

# PRAKTIKUM 27

---

## Pointer 4

---

### A. TUJUAN PEMBELAJARAN

1. Memahami tentang Pointer dalam Fungsi
2. Memahami tentang Pointer sebagai Parameter Fungsi
3. Memahami tentang Pointer sebagai Keluaran Fungsi

### B. DASAR TEORI

#### Pointer dalam Fungsi

Pointer dan kaitannya dengan fungsi yang akan dibahas berikut meliputi :

- Pointer sebagai parameter fungsi
- Pointer sebagai keluaran fungsi

#### Pointer Sebagai Parameter Fungsi

- Penerapan pointer sebagai parameter yaitu jika diinginkan agar nilai suatu variabel internal dapat diubah oleh fungsi yang dipanggil.
- Sebagai contoh dapat dilihat pada fungsi berikut.

```
void naikkan_nilai (int *x, int *y){  
    *x = *x + 2;  
    *y = *y + 2;  
}
```

- Fungsi di atas dimaksudkan agar kalau dipanggil, variabel yang berkenaan dengan parameter aktual dapat diubah nilainya, masing-masing dinaikkan sebesar 2. Contoh pemanggilan :

```
naikkan_nilai(&a, &b);
```

- Perhatikan, dalam hal ini variabel **a** dan **b** harus ditulis diawali operator alamat (**&**) yang berarti menyatakan alamat variabel, sebab parameter fungsi dalam pendefinisian berupa pointer.

### Pointer Sebagai Keluaran Fungsi (*return value*)

- Suatu fungsi dapat dibuat agar keluarannya berupa pointer. Misalnya, suatu fungsi menghasilkan keluaran berupa pointer yang menunjuk ke string `nama_bulan`, seperti pada contoh berikut.

```
char *nama_bulan(int n) {
    static char *bulan[]=
        {"Kode bulan salah", "Januari", "Februari", "Maret",
         "April", "Mei", "Juni", "Juli", "Agustus",
         "September", "Oktober", "Nopember", "Desember"};
    return ( (n<1 || n>12) ? bulan[0] : bulan[n] );
}
```

- Pada definisi fungsi di atas,

```
char *nama_bulan()
```

menyatakan bahwa keluaran fungsi `nama_bulan()` berupa pointer yang menunjuk ke obyek char (atau string).

- Dalam fungsi `nama_bulan()`, mula-mula array bernama `bulan` dideklarasikan dan sekaligus diinisialisasi agar menunjuk sejumlah string yang menyatakan nama bulan. Di bagian akhir fungsi, pernyataan

```
return ( (n<1 || n>12) ? bulan[0] : bulan[n] );
```

menyatakan bahwa hasil fungsi berupa pointer yang menunjuk ke

→ string "Kode bulan salah" (`bulan[0]`) jika masukan fungsi `n<1` atau `n>12`

→ `bulan[n]` untuk `n` yang terletak antara 1 sampai dengan 12.

### C. PERCOBAAN

Untuk setiap program di bawah ini,

- gambarkan ilustrasi alokasi memori dari setiap baris pernyataan yang diproses
- perkirakan hasil eksekusinya

## 1. Fungsi dengan argumen berupa pointer.

```
#include <stdio.h>
void naikan_nilai(int *x, int *y);

main() {
    int a = 3;
    int b = 7;

    printf("SEMULA : a = %d b = %d\n", a, b);
    naikan_nilai(&a, &b);
    printf("KINI    : a = %d b = %d\n", a, b);
}

void naikan_nilai(int *x, int *y){
    *x = *x + 2;
    *y = *y + 2;
}
```

## 2. Fungsi dengan keluaran berupa pointer yang menunjuk string.

```
#include <stdio.h>
char *nama_bulan(int n);

main(){
    int bl;

    printf("Bulan 1..12 : ");
    scanf("%d", &bl);
    printf("Bulan ke-%d adalah %s\n", bl, nama_bulan(bl));
}

char *nama_bulan(int n){
    static char *bulan[] = {
        "Ngawur",
        "Januari",
        "Februari",
        "Maret",
        "April",
        "Mei",
        "Juni",
        "Juli",
        "Agustus",
        "September",
        "Oktober",
        "November",
        "Desember"
    };
};
```

```

    return ((n<1||n>12) ? bulan[0] : bulan[n]);
}

```

3. Berikan ilustrasi dan jelaskan apa yang dilakukan oleh program di bawah ini dan tampilkan hasil eksekusinya.

```

#include <stdio.h>

char strA[80] = "A string to be used for demonstration";
char strB[80];

main(){
    char *pA, *pB;

    puts(strA);
    pA = strA;
    puts(pA);
    pB = strB;
    putchar('\n');

    while(*pA != '\0')
        *pB++ = *pA++;

    *pB = '\0';
    puts(strB);
}

```

Proses apakah yang sebenarnya dilakukan pada program tersebut ?

4. Berikan ilustrasi dan jelaskan apa yang dilakukan oleh program di bawah ini

```

#include <stdio.h>
char *my_strcpy(char *, char *);

main(){
    char strA[80]="A string to be used for demonstration";
    char strB[80];

    my_strcpy(strB, strA);
    puts(strB);
}

char *my_strcpy(char *destination, char *source) {
    char *p = destination;
    while (*source != '\0')
        *p++ = *source++;
}

```

```

    *p = '\0';
    return destination;
}

```

5. Bandingkan fungsi `my_strcpy()` di atas dengan fungsi `my_strcpy()` di bawah ini. Berikan penjelasan terhadap perbedaan proses dari kedua fungsi tersebut

```

char *my_strcpy(char dest[], char source[]){
    int i = 0;
    while (source[i] != '\0')
    {
        dest[i] = source[i];
        i++;
    }
    dest[i] = '\0';
    return dest;
}

```

6. Buatlah sebuah program dengan mendefinisikan sebuah fungsi `rotasi()` yang menerima tiga parameter berupa variabel `a`, `b`, dan `c`. Fungsi ini melakukan rotasi sehingga nilai `a` berpindah ke `b`, `b` ke `c` dan nilai `c` ke `a` sekembalinya ke fungsi `main()`.

#### **D. LAPORAN RESMI**

Kumpulkan listing program, ilustrasi alokasi memorinya beserta hasil eksekusinya.