

**PENERAPAN METODE *CONTENT BASED IMAGE
RETRIEVAL* (CBIR) DENGAN ALGORITMA
SHINGLING PADA APLIKASI
KEMIRIPAN GAMBAR**

SKRIPSI

**RIKA AMALIA
0701162040**



**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
MEDAN
2021**

**PENERAPAN METODE *CONTENT BASED IMAGE
RETRIEVAL* (CBIR) DENGAN ALGORITMA
SHINGLING PADA APLIKASI
KEMIRIPAN GAMBAR**

SKRIPSI

Diajukan untuk Memenuhi Syarat Mencapai Gelar Sarjana Komputer

**RIKA AMALIA
0701162040**



**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
MEDAN
2021**

PERSETUJUAN

Hal : Surat Persetujuan

Lamp :-

Kepada Yth,
Dekan Fakultas Sains dan Teknologi
Universitas Islam Negeri Sumatera Utara Medan

Assalamualaikum Waramatullahi Wabarakatuh.

Setelah membaca, meneliti, memberikan petunjuk, dan mengoreksi serta mengadakan perbaikan, kami selaku pembimbing berpendapat skripsi saudara.

Nama : Rika Amalia

NIM : 0701162040

Program Studi : Ilmu Komputer

Judul : Penerapan Metode *Content Based Image Retrieval* (CBIR)
Dengan Algoritma *Shingling* Pada Aplikasi Kemiripan
Gambar

Dapat disetujui untuk segera di *munaqasyahkan*. Atas perhatiannya kami ucapkan terima kasih.

Medan, 31 Maret 2021 M

17 Sya'ban 1442 H

Komisi Pembimbing,

Pembimbing I,



Dr. Mhd. Furqan, S.Si, M.Comp.Sc.
NIP. 198008062006041003

Pembimbing II,



Yusuf Ramadhan Nasution, M.Kom.
NIB. 1100000075



KEMENTERIAN AGAMA REPUBLIK INDONESIA
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA MEDAN
FAKULTAS SAINS DAN TEKNOLOGI
Jl. IAIN No. 1 Medan 20235
Telp. (061) 6615683-6622925, Fax. (061) 6615683
Url: <http://saintek.uinsu.ac.id>, E-mail: saintek@uinsu.ac.id

PENGESAHAN SKRIPSI

Nomor : B.99/ST/ST.V.2/PP.01.1/05/2021

Judul : Penerapan Metode Content Based Image Retrieval (CBIR)
Dengan Algoritma *Shingling* Pada Aplikasi Kemiripan Gambar

Nama : Rika Amalia
Nomor Induk Mahasiswa : 0701162040
Program Studi : Ilmu Komputer
Fakultas : Sains dan Teknologi

Telah dipertahankan di hadapan Dewan Penguji Skripsi Program Studi Ilmu Komputer
Fakultas Sains dan Teknologi UIN Sumatera Utara Medan dan dinyatakan **LULUS**.

Pada hari/tanggal : Selasa, 31 Maret 2021
Tempat : Via Zoom Meeting

Tim Ujian Munaqasyah,
Ketua,

Ilka Zufria, M.Kom
NIP. 198506042015031006

Dewan Penguji,

Penguji I,

Rakhmat Kurniawan R, S.T, M.Kom
NIP. 198503162015031003

Penguji II,

Abdul Halim Haugian, M.Kom
NIB. 1100000113

Penguji III,

Dr. Mhd. Furqan, S.Si, M.Comp.Sc.
NIP. 198008062006041003

Penguji IV,

Yusuf Ramadhan Nasution, M.Kom.
NIB. 1100000075

Mengesahkan,
Dekan Fakultas Sains dan Teknologi
UIN Sumatera Utara Medan,

Dr. Mhd.Syahnan,M.A
NIP: 196609051991031002

SURAT PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan dibawah ini :

Nama : Rika Amalia

NIM : 0701162040

Program Studi : Ilmu Komputer

Judul : Penerapan Metode *Content Based Image Retrieval* (CBIR) dengan Algoritma *Shingling* Pada Aplikasi Kemiripan Gambar

Menyatakan bahwa skripsi ini hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing – masing disebutkan sumbernya. Apabila dikemudian hari ditemukan plagiat dalam skripsi ini maka saya bersedia menerima sanksi pencabutan gelar akademik yang saya peroleh dan sanksi lainnya sesuai dengan peraturan yang berlaku.

