

28/09/14/07



ITS
Institut
Teknologi
Sepuluh Nopember

RSif
006.42
Wah
i-1

2007

TESIS - CL 2541

IDENTIFIKASI CITRA BUAH BERDASARKAN DISTRIBUSI WARNA MENGUNAKAN HISTOGRAM INDEKS

AGUNG WAHYUDI
NRP. 5102 201 008

DOSEN PEMBIMBING
Febriyana Samopa, S.Kom., M.Kom.

PROGRAM MAGISTER
BIDANG KEAHLIAN TEKNIK INFORMATIKA
PROGRAM STUDI/JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2007

PERPUSTAKAAN ITS	
Tgl. Terima	28 - 2 - 2007
Terima Dari	H
No. Agenda Prp.	227554



TEKNIK - 01 2011

IDENTIFIKASI CIRI BUAH
BERDASARAN DISTRIBUSI WARNA
MENGUNAKAN HISTOGRAM KOLORS

ADANG WAHYUDI
NRP. 2102 201 008

DOSEN PEMBIMBING
Fahmyan Samudra, S.Kom., M.Kom.

PROGRAM MAGISTER
BIDANG KEAHLIAN TEKNIK INFORMATIKA
PROGRAM STUDI JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMATIKA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2012

No. Urut	Nama File	Ukuran
1	1.jpg	1.2 MB
2	2.jpg	1.2 MB
3	3.jpg	1.2 MB
4	4.jpg	1.2 MB
5	5.jpg	1.2 MB

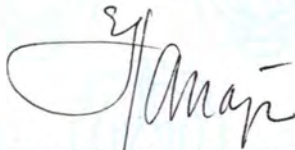
IDENTIFIKASI CITRA BUAH BERDASARKAN DISTRIBUSI WARNA MENGGUNAKAN HISTOGRAM INDEKS

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Komputer (M.Kom.)
di
Institut Teknologi Sepuluh Nopember Surabaya

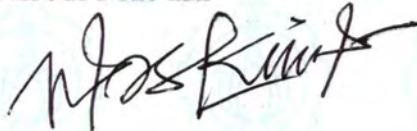
oleh :

Agung Wahyudi
Nrp. 5102.201.008

Disetujui oleh Tim Penguji Tesis :



1. **Ir. Esther Hanaja, M.Sc.**
NIP. 130 816 212



2. **Waskitho Wibisono, S.Kom., M.Eng.**
NIP. 132 256 272

3. **Irfan Surbakti, S.Kom., M.Eng.Sc.**
NIP. 132 300 412

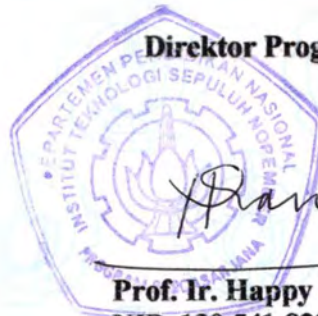
Dosen Pembimbing

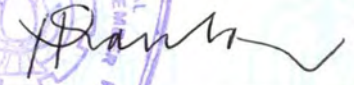


Febriliyan Samoga, S.Kom., M.Kom.
NIP. 132 206 858

Tanggal Ujia : 3 Oktober 2006
Periode Wisuda : Maret 2007

Direktor Program Pascasarjana,




Prof. Ir. Happy Ratna S., M.Sc., Ph.D
NIP. 130 541 829

IDENTIFIKASI CITRA BUAH BERDASARKAN DISTRIBUSI
WARNA MENGGUNAKAN NIS OGANI INDIKATOR

Tesis disusun untuk memenuhi salah satu syarat penyelesaian
Magister Komputer (S2.Kom) di
Fakultas Teknologi Informasi Himpunan Sarjana

oleh :

Angus Winardi
NIM. 2107.201.008

Dosen Pembimbing

Pembimbing dan Tim Penguji Tesis :

Febrianingrum, S.Kom, M.Kom,
NIP. 132 206 832

Dr. Robert Hanjaja, M.Sc.
NIP. 132 216 212

Wahidno Widiono, S.Kom, M.Eng.
NIP. 132 226 272

Tanggal Ujian : 3 Oktober 2006
Periode Wisuda : Maret 2007

Dr. Dedy S. Kurniawan, M.Eng.Sc.
NIP. 132 226 272

Direktor Program Pascasarjana

Prof. Dr. Hanny Ratus S. M. S. S. S.
NIP. 132 226 272

ABSTRAK

IDENTIFIKASI CITRA BUAH BERDASARKAN DISTRIBUSI WARNA MENGGUNAKAN HISTOGRAM INDEKS

Nama mahasiswa : Agung Wahyudi
NRP : 5102.201.008
Pembimbing : Febriliyan Samopa, S.Kom., M.Kom.

ABSTRAK

Sistem identifikasi citra buah dapat dilakukan berdasarkan gambar buah dengan memanfaatkan ciri dasar dari gambar yang berupa ciri warna, karena setiap buah memiliki warna yang spesifik. Ciri warna merupakan model distribusi warna yang dapat dinyatakan dalam bentuk histogram dari warna. Histogram warna secara sederhana merupakan distribusi kemunculan setiap warna sehingga perbedaan warna menyebabkan perbedaan yang signifikan pada vektor ciri yang dihasilkan.

Model ciri warna dengan memanfaatkan histogram ini dapat membedakan objek gambar yang mempunyai perbedaan warna, seperti pada gambar buah. Model histogram yang digunakan dalam beberapa sistem identifikasi dan retrieval dalam pengolahan gambar menggunakan teknik histogram indeks dan histogram segmentasi dengan Euclidian. Teknik histogram indeks merupakan teknik yang cukup baik dengan komputasi yang cepat karena warna dapat disajikan secara langsung dalam bentuk indeks tanpa perlu tahu warnanya apa. Dalam penelitian yang telah dikembangkan untuk keperluan image retrieval, teknik histogram indeks menggunakan 25 skala warna pada masing-masing elemen warna dengan format RGB. Karena penelitian ini menggunakan gambar buah yang variansnya lebih sedikit dari gambar-gambar yang digunakan, maka skala warna diturunkan hingga skala terkecil dengan hasil yang masih akurat.

Penurunan skala warna pada elemen warna diharapkan mampu mempercepat proses komputasi dengan mempertahankan akurasi. Sehingga pada akhirnya sistem identifikasi citra buah ini dapat berjalan secara online dengan waktu proses yang sesuai dengan kebutuhan identifikasi otomatis pada machine vision masa depan.

Hasil pengujian yang diperoleh dalam penelitian ini adalah bahwa histogram indeks dapat digunakan untuk mengidentifikasi citra buah dengan skala warna lebih besar atau sama dengan 8 dan berpengaruh secara non-linier terhadap waktu proses identifikasi.

Kata Kunci: histogram indeks, identifikasi citra buah, distribusi warna, skala warna

ABSTRACT

FRUIT IMAGE IDENTIFICATION BASED ON COLOR DISTRIBUTION USING INDEXING HISTOGRAM

By : Agung Wahyudi
Student Identity Number : 5102201008
Supervisor : Febriliyan Samopa, S.Kom, M.Kom.

ABSTRACT

Fruit image identification system can build based on image color using basic feature from image such as color feature, because every fruit have a specific color. Color feature is a color distribution model that can express in color histogram. By simply, color histogram is distribution of event for each color then color difference cause by significant difference on color feature that had resulted

Color feature model using this histogram can dispartate image object that have color difference, like as fruit images. Histogram model had used in some identification and retrieval system in image processing using indexing histogram technique and segmentation histogram with Euclidian distance. Indexing histogram is good techniques that have a fast computation because color can be expressed directly in indexing without have known the color. In research had developed for image retrieval usage, indexing histogram technique using 25 color-scales in each color element in RGB format. This research using fruit images that have variance smaller than images is used on the last research, then color-scale is decreased until minimum scale with the result is accurate.

Color-scale decreasing on color element is estimated have acceleration in computation process among accuration still good. Finally fruit image identification system can run online with processing time as like as automatically identification needed on future machine vision.

The result in this result is indexing histogram is used to identify fruit images with color-scale bigger or same than 8 and non linear dependent on identification time processing.

Keywords: indexing histogram, fruit image identification, color distribution, color-scale

Kupersembahkan khusus:
Isteriku tersayang Rizky Niswarini Utami,
anakku Chika dan Galih
saudara-saudaraku Suprayitno, Yayuk, Tulus dan Sugeng
serta ayah Bambang Wiratsongko

Yang Kukenang:
Ayah Dwidjosiswojo dan Ibu Mukilah
Dan doa yang selalu menyertai

hanyalah kejujuran,
yang menjadi pengikat ikhlas dan sabar
dalam langkah istiqomah
merengkuh rahmat Allah dibelantara
kehidupan ini,
karena dzat-Nya telah melingkupi yang dibutuhkan.
Subhanallah, marilah selalu tafakur padaNya.

KATA PENGANTAR

KATA PENGANTAR

Segala puji dan syukur penulis panjatkan kehadiran Tuhan yang Maha Esa atas segala rahmat dan hidayahnya sehingga penulis dapat menyelesaikan tesis ini yang berjudul “ *Identifikasi Citra Buah Berdasarkan Distribusi Warna Menggunakan Histogram Indeks*”, yang merupakan salah satu syarat untuk memperoleh gelar Magister Komputer pada Program Pascasarjana, Program studi Teknik informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember Surabaya.

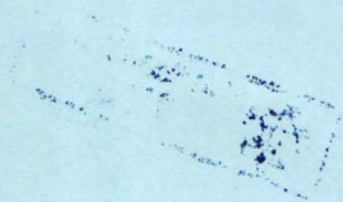
Penulis menyadari bahwa penulisan tesis ini tidak lepas dari bantuan dan dukungan berbagai pihak, baik moral maupun material. Untuk itu penulis menyampaikan terima kasih kepada :

1. Terutama penulis sampaikan terima kasih kepada isteri tersayang Rizky Niswarini Utami dan anak-anakku Chika dan galih yang tiada henti selalu mendorong dengan cinta, kasih dan doa untuk terselesaikannya tesis ini.
2. Bapak Febriliyan Samopa, S.Kom.,M.Kom, selaku dosen pembimbing tesis yang telah mendorong dan memberi pengarahan pada penulis untuk menyelesaikan tesis ini.
3. Bapak Prof. Drs. Ir. Riyanarto Sarno, M.Sc., Ph.D. ,selaku dosen wali penulis yang banyak memberikan dorongan pada penulis sehingga lebih memacu untuk menyelesaikan tesis ini.
4. Bapak Ir. F.X. Arunanto, M.Sc., selaku Ketua Program Studi Program Pascasarjana Teknik Informatika ITS yang telah banyak memberikan bantuan dan arahan untuk memecahkan permasalahan akademik.
5. Bapak Prof. Ir. Arif Djunaedi, M.Sc., Ph.D., selaku Dekan Fakultas Teknologi Informasi Institut Teknologi Sepuluh Nopember Surabaya.
6. Seluruh staf pengajar S-2 Teknik Informatika ITS yang telah memberi bekal ilmu pengetahuan bagi penulis, serta seluruh staf administrasi jurusan teknik informatika yang banyak membantu kelancaran proses studi di Program Studi ini.

DAFTAR ISI

AL-AMUDY
XEROX

DAFTAR ISI



DAFTAR ISI

Halaman Judul	i
Halaman Pengesahan	iii
Abstrak	v
Abstract	vii
Kata Pengantar	xv
Daftar Isi	xvii
Daftar gambar	xxi
Daftar Tabel	xxiii
Bab 1 Pendahuluan	1
1.1 Latar Belakang	1
1.2 Perumusan masalah	3
1.3 Tujuan dan Manfaat Penelitian	4
1.4 Batasan Penelitian	4
1.5 Kontribusi Penelitian	5
1.6 Sistematika Penulisan	5
Bab 2 Tinjauan Pustaka	7
2.1 Content Based Image Retrieval (CBIR)	7
2.2 Query by Image Content	9
2.2.1 Query Basis Data Citra	10
2.2.2 Query Berdasarkan Contoh	12
2.3 Konsep Warna	13
2.4 Histogram Warna	15
2.5 Indeks Warna dan Histogram Indeks	15
2.5.1 Indeks warna	16
2.5.2 Histogram Indeks	18
2.6 Basis Data Fitur	19
2.7 Teknik Matching	20



Bab 3 Perancangan	23
3.1 Blok Diagram Sistem	23
3.2 Penentuan Fitur Warna Dengan Histogram Indeks	25
3.2.1 Definisi dan Format warna	25
3.2.2 Indeks Warna	26
3.2.3 Histogram Indeks	30
3.3 Pembuatan Fitur	32
3.4 Penggabungan Fitur	33
3.5 Proses Identifikasi	37
Bab 4 Pengujian Dan Analisis	39
4.1 Lingkungan pengujian	39
4.2 Hasil Implementasi	40
4.3 Pengujian Fitur Dalam Satu Kelompok	45
4.4 Pengujian fitur Antar Kelompok	48
4.5 Pengujian Jumlah Skala Warna	49
4.6 Pengujian Teknik Matching	53
4.7 Pengujian Waktu Proses	55
Bab 5 Simpulan dan Saran	59
5.1 Simpulan	59
5.2 Saran	59
Daftar Pustaka	61
Lampiran	63

DAFTAR GAMBAR

DAFTAR GAMBAR

Gambar	Halaman
Gambar 2.1 contoh query image di Yahoo Search Engine dengan key=pig	12
Gambar 2.2 Ruang warna RCB model kubik	14
Gambar 2.3 Konversi kuantisasi komponen warna dari 256 menjadi n, dengan $n < 255$	16
Gambar 3.1 Blok Diagram Sistem	24
Gambar 3.2 Model Kubik Format Warna RGB	25
Gambar 3.3 Format Warna RGB dalam Komposisi bit	26
Gambar 3.4 Konversi kuantisasi komponen warna dari 256 menjadi 25	27
Gambar 3.5 Model kubik dan model indeks	27
Gambar 3.6 Contoh histogram indeks pada beberapa gambar buah	31
Gambar 3.7 Histogram sebelum dinormalisasi dan sesudah dinormalisasi	32
Gambar 3.8 Proses pengambilan fitur	33
Gambar 3.9 Histogram indeks skala 4 pada gambar-gambar jeruk sejenis	35
Gambar 3.10 Histogram rata-rata indeks 4 untuk buah jeruk	36
Gambar 3.11 Histogram gabungan dari dua gambar buah jeruk yang dipilih	36
Gambar 3.12 Proses penggabungan fitur dengan histogram rata-rata	36
Gambar 3.13 Blok diagram untuk proses identifikasi buah	38
Gambar 4.1 Perbandingan histogram indeks rata-rata jeruk dan apel hijau	44
Gambar 4.2 Histogram indeks buah apel hijau dan jeruk	45
Gambar 4.3 Histogram indeks untuk buah jeruk dengan $n=4$	46
Gambar 4.4 Histogram indeks beberapa buah pisang sejenis	47
Gambar 4.5 Histogram rata-rata kelompok buah pisang, $n=4$	48

Gambar 4.6 Histogram indeks 3 kelompok buah berbeda	49
Gambar 4.7 Histogram indeks dari buah apel merah pada beberapa skala warna	51
Gambar 4.8 Grafik akurasi pengenalan gambar buah	52
Gambar 4.9 Contoh perhitungan kemiripan buah untuk $n=4$	54
Gambar 4.10 Contoh hasil perhitungan kemiripan gambar buah untuk $n=5$	54
Gambar 4.11 Hasil identifikasi apel hijau dengan skala warna, $n=4$	55
Gambar 4.12 Hasil identifikasi apel hijau dengan skala warna, $n=5$	55
Gambar 4.13 Hasil identifikasi apel hijau dengan skala warna, $n=6$	55
Gambar 4.14 Pengaruh jumlah skala warna dan waktu proses pengenalan	57

DAFTAR TABEL

AL-AMUDY
XEROX

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Perkembangan *computer vision* dan teknologi pengolahan citra saat ini memungkinkan untuk melakukan sistem *retrieval* suatu objek melalui gambar objek tersebut. Sistem *retrieval* melalui gambar ini dapat menghasilkan informasi utuh mengenai suatu objek melalui gambar yang dimasukkan. Sehingga pemakai sistem ini dengan mudah memperoleh informasi mengenai sebuah objek yang dibutuhkan dengan hanya memasukkan gambar objek tersebut pada sistem. Hal ini akan sangat berguna untuk membantu banyak hal dalam sisi kehidupan manusia yang berhubungan dengan kemudahan informasi.

Sistem *retrieval* pada umumnya menggunakan objek gambar yang umum seperti gambar pemandangan, mobil, gedung, awan, dan foto satelit seperti yang sudah dikembangkan oleh Nana [1] yang menjelaskan sistem *retrieval* menggunakan objek gambar umum dengan mengacu pada pola warna, bentuk dan tekstur dengan penekanan pada tekstur menggunakan filter gabor. Tesis ini menjelaskan bagaimana mendapatkan ciri warna menggunakan histogram, ciri bentuk menggunakan deteksi tepi dan ciri tekstur menggunakan filter gabor, serta bagaimana cara menggabungkan ketiga ciri tersebut menjadi suatu vektor ciri (*feature vector*). Meskipun pada penelitian ini titik beratnya pada filter gabor untuk ciri tekstur dari suatu gambar.

Sistem *retrieval* pada gambar yang bersifat umum ini memerlukan teknik ekstraksi fitur dan sistem pengenalan yang cukup rumit dengan fitur-fitur seperti warna, bentuk, tekstur, cluster objek dengan berbagai teknik pengenalan seperti jaringan syaraf tiruan seperti yang telah dijelaskan dalam penelitian Prasad [2] mengenai *Color and Shape Index* berdasarkan *region* pada gambar menggunakan teknik *indexing* pada objek berdasarkan warna dan bentuk. Atau pada penelitian Soriano [3] yang hanya menggunakan fitur warna, tetapi mengembangkan teknik pengenalan menggunakan jaringan syaraf tiruan karena data fitur yang mempunyai derajat biasa yang cukup tinggi.

Berdasarkan kebutuhan akan sistem retrieval, Indonesia merupakan sebuah negara yang mempunyai ragam buah-buahan yang sangat banyak dan sangat berguna bila informasi mengenai keragaman buah di Indonesia ini akan berguna bila hal tersebut dapat diketahui dengan mudah. Jangankan orang luar negeri, orang Indonesia aslipun terkadang banyak yang belum tahu informasi mengenai suatu buah unik yang hanya ada di suatu daerah. Keberadaan sistem retrieval berbasis gambar ini diharapkan mampu memberikan kemudahan informasi akan buah-buahan hanya dengan memasukkan gambar buah-buahan hanya dengan mengambil gambar buah kemudian dimasukkan ke dalam sistem.

Sistem retrieval ini hanya menggunakan objek buah-buahan dalam proses pengenalannya. Sehingga fitur warna dianggap cukup untuk digunakan dalam sistem ini, seperti dijelaskan oleh Soriano [3]. Tetapi batasan objek ini membuat sistem pengenalan menjadi lebih sederhana sehingga tidak memerlukan sistem pengenalan yang rumit seperti jaringan syaraf tiruan, bila fitur yang dibangkitkan dapat diperoleh dengan derajat bias yang sangat kecil. Sehingga titik berat penelitian ini adalah bagaimana mengembangkan teknik ekstraksi fitur berdasarkan warna untuk kebutuhan sistem retrieval.

Fitur warna yang banyak digunakan dalam sistem retrieval adalah distribusi warna yang dinyatakan sebagai histogram. Dalam penelitian sistem pengenalan buah yang dijelaskan oleh Dwi [4], teknik histogram yang digunakan adalah teknik histogram tiga dimensi $H(r,g,b)$ sehingga fitur yang diperoleh membutuhkan ruang vektor berderajat tinggi. Teknik histogram lain yang baik untuk sistem retrieval adalah teknik histogram indeks seperti yang dijelaskan oleh Ravishankar [5], dalam penelitian ini indeks warna yang digunakan adalah 25 skala warna untuk setiap elemen warna. Bila format warna yang digunakan adalah RGB, maka ada 3 buah elemen warna yaitu R, G dan B, sehingga histogram indeks ini menghasilkan 25^3 indeks warna yang merupakan jumlah data fitur yang sangat besar, hal ini karena penelitian ini digunakan untuk objek gambar umum.

Untuk pengenalan buah ini, fitur yang digunakan adalah histogram indeks dengan menurunkan derajat warna pada setiap elemen sekecil-kecilnya sehingga diperoleh fitur optimal dengan ukuran yang kecil. Dengan demikian penggunaan waktu dapat ditekan sehingga bisa dikatakan tak ada waktu tunggu

dalam memperoleh informasi mengenai buah-buahan yang ada di Indonesia. Harapan lebih lanjut adalah menarik para wisatawan untuk mengetahui lebih lanjut mengenai Indonesia melalui informasi buah-buahan yang sangat beragam dan menarik sebagai media informasi dan promosi mengenai kekayaan alam Indonesia. Hal ini adalah tujuan lebih lanjut setelah penelitian ini mendapatkan hasil yang baik dalam mengenali jenis buah.

1.2 Rumusan Permasalahan

Sistem identifikasi buah ini merupakan suatu proses yang hasilnya adalah informasi mengenai buah berdasarkan gambar buah yang dimasukkan ke dalam sistem. Untuk bisa melakukan hal tersebut ada beberapa permasalahan yang harus diselesaikan, yang dirumuskan sebagai berikut :

1. Pengambilan fitur dari gambar buah, dalam penelitian ini menggunakan histogram indeks dengan derajat skala warna yang diturunkan sehingga mengurangi ukuran fitur. Mengacu pada penelitian Ravishankar [5] yang menggunakan skala warna 25 diharapkan pada penelitian ini dapat digunakan skala warna yang lebih kecil, dengan asumsi bahwa keseragaman tema (buah) yang lebih sempit dari sistem *image retrieval* yang biasa untuk semua macam gambar akan dapat mengurangi skala warna dengan jumlah kurang dari 16. Pada penelitian ini juga akan diperhatikan beberapa proses normalisasi histogram yang mana yang bisa menunjukkan fitur yang lebih baik untuk menyatakan sebuah objek buah.
2. Pembuatan Basis data fitur, dalam proses ini dilakukan penyimpanan fitur setiap buah yang disimpan sebagai data acuan (*image basis data*). Dari setiap fitur ini akan disajikan fitur setiap objek buah menggunakan rata-rata fitur dari buah yang sama yang merupakan penurunan sifat konveksitas vektor fitur yang merupakan salah satu cara untuk mendapatkan homogenitas dari vektor fitur pada objek buah yang sama.
3. Proses Identifikasi buah, untuk proses identifikasi menggunakan *template matching* berdasarkan fitur objek buah yang sudah dikelompokkan sehingga dapat mengurangi penggunaan waktu. *Template matching* ini adalah suatu

ukuran berdasarkan jarak antara vektor ciri acuan dan vektor ciri dari gambar buah yang dimasukkan. Penelitian ini akan menggunakan teknik jarak euclidian dalam menghitung nilai kemiripan dari objek gambar buah.

