

PEMBAHASAN QUIZ
PRAKTIKUM PEMROGRAMAN KOMPUTER
PERTEMUAN 10



Disusun oleh:

Nama : Hanan Fijananto
NIM : 24/538946/SV/24555
Kelas : A2
Dosen Pengampu : Firma Syahrian, S.Kom., M.Cs.

PROGRAM STUDI D-IV TEKNOLOGI REKAYASA PERANGKAT LUNAK
DEPARTEMEN TEKNIK ELEKTRO DAN INFORMATIKA
SEKOLAH VOKASI
UNIVERSITAS GADJAH MADA
YOGYAKARTA
2024

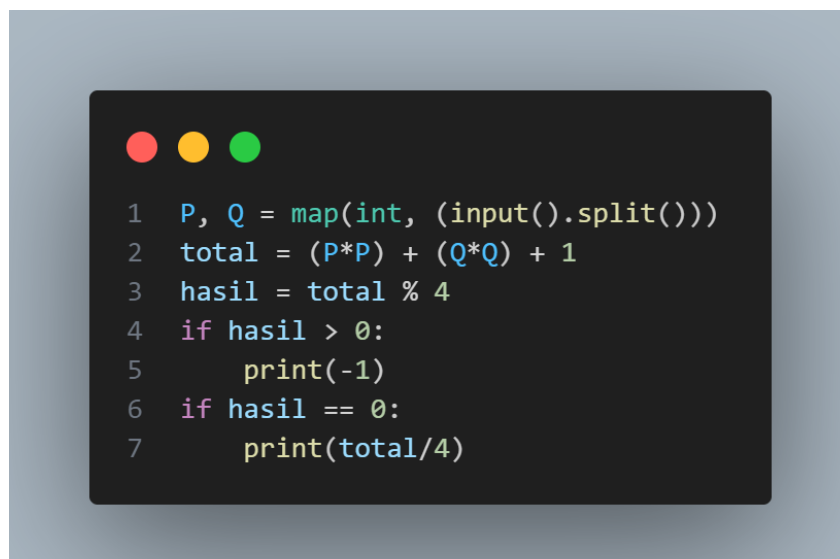
Hasil dan Pembahasan

1. Bunga Gabungan

Diberikan soal:

Di hari pertama, Antileven membeli P ikatan bunga (setiap ikatan berisi P bunga). Di hari kedua, Antileven membeli Q kotak bunga, setiap kotak berisi Q bunga. Di hari ketiga, Antileven membeli 1 bunga. Setelah itu banyaknya bunga yang dibeli Antileven dibagi sama rata ke 4 temannya. Akan dibuat program untuk menghitung banyak bunga yang diberikan ke salah satu temannya. Jika banyaknya bunga yang diberikan ke masing-masing temannya tidak dapat dibagi rata, maka outputnya akan -1.

Solusi:



```
1 P, Q = map(int, (input().split()))
2 total = (P*P) + (Q*Q) + 1
3 hasil = total % 4
4 if hasil > 0:
5     print(-1)
6 if hasil == 0:
7     print(total/4)
```

Gambar 1. 1 Bunga Gabungan

Berdasarkan gambar 1.1, langkah awalnya yaitu membuat variabel input berupa P dan Q yang di-split dan di-map agar berupa integer. Setelah itu membuat variabel total yang berfungsi untuk menghitung banyaknya bunga yang dibeli oleh Antileven. $P * P$ untuk menghitung total bunga yang dibeli Antileven di hari pertama, setelah itu $Q * Q$ untuk menghitung total bunga yang dibeli di hari kedua. Terakhir ditambah 1 sesuai pembelian bunga di hari ketiga. Setelah menghitung total bunga yang dibeli Antileven,

