**Tugas 06**

**Petunjuk pengerjaan:**

1. Jawablah setiap soal-soal di bawah ini dengan menggunakan editor python yang anda miliki.
2. Screenshot hasil pengerjaan **berupa kode dan running program** dan tempelkan pada jawaban.
3. Diperkenankan untuk berdiskusi dengan peserta lain untuk soal yang dianggap susah.

**Soal 1 Fungsi Genap**

Buatlah fungsi bernama genap untuk memeriksa apakah bilangan yang diinputkan sebagai argumen adalah bilangan genap atau bukan. Fungsi memiliki 1 parameter dengan tipe data bilangan bulat. Fungsi akan mengembalikan nilai berupa boolean **True** jika bilangan genap dan **False** jika bilangan tidak genap.

Selanjutnya fungsi akan dipanggil dalam program utama dengan masukan dan keluaran sebagai berikut:

**Masukan** berupa satu bilangan bulat.

**Keluaran berupa string "bilangan genap" atau "bukan bilangan genap"** bergantung pada kondisinya.

Petunjuk: Gunakan pengkondisian pada program utama.

Contoh masukan dan keluaran:

| **Masukan** | **Keluaran** |
| --- | --- |
| 2 | bilangan genap |
| 1 | bukan bilangan genap |

Jawaban:

# pendefinisian fungsi genap

def genap(...):

return ...

# pendefinisian program utama

def main():

# input dan casting

bil = ...

# pemanggilan fungsi dan pencetakan hasil

if ...:

print("bilangan genap")

else:

print("bukan bilangan genap")

# pemanggilan program utama

main()

|  |  |
| --- | --- |
|  |  |

**Soal 2 Fungsi Penjumlahan**

Buatlah fungsi bernama penjumlahan untuk menjumlahkan dua bilangan bulat. Fungsi memiliki 2 parameter dengan tipe data keduanya berupa bilangan bulat. Fungsi akan mengembalikan nilai berupa bilangan bulat.

Selanjutnya fungsi akan dipanggil dalam program utama dengan masukan dan keluaran sebagai berikut:

**Masukan** berupa dua bilangan bulat.

**Keluaran** berupa bilangan bulat,

Contoh masukan dan keluaran:

| **Masukan** | **Keluaran** |
| --- | --- |
| 1  2 | 3 |
| 45  67 | 112 |

Jawaban:

# pendefinisian fungsi penjumlahan

def penjumlahan(..., ...):

return ...

# pendefinisian program utama

def main():

# input dan casting

bil1 = ...

bil2 = ...

# pemanggilan fungsi

hasil\_penjumlahan = penjumlahan(..., ...)

# pencetakan hasil penjumlahan

print(...)

# pemanggilan program utama

if \_\_name\_\_ == '\_\_main\_\_':

main()

|  |  |
| --- | --- |
|  |  |

**Soal 3 Fungsi Perkalian**

Buatlah fungsi bernama perkalian untuk mengalikan dua bilangan. Parameter fungsi ada 2 dengan jenis tipe data keduanya berupa bilangan desimal. Fungsi akan mengembalikan nilai berupa bilangan desimal yang dibulatkan 2 digit di belakang koma.

Selanjutnya fungsi akan dipanggil dalam program utama dengan masukan dan keluaran sebagai berikut:

**Masukan** berupa dua bilangan desimal.

**Keluaran** berupa bilangan desimal.

Petunjuk: Gunakan fungsi **round()** untuk membulatkan 2 digit.

Contoh masukan dan keluaran:

| **Masukan** | **Keluaran** |
| --- | --- |
| 1.5  2.5 | 3.75 |
| 45.7  67.6 | 3089.32 |

Jawaban:

# pendefinisian fungsi perkalian

def perkalian(bilangan\_1, bilangan\_2):

return ...

# pendefinisian program utama

def main():

# input dan casting

bil1 = ...

bil2 = ...

# pemanggilan fungsi

hasil\_perkalian = perkalian(..., ....)

# pencetakan hasil perkalian

print(...)

