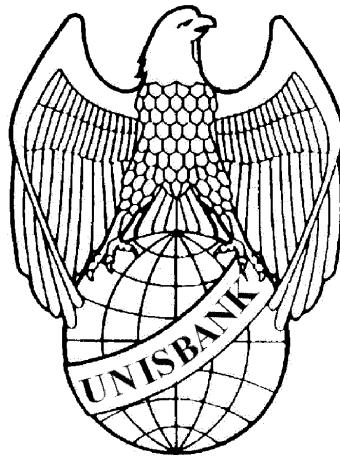


**SISTEM DETEKSI JENIS BUNGA MENGGUNAKAN NILAI HSV DARI  
CITRA MAHKOTA BUNGA**

Tugas Akhir disusun untuk memenuhi syarat  
mencapai gelar Kesarjanaan Komputer pada  
Program Studi Teknik Informatika  
Jenjang Program Strata 1



Oleh :

**Rizqa Puji Rakhmawati**

**08.01.53.0185**

**9983**

**FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS STIKUBANK (UNISBANK)**

**SEMARANG**

**2013**

## **PERNYATAAN KESIAPAN UJIAN TUGAS AKHIR**

Saya, Rizqa Puji Rakhmawati, dengan ini menyatakan bahwa Laporan Tugas Akhir yang berjudul “SISTEM DETEKSI JENIS BUNGA BERDASARKAN NILAI HSV DARI CITRA MAHKOTA BUNGA”

Adalah benar hasil karya saya dan belum pernah diajukan sebagai karya ilmiah, sebagian atau seluruhnya, atas nama saya atau pihak lain.

( Rizqa Puji Rakhmawati )

NIM : 08.01.53.0185

Disetujui oleh pembimbing :

Kami setuju Laporan tersebut diajukan untuk Ujian Tugas Akhir

Semarang : Februari 2013

(Eddy Nurraharjo,ST, M.Cs)

Pembimbing I

Semarang : Februari 2013

(Wiwien Hadikurniawati,ST, M.Kom)

Pembimbing II

## HALAMAN PENGESAHAN

Telah dipertahankan di depan tim dosen penguji Skripsi Fakultas Teknologi Informasi UNIVERSITAS STIKUBANK (UNISBANK) Semarang dan diterima sebagai salah satu syarat guna menyelesaikan Jenjang Program Strata 1, Program Studi Teknik Informatika.

Semarang, Februari 2013

Ketua

(Eddy Nurraharjo, S.T, M.Cs)

Sekretaris

(Wiwien Hadikurniawati, S.T, M.Kom)

Anggota

(Eka Ardhianto, S.Kom, M.Cs)

Mengetahui :

UNIVERSITAS STIKUBANK (UNISBANK) SEMARANG

Fakultas Teknologi Informasi

Dekan

(Dwi Agus Diartono, S.Kom, M.Kom)

## MOTTO DAN PERSEMBAHAN

### MOTTO

- Jangan tunggu bahagia baru kau bersyukur, bersyukurlah maka kau akan bahagia.
- Hai orang-orang yang beriman, Jadikanlah sabar dan shalatmu sebagai penolongmu, sesungguhnya Allah beserta orang-orang yang sabar” (Al-Baqarah: 153)
- Benar menurutmu belum tentu benar di mata orang lain, Jadi berusaha lah menjadi yang terbaik. (RizqaPuji)

### PERSEMBAHAN

- ❖ Allah SWT, yang telah memberikan ilmu kepada penulis.
- ❖ H. Mudlofir, S.Kom dan Hj. Masruroh orang tuaku tercinta yang tidak pernah putus untuk memberikan doa dan semangat.
- ❖ Afrizal Abdi M dan Habib Maulana, kedua adikku tersayang yang sudah memberikan support dan doanya.
- ❖ Keluarga besar H.Dimyati (Alm)-Maenah (Alm) dan Makful-Rastoni (Alm).
- ❖ Dyah, Sasa, Yani, Yati, Cik Yauw Liani yang selalu memberikan dukungan serta doa dalam menyelesaikan tugas akhir ini. Thanks 😊
- ❖ Seseorang yang namanya masih tertulis di Lauhful Mahjudz. Semoga kita cepat dipertemukan dan dipersatukan.

**FAKULTAS TEKNOLOGI INFORMASI**  
**UNIVERSITAS STIKUBANK (UNISBANK) SEMARANG**

Program Studi : Teknik Informatika

Tugas Akhir Sarjana Komputer

Semester Ganjil Tahun 2013

**SISTEM DETEKSI JENIS BUNGA MENGGUNAKAN NILAI HSV**  
**DARI CITRA MAHKOTA BUNGA**

**Rizqa Puji Rakhmawati**

**08.01.53.0185**

**ABSTRAKSI**

Warna adalah salah satu atribut yang berperan dalam mengidentifikasi objek tertentu. Pada penelitian ini model warna yang digunakan adalah model warna HSV yaitu citra (RGB) dikonversi kedalam HSV untuk diketahui nilai mean dari HSV yang digunakan sebagai acuan dalam klasifikasi bunga. Rata – rata keberhasilan sistem untuk mendeteksi objek dengan menggunakan nilai HSV adalah dengan acuan nilai Hue sebesar 36.68%, acuan nilai Saturation sebesar 26.67 %, acuan nilai Value sebesar 20 %. Gabungan nilai HS sebesar 26.67%, nilai HV sebesar 33,33%, nilai SV 20% dan HSV sebesar 13.33%. tidak terdeteksi.

**Kata Kunci** : Model warna HSV, Klasifikasi

Semarang, Februari 2013

Pembimbing I

Pembimbing II

(Eddy Nurraharjo,S.T, M.Cs)

(Wiwien Hadikurniawati,S.T,M.Kom)

## KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah SWT, atas limpahan rahmat dan hidayahNya sehingga penulis dapat menyelesaikan Skripsi ini yang berjudul “Sistem Deteksi Jenis Mahkota Bunga Menggunakan Nilai HSV dari Citra Mahkota Bunga”. Penulisan Skripsi ini disusun sebagai pelaksanaan tugas akhir yang merupakan salah satu persyaratan kelulusan pada program studi Teknik Informatika di Universitas Stikubank Semarang.

Menyadari bahwa dalam penyelesaian penulisan skripsi ini penulis tidak dapat berbuat banyak tanpa bantuan pihak lain, dalam kesempatan ini penulis mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Allah SWT Tuhan semesta alam, Maha Suci, Maha Agung dan Maha segalanya.
2. Bapak Dr. Bambang Suko Priyono, M.M selaku Rektor Universitas Stikubank Semarang.
3. Bapak Dwi Agus Diartono, S.Kom, M.Kom selaku Dekan Fakultas Teknologi Informasi Universitas Stikubank.
4. Ibu Dewi Handayani, S.Kom selaku Kepala Program Studi S1 Teknik Informatika.
5. Bapak Eddy Nurraharjo, ST, M.Cs selaku Dosen Pembimbing I, yang telah memberikan bimbingan, masukan serta waktunya dalam penyusunan skripsi ini.

6. Ibu Wiwien Hadikurniawati,ST, M.Kom selaku Dosen Pembimbing II, yang telah memberikan bimbingan dan masukannya.
7. Ibu Dewi Handayani, S.Kom selaku Dosen Wali, yang telah menjadi ibu wali Fakultas Teknologi Informasi Jurusan Teknik Informatika Universitas Stikubank Semarang.
8. Dosen – dosen Program Studi Teknik Informatika yang telah memberikan ilmunya kepada saya selama saya kuliah di Universitas Stikubank Semarang.
9. Semua teman-teman di Fakultas Teknologi Informasi jurusan Teknik Informatika Universitas Stikubank Semarang yang selalu mendukung.

Penulis menyadari bahwa dalam penyusunan penulisan skripsi ini masih jauh dari kata sempurna dan masih banyak kekurangan, namun demikian penulis telah berusaha sekuat tenaga untuk menyelesaikan penulisan skripsi ini.

Akhir kata, semoga penulisan skripsi ini dapat bermanfaat bagi teman-teman mahasiswa pada khususnya dan pembaca pada umumnya, serta semoga dapat bermanfaat bagi perkembangan teknologi pengolahan citra.

Semarang,

Penulis

Rizqa Puji Rakhmawati

## DAFTAR ISI

<b>HALAMAN JUDUL.....</b>	<b>i</b>
<b>HALAMAN PERSETUJUAN.....</b>	<b>ii</b>
<b>HALAMAN PENGESAHAN.....</b>	<b>iii</b>
<b>ABSTRAK.....</b>	<b>iv</b>
<b>MOTTO DAN PERSEMBAHAN.....</b>	<b>v</b>
<b>KATA PENGANTAR.....</b>	<b>vi</b>
<b>DAFTAR ISI.....</b>	<b>viii</b>
<b>DAFTAR GAMBAR.....</b>	<b>xi</b>
<b>DAFTAR TABEL.....</b>	<b>xiii</b>
<b>BAB I    PENDAHULUAN.....</b>	<b>1</b>
1.1    Latar Belakang Masalah.....	1
1.2    Perumusan Masalah.....	2
1.3    Batasan Masalah.....	3
1.4    Tujuan Penelitian .....	3
1.5    Manfaat Penelitian.....	3
1.6    Metodologi Penelitian.....	4
1.7    Sistematika Penulisan.....	6
<b>BAB II    TINJAUAN PUSTAKA.....</b>	<b>8</b>
2.1    Pengenalan Telur Berdasarkan Karakteristik Warna Citra .....	8
2.2    Pengenalan Buah Berdasarkan Karakteristik Warna Citra .....	8



2.3	Deteksi Dan Klasifikasi Citra Berdasarkan Warna Kulit Menggunakan HSV .....	9
2.4	Klasifikasi Tingkat Kematangan Varietas Tomat Merah Dengan Metode Perbandingan Kadar Warna .....	9
<b>BAB III</b>	<b>LANDASAN TEORI.....</b>	<b>11</b>
3.1	Pengolahan Citra .....	11
3.1.1.	Operasi-Operasi Pengolahan Citra.....	12
3.2	Citra Digital .....	14
3.2.1.	Jenis – jenis Citra Digital.....	16
3.3	Model Warna HSV.....	18
3.4	Konversi Citra.....	20
3.4.1.	Konversi RGB ke HSV.....	20
3.5	Treshold.....	22
3.6	DFD (Data Flow Diagram) .....	22
3.6.1.	Terminator / External Entity.....	23
3.6.2.	Proses.....	24
3.6.3.	Data Store.....	24
3.6.4.	Alur Data.....	25
3.7	Matlab.....	27
3.8	Flowchart .....	30
3.9	Variabel Formulasi Data.....	31
<b>BAB IV</b>	<b>PERANCANGAN SISTEM.....</b>	<b>34</b>

4.1	Kebutuhan Sistem .....	34
4.1.1	Perangkat Keras (Hardware).....	34
4.1.2	Sistem Akuisisi Data.....	34
4.1.3	Perangkat Lunak (Software).....	35
4.2	Perancangan Sistem.....	36
4.2.1	DFD Sistem Deteksi Jenis Bunga.....	36
4.2.2	Flowchart .....	40
4.2.3	Perancangan Antarmuka (Interface) .....	43
<b>BAB V</b>	<b>IMPLEMENTASI SISTEM.....</b>	<b>45</b>
5.1.	Implementasi Sistem.....	45
5.1.1.	Sistem Akuisisi Data.....	45
5.1.2.	Antarmuka Sistem.....	46
5.2.	Pelatihan Sistem.....	50
5.2.1.	Hasil Pelatihan .....	51
5.3.	Pengujian Sistem.....	61
5.3.1.	Hasil Pengujian .....	75
<b>BAB VI</b>	<b>PENUTUP.....</b>	<b>85</b>
6.1.	Kesimpulan .....	85
6.2.	Saran.....	85

**DAFTAR PUSTAKA**

**LAMPIRAN – LAMPIRAN**

## DAFTAR GAMBAR

Gambar 3.1	Proses Pengolahan Citra.....	11
Gambar 3.2	Citra Biner.....	16
Gambar 3.3	Representasi Citra Biner.....	16
Gambar 3.4	Spektrum Warna.....	18
Gambar 3.5	Model Warna HSV.....	19
Gambar 3.6	Skala Hue.....	20
Gambar 3.7	Komponen Data Flow Diagram.....	23
Gambar 4.1	Sistem Akuisisi Data.....	35
Gambar 4.2	DFD Level 0.....	36
Gambar 4.3	DFD Level 1.....	37
Gambar 4.4	DFD Level 2 Proses Latih.....	37
Gambar 4.5	DFD Level 2 Proses Uji.....	39
Gambar 4.6	Flowchart Latih.....	40
Gambar 4.7	Flowchart Uji .....	42
Gambar 4.8	Rancangan Interface.....	43
Gambar 5.1	Sistem Akuisisi Data.....	46
Gambar 5.2	Antarmuka Sistem.....	46
Gambar 5.3	Listing Program Latih.....	47
Gambar 5.4	Listing Menampilkan nilai mean HSV Latih .....	48
Gambar 5.5	Listing Menampilkan Nilai.....	48
Gambar 5.6	Listing Tombol Hapus.....	49

Gambar 5.7	Listing Tombol Reset.....	49
Gambar 5.8	Listing Tombol Keluar.....	49
Gambar 5.9	Tampilan Program latihan.....	50
Gambar 5.10	Tampilan Program Latih.....	51
Gambar 5.11	Tampilan Uji pada Button H.....	62
Gambar 5.12	Tampilan Hasil Uji dengan Acuan Nilai H.....	63
Gambar 5.13	Listing Program pada Button H.....	63
Gambar 5.14	Tampilan Hasil Uji dengan Acuan Nilai S.....	64
Gambar 5.15	Listing Program pada Button S.....	65
Gambar 5.16	Tampilan Hasil Uji dengan Acuan Nilai V.....	66
Gambar 5.17	Listing Program pada Button V.....	66
Gambar 5.18	Tampilan Hasil Uji dengan Acuan Nilai HS.....	67
Gambar 5.19	Listing Program pada Button HS.....	68
Gambar 5.20	Tampilan Hasil Uji dengan Acuan Nilai HV.....	69
Gambar 5.21	Listing Program pada Button HV.....	70
Gambar 5.22	Tampilan Hasil Uji dengan Acuan Nilai SV.....	71
Gambar 5.23	Listing Program pada Button SV.....	72
Gambar 5.24	Tampilan Hasil Uji dengan Acuan Nilai HSV.....	73
Gambar 5.25	Listing Program pada Button HSV.....	74

## DAFTAR TABEL

Tabel 3.1	Simbol – simbol Flowchart.....	30
Tabel 5.1	Tabel hasil latih citra bunga Alamanda .....	51
Tabel 5.2	Tabel hasil latih citra bunga Dahlia.....	53
Tabel 5.3	Tabel hasil latih citra bunga Kamboja.....	54
Tabel 5.4	Tabel hasil latih citra bunga Mawar .....	55
Tabel 5.5	Tabel hasil latih citra bunga Sepatu.....	56
Tabel 5.6	Tabel Data Mean HSV.....	58
Tabel 5.7	Tabel Data Kadar HSV .....	58
Tabel 5.8	Tabel Hasil Uji Bunga Alamanda .....	75
Tabel 5.9	Tabel Hasil Uji Bunga Dahlia .....	77
Tabel 5.10	Tabel Hasil Uji Bunga Kamboja .....	79
Tabel 5.11	Tabel Hasil Uji Bunga Mawar .....	80
Tabel 5.12	Tabel Hasil Uji Bunga Sepatu .....	82
Tabel 5.13	Tabel Rata-rata Presesntase Keberhasilan .....	84

# BAB I

## PENDAHULUAN

### 1.1 LATAR BELAKANG

Perkembangan teknologi pengolahan citra (image processing) sekarang ini menyediakan kemungkinan manusia untuk membuat suatu sistem yang dapat mengenali suatu citra digital. Pengolahan citra merupakan salah satu jenis teknologi untuk menyelesaikan masalah mengenai pemrosesan gambar. Dalam pengolahan citra, gambar diolah sedemikian rupa sehingga gambar tersebut dapat digunakan untuk aplikasi lebih lanjut.

Warna adalah salah satu atribut yang berperan dalam mengidentifikasi objek tertentu, pemrosesan warna termasuk didalamnya adalah ekstraksi informasi tentang *spectral properties* dari permukaan objek dan mencari kesamaan terbaik dari sekumpulan deskripsi yang telah diketahui untuk melakukan pengenalan.

Warna dari mahkota bunga antara bunga yang satu dengan bunga yang lain memiliki perbedaan. Inilah yang merupakan ciri khas tersendiri yang dimiliki oleh setiap bunga. Setiap mahkota bunga memiliki warna yang berbeda-beda, tidak semua mahkota bunga memiliki warna yang sama meskipun dalam satu jenis.

Sebelumnya sudah pernah dilakukan beberapa penelitian yang menyangkut dengan metode yang serupa, diantaranya :

- a. Agus Purwo Handoko, Yustina Retno Wahyu Utami. *Pengenalan Buah Berdasarkan Karakteristik Warna Citra*. STIMIK Sinar Nusantara Surakarta. 2009. Dalam penelitian ini, akan dikenali beberapa jenis buah

berdasarkan karakteristik warna (hue dan saturasi), untuk mengklasifikasi ciri digunakan menggunakan metode *K-Means*.

- b. Yustina Retno Wahyu Utami. *Pengenalan Telur Berdasarkan Karakteristik Warna Citra*. STIMIK Sinar Nusantara Surakarta. 2009. Penelitian ini menggunakan sampel telur sebagai objeknya dengan mengambil ciri citra berdasarkan karakteristik warna obyek pada citra HSV, ciri citra diklasifikasikan dengan menggunakan metode *Minimum Distance Classifier*.

Hal inilah yang melatarbelakangi penulis untuk mengetahui lebih jelas dan mengembangkan bahasa pemrograman matlab dengan membuat suatu aplikasi yang dapat mendeteksi jenis bunga berdasarkan warna. Sehingga gambaran diatas menjadi suatu pertimbangan bagi penulis untuk membuat judul “**SISTEM DETEKSI JENIS BUNGA MENGGUNAKAN NILAI HSV DARI CITRA MAHKOTA BUNGA**”.

## **1.2 PERUMUSAN MASALAH**

Berdasarkan latar belakang permasalahan yang telah di uraikan di atas maka dapat di ambil rumusan yang akan menjadi pembahasan penelitian ini yaitu :

Bagaimana citra mahkota bunga dapat dikenali oleh pengolahan citra dengan menggunakan nilai hsv.

### **1.3 BATASAN MASALAH**

Dalam penelitian ini agar tidak menyimpang dari maksud dan tujuan penyusunan skripsi ini juga mengingat adanya keterbatasan waktu penelitian maka dalam penyusunan skripsi hanya membatasi masalah pada:

1. Jenis bunga yang akan di teliti adalah bunga alamanda, dahlia, kamboja, mawar, sepatu.
2. Hardware yang digunakan dalam penelitian ini menggunakan media webcam resolusi 1.3 MP.
3. Jenis citra yang digunakan adalah citra berwarna, dengan format JPEG.
4. Citra yang diambil merupakan citra tunggal.

### **1.4 TUJUAN PENELITIAN**

Tujuan penelitian yang ingin dicapai penulis adalah menghasilkan suatu sistem yang dapat digunakan untuk mendeteksi jenis bunga berdasarkan warna dengan menggunakan nilai HSV.

### **1.5 MANFAAT PENELITIAN**

Adapun manfaat yang diharapkan dalam penulisan skripsi ini adalah:

1. Bagi Penulis

Sebagai sarana untuk menerapkan metode HSV dalam pengolahan citra yang berkaitan dengan pembuatan sistem deteksi jenis bunga pada citra mahkota bunga.



## 2. Bagi Akademik

Sebagai bahan referensi baru yang dapat dipergunakan untuk penelitian berikutnya.

