

# **Algoritma dan Struktur Data**

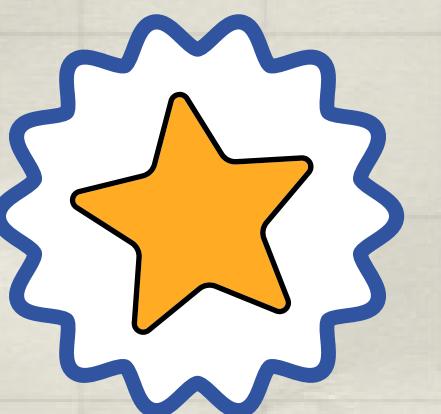
## **Kelompok 1**

### **Polymorphism**

**Nama Kelompok:**

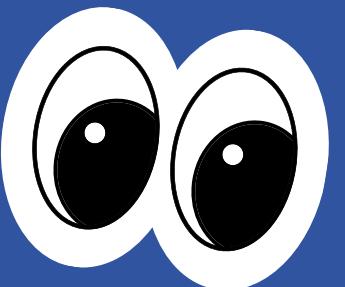
- Tarisma Dwi Putri (1206210001)**
- Rahmat Sigit Hidayat (1206210005)**
- Winalda Risky Febriyanti (1206210018)**
- Astikhatul Mufaidah (1206210020)**

*lele*



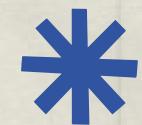


# APA ITU OOP?



OOP (Object Oriented Programming) adalah salah satu metodologi yang menjadi sangat populer selama beberapa tahun terakhir.

Object Oriented adalah artinya diarahkan pada objek. dengan kata lain, itu berarti secara fungsional diarahkan pada pemodelan objek. Ini adalah salah satu dari banyak teknik yang digunakan untuk memodelkan sistem yang kompleks dengan menggambarkan kumpulan objek yang berinteraksi melalui data dan perilakunya.



Object Oriented Programming (OOP) Python adalah cara pemrograman yang berfokus pada penggunaan object dan classes untuk merancang dan membangun aplikasi. Pilar utama Object Oriented Programming (OOP) adalah Inheritance, Polymorphism, Abstraction dan Encapsulation.

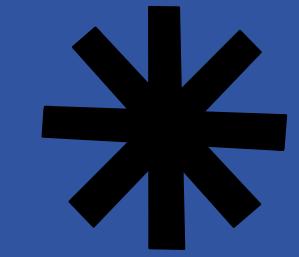


# \* APA ITU POLYMORPHISM ?

Polymorphism merupakan suatu object yang dapat memiliki berbagai bentuk object dari class sendiri atau object dari superclass.

- Overloading : penggunaan satu nama untuk beberapa method yang berbeda (beda parameter)
- Overriding : terjadi ketika deklarasi method subclass dengan nama dan parameter yang sama dengan method superclass.