# pemanggilan program utama

if \_\_name\_\_ == '\_\_main\_\_':

main()

|  |  |
| --- | --- |
|  |  |

**Soal 4 Fungsi luas lingkaran**

Buatlah fungsi bernama luas\_lingkaran untuk menghitung luas lingkaran. Fungsi memiliki 1 parameter dengan tipe data bilangan desimal. Fungsi akan mengembalikan nilai berupa bilangan desimal yang dibulatkan 2 digit di belakang koma.

Selanjutnya fungsi akan dipanggil oleh program utama dengan masukan dan keluaran sebagai berikut:

**Masukan** berupa satu bilangan desimal.

**Keluaran** berupa bilangan desimal.

Petunjuk: Gunakan fungsi **round()** untuk membulatkan 2 digit. Gunakan nilai pi = 3.14.

Contoh masukan dan keluaran:

| **Masukan** | **Keluaran** |
| --- | --- |
| 10.5 | 346.19 |
| 23.4 | 1719.34 |

Jawaban:

# pendefinisian fungsi perkalian

def perkalian(bilangan\_1, bilangan\_2):

return ...

# pendefinisian program utama

def main():

# input dan casting

bil1 = ...

bil2 = ...

# pemanggilan fungsi

hasil\_perkalian = perkalian(..., ....)

# pencetakan hasil perkalian

print(...)

# pemanggilan program utama

if \_\_name\_\_ == '\_\_main\_\_':

main()

|  |  |
| --- | --- |
|  |  |

**Soal 5 Fungsi keliling lingkaran**

Buatlah fungsi untuk bernama keliling\_lingkaran untuk menghitung keliling lingkaran. Parameter fungsi ada 1 dengan jenis tipe data bilangan desimal. Fungsi akan mengembalikan nilai berupa bilangan desimal yang dibulatkan 2 digit di belakang koma.

Selanjutnya, fungsi akan dipanggil pada program utama dengan masukan dan keluaran sebagai berikut:

**Masukan** berupa satu bilangan desimal yang menyatakan jari-jari lingkaran..

**Keluaran** berupa bilangan desimal.

Petunjuk: Gunakan fungsi **round()** untuk membulatkan 2 digit. Gunakan nilai pi = 3.14.

Contoh masukan dan keluaran:

| **Masukan** | **Keluaran** |
| --- | --- |
| 10.5 | 65.94 |
| 23.4 | 146.95 |

Jawaban:

# pendefinisian fungsi keliling lingkaran

def keliling\_lingkaran(radius):

pi = ...

return ...

# pendefinisian program utama

def main():

# input dan casting

jari\_jari = ...

# pemanggilan fungsi

keliling = ...

# pencetakan keliling

print(...)

# pemanggilan program utama

if \_\_name\_\_ == '\_\_main\_\_':

main()

|  |  |
| --- | --- |
|  |  |

**Soal 6 Fungsi sapa nama**

Buatlah fungsi bernama sapa\_nama untuk memberi sapaan. Parameter fungsi ada 1 dengan jenis tipe data string. Fungsi akan mengembalikan nilai berupa  string dengan format "Halo, <nama>".

Selanjutnya, fungsi akan dipanggil oleh program utama dengan masukan keluaran sebagai berikut.

**Masukan** berupa string yang menyatakan nama.

**Keluaran** berupa string dengan format "Halo, <nama>".

Contoh masukan dan keluaran:

| **Masukan** | **Keluaran** |
| --- | --- |
| tel-u | Halo, tel-u |
| sayang | Halo, sayang |

Jawaban:

# pendefinisian fungsi sapa nama

def sapa\_nama(...):

return "Halo, " + ...

# pendefinisian program utama

def main():

# input

nama = ...

# pemanggilan fungsi

hasil = sapaan\_nama(...)

# output

print(...)

# pemanggilan program utama

if \_\_name\_\_ == '\_\_main\_\_':

main()

|  |  |
| --- | --- |
|  |  |

**Soal 7 Validasi**

Buatlah fungsi bernama validasi untuk memvalidasi nilai. Terdapat 1 parameter dengan jenis tipe data bilangan desimal. Fungsi akan mengembalikan nilai berupa boolean True jika nilai antara 0 hingga 100 atau False jika nilai di luar 0 hingga 100.

