process returned 0 (0x0)   execution time : 15.129 s
Press any key to continue.

```

## 7. Latihan

- Terdapat sebuah array dua dimensi dengan ukuran 3 x 4, dimana elemen masing-masing array ditentukan oleh user. Buatlah sebuah program yang **mengganti isi elemen array, kalau genap diubah menjadi 0 dan ganjil diubah menjadi 1**.
- Terdapat dua buah array dua dimensi (matriks) dimana ukuran dan isi masing-masing array ditentukan oleh user. Buatlah program yang **menjumlahkan kedua matriks**. Hal yang perlu diingat adalah proses penjumlahan hanya berlaku apabila kedua matriks memiliki ukuran yang sama.

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>5</b>
	<b>ARRAY DUA DIMENSI</b>	

- c. Buatlah program yang **menampilkan Segitiga Pascal** dengan jumlah baris segitiga ditentukan oleh user.
- d. Terdapat sebuah matriks dengan ukuran dan nilai ditentukan oleh user. Buatlah program yang **menampilkan matriks transpose** dari matriks tersebut.
- e. Terdapat dua buah array dua dimensi (matriks) dimana ukuran dan isi masing-masing array ditentukan oleh user. Buatlah program yang **menampilkan hasil perkalian kedua matriks**. Ingat aturan perkalian matriks



#### **8. Keselamatan dan Kesehatan Kerja (K3)**

- a. Periksa komputer yang anda gunakan, apakah terjadi kerusakan atau tidak. Jika terjadi kerusakan, segera melapor pada Dosen ataupun Teknisi Labor.
- b. Jika akan melakukan peng-copy-an data, pastikan komputer ataupun alat penyimpanan data Anda terbebas dari virus.
- c. Selalu matikan komputer setiap Anda selesai praktikum.
- d. Jangan meninggalkan sampah apapun di ruangan labor komputer.

#### **9. Hasil Pengujian dan Pembahasan**

Selesaikanlah soal latihan dan kemudian cocokan hasil output program Anda dengan hasil program berikut ini. Jalankan juga program Anda untuk nilai-nilai lainnya.

- a. Mengganti isi elemen array, genap diubah 0 dan ganjil diubah 1

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>5</b>
	<b>ARRAY DUA DIMENSI</b>	

```

"D:\Perkuliahan PNP\Materi Kuliah Praktek Pemrograman\Praktek...
Masukkan angka untuk matriks ukuran 3 x 4
A[1][1]= 2
A[1][2]= 3
A[1][3]= 4
A[1][4]= 5
A[2][1]= 6
A[2][2]= 7
A[2][3]= 8
A[2][4]= 9
A[3][1]= 1
A[3][2]= 2
A[3][3]= 3
A[3][4]= 4

Matriks anda adalah :
2 3 4 5
6 7 8 9
1 2 3 4

Matriks setelah pertukaran elemen :
0 1 0 1
0 1 0 1
1 0 1 0

Process returned 0 (0x0)   execution time : 8.980 s
Press any key to continue.

```

b. Menjumlahkan dua buah matriks

```

"D:\Perkuliahan PNP\Materi Kuliah Praktek Pemrograman\Praktek Code B...
====Matriks A====
Masukkan ukuran baris matriks A : 2
Masukkan ukuran kolom matriks A : 2
Masukkan angka untuk matriks A ukuran 2 x 2
A[0][0] = 1
A[0][1] = 2
A[1][0] = 3
A[1][1] = 4
====Matriks B====
Masukkan ukuran baris matriks B : 2
Masukkan ukuran kolom matriks B : 2
Masukkan angka untuk matriks B ukuran 2 x 2
B[0][0] = 5
B[0][1] = 6
B[1][0] = 7
B[1][1] = 8

Matriks A anda adalah :
1 2
3 4



Matriks B anda adalah :
5 6
7 8

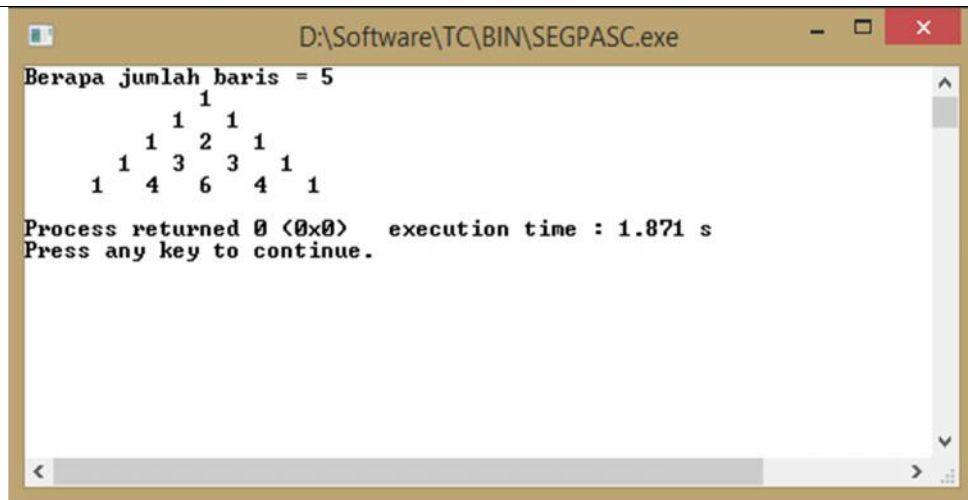
Matriks hasil penjumlahan anda adalah :
6 8
10 12

Process returned 0 (0x0)   execution time : 10.340 s
Press any key to continue.

```

c. Menampilkan segitiga Pascal

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>5</b>
	<b>ARRAY DUA DIMENSI</b>	



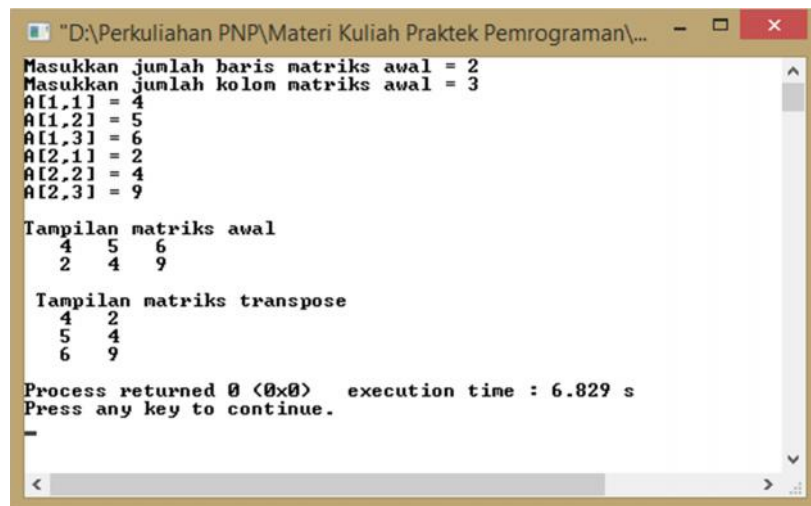
```

D:\Software\TC\BIN\SEGPASC.exe
Berapa jumlah baris = 5
      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1

Process returned 0 (0x0)   execution time : 1.871 s
Press any key to continue.

```

d. Menampilkan matriks transpose



```

"D:\Perkuliahan PNP\Materi Kuliah Praktek Pemrograman\...
Masukkan jumlah baris matriks awal = 2
Masukkan jumlah kolom matriks awal = 3
A[1,1] = 4
A[1,2] = 5
A[1,3] = 6
A[2,1] = 2
A[2,2] = 4
A[2,3] = 9

Tampilan matriks awal
4 5 6
2 4 9



Tampilan matriks transpose
4 2
5 4
6 9

Process returned 0 (0x0)   execution time : 6.829 s
Press any key to continue.

```

e. Menampilkan hasil perkalian dua buah matriks



	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>5</b>
	<b>ARRAY DUA DIMENSI</b>	

```

D:\Perkuliahan PNP\Materi Kuliah Praktek Pemrograman\...
Baris matriks pertama = 2
Kolom matriks pertama = 3
Baris matriks kedua = 3
Kolom matriks kedua = 2
A[1,1] = 1
A[1,2] = 2
A[1,3] = 4
A[2,1] = 5
A[2,2] = 6
A[2,3] = 3
B[1,1] = 4
B[1,2] = 2
B[2,1] = 3
B[2,2] = 5
B[3,1] = 4
B[3,2] = 3
Matriks pertama =
  1  2  4
  5  6  3
Matriks kedua =
  4  2
  3  5
  4  3
Hasil perkalian matriks =
  26  24
  50  49
Process returned 0 (0x0)   execution time : 16.964 s
Press any key to continue.

```



## 10. Kesimpulan

Setelah menyelesaikan topik ini, mahasiswa mampu:

- Memahami konsep array dua dimensi dan penggunaannya.
- Mengisi dan menuliskan nilai array satu dua dimensi menggunakan bahasa C.
- Membuat program yang memproses isi array dua dimensi dalam bahasa C.

## 11. Lembar Penilaian

Item Penilaian	Bobot	Nilai	Nilai X Bobot
Persiapan	5%		
Proses, Contoh,Cara melakukan pekerjaan, sistimatika langkah kerja, kemandirian, partisipasi,	25%		
Sikap dan keselamatan kerja,	10%		

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>5</b>
	<b>ARRAY DUA DIMENSI</b>	



Contoh sikap duduk, sikap menggunakan alat kerja			
Hasil/Data/Produk; Program, Logika program, Kerapian Program, Tampilan Output	50%		
Waktu	10%		
Total	100%		

**12. Daftar Pustaka**

Kelley, Al and Pohl, Ira. 2003. *C by Dissection: The Essentials of C Programming*. Addison-Weasley.

Khannedy, Eko Kurniawan. 2007. Diktat Pemrograman C. Unikom. Bandung.

Liem, Inggriani. 2003. Catatan Singkat Bahasa C. Jurusan Teknik Informatika. Institut Teknologi Bandung.

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>6</b>
	<b>TIPE BENTUKAN</b>	

### 1. Judul : Tipe Bentuk

### 2. Jumlah Jam/Pertemuan : 3 jam x 2 pertemuan = 6 jam praktek



### 3. Tujuan Praktek

- Mahasiswa mampu mengenal dan menjelaskan konsep konsep tipe bentuk dalam Bahasa C
- Mahasiswa mampu mendefinisikan dan mendeklarasikan tipe bentuk
- Mahasiswa mampu membuat program yang memproses tipe bentuk

### 4. Teori Dasar.

Tipe data bentuk adalah suatu tipe data yang didefinisikan sendiri untuk memenuhi kebutuhan-kebutuhan program yang akan dibuat (*user-defined data type*). Tipe bentuk disusun oleh satu atau lebih tipe dasar. Ada dua macam tipe bentuk:

- Tipe dasar yang diberi nama dengan nama tipe baru  
Pemberian nama baru pada sebuah tipe dasar kadang-kadang dilakukan dengan berbagai alasan seperti agar lebih mudah dipahami oleh pembaca program atau untuk mempersingkat penulisan tipe dasar.
- Tipe terstruktur  
Tipe bentuk ini adalah tipe yang berbentuk rekaman (*record*). *Record* terdiri atas satu atau lebih *field*. Tiap *field* menyimpan data dari tipe dasar tertentu atau dari tipe bentuk lain yang susah didefinisikan sebelumnya. Nama *record* yang merupakan nama tipe bentuk ditentukan sendiri oleh pemrogram.

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>6</b>
	<b>TIPE BENTUKAN</b>	

Dalam jobsheet ini lebih berfokus pada tipe bentukan yang terstruktur. Pendefinisian tipe bentukan pada bahasa C dilakukan dengan 2 cara yaitu :

Cara 1 :

```
typedef struct
{
    tipe_data_dasar namafield_1;
    tipe_data_dasar namafield_2;
    ...
    tipe_data_dasar namafield_n;
}nama_tipe_bentukan;
```

Cara 2 :



```
struct nama_tipe_bentukan
{
    tipe_data_dasar namafield_1;
    tipe_data_dasar namafield_2;
    ...
    tipe_data_dasar namafield_n;
};
```

Tipe bentukan telah didefinisikan tersebut akan digunakan sebagai tipe data baru untuk nama variabel yaitu dengan format:

```
nama_tipe_bentukan namavariabel;
```

Untuk pendefinisian tipe bentukan terstruktur dengan cara ke 2, pendeklarasian variabelnya bisa digabungkan dalam satu deklarasi yang sama dengan cara :

```
struct nama_tipe_bentukan
{
    tipe_data_dasar namafield_1;
    tipe_data_dasar namafield_2;
    ...
    tipe_data_dasar namafield_n;
} namavariabel;
```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>6</b>
	<b>TIPE BENTUKAN</b>	

Contoh :

Pendefinisian tipe bentukan Titik yang terdiri dari nilai absis(x) dan ordinat(y)

```
//pendefinisian tipe bentukan Titik
typedef struct
{
    int x;
    int y;
}Titik;
//penggunaan tipe bentukan
Titik T1,T2;
```

atau

```
//pendefinisian tipe bentukan Titik sekaligus variabelnya
Struct Titik
{
    int x;
    int y;
}T1,T2;
```

## 5. Bahan/Peralatan yang digunakan

- Komputer
- Aplikasi CodeBlocks



## 6. Langkah Kerja

- Jalankan program CodeBlocks.
- Buatlah beberapa program berikut di komputer anda. Compile dan jalankan (Run). Perhatikan hasil keluarannya. Jalankan program dan analisa hasilnya.
  - Program yang meminta user menginputkan koordinat sebuah titik, kemudian menampilkan hasil pencerminan terhadap sumbu X maupun sumbu Y.

```
/* Nama File : titik1.c
Pembuat : tuliskan nama dan no.bp anda
Tgl pembuatan : tuliskan tanggal hari ini
Deskripsi : menuliskan isi matriks */

#include "stdio.h"

int main()
{
```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>6</b>
	<b>TIPE BENTUKAN</b>	

```
typedef struct{
    int x;
    int y;
} titik;

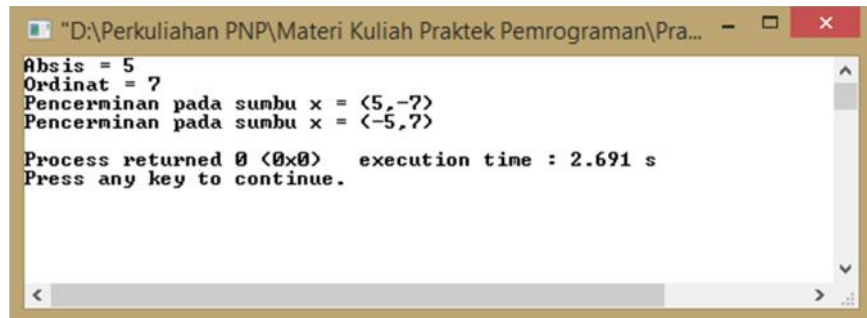
titik A;

printf("Absis = "); scanf("%i",&A.x);
printf("Ordinat = "); scanf("%i",&A.y);

printf("Pencerminan pada sumbu x = (%i,%i)\n",A.x,-A.y);
printf("Pencerminan pada sumbu x = (%i,%i)\n",-A.x,A.y);

return 0;
}
```

### Output Program





- b) Program yang menampilkan hasil penjumlahan dari dua buah pecahan, dimana pembilang dan penyebut masing-masing pecahan diinputkan oleh user.

