

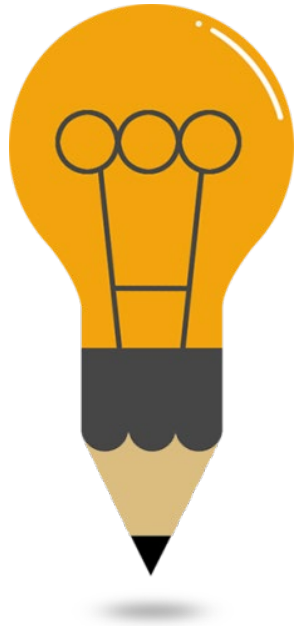
STRUKTUR DATA

Pertemuan 1

(TIM)



Agenda Pertemuan



1

Review Algoritma dan Struktur Data

Ulang materi tingkat I, bagian mana yang perlu dijelaskan kembali?

Review: Dasar Algoritma

- **Algoritma** : Urutan langkah-langkah untuk memecahkan masalah yang disusun secara sistematis dan logis
- Dalam operasi komputer, algoritma merupakan langkah-langkah yang didefinisikan untuk memerintah komputer dalam menyelesaikan suatu masalah
- **Pertimbangan dalam pemilihan algoritma**
 - Algoritma haruslah benar
 - Seberapa baik hasil yang dicapai oleh algoritma tersebut
 - Efisiensi algoritma (waktu dan memori)

Review: Dasar Algoritma

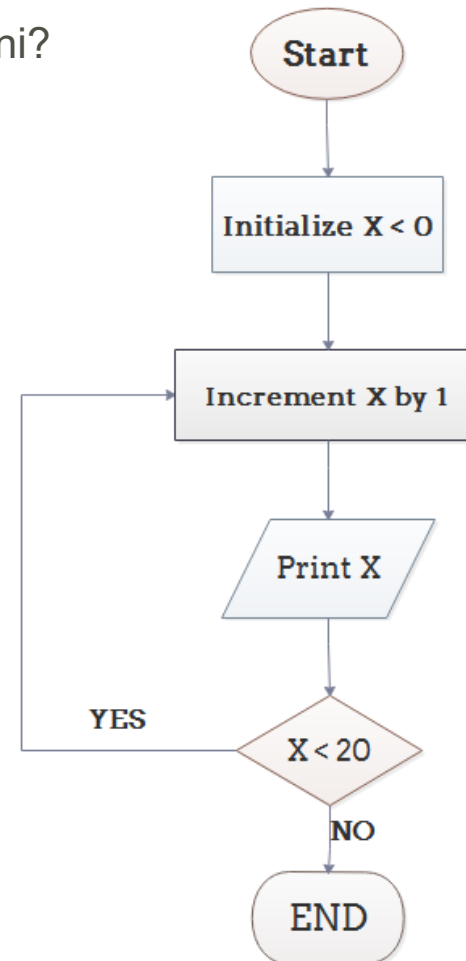
- Program adalah kumpulan "instruksi" komputer (realisasi dari algoritma yang diimplementasikan menggunakan bahasa pemrograman)
- Program = Algoritma + Bahasa Pemrograman
- Pembuatan atau penulisan algoritma tidak tergantung pada bahasa pemrograman apapun
- Notasi algoritma dapat diterjemahkan ke dalam berbagai bahasa pemrograman
- Apapun bahasa pemrogramannya, output yang akan dikeluarkan sama karena algoritmanya sama

Review: Dasar Algoritma

■ Contoh Algoritma: Apa yang dihasilkan algoritma di bawah ini?