Selanjutnya fungsi akan dipanggil dalam program utama dengan masukan dan keluaran sebagai berikut:

**Masukan** berupa bilangan desimal yang menyatakan nilai.

**Keluaran** berupa string dengan format "valid" atau "tidak valid".

Contoh masukan dan keluaran:

| **Masukan** | **Keluaran** |
| --- | --- |
| 100 | valid |
| 101 | tidak valid |

Jawaban:

# pendefinisian fungsi validasi

def validasi(n):

return ...

# pendefinisian program utama

def main():

# input dan casting

nilai = ...

# pemanggilan fungsi

if validasi(...):

print("valid")

else:

print("tidak valid")

# pemanggilan program utama

if \_\_name\_\_ == '\_\_main\_\_':

main()print(...)

|  |  |
| --- | --- |
|  |  |

**Soal 8 Konversi rating**

Buatlah fungsi bernama **konversi\_rating** untuk mengkonversi rating dari bilangan menjadi string. Terdapat 1 parameter dengan jenis tipe data bilangan bulat n yang bernilai dari 1 hingga 5. Fungsi akan mengembalikan nilai string dengan ketentuan sebagai berikut:

* Untuk n = 1, akan di-return string "kurang sekali"
* Untuk n = 2, akan di-return string "kurang"
* Untuk n = 3, akan di-return string "sedang"
* Untuk n = 4, akan di-return string "bagus"
* Untuk n = 5, akan di-return string "bagus sekali"

Untuk selain 1 hingga 5, akan di-return string "invalid"

Contoh masukan dan keluaran:

| **Test** | **Keluaran** |
| --- | --- |
| h = konversi\_rating(1)  print(h) | kurang sekali |
| nilai = 2  h = konversi\_rating(nilai)  print(h) | kurang |

Jawaban:

|  |  |  |
| --- | --- | --- |
| Test 1: |  | Hasilnya: |
| Test 2: |  | Hasilnya: |

**Soal 10 Persamaan kuadrat**

Buatlah fungsi untuk menghitung akar persamaan kuadrat 𝑎𝑥2+𝑏𝑥+𝑐 = 0. Parameter fungsi ada 3, yaitu a, b, dan c yang merupakan koefisien variabel. Proses yang dilakukan adalah:

* Jika nilai diskriminan 𝐷=𝑏2−4𝑎𝑐 < 0 return string "tidak ada akar nyata"
* Jika nilai diskriminan 𝐷=𝑏2−4𝑎𝑐 = 0 return string "ada akar nyata, x = ..."
* Jika nilai diskriminan 𝐷=𝑏2−4𝑎𝑐 > 0 return string "ada akar nyata, x1 = ..., x2 = ...."

Petunjuk: Gunakan [rumus ABC](https://www.statmat.net/rumus-abc/) untuk menghitung akar.

Jawaban:

# pendefinisian fungsi

def akar\_pers\_kuadrat(a, b, c):

D = ...

if D ... ...:

return "Tidak ada akar nyata"

elif D ... ...:

x = ...

return "Ada akar nyata, x = " + str(x)

else:

x1 = ...

x2 = ...

return "Ada akar nyata, x1 = " + str(x1) + " x2 = " + str(x2)

|  |  |
| --- | --- |
|  |  |

**Soal 11 Gaji Mingguan**

Buatlah fungsi untuk menghitung berapa gaji yang diterima pegawai mingguan. Parameter fungsi ada 2, yaitu  total jam kerja dalam seminggu dan jenis pegawai ("A"/"B"). Aturan penggajian adalah sebagai berikut:

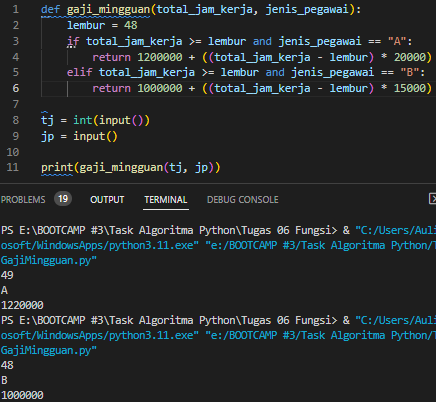
* Lembur dihitung jika jam kerja lebih dari 48 jam.
* Untuk pegawai "A", gaji dasarnya adalah Rp 1.200.000 setiap minggunya. Setiap jam lembur diberi gaji tambahan Rp 20.000,- per jam.
* Untuk pegawai "B", gaji dasarnya adalah Rp 1.000.000 setiap minggunya. Setiap jam lembur diberi gaji tambahan Rp 15.000,- per jam.

Nilai yang dikembalikan (return value) adalah bilangan bulat,

Contoh masukan dan keluaran:

| **Test** | **Masukan** | **Keluaran** |
| --- | --- | --- |
| tj = int(input())  jp = input()  print(gaji\_mingguan(tj, jp)) | 49  A | 1220000 |
| tj = int(input())  jp = input()  print(gaji\_mingguan(tj, jp)) | 48  B | 1000000 |

Jawaban:



**Soal 11 Tabungan Jumat**

Buatlah program dengan menggunakan fungsi untuk menghitung besaran diskon. Fungsi memiliki 1 parameter  berupa bilangan bulat yang menyatakan total belanja. Diskon akan diberikan sebesar 15% jika total belanja minimal Rp 250.000,00.

**Masukan** program berupa total belanja.

**Keluaran** program berupa besarnya diskon.

Contoh masukan dan keluaran:

| **Masukan** | **Keluaran** |
| --- | --- |
| 250000 | 37500.0 |
| 24999 | 0 |

Jawaban:

|  |  |
| --- | --- |
|  |  |

**Soal 12 Parkir**

Buatlah fungsi untuk menghitung besar biaya parkir di suatu swalayan. Fungsi memiliki 2 parameter, yaitu jenis\_kendaraan  dan lama\_parkir dalam jam. Aturan biaya parkirnya sebagai berikut:

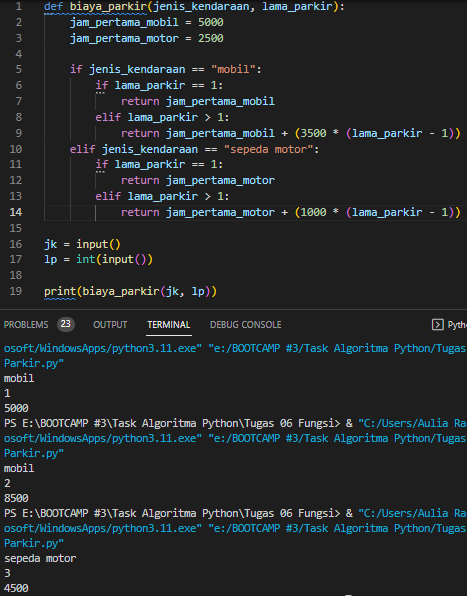
* mobil untuk jam pertama Rp 5.000,-. Untuk jam selanjutnya Rp 3.500,-.
* sepeda motor untuk jam pertama Rp 2.500,-. Untuk jam selanjutnya Rp 1.000,-

Fungsi akan me-return biaya parkir.

Contoh masukan dan keluaran:

| **Test** | **Masukan** | **Keluaran** |
| --- | --- | --- |
| jk = input()  lp = int(input())  print(biaya\_parkir(jk, lp)) | mobil  1 | 5000 |
| jk = input()  lp = int(input())  print(biaya\_parkir(jk, lp)) | mobil  2 | 8500 |
| jk = input()  lp = int(input())  print(biaya\_parkir(jk, lp)) | sepeda motor  3 | 4500 |

Jawaban:



**Soal 13 Validasi email**

Buatlah fungsi untuk mengetahui apakah string tertentu berupa email memiliki 1 buah simbol '@' atau tidak. Fungsi memiliki 1 parameter, yaitu email. Fungsi akan me-return nilai boolean True jika tepat terdapat 1 simbol '@', selain dari itu False.