```
/* Nama File : pecahan1.c
   Pembuat   : tuliskan nama dan no.bp anda
   Tgl pembuatan : tuliskan tanggal hari ini
   Deskripsi  : menuliskan isi matriks */

#include "stdio.h"

int main()
{
    typedef struct
    {
        int x;
        int y;
    } pecahan;
    pecahan p1,p2,p3;
    float hsl;
```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>6</b>
	<b>TIPE BENTUKAN</b>	

```

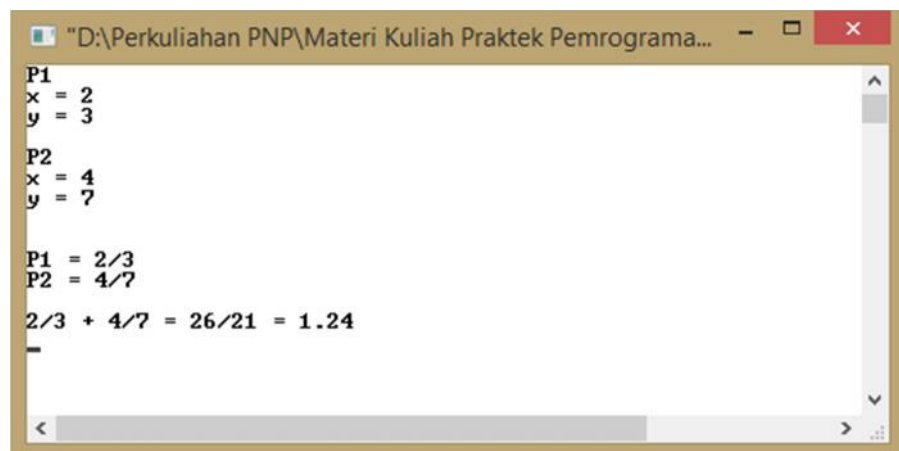
printf("P1\n");
printf("x = "); scanf("%d",&p1.x);
printf("y = "); scanf("%d",&p1.y);
printf("\nP2\n");
printf("x = "); scanf("%d",&p2.x);
printf("y = "); scanf("%d",&p2.y);

printf("\n\nP1 = %d/%d\n",p1.x,p1.y);
printf("P2 = %d/%d\n",p2.x,p2.y);
//penjumlahan
p3.x=(p1.x*p2.y)+(p1.y*p2.x);
p3.y=p1.y*p2.y;
hsl = (float) p3.x/p3.y;
printf("\n%d/%d + %d/%d = %d/%d = %0.2f\n",
p1.x,p1.y,p2.x,p2.y,p3.x,p3.y,hsl);

return 0;
}

```

### Output Program





```

D:\Perkuliahan PNP\Materi Kuliah Praktek Pemrograma...
P1
x = 2
y = 3
P2
x = 4
y = 7
P1 = 2/3
P2 = 4/7
2/3 + 4/7 = 26/21 = 1.24

```

## 7. Latihan

- Buatlah program yang **menampilkan jarak dan gradien (kemiringan)** dari 2 buah titik yang diinputkan user.
- Buatlah program yang **menampilkan lama perjalanan** dalam jam, menit dan detik, apabila waktu kedatangan dan keberangkatan diinputkan user
- Buatlah program yang **memproses hasil perkalian dan pembagian dua buah pecahan** yang diinputkan user.

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>6</b>
	<b>TIPE BENTUKAN</b>	

- d. Buatlah program yang **menampilkan tanggal sehari setelah tanggal yang diinputkan user**. Jangan lupa untuk melakukan pengecekan/ validasi terhadap tanggal yang diinputkan oleh user.

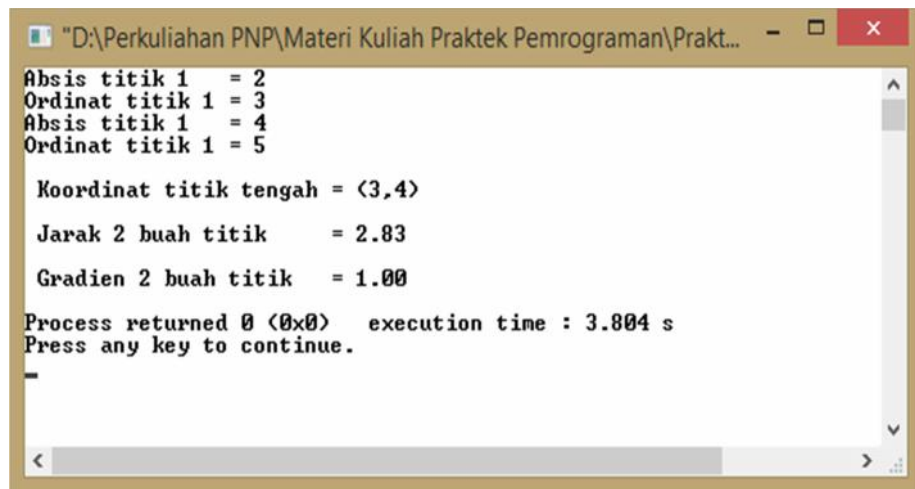
## 8. Keselamatan dan Kesehatan Kerja (K3)

- Periksalah komputer yang anda gunakan, apakah terjadi kerusakan atau tidak. Jika terjadi kerusakan, segera melapor pada Dosen ataupun Teknisi Labor.
- Jika akan melakukan peng-copy-an data, pastikan komputer ataupun alat penyimpanan data Anda terbebas dari virus.
- Selalu matikan komputer setiap Anda selesai praktikum.
- Jangan meninggalkan sampah apapun di ruangan labor komputer.

## 9. Hasil Pengujian dan Pembahasan

Selesaikanlah soal latihan dan kemudian cocokan hasil output program Anda dengan hasil program berikut ini. Jalankan juga program Anda untuk nilai-nilai lainnya.

- Menampilkan titik tengah, jarak dan gradien (kemiringan garis) dari dua buah titik



```

"D:\Perkuliahan PNP\Materi Kuliah Praktek Pemrograman\Prakt... - [X]
Absis titik 1 = 2
Ordinat titik 1 = 3
Absis titik 1 = 4
Ordinat titik 1 = 5



Koordinat titik tengah = <3,4>

Jarak 2 buah titik      = 2.83

Gradien 2 buah titik    = 1.00

Process returned 0 (0x0)   execution time : 3.804 s
Press any key to continue.
-
```



	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>6</b>
	<b>TIPE BENTUKAN</b>	

- b. Menampilkan lama perjalanan

```

"D:\Perkuliahan PNP\Materi Kuliah Praktek Pemrogra...
Waktu keberangkatan =
Jam = 8
Menit = 30
Detik = 0

Waktu kedatangan =
Jam = 10
Menit = 15
Detik = 30
Lama perjalanan = 1 45 30

Process returned 0 (0x0)   execution time : 11.826 s
Press any key to continue.

```

- c. Memproses hasil perkalian dan pembagian dua buah pecahan

```



"D:\Perkuliahan PNP\Materi Kuliah Prakte...
Pecahan 1
x = 1
y = 2

Pecahan 2
x = 1
y = 4

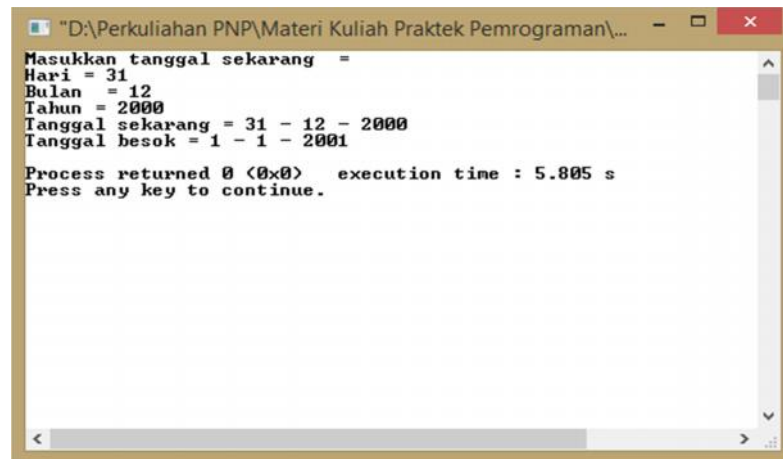
P1 = 1/2
P2 = 1/4

1/2 x 1/4 = 1/8
1/2 : 1/4 = 4/2

```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>6</b>
	<b>TIPE BENTUKAN</b>	

d. Menampilkan tanggal sehari setelah tanggal yang diinputkan oleh user



```

"D:\Perkuliahan PNP\Materi Kuliah Praktek Pemrograman\...
Masukkan tanggal sekarang =
Hari = 31
Bulan = 12
Tahun = 2000
Tanggal sekarang = 31 - 12 - 2000
Tanggal besok = 1 - 1 - 2001

Process returned 0 (0x0)   execution time : 5.805 s
Press any key to continue.

```



## 10. Kesimpulan

Setelah menyelesaikan topik ini, mahasiswa mampu:

- Memahami konsep tipe bentukan terstruktur
- Mendefinisikan dan mendeklarasikan tipe bentukan terstruktur
- Membuat program yang memproses tipe bentukan dalam bahasa C.

## 11. Lembar Penilaian

Item Penilaian	Bobot	Nilai	Nilai X Bobot
Persiapan	5%		
Proses, Contoh, Cara melakukan pekerjaan, sistimatika langkah kerja, kemandirian, partisipasi,	25%		
Sikap dan keselamatan kerja, Contoh sikap duduk, sikap menggunakan alat kerja	10%		
Hasil/Data/Produk; Program, Logika program, Kerapian Program, Tampilan Output	50%		
Waktu	10%		

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>6</b>
	<b>TIPE BENTUKAN</b>	



Total	100%		
-------	------	--	--

**12. Daftar Pustaka**

Kelley, Al and Pohl, Ira. 2003. *C by Dissection: The Essentials of C Programming*. Addison-Weasley.

Khannedy, Eko Kurniawan. 2007. Diktat Pemrograman C. Unikom. Bandung.

Liem, Inggriani. 2003. Catatan Singkat Bahasa C. Jurusan Teknik Informatika. Institut Teknologi Bandung.

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>7</b>
	<b>TIPE BENTUKAN DAN ARRAY</b>	

### 1. Judul : Tipe Bentuk dan Array

### 2. Jumlah Jam/Pertemuan : 3 jam x 1 pertemuan = 3 jam praktek

### 3. Tujuan Praktek



- Mahasiswa mampu mengenal dan menjelaskan konsep gabungan tipe bentuk dan array
- Mahasiswa mampu mendefinisikan dan mendeklarasikan tipe bentuk dengan elemen berupa array atau array dari data-data yang bertipe bentuk terstruktur
- Mahasiswa mampu membuat program yang memproses gabungan tipe bentuk maupun array.

### 4. Teori Dasar.

Dalam mendeklarasikan elemen-elemen sebuah tipe bentuk seperti pada Job 6 juga disertakan dengan deklarasi tipe data masing-masing elemen. Elemen-elemen pembentuk tipe terstruktur tersebut bisa saja berupa array, sehingga aturan deklarasi dan pengoperasian array juga berlaku. Berikut ini adalah proses pendeklarasian tipe bentuk dengan elemen berupa array.

```
typedef struct
{
    tipe_data_dasar namafieldarray_1[ukuran_array];
    tipe_data_dasar namafieldarray_2[ukuran_array];
    ...
    tipe_data_dasar namafield_n;
}nama_tipe_bentuk;

nama_tipe_bentuk nama_variabel;
```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>7</b>
	<b>TIPE BENTUKAN DAN ARRAY</b>	

Contoh :

```
typedef struct
{
    char NIM[10];
    char nama[25];
    int nilai[4];
}mahasiswa;

Mahasiswa A;
```

Contoh pengaksesan elemen nilai dari variabel A yang memiliki tipe bentukan terstruktur adalah :



```
A.nilai[0] = 90;
```

Elemen-elemen dari suatu larik/array dapat juga terdiri dari variabel dengan tipe bentukan terstruktur. Larik/ array struktur dimensi satu membentuk suatu tabel (mirip dengan array dimensi dua) dengan barisnya menunjukkan elemen dari lariknya dan dengan kolom yang menunjukkan elemen dari struktur bentukan. Tiap-tiap elemen dari struktur dapat diakses untuk tiap-tiap elemen array dengan menuliskan nama array diikuti dengan indeksinya, diikuti oleh operator titik dan nama dari elemen strukturnya.

Contoh deklarasi array dengan elemen tipe bentukan terstruktur sebagai berikut :

```
typedef struct
{
    tipe_data_dasar namafield_1;
    tipe_data_dasar namafield_2;
    ...
    tipe_data_dasar namafield_n;
}nama_tipe_bentukan;

nama_tipe_bentukan nama_array[ukuran_array];
```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>7</b>
	<b>TIPE BENTUKAN DAN ARRAY</b>	

Contoh deklarasi array dari tipe bentukan.

```
typedef struct
{
    char NIM[10];
    char nama[25];
    int nilai[5];
}mahasiswa;

mhs mahasiswa[10];
```

Dari contoh diatas, dideklarasikan terdapat sebuah array, dimana masing-masing elemen array merupakan tipe bentukan, dan beberapa elemen data dalam tipe bentukan juga berupa array.

Contoh pengaksesan :

```
mahasiswa[0].nilai[0] = 90;
```

## 5. Bahan/Peralatan yang digunakan



- Komputer
- Aplikasi CodeBlocks

## 6. Langkah Kerja

- Jalankan program CodeBlocks.
- Buatlah beberapa program berikut di komputer anda. Compile dan jalankan (Run). Perhatikan hasil keluarannya. Jalankan program dan analisa hasilnya.
  - Program yang meminta user menginputkan data seorang siswa yang terdiri dari NIS, nama dan nilai. Terdapat 3 buah nilai, cari total dan rata-rata nilai siswa tersebut.

```
/* Nama File : bentukan1.c
   Pembuat   : tuliskan nama dan no.bp anda
   Tgl pembuatan : tuliskan tanggal hari ini
   Deskripsi : menuliskan isi matriks */

#include "stdio.h"
int main()
{
```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>7</b>
	<b>TIPE BENTUKAN DAN ARRAY</b>	

```

typedef struct {
    char nis[10];
    char nama[25];
    int nilai[3];
}mahasiswa;

mahasiswa mhs;
int i, total;
float rata;

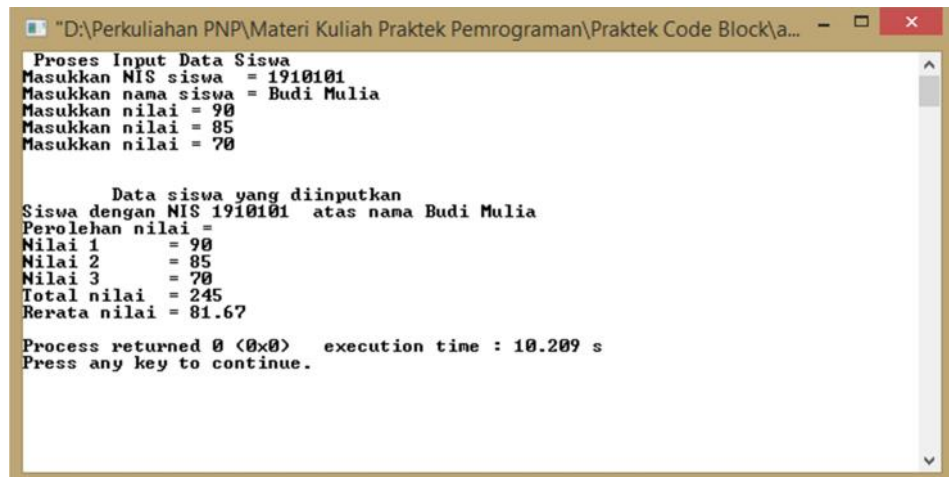
printf(" Proses Input Data Siswa \n");
printf("Masukkan NIS siswa = "); gets(mhs.nis);
printf("Masukkan nama siswa = "); gets(mhs.nama);
total=0;
for(i=0;i<3;i++)
{
    printf("Masukkan nilai = ");
    scanf("%i",&mhs.nilai[i]);
    total=total+mhs.nilai[i];
}
rata=(float)total/3;

printf("\n\n\tData siswa yang diinputkan \n");
printf("Siswa NIS %s atas nama %s\n",mhs.nis,mhs.nama);
printf("Perolehan nilai = \n");
for(i=0;i<3;i++)
    printf("Nilai %i = %i\n",i+1,mhs.nilai[i]);
printf("Total nilai = %i\n",total);
printf("Rerata nilai = %.2f\n",rata);

return 0;
}

```

### Output Program





```

"D:\Perkuliahan PNP\Materi Kuliah Praktek Pemrograman\Praktek Code Block\...
Proses Input Data Siswa
Masukkan NIS siswa = 1910101
Masukkan nama siswa = Budi Mulia
Masukkan nilai = 90
Masukkan nilai = 85
Masukkan nilai = 70

Data siswa yang diinputkan
Siswa dengan NIS 1910101 atas nama Budi Mulia
Perolehan nilai =
Nilai 1 = 90
Nilai 2 = 85
Nilai 3 = 70
Total nilai = 245
Rerata nilai = 81.67

Process returned 0 (0x0) execution time : 10.209 s
Press any key to continue.

```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>7</b>
	<b>TIPE BENTUKAN DAN ARRAY</b>	

b) Program yang menampilkan data persediaan barang, yang terdiri dari :

Kode	Nama	Unit	Harga
M123	Monitor	5	125.000
D124	Disket	10	10.000
P125	Printer	7	300.000

Harga yang tercantum berlaku untuk 1 unit, tambahkan kolom untuk total harga per barang. Tampilkan total harga keseluruhan barang.

