

# **ALGORITMA DAN PEMROGRAMAN I**

## WHILE & ARRAY OF TIPE TERSTRUKTUR

# ROSA ARIANI SUKAMTO

Blog: <http://hariiniadalahhadiah.wordpress.com>

Facebook: <https://www.facebook.com/rosa.ariani.sukamto>

Email: [rosa\\_if\\_itb\\_01@yahoo.com](mailto:rosa_if_itb_01@yahoo.com)

Website: <https://rosa-as.id>



# PERULANGAN - WHILE (1)

Perulangan memiliki kemungkinan berhenti di tengah ketika tujuan perulangan telah terpenuhi

Biasanya memiliki minimal 2 syarat kondisi perulangan:

- Batas maksimal perulangan dilakukan
- Batas jika perulangan berhenti di tengah

```
// inisialisasi
.....
while(kondisi_pengulangan) {

    // proses
    .....
    // iterasi
}
```

# PERULANGAN - WHILE (2)

```
int i;

// inisialisasi + iterasi
for(i=1; i<=9; i++){

    // proses
    .....

}
```

```
int i;

// inisialisasi
i = 1;
while(i <= 9){

    // proses
    .....

    // iterasi
    i = i + 1;

}
```

## PERULANGAN - WHILE (3)

algoritma pencarian pada sebuah *array* bilangan integer apakah di dalam *array* bilangan integer ada angka yang merupakan bilangan ganjil



# PERULANGAN - WHILE (4)

```
#include <stdio.h>

int main() {
    int penghitung;
    int tabInt[5];
    int ketemu;

    for (penghitung=0; penghitung<5;
        penghitung++) {
        printf("masukkan angka:\n");
        scanf("%d",
            &tabInt[penghitung]);
        printf("\n");
    }

    ketemu = 0;
    penghitung = 0;
```

```
while ((ketemu == 0) &&
    (penghitung < 5)) {
        if ((tabInt[penghitung] % 2)
            == 1) {
            ketemu = 1;
        } else {
            penghitung = penghitung +
                1;
        }
    }

    if (ketemu == 0) {
        printf("tidak ada angka
            ganjil\n");
    } else {
        printf("ada angka
            ganjil\n");
    }

    return 0;
}
```

## PERULANGAN - WHILE (5)

Buat algoritma yang mengisi sebuah *array integer* dan menampilkan isi *array*, tapi berhenti menampilkan jika ditemukan angka 999!

# PERULANGAN - WHILE (6)

```
#include <stdio.h>

int main(){
    int tabInt[5];
    int penghitung;
    for(penghitung=0; penghitung<5; penghitung++){
        printf("masukkan angka:\n");
        scanf("%d", &tabInt[penghitung]);
        printf("\n");
    }
    penghitung = 0;
    while((tabInt[penghitung] != 999) && (penghitung < 5)){
        printf("%d\n", tabInt[penghitung]);
        penghitung = penghitung + 1;
    }
    return 0;
}
```



## PERULANGAN - WHILE (7)

**Tampilkan maksimal 3 angka genap yang ada pada sebuah array of integer?**



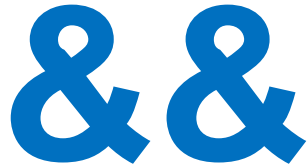
# PERULANGAN - WHILE (8)

```
#include <stdio.h>

int main(){
    int n; scanf("%d", &n);
    int tabInt[n];
    int penghitung;
    for(penghitung=0; penghitung<n; penghitung++){
        printf("masukkan angka:\n");
        scanf("%d", &tabInt[penghitung]);
        printf("\n");
    }
    penghitung = 0;
    int maks = 0;
    while((maks <3) && (penghitung < n)){
        if(tabInt[penghitung] % 2 == 0){
            printf("%d\n", tabInt[penghitung]);
            maks++;
        }
        penghitung = penghitung + 1;
    }
    return 0;
}
```

