



Tutor Sebaya (1%) Pengenalannya Komputasi

*Stream Programming
Input/Output, dan Percabangan*

Made with love by Acads BPA STEI-K '23

Darrel Adinarya S.

19623199

LINE: darrelas

Z. Nayaka Athadiansyah

19623116

LINE: nayaka.zna

Petunjuk Teknis



Praktikum Pengkom gimana, sih?

- Praktikum dilaksanakan secara **daring** lewat **Microsoft Teams**.
- Sesi praktikum dibagi menjadi tiga (total 120 menit):
 1. Sesi tanya jawab selama 10 menit.
 2. **Pengerjaan praktikum selama 100 menit.**
 3. Waktu keluar mahasiswa selama 10 menit.
- Diwajibkan untuk **merekam layarnya** selama pengerjaan. Disarankan pakai **OBS**.
- Seluruh jawaban dikompres dalam file **.zip** lalu dikumpulkan pada pengkomitb23.site.
- Dilarang baca sumber dari luar kecuali modul praktikum Pengkom.
(Kecuali pada Praktikum 5 diperbolehkan untuk membaca dokumentasi library.)





Contoh Program

```
# NIM>Nama      : 19623199/Darrel Adinarya Sunanda
# Tanggal      : 29-02-2023
# Deskripsi    : Program Pertamaku di Python

# KAMUS
# bulat : integer
# koma  : float
# iyakah : boolean

# ALGORITMA
# Input
bulat = int(input("Masukkan Angka: ")) # Lalala ini comment
...
# Output
print(bulat)
```

Materi

Output





Fungsi `print()`

Dalam python, cara basic untuk menghasilkan output adalah menggunakan fungsi `print()`.

Contoh yang sering digunakan untuk menghasilkan output langsung:

```
print("Hello, World!")
```

Fungsi `print()` juga dapat digunakan untuk mengeluarkan sesuatu variabel.

Contoh:

```
x = "Interupsi, Mayor!"  
print(x)  
# Output: Interupsi, Mayor!
```

Intermezzo
Variabel





Apa sih variabel dalam coding?

Dalam python (dan juga bahasa programming lainnya), variabel digunakan untuk menyimpan suatu nilai dalam memori dan mengaitkannya dengan suatu nama.

Python adalah bahasa yang diketik secara dinamis, artinya **Python mengenali tipe variabel secara otomatis**. Ini memudahkan kita karena tidak lagi perlu untuk mendeklarasi tipe variabel di awal.

Namun, perlu diperhatikan juga bahwa artinya kita harus menyesuaikan tipe variabel ketika melakukan operasi tertentu antara variabel (dijelaskan dalam Tipe Data).

Cara untuk mendeklarasi variabel:

```
usia = 35  
nama = "Gibran"
```




Fungsi `print(f"string")`

Semenjak PEP 498 (update Python), dikenalkan format string baru yang sering disebut dengan F-string (karena ada huruf “f” di depannya, artinya “format”).

F-string ini bertujuan untuk memudahkan kita dalam menghasilkan output yang bergantung pada variabel lain.

Contoh:

```
lokasi = "Jatinangor"
print(f"Kampusku ada di {lokasi}.")
# Output: Kampusku ada di Jatinangor.
```

<https://www.geeksforgeeks.org/formatted-string-literals-f-strings-python/>



Parameter Fungsi `print()`

Selain itu, fungsi `print()` ini masih dapat dimodifikasi lebih lanjut. Salah satu modifikasi yang sering digunakan dalam praktikum adalah parameter `end="end"`.

Parameter "`end=`" akan mengubah akhiran (`end`) dari output kalian.

Memiliki nilai default "`\n`" (new line)

Cara mengimplementasi parameter dalam suatu fungsi adalah dengan menambahkan tanda koma (,) lalu diikuti dengan parameternya.

Contoh:

```
print("But would you lose?", end="")  
print("Nah, I'd Win.")
```

https://www.w3schools.com/python/ref_func_print.asp

Materi

Input





Gimana sih cara melakukan input?

Cara untuk melakukan input sangat mudah, cukup menggunakan fungsi `input()`.

Dalam fungsi `input()`, kita dapat memasukan sebuah string yang bertindak sebagai “perintah” atau “arahan” untuk user.

