

Pratikum Modul - 3



✓ Acc
1/2 '25

Disusun oleh :

Nama :Imam Ardi Perdana

Nim :24241020

Kelas :PTI A

Dosen Pembimbing :

Adam Bachtiar, S.Kom. ,M.MT.

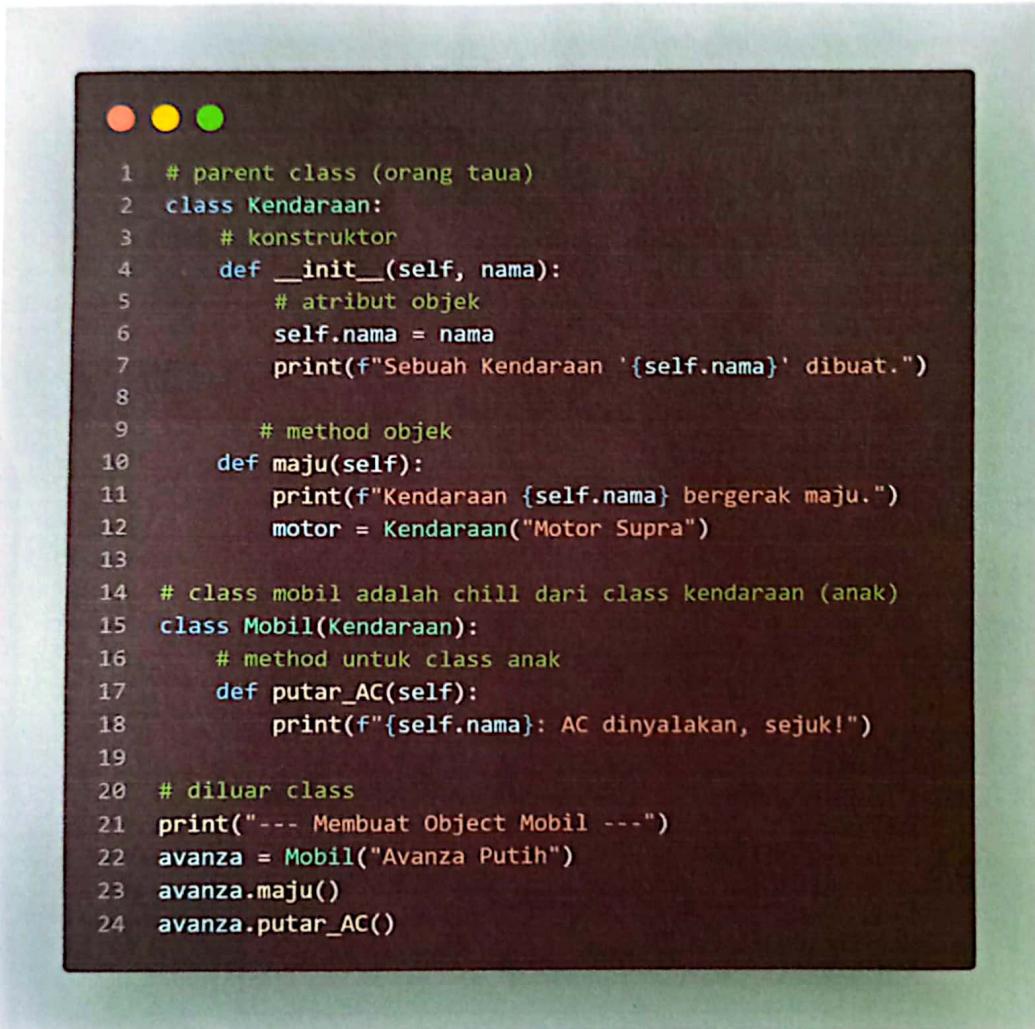
PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI (PTI)

FAKULTAS SAINS DAN ILMU TERAPAN (FSTT)

UNIVERSITAS MANDALIKA MATARAM

2020/2025

Praktikum Percobaan - 1



```
1 # parent class (orang tua)
2 class Kendaraan:
3     # konstruktor
4     def __init__(self, nama):
5         # atribut objek
6         self.nama = nama
7         print(f"Sebuah Kendaraan '{self.nama}' dibuat.")
8
9     # method objek
10    def maju(self):
11        print(f"Kendaraan {self.nama} bergerak maju.")
12        motor = Kendaraan("Motor Supra")
13
14 # class mobil adalah child dari class kendaraan (anak)
15 class Mobil(Kendaraan):
16     # method untuk class anak
17     def putar_AC(self):
18         print(f"{self.nama}: AC dinyalakan, sejuk!")
19
20 # diluar class
21 print("--- Membuat Object Mobil ---")
22 avanza = Mobil("Avanza Putih")
23 avanza.maju()
24 avanza.putar_AC()
```

