

Bentuk 2D dasar

✓ Titik

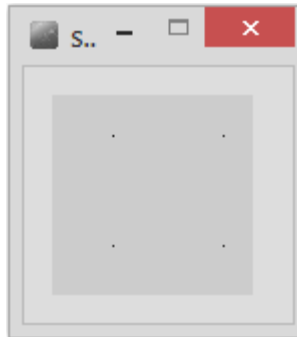
Syntax

```
point(x, y)
```

x: koordinat x
y: koordinat y

Contoh

```
point(30, 20);  
point(85, 20);  
point(85, 75);  
point(30, 75);
```



✓ Garis

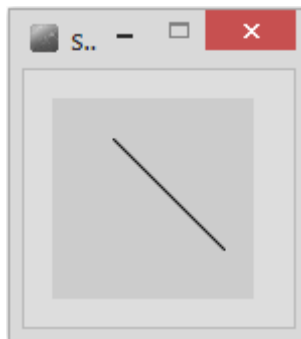
Syntax

```
line(x1, y1, x2, y2)
```

x1: koordinat titik pertama garis X
x2: koordinat titik kedua garis X
y1: koordinat titik pertama garis Y
y2: koordinat titik kedua garis Y
z1: koordinat titik pertama garis Z
z2: koordinat titik kedua garis Z

Contoh 1

```
line(30, 20, 85, 75);
```



Contoh 2

```
line(30, 20, 85, 20);  
stroke(126);  
line(85, 20, 85, 75);  
stroke(255);  
line(85, 75, 30, 75);
```



✓ Ellipse

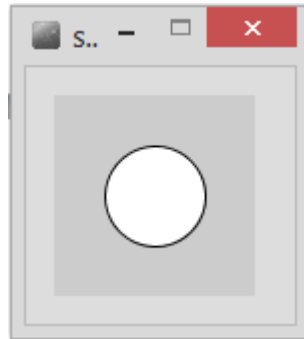
Syntax

```
ellipse(a, b, c, d)
```

a: koordinat x
b: koordinat y
c: Panjang ellipse
d: Lebar ellipse

Contoh

```
ellipse(50, 50, 50, 50);
```



✓ Kotak

Syntax:

```
rect(a, b, c, d)
```

```
rect(a, b, c, d, r)
```

```
rect(a, b, c, d, tl, tr, br, bl)
```

a: koordinat x
b: koordinat y
c: lebar kotak
d: tinggi kotak
r: garis lengkung pada semua sudut kotak
tl: garis lengkung pada titik kiri atas
tr: garis lengkung pada titik kanan atas
br: garis lengkung pada titik kanan bawah
bl: garis lengkung pada titik kiri bawah

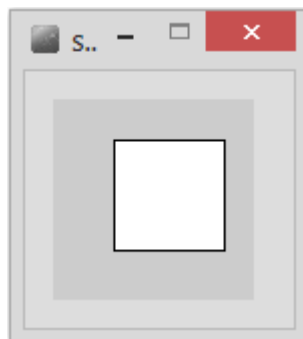
Contoh 1

```
rect(30, 20, 55,  
55);
```

Contoh 2

```
rect(30, 20, 55, 55,  
7);
```

Contoh 3



```
rect(30, 20, 55, 55, 3, 6, 12, 18);
```



Segi tiga

Syntax:

```
triangle(x1, y1, x2, y2, x3, y3)
```

x1: koordinat titik x pertama
y1: koordinat titik y pertama
x2: koordinat titik x kedua
y2: koordinat titik y kedua
x3: koordinat titik x ketiga
y3: koordinat titik y ketiga

Contoh

```
triangle(30, 75, 58, 20, 86,  
75);
```

✓ Background

Latar belakang dari sketch kita. Berupa warna rgb, gray atau gambar.

Syntax

```
background(rgb)  
background(rgb, alpha)  
background(gray)  
background(gray, alpha)  
background(v1, v2, v3)  
background(v1, v2, v3, alpha)  
background(image)
```

rgb : semua harga dari data warna (format 0-255 atau hexa)

alpha: opacity of the background

gray: specific a value between white and black

v1 : harga red atau hue (bergantung pada penggunaan mode warna)

v2 : harga green atau saturation (bergantung pada penggunaan mode warna)

v3 : harga blue atau brightness (bergantung pada penggunaan mode warna)

image : PImage yang dijadikan sebagai background (gambar harus terdefiniskan dalam project dengan mendrag gambar pada sketch dan ukuran gambar harus sesuai dengan ukuran tampilan program "size(x,y);")

Contoh 1

```
background(51); //kalian ubah sendiri nilainya  
triangle(30, 75, 58, 20, 86, 75);
```

Contoh 2

```
background(65,255,204);
```

```
triangle(30, 75, 58, 20, 86, 75);
```

Contoh 3

```
size(1024,768); //ukuran dimensi file Gambar 1024 x 768 pixels  
PImage img;  
img = loadImage("Gambar.jpg"); //drag pada sketch file JPG dengan nama Gambar  
background(img);  
triangle(67,88,90,33,123,234); //buat segitiga
```

✓ Fill

Merupakan warna di dalam sebuah object baik kotak, ellipse maupun triangle.

Syntax

```
fill(rgb)  
fill(rgb, alpha)  
fill(gray)  
fill(gray, alpha)  
fill(v1, v2, v3)  
fill(v1, v2, v3, alpha)
```

rgb : semua harga dari data warna (format 0-255 atau hexa)

alpha: opacity of the background

gray: specific a value between white and black

v1 : harga red atau hue (bergantung pada penggunaan mode warna)

v2 : harga green atau saturation (bergantung pada penggunaan mode warna)

v3 : harga blue atau brightness (bergantung pada penggunaan mode warna)

Contoh 1

```
fill(153);  
rect(30, 20, 55, 55);
```

Contoh 2

```
fill(204, 152, 0);  
rect(30, 20, 55, 55);
```

✓ Stroke

Merupakan warna garis terluar sebuah object.

Syntax

```
stroke(rgb)  
stroke(rgb, alpha)  
stroke(gray)
```

```
stroke(gray, alpha)
stroke(v1, v2, v3)
stroke(v1, v2, v3,
alpha)
```

rgb : semua harga dari data warna (format 0-255 atau hexa)
alpha: opacity of the background
gray: specific a value between white and black
v1 : harga red atau hue (bergantung pada penggunaan mode warna)
v2 : harga green atau saturation (bergantung pada penggunaan mode warna)
v3 : harga blue atau brightness (bergantung pada penggunaan mode warna)

Contoh 1

```
stroke(159);
fill(209);
rect(30, 20, 55, 55);
```

Contoh 2

```
stroke(0, 204, 152);
fill(204, 152, 0);
rect(30, 20, 55, 55);
```

✓ noFill()

fungsi untuk disable warna pada background object.

Syntax

```
noFill()
```

Contoh

```
fill(255,60,80);
rect(15, 10, 55, 55);
noFill();
rect(30, 20, 55, 55);
```

✓ noStroke()

Fungsi untuk disable garis luar pada object

Syntax

```
noStroke()
```

```
)
```

Contoh

```
stroke(255,60,80);  
rect(15, 10, 55, 55);  
noStroke();  
rect(30, 20, 55, 55);
```

Input Mouse

✓ mouseButton

Fungsi yang digunakan untuk menjadikan mouse sebagai tombol

Contoh 1

```
void draw() {  
  if (mousePressed && (mouseButton == LEFT)) {  
    fill(0);  
  } else if (mousePressed && (mouseButton == RIGHT)) {  
    fill(255);  
  } else {  
    fill(126);  
  }  
  rect(25, 25, 50, 50);  
}
```

Contoh 2

```
void draw() {  
  rect(25, 25, 50, 50);  
}  
  
void mousePressed() {  
  if (mouseButton == LEFT) {  
    fill(0);  
  } else if (mouseButton == RIGHT) {  
    fill(255);  
  } else {  
    fill(126);  
  }  
}
```

❖ Apa perbedaan dua contoh program diatas?

✓ mouseClicked()

Sebuah program akan dijalankan setelah tombol mouse ditekan dan dilepas

Contoh

```
int value = 0;

void draw() {
    fill(value);
    rect(25, 25, 50, 50);
}

void mouseClicked() {
    if (value == 0) {
        value = 255;
    } else {
        value = 0;
    }
}
```

❖ Apa yang terjadi pada program diatas?

✓ mouseDragged()

Sebuah fungsi yang digunakan untuk menjalankan program selama mouse dilakukan dragging.

Contoh

```
int value = 0;

void draw() {
    fill(value);
    rect(25, 25, 50, 50);
}

void mouseDragged()
{
    value = value + 5;
    if (value > 255) {
        value = 0;
    }
}
```

❖ Apa yang terjadi pada program diatas?

✓ mouseX dan mouseY

Contoh 1

```
void draw()
```

```
{
  background(204);
  line(mouseX, 20, mouseX, 80);
}
```

Contoh 2

```
void draw()
{
  background(204);
  line(20, mouseY, 80,
mouseY);
}
```

❖ **Apa perbedaan kedua program diatas?**

Input Keyboard

✓ **Key**

Fungsi untuk menjalankan program ketika sebuah tombol keyboard ditekan

Contoh

```
// tekan huruf b
void draw() {
  if (keyPressed) {
    if (key == 'b' || key == 'B') {
      fill(0);
    }
  } else {
    fill(255);
  }
  rect(25, 25, 50, 50);
}
```

❖ **Apa yang terjadi?**

✓ **keyPressed()**

Contoh

```
//Tekan sembarang
tombol
```



```
int value = 0;

void draw() {
  fill(value);
  rect(25, 25, 50, 50);
}

void keyPressed() {
  if (value == 0) {
    value = 255;
  } else {
    value = 0;
  }
}
```

✓ keyPressed()

Contoh

```
int value = 0;

void draw() {
  fill(value);
  rect(25, 25, 50, 50);
}

void keyReleased() {
  if (value == 0) {
    value = 255;
  } else {
    value = 0;
  }
}
```

❖ Apa perbedaan keyPressed() dan keyRelease() ?

Apakah semua program diatas telah Anda kuasai?
Mari menuju pembelajaran selanjutnya tentang kamera

Demo program

```
import processing.video.*;

Capture cam;
```

```

void setup() {
    size(640, 480);

    String[] cameras = Capture.list();

    if (cameras == null) {
        println("Gagal mencari kamera... Coba lagi...");
        cam = new Capture(this, 640, 480);
    } if (cameras.length == 0) {
        println("Tidak ada kamera terdeteksi.");
        exit();
    } else {
        println("Kamera tersedia:");
        for (int i = 0; i < cameras.length; i++) {
            println(cameras[i]);
        }

        //Memilih kamera yang pas untuk project
        cam = new Capture(this, cameras[0]);
        //atau langsung menggunakan syntax berikut untuk setting langsung camera
        //cam = new Capture(this, 640, 480, "Built-in iSight", 30);

        // Mulai memindai gambar dari kamera
        cam.start();
    }
}

void draw() {
    if (cam.available() == true) {
        cam.read();
    }
    image(cam, 0, 0);
}

```

❖ Jelaskan apa yang terjadi pada program diatas! Beri screenshot

Memasang Library OpenCV pada Processing (Online)

1. Klik sketch pada menu bar
2. Import library
3. Add library. Akan muncul library manager.
4. Pilih berdasarkan category (video and Vision) atau ketik langsung OpenCV for Processing.
5. Instal library OpenCV for Processing.

***membutuhkan koneksi stabil dan kuota kurang lebih 100 Mb.**

**** pastikan library telah terinstal dengan adanya tombol "Remove" pada kolom pilihan OpenCV for Processing di Library Manager.**