## 3. Bagi Pembaca

Dengan adanya penelitian ini diharapkan dapat memberikan wawasan dalam melakukan deteksi pada citra mahkota bunga.

## **1.6 METODOLOGI PENELITIAN**

Metode yang akan digunakan dalam penelitian ini terdiri dari langkah-langkah berikut:

### **1.6.1 Obyek Penelitian**

Obyek yang diteliti dalam penelitian ini adalah citra mahkota bunga yang terdapat pada bunga alamanda, dahlia, kamboja, mawar, sepatu.

### **Metode Pengumpulan Data**

Metode pengumpulan data dalam penelitian ini adalah sebagai berikut :

#### 1. Studi Pustaka

Merupakan metode pengumpulan data dengan cara mengumpulkan data-data dari berbagai sumber yang mendukung penelitian baik itu dari buku, majalah, jurnal ilmiah, maupun artikel lainnya yang mendukung penelitian.

## 2. Observasi

Merupakan metode pengumpulan data dengan cara melakukan pengamatan secara langsung pada obyek yang diteliti.

## 3. Internet

Selain dari media konvensional, peneliti akan mencari di media internet sebagai penunjang penelitian ini.

Sedangkan data yang digunakan :

### a. Data primer

Data yang diperoleh langsung dari sumber atau objek penelitian melalui observasi dan interview.

### b. Data sekunder

Data yang diperoleh dari membaca buku – buku sebagai pelengkap untuk menganalisa masalah dalam penelitian ini.

## 1.6.2 Metode Pengembangan Sistem

Metode yang digunakan untuk membuat atau mengembangkan piranti lunak ini adalah model *Prototype* (Pressman,2002). Metode ini merupakan metode pengembangan sistem di mana hasil analisa perbagian langsung ditetapkan ke dalam sebuah metode tanpa harus menunggu seluruh sistem selesai dianalisa. Adapun tahap-tahap dalam metode ini adalah:

1. Analisa

Pada tahap ini kegiatan yang dilakukan adalah menganalisa keperluan yang terdapat pada masalah yang ada serta identifikasi obyektif keseluruhan dari perangkat lunak yang dibutuhkan.

2. Perancangan Sistem

Pada tahap ini kegiatan yang dilakukan adalah merancang model atau *prototype* dari permasalahan yang ada.

3. Evaluasi

Pada tahap ini merupakan kegiatan evaluasi terhadap *prototype* atau model yang sudah dibuat. Apabila ada bagian-bagian yang tidak sesuai dengan keinginan maka perlu diubah. *Prototype* tersebut dievaluasi oleh pemakai dan dipakai untuk menyaring kebutuhan pengembang perangkat lunak. Iterasi yang terjadi pada saat *Prototyping* memungkinkan pengembang untuk mengetahui serta memenuhi keinginan dan kebutuhan pemakai.

4. Hasil

Pada tahap ini merupakan hasil dari *Prototyping* atau model akhir yang telah dibuat sesuai dengan yang diinginkan.

## 1.7 SISTEMATIKA PENULISAN

Sistematika penulisan laporan penelitian ini adalah sebagai berikut:

**Bab I   Pendahuluan**

Berisi tentang Latar Belakang, Perumusan Masalah, Batasan Masalah, Tujuan Penelitian, Manfaat Penelitian, Metodologi Penelitian dan Sistematika Penulisan.

**Bab II   Tinjauan Pustaka**

Berisi tentang abstraksi, simpulan, hasil, serta metode penelitian sebelumnya.

**Bab III   Landasan Teori**

Meliputi uraian teori yang menunjang penelitian yang akan digunakan sebagai dasar. Diantaranya adalah mengenai Pengolahan Citra, Citra Digital, Model Warna HSV, Treshold, Data Flow Diagram dan Bahasa Pemograman Matlab.

**Bab IV   Perancangan Sistem**

Bab ini berisi tentang perancangan kebutuhan sistem yang akan digunakan dalam mendesain program aplikasi ini.

**Bab V   Implementasi Sistem**

Berisi tentang implementasi perancangan sistem dan implementasi perancangan antarmuka yang telah di desain sebelumnya.

**Bab VI   Penutup**

Dalam bab ini berisi tentang kesimpulan dan saran yang berhubungan dengan permasalahan yang telah dibahas serta tindakan yang harus di ambil atas hasil penelitian.

## BAB II

### TINJAUAN PUSTAKA

**2.1** Yustina Retno Wahyu Utami, 2009, **“Pengenalan Telur Berdasarkan Karakteristik Warna Citra”**, menjelaskan tentang pengenalan obyek pada citra berdasarkan karakteristik warna obyek pada citra HSV dengan menggunakan metode *minimum distance classifier*. Dalam penelitiannya obyek yang akan digunakan sebagai sampel adalah obyek telur, dimana obyek ini akan dibedakan berdasarkan warnanya. Telur yang akan digunakan sebagai sampel adalah telur ayam negeri, ayam kampung dan bebek. Pada akuisisi citra, citra yang digunakan berupa citra berwarna 24 bit. Citra ini memiliki komponen red, green dan blue (RGB). Citra RGB ditransformasikan ke model warna HSV (Hue Saturation Value), dimana pada citra HSV ini akan dilakukan ekstraksi ciri. Dari proses ekstraksi ciri digunakan untuk mengklasifikasikan citra ke dalam 3 jenis telur yaitu Telur bebek, Telur ayam negeri dan Telur ayam kampung. Metode pengklasifikasian yang digunakan adalah *Minimum Distance Classifier* yaitu metode untuk mengelaskan suatu objek berdasarkan data latih. Data yang sudah dilatih ditentukan kelas apa, kemudian data baru dikelaskan dengan menggunakan jarak terpendek Euclidean.

**2.2** Agus Purwo Handoko, Yustina Retno Wahyu Utami, 2009, **“Pengenalan Buah Berdasarkan Karakteristik Warna Citra”**, klasifikasi obyek dapat

dilakukan dengan menggunakan metode *K-means Clustering*. Citra obyek ditransformasikan ke model HSV. Dari citra hue dan saturasi, diperoleh cirri rata-rata dan standard deviasi dari hue dan saturasi. Citra tersebut dikelompokkan menggunakan metode *k-means clustering*. Citra akan dikelompokkan dalam kelas berdasarkan jarak terdekat dengan pusat kluster. Dalam penelitian ini terdapat 4 kelas obyek yaitu kelas apel, kelas pear, kelas jeruk dan kelas salak.

**2.3** Jati Sasongko Wibowo, 2011, “**Deteksi Dan Klasifikasi Citra Berdasarkan Warna Kulit Menggunakan HSV**”, menjelaskan tentang segmentasi warna kulit manusia untuk mengklasifikasi citra pornografi. Sample yang digunakan dalam penelitian ini adalah citra warna kulit campuran warna kulit Eropa dan Asia. Pendeteksian dilakukan dengan membandingkan pixel sample image warna kulit dengan image yang akan dideteksi sehingga diketahui jumlah pixel warna kulit pada tiap gambar. Rentang pixel yang digunakan dalam penelitian antara 0%- 20 %, 21%- 50%, dan 51% - 100%. Hasil dari deteksi dikelompokkan dalam tiga kategori yaitu gambar bukan pornografi, gambar semi pornografi dan gambar pornografi.

**2.4** Ary Noviyanto, 2010, “**Klasifikasi Tingkat Kematangan Varietas Tomat Merah Dengan Metode Perbandingan Kadar Warna**”, menjelaskan tentang perbandingan kadar warna dalam pengklasifikasian varietas buah tomat merah berdasarkan tingkat atau tahapan kematangan. Buah tomat di capture dengan kamera kemudian diambil sebagian (cropping) untuk diekstrak cirri warnanya, dan dikalkulasi tingkat kadar warna dari R(red), G (green) dan B

(blue). Tingkat kematangan diklasifikasikan menjadi 3, yaitu *Ripe* (matang), *Half-Ripe* (setengah matang), dan *Un-Ripe* (tidak matang). Dari pengujian yang dilakukan total keberhasilan sistem mencapai 95%, dengan kemungkinan error 5%. Faktor pencahayaan mempengaruhi akurasi dari metode ini karena perbedaan intensitas pencahayaan.

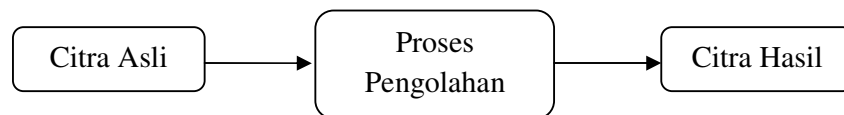
## **BAB III**

### **LANDASAN TEORI**

#### **3.1. PENGOLAHAN CITRA**

Pengolahan citra adalah kegiatan memperbaiki citra agar mudah diinterpretasi oleh manusia atau mesin (komputer). Inputnya adalah citra dan outputnya citra tetapi dengan kualitas lebih baik dari pada citra masukan, misal citra warnanya kurang tajam, kabur (bluring), mengandung noise dan lain-lain sehingga perlu ada pemrosesan untuk memperbaiki citra karena citra tersebut menjadi sulit diinterpretasikan karena informasi yang disampaikan menjadi kurang.

Pengolahan citra bertujuan memperbaiki kualitas agar mudah diinterpretasi oleh manusia atau mesin (dalam hal ini komputer). Teknik pengolahan citra mentransformasikan citra menjadi citra lain. Jadi, masukannya adalah citra dan keluarannya juga citra, namun citra keluarannya harus mempunyai kualitas lebih baik daripada citra masukan.



**Gambar 3.1** Proses Pengolahan Citra

Peningkatan kualitas citra bertujuan menghasilkan citra dengan kualitas yang lebih baik dibandingkan dengan citra semula. Analisis citra bertujuan mengidentifikasi parameter yang di asosiasikan dengan ciri dari objek dalam citra,



untuk selanjutnya parameter tersebut digunakan dalam menginterpretasikan citra.

Analisis citra pada dasarnya terdiri dari tiga tahapan, yaitu:

1. Ekstraksi Citra
2. Segmentasi
3. Klasifikasi

Faktor kunci dalam mengekstraksi citra adalah kemampuan mendeteksi keberadaan tepi di dalam citra. Ada beberapa metode deteksi tepi. Penggunaan metode deteksi tepi yang tidak tepat, akan menghasilkan pendeteksi tepian yang gagal. Pendeteksian tepi merupakan tahapan untuk melingkupi informasi di dalam citra. Tepi mencirikan batas objek dan karena itu tepi berguna untuk proses segmentasi dan identifikasi objek di dalam citra (Munir,2004).

Umumnya, operasi-operasi pada pengolahan citra diterapkan pada citra bila (JAI89) :

1. Perbaiki atau modifikasi citra untuk meningkatkan kualitas visual atau menonjolkan beberapa aspek informasi yang terkandung dalam citra.
2. Elemen di dalam citra perlu di kelompokkan, dicocokkan atau diukur.
3. Sebagian citra perlu digabung dengan bagian citra yang lain.

### **3.1.1. Operasi Pengolahan Citra**

Operasi – operasi yang dilakukan pengolahan citra banyak ragamnya. Secara umum, operasi pengolahan citra dapat diklasifikasikan dalam beberapa jenis, yaitu:

- a. Perbaiki Kualitas Citra (Image Enhancement)

Jenis operasi ini bertujuan untuk memperbaiki kualitas citra dengan cara memanipulasi parameter-parameter citra. Dengan operasi ini ciri-ciri khusus yang terdapat pada citra lebih ditonjolkan. Yang termasuk dalam klasifikasi ini antara lain :

- ❖ Perbaikan kontras gelap / terang (contrast enhancement)
- ❖ Perbaikan tepian objek (edge enhancement)
- ❖ Penajaman (sharpening)
- ❖ Pemberian warna semu (pseudocoloring)
- ❖ Penyaringan derau (noise filtering)

b. Pemugaran Citra (Image Restoration)

Operasi ini bertujuan untuk menghilangkan / meminimumkan cacat pada citra. Tujuan pemugaran citra hampir sama dengan perbaikan. Yang termasuk dalam klasifikasi ini antara lain:

- ❖ Penghilang kesamaran (debluring)
- ❖ Penghilang derau (noise)

c. Pemampatan Citra (Image Compression)

Operasi ini bertujuan untuk memampatkan citra sehingga memori yang dibutuhkan untuk menyimpan citra lebih kecil, tetapi hasil citra yang telah dimampatkan tetap memiliki kualitas gambar yang bagus.

d. Segmentasi Citra (Image Segmentation)

Operasi ini bertujuan untuk memecah suatu citra ke dalam beberapa segmen dengan suatu kriteria tertentu. Jenis operasi ini erat kaitannya dengan pengenalan pola.

e. Pergorakan Citra (Image Analysis)

Operasi ini bertujuan untuk menghitung besaran kuantitatif citra untuk menghasilkan deskripsinya. Teknik analisis citra mengekstraksi ciri-ciri tertentu yang membantu dalam identifikasi objek. Proses segmentasi kadangkala diperlukan untuk melokalisasi objek yang diinginkan dari sekelilingnya. Yang termasuk dalam klasifikasi ini antara lain:

- ❖ Pendeteksian tepi (edge detection)
- ❖ Ekstraksi batas (boundary)
- ❖ Representasi daerah (region)

f. Rekonstruksi Citra (Image Reconstruction)

Operasi ini bertujuan untuk membentuk ulang objek dari beberapa citra hasil proyeksi. Operasi rekonstruksi citra banyak digunakan dalam bidang medis. Contohnya adalah foto rontgen dan sinar X yang digunakan untuk membentuk ulang gambar organ tubuh.

### 3.2. CITRA DIGITAL

Citra (image) merupakan fungsi kontinu dari intensitas cahaya pada bidang dua dimensi. Secara matematis fungsi intensitas cahaya pada bidang dua dimensi dinotasikan sebagai  $f(x,y)$ , dimana  $(x,y)$  merupakan koordinat pada bidang dua dimensi dan  $f(x,y)$  merupakan intensitas cahaya pada bidang dua dimensi (Munir,2004).

Sedangkan menurut Fahmi (2007: II-1) citra digital adalah citra kontinu yang diubah dalam bentuk diskrit, baik koordinat ruang maupun intensitas cahayanya. Pengolahan digitalisasi terdiri dari dua proses, yaitu pencuplikan (*sampling*) posisi, dan kuantisasi intensitas. Citra digital dapat dinyatakan dalam matriks dua dimensi  $f(x,y)$  dimana 'x' dan 'y' merupakan koordinat piksel dalam matriks dan 'f' merupakan derajat intensitas piksel tersebut. Citra digital berbentuk matriks dengan ukuran M x N akan tersusun sebagai berikut :

$$f(x,y) = \begin{pmatrix} f(0,0) & f(0,1) & f(0,2) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & f(1,2) & \dots & f(1,N-1) \\ f(2,0) & f(2,1) & f(2,2) & \dots & f(2,N-1) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ f(M-1,0) & f(M-1,1) & f(M-1,2) & \dots & f(M-1,N-1) \end{pmatrix} \quad (3.1)$$

Suatu citra  $f(x,y)$  dalam fungsi matematis dapat dituliskan sebagai berikut :

$$0 \leq x \leq M-1$$

$$0 \leq y \leq N-1$$

$$0 \leq f(x,y) \leq G-1$$

dimana : M = banyaknya baris pada array citra

N = banyaknya kolom pada array citra

G = banyaknya skala keabuan (*graylevel*)

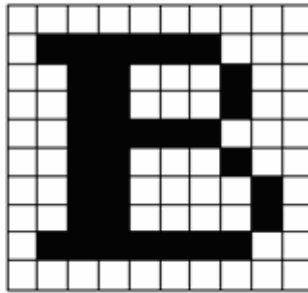
Interval (0,G) disebut skala keabuan (*grayscale*). Besar G tergantung pada proses digitalisasinya. Biasanya keabuan 0 (nol) menyatakan intensitas hitam dan G menyatakan intensitas putih untuk citra 8 bit, nilai G sama dengan  $2^8 = 256$  warna (derajat keabuan).

### 3.2.1. Jenis – jenis Citra Digital

Jenis – jenis citra digital adalah sebagai berikut :

#### 1. Citra Biner ( Monokrom)

Citra biner adalah citra yang hanya memiliki dua buah pixel yaitu hitam bernilai 0 dan putih bernilai 1. Oleh karena itu, setiap *pixel* pada citra biner cukup direpresentasikan dengan 1 bit.



**Gambar 3.2** Citra Biner

0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	0	0	0
0	0	1	1	0	0	0	1	0	0
0	0	1	1	0	0	0	1	0	0
0	0	1	1	1	1	1	0	0	0
0	0	1	1	0	0	0	1	0	0
0	0	1	1	0	0	0	0	1	0
0	0	1	1	0	0	0	0	1	0
0	1	1	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0

**Gambar 3.3** Representasi Citra Biner

#### 2. Citra Grayscale

Citra grayscale adalah citra yang nilai piksel-pikselya berada diantara 0 (hitam) dan 255 (putih). Proses awal yang banyak dilakukan dalam

pengolahan citra adalah mengubah citra berwarna menjadi citra grayscale, hal ini digunakan untuk menyederhanakan model citra.

Tujuan image grayscale antara lain :

- a. Mempresentasikan aras abu-abu (grayscale) atau kode warna.
- b. Kisaran nilai ditentukan oleh bit yang dipakai dan akan menunjukkan resolusi aras abu-abu (gray level resolution).
  1. 1 bit - 2 aras atau warna : [0,1]
  2. 4 bit – 16 aras atau warna : [0,15]
  3. 8 bit – 256 aras atau warna : [0,255]
  4. 24 bit – 16 juta warna (true color)
    - a. Kanal merah – red (R) : [0,255]
    - b. Kanal hijau – green (G) : [0,255]
    - c. Kanal biru – blue (B) : [0,255]

Warna abu-abu adalah satu-satunya warna pada ruang RGB dengan komponen merah, hijau dan biru mempunyai intensitas yang sama. Pada citra beraras keabuan hanya perlu menyatakan nilai intensitas untuk tiap pixel sebagai nilai tunggal, sedangkan pada citra berwarna perlu tiga nilai intensitas untuk tiap pikselnya.

Untuk mengubah citra berwarna yang mempunyai nilai matrik masing-masing r, g dan b menjadi citra grayscale dengan nilai s, maka konversi dapat dilakukan dengan mengambil rata-rata dari nilai r, g dan b sehingga dapat dituliskan menjadi :

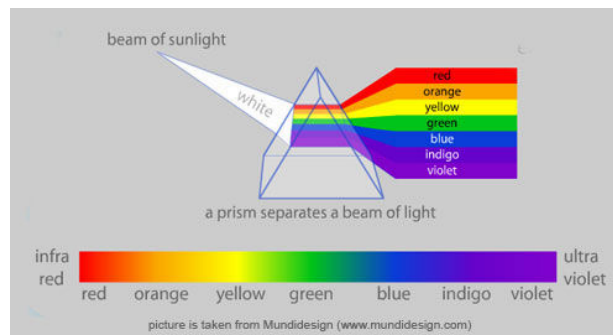
$$s = \frac{r+g+b}{3} \quad (3.2)$$

### 3. Citra Warna ( True Color)

Citra warna adalah citra yang setiap pikselnya mewakili warna yang merupakan kombinasi dari tiga warna dasar yaitu merah, hijau dan biru. Setiap warna dasar ini menggunakan penyimpanan 8 bit atau 1 byte, yang berarti setiap warna mempunyai gradasi sebanyak 256 warna. Hal ini berarti setiap piksel dalam citra warna mempunyai kombinasi warna sebanyak  $2^8 \cdot 2^8 \cdot 2^8 = 2^{24} = 16$  juta lebih. Oleh sebab itu citra ini disebut dengan true color karena mempunyai jumlah warna yang besar sehingga bisa dikatakan hampir mencakup semua warna di alam.

### 3.3 MODEL WARNA HSV

Warna yang kita lihat adalah spektrum cahaya yang dipantulkan oleh benda yang kemudian ditangkap oleh indra penglihatan kita (mata) lalu diterjemahkan oleh otak sebagai sebuah warna tertentu.

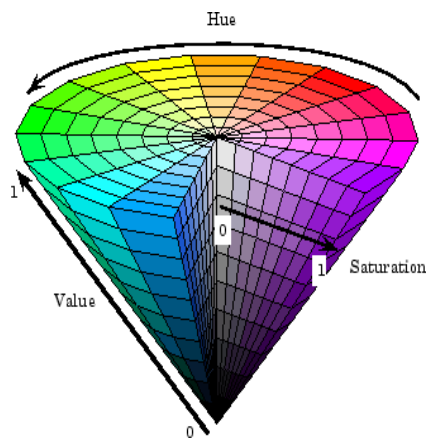


**Gambar 3.4** Spektrum Warna

Model warna HSV mendefinisikan warna dalam terminologi Hue, Saturation, dan Value. Hue menyatakan warna sebenarnya, seperti merah, violet, dan kuning. Hue digunakan untuk membedakan warna-warna dan menentukan

kemerahan (redness), Kehijauan (greenness), dsb dari cahaya. Hue berasosiasi dengan panjang gelombang cahaya. Saturation menyatakan tingkat kemurnian suatu warna, yaitu mengindikasikan seberapa banyak warna putih diberikan pada warna. Value adalah atribut yang menyatakan banyaknya cahaya yang diterima oleh mata tanpa memperdulikan warna. Selain itu jarak warna HSV adalah murni dan konsepnya yang hampir seragam maka proses kuantitasi pada HSV dapat dihasilkan dari mengumpulkan warna yang padat dan lengkap.

Nilai *hue* antara 0 sampai 1 berarti warna antara merah melewati kuning, hijau, cyan, biru dan magenta dan kembali menjadi merah. Nilai *saturation* antara 0 sampai 1 berarti dari tidak tersaturutasi (keabuan) sampai tersaturutasi penuh (tidak putih). Nilai *value* atau *brightness* antara 0 sampai 1 berarti warna semakin cerah seperti ditunjukkan pada Gambar 3.5.



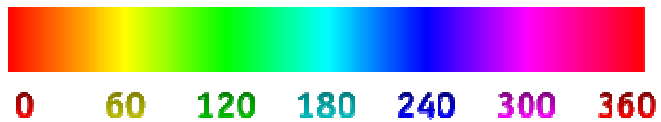
**Gambar 3.5** Model Warna HSV

Karena model warna HSV merupakan model warna yang diturunkan dari model warna RGB, maka untuk mendapatkan warna HSV ini kita harus



melakukan proses konversi warna dari RGB ke HSV. HSV merupakan salah satu cara untuk mendefinisikan warna yang didasarkan pada roda warna.

Hue merupakan variable yang menyatakan warna dari merah hingga *violet*. Hue mengukur sudut sekitar roda warna (merah pada 0°, 120° di hijau, dan biru di 240°). Nilai dari Hue berkisar antara 0° sampai dengan 360°.



**Gambar 3.6** Skala Hue

Saturation merupakan variable yang menyatakan *vibrancy* dari suatu warna, Saturation bisa disebut juga dengan *purity*. Semakin kecil nilai dari Saturation, maka warna yang ditampilkan condong ke warna abu-abu. Skala nilai dari Saturation berkisar antara 0% hingga 100%.

Value menunjukkan nilai kecerahan dari suatu warna. Skala dari Value berkisar antara 0% hingga 100%.

Model HSV memisahkan komponen intensitas dari citra warna, sehingga model ini merupakan model yang ideal untuk mengembangkan algoritma pemrosesan citra yang intuitif dan natural ( Gonzalez & Woods, 2002, p. 295 )

### **3.4 KOVERSI CITRA**

#### **3.4.1. Konversi RGB Ke HSV**

RGB merupakan kependekan dari Red, Green, Blue. Warna – warna yang dibentuk oleh model warna merupakan hasil campuran dari warna-warna primer merah, hijau, dan biru berdasarkan komposisi tertentu.

Perhitungan konversi RGB menjadi HSV dapat dirumuskan sebagai berikut :

$$H = \tan\left[\frac{3(G-B)}{(R-G)+(R-B)}\right] \quad (3.3)$$

$$S = 1 - \frac{\min(R,G,B)}{V} \quad (3.4)$$

$$V = \frac{R+G+B}{3} \quad (3.5)$$

Pada rumus di atas, apabila  $S = 0$  maka  $H$  tidak dapat ditentukan. Untuk itu diperlukan normalisasi RGB terlebih dahulu dengan rumus berikut :

$$r = \frac{R}{R+G+B} \quad (3.6)$$

$$g = \frac{G}{R+G+B} \quad (3.7)$$

$$b = \frac{B}{R+G+B} \quad (3.8)$$

Setelah nilai  $r$ ,  $g$ , dan  $b$  sudah di normalisasi, maka rumus transformasi RGB ke HSV sebagai berikut :

$$V = \max(r, g, b) \quad (3.9)$$

$$S = \begin{cases} 0 & \text{Jika } V = 0 \\ V - \frac{\min(r,g,b)}{V} & \text{Jika } V > 0 \end{cases} \quad (3.10)$$

$$H = \begin{cases} 0 & \text{Jika } S = 0 \\ \frac{60X(g-b)}{SXV} & \text{Jika } V = r \\ 60X\left[2 + \frac{(b-r)}{SXV}\right] & \text{Jika } V = g \\ 60X\left[4 + \frac{(r-g)}{SXV}\right] & \text{Jika } V = b \end{cases} \quad (3.11)$$

$$H = H+360 \qquad \text{Jika } H < 0 \qquad (3.12)$$

### 3.5 THRESHOLD

Thresholding merupakan salah satu teknik segmentasi yang baik digunakan untuk citra dengan perbedaan nilai intensitas yang signifikan antara latar belakang dan objek utama (*Katz, 2000*). Dalam pelaksanaannya thresholding membutuhkan suatu nilai yang digunakan sebagai nilai pembatas antara objek utama dengan latar belakang dan nilai tersebut dinamakan dengan threshold. Umumnya menurut *Fisher dkk(2003)* nilai T dihitung dengan menggunakan persamaan (3.13) :

$$T = \frac{f_{maks}+f_{min}}{2} \qquad (3.13)$$

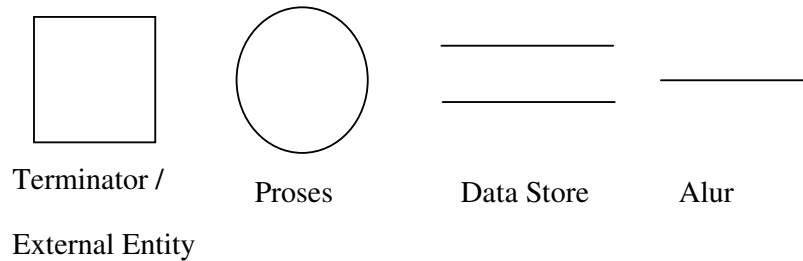
Dimana *f<sub>maks</sub>* adalah nilai intensitas maksimum pada citra dan *f<sub>min</sub>* adalah nilai intensitas minimum pada citra atau dengan kata lain rata-rata dari nilai minimum dan maksimum.

### 3.6 DFD ( Data Flow Diagram )

Data Flow Diagram (DFD) adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan system sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi. DFD ini sering disebut juga dengan nama Bubble chart, Bubble diagram, model proses, diagram alur kerja, atau model fungsi (*Parno : 2011*)

DFD ini merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa

maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program.



**Gambar 3.7** Komponen Data Flow Diagram

### 3.6.1. Terminator /External Entity

Terminator mewakili entitas eksternal yang berkomunikasi dengan sistem yang sedang dikembangkan. Biasanya terminator dikenal dengan nama entitas luar (*external entity*).

Ada tiga hal penting yang harus diingat tentang terminator :

- a. Terminator merupakan bagian/lingkungan luar sistem. Alur data yang menghubungkan terminator dengan berbagai proses sistem, menunjukkan hubungan sistem dengan dunia luar.
- b. Profesional Sistem Tidak berhak mengubah isi atau cara kerja organisasi atau prosedur yang berkaitan dengan terminator.
- c. Hubungan yang ada antar terminator yang satu dengan yang lain tidak digambarkan pada DFD.

### **3.6.2. Proses**

Komponen proses menggambarkan bagian dari sistem yang mentransformasikan input menjadi output.

Proses diberi nama untuk menjelaskan proses/kegiatan apa yang sedang/akan dilaksanakan. Pemberian nama proses dilakukan dengan menggunakan kata kerja transitif (kata kerja yang membutuhkan obyek).

Ada beberapa hal yang perlu diperhatikan tentang proses :

- a. Proses harus memiliki input dan output.
- b. Proses dapat dihubungkan dengan komponen terminator, data store atau proses melalui alur data.
- c. Sistem/bagian/divisi/departemen yang sedang dianalisis oleh profesional sistem digambarkan dengan komponen proses.

### **3.6.3. Data Store**

Komponen ini digunakan untuk membuat model sekumpulan paket data dan diberi nama dengan kata benda jamak.

Data store ini biasanya berkaitan dengan penyimpanan-penyimpanan, seperti file atau database yang berkaitan dengan penyimpanan secara komputerisasi, misalnya file disket, file harddisk, file pita magnetik. Data store juga berkaitan dengan penyimpanan secara manual seperti buku alamat, file folder, dan agenda.

Suatu data store dihubungkan dengan alur data hanya pada komponen proses, tidak dengan komponen DFD lainnya. Alur data yang menghubungkan data store dengan suatu proses mempunyai pengertian sebagai berikut :

1. **Alur data dari data store** yang berarti sebagai pembacaan atau pengaksesan satu paket tunggal data, lebih dari satu paket data, sebagian dari satu paket tunggal data, atau sebagian dari lebih dari satu paket data untuk suatu proses.
2. **Alur data ke data store** yang berarti sebagai pengupdetan data, seperti menambah satu paket data baru atau lebih, menghapus satu paket atau lebih, atau mengubah/modifikasi satu paket data atau lebih.

#### **3.6.4. Alur Data**

Suatu data flow atau alur data digambarkan dengan anak panah, yang menunjukkan arah menuju ke dan keluar dari suatu proses. Alur data ini digunakan untuk menerangkan perpindahan data atau paket data/informasi dari satu bagian sistem ke bagian lainnya.

Selain menunjukkan arah, alur data pada model yang dibuat oleh profesional sistem dapat merepresentasikan bit, karakter, pesan, formulir, bilangan real, dan macam-macam informasi yang berkaitan dengan komputer. Alur data juga dapat merepresentasikan data atau informasi yang tidak berkaitan dengan komputer.

Ada empat konsep yang perlu diperhatikan dalam penggambaran alur data, yaitu :

1. Konsep Paket Data (Packets of Data)

Apabila dua data atau lebih mengalir dari suatu sumber yang sama menuju ke tujuan yang sama dan mempunyai hubungan, dan harus

dianggap sebagai satu alur data tunggal, karena data itu mengalir bersama-sama sebagai satu paket.

2. Konsep Alur Data Menyebar (Diverging Data Flow)

Alur data menyebar menunjukkan sejumlah tembusan paket data yang yang berasal dari sumber yang sama menuju ke tujuan yang berbeda, atau paket data yang kompleks dibagi menjadi beberapa elemen data yang dikirim ke tujuan yang berbeda, atau alur data ini membawa paket data yang memiliki nilai yang berbeda yang akan dikirim ke tujuan yang berbeda.

3. Konsep Alur Data Mengumpul (Converging Data Flow)

Beberapa alur data yang berbeda sumber bergabung bersama-sama menuju ke tujuan yang sama.

4. Konsep Sumber atau Tujuan Alur Data

Semua alur data harus minimal mengandung satu proses. Maksud kalimat ini adalah :

- a. Suatu alur data dihasilkan dari suatu proses dan menuju ke suatu data store dan atau terminator .
- b. Suatu alur data dihasilkan dari suatu data store dan atau terminator dan menuju ke suatu proses.
- c. Suatu alur data dihasilkan dari suatu proses dan menuju ke suatu proses.

### 3.7 MATLAB

Matlab adalah sebuah bahasa dengan (high-performance) kinerja tinggi untuk komputasi masalah teknik. Matlab mengintegrasikan komputasi, visualisasi, dan pemrograman dalam suatu model yang sangat mudah untuk pakai dimana masalah-masalah dan penyelesaiannya diekspresikan dalam notasi matematika yang familiar. Penggunaan Matlab meliputi bidang–bidang:

- Matematika dan Komputasi
- Pembentukan Algorithm
- Akusisi Data
- Pemodelan, simulasi, dan pembuatan prototipe
- Analisa data, explorasi, dan visualisasi
- Grafik Keilmuan dan bidang Rekayasa

Matlab adalah sebuah sistem interaktif yang menggunakan elemen data dasarnya adalah *array* yang tidak membutuhkan dimensi. Hal ini mempermudah kita dalam menyelesaikan masalah komputasi, terutama yang menyangkut matriks dan vektor.

Nama matlab merupakan singkatan dari '*matrix laboratory*' yang pada awalnya dibuat untuk mempermudah pengembangan perangkat lunak berbasis matriks oleh proyek LINPACK (Linear System Package) dan EISPACK (Eiges System Package).

#### 3.7.1. Bagian Terpenting Matlab :

- a. Command Window



Command Window merupakan jendela perintah yang mana semua perintah matlab seperti perhitungan biasa, memanggil fungsi, mencari informasi tentang bantuan (Help), demo program dan lainnya dapat ditulis dan dieksekusi disini. Setiap penulisan perintah disini harus diawali dengan prompt ">>".

b. Workspace

Workspace merupakan jendela matlab yang berisi informasi pemakaian variabel di dalam memori matlab.

c. Command History

Command History merupakan jendela matlab yang berisi informasi tentang perintah yang pernah dituliskan sebelumnya.

d. Current Directory

Current Directory merupakan jendela matlab yang digunakan untuk menentukan direktori aktif yang digunakan oleh matlab.

e. Preferences

Preferences merupakan fasilitas yang digunakan untuk mengatur segala sesuatu tentang matlab. Fasilitas yang tersedia dalam preferences antara lain : general, command window, command history, editor/ debugger, help, current directory, workspace, array editor, guide, figure copy template dan simulink

f. M-File

M-File merupakan sederetan perintah matlab yang dituliskan secara berurutan sebagai sebuah file. M-File dapat dituliskan sebagai sebuah

script yang sederhana yang tidak memerlukan argumen input dan output, atau dapat pula ditulis sebagai fungsi yang menerima argumen input dan output.

### 3.7.2. Guide Matlab

Guide atau Gui Builder merupakan sebuah aplikasi graphical user interface (GUI) yang dibangun dengan obyek grafis. Komponen-komponen guide antara lain :

1. Pushbutton merupakan jenis kontrol yang menghasilkan tombol tekan jika di klik.
2. Togglebutton seperti pushbotton hanya untuk kembali seperti semula perlu ditekan lagi.
3. Radiobutton merupakan jenis kontrol untuk pemilihan.
4. Checkboxes merupakan jenis kontrol pemilihan yang tidak bergantung pada pilihan lainnya.
5. Edit text merupakan jenis kontrol untuk tempat menulis inputan.
6. Static text merupakan jenis kontrol untuk menulis teks yang bersifat tetap.
7. Slider merupakan komponen gui yang dapat digeser secara horisontal maupun vertikal.
8. Frame merupakan komponen gui yang berupa kotak tertutup untuk mengelompokkan kontrol yang berhubungan.

9. Listboxes merupakan jenis kontrol yang dapat menampilkan semua jenis item untuk dipilih.
10. Popmenu merupakan jenis kontrol untuk membuka tampilan daftar pilihan dengan mengklik tanda panah pada aplikasi.


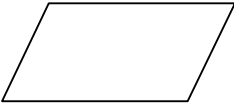
### 3.8 FLOWCHART

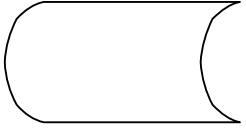

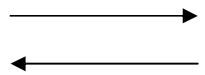
Flowchart merupakan gambar atau bagan yang memperlihatkan urutan dan hubungan antar proses beserta instruksinya. Gambaran ini dinyatakan dengan simbol.

Sistem flowchart adalah bagan yang memperlihatkan urutan prosedur dan proses dari beberapa file di dalam media tertentu. Program flowchart adalah bagan yang memperlihatkan urutan dan hubungan proses dalam suatu program.

Berikut ini adalah simbol-simbol standar yang telah banyak digunakan pada penggambaran program flowchart :

**TABEL 3.1** Simbol-simbol Flowchart

Simbol	Keterangan
	Proses
	Input / Output

	Data Store
	Awa / Akhir Program
	Arus

### 3.9 VARIABEL FORMULASI DATA

a. Max Value (Max V)

Max value adalah nilai tertinggi dalam suatu data.

b. Min Value (Min V)

Min value adalah nilai terendah dalam suatu data.

c. Mean ( $\bar{x}$ )

Mean adalah nilai rata-rata dari beberapa buah data (sampel data). Nilai mean dapat ditentukan dengan membagi jumlah data dengan banyaknya data.

$$\bar{x} = \frac{x_1+x_2+x_3+\dots+x_n}{n} \dots (3.14)$$

dimana,

$\bar{x}$  adalah nilai rata-rata

$x_n$  adalah data ke n

n adalah banyaknya data

d. Standar deviasi (S)

Standar deviasi adalah property data yang menggambarkan keragaman suatu kumpulan data (sampel data). Secara sistematis dirumuskan sebagai berikut.

$$S = \sqrt{\sum \frac{(x_n - \bar{x})^2}{n}} \dots (3.15)$$

dimana,

S adalah standar deviasi

$\bar{x}$  adalah nilai rata-rata

$x_n$  adalah data ke n

n adalah banyaknya data

e. Max

Max adalah nilai dari hasil penjumlahan nilai mean dengan nilai standar deviasi.

$$Max = \bar{x} + S \dots (3.16)$$

dimana,

$\bar{x}$  adalah nilai rata-rata

S adalah standar deviasi

f. Min

Min adalah nilai dari hasil pengurangan nilai mean dengan nilai standar deviasi.

$$Min = \bar{x} - S \dots (3.17)$$

dimana,

$\bar{x}$  adalah nilai rata-rata

S adalah standar deviasi

g. Nilai Delta

Nilai delta adalah nilai dari rata-rata selisih nilai min H pada tiap kelas.

h. Max Akhir

Max Akhir adalah nilai dari hasil penjumlahan nilai max dengan nilai delta.

i. Min akhir

Min akhir adalah nilai dari hasil pengurangan nilai min dengan nilai delta.

## **BAB IV**

### **PERANCANGAN SISTEM**

#### **4.1 KEBUTUHAN SISTEM**

##### **4.1.1. Perangkat Keras ( Hardware )**

Perangkat keras adalah alat elektronik yang terdiri dari beberapa unit yaitu unit masukan ( *input* ), unit pengolahan ( *processing* ), unit keluaran ( *output* ) yang dapat terlihat dan dijamaah.

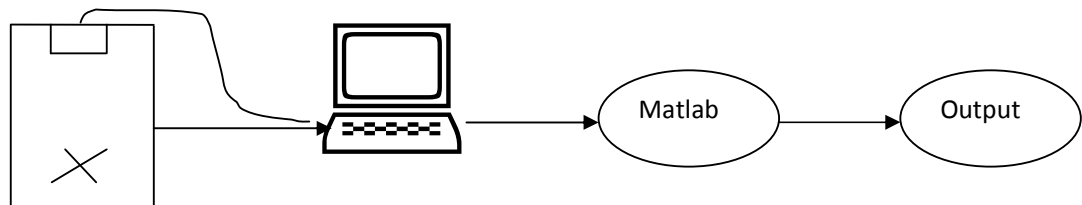
Adapun spesifikasi perangkat keras yang digunakan untuk membuat program ini adalah :

1. Processor Intel Pentium Dual Core 2.0 GHz
2. RAM ( *Random Access Memory* ) 1 GB
3. Keyboard dan mouse standar
4. Hardisk dengan kapasitas 160 GB
5. Webcam 1,3 MP

##### **4.1.2. Sistem Akuisisi Data**

Sistem akuisisi data dapat didefinisikan sebagai suatu sistem yang berfungsi untuk mengambil, mengumpulkan dan menyiapkan data hingga memprosesnya untuk menghasilkan data yang di kehendaki, jenis serta metode yang dipilih pada umumnya bertujuan untuk menyederhanakan setiap langkah yang dilaksanakan pada keseluruhan proses.

Pada sistem akuisisi data pada umumnya di bentuk sedemikian rupa sehingga sistem tersebut berfungsi untuk mengambil, mengumpulkan dan menyiapkan data yang siap untuk di proses lebih lanjut.



**Gambar 4.1** Sistem Akuisisi Data

Dari gambar di atas user menggunakan kotak kayu yang berukuran 20cmX20cm untuk menyangga webcam. Alas untuk tempat mahkota bunga dengan background hitam berukuran 16cmX16cm. Objek bunga yang akan dicapture ditaruh dibawah webcam secara vertikal dan objek merupakan bunga tunggal. Setelah itu image diproses menggunakan software matlab untuk mengkalkulasi metode yang dipakai sehingga user mengetahui hasil akhir yang akan diberikan oleh sistem.

#### **4.1.3. Perangkat Lunak ( Software )**

Perangkat lunak yang menggambarkan sistem operasi yang akan digunakan untuk membuat program yang akan mengolah data menjadi informasi.

Perangkat lunak yang digunakan dalam pembuatan program ini adalah :

1. Sistem Operasi Windows XP Profesional Service Pack 2
2. Bahasa Pemrograman Matlab 7.1

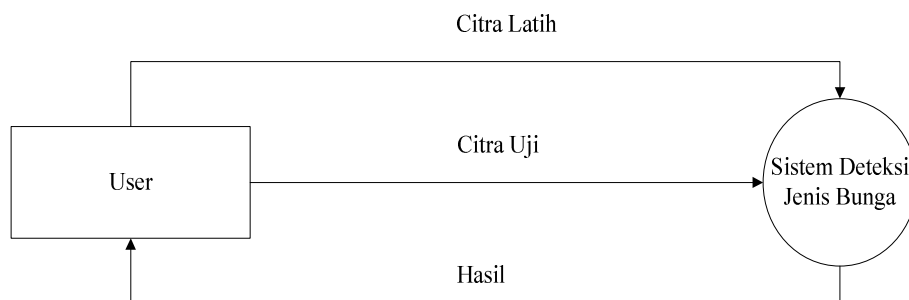


## 4.2 PERANCANGAN SISTEM

Perancangan sistem adalah sekumpulan elemen-elemen yang saling berkaitan dan berinteraksi dalam memproses suatu masukan atau keluaran sehingga dapat menghasilkan keluaran atau output yang diinginkan. Dalam perancangan sistem ini akan menjelaskan pemodelan sistem yang digunakan dan perancangan tampilan atau *user-interface* yang digunakan dalam aplikasi “ Sistem Deteksi Jenis Bunga Menggunakan Nilai HSV Dari Citra Mahkota Bunga”.

### 4.2.1 Data Flow Diagram Sistem Deteksi Jenis Bunga

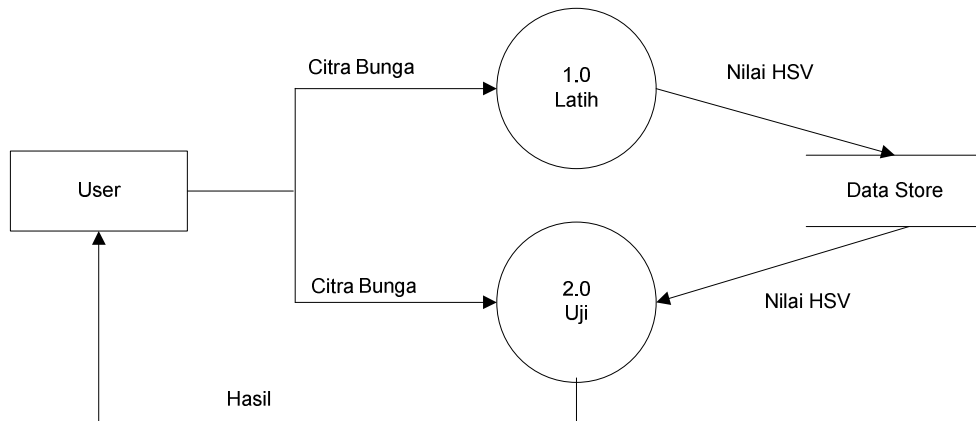
Data Flow Diagram (DFD) adalah suatu diagram yang menggunakan notasi – notasi untuk menggambarkan arus dari data sistem, yang penggunaannya sangat membantu untuk memahami sistem secara logika, tersruktur dan jelas.



Gambar 4.2 DFD Level 0

DFD Level 0 :

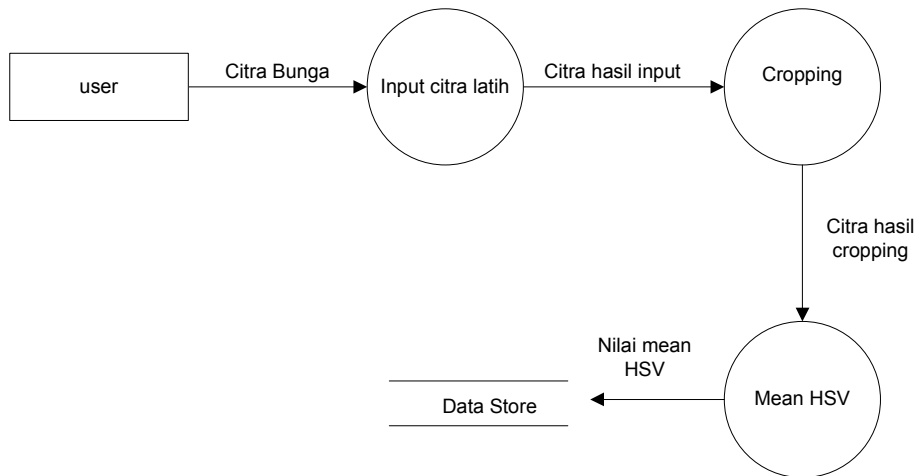
User melakukan 2 input yaitu citra latih dan citra uji. Citra latih digunakan untuk melatih data yang akan diolah untuk mengetahui nilai rata-rata dari HSV. Sedangkan citra uji digunakan untuk menguji data yang akan di eksekusi pada program sehingga menghasilkan output berupa nama jenis bunga.



**Gambar 4.3** DFD Level 1

DFD level 1 :

user melakukan 2 proses yaitu Latih dan Uji. Pada proses Latih menghasilkan data berupa nilai mean HSV yang di simpan di data store. Sedangkan proses uji, mengambil nilai dari data store untuk di jalankan dan mendapatkan hasil akhir berupa nama jenis bunga.



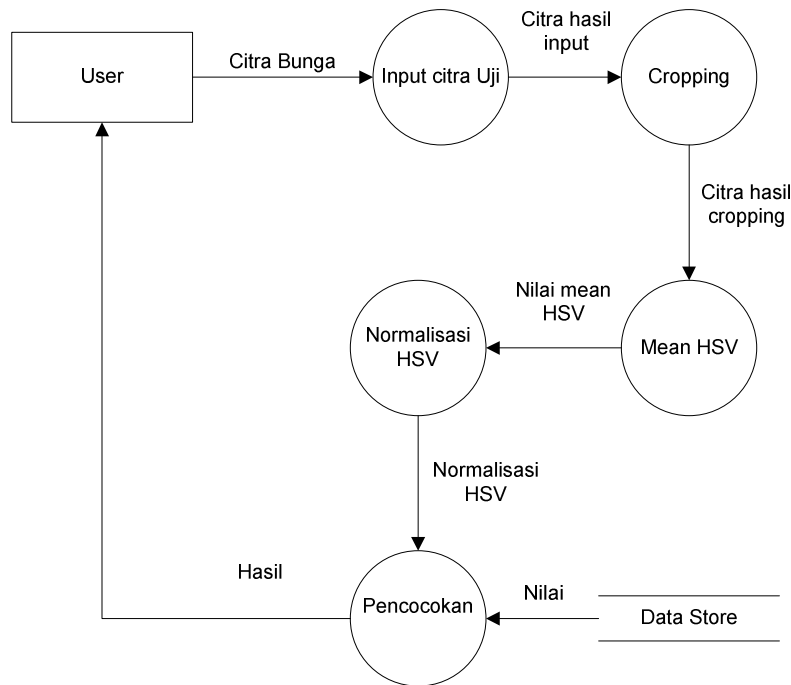
DFD Level 2 dari proses latih

**Gambar 4.4** DFD Level 2 dari proses latih

DFD Level 2 dari proses latih :

User menginput citra bunga untuk dilatih, sebelumnya citra bunga diambil dengan menggunakan webcam yang secara vertikal sejajar dengan objek bunga yang akan dicapture dengan jarak pengamatan 20cm. Citra yang dihasilkan berupa citra berwarna (RGB) dengan resolusi 640X480. Citra yang di input akan tercropping secara otomatis oleh sistem sehingga memiliki ukuran 135X135 piksel. Hal ini dilakukan untuk memisahkan antara objek dengan background. Citra yang di input (RGB) dikonversi ke HSV untuk memudahkan dalam menghitung nilai mean HSV. Ciri yang dipakai pada proses HSV adalah rata-rata Hue dan rata-rata Saturutasi. Proses latih data dilakukan berulang selama 6 kali.

Setelah mendapatkan nilai mean H,S, dan V kemudian di copy secara manual dan disimpan di data store (Excel). Kemudian menentukan nilai max value dan min value, nilai mean, nilai standar deviasi dari tiap kolom H,S,V. Kemudian hitung nilai max yang didapat dari penjumlahan nilai mean dengan nilai standar deviasi, dan nilai min dari pengurangan nilai mean dan standar deviasi. Selanjutnya hitung nilai delta yang digunakan sebagai threshold agar data yang berada pada batas nilai dapat teridentifikasi. Nilai delta diperoleh dari rata-rata selisih nilai H pada tiap kelas. Dari nilai delta kemudian hitung nilai max dan min akhir. Nilai max akhir diperoleh dari penjumlahan nilai delta dengan nilai max, sedangkan nilai min akhir diperoleh dari pengurangan nilai min dengan nilai delta. Nilai max dan min akhir digunakan sebagai acuan dalam penggolongan klasifikasi kelas bunga. Penyimpanan dan penghitungan data dilakukan di excel.



DFD Level 2 dari proses uji

**Gambar 4.5** DFD Level 2 dari proses uji

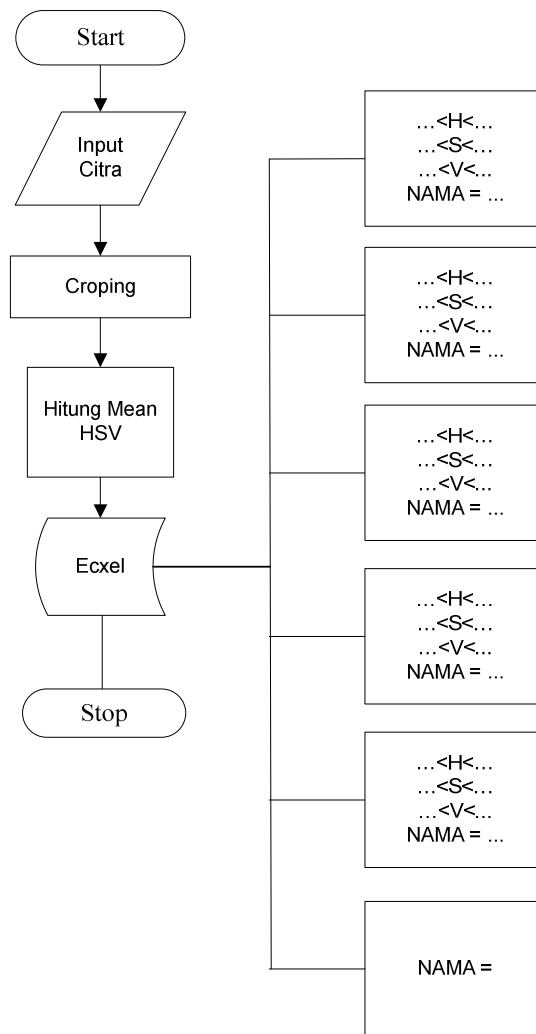
DFD Level 2 dari proses uji :

User menginput citra bunga yang sebelumnya sudah diambil dengan menggunakan webcam yang secara vertikal sejajar dengan objek bunga yang akan dicapture. Citra yang di input (RGB) dikonversi ke HSV untuk memudahkan dalam menghitung nilai mean HSV. Setelah mendapatkan nilai mean H,S, dan V kemudian di simpan di data store (Excel). Kemudian data tersebut di cocokkan dengan data latih yang sebelumnya sudah tersimpan di data store (Excel). Hasil akhir berupa nama jenis bunga.

## 4.2.2 Flowchart

Flowchart merupakan gambar atau bagan yang memperlihatkan urutan dan hubungan antar proses beserta instruksinya. Gambaran ini dinyatakan dengan simbol.

### 4.2.2.1 Flowchart Latih



**Gambar 4.6** Flowchart Latih

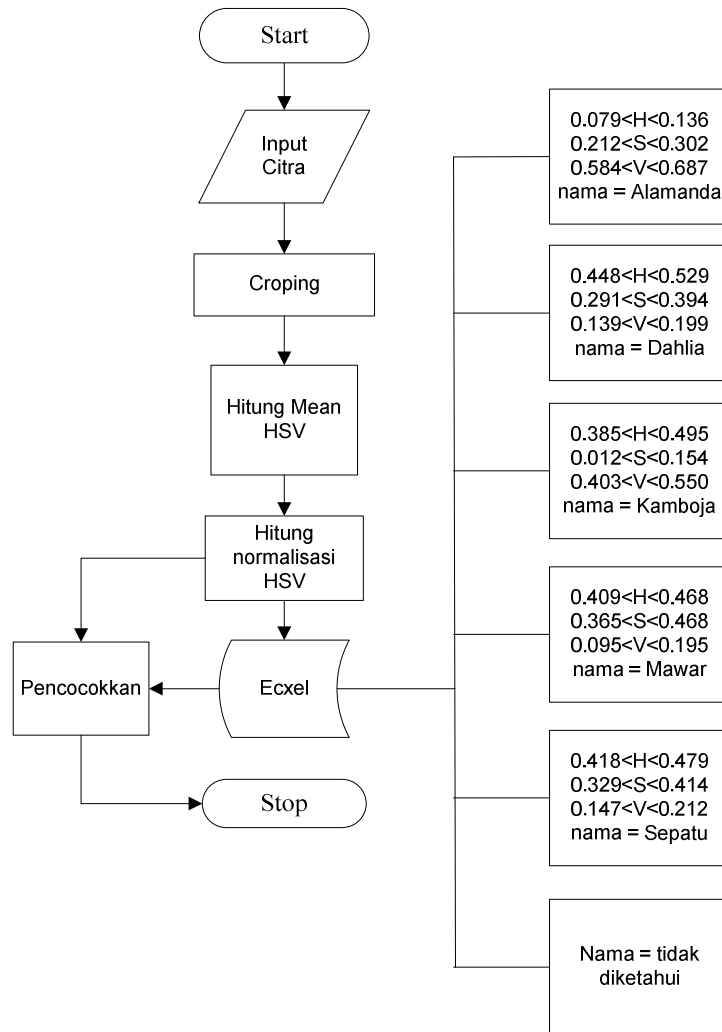
Keterangan Gambar 4.6 :

Mulai dengan menginput citra, kemudian citra di cropping di dalam sistem untuk mendapatkan citra bunga. Setelah tercropping selanjutnya hitung mean HSV yang dilakukan secara berulang sebanyak sampel. Dari mean HSV kemudian hitung kadar HSV dimana terdapat nilai normalisasi, nilai max value dan min value, nilai mean, standar deviasi tiap kolom H,S, dan V.

Kemudian hitung nilai max dari penjumlahan nilai mean dengan nilai standar deviasi dan nilai min dari pengurangan nilai mean dengan nilai standar deviasi. Selanjutnya hitung nilai delta dari rata-rata selisih nilai H tiap kelas. Nilai max akhir dari penjumlahan nilai max dengan nilai delta dan minimum akhir dari pengurangan nilai min dengan nilai delta yang kemudian disimpan di data store ( Excel). Nilai max akhir dan min akhir tersebut digunakan sebagai nilai acuan dalam penggolongan tiap jenis bunga atau sebagai nilai range acuan penggolongan.

Jenis bunga kemudian digolongkan menjadi 5 kategori yaitu bunga alamanda, dahlia, kamboja, mawar dan sepatu. Hasil perhitungan disimpan dan proses berhenti.

#### 4.2.1.2 Flowchart Uji

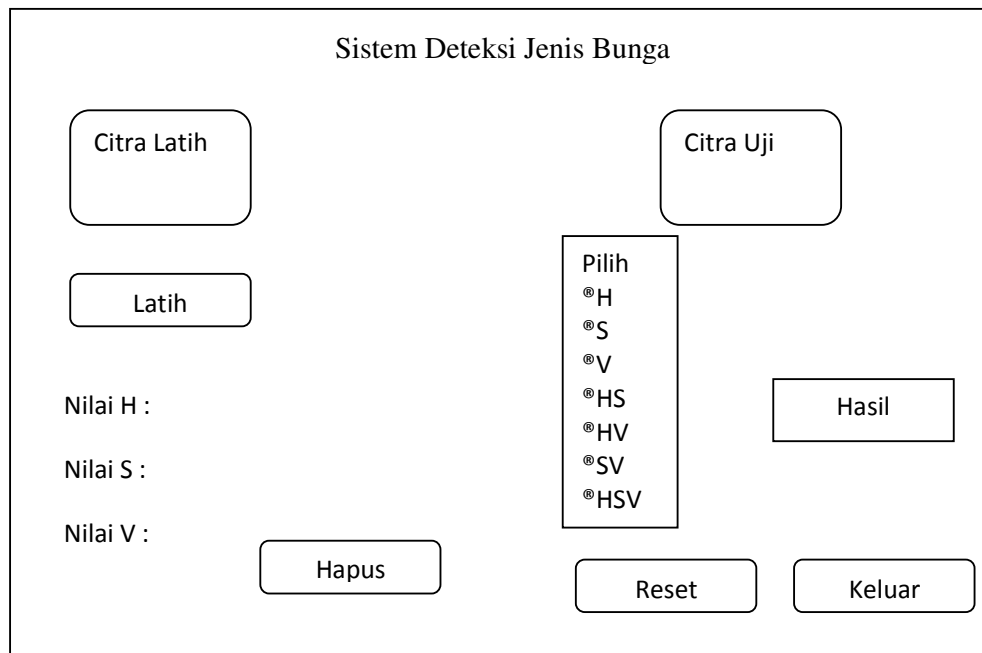


**Gambar 4.7** Flowchart Uji

Dimulai dengan input citra, kemudian citra di cropping untuk mendapatkan citra bunga. Setelah tercropping selanjutnya hitung mean HSV. Dari mean HSV kemudian dinormalisasi nilai HSV. Nilai normalisasi yang sudah dihitung kemudian dicocokkan di data store (Excel) untuk dicocokkan dengan citra yang di

uji untuk melihat citra tersebut masuk ke dalam kategori jenis bunga apa. Hasil akhir berupa nama bunga yang sedang di uji kemudian proses berhenti.

#### 4.2.3 Perancangan Antarmuka ( Interface )



**Gambar 4.8** Rancangan Interface

Keterangan gambar :

1. Citra Latih untuk menampilkan citra hasil inputan yang akan digunakan untuk latihan.
2. Citra Uji untuk menampilkan citra hasil inputan yang akan di uji.
3. Latih untuk memasukkan file citra berekstensi \*.jpg.
4. Hapus untuk menghapus semua data yang sudah di masukkan.
5. Pilih digunakan untuk memilih komponen nilai HSV yang dijadikan sebagai acuan dalam mendeteksi sistem sekaligus untuk menguji program.



6. Reset digunakan untuk menguji ulang program.
7. Hasil untuk menampilkan keterangan jenis bunga dari hasil uji.
8. Keluar untuk mengakhiri program.

## **BAB V**

### **IMPLEMENTASI SISTEM**

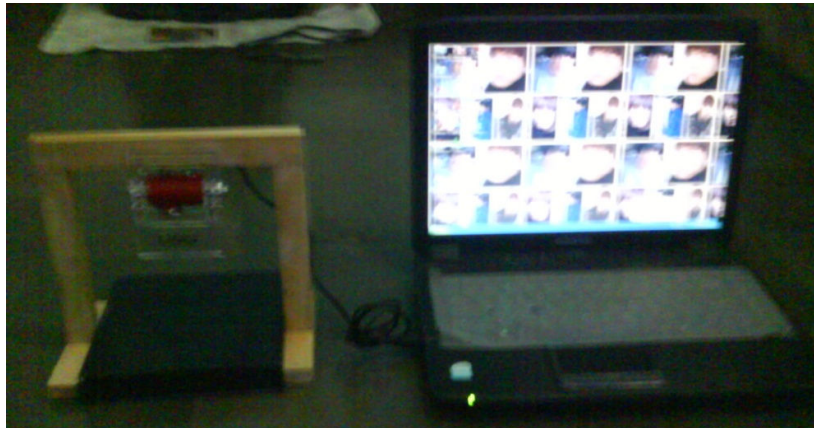
Dari analisis dan perancangan sistem pada bab sebelumnya maka dalam bab ini menggambarkan bagaimana hasil implementasi sistem yang telah dibuat dan dituangkan dalam skrip pemrograman dan apa saja yang diperlukan dalam pengaplikasian rancangan tersebut, maka akan diadakan pengujian program dengan menggunakan emulator.

#### **5.1 IMPLEMENTASI SISTEM**

Pada tahap implementasi sistem ini merupakan tahap awal untuk mendeteksi jenis bunga yang merupakan aplikasi untuk mendeteksi nama bunga. Terdapat 2 proses sebelum melakukan pendeteksian yaitu proses pelatihan sistem dan proses pengujian Sistem.

##### **5.1.1. SISTEM AKUISISI DATA**

Sistem akuisisi data dapat didefinisikan sebagai suatu sistem yang berfungsi untuk mengambil, mengumpulkan dan menyiapkan data hingga memprosesnya untuk menghasilkan data yang di kehendaki, jenis serta metode yang dipilih pada umumnya bertujuan untuk menyederhanakan setiap langkah yang dilaksanakan pada keseluruhan proses.



**Gambar 5.1** Sistem Akuisisi Data

Citra yang digunakan dalam sistem ini berupa citra warna dengan resolusi 640 X 480 piksel. Citra yang di gunakan ada 5 jenis bunga yaitu bunga alamanda, bunga dahlia, bunga kamboja, bunga mawar, bunga sepatu. Citra diambil dengan menggunakan webcam yang diletakkan secara vertikal dengan alas yang diberi background warna hitam dengan cahaya di dalam ruangan. Obyek diamati dengan jarak pengamatan 20 cm dan zoom tetap.

### 5.1.2. Antarmuka Sistem



**Gambar 5.2** Antarmuka Sistem

Keterangan gambar :

1. Latih untuk menampilkan citra yang dilatih sekaligus akan digunakan untuk menampilkan hasil citra masukan.
2. Hapus untuk menghapus semua data yang sudah di masukkan.
3. Uji untuk menampilkan citra hasil inputan yang akan di uji.
4. Pilih digunakan untuk memilih komponen nilai HSV yang dijadikan sebagai acuan dalam mendeteksi sistem sekaligus untuk menguji sistem.
5. Reset digunakan untuk menguji ulang sistem.
6. Hasil untuk menampilkan keterangan jenis bunga dari hasil uji.
7. Keluar untuk mengakhiri program.

Keterangan pada implementasi program :

a) Tombol Latih

Tombol ini digunakan untuk membuka dan memilih file citra yang akan dilatih dengan ekstensi \*.jpg serta memproses nilai mean HSV pada citra dan menampilkan nilai masing-masing mean H, S,V pada kolom edit text.

```
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

[a,b]=uigetfile({'*.jpg'},'buka gbr');
if isequal ([a,b],[0,0])
    return
end
I=imread ([b,a], 'jpg');
axes(handles.axes1);
%cropping
I = imcrop(I,[ 135 135 135 135 ]);
axes(handles.axes1);
imshow (I);
```

**Gambar 5.3** Listing program latih

Keterangan Gambar 5.3 :

- i. Uigetfile untuk membuka file gambar.
- ii. Imread untuk membaca citra berkecstensi \*.jpg.
- iii. Imcrop digunakan untuk memotong area pada gambar yang bertujuan untuk memisahkan antara objek dengan background.
- iv. Axes untuk tempat citra yang akan ditampilkan

```
hsv= rgb2hsv(I);  
h = hsv(:, :, 1); %untuk mengambil nilai h  
meanH = mean2(h);  
s = hsv(:, :, 2);  
meanS = mean2(s);  
v = hsv(:, :, 3);  
meanV = mean2(v);
```

**Gambar 5.4** Listing menampilkan nilai mean HSV pada Latih

Ketererangan Gambar 5.4 :

Untuk mengubah citra dari RGB ke hSV sekaligus untuk menghitung masing-masing nilai H,S,dan V.

```
normH=meanH/(meanH+meanS+meanV);  
normS=meanS/(meanH+meanS+meanV);  
normV=meanV/(meanH+meanS+meanV);
```

**Gambar 5.5** Listing menampilkan nilai pada Latih

Keterangan Gambar 5.5 :

Untuk menampilkan nilai mean H, mean S, mean V pada masing-masing kotak edit..

b) Tombol Hapus

Tombol ini digunakan untuk menghapus citra dan nilai yang terdapat pada tampilan program latih.

```

function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.edit1,'string','');
set(handles.edit2,'string','');
set(handles.edit3,'string','');
axes(handles.axes1);
plot(0);

```

**Gambar 5.6** Listing Tombol Hapus

c) Tombol Reset

Tombol ini digunakan untuk mereset citra pada axes uji dan hasil nama jenis bunga.

```

function reset_Callback(hObject, eventdata, handles)
% hObject    handle to reset (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.edit4,'String','');
axes(handles.axes2);
plot(0);

```

**Gambar 5.7** Listing Tombol Reset

d) Tombol Keluar

Tombol ini digunakan untuk keluar / mengakhiri dari program sistem deteksi jenis bunga.

```

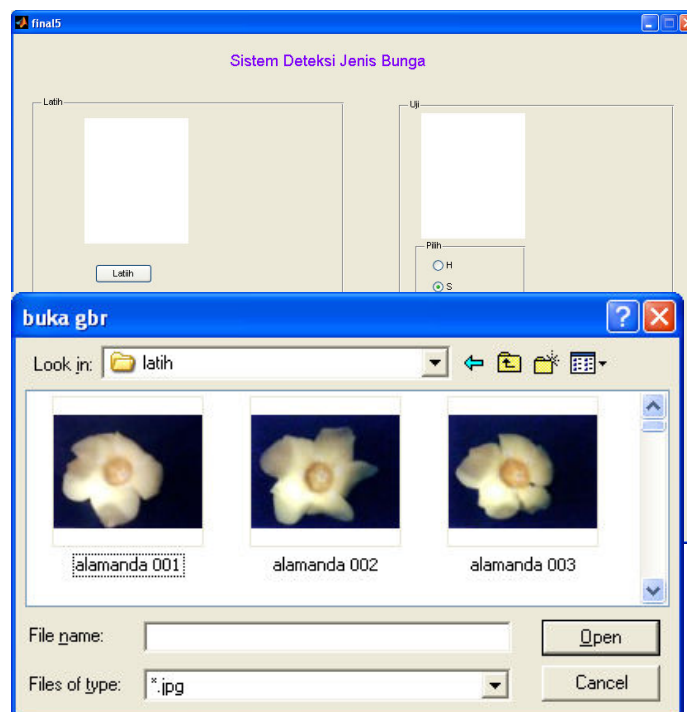
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
keluar = questdlg('Yakin akan keluar?', ...
    'Main', ...
    'Ya', 'Tidak', 'Ya');
switch keluar,
case 'Ya',
    clear handles.Citra;
    clear all;
    close;
case 'Tidak',
    return
end

```

**Gambar 5.8** Listing tombol Keluar pada Program

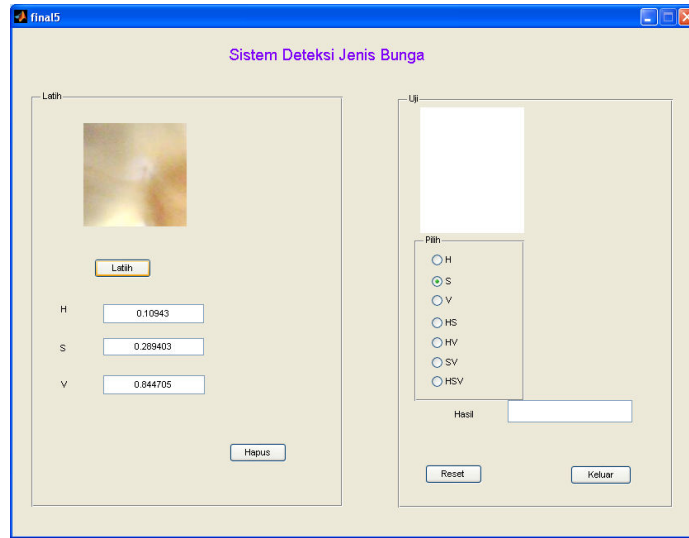
## 5.2 PELATIHAN SISTEM

Citra yang digunakan dalam pelatihan ini berupa citra warna true color (RGB) dengan file berekstensi \*.jpg. Citra yang digunakan dalam penelitian ada 5 jenis bunga yaitu bunga alamanda, dahlia, kamboja, mawar dan sepatu.



**Gambar 5.9** Tampilan Program Latih

Capture citra dengan menggunakan kamera webcame. Citra yang dihasilkan, kemudian disimpan didalam folder latih. Untuk melatih file citra dapat dilakukan dengan cara menekan tombol latih maka akan keluar kotak “buka gbr” untuk memilih file citra bunga yang akan di latih. Gambar yang ditampilkan hasil cropping oleh sistem dengan ukuran 135X135 piksel.




**Gambar 5.10** Tampilan Program Latih

Serta akan menampilkan citra bunga true color beserta nilai mean masing-masing H,S,dan V. Kemudian dari masing-masing nilai mean HSV disimpan dalam data store (Excel) dengan cara mengcopy secara manual dari masing-masing nilai tersebut. Lihat tabel 5.6






### 5.2.1. HASIL PELATIHAN

Berikut akan di tampilkan hasil dari proses pengambilan data pelatihan dan nilai mean dari hasil pelatihan masing-masing obyek :






**Tabel 5.1** : Tabel hasil latih citra bunga Alamanda


DATA ke	CITRA LATIH	PENENTU OBYEK
1		Alamnda 001




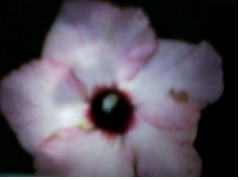


2		Alamanda 002
3		Alamanda 003
4		Alamanda 004
5		Alamanda 005
6		Alamanda 006



**Tabel 5.2** : Tabel hasil latih citra bunga Dahlia

DATA ke	CITRA LATIH	PENENTU OBYEK
1		Dahlia 001
2		Dahlia 003
3		Dahlia 004
4		Dahlia 005
5		Dahlia 006


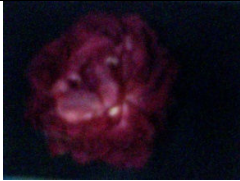
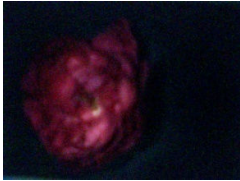
6		Dahlia 007
---	---	------------

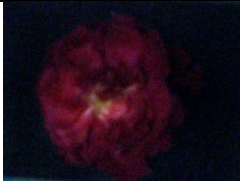

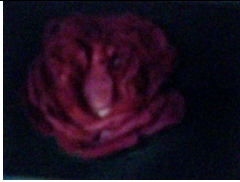
**Tabel 5.3 :** Tabel hasil latih citra bunga Kamboja

DATA Ke	CITRA LATIH	PENENTU OBYEK
1		Kamboja
2		Kamboja 003
3		Kamboja 005
4		Kamboja 007

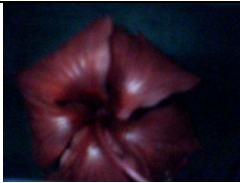

5		Kamboja 008
6		Kamboja 010

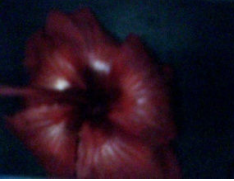

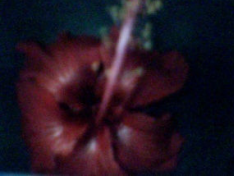
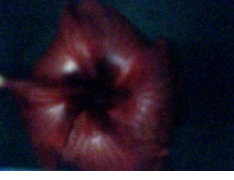
**Tabel 5.4** : Tabel hasil latih citra bunga Mawar

DATA Ke	CITRA LATIH	PENENTU OBYEK
1		Mawar merah 001
2		Mawar merah 002
3		Mawar merah 004

4		Mawar merah 005
5		Mawar merah 006
6		Mawar merah 007

**Tabel 5.5 :** Tabel hasil latih citra bunga Sepatu

DATA ke	CITRA LATIH	PENENTU OBYEK
1		Sepatu_
2		Sepatu_001

3		Sepatu_002
4		Sepatu_003
5		Sepatu_004
6		Sepatu_007

**Tabel 5.6** Data Mean HSV

Data Ke-	Alamanda			Dahlia		
	H	S	V	H	S	V
1	0.109	0.289	0.845	0.770	0.490	0.273
2	0.133	0.273	0.807	0.805	0.634	0.282
3	0.129	0.341	0.766	0.793	0.627	0.273
4	0.139	0.301	0.808	0.780	0.625	0.261
5	0.140	0.345	0.739	0.784	0.441	0.285

Data Ke-	Kamboja			Mawar		
	H	S	V	H	S	V
1	0.669	0.096	0.670	0.748	0.808	0.142
2	0.741	0.146	0.708	0.864	0.733	0.308
3	0.672	0.119	0.731	0.842	0.770	0.345
4	0.561	0.254	0.582	0.785	0.808	0.271
5	0.539	0.043	0.805	0.855	0.779	0.287

Data Ke-	Sepatu		
	H	S	V
1	0.870	0.616	0.367
2	0.742	0.629	0.268
3	0.761	0.661	0.287
4	0.810	0.636	0.350
5	0.765	0.649	0.319

Setelah didapatkan data nilai mean HSV sampel, kemudian dihitung kadar HSV.

**Tabel 5.7** Data Kadar HSV

Data Ke -	Alamanda			Dahlia			Kamboja		
	H	S	V	H	S	V	H	S	V
1	0.088	0.233	0.679	0.502	0.320	0.178	0.466	0.067	0.467
2	0.110	0.225	0.665	0.468	0.368	0.164	0.465	0.091	0.444
3	0.104	0.276	0.620	0.468	0.371	0.161	0.442	0.078	0.480
4	0.111	0.241	0.648	0.468	0.375	0.157	0.401	0.182	0.417
5	0.115	0.282	0.603	0.519	0.292	0.189	0.389	0.031	0.580

6	0.118	0.285	0.597	0.502	0.332	0.166	0.478	0.049	0.472
Min V	0.118	0.285	0.679	0.519	0.375	0.189	0.478	0.182	0.580
Max V	0.088	0.225	0.597	0.468	0.292	0.157	0.389	0.031	0.417
Mean	0.108	0.257	0.635	0.488	0.343	0.169	0.440	0.083	0.477
STD	0.011	0.027	0.034	0.023	0.034	0.012	0.037	0.053	0.056
Max	0.118	0.284	0.669	0.511	0.377	0.181	0.477	0.136	0.532
Min	0.097	0.230	0.602	0.465	0.309	0.157	0.403	0.030	0.421

Data ke -	Mawar			Sepatu		
	H	S	V	H	S	V
1	0.441	0.476	0.083	0.470	0.332	0.198
2	0.453	0.385	0.162	0.453	0.384	0.163
3	0.430	0.393	0.176	0.445	0.387	0.168
4	0.421	0.433	0.145	0.451	0.354	0.195
5	0.445	0.406	0.149	0.442	0.374	0.184
6	0.443	0.403	0.154	0.430	0.399	0.171
Min V	0.453	0.476	0.176	0.470	0.399	0.198
Max V	0.421	0.385	0.083	0.430	0.332	0.163
Mean	0.439	0.416	0.145	0.448	0.372	0.180
STD	0.011	0.034	0.032	0.013	0.024	0.015
Max	0.450	0.450	0.177	0.461	0.396	0.195
Min	0.427	0.382	0.113	0.435	0.347	0.165



Perhitungan kadar HSV meliputi proses hitung normalisasi, nilai max value dan nilai mean value, nilai mean, nilai standar deviasi, nilai max dan min. Normalisasi penting dilakukan terutama bila sejumlah citra ditangkap dengan penerangan yang berbeda-beda. Hasil perhitungan nilai tiap nilai HSV yang telah dinormalisasi akan menghilangkan pengaruh penerangan, sehingga nilai untuk setiap HSV dapat dibandingkan satu sama lainnya walaupun berasal dari citra dengan kondisi penerangan yang tidak sama.

Setelah data normalisasi, kemudian di cari nilai maksimal (max value) dan nilai minimal (min value), nilai mean dan standar deviasi (STD) yang didapat dari nilai normalisasi perkolom masing-masing H, S dan V. Dari hasil tersebut didapatkan nilai max yang berasal dari penjumlahan nilai mean dan standar deviasi, dan untuk nilai min berasal dari hasil pengurangan nilai mean dan standar deviasi.

Dari tabel 5.7 perlu ada penyesuaian, setiap nilai min akan dikurangi dengan nilai delta = 0.017 sebagai threshold agar data yang berada pada batas-batas nilai dapat teridentifikasi dan begitu juga nilai max akan ditambah delta = 0.017.

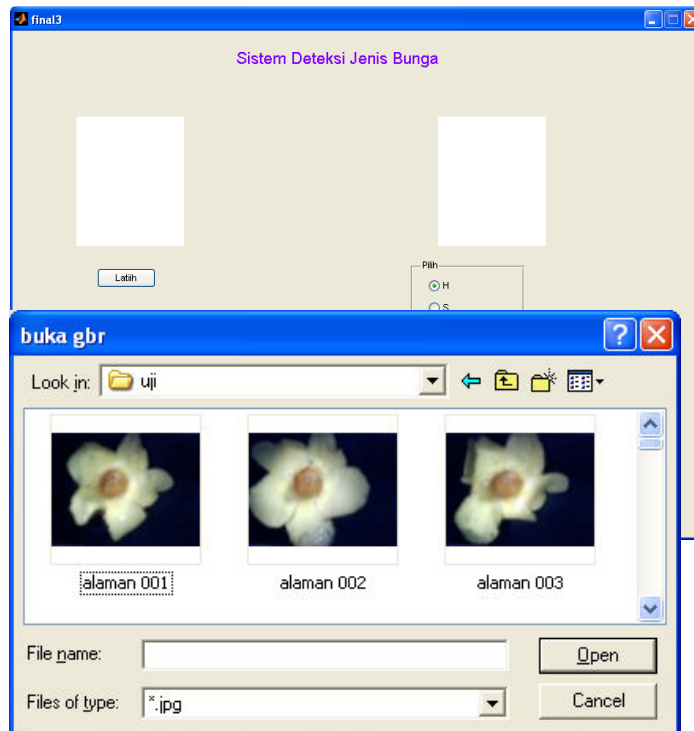
Nilai delta diperoleh dari rata-rata selisih nilai min H pada tiap kelas. Dari penyesuaian tersebut maka nilai range akan menjadi :

- i. Untuk bunga Alamanda
  - a. Min H = 0.079 ; Max H = 0.136
  - b. Min S = 0.212 ; Max H = 0.302
  - c. Min V = 0.584 ; Max V = 0.687
- ii. Untuk bunga Dahlia

- a.  $\text{Min H} = 0.448$  ;  $\text{Max H} = 0.529$
  - b.  $\text{Min S} = 0.291$  ;  $\text{Max S} = 0.394$
  - c.  $\text{Min V} = 0.139$  ;  $\text{Max V} = 0.199$
- iii. Untuk bunga Kamboja
- a.  $\text{Min H} = 0.385$  ;  $\text{Max H} = 0.495$
  - b.  $\text{Min S} = 0.012$  ;  $\text{Max S} = 0.154$
  - c.  $\text{Min V} = 0.403$  ;  $\text{Max V} = 0.550$
- iv. Untuk bunga Mawar
- a.  $\text{Min H} = 0.409$  ;  $\text{Max H} = 0.468$
  - b.  $\text{Min S} = 0.365$  ;  $\text{Max S} = 0.468$
  - c.  $\text{Min V} = 0.095$  ;  $\text{Max V} = 0.195$
- v. Untuk bunga Sepatu
- a.  $\text{Min H} = 0.418$  ;  $\text{Max H} = 0.479$
  - b.  $\text{Min S} = 0.329$  ;  $\text{Max S} = 0.414$
  - c.  $\text{Min V} = 0.147$  ;  $\text{Max V} = 0.212$

### 5.3 PENGUJIAN SISTEM

Untuk pengujian program, dengan cara memilih button pada kotak “pilih”.Misalnya untuk menguji sistem dengan acuan nilai H,maka klik tombol H pada radio button. Kemudian akan keluar kotak ‘buka gambar’ untuk memilih file citra bunga yang akan di uji. Sebelumnya, citra bunga sudah dicapture dan tersimpan dalam folder uji.



**Gambar 5.11** Tampilan Uji pada button H

Setelah memilih data yang akan diuji kemudian klik open akan muncul nama bunga dari bunga yang di uji.



**Gambar 5.12** Tampilan hasil uji dengan acuan nilai H

```
[a,b]=uigetfile({'*.jpg'},'buka gbr');
if isequal ([a,b],[0,0]) %-----untuk mencegah error jika user tidak
jadi mengambil file-----%
    return
end
I=imread ([b,a],'jpg');
%cropping
I = imcrop(I,[ 135 135 135 135 ]);
axes(handles.axes2);
imshow (I);
%MENAMPILKAN NILAI MASING-MASING H,S,V
hsv= rgb2hsv(I);
h = hsv(:, :, 1); %untuk mengambil nilai h
meanH = mean2(h);
s = hsv(:, :, 2);
meanS = mean2(s);
v = hsv(:, :, 3);
meanV = mean2(v);
normH=meanH/(meanH+meanS+meanV);
if normH>0.07904 && normH<0.13625
    set(handles.edit4,'string','Bunga Alamanda');
elseif normH>0.44756 && normH<0.52851
    set(handles.edit4,'string','Bunga Dahlia');
elseif normH>0.38528 && normH<0.49521
    set(handles.edit4,'string','Bunga Kamboja');
elseif normH>0.40947 && normH<0.46817
    set(handles.edit4,'string','Bunga Mawar');
elseif normH>0.41756 && normH<0.47933
    set(handles.edit4,'string','Bunga Sepatu');
else set(handles.edit4,'string','tidak diketahui');
end
```

**Gambar 5.13** Listing Program pada button H

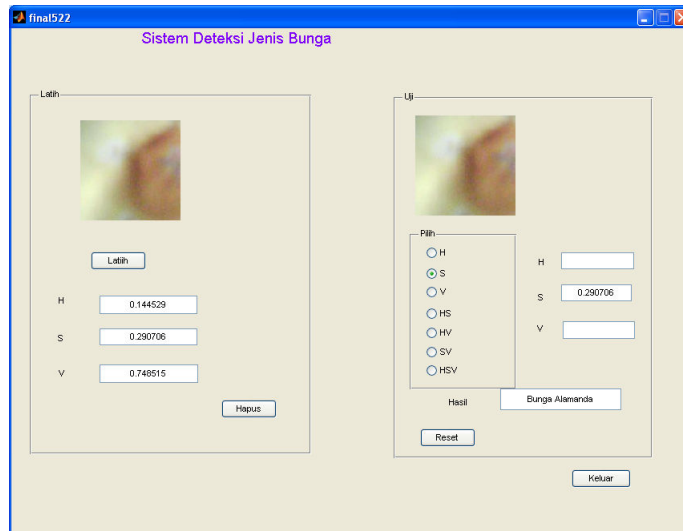
Keterangan Gambar 5.13 :

- Uigetfile untuk membuka file gambar.
- Imread untuk membaca citra berkeestensi \*.jpg.
- Axes untuk tempat citra yang akan ditampilkan
- `h=hsv(:, :, :1);` untuk mengambil nilai h
- `if normH>0.079 && normH<0.136` penggolongan jenis bunga berdasarkan nilai H.
- `set(handles.edit4,'string','Bunga Alamanda');` menampilkan nama bunga.

Proses yang sama juga dilakukan untuk menguji sistem dengan acuan S,V,HS,HV,SV dan HSV.

- Untuk pengujian dengan acuan nilai S

Sistem akan mendeteksi objek dengan menggunakan acuan nilai mean S yang sebelumnya sudah dilatih dan disimpan dalam data store.



**Gambar 5.14** Tampilan hasil uji dengan acuan nilai S

```

[a,b]=uigetfile({'*.jpg'},'buka gbr');
if isequal ([a,b],[0,0]) %-----untuk mencegah error jika user tidak
jadi mengambil file-----%
    return
end
I=imread ([b,a],'jpg');
%cropping
I = imcrop(I,[ 135 135 135 135 ]);
axes(handles.axes2);
imshow (I);
%MENAMPILKAN NILAI MASING-MASING H,S,V
hsv= rgb2hsv(I);
h = hsv(:, :, 1); %untuk mengambil nilai h
meanH = mean2(h);
s = hsv(:, :, 2);
meanS = mean2(s);
v = hsv(:, :, 3);
meanV = mean2(v);
normS=meanS/(meanH+meanS+meanV);
if normS>0.21217 && normS<0.30165
    set(handles.edit4,'string','Bunga Alamanda');
elseif normS>0.29129 && normS<0.39450
    set(handles.edit4,'string','Bunga Dahlia');
elseif normS>0.01248 && normS<0.15368
    set(handles.edit4,'string','Bunga Kamboja');
elseif normS>0.36464 && normS<0.46761
    set(handles.edit4,'string','Bunga Mawar');
elseif normS>0.32939 && normS<0.41387
    set(handles.edit4,'string','Bunga Sepatu');
else set(handles.edit4,'string','tidak diketahui');
end

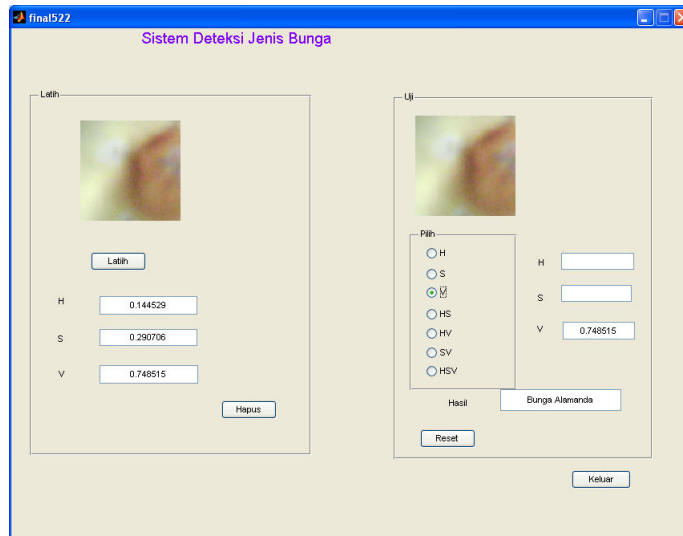
```

**Gambar 5.15** Listing Program pada button S

Keterangan Gambar 5.15 :

- a. Uigetfile untuk membuka file gambar.
  - b. Imread untuk membaca citra berkecstensi \*.jpg.
  - c. Axes untuk tempat citra yang akan ditampilkan
  - d.  $h=hsv(:, :, :1)$ ; untuk mengambil nilai h
  - e. if  $normS>0.212 \ \&\& \ normS<0.302$  penggolongan jenis bunga berdasarkan nilai S
  - f.  $set(handles.edit4,'string','Bunga Alamanda');$  menampilkan nama bunga.
- Untuk pengujian dengan acuan nilai V

Sistem akan mendeteksi objek dengan menggunakan acuan nilai mean V yang sebelumnya sudah dilatih dan disimpan dalam data store.



**Gambar 5.16** Tampilan hasil uji dengan acuan nilai V

```
[a,b]=uigetfile({'*.jpg'},'buka gbr');
if isequal ([a,b],[0,0]) %-----untuk mencegah error jika user tidak
jadi mengambil file-----%
    return
end
I=imread ([b,a],'jpg');
%cropping
I = imcrop(I,[ 135 135 135 135 ]);
axes(handles.axes2);
imshow (I);
%MENAMPILKAN NILAI MASING-MASING H,S,V
hsv= rgb2hsv(I);
h = hsv(:, :, 1); %untuk mengambil nilai h
meanH = mean2(h);
s = hsv(:, :, 2);
meanS = mean2(s);
v = hsv(:, :, 3);
meanV = mean2(v);
normV=meanV/(meanH+meanS+meanV);
if normV>0.58380 && normV<0.68709
    set(handles.edit4,'string','Bunga Alamanda');
elseif normV>0.13923 && normV<0.19892
    set(handles.edit4,'string','Bunga Dahlia');
elseif normV>0.40323 && normV<0.55012
    set(handles.edit4,'string','Bunga Kamboja');
elseif normV>0.09510 && normV<0.19501
    set(handles.edit4,'string','Bunga Mawar');
elseif normV>0.14748 && normV<0.21237
    set(handles.edit4,'string','Bunga Sepatu');
else set(handles.edit4,'string','tidak diketahui');
end
```

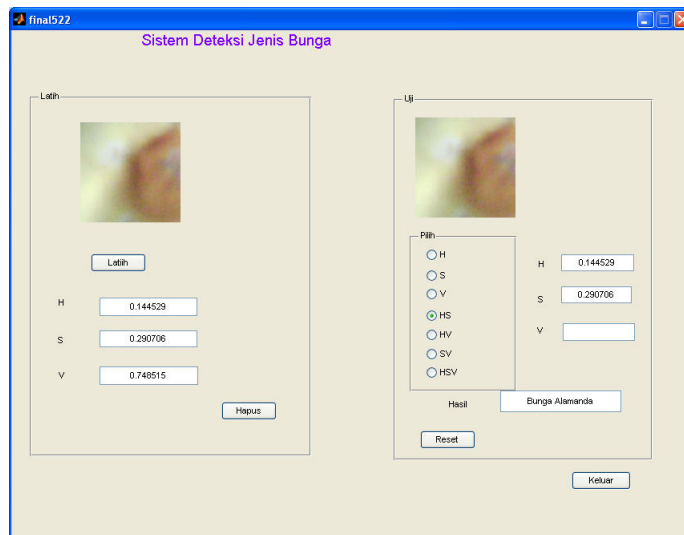
**Gambar 5.17** Listing Program pada button V

Keterangan Gambar 5.17 :

- a. Uigetfile untuk membuka file gambar.
- b. Imread untuk membaca citra berkestensi \*.jpg.
- c. Axes untuk tempat citra yang akan ditampilkan
- d. `h=hsv(:, :, :1);` untuk mengambil nilai h
- e. `if normV>0.584 && normV<0.687` penggolongan jenis bunga berdasarkan nilai S
- f. `set(handles.edit4,'string','Bunga Alamanda');` menampilkan nama bunga.

➤ Untuk pengujian dengan acuan nilai HS

Sistem akan mendeteksi objek dengan menggunakan acuan nilai mean H dan mean S. yang sebelumnya sudah dilatih dan disimpan dalam data store.



**Gambar 5.18** Tampilan hasil uji dengan acuan nilai HS



```

[a,b]=uigetfile({'*.jpg'},'buka gbr');
if isequal ([a,b],[0,0]) %-----untuk mencegah error jika user tidak
jadi mengambil file-----%
    return
end
I=imread ([b,a],'jpg');
%cropping
I = imcrop(I,[ 135 135 135 135 ]);
axes(handles.axes2);
imshow (I);
%MENAMPILKAN NILAI MASING-MASING H,S,V
hsv= rgb2hsv(I);
h = hsv(:, :, 1); %untuk mengambil nilai h
meanH = mean2(h);
s = hsv(:, :, 2);
meanS = mean2(s);
v = hsv(:, :, 3);
meanV = mean2(v);
normH=meanH/(meanH+meanS+meanV);
normS=meanS/(meanH+meanS+meanV);
if normH>0.07904 & normH<0.13625 && normS>0.21217 & normS<0.30165
    set(handles.edit4,'string','Bunga Alamanda');
elseif normH>0.44756 & normH<0.52851 && normS>0.29129 & normS<0.39450
    set(handles.edit4,'string','Bunga Dahlia');
elseif normH>0.38528 & normH<0.49521 && normS>0.001248 & normS<0.15368
    set(handles.edit4,'string','Bunga Kamboja');
elseif normH>0.40947 & normH<0.46817 && normS>0.36464 & normS<0.46761
    set(handles.edit4,'string','Bunga Mawar');
elseif normH>0.41756 & normH<0.47933 && normS>0.32939 & normS<0.41387
    set(handles.edit4,'string','Bunga Sepatu');
else set(handles.edit4,'string','tidak diketahui');
end

```

**Gambar 5.19** Listing Program pada button HS

Keterangan Gambar 5.19:

- a. Uigetfile untuk membuka file gambar.
- b. Imread untuk membaca citra berkestensi \*.jpg.
- c. Axes untuk tempat citra yang akan ditampilkan
- d. `h=hsv(:, :, :1)`; untuk mengambil nilai h
- e. `if normH>0.079 & normH<0.136 && normS>0.212 & normS<0.302` penggolongan jenis bunga berdasarkan nilai HS.
- f. `set(handles.edit4,'string','Bunga Alamanda');` menampilkan nama bunga.

- Untuk pengujian dengan acuan nilai HV  
Sistem akan mendeteksi objek dengan menggunakan acuan nilai mean H dan mean V. yang sebelumnya sudah dilatih dan disimpan dalam data store.



**Gambar 5.20** Tampilan hasil uji dengan acuan nilai HV

```

[a,b]=uigetfile({'*.jpg'},'buka gbr');
if isequal ([a,b],[0,0]) %-----untuk mencegah error jika user tidak
jadi mengambil file-----%
    return
end
I=imread ([b,a],'jpg');
%cropping
I = imcrop(I,[ 135 135 135 135 ]);
axes(handles.axes2);
imshow (I);
%MENAMPILKAN NILAI MASING-MASING H,S,V
hsv= rgb2hsv(I);
h = hsv(:, :, 1); %untuk mengambil nilai h
meanH = mean2(h);
s = hsv(:, :, 2);
meanS = mean2(s);
v = hsv(:, :, 3);
meanV = mean2(v);
normH=meanH/(meanH+meanS+meanV);
normV=meanV/(meanH+meanS+meanV);
if normH>0.079 & normH<0.136 && normV>0.584 & normV<0.687
    set(handles.edit4,'string','Bunga Alamanda');
elseif normH>0.448 & normH<0.529 && normV>0.139 & normV<0.139
    set(handles.edit4,'string','Bunga Dahlia');
elseif normH>0.385 & normH<0.495 && normV>0.403 & normV<0.550
    set(handles.edit4,'string','Bunga Kamboja');
elseif normH>0.409 & normH<0.468 & normV>0.095 & normV<0.195
    set(handles.edit4,'string','Bunga Mawar');
elseif normH>0.418 & normH<0.479 && normV>0.147 & normV<0.212
    set(handles.edit4,'string','Bunga Sepatu');
else set(handles.edit4,'string','tidak diketahui');
end

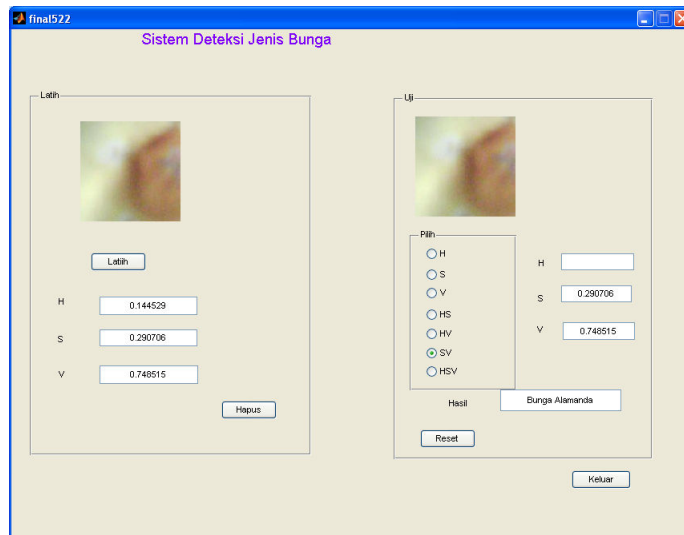
```

**Gambar 5.21** Listing gambar pada button HV

Keterangan Gambar 5.21 :

- a. Uigetfile untuk membuka file gambar.
- b. Imread untuk membaca citra berkestensi \*.jpg.
- c. Axes untuk tempat citra yang akan ditampilkan
- d. `h=hsv(:, :, :1)`; untuk mengambil nilai h
- e. `if normH>0.079 & normH<0.136 && normV>0.699 & normV<0.827` penggolongan jenis bunga berdasarkan nilai HV.
- f. `set(handles.edit4,'string','Bunga Alamanda');` menampilkan nama bunga.

- Untuk pengujian dengan acuan nilai SV  
Sistem akan mendeteksi objek dengan menggunakan acuan nilai mean S dan mean V yang sebelumnya sudah dilatih dan disimpan dalam data store.



**Gambar 5.22** Tampilan hasil uji dengan acuan nilai SV

```

[a,b]=uigetfile({'*.jpg'},'buka gbr');
if isequal ([a,b],[0,0]) %-----untuk mencegah error jika user tidak
jadi mengambil file-----%
    return
end
I=imread ([b,a],'jpg');
%cropping
I = imcrop(I,[ 135 135 135 135 ]);
axes(handles.axes2);
imshow (I);
%MENAMPILKAN NILAI MASING-MASING H,S,V
hsv= rgb2hsv(I);
h = hsv(:, :, 1); %untuk mengambil nilai h
meanH = mean2(h);
s = hsv(:, :, 2);
meanS = mean2(s);
v = hsv(:, :, 3);
meanV = mean2(v);
normS=meanS/(meanH+meanS+meanV);
normV=meanV/(meanH+meanS+meanV);
if normS>0.212 & normS<0.302 && normV>0.584 & normV<0.687
    set(handles.edit4,'string','Bunga Alamanda');
elseif normS>0.291 & normS<0.394 && normV>0.139 & normV<0.199
    set(handles.edit4,'string','Bunga Dahlia');
elseif normS>0.012 & normS<0.154 && normV>0.403 & normV<0.550
    set(handles.edit4,'string','Bunga Kamboja');
elseif normS>0.365 & normS<0.468 && normV>0.095 & normV<0.195
    set(handles.edit4,'string','Bunga Mawar');
elseif normS>0.329 & normS<0.414 && normV>0.147 & normV<0.212
    set(handles.edit4,'string','Bunga Sepatu');
else set(handles.edit4,'string','tidak diketahui');
end

```

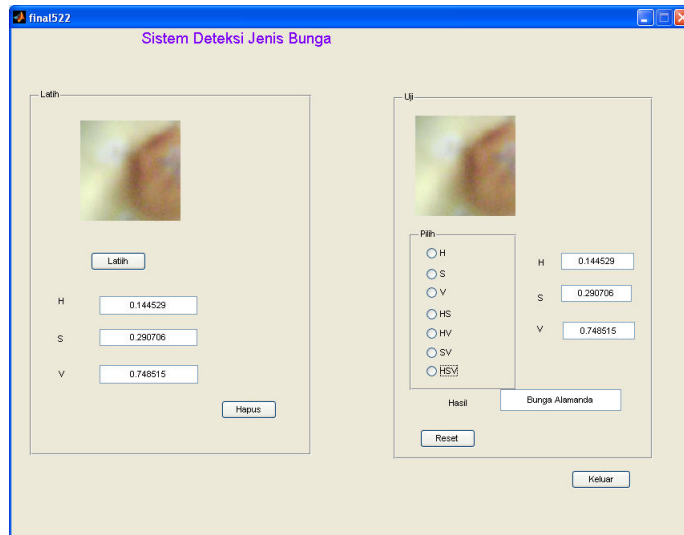
**Gambar 5.23** Listing gambar pada button SV

Keterangan Gambar 5.23 :

- a. Uigetfile untuk membuka file gambar.
- b. Imread untuk membaca citra berkeestensi \*.jpg.
- c. Axes untuk tempat citra yang akan ditampilkan
- d. `h=hsv(:, :, 1);` untuk mengambil nilai h
- e. `if normS>0.212 & normS<0.302 && normV>0.584 & normV<0.687` penggolongan jenis bunga berdasarkan nilai SV.
- f. `set(handles.edit4,'string','Bunga Alamanda');` menampilkan nama bunga.

- Untuk pengujian dengan acuan nilai HSV

Sistem akan mendeteksi objek dengan menggunakan acuan nilai mean H, mean S dan mean V yang sebelumnya sudah dilatih dan disimpan dalam data store.



**Gambar 5.24** Tampilan hasil uji dengan acuan nilai HSV

```

[a,b]=uigetfile({'*.jpg'}, 'buka gbr');
if isequal ([a,b],[0,0]) %-----untuk mencegah error jika user tidak
jadi mengambil file-----%
    return
end
I=imread ([b,a], 'jpg');
%cropping
I = imcrop(I,[ 135 135 135 135 ]);
axes(handles.axes2);
imshow (I);
%MENAMPILKAN NILAI MASING-MASING H,S,V
hsv= rgb2hsv(I);
h = hsv(:, :, 1); %untuk mengambil nilai h
meanH = mean2(h);
s = hsv(:, :, 2);
meanS = mean2(s);
v = hsv(:, :, 3);
meanV = mean2(v);
normH=meanH/(meanH+meanS+meanV);
normS=meanS/(meanH+meanS+meanV);
normV=meanV/(meanH+meanS+meanV);
if normH>0.07905 & normH<0.13625 && normS>0.21217 & normS<0.30165 &&
normV>0.58380 & normV<0.68709
    set(handles.edit4,'string','Bunga Alamanda');
elseif normH>0.44756 & normH<0.52851 && normS>0.29129 & normS<0.39450 &&
normV>0.13923 & normV<0.19892
    set(handles.edit4,'string','Bunga Dahlia');
elseif normH>0.52807 & normH<0.75983 && normS>0.00105 & normS<0.24153 &&
normV>0.57700 & normV<0.81225
    set(handles.edit4,'string','Bunga Kamboja');
elseif normH>0.73797 & normH<0.90513 && normS>0.70169 & normS<0.85170 &&
normV>0.16058 & normV<0.38731
    set(handles.edit4,'string','Bunga Mawar');
elseif normH>0.67506 & normH<0.87230 && normS>0.57699 & normS<0.65477 &&
normV>0.22631 & normV<0.39593
    set(handles.edit4,'string','Bunga Sepatu');
else set(handles.edit4,'string','tidak diketahui');
end

```

**Gambar 5.25** Listing gambar pada button HSV

Keterangan Gambar 5.25 :

- g. Uigetfile untuk membuka file gambar.
- a. Imread untuk membaca citra berkecstensi \*.jpg.
- b. Axes untuk tempat citra yang akan ditampilkan
- c.  $h=hsv(:, :, 1)$ ; untuk mengambil nilai h
- d.  $if\ normH > 0.079 \ \& \ normH < 0.136 \ \&\& \ normS > 0.212 \ \&$   
 $normS < 0.302 \ \&\& \ normV > 0.584 \ \& \ normV < 0.687$  penggolongan  
jenis bunga berdasarkan nilai HSV.




e. `set(handles.edit4,'string','Bunga Alamanda');` menampilkan nama bunga.

Ulangi pengujian program dengan acuan nilai H, S, V, HS, HV, SV dan HSV masing-masing sebanyak 30 kali. Hasil dari 30 sampel pengujian lihat tabel 5.7



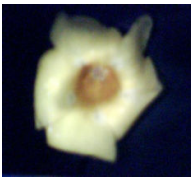
### 5.3.1. HASIL PENGUJIAN

Data hasil pengujian bunga yang sudah dilakukan sebagai berikut :

**Tabel 5.8** Hasil Uji Bunga Alamanda

Da ta	Citra Uji	H	S	V	HS	HV	SV	HSV
1		Tidak diketa hui	Bunga Alam anda	Tidak diketa hui	Tidak diketa hui	Tidak diketa hui	Tidak diketa hui	Tidak diketa hui
2		Bunga Alam anda	Bunga Alam anda	Bunga Alam anda	Bunga Alam anda	Bunga Alam anda	Bunga Alam anda	Bunga Alam anda
3		Bunga Alam anda	Bunga Alam anda	Bunga Alam anda	Bunga Alam anda	Bunga Alam anda	Bunga Alam anda	Bunga Alam anda



4		Tidak diketa hui	Tidak diketa hui	Bunga Alam anda	Tidak diketa hui	Tidak diketa hui	Tidak diketa hui	Tidak diketa hui
5		Bunga Alam anda	Bunga Alam anda	Bunga Alam anda	Bunga Alam anda	Bunga Alam anda	Bunga Alam anda	Bunga Alam anda
6		Bunga Alam anda	Bunga Alam anda	Bunga Alam anda	Bunga Alam anda	Bunga Alam anda	Bunga Alam anda	Bunga Alam anda
Keberhasilan		66.67 %	83.33 %	83.33 %	66.67 %	66.67 %	66.67 %	66.67 %
Kegagalan		33.33 %	16.67 %	16.67 %	33.33 %	33.33 %	33.33 %	33.33 %

Dari 6 sampel data yang di uji pada masing-masing acuan nilai H, S, V, HS, HV, SV dan HSV dapat diperoleh persentase keberhasilan sebagai berikut :

$$\text{Nilai H} = \frac{4}{6} \times 100\% = 66.67\%$$

$$\text{Nilai S} = \frac{5}{6} \times 100\% = 83.33\%$$

$$\text{Nilai V} = \frac{5}{6} \times 100\% = 83.33\%$$



$$\text{Nilai HS} = \frac{4}{6} \times 100\% = 66.67\%$$




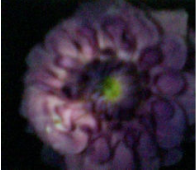
$$\text{Nilai HV} = \frac{4}{6} \times 100\% = 66.67\%$$

$$\text{Nilai SV} = \frac{4}{6} \times 100\% = 66.67\%$$

$$\text{Nilai HSV} = \frac{4}{6} \times 100\% = 66.67\%$$

**Tabel 5.9** Hasil Uji Bunga Dahlia


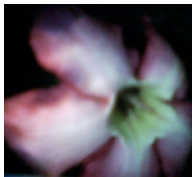


Da ta	Citra Uji	H	S	V	HS	HV	SV	HSV
1		Bun ga Dahl ia	Bunga Alama nda	Tidak diketa hui	Tidak diketa hui	Tidak diketa hui	Tidak diketa hui	Tidak diketa hui
2		Bun ga Dahl ia	Bunga Alama nda	Tidak diketa hui	Tidak diketa hui	Tidak diketa hui	Tidak diketa hui	Tidak diketa hui



3		Bun ga Dahl ia	Bunga Alama nda	Tidak diketa hui	Tidak diketa hui	Tidak diketa hui	Tidak diketa hui	Tidak diketa hui
4		Bun ga Dahl ia	Bunga Alama nda	Tidak diketa hui	Tidak diketa hui	Tidak diketa hui	Tidak diketa hui	Tidak diketa hui
5		Bun ga Dahl ia	Bunga Alama nda	Tidak diketa hui	Tidak diketa hui	Tidak diketa hui	Tidak diketa hui	Tidak diketa hui
6		Bun ga Dahl ia	Bunga Alama nda	Tidak diketa hui	Tidak diketa hui	Tidak diketa hui	Tidak diketa hui	Tidak diketa hui
% Keberhasilan		100 %	0%	0%	0%	0%	0%	0%
% Kegagalan		0%	100%	100%	100%	100%	100%	100%

Dari 6 sampel data yang di uji pada masing-masing acuan nilai H, S, V, HS, HV, SV dan HSV dapat diperoleh persentase keberhasilan sebagai berikut :

$$\text{Nilai H} = \frac{6}{6} \times 100\% = 100\%$$

**Tabel 5.10** Hasil Uji Bunga Kamboja

Da ta	Citra Uji	H	S	V	HS	HV	SV	HSV
1		Tidak diketa hui	Tidak diketah ui	Bunga Kamb oja	Tidak diketa hui	Tidak diketa hui	Tidak diketa hui	Tidak diketa hui
2		Bunga Dahlia	Bunga Kambo ja	Tidak diketa hui	Bunga Kamb oja	Bunga Mawa r	Tidak diketa hui	Tidak diketa hui
3		Bunga Dahlia	Bunga Alama nda	Tidak diketa hui	Tidak diketa hui	Bunga Mawa r	Tidak diketa hui	Tidak diketa hui
4		Bunga Kamb oja	Bunga Alama nda	Tidak diketa hui	Tidak diketa hui	Bunga Mawa r	Tidak diketa hui	Tidak diketa hui





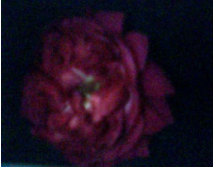
5		Bunga Dahlia	Bunga Alamanda	Tidak diketahui	Tidak diketahui	Bunga Mawar	Tidak diketahui	Tidak diketahui
6		Bunga Dahlia	Tidak diketahui	Tidak diketahui	Tidak diketahui	Tidak diketahui	Tidak diketahui	Tidak diketahui
% Keberhasilan		16.67 %	16.67 %	16.67 %	16.67 %	0%	0%	0%
% Kegagalan		83.33 %	83.33 %	83.33 %	83.33 %	100%	100%	100%


Dari 6 sampel data yang di uji pada masing-masing acuan nilai H, S, V, HS, HV, SV dan HSV dapat diperoleh persentase keberhasilan sebagai berikut :

$$\text{Nilai H, Nilai S, Nilai V dan Nilai HS} = \frac{1}{6} \times 100\% = 16.67\%.$$

**Tabel 5. 11** Hasil Uji Bunga Mawar

Dat	Citra Uji	H	S	V	HS	HV	SV	HSV
a								

1		Bunga Dahlia	Bung a Mawa r	Bung a Dahl ia	Bung a Maw ar	Bung a Maw ar	Bung a Mawa r	Tidak diketah ui
2		Bunga Dahlia	Bung a Dahli a	Bung a Dahl ia	Bung a Dahli a	Bung a Maw ar	Bung a Dahli a	Bunga Dahlia
3		Bunga Dahlia	Bung a Dahli a	Bung a Dahl ia	Bung a Dahli a	Bung a Maw ar	Bung a Dahli a	Bunga Dahlia
4		Bunga Kambo ja	Bung a Dahli a	Bung a Dahl ia	Bung a Maw ar	Bung a Maw ar	Bung a Dahli a	Tidak diketah ui
5		Bunga Dahlia	Bung a Dahli a	Bung a Dahl ia	Bung a Dahli a	Bung a Maw ar	Bung a Dahli a	Bunga Dahlia

6		Bunga Kambo ja	Bung a Mawa r	Bung a Dahl ia	Bung a Maw ar	Bung a Maw ar	Bung a Mawa r	Tidak diketah ui
% Keberhasilan		0%	33.33 %	0%	50%	100 %	33.33 %	0%
% Kegagalan		100%	66.67 %	100 %	50%	0%	66.67 %	100%

Dari 6 sampel data yang di uji pada masing-masing acuan nilai H, S, V, HS, HV, SV dan HSV dapat diperoleh persentase keberhasilan sebagai berikut :

$$\text{Nilai S} = \frac{2}{6} \times 100\% = 33.33 \%$$

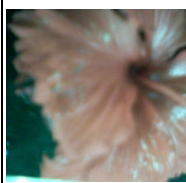
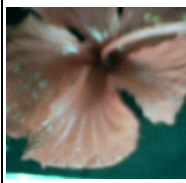
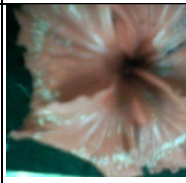

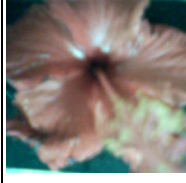
$$\text{Nilai HS} = \frac{3}{6} \times 100\% = 50 \%$$

$$\text{Nilai HV} = \frac{6}{6} \times 100\% = 100 \%$$

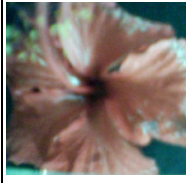
$$\text{Nilai SV} = \frac{2}{6} \times 100\% = 33.33 \%$$

**Tabel 5. 12** Hasil Uji Bunga Sepatu

Da ta	Citra Uji	H	S	V	HS	HV	SV	HSV
----------	-----------	---	---	---	----	----	----	-----

1		Bunga Alama nda	Bunga Alama nda	Bunga Alama nda	Bunga Alama nda	Bunga Alama nda	Bunga Alama nda	Bunga Alama nda
2		Tidak diketa hui	Tidak diketa hui	Bunga Kamb oja	Tidak diketa hui	Tidak diketa hui	Tidak diketa hui	Tidak diketa hui
3		Tidak diketa hui	Bunga Mawa r	Bunga Kamb oja	Tidak diketa hui	Tidak diketa hui	Tidak diketa hui	Tidak diketa hui
4		Tidak diketa hui	Tidak diketa hui	Bunga Kamb oja	Tidak diketa hui	Tidak diketa hui	Tidak diketa hui	Tidak diketa hui
5		Tidak diketa hui	Bunga Dahlia	Bunga Kamb oja	Tidak diketa hui	Tidak diketa hui	Tidak diketa hui	Tidak diketa hui



6		Tidak diketa hui	Bunga Mawa r	Tidak diketa hui	Tidak diketa hui	Tidak diketa hui	Tidak diketa hui	Tidak diketa hui
% Keberhasilan		0%	0%	0%	0%	0%	0%	0%
% Kegagalan		100%	100%	100%	100%	100%	100%	100%

Dari 6 sampel data bunga sepatu yang di uji dengan menggunakan acuan nilai H,S,V,HS,HV,SV dan HSV sistem tidak dapat mendeteksi dengan sempurna. Dengan prosentase kegagalan mencapai 100%.

**Tabel 5.13** Rata-rata presentase keberhasilan

No	Nama Bunga	%H	%S	%V	%HS	%HV	%SV	%HSV
1	Alamanda	66.67	83.3	83.3	66.7	66.7	66.7	66.67
2	Dahlia	100	0	0	0	0	0	0
3	Kamboja	16.67	16.7	16.7	16.7	0	0	0
4	Mawar	0	33.3	0	50	100	33.3	0
5	Sepatu	0	0	0	0	0	0	0
% Keberhasilan		36.668	26.7	20	26.7	33.3	20	13.33

## **BAB VI**

### **PENUTUP**

#### **6.1 Kesimpulan**

Dari hasil penelitian, dapat disimpulkan sebagai berikut :

Rata – rata keberhasilan sistem untuk mendeteksi objek dengan menggunakan nilai HSV adalah dengan acuan nilai Hue sebesar 36.68%, acuan nilai Saturation sebesar 26.67 %, acuan nilai Value sebesar 20 %. Gabungan nilai Hue-Saturation sebesar 26.67%, nilai Hue-Value sebesar 33,33%, nilai Saturation-Value 20% dan Hue-Saturation-Value sebesar 13.33%. Sehingga untuk mendeteksi objek dengan menggunakan nilai HSV, sebaiknya menggunakan nilai acuan Hue karena nilai Hue memberikan tingkat keberhasilan sistem lebih tinggi dibanding dengan acuan nilai yang lain.

#### **6.2 Saran**

Hasil deteksi masih kurang akurat diharapkan kedepannya dapat dikembangkan dengan metode lain yang lebih baik.

1. Perhitungan dan penyimpanan data masih menggunakan cara manual, agar dikembangkan perhitungan secara otomatis dengan memanfaatkan fungsi-fungsi pada aplikasi matlab.
2. Sampel data yang digunakan dalam pengujian sistem masih sedikit, maka perlu dicoba dengan sampel yang lebih banyak.

## DAFTAR PUSTAKA

- Agus Prijono, Marvin Ch. Wijaya, 2007, *Pengolahan Citra Digital Menggunakan Matlab*, Cetakan Pertama, Informatika, Bandung.
- Agus Purwo Handoko, Yustina Retno Wahyu Utami, 2009. *Pengenalan Buah Berdasarkan Karakteristik Warna Citra*. CSRID 1, 114-120.
- Efendi,Susanto, 2010, *Deteksi Image Digital Menggunakan Metode Edge Detection*. Teknik Informatika, Universitas Stikubank Semarang.
- Jati Sasongko Wibowo, 2011, *Deteksi dan Klasifikasi Cara Berdasarkan Warna Kulit Menggunakan HSV*. Dinamik 16, 118-123.
- Jogiyanto, 1990, *Analisis dan Desain Sistem Operasi*. Andi Offset, Yogyakarta.
- Munir,Rinaldi, 2004, *Pengolahan citra Digital dengan Pendekatan Algoritmik*, Cetakan Pertama, Informatika, Bandung.
- Nur Indah Fadjariani, 2010, Aplikasi Monitoring Pajak, [online], (<http://www.scribd.com/doc/50064628/17/Diagram-Konteks>), diakses tanggal 13 Januari 2012 pukul 20.30 WIB)
- Pressman, Roger S, 2002, *Rekayasa Perangkat Lunak ; Pendekatan Praktisi* ( buku 1). Andi.
- Putra,Darma. *Pengolahan Citra Digital*. Andi Offset, Yogyakarta. 2010.
- Sukendar,Hari, 2010, *Identifikasi Buah Menggunakan Ciri Warna dan Bentuk*. Teknik Informatika, Universitas Dian Nuswantoro Semarang.

“\_\_\_\_\_”, *Landasan Teori*, [pdf],  
(<http://library.binus.ac.id/eColls/eThesis/Bab2/2007-2-00459-MTIF-Bab%202.pdf>, diakses tanggal 15 November 2011 pukul 09.00 WIB)

“\_\_\_\_\_”, *Landasan Teori*, [online]  
(<http://digilib.petra.ac.id/viewer.php?page=1&submit.x=0&submit.y=0&query=high&fname=/jiunkpe/s1/info/2009/jiunkpe-ns-s1-2009-26406036-14553-karikatur-chapter2.pdf>, diakses tanggal 3 Maret 2012 pukul 19.15 WIB)

“\_\_\_\_\_”, *Landasan Teori*, [pdf]  
(<http://library.binus.ac.id/eColls/eThesis/Bab2/2012-1-00523-mtif%202.pdf>, diakses tanggal 9 Desember 2011 pukul 20.25 WIB)

## LISTING PROGRAM

```
function varargout = final522(varargin)
% FINAL522 M-file for final522.fig
%     FINAL522, by itself, creates a new FINAL522 or raises the
existing
%     singleton*.
%
%     H = FINAL522 returns the handle to a new FINAL522 or the
handle to
%     the existing singleton*.
%
%     FINAL522('CALLBACK', hObject,eventData,handles,...) calls
the local
%     function named CALLBACK in FINAL522.M with the given input
arguments.
%
%     FINAL522('Property','Value',...) creates a new FINAL522 or
raises the
%     existing singleton*. Starting from the left, property
value pairs are
%     applied to the GUI before final522_OpeningFunction gets
called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to final522_OpeningFcn via
varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help final522

% Last Modified by GUIDE v2.5 19-Feb-2013 09:46:15

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @final522_OpeningFcn, ...
                  'gui_OutputFcn',  @final522_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
```

```

% End initialization code - DO NOT EDIT

% --- Executes just before final522 is made visible.
function final522_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject handle to figure
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)
% varargin command line arguments to final522 (see VARARGIN)

% Choose default command line output for final522
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes final522 wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = final522_OutputFcn(hObject, eventdata,
handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton1 (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)

[a,b]=uigetfile({'*.jpg'}, 'buka gbr');
if isequal ([a,b],[0,0])
    return
end
I=imread ([b,a], 'jpg');
axes(handles.axes1);
I = imcrop(I, [ 135 135 135 135 ]);
imshow(I);
%cropping

%MENAMPILKAN NILAI MASING-MASING H,S,V
hsv= rgb2hsv(I);
h =hsv(:, :, 1); %untuk mengambil nilai h
meanH = mean2(h);
s = hsv(:, :, 2);
meanS = mean2(s);

```

```

v = hsv(:, :, 3);
meanV = mean2(v);
set(handles.edit1, 'string', meanH);
set(handles.edit2, 'string', meanS);
set(handles.edit3, 'string', meanV);

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton2 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)
[a,b]=uigetfile({'*.jpg'}, 'buka gbr');
if isequal ([a,b],[0,0]) %-----untuk mencegah error jika user
tidak jadi mengambil file-----%
    return
end
I=imread ([b,a], 'jpg');
%cropping
I = imcrop(I, [ 135 135 135 135 ]);
axes(handles.axes2);
imshow (I);

function edit1_Callback(hObject, eventdata, handles)
% hObject      handle to edit1 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of
edit1 as a double

% --- Executes during object creation, after setting all
properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit1 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc      &&      isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)
% hObject      handle to edit2 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text

```

```

%          str2double(get(hObject,'String')) returns contents of
edit2 as a double

% --- Executes during object creation, after setting all
properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if    ispc    &&    isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of edit3 as text
%          str2double(get(hObject,'String')) returns contents of
edit3 as a double

% --- Executes during object creation, after setting all
properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if    ispc    &&    isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of edit4 as text
%          str2double(get(hObject,'String')) returns contents of
edit4 as a double
% --- Executes during object creation, after setting all
properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)

```



```

% eventdata reserved - to be defined in a future version of
MATLAB
% handles empty - handles not created until after all
CreateFcns called
% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton3 (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)
keluar = questdlg('Yakin akan keluar?', ...
    'Main', ...
    'Ya', 'Tidak', 'Ya');

switch keluar,
    case 'Ya',
        clear handles.Citra;
        clear all;
        close;
    case 'Tidak',
        return
end

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton4 (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)
set(handles.edit1,'string','');
set(handles.edit2,'string','');
set(handles.edit3,'string','');
axes(handles.axes1);
plot(0);

% --- If Enable == 'on', executes on mouse press in 5 pixel
border.
% --- Otherwise, executes on mouse press in 5 pixel border or over
pushbutton2.
function pushbutton2_ButtonDownFcn(hObject, eventdata, handles)
% hObject handle to pushbutton2 (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)

% --- Executes on key press over radiobutton1 with no controls
selected.
function radiobutton1_KeyPressFcn(hObject, eventdata, handles)
% hObject handle to radiobutton1 (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)

% --- Executes on key press over pushbutton2 with no controls
selected.
function pushbutton2_KeyPressFcn(hObject, eventdata, handles)
% hObject handle to pushbutton2 (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)

% --- Executes on button press in H.
function H_Callback(hObject, eventdata, handles)
% hObject handle to H (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of H

[a,b]=uigetfile({'*.jpg'},'buka gbr');
if isequal ([a,b],[0,0]) %-----untuk mencegah error jika user
tidak jadi mengambil file-----%
    return
end
I=imread ([b,a], 'jpg');
%cropping
I = imcrop(I,[ 135 135 135 135 ]);
axes(handles.axes2);
imshow (I);
%MENAMPILKAN NILAI MASING-MASING H,S,V
hsv= rgb2hsv(I);
h = hsv(:, :, 1); %untuk mengambil nilai h
s = hsv(:, :, 2);
v = hsv(:, :, 3);
meanH = mean2(h);
meanS = mean2(s);
meanV = mean2(v);
normH=meanH/(meanH+meanS+meanV);
set(handles.edit9,'string',meanH);
set(handles.edit10,'String','');
set(handles.edit11,'String','');
if normH>0.07904 && normH<0.13625
    set(handles.edit4,'string','Bunga Alamanda');
elseif normH>0.44756 && normH<0.52851
    set(handles.edit4,'string','Bunga Dahlia');
elseif normH>0.38528 && normH<0.49521
    set(handles.edit4,'string','Bunga Kamboja');
elseif normH>0.40947 && normH<0.46817
    set(handles.edit4,'string','Bunga Mawar');
elseif normH>0.41756 && normH<0.47933
    set(handles.edit4,'string','Bunga Sepatu');
else set(handles.edit4,'string','tidak diketahui');
end

```

```

% --- Executes on button press in S.
function S_Callback(hObject, eventdata, handles)
% hObject      handle to S (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of S
[a,b]=uigetfile({'*.jpg'},'buka gbr');
if isequal ([a,b],[0,0]) %-----untuk mencegah error jika user
tidak jadi mengambil file-----%
    return
end
I=imread ([b,a], 'jpg');
%cropping
I = imcrop(I,[ 135 135 135 135 ]);
axes(handles.axes2);
imshow (I);
%MENAMPILKAN NILAI MASING-MASING H,S,V
hsv= rgb2hsv(I);
h = hsv(:, :, 1); %untuk mengambil nilai h
meanH = mean2(h);
s = hsv(:, :, 2);
meanS = mean2(s);
v = hsv(:, :, 3);
meanV = mean2(v);
normS=meanS/(meanH+meanS+meanV);
set(handles.edit10,'string',meanS);
set(handles.edit9,'String','');
set(handles.edit11,'String','');
if normS>0.21217 && normS<0.30165
    set(handles.edit4,'string','Bunga Alamanda');
elseif normS>0.29129 && normS<0.39450
    set(handles.edit4,'string','Bunga Dahlia');
elseif normS>0.01248 && normS<0.15368
    set(handles.edit4,'string','Bunga Kamboja');
elseif normS>0.36464 && normS<0.46761
    set(handles.edit4,'string','Bunga Mawar');
elseif normS>0.32939 && normS<0.41387
    set(handles.edit4,'string','Bunga Sepatu');
else set(handles.edit4,'string','tidak diketahui');
end

% --- Executes on button press in V.
function V_Callback(hObject, eventdata, handles)
% hObject      handle to V (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of V

[a,b]=uigetfile({'*.jpg'},'buka gbr');
if isequal ([a,b],[0,0]) %-----untuk mencegah error jika user
tidak jadi mengambil file-----%
    return
end

```

```

I=imread ([b,a], 'jpg');
%cropping
I = imcrop(I, [ 135 135 135 135 ]);
axes(handles.axes2);
imshow (I);
%MENAMPILKAN NILAI MASING-MASING H,S,V
hsv= rgb2hsv(I);
h = hsv(:, :, 1); %untuk mengambil nilai h
meanH = mean2(h);
s = hsv(:, :, 2);
meanS = mean2(s);
v = hsv(:, :, 3);
meanV = mean2(v);
normV=meanV/(meanH+meanS+meanV);
set(handles.edit11, 'string', meanV);
set(handles.edit9, 'String', '');
set(handles.edit10, 'String', '');
if normV>0.58380 && normV<0.68709
    set(handles.edit4, 'string', 'Bunga Alamanda');
elseif normV>0.13923 && normV<0.19892
    set(handles.edit4, 'string', 'Bunga Dahlia');
elseif normV>0.40323 && normV<0.55012
    set(handles.edit4, 'string', 'Bunga Kamboja');
elseif normV>0.09510 && normV<0.19501
    set(handles.edit4, 'string', 'Bunga Mawar');
elseif normV>0.14748 && normV<0.21237
    set(handles.edit4, 'string', 'Bunga Sepatu');
else set(handles.edit4, 'string', 'tidak diketahui');
end

%function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of edit1 as text
%         str2double(get(hObject,'String')) returns contents of
edit1 as a double
% --- Executes during object creation, after setting all
properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in HS.
function HS_Callback(hObject, eventdata, handles)

```

```

% hObject      handle to HS (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of HS
[a,b]=uigetfile({'*.jpg'},'buka gbr');
if isequal ([a,b],[0,0]) %-----untuk mencegah error jika user
tidak jadi mengambil file-----%
    return
end
I=imread ([b,a], 'jpg');
%cropping
I = imcrop(I,[ 135 135 135 135 ]);
axes(handles.axes2);
imshow (I);
%MENAMPILKAN NILAI MASING-MASING H,S,V
hsv= rgb2hsv(I);
h = hsv(:, :, 1); %untuk mengambil nilai h
meanH = mean2(h);
s = hsv(:, :, 2);
meanS = mean2(s);
v = hsv(:, :, 3);
meanV = mean2(v);
normH=meanH/(meanH+meanS+meanV);
normS=meanS/(meanH+meanS+meanV);
set(handles.edit9,'String',meanH);
set(handles.edit10,'string',meanS);
set(handles.edit11,'String','');
if normH>0.07904 & normH<0.13625 && normS>0.21217 & normS<0.30165
    set(handles.edit4,'string','Bunga Alamanda');
elseif normH>0.44756 & normH<0.52851 && normS>0.29129 &
normS<0.39450
    set(handles.edit4,'string','Bunga Dahlia');
elseif normH>0.38528 & normH<0.49521 && normS>0.001248 &
normS<0.15368
    set(handles.edit4,'string','Bunga Kamboja');
elseif normH>0.40947 & normH<0.46817 && normS>0.36464 &
normS<0.46761
    set(handles.edit4,'string','Bunga Mawar');
elseif normH>0.41756 & normH<0.47933 && normS>0.32939 &
normS<0.41387
    set(handles.edit4,'string','Bunga Sepatu');
else set(handles.edit4,'string','tidak diketahui');
end

% --- Executes on button press in HV.
function HV_Callback(hObject, eventdata, handles)
% hObject      handle to HV (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of HV
[a,b]=uigetfile({'*.jpg'},'buka gbr');
if isequal ([a,b],[0,0]) %-----untuk mencegah error jika user
tidak jadi mengambil file-----%
    return

```

```

end
I=imread ([b,a], 'jpg');
%cropping
I = imcrop(I, [ 135 135 135 135 ]);
axes(handles.axes2);
imshow (I);
%MENAMPILKAN NILAI MASING-MASING H,S,V
hsv= rgb2hsv(I);
h = hsv(:, :, 1); %untuk mengambil nilai h
meanH = mean2(h);
s = hsv(:, :, 2);
meanS = mean2(s);
v = hsv(:, :, 3);
meanV = mean2(v);
normH=meanH/(meanH+meanS+meanV);
normV=meanV/(meanH+meanS+meanV);
set(handles.edit9, 'String', meanH);
set(handles.edit10, 'string', '');
set(handles.edit11, 'String', meanV);
if normH>0.079 & normH<0.136 && normV>0.584 & normV<0.687
    set(handles.edit4, 'string', 'Bunga Alamanda');
elseif normH>0.448 & normH<0.529 && normV>0.139 & normV<0.139
    set(handles.edit4, 'string', 'Bunga Dahlia');
elseif normH>0.385 & normH<0.495 && normV>0.403 & normV<0.550
    set(handles.edit4, 'string', 'Bunga Kamboja');
elseif normH>0.409 & normH<0.468 & normV>0.095 & normV<0.195
    set(handles.edit4, 'string', 'Bunga Mawar');
elseif normH>0.418 & normH<0.479 && normV>0.147 & normV<0.212
    set(handles.edit4, 'string', 'Bunga Sepatu');
else set(handles.edit4, 'string', 'tidak diketahui');
end

% --- Executes on button press in SV.
function SV_Callback(hObject, eventdata, handles)
% hObject    handle to SV (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of SV
[a,b]=uigetfile({'*.jpg'}, 'buka gbr');
if isequal ([a,b],[0,0]) %-----untuk mencegah error jika user
tidak jadi mengambil file-----%
    return
end
I=imread ([b,a], 'jpg');
%cropping
I = imcrop(I, [ 135 135 135 135 ]);
axes(handles.axes2);
imshow (I);
%MENAMPILKAN NILAI MASING-MASING H,S,V
hsv= rgb2hsv(I);
h = hsv(:, :, 1); %untuk mengambil nilai h
meanH = mean2(h);
s = hsv(:, :, 2);
meanS = mean2(s);
v = hsv(:, :, 3);

```

```

meanV = mean2(v);
normS=meanS/(meanH+meanS+meanV);
normV=meanV/(meanH+meanS+meanV);
set(handles.edit9,'String','');
set(handles.edit10,'string',meanS);
set(handles.edit11,'String',meanV);
if normS>0.212 & normS<0.302 && normV>0.584 & normV<0.687
    set(handles.edit4,'string','Bunga Alamanda');
elseif normS>0.291 & normS<0.394 && normV>0.139 & normV<0.199
    set(handles.edit4,'string','Bunga Dahlia');
elseif normS>0.012 & normS<0.154 && normV>0.403 & normV<0.550
    set(handles.edit4,'string','Bunga Kamboja');
elseif normS>0.365 & normS<0.468 && normV>0.095 & normV<0.195
    set(handles.edit4,'string','Bunga Mawar');
elseif normS>0.329 & normS<0.414 && normV>0.147 & normV<0.212
    set(handles.edit4,'string','Bunga Sepatu');
else set(handles.edit4,'string','tidak diketahui');
end

% --- Executes on button press in HSV.
function HSV_Callback(hObject, eventdata, handles)
% hObject    handle to HSV (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of HSV
[a,b]=uigetfile({'*.jpg'},'buka gbr');
if isequal ([a,b],[0,0]) %-----untuk mencegah error jika user
tidak jadi mengambil file-----%
    return
end
I=imread ([b,a], 'jpg');
%cropping
I = imcrop(I,[ 135 135 135 135 ]);
axes(handles.axes2);
imshow (I);
%MENAMPILKAN NILAI MASING-MASING H,S,V
hsv= rgb2hsv(I);
h = hsv(:, :, 1); %untuk mengambil nilai h
meanH = mean2(h);
s = hsv(:, :, 2);
meanS = mean2(s);
v = hsv(:, :, 3);
meanV = mean2(v);
normH=meanH/(meanH+meanS+meanV);
normS=meanS/(meanH+meanS+meanV);
normV=meanV/(meanH+meanS+meanV);
set(handles.edit9,'String',meanH);
set(handles.edit10,'string',meanS);
set(handles.edit11,'String',meanV);
if normH>0.07905 & normH<0.13625 && normS>0.21217 & normS<0.30165
&& normV>0.58380 & normV<0.68709
    set(handles.edit4,'string','Bunga Alamanda');
elseif normH>0.44756 & normH<0.52851 && normS>0.29129 &
normS<0.39450 && normV>0.13923 & normV<0.19892
    set(handles.edit4,'string','Bunga Dahlia');

```

```

elseif normH>0.52807 & normH<0.75983 && normS>0.00105 &
normS<0.24153 && normV>0.57700 & normV<0.81225
    set(handles.edit4,'string','Bunga Kamboja');
elseif normH>0.73797 & normH<0.90513 && normS>0.70169 &
normS<0.85170 && normV>0.16058 & normV<0.38731
    set(handles.edit4,'string','Bunga Mawar');
elseif normH>0.67506 & normH<0.87230 && normS>0.57699 &
normS<0.65477 && normV>0.22631 & normV<0.39593
    set(handles.edit4,'string','Bunga Sepatu');
else set(handles.edit4,'string','tidak diketahui');
end

% -----
% -----
function uipanel2_SelectionChangeFcn(hObject, eventdata, handles)
% hObject    handle to uipanel2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

%if hObject,'Value'==handles.H
% set(get(hObject,'Value')) ;
%elseif
% hObject==handles.S
% set(handles.edit4,'String','S');
% else
% set(handles.edit4,'String','V');
%end

function edit6_Callback(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
% str2double(get(hObject,'String')) returns contents of
edit6 as a double

% --- Executes during object creation, after setting all
properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called
% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```



```

% --- Executes on button press in reset.
function reset_Callback(hObject, eventdata, handles)
% hObject      handle to reset (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)
set(handles.edit4,'String','');
set(handles.edit9,'String','');
set(handles.edit10,'String','');
set(handles.edit11,'String','');
axes(handles.axes2);
plot(0);
function edit7_Callback(hObject, eventdata, handles)
% hObject      handle to edit7 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of edit7 as text
%         str2double(get(hObject,'String')) returns contents of
edit7 as a double
% --- Executes during object creation, after setting all
properties.
function edit7_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit7 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all
properties.
function uipanel2_CreateFcn(hObject, eventdata, handles)
% hObject      handle to uipanel2 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

function edit8_Callback(hObject, eventdata, handles)
% hObject      handle to edit8 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of edit8 as text
%         str2double(get(hObject,'String')) returns contents of
edit8 as a double

% --- Executes during object creation, after setting all
properties.

```

```

function edit8_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit8 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc      &&      isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)
% hObject      handle to edit9 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of edit9 as text
%           str2double(get(hObject,'String')) returns contents of
edit9 as a double
% --- Executes during object creation, after setting all
properties.
function edit9_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit9 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc      &&      isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)
% hObject      handle to edit10 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit10 as text
%           str2double(get(hObject,'String')) returns contents of
edit10 as a double
% --- Executes during object creation, after setting all
properties.
function edit10_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit10 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%           See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)
% hObject    handle to edit11 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of edit11 as text
%         str2double(get(hObject,'String')) returns contents of
edit11 as a double
% --- Executes during object creation, after setting all
properties.
function edit11_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit11 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```