Outpunya:

```
● PS E:\algoritma\Tugas-semester-3\modul 3> & C:/Users/USER/AppData/Local/Programs/Python/aktikum Percobaan - 1.py"
--- Membuat Object Mobil ---
Sebuah Kendaraan 'Avanza Putih' dibuat.
Kendaraan Avanza Putih bergerak maju.
Sebuah Kendaraan 'Motor Supra' dibuat.
Avanza Putih: AC dinyalakan, sejuk!
○ PS E:\algoritma\Tugas-semester-3\modul 3>
```

Penjelasan:

Baris 1: Parent class (kendaraan).

ini adalah komentar

Baris 2: class kendaraan

Baris ini mendefinisikan sebuah class bernama kendaraan

Baris 3: konstruktor

ini adalah komentar yang mendeklarasi bahwa blok kode selanjutnya adalah konstruktor dari class kendaraan

Baris 4: def __init__(self, nama)

Baris ini mendefinisikan konstruktor (method __init__) untuk kelas kendaraan

Baris 5: atribut objek

ini adalah komentar

Baris 6: self.nama = nama

Baris ini membuat atribut nama pada objek kendaraan

Baris 7: Print(f" sebuah kendaraan '{self.nama}' dibuat.")

ini mencetak pesan ke konsol yang memberitahu sebuah objek kendaraan

Baris 8: method objek

ini adalah komentar yang mendeklarasi selanjutnya adalah method objek

Baris 9: def maju(self)

Baris ini mendefinisikan sebuah method bernama maju di dalam kelas kendaraan

Baris 10: Print(f" kendaraan {self.nama} bergerak maju")

ini mencetak pesan ke konsol yang memberitahu bahwa kendaraan dengan nama tertentu bergerak maju

Baris 11: motor = kendaraan ("motor supra")

ini membuat objek baru dari kelas kendaraan dengan nama "motor supra"

Baris 13: class mobil(baru): class dari class kendaraan (lunatic)

ini adalah komentar

- Baris 14 : class mobil (Kendaraan)
ini mendefinisikan sebuah kelas bernama
mobil yang mewarisi dari kelas kendaraan
- Baris 15 : method untuk class anak
ini adalah komentar
- Baris 16 : def Putar_Ac (self)
ini mendefinisikan sebuah method bernama
Putar_Ac di dalam kelas mobil
- Baris 17 : Print (F" {self.nama} Ac dinyalakan, sejuk!")
ini mencetak Pesan ke konsol yang memberi-
tahu bahwa AC mobil dengan nama tertentu
dinyalakan
- Baris 18 : diwar class
ini adalah komentar
- Baris 19 : Print ("--- membuat objek mobil ---")
ini mencetak Pesan ke konsol bahwa
objek mobil akan dibuat
- Baris 20 : Avanza = mobil ("Avanza Putih")
membuat objek baru dari kelas mobil
bernama "Avanza Putih"
- Baris 21 : Avanza.maju ()
memanggil method maju Pada objek avanza.
- Baris 22 : Avanza.Putar_Ac
memanggil method Putar_Ac Pada objek
avanza. method ini khusus untuk kelas mobil.

Praktikum Percobaan – 2

```
1 class Kendaraan:
2     def __init__(self, nama):
3         print("-> (Parent __init__ dipanggil)")
4         self.nama = nama
5
6 class Mobil(Kendaraan):
7     def __init__(self, nama, jumlah_pintu):
8         print("-> (Child __init__ dipanggil)")
9
10    super().__init__(nama)
11
12    self.jumlah_pintu = jumlah_pintu
13    print(f"    -> Mobil {self.nama} dengan {self.jumlah_pintu} pintu dibuat.")
14
15 xenia = Mobil("Xenia Hitam", 4)
16 print(f"Nama object: {xenia.nama}")
```