Contoh:

```
x = input("Masukkan nilai X: ")
print(x)
print(f"Nilai x adalah {x}")
```

Output:

```
Masukkan nilai X: 10
10
Nilai x adalah 10
```

Materi

Tipe Data



Apa sih yang dimaksud tipe data?

Ada beberapa macam tipe data, namun dalam Pengenalan Komputasi kita hanya akan banyak menggunakan tipe data berikut:

<code>bool</code>	Boolean	True atau False
<code>int</code>	Bilangan bulat	Seluruh bilangan bulat
<code>float</code>	Bilangan real	Seluruh bilangan real
<code>string</code>	Teks	Kumpulan Karakter

```
B = True # Boolean
```

```
I = 123456789012 # Bilangan bulat
```

```
R = 2.331973 # Bilangan real
```

```
S = "abc" # Teks
```

```
S = 'def'
```

Kalo udah ada **float**, kenapa masih harus pake **int**?
Bagaimana interaksi antara data **bilangan** dengan data **teks**?

Intermezzo

Operasi



1.4.1 Operator Aritmatika

Operator	Deskripsi	Contoh
+	Penjumlahan	2 + 3 bernilai 5
-	Pengurangan	1 - 8 bernilai -7
*	Perkalian	5 * 6 bernilai 30
/	Pembagian	13 / 5 bernilai 2.6
//	Pembagian (dibulatkan ke bawah)	13 // 5 bernilai 2
%	Sisa Bagi / Modulo	13 % 5 bernilai 3

1.4.4 Operator Logika

Operator	Deskripsi	Contoh True	Contoh False
and	Dan: True jika kedua operand True	(1 < 2) and (3 == 3)	(1 == 2) and (3 == 3)
or	Atau: True jika salah satu operand True	(1 < 2) or (4 == 3)	(3 < 2) or (2 == 3)
not	Negasi: True jika operand False	not (3 < 2)	not (1 < 2)

1.4.2 Operator Assignment

Operator	Deskripsi	Contoh
=	Assignment	N = 5
+=	Penjumlahan	N += 5, N akan ditambah 5.
-=	Pengurangan	N -= 5, N akan dikurang 5.
*=	Perkalian	N *= 5, N akan dikali 5.
/=	Pembagian	N /= 5, N akan dibagi 5.
//=	Pembagian (dibulatkan ke bawah)	N //= 5, N akan dibagi 5 (dibulatkan ke bawah).
%=	Sisa Bagi / Modulo	N %= 5, N akan dimodulo 5.

Operator dan Operasinya

Merupakan kegiatan utama yang dilakukan dalam program, mulai dari tambah-kurang hingga kali-bagi. Inilah proses yang kalian akan rancang untuk menyelesaikan permasalahan dalam soal.

1.4.3 Operator Relasional

Operator	Deskripsi	Contoh True	Contoh False
==	Sama dengan	2 == 2	2 == 3
!=	Tidak sama dengan	3 != 2	3 != 3
<	Kurang dari	2 < 3	2 < 2
>	Lebih dari	3 > 2	2 > 3
<=	Kurang dari sama dengan	2 <= 2	3 <= 1
>=	Lebih dari sama dengan	6 >= 5	2 >= 4

Jangan dihafal!
Cukup pahami dan
Sontek syntax-nya!

