

# Halo, *Turtle*!

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

Selamat datang di Tutorial 1 DDP1! Pada sesi tutorial ini, kita akan belajar mengenai *turtle graphics*. Selain itu, kita juga akan menggunakan konsep tutorial sebelumnya serta *control flow* dengan *if*. Spyder mungkin dapat mengalami *crash* saat menjalankan program *turtle*, untuk itu program dapat dijalankan dengan CLI. Pengenalan CLI dan *if* dapat dibaca pada dokumen terpisah di SCellE.

**Mohon kumpulkan semua file jawaban Anda dalam bentuk zip dengan format Lab1\_Nama\_NPM\_KodeAsdos.zip. Contoh: Lab1\_WindiChandra\_1606862721\_YE.zip**

## Latihan #4: Menggambar Persegi Panjang (latihan4.py, skor: +35)

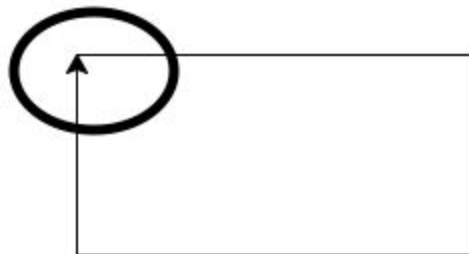
Konsep penting: *Turtle*

Dengan menggunakan *turtle graphics*, kita akan menggambar sebuah persegi panjang seperti berikut:



### Pengenalan *Turtle*

*Turtle graphics* merupakan *module/library* bawaan Python yang memuat fitur untuk memungkinkan kita menggambar layaknya pada papan tulis. Modul ini pun cukup menarik untuk para pemula dalam mempelajari pemrograman. Konsep menggambar menggunakan *turtle graphics* adalah kita memiliki suatu 'pulpen' (*pen*) yang mempunyai posisi dan arah menulis (ditunjukkan oleh bentuk panah dalam lingkaran).



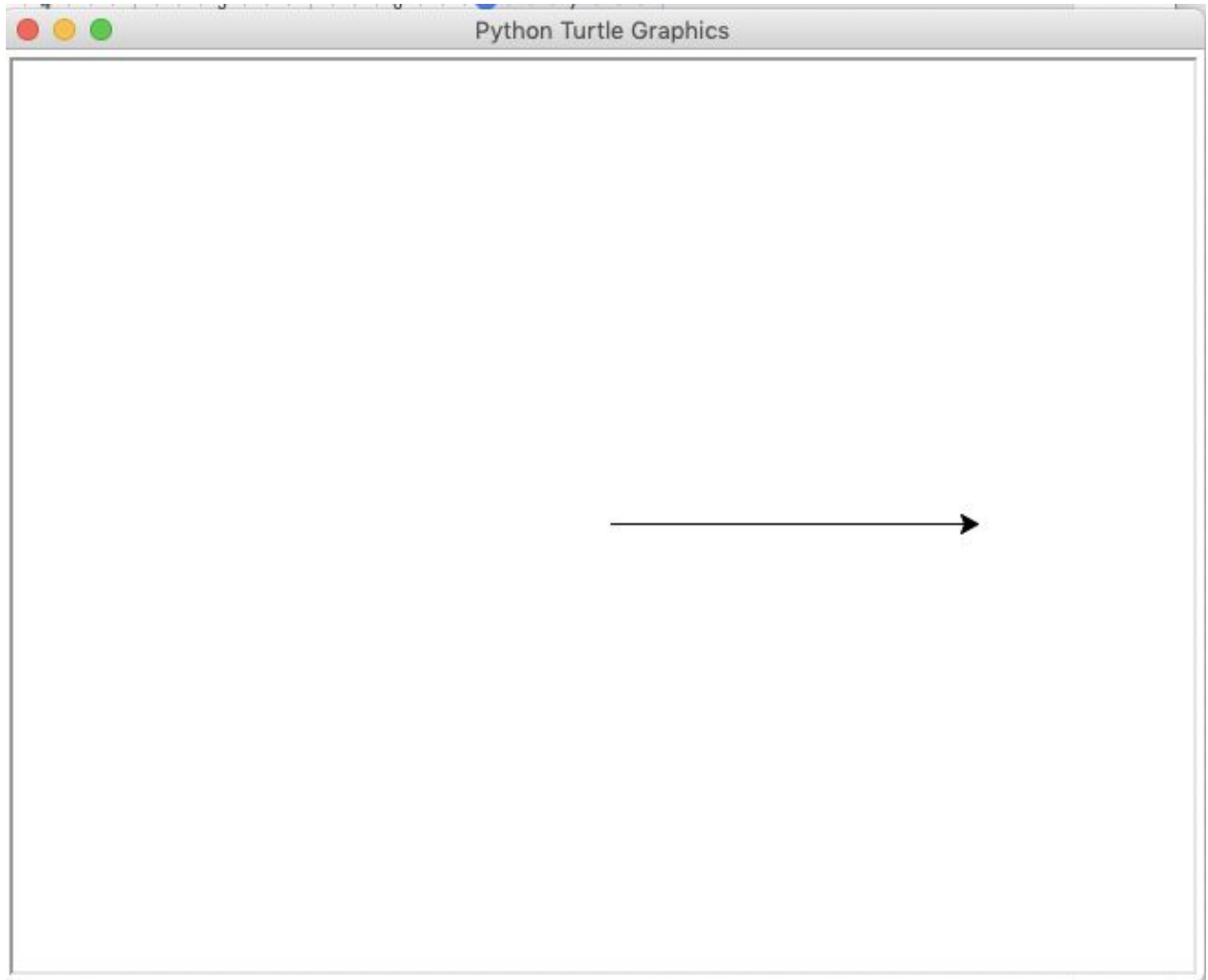
Untuk membentuk sebuah garis/lekukan, kita akan memerintah *pen* tersebut untuk 'bergerak' seperti maju, berputar ke kanan/kiri, menggambar lingkaran dengan jari-jari tertentu, dan masih banyak lagi. Anda dapat mengetahui apa saja yang bisa dilakukan oleh *turtle* dengan membaca [dokumentasinya](#).

