

# Latihan Percabangan

## Tutorial 2 [G, F] - Dasar-Dasar Pemrograman 1 Gasal 2019/2020

Selamat datang di Tutorial 2 DDP1! Pada sesi tutorial kali ini, kita akan mengerjakan soal latihan pemrograman yang memanfaatkan konsep program control/branching. Ingat, kalian sudah berkenalan dengan konsep ini minggu lalu, masih ingat *kan*? Karena kalian sudah berkenalan dengan konsep ini minggu lalu, maka tidak akan ada *step by step* pada tutorial ini. Semangat!

**Mohon kumpulkan semua file jawaban Anda dalam bentuk zip dengan format Lab2\_Nama\_NPM\_KodeAsdos.zip. Contoh: Lab2\_WindiChandra\_1606862721\_YE.zip**

## Latihan #7: Persegi dan Titik (latihan7.py, skor: +75)

Konsep penting: variabel, branching

Buatlah program yang dapat menentukan apakah sebuah titik berada di dalam suatu persegi atau tidak!

Program Anda akan menerima masukan berikut:

1. Panjang sisi persegi,
2. Koordinat titik kiri bawah dari persegi, dan
3. Koordinat dari titik yang akan dicek.

Sistem koordinat yang digunakan adalah sistem koordinat kartesius, yang artinya titik pusat berada pada titik (0,0) dan arah ke kanan adalah arah X positif dan arah ke atas merupakan arah Y positif.

Tampilkan 'Di dalam' apabila titik berada di dalam persegi, 'Di tepi' apabila titik berada di sisi persegi, dan 'Di luar' apabila titik berada di luar persegi.

### Contoh Program

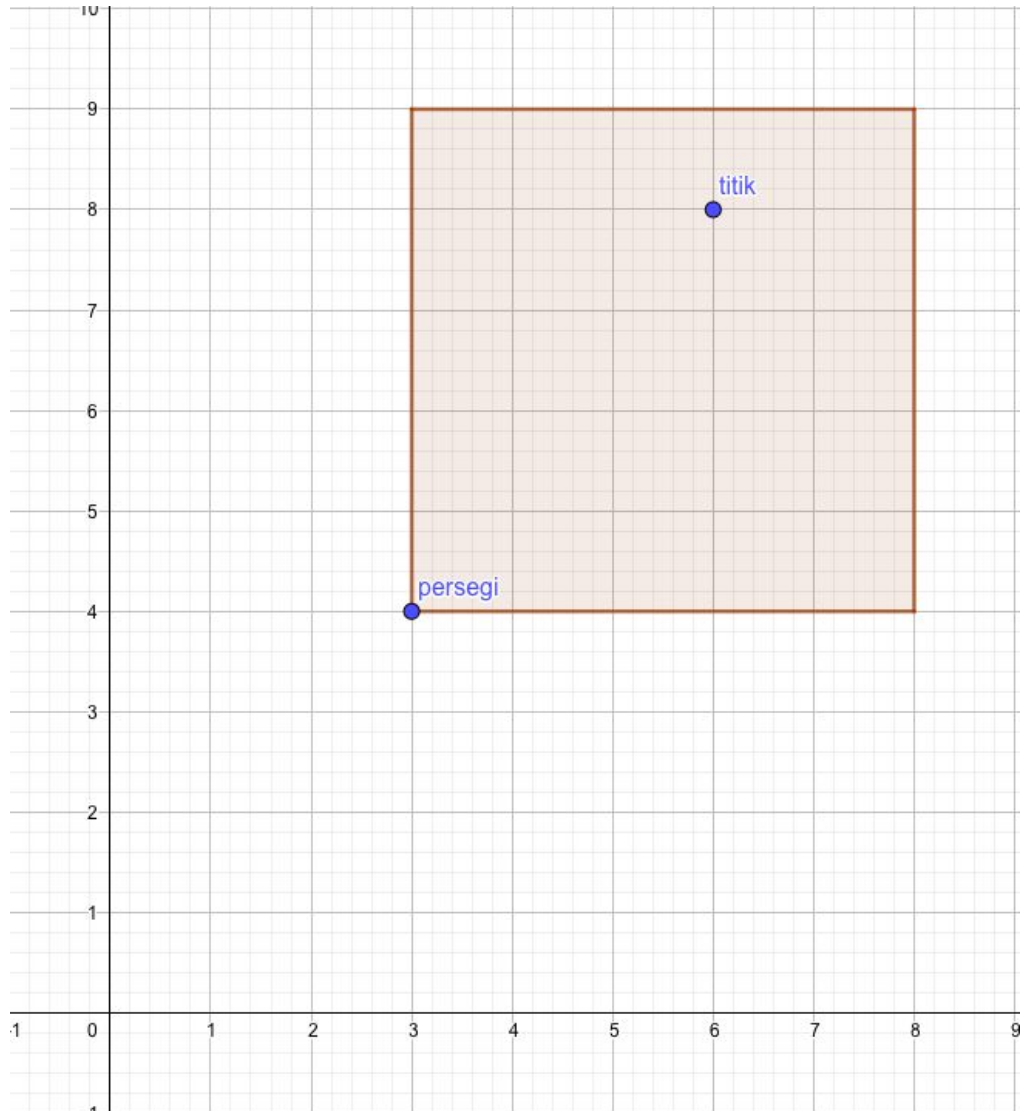
```
Masukkan panjang sisi persegi: 5
Masukkan komponen X dari titik kiri bawah persegi: 3
Masukkan komponen Y dari titik kiri bawah persegi: 4
Masukkan komponen X dari titik yang ingin dicek: 6
Masukkan komponen Y dari titik yang ingin dicek: 8
```