Materi

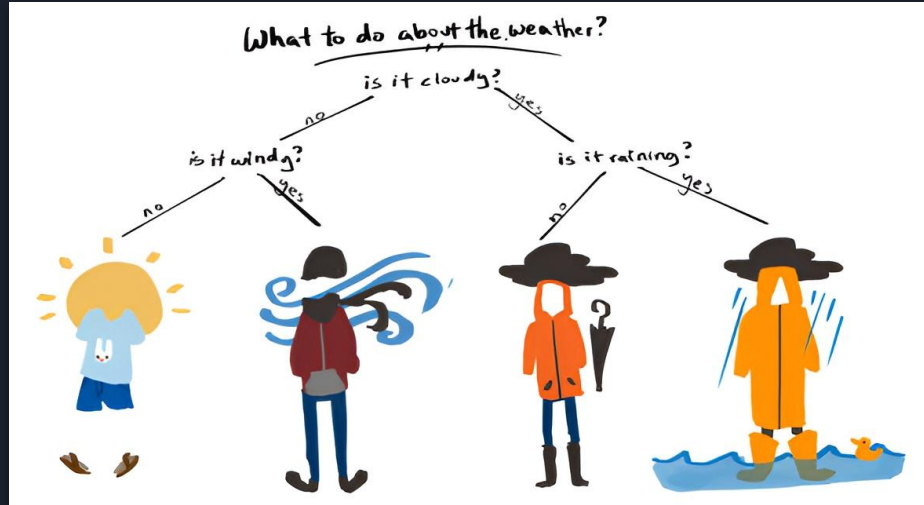
Percabangan



Maksudnya percabangan gimana sih?

Pada modul pertama ini, kalian akan fokus pada soal-soal percabangan.

Secara singkat, soal percabangan membahas tentang kasus yang memiliki hasil berbeda tergantung pada kondisi awal atau inputnya.



<https://micamaps.wordpress.com/2013/04/10/1299/>



Statement **if**

Dalam pemrograman komputer, **if** merupakan pernyataan bersyarat yang digunakan untuk mengeksekusi blok kode hanya ketika kondisi tertentu terpenuhi.

Dalam Python, perlu diperhatikan bahwa suatu blok kode dibedakan oleh indentasi-nya (seberapa dalam ke kanan teksnya) yang memisahkan suatu bagian dengan yang lainnya.

Sintaks:

```
if (<syarat-1>) :
```

```
    <aksi-1>
```

Kalau <syarat-1> terpenuhi,
<aksi-1> dilakukan.

Kalau nggak? Lanjut ke bagian
kode setelahnya.



Statement **else**

Pernyataan **else** digunakan untuk mengcover kemungkinan apabila kondisi/syarat pada pernyataan **if** tidak terpenuhi.

Sintaks:

```
if (<syarat-1>) :  
    <aksi-1>  
else :  
    <aksi-2>
```

Kalau <syarat-1> terpenuhi,
<aksi-1> dilakukan.

Kalau nggak? Lakukan <aksi-2>.



Statement `elif`

Pernyataan `elif` seperti namanya merupakan gabungan dari pernyataan `else` dan `if`.

Dalam arti, pernyataan `elif` akan berjalan apabila kondisi/syarat `if` sebelumnya tidak terpenuhi, lalu akan mengecek kondisi/syarat baru untuk menjalankan aksinya.

Sintaks:

```
if (<syarat-1>):  
    <aksi-1>  
elif (<syarat-2>):  
    <aksi-2>  
elif (<syarat-3>):  
    <aksi-3>  
...
```

Kalau <syarat-1> terpenuhi,
<aksi-1> dilakukan.

Kalau nggak? Cek apakah <syarat-2> terpenuhi.
Kalau ya, lakukan <aksi-2>.

Kalau nggak? Cek apakah <syarat-3> terpenuhi.
Kalau ya, lakukan <aksi-3>.

Bantai soal, kuy!



Problem 1:

Dasar I/O & If-Else

Nayaka sedang menyusun rangkaian paralel dengan 3 resistor yang berbeda. Nayaka ingin mengetahui nilai total hambatan rangkaian yang telah disusunnya dengan rumus

$$\frac{1}{R_{total}} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3}$$

Bantulah Nayaka menentukan nilai total hambatan rangkaian yang telah disusunnya! Masukkan nilai resistor dalam satuan ohm. Hambatan harus bernilai positif.

[Diadaptasi dari Praktikum Pengenalan Komputasi sesi 4.1 (STEI 2023)]

Test	Test	Case	Case	2:	Test	Case	1:	3:	
Masukkan hambatan	Masukkan hambatan	resistor 1:	2	Masukkan hambatan	resistor 1:	2	Masukkan hambatan	resistor 1:	
Masukkan hambatan	Masukkan hambatan	resistor 2:	2	Masukkan hambatan	resistor 2:	2	Masukkan hambatan	resistor 2:	
Masukkan hambatan	Masukkan hambatan	resistor 3:	3	Masukkan hambatan	resistor 3:	3	Masukkan hambatan	resistor 3:	
Hambatan total rangkaian adalah 1 ohm.				Hambatan total rangkaian adalah 0.75 ohm.				Hambatan tidak dapat dihitung.	