Medan, 25 Maret 2021



Rika Amalia

NIM. 0701162040

ABSTRAK

Content Based Image Retrieval (CBIR) adalah salah satu aplikasi pengolahan citra yang dapat membantu pengguna menemukan suatu citra dari database citra secara cepat dengan menggunakan citra sebagai query atau gambar A dan gambar B. Dalam CBIR, fitur-fitur visual citra dalam database diekstraksi dan disimpan dalam vektor yang disebut vektor fitur. Fitur citra query juga diekstraksi dan diukur jaraknya dengan vektor fitur yang tersimpan di dalam database. Citra hasil temu kembali diurutkan berdasarkan jarak vektor fitur dari terkecil hingga terbesar. Warna dan tekstur adalah dua fitur penting dari sebuah citra. Dalam penelitian ini, fitur warna diekstraksi dengan metode Colour Histogram, Colour Histogram menunjukkan probabilitas suatu intensitas warna. Model warna citra yang digunakan adalah RGB, Berdasarkan hasil pengujian yang dilakukan oleh pengguna yang masing-masing menggunakan 2 citra query, dari masing-masing 2 citra tersebut kemudian dipadukan dengan algoritma shingling, dengan algoritma ini gambar dari kedua citra bitmap tersebut dibangkitkan nilai 64 bit (8 byte)

Kata kunci : *Content Based Image Retrieval, Citra, Colour Histogram, Grey Level Co-Occurrence Matrices, Shingling*

ABSTRACT

Content Based Image Retrieval (CBIR) is an image processing application that can help users find an image from an image database quickly by using an image as a query or image A and image B. In CBIR, the visual features of the image in the database are extracted and stored in a vector called a feature vector. The query image feature is also extracted and its distance is measured with the feature vector stored in the database. The retrieved images are sorted based on the distance of the feature vector from the smallest to the largest. Color and texture are two important features of an image. In this study, color features are extracted using the Color Histogram method. The image color model used is RGB. Based on the results of tests carried out by users, each of which uses 2 query images, from each of the 2 images then combined with the shingling algorithm, with this algorithm the images from the two bitmap images are generated 64 bit (8 bytes) values.

Keyword :*Content Based Image Retrieval, Citra, Colour Histogram, Grey Level Co-Occurrence Matrices, Shingling*

KATA PENGANTAR



Puji dan syukur penulis ucapkan kepada Allah SWT yang telah memberikan nikmat berupa kesehatan, kesempatan kepada penulis sehingga mampu menyelesaikan skripsi ini.

Maka sehubungan dengan skripsi ini, segala bentuk hasil dituliskan kedalam bentuk skripsi ini sebagai persyaratan dalam menyelesaikan mata kuliah skripsi di Program Studi Ilmu Komputer. Tujuan utama dari penulisan skripsi ini adalah untuk memantapkan teori dan praktek yang telah dipelajari di kampus dan dapat diselesaikan dengan serta diaplikasikan.