Tulisan yang ditandai dengan warna biru adalah masukan dari pengguna.

Apabila diberi masukan seperti di atas, program akan mengeluarkan:

```
Di dalam
```

Persegi dan titik divisualisasikan oleh gambar berikut:



Terlihat bahwa titik berada di dalam persegi.

Contoh lain:

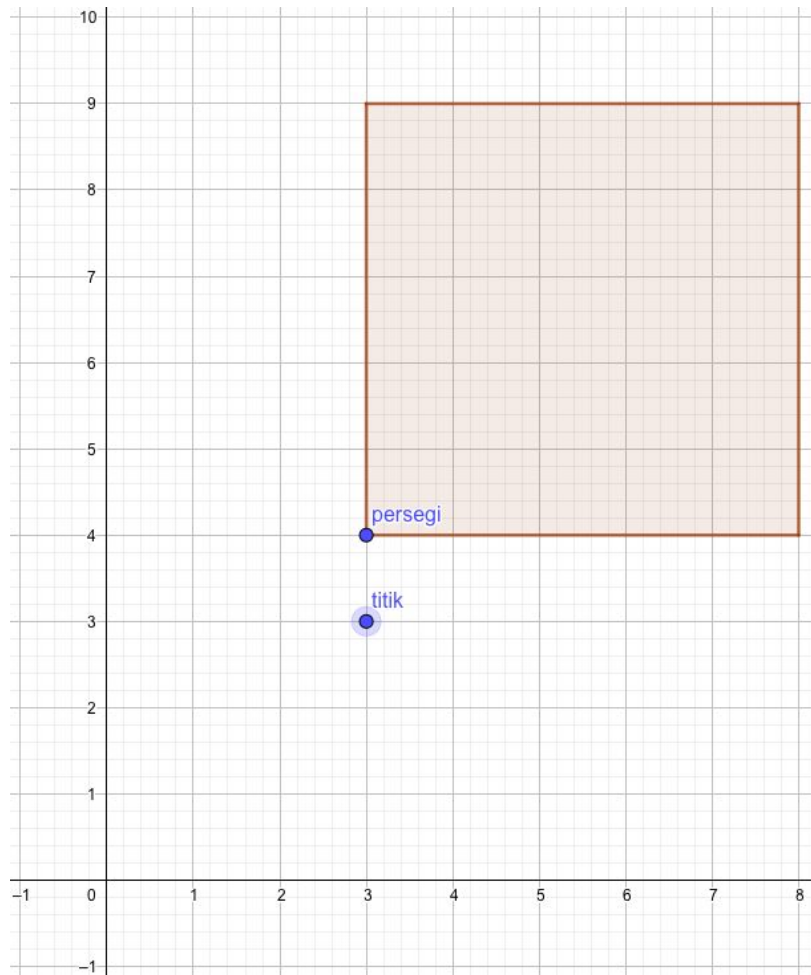
```
Masukkan panjang sisi persegi: 5
Masukkan komponen X dari titik kiri bawah persegi: 3
Masukkan komponen Y dari titik kiri bawah persegi: 4
Masukkan komponen X dari titik yang ingin dicek: 3
Masukkan komponen Y dari titik yang ingin dicek: 3
```

Tulisan yang ditandai dengan warna biru adalah masukan dari pengguna.