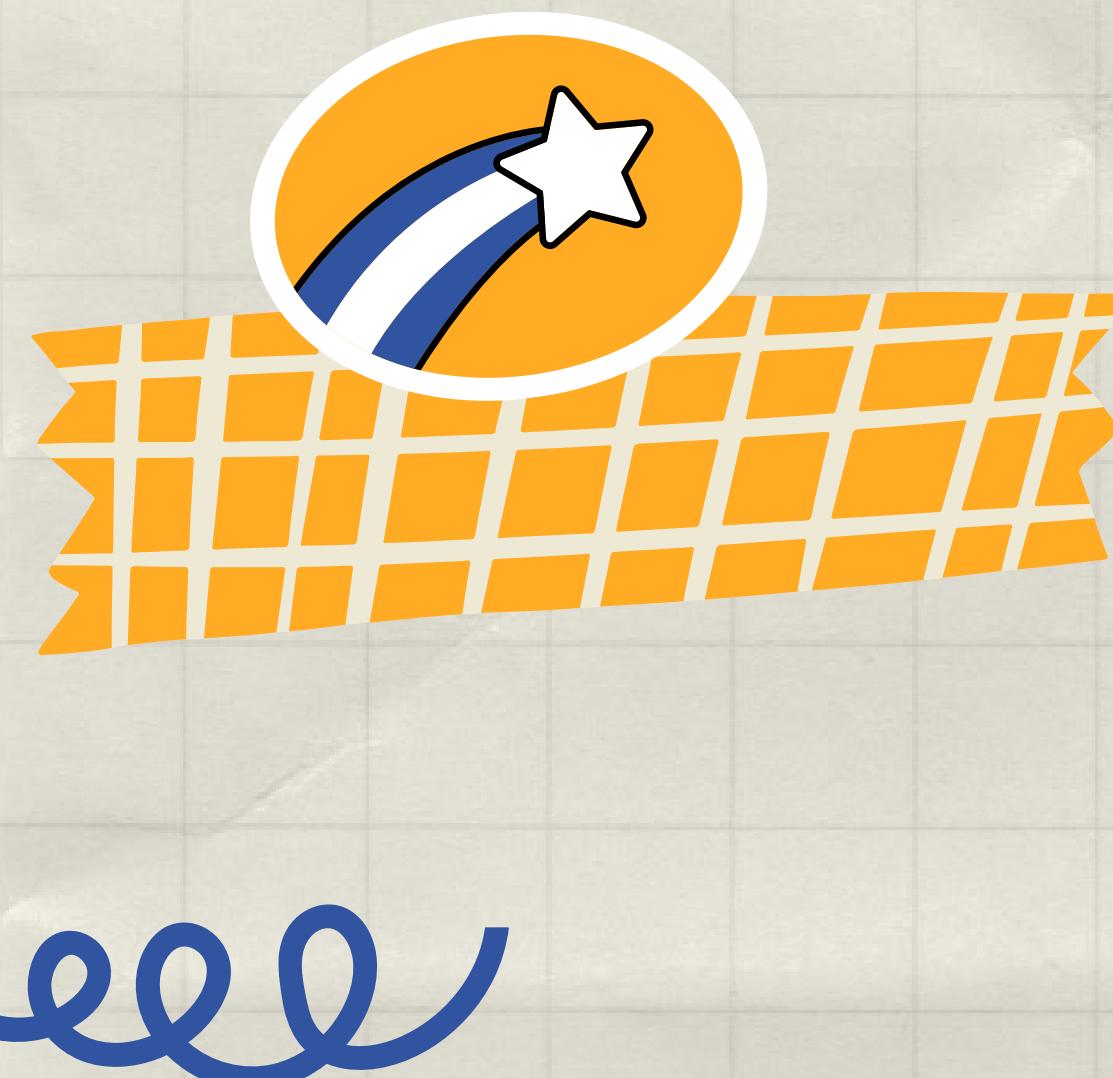
# KEUNTUNGAN PEMROGRAMAN DENGAN MENGGUNAKAN POLYMORPHISM YAITU:

- Dapat menggunakan beberapa class yang kita buat (sebagai superclass), kemudian membuat class yang baru berdasarkan superclass tersebut dengan karakteristik yang lebih khusus dari behavior umum yang dimiliki superclass.
- Dapat membuat superclass yang hanya mendefinisikan behavior namun tidak memberikan implementasi dari metode-metode yang ada.
- Menghindari duplikasi object yang dapat menciptakan class baru dari class yang sudah ada, sehingga tidak perlu menuliskan code dari nol ataupun mengulangnya.



## PADA DASARNYA ADA 2 TIPE POLYMORPHISM YAITU:

- A. Static atau trivial : function overloading  
(penggunaan kembali nama fungsi yang sama tetapi dengan argumen yang berbeda).
  
- B. Dynamic atau true : function overriding  
(sebuah fungsi dalam class turunan yang memiliki nama, return type argumen function yang sama dengan fungsi dalam class induk).  
Menggunakan virtual method.