Outpunya:

```
PS E:\algoritma\Tugas-semester-3\modul 3> & C:/Users/USER/AppData/Local/Programs/Python/Python313
  praktikum Percobaan - 2.py"
-> (Child __init__ dipanggil)
-> (Parent __init__ dipanggil)
-> Mobil Xenia Hitam dengan 4 pintu dibuat.
Nama object: Xenia Hitam
PS E:\algoritma\Tugas-semester-3\modul 3>
```

Penjelasan:

Baris 1 : class kendaraan

membuat sebuah "cetakan" (class) bernama kendaraan

Baris 2 : def __init__(self,nama)

"Pembuatan" (constructor) untuk class kendaraan

Baris 3 : Print ("-> (Parent __init__ dipanggil)")
mencetak pesan ketemu

Baris 4 : self.nama = nama

- menimpan informasi numu ke dalam objek kendaraan itu sendiri
- Baris 6 : class mobil (kendaraan)
membuat objekan (class) baru bernama mobil
- Baris 7 : def __init__(self, numu, Jumlah_Pintu)
"Pembuat" (constructor) untuk class mobil
- Baris 8 : Print ("→ (child_init_dipanggil)")
mencetak pesan ke layar.
- Baris 9 : super().__init__(numu)
menjalankan __init__ dari kendaraan dan memberikan informasi numu
- Baris 11 : Self.Jumlah_Pintu = Jumlah_Pintu
Setiap objek mobil akan punya jumlah pintu masing-masing
- Baris 12 : Print(f"→ mobil {self.nama} dengan {self.Jumlah_Pintu} pintu di buat").
sebuah objek mobil dengan nama dan jumlah pintu tertentu telah dibuat
- Baris 14 : Xenia=mobil ("Xenia hitam")
membuat objek mobil baru bernama xenia
- Baris 15 : Print(f"Nama Object: {Xenia.nama}")
mencetak pesan ke layar yang menampilkan numu dari objek Xenia

Praktikum Percobaan – 3

```
1  class Hero:
2      def __init__(self, name, health):
3          self.name = name
4          self.health = health
5
6      def showInfo(self):
7          print("showInfo di class Hero")
8          print("{} health: {}".format(
9              self.name,
10             self.health
11         )
12     )
13
14 class Hero_intelligent(Hero):
15     def __init__(self, name):
16         super().__init__(name, 100)
17
18     def showInfo(self):
19         print("showInfo di subclass Hero_intelligent")
20         print("{} \n\tTipe: intelligent, \n\thealth: {}".format(
21             self.name,
22             self.health
23         )
24     )
25
26 class Hero_strength(Hero):
27     def __init__(self, name):
28         super().__init__(name, 200)
29
30
31 lina = Hero_intelligent('lina')
32 axe = Hero_strength('axe')
33
34 lina.showInfo()
```

Outpunya:

```
• PS E:\algoritma\Tugas-semester-3\modul 3> & C:/Users/USER/AppData/Local/Programs/Python/aklikum Percobaan 3.py"
showInfo di subclass Hero_intelligent
lina
    Tipe: intelligent,
    health: 100
• PS E:\algoritma\Tugas-semester-3\modul 3>
```

Penjelasan:

Baris 1 : class Hero

Baris ini membuat "blueprint" atau cetakan bernama Hero. ini adalah dasar untuk membuat karakter Pahlawan dalam game.

Baris 2 : def __init__(self, name, health)

ini adalah "Pembangun" atau constructor untuk class Hero. Setiap kali kita membuat hero baru, kode di sini akan dipakai. self, adalah sebutan untuk hero itu sendiri, name adalah nama hero, dan health adalah jumlah kesehatan yang dia punya.

Baris 3 : self.name = name

Baris ini menyimpan nama yang kita berikan ke dalam objek hero itu sendiri.

Baris 4 : self.health = health

Baris ini menyimpan jumlah kesehatan yang kita berikan ke dalam objek itu sendiri.

Baris 6 : def showinfo(self)

Baris ini membuat fungsi bernama showinfo di dalam class Hero. Fungsi ini akan menampilkan informasi tentang hero

Baris 7 : Print("showinfo di class Hero")

Baris ini mencetak pesan yang memberitahu kita bahwa kita sedang melihat informasi dari class Hero (induk).

Baris 8 : Print("{} health: {} format({self.name, self.health}))

mencetak nama dan kesehatan hero.