Solusi Problem 1:

Dasar I/O & If-Else

AWAS, JANGAN LANGSUNG HITUNG R TOTAL!
Gimana kalo salah satu dari R1, R2, atau R3 nilainya nol?

$$\frac{1}{R_{total}} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3}$$
$$R_{total} = 1 / \left(\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} \right)$$

```
1 r1 = int(input("Masukkan hambatan resistor 1: "))
2 r2 = int(input("Masukkan hambatan resistor 2: "))
3 r3 = int(input("Masukkan hambatan resistor 3: "))
4
5 # Proses
6 # r_total = 1/(1/r1 + 1/r2 + 1/r3)
7
8 if (r1 <= 0) or (r2 <= 0) or (r3 <= 0):
9     print("Hambatan tidak dapat dihitung.")
10 else: # (r1 >= 0) and (r2 >= 0) and (r3 >= 0)
11     r_total = 1/(1/r1 + 1/r2 + 1/r3)
12     print(f"Hambatan total rangkaian adalah {r_total:.2f} ohm.")
```


Problem 2: Dasar Percabangan

Karena Darrel mendapat nilai 100 di semua praktikum PengKom, Nayaka menghadihinya dua macam koin, yakni koin Nay sebanyak x buah dan koin Aka sebanyak y buah di mana Darrel harus memilih salah satu jenis koin. Tiap jenis koin punya nilai konversinya sendiri terhadap rupiah, yakni a dan b secara berturut-turut. Bantu Darrel memilih koin yang nominalnya **lebih besar** dalam rupiah. Nominal rupiah dari kedua koin dijamin tidak akan sama.


[Diadaptasi dari Praktikum Pengenalan Komputasi sesi 4.3 (STEI 2023)]

Test	Case	1:
Banyak koin Nay:		10
Banyak koin Aka:		20
Nilai konversi Nay ke rupiah:		3000
Nilai konversi Aka ke rupiah:		1000
Darrel lebih memilih koin Nay.		

#	Penjelasan:
koin Nay nilainya $10 \cdot 3000$	$= \text{Rp}30.000$
koin Aka nilainya $20 \cdot 1000$	$= \text{Rp}20.000$

Test	Case	2:
Banyak koin Nay:		15
Banyak koin Aka:		30
Nilai konversi Nay ke rupiah:		2000
Nilai konversi Aka ke rupiah:		2000
Darrel lebih memilih koin Aka.		

#	Penjelasan:
koin Nay nilainya $15 \cdot 2000$	$= \text{Rp}30.000$
koin Aka nilainya $30 \cdot 2000$	$= \text{Rp}60.000$



Solusi Problem 2: Dasar Percabangan

Ide: Cukup hitung nilai rupiah tiap koin lalu bandingkan mana yang lebih besar.

```
1 # Input
2 nay = int(input("Banyak koin Nay: "))
3 aka = int(input("Banyak koin Aka: "))
4 nay *= int(input("Nilai konversi Nay ke rupiah: "))
5 aka *= int(input("Nilai konversi Aka ke rupiah: "))
6
7 # Proses
8 if nay > aka:
9     print("Darrel lebih memilih koin Nay.")
10 else:
11     print("Darrel lebih memilih koin Aka.")
```



Problem 3: Prioritas dalam Percabangan (1)

Tuan Rel sudah kecanduan ngerakit PC sehingga ia ingin mengklasifikasikan komputer ke dalam kelompok kemampuan 1, 2, 3, 4, dan 5. Berdasarkan kemampuan CPU, GPU, dan hardisk dengan ketentuan sebagai berikut.

1. Jika kemampuan CPU lebih dari 7, kemampuan GPU lebih dari 7, **dan** kemampuan harddisk lebih dari 7, maka termasuk **kelompok 5**.
2. Jika setidaknya satu kemampuan bernilai kurang dari sama dengan 7, maka termasuk **kelompok 4**.
3. Jika nilai kemampuan CPU, GPU, **dan** harddisk kurang dari sama dengan 7, maka termasuk **kelompok 3**.
4. Jika nilai kemampuan CPU **atau** GPU kurang dari 5, maka termasuk **kelompok 2**.
5. Jika ada salah satu nilai kemampuan yang bernilai kurang dari 2, maka termasuk **kelompok 1** (tidak peduli dengan nilai kemampuan lainnya).