**YUK KITA MASUK  
KE PEMBAHASAN  
CODINGAN**

# INPUT

## CASE 3 : POLYMORPHISM

### DEFINE FUNCTION

FUNGSI PENCARIAN LINEAR UNTUK KEBUTUHAN INSERT DATA OLEH USER

```
In [1]: 1 def linear(a, x):
2     for i in range(len(a)):
3         if a[i] == x:
4             return True
5     return False
```

yang pertama yaitu melakukan declare fungsi umum pencarian linear untuk insert data maupun pemanggilan value dalam database



## DEFINE CLASS OBJECT

CLASS HEWAN SEBAGAI INDUK DAN PENYIMPANAN DATA DARI INPUT USER

In [2]:

```
1 class hewan:
2     def __init__(self):
3         self.data = {}
4         self.check=[]
5
6     def insert(self,jenis,nama,spesies,id_hewan,ciri):
7         find = linear(self.check,jenis)
8         self.check.append(jenis)
9         if find==False:
10             self.data[jenis]={nama:{'spesies':{spesies:{'id':id_hewan,'ciri':ciri}}}}#jenis baru
11         elif find==True:
12             spec=[]
13             for i in self.data[jenis]:
14                 spec.append(i)
15             find2 = linear(spec,nama)
16             if find2== False:
17                 self.data[jenis][nama]={'spesies':{spesies:{'id':id_hewan,'ciri':ciri}}}}#nama baru
18             elif find2== True:
19                 self.data[jenis][nama]['spesies'][spesies]={'id':id_hewan,'ciri':ciri}#spesies baru
20
21     def id_char(self,jenis,nama,spesies):
22         print('\n■ ID',spesies,'\t:',self.data[jenis][nama]['spesies'][spesies]['id'],'\n■ Dengan ciri-ciri :\n')
23         for i in self.data[jenis][nama]['spesies'][spesies]['ciri']:
24             print('•',i)
```

selanjutnya adalah Define Class induknya, yaitu hewan. Dibuka dengan method construct berisi self.data, sebagai wadah database dalam bentuk dictionary bertingkat, serta self.check untuk pencarian value pada saat memasukkan data.



```
25
26     def showdatabase(self):
27         print('*'*20,'DATABASE KEBUN BINATANG','*'*20)
28         for a in self.data:
29             print('---',a.upper(),'---')
30             for b in self.data[a]:
31                 print(b)
32                 for c in self.data[a][b]:
33                     for d in self.data[a][b][c]:
34                         print('\t•',d)
35                         for e in self.data[a][b][c][d]:
36                             print('\t\t',e,'= ',self.data[a][b][c][d][e])
37             dic=input('Ingin melihat data dengan format dictionary? [y/n]\t')
38             if dic=='y':
39                 print(self.data)
40
41     def lookdata(self,jenis,nama,x):
42         target=self.data[jenis][nama]['spesies']
43         c=[]
44         for i in target:
45             c.append(i)
46         return linear(c,x)
47
```

Selanjutnya ada method showdatabase, dimana method tersebut akan memberikan output bercabang dari value di dalam database dengan menggunakan nested for loop. Selain print out, user juga bisa memilih untuk menampilkan dalam bentuk dictionary atau tidak. Lalu ada method lookdata yang berfungsi sebagai pencari value spesies di dalam database. Method tersebut akan berguna di dalam pencarian pada main program nantinya.



```
48     def sounds(self):
49         if self.data['mamalia']=='paus':
50             print('suara paus')
51         elif self.data['mamalia']=='singa':
52             print('suara singa')
53         elif self.data['aves']=='elang':
54             print('suara elang')
55         elif self.data['insektta']=='lebah':
56             print('suara lebah')
57
58     def mobility(self):
59         if self.data['mamalia']=='paus':
60             print('berenang')
61         elif self.data['mamalia']=='singa':
62             print('berlari')
63         elif self.data['aves']=='elang':
64             print('terbang')
65         elif self.data['insektta']=='lebah':
66             print('terbang')
```

Pada bagian ini, kami mendeklarasikan method sounds dan mobility yang telah dipetakan berdasarkan nama hewannya