Langkah pengerjaan:

1. Tuliskan kode berikut pada *file* latihan3.py:

```
1 import turtle
2
3 turtle.forward(200)
4
5 turtle.mainloop()
```

2. Jalankan program tersebut, seharusnya akan muncul hasil seperti ini:



Penjelasan kode:

<code>import turtle</code>	Meng- <i>import</i> modul <i>turtle</i> agar kita bisa menggunakan fitur <i>turtle graphics</i> .
<code>turtle.forward(200)</code> <sup>1</sup>	Memerintahkan <i>pen</i> untuk maju sebanyak 200 unit sehingga bisa menggoreskan suatu garis. Window <i>turtle</i> akan terbuka ketika perintah “ <i>turtle.</i> ” dipanggil pertama kali
<code>turtle.mainloop()</code>	Mencegah error terkait loop GUI. <a href="https://docs.python.org/3.1/library/turtle.html#turtle.mainloop">https://docs.python.org/3.1/library/turtle.html#turtle.mainloop</a>  Perintah ini harus selalu ada di setiap akhir program!

Pada awalnya, posisi *pen* berada pada koordinat (0,0) (tengah) dan menghadap ke kanan. Apabila kita memanggil perintah `turtle.forward(200)` dengan kondisi *pen* seperti di atas, maka *pen* akan bergerak ke kanan dan akan menggoreskan garis sepanjang 200 unit dan posisi *pen* sekarang berada pada koordinat (200, 0).