```

/* Nama File : bentukan2.c
Pembuat : tuliskan nama dan no.bp anda
Tgl pembuatan : tuliskan tanggal hari ini
Deskripsi : menuliskan isi matriks */



#include "stdio.h"
int main()
{
    typedef struct {
        char kode[5];
        char nama[20];
        int unit;
        float harga;
    }psd;

    psd brg[3] = {
        "M123","Monitor",5,12500,
        "D124","Disket",10,10000,
        "P125","Printer",7,300000};

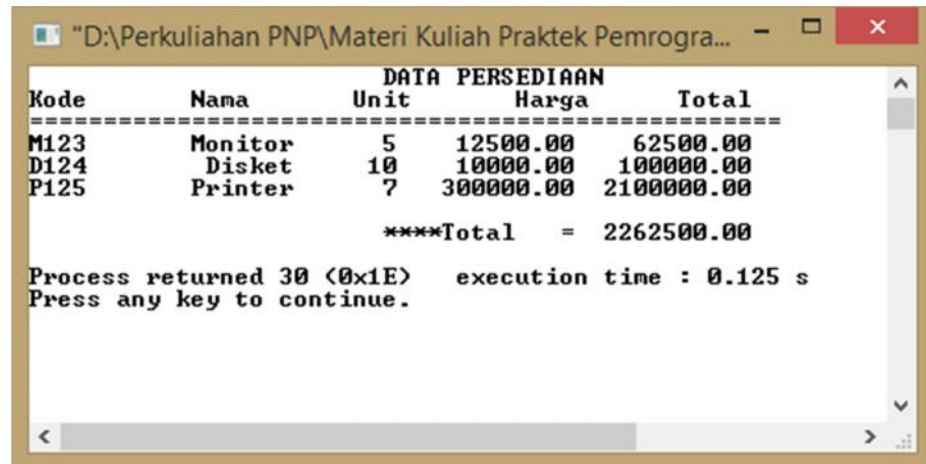
    int i;
    float total;
    total=0;
    printf("\t\t\tDATA PERSEDIAAN\n");
    printf("Kode\t\t\tNama\t\t\tUnit\t\t\tHarga\t\t\tTotal \n");
    printf("=====\n");
    for(i=0;i<3;i++)
    {
        printf("%s\t\t\t%12s\t\t\t%5d\t\t\t%10.2f\t\t\t%10.2f\n", brg[i].kode,
        brg[i].nama,
        brg[i].unit,
        brg[i].harga,
        brg[i].unit*brg[i].harga);
        total=total+(brg[i].unit*brg[i].harga);
    }
    printf("\n\t\t\t\t\t****Total\t\t\t= %10.2f\n",total);
    return 0;
}

```



	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>7</b>
	<b>TIPE BENTUKAN DAN ARRAY</b>	

### Output Program :



```

D:\Perkuliahan PNP\Materi Kuliah Praktek Pemrogra...
DATA PERSEDIAAN
=====
Kode      Nama      Unit      Harga      Total
=====
M123      Monitor    5          12500.00    62500.00
D124      Disket     10         10000.00    100000.00
P125      Printer    7          300000.00   2100000.00

          ****Total   = 2262500.00

Process returned 30 (0x1E)   execution time : 0.125 s
Press any key to continue.

```



## 7. Latihan

- Buatlah **program array bentukan statis** yang memgolah data 4 orang mahasiswa yang terdiri dari NIM, Nama dan 3 buah nilai, kemudian menampilkan total dan rata-rata. Tampilkan hasil akhir dalam bentuk tabel.
- Buatlah **program array bentukan dinamis** yang memgolah data sejumlah mahasiswa yang terdiri dari NIM, Nama dan 3 buah nilai, kemudian menampilkan total dan rata-rata. Banyak mahasiswa ditentukan oleh user (array dinamis). Konversikan rata-rata nilai kedalam nilai huruf, dengan ketentuan :

Nilai 80 : A  
 65 Nilai < 80 : B  
 55 Nilai < 65 : C  
 40 Nilai < 55 : D  
 Nilai < 40 : E

Tampilkan hasil akhir dalam bentuk tabel.

- Buatlah **program pengurutan array tipe bentukan dinamis** yang memgolah data sejumlah mahasiswa yang terdiri dari NIM, Nama dan 3 buah nilai, kemudian menampilkan total dan rata-rata. Banyak mahasiswa ditentukan oleh

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>7</b>
	<b>TIPE BENTUKAN DAN ARRAY</b>	

user (array dinamis), tampilkan nilai tertinggi dan terendah. Kemudian urutkan dari mahasiswa dengan rata-rata tertinggi ke terendah. Tampilkan tabel sebelum dan sesudah pengurutan.

## 8. Keselamatan dan Kesehatan Kerja (K3)

- Periksalah komputer yang anda gunakan, apakah terjadi kerusakan atau tidak. Jika terjadi kerusakan, segera melapor pada Dosen ataupun Teknisi Labor.
- Jika akan melakukan peng-copy-an data, pastikan komputer ataupun alat penyimpanan data Anda terbebas dari virus.
- Selalu matikan komputer setiap Anda selesai praktikum.
- Jangan meninggalkan sampah apapun di ruangan labor komputer.

## 9. Hasil Pengujian dan Pembahasan

Selesaikanlah soal latihan dan kemudian cocokan hasil output program Anda dengan hasil program berikut ini. Jalankan juga program Anda untuk nilai-nilai lainnya.

- Program array bentukan statis



```

D:\Perkuliahan PNP\Materi Kuliah Praktek Pemrograman\Praktek Code Block\...
Input data mahasiswa
Nim mahasiswa 1 = 101
Nama mahasiswa 1 = Ali
Nilai = 76
Nilai = 67
Nilai = 80
Nim mahasiswa 2 = 102
Nama mahasiswa 2 = Budi
Nilai = 80
Nilai = 90
Nilai = 70
Nim mahasiswa 3 = 103
Nama mahasiswa 3 = Cici
Nilai = 90
Nilai = 80
Nilai = 95
Nim mahasiswa 4 = 104
Nama mahasiswa 4 = Daniel
Nilai = 65
Nilai = 78
Nilai = 90
90

Tabel Mahasiswa
NIM      NAMA      Nilai1  Nilai2  Nilai3  Total  Rata-Rata
=====
101      Ali       76      67      80      223    74.33
102      Budi      80      90      70      240    80.00
103      Cici      90      80      95      265    88.33
104      Daniel    65      78      90      233    77.67

Process returned 21 (0x15)   execution time : 45.336 s
Press any key to continue.

```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>7</b>
	<b>TIPE BENTUKAN DAN ARRAY</b>	

b. Program array bentukan dinamis



```

"D:\Perkuliahan PNP\Materi Kuliah Praktek Pemrograman\Praktek Code Block\...
Masukkan jumlah mahasiswa = 3
Input data mahasiswa
Nim mahasiswa 1 = 101
Nama mahasiswa 1 = Ayu
Nilai 1 = 75
Nilai 2 = 65
Nilai 3 = 70
Nim mahasiswa 2 = 102
Nama mahasiswa 2 = Nagita
Nilai 1 = 95
Nilai 2 = 90
Nilai 3 = 85
Nim mahasiswa 3 = 103
Nama mahasiswa 3 = Raffi
Nilai 1 = 55
Nilai 2 = 60
Nilai 3 = 50

Tabel Mahasiswa
NIM      NAMA      Nilai
                        Nilai1  Nilai2  Nilai3
=====
101      Ayu       75      65      70      210     70.00     B
102      Nagita    95      90      85      270     90.00     A
103      Raffi     55      60      50      165     55.00     C
Total nilai kelas = 215
rata-rata kelas = 71.67

Process returned 1 (0x1)  execution time : 40.450 s
Press any key to continue.

```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>7</b>
	<b>TIPE BENTUKAN DAN ARRAY</b>	

c. Program pengurutan array bentukan dinamis

```

D:\Perkuliahan PNP\Materi Kuliah Praktek Pemrograman\Praktek Code Block\...
:: DATA MAHASISWA ::
Masukkan Jumlah Data = 3

Nama : Ayu
Nim : 101
Nilai 1: 70
Nilai 2: 75
Nilai 3: 65

Nama : Budi
Nim : 102
Nilai 1: 90
Nilai 2: 95
Nilai 3: 88

Nama : Citra
Nim : 103
Nilai 1: 70
Nilai 2: 80
Nilai 3: 85

+-----+-----+-----+-----+-----+-----+-----+-----+
| NO | NAMA | NIM | NILAI | NILAI | NILAI | TOTAL | RATA-RATA | NILAI |
|    |      |     | 1     | 2     | 3     |       |           | HURUF |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1  | Ayu  | 101 | 70    | 75    | 65    | 210   | 70.00    | B     |
| 2  | Budi | 102 | 90    | 95    | 88    | 273   | 91.00    | A     |
| 3  | Citra| 103 | 70    | 80    | 85    | 235   | 78.33    | B     |
+-----+-----+-----+-----+-----+-----+-----+

TOTAL KESELURUHAN          718.00
RATA-RATA KESELURUHAN      79.78
NILAI MAKSIMUM              95
NILAI MINIMUM               65

+-----+-----+-----+-----+-----+-----+-----+-----+
| NO | NAMA | NIM | NILAI | NILAI | NILAI | TOTAL | RATA-RATA | NILAI |
|    |      |     | 1     | 2     | 3     |       |           | HURUF |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1  | Budi | 102 | 90    | 95    | 88    | 273   | 91.00    | A     |
| 2  | Citra| 103 | 70    | 80    | 85    | 235   | 78.00    | B     |
| 3  | Ayu  | 101 | 70    | 75    | 65    | 210   | 70.00    | B     |
+-----+-----+-----+-----+-----+-----+-----+



TOTAL KESELURUHAN          718.00
RATA-RATA KESELURUHAN      79.78
NILAI MAKSIMUM              95
NILAI MINIMUM               65

```

## 10. Kesimpulan

Setelah menyelesaikan topik ini, mahasiswa mampu:

- Memahami konsep gabungan antara tipe bentukan terstruktur dan array
- Mendefinisikan dan mendeklarasikan gabungan tipe bentukan terstruktur dan array
- Membuat program yang memproses gabungan array dan tipe bentukan terstruktur dalam bahasa C.

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>7</b>
	<b>TIPE BENTUKAN DAN ARRAY</b>	

### 11. Lembar Penilaian



Item Penilaian	Bobot	Nilai	Nilai X Bobot
Persiapan	5%		
Proses, Contoh, Cara melakukan pekerjaan, sistematika langkah kerja, kemandirian, partisipasi,	25%		
Sikap dan keselamatan kerja, Contoh sikap duduk, sikap menggunakan alat kerja	10%		
Hasil/Data/Produk; Program, Logika program, Kerapian Program, Tampilan Output	50%		
Waktu	10%		
Total	100%		

### 12. Daftar Pustaka

Kelley, Al and Pohl, Ira. 2003. *C by Dissection: The Essentials of C Programming*. Addison-Weasley.

Khannedy, Eko Kurniawan. 2007. Diktat Pemrograman C. Unikom. Bandung.

Liem, Inggriani. 2003. Catatan Singkat Bahasa C. Jurusan Teknik Informatika. Institut Teknologi Bandung.

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>8</b>
	<b>FUNGSI DENGAN NILAI BALIK</b>	

### **1. Judul : Fungsi Dengan Nilai Balik**

### **2. Jumlah Jam/Pertemuan : 3 jam x 1 pertemuan = 3 jam praktek**

### **3. Tujuan Praktek**



- Mahasiswa mampu mengenal dan menjelaskan konsep dan kegunaan fungsi bertipe data (fungsi dengan nilai kembalian/nilai balik)
- Mahasiswa mampu mendefinisikan dan mendeklarasikan fungsi, baik fungsi yang sudah ada didalam Bahasa C maupun yang harus dideklarasikan sendiri.
- Mahasiswa mampu membuat program menggunakan fungsi bertipe data (memiliki nilai kembalian/return)

### **4. Teori Dasar.**

Suatu fungsi adalah suatu bagian dari program yang dimaksudkan untuk mengerjakan suatu tugas tertentu (blok program) dan letaknya dipisahkan dari bagian program yang menggunakannya. Fungsi-fungsi merupakan elemen utama dari program Bahasa C. Program dengan menggunakan bahasa C dibentuk dari kumpulan fungsi, mulai dari fungsi utama dengan nama **main()**, fungsi-fungsi pustaka(standard) dan fungsi-fungsi yang dibuat sendiri oleh pemrogram.

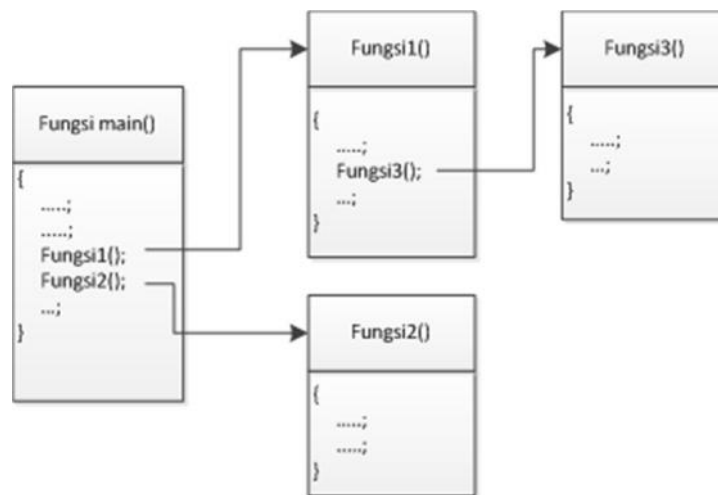
Fungsi-fungsi banyak digunakan dengan 2 alasan utama :

- Fungsi-fungsi menjadikan program mempunyai struktur yang jelas. Dengan memisahkan langkah-langkah detail ke satu atau lebih fungsi-fungsi, maka fungsi utama ( **main()** ) akan menjadi lebih pendek, jelas dan mudah dipahami. Hal seperti ini menunjukkan suatu struktur program yang baik.
- Fungsi-fungsi dapat digunakan untuk menghindari penulisan kode program yang sama yang ditulis secara berulang-ulang. Langkah-langkah program yang sama dan sering digunakan berulang-ulang di program dapat dituliskan sekali

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>8</b>
	<b>FUNGSI DENGAN NILAI BALIK</b>	



saja secara terpisah dalam bentuk fungsi-fungsi. Selanjutnya bagian program yang membutuhkan langkah-langkah ini tidak perlu selalu dituliskan, tetapi cukup dengan pemanggilan fungsi-fungsi tersebut.

Gambar 8.1 berikut akan memperlihatkan mekanisme pemanggilan fungsi dalam Bahasa C.



**Gambar 8.1. Pemanggilan Fungsi dalam Bahasa C**

Mula-mula fungsi `main()` akan dieksekusi oleh kompilator. Oleh karena fungsi `main()` tersebut memanggil `Fungsi1()`, maka kompilator akan mengeksekusi `Fungsi1()`. Dalam `Fungsi1()` juga terdapat pemanggilan `Fungsi3()`, maka kompilator akan mengeksekusi `Fungsi3()` dan kembali lagi mengeksekusi baris selanjutnya yang terdapat di `Fungsi1()` (jika ada). Setelah `Fungsi1()` selesai dieksekusi, kompilator akan kembali mengeksekusi baris berikutnya pada fungsi `main()`, yaitu dengan mengeksekusi kode-kode yang terdapat pada `Fungsi2()`. Setelah selesai, maka kompilator akan melanjutkan pengeksekusian kode pada baris-baris selanjutnya dalam fungsi `main()`. Apabila ternyata dalam fungsi `main()` tersebut kembali terdapat pemanggilan fungsi lain, maka

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>8</b>
	<b>FUNGSI DENGAN NILAI BALIK</b>	

kompatro akan meloncat ke fungsi lain tersebut, begitu seterusnya sampai semua baris kode dalam fungsi `main()` selesai dieksekusi.

Dalam Bahasa C, fungsi terbagi menjadi 2 macam, yakni fungsi yang mengembalikan nilai (*return value*) dan fungsi yang tidak mengembalikan nilai. Fungsi yang tidak mengembalikan nilai tersebut dinamakan dengan *void function*. Sebelum digunakan (dipanggil), suatu fungsi harus dideklarasikan dan didefinisikan terlebih dahulu. Bentuk umum pendeklarasian fungsi adalah:

```
tipe_fungsi nama_fungsi(parameter_fungsi);
```



Sedangkan bentuk umum pendefinisian fungsi adalah:

```
Tipe_fungsi nama_fungsi(parameter_fungsi)
{
    Statement;
    Statement;
    .....;
    .....;
    return nilai_balik;
}
```

Hal-hal yang perlu diperhatikan dalam penggunaan fungsi:

- ✓ Kalau tipe fungsi tidak disebutkan, maka akan dianggap sebagai fungsi dengan nilai keluaran bertipe integer.
- ✓ Untuk fungsi yang memiliki keluaran bertipe bukan integer, maka diperlukan pendefinisian penentu tipe fungsi.
- ✓ Untuk fungsi yang tidak mempunyai nilai keluaran maka dimasukkan ke dalam tipe void
- ✓ Pernyataan yang diberikan untuk memberikan nilai akhir fungsi berupa pernyataan return.
- ✓ Penulisan struktur fungsi bisa dilakukan dengan 2 cara :
  - Diawali dengan judul fungsi beserta parameternya terlebih dahulu sebelum fungsi utama, sedangkan badan fungsi dikerjakan setelah



	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>8</b>
	<b>FUNGSI DENGAN NILAI BALIK</b>	

berakhirnya fungsi utama.

- Deklarasi fungsi lengkap (judul, parameter beserta badan fungsi) diselesaikan lebih dahulu sebelum deklarasi fungsi utama.

## 5. Bahan/Peralatan yang digunakan

- Komputer
- Aplikasi CodeBlocks



## 6. Langkah Kerja

- Jalankan program CodeBlocks.
- Buatlah beberapa program berikut di komputer anda. Compile dan jalankan (Run). Perhatikan hasil keluarannya. Jalankan program dan analisa hasilnya.
  - Program yang menghitung pangkat dan akar kuadrat menggunakan fungsi yang telah disediakan bahasa C (pada library math.h)

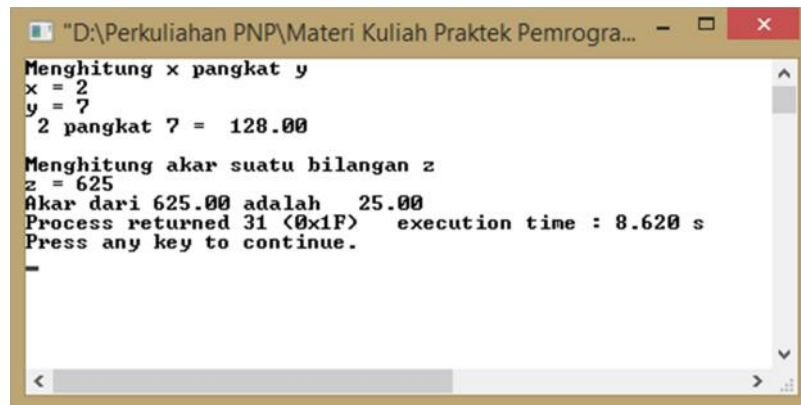
```
/* Nama File : pangkat_akar.c
   Pembuat   : tuliskan nama dan no.bp anda
   Tgl pembuatan : tuliskan tanggal hari ini
   Deskripsi  : menghitung pangkat dan akar kuadrat satu
                bilangan*/