Prioritas kelompok terurut dari nomor **kelompok terkecil** hingga terbesar. Misalnya, apabila masukan cocok dengan kelompok 2 dan 4, maka yang dianggap benar adalah kelompok 2.

Solusi 3:

Prioritas dalam Percabangan (1)



```
1  # Input
2  cpu = int(input("Masukkan nilai kemampuan CPU: "))
3  gpu = int(input("Masukkan nilai kemampuan GPU: "))
4  hdd = int(input("Masukkan nilai kemampuan hard disk:
5  "))
6  # Proses
7  if (cpu < 2) or (gpu < 2) or (hdd < 2):
8      kelompok = 1
9  elif (cpu < 5) or (gpu < 5):
10     kelompok = 2
11  elif (cpu <= 7) and (gpu <= 7) and (hdd <= 7):
12     kelompok = 3
13  elif (cpu <= 7) or (gpu <= 7) or (hdd <= 7):
14     kelompok = 4
15  else:
16     kelompok = 5
17
18  print(f"PC Tuan Rel termasuk kelompok {kelompok}.")
```

not (cpu <= 7 or gpu <= 7 or hdd <= 7)
= cpu > 7 and gpu > 7 and hdd > 7
-> kelompok 5

Problem 4: Prioritas dalam Percabangan (2)

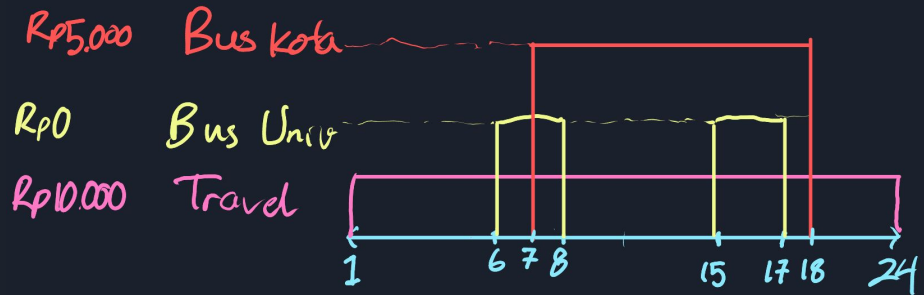
Mas Hanies punya beberapa pilihan transportasi untuk pergi ke Jatinangor dari Ciumbuleuit dan pulang kembali:

- Bus Kota: Harga Rp5.000. Beroperasi dari pukul 7–18.
- Bus Universitas: Gratis. Tersedia pada pukul 6–8 dan 15–17.
- Travel: Rp10.000. Selalu tersedia.

Ia harus menghemat uang karena harga beras kian melambung tinggi. Diberikan jam keberangkatan dan kepulangan Mas Hanies (berupa bilangan bulat 1–24), buatlah program untuk menentukan moda transportasi untuk berangkat dan pulang (bisa berbeda) dengan biaya termurah. [Diadaptasi dari Praktikum Pengenalan Komputasi shift 4.1 (STEI 2023)]

Test	Case	1:	Test	Case	2:	Test	Case	3:
Jam	keberangkatan:	14	Jam	keberangkatan:	8	Jam	keberangkatan:	2
Jam	kepulangan:	21	Jam	kepulangan:	9	Jam	kepulangan:	16
Mas Hanies berangkat naik Bus Kota dan pulang naik Travel dengan biaya 15000.			Mas Hanies berangkat naik Bus Universitas dan pulang naik Bus Kota dengan biaya 5000.			Mas Hanies berangkat naik Travel dan pulang naik Bus Universitas dengan biaya 10000.		

Strategi Problem 4: Prioritas dalam Percabangan (2)



Skema transportasi paling optimal:

- Kalo di antara pukul 6–8 atau 15–17, naik Bus Universitas aja soalnya gratis.
- Di rentang pukul 9–14 atau pada pukul 18, naiklah Bus Kota.
- Selebihnya naik Travel karena itu satu-satunya moda transportasi yang tersedia.