Baris 14 : class Hero_intelligent(Hero)

membuat class baru bernama Hero_intelligent

Baris 15 : def __init__(self, name)

constructor untuk class Hero_intelligent

Baris 16 : super().__init__(name=100)

Baris ini penting! Karena Hero_intelligent mewarisi dari Hero, kita perlu jalankan constructor dari hero juga.

Baris 18 : def showinfo(self)

membuat fungsi showInfo di dalam class
Hero-intelligent

Baris 14: Print ("showInfo di subclass Hero-Intelligent")
mencetak pesan yang memberitahu kita bahwa
sedang melihat informasi dari class Hero-
Intelligent (anak)

Baris 20: Baris ini mencetak nama, type (intelligent)
dan keselatan Hero-Intelligent

Baris 26: class Hero-Strength (hero)
membuat class baru bernama Hero-strength

Baris 27: def __init__(self, name)
constructor untuk class Hero-strength

Baris 28: super().__init__(name, 200)
menjalankan constructor dari hero, memberikan
nama, dan memberikan nilai keselatan
default 200

Baris 31: linu = hero_intelligent ('linu')
membuat objek Hero-intelligent bernama linu
dengan nama 'linu'

Baris 32: axe = Hero-strength ('axe')
membuat objek Hero-strength bernama axe
dengan nama 'axe'

Baris 33: linu.showInfo()
memunggii fungsi showInfo pada objek linu.

Praktikum Percobaan – 4

```
1  class Ayah:
2      def __init__(self, nama_ayah):
3          self.nama_ayah = nama_ayah
4
5      def bekerja(self):
6          print(f"{self.nama_ayah} sedang bekerja.")
7
8  class Ibu:
9      def __init__(self, nama_ibu):
10         self.nama_ibu = nama_ibu
11
12     def memasak(self):
13         print(f"{self.nama_ibu} sedang memasak.")
14
15 class Anak(Ayah, Ibu):
16     def __init__(self, nama_anak, nama_ayah, nama_ibu):
17         Ayah.__init__(self, nama_ayah)
18         Ibu.__init__(self, nama_ibu)
19         self.nama_anak = nama_anak
20
21     def bermain(self):
22         print(f"{self.nama_anak} sedang bermain.")
23
24 budi = Anak("Budi", "Hendra (Ayah)", "Wati (Ibu)")
25
26 budi.bekerja()
27 budi.memasak()
28 budi.bermain()
```

Outpunya:

```
PS E:\algoritma\Tugas-semester-3\modul 3> & C:/Users/USER/AppData/Local/Programs/Python, aktikum Percobaan 4.py"
Hendra (Ayah) sedang bekerja.
Wati (Ibu) sedang memasak.
Budi sedang bermain.
PS E:\algoritma\Tugas-semester-3\modul 3>
```

Penjelasan:

Baris 1 : class ayah

menciptakan sebuah blueprint atau cetakan bernama Ayah.

Baris 2 : def __init__(self, nama_ayah)

"Pembangun" atau constructor untuk class ayah

Baris 3 : self.nama_ayah = nama_ayah

menyimpan nama ayah yang kita berikan ke dalam objek ayah itu sendiri

Baris 5 : def bekerja(self)

membuat fungsi bernama (bekerja) di dalam class ayah

Baris 6 : print(f"\{self.nama_ayah\} sedang bekerja")

mencetak pesan yang memberitahu kita bahwa ayah (dengan nama yang sesuai) sedang bekerja

Baris 8 : class ibu

membuat sebuah blueprint atau cetakan bernama ibu

Baris 9 : def __init__(self, nama_ibu)

"Pembangun" atau constructor untuk class ibu

Baris 10 : self.nama_ibu = nama_ibu

menyimpan nama ibu yang kita berikan ke dalam objek ibu itu sendiri

Baris 12 : def memasuk(self)

membuat fungsi bernama memasuk di dalam class ibu

Baris 13 : mencetak pesan yang memberitahu kita bahwa ibu (dengan nama yang sesuai)

sedang memasuk

Baris 15 : class anak (Ayah, ibu)

menciptakan sebuah "blueprint" atau cetakan bernama (Anak)

Baris 16 : constructor untuk class anak

Baris 17 : Ayah.__init__(self, nama_ayah)

menjalankan constructor dari class ayah dan memberikan nama ayah.

Baris 18 : ibu.__init__(self, nama_ibu)

menjalankan constructor dari class ibu

dan memberikan nama ibu.

Buris 19 : menyimpan nama anak yang kita berikan ke dalam objek anak itu sendiri.

