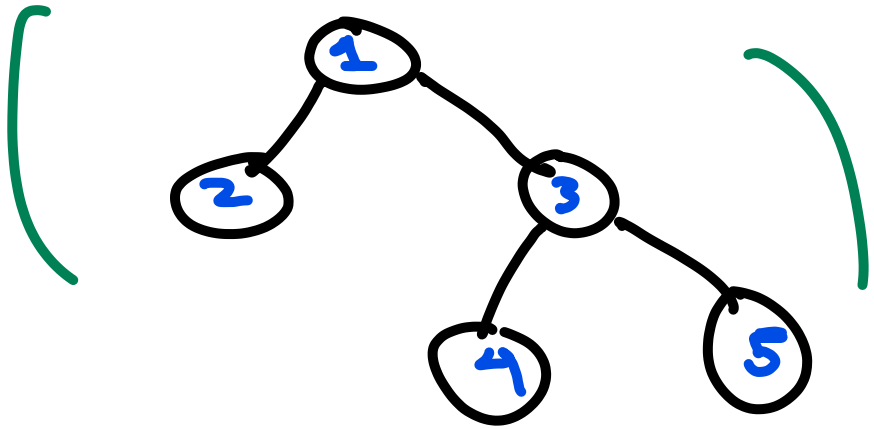
ADT

(Struct comprehension, Pointer)

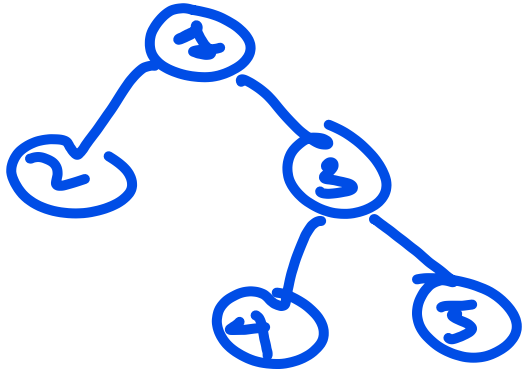
Without ADT



→ (1 : {2,3}, 3 : {4,5}) → [1 0 / 0 1 / ..]

Representasi

With ADT



→

interface

```
Tree {
    Left
    Right
}
```

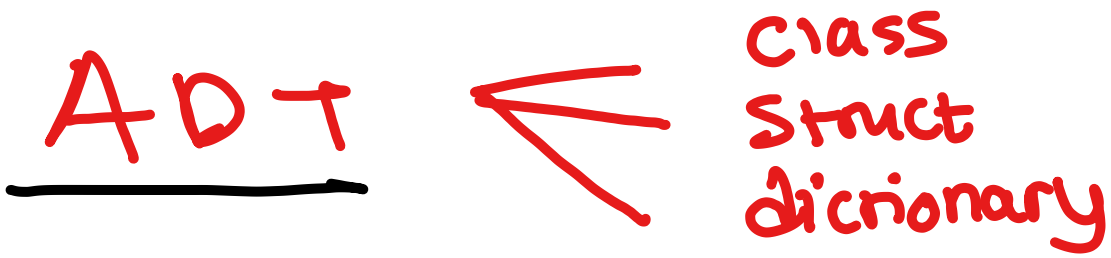
Node

1

Left R

2

Map ⇒



Extrak fitur penting

Abstraksi

```
Animal{
  string name;
  int age;
}
```

object

* atribut / Properti

Behaviour ⇒ Method / Func

move()

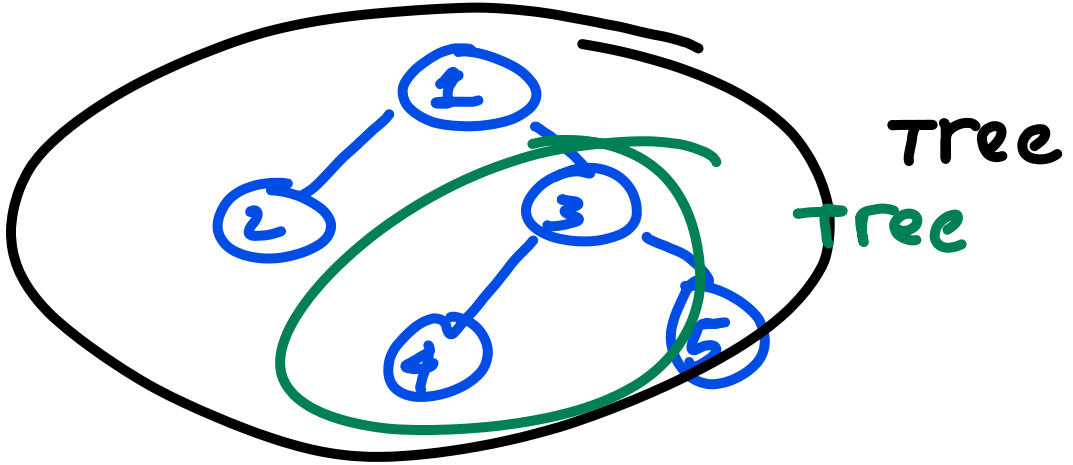
sound()