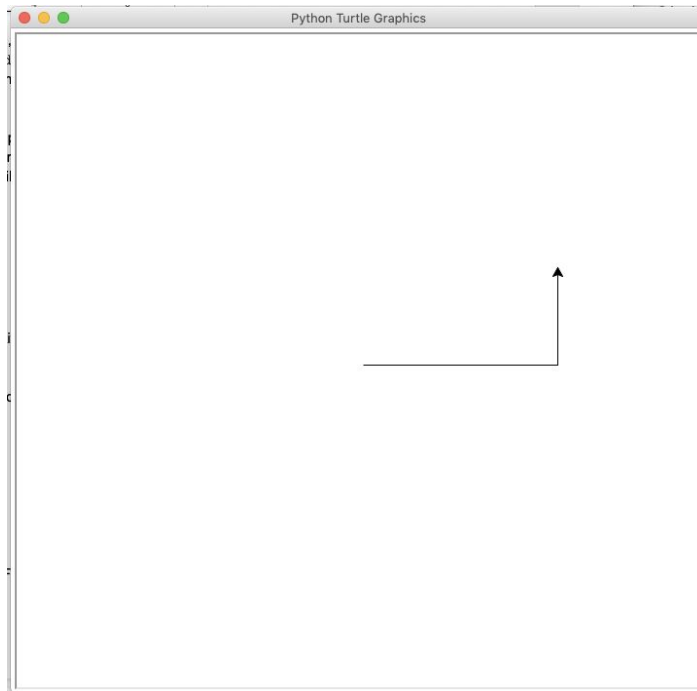
---

<sup>1</sup> <https://docs.python.org/3.1/library/turtle.html#turtle.forward>

3. Kita sudah berhasil membuat satu sisi dari persegi panjang, selanjutnya kita akan memerintahkan *pen* tersebut untuk berputar ke kiri sebanyak 90 derajat dengan perintah `"turtle.left(90)"`<sup>2</sup>. Kemudian tarik garis lagi sepanjang 100 unit:

```
1 import turtle
2
3 turtle.forward(200)
4
5 turtle.left(90)
6 turtle.forward(100)
7
8 turtle.mainloop()
```

4. Jalankan program, dan perhatikan bahwa kita telah berhasil membuat 2 sisi dari persegi panjang:



Perintah `turtle.left(90)` juga dapat diganti dengan `turtle.right(270)` (mengapa?)

---

<sup>2</sup> <https://docs.python.org/3.1/library/turtle.html#turtle.left>

5. Kemudian, dengan cara yang sama kita akan menggambarkan 2 sisi yang tersisa dengan berputar ke kiri 90 derajat, menarik garis, berputar lagi ke kiri 90 derajat, dan tarik garis lagi.

```
1 import turtle
2
3 turtle.forward(200)
4
5 turtle.left(90)
6 turtle.forward(100)
7
8 turtle.left(90)
9 turtle.forward(200)
10
11 turtle.left(90)
12 turtle.forward(100)
13
14 turtle.mainloop()
```

6. Selamat, Anda telah berhasil menggambar persegi panjang!

## Latihan #5: TurtleTRON-3999 (latihan5.py, skor: +35)

Konsep penting: variabel, *program control/branching*, *turtle*

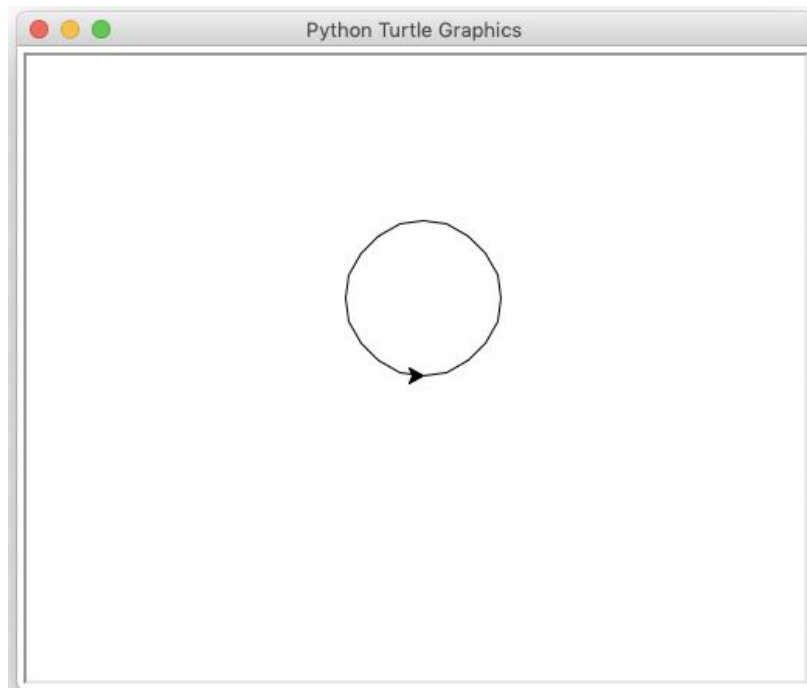
Buatlah program yang dapat menggambar lingkaran atau segitiga sama sisi. Program akan meminta input kepada pengguna. Apabila pengguna memasukkan kata 'lingkaran', maka gambarlah lingkaran dengan jari-jari 50 *unit*, apabila pengguna memasukkan kata 'segitiga', maka gambarlah segitiga sama sisi dengan panjang sisi 100 unit.

Contoh program:

```
Selamat datang di TurtleTRON-3999!  
Masukkan bentuk yang ingin digambar ('lingkaran' atau 'segitiga'): lingkaran
```

Tulisan yang ditandai dengan warna biru adalah masukan dari pengguna.

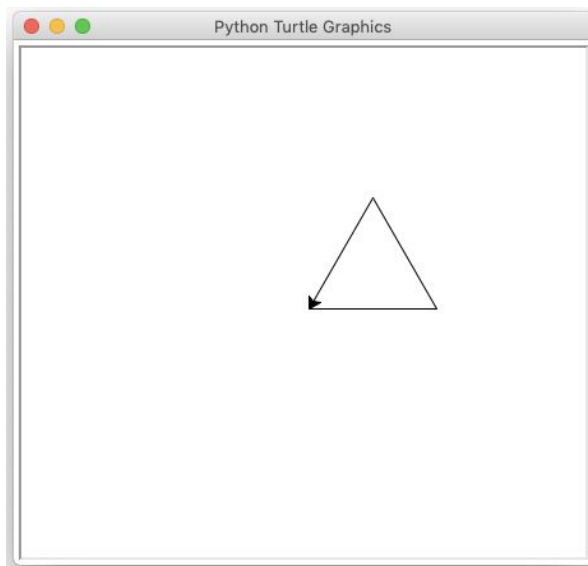
Apabila diberikan masukan seperti di atas, maka program akan mengeluarkan seperti di bawah:



Contoh lain:

```
Selamat datang di TurtleTRON-3999!  
Masukkan bentuk yang ingin digambar ('lingkaran' atau 'segitiga'): segitiga
```

Apabila diberikan masukan seperti di atas, maka program akan mengeluarkan gambar seperti di bawah:



Petunjuk:

- Perintah `turtle.circle(150)`<sup>3</sup> akan memerintahkan *turtle* untuk menggambar lingkaran dengan panjang jari-jari 150 unit.
- Format input tidak harus sama dengan soal, Anda boleh menentukan format input sesuai kreativitas Anda :).

---

<sup>3</sup> <https://docs.python.org/3.1/library/turtle.html#turtle.circle>



## Latihan #6: TurtleTRON-3999 v2 (latihan6.py, skor: +30)

Konsep penting: variabel, program control/branching, turtle

Program Anda pada latihan kelima diterima sangat baik oleh komunitas internet! Bahkan, majalah teknologi ternama memberikan nilai *review* sebesar 90/100 untuk program Anda! Oleh karena itu, Anda menjadi semangat dan ingin mengembangkan lebih lanjut program tersebut. Anda akan menambahkan fitur-fitur baru, yaitu:

- Nama program Anda terpampang di atas gambar, untuk *branding*. **(3 poin)**
- **Terdapat nama Anda** di sebelah kiri bawah gambar, untuk *copyright*. **(3 poin)**
- Anda dapat memberi warna pada goresan gambar Anda (*outline*). **(10 poin)**
- Anda dapat memberi warna di dalam gambar Anda (*fill*). **(9 poin)**
- Anda dapat menentukan panjang keliling segitiga/lingkaran Anda. **(5 poin)**
  - (Apabila Anda membutuhkan nilai konstanta pi, Anda boleh menggunakan nilai `math.pi` dari module *math*, atau menggunakan konstanta 3.1415)

Warna yang didukung oleh program Anda adalah merah, hijau, biru, kuning, oranye, dan hitam.

Tentu saja, program Anda akan menerima input tambahan yaitu panjang keliling, warna yang ingin digunakan, dan pilihan apakah ingin melakukan *fill* ('menyiram' warna ke dalam gambar) atau tidak. Urutan input tidak dipermasalahkan dan Anda boleh menentukan urutan input sesuai selera Anda. Program Anda juga mengasumsikan input pengguna selalu benar sehingga Anda tidak perlu menangani kasus-kasus khusus dimana input pengguna tidak valid seperti menggambar segilima dengan warna pink. Tidak apa-apa bila gambar lingkaran/segitiga Anda 'menimpa' teks nama program/nama Anda karena gambar tersebut terlalu besar.

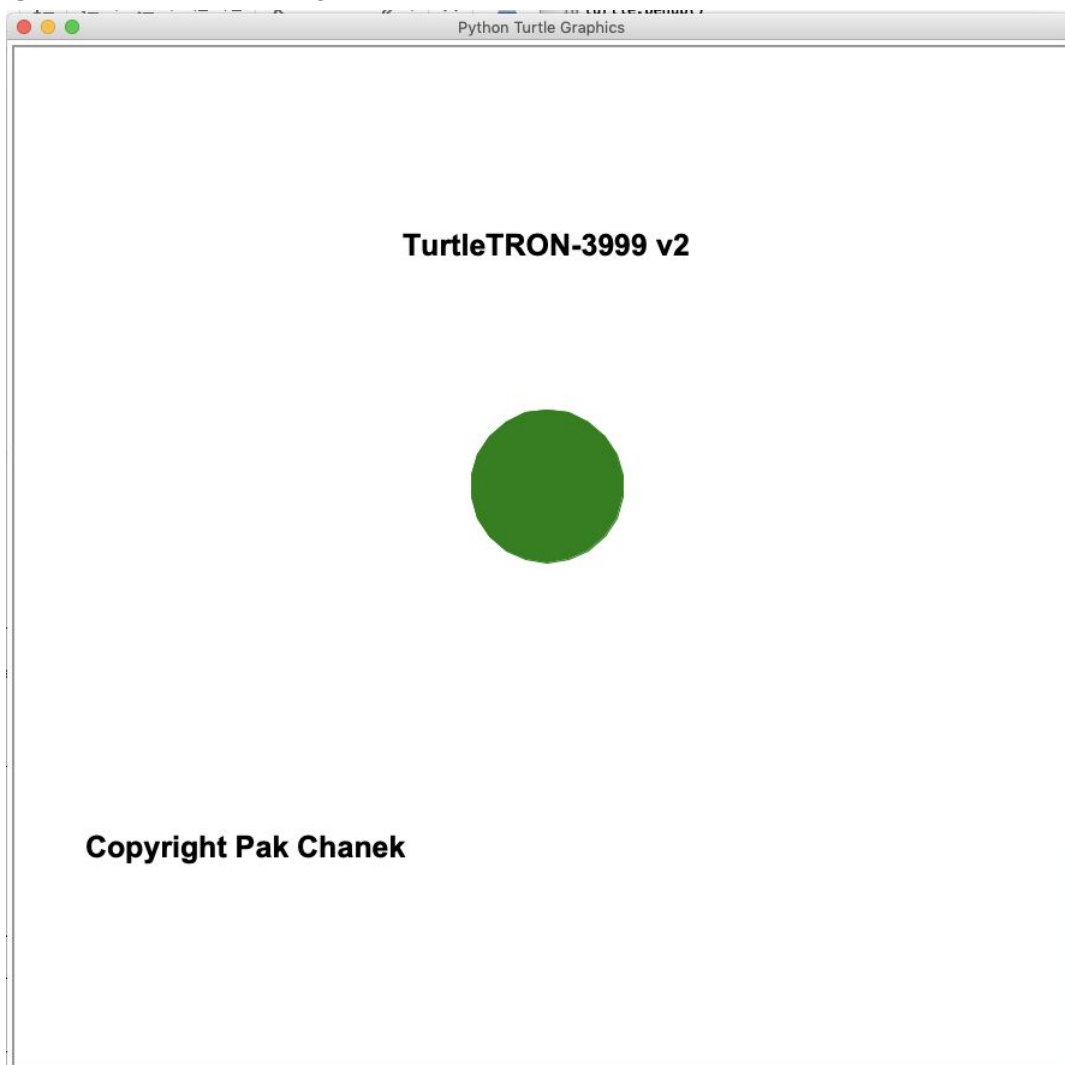
Anda diperbolehkan untuk meng-*copy paste* jawaban Anda dari latihan 5.

Contoh interaksi program ada pada halaman berikutnya.

Contoh program:

```
Selamat datang di TurtleTRON-3999 v2!  
Masukkan bentuk yang ingin digambar (lingkaran/segitiga): lingkaran  
Masukkan panjang keliling gambar: 400  
Masukkan warna gambar: hijau  
Apakah Anda ingin mengisi warna ke dalam gambar?: ya
```

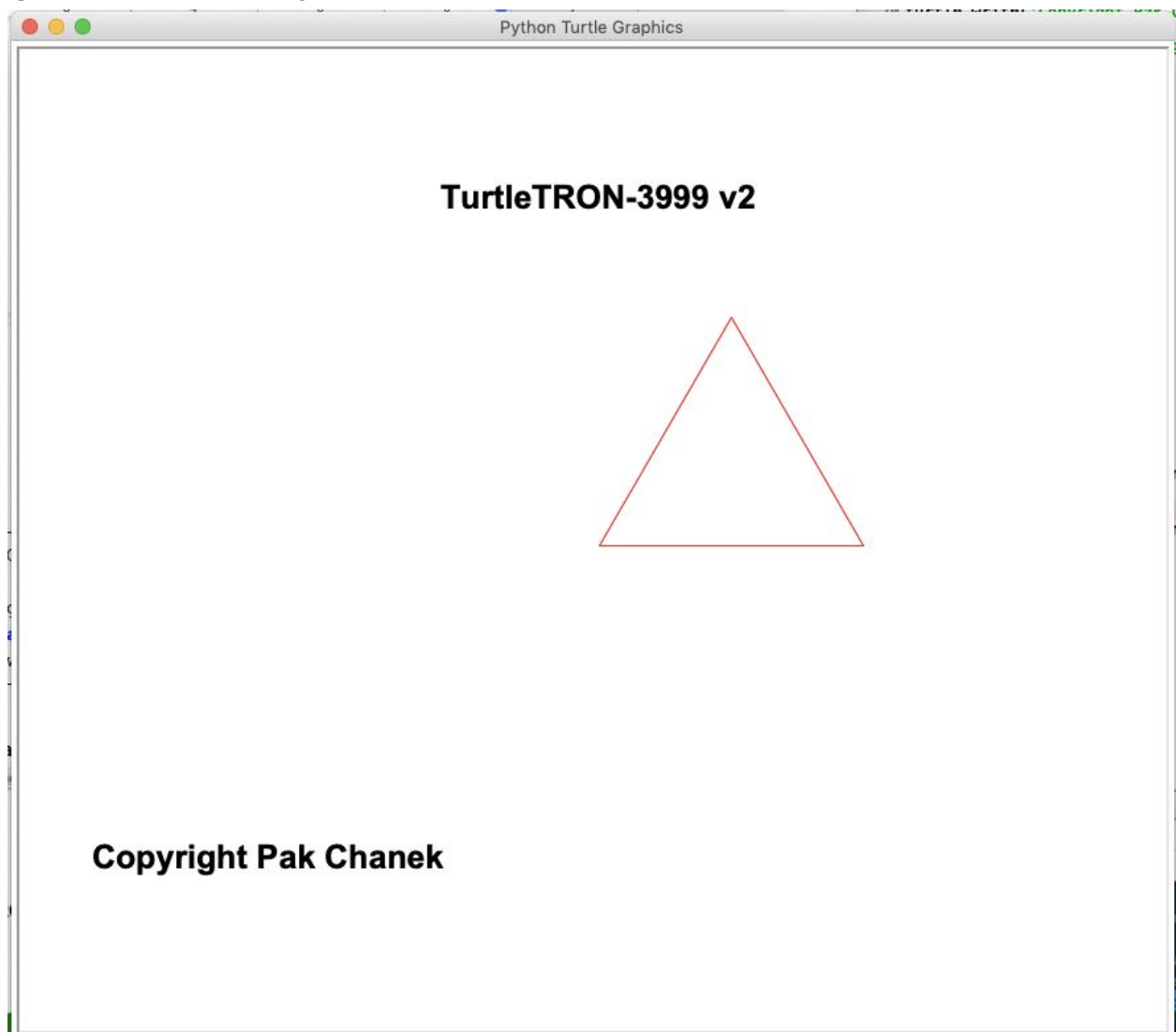
Apabila diberi masukan seperti di atas, program akan memberikan gambar seperti berikut (**ganti Pak Chanek** dengan nama Anda!):



Contoh program:

```
Selamat datang di TurtleTRON-3999 v2!  
Masukkan bentuk yang ingin digambar (lingkaran/segitiga): segitiga  
Masukkan panjang keliling gambar: 600  
Masukkan warna gambar: merah  
Apakah Anda ingin mengisi warna ke dalam gambar?: tidak
```

Apabila diberi masukan seperti di atas, program akan memberikan gambar seperti berikut (**ganti Pak Chanek** dengan nama Anda!):



Petunjuk:

1. Warna *pen* dapat diganti dengan `turtle.color('kode_warna')`. Anda harus mengganti `kode_warna` dengan kode warna yang valid. Untuk soal ini, Anda dapat menggunakan kode warna yang valid di bawah ini:

Warna	Kode Warna
Merah	red
Hijau	green
Kuning	yellow
Oranye	orange
Hitam	black
Biru	blue

Apabila Anda mengganti warna *pen*, warna setiap goresan akan berubah sesuai warna dari *pen*.

2. Anda dapat menggunakan perintah seperti berikut untuk mengisi lingkaran dengan suatu warna:

```
turtle.begin_fill()
turtle.circle(100)
turtle.end_fill()
```

Penjelasan:

<code>turtle.begin_fill()</code> <sup>4</sup>	Menandakan bahwa garis yang akan digambar berikutnya harus diwarnai secara <i>flood fill</i> , (ingat fill pada program paint!)
<code>turtle.end_fill()</code> <sup>5</sup>	Memulai mewarnai secara <i>flood fill</i>

---

<sup>4</sup> [https://docs.python.org/3.1/library/turtle.html#turtle.begin\\_fill](https://docs.python.org/3.1/library/turtle.html#turtle.begin_fill)

<sup>5</sup> [https://docs.python.org/3.1/library/turtle.html#turtle.end\\_fill](https://docs.python.org/3.1/library/turtle.html#turtle.end_fill)

3. Apabila Anda melakukan *fill* ke dalam gambar Anda, Anda mungkin saja mengalami hal di mana gambar Anda terlihat tidak sempurna seperti berikut:



Hal ini terjadi karena bentuk 'tajak' di sisi bawah lingkaran sebenarnya adalah *pen* anda (bentuk panah berwarna hijau). Gunakan `turtle.hideturtle()`<sup>6</sup> setelah Anda selesai menggambar untuk menyembunyikan *pen* Anda.

4. Untuk menulis teks, Anda dapat menggunakan perintah `turtle.write()`<sup>7</sup>. Perintah tersebut akan menulis teks di tempat *pen* Anda berada.
5. Anda perlu menggeser *pen* Anda sebelum menulis teks, perintah yang bisa digunakan adalah `turtle.goto()`<sup>8</sup>. Namun, menggeser *pen* akan menimbulkan goresan. Untuk menghindari hal tersebut, anda dapat menggunakan `turtle.penup()`<sup>9</sup> dan `turtle.pendown()`<sup>10</sup>. Sistem koordinat pada turtle mengikuti sistem koordinat kartesius (y positif ke atas, x positif ke kanan).
6. Jangan takut untuk mencoba dan tetap semangat!

---

<sup>6</sup> <https://docs.python.org/3.1/library/turtle.html#turtle.hideturtle>

<sup>7</sup> <https://docs.python.org/3.1/library/turtle.html#turtle.write>

<sup>8</sup> <https://docs.python.org/3.1/library/turtle.html#turtle.goto>

<sup>9</sup> <https://docs.python.org/3.1/library/turtle.html#turtle.penup>

<sup>10</sup> <https://docs.python.org/3.1/library/turtle.html#turtle.pendown>