Apabila diberi masukan seperti di atas, program akan mengeluarkan:

Di luar

Persegi dan titik divisualisasikan oleh gambar berikut:



Terlihat bahwa titik berada di luar persegi.

Contoh lain:

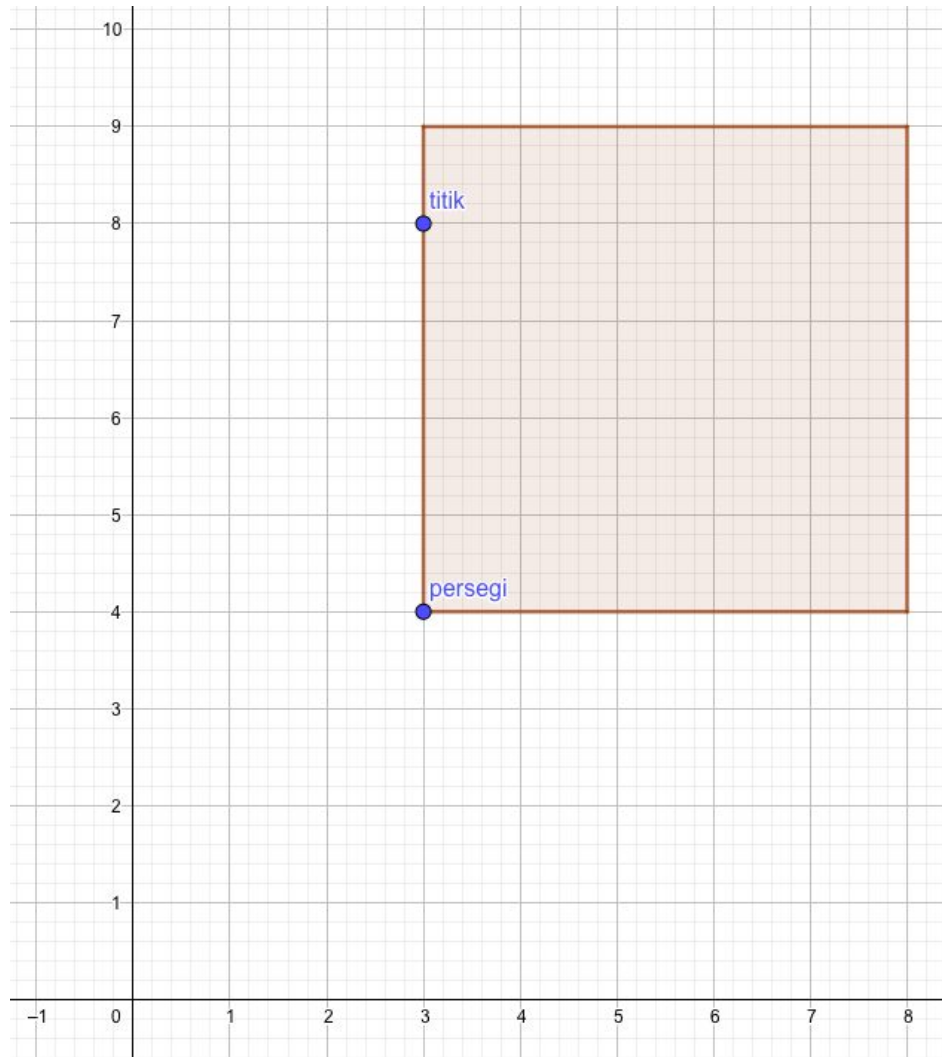
Masukkan panjang sisi persegi: 5  
 Masukkan komponen X dari titik kiri bawah persegi: 3  
 Masukkan komponen Y dari titik kiri bawah persegi: 4  
 Masukkan komponen X dari titik yang ingin dicek: 3  
 Masukkan komponen Y dari titik yang ingin dicek: 8

Tulisan yang ditandai dengan warna biru adalah masukan dari pengguna.

Apabila diberi masukan seperti di atas, program akan mengeluarkan:

Di tepi

Persegi dan titik divisualisasikan oleh gambar berikut:



Terlihat bahwa titik berada di tepi/sisi persegi.

## Latihan #8: Dua Persegi (latihan8.py, skor: +25)

Konsep penting: variabel, branching

Buatlah program yang dapat menentukan apakah dua buah persegi dengan panjang sisi yang sama beririsan atau tidak!