#include "stdio.h"
#include "math.h"

void main()
{
    int x, y;
    float z;
    printf("Menghitung x pangkat y\n");
    printf("x = "); scanf("%i",&x);
    printf("y = "); scanf("%i",&y);
    printf(" %i pangkat %i = %7.2lf\n", x, y, pow(x, y));
    printf("\nMenghitung akar suatu bilangan z\n");
    printf("z = "); scanf("%f", &z);
    printf("Akar dari %f adalah %7.2lf", z, sqrt(z));
}
```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>8</b>
	<b>FUNGSI DENGAN NILAI BALIK</b>	

### Output Program



- b) Program yang mencari nilai  $f(a)-f(b)$ , dimana a dan b diinputkan user dan fungsi  $f(x) = x^2+2x-5$

```

/* Nama File : fungsi1.c
   Pembuat   : tuliskan nama dan no.bp anda
   Tgl pembuatan : tuliskan tanggal hari ini
   Deskripsi : menghitung pangkat dan akar kuadrat satu
              bilangan*/



#include "stdio.h"

int f(int x);
void main()
{
    int a,b,x;

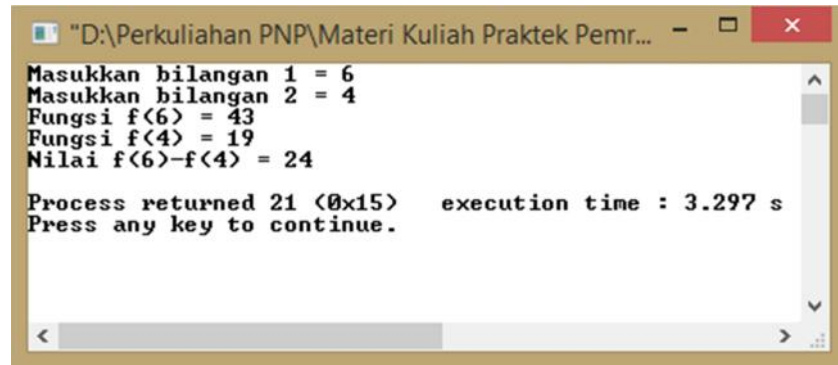
    printf("Masukkan bilangan 1 = ");
    scanf("%i",&a);
    printf("Masukkan bilangan 2 = ");
    scanf("%i",&b);
    printf("Fungsi f(%i) = %i\n",a,f(a));
    printf("Fungsi f(%i) = %i\n",b,f(b));
    x=f(a)-f(b);
    printf("Nilai f(%i)-f(%i) = %i\n",a,b,x);
}

int f(int x)
{
    return(x*x+2*x-5);
}

```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>8</b>
	<b>FUNGSI DENGAN NILAI BALIK</b>	

### Output Program



```

"D:\Perkuliahan PNP\Materi Kuliah Praktek Pemr... - [X]
Masukkan bilangan 1 = 6
Masukkan bilangan 2 = 4
Fungsi f(6) = 43
Fungsi f(4) = 19
Nilai f(6)-f(4) = 24

Process returned 21 (0x15)   execution time : 3.297 s
Press any key to continue.

```

- c) Program untuk mencari luas dan keliling persegi panjang dengan input panjang dan lebar



```

/* Nama File : fungsi_fgsluaskel.c
Pembuat   : tuliskan nama dan no.bp anda
Tgl pembuatan : tuliskan tanggal hari ini
Deskripsi : menjumlahkan dua buah bil bulat*/

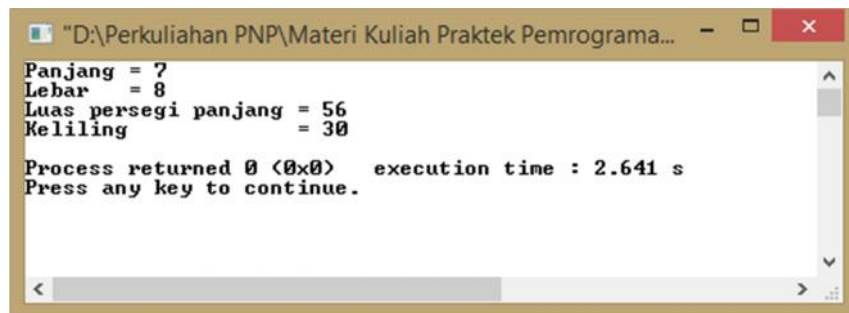
#include "stdio.h"

int luas(int p,int l)
{
    return(p*l);
}
int keliling(int p,int l)
{
    return(2*p+2*l);
}
int main()
{
    int a,b;
    printf("Panjang = "); scanf("%i",&a);
    printf("Lebar   = "); scanf("%i",&b);
    printf("Luas persegi panjang = %i\n",luas(a,b));
    printf("Keliling           = %i\n",keliling(a,b));
    return 0;
}

```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>8</b>
	<b>FUNGSI DENGAN NILAI BALIK</b>	

### Output Program



```

"D:\Perkuliahan PNP\Materi Kuliah Praktek Pemrograman...
Panjang = 7
Lebar = 8
Luas persegi panjang = 56
Keliling = 30

Process returned 0 (0x0)   execution time : 2.641 s
Press any key to continue.

```

#### d) Program mencari nilai maksimum dan minimum

```

/* Nama File : fungsi_maxmin.c
   Pembuat   : tuliskan nama dan no.bp anda
   Tgl pembuatan : tuliskan tanggal hari ini
   Deskripsi  : menentukan nilai max dan min dari 3 bil
   bulat*/



#include "stdio.h"

//Definisi fungsi
int max2 (int x, int y);
int min2 (int x, int y);

void main()
{
    int a, b, c;
    int max, min;
    printf("A = "); scanf("%d", &a);
    printf("B = "); scanf("%d", &b);
    printf("C = "); scanf("%d", &c);
    max = max2(max2(a, b),c); /* pemanggilan fungsi max2() */
    min = min2(min2(a, b),c);
    printf("\nNilai terbesar = %d \n", max);
    printf("\nNilai terkecil = %d \n", min);
}

//Fungsi
/* Definisi fungsi , tanpa titik koma */
int max2 (int x, int y)
{
    if (x>=y)
        return x;
    else

```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>8</b>
	<b>FUNGSI DENGAN NILAI BALIK</b>	

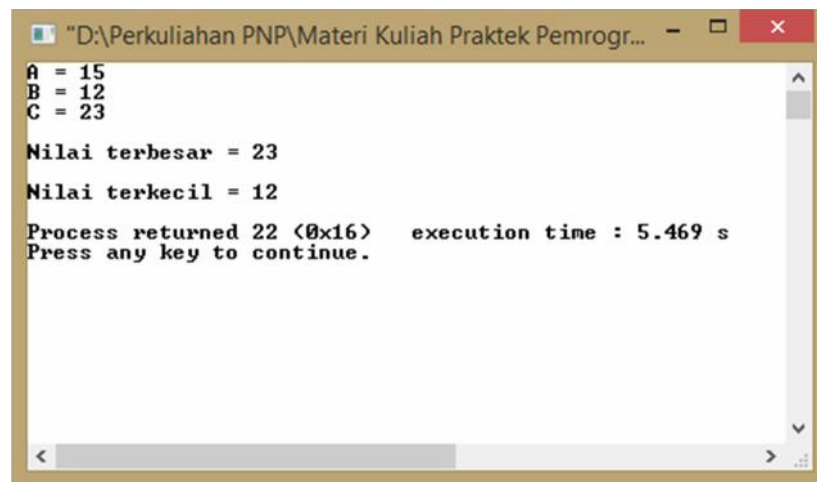
```

        return y;
    }

    int min2 (int x, int y)
    {
        if (x<=y)
            return x;
        else
            return y;
    }

```

### Output Program



```

"D:\Perkuliahan PNP\Materi Kuliah Praktek Pemrogr...
A = 15
B = 12
C = 23

Nilai terbesar = 23
Nilai terkecil = 12

Process returned 22 (0x16)    execution time : 5.469 s
Press any key to continue.



```

## 7. Latihan

- Buatlah **program yang mencari nilai faktorial** menggunakan fungsi.
- Buatlah **program yang mencari nilai permutasi dan kombinasi** dari dua buah angka yang diinputkan oleh user.
- Buatlah **program Kabataku** menggunakan fungsi dengan dua angka yang diinputkan user
- Buatlah **fungsi yang menghitung nilai**  $f(x,y) = x^3 + 3x^2y + 6xy^2 + y^3$

## 8. Keselamatan dan Kesehatan Kerja (K3)

- Periksalah komputer yang anda gunakan, apakah terjadi kerusakan atau tidak. Jika terjadi kerusakan, segera melapor pada Dosen ataupun Teknisi Labor.

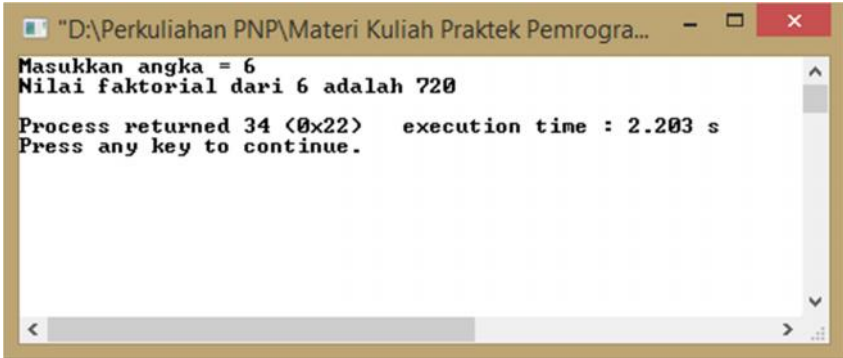
	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>8</b>
	<b>FUNGSI DENGAN NILAI BALIK</b>	

- Jika akan melakukan peng-copy-an data, pastikan komputer ataupun alat penyimpanan data Anda terbebas dari virus.
- Selalu matikan komputer setiap Anda selesai praktikum.
- Jangan meninggalkan sampah apapun di ruangan labor komputer.

## 9. Hasil Pengujian dan Pembahasan

Selesaikanlah soal latihan dan kemudian cocokan hasil output program Anda dengan hasil program berikut ini. Jalankan juga program Anda untuk nilai-nilai lainnya.

- Program mencari nilai faktorial

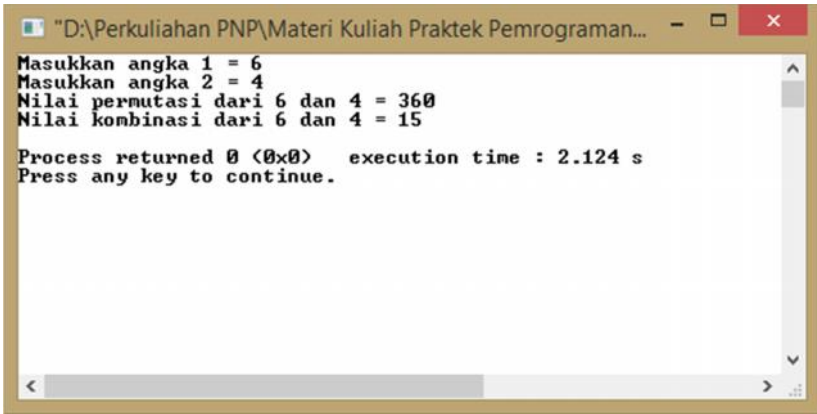


```

"D:\Perkuliahan PNP\Materi Kuliah Praktek Pemrograman...
Masukkan angka = 6
Nilai faktorial dari 6 adalah 720
Process returned 34 (0x22)   execution time : 2.203 s
Press any key to continue.

```



- Program yang mencari nilai permutasi dan kombinasi array bentukan dinamis



```

"D:\Perkuliahan PNP\Materi Kuliah Praktek Pemrograman...
Masukkan angka 1 = 6
Masukkan angka 2 = 4
Nilai permutasi dari 6 dan 4 = 360
Nilai kombinasi dari 6 dan 4 = 15
Process returned 0 (0x0)   execution time : 2.124 s
Press any key to continue.

```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>8</b>
	<b>FUNGSI DENGAN NILAI BALIK</b>	

c. Program Kabataku

```

"D:\Perkuliahan PNP\Materi Kuliah Praktek Pemro...
A = 7.5
B = 3.5
7.50 + 3.50 = 11.00
7.50 - 3.50 = 4.00
7.50 * 3.50 = 26.25
7.50 / 3.50 = 2.14
Process returned 20 (0x14)   execution time : 8.041 s
Press any key to continue.

```

d. Fungsi yang menghitung nilai  $f(x,y)=x^3+3x^2y+6xy^2+y^3$

```



"D:\Perkuliahan PNP\Materi Kuliah Praktek Pemrograman...
Masukkan angka 1 = 2
Masukkan angka 2 = 3
Nilai f(2,3) = 179
Nilai f(3,2) = 161
Process returned 0 (0x0)   execution time : 3.234 s
Press any key to continue.

```

## 10. Kesimpulan

Setelah menyelesaikan topik ini, mahasiswa mampu:

- Memahami konsep dan kegunaan fungsi bertipe data (dengan nilai kembalian)
- Mendefinisikan dan mendeklarasikan fungsi bertipe data (dengan nilai kembalian) serta melakukan pemanggilannya.
- Membuat program yang memproses fungsi bertipe data (dengan nilai kembalian)

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>8</b>
	<b>FUNGSI DENGAN NILAI BALIK</b>	

## 11. Lembar Penilaian

Item Penilaian	Bobot	Nilai	Nilai X Bobot
Persiapan	5%		
Proses, Contoh, Cara melakukan pekerjaan, sistematika langkah kerja, kemandirian, partisipasi,	25%		
Sikap dan keselamatan kerja, Contoh sikap duduk, sikap menggunakan alat kerja	10%		
Hasil/Data/Produk; Program, Logika program, Kerapian Program, Tampilan Output	50%		
Waktu	10%		
Total	100%		

## 12. Daftar Pustaka



Hartono, J. 2001. *Konsep Dasar Bahasa Pemrograman C*. Andi Offset Yogyakarta

Kelley, Al and Pohl, Ira. 2003. *C by Dissection: The Essentials of C Programming*. Addison-Weasley.

Khannedy, Eko Kurniawan. 2007. *Diktat Pemrograman C*. Unikom. Bandung.

Liem, Inggriani. 2003. *Catatan Singkat Bahasa C*. Jurusan Teknik Informatika. Institut Teknologi Bandung.



	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>9</b>
	<b>FUNGSI BERTIPE VOID</b>	

### 1. Judul : Fungsi Bertipe Void

### 2. Jumlah Jam/Pertemuan : 3 jam x 2 pertemuan = 6 jam praktek

### 3. Tujuan Praktek

- Mahasiswa mampu mengenal dan menjelaskan konsep dan kegunaan fungsi bertipe void (fungsi tanpa nilai kembalian)
- Mahasiswa mampu mendefinisikan dan mendeklarasikan fungsi bertipe void (dengan beragam parameter)
- Mahasiswa mampu membuat program menggunakan fungsi bertipe void

### 4. Teori Dasar.



#### Variabel Lokal

Variabel lokal merupakan variabel yang namanya dan nilainya hanya dikenal di suatu blok statemen tertentu saja atau di dalam suatu fungsi. Untuk membuat suatu variabel menjadi bersifat lokal, maka variabel ini harus dideklarasikan di dalam blok bersangkutan. Variabel lokal akan dihapus dari memori jika proses sudah meninggalkan blok statemen letak variabel lokalnya.

#### Variabel Global

Variabel global merupakan variabel yang dikenal disemua bagian-bagian tertentu dari program. Variabel-variabel global dapat dibuat dengan cara mendeklarasikannya diluar suatu blok statemen atau diluar fungsi-fungsi yang menggunakannya.

Fungsi bertipe void, kalau dalam program pascal atau delphi disebut sebagai **prosedur**. Fungsi ini tidak mempunyai nilai kembalian, jadi fungsi bertipe ini hanya merupakan sekumpulan kode program yang bekerja sesuai dengan

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>9</b>
	<b>FUNGSI BERTIPE VOID</b>	

parameter yang diberikan.

### **Parameter Formal dan Parameter Aktual**

- ✓ Parameter Formal adalah variabel yang ada pada daftar parameter dalam definisi fungsi.
- ✓ Parameter Aktual adalah variabel (parameter) yang dipakai dalam pemanggilan fungsi.

Dalam contoh program penambahan pada job sebelumnya, parameter formal terdapat pada pendefinisian fungsi :