exist() → initiate() → constructor()

decomposition

inheritance

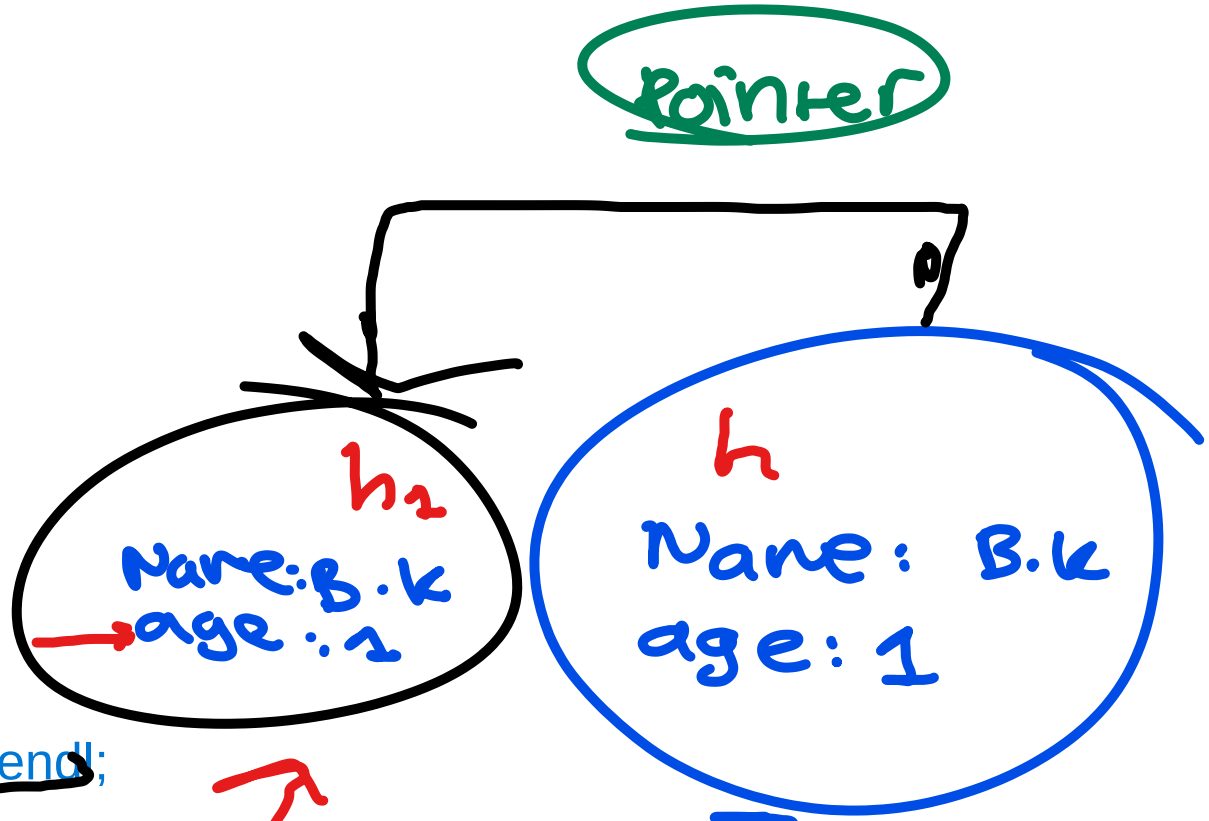
Polymorphism



The main tree has a sub tree decomposition

```
void ultah(Human h){
    h.age++;
}
```

```
Human bintang;
    bintang.born("Bintang Krisnanda");
    cout<<bintang.name<<endl;
    cout<<"Usia sebelum"<<bintang.age<<endl;
    ultah(bintang);
    cout<<"Usia sekarang"<<bintang.age<<endl;
```



ultah (bintang)

