

## Jobsheet III

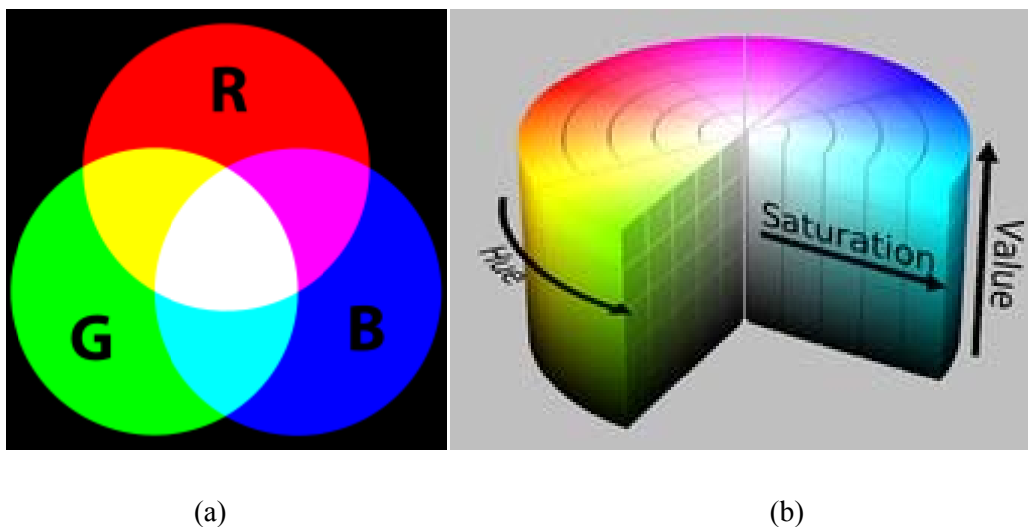
### Warna RGB dan HSB

#### A. Tujuan

- Mahasiswa mampu menjelaskan penggunaan filter warna pada OpenCV untuk Processing
- Mahasiswa mampu mengoperasikan filter warna pada OpenCV untuk Processing.

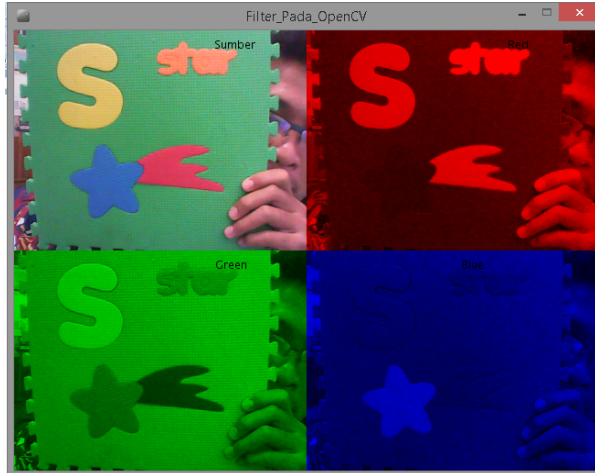
#### B. Dasar Teori

Ada dua mode warna pada OpenCV, yaitu RGB (Red, Green, Blue) dan HSB (Hue, Saturation, Brightness). Berikut merupakan gambaran warna RGB dan HSB.