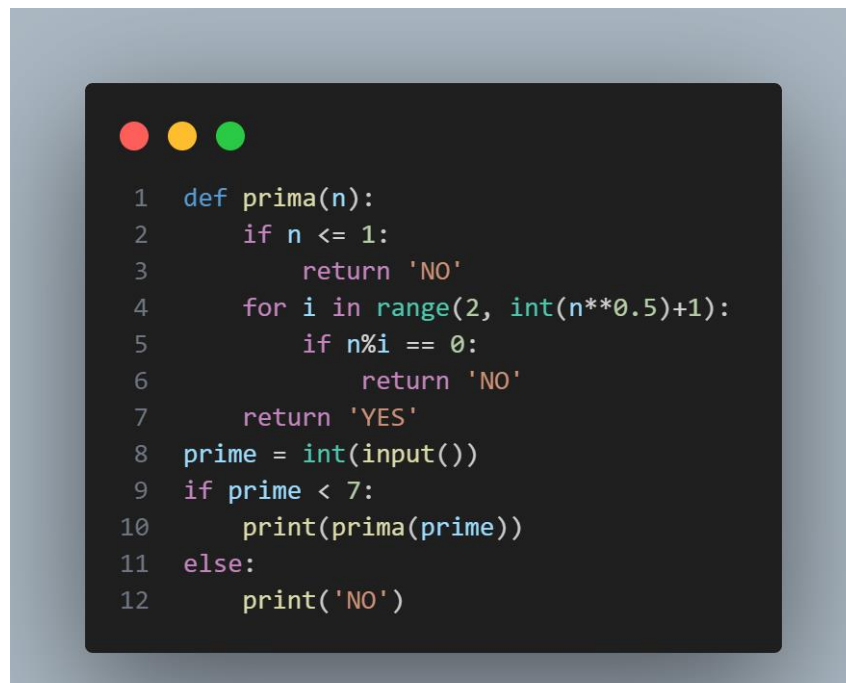
buat variabel hasil yang akan menjadi parameter dengan cara variabel total di-modulo dengan 4. Jika variabel hasil lebih dari 0, maka keluarkan output -1. Jika variabel hasil sama dengan 0 (yang artinya total bunga yang dibeli Antileven dapat dibagi sama rata) maka keluarkan variabel total yang dibagi dengan 4 agar dapat melihat banyak bunga yang diberikan ke salah satu temannya.

2. Angka Kesukaan

Diberikan soal:

Tzaph mencintai bilangan prima, sedangkan Nevest mencintai bilangan yang kurang dari 7. Tentukan apakah mereka berdua mencintai bilangan bulat N. Output akan berisi YES jika mereka berdua sama-sama mencintai bilangan bulat N, atau NO apabila sebaliknya.

Solusi:



```
1 def prima(n):
2     if n <= 1:
3         return 'NO'
4     for i in range(2, int(n**0.5)+1):
5         if n%i == 0:
6             return 'NO'
7     return 'YES'
8 prime = int(input())
9 if prime < 7:
10     print(prima(prime))
11 else:
12     print('NO')
```

Gambar 1. 2 Angka Kesukaan

Berdasarkan gambar 1.2, Untuk mengecek apakah bilangan yang di-input bernilai prima atau tidak maka dibuat fungsi def prima(n). Jika n kurang dari sama dengan satu, maka mengembalikan nilai NO karena bilangan prima. Jika tidak, maka jalankan iterasi for i in

range 2 hingga akar n, jika n di-modulo dengan i sama dengan 0, maka return NO karena jika ada pembagi selain 1 dan n, maka bukan bilangan prima. Terakhir return NO untuk selain if dan iterasi di atas (artinya bilangan prima).

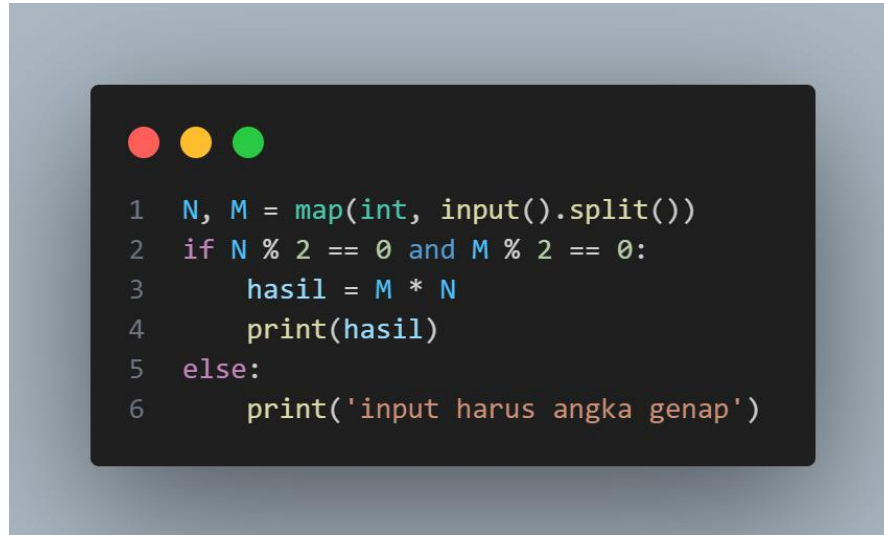
Setelah itu buat variabel prime yang digunakan untuk menginput angka. Jika variabel prime kurang dari 7, maka panggil fungsi prima(prime). Jika tidak, maka print NO

3. Permainan Ubin

Diberikan soal:

Nevest memiliki banyak segitiga sama kaki dengan panjang alas 2 dan tinggi 1. Nevest juga mempunyai sebuah persegi panjang dengan panjang sisi N dan M, dengan N dan M merupakan bilangan genap. Nevest ingin menyusun segitiga tersebut sedemikian rupa sehingga persegi panjang tersebut terisi penuh oleh segitiga. Hitung banyak segitiga minimum yang diperlukan Nevest.