Program Anda akan menerima tiga buah masukan:

1. Panjang sisi kedua persegi,
2. Koordinat titik kiri bawah dari persegi pertama, dan
3. Koordinat titik kiri bawah dari persegi kedua.

Sistem koordinat yang digunakan adalah sistem koordinat kartesius, yang artinya titik pusat berada pada titik (0,0) dan arah ke kanan adalah arah X positif dan arah ke atas merupakan arah Y positif.

Tampilkan 'Persegi beririsan' apabila kedua buah persegi mempunyai irisan (atau menempel), atau 'Persegi terpisah' apabila tidak.

### Contoh Program

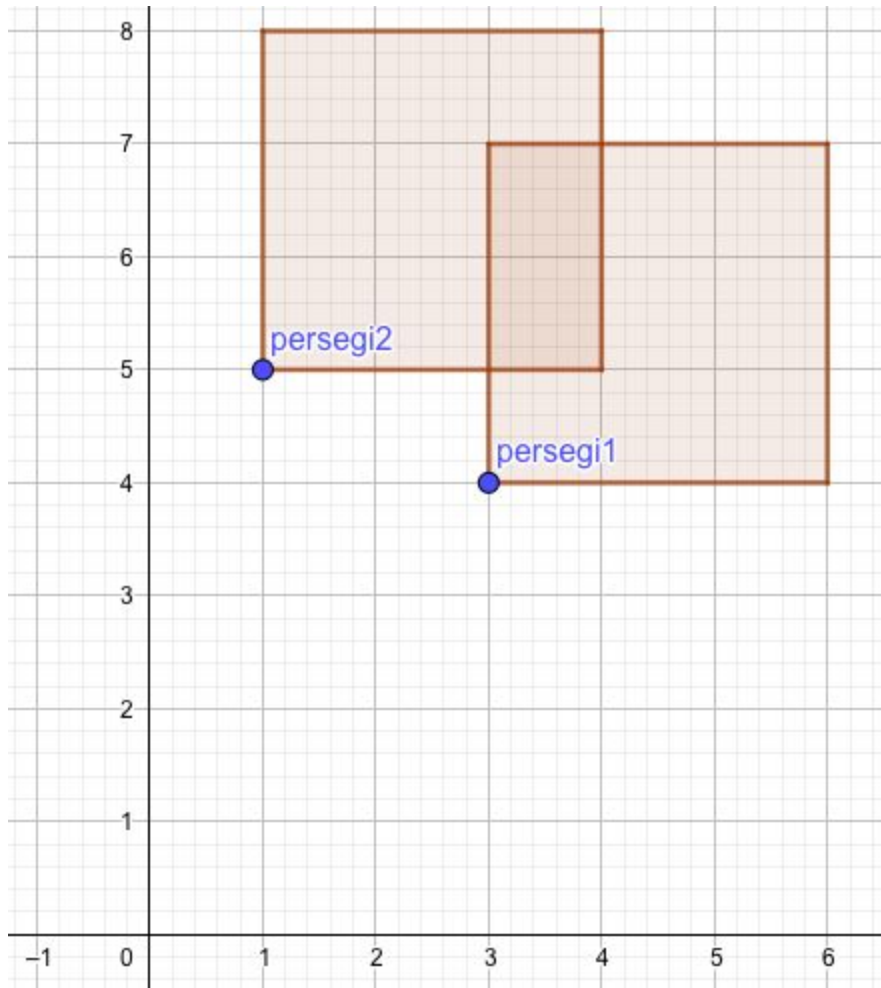
```
Masukkan panjang sisi persegi: 3
Masukkan komponen X dari titik kiri bawah persegi pertama: 3
Masukkan komponen Y dari titik kiri bawah persegi pertama: 4
Masukkan komponen X dari titik kiri bawah persegi kedua: 1
Masukkan komponen Y dari titik kiri bawah persegi kedua: 5
```

Tulisan yang ditandai dengan warna biru adalah masukan dari pengguna.