```
float tambah(float x, float y) //parameter formal
{
    return (a+b);
}
```



Sedangkan parameter aktual terdapat pada pemanggilan fungsi :

```
void main()
{
    .....
    .....
    c = tambah(a, b); //parameter aktual
    .....
}
```

### **Cara Melewatkan Parameter**

Cara melewati suatu parameter dalam Bahasa C ada dua cara yaitu :

- Pemanggilan Secara Nilai (*Call by Value*)
  - Call by value akan menyalin nilai dari parameter aktual ke parameter formal.
  - Yang dikirimkan ke fungsi adalah nilai dari datanya, bukan alamat memori letak dari datanya.
  - Fungsi yang menerima kiriman nilai akan menyimpannya di alamat terpisah dari nilai aslinya yang digunakan oleh bagian program yang memanggil fungsi.

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>9</b>
	<b>FUNGSI BERTIPE VOID</b>	

- Perubahan nilai di fungsi (parameter formal) tidak akan merubah nilai asli di bagian program yang memanggilnya.
- Pengiriman parameter secara nilai adalah pengiriman searah, yaitu dari bagian program yang memanggil fungsi ke fungsi yang dipanggil.
- Pengiriman suatu nilai dapat dilakukan untuk suatu ungkapan, tidak hanya untuk sebuah variabel, elemen array atau konstanta saja.

➤ **Pemanggilan Secara Referensi (*Call by Reference*)**



- Pemanggilan secara Referensi merupakan upaya untuk melewati alamat dari suatu variabel ke dalam fungsi.
- Yang dikirimkan ke fungsi adalah alamat letak dari nilai datanya, bukan nilai datanya.
- Fungsi yang menerima kiriman alamat ini akan menggunakan alamat yang sama untuk mendapatkan nilai datanya.
- Perubahan nilai di fungsi akan merubah nilai asli di bagian program yang memanggil fungsi.
- Pengiriman parameter secara referensi adalah pengiriman dua arah, yaitu dari fungsi pemanggil ke fungsi yang dipanggil dan juga sebaliknya.
- Pengiriman secara acuan tidak dapat dilakukan untuk suatu ungkapan.

**5. Bahan/Peralatan yang digunakan**

- a. Komputer
- b. Aplikasi CodeBlocks

**6. Langkah Kerja**

- 1) Jalankan program CodeBlocks.
- 2) Buatlah beberapa program berikut di komputer anda. Compile dan jalankan (Run). Perhatikan hasil keluarannya. Jalankan program dan analisa hasilnya.

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>9</b>
	<b>FUNGSI BERTIPE VOID</b>	

- a) Program dengan prosedur hitung luas segitiga tanpa menggunakan parameter

```

/* Nama File : hit_luas1.c
   Pembuat   : tuliskan nama dan no.bp anda
   Tgl pembuatan : tuliskan tanggal hari ini
   Deskripsi  : prosedur hitung luas segitiga tanpa
               parameter*/



#include "stdio.h"

void Hit_LuasSg3();

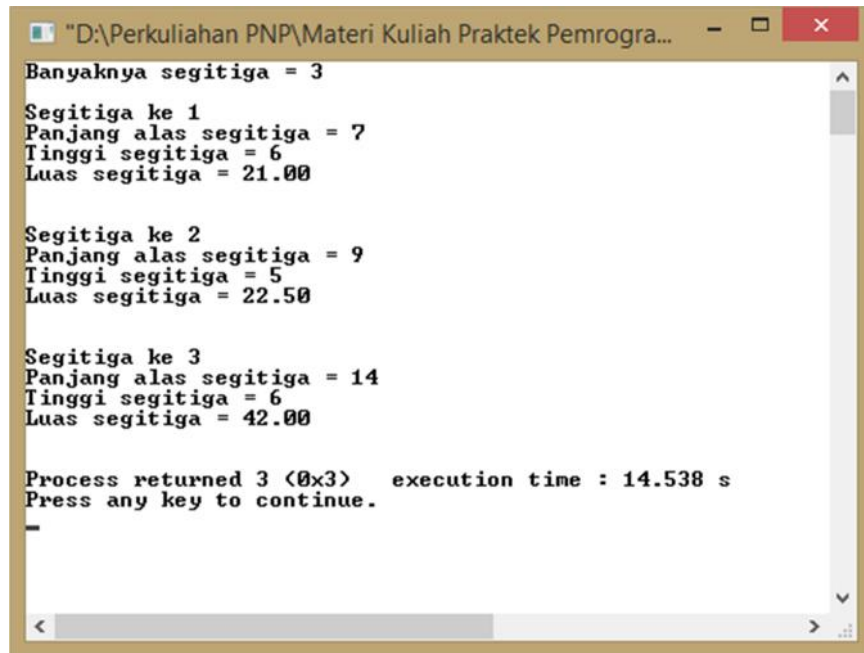
void main()
{
    int i, N;
    printf("Banyaknya segitiga = ");
    scanf("%d",&N);
    for(i=1; i<=N; i++)
    {
        printf("\nSegitiga ke %i\n",i);
        Hit_LuasSg3();
        printf("\n");
    }
}

void Hit_LuasSg3()
{
    float alas, tinggi, luas;
    printf("Panjang alas segitiga = ");
    scanf("%f",&alas);
    printf("Tinggi segitiga = ");
    scanf("%f",&tinggi);
    luas = (alas * tinggi)/2.0;
    printf("Luas segitiga = %0.2f\n",luas);
}

```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>9</b>
	<b>FUNGSI BERTIPE VOID</b>	

### Output Program :



```

"D:\Perkuliahan PNP\Materi Kuliah Praktek Pemrogra...
Banyaknya segitiga = 3

Segitiga ke 1
Panjang alas segitiga = 7
Tinggi segitiga = 6
Luas segitiga = 21.00

Segitiga ke 2
Panjang alas segitiga = 9
Tinggi segitiga = 5
Luas segitiga = 22.50

Segitiga ke 3
Panjang alas segitiga = 14
Tinggi segitiga = 6
Luas segitiga = 42.00

Process returned 3 (0x3)   execution time : 14.538 s
Press any key to continue.

```

- b) Program dengan prosedur hitung luas segitiga dengan menggunakan parameter masukan (input)

```



/* Nama File : hit_luas2.c
Pembuat   : tuliskan nama dan no.bp anda
Tgl pembuatan : tuliskan tanggal hari ini
Deskripsi : prosedur hitung luas segitiga dgn parameter
            masukan (input)*/

#include "stdio.h"

void Hit_LuasSg3 (float alas, float tinggi);

void main()
{
    int i, N;
    float a, t ;
    printf("Banyaknya segitiga = ");
    scanf("%d",&N);
    for(i=1; i<=N; i++)

```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>9</b>
	<b>FUNGSI BERTIPE VOID</b>	

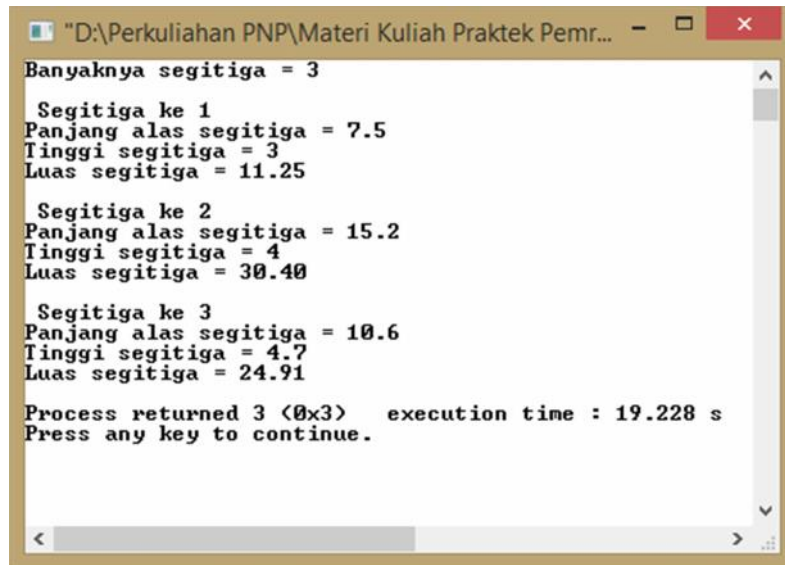
```

{
    printf("\n Segitiga ke %i\n",i);
    printf("Panjang alas segitiga = ");
    scanf("%f",&a);
    printf("Tinggi segitiga = ");
    scanf("%f",&t);
    Hit_LuasSg3(a,t);
}

void Hit_LuasSg3 (float alas, float tinggi)
{
    float luas;
    luas = (alas * tinggi)/2.0;
    printf("Luas segitiga = %0.2f\n",luas);
}

```

#### Output Program :



```

"D:\Perkuliahan PNP\Materi Kuliah Praktek Pemr...
Banyaknya segitiga = 3

Segitiga ke 1
Panjang alas segitiga = 7.5
Tinggi segitiga = 3
Luas segitiga = 11.25

Segitiga ke 2
Panjang alas segitiga = 15.2
Tinggi segitiga = 4
Luas segitiga = 30.40

Segitiga ke 3
Panjang alas segitiga = 10.6
Tinggi segitiga = 4.7
Luas segitiga = 24.91

Process returned 3 (0x3)   execution time : 19.228 s
Press any key to continue.



```

- c) Program dengan prosedur hitung luas segitiga dengan menggunakan parameter masukan (input) dan parameter keluaran (output)

```

/* Nama File : hit_luas3.c
Pembuat : tuliskan nama dan no.bp anda
Tgl pembuatan : tuliskan tanggal hari ini
Deskripsi : prosedur hitung luas segitiga dgn parameter
masukan (input) dan keluaran (output)*/

```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>9</b>
	<b>FUNGSI BERTIPE VOID</b>	

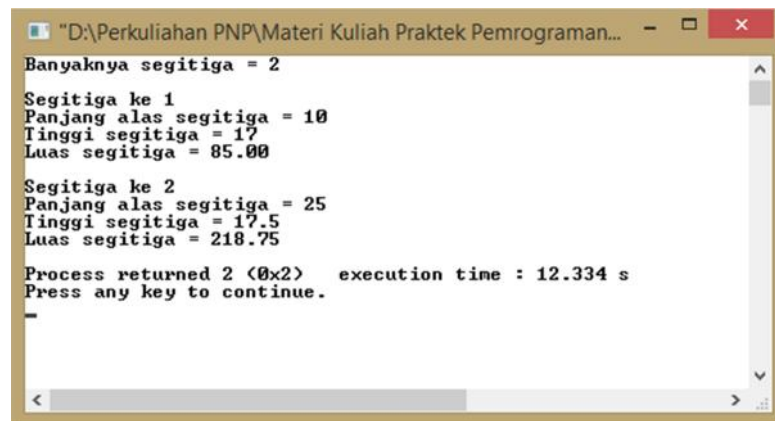
```
#include "stdio.h"

void Hit_LuasSg3 (float alas, float tinggi, float *luas);

void main()
{
    int i, N;
    float a, t, L ;
    printf("Banyaknya segitiga = ");
    scanf("%d",&N);
    for(i=1; i<=N; i++)
    {
        printf("\nSegitiga ke %i\n",i);
        printf("Panjang alas segitiga = ");
        scanf("%f",&a);
        printf("Tinggi segitiga = ");
        scanf("%f",&t);
        Hit_LuasSg3(a,t,&L);
        printf("Luas segitiga = %0.2f\n",L);
    }
}

void Hit_LuasSg3 (float alas, float tinggi, float *luas)
{
    *luas = (alas * tinggi)/2.0;
}
```

#### Output Program :



```
"D:\Perkuliahan PNP\Materi Kuliah Praktek Pemrograman..."
Banyaknya segitiga = 2



Segitiga ke 1
Panjang alas segitiga = 10
Tinggi segitiga = 17
Luas segitiga = 85.00

Segitiga ke 2
Panjang alas segitiga = 25
Tinggi segitiga = 17.5
Luas segitiga = 218.75

Process returned 2 (0x2)   execution time : 12.334 s
Press any key to continue.
-
```

- d) Program yang menukarkan isi dua buah variabel menggunakan konsep *call by value*

```
/* Nama File : tukar_value.c
Pembuat : tuliskan nama dan no.bp anda
Tgl pembuatan : tuliskan tanggal hari ini
Deskripsi : prosedur tukar menggunakan call by value*/
```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>9</b>
	<b>FUNGSI BERTIPE VOID</b>	

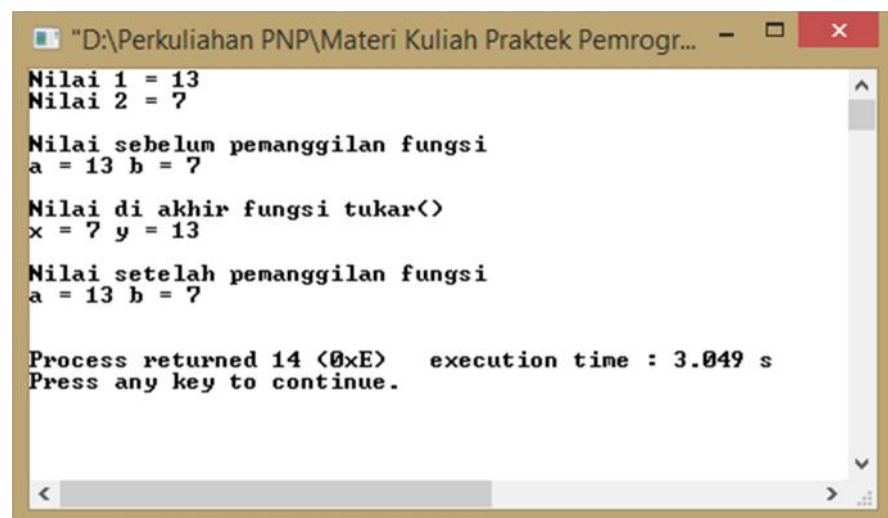
```
#include "stdio.h"

void tukar(int x, int y); /* pendeklarasian fungsi */

void main()
{
    int a,b;
    printf("Nilai 1 = ");scanf("%i",&a);
    printf("Nilai 2 = ");scanf("%i",&b);
    printf("\nNilai sebelum pemanggilan fungsi\n");
    // a dan b sebelum pemanggilan fungsi
    printf("a = %i b = %i\n\n", a, b);
    tukar(a,b); /* pemanggilan fungsi tukar() */
    printf("Nilai setelah pemanggilan fungsi\n");
    // a dan b setelah pemanggilan fungsi
    printf("a = %i b = %i\n\n", a, b);
}

/* Pendefinisian fungsi tukar() */
void tukar(int x, int y)
{
    int z; /* variabel sementara */
    z = x;
    x = y;
    y = z;
    printf("Nilai di akhir fungsi tukar()\n");
    printf("x = %i y = %i\n\n", x, y);
}
```

### Output Program



```
"D:\Perkuliah PNP\Materi Kuliah Praktek Pemrogr...
Nilai 1 = 13
Nilai 2 = 7



Nilai sebelum pemanggilan fungsi
a = 13 b = 7

Nilai di akhir fungsi tukar()
x = 7 y = 13

Nilai setelah pemanggilan fungsi
a = 13 b = 7

Process returned 14 (0xE)   execution time : 3.049 s
Press any key to continue.
```



	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>9</b>
	<b>FUNGSI BERTIPE VOID</b>	

- e) Program yang menukarkan isi dua buah variabel menggunakan konsep *call by references* (input/output)

```

/* Nama File : tukar_references.c
Pembuat : tuliskan nama dan no.bp anda
Tgl pembuatan : tuliskan tanggal hari ini
Deskripsi : prosedur tukar menggunakan call by
references*/



#include "stdio.h"

void tukar(int *px, int *py); /* pendeklarasian fungsi */

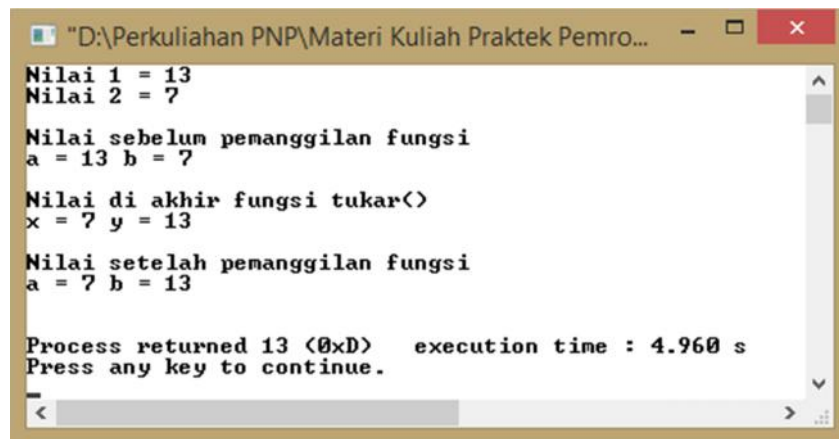
void main()
{
    int a,b;
    printf("Nilai 1 = ");scanf("%i",&a);
    printf("Nilai 2 = ");scanf("%i",&b);
    printf("\nNilai sebelum pemanggilan fungsi\n");
    // a dan b sebelum pemanggilan fungsi
    printf("a = %i b = %i\n\n", a, b);
    tukar(&a,&b); /* pemanggilan fungsi tukar() */
    printf("Nilai setelah pemanggilan fungsi\n");
    // a dan b setelah pemanggilan fungsi
    printf("a = %i b = %i\n\n", a, b);
}

