ALGORIMA DAN PENROGRAMANI

NHILE & ARRAY OF TIPE RESTRUKTUR

ROSA ARIANI SUKAMTO

Blog: http://hariiniadalahhadiah.wordpress.com

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

Email: 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 \leq 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;</pre>
  penghitung++) {
    printf("masukkan angka:\n");
    scanf("%d",
   &tabInt[penghitung]);
    printf("\n");
  ketemu = 0;
  penghitung = 0;
```

```
while((ketemu == 0) &&
   (penghitung < 5)){</pre>
     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++){</pre>
    printf("masukkan angka:\n");
    scanf("%d", &tabInt[penghitung]);
    printf("\n");
  }
 penghitung = 0;
  while((tabInt[penghitung] != 999) && (penghitung < 5)){</pre>
       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++) {</pre>
    printf("masukkan angka:\n");
    scanf("%d", &tabInt[penghitung]);
    printf("\n");
  penghitung = 0;
  int maks = 0;
  while((maks <3) && (penghitung < n)){</pre>
     if(tabInt[penghitung] % 2 == 0){
       printf("%d\n", tabInt[penghitung]);
       maks++;
     }
      penghitung = penghitung + 1;
  return 0;
```

SYARAT ABSOLUT BISA ALPRO

Latihan Menyelesaikan Soal



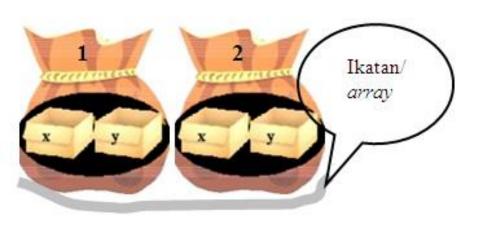
Bertanya jika tidak bisa di dalam latihan

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



PEMROGRAMAN

□ Komentar
□ Tipe Data

Masukan (Input) dan Keluaran (Output)

□ Operator □ Array (Larik)

p Fungsi Matriks dalam Larik (Array) Dua Dimensi

Rekursif
 Pengurutan (Sorting)

Penggabungan Larik
 Pencarian (Searching)
 Arsip Beruntun (Sequential File)

Mesin Abstrak

Algoritma berarti solusi. Ketika orang berbicara mengenai algoritma di bidang pemrograman, maka yang dimaksud adalah solusi dari suatu masalah yang harus dipecahkan dengan menggunakan komputer. Algoritma harus dibuat secara runut agar komputer mengerti dan mampu mengeksekusinya. Analisis kasus sangat dibutuhkan dalam membuat sebuah algoritma, misalnya proses apa saja yang sekiranya dibutuhkan untuk menyelesaikan masalah yang harus diselesaikan. Ketajaman dalam menganalisis sebuah kasus dapat dilatih dengan berlatih menyelesaikan kasus-kasus algoritma, mulai dari yang paling sederhana sampai rumit. Kesabaran sangat dibutuhkan dalam mempelajari algoritma.

Esensi dari belajar algoritma adalah membuat solusi untuk menyelesaikan permasalahan, jadi hasilnya adalah dapat menyelesaikan permasalahan. Maka sangat tidak sesuai dengan esensi jika algoritma dipelajari dengan hanya tahu polapola atau teknik-teknik algoritma tapi tidak mampu secara logika menyelesaikan permasalahan yang berbeda-beda. Maka dari itu sangat dibutuhkan latihan soal dalam mempelajari algoritma dan pemrograman agar logika terlatih untuk membuat solusi dari permasalahan.

Penulis berharap buku ini dapat membantu para pembaca pada umumnya dan mahasiswa, siswa SMP, SMK, SMA khususnya dalam memahami pemrograman secara lebih baik. Dalam buku ini banyak diberikan penggambaran/ilustrasi secara visual agar pembaca lebih mudah dalam memahami isi buku. Selain itu, dalam buku ini juga diberikan bagaimana mengimplementasikan suatu algoritma dalam bahasa algoritmik, bahasa Pascal, bahasa C, C++, dan Java agar pembaca dapat lebih mudah mengimplementasikannya secara langsung.





Pemasaran: BI-OBSES Pasar buku Palasari 82 Bandung 40264 Tel.(022) 7317812 Fax.(022) 7317896 Rosa A.

LOGIKA ALGORITMA **PEMROGRAMAN**

Rosa A. S.

- □ Pendahuluan Algoritma dan Pemrograman
- Komentar ■ Tipe Data
- Masukan (Input) dan Keluaran (Output)
- Operator
- Array (Larik)
- Percabangan / Pemilihan If ■ Perulangan (Looping)
- Prosedur
- Fungsi
- Matriks dalam Larik (Array) Dua Dimensi
- Rekursif
- Pengurutan (Sorting)
- Penggabungan Larik ■ Pencarian (Searching)
- Arsip Beruntun (Sequential File)
- Mesin Abstrak