Buris 21 : membuat fungsi bernama bermuin (bermain) di dalam class Anak.

Buris 22 : mencetak pesan yang memberitahu kita bahwa anak (dengan nama yang sesuai) sedang bermain.

Buris 24 : membuat objek anak bernama bernama "budi", nama ayah, "Hendra" dan nama ibu "wati" (Ibu).

Buris 25 : memanggil fungsi belcerja pada objek budi karena budi adalah anak yang mewarisi dari Ayah, dia memiliki fungsi belcerja

Buris 26 : memanggil fungsi memasak pada objek budi. Karena budi adalah anak yang mewarisi ~~dari~~ ibu dia memiliki fungsi memasak

Buris 27 : Buris ini memanggil Fungsi bermain pada objek budi.

Tugas Praktikum Modul 3

```
1 class Pegawai:
2     def __init__(self, nama, nip, gaji_pokok):
3         self.nama = nama
4         self.nip = nip
5         self.__gaji_pokok = gaji_pokok # Atribut private
6
7     def get_gaji_pokok(self):
8         return self.__gaji_pokok
9
10    def hitung_bonus(self):
11        # Metode ini akan di-override oleh subclass
12        return 0
13
14    def get_gaji_total(self):
15        return self.__gaji_pokok + self.hitung_bonus()
16
17    def tampilan_info(self):
18        print(f"Nama: {self.nama}, NIP: {self.nip}")
19        print(f"Gaji Pokok: Rp {self.get_gaji_pokok():,.0f}")
20
21 class Manager(Pegawai):
22     def __init__(self, nama, nip, gaji_pokok, tunjangan_jabatan):
23         super().__init__(nama, nip, gaji_pokok)
24         self.tunjangan_jabatan = tunjangan_jabatan
25
26     def hitung_bonus(self):
27         return 0.15 * self.get_gaji_pokok()
28
29     def get_gaji_total(self):
30         return super().get_gaji_total() + self.tunjangan_jabatan
31
32     def tampilan_info(self):
33         print("--- Info Manager ---")
34         super().tampilan_info()
35         print(f"Tunjangan: Rp {self.tunjangan_jabatan:,.0f}")
36         print(f"Gaji Total Manager: Rp {self.get_gaji_total():,.0f}")
37
38
39 class StaffTeknis(Pegawai):
40     def __init__(self, nama, nip, gaji_pokok, jumlah_proyek):
41         super().__init__(nama, nip, gaji_pokok)
42         self.jumlah_proyek = jumlah_proyek
43
44     def hitung_bonus(self):
45         return 500000 * self.jumlah_proyek
46
47     def tampilan_info(self):
48         print("--- Info Staff Teknis ---")
49         super().tampilan_info()
50         print(f"Jumlah Proyek: {self.jumlah_proyek}")
51         print(f"Gaji Total Staff: Rp {self.get_gaji_total():,.0f}")
52
53
54 # Membuat instance dari kelas Manager dan StaffTeknis
55 manager = Manager("Budi Hartomo", "M-001", 10000000, 5000000)
56 staff = StaffTeknis("Susi Susanti", "S-001", 6000000, 3)
57
58 # Menampilkan informasi pegawai
59 manager.tampilan_info()
60 print("\n-----\n")
61 staff.tampilan_info()
62 print("\n-----\n")
63
64 # Tes Enkapsulasi
65 print("... Tes Keamanan (Encapsulasi) ...")
66 try:
67     print(staff.__gaji_pokok) # Mencoba mengakses atribut private langsung dari luar kelas
68 except AttributeError as e:
69     print(f"ERROR: {e}")
70     print("--> TIDAK BISA diakses langsung dari luar!")
71
72 print(f"Gaji Total Susi (tetap): Rp {staff.get_gaji_total():,.0f}") # Memastikan gaji total tetap dapat diakses melalui getter
```