Apabila diberi masukan seperti di atas, program akan mengeluarkan:

```
Persegi beririsan
```

Kedua persegi divisualisasikan oleh gambar berikut:



Contoh lain:

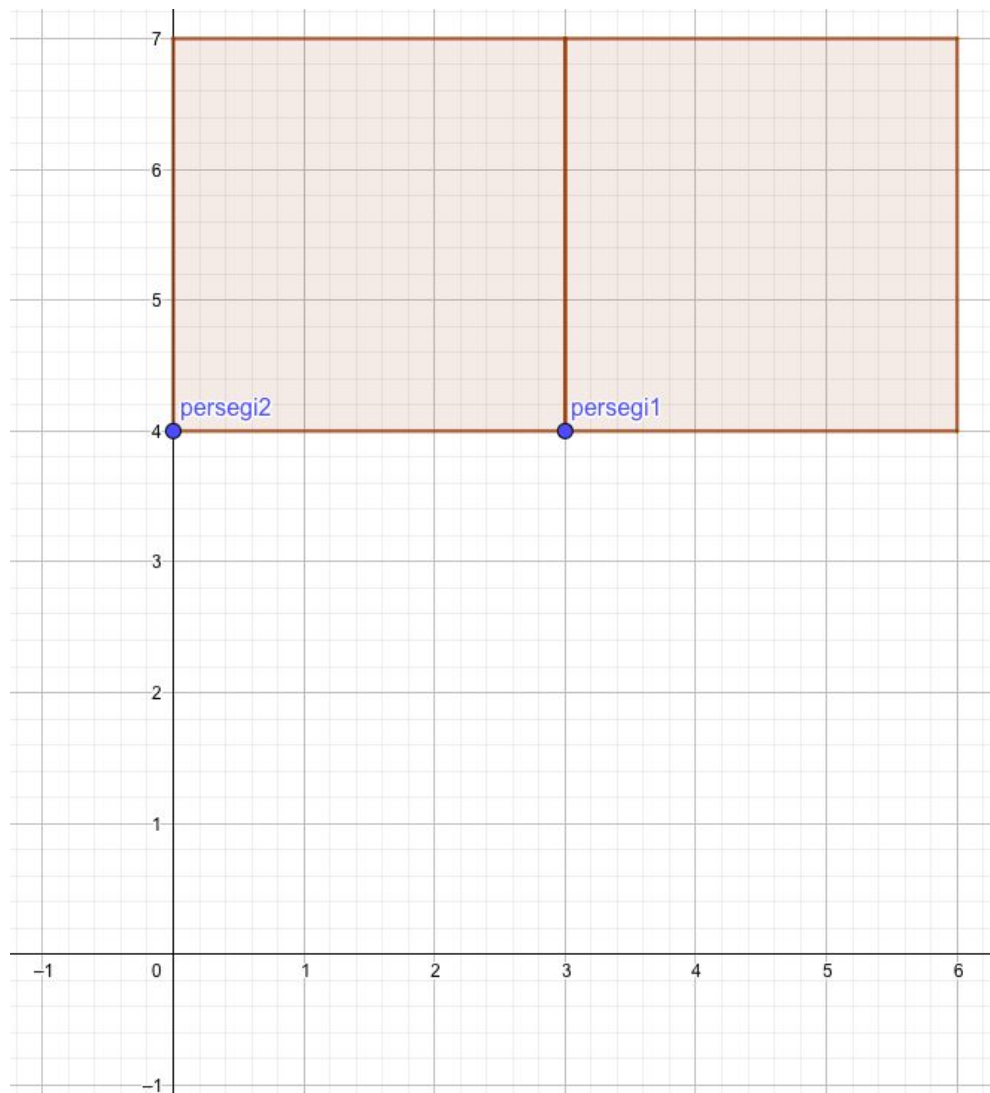
```
Masukkan panjang sisi persegi: 3
Masukkan komponen X dari titik kiri bawah persegi pertama: 3
Masukkan komponen Y dari titik kiri bawah persegi pertama: 4
Masukkan komponen X dari titik kiri bawah persegi kedua: 0
Masukkan komponen Y dari titik kiri bawah persegi kedua: 4
```

Tulisan yang ditandai dengan warna biru adalah masukan dari pengguna.

Apabila diberi masukan seperti di atas, program akan mengeluarkan:

```
Persegi beririsan
```

Kedua persegi divisualisasikan oleh gambar berikut:



Terlihat bahwa walaupun kedua persegi hanya 'menempel', program tetap mengeluarkan 'Persegi beririsan'.