## DEFINE SUB CLASS OBJECT

CLASS SINGA, PAUS, LEBAH, DAN ELANG SEBAGAI SUB-CLASS DAN PEMANGGILAN INFORMASI MENGENAI HEWAN-HEWAN TERTENTU

In [3]:

```
1 class singa(hewan):
2     def __init__(self):
3         super().__init__()
4     def sounds(self):
5         print('Singa berkomunikasi melalui vokalisasi seperti auman keras,',
6               ' dengan meninggalkan tanda aroma, dan melalui isyarat visual, seperti kegelapan surai.',
7               ' Singa Jantan mengaum untuk melindungi wilayah mereka dari pejantan lain, untuk menarik pasangan kawin,',
8               ' dan untuk mencegah pemangsa lainnya.\n')
9     def mobility(self):
10        print('Mobilitas Singa adalah dengan berlari/berjalan dengan 4 kakinya.',
11              ' Singa tidak dapat berlari untuk jarak jauh tetapi hanya dalam jarak pendek.',
12              ' Bagian bawah cakar singa memiliki bantalan lembut dengan cakar yang dapat ditarik di setiap jari kaki,',
13              ' yang memungkinkan mereka berlari untuk jarak pendek. Tumit mereka tidak menyentuh tanah ketika berjalan.',
14              ' Ini karena mereka memiliki jari kaki yang besar dan bantalan tersebut,',
15              ' yang memungkinkan mereka untuk bergerak dengan tenang.\n')
16     def breath(self):
17         print('Singa adalah mamalia dan sewajarnya bernafas dengan paru-paru\n\n')
```



```
19 class paus(hewan):
20     def __init__(self):
21         super().__init__()
22     def sounds(self):
23         print('Mamalia laut sangat bergantung pada indra pendengaran untuk berkomunikasi dan makan,',
24             ' karena samudra menghamburkan Cahaya yang mengakibatkan Penglihatan kurang efektif.',
25             ' Indra penciuman juga terbatas karena molekul berdifusi lebih lambat di air.',
26             ' Paus menggunakan Vokalisasi untuk berkomunikasi. Suara yang dihasilkan oleh paus bisa',
27             ' mencapai hingga 188 db, tujuannya adalah untuk menjaga jarak antar individu, mengenali ',
28             ' spesies, bahkan untuk mencari makanan.\n')
29     def mobility(self):
30         print('Karena hidup di lautan, mobilitas paus bergantung kepada sirip dan ekornya.',
31             ' Paus berenang dengan gerakan ekor vertikal', 'Menurut ahli dari University In Glassboro,',
32             ' Kenneth Lacovara, alasan paus menggerakkan ekornya ke atas dan ke bawah adalah karena paus',
33             ' berevolusi dari mamalia sekitar 50 juta tahun yang lalu. Mamalia empat kaki yang tinggal di',
34             ' darat menggerakkan tulang belakangnya ke atas dan ke bawah ketika berlari.\n')
35     def breath(self):
36         print('Sebagaimana Mamalia, meskipun hidup di lautan, Paus bernafas dengan paru-paru\n\n')
37
38 class elang(hewan):
39     def __init__(self):
40         super().__init__()
41     def sounds(self):
42         print('Suara dari Elang adalah kombinasi dari siulan dan "piping" bernada tinggi.',
43             ' Jeritan Elang atau suara memekiknya sering diberikan sebagai tanggapan terhadap manusia yang',
44             ' mendekati Elang dan merupakan tangisan bernada tinggi yang memiliki tiga hingga lima nada yang ',
45             ' terdengar seperti burung camar diikuti oleh enam atau tujuh nada cepat.\n')
46     def mobility(self):
47         print('Karena hidup di darat dan udara(dominan), cara mobilitas Elang adalah terbang.',
48             ' Elang terbang dengan cara mengepakan sayapnya. Saat sayap burung dikenakan, udara akan didorong kebawah.',
49             ' Dorongan kebawah akan menghasilkan gaya yang berlawanan sehingga mengangkat tubuh burung keatas.',
50             ' Untuk mengendalikan gerakan baik menuik, membelok, burung menggunakan ekornya.\n')
51     def breath(self):
52         print('Elang bernafas dengan paru-paru. Namun Ketika mengudara,',
53             ' Elang menggunakan udara kantong/pundi-pundi udara sebagai alat pernapasannya.',
54             ' ketika tidak terbang, maka Elang mengisi pundi-pundi udaranya.\n')
```



```
56 class lebah(hewan):
57     def __init__(self):
58         super().__init__()
59     def sounds(self):
60         print('Manusia mendengar lebah mendengung, namun sebenarnya suara tersebut dihasilkan oleh',
61               'kepakan sayap serangga yang cepat saat mereka terbang di udara dan menciptakan getaran',
62               'yang dideteksi telinga manusia sebagai dengung.\n\tBzzzzzz\n')
63     def mobility(self):
64         print('lebih kurang sama seperti Elang, mobilitas lebah dominan terbang. ',
65               'Sayap mereka tidak kaku, tetapi *twist-and-rotate* saat terbang.',
66               ' Sayap lebah membuat gerakan menyapu pendek dan cepat ke depan dan belakang, depan dan belakang.',
67               ' Gerakan ini menciptakan daya angkat yang cukup untuk memungkinkan lebah terbang.\n')
68     def breath(self):
69         print('Lebah juga memiliki sistem peredaran darah terbuka, yang berarti bahwa mereka tidak memiliki',
70               'pembuluh darah atau arteri, melainkan semua organ internal mereka bermandikan cairan yang disebut',
71               '**hemolymph** (campuran darah dan cairan limfatik). Lebah bernafas melalui struktur kompleks jaringan',
72               'trachea dan kantung udara.\n\n')
```