/* Pendefinisian fungsi tukar() */
void tukar(int *px, int *py)
{
    int z; /* variabel sementara */
    z = *px;
    *px = *py;
    *py = z;
    printf("Nilai di akhir fungsi tukar()\n");
    printf("x = %i y = %i\n\n", *px, *py);
}

```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>9</b>
	<b>FUNGSI BERTIPE VOID</b>	

### Output Program



```

"D:\Perkuliahan PNP\Materi Kuliah Praktek Pemro...
Nilai 1 = 13
Nilai 2 = 7

Nilai sebelum pemanggilan fungsi
a = 13 b = 7

Nilai di akhir fungsi tukar()
x = 7 y = 13

Nilai setelah pemanggilan fungsi
a = 7 b = 13

Process returned 13 (0xD)   execution time : 4.960 s
Press any key to continue.



```

### 7. Latihan

- Buatlah **program untuk menghitung volume dan luas permukaan bola** menggunakan prosedur dengan parameter input
- Buatlah **program yang mengkonversikan angka yang diinputkan user kedalam nama bulan** secara berulang dengan memanfaatkan prosedur.

### 8. Keselamatan dan Kesehatan Kerja (K3)

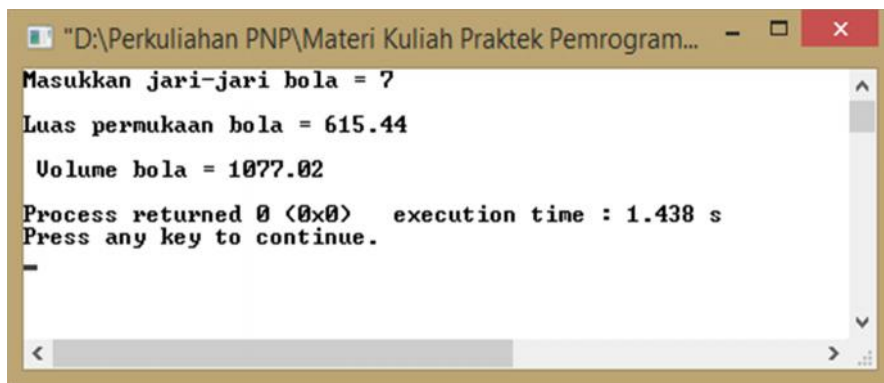
- Periksalah komputer yang anda gunakan, apakah terjadi kerusakan atau tidak. Jika terjadi kerusakan, segera melapor pada Dosen ataupun Teknisi Labor.
- Jika akan melakukan peng-copy-an data, pastikan komputer ataupun alat penyimpanan data Anda terbebas dari virus.
- Selalu matikan komputer setiap Anda selesai praktikum.
- Jangan meninggalkan sampah apapun di ruangan labor komputer.

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>9</b>
	<b>FUNGSI BERTIPE VOID</b>	

## 9. Hasil Pengujian dan Pembahasan

Selesaikanlah soal latihan dan kemudian cocokan hasil output program Anda dengan hasil program berikut ini. Jalankan juga program Anda untuk nilai-nilai lainnya.

- a. Program menghitung volume dan luas permukaan bola.

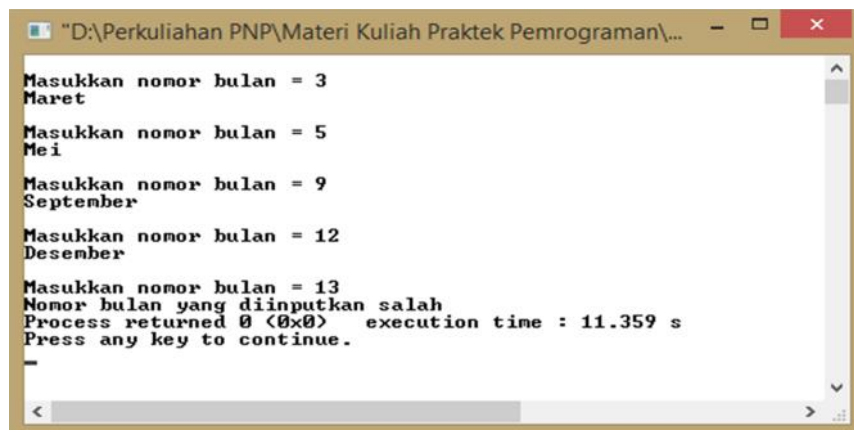


```

"D:\Perkuliahan PNP\Materi Kuliah Praktek Pemrograman..."
Masukkan jari-jari bola = 7
Luas permukaan bola = 615.44
Volume bola = 1077.02
Process returned 0 (0x0)   execution time : 1.438 s
Press any key to continue.

```

- b. Program yang mengkonversikan angka yang diinputkan user kedalam nama bulan



```



"D:\Perkuliahan PNP\Materi Kuliah Praktek Pemrograman\..."
Masukkan nomor bulan = 3
Maret
Masukkan nomor bulan = 5
Mei
Masukkan nomor bulan = 9
September
Masukkan nomor bulan = 12
Desember
Masukkan nomor bulan = 13
Nomor bulan yang diinputkan salah
Process returned 0 (0x0)   execution time : 11.359 s
Press any key to continue.

```

## 10. Kesimpulan

Setelah menyelesaikan topik ini, mahasiswa mampu:

- a. Memahami konsep dan kegunaan fungsi bertipe void (prosedur)

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>9</b>
	<b>FUNGSI BERTIPE VOID</b>	

- b. Mendefinisikan dan mendeklarasikan fungsi bertipe void (dengan beragam parameter) serta melakukan pemanggilannya.
- c. Membuat program yang memproses fungsi bertipe void

### 11. Lembar Penilaian

Item Penilaian	Bobot	Nilai	Nilai X Bobot
Persiapan	5%		
Proses, Contoh, Cara melakukan pekerjaan, sistematika langkah kerja, kemandirian, partisipasi,	25%		
Sikap dan keselamatan kerja, Contoh sikap duduk, sikap menggunakan alat kerja	10%		
Hasil/Data/Produk; Program, Logika program, Kerapian Program, Tampilan Output	50%		
Waktu	10%		
Total	100%		



### 12. Daftar Pustaka

Hartono, J. 2001. *Konsep Dasar Bahasa Pemrograman C*. Andi Offset Yogyakarta

Kelley, Al and Pohl, Ira. 2003. *C by Dissection: The Essentials of C Programming*. Addison-Weasley.

Khannedy, Eko Kurniawan. 2007. *Diktat Pemrograman C*. Unikom. Bandung.

Liem, Inggriani. 2003. *Catatan Singkat Bahasa C*. Jurusan Teknik Informatika. Institut Teknologi Bandung.

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>10</b>
	<b>GABUNGAN FUNGSI DENGAN ARRAY</b>	

### **1. Judul : Gabungan Fungsi dengan Array**

### **2. Jumlah Jam/Pertemuan : 3 jam x 1 pertemuan = 3 jam praktek**

### **3. Tujuan Praktek**

- Mahasiswa mampu mengenal dan menjelaskan konsep gabungan fungsi dengan parameter array
- Mahasiswa mampu mendefinisikan dan mendeklarasikan gabungan fungsi dengan parameter berupa array baik satu maupun dua dimensi
- Mahasiswa mampu membuat program menggunakan gabungan fungsi dengan parameter array

### **4. Teori Dasar.**



Pendefinisian fungsi tidak lepas dari pendefinisian parameter. Pada kegiatan praktek kali ini akan dibahas tentang parameter berupa array.

- Pengiriman parameter berupa array dimensi satu

Pengiriman parameter berupa array/ larik dimensi satu sebenarnya merupakan pengiriman acuan, karena sebenarnya yang dikirimkan adalah alamat dari elemen pertama lariknya, bukan seluruh nilai elemen-elemennya. Alamat elemen pertama dari larik dapat ditunjukkan oleh nama lariknya yang tidak ditulis dengan indeksnya. Bentuk ini akan tampak sebagai argument di parameter nyata untuk bagian yang memanggil fungsi.

- Pengiriman parameter berupa array dimensi dua

Pengiriman parameter berupa array dimensi dua, untuk parameter nyatanya dapat dilakukan dengan cara yang sama dengan pengiriman parameter berupa array dimensi satu, yaitu dengan mengirimkan alamat elemen pertama dari larik. Perbedaannya adalah terletak di parameter formalnya. Di parameter

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>10</b>
	<b>GABUNGAN FUNGSI DENGAN ARRAY</b>	

formal ini, jumlah dari maksimum kolom yang telah dideklarasikan di bagian program yang memanggil fungsi harus ikut disebutkan. Hal ini diperlukan supaya fungsi yang menggunakan array dimensi dua ini dapat mengetahui alamat dari nilai-nilai elemen kolom pertama tiap-tiap barisnya.

## 5. Bahan/Peralatan yang digunakan

- Komputer
- Aplikasi CodeBlocks

## 6. Langkah Kerja

- Jalankan program CodeBlocks.
- Buatlah beberapa program berikut di komputer anda. Compile dan jalankan (Run). Perhatikan hasil keluarannya. Jalankan program dan analisa hasilnya.
  - Program yang mencari total dari sejumlah data yang diinputkan user, dimana proses input dan mencari total dalam fungsi yang berbeda

```

/* Nama File : arrfung.c
   Pembuat   : tuliskan nama dan no.bp anda
   Tgl pembuatan : tuliskan tanggal hari ini
   Deskripsi  : fungsi dengan parameter array 1 D */

#include "stdio.h"



int i; // variabel global
void Input(int X[], int Jumlah); //Prototype fungsi
int Hitung_Total(int X[],int Jumlah); //Prototype fungsi

void main()
{
    int n,total;
    int nilai[1000];
    printf("Jumlah Data = "); scanf("%i",&n);
    Input(nilai,n); // pemanggilan fungsi dengan tipe void
    total = Hitung_Total(nilai,n); //pemanggilan fungsi dengan
    nilai balik

    printf("Total nilai = %i\n",total);
}

/* Fungsi memasukkan data array 1 D */

```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>10</b>
	<b>GABUNGAN FUNGSI DENGAN ARRAY</b>	

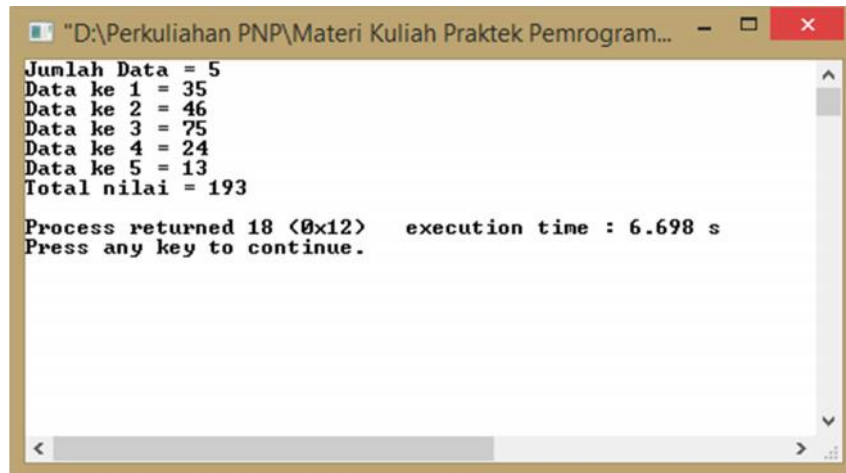
```

void Input(int X[], int Jumlah)
{
    for(i=0;i<Jumlah;i++)
    {
        printf("Data ke %i = ",i+1);
        scanf("%i",&X[i]);
    }
}

/*Fungsi menghitung total dari elemen array*/
int Hitung_Total(int X[],int Jumlah)
{
    int total;
    total = 0;
    for(i=0;i<Jumlah;i++)
        total=total+X[i];
    return(total);
}

```

### Output Program



```

D:\Perkuliahan PNP\Materi Kuliah Praktek Pemrogram...
Jumlah Data = 5
Data ke 1 = 35
Data ke 2 = 46
Data ke 3 = 75
Data ke 4 = 24
Data ke 5 = 13
Total nilai = 193
Process returned 18 (0x12)   execution time : 6.698 s
Press any key to continue.

```

- b) Program untuk menjumlahkan dua buah matriks dimana proses isi, jumlah dan menampilkan dalam fungsi yang berbeda-beda.



```

/* Nama File : arrfung2.c
   Pembuat   : tuliskan nama dan no.bp anda
   Tgl pembuatan : tuliskan tanggal hari ini
   Deskripsi  : fungsi dengan parameter array 2 D */

#include "stdio.h"

// penentuan konstanta untuk maksimum baris da
#define mak 100n kolom

```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>10</b>
	<b>GABUNGAN FUNGSI DENGAN ARRAY</b>	

```

void isi (int A[][mak],int brs,int klm);
void tampil(int A[][mak],int brs,int klm) ;
void tambah (int A[][mak],int B[][mak],int C[][mak],int
brs,int klm);

int main()
{



int M[100][100],N[100][100],X[100][100],b,k;
printf("Masukkan jumlah baris matrik  = "); scanf("%i",&b);
printf("Masukkan jumlah kolom matrik  = "); scanf("%i",&k);
isi(M,b,k);
printf("\n");
isi(N,b,k);
printf("\n\n Hasil matriks 1 = \n");
tampil(M,b,k);
printf("\n Hasil matriks 2 = \n");
tampil (N,b,k);
tambah(M,N,X,b,k);
printf("Hasil penambahan = \n");
tampil(X,b,k);
return 0;
}

//deklarasi fungsi isi matriks
void isi(int A[][mak],int brs,int klm)
{
int i,j;
for (i=0;i<brs;i++)
for (j=0;j<klm;j++)
{
printf ("masukkan baris ke %i klm ke %i = ",i+1,j+1);
scanf("%i",&A[i][j]);
}
}

//deklarasi fungsi tampil matriks
void tampil(int A[][mak],int brs,int klm)
{
int i,j;
for(i=0;i<brs;i++)
{
for (j=0;j<klm;j++)
printf("%4i",A[i][j]);
printf("\n");
}
}

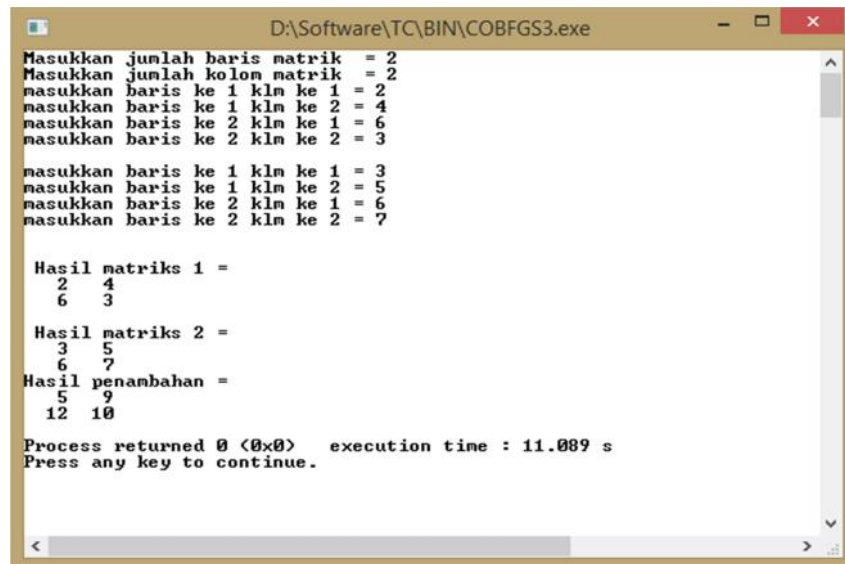
```



	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>10</b>
	<b>GABUNGAN FUNGSI DENGAN ARRAY</b>	

```
// deklarasi fungsi penjumlahan matriks
void tambah (int A[][mak],int B[][mak],int C[][mak],int
brs,int klm)
{
int i,j;
for(i=0;i<brs;i++)
for(j=0;j<klm;j++)
C[i][j]=A[i][j]+B[i][j];
}
```

### Output Program



```
D:\Software\TC\BIN\COBFGS3.exe
Masukkan jumlah baris matrik = 2
Masukkan jumlah kolom matrik = 2
masukkan baris ke 1 klm ke 1 = 2
masukkan baris ke 1 klm ke 2 = 4
masukkan baris ke 2 klm ke 1 = 6
masukkan baris ke 2 klm ke 2 = 3

masukkan baris ke 1 klm ke 1 = 3
masukkan baris ke 1 klm ke 2 = 5
masukkan baris ke 2 klm ke 1 = 6
masukkan baris ke 2 klm ke 2 = 7