1. Teristimewa penulis ucapkan terima kasih kepada kedua orang tua penulis yaitu Bapak Budi Hartono dan Ibu Sri Murni yang sangat penulis cintai telah memberikan bantuan moril maupun materil, semangat dan doa yang begitu besar kepada penulis.
2. Bapak Prof. Dr. Syahrin Harahap, M.A, Selaku Rektor Universitas Islam Negeri Sumatera Utara Medan yang telah memberikan kesempatan kepada penulis untuk pelaksanaan kegiatan penyusunan tugas akhir khusus tahun akademik 2020/2021.
3. Bapak Dr. Mhd Syahnan, M.A, selaku Dekan Fakultas Sains dan Teknologi.
4. Bapak Ilka Zufria, M.Kom, selaku Ketua Program Studi Ilmu Komputer
5. Bapak Dr. Mhd. Furqan, S.Si., M.Comp.Sc, selaku Dosen Pembimbing I yang telah banyak memberi gagasan serta pemahaman kepada penulis selama proses awal sampai akhir skripsi ini rampung.
6. Bapak Yusuf Ramadhan Nasution, M.Kom selaku Dosen Pembimbing II penulis yang telah berkontribusi membantu dalam memberikan ide, saran, kritik, dan bimbingannya kepada penulis selama mengerjakan skripsi ini.
7. Bapak Rakhmat Kurniawan R, S.T, M.Kom, selaku Dosen Penguji I.

8. Bapak Abdul Halim Hasugian, M.Kom, Selaku Dosen Penguji II penulis.
9. Kepada diri saya sendiri yang sangat kuat dan tangguh untuk bisa mencapai dan melalui semua proses selama perkuliahan hingga akhir mencapai gelar sarjana.
10. Kepada Anisa Alfu Laila Selaku sahabat yang selalu memberikan semangat, dukungan dan relasi kepada penulis.
11. Kepada Muhammad Iqbal Mutakin, SE, terima kasih telah memberikan masukan dan motivasi untuk penulis.
12. Kepada Adi Mawardi, S.Kom, terima kasih telah memberikan dukungan dan informasi,
13. Kepada Fauzan Rochari Santoso, Rahma Aulia Hasanah, Nabila Melinda, Alvina Damayanti Selaku adik penulis yang juga ikut membantu dan memberikan semangat kepada penulis.
14. Semua pihak yang telah membantu penulis namun tidak dapat disebutkan satu persatu.

Penulis menyadari bahwa Skripsi ini masih jauh dari kata sempurna untuk itu penulis mengharapkan saran perbaikan dari para penguji, pembimbing dan pembaca demi kesempurnaan dari skripsi ini, Demikian penyusunan skripsi ini di ditulis, semoga skripsi ini dapat bermanfaat bagi pembaca dan dapat digunakan dengan semestinya.

Medan, Maret 2021

Penulis,



Rika Amalia

NIM. 0701162040

DAFTAR ISI

	Halaman
ABSTRAK	i
ABSTRACT	ii
KATA PENGANTAR.....	iii
DAFTAR ISI	v
DAFTAR GAMBAR.....	viii
DAFTAR TABEL	x
DAFTAR LAMPIRAN	xi
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah	2
1.4. Tujuan Penelitian	3
1.5. Manfaat Penelitian	3
BAB II TINJAUAN PUSTAKA.....	4
2.1 Pengenalan Citra	4
2.2 Citra Digital.....	5
2.2.1. Jenis Citra Digital	6
2.2.2. Format File Citra Digital.....	8
2.2.3. Elemen-Elemen Citra Digital.....	10

2.2.4. Warna.....	12
2.2.4.1. Model RGB (<i>Read, Grean, Blue</i>	13
2.2.4.2. Model HSV (<i>Hue, Saturation, and Value</i>)	14
2.2.4.3. Model HSL (<i>Hue, saturation , and Lightness</i>).....	15
2.2.4.4. Model CMYK (Cyan Magenta Yellow Key)	15
2.3. Perbaikan Kualitas Citra (<i>Image Enhancement</i>).....	16
2.4. <i>Content Based Image Retrieval</i> (CBIR).....	17
2.5. Metode Pengukuran Kemiripan (<i>Similarity Meassure</i>)	23
2.6. Algoritma	26
2.6.1. Definisi Analisis Algoritma	27
2.6.2. Algoritma Shingling.....	27
2.7. <i>Visual Basic .Net</i>	29
2.8. <i>MySQL</i>	30
2.9. Mengenal XAMPP	31
2.10. <i>Flow Chart</i>	32
BAB III METODOLOGI PENELITIAN	32
3.1. Waktu dan Tempat Penelitian	32
3.2. Alat dan Bahan Penelitian.....	32
3.3. Prosedur Penelitian.....	32
3.4. Desain Sistem.....	36
3.5 Perancangan Antarmuka	48