Contoh lain:

```
Masukkan panjang sisi persegi: 3
Masukkan komponen X dari titik kiri bawah persegi pertama: 3
Masukkan komponen Y dari titik kiri bawah persegi pertama: 4
Masukkan komponen X dari titik kiri bawah persegi kedua: -1
Masukkan komponen Y dari titik kiri bawah persegi kedua: 4
```

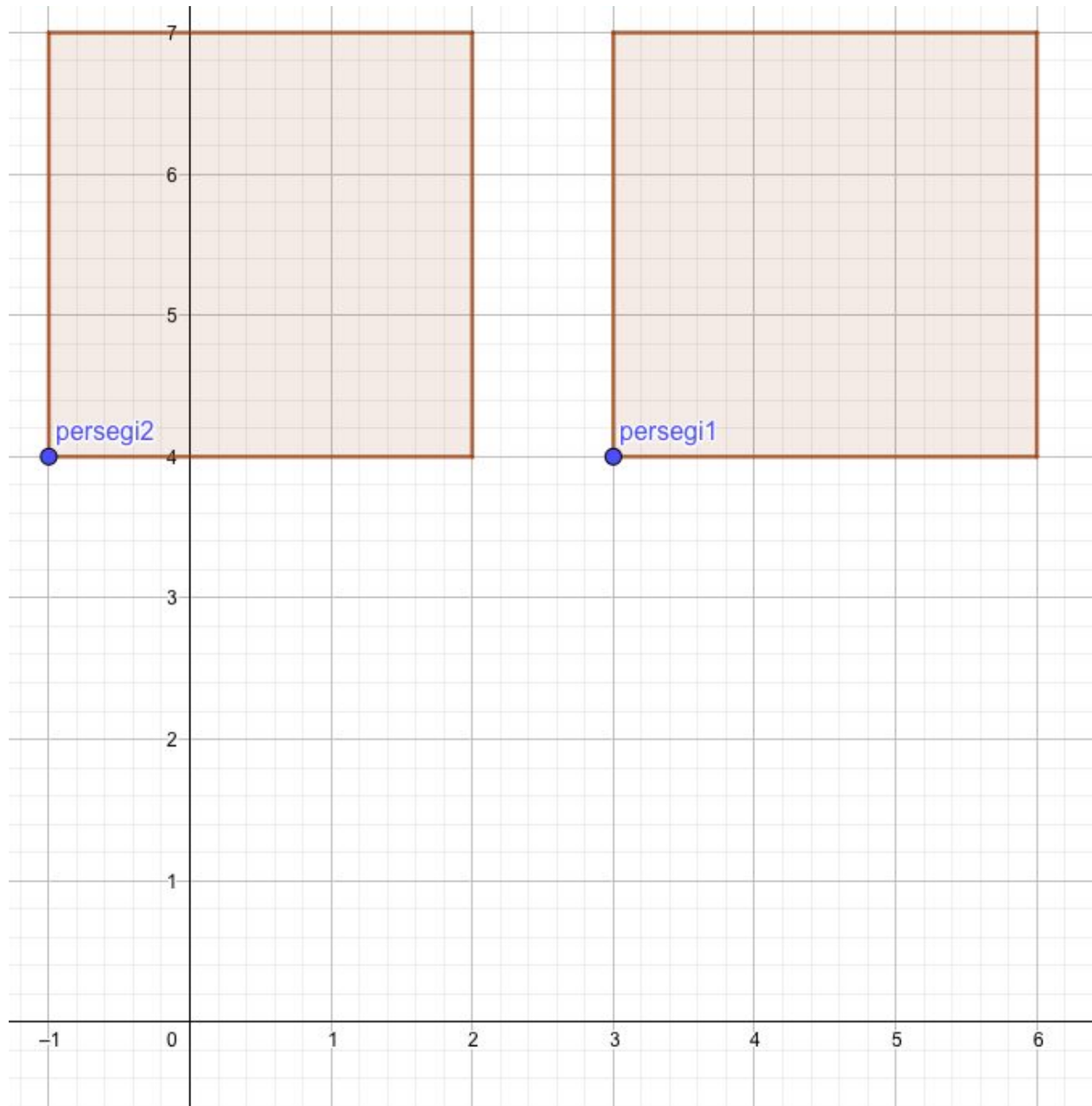
Tulisan yang ditandai dengan warna biru adalah masukan dari pengguna.



Apabila diberi masukan seperti di atas, program akan mengeluarkan:

Persegi terpisah

Kedua persegi divisualisasikan oleh gambar berikut:



Terlihat bahwa kedua persegi terpisah.