Hasil matriks 1 =
2 4
6 3

Hasil matriks 2 =
3 5
6 7
Hasil penambahan =
5 9
12 10



Process returned 0 (0x0)   execution time : 11.089 s
Press any key to continue.
```

## 7. Latihan

- Buatlah **program pengurutan Bubble Sort menggunakan fungsi** untuk mengurutkan angka dari kecil ke besar.
- Buatlah **program menyisipkan angka menggunakan fungsi**
- Buatlah **program yang mengalikan dua buah matriks**, dimana setiap proses dibuat dalam blok fungsi.

## 8. Keselamatan dan Kesehatan Kerja (K3)

- Periksalah komputer yang anda gunakan, apakah terjadi kerusakan atau tidak. Jika terjadi kerusakan, segera melapor pada Dosen ataupun Teknisi Labor.

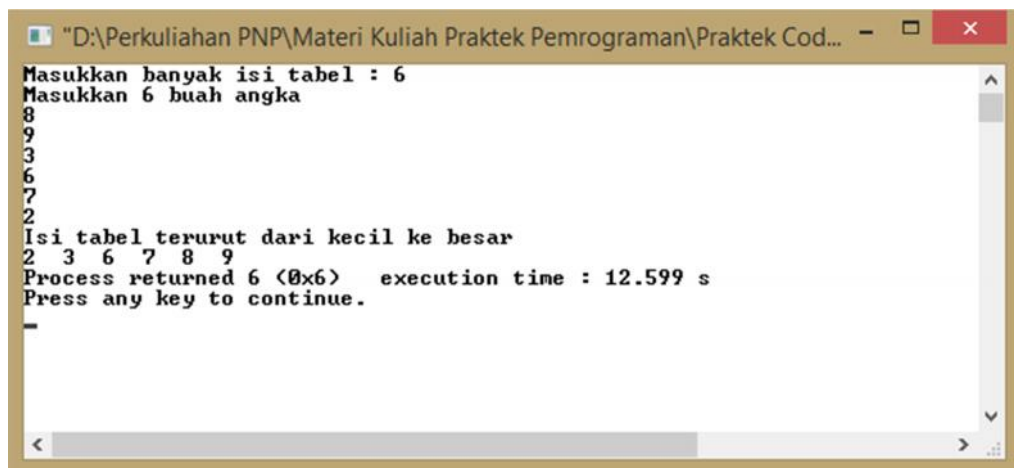
	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>10</b>
	<b>GABUNGAN FUNGSI DENGAN ARRAY</b>	

- b. Jika akan melakukan peng-copy-an data, pastikan komputer ataupun alat penyimpanan data Anda terbebas dari virus.
- c. Selalu matikan komputer setiap Anda selesai praktikum.
- d. Jangan meninggalkan sampah apapun di ruangan labor komputer.

## 9. Hasil Pengujian dan Pembahasan

Selesaikanlah soal latihan dan kemudian cocokan hasil output program Anda dengan hasil program berikut ini. Jalankan juga program Anda untuk nilai-nilai lainnya.

- a. Program pengurutan Bubble Sort menggunakan fungsi





```

"D:\Perkuliahan PNP\Materi Kuliah Praktek Pemrograman\Praktek Cod...
Masukkan banyak isi tabel : 6
Masukkan 6 buah angka
8
9
3
6
7
2
Isi tabel terurut dari kecil ke besar
2 3 6 7 8 9
Process returned 6 (0x6) execution time : 12.599 s
Press any key to continue.

```

- b. Program menyisipkan angka menggunakan fungsi



	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>10</b>
	<b>GABUNGAN FUNGSI DENGAN ARRAY</b>	

c. Membuat program yang memproses fungsi dengan parameter array

### 11. Lembar Penilaian

Item Penilaian	Bobot	Nilai	Nilai X Bobot
Persiapan	5%		
Proses, Contoh, Cara melakukan pekerjaan, sistematika langkah kerja, kemandirian, partisipasi,	25%		
Sikap dan keselamatan kerja, Contoh sikap duduk, sikap menggunakan alat kerja	10%		
Hasil/Data/Produk; Program, Logika program, Kerapian Program, Tampilan Output	50%		
Waktu	10%		
Total	100%		



### 12. Daftar Pustaka

Hartono, J. 2001. *Konsep Dasar Bahasa Pemrograman C*. Andi Offset Yogyakarta

Kelley, Al and Pohl, Ira. 2003. *C by Dissection: The Essentials of C Programming*. Addison-Weasley.

Khannedy, Eko Kurniawan. 2007. *Diktat Pemrograman C*. Unikom. Bandung.

Liem, Inggriani. 2003. *Catatan Singkat Bahasa C*. Jurusan Teknik Informatika. Institut Teknologi Bandung.

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>11</b>
	<b>GABUNGAN FUNGSI DENGAN TIPE BENTUKAN</b>	

### 1. Judul : Gabungan Fungsi Tipe Bentukkan

2. Jumlah Jam/Pertemuan : 3 jam x 1 pertemuan = 3 jam praktek

### 3. Tujuan Praktek

- Mahasiswa mampu menjelaskan kapan fungsi ataupun prosedur digunakan pada proses tipe bentukkan.
- Mahasiswa mampu membuat program yang mengaplikasikan prosedur serta fungsi pada tipe bentukkan.

### 4. Teori Dasar.

Pada dasarnya pengaplikasian prosedur maupun fungsi pada tipe (data) bentukkan tidak ada bedanya dengan tipe data dasar. Hal ini dikarenakan setelah didefinisikan, sebuah tipe bentukkan memiliki peran dan kegunaan yang sama dengan tipe data dasar.

Sebagai contoh, kita ambil tipe bentukkan POINT yang terdiri atas dua buah bilangan bulat yang mewakili nilai absis dan ordinat sebuah titik. Contoh prosedur dan fungsi yang dapat dibuat pada tipe bentukkan POINT ini antara lain :

- function MakePOINT (X : integer; Y : integer)    POINT**



{ Membentuk sebuah POINT dari komponen-komponennya }

```
Point MakePoint(int a,int b)
```

```
{
    Point P;
    P.X = a;
    P.Y = b;
    return(P);
}
```

- procedure BacaPOINT (output P : POINT)**

{ Membaca nilai absis dan ordinat dari keyboard dan membentuk POINT P berdasarkan dari nilai absis dan ordinat tersebut }

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>11</b>
	<b>GABUNGAN FUNGSI DENGAN TIPE BENTUKAN</b>	

```
void BacaPoint(Point *P)
{
    int x,y;
    printf("Input Point (X Y): ");
    scanf("%d %d",&x,&y);
    (*P)=MakePoint(x,y);
}
```

Dari contoh di atas terlihat bahwa penggunaan prosedur dan fungsi pada tipe bentukan tidak berbeda jauh dengan penggunaannya pada tipe data biasa.

## 5. Bahan/Peralatan yang digunakan

- Komputer
- Aplikasi CodeBlocks

## 6. Langkah Kerja



- Jalankan program CodeBlocks.
- Buatlah program berikut di komputer anda. Compile dan jalankan (Run). Perhatikan hasil keluarannya. Jalankan program dan analisa hasilnya.
  - Program yang mengaplikasikan konsep prosedur dan fungsi pada tipe bentukan POINT.

```
/* Nama File : point_func.c
   Pembuat   : tuliskan nama dan no.bp anda
   Tgl pembuatan: tuliskan tanggal hari ini
   Deskripsi  : penggunaan fungsi dan prosedur pada tipe
   bentukan*/

#include "stdio.h"
//pendefinisian boolean
#define true 1
#define false 0
#define boolean unsigned char

typedef struct{
    int X; //absis
    int Y; //ordinat
}Point;

Point MakePoint(int X,int Y);
```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>11</b>
	<b>GABUNGAN FUNGSI DENGAN TIPE BENTUKAN</b>	

```

void BacaPoint(Point *P);
void Tulis(Point P);
int GetAbsis(Point P);
void SetOrdinat(Point *P,int newY);
boolean EQP(Point P1, Point P2);
boolean IsOrigin(Point P);
Point MirrorOf(Point p,boolean sbx,boolean sby);

void main()
{
    Point P1,P2,P3,P4,P5;



    P1=MakePoint(4,4); printf(" P1 = "); Tulis(P1);
    P2=MakePoint(-4,4); printf(" P2 = "); Tulis(P2);
    P3=MakePoint(4,-4); printf(" P3 = "); Tulis(P3);
    P4=MakePoint(-4,-4); printf(" P4 = "); Tulis(P4);
    BacaPoint(&P5); printf(" P5 = "); Tulis(P5);
    printf(" GetAbsis P1 : %d\n",GetAbsis(P1));
    printf(" GetAbsis P2 : %d\n",GetAbsis(P2));
    printf(" GetAbsis P5 : %d\n",GetAbsis(P5));
    SetOrdinat(&P1,2);
    printf(" P1 stlh diset ordinat dengan 2      : ");
    Tulis(P1);
    P5=MakePoint(-4,4);
    printf(" P5 stlh diset -4,4 dgn MakePoint   : ");
    Tulis(P5);
    printf(" Equal P1 P2 ? %d\n",EQP(P1,P2));
    printf(" Equal P2 P5 ? %d\n",EQP(P2,P5));
    printf(" Origin P1 ? %d\n",IsOrigin(P1));
    P3= MirrorOf(P3, true, false);
    printf(" Miror P3 terhadap SbX : ");Tulis(P3);
    P4= MirrorOf(P4, false, true);
    printf(" Miror P4 terhadap Sby : ");Tulis(P4);
}

Point MakePoint(int a,int b)
{
    Point P;
    P.X = a;
    P.Y = b;
    return(P);
}

void BacaPoint(Point *P)
{
    int x,y;
    printf("Input Point (X Y): ");
    scanf("%d %d",&x,&y);
    (*P)=MakePoint(x,y);
}

void Tulis(Point P)
{

```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>11</b>
	<b>GABUNGAN FUNGSI DENGAN TIPE BENTUKAN</b>	



```

        printf("(%d,%d)\n",P.X,P.Y);
    }
    /*Selektor*/
    int GetAbsis(Point P)
    {
        return P.X ;
    }
    void SetOrdinat(Point *P,int newY)
    {
        (*P).Y=newY;
    }
    boolean EQP(Point P1, Point P2)
    {
        boolean same;
        if ((P1.X==P2.X ) && (P1.Y==P2.Y))
        {
            same = true;
        }
        else
            same = false;
        return same;
    }
    boolean IsOrigin(Point P)
    {
        boolean ori;
        if ((P.X==0) && (P.Y==0))
        {
            ori = true;
        }
        else
            ori = false;
        return ori;
    }

    Point MirrorOf(Point p,boolean sbx,boolean sby)
    {
        Point mirror;
        if (sbx == true)
        {
            mirror.X = p.X;
            mirror.Y = p.Y * -1;
        }
        if (sby == true)
        {
            mirror.X = p.X * -1;
            mirror.Y = p.Y;
        }
        return mirror;
    }

```



	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>11</b>
	<b>GABUNGAN FUNGSI DENGAN TIPE BENTUKAN</b>	

### Output Program

```

D:\KERJAAN\Kelas_NoBP\point_func.exe
P1 = (4,4)
P2 = (-4,4)
P3 = (4,-4)
P4 = (-4,-4)
Input Point (X Y): 10 -5
P5 = (10,-5)
GetAbsis P1 : 4
GetAbsis P2 : -4
GetAbsis P5 : 10
P1 stlh diset ordinat dengan 2 : (4,2)
P5 stlh diset -4,4 dgn MakePoint : (-4,4)
Equal P1 P2 ? 0
Equal P2 P5 ? 1
Origin P1 ? 0
Miror P3 terhadap SbX : (4,4)
Miror P4 terhadap Sby : (4,-4)

Process returned 7 (0x7)   execution time : 4.764 s
Press any key to continue.

```

- b) Program yang mengaplikasikan konsep prosedur fungsi pada tipe bentukan JAM.

```

/* Nama File : jam_func.c
   Pembuat   : tuliskan nama dan no.bp anda
   Tgl pembuatan : tuliskan tanggal hari ini
   Deskripsi  : prosedur dan fungsi pada JAM */

#include "stdio.h"

typedef struct
{
    long int HH,MM,SS;
}jam;



/*--PROTOTYPE FUNGSI DAN PROSEDUR YANG DIGUNAKAN--*/
void ResetJam (jam *J);
/*mengisi sebuah jam J dengan 00:00:00 */

void TulisJam (jam J);
/*menulis sebuah jam dengan format hh:mm:ss*/



long int JamToDetik (jam J);
/*konversi jam ke detik*/

jam DetikToJam (long int d);

```

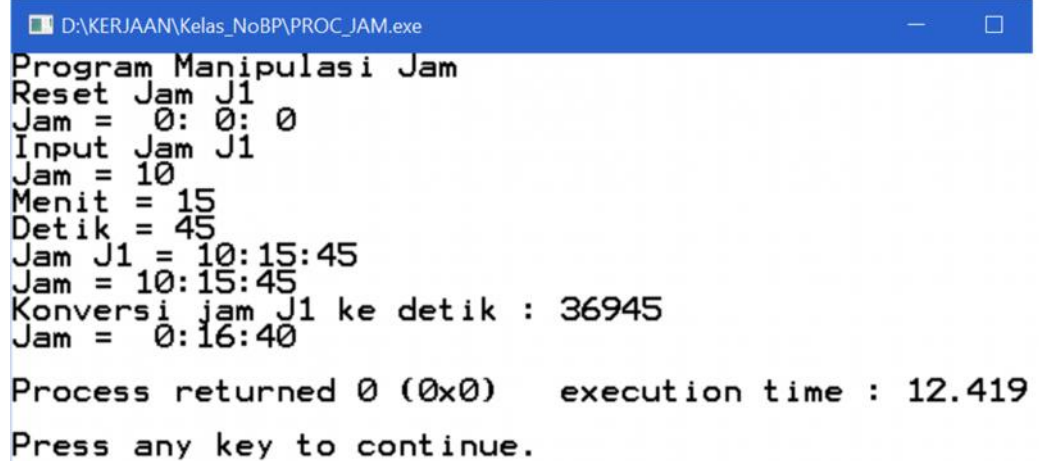
	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>11</b>
	<b>GABUNGAN FUNGSI DENGAN TIPE BENTUKAN</b>	

<pre> /*konversi detik ke jam*/  /*--PROGRAM UTAMA---*/ void main() {     jam J1,J2;     long int det=1000;     clrscr();     printf("Program Manipulasi Jam\n");     printf("Reset Jam J1\n");     ResetJam(&amp;J1);     TulisJam(J1);     printf("Input Jam J1\n");     printf("Jam = ");scanf("%ld",&amp;J1.HH);     printf("Menit = ");scanf("%ld",&amp;J1.MM);     printf("Detik = ");scanf("%ld",&amp;J1.SS);     printf("Jam J1 = %ld:%ld:%ld\n",J1.HH,J1.MM,J1.SS);     TulisJam(J1);     printf("Konversi jam J1 ke detik : %ld\n",JamToDetik(J1));     J2=DetikToJam(det);     TulisJam(J2); }  /*--BODY PROTOTYPE--*/ void ResetJam (jam *J) /*mengisi sebuah jam J dengan 00:00:00 */ {     (*J).HH=0;     (*J).MM=0;     (*J).SS=0; }  void TulisJam (jam J) /*menulis sebuah jam dengan format hh:mm:ss*/ {     printf("Jam = %2ld:%2ld:%2ld\n",J.HH,J.MM,J.SS); }  long int JamToDetik (jam J) /*konversi jam ke detik*/ {     return((J.HH*3600)+(J.MM*60)+J.SS); }  jam DetikToJam (long int d) /*konversi detik ke jam*/ {     //kamus lokal     jam J;     int sisa;     //program </pre>	
---	--

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>11</b>
	<b>GABUNGAN FUNGSI DENGAN TIPE BENTUKAN</b>	

```
J.HH=(d/3600)%24;
sisad=d%3600;
J.MM=sisad/60;
J.SS=sisad%60;
return (J);
}
```

### Output Program



```
D:\KERJAAN\Kelas_NoBP\PROC_JAM.exe
Program Manipulasi Jam
Reset Jam J1
Jam = 0: 0: 0
Input Jam J1
Jam = 10
Menit = 15
Detik = 45
Jam J1 = 10:15:45
Jam = 10:15:45
Konversi jam J1 ke detik : 36945
Jam = 0:16:40



Process returned 0 (0x0)   execution time : 12.419
Press any key to continue.
```

## 7. Latihan

Buatlah program yang mengaplikasikan konsep fungsi dan prosedur pada tipe bentukan JAM (spesifikasi fungsi dan prosedur diberikan saat praktikum).

## 8. Keselamatan dan Kesehatan Kerja (K3)

- Periksalah komputer yang anda gunakan, apakah terjadi kerusakan atau tidak. Jika terjadi kerusakan, segera melapor pada Dosen ataupun Teknisi Labor.
- Jika akan melakukan peng-copy-an data, pastikan komputer ataupun alat penyimpanan data Anda terbebas dari virus.
- Selalu matikan komputer setiap Anda selesai praktikum.
- Jangan meninggalkan sampah apapun di ruangan labor komputer.

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>11</b>
	<b>GABUNGAN FUNGSI DENGAN TIPE BENTUKAN</b>	

## 9. Hasil Pengujian dan Pembahasan

Slesaikanlah soal latihan dan kemudian cocokan hasil output program Anda dengan hasil program berikut ini. Jalankan juga program Anda untuk nilai-nilai lainnya.

a. Prosedur dan fungsi pada tipe bentukan JAM