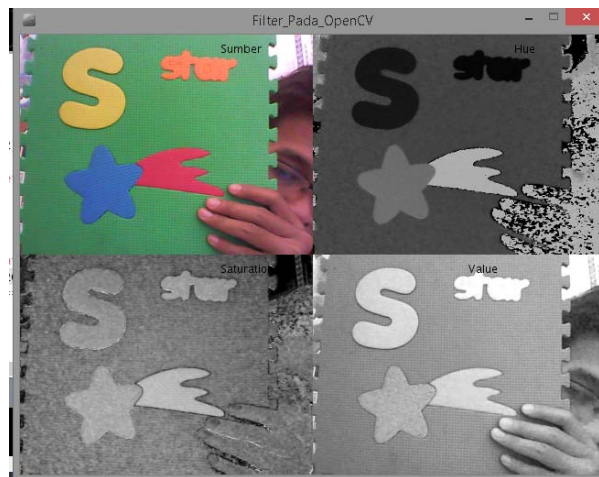


**Gambar 3.1** Merupakan Mode Warna pada OpenCV (a) RGB (b) HSB

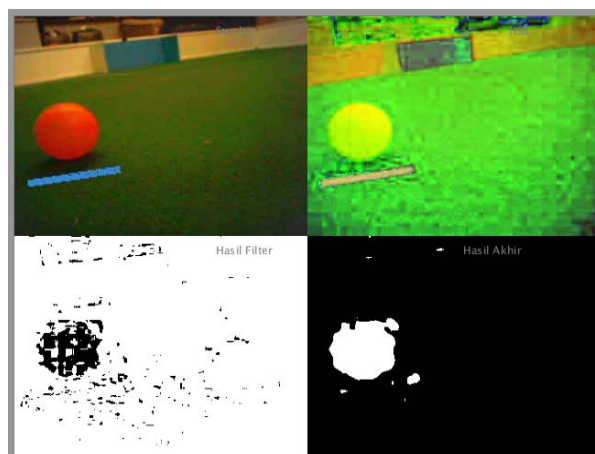
Pada warna RGB, setiap warna akan sulit untuk dipisahkan. Dikarenakan dari ketiga warna membutuhkan warna yang lainnya untuk memperoleh warna baru. Misalkan, ketika kita membutuhkan warna kuning, maka harus memadukan warna merah dan hijau. Warna cyan diperoleh dari penggabungan warna biru dan hijau. Sedangkan warna magenta diperoleh dari penggabungan warna biru dan merah. Namun, ketika kita membutuhkan warna putih, harus menggabungkan ketiganya. Maka, untuk menggunakan filter warna, mode warna RGB akan mempersulit proses pengambilan warna. Oleh sebab itu, pada penggunaan filter warna, digunakan mode warna HSB atau HSV.



**Gambar 3.2** Kanal Warna RGB pada OpenCV



**Gambar 3.3** Kanal Warna HSV pada OpenCV



**Gambar 3.4** Filter Warna Oranye pada OpenCV

Pada processing penulisan penggunaan mode warna dapat dilakukan dengan kode berikut:

```
opencv.useColor( RGB );  
opencv.useColor( HSB );
```

Sedangkan untuk pengambilan setiap warna menggunakan kode berikut:

```
r = opencv.getSnapshot( opencv.getR() );  
g = opencv.getSnapshot( opencv.getG() );  
b = opencv.getSnapshot( opencv.getB() );  
  
h = opencv.getSnapshot( opencv.getH() );  
s = opencv.getSnapshot( opencv.getS() );  
v = opencv.getSnapshot( opencv.getV() );
```

Untuk pemfilteran grayscale dari setiap warna yang sudah diambil dapat menggunakan kode berikut tergantung warna apa yang digunakan.

```
opencv.setGray( opencv.getR().clone() );  
opencv.inRange( min, max );  
  
opencv.setGray( opencv.getG().clone() );  
opencv.inRange( min, max );  
  
opencv.setGray( opencv.getB().clone() );  
opencv.inRange( min, max );  
  
opencv.setGray( opencv.getH().clone() );  
opencv.inRange( min, max );  
  
opencv.setGray( opencv.getS().clone() );  
opencv.inRange( min, max );  
  
opencv.setGray( opencv.getV().clone() );  
opencv.inRange( min, max );
```

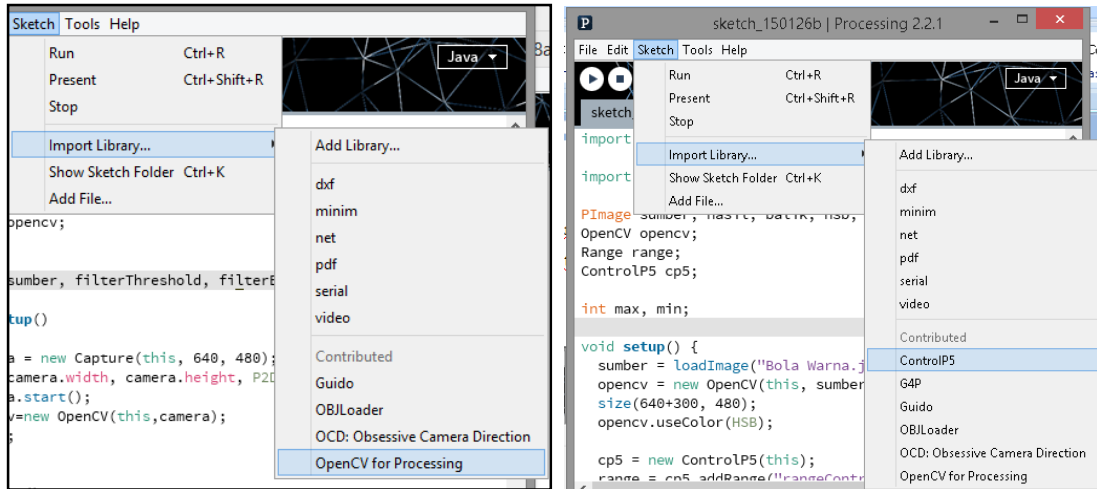
### C. Alat dan Bahan

- 1 set PC atau Notebook.
- Program Processing.

- Library OpenCV for Processing dan ControlP5 yang telah terunduh.

#### D. Langkah Kerja

1. Baca dan pahami petunjuk praktikum sebelum melakukan praktikum. Pastikan semua alat dan bahan sudah terpenuhi. Hubungi pengajar ketika terdapat masalah atau error.
2. Buat sketsa baru pada program Processing.
3. Import library ControlP5 dan OpenCV.



**Gambar 3.5 Import Library ControlP5 dan OpenCV for Processing**

4. Buat inisialisasi berikut.

```
PImage sumber, hsb, hasil, akhir;  
OpenCV opencv;  
Range range;  
ControlP5 scroll;  
int max, min;
```

5. Masukkan 3 gambar dengan nama “Bola.jpg”, “Bola Warna.jpg” dan “Pelangi.jpg” yang telah disediakan ke dalam folder sketch kita.



Gambar 3.5 Masukkan Gambar ke Folder Sketch

6. Buat setting project sesuai dengan kode berikut.

```
void setup() {
  sumber = loadImage("Bola Warna.jpg");
  opencv = new OpenCV(this, sumber);
  size(640, 480+40);
  background(#005250);
  opencv.useColor(HSB);

  scroll = new ControlP5(this);
  range = scroll.addRange("Min, Range, Max")
    .setBroadcast(false)
    .setPosition(640/4, 490)
    .setSize(320, 20)
    .setHandleSize(10)
    .setRange(0, 255)
    .setRangeValues(50, 100)
    .setBroadcast(true)

  ;

  println("-----");
  println("|      Filter Warna pada OpenCV      |");
  println("|  Praktikum Pemrosesan Sinyal @G4.206  |");
  println("|      Universitas Negeri Malang      |");
  println("-----");
}
```

7. Masukkan kode berikut pada void draw()

```
void draw() {
    opencv.loadImage(sumber);
    opencv.useColor(HSB);
    hsb=opencv.getSnapshot();

    opencv.setGray(opencv.getH().clone());
    opencv.inRange(min, max);
    hasil=opencv.getSnapshot();

    opencv.blur(12);
    opencv.threshold(50);
    opencv.invert();
    opencv.dilate();
    opencv.erode();
    akhir=opencv.getSnapshot();

    image(sumber, 0, 0, 640/2, 480/2);
    image(hsb, 640/2, 0, 640/2, 480/2);
    image(hasil, 0, 480/2, 640/2, 480/2);
    image(akhir, 640/2, 480/2, 640/2, 480/2);

    fill(128);
    text("Sumber", 640/2 - 100, 20 );
    text("HSB", 640 - 100, 20 );
    text("Hasil Filter", 640/2 - 100, 480/2 + 20 );
    text("Hasil Akhir", 640 - 150, 480/2 + 20 );
    text("Nama: Yogi Dwi Mahandi", 640 - 150, 480/2 + 40 );
}

void controlEvent(ControlEvent theControlEvent) {
    if (theControlEvent.isFrom("Min, Range, Max")) {
        min = int(theControlEvent.getController().getArrayValue(0));
        max = int(theControlEvent.getController().getArrayValue(1));
    }
}
```

8. Run sketch

9. Ubah nilai min, range, dan max untuk memfilter warna yang di kehendaki.

10. Masukkan pada Tabel 3.1.

11. Ubah mode filter menggunakan Saturation dan Value. Atur sendiri nilai min, range, dan max nya, lalu amati apa perbedaan ketiga filter tersebut.

12. Masukkan pada Tabel 3.2.

**E. Hasil Pengamatan**

**Tabel 3.1 Hasil Pengamatan Min, Range, Max Bola**

No	Warna Bola	Min	Range	Max
1	Merah			
2	Oranye			
3	Hijau muda			
4	Hijau Tua			
5	Biru			

**Tabel 3.2 Hasil Pengamatan Perbedaan Filter HSV**

No	Filter	Hasil Pengamatan
1	Hue	
2	Saturation	
3	Value	

**F. Analisa Data**

1. Berdasarkan Tabel 3.1, mengapa bola hijau muda dan hijau tua sulit untuk dibedakan?
2. Mengapa filter saturation tidak dapat membedakan warna bola? Lalu apa kegunaan filter saturation?
3. Gunakan gambar “Pelangi.jpg” dan filter saturation. Mengapa warna putih pada tengah-tengah gambar lebih dominan tampak jika dibandingkan dengan warna lainnya?

**G. Tugas**

1. Buatlah komposisi filter HSV yang tepat untuk dapat mendeteksi bola oranye pada gambar “Bola.jpg” dan sertakan cuplikan gambarnya.

**H. Kesimpulan**

.....  
.....  
.....