1.3 Tujuan dan Manfaat Penelitian

Tujuan penelitian ini adalah menghasilkan sebuah sistem identifikasi berbasis pengolahan citra digital yang digunakan untuk mengidentifikasi buah. Untuk bisa mengidentifikasi buah digunakan histogram warna sebagai distribusi kemunculan setiap warna dengan model setiap warna dijadikan sebagai nomor indeks warna. Tujuan lebih khusus adalah membuat jumlah indeks menjadi minimal dengan akurasi yang tinggi, sehingga waktu komputasinya menjadi lebih cepat.

Manfaat yang diperoleh dari tesis ini adalah dapat digunakan sebagai pengganti mesin *bar code* untuk mengidentifikasi buah-buahan.

1.4 Batasan Penelitian

Batasan permasalahan untuk penelitian sistem identifikasi citra buah dalam tesis ini antara lain:

1. Latar belakang yang digunakan adalah uniform, khususnya berwarna putih
2. Jarak buah dan kamera, dalam jarak jangkauan kamera sehingga gambar buah terlihat dengan jelas.
3. Gambar buah dapat diambil gambarnya secara utuh
4. Satu gambar hanya berisi satu jenis buah, tetapi jumlah buah bisa lebih dari satu.
5. Format warna yang digunakan dalam proses identifikasi adalah RGB, karena format warna ini adalah format warna dasar di dalam image capture, sehingga tidak membutuhkan transformasi warna yang akan menambah proses komputasi.

BAB 2

TINJAUAN PUSTAKA

2.1 *Content Based Image Retrieval (CBIR)*

Content-based Image Retrieval (CBIR) telah menjadi sebuah topik penelitian yang sering dilakukan banyak orang karena permintaan untuk image retrieval yang efisien dalam basis data image yang besar. Cara tradisional untuk mendapatkan kembali image adalah dengan cara menyatakan kata kunci (keywords) secara manual (*text-based*). Ada dua kerugian utama. Pertama, secara intensif tidak wajar karena menghabiskan waktu dan mahal. Kedua, semantik yang pantas/sesuai dari sebuah image sulit digambarkan secara tepat dan orang yang berbeda dapat menggambarkan image yang sama dengan cara yang berbeda [6]. Untuk mengatasi kekurangan dari pendekatan *text-based*, pendekatan *content-based image retrieval (CBIR)* mencoba untuk menghasilkan kembali *image-image* secara langsung dan secara otomatis didasarkan pada isi visual mereka seperti warna, tekstur, dan bentuk yang dipunyai [6]. Dalam sebuah sistem *content-based image retrieval* yang tertentu, pola querinya adalah query berdasarkan contoh (*query by example*), dengan mencari N image pertama yang sama untuk sebuah contoh *image*. Sebelum menghasilkan *image retrieval*, fitur-fitur visual diekstraks dari semua *image* dalam sebuah basis data image secara *offline*. Selama menghasilkan *image retrieval*, fitur-fitur visual dari contoh *image* dibandingkan terhadap semua *image* dalam basis data *image* dan N *image* pertama dihasilkan kembali sebagai hasil query [6].

Ada banyak sistem *content-based image retrieval* yang telah dibangun sebagai referensi/pustaka. Dalam sistem QBIC [3], informasi warna, tekstur atau bentuk digunakan dalam representasi image. Informasi warna dikodekan dengan menggunakan dua vektor warna. Pertama adalah vektor warna rata-rata 3D dari sebuah citra dalam ruang warna RGB, YIQ, Lab, and Munsell. Yang kedua adalah sebuah histogram warna bin 256 dalam ruang warna RGB. Vektor warna rata-rata

berfungsi sebagai sebuah filter untuk membatasi komputasi yang mahal yang dibutuhkan dalam komputasi histogram warna. Selama menghasikan *image retrieval*, kueri-kueri yang didasarkan pada fitur-fitur individu diproses pertama kali secara bebas. Kemudian hasil secara individu tadi dikombinasi dengan menggunakan suatu jumlahan jarak yang terbobot [6]. Sistem dalam [6] menyaring *image-image* yang tidak diharapkan melewati sebuah icon filter, sebuah detektor *graph-photo*, sebuah filter histogram warna, sebuah filter tekstur, dan sebuah filter bentuk [6].

Basis data citra diadakan untuk menyimpan sekumpulan gambar seni, citra satelit, citra hasil kedokteran dan secara umum sekumpulan foto. Penggunaannya bermacam-macam bergantung pada aplikasinya. Pemakai dari kelompok seni mungkin ingin menemukan hasil karya seniman tertentu atau menemukan siapa yang melukis gambar tertentu yang telah mereka lihat. Pemakai pada basis data kedokteran mungkin digunakan oleh pelajar pada sekolah kedokteran untuk belajar anatomi atau dokter untuk mencari contoh sample dari penyakit yang ditemukan. Kumpulan basis data citra secara umum yang berisi foto mungkin diakses oleh ilustrator untuk mencari gambar yang sesuai untuk artikel atau bukunya. Bidang aplikasi ini sangat luas cakupannya, seorang pemakai mungkin ingin menemukan citra kuda, yang lainnya mungkin ingin menemukan citra dari matahari terbenam, dan orang ketiga mungkin mencari sesuatu konsep yang abstrak, misalnya cinta. Penelitian ini memilih citra buah-buahan sebagai batasan objek gambar.

Basis data citra dapat menjadi sangat besar, berisi ratusan ribu atau jutaan citra. Pada kebanyakan kasus basis data citra ini hanya diindeks berdasarkan kata kuncinya yang harus sudah diputuskan pada masing-masing citra dan dimasukkan pada sistem basis data oleh manusia sebagai pengelompoknya. Tetapi, citra dapat di-*retrieve* berdasarkan isinya masing-masing, dimana isi mungkin mengacu pada distribusi warna, tekstur, bentuk region atau klasifikasi objek. Sampai saat ini sistem penelitian dan di bidang komersial yang telah dibangun dan yang sekarang sedang digunakan masih primitif. Demo sistem CBIR ini seringkali disediakan di world wide

web. Pada sub bab berikut ini akan dibahas metode yang mana orang dapat *retrieve* citra tanpa melakukan resorting seperti pencarian berbasis kata kunci [6].

2.2 *Query by Image Content (QBIC)*

Beberapa citra basis data telah dibuat hanya untuk menunjukkan bagaimana secara khusus sistem retrieval bekerja. IBM dengan *Query by Image Content (QBIC)* basis data proyek adalah contoh dari sistem CBIR ini. QBIC adalah sebuah sistem penelitian yang menghasilkan produk komersial yang dikembangkan dan dijual oleh IBM. Sistem QBIC ini *retrieve* citra berbasis pada *visual content*, termasuk propertinya seperti persentase warna, layout warna dan tekstur. Virage, Inc. mengembangkan sebuah produk yang menggunakan QBIC, Virage Search Engine yang dapat *retrieve* citra berdasarkan warna, komposisi, tekstur dan struktur. Mesin pencari citra yang dibuat oleh institusi yang lainnya dapat juga digunakan sebagai pencarian basis data. Contoh museum Fine Arts di San Fransisco yang memberikan akses ke QBIC melalui sekumpulan lukisan digitalisasinya.

Di samping pada kelompok seni, disini ada sekumpulan citra umum dimana masing-masing citra disediakan lisensi oleh pelanggan khusus, yaitu siapa yang ingin citra pilihannya menjadi produk pasarnya atau untuk ilustrasi artikel. Salah satu yang terbesar adalah Corbis Archive yang berisi lebih dari 17 juta citra. Corbis Archive mencoba menangkap semua range ekspresi dan persepsi manusia yang berisi kategori-kategori seperti sejarah, seni, hiburan, ilmiah, industri, dan binatang. *Retrieval* citra pada Corbis menggunakan kata kunci dan dengan *browsing*. Perusahaan lainnya, Getty Images, sekarang memberikan beberapa basis data citra secara *on/line* yang disusun berdasarkan kategorinya dan dapat dicari melalui kata kuncinya.

Di samping pada bidang pekerjaan seni dan fotografi, terdapat juga kelompok citra ilmiah dan medical. National Library of Medicine memberikan basis data dari *X-rays*, CT scans, citra MRI dan warna *cross-section* yang diambil dari sayatan kecil sepanjang tubuh mayat laki-laki dan perempuan. Disini disediakan lebih

dari 14.000 citra yang disediakan bagi mereka yang ingin menggunakannya untuk penelitian dibidang kesehatan. National Aeronautics dan Space Administration (NASA) mengumpulkan basis data citra yang besar dari satelit. United States Geological Survey (USGS) menyediakan kemampuan *web search* bagi pemakai yang ingin menemukan dan meminta sekumpulan data termasuk foto satelit dan citra udara. Pada akhirnya, world wide web yang ada saat ini adalah basis data yang berisi teks dan sejumlah besar citra. Mesin pencari untuk menemukan citra pada Web, berbasis pada kata kunci dan terbatas dalam me-*retrieve* citra dengan benar, sedangkan *search engine* yang berbasis pada *content* citra sedang dikembangkan [6].

2.2.1 Query Basis Data Citra

Jika diberikan sebuah basis data citra, bagaimana cara mencari citra tertentu dari keseluruhan basis data citra, untuk kemudian me-*retrieve* citra yang sama. Perusahaan yang membuat basis data citra yang disediakan untuk pemakainya umumnya mempunyai proses pemilihan untuk menyatakan citra yang mana yang seharusnya ditambahkan pada kumpulan dan proses untuk mengelompokkan untuk memberikan kategori yang umum dan kata kunci lain untuk citra terpilih. Citra yang tampil pada world wide web biasanya mempunyai label dari kata kunci yang mana yang dapat dihasilkan otomatis.

Pada sistem basis data *relational*, *entity* dapat di-*retrieve* berbasis nilai atribut tekstualnya. Atribut yang digunakan untuk *retrieve* citra mungkin memasukkan kategori umum, nama objek, nama orang, tanggal dibuat, dan sumbernya. Citra dapat diberi indeks menurut atribut-atribut ini, sedemikian sehingga citra dapat dengan cepat di-*retrieve* ketika *query* dimasukkan. Tipe *query* tekstual ini dapat diekspresikan dengan bahasa SQL basis data relasional yang disediakan oleh semua standard sistem relasional. Adapun contoh *query* adalah:

```
SELECT * FROM IMAGEDB  
WHERE CATEGORY = 'MUTIARA' AND SOURCE = 'SMITHSONIAN'
```


Query ini akan menemukan dan mengembalikan semua citra dari basis data citra bernama IMAGEDB yang atribut CATEGORY ada pada himpunan 'MUTIARA' dan atribut SOURCE diambil dari data 'SMITHSONIAN'. Objek akan *me-retrieve* citra dari kelompok mutiara dari institute Smithsonian. Hasil citra *retrieve* dengan kategori mutiara dapat berupa citra dari batu permata. Untuk mengijinkan *retrieval* lebih selektif, digunakan kumpulan kata kunci yang deskriptif yang akan disimpan pada masing-masing citra. Pada basis data relasional kata kunci akan menjadi atribut yang dapat mempunyai banyak nilai untuk masing-masing citra. Jadi citra batu permata dapat mempunyai nilai batu permata, kristal dan berwarna ungu dengan kata kunci ini, dapat *diretrieve* berdasarkan semua dari tiga atribut atau salah satu dari tiga atribut, bergantung pada keinginan pemakai. Contoh query SQLnya:

```
SELECT * FROM IMAGEDB
WHERE CATEGORY = 'MUTIARA' AND SOURCE = 'SMITHSONIAN'
AND (KEYWORD = 'BATU PERMATA' OR KEYWORD = 'KRISTAL'
OR KEYWORD = 'UNGU')
```

Akan *me-retrieve* semua citra dari kumpulan basis data citra IMAGEDB, yang mempunyai CATEGORY = 'MUTIARA', dari SOURCE = 'SMITHSONIAN' dan mempunyai nilai KEYWORD = ['BATU PERMATA', 'KRISTAL', 'UNGU']. Dengan model sintaks SQL ini akan *me-retrieve* citra lebih banyak dibandingkan sintaks SQL sebelumnya yang hanya menampilkan citra batu permata. Selanjutnya pemakai dapat *mem-browse* dan memilih citra dari sekumpulan citra yang dihasilkan.

Terdapat keterbatasan yang dapat dikerjakan dengan pendekatan kata kunci. Kata kunci yang dikodekan orang adalah mahal dan terbatas pada beberapa istilah yang dihasilkan untuk masing-masing referensi citra. Lebih lanjut beberapa citra yang dihasilkan akan tampak sangat berbeda dibandingkan dengan keinginan user dari otomatisasi pencarian menggunakan kata kunci, contoh perhatikan gambar 2.1 dengan memasukkan kata kunci babi pada Mesin Pencari Yahoo akan dihasilkan citra yang ada gambar babinya, dan citra yang tidak ada gambar babinya.





a) Gambar ada Babi



b) Gambar tidak ada babi

Gambar 2.1 Contoh *Query Image* di Mesin Pencari Yahoo dengan Key = pig

Dengan memberikan kata kunci saja adalah tidak cukup, harus dikembangkan dengan metode lain untuk *retrieve* citra yang dapat digunakan sebagai pengganti atau ditambahkan pada kata kunci [6].

2.2.2 *Query* Berdasarkan Contoh

Pada basis data citra, ide *query* berdasarkan contoh (*query-by-example*) adalah menarik. Sebagai pengganti mengetikkan sebuah *query*, pemakai basis data citra seharusnya dapat menunjukkan sistem contoh citra, atau menggambarkannya secara interaktif pada *screen*, atau hanya melakukan sketsa bentuk dasar dari objek. Sistem seharusnya dapat mengembalikan citra yang mirip atau citra yang berisi objek yang mirip. Dalam hal ini tujuannya adalah sistem *content based image retrieval*, dimana masing-masing mempunyai cara sendiri dalam menyatakan *query*, menyatakan kemiripan diantara *query* dan citra pada basis data, dan memilih citra yang akan dikembalikan.

Sebuah *query* adalah contoh citra ditambah sekumpulan batasan-batasan. Citra dapat berupa hasil fotografi, digambar sendiri oleh user, sketsa garis, atau kosong, pada kasus seperti ini himpunan *retrieval* harus hanya cocok dengan batasan-batasan yang diberikan. Batasan-batasan yang diberikan dapat berupa kata kunci yang seharusnya ada pada beberapa sistem indeks atau mungkin menyatakan objek yang seharusnya ada pada sebuah citra dan hubungan diantara objek-objek tersebut. Pada beberapa kasus yang umum *query* adalah citra digital yang dibandingkan dengan citra

pada basis data menurut pengukuran jarak diantara citra *query* dan citra pada basis data (*image distance measure*). Ketika jarak yang dihasilkan nol, citra persis sama dengan citra *query* (*match*) atau dikatakan bahwa nilai kemiripannya adalah 100. Nilai yang lebih besar dari nol menunjukkan bermacam-macam derajat kemiripan *query* (*similarity*). Search Engine berbasis citra biasanya menghasilkan sekumpulan citra berdasarkan jarak citra dengan *query*.

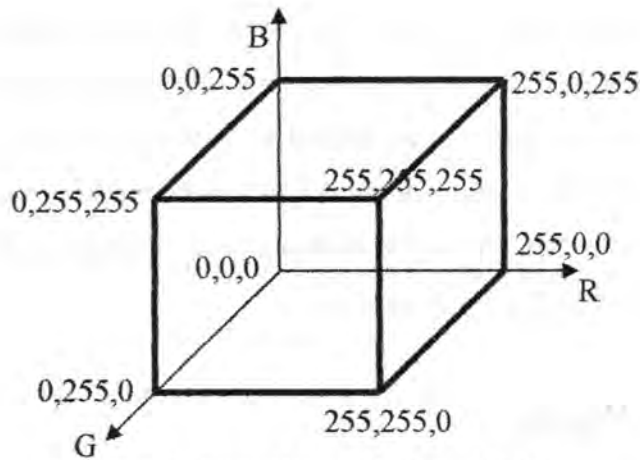
2.3 Konsep Warna

Suatu citra I yang akan diolah (*image processing*) pada dasarnya berupa suatu matriks dua dimensi dengan ukuran $n_x \times n_y$ dengan nilai setiap piksel $I(x, y)$ menyatakan nilai warna dari piksel itu sendiri. Di sisi lain, warna bukanlah suatu besaran numerik, sehingga dalam merepresentasikan warna tersebut ke dalam besaran numerik, perlu adanya transformasi warna, yaitu dengan menyatakan nilai derajat kecerahan dari setiap komponen warna.

Model warna RGB (red, green, blue) mendeskripsikan warna sebagai kombinasi positif dari 3 warna, yaitu merah, hijau, dan biru sehingga membentuk sebuah warna W dengan rumusan sebagai berikut:

$$W = (rR, gG, bB) \tag{2.1}$$

Dimana r, g, b berturut-turut adalah skalar atau konstanta dari $R, G,$ dan B . Jika skalar r, g, b diberi harga antara 0 dan 255, maka semua definisi warna akan memiliki nilai yang berbeda-beda. Sebagai contoh tentang kombinasi nilai-nilai tersebut berada dalam kubus seperti yang diperlihatkan pada gambar 2.2. Warna Red dinyatakan dengan kombinasi $(255,0,0)$, Green dengan $(0,255,0)$ serta Blue $(0,0,255)$. Sementara kombinasi $(0,0,0)$ merepresentasikan warna hitam dan $(255,255,255)$ adalah warna putih, sedang garis yang menghubungkan keduanya menyatakan warna antara putih dan hitam, yaitu abu-abu.



Gambar 2.2 Ruang Warna RGB model Kubik

Ruang warna sebagaimana yang diperlihatkan pada gambar 2.2. adalah dasar warna dari display monitor komputer. Garis sepanjang titik hitam (0,0,0) RGB hingga titik abu (255,255,255) RGB disebut dengan titik keabuan atau gray level. Sehingga kita dapatkan hubungan antara RGB dengan gray level yaitu:

$$(a)_{GL} \Leftrightarrow (a, a, a)_{RGB} \quad (2.2)$$

Pada proses penjumlahan warna, warna yang dideskripsikan dengan RGB adalah pemetaan yang mengacu pada panjang gelombang dari RGB. Pemetaan tersebut menghasilkan nuansa warna sekitar 16 juta ($256^3 = 16.777.216$) yaitu 256 (8 bit) untuk masing-masing R, G, dan B. Masing-masing R, G dan B didiskritkan dalam skala 256, sehingga RGB akan memiliki indeks antara 0 sampai 255. Penjumlahan warna RGB ini adalah:

$$C = C_1 + C_2 = \frac{(rC_1 + gC_1 + bC_1) + (rC_2 + gC_2 + bC_2)}{2} \quad (2.3)$$

Penjumlahan ini menghasilkan nilai RGB yang tidak bulat, sehingga kita harus membulatkan nilainya, yaitu dengan metode bulat kebawah atau keatas. Misalkan dua warna yaitu hitam (0, 0, 0) kita jumlahkan dengan warna putih (255, 255, 255) akan menghasilkan warna abu-abu antara hitam dan putih (127, 127, 127). Hasil akhir dibulatkan kebawah, contoh lain misalkan menjumlahkan warna merah

(255, 0, 0) dengan warna hijau (0, 255, 0) akan menghasilkan warna kuning (127,127,0).

2.4 Histogram Warna

Histogram warna merupakan hubungan dari intensitas tiga macam warna. Histogram warna tersebut didefinisikan sebagai berikut:

$$H_{R,G,B}[r,g,b] = N.Prob\{R=r, G=g, B=b\} \quad (2.4)$$

dimana R, G, B merupakan tiga macam masing-masing warna merah, hijau, dan biru, N adalah jumlah pixel pada citra. Histogram warna dihitung dengan cara mengkonversi nilai warna dalam bentuk diskret, dan menghitung jumlah dari tiap-tiap pixel pada citra untuk masing-masing nilai warna yang sama. Karena jumlah dari tiap-tiap warna terbatas, maka untuk lebih tepatnya dengan cara menstransform tiga histogram ke dalam histogram yang mempunyai variabel tunggal. Misalkan pada citra RGB, salah satu transformasinya didefinisikan dengan:

$$m = rN_r + gN_g + bN_b \quad (2.5)$$

dimana N_r , N_g , dan N_b merupakan jumlah nilai biner dari warna merah, hijau, dan biru secara berturut-turut. Di bawah ini merupakan persamaan dari histogram yang mempunyai variabel tunggal:

$$h[m] = N.Prob\{M=m\} \quad (2.6)$$

dimana M adalah jumlahan dari nilai RGB pada masing-masing piksel.

2.5 Indeks Warna dan Histogram Indeks

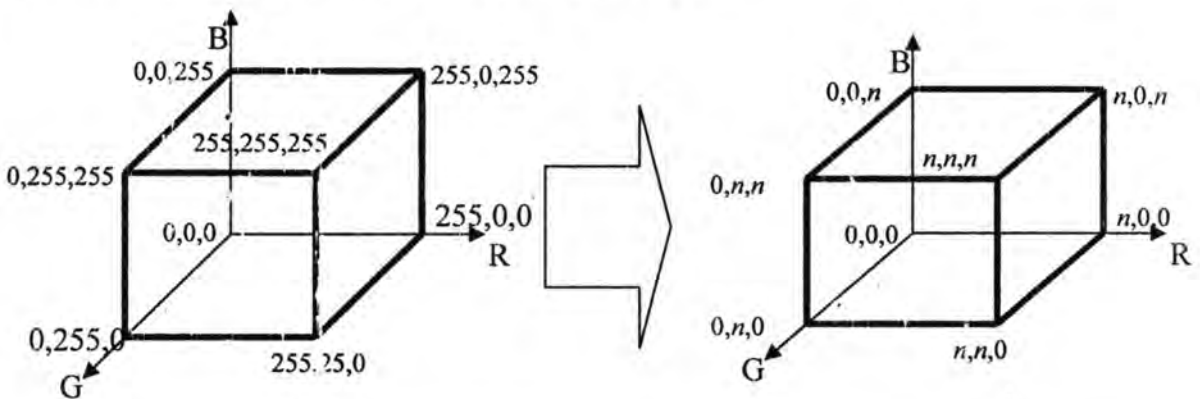
Pada bagian ini akan diperdalam tentang warna beserta histogramnya. Sebagaimana fitur yang digunakan dalam penelitian ini adalah fitur warna di sisi lain warna itu sendiri direpresentasikan dengan derajat keabuan yang merupakan gabungan antara 3 komponen warna R, G, dan B, maka pernyataan nilai warna dapat ditulis dalam bentuk indeks warna beserta histogramnya (histogram indeks warna).

2.5.1 Indeks Warna

Suatu warna dengan format RGB dinyatakan sebagai komposisi dari nilai komponen dasar warna Red (R), Green (G) dan Blue (B). Sehingga suatu fungsi F terhadap komponen warna R, G dan B, dan dituliskan dengan $F(r,g,b)$, bentuk ini adalah bentuk array 3 dimensi. Bila setiap komponen warna mempunyai kuantisasi 256 derajat keabuan maka jumlah warna yang mungkin adalah $256^3=16.777.216$ warna yang disusun dari H (0,0,0) sampai dengan H (255,255,255). Pernyataan penulisan atau format warna seperti ini adalah suatu jumlah yang sangat besar.

Dari jumlah warna yang sangat besar tersebut, diadakan pengurangan jumlah warna dengan tidak mengurangi *content* yang ada pada sebuah citra. Cara yang digunakan dalam pengurangan jumlah warna ini adalah dengan penurunan kuantisasi

pada setiap komponen warna menjadi n komponen dengan $n < 255$ sehingga diperoleh konversi koordinat warna seperti gambar 2.3.



Gambar 2.3 Konversi kuantisasi komponen warna dari 256 menjadi n , dengan $n < 255$

Dari cara penulisan warna sebagaimana tersebut di atas, dapat dinyatakan bahwa penulisan atau pernyataan warna dalam bentuk fungsi 3 dimensi (3D). Bila dalam fungsi 3 dimensi seperti ini maka representasi warna yang ada tidak dapat dijelaskan secara baik. Hal ini disebabkan visualisasi dari model 3D yang ditulis

dengan $F(x,y,z)$ tidak dapat dinyatakan dengan mudah ($F(x,y,z)$ adalah nilai histogram dari nilai warna (x, y, z)). Dari kondisi inilah dilakukan suatu konversi dari model 3D menjadi model fungsi 1D untuk pernyataan warna tersebut. Jadi fungsi 3D: $F(x,y,z)$ dikonversi menjadi fungsi 1D: $F(w)$, dimana w adalah kombinasi warna, sedangkan $F(w)$ adalah nilai histogramnya. Bila $F(x,y,z)$ dimana x, y, z , masing-masing adalah mewakili nilai read R, green G, dan blue B, maka fungsi tersebut dapat ditulis menjadi $F(r,g,b)$. Untuk selanjutnya proses konversi fungsi 3D: $F(r,g,b)$ ke dalam bentuk fungsi 1D: $F(w)$ inilah yang dinamakan dengan nilai histogram indeks warna.

Dari hasil penelitian sebelumnya dikatakan bahwa distribusi warna menjadi lebih mudah diamati bila model warna yang digunakan adalah model indeks [2]. Suatu warna indeks merupakan gabungan dari nilai kuantisasi setiap komponen warna R, G dan B, dan dibentuk dengan didasarkan pada sebuah tabel warna yang terdiri dari r indeks warna (kuantisasi warna). Indeks adalah unik dan diberikan dengan persamaan berikut:

$$\text{Indeks} = \sum_{i=1}^n C_i * r^{n-i} \quad (2.7)$$

dimana:

n = jumlah daerah dominan,

C_i = indeks warna untuk daerah dominan ke- i , dan

r = kuantisasi derajat keabuan setiap komponen warna.

Dengan jumlah derajat keabuan setiap komponen warna sebesar n , yang selanjutnya disebut dengan skala warna, maka akan menghasilkan jumlah indeks warna sebesar n^3 . Sehingga bila skala warna adalah 256 sebagaimana pada true-color, maka akan menghasilkan jumlah indeks warna sebesar $256^3=16.777.216$ indeks warna. Dalam penelitian ini dilakukan pengurangan indeks warna dengan tidak menggunakan true-color lagi, sehingga sangat mereduksi jumlah indeks warna yang mana kondisi ini akan membawa manfaat pada efektifitas proses sistem yang ada.

2.5.2 Histogram Indeks

Sebagaimana arti dari suatu histogram dalam statistika, maka histogram $H(x)$ adalah suatu cara untuk mendapatkan jumlah kemunculan suatu kejadian X , atau dengan kata lain menggambarkan pola peluang terjadinya suatu kejadian. Maka dengan pengertian yang sama, histogram pada warna dalam suatu citra, menyatakan peluang munculnya suatu warna di dalam suatu citra.

Dalam permasalahan model warna indeks, variabel X identik dengan nomor indeks warna W . Jadi dapat dimengerti bahwa histogram indeks adalah suatu bentuk histogram yang menggambarkan peluang terjadinya suatu indeks warna di dalam sebuah citra. Persamaan histogram indeks warna tersebut dapat dinyatakan dengan :

$$H(w) = \Pr(w | I(x,y) \in I, w \in W) \quad (2.8)$$

dimana:

- W = Himpunan seluruh indeks yang akan digunakan dengan jumlah indeks warna sebesar w^3
- dengan w adalah skala warna pada masing-masing komponen warna.

Untuk dapat membandingkan antara citra dengan skala yang berbeda-beda maka terhadap histogram indeks perlu dilakukan normalisasi, sehingga hasil histogramnya menjadi fungsi probabilitas kemunculan setiap indeks warna. Persamaan fungsi probabilitas kemunculan setiap indeks warna tersebut adalah

$$P(x) = \frac{H(x)}{\sum_{i=0}^N H(i)} \quad (2.9)$$

Dimana:

$P(x)$ = probabilitas kemunculan indeks warna x ,

$H(x)$ = histogram indeks warna x ,

$H(i)$ = histogram indeks warna ke- i .

Sebagaimana pada nilai fungsi probabilitas yang terletak antara range 0 sampai dengan 1, demikian pula hasil normalisasi dari histogram indeks warna ini

juga terletak antara range 0 sampai dengan 1, dan jumlah dari seluruh kemunculan dalam histiogram sama dengan 1.

2.6 Basis data Fitur

Dalam proses pengolahan citra untuk menyimpan suatu hasil fitur dinyatakan dalam basis data fitur. Bila tiga fitur dasar dalam pengolahan citra yaitu fitur bentuk, warna dan tekstur digunakan semua dalam proses identifikasi, maka ketiga-tiganya disimpan dalam basis data fitur. Berarti ada tiga tabel basis data fitur dalam penyimpanan tersebut. Fitur yang dihasilkan dalam pengolahan citra dapat dipandang sebagai vektor fitur. Sedangkan representasi vektor fitur dapat menggunakan histogram. Vektor fitur yang berupa histogram tadi disimpan ini disimpan dalam sebuah tabel basis data.

Untuk n jenis gambar dengan objek yang sama maka dalam tabel akan mempunyai n *record* data. Bila terdapat m jenis objek gambar yang berbeda maka di dalam tabel basis data terdapat $m \times n$ *record*. Sedangkan bila digunakan skala warna *true-color* 256 setiap komponen warna maka $256^3 = 16.777.216$ *attribut*. Kondisi ini akan menyebabkan proses sistem (*retrieval*) menjadi sangat lambat. Salah satu cara untuk mengatasi kelambatan ini, digunakan sistem pengelompokan setiap n objek gambar yang sama. Pengelompokan ini dilakukan pada semua jenis objek gambar yang berbeda (m objek gambar yang berbeda). Dan bersamaan dengan itu dilakukan pengurangan jumlah indeks warna, tidak lagi sebesar 256^3 indeks warna.

Dengan adanya pengelompokan, maka yang disimpan dalam tabel basis data adalah fitur tiap-tiap kelompok saja (ada m kelompok, berarti ada m *record*). Cara membuat fitur dalam suatu kelompok dapat menggunakan nilai rata-rata pada setiap indeks warna dalam satu kelompok tersebut. Persamaan nilai rata-rata untuk setiap indeks warna adalah:

$$H_i = \frac{\sum_{k \in K} h_i^k}{N} \quad (2.10)$$

dimana:

- H_i = histogram rata-rata pada indeks warna ke-i dalam satu kelompok (kelompok ke-k).
- h_i = nilai histogram pada indeks warna ke-i dalam satu kelompok (kelompok ke-k)
- N = jumlah data pada setiap objek gambar.

Dari hasil histogram rata-rata pada masing-masing jenis kelompok objek gambar, maka baru dilakukan penyimpanan terhadap vektor fitur yang berupa histogram rata-rata tersebut. Berdasarkan fitur yang dihasilkan maka variabel yang disimpan dalam tabel basis data fitur adalah nama buah dan histogram rata-rata. Sehingga setiap record data pada tabel basis data fitur berisi Nama buah dan $H(0)$, $H(1)$, ..., $H(n)$, dimana n adalah jumlah indeks warna.

2.7 Teknik Matching

Teknik *matching* paling dasar adalah template matching, yaitu mencari selisih dari kedua fitur. Selisih inilah yang digunakan sebagai dasar untuk menyatakan apakah kedua objek dikatakan mirip (sama) atau tidak. Template Matching dikembangkan dengan dua cara yaitu dengan selisih Euclidian dan selisih Manhattan.

$$\text{Selisih Euclidian: } d = \sqrt{\sum_i (h_i^1 - h_i^2)^2} \quad (2.11)$$

$$\text{Selisih Manhattan: } d = \sum_i \|h_i^1 - h_i^2\| \quad (2.12)$$

dimana:

h_i^1 = adalah vektor berupa histogram indeks

h_i^2 = adalah vektor berupa histogram dalam database

Suatu objek dikatakan sama bila selisih d kurang dari toleransi ϵ yang ditentukan sebelumnya. Teknik lain yang dapat digunakan adalah menggunakan

korelasi dari kedua buah fitur. Teknik ini cukup baik dalam melakukan identifikasi. Untuk menghitung korelasi kedua fitur distribusi warna dilakukan dengan:

$$\text{Corr}(h^{(1)}, h^{(2)}) = \frac{\sum_i (h_i^{(1)} - h^{(1)})(h_i^{(2)} - h^{(2)})}{n \cdot s^{(1)} \cdot s^{(2)}} \quad (2.13)$$

Dimana:

n = ukuran fitur

$s^{(1)}$ = standard deviasi dari distribusi warna $h^{(1)}$

$s^{(2)}$ = standard deviasi dari distribusi warna $h^{(2)}$

Sayangnya teknik ini tidak begitu baik untuk identifikasi dengan melibatkan latar belakang yang cukup besar prosentasenya dibandingkan dengan objeknya sehingga korelasi kedua fitur akan cenderung positif. Teknik lainnya adalah teknik Bayes, dimana dicari probabilitas suatu warna pada h^1 ada dalam warna pada h^2 , yang didefinisikan dengan:

$$\Pr(x \in h^1 | y \in h^2) \cong \Pr(y \in h^2 | x \in h^1) \cdot \Pr(x \in h^1) \quad (2.14)$$

Teknik matching pada image processing merupakan pencocokan antara citra *query* dengan citra yang ada pada basis data. Citra *query* adalah suatu citra yang dijadikan acuan informasi (*content*) dalam proses pencarian. Sedangkan Citra Basis data adalah sekumpulan citra yang akan digunakan sebagai basis data citra. Dari kumpulan citra ini akan dicari citra yang mempunyai kemiripan dengan gambar *template*-nya.

Salah satu metode teknik matching adalah dengan menggunakan konsep perhitungan jarak. Bila fitur yang digunakan dalam ekstraksi fitur adalah fitur warna, dan proses fitur warna dilakukan dengan cara histogram warna, maka dengan menghitung jarak dari nilai histogram RGB pada masing-masing citra (citra *query* dengan citra basis data) dapat ditentukan atau diputuskan identifikasi dari sistem yang ada dalam image processing. Citra yang mempunyai jarak yang minimal adalah citra basis data yang paling mirip dengan citra *query*. Persamaan yang digunakan untuk menghitung jarak di atas adalah sebagai berikut:

$$d = \sum_x \sum_y \sqrt{(R_{x,y}^Q - R_{x,y}^T)^2 + (G_{x,y}^Q - G_{x,y}^T)^2 + (B_{x,y}^Q - B_{x,y}^T)^2} \quad (2.15)$$

Dimana $R_{x,y}^Q$, $G_{x,y}^Q$, $B_{x,y}^Q$ adalah warna pada citra *query* sedangkan $R_{x,y}^T$, $G_{x,y}^T$, $B_{x,y}^T$ warna pada citra target dalam basis data.

BAB III

PERANCANGAN

BAB 3

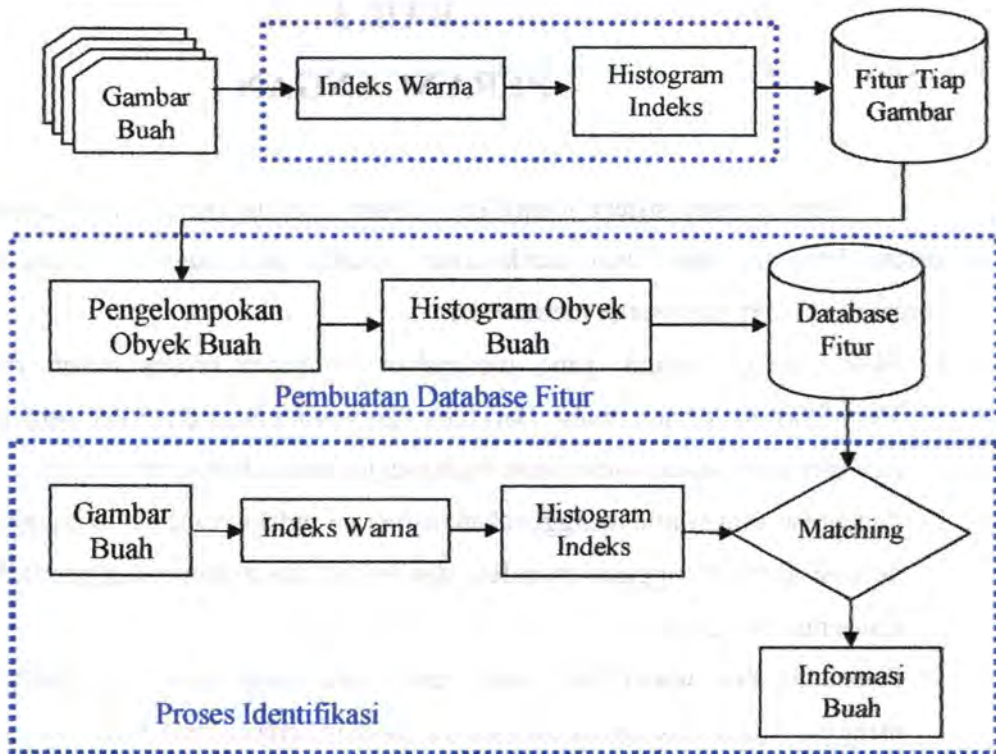
PERANCANGAN

Perancangan sistem identifikasi dalam bab ini membahas bagaimana sistem dirancang dan dibuat berdasarkan pustaka yang diambil. Perancangan sistem disini bertujuan untuk merumuskan:

1. Blok diagram sistem, yang merupakan gambaran proses dalam sistem identifikasi ini secara umum. Dari blok diagram ini akan diperoleh penjelasan dari cara kerja sistem identifikasi buah menggunakan histogram indeks.
2. Penentuan fitur warna menggunakan histogram indeks, yang berisi bagaimana definisi warna menggunakan indeks dan bagaimana merumuskan perhitungan histogram itu sendiri.
3. Penentuan fitur warna pada setiap jenis buah, yang berisi bagaimana cara menggabungkan histogram dari semua gambar dengan jenis buah yang sama dan kemudian memperoleh histogram untuk satu jenis gambar. Dengan demikian satu jenis buah hanya mempunyai satu record data fitur dan ini menambah kecepatan dalam template matching.
4. Proses identifikasi, yang berisi teknik template matching yang digunakan dalam sistem identifikasi buah ini. Teknik template matching ini menggunakan jarak antar vektor ciri dari gambar query dan vektor-vektor ciri dari histogram bersama pada setiap jenis buah.

3.1 Blok Diagram Sistem

Blok diagram sistem identifikasi buah secara keseluruhan digambarkan pada gambar 3.1 berikut.



Gambar 3.1 Blok diagram sistem

Berdasarkan blok diagram di atas, sistem identifikasi buah ini secara global mempunyai 3 kelompok blok diagram proses yang terdiri dari:

1. **Pengambilan fitur dari gambar buah**, dalam penelitian ini menggunakan histogram indeks dengan derajat skala warna yang diturunkan sehingga mengurangi ukuran fitur. Mengacu pada penelitian Ravishanka[5] yang menggunakan skala warna 25 diharapkan pada penelitian ini dapat digunakan skala warna yang lebih kecil dari 16. Penurunan ini diharapkan mampu menambah performa kecepatan perhitungan.
2. **Pembuatan Database Fitur**, dalam proses ini dilakukan penyimpanan fitur setiap buah yang disimpan sebagai data acuan (*image database*). Dari setiap fitur ini akan disajikan fitur setiap obyek buah menggunakan rata-rata fitur dari buah yang sama yang merupakan penurunan sifat homogenitas vektor fitur.

3. **Proses Identifikasi buah**, untuk proses identifikasi menggunakan *template matching* berdasarkan fitur obyek buah yang sudah dikelompokkan sehingga dapat mengurangi penggunaan waktu.

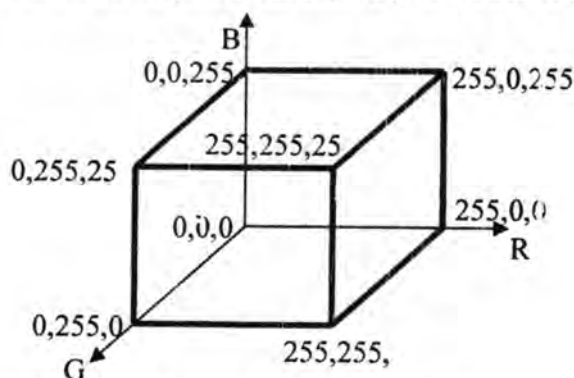
3.2 Penentuan Fitur Warna Dengan Histogram Indeks

3.2.1 Definisi dan Format Warna

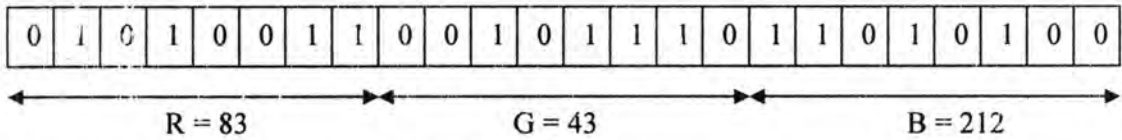
Pada pengolahan citra, definisi citra I adalah suatu matriks dua dimensi dengan ukuran $n_x \times n_y$ dengan nilai setiap piksel $I(x,y)$ menyatakan nilai warna dari piksel itu sendiri. Padahal warna bukanlah besaran numerik, sehingga untuk menyatakan suatu warna perlu proses transformasi warna menjadi komponen-komponen warna dasar sebagai pembentuk warna dengan menyatakan nilai derajat kecerahan dari setiap komponen warna.

Salah satu format standar di dalam pengolahan citra adalah format warna RGB, yang mempunyai tiga komponen dasar yaitu R (*red*), G (*green*) dan B (*blue*). Pada format warna RGB ini, setiap komponen warna R, G dan B mempunyai kuantisasi sebesar 256 derajat keabuan. Format warna RGB ini adalah digambarkan seperti pada gambar 3.2.

Struktur data pada format warna RGB yang dinyatakan dalam format *true-color*, dituliskan dalam format biner dengan masing-masing komponen warna R, G dan B mempunyai panjang 8 bit atau 256 nilai derajat keabuan. Sehingga jumlah bit pada sebuah warna menggunakan format RGB adalah 24 bit yang terdiri 8 bit R, 8 bit G dan 8 bit B. Model data dari format warna *true-color* menggunakan RGB digambarkan seperti gambar 3.3 di bawah ini.



Gambar 3.2 Model Kubik Format Warna RGB



Gambar 3.3 Format warna RGB dalam komposisi bit

Tabel 3.1 Contoh Warna dengan nilai komponen R, G dan B

Warna	R	G	B	Warna	R	G	B
Hitam	0	0	0	Abu-abu	128	128	128
Putih	255	255	255	Coklat	172	82	0
Merah	255	0	0	Orange	255	166	77
Hijau	0	255	0	Pink	235	157	200
Biru	0	0	255	Navy	25	0	160
Kuning	255	255	0	Ungu	128	0	255
Magenta	255	0	255	Dark Green	0	90	0
Cyan	0	255	255	Light Green	190	255	199

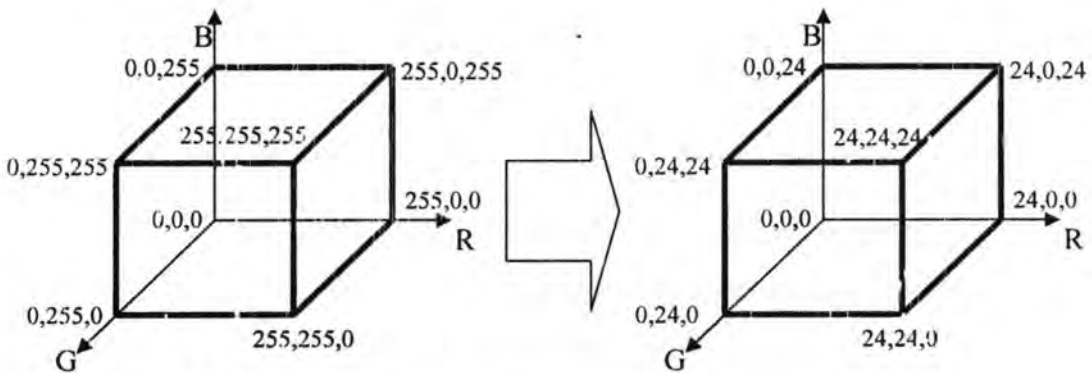
Terlihat pada tabel 3.1 dan gambar 3.3 bahwa sebuah warna dalam format RGB merupakan komposisi bersama dari tiga komponen R, G dan B. Sehingga sebuah warna w dituliskan dengan:

$$w = (r, g, b) \tag{3.1}$$

3.2.2 Indeks Warna

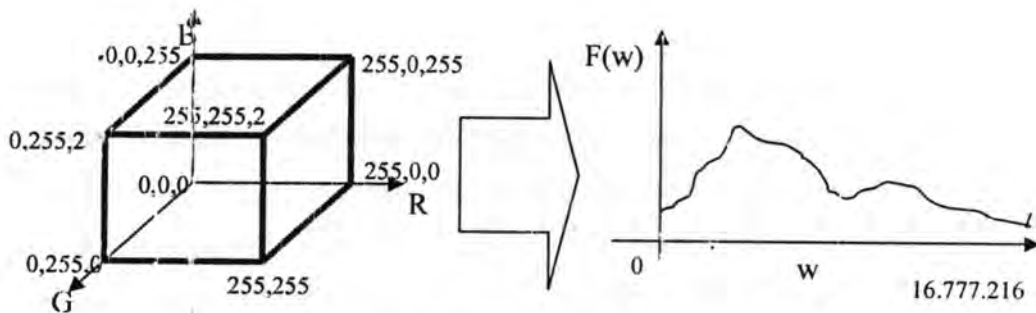
Seperti telah dijelaskan pada sub bab 3.2.1, bahwa suatu warna dengan format RGB dinyatakan sebagai komposisi dari nilai komponen dasar warna Red (R), Green (G) dan Blue (B). Dengan demikian suatu fungsi F terhadap komponen warna R, G dan B, dan dituliskan dengan $F(r, g, b)$. Bentuk ini adalah bentuk array 3 dimensi. Bila setiap komponen warna mempunyai kuantisasi 256 derajat keabuan maka jumlah warna yang mungkin adalah $256^3 = 16.777.216$ warna yang disusun dari $H(0,0,0)$ sampai dengan $H(255,255,255)$. Ini suatu jumlah yang sangat besar.

Untuk mengurangi jumlah warna menjadi lebih sedikit dengan tidak mengurangi content yang ada pada sebuah gambar dilakukan penurunan kuantisasi pada setiap komponen warna menjadi 25 sehingga diperoleh konversi koordinat warna seperti gambar 3.4 berikut.



Gambar 3.4 Konversi kuantisasi komponen warna dari 256 menjadi 25

Berdasarkan format warna di atas dapat dikatakan bahwa suatu warna dinyatakan dalam model fungsi 3D. Model ini tidak dapat menjelaskan fungsi warna secara baik karena membayangkan dan melihat model fungsi 3D yang ditulis dengan $F(x,y,z)$ merupakan hal yang sulit. Untuk memudahkan dalam memberi penjelasan nilai warna dilakukan konversi dari fungsi 3D $F(x, y, z)$ menjadi fungsi 1D $F(w)$. Konversi dari (r, g, b) dalam bentuk kubik menjadi w inilah yang dinamakan dengan nilai indeks warna.



Gambar 3.5 Model Kubik dan model Indeks

Dari gambar 3.5 di atas terlihat bahwa pola distribusi warna menjadi lebih mudah diamati bila model warna yang digunakan adalah model indeks. Suatu warna indeks merupakan gabungan dari nilai kuantisasi setiap komponen warna, dan dibentuk dengan didasarkan pada sebuah tabel warna yang terdiri dari r indeks warna (kuantisasi warna). Indeks adalah unik dan diberikan dengan persamaan berikut

$$\text{Indeks} = \sum_{i=1}^n C_i * r^{n-i} \quad (3.2)$$

dimana:

n = jumlah daerah dominan,

C_i = indeks warna untuk daerah dominan ke- i , dan

r = kuantisasi derajat keabuan setiap komponen warna.

Pada penelitian ini komponen warna yang digunakan adalah R,G dan B. Bila masing-masing komponen warna RGB mempunyai nilai r , g dan b dengan skala derajat keabuan sebesar n , maka indeks dapat dirumuskan dengan:

1. Untuk dominasi warna merah:

$$\text{Indeks} = n^2.r + n.g + b \text{ atau } \text{Indeks} = n^2.r + g + n.b \quad (3.3)$$

2. Untuk dominasi warna hijau:

$$\text{Indeks} = n.r + n^2.g + b \text{ atau } \text{Indeks} = r + n^2.g + n.b \quad (3.4)$$

3. Untuk dominasi warna biru:

$$\text{Indeks} = r + n.g + n^2.b \text{ atau } \text{Indeks} = n.r + g + n^2.b \quad (3.5)$$

Der gan jumlah derajat keabuan setiap komponen warna sebesar n , yang selanjutnya disebut dengan skala warna, maka akan menghasilkan jumlah indeks warna sebesar n^3 . Sehingga bila skala warna=256, seperti pada *true-color*, maka akan menghasilkan jumlah indeks warna sebesar $256^3=16.777.216$ indeks warna.

Mengacu pada penelitian [5] yang menggunakan skala warna 25, maka jumlah indeks warna yang digunakan sebesar $25^3=15.625$ indeks warna. Sehingga satu kali perhitungan indeks warna akan membutuhkan array yang berukuran 15.625, yang merupakan jumlah ukuran yang sangat besar untuk proses

komputasi. Bayangkan bila ukuran gambar yang digunakan adalah 320×240 piksel (ukuran default yang diberikan oleh webcam, dan ukuran terkecil pada sebuah kamera digital), maka bila diambil setiap pikselnya sebagai suatu fitur akan dihasilkan 76.800 nilai fitur. Sehingga terjadi pengurangan fitur sebesar 20.35% dengan skala warna 25, tentu pada ukuran tersebut sudah cukup baik sebagai jumlah fitur yang mewakili suatu gambar.

Untuk menunjukkan indeks mana yang baik di dalam implementasi sistem pengenalan buah, akan dicoba beberapa warna yang banyak muncul di dalam gambar buah seperti merah, hijau, kuning, orange, coklat dan ungu seperti tabel 3.1 di bawah ini.

Tabel 3.2 Model indeks warna-warna dominan pada gambar buah dengan skala warna 16

Warna	R	G	B	1(a)	1(b)	2(a)	2(b)	3(a)	3(b)
Merah	15	0	0	3840	3840	240	15	15	240
Orange	15	8	1	3969	3864	2289	2079	399	504
Kuning	15	15	0	4080	3855	4080	3855	255	255
Hijau	0	15	0	240	15	3840	3840	240	15
Coklat	9	4	0	2368	2308	1168	1033	73	148
Ungu	6	0	13	1549	1744	109	214	3334	3424

Dimana:

1. untuk warna dominan merah:

(a)

(b)

$$\text{Indeks} = 16^2 \cdot r + 16 \cdot g + b \quad \text{atau} \quad \text{Indeks} = 16^2 \cdot r + g + 16 \cdot b \quad (3.6)$$

2. untuk warna dominan hijau:

$$\text{Indeks} = 16 \cdot r + 16^2 \cdot g + b \quad \text{atau} \quad \text{Indeks} = r + 16^2 \cdot g + 16 \cdot b \quad (3.7)$$

3. untuk warna dominan biru:

$$\text{Indeks} = r + 16 \cdot g + 16^2 \cdot b \quad \text{atau} \quad \text{Indeks} = 16 \cdot r + g + 16^2 \cdot b \quad (3.8)$$

Dalam penelitian ini rumus untuk menjadikan indeks warna adalah 3.6(a), karena warna merah dan hijau dominan di hampir semua gambar buah.

Sebetulnya buah yang berwarna kuning juga banyak. Tetapi dalam format RGB, warna kuning adalah kombinasi dari warna merah dan warna hijau.

3.2.3 Histogram Indeks

Histogram $H(x)$ adalah suatu cara untuk mendapatkan jumlah kemunculan suatu kejadian X , atau dengan kata lain menggambarkan pola peluang terjadinya suatu kejadian X . Demikian pula, histogram pada warna menyatakan peluang munculnya suatu warna di dalam suatu citra.

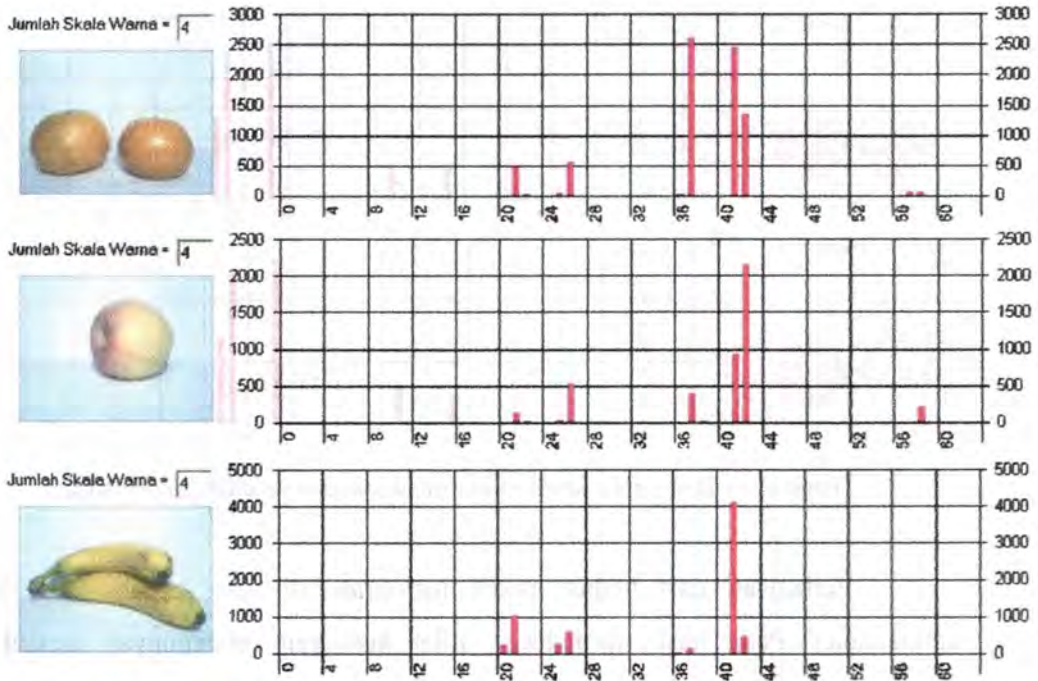
Dengan menggunakan model warna indeks, maka X menyatakan nomor indeks warna W . Sehingga histogram indeks adalah suatu bentuk histogram yang menggambarkan peluang terjadinya suatu indeks warna di dalam sebuah citra, yang didefinisikan dengan:

$$H(w) = \Pr(w | I(x, y) \in I, w \in W) \quad (3.9)$$

dimana:

- N =Himpunan seluruh indeks yang akan digunakan dengan jumlah indeks warna sebesar n^3
- dengan n adalah skala warna pada masing-masing komponen warna.

Gambar 3.6 menggambarkan bagaimana histogram indeks dengan skala 8 bekerja pada sebuah citra berwarna. Dan terlihat semua histogram akan berbeda, tetapi antara buah jeruk dan apel hijau menunjukkan sedikit kemiripan. Tetapi hanya dengan menunjukkan histogram pada satu buah dengan satu buah yang lain tidak dapat mewakili bahwa histogram itu adalah histogram acuan dari satu jenis buah. Sehingga dalam penelitian ini jumlah gambar untuk satu jenis buah digunakan 30 gambar, yang kemudian dikelompokkan sesuai dengan jenis buahnya dan dihitung histogram setelah pengelompokan.

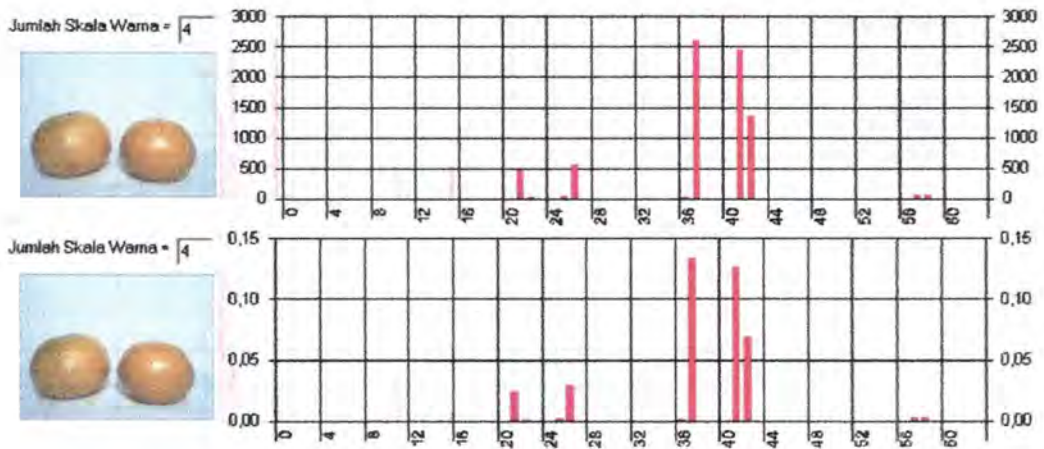


Gambar 3.6 Contoh histogram indeks pada beberapa gambar buah

Gambar 3.6 sebenarnya histogram yang masing-masing tidak bisa dibandingkan secara grafis, karena mempunyai skala yang berbeda. Terlihat skala pada tiap-tiap gambar adalah 3000, 1500 dan 4000. Sehingga untuk dapat membandingkan, perlu dilakukan normalisasi, sehingga hasil histogramnya menjadi fungsi probabilitas kemunculan setiap indeks warna, yang dituliskan dengan:

$$P(x) = \frac{H(x)}{\sum_{i=0}^N H(i)} \tag{3.10}$$

Hasil normalisasi ini akan menyebabkan nilai dari setiap histogram akan berada pada range 0 sampai dengan 1, dan jumlah dari seluruh kemunculan dalam histogram sama dengan 1. Proses normalisasi dari histogram di atas, dapat juga dikatakan sebagai fungsi kepadatan probabilitas, dan nilainya bukan lagi jumlah kemunculan suatu indeks warna tetapi lebih merupakan nilai probabilitas dari munculnya suatu indeks.



Gambar 3.7 Histogram sebelum dinormalisasi dan sesudah dinormalisasi

Perbedaan dari kedua model histogram di atas adalah pada nilai kemunculan. Pada hasil normalisasi, nilai histogram mempunyai spesifikasi seperti halnya sebuah fungsi kepadatan probabilitas yaitu:

1. Nilai probabilitas: $0 \leq \Pr(w) \leq 1$
2. Jumlah semua kejadian adalah 1, atau ditulis dengan: $\sum_{w \in W} \Pr(w) = 1$

3.3 Pembuatan Fitur

Proses pembuatan fitur dilakukan dengan mengubah format warna RGB menjadi suatu indeks satu dimensi. Proses ini dapat dibagi menjadi 3 proses yaitu:

1. Proses Capture untuk membaca data image dari kamera
2. Membuang latar belakang gambar sehingga yang diperoleh gambar buah saja. Teknik untuk membuang latar belakang digunakan differensial antara gambar buah dan gambar latar belakang.
3. Proses pengubahan skala warna menggunakan indeks warna dengan formula yang diambil dari [5] sebagai berikut:

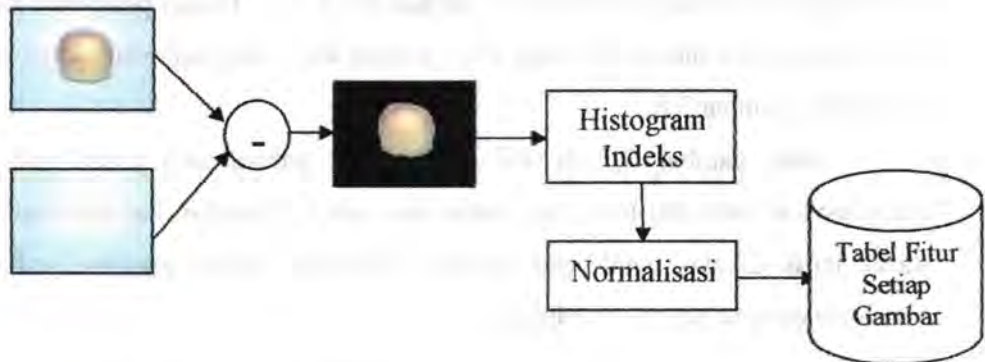
$$\text{Indeks} = n^2 w_1 + n w_2 + w_3 \quad (3.11)$$

dimana :

n = adalah skala warna

w_1, w_2 dan w_3 = adalah nilai setiap elemen warna

4. Proses perhitungan histogram semua piksel pada gambar 3.8. Untuk mendapatkan hasil yang optimal, maka diperlukan normalisasi dengan menjadikan histogram menjadi fungsi kepadatan probabilitas.



Gambar 3.8 Proses Pengambilan Fitur

3.4 Penggabungan Fitur

Hasil dari seluruh vektor fitur yang berupa histogram ini disimpan dalam sebuah tabel database. Bila asumsi satu jenis buah menggunakan 30 gambar maka dalam tabel akan mempunyai 30 *record* data, dan bila jenis buah yang digunakan adalah 20 buah maka di dalam tabel terdapat 600 *record*. Dan bila menggunakan skala warna 16 dalam tabel terdapat $16^3=4.096$ atribut. Hal ini akan menyebabkan proses *retrieval* menjadi sangat lambat.

Untuk bisa mengatasi hal di atas, maka untuk setiap gambar yang dikelompokkan sesuai dengan jenis buah yang ada dalam gambar tersebut. Untuk membuat fitur pada masing-masing kelompok digunakan nilai rata-rata pada setiap indeks warna yang dinyatakan sebagai berikut:

$$H_i = \frac{\sum_{k \in K} h_i^k}{N} \quad (3.12)$$

dimana:

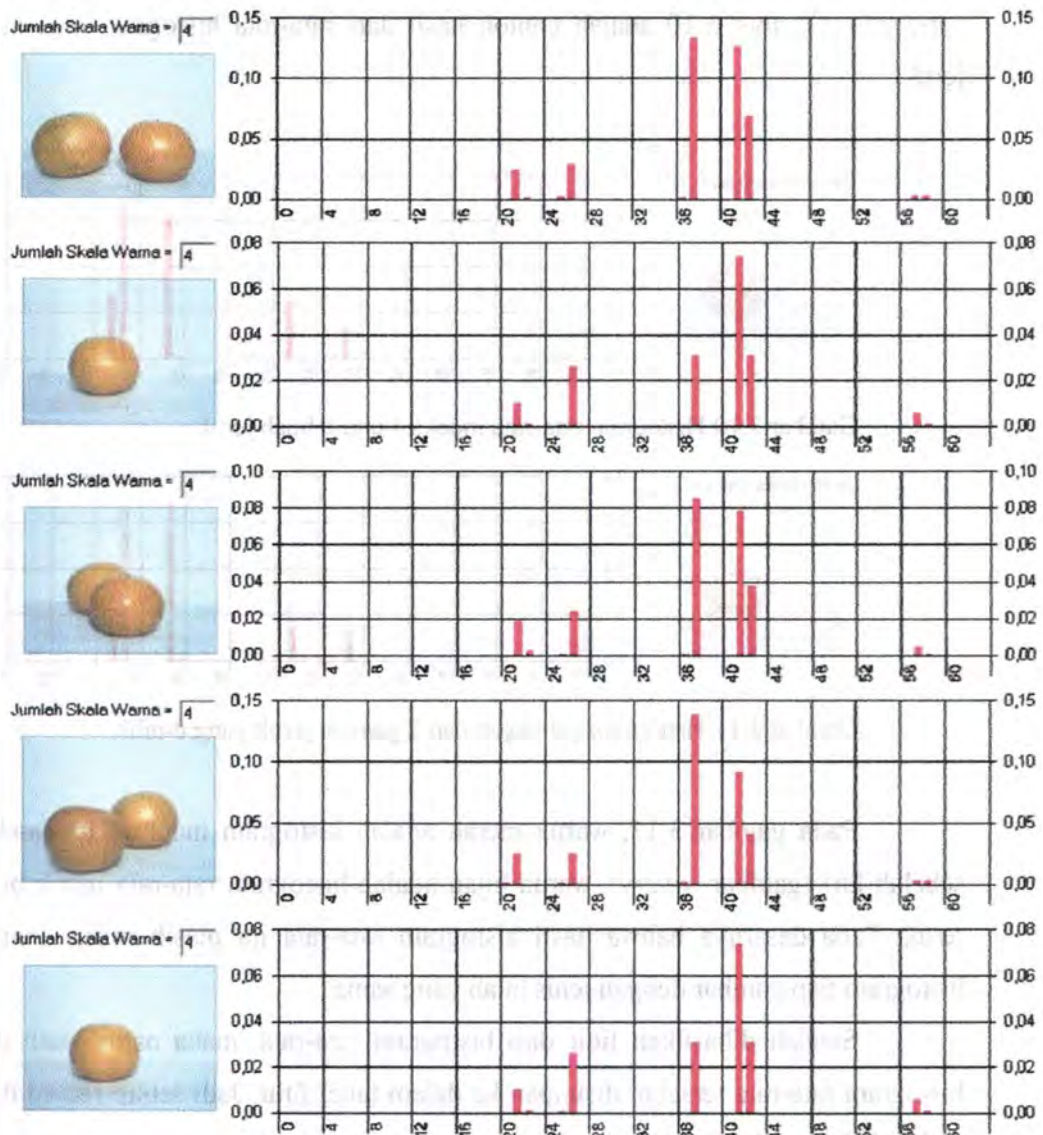
H adalah histogram dari setiap buah (kelompok gambar)

h adalah histogram pada setiap file/record dari data

N adalah jumlah data pada setiap buah, dalam hal ini digunakan 30 karena setiap buah diwakili oleh 30 record.

Data gambar buah yang digunakan dalam penelitian ini menggunakan 8 macam buah dengan setiap buah mempunyai 30 gambar untuk training (pembuatan basis data fitur) dan 10 gambar untuk test. Proses pengambilan fitur dengan histogram indeks ini untuk jenis gambar dari jenis buah yang sama seperti terlihat ada gambar 3.8.

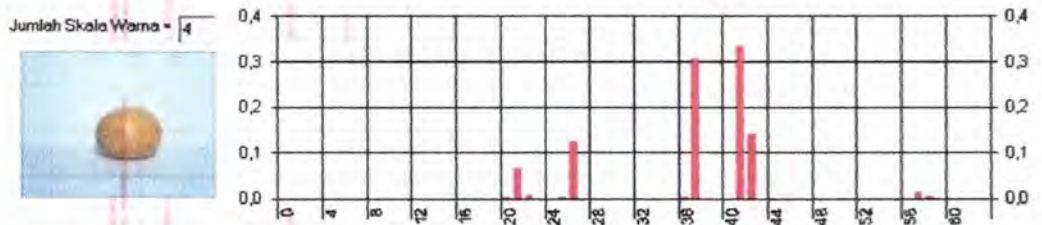
Pada gambar 3.7 di atas ditunjukkan bahwa satu jenis buah akan menghasilkan pola distribusi yang sama atau mirip. Mungkin hal ini disebabkan warna jeruk di atas betul-betul orange. Kemudian dipilih gambar buah yang warnanya campur seperti apel hijau.



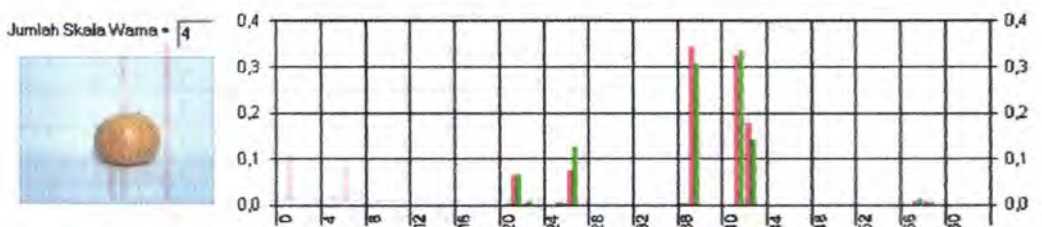
Gambar 3.9 Histogram indeks skala 4 pada gambar-gambar jeruk sejenis

Pada gambar 3.9 di atas, terlihat bahwa ketiga gambar jeruk tersebut mempunyai komponen indeks warna yang sama. Sehingga untuk menguji apakah sebuah gambar buah jeruk tidak diperlukan membandingkan histogram yang baru yang selanjutnya dinamakan dengan histogram query dengan histogram dari semua gambar, cukup dibandingkan dengan satu histogram yang bisa mewakili pola dari semua gambar jeruk. Untuk mendapatkan histogram gabungan dari semua gambar buah yang sama, digunakan nilai rata-rata dari semua histogram

yang ada. Gambar 3.10 adalah contoh hasil dari rata-rata histogram dari buah jeruk.



Gambar 3.10 Histogram rata-rata indeks 4 untuk buah jeruk

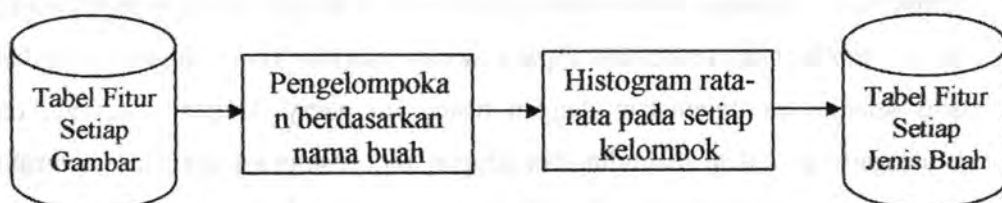


Gambar 3.11 Histogram gabungan dan 2 gambar jeruk yang dipilih

Pada gambar 3.11, warna merah adalah histogram indeks dari gambar sebelah kiri (gambar *query*), warna hijau adalah histogram rata-rata untuk buah jeruk. Pada dasarnya bahwa hasil histogram rata-rata ini masih mirip dengan histogram tiap gambar dengan jenis buah yang sama

Setelah dihasilkan fitur dari histogram rata-rata, maka nama buah dan histogram rata-rata tersebut disimpan ke dalam tabel fitur. Jadi setiap record data pada tabel fitur akan berisi:

- Nama buah
- $H(0)$ sampai dengan $H(N)$, dimana $N = \text{skala}^3$.



Gambar 3.12 Proses penggabungan fitur dengan histogram rata-rata

Berdasarkan gambar 3.12 di atas terlihat bahwa terdapat dua tabel fitur, yang pertama adalah tabel fitur pada setiap gambar, dan yang kedua adalah tabel fitur untuk setiap jenis buah. Untuk proses identifikasi digunakan tabel fitur pada setiap buah. Dengan setiap jenis buah mempunyai 30 gambar dan jumlah jenis buah 20, pada tabel fitur gambar terdapat 30x20 record data, sedangkan tabel fitur jenis buah hanya mempunyai 20 record data, Dengan demikian akan membuat proses pengenalan menjadi lebih cepat.

3.5 Proses Identifikasi

Proses identifikasi dilakukan dengan menggunakan template matching antara gambar yang dimasukkan dengan fitur yang dibuat dalam database fitur. Nilai matching ditentukan oleh nilai jarak Euclidian dari vektor gambar query dan vektor di database yang dituliskan sebagai berikut:

1. Rumus jarak menggunakan Euclidian

$$d = \sqrt{\sum_i (H_i - \hat{H}_i)^2} \quad (3.13)$$

2. Rumus jarak menggunakan Manhattan

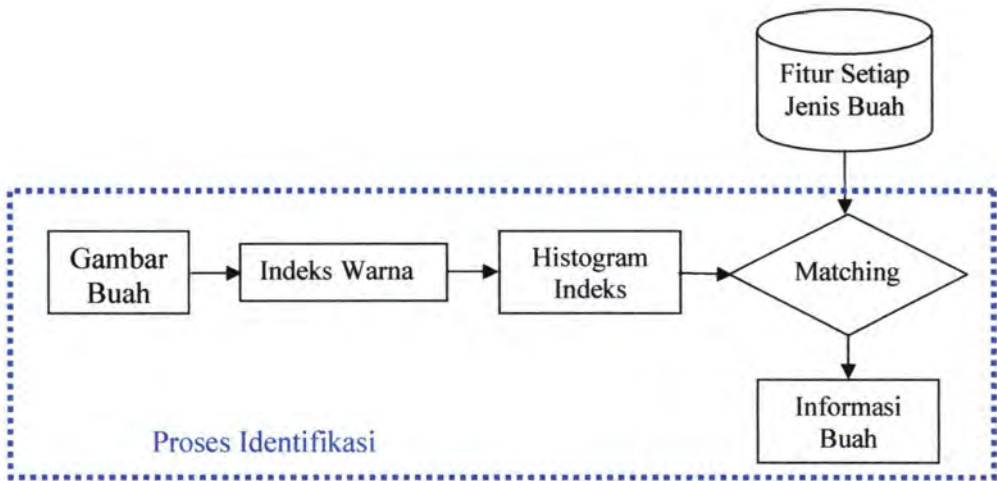
$$d = \sum_i |H_i - \hat{H}_i| \quad (3.14)$$

dimana:

H_i adalah vektor berupa histogram indeks dari gambar query

\hat{H}_i adalah vektor dari database

Buah yang dipilih menggunakan nilai minimum dari jarak euclidian pada masing-masing jenis buah. Proses identifikasi secara keseluruhan dapat digambarkan dengan blok diagram berikut:



Gambar 3.13 Blok diagram untuk proses identifikasi buah

BAB IV

PENGUJIAN DAN ANALISIS

BAB 4

PENGUJIAN DAN ANALISIS

Dalam bab ini akan dijelaskan tentang implementasi program sebagai hasil uji coba perancangan sistem perangkat lunak (*software*) pada bab sebelumnya. Juga akan dilakukan analisis terhadap hasil dari identifikasi terhadap obyek buah-buahan yang terdiri dari 8 jenis buah dengan masing-masing 30 gambar buah yang berbeda.

Implementasi program akan dibagi menjadi 2 tahap, yaitu tahap pelatihan (*training*) dan tahap pengenalan (*testing*). Pada tahap training dilakukan pengambilan fitur dari gambar buah, yang dalam penelitian ini menggunakan histogram indeks dengan derajat skala warna yang diturunkan sehingga mengurangi ukuran fitur. Mengacu pada penelitian [5] yang menggunakan skala warna 25 diharapkan pada penelitian ini dapat digunakan skala warna yang lebih kecil dari 16. Sementara itu pada tahap testing dilakukan uji coba untuk mengetahui akurasi pengenalan terhadap objek buah-buahan yang disediakan pada masing-masing skala warna yang digunakan serta menghitung prosentase kesalahannya.

4.1 Lingkungan Pengujian

Pengujian dilakukan pada PC compatible dengan spesifikasi sebagai berikut:

- Prosesor Intel Pentium IV 2 GHz
- Memori 256 MB
- Monitor: resolusi 1024 x 768 piksel, 120 dpi
- Jumlah data buah sebanyak 8 jenis buah dan jumlah gambar sebanyak 320, dengan 240 diambil sebagai data training dan 80 sebagai data test.



























4.2 Hasil Implementasi

Berdasarkan spesifikasi seperti pada (4.1), hasil implementasi pada setiap buah dengan menggunakan skala warna 4 adalah sebagai berikut:

Tabel 4.1 Hasil Pengenalan Buah

Nama file	Dikenali Sebagai	Error (%)	Keputusan	Nama file	Dikenali Sebagai	Error (%)	Keputusan
	ANGGUR	1,27%	BENAR		APEL MERAH	1,803%	BENAR
	ANGGUR	1,528%	BENAR		APEL MERAH	3,102%	BENAR
	ANGGUR	1,315%	BENAR		APEL MERAH	2,871%	BENAR
	ANGGUR	1,13%	BENAR		APEL MERAH	2,632%	BENAR
	ANGGUR	0,917%	BENAR		APEL MERAH	2,295%	BENAR
	ANGGUR	1,168%	BENAR		APEL MERAH	2,266%	BENAR
	ANGGUR	0,629%	BENAR		LOMBOK	2,587%	SALAH
	ANGGUR	0,833%	BENAR		APEL MERAH	3,235%	BENAR
	ANGGUR	1,454%	BENAR		APEL MERAH	5,104%	BENAR
	ANGGUR	1,246%	BENAR		APEL MERAH	3,335%	BENAR









Tabel 4.1 Hasil Pengenalan Buah (lanjutan)

Nama file	Dikenali Sebagai	Error (%)	Keputusan	Nama file	Dikenali Sebagai	Error (%)	Keputusan
	APEL HIJAU	2,378%	BENAR		JERUK	2,409%	BENAR
	APEL HIJAU	1,497%	BENAR		JERUK	1,832%	BENAR
	APEL HIJAU	3,497%	BENAR		JERUK	2,639%	BENAR
	APEL HIJAU	1,405%	BENAR		JERUK	1,286%	BENAR
	PEAR	2,596%	SALAH		APEL HIJAU	2,066%	SALAH
	JERUK	2,646%	SALAH		JERUK	3,105%	BENAR
	APEL HIJAU	2,283%	BENAR		JERUK	2,26%	BENAR
	APEL HIJAU	1,615%	BENAR		JERUK	2,023%	BENAR
	JERUK	1,263%	SALAH		JERUK	2,023%	BENAR
	APEL HIJAU	1,497%	BENAR		JERUK	0,664%	BENAR
	KLENGKE NG	1,323%	BENAR		LOMBOK	1,113%	BENAR
	KLENGKE NG	1,738%	BENAR		APEL MERAH	2,317%	SALAH
	KLENGKE NG	1,234%	BENAR		LOMBOK	1,111%	BENAR

Tabel 4.1 Hasil Pengenalan Buah (lanjutan)

Nama file	Dikenali Sebagai	Error (%)	Keputusan	Nama file	Dikenali Sebagai	Error (%)	Keputusan
	KLENGKE NG	1,228%	BENAR		APEL MERAH	0,738%	SALAH
	APEL MERAH	2,169%	SALAH		LOMBOK	0,844%	BENAR
	KLENGKE NG	1,087%	BENAR		LOMBOK	0,576%	BENAR
	KLENGKE NG	1,18%	BENAR		LOMBOK	1,872%	BENAR
	KLENGKE NG	1,49%	BENAR		LOMBOK	2,164%	BENAR
	KLENGKE NG	1,297%	BENAR		LOMBOK	2,91%	BENAR
	KLENGKE NG	0,917%	BENAR		APEL MERAH	3,16%	SALAH
	PEAR	0,891%	BENAR		KLENGKENG	2,252%	SALAH
	PEAR	1,455%	BENAR		PISANG	0,81%	BENAR
	PEAR	0,535%	BENAR		PISANG	2,225%	BENAR
	PEAR	0,698%	BENAR		PISANG	0,733%	BENAR
	PEAR	0,651%	BENAR		PISANG	1,738%	BENAR
	PEAR	1,132%	BENAR		PISANG	0,419%	BENAR

Tabel 4.1 Hasil Pengenalan Buah (lanjutan)

Nama file	Dikenali Sebagai	Error (%)	Keputusan	Nama file	Dikenali Sebagai	Error (%)	Keputusan
	PEAR	2,192%	BENAR		PISANG	2,232%	BENAR
	PEAR	1,43%	BENAR		PISANG	1,723%	BENAR
	PEAR	1,448%	BENAR		PISANG	0,724%	BENAR
	PEAR	1,621%	BENAR		PISANG	0,863%	BENAR

Tingkat akurasi pengenalan buah untuk skala warna 4 adalah 70 berhasil dikenali dari 80 buah atau sebesar 87,5%. Akurasi untuk masing-masing jenis buah dapat ditunjukkan dengan tabel 4.2.

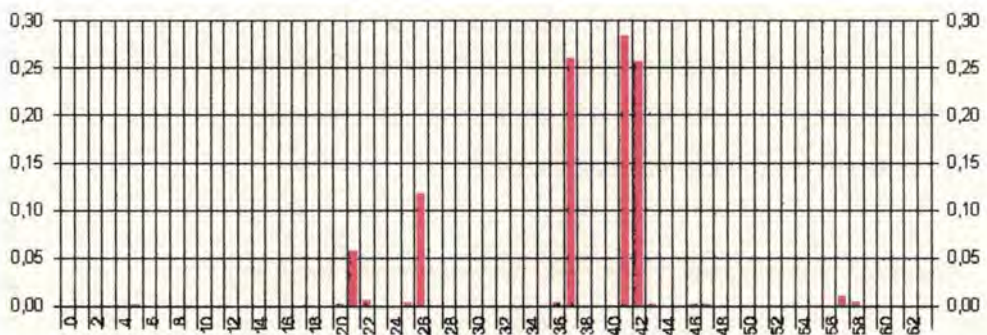
Tabel 4.2 Akurasi pada setiap jenis buah dalam %, untuk n=4

	Anggur	Apel Merah	Apel Hijau	Klengkeng	Lombok	Pisang	Pear	Jeruk
Anggur	100	0	0	0	0	0	0	0
Apel Merah	0	90	0	0	10	0	0	0
Apel Hijau	0	0	70	0	0	0	10	20
Klengkeng	10	0	0	90	0	0	0	0
Lombok	0	30	0	0	70	0	0	0
Pisang	0	0	0	10	0	90	0	0
Pear	100	0	0	0	0	0	0	0
Jeruk	0	10	0	0	0	0	0	90

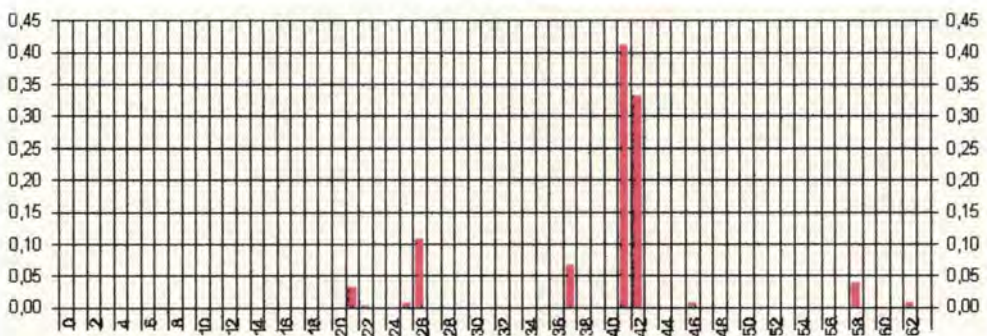
Dari data dalam tabel diatas terlihat bahwa kesalahan pengenalan terbesar terjadi pada buah apel hijau dan lombok. Jika diperhatikan kesalahan yang terjadi

dalam pengenalan terhadap buah apel hijau adalah kesalahan mengidentifikasi sebagai jeruk sebanyak 20%. Hal ini dapat dipelajari dari adanya kemiripan vektor fitur buah apel hijau dan jeruk yang ditunjukkan dalam perbandingan kedua histogramnya, sebagaimana gambar 4.1 berikut.

Gambar 4.1 memberikan penjelasan tentang kemiripan distribusi frekwensi nilai indeks warna atau histogram kedua jenis buah. Jika dilihat dari warna-warna yang dominan, kedua gambar memperlihatkan komponen warna yang sama, hanya ada perbedaan pada frekwensi atau jumlah kemunculan setiap indeks warnanya. Sehingga jika dihitung jarak antara kedua vektor fitur akan menjadi minimal, yang pada akhirnya teridentifikasi sebagai buah yang dianggap sama.



(1)

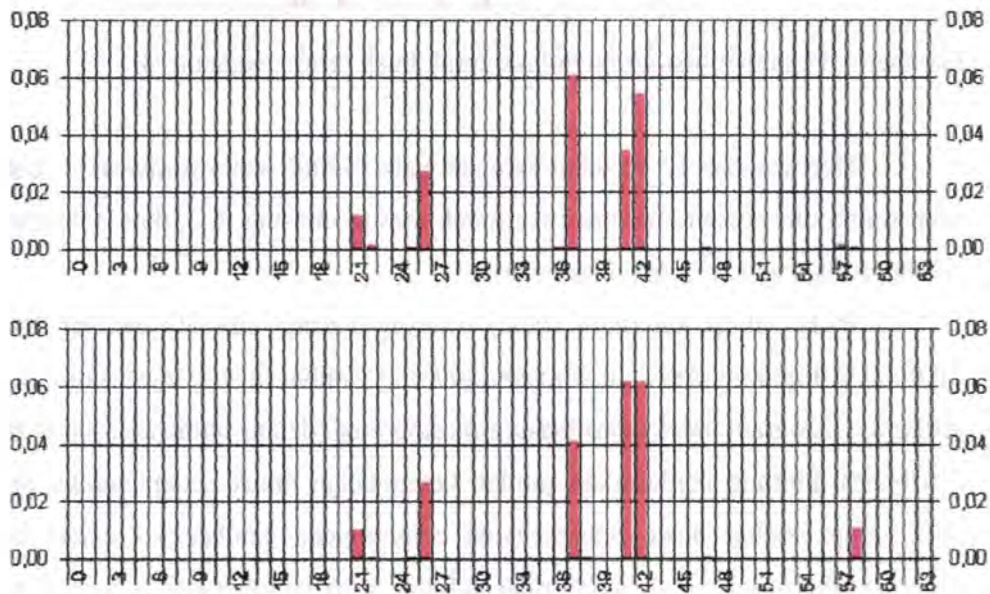


(2)

Gambar 4.1 Perbandingan histogram indeks rata-rata (i) jeruk dan (ii) apel hijau

Selanjutnya jika diperhatikan letak kesalahan identifikasi, maka akan diambil contoh pada apel hijau data ke-6 yang diidentifikasi sebagai jeruk

dan jarak data ke-5 yang teridentifikasi sebagai apel hijau. Hal ini dapat dipelajari dengan membandingkan kedua histogram indeks kedua buah tersebut sebagaimana gambar 4.2. Dengan membandingkan kedua histogramnya, maka akan dapat diketahui mengapa kesalahan identifikasi ini dapat terjadi, yaitu dengan melihat seberapa dekat nilai masing-masing histogram kedua buah tersebut.



Gambar 4.2 Histogram indeks buah apel hijau dan jeruk

Gambar 4.2 diatas memberikan gambaran tentang kemiripan histogram indeks kedua jenis buah dengan komponen warna yang ditunjukkan oleh kesamaan nilai indeks warna yang sama. Tetapi jumlah kemunculan indeks-indeks warna yang bersesuaian tidak sama dengan fluktuasi yang tidak tajam. Sehingga jika dua fitur ini dihitung nilai kemiripannya akan diperoleh jarak yang minimal, yang berarti akan teridentifikasi sebagai buah yang sama.

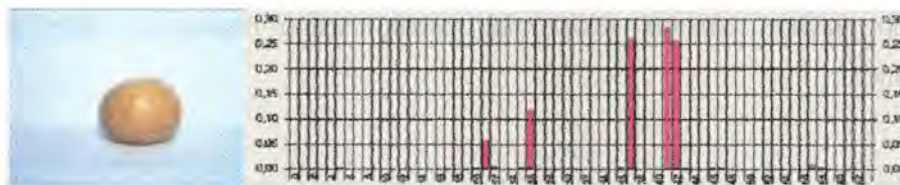
4.3 Pengujian Fitur Dalam Satu Kelompok

Tahap ini bertujuan untuk mengambil fitur vektor setiap jenis/kelompok buah dengan mengambil rata-rata dari 30 gambar buah yang disediakan untuk masing-masing kelompok dari keseluruhan buah yang akan diidentifikasi,



sehingga diperoleh fitur dari setiap jenis buah yang akan disimpan dalam basis data.

Berikut ini contoh histogram indeks untuk buah jeruk dengan mengambil skala warna $n = 4$.



Gambar 4.3 Gambar histogram indeks untuk buah jeruk dengan $n=4$

Dari gambar 4.3 tersebut nampak bahwa untuk skala warna $n=4$, berarti nilai warna diturunkan kuantisasinya untuk masing-masing R,G, dan B menjadi 4 sehingga akan ada $n^3 = 64$ indeks warna.

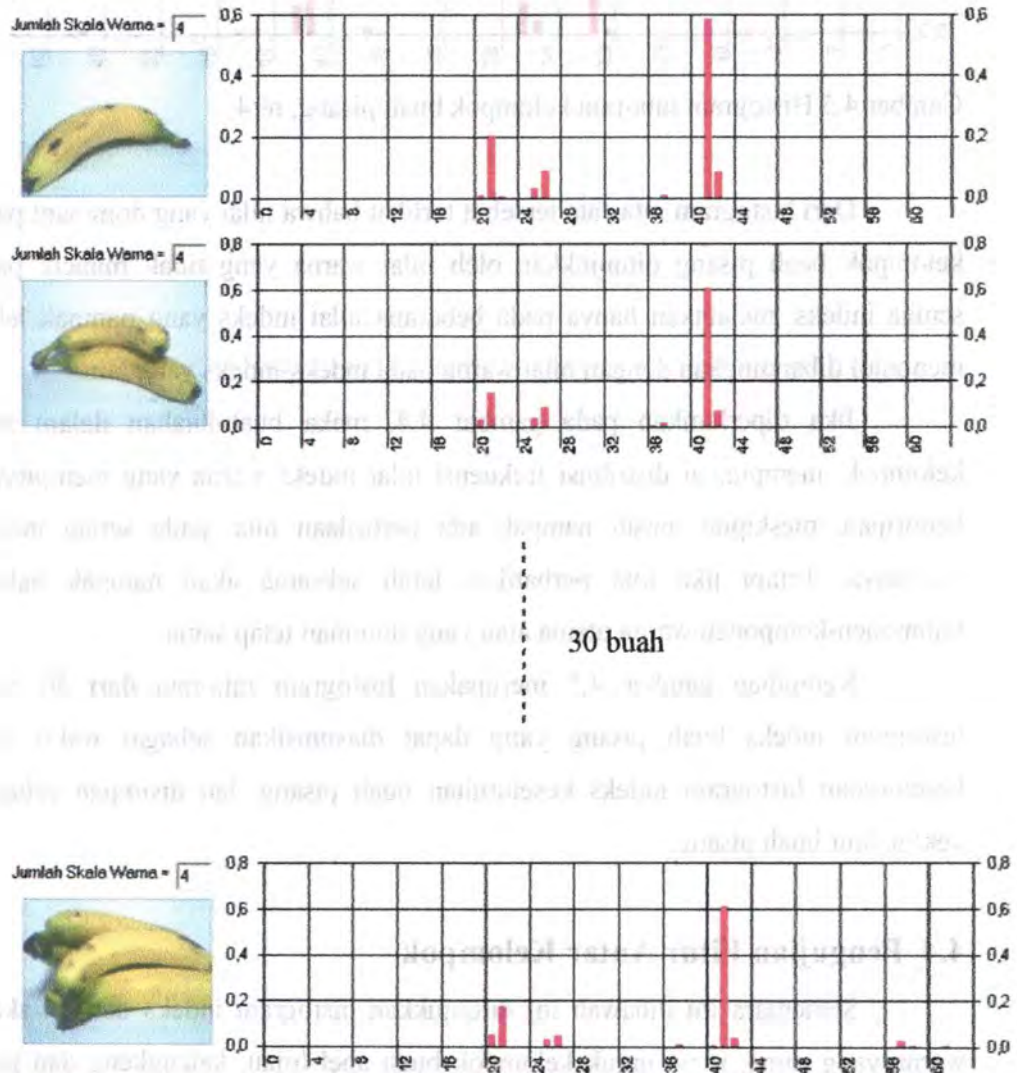
Pada setiap kelompok buah (masing-masing ada 30 gambar buah), dihitung histogram indeks untuk masing-masing gambar buah dengan skala warna tertentu. Histogram indeks dari masing-masing buah dalam setiap kelompok buah (30 buah) dihitung berdasarkan jumlah kemunculan untuk setiap indeks warna dari setiap gambar buah. Sebagaimana contoh pada gambar 4.3 diatas maka distribusi frekwensi untuk setiap nilai indeks warna memberikan informasi yang spesifik tentang buah masing-masing.

Histogram rata-rata dengan skala warna n untuk setiap jenis buah dihitung berdasarkan rata-rata dari seluruh histogram indeks dalam setiap jenis buah, yaitu dengan menghitung rata-rata jumlah kemunculan setiap indeks warna yang bersesuaian untuk setiap skala warna n yang sama pada setiap jenis buah. Kemudian rata-rata nilai histogram indeks dari 30 gambar buah pada setiap jenis buah yang selanjutnya disimpan dalam basis data sebagai vektor fitur buah untuk masing-masing jenis buah.

Ilustrasi dari uraian tersebut diatas diberikan melalui gambar 4.4. Sebagai contoh pembahasan tentang hal ini, berikut diberikan beberapa gambar histogram indeks dari beberapa gambar buah sejenis, dalam hal ini dicontohkan buah pisang. Kemudian dari 30 histogram indeks dalam kelompok buah pisang ini akan dihitung nilai rata-rata kemunculan setiap indeks warna yang bersesuaian

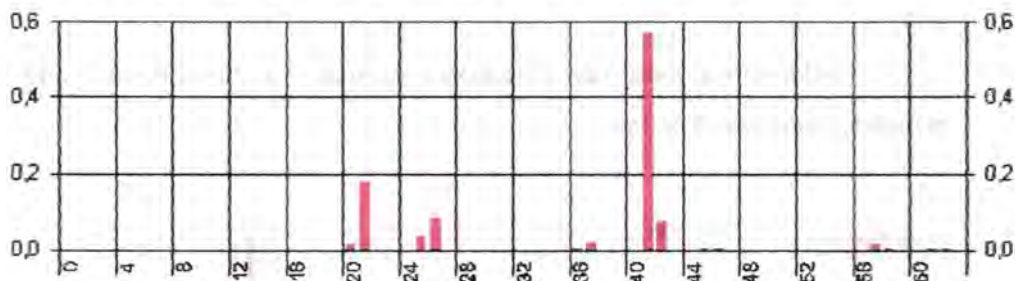
dengan mengambil skala warna yang sama (sebagai contoh $n=4$). Nilai rata-rata tersebut adalah histogram rata-rata untuk jenis buah pisang yang selanjutnya dalam sistem ini disimpan sebagai vektor fitur buah pisang untuk skala warna $n=4$.

Selanjutnya mari kita perhatikan gambar 4.4 dibawah ini yang akan memperjelas maksud diatas.



Gambar 4.4 Histogram indeks beberapa buah pisang sejenis

Histogram rata-rata dari kelompok buah pisang sebagaimana yang dijelaskan diatas merupakan rata-rata dari nilai histogram indeks dari seluruh 30 gambar buah pisang seperti yang ditunjukkan pada gambar 4.5.



Gambar 4.5 Histogram rata-rata kelompok buah pisang, n=4

Dari histogram rata-rata tersebut terlihat bahwa nilai yang dominant pada kelompok buah pisang ditunjukkan oleh nilai warna yang tidak muncul pada semua indeks, melainkan hanya pada beberapa nilai indeks yang nampak lebih menonjol dibandingkan dengan nilai warna pada indeks-indeks yang lain.

Jika diperhatikan pada gambar 4.4, maka buah-buahan dalam satu kelompok, mempunyai distribusi frekuensi nilai indeks warna yang mempunyai kemiripan, meskipun masih nampak ada perbedaan nilai pada setiap indeks warnanya. Tetapi jika kita perhatikan lebih seksama akan nampak bahwa komponen-komponen warna utama atau yang dominan tetap sama.

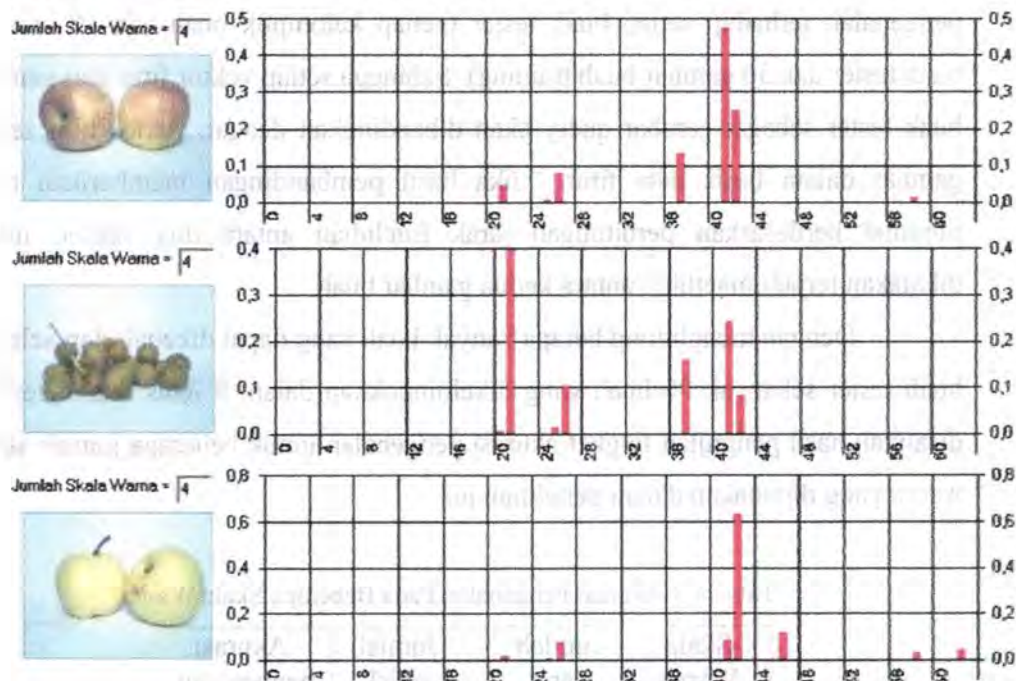
Kemudian gambar 4.5 merupakan histogram rata-rata dari 30 nilai histogram indeks buah pisang yang dapat diasumsikan sebagai wakil dari keseluruhan histogram indeks keseluruhan buah pisang dan disimpan sebagai vektor fitur buah pisang.

4.4 Pengujian Fitur Antar Kelompok

Sementara itu dibawah ini ditunjukkan histogram indeks dengan skala warna yang sama, n =4 untuk kelompok buah apel hijau, kelengkeng dan pear seperti terlihat pada gambar 4.6.

Gambar 4.6 diatas menunjukkan bahwa histogram untuk 3 contoh jenis buah dari kelompok buah yang berbeda, yaitu buah kelengkeng, apel dan pear dengan menggunakan skala warna yang sama n=4, sangat jelas nampak perbedaan

distribusi frekwensi nilai indeks warnanya dan juga terlihat perbedaan nilai-nilai indeks warna yang dominan dengan mencolok. Hal inilah yang digunakan sebagai fitur pembeda antara buah yang satu dengan yang lain.



Gambar 4.6 Histogram indeks 3 kelompok buah

Namun untuk buah-buahan yang mempunyai kemiripan warna meskipun bentuknya berbeda, akan didapatkan gambar histogram yang juga mempunyai kemiripan. Oleh karena itu untuk memperlihatkan perbedaan nilai warna pada histogram sebagai fitur pembeda dapat ditingkatkan kuantisasi warna dengan menambah skala warnanya. Dengan demikian akan diperoleh histogram yang berbeda. Semakin tinggi skala warna yang digunakan, maka akan semakin terlihat perbedaan yang nyata sebaran nilai pada histogramnya.

4.5 Pengujian Jumlah Skala Warna

Pengujian jumlah skala warna dimaksudkan untuk mengetahui seberapa jauh skala warna dapat diturunkan berdasarkan pengelompokan masing-masing obyek buah untuk menentukan vektor fitur masing-masing kelompok yang akan

digunakan sebagai parameter dalam proses identifikasi buah. Pengujian dilakukan dengan menggunakan skala warna mulai $n=4$ sampai target $n=16$.

Pada masing-masing skala warna yang digunakan, dilakukan pengujian pengenalan terhadap setiap buah tester (setiap kelompok buah ada 10 gambar buah tester dan 30 gambar buah training). Sehingga setiap vektor fitur dari gambar buah tester sebagai gambar query akan dibandingkan dengan vektor fitur setiap gambar dalam basis data fitur. Jika hasil perbandingan memberikan nilai minimal berdasarkan perhitungan jarak Euclidian antara dua vektor, maka dikatakan terjadi matching antara kedua gambar buah.

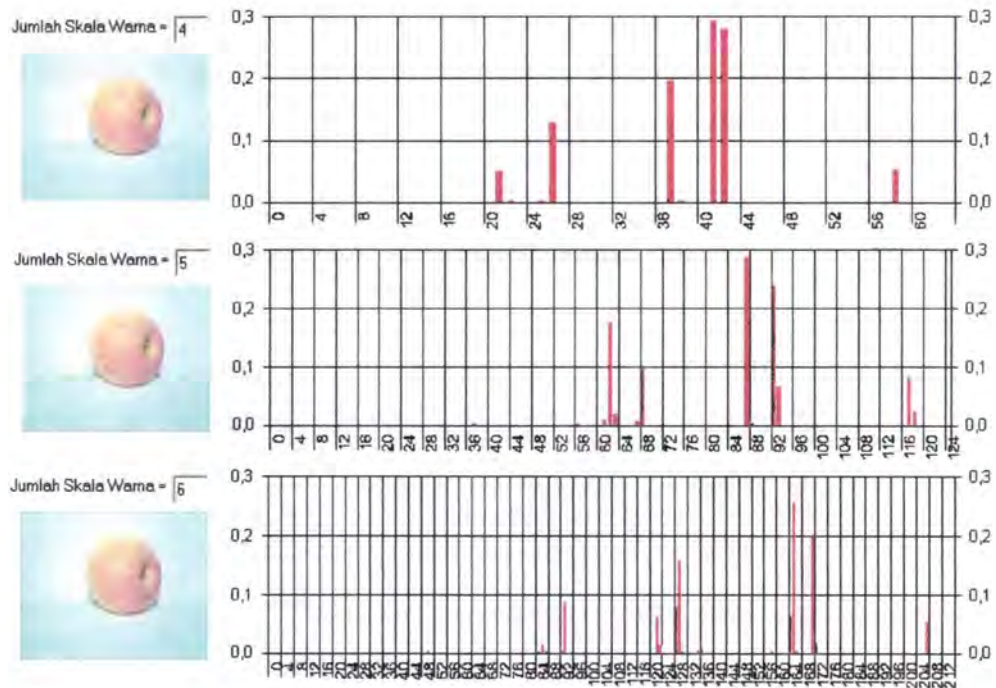
Dengan menghitung berapa banyak buah yang dapat dikenali dari seluruh buah tester sebanyak 80 buah yang dikelompokkan dalam 8 jenis buah. Berikut disajikan hasil pengujian tingkat akurasi pengenalan untuk beberapa jumlah skala warna yang digunakan dalam penelitian ini:

Tabel 4.3 Akurasi Pengenalan Pada Beberapa Skala Warna

Skala Warna	Jumlah yang dikenali	Jumlah objek	Akurasi pengenalan
4	66	80	88,75%
5	79	80	98,75%
6	76	80	93,75%
7	78	80	97,5%
8	80	80	100%
9	80	80	100%
10	80	80	100%
11	80	80	100%
12	80	80	100%
13	80	80	100%
14	80	80	100%
15	80	80	100%
16	80	80	100%

Dari tabel 4.3 diatas dapat dilihat dengan jelas bahwa pada skala warna $n=4$ akurasi pengenalan masih 88,75% atau dapat mengenali dengan baik 66 buah dari 80 buah yang dicobakan. Ini berarti masih ada kesalahan pengenalan terhadap buah yang digunakan sebagai tester sebesar 11,25%.

Untuk skala warna $n=5$ terjadi peningkatan akurasi pengenalan 10% dari sebelumnya menjadi 98,75%, tetapi pada kenaikan skala warna berikutnya, $n=6$ ternyata justru akurasi pengenalan turun menjadi 93,75%. Hal ini dimungkinkan karena faktor kuantisasi yang dilakukan dalam sistem ini, dimana nilai indeks pada beberapa komponen-komponen warna ikut terkuantisasi menjadi nilai pada indeks warna tetangganya. Sebagai gambaran, berikut ini disajikan hasil uji coba pengambilan fitur berupa histogram warna buah apel hijau yang sama untuk beberapa skala warna $n=4$, $n=5$ dan $n=6$.



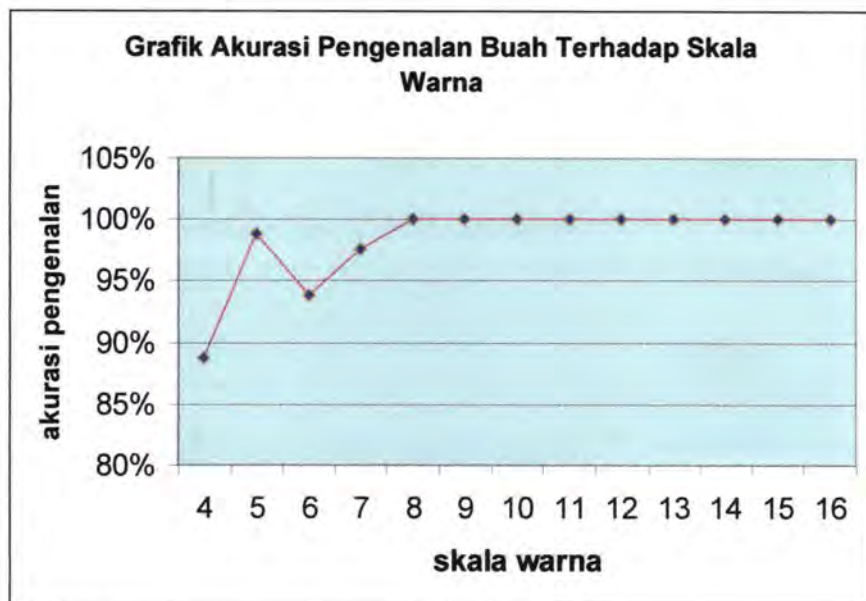
Gambar 4.7 Histogram indeks dari buah apel merah pada beberapa skala warna

Dari gambar diatas nampak bahwa ketiga histogram indeks untuk buah yang sama, apel hijau memberikan ilustrasi semakin menyebarnya komponen indeks warna yang terkandung pada suatu buah seiring dengan meningkatnya

jumlah skala warna yang digunakan. Hal ini berarti semakin tinggi jumlah skala warna yang digunakan, maka akan semakin menyebar nilai indeks warna yang mempunyai frekuensi ≥ 0 . Dengan demikian akan semakin lengkap informasi tentang setiap buah karena sebaran nilai indeks warna memberikan konsekuensi sebaran nilai yang dibandingkan dalam proses *matching*. Sehingga akan semakin akurat proses identifikasinya yang pada akhirnya akan meningkatkan akurasi.

Pada skala warna berikutnya, $n=7$ terjadi kenaikan akurasi pengenalan terhadap obyek buah-buahan menjadi 97,50%. Demikian juga terjadi kenaikan akurasi pengenalan untuk skala warna $n=8$ dan $n=9$ menjadi mutlak 100%.

Hal ini dapat lebih jelas dilihat pada gambar 4.8 tentang grafik akurasi pengenalan buah-buahan yang disuguhkan dengan menggunakan skala warna mulai dari $n=4$ sampai $n=16$. Pada grafik tersebut nampak adanya pergerakan naik turun tingkat akurasi pengenalan buah yang dimulai level 88,75% pada skala warna $n=4$, kemudian naik pada skala $n=5$ tetapi turun kembali pada skala berikutnya. Tetapi pada skala selanjutnya pergerakan tingkat akurasinya tidak menunjukkan fluktuasi yang tajam.



Gambar 4.8 Grafik akurasi pengenalan buah

Dari gambar 4.8 diatas dapat dilihat bahwa akurasi pengenalan telah mencapai maksimal pada $n \geq 8$, yaitu 100%, dimana pada skala warna sebelumnya masih menunjukkan pergerakan naik turun. Sehingga pada skala warna $n=8$ tersebut dapat dianggap sebagai skala warna yang optimal dalam penelitian identifikasi terhadap buah dengan batasan serta ruang lingkup penelitian yang telah ditentukan sebelumnya.

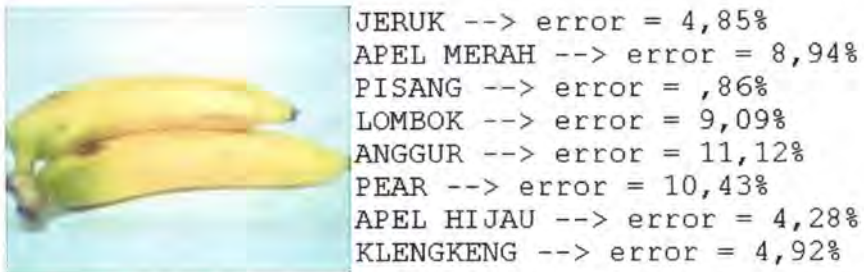
Setelah memperhatikan pada skala warna berikutnya tidak turun kembali dan telah mencapai maksimal akurasi pengenalan, maka dapat dianggap pengujian terhadap akurasi pengenalan buah tidak perlu dilakukan dengan menambah skala warna yang digunakan, dianggap cukup sampai pada skala warna $n=16$.

4.6 Pengujian Teknik Matching

Pada bagian ini akan dilakukan uji coba pada proses identifikasi buah dengan menghitung seberapa besar nilai kesalahan masing-masing pengenalan terhadap buah tester yang dicobakan. Hal ini dimaksudkan untuk melihat secara detil tingkat kesalahan dalam besaran numerik.

Proses identifikasi ini dilakukan dengan menggunakan template matching antara gambar query dengan gambar yang ada dalam basis data, yang dilakukan dengan membandingkan berdasarkan vektor fitur keduanya. Nilai matching ditentukan oleh nilai jarak Euclidian dari vektor gambar query dan vektor di basis data fitur yang dituliskan pada rumus jarak Euclidian sebagaimana dituliskan dalam bab 3. Kesesuaian atau kemiripan antara dua gambar dihitung berdasarkan nilai minimal jarak antara vektor gambar query dan vektor dalam basis data fitur.

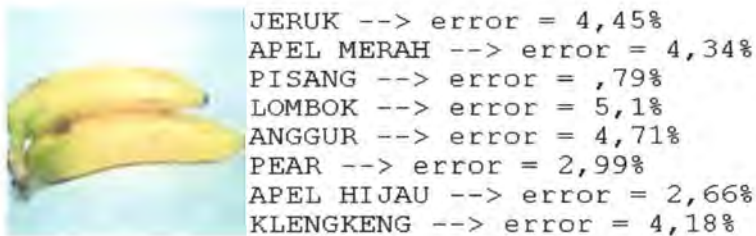
Kesesuaian kedua gambar ditentukan berdasarkan nilai minimal jarak antara dua vector fitur. Berikut ini disajikan perhitungan jarak euclidian antara kedua gambar yang memberikan gambaran tingkat kemiripannya.



Gambar 4.9 contoh penghitungan kemiripan buah untuk $n=4$

Dari gambar tersebut dapat dilihat bahwa gambar buah pisang dikenali sebagai buah pisang dengan kesalahan sebesar 0,86% yang jauh lebih kecil dibandingkan dengan tingkat kesalahan dalam mengenali jenis buah-buah yang lain.

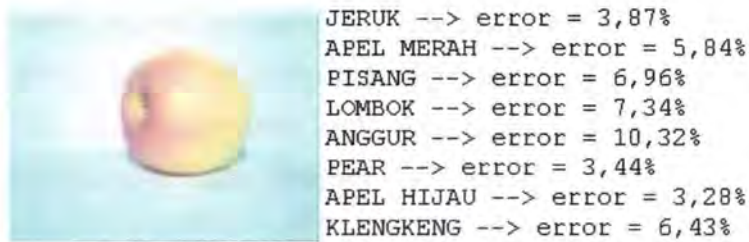
Untuk skala warna yang ditingkatkan menjad $n=5$ data kesalahan pengenalan adalah sebagai berikut:



Gambar 4.10 contoh hasil penghitungan kemiripan buah untuk $n=5$

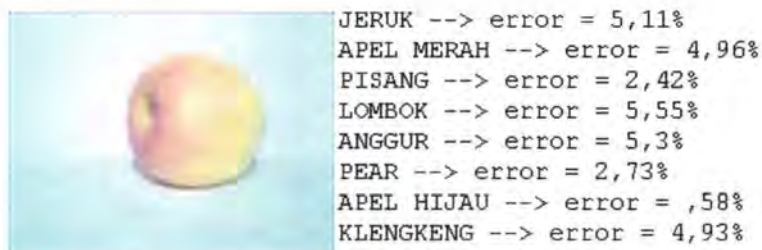
Dari data tersebut terlihat bahwa untuk $n=5$, kesalahan pengenalan terhadap obyek buah pisang yang sama dengan sebagai buah pisang menjadi 0,79%, berarti terjadi penurunan kesalahan sebesar 0,07%. Demikian seterusnya nilai error pengenalan terhadap obyek untuk skala warna $n=4$ yang diujikan.

Salah satu pengujian yang dilakukan untuk proses template matching ini adalah mengetahui pengaruh nilai skala warna dan akurasi dengan mempertimbangkan kemiripan dari dua buah berbeda dengan warna yang hampir sama seperti apel hijau dan pear, yang ditunjukkan pada gambar 4.11.



Gambar 4.11 Hasil identifikasi apel hijau dengan skala warna 4

Gambar 4.11 ini menunjukkan bahwa jarak antara histogram dari gambar query dan histogram pada basis data untuk buah apel hijau, pear dan jeruk masing-masing adalah 3.28%, 3.44% dan 3.87%. Nilai tersebut mempunyai perbedaan yang kecil yang disebabkan kemiripan warna dasar, dan perbedaan pada komponen warna tambahan. Hal tersebut di atas tidak terjadi lagi ketika skala warna dinaikkan menjadi 5 dan 6 seperti terlihat pada gambar 4.12 dan 4.13



Gambar 4.12 Hasil identifikasi apel hijau dengan skala warna 5



Gambar 4.13 Hasil identifikasi apel hijau dengan skala warna 6

4.7 Pengujian Waktu Proses

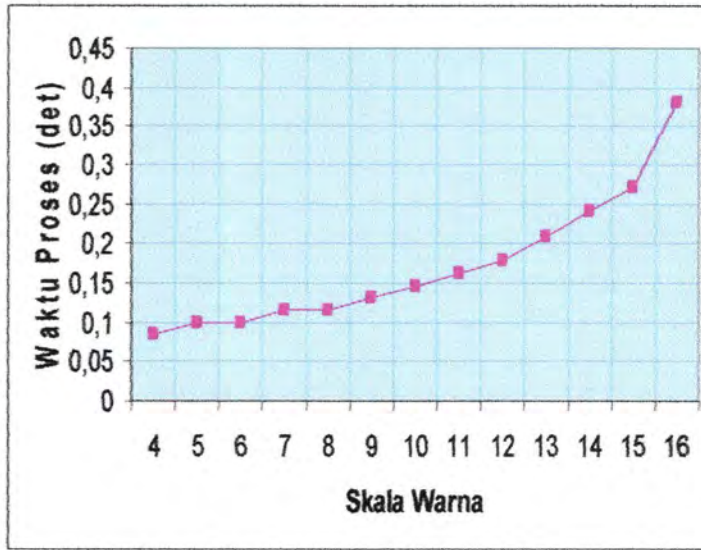
Pada bagian ini akan dilakukan uji coba pada proses identifikasi buah dengan mengetahui pengaruh besarnya skala warna dengan waktu proses

identifikasi. Pengambilan waktu proses ini dilakukan dengan menghitung selisih waktu dari proses perhitungan jarak dari histogram indeks gambar query dan histogram fitur dalam database fitur. Hasil percobaan dengan spesifikasi yang sudah dijelaskan di atas menunjukkan memang ada peningkatan seperti terlihat pada tabel 4.4

Tabel 4.4 Waktu proses untuk setiap skala warna dari 4 sampai dengan 16

SkalaWarna	Waktu Proses (Detik)
4	0.08438
5	0.10000
6	0.10000
7	0.11563
8	0.11563
9	0.13125
10	0.14688
11	0.16250
12	0.17813
13	0.20938
14	0.24063
15	0.27188
16	0.38175

Perkembangan waktu proses akibat dari penambahan jumlah skala warna dapat lebih jelas ditunjukkan oleh gambar 4.14. Terlihat pada gambar bahwa perkembangan waktu proses tidak berjalan secara linier.



Gambar 4.14 Pengaruh jumlah skala warna dan waktu proses pengenalan

Dengan memperhatikan grafik diatas, maka semakin jelas terlihat bahwa waktu proses sangat dipengaruhi oleh jumlah skala warna yang digunakan dalam proses ini.

BAB VI

SIMPULAN DAN SARAN

BAB 5

SIMPULAN DAN SARAN

5.1 Simpulan

Berdasarkan hasil implementasi yang dibagi menjadi dua tahap, training dan testing terhadap citra buah dengan menggunakan histogram indeks dan pada lingkungan pengujian sebagaimana telah dijelaskan pada bab sebelumnya, maka simpulan yang dapat diambil dari hasil pengujian pada tesis ini adalah bahwa:

1. Histogram indeks dapat digunakan untuk mengidentifikasi gambar buah dengan jumlah skala warna di atas atau sama dengan 8, yang berarti jauh lebih kecil dari jumlah skala pada objek gambar menyeluruh yang menggunakan 25 skala warna.
2. Jumlah skala warna pada histogram indeks berpengaruh secara non-linier pada lamanya waktu proses identifikasi.

5.2 Saran

Metode histogram indeks pada sistem identifikasi gambar buah yang merupakan salah satu bagian dari persoalan *content based image retrieval*, dapat juga diimplementasikan untuk obyek gambar yang lain.

Metode histogram indeks tidak memperdulikan indeks ke n adalah warna apa, karena metode ini hanya memberi simbol berupa indeks sesuai dari nilai pada setiap komponen warna dasarnya. Proses kuantisasi warna akan mempengaruhi nilai indeks yang dihasilkan.

DAFTAR PUSTAKA

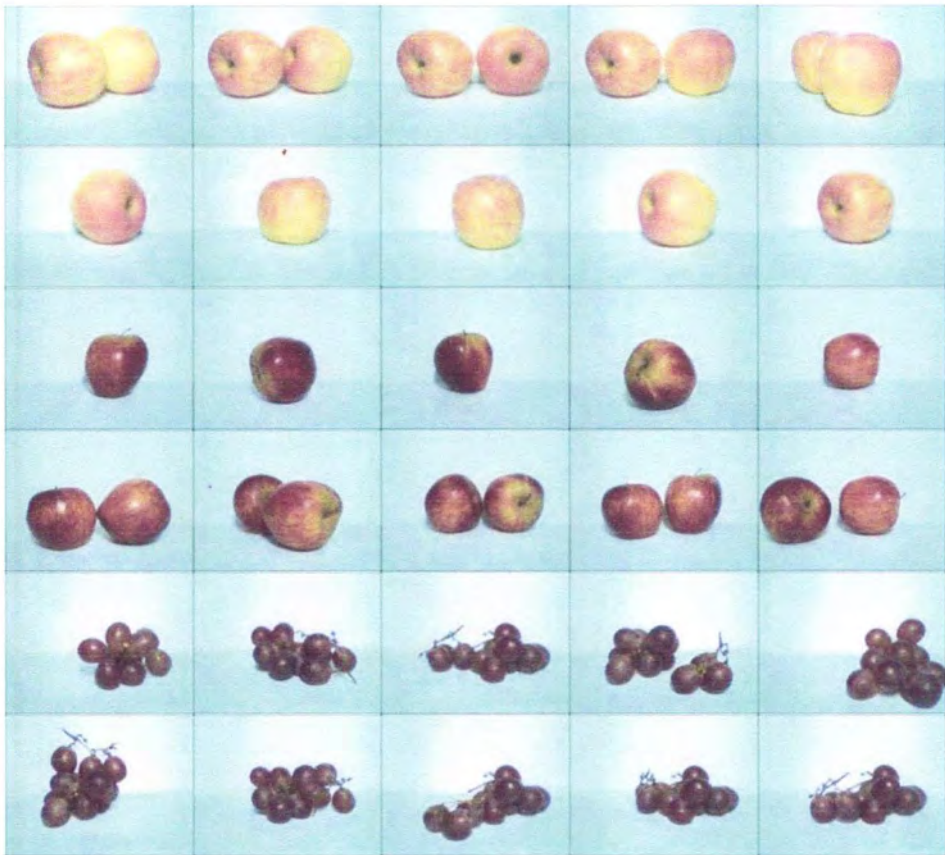
DAFTAR PUSTAKA

- [1] Nana Ramadijanti, "*Sistem Content Based Image Retrieval Menggunakan Filter Gabor*", *Tesis*, Jurusan Teknik Informatika, ITS, 2003
- [2] B.G. Prasad, S.K. Gupta, K.K. Biswas, "Color and Shape Index for Region-Based Image Retrieval", *IWVF4*, LNCS 2059, pp. 716-725, 2001. Springer-Verlag Berlin Heidelberg 2001
- [3] M. Soriano, L. Garcia and C. Saloma, "Fluorescent image classification by major color histograms and a neural network", 26 February 2001 / Vol. 8, No. 5 / OPTICS EXPRESS 271
- [4] Dwi Sulistyowati, "Pengenalan Buah Menggunakan Image Processing", *Tugas Akhir*, Politeknik Elektronika Negeri Surabaya, Surabaya, 2004.
- [5] K. C. Ravishankar, B. G. Prasad, S. K. Gupta, and K. K. Biswas, "Dominant Color Region Based Indexing For CBIR", *In proceeding of the International Conference on Image Analysis and Processing (ICIAP)*, pages 887—892, Venice, Italy, 1999.
- [6] Xiuqi Li, Shu-Ching Chen, Mei-Ling Shyu and Borko Furht, "Image Retrieval By Color, Texture, And Spatial Information", 2002.

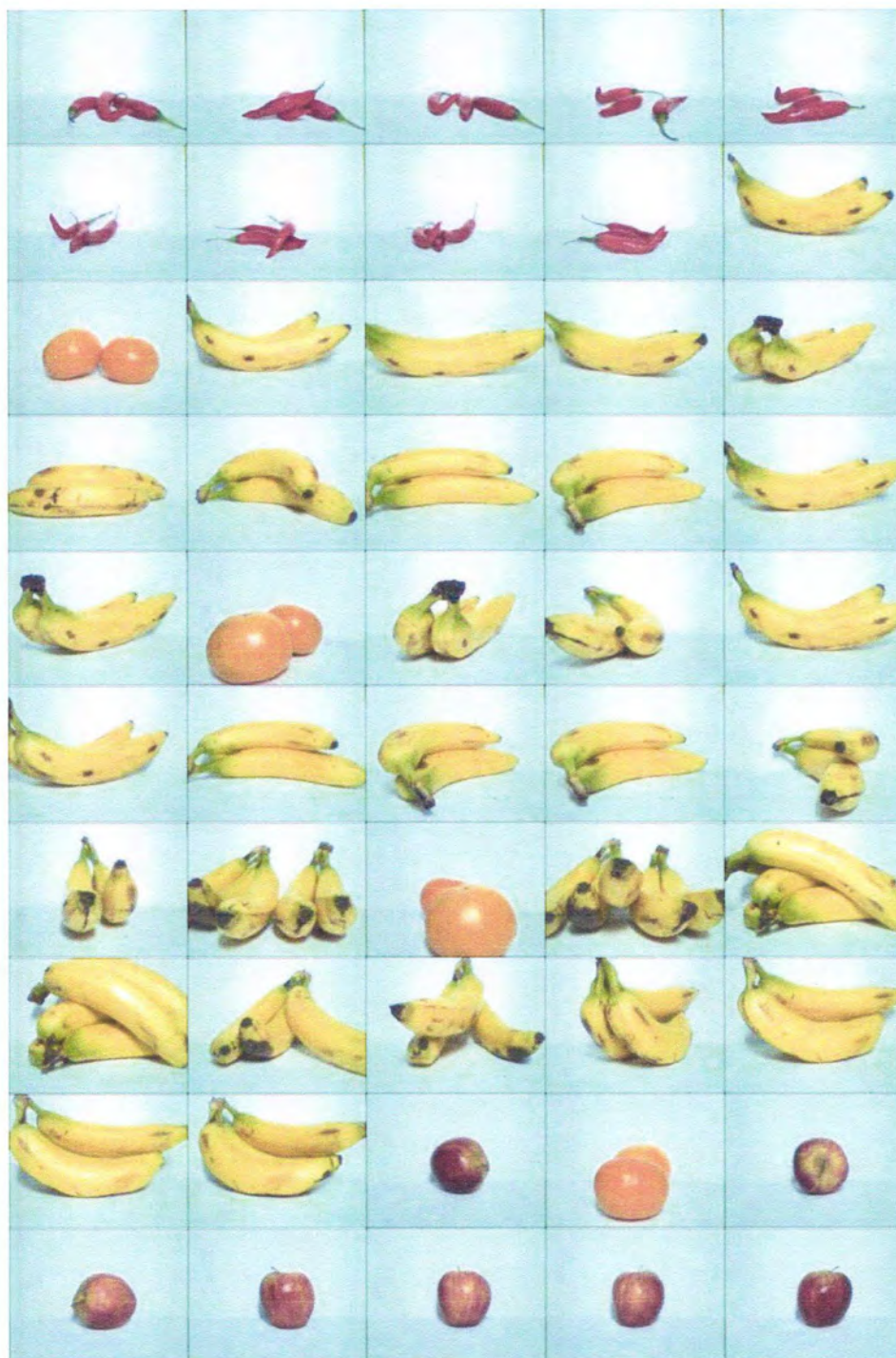
LAMPIRAN

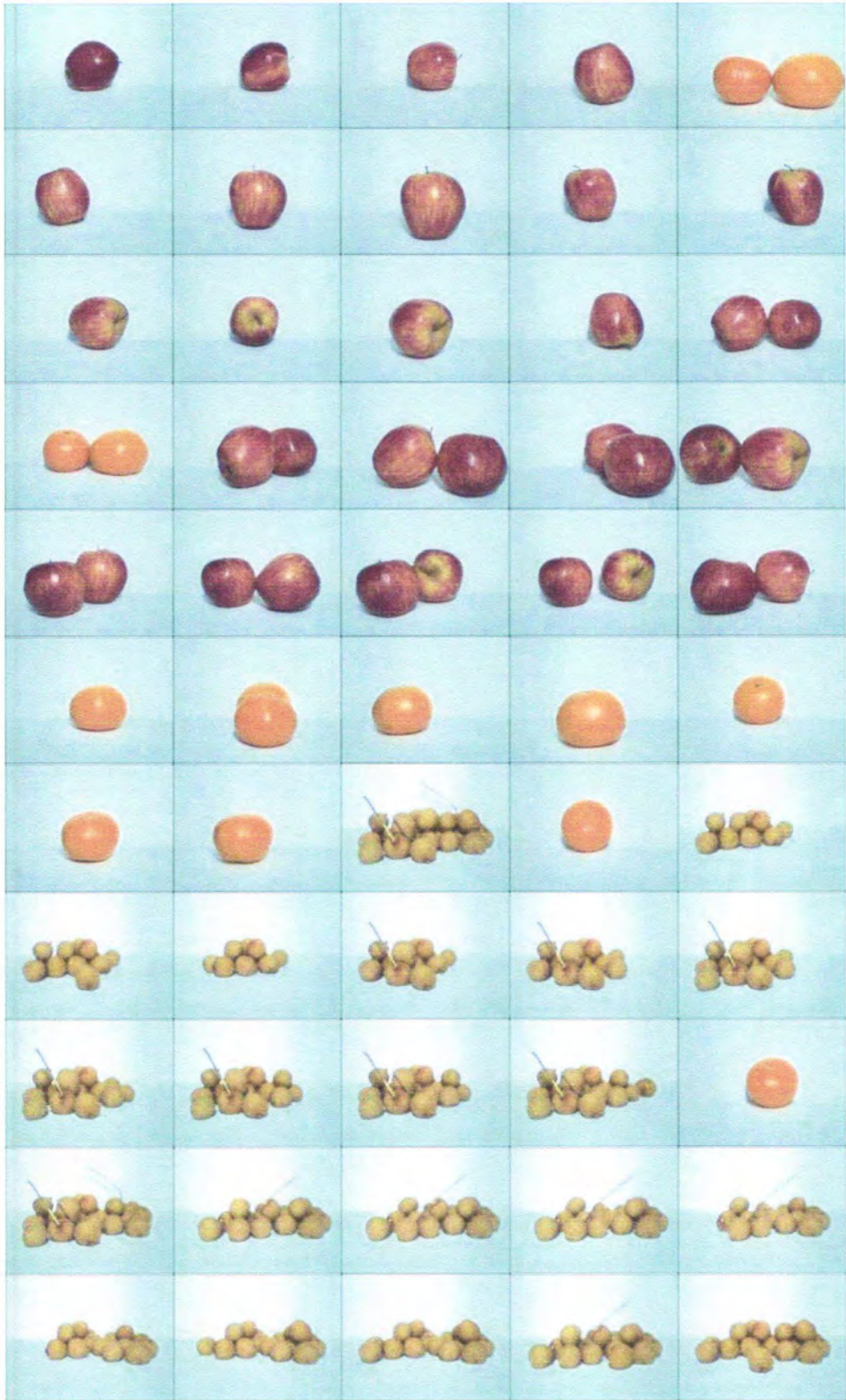
LAMPIRAN 1
GAMBAR BUAH TEST



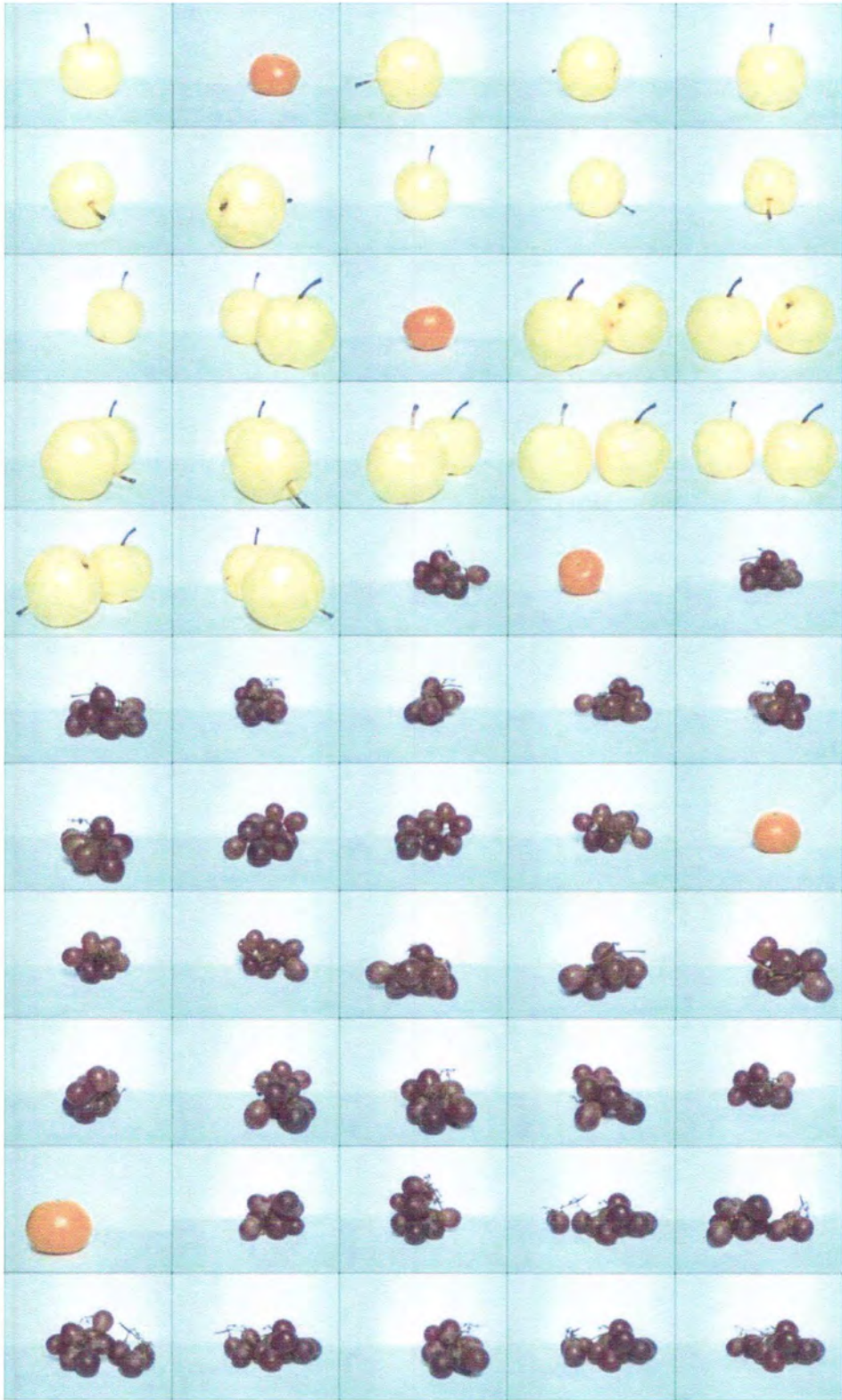


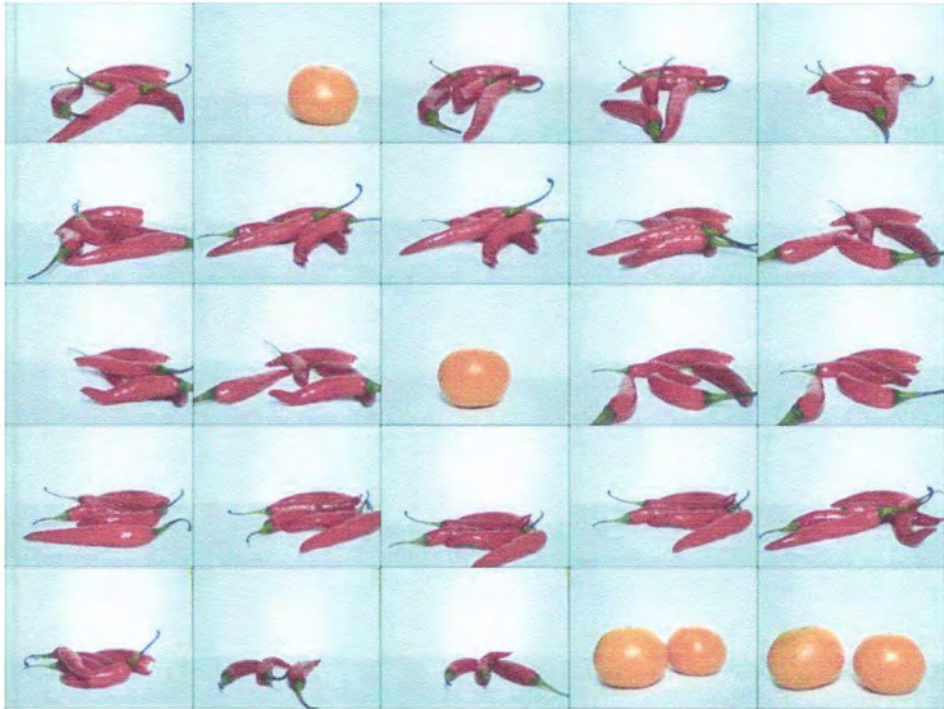
LAMPIRAN 2
GAMBAR BUAH UNTUK TRAINING











Handwritten text, likely bleed-through from the reverse side of the page. The characters are faint and difficult to decipher, but appear to be arranged in several lines.

Handwritten text, possibly a signature or a date, located in the lower-left corner of the page.

LAMPIRAN 3

LISTING PROGRAM

```
Dim N As Integer 'N adalah jumlah file gambar di dalam list box
Private Sub Combo1_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
    N = Combo1.ListIndex 'Saat combo1 di tekan nilai N berubah
End If
End Sub
Private Sub Combo1_LostFocus()
    N = Combo1.ListIndex 'Saat combo1 di tekan nilai N berubah
End Sub
Private Sub Command1_Click()
    Dim h(4096) As Single
    ni = Val(Text1) 'ni adalah jumlah skala warna
    s = Int(255 / (ni - 1)) 's adalah pembagi untuk mendapatkan skala warna ni

    'Proses inisialisa 0 pada nilai histogram
    For i = 0 To ni ^ 3
        h(i) = 0
    Next i

    'Pembacaan pixel by pixel
    For i = 0 To Picture1.ScaleWidth - 1
        For j = 0 To Picture1.ScaleHeight - 1
            ' membaca warna dan konversi ke RGB

            ' Threshold warna hitam
            If r0 < 40 And g0 < 40 And b0 < 40 Then r1 = 0: g1 = 0: b1 = 0
            Picture3.PSet (i, j), RGB(r1, g1, b1)
            r = Int(r1 / s)
            g = Int(g1 / s)
```



```

b = Int(b1 / s)

'Indek warna
m = ni ^ 2 * r + ni * g + b

'Histogram sebagai akumulator
h(m) = h(m) + 1
Next j
Next i

'Menggambar histogram dengan MSChart
MSChart1.RowCount = ni ^ 3 'Menjamin ukuran mschart sama dengan ukuran
data
MSChart1.Refresh
MSChart1.Row = 1
MSChart1.RowLabel = "0"
MSChart1.Data = 0
h(0) = 0
jh = 0
For i = 0 To ni ^ 3 - 1
    jh = jh + h(i)
Next i
For i = 1 To ni ^ 3 - 1
    h(i) = h(i) / jh ' Histogram yang dinormalisasi
    MSChart1.Row = i + 1
    MSChart1.RowLabel = Str(i)
    MSChart1.Data = h(i)
Next i
End Sub

Private Sub Command2_Click()
Dim h(5000) As Single

```

```

ni = Val(Text1)
s = Int(255 / (ni - 1))
For i = 0 To ni ^ 3 - 1
    h(i) = 0
Next i
For t = N * 30 + 1 To (N + 1) * 30
    namafile = App.Path & "\training\" & Trim(Str(t)) & ".jpg"
    Picture1 = LoadPicture(namafile)
    Picture1.Refresh
    For i = 0 To Picture1.ScaleWidth - 1
        For j = 0 To Picture1.ScaleHeight - 1
            w = Picture1.Point(i, j)
            r1 = w And RGB(255, 0, 0)
            g1 = Int((w And RGB(0, 255, 0)) / 256)
            b1 = Int(Int((w And RGB(0, 0, 255)) / 256) / 256)
            w = Picture2.Point(i, j)
            r2 = w And RGB(255, 0, 0)
            g2 = Int((w And RGB(0, 255, 0)) / 256)
            b2 = Int(Int((w And RGB(0, 0, 255)) / 256) / 256)
            r0 = Abs(r1 - r2)
            g0 = Abs(g1 - g2)
            b0 = Abs(b1 - b2)
            If r0 < 40 And g0 < 40 And b0 < 40 Then r1 = 0: g1 = 0: b1 = 0
            r = Int(r1 / s)
            g = Int(g1 / s)
            b = Int(b1 / s)
            m = ni ^ 2 * r + ni * g + b
            h(m) = h(m) + 1
        Next j
    Next i
Next t
h(0) = 0

```

```

jh = 0
For i = 0 To ni ^ 3 - 1
    jh = jh + h(i)
Next i
For i = 0 To ni ^ 3 - 1
    h(i) = h(i) / jh
Next i

If ni < 7 Then
MSChart2.RowCount = ni ^ 3
MSChart2.ColumnCount = 1
For i = 0 To ni ^ 3 - 1
    MSChart2.Row = i + 1
    MSChart2.RowLabel = Str(i)
    MSChart2.Data = h(i)
Next i
End If
End Sub

```

```

Private Sub Command3_Click()
    Dim h(4096) As Single

    ni = Val(Text1)
    s = Int(255 / (ni - 1))
    MSChart1.ColumnCount = 3
    MSChart1.RowCount = ni ^ 3
    MSChart1.Refresh
    MSChart1.chartType = VtChChartType2dLine

    For i = 0 To ni ^ 3 - 1
        h(i) = 0
    Next i

```



```

For t = N * 30 + 1 To (N + 1) * 30
    namafile = App.Path & "\training\" & Trim(Str(t)) & ".jpg"
    Picture1 = LoadPicture(namafile)
    Picture1.Refresh
    For i = 0 To Picture1.ScaleWidth - 1
        For j = 0 To Picture1.ScaleHeight - 1
            w = Picture1.Point(i, j)
            r1 = w And RGB(255, 0, 0)
            g1 = Int((w And RGB(0, 255, 0)) / 256)
            b1 = Int(Int((w And RGB(0, 0, 255)) / 256) / 256)
            w = Picture2.Point(i, j)
            r2 = w And RGB(255, 0, 0)
            g2 = Int((w And RGB(0, 255, 0)) / 256)
            b2 = Int(Int((w And RGB(0, 0, 255)) / 256) / 256)
            r0 = Abs(r1 - r2)
            g0 = Abs(g1 - g2)
            b0 = Abs(b1 - b2)
            If r0 < 32 And g0 < 32 And b0 < 32 Then r1 = 0: g1 = 0: b1 = 0
            r = Int(r1 / s)
            g = Int(g1 / s)
            b = Int(b1 / s)
            m = ni ^ 2 * r + ni * g + b
            h(m) = h(m) + 1
        Next j
    Next i
Next t
h(0) = 0
jh = 0
For i = 0 To ni ^ 3 - 1
    jh = jh + h(i)
Next i
For i = 0 To ni ^ 3 - 1

```

```

    h(i) = h(i) / jh
Next i
MSChart1.Row = 1
MSChart1.RowLabel = "0"
MSChart1.Column = 1
MSChart1.Data = 0
For i = 1 To ni ^ 3 - 1
    MSChart1.Row = i + 1
    MSChart1.Column = 1
    MSChart1.RowLabel = Str(i)
    MSChart1.Data = h(i)
Next i
t = 0
For k = 1 To 2
    t = t + 1 + Int(Rnd * 4)
    namafile = App.Path & "\training\" & Trim(Str(t)) & ".jpg"
    Picture1 = LoadPicture(namafile)
    Picture1.Refresh
    For i = 0 To ni ^ 3
        h(i) = 0
    Next i
    For i = 0 To Picture1.ScaleWidth - 1
        For j = 0 To Picture1.ScaleHeight - 1
            w = Picture1.Point(i, j)
            r1 = w And RGB(255, 0, 0)
            g1 = Int((w And RGB(0, 255, 0)) / 256)
            b1 = Int(Int((w And RGB(0, 0, 255)) / 256) / 256)
            w = Picture2.Point(i, j)
            r2 = w And RGB(255, 0, 0)
            g2 = Int((w And RGB(0, 255, 0)) / 256)
            b2 = Int(Int((w And RGB(0, 0, 255)) / 256) / 256)
            r0 = Abs(r1 - r2)

```

```

g0 = Abs(g1 - g2)
b0 = Abs(b1 - b2)
If r0 < 32 And g0 < 32 And b0 < 32 Then r1 = 0: g1 = 0: b1 = 0
Picture3.PSet (i, j), RGB(r1, g1, b1)
r = Int(r1 / s)
g = Int(g1 / s)
b = Int(b1 / s)
m = ni ^ 2 * r + ni * g + b
h(m) = h(m) + 1
Next j
Next i
h(0) = 0
jh = 0
For i = 0 To ni ^ 3 - 1
    jh = jh + h(i)
Next i
MSChart1.Row = 1
MSChart1.RowLabel = "0"
MSChart1.Column = k + 1
MSChart1.Data = 0
For i = 1 To ni ^ 3 - 1
    h(i) = h(i) / jh
    MSChart1.Row = i + 1
    MSChart1.Column = k + 1
    MSChart1.Data = h(i)
Next i
Next k
End Sub

Private Sub File1_Click()
namafile = App.Path & "\test\" & File1.List(File1.ListIndex)
Picture1 = LoadPicture(namafile)

```

```
Picture1.Refresh
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
File1.Path = App.Path & "\test"
```

```
File1.FileName = "*.jpg"
```

```
End Sub
```

```
Dim N As Integer
```

```
Private Sub Combo1_Click()
```

```
N = Combo1.ListIndex
```

```
End Sub
```

```
Private Sub Command1_Click()
```

```
Dim h(5000) As Single
```

```
ni = Val(Text1)
```

```
s = Int(255 / (ni - 1))
```

```
For i = 0 To ni ^ 3 - 1
```

```
    h(i) = 0
```

```
Next i
```

```
For t = N * 30 + 1 To (N + 1) * 30
```

```
    namafile = App.Path & "\training\" & Trim(Str(t)) & ".jpg"
```

```
    Picture1 = LoadPicture(namafile)
```

```
    Picture1.Refresh
```

```
    For i = 0 To Picture1.ScaleWidth - 1
```

```
        For j = 0 To Picture1.ScaleHeight - 1
```

```
            w = Picture1.Point(i, j)
```

```
            r1 = w And RGB(255, 0, 0)
```

```
            g1 = Int((w And RGB(0, 255, 0)) / 256)
```

```
            b1 = Int(Int((w And RGB(0, 0, 255)) / 256) / 256)
```

```
            w = Picture2.Point(i, j)
```

```
            r2 = w And RGB(255, 0, 0)
```

```

g2 = Int((w And RGB(0, 255, 0)) / 256)
b2 = Int(Int((w And RGB(0, 0, 255)) / 256) / 256)
r0 = Abs(r1 - r2)
g0 = Abs(g1 - g2)
b0 = Abs(b1 - b2)
If r0 < 32 And g0 < 32 And b0 < 32 Then r1 = 0: g1 = 0: b1 = 0
r = Int(r1 / s)
g = Int(g1 / s)
b = Int(b1 / s)
m = ni ^ 2 * r + ni * g + b
h(m) = h(m) + 1

```

```

Next j

```

```

Next i

```

```

Next t

```

```

h(0) = 0

```

```

jh = 0

```

```

For i = 0 To ni ^ 3 - 1

```

```

    jh = jh + h(i)

```

```

Next i

```

```

For i = 0 To ni ^ 3 - 1

```

```

    h(i) = h(i) / jh

```

```

Next i

```

```

If ni < 7 Then

```

```

    MSChart1.RowCount = ni ^ 3

```

```

    MSChart1.ColumnCount = 1

```

```

    For i = 0 To ni ^ 3 - 1

```

```

        MSChart1.Fow = i + 1

```

```

        MSChart1.FowLabel = Str(i)

```

```

        MSChart1.Data = h(i)

```

```

    Next i

```

```

End If

```

```

End Sub

```

```

Private Sub Command2_Click()
Dim h(5000) As Single
ni = Val(Text1)
s = Int(255 / (ni - 1))
Open App.Path & "\fitur" & Trim(Str(ni)) & ".txt" For Output As #1
For k = 0 To 7
For i = 0 To ni ^ 3 - 1
    h(i) = 0
Next i
For t = k * 30 + 1 To (k + 1) * 30
    namafile = App.Path & "\training\" & Trim(Str(t)) & ".jpg"
    Picture1 = LoadPicture(namafile)
    Picture1.Refresh
    For i = 0 To Picture1.ScaleWidth - 1
        For j = 0 To Picture1.ScaleHeight - 1
            w = Picture1.Point(i, j)
            r1 = w And RGB(255, 0, 0)
            g1 = Int((w And RGB(0, 255, 0)) / 256)
            b1 = Int(Int((w And RGB(0, 0, 255)) / 256) / 256)
            w = Picture2.Point(i, j)
            r2 = w And RGB(255, 0, 0)
            g2 = Int((w And RGB(0, 255, 0)) / 256)
            b2 = Int(Int((w And RGB(0, 0, 255)) / 256) / 256)
            r0 = Abs(r1 - r2)
            g0 = Abs(g1 - g2)
            b0 = Abs(b1 - b2)
            If r0 < 32 And g0 < 32 And b0 < 32 Then r1 = 0: g1 = 0: b1 = 0
            r = Int(r1 / s)
            g = Int(g1 / s)
            b = Int(b1 / s)
            m = ni ^ 2 * r + ni * g + b

```

```

        h(n) = h(m) + 1
    Next j
Next i
Next t
h(0) = 0
jh = 0
For i = 0 To ni ^ 3 - 1
    jh = jh + h(i)
Next i
Write #i, Trim(Combo1.List(k))
For i = 0 To ni ^ 3 - 1
    h(i) = h(i) / jh
    Write #1, h(i)
Next i
Next k
Close #1
End Sub

```

```

Private Sub Command3_Click()
Dim h(5000) As Single
ni = Val(Text1)
If ni < 7 Then
Open App.Path & "\fitur" & Trim(Str(ni)) & ".txt" For Input As #1
MSChart1.ColumnCount = 6
MSChart1.RowCount = ni ^ 3
For k = 0 To 5
Input #1, namabuah
List1.AddItem namabuah
For i = 0 To ni ^ 3 - 1
Input #1, h(i)
List1.AddItem Str(h(i))
MSChart1.Row = i + 1

```

```

MSChart1.RowLabel = i
MSChart1.Column = k + 1
MSChart1.Data = h(i)
Next i
Next k
Close #1
End If
End Sub

```

```

Private Sub Form_Load()
N = 0
End Sub

```

```

Private Sub Command1_Click()
FormLearning.Show
End Sub

```

```

Private Sub Command2_Click()
FormTest.Show
End Sub

```

```

Private Sub Command3_Click()
Form1.Show
End Sub

```

```

Dim cbuah(20) As String
Private Sub Command1_Click()
Dim ramabuah(20) As String
Dim hs(20, 5000), h(5000) As Single
ni = Val(Text1)
s = Int(255 / (ni - 1))
Open App.Path & "\fitur" & Trim(Str(ni)) & ".txt" For Input As #1

```



```

For k = 0 To 7
    Input #1, namabuah(k)
    For i = 0 To ni ^ 3 - 1
        Input #1, hs(k, i)
    Next i
Next k
Close #1

```

```
waktu1 = Timer
```

```

For i = 0 To Picture1.ScaleWidth - 1
    For j = 0 To Picture1.ScaleHeight - 1
        w = Picture1.Point(i, j)
        r1 = w And RGB(255, 0, 0)
        g1 = Int((w And RGB(0, 255, 0)) / 256)
        b1 = Int(Int((w And RGB(0, 0, 255)) / 256) / 256)
        w = Picture2.Point(i, j)
        r2 = w And RGB(255, 0, 0)
        g2 = Int((w And RGB(0, 255, 0)) / 256)
        b2 = Int(Int((w And RGB(0, 0, 255)) / 256) / 256)
        r0 = Abs(r1 - r2)
        g0 = Abs(g1 - g2)
        b0 = Abs(b1 - b2)
        If r0 < 32 And g0 < 32 And b0 < 32 Then r1 = 0: g1 = 0: b1 = 0
        r = Int(r1 / s)
        g = Int(g1 / s)
        b = Int(b1 / s)
        m = ni ^ 2 * r + ni * g + b
        h(m) = h(m) + 1
    Next j
Next i

```

```

Next i
h(0) = 0
jh = 0

```

```
For i = 0 To ni ^ 3 - 1
```

```
    jh = jh + h(i)
```

```
Next i
```

```
For i = 0 To ni ^ 3 - 1
```

```
    h(i) = h(i) / jh
```

```
Next i
```

```
List1.Clear
```

```
emin = 1000
```

```
m = 0
```

```
For k = 0 To 7
```

```
    e = 0
```

```
    For i = 0 To ni ^ 3 - 1
```

```
        e = e + (h(i) - hs(k, i)) ^ 2
```

```
    Next i
```

```
    e = 100 * (e / ni ^ 3) ^ 0.5
```

```
    If e < emin Then
```

```
        emin = e
```

```
        m = k
```

```
    End If
```

```
    kal = Trim(namabuah(k)) & " --> error = " & Format(e, "#.###") & "%"
```

```
    List1.AddItem kal
```

```
Next k
```

```
kal := namabuah(m) & " (" & Format(emin, "#.###") & "%)"
```

```
Label1 = kal
```

```
MSChart1.RowCount = ni ^ 3
```

```
For i = 0 To ni ^ 3 - 1
```

```
    MSChart1.Row = i + 1
```

```
    MSChart1.RowLabel = Str(i)
```

```
    MSChart1.Column = 1
```

```
    MSChart1.Data = h(i)
```

```

namafile = App.Path & "\test\" & File1.List(nt)
Picture1 = LoadPicture(namafile)
Picture1.Refresh
For i = 0 To Picture1.ScaleWidth - 1
    For j = 0 To Picture1.ScaleHeight - 1
        w = Picture1.Point(i, j)
        r1 = w And RGB(255, 0, 0)
        g1 = Int((w And RGB(0, 255, 0)) / 256)
        b1 = Int(Int((w And RGB(0, 0, 255)) / 256) / 256)
        w = Picture2.Point(i, j)
        r2 = w And RGB(255, 0, 0)
        g2 = Int((w And RGB(0, 255, 0)) / 256)
        b2 = Int(Int((w And RGB(0, 0, 255)) / 256) / 256)
        r0 = Abs(r1 - r2)
        g0 = Abs(g1 - g2)
        b0 = Abs(b1 - b2)
        If r0 < 32 And g0 < 32 And b0 < 32 Then r1 = 0: g1 = 0: b1 = 0
        r = Int(r1 / s)
        g = Int(g1 / s)
        b = Int(b1 / s)
        m = ni ^ 2 * r + ni * g + b
        h(m) = h(m) + 1
    Next j
Next i
Next i
h(0) = 0
jh = 0
For i = 0 To ni ^ 3 - 1
    jh = jh + h(i)
Next i
For i = 0 To ni ^ 3 - 1
    h(i) = h(i) / jh
Next i

```

```

MSChart1.Column = 2
MSChart1.Data = hs(m, i)
Next i
waktu2 = Timer
waktu = waktu2 - waktu1 - 0.4
Label3 = "Waktu = " & Format(waktu, "#.#####") & " detik"

```

```
End Sub
```

```

Private Sub Command2_Click()
Dim namabuah(20) As String
Dim hs(20, 5000), h(5000) As Single
ni = Val(Text1)
s = Int(255 / (ni - 1))
Open App.Path & "\fitur" & Trim(Str(ni)) & ".txt" For Input As #1
For k = 0 To 7
    Input #1, namabuah(k)
    For i = 0 To ni ^ 3 - 1
        Input #1, hs(k, i)
    Next i
Next k
Close #1
Open App.Path & "\hasil" & Trim(Str(ni)) & ".html" For Output As #1
nk = File1.ListCount
List1.Clear
benar = 0
salah = 0
Print #1, "<html><body>"
Print #1, "<table border=1>"
Print #1, "          #1,          "<tr><td>Nama          file</td><td>Dikenali
Sebagai</td><td>Error(%</td><td>Keputusan</td></tr>"
For nt = 0 To nk - 1

```

```

    kal = kal & "</td><td align=right>" & Format(emin, "#.###") & "%</td>"
    kal = kal & "<td>SALAH</td></tr>"
    salah = salah + 1
End If
Print #1, kal.
Next nt
hasil = 100 * benar / (benar + salah)
Print #1, "</table><br>"
Print #1, " Tingkat akurasi pengenalan buah untuk skala warna " & Str(ni) & "
adalah"
Print #1, Str(benar) & " dari" & Str(benar + salah)
Print #1, " buah atau " & Format(hasil, "#.###") & "%<br>"
Print #1, "</body></html>"
Close #1
End Sub

```

```

Private Sub File1_Click()
    namafile = App.Path & "\test\" & File1.List(File1.ListIndex)
    Picture1 = LoadPicture(namafile)
    Picture1.Refresh
End Sub

```

```

Private Sub Form_Load()
    File1.Path = App.Path & "\test"
    File1.FileName = "*.jpg"
    cbuah(0) = "ANGGUR"
    cbuah(1) = "APEL MERAH"
    cbuah(2) = "APEL HIJAU"
    cbuah(3) = "JERUK"
    cbuah(4) = "KLENGKENG"
    cbuah(5) = "LOMBOK"
    cbuah(6) = "PEAR"

```

```

emin = 1000
m = 0
For k = 0 To 7
    e = 0
    For i = 0 To ni ^ 3 - 1
        e = e + (h(i) - hs(k, i)) ^ 2
    Next i
    e = 100 * (e / ni ^ 3) ^ 0.5
    If e < emin Then
        emin = e
        m = k
    End If
Next k
ntu = Int(nt / 10)
nm = File1.List(nt)
kal = nm & " >> dikenali sbi: " & namabuah(m) & " (" & Format(emin, "#.##") &
"%)"
List1.AddItem kal
List1.Refresh
namabuah(m) = Trim(namabuah(m))
If namabuah(m) = cbuah(ntu) Then
    kal = "<t><td align=center></td><td align=center>" & Trim(namabuah(m))
    'kal = kal & "(" & cbuah(ntu) & Str(ntu) & ")"
    kal = kal & "</td><td align=right>" & Format(emin, "#.###") & "%</td>"
    kal = kal & "<td>BENAR</td></tr>"
    benar = benar + 1
Else
    kal = "<tr bgcolor=yellow><td align=center></td><td align=center>" &
Trim(namabuah(m))
    'kal = kal & "(" & cbuah(ntu) & Str(ntu) & ")"

```



```
cbuah(7) = "PISANG"
```

```
End Sub
```