BAB IV HASIL DAN PEMBAHASAN	43
4.1. Tampilan Hasil	43
4.1.1 Perhitungan Manual	43
4.1.2 Tampilan Hasil Antarmuka Pemakai	53
4.1.3 Pencocokan Citra Untuk Menentukan Kematangan	58
4.1.4 Kelebihan Dan Kekurangan Aplikasi yang dirancang	58
BAB V KESIMPULAN DAN SARAN	59
5.1. Kesimpulan	59
5.2. Saran.....	59
DAFTAR PUSTAKA	60

DAFTAR GAMBAR

No. Judul	Halaman
2.1 Citra Biner.....	7
2.2 Citra Grayscale.....	7
2.3 Citra Warna.....	8
2.4 Gambar Citra .bmp.....	9
2.5 Contoh Citra JPEG.....	10
2.6 Struktur Warna RGB.....	11
2.7 Model Warna HSV.....	14
2.8 Diagram arsitektur CBIR.....	21
2.9 Struktur Hubungan dan Jenis Algoritma.....	27
3.1 Flowchart Prosedur Penelitian.....	35
3.2 Flowchart halaman utama.....	36
3.3 Flowchart Masuk ke Halaman <i>Login</i>	36
3.4 Flowchart Memilih Gambar.....	37
3.5 Flowchart Halaman Menghitung Kemiripan Gambar.....	38
3.6 Form Judul Perangkat Lunak.....	39
3.7 Form <i>Login</i> Perangkat Lunak.....	39
3.8 Form Perbandingan Kemiripan Gambar.....	40

3.9 Form Memilih Gambar	40
3.10 Form Penguji Kemiripan Gambar	41
3.11 Pengolahan CBIR.....	42
4.1 Gambar Pembentukan Single Rika.bmp	47
4.2 Citra Pemilihan Single Rika.....	50
4.3 Citra Pembentukan	50
4.4 Flowchart Proses Kerja	53
4.5 Tampilan Flowchart Proses <i>Login</i>	54
4.6 Tampilan Flowchart Proses Memasukan Gambar	54
4.7 Tampilan Flowchart Proses Hitung Kemiripan Citra	55
4.8 Tampilan Halaman Utama Kemiripan Gambar	55
4.9 Tampilan Halaman <i>Login</i>	56
4.10 Tampilan Halaman Kemiripan Gambar	56
4.11 Tampilan Halaman Hitung Kemiripan Gambar.....	57

DAFTAR TABEL

No. Judul	Halaman
2.1 Kode Warna RGB	13
2.2 Flowchart	32
4.1 Tabel Kemiripan Gambar.....	57

DAFTAR LAMPIRAN

Lampiran	Judul Lampiran
1.	Listing Program
2.	Uji Coba Tingkat Keberhasilan
3.	Daftar Riwayat Hidup
4.	Karu Bimbingan Skripsi

BAB I

PENDAHULUAN

1.1 Latar Belakang

Gambar adalah suatu representasi, kemiripan, atau imitasi dari suatu objek. Gambar mengandung informasi tentang objek yang direpresentasikan. Sehingga gambar mampu memberikan informasi yang lebih banyak dibandingkan data teks. Untuk merepresentasikan objek lebih akurat dilakukan pengolahan citra. Pengolahan citra merupakan proses memanipulasi dan menganalisis gambar menggunakan bantuan komputer yang bertujuan untuk memperbaiki, mengekstrak informasi, dan menambah kualitas gambar. Secara umum operasi pengolahan citra dapat diklasifikasikan sebagai perbaikan kualitas gambar, restorasi gambar, pemampatan gambar, segmentasi gambar, pengerokan gambar, dan rekonstruksi gambar.