Pada bagian subclass, kami telah menyiapkan 4 buah sub class yang terhubung dengan class induknya dengan method yang memiliki nama serupa, kecuali breath. Subclass ini nantinya akan berguna dalam pemanggilan output di luar dari database.



## MAIN PROGRAM

KAMI MENGAPLIKASIKAN POLYMORPH DALAM OOP SEBAGAI PROGRAM YANG DIGUNAKAN UNTUK MENAMBAHKAN DATA MAUPUN MENCARI INFORMASI SEPUTAR HEWAN DI KEBUN BINATANG

```
[1]: hewan=hewan()
[2]: while True:
[3]:     print('\n','■'*50,'\\n    SELAMAT DATANG DI KEBUN BINATANG~\\n','■'*50)
[4]:     user=input('\\n1. Login Petugas\\n2. Pengunjung\\n3. Stop Program\\n\\t *[1/2/3]')
[5]:     if user=='1':
[6]:         loop=True
[7]:         while loop== True:
[8]:             uname=input('>\\tMasukkan Username\\t:')
[9]:             pword=input('>\\tMasukkan Password\\t:')
[10]:            if uname=='penjagamonyet' and pword=='pendek':
[11]:                print('■'*40,'>>>\\nHALO KAKAK PETUGAS!')
[12]:                while True:
[13]:                    update=input('\\n>\\t APA YANG INGIN KAKAK LAKUKAN ?\\n1.Update Database\\n2.Lihat Database\\n3.Tidak Ada\\n\\t')
[14]:                    if update=='1':
[15]:                        for i in range(int(input('Berapa jumlah data yang ingin dimasukkan? :\\n\\t'))):
[16]:                            print('='*50,'Data ke',i+1,'\\n')
[17]:                            jenis = input('●Masukkan nama jenis dari hewan [mamalia/aves/insekta] :\\n\\t')
[18]:                            id_hewan = int(input('●Masukkan id hewan [numerik] :\\n\\t'))
[19]:                            spesies = input('●Masukkan nama spesies dari hewan [ex : singa afrika; elang jawa] :\\n\\t')
[20]:                            nama = spesies[:5]
[21]:                            ciri=[]
[22]:                            for j in range(int(input('●Berapa ciri-ciri yang ingin dimasukkan? :\\t'))):
[23]:                                print(j+1,end='.')
[24]:                                ciri.append(input('\\t'))
[25]:                            hewan.insert(jenis,nama,spesies,id_hewan,ciri)
[26]:                            print('\\n\\t=====Data ke',i+1,'berhasil ditambahkan!=====\\n')
[27]:                    elif update=='2':
[28]:                        hewan.showdatabase()
[29]:                    else:
[30]:                        break
[31]:                loop=False
[32]:            elif uname!='penjagamonyet' and pword!='pendek':
[33]:                print('\\nxxx=====username atau password belum sesuai, silahkan coba lagi=====xxx\\n')
```



```
34 elif user=='2':
35     print('*'*40,'>>>\nHALO KAKAK PENGUNJUNG!\n')
36     search=input('>\t HEWAN APA YANG INGIN KAKAK CARI TAHU? [ex : singa afrika; elang jawa] :\n\t')
37     if search[:5]=='singa':
38         if hewan.lookdata('mamalia',search[:5],search)==True:
39             print('\n\t\t',search.upper())
40             singa=singa()
41             hewan.id_char('mamalia',search[:5],search)
42             print('\t\t DESKRIPSI\n')
43             singa.mobility()
44             singa.sounds()
45             singa.breath()
46     elif hewan.lookdata('mamalia',search[:5],search)==False:
47         print('Spesies tidak ditemukan')
48     elif search[:5]=='paus':
49         if hewan.lookdata('mamalia',search[:5],search)==True:
50             print('\n\t\t',search.upper())
51             paus=paus()
52             hewan.id_char('mamalia',search[:5],search)
53             print('\t\t DESKRIPSI\n')
54             paus.mobility()
55             paus.sounds()
56             paus.breath()
57     elif hewan.lookdata('mamalia',search[:5],search)==False:
58         print('Spesies tidak ditemukan')
59     elif search[:5]=='elang':
60         if hewan.lookdata('aves',search[:5],search)==True:
61             print('\n\t\t',search.upper())
62             elang=elang()
63             hewan.id_char('aves',search[:5],search)
64             print('\t\t DESKRIPSI\n')
65             elang.mobility()
66             elang.sounds()
67             elang.breath()
68     elif hewan.lookdata('aves',search[:5],search)==False:
69         print('Spesies tidak ditemukan')
70     elif search[:5]=='lebah':
71         if hewan.lookdata('insekta',search[:5],search)==True:
72             lebah=lebah()
73             print('\n\t\t',search.upper())
74             hewan.id_char('insekta',search[:5],search)
75             print('\t\t DESKRIPSI\n')
76             lebah.mobility()
77             lebah.sounds()
78             lebah.breath()
79     elif hewan.lookdata('insekta',search[:5],search)==False:
80         print('Spesies tidak ditemukan')
81     else:
82         print('maaf hewan tidak ditemukan')
```



```
84  
85 elif user=='3':  
86     break  
87 print('\n\n', '='*12, 'TERIMA KASIH TELAH MENGGUNAKAN PROGRAMINI, SAMPAI JUMPA!', '='*12, '\n')
```