■ *Sequence, Selection, Iteration Process*

1. Start
2. Menginisialisasi $X = 0$
3. Menambahkan X dengan 1
4. Tampilkan X
5. Jika X kurang dari 20, maka kembali ke langkah ke-3
6. End



Review: Konsep Dasar Bahasa PASCAL

```
program konstanta;  
uses crt;  
  
const  
    phi=3.14;  
    dollar=13000;  
    nama='Joko';  
  
var  
    situs:string='www.duniailkom.com';  
  
begin  
    clrscr;  
    writeln(phi:4:2);  
    writeln(dollar);  
    writeln(nama);  
    writeln(situs);  
    readln;  
end.
```

1. Judul Program

2. Blok Program atau Badan Program

- Bagian Deklarasi (deklarasi label, konstanta, tipe, variable, prosedur, fungsi)
- Bagian Pernyataan

Review: Konsep Dasar Bahasa PASCAL

■ Konsep Pemilihan

- Pemilihan Sederhana (If-Then)
- Konsep If-then-else
- Else If berjenjang
- Kondisi bersarang

- Konsep Case
- Percabangan Tidak Bersyarat (GOTO)
- Repetition Loops

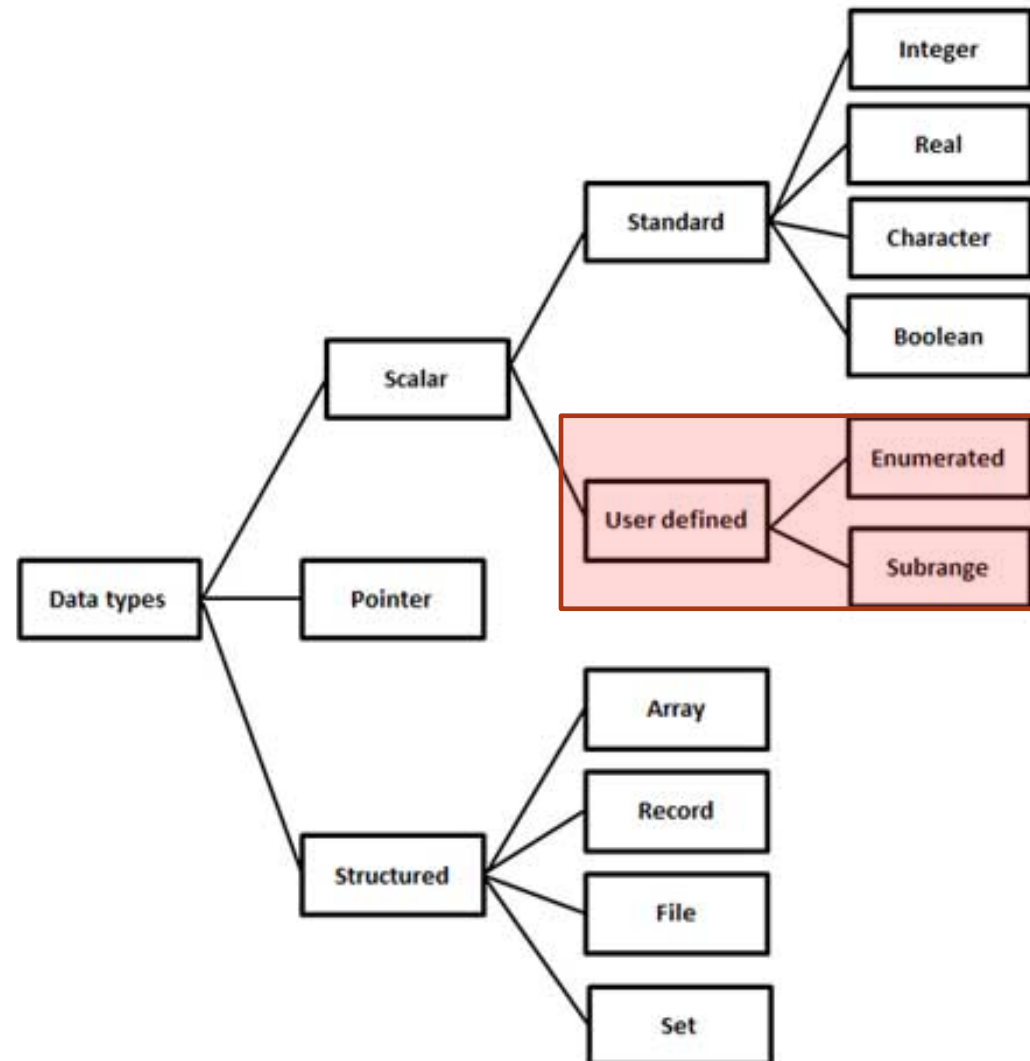
Review: Konsep Dasar Bahasa PASCAL

■ Konsep Pengulangan

- Struktur WHILE - DO
- Struktur REPEAT - UNTIL

- Konsep Case
- Struktur FOR (Menaik / Menurun)
- Nested Loop / Pengulangan bersarang

Review: Tipe Data Pascal



Review: Tipe Data Pascal

■ User defined data type:

- **Enumerated** : Nilai dari variable dituliskan ke dalam daftar. Nilai yang ada di dalam daftar dapat direpresentasikan dalam *small integer* 0 sampai n-1.
- Operasi ORD, PRED dan SUCC terhadap elemen

```
VAR
  BULAN: (JAN, FEB, MAR, APR)
           0      1      2      3
```

- **Subrange** : Nilai yang berada di luar range tidak diperbolehkan

```
VAR
  Month : 1..12; {INTEGER}
  Uppercase:'A'..'Z'; {CHAR}
  Lowercase:'a'..'z';
  Month :=13; {Akan menghasilkan error}
```

Review: Tipe Data Pascal

- Subrange sebagai subset Enumerated

VAR

```
months = (Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec);  