Outpunya:

```
PS E:\algoritma\Tugas-semester-3\modul 3> & C:/Users/USR/AppData/Local/Programs/Python/Python33/python.exe  
gas_praktikum_3.py  
... Info Manager ...  
Nama: Budi Hartono, NIP: M-001  
Gaji Pokok: Rp 10,000,000  
Tunjangan: Rp 5,000,000  
Gaji Total Manager: Rp 15,000,000  
-----  
... Info Staff Teknis ...  
Nama: Susi Susanti, NIP: S-001  
Gaji Pokok: Rp 6,000,000  
Jumlah Proyek: 3  
Gaji Total Staff: Rp 7,500,000  
-----  
--- Tes Keamanan (Encapsulasi) ---  
ERROR: 'StaffTeknis' object has no attribute '_gaji_pokok'  
-> TIDAK BISA diakses langsung dari luar!  
Gaji Total Susi (tetap): Rp 7,500,000  
PS E:\algoritma\Tugas-semester-3\modul 3>
```

Penjelasan:

- Baris 1 : membuat "blueprint" atau cetakan bernama Pegawai (Pegawai / karyawan).
- Baris 2 : "Pembangun" atau constructor untuk class Pegawai
- Baris 3 : menyimpan nama Pegawai yang kita berikan ke dalam objek Pegawai itu sendiri.
- Baris 4 : menyimpan NIP Pegawai yang kita berikan ke dalam obsek Pegawai itu sendiri
- Baris 5 : menyimpan Gaji Pokok Pegawai yang kita berikan ke dalam obsek Pegawai itu sendiri
- Baris 6 : membuat fungsi bernama get_gaji_Pokok di dalam class Pegawai
- Baris 7 : menyembunyikan nilai gaji pokok.
- Baris 8 : membuat fungsi bernama hitung_bonus() di dalam class Pegawai

- Baris 11 : komentar yang memberitahukan kita bahwa fungsi hitung - bonus ini akan di ganti (override) oleh class anak (subclass)
- Baris 12 : mengembalikan nilai 0 sebagai bonus default .
- Baris 13 : membuat fungsi bernama get - gaji - total (dapatkan gaji total) di dalam class Pegawai .
- Baris 14 : menghitung gaji total dengan mensumblak - kan gaji pokok dan bonus .
- Baris 15 : membuat fungsi bernama tampilkan - info di dalam class Pegawai .
- Baris 16 : mencetak nama dan NIP Pegawai .
- Baris 17 : mencetak gaji pokok Pegawai
- Baris 18 : membuat class baru bernama manager .
- Baris 19 : constructor untuk class manager
- Baris 20 : menjalankan constructor dari class Pegawai (induk) dan memberikan nama , NIP , dan gaji pokok .
- Baris 21 : menyimpan fungsi jabatan yang kita berikan ke dalam objek manager .
- Baris 22 : mengganti (override) fungsi hitung - bonus yang ada di class Pegawai .
- Baris 23 : menghitung bonus manager sebesar 15 % dari gaji pokok
- Baris 24 : mengganti (override) fungsi get - gaji - total yang ada di class Pegawai .
- Baris 25 : menghitung gaji total manager dengan mensumblakan gaji total dari class Pegawai (gaji pokok + bonus) dan fungsi jabatan .
- Baris 26 : menghitung gaji total manager mengganti (override) fungsi tampilkan - info yang ada di class Pegawai
- Baris 27 : mencetak pesan yang memberitahu kita bahwa kita sedang melihat informasi manager .

Baris 34 : menyalankan fungsi tampilan - info dari class Pegawai (induk)

Baris 35 : mencetak fungsi gaji total manager

Baris 36 : mencetak gaji total manager

Baris 38 : membuat class baru bernama StaffTeknis
dari class Pegawai.

Baris 39 : constructor untuk class StaffTeknis

Baris 40 : menyalankan constructor dari class Pegawai
(induk) dan memberikan nama, nip, dan gaji
Pokok

Baris 41 : menyimpan jumlah Projek yang kita berikan
ke dalam objek staffteknik itu sendiri

Baris 43 : mengganti (override) fungsi hitung - bonus
yang ada di class Pegawai.

Baris 44 : menghitung bonus staff teknis sebesar 500000
dikalii jumlah Projek

Baris 46 : mengganti (override) fungsi tampilan - info
yang ada di class Pegawai

Baris 47 : mencetak pesan yang memberitahu kita bahwa
kita sedang merihal informasi staff teknis.

Baris 48 : menyalankan fungsi tampilan - info dari
class Pegawai (induk)

Baris 49 : mencetak jumlah Projek staff teknis

Baris 50 : mencetak gaji total staff teknis

Baris 51 : komentar yang memberitahukan kita bahwa
kita akan membuat objek dari class
manager dan staff teknis

Baris 55 : manager = manager ("Budi Hart")