```



D:\KERJAAN\PNP\OBSHEET 2016\Program\jam.exe
Test MakeJam : 23:45:50
Jam J1 : 1:0:0
Jam J2 : 2:0:0
GetJam J1 : 1
GetMinute J1 : 0
GetSecond J1 : 0
Jam ke Detik J1 : 3600
Jam ke Detik J2 : 7200
Detik ke Jam J1 : 1:0:0
Detik ke Jam J2 : 2:0:0
J1 == J2 : 0
J1 != J2 : 1
J1 < J2 : 1
J1 > J2 : 0
J1 + J2 : 3:0:0
J1 - J2 : 23:0:0
Next 1 detik J1 : 1:0:1
Next 1 detik J2 : 2:0:1
Next 15 detik J1 : 1:0:15
Next 15 detik J2 : 2:0:15
Prev 1 detik J1 : 0:59:59
Prev 1 detik J2 : 1:59:59
Prev 15 detik J1 : 0:59:45
Prev 15 detik J2 : 1:59:45
Durasi : 3600 detik
Process returned 19 (0x13)    execution time : 0.102 s
Press any key to continue.

```

## 10. Kesimpulan

Setelah menyelesaikan topik ini, mahasiswa mampu:

- Menjelaskan kapan fungsi dan prosedur digunakan pada proses tipe bentukan.
- Membuat program yang mengaplikasikan konsep prosedur dan fungsi pada tipe bentukan menggunakan bahasa C.

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>11</b>
	<b>GABUNGAN FUNGSI DENGAN TIPE BENTUKAN</b>	

### 11. Lembar Penilaian



Item Penilaian	Bobot	Nilai	Nilai X Bobot
Persiapan	5%		
Proses, Contoh, Cara melakukan pekerjaan, sistematika langkah kerja, kemandirian, partisipasi,	25%		
Sikap dan keselamatan kerja, Contoh sikap duduk, sikap menggunakan alat kerja	10%		
Hasil/Data/Produk; Program, Logika program, Kerapian Program, Tampilan Output	50%		
Waktu	10%		
Total	100%		

### 12. Daftar Pustaka

Kelley, Al and Pohl, Ira. 2003. *C by Dissection: The Essentials of C Programming*. Addison-Weasley.

Khannedy, Eko Kurniawan. 2007. Diktat Pemrograman C. Unikom. Bandung.

Liem, Inggriani. 2003. Catatan Singkat Bahasa C. Jurusan Teknik Informatika. Institut Teknologi Bandung.

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>12</b>
	<b>OPERASI FILE</b>	

### 1. Judul : Operasi File

### 2. Jumlah Jam/Pertemuan : 3 jam x 1 pertemuan = 3 jam praktek

### 3. Tujuan Praktek

- Mahasiswa mampu menjelaskan konsep file pada pemrograman.
- Mahasiswa mampu mendefinisikan, mendeklarasikan, membuka, mengakses, serta menutup file menggunakan bahasa C.
- Mahasiswa mampu membuat program yang melakukan operasi file menggunakan bahasa C.

### 4. Teori Dasar

File adalah sebuah organisasi dari sejumlah record. Masing-masing record bisa terdiri dari satu atau beberapa field. Setiap field terdiri dari satu atau beberapa byte.



#### **PENDEKLARASIAN VARIABEL FILE**

- ✓ Sebuah file dalam bahasa C dideklarasikan variabelnya dengan nama tipe data FILE (harus huruf besar).
- ✓ Nama variable yang bertipe data FILE tersebut harus diawali dengan lambing “bintang” (\*)

```
FILE *fp;
```

#### **MEMBUKA FILE**

- ✓ Untuk membuka atau mengaktifkan file, fungsi yang digunakan adalah fungsi fopen().
- ✓ Prototipe fungsi fopen() ada di header fungsi “stdio.h”

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>12</b>
	<b>OPERASI FILE</b>	

- ✓ Jika lokasi file yang akan dibuka tidak dituliskan dengan jelas pada perintah fopen, maka lokasi penyimpanan file yang akan dibuka HARUS SAMA dengan lokasi file program.

- ✓ Bentuk umum :

```
file *fopen(char *namafile, char *mode);
```

Keterangan :

namafile adalah nama dari file yang akan dibuka/diaktifkan.

mode adalah jenis operasi file yang akan dilakukan terhadap file.



#### **Jenis-jenis operasi file:**

- ✓ r : menyorakan file hanya dapat dibaca (file harus sudah ada)
- ✓ w : menyatakan file baru akan dibuat/diciptakan (file yang sudah ada akan dihapus)
- ✓ a : untuk membuka file yang sudah ada dan akan dilakukan proses penambahan data (jika file belum ada, otomatis akan dibuat)
- ✓ r+ : untuk membuka file yang sudah ada dan akan dilakukan proses pembacaan dan penulisan.
- ✓ w+ : untuk membuka file dengan tujuan untuk pembacaan atau penulisan. Jika file sudah ada, isinya akan dihapus.
- ✓ a+ : untuk membuka file, dengan operasi yang akan dilakukan berupa perekaman maupun pembacaan. Jika file sudah ada, isinya akan dihapus.
- ✓ Contoh :

```
fp = fopen("coba.txt", "w");
```

#### **MENUTUP FILE**

- ✓ Untuk menutup file, fungsi yang digunakan adalah fclose().
- ✓ Prototype fungsi fclose() ada di header file "stdio.h"

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>12</b>
	<b>OPERASI FILE</b>	

✓ Bentuk Umum :

```
fclose(fp);
```

atau

```
fcloseall(void);
```

## 5. Bahan/Peralatan yang digunakan

- Komputer
- Aplikasi CodeBlocks

## 6. Langkah Kerja

- Jalankan aplikasi CodeBlocks.
- Buatlah beberapa program berikut di komputer anda. Compile dan jalankan (Run). Perhatikan hasil keluarannya. Jalankan program dan analisa hasilnya.
  - Program yang menginputkan sejumlah string dan menuliskannya ke dalam sebuah file. Program akan berhenti meminta input string ketika tidak diinputkan apa-apa (langsung ENTER).



```
/* Nama File : op_file1.c
Pembuat   : tuliskan nama dan no.bp anda
Tgl pembuatan : tuliskan tanggal hari ini
Deskripsi : menuliskan ke dalam sebuah file*/

#include <stdio.h>
#include <String.h>

void main()
{
    FILE *f;
    char st[81]="";
    f=fopen("coba.txt","w");
    if (f!=NULL)
    {
        do
        {
            printf("Masukan string : ");gets(st);
            fprintf(f,"%s\n",st); //menuliskan ke file
        }while (strlen(st)>0);
        fclose(f);
        printf("File selesai di buat");
    }
    else
    {

```



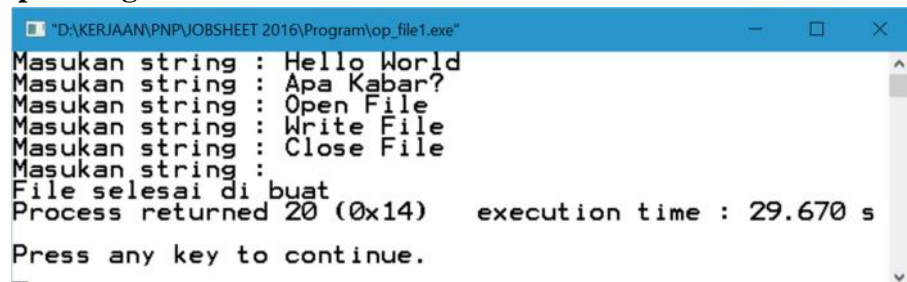
	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>12</b>
	<b>OPERASI FILE</b>	

```

        printf("File gagal dibuat");
    }
}

```

### Output Program

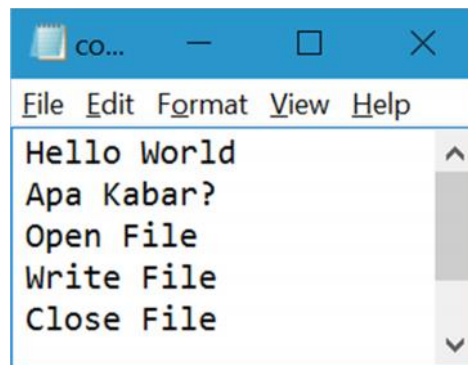


```

Masukan string : Hello World
Masukan string : Apa Kabar?
Masukan string : Open File
Masukan string : Write File
Masukan string : Close File
Masukan string : 
File selesai di buat
Process returned 20 (0x14)   execution time : 29.670 s
Press any key to continue.

```

Buka file explorer di komputer anda, masuk ke folder tempat program anda disimpan kemudian cari file dengan nama “coba.txt”. Buka file “coba.txt” tersebut dan periksa isinya apakah sama dengan yang anda tulis.



```

Hello World
Apa Kabar?
Open File
Write File
Close File

```

- b) Program yang membaca file “coba.txt” yang dihasilkan program sebelumnya dan menuliskan isinya ke layar.


```

/* Nama File : op_file2.c
   Pembuat   : tuliskan nama dan no.bp anda
   Tgl pembuatan : tuliskan tanggal hari ini
   Deskripsi : membuka file dan menuliskan isinya ke layar*/

#include <stdio.h>
#include <String.h>

void main()
{

```

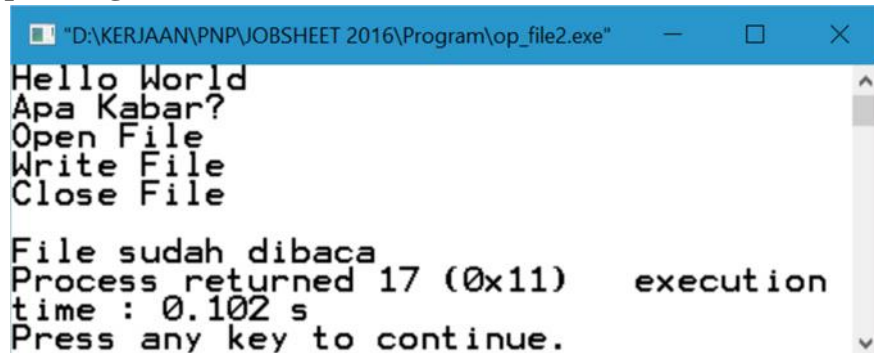
	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>12</b>
	<b>OPERASI FILE</b>	

```

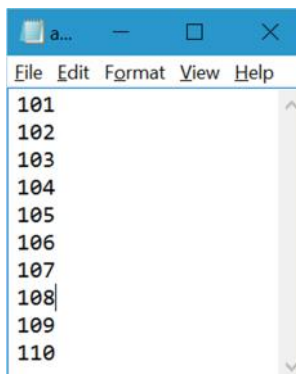
FILE *f;
char st[81]="";
f=fopen("coba.txt","r");
if (f!=NULL)
{
    while (fgets(st,80,f)!=NULL)
    {
        printf("%s",st);
    }
    fclose(f);
    printf("File sudah dibaca");
}
else
{
    printf("File gagal dibaca");
}
}



```

### Output Program



- c) Program yang membaca file “angka.txt” kemudian menyimpan isinya dalam array dan menampilkan ke layar.



	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>12</b>
	<b>OPERASI FILE</b>	

```

/* Nama File : op_file3.c
   Pembuat   : tuliskan nama dan no.bp anda
   Tgl pembuatan : tuliskan tanggal hari ini
   Deskripsi  : membuka file, membaca isi dan menyimpan ke
array*/

#include <stdio.h>

void main()
{
    FILE *file = NULL;
    int angka[10];
    int i;



    file = fopen("angka.txt", "r");

    if (file != NULL)
    {
        for(i=0;i<=9;i++)
        {
            fscanf(file, "%d", &angka[i]);
        }

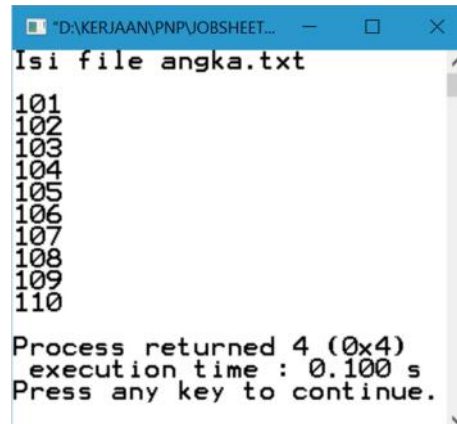
        fclose(file);
    }
    else
    {
        perror("Error on opening the input file");
    }

    printf("Isi file angka.txt\n\n");
    for(i=0;i<10;i++)
        printf("%d\n", angka[i]);
}

```

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>12</b>
	<b>OPERASI FILE</b>	

### Output Program



```



D:\KERJAAN\PNP\JOBSHEET...
Isi file angka.txt
101
102
103
104
105
106
107
108
109
110
Process returned 4 (0x4)
execution time : 0.100 s
Press any key to continue.

```

### 7. Latihan

Buatlah sebuah file txt yang berisi angka 1 dan 0 dengan banyak angka tidak diketahui. Buatlah sebuah program yang membaca file tersebut kemudian menghitung ada berapa banyak angka di file tersebut serta berapa banyak angka 1 saja. Periksalah angka 1 ada di posisi berapa saja, simpan ke array dan kedalam sebuah file txt yang baru.

Contoh file input :

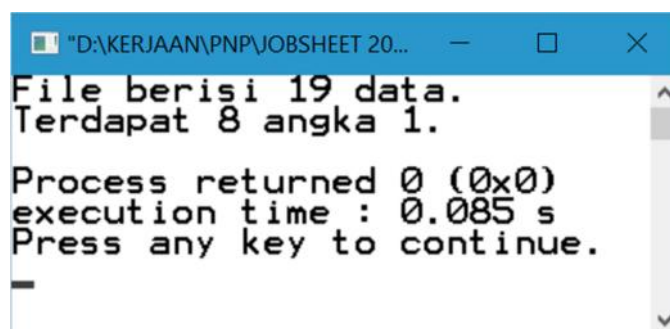
	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>12</b>
	<b>OPERASI FILE</b>	





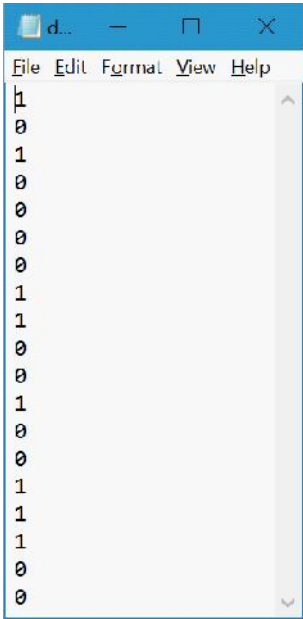
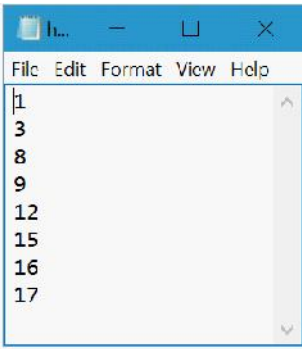
## 8. Keselamatan dan Kesehatan Kerja (K3)

- Periksalah komputer yang anda gunakan, apakah terjadi kerusakan atau tidak. Jika terjadi kerusakan, segera melapor pada Dosen ataupun Teknisi Labor.
- Jika akan melakukan peng-copy-an data, pastikan komputer ataupun alat penyimpanan data Anda terbebas dari virus.
- Selalu matikan komputer setiap Anda selesai praktikum.
- Jangan meninggalkan sampah apapun di ruangan labor komputer.

## 9. Hasil Pengujian dan Pembahasan



	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>12</b>
	<b>OPERASI FILE</b>	

<p>File input</p> 	<p>Posisi angka 1 (hasil.txt)</p> 
--	--



## 10. Kesimpulan

Setelah menyelesaikan topik ini, mahasiswa mampu:

- Menjelaskan konsep file pada pemrograman.
- Mendefinisikan, mendeklarasikan, membuka, mengakses, serta menutup file menggunakan bahasa C.
- Membuat program yang melakukan operasi file menggunakan bahasa C.

## 11. Lembar Penilaian

Item Penilaian	Bobot	Nilai	Nilai X Bobot
Persiapan	5%		
Proses, Contoh, Cara melakukan pekerjaan, sistematika langkah kerja, kemandirian, partisipasi,	25%		
Sikap dan keselamatan kerja, Contoh sikap duduk, sikap menggunakan alat kerja	10%		
Hasil/Data/Produk;	50%		

	<b>POLITEKNIK NEGERI PADANG</b> <b>PROGRAM STUDI : MANAJEMEN INFORMATIKA</b>	
	<b>JUDUL PRAKTEK: PEMROGRAMAN LANJUT</b>	
	<b>TOPIK KEGIATAN PRAKTEK:</b>	<b>12</b>
	<b>OPERASI FILE</b>	

Program, Logika program, Kerapian Program, Tampilan Output			
Waktu	10%		
Total	100%		

## 12. Daftar Pustaka

Kelley, Al and Pohl, Ira. 2003. *C by Dissection: The Essentials of C Programming*. Addison-Weasley.

Khannedy, Eko Kurniawan. 2007. Diktat Pemrograman C. Unikom. Bandung.

Liem, Inggriani. 2003. Catatan Singkat Bahasa C. Jurusan Teknik Informatika. Institut Teknologi Bandung.

<http://www.phy225.dept.shef.ac.uk/mediawiki/index.php/>. *Arrays, Dynamic Array Allocation - Programming in C*. Tanggal akses 17 April 2014