Solusi:



Gambar 1. 3 Permainan Ubin

Berdasarkan gambar 1.3, Langkah awal program yaitu dengan membuat variabel input N dan M yang di-map dan di-split lalu diganti ke dalam integer. N dan M sebagai panjang sisi persegi panjang. Setelah itu, buat percabangan if untuk menentukan apakah bilangan yang di-input berupa bilangan genap dengan cara M dan N di-dimodulokan dengan 2. Jika

hasilnya 0, maka jalankan program berikut: variabel hasil yang berisi perkalian antara M dan N. Hal tersebut dikarenakan banyak segitiga (dengan panjang alas 2 dan tinggi 1) yang dimasukkan ke persegi panjang dengan sisi genap pasti perkalian dari kedua panjang sisi

Contoh Masukan 1

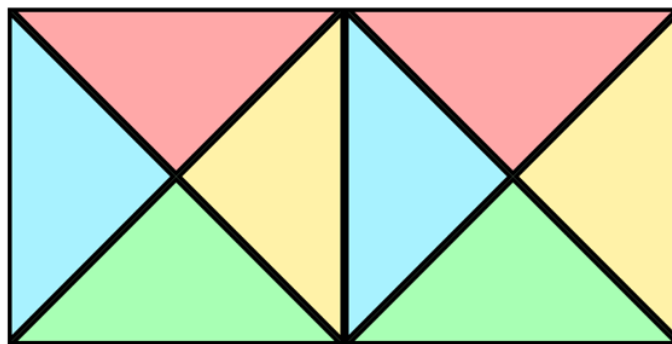
2 4

Contoh Keluaran 1

8

Penjelasan

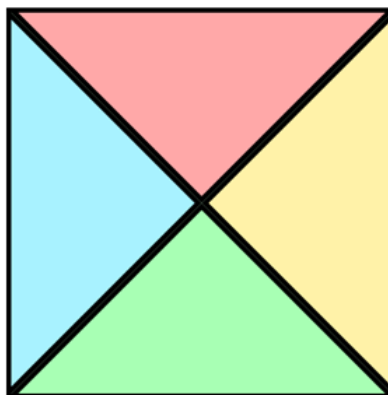
Pada contoh masukan pertama, berikut ilustrasi pemasangan segitiga yang dilakukan.



Gambar 1. 4 contoh

persegi panjang tersebut. Contoh dapat dilihat pada gambar 1.4.

Berdasarkan gambar tersebut, jika panjang sisi sisi persegi panjang 2 dan 4, maka banyaknya segitiga adalah 8. Jika persegi tersebut dipecah menjadi ukuran 2 x 2 maka banyaknya segitiga menjadi 4. Berdasarkan logika tersebut, dapat disimpulkan bahwa



Gambar 1. 5 pembuktian logika

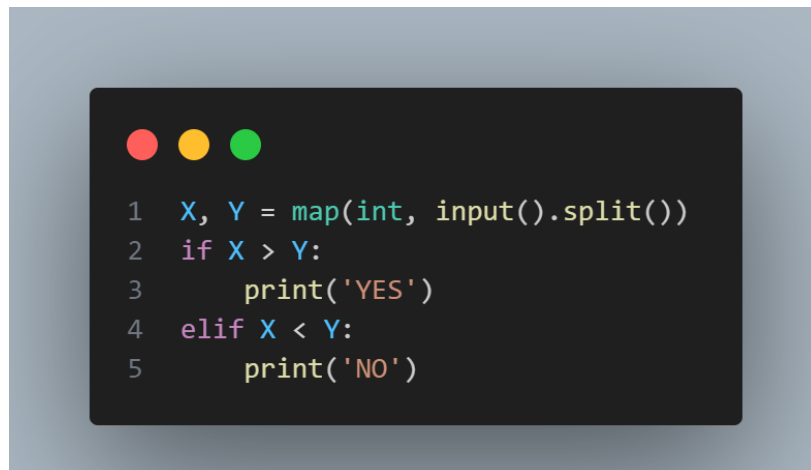
banyaknya segitiga yang dapat dimasukkan ke dalam persegi panjang sama dengan luas persegi panjang tersebut.