وَمَنْ أَظْلَمُ مِمَّنْ ذَهَبَ يَخْلُقُ كَخَلْقِي فَالْيَخْلُقُوا ذَرَّةً، فَالْيَخْلُقُوا شُعَيْرَةً

Artinya : “Siapakah yang lebih zalim daripada orang yang sengaja menciptakan (sesuatu) seperti ciptaan-Ku, maka hendaklah mereka menciptakan semut kecil (jika mampu), dan hendaklah mereka menciptakan sebiji gandum. (Firman Allah SWT.)

Content Based Image Retrieval (CBIR) merupakan salah satu bentuk implementasi dari pengolahan citra. CBIR sendiri merupakan teknik pencarian kembali gambar yang mempunyai kemiripan karakteristik atau content dari sekumpulan gambar. Secara umum, proses dalam CBIR dilakukan dengan proses ekstraksi fitur. Parameter fitur yang dapat digunakan untuk retrieval dapat berupa histogram, susunan warna, tekstur, bentuk, tipe spesifik dari objek, dan lain-lain.

CBIR dalam penelitian ini akan mencoba menggunakan fitur bentuk dan karakteristik histogram dengan ekstraksi ciri statistik. Metode ini dipilih karena melihat dari penelitian mengenai ekstraksi ciri statistik. Selain itu, dalam penelitian ini, akan diimplementasikan metode klasifikasi untuk dapat mengelompokkan gambar-gambar dalam sistem. Algoritma yang digunakan adalah

algoritma *Shingling*.

Kemiripan gambar merupakan permasalahan yang tidak hanya melanggar hak cipta atau kepemilikan. Kemiripan gambar tersebut hanya tinggal melakukan pencarian gambar A atau gambar query dengan gambar B atau gambar sumber, bahkan seluruh isi gambar warna, tekstur, bentuk tinggal melakukan kemiripannya dari kedua gambar tersebut maka gambar yang dibandingkan disederhanakan dalam bentuk set, melalui proses shingling. Algoritma *shingling* tersebut merupakan algoritma yang terbukti dapat melakukan klasifikasi dan menghasilkan akurasi.

Berdasarkan uraian diatas penulis mengangkat judul ” Penerapan *Metode Content Based Image Retrieval* Dengan Algoritma *Shingling* Pada Aplikasi Kemiripan Gambar”.

1.2 Rumusan Masalah

Berdasarkan identifikasi masalah diatas ada pun yang menjadi rumusan masalah pada penulisan skripsi ini adalah:

1. Apakah aplikasi kemiripan gambar yang akan dibuat mampu memproses melakukan perbandingan kemiripan gambar dengan metode CBIR dengan algoritma shingling.
2. Apakah penerapan CBIR dengan algoritma Shingling merupakan algoritma yang efektif untuk melakukan perbandingan dari kedua gambar yaitu gambar A dan gambar B
3. Bagaimana cara mengoptimalkan pencarian gambar dalam sistem Content Based Image Retrieval (CBIR).

1.3 Batasan Masalah

Agar pembahasan tidak menyimpang dari tujuannya maka dilakukan pembatasan masalah sebagai berikut:

1. Gambar yang dapat diproses terfokus gambar dengan ekstensi BMP.
2. Perbandingan kemiripan gambar menggunakan metode *Content Based Image Retrieval* dengan algoritma *Shingling*.

3. Perbandingan warna di lakukan dengan biner 24 bit, 256 color dan 16 color.
4. Bahasa pemrograman yang digunakan PHP dan Visual Basic .Net dan Database MySql

1.4 Tujuan Penelitian

Adapun yang menjadi tujuan dari penelitian ini adalah:

1. Aplikasi kemiripan gambar menggunakan CBIR untuk mengukur kemiripan gambar.
2. Melakukan perbandingan kemiripan gambar menggunakan algoritma shingling.
3. Melakukan proses pixel dari gambar B dengan Gambar A.

1.5 Manfaat Penelitian

Adapun yang menjadi manfaat dari penelitian ini adalah:

1. Aplikasi kemiripan gambar dapat menghasilkan untuk menemukan gambar dalam membandingkan gambar A dan gambar B dengan akurasi yang lebih baik
2. Mengetahui persen kemiripan gambar dari perbandingan gambar B dan gambar A dengan tingkat pixel.
3. Mengetahui sisa persen dari kemiripan gambar dalam perbandingan dari kedua gambar.