**SYARAT ABSOLUT BISA ALPRO**

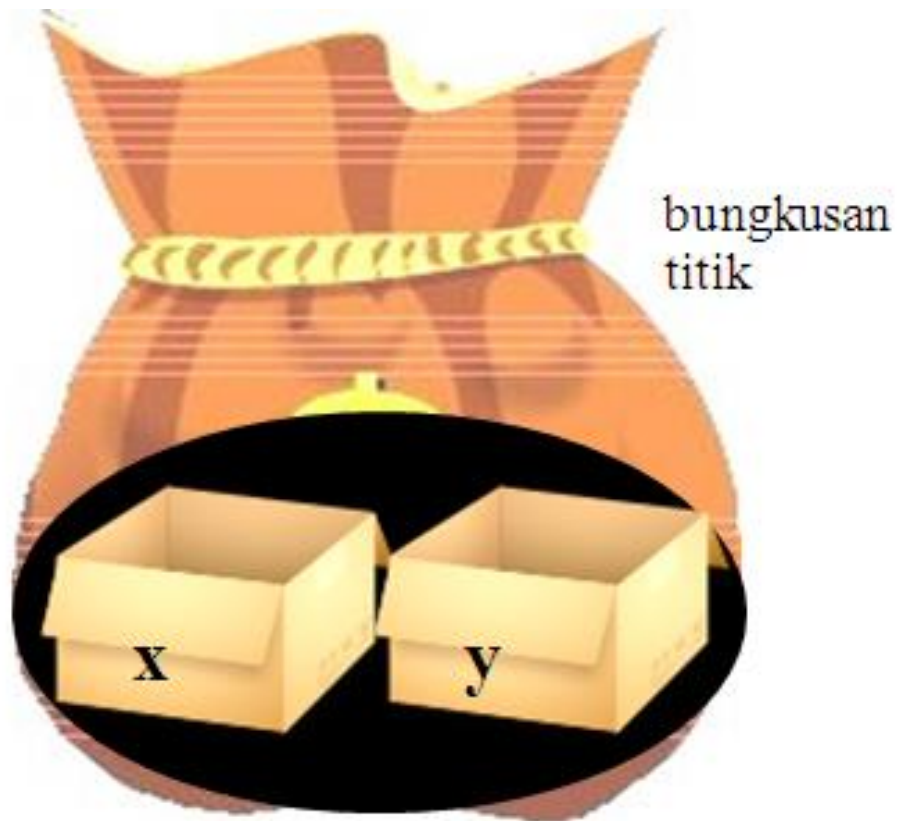
# **Latihan Menyelesaikan Soal**



**Bertanya jika tidak bisa di  
dalam latihan**

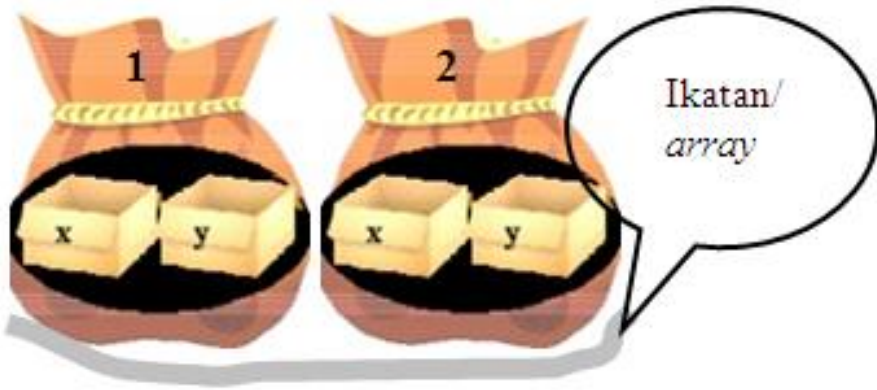


# TIPE TERSTRUKTUR



```
typedef struct{  
    int x;  
    int y;  
}titik;  
  
int main(){  
    titik titik1;  
    titik1.x = 9;  
    .....  
    return 0;  
}
```

# ARRAY OF TIPE TERSTRUKTUR (1)



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

```
int main() {  
    titik ikatan_titik[2];  
    ikatan_titik[0].x = 9;  
    .....  
    return 0;  
}
```

## ARRAY OF TIPE TERSTRUKTUR (2)

**Jumlahkan isi X dan isi Y yang ada di dalam Array of Titik**



```
#include <stdio.h>

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

int main(){
    int n, i, jx = 0, jy = 0;
    scanf("%d", &n);
    titik ikatan_titik[n];
    for(i=0;i<n;i++){
        scanf("%d", &ikatan_titik[i].x);
        scanf("%d", &ikatan_titik[i].y);
    }
    for(i=0;i<n;i++){
        jx = jx + ikatan_titik[i].x;
        jy = jy + ikatan_titik[i].y;
    }
    printf("%d %d\n", jx, jy);
    return 0;
}
```

## ARRAY OF TIPE TERSTRUKTUR (4)

**Tampilkan semua titik yang x-nya ganjil dan y-nya lebih besar dari x**





```
#include <stdio.h>

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

int main(){
    int n, i, jx = 0, jy = 0;  scanf("%d", &n);
    titik ikatan_titik[n];
    for(i=0;i<n;i++){
        scanf("%d", &ikatan_titik[i].x);
        scanf("%d", &ikatan_titik[i].y);
    }
    for(i=0;i<n;i++){
        if((ikatan_titik[i].x % 2 == 1) && (ikatan_titik[i].x <
ikatan_titik[i].y)){
            printf("%d %d\n", ikatan_titik[i].x, ikatan_titik[i].y);
        }
    }
    return 0;
}
```

# MARI MENCOBA UNTUK MASA DEPAN CERIA 😊

- Diberikan sebuah array bungkusan bertipe segitiga (alas, tinggi, sisiMiring), tampilkan maksimal 3 segitiga yang merupakan segitiga siku-siku (gunakan rumus phitagoras untuk memeriksa).
- Diberikan sebuah array bertipe float, tampilkan semua nilai float dimana nilai di depan koma dan nilai di belakang koma adalah genap.
- Diberikan sebuah array bungkusan bertipe pecahan (pembilang, penyebut), tampilkan maksimal 3 pecahan yang merupakan bilangan bulat (pembilang dapat dibagi habis oleh penyebut)
- Diberikan sebuah array bungkusan bertipe pecahan (pembilang dan penyebut), jumlahkan semua pecahan yang merupakan bilangan bulat (hasil bagi pembilang oleh penyebut)

# DAFTAR PUSTAKA