Pada bagian main program, kami menggunakan nested while loop, yang pertama sebagai perulangan untuk pilihan antara petugas dan pengunjung. Kedua adalah perulangan untuk input username dan password. Program kami akan terus berputar sampai nanti user memutuskan untuk berhenti dari program dengan memecah kondisi perulangan pertama(untuk memecahnya akan kami jelaskan pada bagian output). Loop pertama membawa pada percabangan 3 kondisi; untuk petugas, pengunjung, dan pemberhentian program.

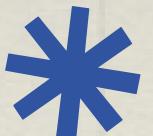
Jika sudah terjadi break loop kedua, maka user akan dikembalikan pada 3 percabangan awal. Selanjutnya adalah kondisi percabangan pengunjung, di sini user akan diminta inputan nama spesies yang akan dicari informasinya. Jika ada value inputan user di dalam database, maka akan keluar output detail mengenai spesies tersebut. Sedangkan jika salah, akan keluar output bahwa nama tidak ditemukan atau spesies tidak ditemukan.



# OUTPUT

```
SELAMAT DATANG DI KEBUN BINATANG~  
  
1. Login Petugas  
2. Pengunjung  
3. Stop Program  
    *[1/2/3]1  
>     Masukkan Username      :asda  
>     Masukkan Password       :afsaf  
  
xxx=====username atau password belum sesuai, silahkan coba lagi=====xxx  
>     Masukkan Username      :penjagamonyet  
>     Masukkan Password       :pendek  
----->>>  
HALO KAKAK PETUGAS!  
  
>     APA YANG INGIN KAKAK LAKUKAN ?  
1.Update Database  
2.Lihat Database  
3.Tidak Ada  
    1  
Berapa jumlah data yang ingin dimasukkan? :  
    2  
===== Data ke 1  
  
•Masukkan nama jenis dari hewan [mamalia/aves/insekta] :  
    mamalia  
•Masukkan id hewan [numerik] :  
    1234  
•Masukkan nama spesies dari hewan [ex : singa afrika; elang jawa] :  
    paus bungkuk  
•Berapa ciri-ciri yang ingin dimasukkan? :      3  
1.) habitat samudra pasifik  
2.) berat ribuan ton  
3.) hewan yang jarang ditemui  
  
=====Data ke 1 berhasil ditambahkan!=====  
  
===== Data ke 2  
  
•Masukkan nama jenis dari hewan [mamalia/aves/insekta] :  
    mamalia  
•Masukkan id hewan [numerik] :  
    1233  
•Masukkan nama spesies dari hewan [ex : singa afrika; elang jawa] :  
    singa macau  
•Berapa ciri-ciri yang ingin dimasukkan? :      2  
1.) surai pendek  
2.) badan ramping  
  
=====Data ke 2 berhasil ditambahkan!=====
```