Contoh masukan dan keluaran:

| **Test** | **Input** | **Keluaran** |
| --- | --- | --- |
| email = input()  print(validasi\_email(email)) | mahasiswa@telkomuniversity.com | True |
| print(validasi\_email('@admin@wkwkwk.com')) |  | False |
| print(validasi\_email('admin@wkwkwk.com')) |  | True |
| print(validasi\_email('adminwkwkwk.com')) |  | False |

Jawaban:

|  |  |
| --- | --- |
|  |  |

**Soal 14 Huruf hidup**

Buatlah fungsi untuk menghitung banyaknya huruf hidup pada string. Terdapat 1 parameter dengan jenis tipe data string. Fungsi akan mengembalikan nilai berupa bilangan bulat yang menyatakan banyak huruf hidup.

Contoh masukan dan keluaran:

| **Test** | **Keluaran** |
| --- | --- |
| print(banyak\_huruf\_hidup("bermain")) | 3 |
| string = "hello, world!"  print(banyak\_huruf\_hidup(string)) | 3 |

Jawaban:

|  |  |
| --- | --- |
|  |  |

**Soal 15 Konsonan**

Buatlah fungsi untuk menghitung banyaknya konsonan pada string. Terdapat 1 parameter dengan jenis tipe data string. Fungsi akan mengembalikan nilai berupa bilangan bulat yang menyatakan banyaknya konsonan.

Contoh masukan dan keluaran:

| **Test** | **Keluaran** |
| --- | --- |
| print(banyak\_konsonan("bermain")) | 4 |
| string = "hello, world!"  print(banyak\_konsonan(string)) | 7 |

Jawaban:

|  |  |
| --- | --- |
|  |  |

**Soal 16 Cetak bilangan**

Buatlah fungsi untuk mencetak barisan bilangan dengan range tertentu. Fungsi memiliki 2 parameter  berupa bilangan bulat a dan b. Fungsi akan mencetak barisan bilangan dari a hingga b.

Misalkan jika a = 1 dan b = 3, maka fungsi akan mencetak 1 2 3.

Jawaban:

|  |  |
| --- | --- |
|  |  |

**Soal 17 Jauh dekat**

Buatlah fungsi bernama **apakah\_dekat** untuk menghitung jarak antara dua titik dalam koordinat kartesis 2 dimensi. Fungsi akan memiliki 4 parameter, yaitu x1, y1, x2, dan y2 yang akan diisi oleh titik A(x1, y1) dan B(x2, y2). Fungsi akan mengembalikan string 'dekat' jika jarak lebih kecil atau sama dengan 1, atau string 'jauh' jika jarak lebih dari 1.

For example:

| **Test** | **Masukan** | **Result** |
| --- | --- | --- |
| a1 = float(input())  b1 = float(input())  a2 = float(input())  b2 = float(input())  print(apakah\_dekat(a1, b1, a2, b2)) | 1.2  1.3  1.5  1.8 | dekat |
| a1 = float(input())  b1 = float(input())  a2 = float(input())  b2 = float(input())  print(apakah\_dekat(a1, b1, a2, b2)) | 70.30  55.35  70.30  60.33 | jauh |

Jawaban:

|  |  |
| --- | --- |
|  |  |

**Soal 17 Jumlah bilangan ganjil**

Buatlah fungsi untuk menjumlahkan bilangan ganjil pada suatu deret bilangan. Terdapat 2 parameter, yaitu bilangan bulat n dan bilangan bulat m. Fungsi akan mengembalikan nilai berupa jumlah bilangan ganjil dari n hingga m, dengan n < m.

Contoh masukan dan keluaran:

| **Test** | **Keluaran** |
| --- | --- |
| print(jumlah\_ganjil(1, 5)) | 9 |
| n = 3  m = 4  print(jumlah\_ganjil(n, m)) | 3 |

Jawaban:

