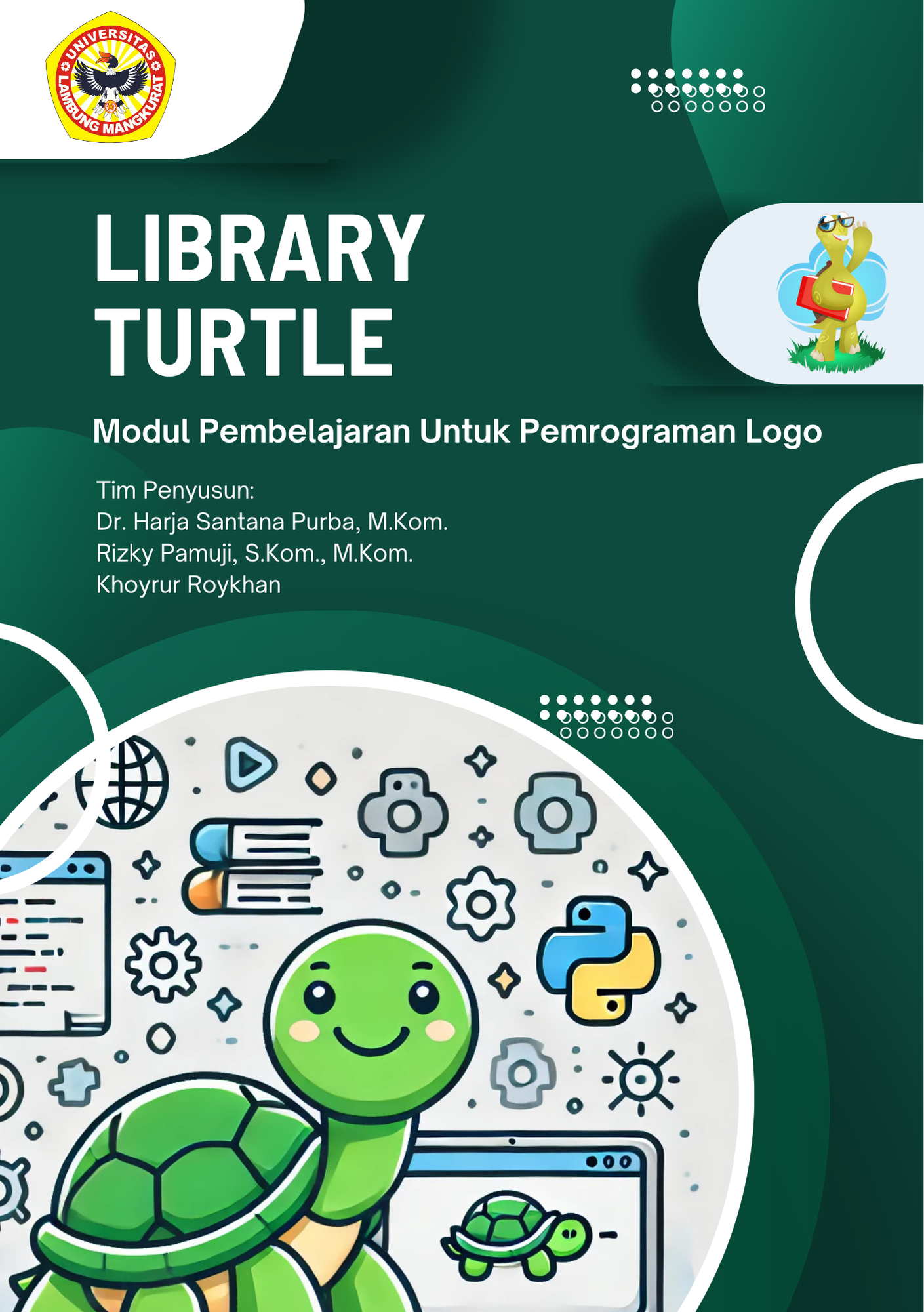
**Cover**

**KATA PENGANTAR**

Puji syukur kehadirat Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya sehingga penyusunan materi ini dapat diselesaikan dengan baik. Materi ini disusun untuk membantu siswa dalam memahami dasar-dasar pemrograman melalui visualisasi yang interaktif dan menyenangkan.

Library Turtle di Python merupakan salah satu modul standar yang dirancang untuk memudahkan pengenalan konsep pemrograman melalui grafik. Modul ini terinspirasi dari bahasa Logo, yang diperkenalkan oleh Wally Feurzeig, Seymour Papert, dan Cynthia Solomon pada tahun 1967. Dengan Turtle, pengguna dapat membuat berbagai bentuk geometris dan pola dengan mengendalikan pergerakan "pen" atau "kura-kura" di atas layar, sehingga sangat cocok untuk pemula yang ingin mempelajari konsep dasar pemrograman, matematika, dan geometri.

Dalam penyusunan materi ini, konsep Turtle dimodifikasi dengan menghadirkan karakter Bidawang sebagai pengganti "kura-kura." Bidawang, yang dikenal dalam kearifan lokal sebagai hewan khas lahan basah, diharapkan dapat memberikan pengalaman belajar yang lebih kontekstual dan relevan bagi siswa, terutama dalam mengenalkan potensi dan lingkungan sekitar.

Akhir kata, penulis berharap materi ini dapat memberikan manfaat dan menjadi sarana pembelajaran yang efektif. Segala kritik dan saran yang membangun sangat diharapkan untuk perbaikan di masa mendatang.

*Penyusun*

**DAFTAR ISI**

[1. PENGENALAN 7](#_Toc191636252)

[2. PERGERAKAN 12](#_Toc191636253)

[A. Left & Right 12](#_Toc191636254)

[B. Forward & Backward 16](#_Toc191636255)

[C. Setposition 21](#_Toc191636256)

[D. Setx & Sety 25](#_Toc191636257)

[E. Setheading 30](#_Toc191636258)

[F. Home 34](#_Toc191636259)

[G. Circle 38](#_Toc191636260)

[H. Dot 43](#_Toc191636261)

[3. MENGETAHUI STATUS 48](#_Toc191636262)

[A. Position 48](#_Toc191636263)

[B. Xcor & Ycor 52](#_Toc191636264)

[C. Heading 56](#_Toc191636265)

[D. Distance 59](#_Toc191636266)

[4. KONTROL PENA 62](#_Toc191636267)

[A. Pendown & Penup 62](#_Toc191636268)

[B. Pensize 65](#_Toc191636269)

[C. Isdown 68](#_Toc191636270)

[5. KONTROL WARNA 71](#_Toc191636271)

[A. Pencolor 71](#_Toc191636272)

[B. Fillcolor, Begin\_fill, & End\_fill 74](#_Toc191636273)

[6. KONTROL GAMBAR LANJUTAN 77](#_Toc191636274)

[A. Reset 77](#_Toc191636275)

[B. Clear 80](#_Toc191636276)

[C. Write 83](#_Toc191636277)

**DAFTAR GAMBAR**

[Gambar 1 (Video ilustrasi bidawang geometri) 7](#_Toc192105826)

[Gambar 2 (Gambaran koordinat canvas) 8](#_Toc192105827)

[Gambar 3 (Contoh pergerakan Bidawang di canvas) 8](#_Toc192105828)

[Gambar 4 (Tampilan lingkungan kerja) 9](#_Toc192105829)

[Gambar 5 (Hasil contoh menggerakan bidawang dalam canvas) 11](#_Toc192105830)

[Gambar 6 (Hasil contoh penggunaan left) 12](#_Toc192105831)

[Gambar 7 (Hasil contoh penggunaan right) 13](#_Toc192105832)

[Gambar 8 (Hasil latihan penggunaan perintah left dan right ) 14](#_Toc192105833)

[Gambar 9 (Tantangan perintah left dan right) 15](#_Toc192105834)

[Gambar 10 (Hasil contoh pengunaan forward) 16](#_Toc192105835)

[Gambar 11 (Hasil contoh penggunaan backward) 17](#_Toc192105836)

[Gambar 12 (Hasil latihan penggunaan forward dan backward) 18](#_Toc192105837)

[Gambar 13 (Tantangan perintah forward dan backward) 19](#_Toc192105838)

[Gambar 14 (Hasil contoh penggunaan setposition) 21](#_Toc192105839)

[Gambar 15 (Hasil latihan pengunaan setposition) 22](#_Toc192105840)

[Gambar 16 (Tantangan penggunaan perintah setposition) 23](#_Toc192105841)

[Gambar 17 (Hasil contoh penggunaan setx) 25](#_Toc192105842)

[Gambar 18 (Hasil contoh penggunaan sety) 26](#_Toc192105843)

[Gambar 19 (Hasil latihan penggunaan setx dan sety) 27](#_Toc192105844)

[Gambar 20 (Tantangan perintah setx dan sety) 28](#_Toc192105845)

[Gambar 21 (Hasil contoh penggunaan setheading) 30](#_Toc192105846)

[Gambar 22 (Hasil latihan penggunaan setheading) 32](#_Toc192105847)

[Gambar 23 (Tantangan perintah setheading) 33](#_Toc192105848)

[Gambar 24 (Hasil contoh penggunaan home) 34](#_Toc192105849)

[Gambar 25 (Hasil latihan penggunaan home) 35](#_Toc192105850)

[Gambar 26 (Tantangan perintah home) 36](#_Toc192105851)

[Gambar 27 (Hasil contoh 1 penggunaan circle) 38](#_Toc192105852)

[Gambar 28 (Hasil contoh 2 penggunaan circle) 39](#_Toc192105853)

[Gambar 29 (Hasil latihan penggunaan circle) 40](#_Toc192105854)

[Gambar 30 (Tantangan perintah circle) 41](#_Toc192105855)

[Gambar 31 (Hasil contoh 1 penggunaan dot) 43](#_Toc192105856)

[Gambar 32 (Hasil contoh 2 penggunaan dot) 44](#_Toc192105857)

[Gambar 33 (Hasil latihan penggunaan dot) 45](#_Toc192105858)

[Gambar 34 (Tantangan perintah dot) 46](#_Toc192105859)

[Gambar 35 (Hasil contoh 1 penggunaan position) 48](#_Toc192105860)

[Gambar 36 (Hasil contoh 2 penggunaan position) 49](#_Toc192105861)

[Gambar 37 (Hasil latihan penggunaan position) 50](#_Toc192105862)

[Gambar 38 (Hasil contoh penggunaan xcor dan ycor) 53](#_Toc192105863)

[Gambar 39 (Hasil contoh penggunaan heading) 56](#_Toc192105864)

[Gambar 40 (Hasil contoh penggunaan distance) 59](#_Toc192105865)

[Gambar 41 (Hasil contoh penggunaan penup dan pendown) 62](#_Toc192105866)

[Gambar 42 (Hasil contoh penggunaan pensize 65](#_Toc192105867)

[Gambar 43 (Hasil contoh penggunan isdown) 68](#_Toc192105868)

[Gambar 44 (Hasil contoh penggunaan pencolor) 72](#_Toc192105869)

[Gambar 45 (Hasil contoh penggunan fillcolor, begin\_fill, dan end\_fill) 75](#_Toc192105870)

[Gambar 46 (Hasil contoh penggunaan reset) 77](#_Toc192105871)

[Gambar 47 (Hasil contoh penggunaan clear) 80](#_Toc192105872)

[Gambar 48 (Hasil contoh penggunaan write) 83](#_Toc192105873)

**DAFTAR TABEL**

# PENGENALAN

PENGENALAN

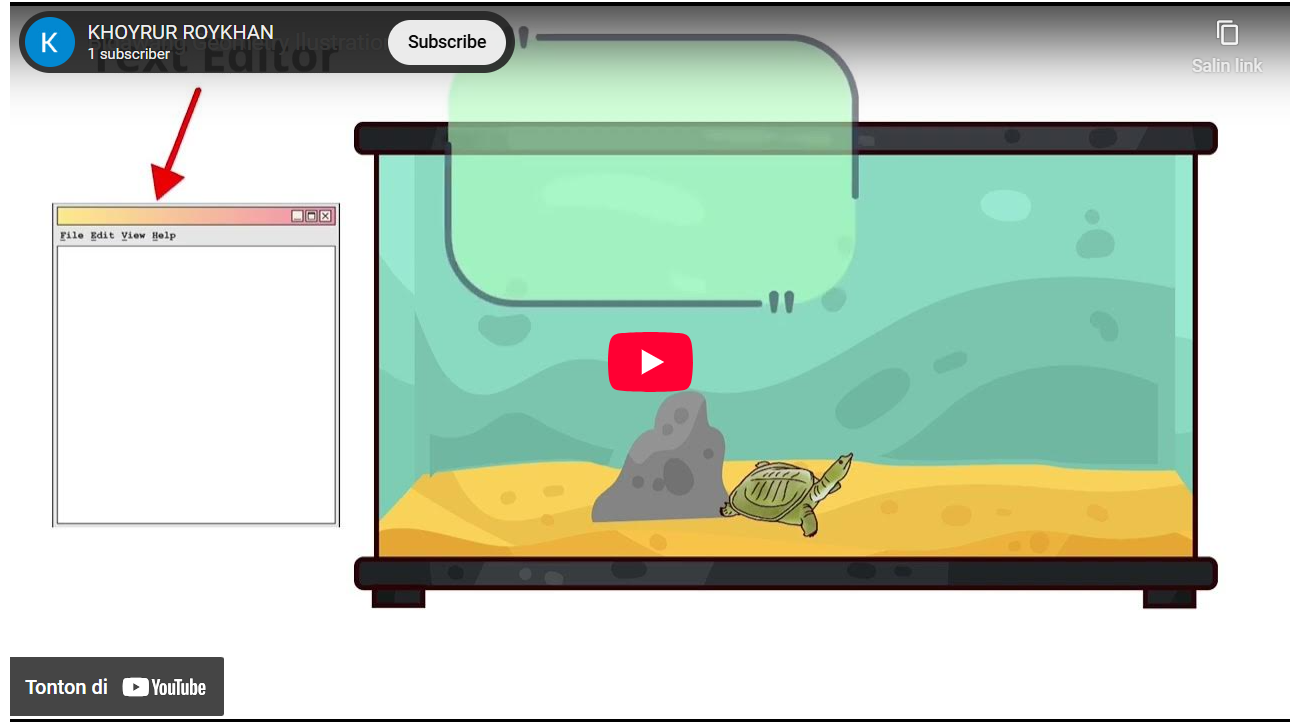
**1**

**TUJUAN PEMBELAJARAN**

* Memahami konsep canvas sebagai ruang pergerakan Bidawang.
* Mengenali tampilan lingkungan kerja dan perintah dasar untuk menggerakan Bidawang pada canvas.

**APERSEPSI**

Perhatikan video ilustrasi dibawah ini:



Gambar 1 (Video ilustrasi bidawang geometri)

Sumber: <https://youtu.be/iefPvNd_diM>

**Canvas**

Canvas adalah area tempat Bidawang bergerak menggambar pola geometri secara interaktif. Bidawang tersebut dapat dikontrol untuk bergerak maju (forward), mundur (backward), berbelok ke kiri (left), berbelok ke kanan (right), dan melakukan berbagai aksi lainnya menggunakan perintah-perintah tertentu. Ukuran default canvas adalah 400x400 piksel. Berikut adalah contoh gambaran canvas dan contoh Bidwang bergerak dalam canvas:

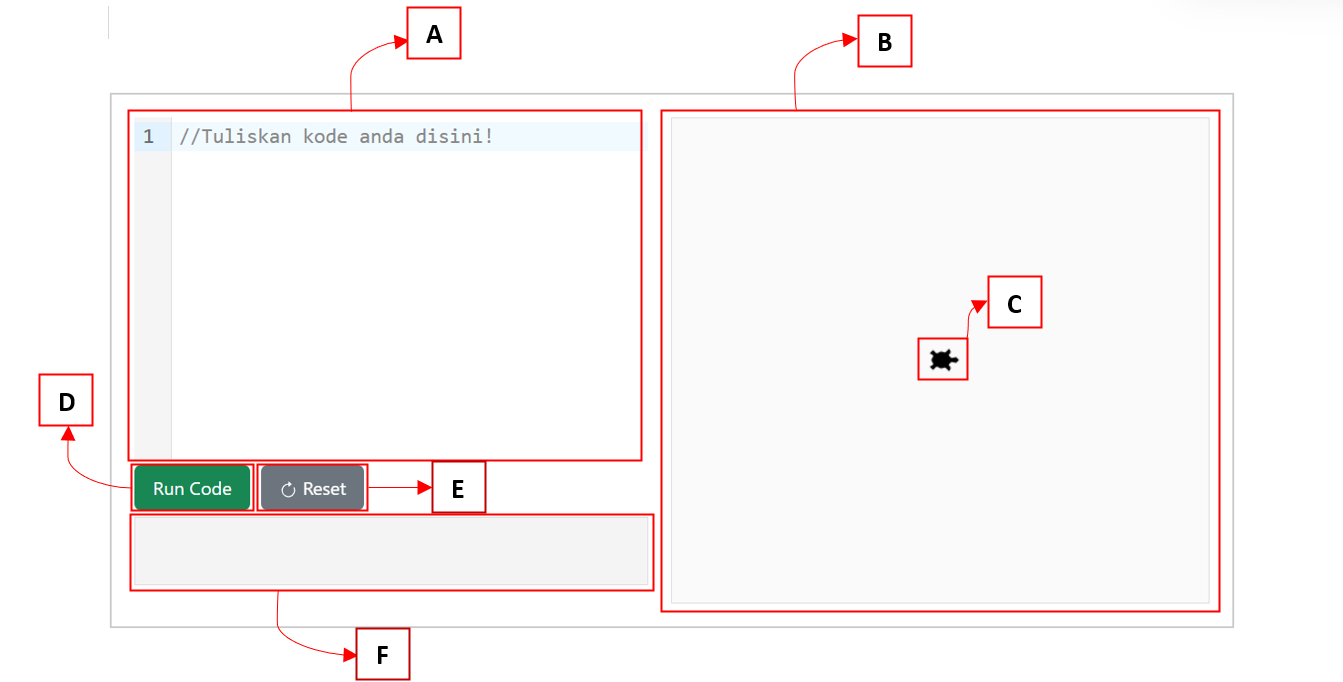
|  |  |
| --- | --- |
| Gambar 2 (Gambaran koordinat canvas) | Gambar 3 (Contoh pergerakan Bidawang di canvas) |

Penjelasan:

* Titik awal posisi Bidawang adalah (0, 0), yang berada di tengah canvas.
* Batas pergerakan ke atas (sumbu Y positif) adalah 200, yang merupakan batas atas canvas.
* Batas pergerakan ke bawah (sumbu Y negatif) adalah -200, yang merupakan batas bawah canvas.
* Batas pergerakan ke kanan (sumbu X positif) adalah 200.
* Batas pergerakan ke kiri (sumbu X negatif) adalah -200.

**Tampilan Lingkungan Kerja**

Untuk mempermudah mengontrol Bidawang, tersedia lingkungan kerja di dalam aplikasi yang sudah di kembangkan, yang terdiri dari beberapa komponen seperti gambar dibawah ini:



Gambar 4 (Tampilan lingkungan kerja)

Penjelasan :

* **(A) Text Editor** : Area tempat pengguna mengetik perintah kode untuk menggerakkan Bidawang.
* **(B) Canvas** : Area tampilan di mana pergerakan Bidawang divisualisasikan. Semua perintah yang dijalankan akan langsung terlihat pada canvas.
* **(C) Bidawang** : Objek yang digerakkan menggunakan perintah kode.
* **(D) Tombol "Run Code"**: Digunakan untuk menjalankan kode yang telah ditulis di text editor. Setelah ditekan, Bidawang akan menjalankan perintah dan menggambar sesuai instruksi.
* **(E) Tombol "Reset"**: Menghapus kode serta hasil gambar di canvas dan mengembalikan Bidawang ke posisi awal.
* **(F) Output Log**: Digunakan untuk menampilkan output dari program yang dijalankan atau pesan eror.

**Contoh menggerakan Bidawang dalam canvas**

Kita bisa menggerakkan Bidawang dengan berbagai perintah. Berikut adalah contoh perintah dasar untuk membuat Bidawang dapat bergerak dan berputar, Agar lebih mudah untuk memahaminya coba ikuti instruksi dibawah ini. Kalian bisa membuka aplikasi yang sudah di siapkan, kemudian ketikkan perintah tersebut, lalu tekan tombol "Run Code" untuk melihat pergerakan Bidawang, lakukan secara bertahap:

1. Maju

Gerakkan Bidawang maju sejauh 100 langkah dengan perintah dibawah ini:

forward(100)

1. Berbelok ke kanan

Kemudian lanjutkan lagi pada baris baru dengan perintah dibawah ini untuk memutar Bidawang ke kanan sebesar 90 derajat:

right(90)

1. Maju

Gerakkan lagi Bidawang maju sejauh 100 langkah:

forward(100)

1. Berbelok ke kiri

Putar Bidawang ke kiri sebesar 45 derajat:

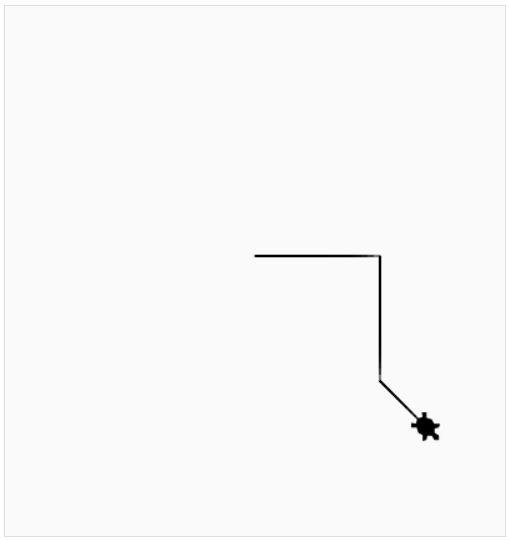
left(45)

1. Maju

Gerakkan Bidawang maju sejauh 50 langkah:

forward(50)

Jika kalian mengikuti semua instruksi dengan benar, maka hasil pergerakan bidawang didalam canvas akan seperti gambar dibawah ini:



Gambar 5 (Hasil contoh menggerakan bidawang dalam canvas)

**Kuis**

1. Jika pergerakan Bidawang ke kanan adalah 200 dan ke bawah adalah -150, bagaimana koordinat tersebut dijelaskan?
2. (200, 150)
3. (-200, 150)
4. (200, -150)
5. (-200, -250)

# PERGERAKAN

PERGERAKAN

**2**

## Left & Right

**Tujuan Pembelajaran**

* Memahami cara mengendalikan arah rotasi Bidawang menggunakan left() dan right().

Perintah **left()** dan **right()** digunakan untuk memutar arah gerakan Bidawang berdasarkan sudut derajat yang diberikan, tanpa harus memindahkan posisinya. Ini berguna untuk mengatur arah Bidawang sebelum melanjutkan dengan perintah lainnya seperti bergerak.

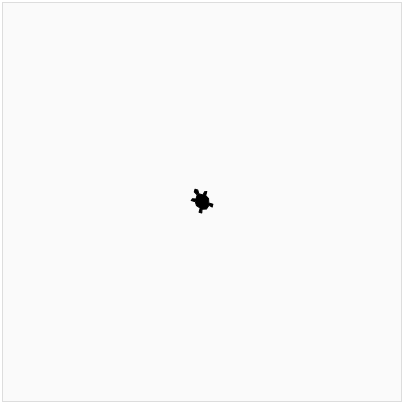
1. **Left(*derajat*)**

Digunakan untuk memutar arah Bidawang berlawanan arah jarum jam (kiri) sebesar derajat yang ditentukan.

Contoh :

left(120)

Hasil :



Gambar 6 (Hasil contoh penggunaan left)

Bidawang yang awalnya menghadap ke kanan layar, akan berputar 120 derajat ke kiri.

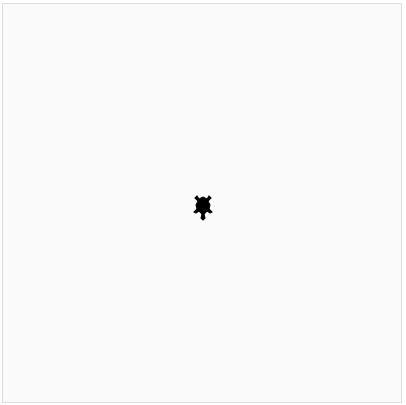
1. **Right(*derajat*)**

Digunakan untuk memutar arah Bidawang searah jarum jam (kanan) sebesar derajat yang ditentukan.

Contoh:

right(90)

Hasil:



Gambar 7 (Hasil contoh penggunaan right)

Bidawang yang awalnya menghadap ke kanan layar, akan berputar 90 derajat ke kanan.

**Latihan**

Untuk lebih memudahkan memahami cara kerja perintah left() dan right(), ikuti instruksi di bawah ini. Kalian bisa membuka aplikasi yang sudah di siapkan untuk menjalankan kodenya, kemudian ketikkan perintah tersebut, lalu tekan tombol "Run Code" untuk melihat pergerakan Bidawang, lakukan secara bertahap:

1. Berputar ke kiri

Buat bidawang berputar 90 derajat ke kiri dengan perintah di bawah ini:

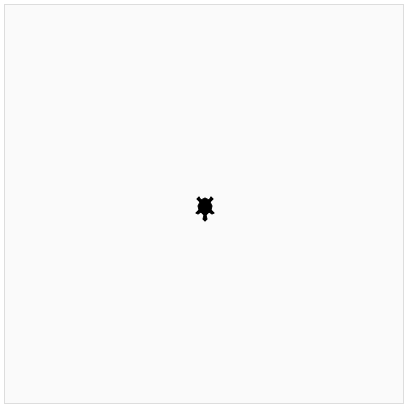
left(90)

1. Berputar ke kanan

Kemudian lanjutkan pada baris baru dengan perintah dibawah ini untuk memutar Bidawang ke kanan sebesar 180 derajat:

right(180)

Jika kalian mengikuti semua instruksi dengan benar, maka hasil pergerakan bidawang didalam canvas akan seperti gambar dibawah ini:



Gambar 8 (Hasil latihan penggunaan perintah left dan right )

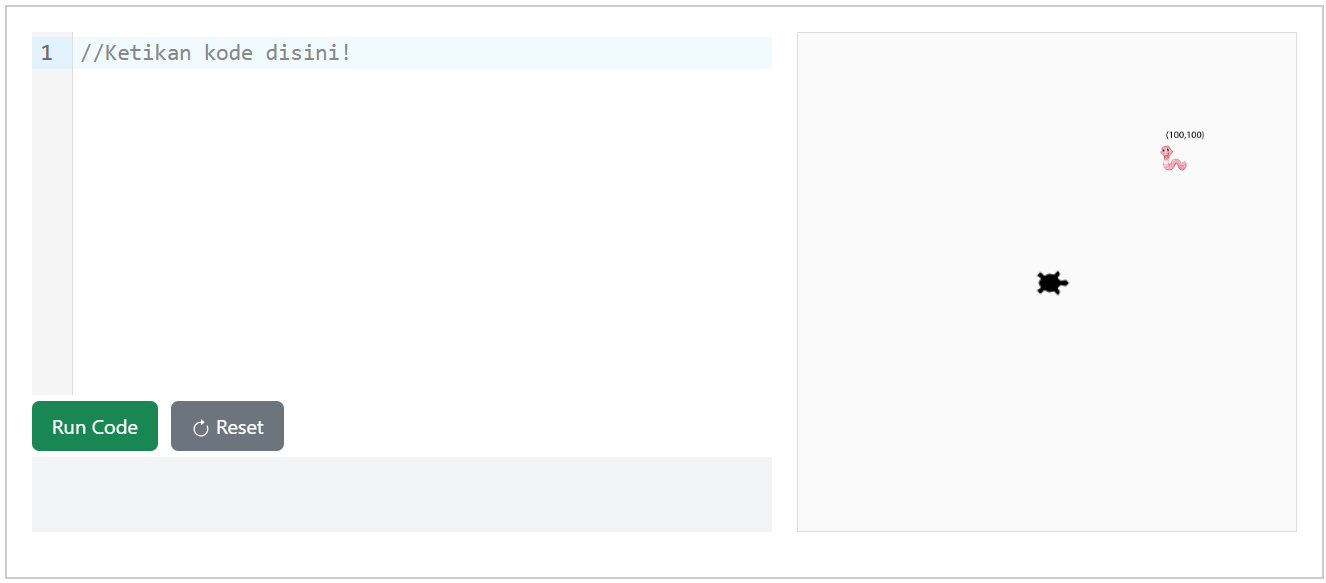
**Kesimpulan**

Perintah `left()` dan `right()` memungkinkan pengaturan arah gerakan gerakan dengan rotasi ke kiri atau ke kanan berdasarkan derajat yang ditentukan. Perintah ini sangat berguna untuk kontrol arah sebelum melakukan perintah lain dalam pembuatan gambar atau pola.

**Tantangan**

Petunjuk pengerjaan:

Bidawang saat ini berada di tengah layar (titik **(0, 0)**), sedangkan cacing berada di titik **(100, 100)**. Tugas kalian adalah **membuat Bidawang menghadap ke arah cacing** dengan 1x perintah, gunakan perintah **left()** atau **right()**:



Gambar 9 (Tantangan perintah left dan right)

Masukan kode perintah :

**Kuis**

1. Apa perbedaan utama antara perintah left() dan right() ?
2. left() memutar bidawang ke arah kanan, sementara right() memutar bidawang ke arah kiri.
3. left() memutar bidawang ke arah kiri, sementara right() memutar bidawang ke arah kanan.
4. left() dan right() hanya digunakan untuk mengubah warna turtle.
5. Keduanya memindahkan turtle ke posisi (0, 0).
6. Jika bidawang menghadap ke timur (kanan layar), dan Anda menggunakan perintah right(90), ke arah mana bidawang akan menghadap?
7. Utara (atas layar)
8. Barat (kiri layar)
9. Selatan (bawah layar)
10. Timur (kanan layar)

## Forward & Backward

**Tujuan Pembelajaran**

* Memahami cara menggerakkan Bidawang ke depan dan ke belakang menggunakan forward() dan backward().

Perintah `forward()` dan `backward()` digunakan untuk menggerakkan Bidawang ke arah depan (forward) searah arah yang sedang dihadapi Bidawang dan belakang (backward) berlawanan arah dengan yang sedang dihadapi Bidawang, berdasarkan jarak yang ditentukan dalam satuan piksel (pixel).

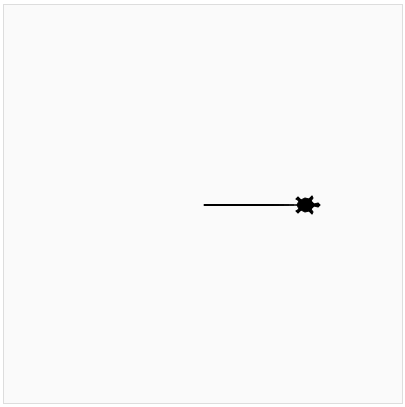
1. **Forward(*jarak*)**

Digunakan untuk menggerakkan Bidawang ke depan sejauh jarak yang ditentukan (dalam piksel), dalam arah yang sedang dihadapi oleh Bidawang.

Contoh:

forward(100)

Hasil:



Gambar 10 (Hasil contoh pengunaan forward)

Bidawang akan bergerak sejauh 100 langkah ke arah yang dihadapnya.

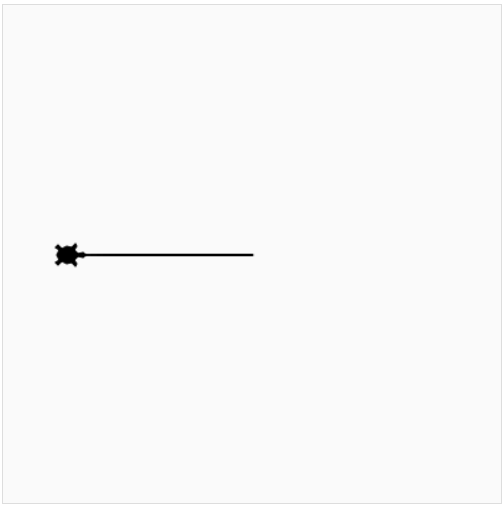
1. **Backward(*jarak*)**

Menggerakkan Bidawang ke belakang sejauh jarak yang ditentukan (dalam piksel), dalam arah berlawanan dengan arah yang sedang dihadapi oleh bidawang.

Contoh:

backward(150)

Hasil:



Gambar 11 (Hasil contoh penggunaan backward)

Bidawang akan mundur sejauh 150 langkah dari arah yang dihadapnya.

**Latihan**

Untuk lebih memudahkan memahami cara kerja perintah forward() dan backward(), ikuti instruksi di bawah ini. Kalian bisa membuka aplikasi yang sudah di siapkan untuk menjalankan kodenya, kemudian ketikkan perintah tersebut, lalu tekan tombol "Run Code" untuk melihat pergerakan Bidawang, lakukan secara bertahap:

1. Maju

Gerakkan Bidawang maju sejauh 100 langkah dengan perintah dibawah ini:

forward(100)

1. Mundur

Kemudian lanjutkan lagi pada baris baru dengan perintah dibawah ini untuk membuat Bidawang mundur 50 langkah:

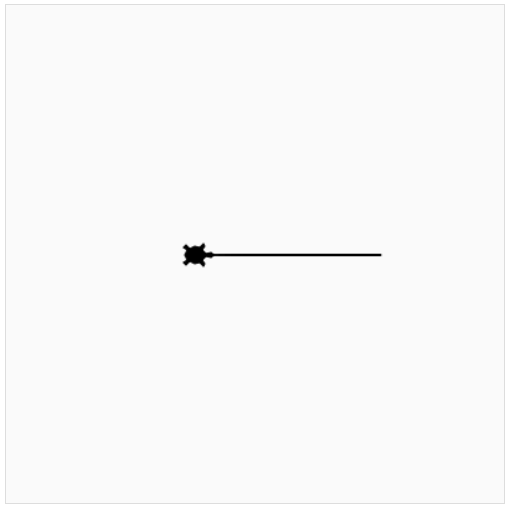
backward(50)

1. Mundur

Gerakkan lagi bidawang mundur sejauh 100 langkah:

backward(100)

Jika kalian mengikuti semua instruksi dengan benar, maka hasil akhir pergerakan bidawang didalam canvas akan seperti gambar dibawah ini:



Gambar 12 (Hasil latihan penggunaan forward dan backward)

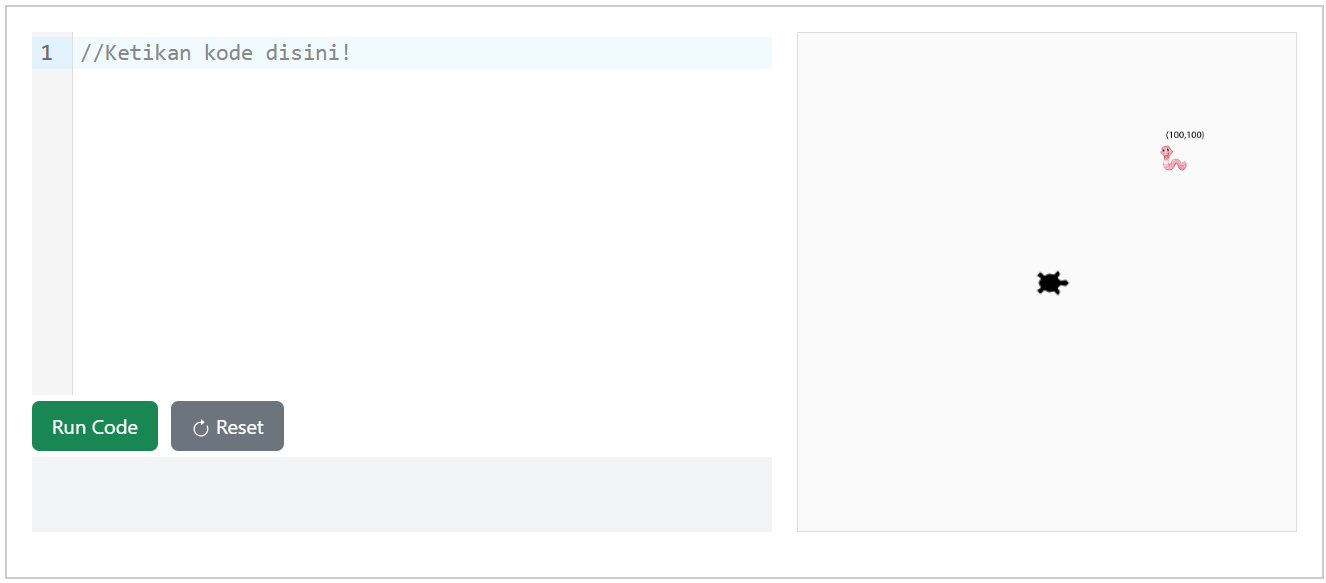
**Kesimpulan**

Perintah forward() dan backward() digunakan untuk menggerakkan Bidawang ke depan atau ke belakang sejauh jarak yang ditentukan dalam piksel. Perintah forward() dan backward() sering dikombinasikan dengan perintah rotasi seperti `left()` dan `right()` untuk membuat pola atau gambar yang lebih kompleks.

**Tantangan**

Petunjuk pengerjaan:

Bidawang saat ini berada di tengah layar (titik **(0, 0)**), sedangkan cacing berada di titik **(100, 100)**. Tugas kalian adalah **menggerakkan Bidawang menuju ke posisi cacing** dengan maksimal menggunakan 4x kode perintah. Gunakan **forward()** atau **backward()** lalu kombinasikan dengan **left()** atau **right()** untuk membuat bidawang berbelok arah.



Gambar 13 (Tantangan perintah forward dan backward)

Masukan kode perintah:

**Kuis**

1. Perintah apa yang digunakan untuk menggerakkan turtle ke depan sejauh 150 piksel?
2. backward(150)
3. forward(150)
4. left(150)
5. right(150)
6. Jika turtle menghadap ke barat dan Anda menggunakan perintah backward(100), ke arah mana turtle akan bergerak?
7. Ke barat
8. Ke timur
9. Ke utara
10. Ke Selatan

## Setposition

**Tujuan Pembelajaran**

* Memahami cara mengatur posisi Bidawang secara langsung menggunakan `setposition()`.

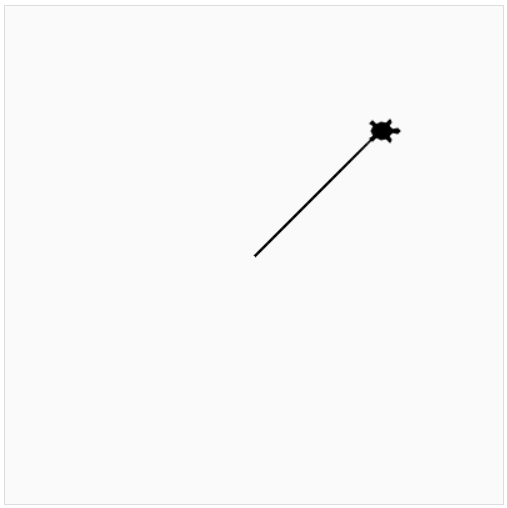
Perintah `setposition()` (bisa juga disingkat `setpos()`) digunakan untuk memindahkan Bidawang ke titik tertentu dalam ruang canvas, berdasarkan koordinat yang ditentukan. Saat perintah ini digunakan, Bidawang akan menggambar jalur dari posisi sebelumnya ke posisi baru.

**Contoh:**

Memindahkan Bidawang ke koordinat (*x* = 100, *y* = 100) dalam ruang canvas:

setposition(100, 100)

Hasil:



Gambar 14 (Hasil contoh penggunaan setposition)

Bidawang akan bergerak dari posisi awal (x=0, y=0) ke posisi x=100 dan y=100.

**Latihan**

Untuk lebih memudahkan memahami cara kerja perintah setposition() ikuti instruksi di bawah ini. Kalian bisa membuka aplikasi yang sudah di siapkan untuk menjalankan kodenya, kemudian ketikkan perintah tersebut, lalu tekan tombol "Run Code" untuk melihat pergerakan Bidawang, lakukan secara bertahap:

1. Berpindah posisi

Ubah posisi bidawang yang berada di titik awal menjadi ke titik (100,100) dengan perintah dibawah ini:

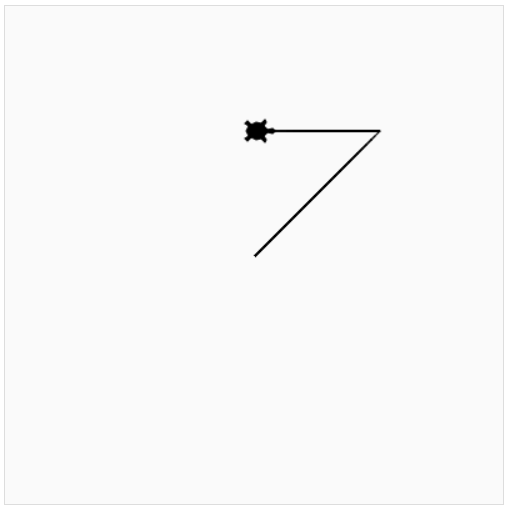
setposition(100,100)

1. Berpindah posisi

Lanjutkan pada baris baru untuk mengubah posisi bidawang lagi ke titik (0,100) dengan perintah dibawah ini:

setposition(0,100)

Jika kalian mengikuti semua instruksi dengan benar, maka hasil akhir pergerakan bidawang didalam canvas akan seperti gambar dibawah ini:

****

Gambar 15 (Hasil latihan pengunaan setposition)

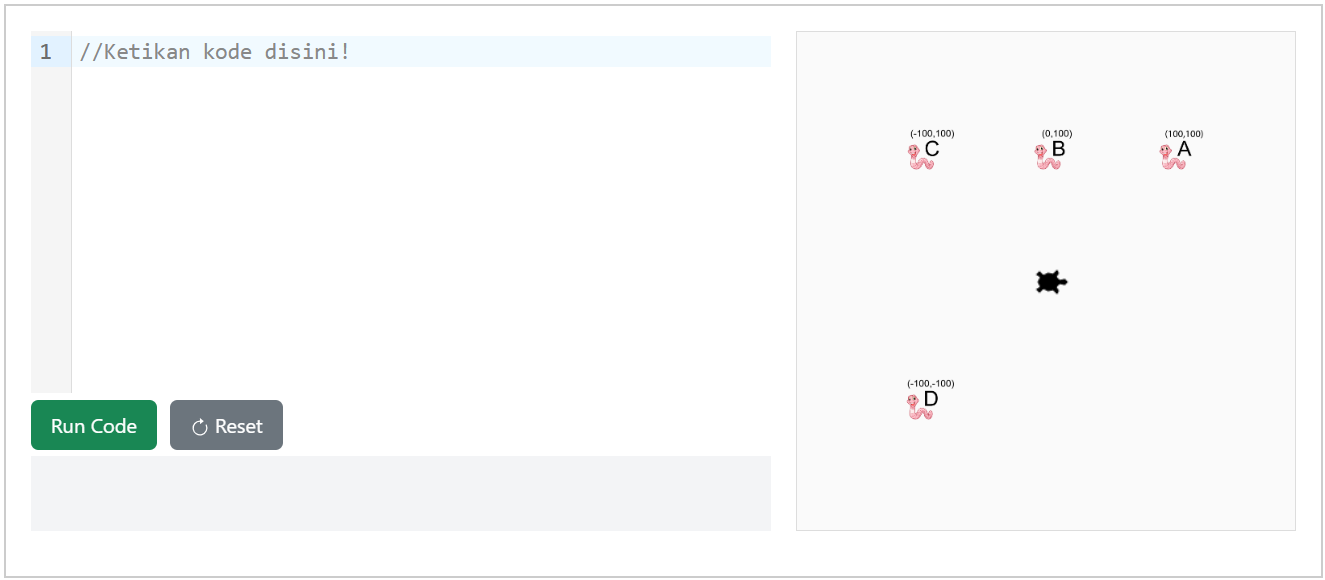
**Kesimpulan**

Perintah `setposition()` atau `setpos()` digunakan untuk memindahkan Bidawang ke koordinat tertentu dalam ruang canvas. Perintah ini sangat berguna untuk mengatur posisi awal Bidawang atau membuat jalur menggambar yang kompleks dengan mengombinasikan pergerakan dan kontrol pena.

**Tantangan**

Petunjuk pengerjaan:

Bidawang saat ini berada di tengah layar (titik **(0, 0)**). Tugas kalian adalah **memindahkan posisi Bidawang menuju ke 4 posisi yang sudah di tandai pada canvas** dengan menggunakan perintah setposition().



Gambar (Tantangan penggunaan perintah setposition)

Masukan kode perintah:

**Kuis**

1. Apa fungsi utama dari perintah setposition(x, y)?
2. Mengatur warna garis.
3. Memutar turtle ke arah tertentu.
4. Memindahkan bidawang ke koordinat tertentu dalam ruang gambar.
5. Mengubah ukuran turtle.
6. Jika posisi awal Bidawang adalah (0, 0), dan Anda menggunakan perintah setposition(50, 100), di mana posisi akhir Bidawang?
7. (100, 50)
8. (50, 0)
9. (0, 100)
10. (50, 100)

## Setx & Sety

**Tujuan Pembelajaran**

* Memahami cara mengatur posisi bidawang pada sumbu x dan y secara terpisah menggunakan `setx()` dan `sety()`.

Fungsi setx() dan sety() digunakan untuk memindahkan Bidawang ke posisi tertentu hanya pada sumbu x (setx()) atau sumbu y (sety()), tanpa mengubah koordinat lainnya. Perintah ini berguna untuk memindahkan turtle secara horizontal atau vertikal.

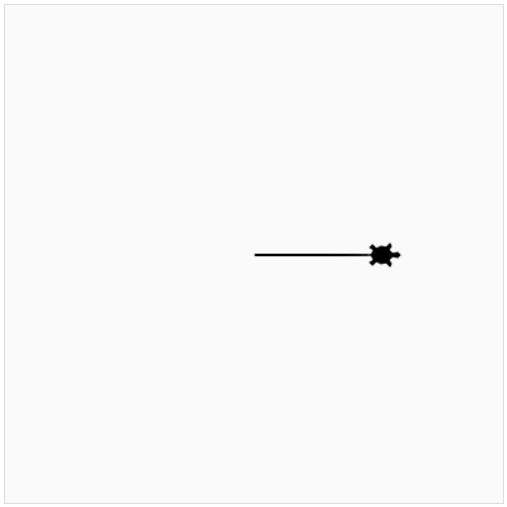
1. **Setx(*koordinat x*)**

Digunakan untuk memindahkan Bidawang ke posisi x yang ditentukan, tanpa mengubah posisi pada sumbu y.

Contoh:

setx(100)

Hasil:



Gambar 17 (Hasil contoh penggunaan setx)

Posisi awal Bidawang adalah x=0,y=0. Setelah dijalankan kode tersebut maka nilai x akan menjadi 100. Dan Bidawang akan berpindah ke titik x=100,y=0.

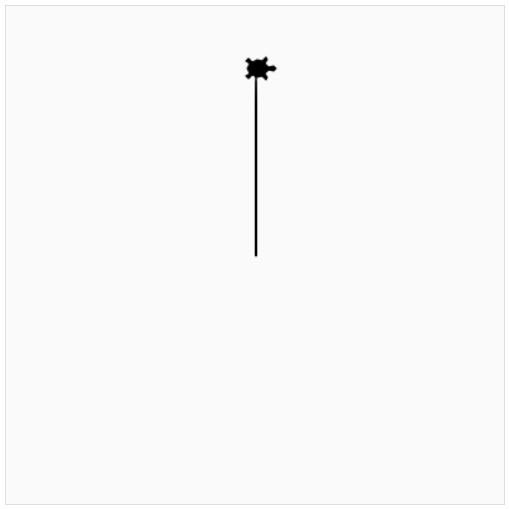
1. **Sety(*koordinat y*)**

Digunakan untuk memindahkan Bidawang ke posisi y yang ditentukan, tanpa mengubah posisi pada sumbu x.

Contoh:

Sety(150)

Hasil:



Gambar 18 (Hasil contoh penggunaan sety)

Posisi awal bidawang adalah x=0, y=0. Setelah dijalankan kode tersebut maka nilai y akan menjadi 150. Dan bidawang akan berpindah ke titik x=0, y=150.

**Latihan**

Untuk lebih memudahkan memahami cara kerja perintah setx() dan sety() ikuti instruksi di bawah ini. Kalian bisa membuka aplikasi yang sudah di siapkan untuk menjalankan kodenya, kemudian ketikkan perintah tersebut, lalu tekan tombol "Run Code" untuk melihat pergerakan Bidawang, lakukan secara bertahap:

1. Ubah titik X

Ubah nilai titik **X** bidawang yang awaknya berada di titik awal (0,0) ubah titik **X** nya menjadi 100 dengan perintah dibawah ini:

setx(100)

1. Ubah titik Y

Kemudian lanjutkan pada baris baru untuk mengubah nilai titik **Y** menjadi -50 dengan perintah dibawah ini:

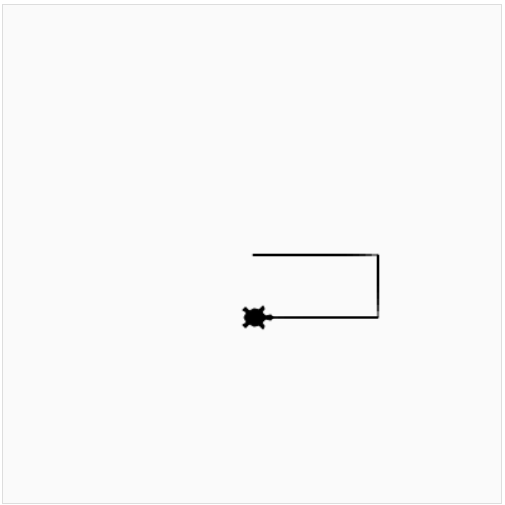
sety(-50)

1. Ubah titik X

Lalu ubah lagi nilai titik **X** menjadi 0 dengan perintah dibawah ini:

setx(0)

Jika kalian mengikuti semua instruksi dengan benar, maka hasil akhir pergerakan bidawang didalam canvas akan seperti gambar dibawah ini:

****

Gambar 19 (Hasil latihan penggunaan setx dan sety)

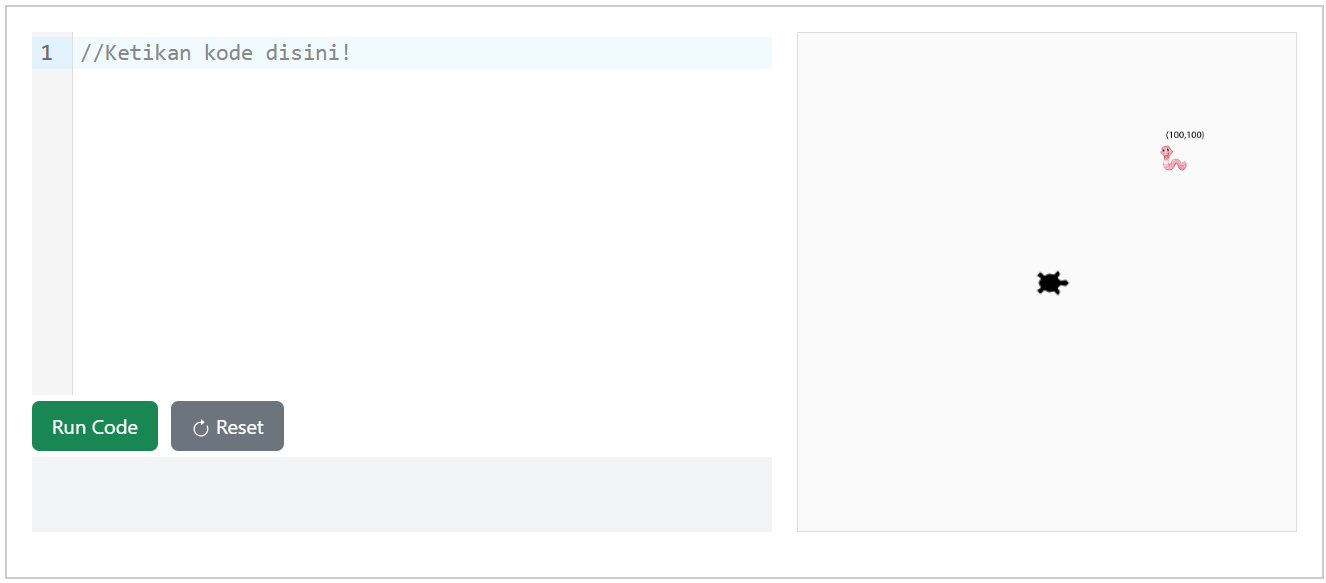
**Kesimpulan**

Perintah setx() dan sety() digunakan untuk memindahkan Bidawang secara horizontal atau vertikal tanpa mengubah koordinat lainnya. Perintah ini berguna untuk memindahkan Bidawang ke posisi yang diinginkan dengan lebih fleksibel, terutama saat mengatur pola atau jalur yang spesifik.

**Tantangan**

Petunjuk pengerjaan:

Bidawang saat ini berada di tengah layar (titik **(0, 0)**), sedangkan cacing berada di titik **(100, 100)**. Tugas kalian adalah **memindahkan posisi Bidawang menuju ke posisi cacing** dengan menggunakan perintah **setx** dan **sety()**.



Gambar 20 (Tantangan perintah setx dan sety)

Masukan kode perintah:

**Kuis**

1. Apa hasil dari perintah setx(200) jika posisi awal turtle adalah (100,50)?
2. Posisi baru menjadi (200, 50).
3. Posisi baru menjadi (100, 200).
4. Posisi baru menjadi (200, 200).
5. Posisi baru tetap (100, 50).
6. Jika Anda ingin memindahkan turtle secara vertikal ke posisi y = -50 tanpa mengubah posisi x, perintah apa yang harus digunakan?
7. sety(-50)
8. setx(-50)
9. setposition(-50, 0)
10. home()

## Setheading

**Tujuan Pembelajaran**

* Memahami cara mengatur arah Bidawang menggunakan setheading().

Perintah setheading() digunakan untuk mengatur arah Bidawang ke sudut tertentu, diukur dalam derajat. Sudut dihitung berlawanan arah jarum jam, dimulai dari arah timur (0 derajat). Dengan menggunakan setheading() dapat mengontrol ke mana bidawang akan menghadap sebelum bergerak.

* **0°** : Timur
* **90°** : Utara
* **180°** : Barat
* **270°** : Selatan

**Contoh**

Mengatur arah pergerakan bidawang dengan setheading():

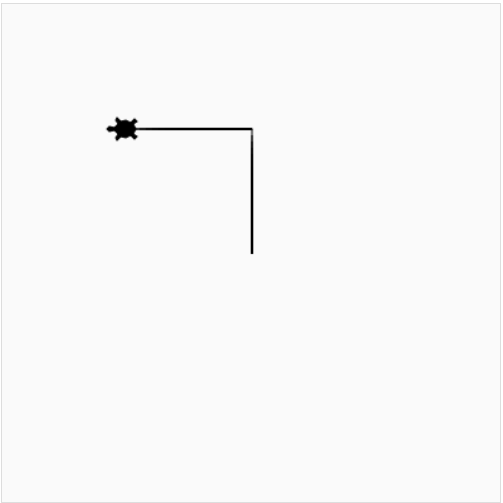
setheading(90)

forward(100) # Bergerak maju ke atas

setheading(180)

forward(100) # Bergerak maju ke kiri

Hasil:



Gambar 21 (Hasil contoh penggunaan setheading)

**Latihan**

Untuk lebih memudahkan memahami cara kerja perintah setheading() ikuti instruksi di bawah ini. Kalian bisa membuka aplikasi yang sudah di siapkan untuk menjalankan kodenya, kemudian ketikkan perintah tersebut, lalu tekan tombol "Run Code" untuk melihat pergerakan Bidawang, lakukan secara bertahap:

1. Hadap Utara

Ubah arah bidawang menjadi menghadap ke arah utara dengan menggunakan perintah dibawah ini:

setheading(90)

1. Maju

Lalu lanjutkan pada baris baru untuk membuat bidawang maju sejauh 100 langkah dengan perintah dibawah ini:

forward(100)

1. Hadap Timur

Lanjutkan lagi dengan mengubah arah bidawang menjadi menghadap ke timur dengan perintah dibawah ini:

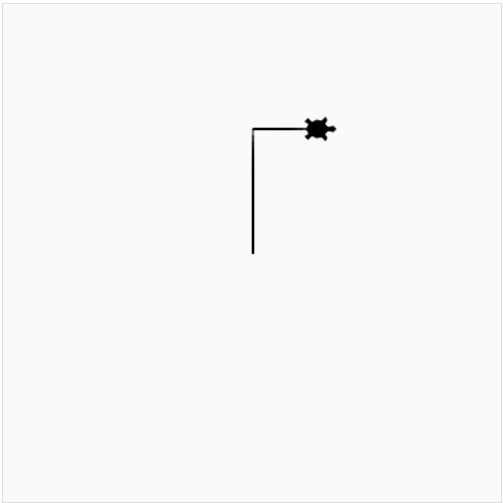
setheading(0)

1. Maju

Buat bidawang maju sejauh 50 langkah lagi.

forward(50)

Jika kalian mengikuti semua instruksi dengan benar, maka hasil akhir pergerakan bidawang didalam canvas akan seperti gambar dibawah ini:

****

Gambar 22 (Hasil latihan penggunaan setheading)

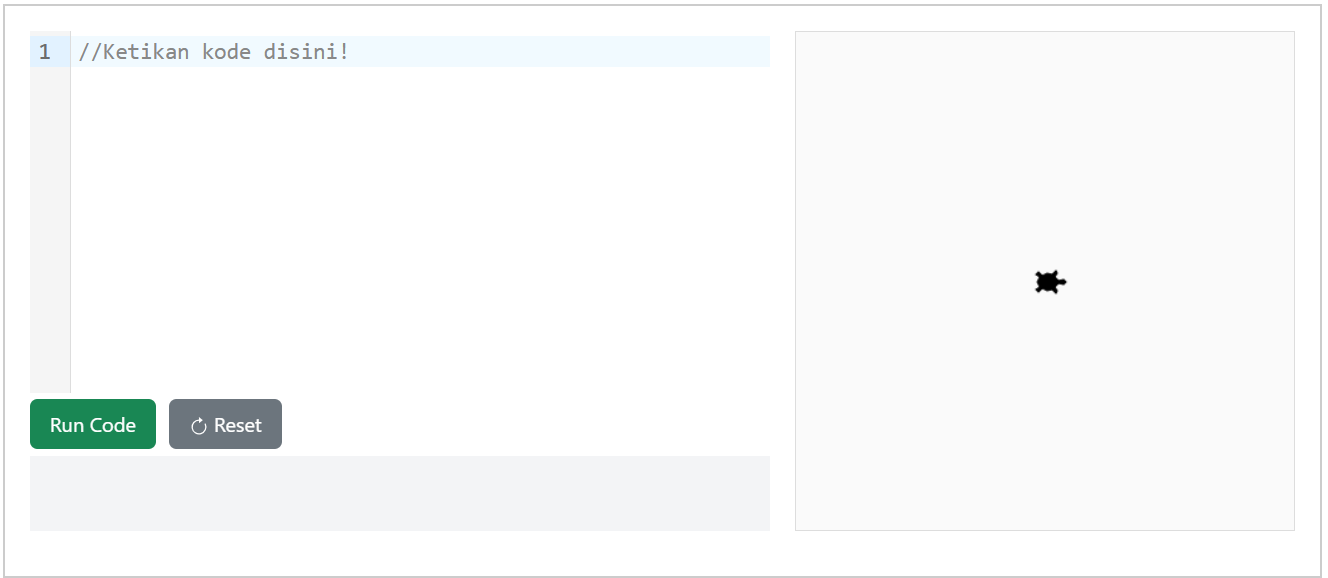
**Kesimpulan**

Perintah setheading() sangat berguna untuk mengontrol arah objek dengan presisi. Dengan mengatur sudut arah secara langsung, Anda dapat membuat pola yang kompleks dan menggambar dengan lebih terstruktur.

**Tantangan**

Petunjuk pengerjaan:

Bidawang saat ini berada di tengah layar (titik **(0, 0)**). Tugas kalian adalah **buat bidawang bergerak mencapai ujung canvas arah selatan**. Gunakan perintah setheading untuk mengubah arahnya.



Gambar (Tantangan perintah setheading)

Masukan kode perintah:

**Kuis**

1. Jika Anda menggunakan perintah setheading(90), ke arah mana Bidawang akan menghadap?
2. Timur
3. Barat
4. Utara
5. Selatan
6. Top of Form
7. Apa fungsi utama dari perintah setheading(*derajat*)?
8. Mengatur posisi turtle ke koordinat (0, 0).
9. Mengatur arah turtle ke sudut tertentu berdasarkan derajat.
10. Menggerakkan turtle ke arah utara.
11. Mengubah warna turtle.Bottom of Form

## Home

**Tujuan Pembelajaran**

* Memahami cara mengembalikan posisi Bidawang ke titik awal menggunakan home().

Perintah `home()` digunakan untuk memindahkan Bidawang kembali ke posisi awalnya, yaitu titik (0, 0). Selain memindahkan Bidawang ke posisi awal, perintah ini juga mengatur arah Bidawang menghadap ke timur (0 derajat). Ini berguna ketika Anda ingin memulai kembali menggambar dari posisi awal.

**Contoh**

Memindahkan bidawang ke posisi awal setelah menjalankan berbagai perintah:

forward(100)

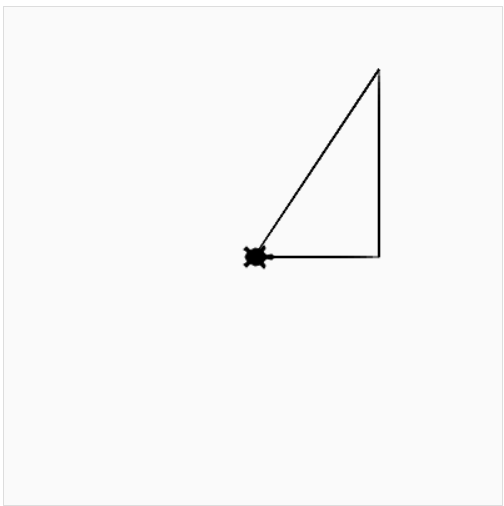
left(90)

forward(150)

# Kembali ke posisi awal

home()

Hasil:



Gambar 24 (Hasil contoh penggunaan home)

Fungsi home() akan membuat Bidawang kembali ke posisi awal setelah menjalankan perintah lainnya.

**Latihan**

Untuk lebih memudahkan memahami cara kerja perintah home() ikuti instruksi di bawah ini. Kalian bisa membuka aplikasi yang sudah di siapkan untuk menjalankan kodenya, kemudian ketikkan perintah tersebut, lalu tekan tombol "Run Code" untuk melihat pergerakan Bidawang, lakukan secara bertahap:

1. Gerakkan Bidawang

Buat bidawang menjadi bergerak ke arah mana saja, sebagai contoh gunakan perintah di bawah ini untuk menggerakan bidawang:

forward(100)

right(45)

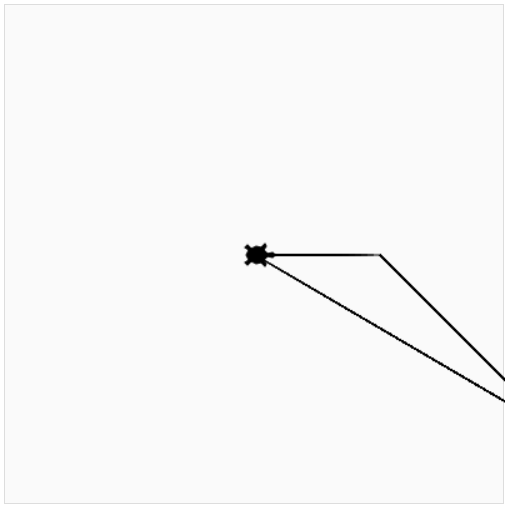
forward(200)

1. Kembali ke posisi awal

Selanjutnya ketikan perintah di bawah ini pada baris baru untuk mengembalikan bidawang ke posisi awal.

home()

Jika kalian mengikuti semua instruksi dengan benar, maka hasil akhir pergerakan bidawang didalam canvas akan seperti gambar dibawah ini:

****

Gambar 25 (Hasil latihan penggunaan home)

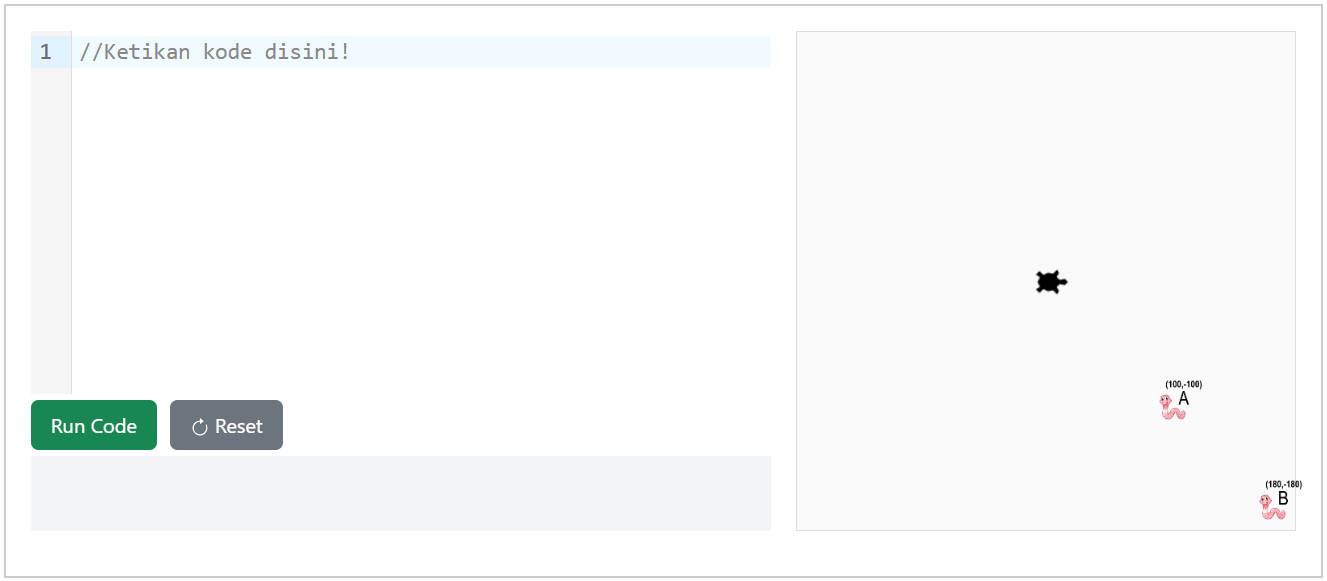
**Kesimpulan**

Perintah home() memudahkan untuk kembali ke posisi awal (0, 0) dan mengatur arah Bidawang ke posisi semula. Ini berguna untuk memulai kembali proses menggambar dari titik awal.

**Tantangan**

Petunjuk pengerjaan:

Bidawang saat ini berada di tengah layar (titik **(0, 0)**). Tugas kalian adalah Gerakan bidawang untuk mencapai 2 titik yang ditandai pada canvas kemudian buat bidawang kembali ke posisi awal.



Gambar (Tantangan perintah home)

Masukan kode perintah:

**Kuis**

1. Apa fungsi dari perintah home()?
2. Menghapus seluruh gambar yang telah dibuat.
3. Mengembalikan Bidawang ke posisi awal (0, 0) dan mengatur arahnya ke timur.
4. Utara
5. Memindahkan Bidawang ke posisi y = 0.
6. Jika posisi awal Bidawang adalah (100, 100) dan arahnya ke barat, apa yang terjadi setelah menggunakan home()?
7. Bidawang tetap di posisi (100, 100).
8. Bidawang kembali ke posisi (0, 0) dengan arah tetap ke barat.
9. Bidawang kembali ke posisi (0, 0) dan menghadap ke timur.
10. Bidawang tetap di posisi (100, 100) tetapi menghadap ke timur.

## Circle

**Tujuan Pembelajaran**

* Memahami cara menggambar lingkaran menggunakan circle().

Fungsi circle() digunakan untuk menggambar lingkaran atau bagian dari lingkaran (busur) dengan jari-jari tertentu. Fungsi ini sangat berguna untuk menggambar bentuk bulat dan pola yang melibatkan lingkaran.

circle(jari\_jari, extent=None): Menggambar lingkaran dengan jari-jari yang ditentukan. Parameter extent opsional dan digunakan untuk menggambar busur lingkaran. Jika extent tidak diberikan, maka lingkaran penuh akan digambar.

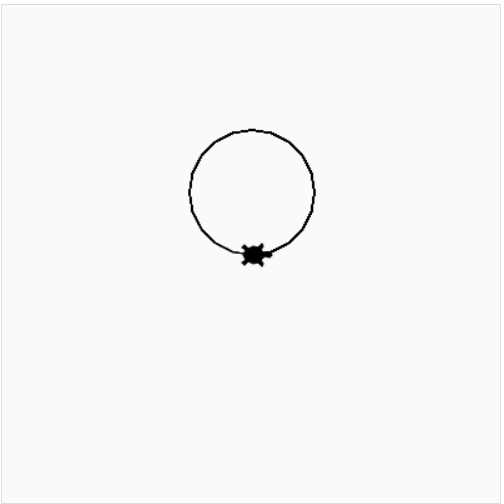
* **jari\_jari :** nilai jari-jari lingkaran yang ingin digambar.
* **Extent :** Sudut (dalam derajat) dari lingkaran yang ingin digambar. Jika tidak ada, menggambar lingkaran penuh (360 derajat).

**Contoh 1**

Menggambar lingkaran penuh dengan menentukan jari-jari:

circle(50)

Hasil:



Gambar 27 (Hasil contoh 1 penggunaan circle)

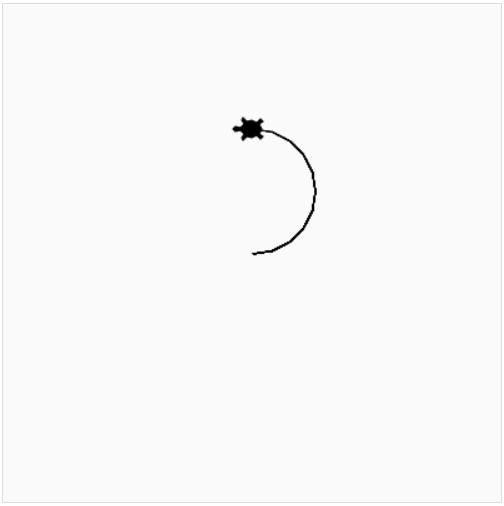
Fungsi circle(50) akan membuat Bidawang menggambar lingkaran dengan jari jari 50.

**Contoh 2**

Menggambar busur dengan menentukan besaran derajatnya *'extent'*:

circle(50, 180)

Hasil:



Gambar 28 (Hasil contoh 2 penggunaan circle)

Fungsi circle(50,180) akan membuat Bidawang menggambar lingkaran dengan jari jari 50, dengan besar busur 180 derajat.

**Latihan**

Untuk lebih memudahkan memahami cara kerja perintah circle() ikuti instruksi di bawah ini. Kalian bisa membuka aplikasi yang sudah di siapkan untuk menjalankan kodenya, kemudian ketikkan perintah tersebut, lalu tekan tombol "Run Code" untuk melihat pergerakan Bidawang, lakukan secara bertahap:

1. Menggambar Lingkaran penuh

Buat bidawang menggambar lingkaran penuh dengan jari-jari 100 menggunakan perintah dibawah ini:

circle(100)

1. Berpindah posisi

Kemudian pada baris berikutnya gunakan perintah setposition untuk memindahkan posisi bidawang ke (0,-200):

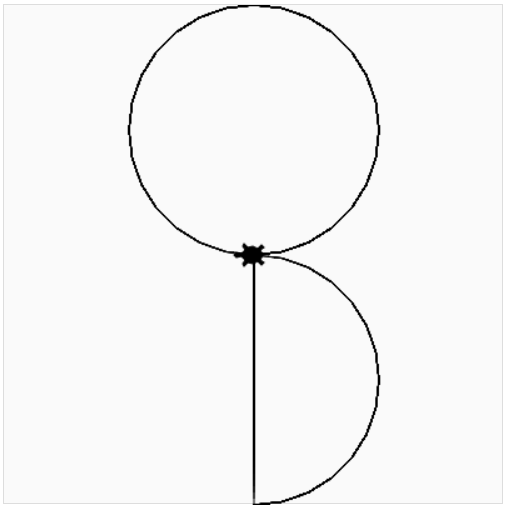
setposition(0,-200)

1. Menggambar busur

Lalu lanjutkan lagi menggambar busur dengan besar 180 derajat dan jari jari 100 menggunakan perintah dibawah ini:

circle(100,180)

Jika kalian mengikuti semua instruksi dengan benar, maka hasil akhir pergerakan bidawang didalam canvas akan seperti gambar dibawah ini:

****

Gambar 29 (Hasil latihan penggunaan circle)

**Kesimpulan**

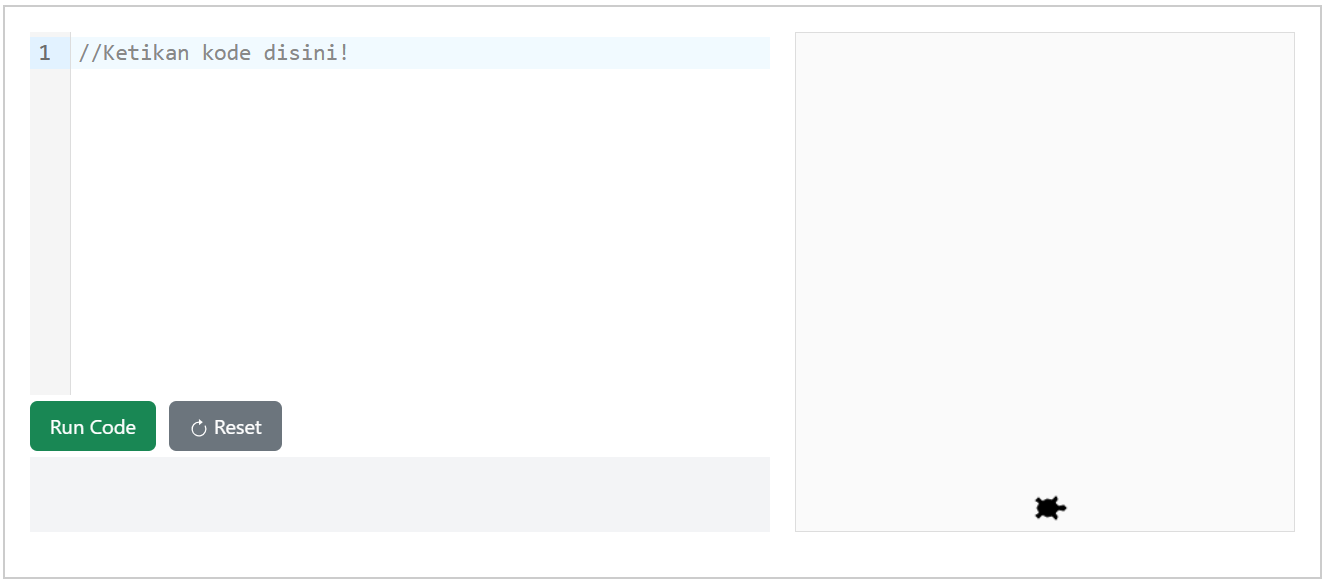
Perintah **circle()** digunakan untuk menggambar lingkaran atau busur dengan mudah. Dengan mengatur jari-jari dan opsi `extent`, kita dapat menggambar berbagai bentuk bulat yang berbeda untuk memperkaya gambar atau pola yang dibuat.

**Tantangan**

Petunjuk pengerjaan:

Bidawang saat ini berada di titik **(0, -200)**. Tugas kalian adalah:

* Buat lingkaran dengan jari jari 180.
* Pindahkan posisi bidawang naik ke atas sejauh 50 langkah kemudian gambar lingkaran lagi dengan jari jari 130.
* Pindahkan lagi posisi bidawang naik ke atas sejauh 50 langkah kemudian gambar busur dengan jari jari 80 dan sudut 180 derajat.



Gambar (Tantangan perintah circle)

Masukan kode perintah:

**Kuis**

1. Apa fungsi dari parameter extent dalam metode circle()?
2. Mengatur arah lingkaran.
3. Menentukan ukuran jari-jari lingkaran.
4. Menentukan besaran sudut busur yang ingin Digambar.
5. Mengubah warna lingkaran.
6. Apa yang terjadi jika Anda menjalankan perintah circle(-50)?
7. Tidak ada lingkaran yang Digambar.
8. Lingkaran dengan jari-jari 50 akan digambar berlawanan dengan arah.
9. Lingkaran dengan jari-jari 50 akan digambar dengan arah normal (berlawanan arah jarum jam)
10. Lingkaran kecil akan digambar di posisi saat ini.

## Dot

**Tujuan Pembelajaran**

* Memahami cara menggambar titik menggunakan dot().

Fungsi dot() digunakan untuk menggambar titik pada posisi Bidawang berada. Titik ini dapat diatur ukuran dan warnanya. Fungsi ini sangat berguna untuk menandai posisi tertentu dalam gambar atau untuk menggambar pola dengan titik.

**dot(ukuran, warna):** Menggambar titik dengan ukuran dan warna yang ditentukan. Parameter warna opsional dan dapat diisi dengan nama warna atau kode heksadesimal.

* **Ukuran** : Ukuran titik yang ingin digambar. Nilai default adalah 5.
* **Warna** : Warna titik yang ingin digambar. Jika tidak ditentukan, warna default adalah warna pena saat ini.

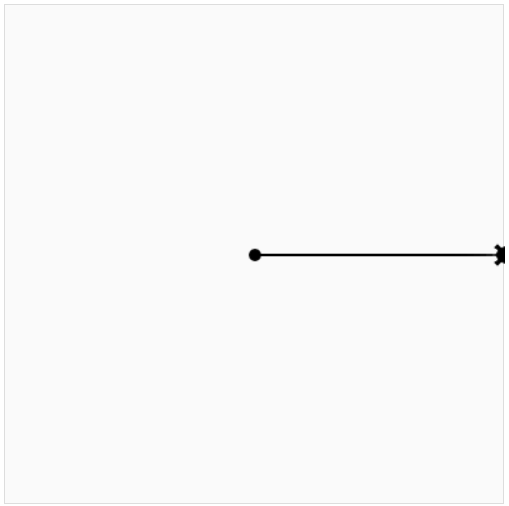
**Contoh 1**

Menggambar titik dengan ukuran 10 dan warna default:

dot(10)

forward(200)

Hasil:



Gambar 31 (Hasil contoh 1 penggunaan dot)

Fungsi dot(10) akan menggambar titik pada posisi bidawang tersebut dengan ukuran 10 dan warna default.

**Contoh 2**

Menggambar Titik dengan Ukuran dan Warna yang Berbeda:

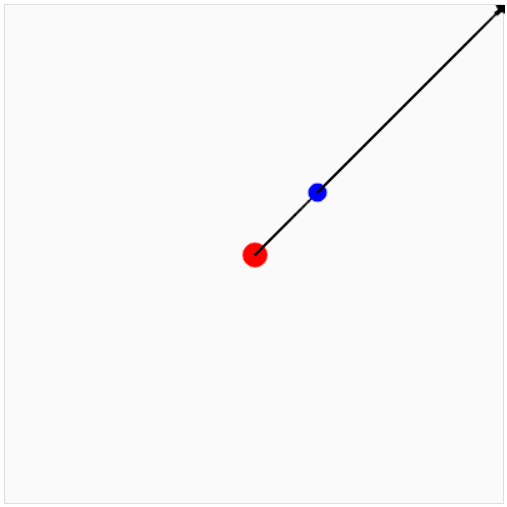
dot(20, "red")

setposition(50, 50)

dot(15, "blue")

setposition(200, 200)

Hasil:



Gambar 32 (Hasil contoh 2 penggunaan dot)

Fungsi **dot(20,"red")** akan membuat Bidawang menggambar titik dengan ukuran 20 dan warna merah, sedangkan **dot(15,"blue")** akan menggambar titik dengan ukuran 15 dan warna biru.

**Latihan**

Untuk lebih memudahkan memahami cara kerja perintah dot() ikuti instruksi di bawah ini. Kalian bisa membuka aplikasi yang sudah di siapkan untuk menjalankan kodenya, kemudian ketikkan perintah tersebut, lalu tekan tombol "Run Code" untuk melihat pergerakan Bidawang, lakukan secara bertahap:

1. Menggambar titik

Buat bidawang menggambar titik dengan warna default dan ukuran 20:

dot(20)

1. Berpindah posisi

Lanjutkan pada baris berikutnya untuk memindahkan posisi bidawang dengan perintah dibawah ini:

forward(100)

1. Menggambar titik

Gambar titik lagi dengan warna biru dan ukuran 20:

dot(20,”blue”)

1. Berpindah posisi

Pindahkan lagi posisi bidawang dengan menggunakan perintah dibawah ini:

left(120)

forward(100)

1. Menggambar titik

Setelah berpindah gambar titik lagi berwarna hijau dengan ukuran 50:

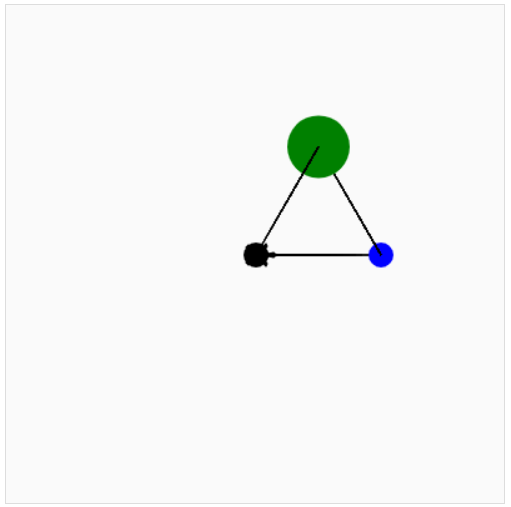
dot(50,”green”)

1. Berpindah posisi

Kemudian pindahkan lagi posisi bidawang ke titik awal dengan perintah home:

home()

Jika kalian mengikuti semua instruksi dengan benar, maka hasil akhir pergerakan bidawang didalam canvas akan seperti gambar dibawah ini:

****

Gambar 33 (Hasil latihan penggunaan dot)

**Kesimpulan**

Perintah dot() berfungsi untuk menggambar titik dengan ukuran dan warna yang dapat disesuaikan. Fungsi ini berguna untuk menandai posisi, membuat pola, atau menambah detail pada gambar yang dibuat.

**Tantangan**

Petunjuk pengerjaan:

Bidawang saat ini berada di titik **(-200, 0)**. Tugas kalian adalah Menggerakan bidawang menuju titik **(200, 0)**, namun pada setiap 50 langkah buat bidawang menggambar titik berwarna merah dan hijau secara selang seling dengan ukuran 15.



Gambar (Tantangan perintah dot)

Masukan kode perintah:

**Kuis**

1. Apa fungsi dari metode dot()?
2. Menggambar lingkaran penuh.
3. Menggambar busur lingkaran.
4. Menggambar titik di posisi saat ini.
5. Menghapus titik pada kanvas.
6. Apa yang terjadi jika Anda memanggil dot(15, "blue")?
7. Titik biru dengan ukuran 15 piksel digambar di posisi saat ini.
8. Lingkaran biru dengan ukuran 15 piksel digambar di posisi saat ini.
9. Lingkaran penuh biru dengan jari-jari 15 digambar di posisi saat ini.
10. Tidak ada yang terjadi.

# MENGETAHUI STATUS

**3**

MENGETAHUI STATUS

## Position

**Tujuan Pembelajaran**

* Memahami fungsi dan penggunaan position().

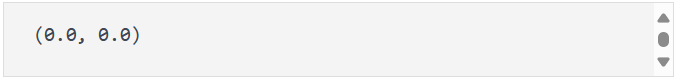
Fungsi position() digunakan untuk mendapatkan posisi saat ini dari bidawang di dalam canvas. Posisi ini dinyatakan dalam bentuk pasangan koordinat (x, y). Untuk menampilkan koordinatnya kita bisa menggunakan fungsi print().

**Contoh 1**

Menampilkan posisi saat ini dari bidawang:

print(position())

Hasil:



Gambar (Hasil contoh 1 penggunaan position)

Fungsi position() digunakan untuk mencari tahu posisi bidawang saat ini, kemudian position() akan membuat posisi tersebut di tampilkan pada bagian output log.

**Contoh 2**

Memeriksa posisi Bidawang sebelum dan sesudah pergerakan:

# Posisi awal

print("Posisi awal:", position())

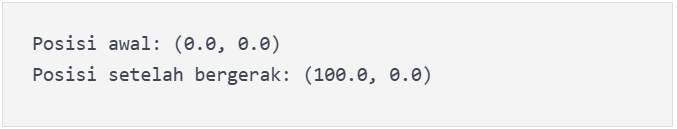
# Gerakkan Bidawang ke depan sejauh 100 langkah

forward(100)

# Cek posisi baru

print("Posisi setelah bergerak:", position())

Hasil:



Gambar (Hasil contoh 2 penggunaan position)

Sebelum di gerakan posisi awal Bidawang adalah *(0,0)*, kemudian setelah diigerakan dengan forward(100) posisinya menjadi *(100, 0).*

**Latihan**

Untuk lebih memudahkan memahami cara kerja perintah position() ikuti instruksi di bawah ini. Kalian bisa membuka aplikasi yang sudah di siapkan untuk menjalankan kodenya, kemudian ketikkan perintah tersebut, lalu tekan tombol "Run Code" untuk melihat pergerakan Bidawang, lakukan secara bertahap:

1. Cek Posisi

Lakukan pemeriksaan posisi bidawang terlebih dahulu menggunakan perintah di bawah ini:

print(position())

1. Maju

Kemudian lanjutkan pada baris baru untuk membuat bidawang maju sejauh 150 langkah dengan perintah dibawah ini:

forward(150)

1. Cek Posisi

Lalu periksa lagi posisi bidawang tersebut:

print(position())

1. Pindah Posisi

Selanjutnya pindahkan lagi bidawang ke posisi x=75 dan y=75 dengan perintah di bawah ini:

setposition(75,75)

1. Cek Posisi

Lanjutkan dengan memeriksa posisi bidawang lagi:

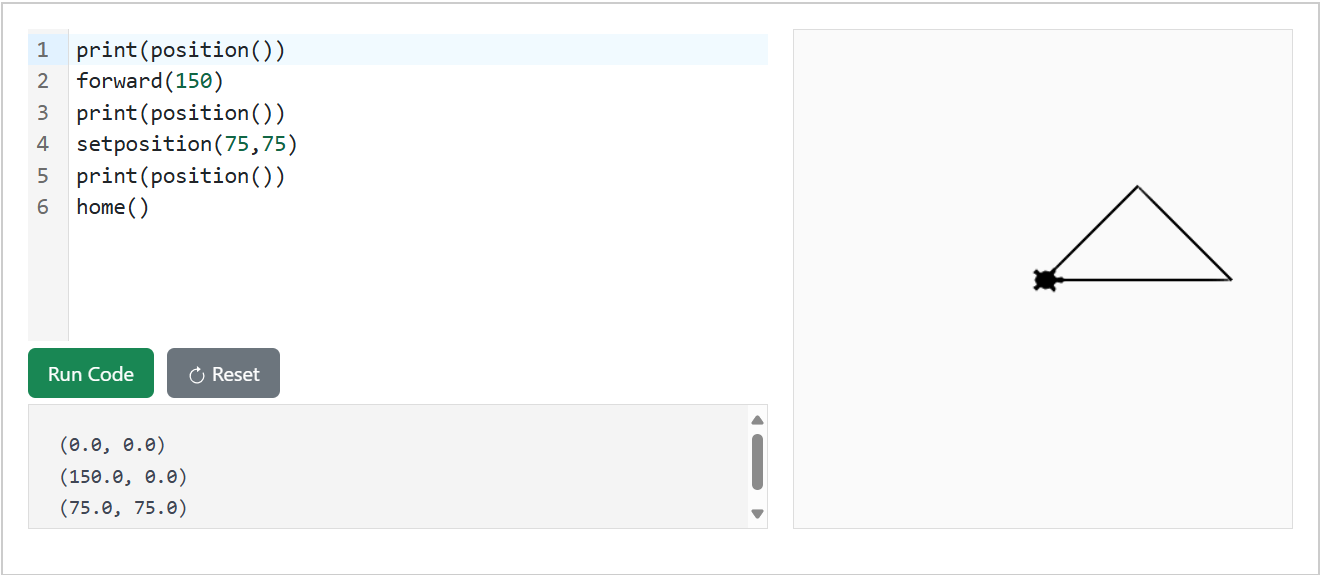
print(position())

1. Kembali ke posisi awal

Gunakan perintah home untuk membuat bidawang kembali ke titik awal (0,0):

home()

Jika kalian mengikuti semua instruksi dengan benar, maka hasil akhir pergerakan bidawang akan menggambar segitiga didalam canvas dan akan menampilkan posisi pada bagian output-log seperti dibawah ini:

****

Gambar (Hasil latihan penggunaan position)

**Kesimpulan**

Perintah position() untuk mengetahui posisi saat ini dari Bidawang. Fungsi ini berguna untuk memantau pergerakan dan membantu dalam menggambar pola atau bentuk yang presisi.

**Tantangan**

PR

**Kuis**

1. Apa fungsi dari position()?
2. Memberitahu arah pergerakan turtle saat ini.
3. Menghapus posisi turtle sebelumnya.
4. Memberitahu posisi turtle saat ini dalam bentuk koordinat (x, y).
5. Mengatur posisi turtle ke titik tertentu.
6. Apa hasil dari perintah berikut jika turtle berada di posisi (50, 100)?

print(position())

1. (50, 100)
2. 0
3. (0, 0)
4. [50, 100]

## Xcor & Ycor

**Tujuan Pembelajaran**

* Mengetahui cara memeriksa posisi horizontal (sumbu x) dan vertikal (sumbu y) dari bidawang.

Fungsi xcor() dan ycor() digunakan untuk mendapatkan koordinat posisi spesifik di sepanjang sumbu horizontal (x) dan vertikal (y). Metode-metode ini ideal untuk memantau posisi terkini, membantu menentukan apakah bidawang telah mencapai batas tertentu, atau digunakan dalam penghitungan pola geometris yang memerlukan pengawasan koordinat. Sama seperti perintah sebelumnya (*Position()*) Untuk menampilkan hasilnya kita bisa menggunakan fungsi print().

* xcor(): Mengembalikan posisi horizontal (sumbu x) bidawang.
* ycor(): Mengembalikan posisi vertikal (sumbu y) bidawang.

**Contoh**

Menampilkan posisi horizontal (sumbu x) dan vertikal (sumbu y) dari bidawang sebelum dan sesudah bergerak:

# Menampilkan posisi awal bidawang

print("Posisi x awal:", xcor())

print("Posisi y awal:", ycor())

# Memindahkan bidawang ke posisi lain

left(45)

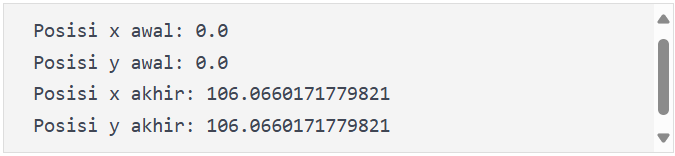
forward(150)

# Menampilkan posisi akhir bidawang

print("Posisi x akhir:", xcor())

print("Posisi y akhir:", ycor())

Hasil:



Gambar (Hasil contoh penggunaan xcor dan ycor)

Fungsi xcor() akan menampilkan posisi horizontal (sumbu x) dari bidawang, kemudian ycor() akan menampilkan posisi vertikal (sumbu y) dari bidawang.

**Latihan**

Untuk lebih memudahkan memahami cara kerja perintah xcor() dan ycor() ikuti instruksi di bawah ini. Kalian bisa membuka aplikasi yang sudah di siapkan untuk menjalankan kodenya, kemudian ketikkan perintah tersebut, lalu tekan tombol "Run Code" untuk melihat pergerakan Bidawang, lakukan secara bertahap:

1. Cek posisi x dan y

Lakukan pemeriksaan posisi x dan y menggunakan perintah di bawah ini:

print("x : ", xcor())

print("y : ", ycor())

1. Pindahkan posisi

Kemudian lanjutkan lagi pada baris baru untuk menggerakan bidawang dengan perintah di bawah ini:

left(90)

forward(100)

1. Cek posisi x dan y

Lakukan pemeriksaan lagi untuk posisi x dan y menggunakan perintah di bawah ini:

print("x setelah bergerak : ", xcor())

print("y setelah bergerak : ", ycor())

**Kesimpulan**

Perintah xcor() dan ycor() berfungsi untuk mengetahui posisi bidawang secara terpisah pada sumbu x dan y. Perintah ini membantu menentukan apakah turtle telah mencapai batas tertentu, atau digunakan dalam penghitungan pola geometris yang memerlukan pengawasan koordinat.

**Tantangan**

PR

**Kuis**

1. Apa perbedaan antara xcor() dan ycor()?
2. xcor() menampilkan posisi horizontal (sumbu x), sementara ycor() menampilkan posisi vertikal (sumbu y).
3. xcor() mengatur posisi horizontal, sementara ycor() mengembalikan posisi vertikal.
4. xcor() menghapus posisi pada sumbu x, sementara ycor() membaca posisi awal.
5. Tidak ada perbedaan, keduanya menampilkan posisi (x, y).
6. Apa hasil dari perintah berikut jika turtle berada di koordinat (30, 20)?

print(xcor())

1. -20
2. 30
3. (30, -20)
4. 0

## Heading

**Tujuan Pembelajaran**

* Mengetahui cara memeriksa arah pergerakan dari bidawang menggunakan heading().

Fungsi heading() digunakan untuk menampilkan arah pergerakan bidawang dalam satuan derajat. Sama seperti perintah sebelumnya (*Position()*) Untuk menampilkan hasilnya kita bisa menggunakan fungsi print().

**Contoh**

Menampilkan posisi arah pergerakan dari bidawang sebelum dan sesudah berotasi:

# Periksa arah awal

print("Arah awal:", heading())

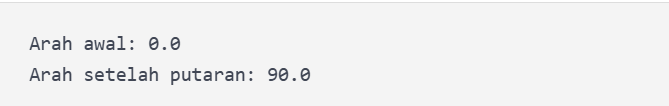
# Putar turtle ke kiri sebesar 90 derajat

left(90)

# Periksa arah setelah putaran

print("Arah setelah putaran:", heading())

Hasil:



Gambar (Hasil contoh penggunaan heading)

Fungsi heading() digunakan untuk mendapatkan arah pergerakan dari bidawang, kemudian untuk menampilkan nilainya kita bisa menggunakan fungsi print().

**Latihan**

Untuk lebih memudahkan memahami cara kerja perintah heading() ikuti instruksi di bawah ini. Kalian bisa membuka aplikasi yang sudah di siapkan untuk menjalankan kodenya, kemudian ketikkan perintah tersebut, lalu tekan tombol "Run Code" untuk melihat pergerakan Bidawang, lakukan secara bertahap:

PR

**Kesimpulan**

Perintah heading() berguna untuk memantau arah gerakan bidawang dengan lebih tepat. Pemahaman tentang sistem derajat akan membantu siswa menggambar bentuk dan pola secara lebih akurat.

**Tantangan**

PR

**Kuis**

1. Dalam canvas Bidawang, arah 0 derajat mengarah ke mana?
2. Utara (atas).
3. Timur (kanan).
4. Barat (kiri).
5. Selatan (bawah).
6. Apa hasil dari perintah berikut jika turtle sudah diputar 90 derajat ke kiri?

print(heading())

1. 0
2. 90
3. 180
4. 270

## Distance

**Tujuan Pembelajaran**

* Memahami cara kerja fungsi distance().

Fungsi distance() digunakan untuk menghitung jarak Euclidean antara posisi turtle saat ini dan titik tertentu dengan koordinat yang diberikan. Sama seperti perintah sebelumnya (*Position()*) Untuk menampilkan hasilnya kita bisa menggunakan fungsi print().

**Contoh**

Menampilkan jarak posisi bidawang ke koordinat tujuan:

# Hitung jarak dari posisi awal ke titik (100, 100)

print("Jarak ke (100, 100):")

print(distance(100,100))

# Gerakkan turtle ke titik (100, 100)

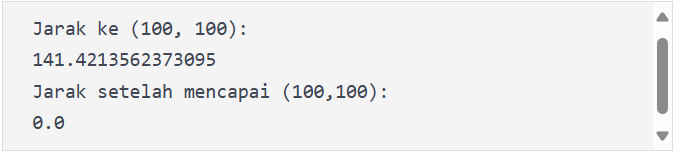
setposition(100, 100)

# Hitung jarak setelah mencapai (100, 100)

print("Jarak setelah mencapai (100,100):")

print(distance(100, 100))

Hasil:



Gambar (Hasil contoh penggunaan distance)

Fungsi distance() akan mendapatkan jarak posisi bidawang ke koordinat tujuan, kemudian untuk menampilkan nilainya kita bisa menggunakan fungsi print().

**Latihan**

Untuk lebih memudahkan memahami cara kerja perintah distance() ikuti instruksi di bawah ini. Kalian bisa membuka aplikasi yang sudah di siapkan untuk menjalankan kodenya, kemudian ketikkan perintah tersebut, lalu tekan tombol "Run Code" untuk melihat pergerakan Bidawang, lakukan secara bertahap:

PR

**Kesimpulan**

Perintah heading() berguna untuk memantau arah gerakan bidawang dengan lebih tepat. Pemahaman tentang sistem derajat akan membantu siswa menggambar bentuk dan pola secara lebih akurat.

**Tantangan**

PR

**Kuis**

1. Apa fungsi dari metode distance(x, y)?
2. Mengembalikan arah turtle menuju titik (x, y).
3. Menghitung jarak Euclidean antara posisi saat ini dan koordinat (x, y).
4. Mengatur turtle untuk bergerak ke posisi (x, y).
5. Menghapus jarak antara dua titik.
6. Jika turtle berada di koordinat (0, 0), apa hasil dari perintah berikut?

print(distance(3, 4))

1. 3
2. 4
3. 7
4. 5

# KONTROL PENA

KONTROL PENA

**4**

## Pendown & Penup

**Tujuan Pembelajaran**

* Memahami perbedaan antara mode menggambar dan mode tidak menggambar.

Fungsi distance() digunakan untuk menurunkan pena, sehingga turtle akan menggambar garis saat bergerak. Sebaliknya, metode penup() mengangkat pena, mencegah turtle menggambar saat bergerak.

**Contoh**

Mengaktifkan dan menonaktifkan mode menggambar:

# Gerakkan tanpa menggambar

penup()

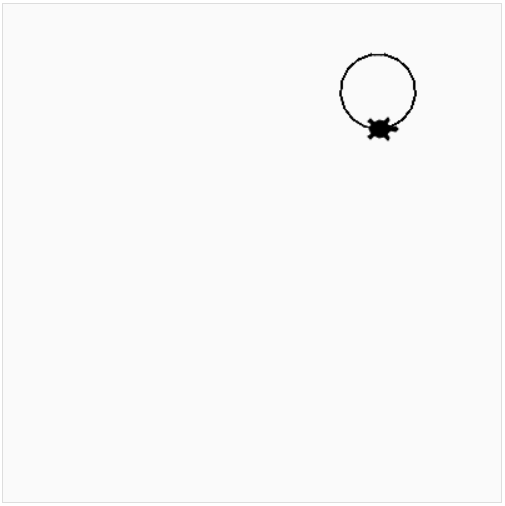
setposition(100, 100)

# Aktifkan menggambar

pendown()

circle(30)

Hasil:



Gambar (Hasil contoh penggunaan penup dan pendown)

Fungsi penup() akan menonaktifkan mode menggambar, jadi meskipun bidawang bergerak tidak akan menggambar garis. Fungsi pendown() akan mengaktifkan mode menggambar.

**Latihan**

Untuk lebih memudahkan memahami cara kerja perintah penup() dan pendown() ikuti instruksi di bawah ini. Kalian bisa membuka aplikasi yang sudah di siapkan untuk menjalankan kodenya, kemudian ketikkan perintah tersebut, lalu tekan tombol "Run Code" untuk melihat pergerakan Bidawang, lakukan secara bertahap:

PR

**Kesimpulan**

Perintah pendown() dan penup() sangat penting untuk mengontrol kapan turtle menggambar atau hanya bergerak tanpa meninggalkan jejak. Penggunaan yang tepat membantu dalam pembuatan bentuk yang terpisah.

**Tantangan**

PR

**Kuis**

1. Apa yang terjadi jika metode pendown() tidak dipanggil setelah penup()?
2. Bidawang akan terus menggambar saat bergerak.
3. Bidawang akan berhenti bergerak.
4. Bidawang tidak akan menggambar garis saat bergerak.
5. Bidawang akan menggambar lingkaran secara otomatis.
6. Perhatikan kode berikut:

penup()

goto(100, 100)

pendown()

43

goto(200, 200)

Apa yang terjadi setelah kode tersebut dijalankan?

1. Bidawang menggambar garis dari titik awal ke (100, 100).
2. Bidawang menggambar garis dari (100, 100) ke (200, 200).
3. Bidawang tidak menggambar sama sekali.
4. Bidawang hanya menggambar lingkaran.

## Pensize

**Tujuan Pembelajaran**

* Memahami cara mengubah ketebalan garis menggunakan perintah pensize().

Fungsi pensize(width) digunakan untuk mengatur ketebalan pena saat menggambar garis. Parameter width menunjukkan ukuran ketebalan dalam piksel. Dengan memodifikasi ketebalan garis, pengguna dapat menciptakan variasi visual, seperti membuat garis tebal untuk bingkai atau garis tipis untuk detail halus.

* pensize(width): Mengatur ketebalan garis menjadi width (piksel).

**Contoh**

Menggambar lingkaran dengan garis tebal:

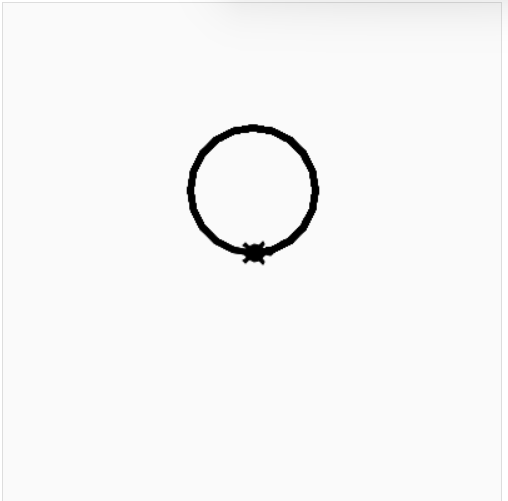
# Ubah ketebalan pena

pensize(5)

# Gambar lingkaran dengan garis tebal

circle(50)

Hasil:



Gambar (Hasil contoh penggunaan pensize

Fungsi pensize(5) akan membuat ketebalan garis menjadi 5 pixel saat bidawang menggambar lingkaran dengan fungsi circle().

**Latihan**

Untuk lebih memudahkan memahami cara kerja perintah pensize() ikuti instruksi di bawah ini. Kalian bisa membuka aplikasi yang sudah di siapkan untuk menjalankan kodenya, kemudian ketikkan perintah tersebut, lalu tekan tombol "Run Code" untuk melihat pergerakan Bidawang, lakukan secara bertahap:

PR

**Kesimpulan**

Dengan perintah pensize(), ketebalan garis dapat disesuaikan untuk meningkatkan estetika atau menciptakan variasi visual dalam gambar.

**Tantangan**

PR

**Kuis**

1. Apa efek dari penggunaan metode pensize(10) sebelum menggambar?
2. Mengganti warna garis menjadi hitam.
3. Mengubah ketebalan garis menjadi 10 piksel.
4. Membuat garis tidak terlihat.
5. Menghapus garis yang telah digambar.
6. Jika kode berikut dijalankan:

pensize(5)

circle(30)

pensize(2)

circle(50)

Apa yang akan terlihat pada hasil akhir?

1. Dua lingkaran dengan ketebalan garis yang sama.
2. Lingkaran pertama memiliki garis lebih tebal daripada lingkaran kedua.
3. Lingkaran kedua memiliki garis lebih tebal daripada lingkaran pertama.
4. Tidak ada lingkaran yang digambar.

## Isdown

**Tujuan Pembelajaran**

* Memahami cara mengecek status pena apakah sedang aktif menggambar atau tidak.

Fungsi isdown() digunakan untuk memeriksa status pena. Perintah ini menyimpan nilai True jika pena sedang dalam posisi turun (aktif menggambar) dan False jika pena diangkat. Pemeriksaan status pena berguna untuk menentukan apakah Bidawang akan meninggalkan jejak atau tidak saat bergerak.

**Contoh**

Mengecek status pena sedang aktif atau tidak:

# Cek status pena

print("Status Pena saat ini:")

print(isdown())

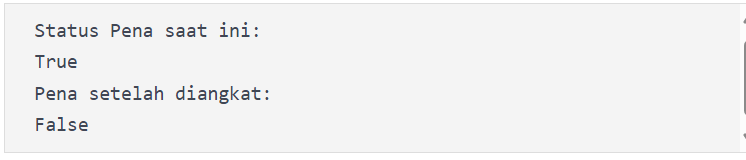
# Mengangkat pena

penup()

print("Pena setelah diangkat:")

print(isdown())

Hasil:



Gambar (Hasil contoh penggunan isdown)

Fungsi isdown() akan menampilkan true jika pena sedang aktif, kemudian ketika pena di angkat dengan fungsi penup() maka menampilkan false.

**Latihan**

Untuk lebih memudahkan memahami cara kerja perintah isdown() ikuti instruksi di bawah ini. Kalian bisa membuka aplikasi yang sudah di siapkan untuk menjalankan kodenya, kemudian ketikkan perintah tersebut, lalu tekan tombol "Run Code" untuk melihat pergerakan Bidawang, lakukan secara bertahap:

PR

**Kesimpulan**

Perintah isdown(), berguna untuk mengecek status pena secara dinamis dan membantu mengontrol logika penggambaran.

**Tantangan**

PR

**Kuis**

1. Apa yang dikembalikan oleh metode isdown() jika pena sedang dalam posisi turun?
2. True
3. False
4. None
5. Error
6. Perhatikan kode berikut:pendown()

print(isdown())

penup()

print(isdown())

Apa output dari kode tersebut?

1. True, False
2. False, True
3. True, True
4. False, False

# KONTROL WARNA

KONTROL WARNA

**5**

## Pencolor

**Tujuan Pembelajaran**

* Memahami cara mengubah warna garis pada saat menggambar.

Fungsi pencolor("*warna*") fungsi yang digunakan untuk mengontrol warna garis pada gambar yang dihasilkan oleh bidawang. Parameter warna bisa berupa nama warna (contoh: "red") atau kode warna RGB (contoh: "FF0000").

**Contoh**

Membuat persegi dengan garis biru:

# Mengatur warna garis menjadi biru

pencolor("blue")

# Membuat persegi

forward(100)

left(90)

forward(100)

left(90)

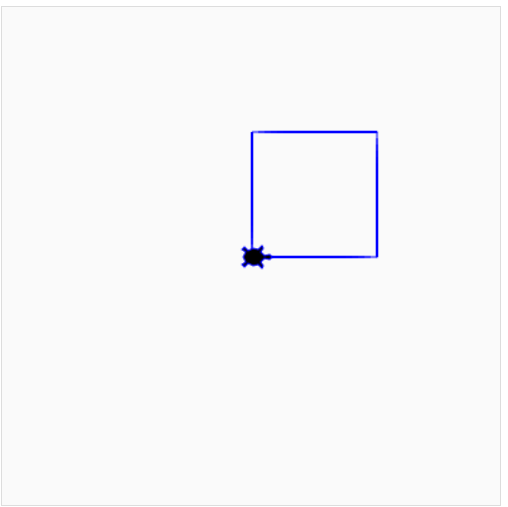
forward(100)

left(90)

forward(100)

left(90)

Hasil:



Gambar (Hasil contoh penggunaan pencolor)

Fungsi pencolor("blue") akan membuat warna pen jadi biru, jadi ketika bidawang bergerak menggambar persegi maka warna dari garis-garis yang membentuk persegi akan berwarna biru.

**Latihan**

Untuk lebih memudahkan memahami cara kerja perintah pencolor() ikuti instruksi di bawah ini. Kalian bisa membuka aplikasi yang sudah di siapkan untuk menjalankan kodenya, kemudian ketikkan perintah tersebut, lalu tekan tombol "Run Code" untuk melihat pergerakan Bidawang, lakukan secara bertahap:

PR

**Kesimpulan**

Perintah pencolor() digunakan untuk mengatur warna garis lintasan yang dilalui oleh bidawang.

**Tantangan**

PR

**Kuis**

1. Apa yang terjadi jika kode berikut dijalankan?

pencolor("red")

forward(100)

1. Bidawang menggambar garis merah sepanjang 100 piksel.
2. Bidawang menggambar garis hitam sepanjang 100 piksel.
3. Bidawang menggambar garis tanpa warna sepanjang 100 piksel.
4. Bidawang tidak menggambar garis.
5. Apa format yang tidak valid untuk parameter warna dalam fungsi pencolor()?
6. "blue"
7. "#00FF00"
8. (1,0,0)
9. "bold\_red"

## Fillcolor, Begin\_fill, & End\_fill

**Tujuan Pembelajaran**

* Memahami cara kerja fungsi fillcolor, begin\_fill, dan end\_fill dalam mengatur warna isian pada bentuk geometris.

Perintah fillcolor("*warna*") Digunakan untuk menentukan warna isian yang akan diterapkan pada bentuk. Parameter warna dapat berupa nama warna atau kode RGB. Untuk mewarnai bentuk, perintah fillcolor("*warna*") harus di sertai dengan begin\_fill() dan end\_fill()

* begin\_fill(): Menandai awal area yang akan diisi warna.
* end\_fill(): Menandai akhir area yang akan diisi warna. Warna isian diterapkan sesuai pengaturan fillcolor.

**Contoh**

Membuat lingkaran berwarna hijau:

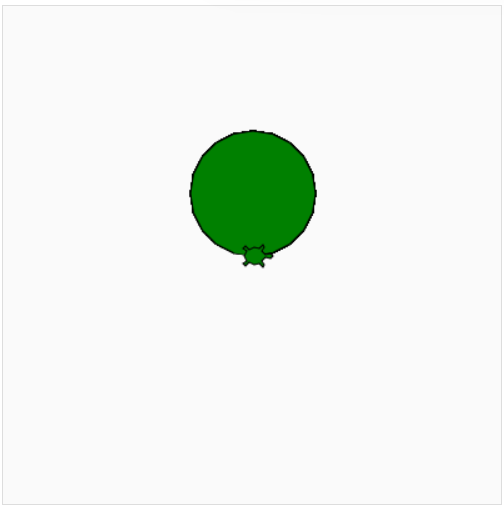
fillcolor("green") # Warna isian hijau

begin\_fill()

circle(50) # Membuat lingkaran

end\_fill()

Hasil:



Gambar (Hasil contoh penggunan fillcolor, begin\_fill, dan end\_fill)

Bidawang akan menggambar lingkaran, kemudian fungsi fillcolor("green") akan membuat warna lingkaran tersebut menjadi hijau.

**Latihan**

Untuk lebih memudahkan memahami cara kerja perintah fillcolor(), begin\_fill(), dan end\_fill() ikuti instruksi di bawah ini. Kalian bisa membuka aplikasi yang sudah di siapkan untuk menjalankan kodenya, kemudian ketikkan perintah tersebut, lalu tekan tombol "Run Code" untuk melihat pergerakan Bidawang, lakukan secara bertahap:

PR

**Kesimpulan**

Perintah fillcolor() digunakan untuk mengatur warna isian, sedangkan proses pengisian warna dilakukan dengan bantuan begin\_fill() dan end\_fill(). Untuk hasil optimal, pastikan area yang digambar adalah bentuk tertutup.

**Tantangan**

PR

**Kuis**

1. Perhatikan kode berikut:

forward(100)

begin\_fill()

circle(50)

end\_fill()

Apa hasil dari kode tersebut?

1. Sebuah lingkaran dengan warna garis kuning tanpa isian.
2. Sebuah lingkaran dengan warna isian kuning.
3. Sebuah lingkaran dengan warna garis hitam dan isian kuning.
4. Tidak ada lingkaran yang digambar.
5. Apa fungsi dari end\_fill() dalam proses pengisian warna?
6. Mengatur warna isian menjadi transparan.
7. Menandai akhir area yang akan diisi warna.
8. Menonaktifkan pengaturan warna pada turtle.
9. Menghapus warna isian dari bentuk yang digambar.

# KONTROL GAMBAR LANJUTAN

KONTROL GAMBAR LANJUTAN

**6**

## Reset

**Tujuan Pembelajaran**

* Memahami fungsi reset().

Perintah reset() Digunakan untuk menghapus semua gambar pada canvas dan mengatur ulang posisi serta atribut seperti warna dan ukuran pena, ke nilai default.

**Contoh**

Menghapus gambar yang sudah dibuat dan mengatur ulang atribut serta posisi:

pencolor("red")

forward(100)

left(90)

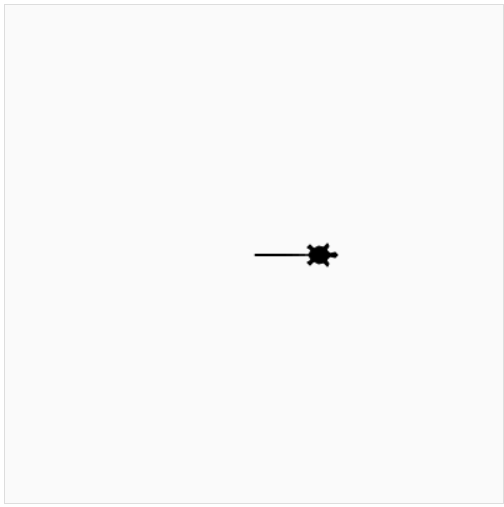
forward(100)

# Reset layar dan posisi turtle

reset()

forward(50) # Memulai gambar baru

Hasil:



Gambar (Hasil contoh penggunaan reset)

Bidawang akan menggambar dengan garis merah, kemudian setelah perintah reset() semua gambar akan hilang dan posisi serta atribut bidawang akan kembali ke nilai default.

**Latihan**

Untuk lebih memudahkan memahami cara kerja perintah reset() ikuti instruksi di bawah ini. Kalian bisa membuka aplikasi yang sudah di siapkan untuk menjalankan kodenya, kemudian ketikkan perintah tersebut, lalu tekan tombol "Run Code" untuk melihat pergerakan Bidawang, lakukan secara bertahap:

PR

**Kesimpulan**

Perintah reset() berguna untuk memulai ulang proses menggambar tanpa harus menghapus semua kode yang sudah ditulis.

**Tantangan**

PR

**Kuis**

Top of Form

1. Apa yang terjadi jika fungsi reset() dipanggil setelah menggambar garis?
2. Semua gambar dihapus, tetapi atribut bidawang tetap sama.
3. Semua gambar dihapus, dan bidawang kembali ke posisi awal dengan atribut default.
4. Garis tidak dihapus, tetapi posisi bidawang berubah.
5. Bidawang akan keluar dari jendela.
6. Perhatikan kode berikut:

pencolor("green")

forward(100)

reset()

circle(50)

Apa warna lingkaran yang akan dihasilkan?

1. Hijau.
2. Hitam.
3. Biru.
4. Merah.

**Bottom of Form**

## Clear

**Tujuan Pembelajaran**

* Memahami fungsi reset().

Perintah clear() digunakan untuk menghapus semua gambar pada canvas tanpa mengatur ulang posisi atau atribut bidawang.

**Contoh**

Menghapus gambar yang sudah dibuat tanpa mengatur ulang atribut dan posisi:

pencolor("blue")

forward(50)

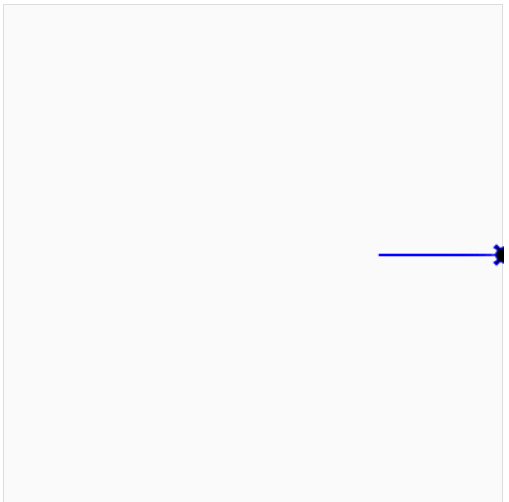
circle(50)

# Menghapus gambar tetapi posisi dan atribut tetap sama

clear()

forward(100) # Memulai gambar baru dari posisi terakhir

Hasil:



Gambar (Hasil contoh penggunaan clear)

Bidawang akan menggambar lingkaran dengan garis biru, Setelah di clear() atribut pencolor “blue” akan tetap aktif.

**Latihan**

Untuk lebih memudahkan memahami cara kerja perintah clear() ikuti instruksi di bawah ini. Kalian bisa membuka aplikasi yang sudah di siapkan untuk menjalankan kodenya, kemudian ketikkan perintah tersebut, lalu tekan tombol "Run Code" untuk melihat pergerakan Bidawang, lakukan secara bertahap:

PR

**Kesimpulan**

Perintah clear() berguna untuk menghapus gambar tanpa mengubah pengaturan posisi atau atribut bidawang.

**Tantangan**

PR

**Kuis**

1. Apa perbedaan utama antara fungsi clear() dan reset()?
2. clear() menghapus gambar tanpa mengubah posisi atau atribut, sedangkan reset() juga mengatur ulang posisi dan atribut bidawang.
3. clear() menghapus gambar beserta posisi bidawang, sedangkan reset() hanya menghapus gambar.
4. clear() tidak menghapus gambar, sedangkan reset() menghapus gambar.
5. clear() menutup jendela, sedangkan reset() tidak.
6. Perhatikan kode berikut:

color("blue")

circle(50)

clear()

forward(100)

circle(30)

Apa warna lingkaran kedua?

1. Tidak Berwarna.
2. Biru.
3. Hitam.
4. Merah.

## Write

**Tujuan Pembelajaran**

* Memahami fungsi write() untuk menampilkan teks pada canvas.

Perintah write() digunakan untuk menulis teks pada canvas di lokasi bidawang berada. write(teks, move=False, align="left", font=("Arial", 8, "normal")):

* Parameter teks: Teks yang akan ditulis.
* Parameter move: Jika True, posisi turtle akan berpindah setelah menulis teks.
* Parameter align: Penyesuaian teks ("left", "center", atau "right").
* Parameter font: Menentukan jenis, ukuran, dan gaya font.

**Contoh**

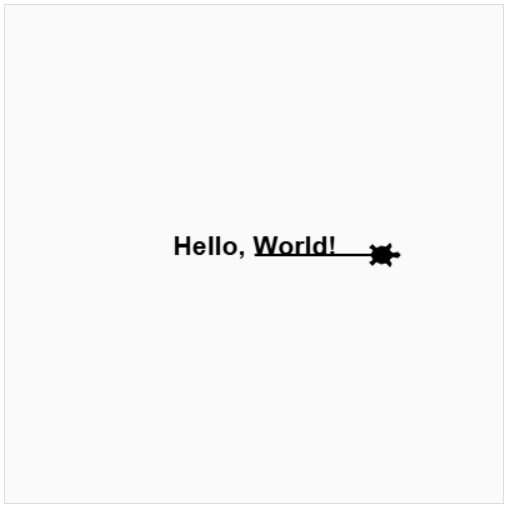
Menulis "Hello, World!" di canvas:

write("Hello, World!", align="center", font=("Arial", 16, "bold"))

# Bidawang tetap dapat menggambar setelah menulis

forward(100)

Hasil:



Gambar (Hasil contoh penggunaan write)

**Latihan**

Untuk lebih memudahkan memahami cara kerja perintah write() ikuti instruksi di bawah ini. Kalian bisa membuka aplikasi yang sudah di siapkan untuk menjalankan kodenya, kemudian ketikkan perintah tersebut, lalu tekan tombol "Run Code" untuk melihat pergerakan Bidawang, lakukan secara bertahap:

PR

**Kesimpulan**

Perintah write() memungkinkan untuk menampilkan teks di layar, sehingga dapat digunakan untuk memberikan informasi tambahan pada gambar.

**Tantangan**

PR

**Kuis**

Top of Form

1. Apa fungsi dari parameter align dalam metode write()?
2. Mengatur posisi turtle setelah menulis teks.
3. Menentukan jenis font yang digunakan.
4. Mengatur perataan teks (kiri, tengah, atau kanan).
5. Mengatur warna teks.
6. Perhatikan kode berikut:

write("Belajar Python!", align="center", font=("Arial", 12, "italic"))

Apa yang akan terjadi?

1. Teks ditulis di layar dengan font Arial, ukuran 12, dan bergaya italic di posisi kiri bidawang.
2. Teks ditulis di layar dengan font Arial, ukuran 12, dan bergaya italic di posisi tengah bidawang.
3. Teks ditulis di layar dengan font Arial, ukuran 12, tetapi tidak bergaya italic.
4. Tidak ada teks yang ditulis karena font tidak valid.

**Bottom of Form**