BAB II

TINJAUAN PUSTAKA

2.1. Pengenalan Citra

Secara harfiah, citra (*image*) adalah gambar pada bidang dwimatra (dua dimensi). Ditinjau dari sudut pandang matematis, citra merupakan fungsi menerus (continue) dari intensitas cahaya pada bidang 2 dimensi. Citra yang terlihat merupakan cahaya yang direfleksikan dari sebuah objek. Sumber cahaya menerangi objek, objek memantulkan kembali sebagian dari berkas cahaya tersebut dan pantulan cahaya ditangkap oleh alat-alat optik, misal mata manusia, kamera, scanner, sensor satelit, dsb, kemudian direkam. Citra merupakan keluaran dari suatu sistem perekaman data yang bersifat optic, analog maupun digital. Perekaman dari citra dapat dibagi menjadi 2 bagian, yaitu :

1. Citra Analog

Citra analog yaitu terdiri dari sinyal-sinyal elektromagnetik yang tidak dapat dibedakan sehingga pada umumnya tidak dapat ditentukan ukurannya. Citra analog mempunyai fungsi yang kontinu. Hasil perekaman citra analog dapat bersifat optis yakni berupa foto (film foto konvensional) dan bersifat sinyal video seperti gambar pada monitor televisi.

2. Citra Digital

Citra digital terdiri dari sinyal-sinyal yang dapat dibedakan dan mempunyai fungsi yang tidak kontinu yakni berupa titik-titik warna pembentuk citra. Hasil perekaman citra digital dapat disimpan pada suatu media magnetik (Munir, 2014:30).

2.2. Citra Digital

Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Citra merupakan salah satu komponen multimedia dimana memegang peranan yang sangat penting sebagai bentuk informasi visual. Citra mempunyai karakteristik yang tidak dimiliki oleh data teks, yaitu citra kaya dengan informasi. Ditinjau dari sudut pandang matematis, citra merupakan fungsi berkelanjutan dari intensitas cahaya pada bidang dwimatra (dua dimensi).

Citra digital merupakan sebuah larik (*array*) yang berisi nilai-nilai riil maupun kompleks yang direpresentasikan dengan deretan bit tertentu. Pada umumnya citra digital berbentuk empat persegi panjang, dan dimensi ukurannya dinyatakan sebagai tinggi x lebar. Citra Digital merupakan suatu fungsi intensitas cahaya $f(x,y)$, dimana harga x dan y merupakan koordinat spasial dan harga fungsi tersebut pada setiap titik (x,y) merupakan tingkat kecerahan citra pada titik tersebut. Citra digital yang tingginya N , lebarnya M , dan memiliki L derajat keabuan dapat dianggap sebagai fungsi:

$$f(x, y) = \begin{cases} 0 \leq x \leq M \\ 0 \leq y \leq N \\ 0 \leq f \leq L \end{cases} \dots\dots\dots (1)$$

Citra digital yang berukuran $N \times M$ lazim dinyatakan dengan matriks berukuran N baris dan M kolom sebagai berikut:

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

Keterangan:

N = jumlah baris, $0 \leq y \leq N-1$

M = jumlah kolom, $0 \leq x \leq M-1$

L = maksimal warna intensitas (derajat keabuan), $0 \leq f(x,y) \leq L-1$

Masing-masing elemen pada citra digital disebut image element atau piksel. Jadi, citra yang berukuran $N \times M$ mempunyai NM -buah piksel. Proses digitalisasi koordinat (x,y) dikenal sebagai pencuplikan citra (*image sampling*), sedangkan proses digitalisasi derajat keabuan $f(x,y)$ disebut kuantisasi derajat