Summer = Apr .. Aug;  
Winter = Oct .. Dec;
```

Review: Array sebagai struktur data

- **Array** adalah sekumpulan variabel yang memiliki **tipe data yang sama** dan dinyatakan dengan **nama yang sama**

- Contoh :

Mahasiswa [1]

Mahasiswa [2]

Mahasiswa [3]

...

- Deklarasi :

```
x : array [1..10] of integer;
```

```
mhs : array [1..10] of array 1..1000] of integer;
```

```
mhs : array [1..10, 1..1000] of integer;
```

Review: Sub Program

■ Pemrograman Modular

- Modul dalam bahasa Pascal di-implementasikan dengan Prosedur dan fungsi
- Prosedur/fungsi dibentuk dengan mengelompokkan sejumlah perintah untuk menyelesaikan tugas tertentu
- Modul sering disebut juga dengan **Sub-Program**

Review: Prosedur

```
Program Judul;  
  
Procedure Bintang;  
Begin  
    write('*');  
End;  
Var i : integer;  
Begin  
    for i:=1 to 9 do bintang;  
    writeln;  
    bintang;  
    write(' Judul ');  
    bintang;  
    writeln;  
    bintang;  
End.
```

Review: Fungsi

```
Program Judul;  
  
Function Luas_Lingkaran(r:real):real;  
Begin  
    Luas_lingkaran:=3.14*r*r;  
End;  
Var luas: real;  
Begin  
    luas:=luas_lingkaran(10);  
    write(' Luas= ',luas:0:2);  
End.
```

Review: Identifier

■ Lingkup identifier meliputi:

- Lokal
- Global

```
Program Jangkauan;  
Var x,y: integer;  
    Function  
fungsi1(x:real):real;  
    Var y: real;  
    Begin  
        ...  
    End;  
    procedure pro;  
    var  
x,y,z:integer;  
    begin  
        ...  
    end;  
Var a,b:integer;  
Begin  
    ...  
End.
```


Review: Transfer Parameter

pass by reference



fillCup()

pass by value



fillCup()

www.mathwarehouse.com

- **By Value** : Yang dikirim ke modul lain atau ke program utama adalah **nilai datanya**
- **By Reference** : Yang ditransfer ke modul lain atau program utama adalah **alamat memorinya**