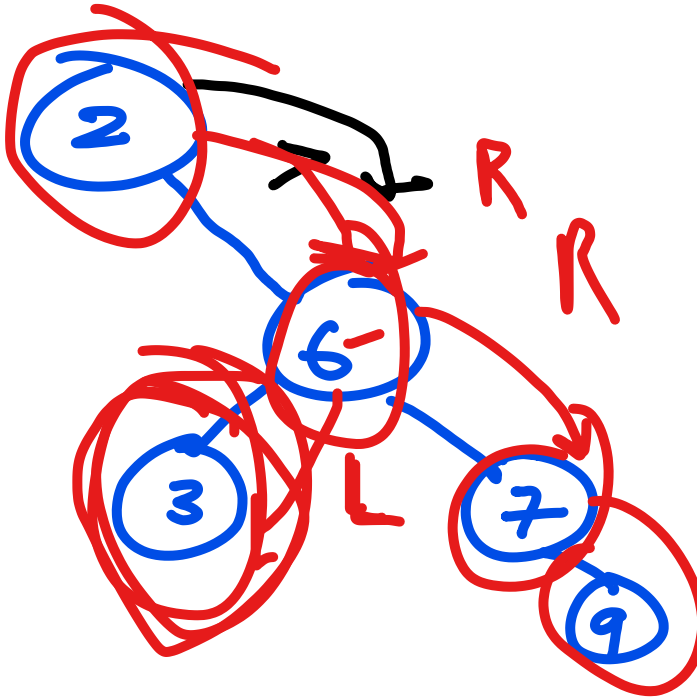
h = bintang

h.age++ → kanan

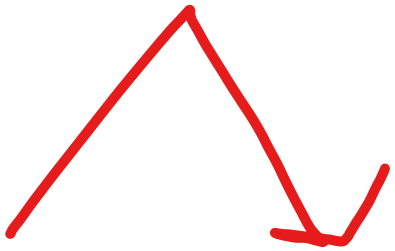
Binary Search Tree -> Left < Right

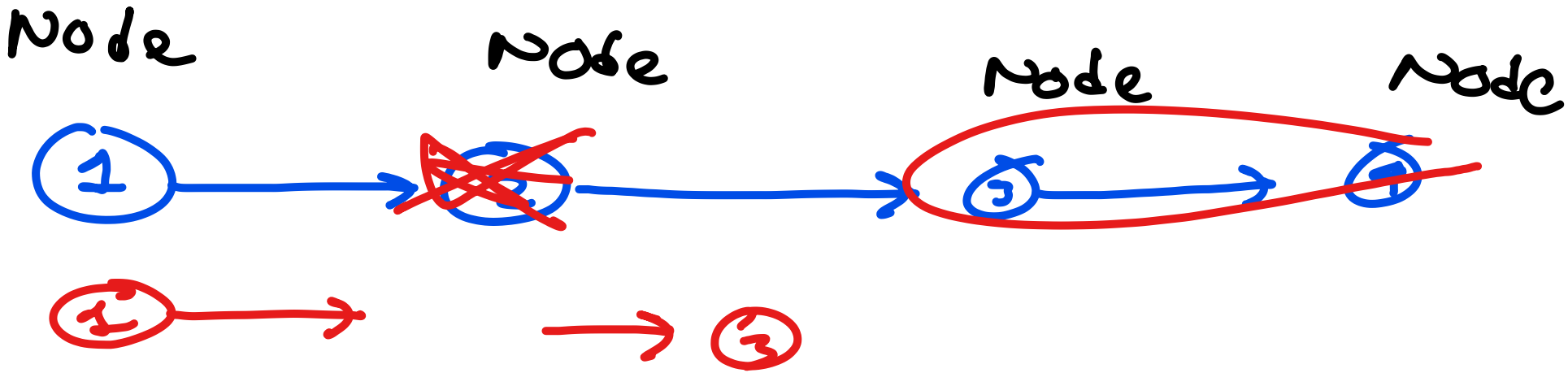
array = [2, 6, 7, 4, 3]

Root



Pre Order : 2 6 3 7 9
In Order :




Node
Node
Node
Node
1
2
3
4
1
3