Contoh lain:

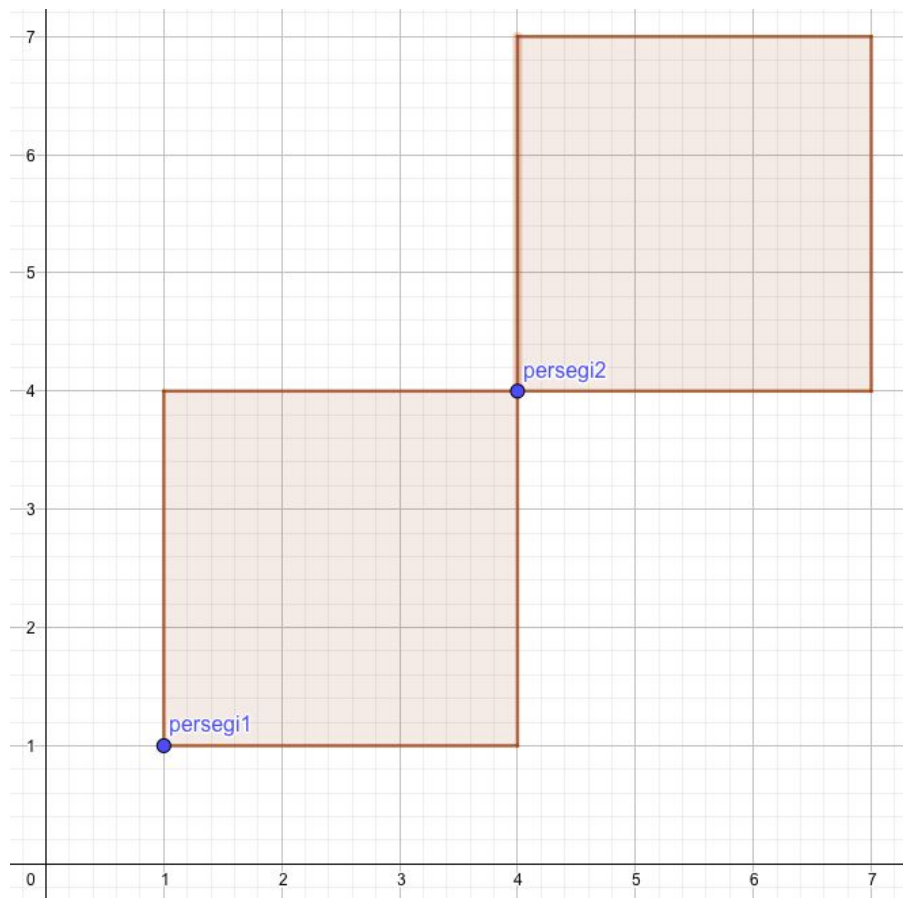
```
Masukkan panjang sisi persegi: 3
Masukkan komponen X dari titik kiri bawah persegi pertama: 1
Masukkan komponen Y dari titik kiri bawah persegi pertama: 1
Masukkan komponen X dari titik kiri bawah persegi kedua: 4
Masukkan komponen Y dari titik kiri bawah persegi kedua: 4
```

Tulisan yang ditandai dengan warna biru adalah masukan dari pengguna.

Apabila diberi masukan seperti di atas, program akan mengeluarkan:

```
Persegi beririsan
```

Kedua persegi divisualisasikan oleh gambar berikut:



Terlihat bahwa walaupun kedua persegi hanya 'menempel' pada ujung, program tetap mengeluarkan 'Persegi beririsan'.

**[Bonus] Challenge #1: ChaNET v0.5** (challenge1.py, skor: +0)

[https://docs.google.com/document/d/1661Mni\\_RelXg5yDvMfYQOlajmy0\\_Rp6fB9-iRWEpPa4/edit#heading=h.jhfsmgc014ho](https://docs.google.com/document/d/1661Mni_RelXg5yDvMfYQOlajmy0_Rp6fB9-iRWEpPa4/edit#heading=h.jhfsmgc014ho)

**[Bonus] Challenge #2: ChaNET Visualization** (challenge2.py, skor: +0)

[https://docs.google.com/document/d/1661Mni\\_RelXg5yDvMfYQOlajmy0\\_Rp6fB9-iRWEpPa4/edit#heading=h.r2qou3svnuue](https://docs.google.com/document/d/1661Mni_RelXg5yDvMfYQOlajmy0_Rp6fB9-iRWEpPa4/edit#heading=h.r2qou3svnuue)