4. Penemuan Palu

Diberikan soal:

Palu ditemukan pada tahun X Sebelum Masehi. Apakah orang-orang bisa menggunakan palu pada tahun Y Sebelum Masehi? Catatan: orang-orang bisa menggunakan palu jika dan hanya jika palu sudah ditemukan. Sebuah baris berisi YES jika orang-orang bisa menggunakan palu pada tahun YY Sebelum Masehi, atau NO bila sebaliknya.

Solusi:



```
1 X, Y = map(int, input().split())
2 if X > Y:
3     print('YES')
4 elif X < Y:
5     print('NO')
```

Gambar 1. 6 Penemuan Palu

Buat variabel input X dan Y yang di-split lalu di-map diganti tipe datanya menjadi integer. Variabel X sebagai tahun ditemukannya palu sebelum masehi. Sedangkan variabel Y sebagai tahun orang menggunakan palu sebelum masehi. Tahun sebelum masehi berkebalikan dengan tahun setelah masehi, artinya semakin besar tahunnya, maka semakin lama usianya. Dibuat percabangan if $X > Y$: maka print YES. Artinya jika orang yang menemukan palu berada pada tahun yang lebih besar daripada orang yang menggunakan palu tersebut, maka logika tersebut benar. Jika X kurang dari Y maka berlaku sebaliknya.

5. Pembatasan Dunia

Diberikan soal:

Ivo berencana untuk membuat jam digital Oivworld. Ivo tahu bahwa di Oivworld, waktu tidak dinyatakan dalam jam dan menit, melainkan dalam *maj* dan *netim*. Untungnya, Ivo memiliki buku petunjuk yang memiliki informasi tentang waktu di Oivworld, yaitu:

- satu *maj* sama dengan AA *netim*;
- format waktu di Oivworld adalah NN *maj* MM *netim*, dengan $0 \leq M < A0 \leq M < A$.

Ivo pun bertanya kepada Anda, seorang penghuni Oivworld. Jika waktu telah berlalu selama XX *netim*, bagaimana format waktunya dalam NN *maj* MM *netim*?

Solusi:

Format Keluaran

Sebuah baris yang berisi dua buah bilangan NN dan MM , yang menyatakan waktu di Oivworld telah berjalan selama NN *maj* MM *netim*.

Contoh Masukan 1

2 5

Contoh Keluaran 1

2 1

Contoh Masukan 2

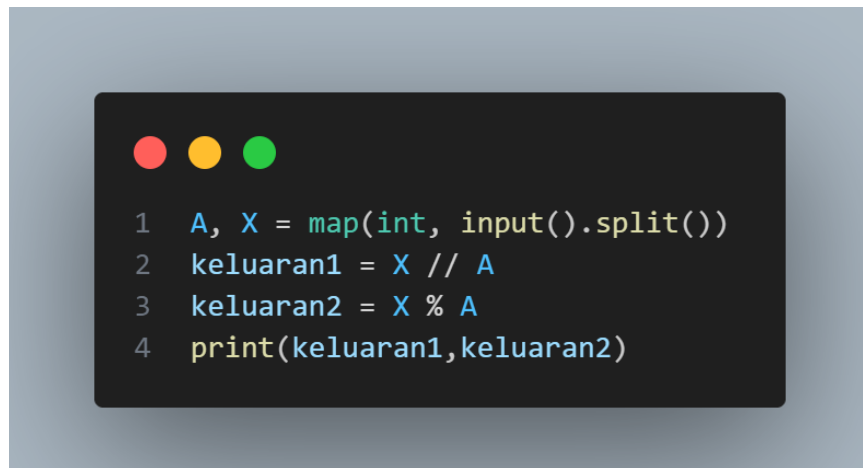
4 20

Contoh Keluaran 2

5 0

Gambar 1. 7 Contoh output

Berdasarkan contoh 1, jika user meng-*input* 2 5, maka outputnya akan 2 1. Artinya, sesuai soal, jika 1 jam adalah 2 menit, maka 5 menit sama dengan 2 jam lebih 1 menit. Begitu juga dengan contoh 2. Jika user meng-*input* 4 20, maka outputnya akan 5 0. Hal tersebut dikarenakan 1 jam adalah 4 menit, maka 20 menit sama dengan 5 jam lebih 0 menit. Dapat disimpulkan bahwa N maj dapat dicari dengan membagi X dengan A . Sedangkan M dapat dicari dengan hasil bagi antara X dengan A .



Gambar 1. 8 solusi

Membuat variabel input A dan X yang telah diubah menjadi integer. Selanjutnya buat variabel keluaran1 sebagai N yang berisi operasi perhitungan X dibagi bulat dengan A. Sedangkan variabel keluaran2 yang berisi hasil bagi dari X dengan A. Lalu print keluaran 1 dan keluaran 2.

Bukti pengisian survei:

