



LIBRARY TURTLE



Modul Pembelajaran Untuk Pemrograman Logo

Tim Penyusun:

Dr. Harja Santana Purba, M.Kom.

Rizky Pamuji, S.Kom., M.Kom.

Khoyrur Roykhan



DAFTAR ISI

DAFTAR ISI	2
DAFTAR GAMBAR	3
1. Pendahuluan	5
1.1 Pemrograman Logo	5
1.2 Canvas	6
2. Turtle Motion	7
2.1 Left & Right	7
2.2 Forward & Backward	10
2.3 Setposition	14
2.4 Setx & sety	17
2.5 Setheading	21
2.6 Home	23
2.7 Circle	25
2.8 Dot	28
3. Turtle Tell State	31
3.1 Position	31
3.2 Xcor & ycor	33
3.3 Heading	35
3.4 Distance	37
4. Pen Control (Drawing State)	40
4.1 Pendown & penup	40
4.2 Pensize	42
4.3 Isdown	44
5. Color Control	46
5.1 Color & pencolor	46
5.2 Pengisian Warna (Fillcolor, Begin_fill, dan End_fill)	48
6. More Drawing Control	51
6.1 Reset	51
6.2 Clear	53
6.3 Write	55

DAFTAR GAMBAR

Gambar 1 (Hasil contoh penggunaan left)	8
Gambar 2 (Hasil contoh penggunaan right)	8
Gambar 3 (Hasil contoh penggunaan left dan right)	9
Gambar 4 (Hasil contoh penggunaan forward)	11
Gambar 5 (Hasil contoh penggunaan backward)	12
Gambar 6 (Hasil contoh penggunaan forward dan backward)	13
Gambar 7 (Hasil contoh penggunaan setposition)	15
Gambar 8 (Hasil contoh penggunaan setposition membentuk pola)	16
Gambar 9 (Hasil contoh penggunaan setx)	18
Gambar 10 (Hasil contoh penggunaan sety)	19
Gambar 11 (Hasil Contoh penggunaan setx dan sety)	20
Gambar 12 (Hasil contoh penggunaan setheading)	22
Gambar 13 (Hasil contoh penggunaan home)	24
Gambar 14 (Hasil contoh 1 penggunaan circle)	26
Gambar 15 (Hasil contoh 2 penggunaan circle)	27
Gambar 16 (Hasil contoh penggunaan dot)	29
Gambar 17 (Hasil contoh 1 penggunaan position)	31
Gambar 18 (Hasil contoh 2 penggunaan position)	32
Gambar 19 (Hasil contoh penggunaan xcor dan ycor)	34
Gambar 20 (Hasil contoh penggunaan heading)	36
Gambar 21 (Hasil contoh penggunaan distance)	38
Gambar 22 (Hasil contoh penggunaan penup)	41
Gambar 23 (Hasil contoh penggunaan pensize)	43
Gambar 24 (Hasil contoh penggunaan isdown)	45
Gambar 25 (Hasil contoh penggunaan pencolor)	47
Gambar 26 (Hasil contoh penggunaan Fillcolor, Begin_fill, dan End_fill)	49
Gambar 27 (Hasil dari contoh penggunaan reset)	52

Gambar 28 (Hasil contoh penggunaan clear).....	54
Gambar 29 (Hasil contoh penggunaan write)	56

Dalam dunia yang semakin maju dengan teknologi, pemrograman telah menjadi salah satu keterampilan penting yang harus dipelajari sejak dini. Salah satu cara yang menyenangkan untuk memulai belajar pemrograman adalah dengan menggunakan Python, sebuah bahasa pemrograman yang sederhana namun sangat kuat. Untuk memperkenalkan konsep dasar pemrograman dengan cara yang interaktif dan visual, library Turtle adalah pilihan yang tepat. Library Turtle menyediakan cara mudah untuk menggambar bentuk dan pola pada layar menggunakan perintah-perintah sederhana. Library ini menggunakan metafora kura-kura yang dapat bergerak di atas "canvas" atau kanvas, meninggalkan jejak di belakangnya seperti pena yang menggambar di atas kertas.

Modul ini dirancang untuk siswa tingkat Sekolah Menengah Pertama (SMP) agar mereka dapat belajar pemrograman sambil mengembangkan keterampilan berpikir logis dan kreatif. Dalam modul ini, kita akan mempelajari konsep dasar dari library Turtle dan bagaimana menggunakannya untuk membuat berbagai bentuk serta pola menarik.

1.1 Pemrograman Logo

Pemrograman Logo adalah salah satu pendekatan klasik yang digunakan untuk mengajarkan dasar-dasar pemrograman kepada pemula. Konsep ini pertama kali diperkenalkan pada akhir 1960-an dan menggunakan metafora kura-kura yang bergerak di atas layar untuk menggambar bentuk-bentuk geometris.

Logo menjadi inspirasi utama di balik library Turtle dalam Python. Seperti halnya Logo, Turtle memungkinkan kita untuk mengontrol pergerakan kura-kura virtual menggunakan perintah-perintah sederhana seperti "forward", "backward", "right" dan "left" untuk menciptakan berbagai gambar dan pola. Dengan menggunakan Turtle, siswa dapat

memahami logika pemrograman melalui aktivitas yang menyenangkan dan langsung terlihat hasilnya.

1.2 Canvas

Canvas adalah area gambar di mana kura-kura menggambar pola dan bentuk sesuai perintah yang diberikan. Dalam library Turtle, canvas adalah jendela grafis yang akan muncul saat program dieksekusi. Kura-kura dapat bergerak di dalam canvas untuk menggambar garis dan bentuk sesuai instruksi yang kita berikan.

Secara default, ukuran canvas dalam library Turtle adalah 400 piksel x 400 piksel. Posisi awal kura-kura berada di tengah canvas dengan koordinat (0, 0), di mana sumbu x positif mengarah ke kanan, dan sumbu y positif mengarah ke atas. Kura-kura memiliki ukuran awal standar yang dapat disesuaikan dengan perintah `turtlesize()`. Warna latar belakang canvas dapat diubah menggunakan fungsi `bgcolor()`, dan kita dapat menyesuaikan warna garis gambar dengan perintah `pencolor()`. Selain itu, kecepatan gerakan kura-kura dapat diatur menggunakan fungsi `speed()`, yang memungkinkan kita untuk mempercepat atau memperlambat pergerakan saat menggambar.

Dengan dasar pemahaman tentang pemrograman Logo dan canvas, siswa akan siap untuk memulai perjalanan mereka dalam belajar library Turtle, yang akan membantu membangun keterampilan pemrograman yang berguna di masa depan.

2.1 Left & Right

A. Tujuan Pembelajaran

- Memahami cara mengendalikan arah gerakan turtle menggunakan *left()* dan *right()*.
- Memahami rotasi kiri dan kanan pada turtle.

B. Definisi Turtle *left()* dan *right()*

Dalam pustaka Turtle Graphics, metode *left()* dan *right()* digunakan untuk memutar arah gerakan turtle berdasarkan sudut derajat yang diberikan, tanpa harus memindahkan posisinya. Ini berguna untuk mengatur arah turtle sebelum melanjutkan dengan perintah lainnya, seperti menggambar atau bergerak.

C. Fungsi

a) Left

Memutar arah turtle berlawanan arah jarum jam (kiri) sebesar derajat yang ditentukan.

Contoh:

```
# Putar turtle ke kiri sejauh 120 derajat  
left(120)
```

Hasil:

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



Gambar 1 (Hasil contoh penggunaan left)

b) Right

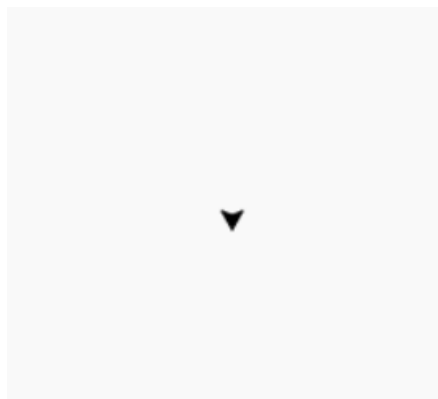
Memutar arah turtle searah jarum jam (kanan) sebesar derajat yang ditentukan.

Contoh:

```
# Putar turtle ke kanan sejauh 90 derajat  
right(90)
```

Hasil:

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



Gambar 2 (Hasil contoh penggunaan right)

D. Contoh Penggunaan Gabungan

Kita bisa mengombinasikan kedua perintah ini untuk membuat pola rotasi tanpa menggerakkan turtle maju. Misalnya, jika ingin memutar turtle ke kiri 90 derajat lalu kembali ke posisi semula dengan memutar ke kanan:

```
# Putar ke kiri sejauh 90 derajat
left(90)

# Putar ke kanan sejauh 90 derajat
right(90)
```

Hasil:

Objek akan berputar 90 derajat ke kiri, kemudian berputar lagi 90 derajat ke kanan. Hasilnya akan kembali ke posisi awal.



Gambar 3 (Hasil contoh penggunaan left dan right)

Meskipun `left()` dan `right()` biasanya dikombinasikan dengan perintah lain seperti `forward()` untuk menggambar, perintah ini juga bisa digunakan secara mandiri untuk mengatur sudut atau arah dari turtle sesuai dengan kebutuhan.

E. Kesimpulan

Perintah `'left()'` dan `'right()'` dalam pustaka Turtle Graphics memungkinkan pengaturan arah gerakan turtle 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.

F. Kuis

- 1) Apa perbedaan utama antara perintah `left()` dan `right()` dalam pustaka Turtle Graphics?
 - a. `left()` memutar turtle ke arah kanan, sementara `right()` memutar turtle ke arah kiri.
 - b. `left()` memutar turtle ke arah kiri, sementara `right()` memutar turtle ke arah kanan.
 - c. `left()` dan `right()` hanya digunakan untuk mengubah warna turtle.
 - d. Keduanya memindahkan turtle ke posisi (0, 0).
- 2) Jika turtle menghadap ke timur, dan Anda menggunakan perintah `right(90)`, ke arah mana turtle akan menghadap?
 - a. Utara
 - b. Barat
 - c. Selatan
 - d. Timur

2.2 Forward & Backward

A. Tujuan Pembelajaran

- Memahami cara menggerakkan turtle ke depan dan ke belakang menggunakan `'forward()'` dan `'backward()'`.

- Memahami konsep gerakan linear pada turtle.

B. Definisi

Dalam pustaka Turtle Graphics, metode `'forward()'` dan `'backward()'` digunakan untuk menggerakkan turtle dalam arah tertentu, berdasarkan jarak yang ditentukan dalam satuan piksel (pixel). Gerakan ini dilakukan ke depan (searah arah yang sedang dihadapi turtle) atau ke belakang (berlawanan arah dengan yang sedang dihadapi turtle).

C. Fungsi

a) Forward

Menggerakkan turtle ke depan sejauh jarak yang ditentukan (dalam piksel), dalam arah yang sedang dihadapi oleh turtle.

Contoh:

```
# Gerakkan turtle ke depan sejauh 100 piksel  
forward(100)
```

Hasil:

Objek akan bergerak sejauh 100 ke arah yang dihadapnya.



Gambar 4 (Hasil contoh penggunaan forward)

b) Backward

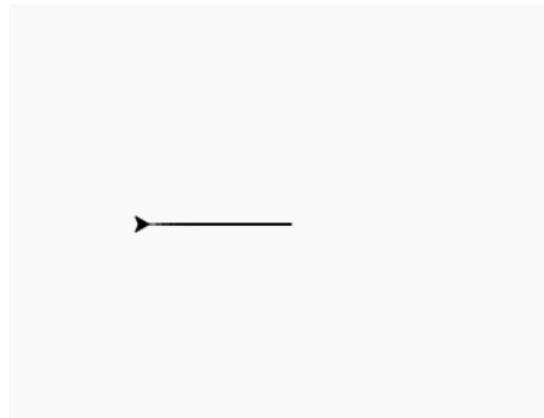
Menggerakkan turtle ke belakang sejauh jarak yang ditentukan (dalam piksel), dalam arah berlawanan dengan arah yang sedang dihadapi oleh turtle.

Contoh:

```
# Gerakkan turtle ke belakang sejauh 50 piksel  
backward(100)
```

Hasil:

Objek akan mundur sejauh 100 dari arah yang dihadapnya.



Gambar 5 (Hasil contoh penggunaan backward)

D. Contoh Penggunaan Gabungan

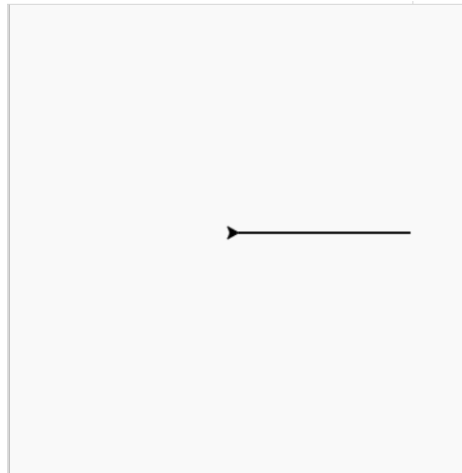
Anda bisa mengombinasikan kedua perintah ini untuk membuat gerakan maju-mundur dengan mudah. Misalnya, untuk membuat turtle bergerak ke depan sejauh 150 piksel, lalu mundur kembali ke posisi awal:

```
# Gerakkan turtle ke depan sejauh 150 piksel  
forward(150)  
  
# Kembali ke posisi awal dengan mundur sejauh 150  
piksel
```

```
backward(150)
```

Hasil:

Objek akan maju sejauh 150 kemudian mundur sejauh 150. Hasilnya objek akan kembali ke posisi (0,0)



Gambar 6 (Hasil contoh penggunaan forward dan backward)

Perintah `'forward()'` dan `'backward()'` sering dikombinasikan dengan perintah rotasi seperti `'left()'` dan `'right()'` untuk membuat pola atau gambar yang lebih kompleks.

E. Kesimpulan

Perintah `'forward()'` dan `'backward()'` dalam pustaka Turtle Graphics digunakan untuk menggerakkan turtle ke depan atau ke belakang sejauh jarak yang ditentukan dalam piksel. Kedua perintah ini sangat berguna untuk membentuk gambar dengan gerakan linear sebelum mengatur arah baru menggunakan perintah rotasi.

F. Kuis

- 1) Perintah apa yang digunakan untuk menggerakkan turtle ke depan sejauh 150 piksel?
 - a. `backward(150)`

- b. `forward(150)`
 - c. `left(150)`
 - d. `right(150)`
- 2) Jika turtle menghadap ke barat dan Anda menggunakan perintah `backward(100)`, ke arah mana turtle akan bergerak?
- a. Ke barat
 - b. Ke timur
 - c. Ke utara
 - d. Ke selatan

2.3 Setposition

A. Tujuan Pembelajaran

- Memahami cara mengatur posisi turtle secara langsung menggunakan ``setposition()`` atau ``setpos()``.
- Memahami koordinat kartesius dalam ruang gambar Turtle Graphics.

B. Definisi

Dalam pustaka Turtle Graphics, metode ``setposition()`` (bisa juga disingkat ``setpos()``) digunakan untuk memindahkan turtle ke titik tertentu dalam ruang gambar, berdasarkan koordinat yang ditentukan. Saat perintah ini digunakan, turtle akan menggambar jalur dari posisi sebelumnya ke posisi baru.

C. Fungsi

``setposition(x, y)`` atau ``setpos(x, y)``: Memindahkan turtle ke koordinat ``(x, y)`` yang ditentukan dalam ruang gambar. Jalur yang digambar tergantung pada apakah pena (`pen`) sedang aktif atau tidak.

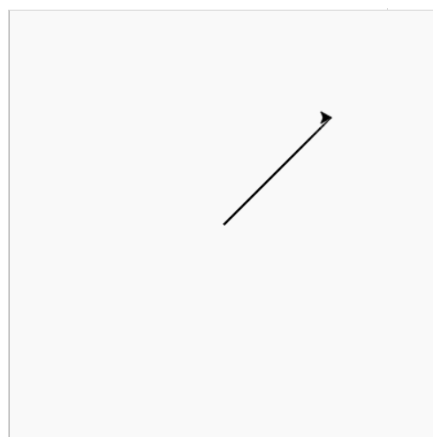
Jika pena aktif, jalur akan tergambar; jika tidak, hanya turtle yang berpindah.

Contoh:

```
# Pindahkan turtle ke koordinat (100, 100)
setposition(100, 100)
```

Hasil:

Objek akan bergerak ke posisi x=100 dan y=100.



Gambar 7 (Hasil contoh penggunaan setposition)

D. Contoh Penggunaan Setposition untuk Membentuk Pola

Kita dapat menggunakan `setposition()` untuk menggambar pola dengan mengatur beberapa titik koordinat. Misalnya, membuat segitiga dengan menentukan tiga titik:

```
# Pindahkan turtle ke titik pertama
setposition(100, 0)

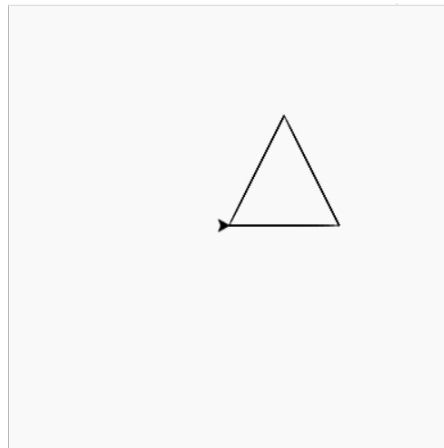
# Pindahkan ke titik kedua
setposition(50, 100)

# Kembali ke titik awal
```

```
setposition(0, 0)
```

Hasil:

Objek akan berpindah ke titik $x=100, y=0$. Kemudian setelah mencapai titik itu objek akan pindah ke titik kedua yaitu $x=50, y=100$. Dan setelah mencapai titik kedua, objek akan berpindah lagi ke titik awal yaitu $x=0, y=0$.



Gambar 8 (Hasil contoh penggunaan setposition membentuk pola)

E. Kesimpulan

Perintah ``setposition()`` atau ``setpos()`` dalam pustaka Turtle Graphics memungkinkan Anda memindahkan turtle ke koordinat tertentu dalam ruang gambar. Perintah ini sangat berguna untuk mengatur posisi awal turtle atau membuat jalur menggambar yang kompleks dengan mengombinasikan pergerakan dan kontrol pena.

F. Kuis

- 1) Apa fungsi utama dari perintah `setposition(x, y)` dalam pustaka Turtle Graphics?
 - a. Mengatur warna garis.

- b. Memutar turtle ke arah tertentu.
 - c. Memindahkan turtle ke koordinat tertentu dalam ruang gambar.
 - d. Mengubah ukuran turtle.
- 2) Jika posisi awal turtle adalah (0, 0), dan Anda menggunakan perintah `setposition(50, 100)`, di mana posisi akhir turtle?
- a. (100, 50)
 - b. (50, 0)
 - c. (0, 100)
 - d. (50, 100)

2.4 Setx & sety

A. Tujuan Pembelajaran

- Memahami cara mengatur posisi turtle pada sumbu x dan y secara terpisah menggunakan `setx()` dan `sety()`.
- Memahami koordinat kartesius dalam ruang gambar Turtle Graphics.

B. Definisi

Dalam pustaka Turtle Graphics, metode `setx()` dan `sety()` digunakan untuk memindahkan turtle ke posisi tertentu hanya pada sumbu x atau sumbu y, tanpa mengubah koordinat lainnya. Perintah ini berguna untuk memindahkan turtle secara horizontal atau vertikal.

C. Fungsi

a) Setx

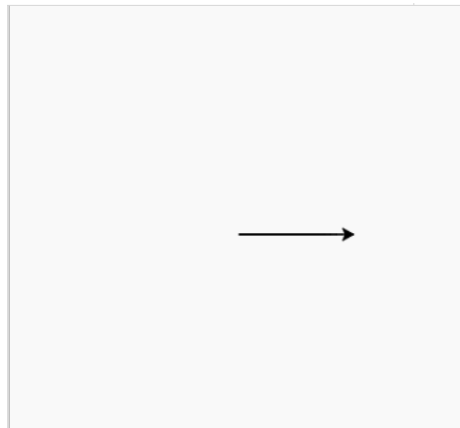
Memindahkan turtle ke posisi x yang ditentukan, tanpa mengubah posisi pada sumbu y. Jika pena (pen) aktif, turtle akan menggambar jalur dari posisi sebelumnya ke posisi baru.

Contoh:

```
# Pindahkan turtle ke posisi x = 100, tanpa  
mengubah posisi y  
setx(100)
```

Hasil:

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



Gambar 9 (Hasil contoh penggunaan setx)

b) Sety

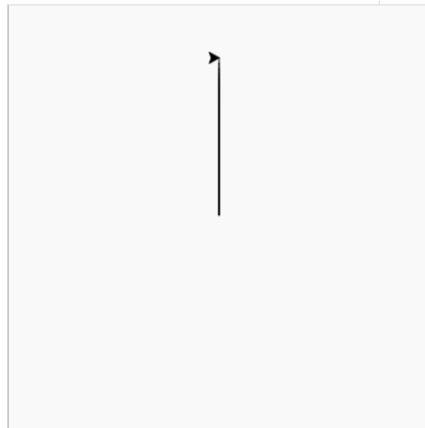
Memindahkan turtle ke posisi y yang ditentukan, tanpa mengubah posisi pada sumbu x . Jika pena aktif, jalur akan tergambar.

Contoh:

```
# Pindahkan turtle ke posisi y = 150, tanpa  
mengubah posisi x  
sety(150)
```

Hasil:

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



Gambar 10 (Hasil contoh penggunaan sety)

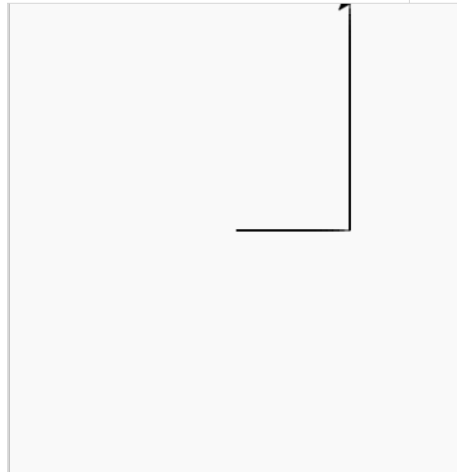
D. Contoh Penggunaan Gabungan

Anda dapat menggunakan kedua perintah ini untuk memindahkan turtle ke posisi tertentu dengan mengatur koordinat x dan y secara terpisah. Misalnya, untuk memindahkan turtle secara horizontal lalu vertikal:

Contoh:

```
# Pindahkan turtle ke posisi x = 100  
setx(100)  
  
# Pindahkan turtle ke posisi y = 200  
sety(200)
```

Hasil:



Gambar 11 (Hasil Contoh penggunaan setx dan sety)

E. Kesimpulan

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

F. Kuis

- 1) Apa hasil dari perintah `setx(200)` jika posisi awal turtle adalah (100, 50)?
 - a. Posisi baru menjadi (200, 50).
 - b. Posisi baru menjadi (100, 200).
 - c. Posisi baru menjadi (200, 200).
 - d. Posisi baru tetap (100, 50).
- 2) Jika Anda ingin memindahkan turtle secara vertikal ke posisi $y = -50$ tanpa mengubah posisi x , perintah apa yang harus digunakan?
 - a. `sety(-50)`
 - b. `setx(-50)`

- c. `setposition(-50, 0)`
- d. `home()`

2.5 Setheading

A. Tujuan Pembelajaran

- Memahami cara mengatur arah turtle menggunakan ``setheading()``.
- Memahami bagaimana sudut arah mempengaruhi gerakan turtle.

B. Definisi

Dalam pustaka Turtle Graphics, metode ``setheading()`` digunakan untuk mengatur arah turtle ke sudut tertentu, diukur dalam derajat. Sudut dihitung searah jarum jam, dimulai dari arah ke timur (0 derajat). Dengan menggunakan ``setheading()``, Anda dapat mengontrol ke mana turtle akan menghadap sebelum bergerak.

C. Fungsi

``setheading(sudut)``: Mengatur arah turtle berdasarkan sudut yang ditentukan dalam derajat. Sudut diukur dari arah ke timur (0 derajat) searah jarum jam:

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

Contoh:

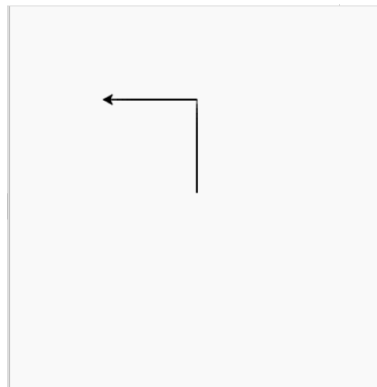
```
# Atur arah turtle menghadap utara (90 derajat)
setheading(90)
forward(100) # Bergerak ke atas

# Atur arah turtle menghadap barat (180 derajat)
```

```
setheading(180)
forward(100) # Bergerak ke kiri
```

Hasil:

Setheading(90) pertama akan membuat objek menghadap keutara kemudian objek akan bergerak maju sejauh 100. Setelah itu setheading(180) akan membuat objek menghadap ke barat dan objek akan maju sejauh 100.



Gambar 12 (Hasil contoh penggunaan setheading)

Dengan `setheading()`, Anda dapat secara langsung mengatur arah turtle ke sudut yang diinginkan tanpa harus menggunakan `left()` atau `right()`. Ini memudahkan pembuatan pola atau gambar yang memerlukan arah spesifik.

D. Kesimpulan

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

E. Kuis

- 1) Jika Anda menggunakan perintah `setheading(90)`, ke arah mana turtle akan menghadap?
 - a. Timur
 - b. Barat
 - c. Utara
 - d. Selatan

- 2) Apa fungsi utama dari perintah `setheading(sudut)`?
 - a. Mengatur posisi turtle ke koordinat (0, 0).
 - b. Mengatur arah turtle ke sudut tertentu berdasarkan derajat.
 - c. Menggerakkan turtle ke arah utara.
 - d. Mengubah warna turtle.

2.6 Home

A. Tujuan Pembelajaran

- Memahami cara mengembalikan posisi turtle ke titik awal menggunakan `'home()'`.
- Mengetahui kegunaan fungsi `'home()'` dalam pengaturan gambar.

B. Definisi

Dalam pustaka Turtle Graphics, metode `'home()'` digunakan untuk memindahkan turtle kembali ke posisi awalnya, yaitu titik (0, 0) pada koordinat kartesius. Selain memindahkan turtle, perintah ini juga mengatur arah turtle menghadap ke timur (0 derajat). Ini berguna ketika Anda ingin memulai kembali menggambar dari posisi awal.

C. Fungsi

`'home()'`: Memindahkan turtle ke koordinat (0, 0) dan mengatur arah turtle menghadap timur.

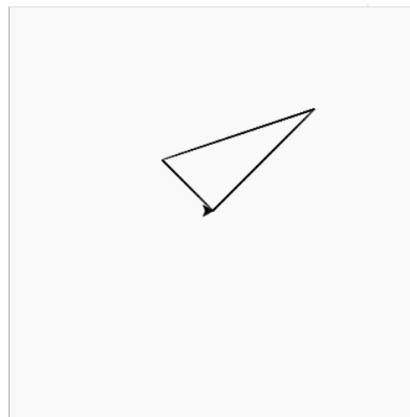
Contoh:

```
# Pindahkan turtle ke beberapa posisi
setposition(100, 100)
setposition(-50, 50)

# Kembali ke posisi awal
home()
```

Hasil:

Home() akan membuat objek kembali ke posisi (0,0).



Gambar 13 (Hasil contoh penggunaan home)

D. Kesimpulan

Perintah `home()` dalam pustaka Turtle Graphics memudahkan untuk kembali ke posisi awal (0, 0) dan mengatur arah turtle. Ini berguna untuk memulai kembali proses menggambar dari titik yang telah ditentukan tanpa harus mengatur koordinat secara manual.

E. Kuis

- 1) Apa fungsi utama dari perintah home()?
 - a. Menghapus seluruh gambar yang telah dibuat.

- b. Mengembalikan turtle ke posisi awal (0, 0) dan mengatur arahnya ke timur.
 - c. Mengganti warna pena turtle.
 - d. Memindahkan turtle ke posisi $y = 0$.
- 2) Jika posisi awal turtle adalah (100, 100) dan arahnya ke barat, apa yang terjadi setelah menggunakan `home()`?
- a. Turtle tetap di posisi (100, 100).
 - b. Turtle kembali ke posisi (0, 0) dengan arah tetap ke barat.
 - c. Turtle kembali ke posisi (0, 0) dan menghadap ke timur.
 - d. Turtle tetap di posisi (100, 100) tetapi menghadap ke timur.

2.7 Circle

A. Tujuan Pembelajaran

- Memahami cara menggambar lingkaran menggunakan `'circle()'`.
- Mengerti parameter yang digunakan dalam fungsi `'circle()'` untuk mengubah ukuran dan arah lingkaran.

B. Definisi

Dalam pustaka Turtle Graphics, metode `'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.

C. Fungsi

`'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``: Jari-jari lingkaran yang ingin digambar.
- ``extent``: Sudut (dalam derajat) dari lingkaran yang ingin digambar. Jika tidak ada, menggambar lingkaran penuh (360 derajat).

D. Contoh Penggunaan `circle()`

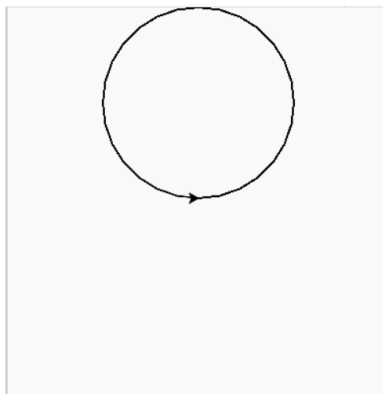
Berikut adalah contoh penggunaan ``circle()`` untuk menggambar lingkaran penuh dan busur:

- Menggambar Lingkaran Penuh

Contoh:

```
# Menggambar lingkaran penuh dengan jari-jari
100
circle(100)
```

Hasil:



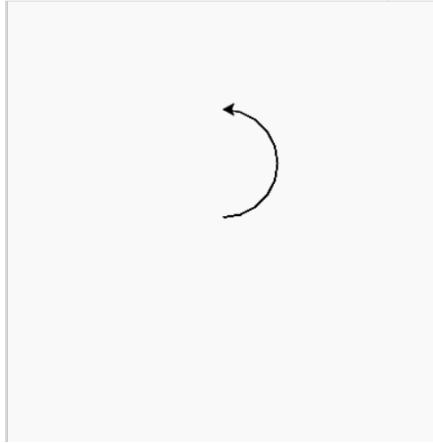
Gambar 14 (Hasil contoh 1 penggunaan `circle`)

- Menggambar Busur Lingkaran

Anda juga dapat menggambar busur dengan menentukan ``extent``. Misalnya, untuk menggambar busur 180 derajat:

```
# Menggambar busur dengan jari-jari 50 dan 180
derajat
circle(50, 180)
```

Hasil:



Gambar 15 (Hasil contoh 2 penggunaan circle)

E. Kesimpulan

Perintah `'circle()'` dalam pustaka Turtle Graphics memungkinkan Anda menggambar lingkaran atau busur dengan mudah. Dengan mengatur jari-jari dan opsi `'extent'`, Anda dapat menggambar berbagai bentuk bulat yang berbeda untuk memperkaya gambar atau pola yang Anda buat.

F. Kuis

- 1) Apa fungsi dari parameter extent dalam metode circle()?
 - a. Mengatur arah lingkaran
 - b. Menentukan ukuran jari-jari lingkaran
 - c. Menentukan sudut lingkaran yang ingin Digambar
 - d. Mengubah warna lingkaran

- 2) Apa yang terjadi jika Anda memanggil circle(-50)?
 - a. Tidak ada lingkaran yang Digambar

- b. Lingkaran dengan jari-jari 50 akan digambar berlawanan arah jarum jam
- c. Lingkaran dengan jari-jari 50 akan digambar searah jarum jam
- d. Lingkaran kecil akan digambar di posisi saat ini

2.8 Dot

A. Tujuan Pembelajaran

- Memahami cara menggambar titik menggunakan ``dot()``.
- Mengerti parameter yang digunakan dalam fungsi ``dot()`` untuk mengubah ukuran dan warna titik.

B. Definisi

Dalam pustaka Turtle Graphics, metode ``dot()`` digunakan untuk menggambar titik pada posisi turtle saat ini. Titik ini dapat diatur ukuran dan warnanya. Fungsi ini sangat berguna untuk menandai posisi tertentu dalam gambar atau untuk menggambar pola dengan titik.

C. Fungsi

`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.

D. Contoh Penggunaan

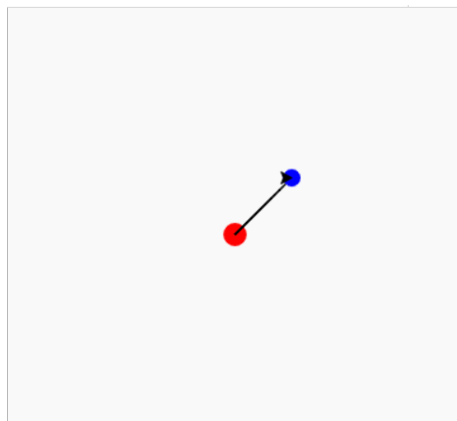
Berikut adalah beberapa contoh penggunaan ``dot()`` untuk menggambar titik:

Contoh:

```
# Menggambar titik dengan ukuran 20 dan warna merah
dot(20, "red")
setposition(50, 50)

# Menggambar titik dengan ukuran 15 dan warna biru
dot(15, "blue")
```

Hasil:



Gambar 16 (Hasil contoh penggunaan dot)

E. Kesimpulan

Perintah `dot()` dalam pustaka Turtle Graphics memungkinkan Anda menggambar titik dengan ukuran dan warna yang dapat disesuaikan. Fungsi ini berguna untuk menandai posisi, membuat pola, atau menambah detail pada gambar yang Anda buat.

F. Kuis

- 1) Apa fungsi dari metode `dot()`?
 - a. Menggambar lingkaran penuh
 - b. Menggambar busur lingkaran
 - c. Menggambar titik di posisi saat ini

d. Menghapus titik pada kanvas

2) Apa yang terjadi jika Anda memanggil `dot(15, "blue")`?

- a. Titik biru dengan ukuran 15 piksel digambar di posisi saat ini
- b. Lingkaran biru dengan ukuran 15 piksel digambar di posisi saat ini
- c. Lingkaran penuh biru dengan jari-jari 15 digambar di posisi saat ini
- d. Tidak ada yang terjadi

3.1 Position

A. Tujuan Pembelajaran

- Memahami fungsi dan penggunaan metode `position()` dalam pustaka Turtle Graphics.
- Memahami cara membaca koordinat posisi saat ini dari turtle.

B. Definisi

Dalam pustaka Turtle Graphics, metode `position()` digunakan untuk mendapatkan posisi saat ini dari turtle di dalam canvas. Posisi ini dinyatakan dalam bentuk pasangan koordinat (x, y).

C. Fungsi

`position()`: Mengembalikan nilai koordinat posisi saat ini dari turtle sebagai pasangan (x, y).

D. Contoh Penggunaan

Contoh 1:

```
# Mendapatkan posisi saat ini dari turtle
pos = position()
print("Posisi saat ini:", pos)
```

Hasil:

```
('Posisi saat ini:', (0.0, 0.0))
```

Gambar 17 (Hasil contoh 1 penggunaan position)

Contoh 2:

Misalnya, untuk memeriksa posisi turtle saat ini sebelum dan sesudah pergerakan.

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

# Gerakkan turtle ke depan sejauh 100 satuan
forward(100)

# Cek posisi baru
print("Posisi setelah bergerak:", position())
```

Hasil:

```
('Posisi awal:', (0.0, 0.0))
('Posisi setelah bergerak:', (100.0, 0.0))
```

Gambar 18 (Hasil contoh 2 penggunaan position)

E. Kesimpulan

Metode `position()` dalam pustaka Turtle Graphics memberikan cara untuk mengetahui posisi saat ini dari turtle. Fungsi ini berguna untuk memantau pergerakan dan membantu dalam menggambar pola atau bentuk yang presisi.

F. Kuis

- 1) Apa fungsi dari metode `position()` dalam Turtle Graphics?
 - a. Mengembalikan arah pergerakan turtle saat ini
 - b. Mengembalikan posisi turtle saat ini dalam bentuk koordinat (x, y)
 - c. Mengatur posisi turtle ke titik tertentu
 - d. Menghapus posisi turtle sebelumnya

- 2) Apa hasil dari perintah berikut jika turtle berada di posisi (50, 100)?


```
print(position())
```

- a. (50, 100)
- b. 0
- c. (0, 0)
- d. [50, 100]

3.2 Xcor & ycor

A. Tujuan Pembelajaran

- Memahami cara menggunakan metode xcor() dan ycor() dalam Turtle Graphics.
- Mengetahui cara memeriksa posisi horizontal (sumbu x) dan vertikal (sumbu y) dari turtle.

B. Definisi

Dalam pustaka Turtle Graphics, metode xcor() dan ycor() memungkinkan kita mendapatkan koordinat posisi spesifik di sepanjang sumbu horizontal (x) dan vertikal (y). Keduanya digunakan untuk menilai atau memonitor pergerakan turtle secara lebih rinci.

- xcor(): Mengembalikan posisi horizontal (sumbu x) turtle.
- ycor(): Mengembalikan posisi vertikal (sumbu y) turtle.

C. Fungsi

xcor(): Digunakan untuk mendapatkan nilai posisi sumbu x saat ini.

ycor(): Digunakan untuk mendapatkan nilai posisi sumbu y saat ini.

Metode-metode ini ideal untuk memantau posisi terkini, membantu menentukan apakah turtle telah mencapai batas tertentu, atau digunakan dalam penghitungan pola geometris yang memerlukan pengawasan koordinat.

D. Contoh Penggunaan

Berikut adalah contoh yang menunjukkan bagaimana menggunakan `xcor()` dan `ycor()` bersama-sama dalam membuat suatu pola:

```
# Menampilkan posisi awal turtle
print("Posisi x awal:", xcor())
print("Posisi y awal:", ycor())

# Memindahkan turtle ke posisi (100, 50)
setposition(100, 50)
print("Posisi setelah goto(100, 50):")
print("Posisi x:", xcor())
print("Posisi y:", ycor())

# Gerakkan maju dan periksa posisi
print("Gerakkan maju 50 satuan")
forward(50)
print("Posisi x setelah gerakan maju:", xcor())
print("Posisi y setelah gerakan maju:", ycor())
```

Hasil:

```
('Posisi x awal:', 0.0)
('Posisi y awal:', 0.0)
Posisi setelah setposition(100, 50):
('Posisi x:', 100.0)
('Posisi y:', 50.0)
Gerakkan maju 50 satuan
('Posisi x setelah gerakan maju:', 150.0)
('Posisi y setelah gerakan maju:', 50.0)
```

Gambar 19 (Hasil contoh penggunaan `xcor` dan `ycor`)

E. Kesimpulan

Metode `xcor()` dan `ycor()` dalam Turtle Graphics sangat penting untuk mengontrol posisi turtle secara terpisah pada sumbu x dan y.

Mereka memungkinkan pengguna memanfaatkan logika posisi lebih efisien dalam menggambar pola yang kompleks dan memverifikasi gerakan dengan tepat.

F. Kuis

- 1) Apa perbedaan antara `xcor()` dan `ycor()`?
 - a. `xcor()` mengembalikan posisi horizontal (sumbu x), sementara `ycor()` mengembalikan posisi vertikal (sumbu y).
 - b. `xcor()` mengatur posisi horizontal, sementara `ycor()` mengembalikan posisi vertikal.
 - c. `xcor()` menghapus posisi pada sumbu x, sementara `ycor()` membaca posisi awal.
 - d. Tidak ada perbedaan, keduanya mengembalikan posisi (x, y).
- 2) Apa hasil dari perintah berikut jika turtle berada di koordinat (30, -20)?

```
print(xcor())
```

 - a. -20
 - b. 30
 - c. (30, -20)
 - d. 0

3.3 Heading

A. Tujuan Pembelajaran

- Memahami konsep arah pergerakan turtle dalam Turtle Graphics.
- Mengetahui cara menggunakan metode `heading()` untuk mendapatkan dan mengatur arah pergerakan turtle.

B. Definisi

Metode `heading()` adalah fungsi bawaan dalam pustaka Turtle Graphics yang mengembalikan arah pergerakan turtle dalam satuan derajat. Dalam sistem koordinat Turtle Graphics:

- 0 derajat mengarah ke timur (kanan),
- 90 derajat mengarah ke utara (atas),
- 180 derajat mengarah ke barat (kiri), dan
- 270 derajat mengarah ke selatan (bawah).

C. Fungsi

Heading ini berfungsi mengembalikan nilai derajat yang menunjukkan arah pergerakan turtle saat ini.

D. Contoh Penggunaan

Contoh:

```
# 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:

```
('Arah awal:', 0.0)
('Arah setelah putaran:', 90.0)
```

Gambar 20 (Hasil contoh penggunaan heading)

E. Kesimpulan

Metode `heading()` pada Turtle Graphics sangat bermanfaat untuk memantau dan mengontrol arah gerakan turtle dengan lebih tepat. Pemahaman tentang sistem derajat akan membantu siswa menggambar bentuk dan pola secara lebih akurat.

F. Kuis

- 1) Dalam Turtle Graphics, arah 0 derajat mengarah ke mana?
 - a. Utara (atas)
 - b. Timur (kanan)
 - c. Selatan (bawah)
 - d. Barat (kiri)
- 2) Apa hasil dari perintah berikut jika turtle sudah diputar 90 derajat ke kiri?

```
print(heading())
```

- a. 0
- b. 90
- c. 180
- d. 270

3.4 Distance

A. Tujuan Pembelajaran

- Memahami cara kerja metode `distance()` dalam Turtle Graphics.
- Menggunakan metode `distance()` untuk menghitung jarak antara dua titik.

B. Definisi

Metode `distance()` adalah fungsi yang digunakan untuk menghitung jarak Euclidean antara posisi turtle saat ini dan titik

tertentu dengan koordinat yang diberikan. Metode ini dapat dipanggil dengan:

- `distance(x, y)`: Menghitung jarak antara posisi saat ini dan koordinat (x, y).
- `distance((x, y))`: Alternatif pemanggilan dengan tuple koordinat.

C. Fungsi

`distance(x, y)` berfungsi mengembalikan nilai jarak Euclidean antara posisi turtle saat ini dan koordinat (x, y).

D. Contoh Penggunaan

```
# Hitung jarak dari posisi awal ke titik (100, 100)
jarak = distance(100, 100)
print("Jarak ke (100, 100):", jarak)

# Gerakkan turtle ke titik (100, 100)
goto(100, 100)

# Hitung jarak setelah mencapai (100, 100)
print("Jarak setelah mencapai (100, 100):",
distance(100, 100))
```

Hasil:

```
('Jarak ke (100, 100):', 141.421356237)
('Jarak setelah mencapai (100, 100):', 0.0)
```

Gambar 21 (Hasil contoh penggunaan distance)

E. Kesimpulan

Metode `distance()` sangat berguna dalam Turtle Graphics untuk menghitung jarak antara posisi awal dan posisi tujuan. Pemahaman metode ini membantu menggambar bentuk geometris yang proporsional dan akurat.

F. Kuis

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

```
print(distance(3, 4))
```

- a. 3
- b. 4
- c. 7
- d. 5

4.1 Pendown & penup

A. Tujuan Pembelajaran

- Memahami perbedaan antara mode menggambar dan mode tidak menggambar dalam Turtle Graphics.
- Menggunakan metode pendown() dan penup() untuk mengatur status pena saat menggambar.

B. Definisi

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

C. Fungsi

pendown(): Mengaktifkan mode menggambar.

penup(): Menonaktifkan mode menggambar.

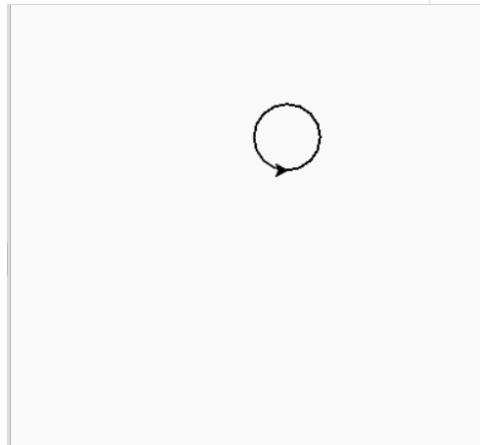
D. Contoh Penggunaan Gabungan

```
# Gerakkan tanpa menggambar
penup()
goto(50, 50)

# Aktifkan menggambar
pendown()
circle(30)
```

Hasil:

Objek akan berpindah terlebih dahulu ketitik (50,50) tanpa menggambar garis.



Gambar 22 (Hasil contoh penggunaan penup)

E. Kesimpulan

Metode 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.

F. Kuis

- 1) Apa yang terjadi jika metode pendown() tidak dipanggil setelah penup()?
 - a. Turtle akan terus menggambar saat bergerak.
 - b. Turtle akan berhenti bergerak.
 - c. Turtle tidak akan menggambar garis saat bergerak.
 - d. Turtle akan menggambar lingkaran secara otomatis.
- 2) Perhatikan kode berikut:

```
penup()  
goto(100, 100)  
pendown()
```

```
goto(200, 200)
```

Apa yang terjadi setelah kode tersebut dijalankan?

- Turtle menggambar garis dari titik awal ke (100, 100).
- Turtle menggambar garis dari (100, 100) ke (200, 200).
- Turtle tidak menggambar sama sekali.
- Turtle hanya menggambar lingkaran.

4.2 Pensize

A. Tujuan Pembelajaran

- Mengubah ketebalan garis menggunakan metode pensize() dalam Turtle Graphics.
- Menerapkan ketebalan garis untuk menciptakan efek visual yang berbeda.

B. Definisi

Metode pensize(width) digunakan untuk mengatur ketebalan pena yang digunakan oleh turtle 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.

C. Fungsi

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

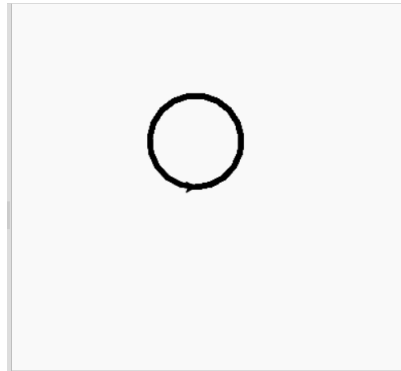
D. Contoh Penggunaan

Contoh:

```
# Ubah ketebalan pena  
pensize(5)
```

```
# Gambar lingkaran dengan garis tebal  
circle(50)
```

Hasil:



Gambar 23 (Hasil contoh penggunaan pensize)

E. Kesimpulan

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

F. Kuis

- 1) Apa efek dari penggunaan metode `pensize(10)` sebelum menggambar?
 - a. Mengganti warna garis menjadi hitam.
 - b. Mengubah ketebalan garis menjadi 10 piksel.
 - c. Membuat garis tidak terlihat.
 - d. Menghapus garis yang telah digambar.
- 2) Jika kode berikut dijalankan:

```
pensize(5)  
circle(30)  
pensize(2)  
circle(50)
```

Apa yang akan terlihat pada hasil akhir?

- a. Dua lingkaran dengan ketebalan garis yang sama.
- b. Lingkaran pertama memiliki garis lebih tebal daripada lingkaran kedua.
- c. Lingkaran kedua memiliki garis lebih tebal daripada lingkaran pertama.
- d. Tidak ada lingkaran yang digambar.

4.3 Isdown

A. Tujuan Pembelajaran

- Mengetahui status pena apakah sedang aktif menggambar atau tidak.

B. Definisi

Metode `isdown()` digunakan untuk memeriksa status pena dalam Turtle Graphics. Metode ini mengembalikan nilai `True` jika pena sedang dalam posisi turun (aktif menggambar) dan `False` jika pena diangkat. Pemeriksaan status pena berguna untuk menentukan apakah turtle akan meninggalkan jejak saat bergerak, memungkinkan kontrol penuh terhadap proses menggambar.

C. Fungsi

`isdown()`: Mengecek apakah pena sedang turun.

D. Contoh Penggunaan Gabungan

Contoh:

```
# Cek status pena
print("Pena aktif menggambar:", isdown())

penup()
```

```
print("Pena setelah diangkat:", isdown())
```

Hasil:

```
('Pena aktif menggambar:', True)  
( 'Pena setelah diangkat:', False)
```

Gambar 24 (Hasil contoh penggunaan isdown)

E. Kesimpulan

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

F. Kuis

1) Apa yang dikembalikan oleh metode isdown() jika pena sedang dalam posisi turun?

- a. True
- b. False
- c. None
- d. Error

2) Perhatikan kode berikut:

```
pendown()  
print(isdown())  
penup()  
print(isdown())
```

Apa output dari kode tersebut?

- a. True, False
- b. False, True
- c. True, True
- d. False, False

5.1 Color & pencolor

A. Tujuan Pembelajaran.

- Mampu mengubah warna garis pada gambar menggunakan fungsi `pencolor`.

B. Definisi

Dalam library Turtle, `pencolor` adalah fungsi yang digunakan untuk mengontrol warna garis pada gambar yang dihasilkan oleh turtle.

- **pencolor**: Fungsi ini digunakan untuk mengatur warna garis yang digambar oleh turtle.

C. Fungsi

Pencolor

- Mengatur warna garis sesuai parameter warna yang diberikan.
- Parameter dapat berupa nama warna (contoh: "red") atau kode warna RGB (contoh: "#FF0000").

D. Contoh Penggunaan

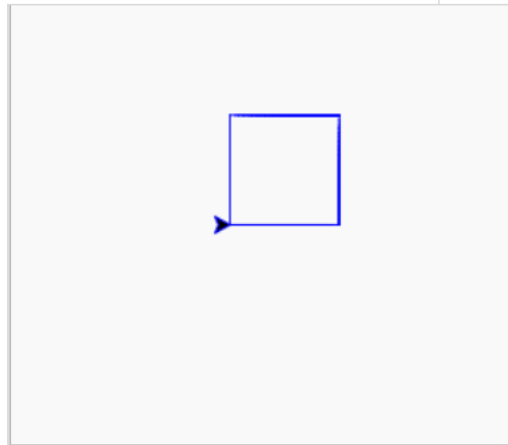
Contoh:

```
# 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 25 (Hasil contoh penggunaan pencolor)

E. Kesimpulan

Fungsi `pencolor` digunakan mengatur warna garis lintasan yang dilalui oleh objek turtle.

F. Kuis

1) Apa yang terjadi jika kode berikut dijalankan?

```
pencolor("red")
```

```
forward(100)
```

- Turtle menggambar garis merah sepanjang 100 piksel.
- Turtle menggambar garis hitam sepanjang 100 piksel.
- Turtle menggambar garis tanpa warna sepanjang 100 piksel.
- Turtle tidak menggambar garis.

2) Apa format yang tidak valid untuk parameter warna dalam fungsi

```
pencolor()?
```

- "blue"

- b. "#00FF00"
- c. (1, 0, 0)
- d. "bold_red"

5.2 Pengisian Warna (Fillcolor, Begin_fill, dan End_fill)

A. Tujuan Pembelajaran

- Siswa memahami cara kerja fungsi fillcolor, begin_fill, dan end_fill dalam mengatur warna isian pada bentuk geometris.
- Siswa mampu menggunakan fungsi-fungsi tersebut secara praktis dalam proses menggambar bentuk.

B. Definisi

Pengisian warna pada library Turtle:

- fillcolor():
Digunakan untuk menentukan warna isian yang akan diterapkan pada bentuk.
Parameter warna dapat berupa nama warna, kode RGB, atau tuple RGB (contoh: (0.5, 0.5, 0.5)).
- begin_fill():
Menandai awal area yang akan diisi warna.
- end_fill():
Menandai akhir area yang akan diisi warna. Warna isian diterapkan sesuai pengaturan fillcolor atau parameter kedua fungsi color.

C. Fungsi

- a) Fungsi fillcolor(warna):
 - Mengatur warna isian pada bentuk.
 - Digunakan sebelum memulai proses menggambar area yang akan diisi warna.

b) Fungsi `begin_fill()` dan `end_fill()`:

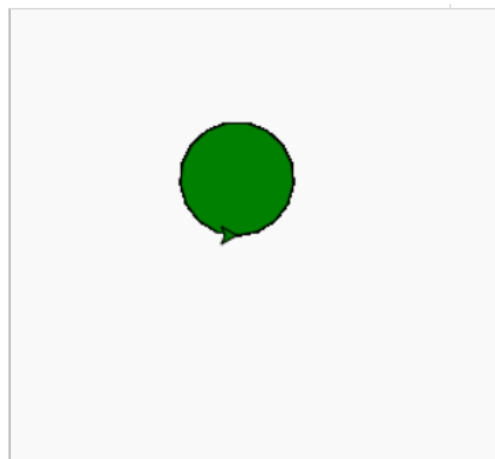
- Proses pengisian warna dimulai dengan `begin_fill()` dan diakhiri dengan `end_fill()`.
- Area yang digambar di antara kedua fungsi ini akan diisi dengan warna sesuai pengaturan sebelumnya.

D. Contoh Penggunaan

Contoh 1: Menggunakan `fillcolor`

```
fillcolor("green") # Warna isian hijau  
  
begin_fill()  
circle(50) # Membuat lingkaran  
end_fill()
```

Hasil:



Gambar 26 (Hasil contoh penggunaan `Fillcolor`, `Begin_fill`, dan `End_fill`)

E. Kesimpulan

Fungsi `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.

F. Kuis

1) Perhatikan kode berikut:

```
fillcolor("yellow")
begin_fill()
circle(50)
end_fill()
```

Apa hasil dari kode tersebut?

- a. Sebuah lingkaran dengan warna garis kuning tanpa isian.
- b. Sebuah lingkaran dengan warna isian kuning.
- c. Sebuah lingkaran dengan warna garis hitam dan isian kuning.
- d. Tidak ada lingkaran yang digambar.

2) Apa fungsi dari `end_fill()` dalam proses pengisian warna?

- a. Mengatur warna isian menjadi transparan.
- b. Menandai akhir area yang akan diisi warna.
- c. Menonaktifkan pengaturan warna pada turtle.
- d. Menghapus warna isian dari bentuk yang digambar.

6.1 Reset

A. Tujuan Pembelajaran

- Siswa memahami fungsi reset dalam library Turtle.

B. Definisi

Fungsi reset digunakan untuk menghapus semua gambar pada layar Turtle dan mengatur ulang posisi serta atribut turtle ke keadaan awal.

C. Fungsi

reset():

- Menghapus semua gambar pada layar Turtle.
- Mengatur ulang posisi turtle ke tengah layar.
- Mengatur ulang semua atribut turtle, seperti warna dan ukuran pena, ke nilai default.

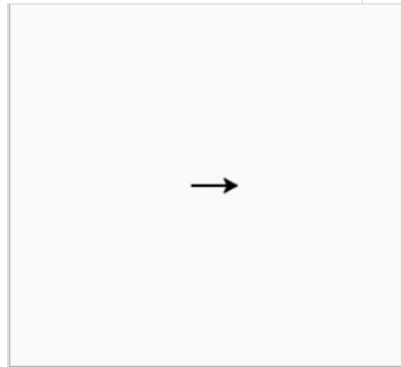
D. Contoh Penggunaan

Contoh:

```
color("red")
forward(100)
left(90)
forward(100)

# Reset layar dan posisi turtle
reset()
forward(50) # Memulai gambar baru
```

Hasil:



Gambar 27 (Hasil dari contoh penggunaan reset)

E. Kesimpulan

Fungsi reset sangat berguna untuk memulai ulang proses menggambar tanpa harus menutup dan membuka ulang jendela Turtle.

F. Kuis

- 1) Apa yang terjadi jika fungsi reset() dipanggil setelah menggambar garis?
 - a. Semua gambar dihapus, tetapi atribut turtle tetap sama.
 - b. Semua gambar dihapus, dan turtle kembali ke posisi awal dengan atribut default.
 - c. Garis tidak dihapus, tetapi posisi turtle berubah.
 - d. Turtle akan keluar dari jendela.

- 2) Perhatikan kode berikut:

```
color("green")
forward(100)
reset()
circle(50)
```

Apa warna lingkaran yang akan dihasilkan?

- a. Hijau

- b. Hitam
- c. Biru
- d. Merah

6.2 Clear

A. Tujuan Pembelajaran

- Siswa memahami fungsi clear dalam library Turtle.
- Siswa dapat menggunakan fungsi ini untuk menghapus gambar tanpa mengubah posisi atau atribut turtle.

B. Definisi

Fungsi clear digunakan untuk menghapus semua gambar pada layar Turtle tanpa mengatur ulang posisi atau atribut turtle.

C. Fungsi

clear():

- Menghapus semua gambar pada layar Turtle.
- Posisi dan atribut turtle tetap sama.

D. Contoh Penggunaan

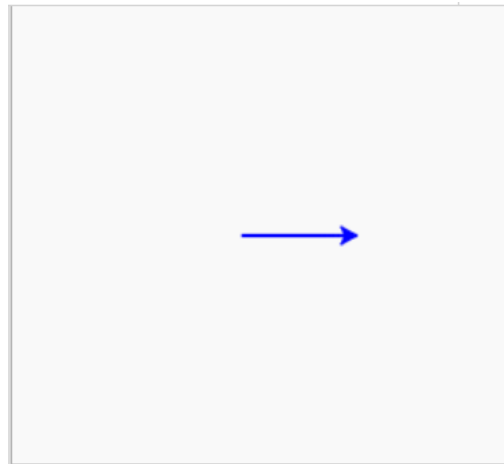
Contoh:

```
color("blue")
circle(50)

# Menghapus gambar tetapi posisi dan atribut tetap sama
clear()
forward(100) # Memulai gambar baru dari posisi terakhir
```

Hasil:

Setelah di `clear()` atribut color “blue” akan tetap aktif.



Gambar 28 (Hasil contoh penggunaan clear)

E. Kesimpulan

Fungsi `clear` berguna untuk menghapus gambar tanpa mengubah pengaturan posisi atau atribut turtle.

F. Kuis

- 1) Apa perbedaan utama antara fungsi `clear()` dan `reset()`?
 - a. `clear()` menghapus gambar tanpa mengubah posisi atau atribut, sedangkan `reset()` juga mengatur ulang posisi dan atribut turtle.
 - b. `clear()` menghapus gambar beserta posisi turtle, sedangkan `reset()` hanya menghapus gambar.
 - c. `clear()` tidak menghapus gambar, sedangkan `reset()` menghapus gambar.
 - d. `clear()` menutup jendela, sedangkan `reset()` tidak.

2) Perhatikan kode berikut:

```
color("blue")
```

```
circle(50)
```

```
clear()
```

```
forward(100)
```

```
circle(30)
```

Apa warna lingkaran kedua?

- a. Tidak berwarna
- b. Biru
- c. Hitam
- d. Merah

6.3 Write

A. Tujuan Pembelajaran

- Siswa memahami fungsi write dalam library Turtle.
- Siswa mampu menampilkan teks pada layar menggunakan fungsi ini.

B. Definisi

Fungsi write digunakan untuk menulis teks pada layar Turtle di lokasi turtle saat ini.

C. Fungsi

```
write(teks, move=False, align="left", font=("Arial", 8, "normal")):
```

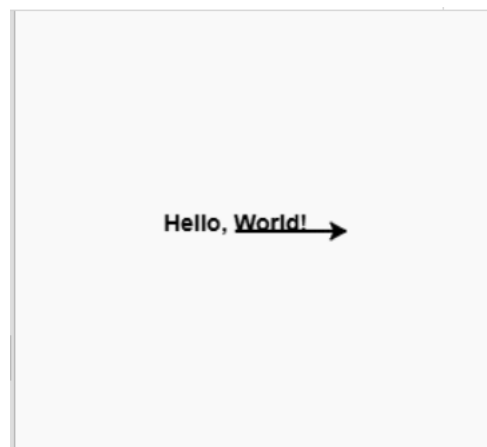
- Parameter teks: Teks yang akan ditulis.
- Parameter move: Jika True, posisi turtle akan berpindah setelah menulis teks.
- Parameter align: Penyesuaian teks ("left", "center", atau "right").
- Parameter font: Menentukan jenis, ukuran, dan gaya font.

D. Contoh Penggunaan

Contoh:

```
write("Hello, World!", align="center",  
font=("Arial", 16, "bold"))  
forward(100) # Turtle tetap dapat menggambar  
setelah menulis
```

Hasil:



Gambar 29 (Hasil contoh penggunaan write)

E. Kesimpulan

Fungsi write memungkinkan turtle untuk menampilkan teks di layar, sehingga dapat digunakan untuk memberikan informasi tambahan pada gambar.

F. Kuis

- 1) Apa fungsi dari parameter align dalam metode write()?
 - A. Mengatur posisi turtle setelah menulis teks.
 - B. Menentukan jenis font yang digunakan.
 - C. Mengatur perataan teks (kiri, tengah, atau kanan).

D. Mengatur warna teks.

2) Perhatikan kode berikut:

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

Apa yang akan terjadi?

- a. Teks ditulis di layar dengan font Arial, ukuran 12, dan bergaya italic di posisi kiri turtle.
- b. Teks ditulis di layar dengan font Arial, ukuran 12, dan bergaya italic di posisi tengah turtle.
- c. Teks ditulis di layar dengan font Arial, ukuran 12, tetapi tidak bergaya italic.
- d. Tidak ada teks yang ditulis karena font tidak valid.