Review: Transfer Parameter

- By Value : Yang dikirim ke modul lain atau ke program utama adalah **nilai datanya**
- By Reference : Yang ditransfer ke modul lain atau program utama adalah **alamat memorinya**

```
1 function increaseAge(int age){  
2     return age + 1  
3 }  
4  
5  
6 int myAge = 14;  
7 myAge = increaseAge(myAge);
```

By Value

...	101	102	103	104
105	106	107	108	109
110	111	112	113	114
115	116	117	118	119
120	121	122	123	...

www.mathwarehouse.com

...	150	151	152	153
154	155	156	157	158
159	160	161	162	163
164	165	166	167	168

By Reference

...	101	102	103	104
105	106	107	108	109
110	111	112	113	114
115	116	117	118	119
120	121	122	123	...

Review: Transfer Parameter

By Value

```
Program Tranfer1;

    Procedure tukar(x,y:real);
        Var z: integer;
    Begin
        z:=x; x:=y; y:=z;
    End;

Var a,b:integer;
Begin
    a:=2; b:=9;
    tukar(a,b);
    writeln(' a= ',a,'      b:= ',b);
End.
```

By Reference

```
Program Tranfer1;

    Procedure tukar(var x,y:real);
        Var z: integer;
    Begin
        z:=x; x:=y; y:=z;
    End;

Var a,b:integer;
Begin
    a:=2; b:=9;
    tukar(a,b);
    writeln(' a= ',a,'      b:= ',b);
End.
```

Review: Rekursi

- Sub program dapat memanggil sub program lainnya
- Apabila sub program **memanggil dirinya sendiri**, inilah yang disebut dengan rekursif

Prosedur Rekursif

```
Procedure proc();  
Begin  
  proc();  
End;
```

Fungsi Rekursif

```
Function func();  
Begin  
  func();  
End;
```

Review: Rekursi

- Direct Call
- Indirect Call

```
Procedure proc;  
begin  
    proc ();  
end;
```

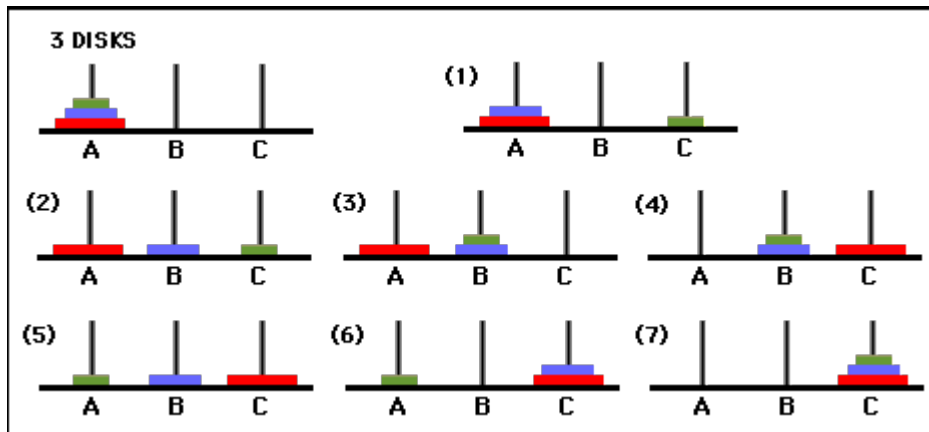
```
procedure proc1;  
begin  
    proc2;  
end;  
  
procedure proc2;  
begin  
    proc3;  
end;  
  
procedure proc3;  
begin  
    proc1;  
end;
```

Review: Rekursi

Faktorial

$$n! = n \times (n - 1) \times (n - 2) \times (n - 3) \times \cdots \times 3 \times 2 \times 1$$

Menara Hanoi



Fibonacci

$$F_0 = 0, \quad F_1 = 1, \quad F_n = F_{n-1} + F_{n-2}$$

Permutasi dan Kombinasi

Review: Iterasi Vs Rekursi

Iterasi

```
Function Factorial(n : Integer) :  
LongInt;  
Var  
    Result : LongInt;  
    i : Integer;  
Begin  
    Result := n;  
    If (n <= 1) then  
        Result := 1  
    Else  
        For i := n-1 DownTo 1 do  
            Result := Result * i;  
        Factorial := Result;  
    End;
```

Rekursi

```
Function Factorial(n : Integer) :  
Integer;  
Var  
    Result : Integer;  
Begin  
    If n = 1 then  
        Factorial := 1  
    Else  
        Factorial := n*Factorial(n-1);  
    End;
```

Review: Searching

- **Proses pencarian** adalah menemukan data tertentu di dalam sekumpulan data yang **bertipe sama**
- Fokus: **Array**

D: array [1..11] of integer

Kar: array [1..8] of character

Const N : integer = 5 {jumlah siswa}

Type Data = record <Nama: string, Usia: integer>

DataSiswa : array[1..N] of Data

21	36	8	7	10	36	68	32	12	10	36
1	2	3	4	5	6	7	8	9	10	11

k	m	t	a	f	m	*	#
1	2	3	4	5	6	7	8

1	Ali	18
2	Tono	24
3	Amir	30
4	Tuti	21
5	Yani	22

Review: Searching

- **Sequential Search** membandingkan setiap elemen array satu per satu secara beruntun, mulai dari elemen pertama, sampai elemen yang dicari ditemukan atau sampai seluruh elemen sudah diperiksa
 - Array tidak terurut
 - Array terurut
- **Binary Search** diterapkan pada sekumpulan data yang sudah terurut (terurut menaik atau menurun) dalam menemukan nilai tertentu dalam sebuah larik (array) linear dengan menghilangkan setengah data pada setiap langkah.

Review: Sorting

- **Pengurutan** adalah proses menyusun kembali data yang sebelumnya telah disusun menurut aturan tertentu (Ascending atau Descending)
 - Bubble Sort, pengurutan berdasar perbandingan
 - Selection Sort, pengurutan berdasar prioritas/ seleksi
 - Insertion Sort, pengurutan berdasarkan penyisipan dan penjagaan terurut

Notasi “O Besar” (Big-O notation)

- Setiap programmer yang baik akan menggunakan cara yang paling efektif dan efisien dalam menyelesaikan suatu permasalahan berarti harus bisa meminimalisir kompleksitas dari algoritma yang kita gunakan
- Kompleksitas suatu algoritma dibagi menjadi 2, yaitu **Time Complexity** dan **Space Complexity**
- **Time Complexity** adalah seberapa lama waktu yang diperlukan untuk menjalankan suatu algoritma. Sedangkan **Space Complexity** adalah seberapa besar memori yang kita gunakan untuk menjalankan suatu algoritma.
- Dalam **Time Complexity** □ **Big O Notation** digunakan untuk mengukur tingkat kompleksitas suatu algoritma (*berapa lama waktu yang dibutuhkan untuk menjalankan suatu algoritma dengan input tertentu (n)*)

Notasi “O Besar” (Big-O notation)

- $O(1)$ — Constant Time
- $O(\log n)$ — Logarithmic Time
- $O(n)$ — Linear Time
- $O(n^2)$ — Quadratic Time
- $O(2^n)$ — Exponential Time

Latihan Soal

- Kamu bertanggung jawab untuk membeli kue ulang tahun keponakanmu. Kamu memutuskan untuk membeli lilin sebanyak jumlah umurnya. Ketika dia meniup lilinnya, dia hanya bisa memadamkan lilin yang paling tinggi. Tugas kamu adalah menghitung berapa banyak jumlah lilin yang dapat dipadamkan!