Kita bisa simpan informasi terkait moda transportasi yang Mas Hanies gunakan ke dalam suatu variabel, katakanlah **modaBerangkat** dan **modaPulang**.

Buatlah variabel **biaya** (mula-mula nol) yang menampung besar biaya total untuk perjalanan pulang-pergi.

Solusi Problem 4:

Prioritas dalam Percabangan (2)

```
1 # Input
2 berangkat = int(input("Jam keberangkatan: "))
3 pulang = int(input("Jam kepulangan: "))
4
5 # Inisialisasi Awal
6 biaya = 0
7
8 # Proses
9 ## Berangkat
10 if (6 <= berangkat <= 8) or (15 <= berangkat <= 17):
11     modaBerangkat = "Bus Universitas"
12 elif (9 <= berangkat <= 14) or (berangkat == 18):
13     modaBerangkat = "Bus Kota"
14     biaya += 5000
15 else:
16     modaBerangkat = "Travel"
17     biaya += 10000
18
19 ## Pulang
20 if (6 <= pulang <= 8) or (15 <= pulang <= 17):
21     modaPulang = "Bus Universitas"
22 elif (9 <= pulang <= 14) or (pulang == 18):
23     modaPulang = "Bus Kota"
24     biaya += 5000
25 else:
26     modaPulang = "Travel"
27     biaya += 10000
28
29 # Output
30 print(f"Mas Hanies berangkat naik {modaBerangkat} dan pulang naik {modaPulang} dengan biaya {biaya}."
```



Problem 5: Bermain-main dengan modulo dan div

Darrel memberi sebuah bilangan empat digit kepada Nayaka yang gabut untuk mengklasifikasikannya menjadi 4 tipe:

1. Bilangan **alfa** adalah bilangan yang setiap digitnya membentuk pola monoton naik (membesar setiap digitnya) atau monoton turun (mengecil setiap digitnya).
2. Bilangan **beta** adalah bilangan yang selisih dua digit pertamanya dengan dua digit terakhirnya minimal 30.
3. Bilangan **gamma** adalah bilangan yang memenuhi persyaratan bilangan alfa dan beta.
4. Bilangan **delta** adalah bilangan yang tidak memenuhi ketiga syarat diatas.

Bantulah Nayaka menentukan tipe bilangan yang Darrel berikan. [Diadaptasi dari Praktikum Pengenalan Komputasi sesi 1.1 (FTMD 2023)]

Test Case 1:
Masukkan bilangan empat digit: 1234
Alfa

Test Case 2:
Masukkan bilangan empat digit: 9812
Beta

Test Case 3:
Masukkan bilangan empat digit: 8731
Gamma

Test Case 4:
Masukkan bilangan empat digit: 1111
Delta

Test Case 5:
Masukkan bilangan empat digit: 1001
Delta



Strategi Problem 5: Bermain-main dengan modulo dan div

Sebagai gambaran, ambil contoh bilangan **1357**. Perhatikan beberapa operasi berikut:

```
>> 1357 // 10
```

```
135
```

```
>> 1357 % 10
```

```
7
```

```
>> 1357 // 100
```

```
13
```

```
>> 1357 % 100
```

```
57
```

```
>> 1357 // 1000
```

```
1
```

```
>> (1357 // 10) % 10
```

```
5
```

Bisa disimpulkan bahwa kalau bilangan 4-digitnya adalah n maka

- `digitPertama = n // 1000`
- `digitKetiga = n // 10 % 10`
- `digitKedua = n // 100 % 10`
- `digitKeempat = n % 10`

$$|x - y| > k \iff x - y > k \quad \vee \quad x - y < -k$$

Solusi Problem 5:

Bermain-main dengan modulo dan div

```
1 # Input
2 klmn = int(input("Masukkan bilangan empat digit: "))
3
4 # Inisialisasi Awal
5 alfa = False
6 beta = False
7
8 # Proses
9 k = klmn // 1000
10 l = klmn // 100 % 10
11 m = klmn // 10 % 10
12 n = klmn % 10
13
14 if (k < l < m < n) or (k > l > m > n):
15     alfa = True
16 if (klmn // 100) - (klmn % 100) > 30 or (klmn // 100) - (klmn % 100) < -30:
17     beta = True
18
19 # Output
20 if alfa and beta:
21     print("Gamma")
22 elif alfa:
23     print("Alfa")
24 elif beta:
25     print("Beta")
26 else:
27     print("Delta")
```



Problem 6: Percabangan Level Tinggi (Nguli)

Tuan Nay memiliki bakso sebanyak N , serta siomay sebanyak M , yang akan ia gunakan untuk meracik seblak. Tuan Nay memiliki 3 resep seblak yang masing masing membutuhkan bahan yang berbeda, yaitu

1. Seblak Mini : 5 Bakso, 3 Siomay
2. Seblak Gacor : 7 Bakso, 5 Siomay
3. Seblak MaxWin : 10 Bakso, 7 Siomay

Tuan Nay ingin membuat sebanyak mungkin seblak dengan prioritas pembuatan adalah seblak MaxWin. Setelah tidak bisa membuat seblak MaxWin, baru ia akan membuat seblakGacor, dan yang terakhir, Seblak Mini. Ia juga ingin bahwa setiap seblak dibuat setidaknya 1 porsi. Tentukan banyaknya seblak yang dapat ia buat untuk masing masing jenis seblak!

[Diadaptasi dari Praktikum Pengenalan Komputasi sesi 4.3 (STEI 2023)]

Problem 6:

Percabangan Level Tinggi (Nguli)

```
1 # Input
2 bakso = int(input("Banyaknya bakso: "))
3 siomay = int(input("Banyaknya siomay: "))
4
5 # Proses
6 # Pastikan bisa bikin satu porsi seblak untuk tiap tipe yang ad
7 bakso -= 22
8 siomay -= 15
9 #
10 mini = 0
11 gacor = 0
12 maxwin = 0
13
14 if (bakso <= 0) or (siomay <= 0):
15     print("Tuan Nay ngambek.")
16 else:
17     mini = 1
18     gacor = 1
19     maxwin = 1
20     # MaxWin
21     if bakso >= 10 and siomay >= 7:
22         if bakso // 10 < siomay // 7:
23             n = bakso // 10
24             maxwin += n
25             bakso -= 10 * n
26             siomay -= 7 * n
27         else: # siomay // 7 <= bakso // 10
28             n = siomay // 7
29             maxwin += n
30             bakso -= 10 * n
31             siomay -= 7 * n
```

```
1 # Gacor
2 if bakso >= 7 and siomay >= 5:
3     if bakso // 7 < siomay // 5:
4         n = bakso // 7
5         gacor += n
6         bakso -= 7 * n
7         siomay -= 5 * n
8     else: # siomay // 5 <= bakso // 7
9         n = siomay // 5
10        gacor += n
11        bakso -= 7 * n
12        siomay -= 5 * n
13
14 # Mini
15 if bakso >= 5 and siomay >= 3:
16     if bakso // 5 < siomay // 3:
17         n = bakso // 5
18         mini += n
19         bakso -= 5 * n
20         siomay -= 3 * n
21     else: # siomay // 3 <= bakso // 5
22         n = siomay // 3
23         mini += n
24         bakso -= 5 * n
25         siomay -= 3 * n
26
27 # Output
28 print(f"Tuan Nay membuat {mini} porsi seblak Mini, {gacor} porsi seblak Gacor, dan {maxwin} porsi seblak MaxWi
n.")
```



Problem 7: Soal Tantangan

Tuan Nay akhirnya menjadi dosen! Di kelasnya ada M buah meja dengan tiap mejanya maksimal ditempati dua orang. Ada sebanyak A orang mahasiswa dan B orang mahasiswi.

Tiba tiba tuan Nay berpikir, jika seluruh mahasiswa ($A + B$ mahasiswa) duduk pada meja-meja tersebut, dengan posisi yang dapat diatur secara bebas, berapa maksimum dan minimum banyaknya meja yang ditempati oleh seorang mahasiswa dan mahasiswi? Sebagai dosen yang baik, ia pun menyerahkan tugas ini kepada Anda. Selesaikan tugas yang diberikan oleh Tuan Nay!

[Diadaptasi dari TROC-33 Soal B)



Problem 7:

Percabangan Level Tinggi (Nguli)

Semangat!

Makasih udah mau belajar bareng malam ini.

Selamat beristirahat! :D