Pada bagian output ini kami akan menjelaskan bagaimana luaran dari program, pembaca akan melihat bagaimana jika input awalnya adalah petugas dan user memasukkan uname atau password yang salah. Lalu pada percobaan kedua user berhasil masuk dan memilih opsi update database dengan memasukkan beberapa data sekaligus. Untuk detailnya akan kami jelaskan di bawahnya.



=====Data ke 2 berhasil ditambahkan!=====

> APA YANG INGIN KAKAK LAKUKAN ?

- 1.Update Database
- 2.Lihat Database
- 3.Tidak Ada

1

Berapa jumlah data yang ingin dimasukkan? :

2

===== Data ke 1

•Masukkan nama jenis dari hewan [mamalia/aves/insekta] :

insekta

•Masukkan id hewan [numerik] :

2222

•Masukkan nama spesies dari hewan [ex : singa afrika; elang jawa] :

lebah madu

•Berapa ciri-ciri yang ingin dimasukkan? : 3

- 1.) berkelompok
- 2.) jinak
- 3.) tidak beracun

=====Data ke 1 berhasil ditambahkan!=====

===== Data ke 2

•Masukkan nama jenis dari hewan [mamalia/aves/insekta] :

mamalia

•Masukkan id hewan [numerik] :

1223

•Masukkan nama spesies dari hewan [ex : singa afrika; elang jawa] :

singa jawa

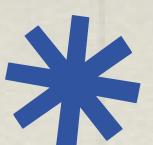
•Berapa ciri-ciri yang ingin dimasukkan? : 2

- 1.) badan tinggi
- 2.) ekor merah

=====Data ke 2 berhasil ditambahkan!=====



```
>      APA YANG INGIN KAKAK LAKUKAN ?
1.Update Database
2.Lihat Database
3.Tidak Ada
2
██████████ DATABASE KEBUN BINATANG ██████████
████ MAMALIA █████
paus
• paus bungkuk
  id    = 1234
  ciri  = ['habitat samudra pasifik', 'berat ribuan ton', 'hewan yang jarang ditemui']
singa
• singa macau
  id    = 1233
  ciri  = ['surai pendek', 'badan ramping']
• singa jawa
  id    = 1223
  ciri  = ['badan tinggi', 'ekor merah']
████ INSEKTA █████
lebah
• lebah madu
  id    = 2222
  ciri  = ['berkelompok', 'jinak', 'tidak beracun']
Ingin melihat data dengan format dictionary? [y/n] y
{'mamalia': {'paus': {'spesies': {'paus bungkuk': {'id': 1234, 'ciri': ['habitat samudra pasifik', 'berat ribuan ton', 'hewan yang jarang ditemui']}}, 'singa': {'spesies': {'singa macau': {'id': 1233, 'ciri': ['surai pendek', 'badan ramping']}, 'singa jawa': {'id': 1223, 'ciri': ['badan tinggi', 'ekor merah']}}}}, 'insekta': {'lebah': {'spesies': {'lebah madu': {'id': 2222, 'ciri': ['berkelompok', 'jinak', 'tidak beracun']}}}}}
```



> APA YANG INGIN KAKAK LAKUKAN ?

1.Update Database

2.Lihat Database

3.Tidak Ada

3

SELAMAT DATANG DI KEBUN BINATANG~

1. Login Petugas

2. Pengunjung

3. Stop Program

\*[1/2/3]2

HALO KAKAK PENGUNJUNG!

> HEWAN APA YANG INGIN KAKAK CARI TAHU? [ex : singa afrika; elang jawa] :

singa madiun

Spesies tidak ditemukan



SELAMAT DATANG DI KEBUN BINATANG~

1. Login Petugas
2. Pengunjung
3. Stop Program

\*[1/2/3]2

>>>

HALO KAKAK PENGUNJUNG!

> HEWAN APA YANG INGIN KAKAK CARI TAHU? [ex : singa afrika; elang jawa] :  
singa macau

■ SINGA MACAU

■ ID singa macau : 1233  
■ Dengan ciri-ciri :

- surai pendek
- badan ramping

DESKRIPSI

Mobilitas Singa adalah dengan berlari/berjalan dengan 4 kakinya. Singa tidak dapat berlari untuk jarak jauh tetapi hanya dalam jarak pendek. Bagian bawah cakar singa memiliki bantalan lembut dengan cakar yang dapat ditarik di setiap jari kaki, yang memungkinkan mereka berlari untuk jarak pendek. Tumit mereka tidak menyentuh tanah ketika berjalan. Ini karena mereka memiliki jari kaki yang besar dan bantalan tersebut, yang memungkinkan mereka untuk bergerak dengan tenang.

Singa berkomunikasi melalui vokalisasi seperti auman keras, dengan meninggalkan tanda aroma, dan melalui isyarat visual, seperti kegelapan surai. Singa Jantan mengaum untuk melindungi wilayah mereka dari pejantan lain, untuk menarik pasangan kawin, dan untuk mencegah pemangsa lainnya.

Singa adalah mamalia dan sejarnya bernafas dengan paru-paru





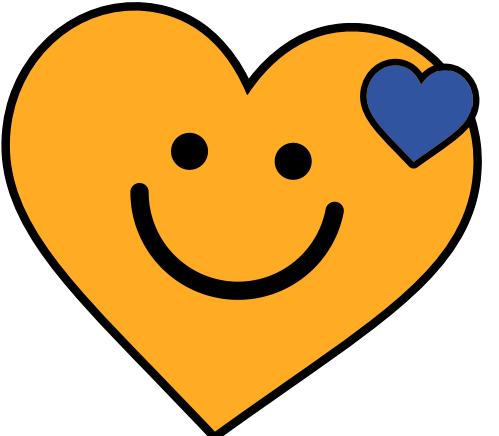
# KESIMPULAN DAN SARAN

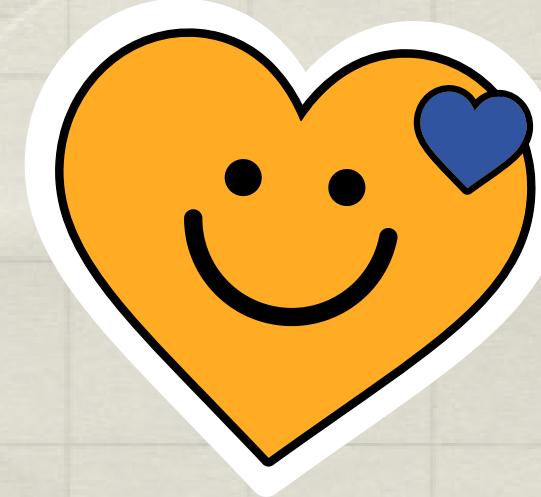
## Kesimpulan

Dari informasi yang telah diperoleh dapat disimpulkan bahwa untuk menuliskan polymorphism dalam pemrograman dapat dibuat dengan menggunakan OOP (Object Oriented Programming), karena pada dasarnya polymorphism merupakan bentuk OOP turunan.

## Saran

Untuk mendapatkan hasil yang lebih baik lagi. penulis memerlukan pengetahuan yang lebih dalam mengenai polymorphism.





eee



TERIMAKASIH! \*