- Hints:
 1. Input umur (contoh : 5)
 2. Input tinggi setiap lilin (contoh : 3, 4, 5, 5, 2)
 3. Output: jumlah lilin yang dapat dipadamkan (contoh : 2)

Solusi *Birthday Cake Candles*

```
program tiuplilin;
var
    umur : integer;
    tinggi : array of integer;
    i : integer;
    temp : integer;
    count : integer;
begin
    writeln('Umur = ');
    readln(umur);

    setLength(tinggi, umur);
    temp := 0;
    count := 0;
    for i := 0 to (umur-1) do
    begin
        writeln('Tinggi lilin ', i, ' = ');
        readln(tinggi[i]);
        if(tinggi[i] > temp) then
        begin
            temp := tinggi[i];
            count := 1;
        end
        else if(tinggi[i] = temp) then
            count := count + 1;
        end;
    end;

    writeln('Jumlah lilin yang dapat dipadamkan = ', count);
end.
```

Tipe Data, Objek Data dan Struktur Data

- **Tipe Data:** jenis data yang mampu ditangani oleh suatu bahasa pemrograman pada komputer
 - misal di Pascal: integer, real dll
- **Objek Data** adalah kumpulan elemen yang mungkin untuk suatu tipe data tertentu
 - misal integer di pascal adalah bilangan antara -32768 s.d. 32767
- **Struktur Data** adalah cara penyimpanan , pengorganisasian , dan pengaturan data di dalam media penyimpanan komputer sehingga data tersebut dapat digunakan secara efisien

Aktivitas Struktur Data

1. Mendeskripsikan kumpulan obyek data yang sah sesuai dengan tipe data yang ada
2. Menunjukkan mekanisme kerja operasi-operasinya

Contoh:

integer (-32768 s/d 32767) dan jenis operasi yang diperbolehkan adalah +, -, *, /, mod, <, >, dll

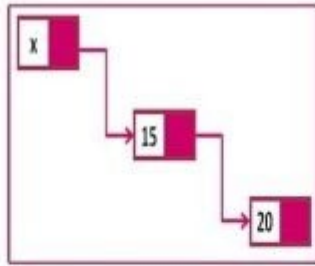
Struktur Data = obyek data + [operasi manipulasi data]

Algoritma + Struktur Data = Program

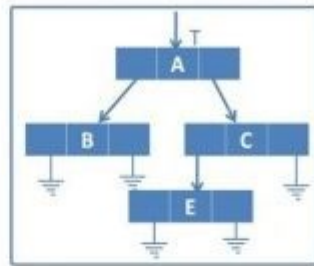
Contoh Struktur Data



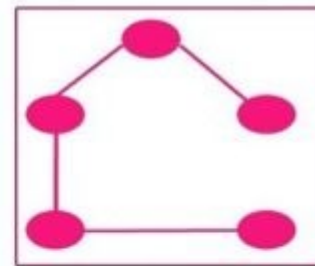
Sorting



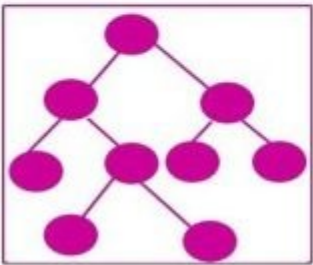
Link list



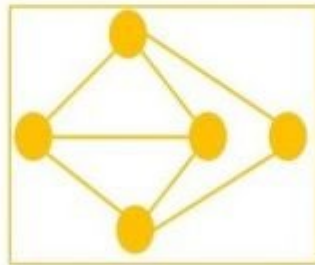
list



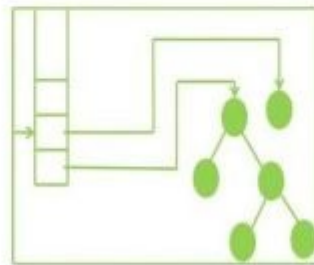
spanning tree



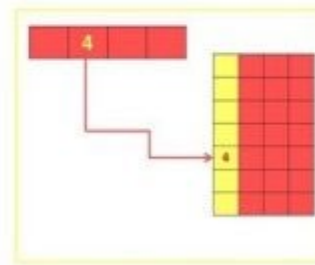
Tree



Graph



Stack



Hashing