keabuan (*gray-level quantization*) (Munir, 2016:34). Citra (*image*) adalah representasi optis dari sebuah objek yang disinari oleh sebuah sumber radiasi, pada dasarnya citra yang dilihat terdiri atas berkas-berkas cahaya yang dipantulkan oleh benda-benda disekitarnya, jadi secara alamiah fungsi intensitas cahaya merupakan fungsi sumber cahaya yang menerangi objek, serta jumlah cahaya yang dipantulkan oleh objek, dinotasikan

$$(x, y) = i(x, y).r(x, y) \dots\dots\dots (3)$$

Dimana : $0 < i(x, y) < \infty$ merupakan iluminasi sumber cahaya

$0 < r(x, y) < 1$ merupakan koefisien pantul objek

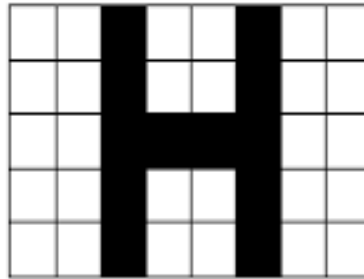
Salah satu bentuk citra adalah citra yang mengandung abstrak dari citra matematis yang berisi fungsi kontinyu dan fungsi diskrit atau citra digital, citra yang memiliki fungsi diskrit inilah yang dapat diolah oleh komputer. Setiap citra digital memiliki beberapa karakteristik, antara lain ukuran citra, resolusi dan format nilainya untuk itu citra digital harus mempunyai *format* tertentu yang sesuai sehingga dapat merepresentasikan objek pencitraan dalam bentuk kombinasi data *biner*. (Muhtadan, Djiwi Harsono, 2018: 468)

Format citra digital yang banyak digunakan adalah citra biner, skala keabuan (*grayscale*), warna dan warna berindeks, film radiografi merupakan citra fisik yang menunjukkan distribusi materi atau energi dari radiasi pengion dimana radiasi pengion menghitamkan film sehingga tingkat kehitaman merupakan wujud dari densitas benda uji, sedangkan bentuk dari struktur benda uji ditunjukkan dengan bentuk citra yang nampak dalam film, pengolahan citra digital memfokuskan transformasi suatu citra pada format digital dan pengolahannya oleh komputer digital input dan output dari sistem pengolahan citra digital adalah citra digital. (Muhtadan, Djiwi Harsono, 2018: 468)

2.2.1 Jenis Citra Digital

Nilai suatu piksel memiliki nilai dalam rentang tertentu, dari nilai minimum sampai nilai maksimum, jangkauan yang digunakan berbeda-beda tergantung dari jenis warnanya. Secara umum jangkauannya adalah 0-255. Berikut adalah jenis-jenis citra berdasarkan nilai pikselnya.

Citra Biner merupakan citra digital yang hanya memiliki dua kemungkinan nilai piksel yaitu hitam dan putih. Citra biner hanya membutuhkan satu bit untuk mewakili nilai setiap piksel dari citra biner. Contoh citra biner dapat dilihat pada gambar 2.1



Gambar 2.1 Citra Biner

Citra Grayscale adalah citra digital yang hanya memiliki satu nilai kanal pada setiap pikselnya, dengan kata lain nilai bagian red = green = blue. Citra grayscale memiliki kedalaman warna 8 bit (256 kombinasi warna keabuan). Contoh citra grayscale dapat dilihat pada gambar 2.2 (Putra, 2017:23).



Gambar 2.2 Citra Grayscale

Citra Warna (24 –Bit) adalah citra yang setiap piksel dari citra warna 24-bit diwakili dengan 24 –bit sehingga total 16.777.216 variasi warna. Variasi ini sudah lebih dari cukup untuk memvisualisasikan seluruh warna yang dapat dilihat penglihatan manusia. Setiap informasi warna disimpan ke dalam 1-byte data.8-bit pertama menyimpan nilai biru. Kemudian diikuti dengan nilai hijau pada 8-bit kedua dan 8-bit terakhir merupakan nilai warna merah (Santoso, 2018:10).

Contoh citra warna dapat dilihat pada gambar 2.3, dibawah ini:



Gambar 2.3 Citra Warna

2.2.2 Format File Citra Digital

File gambar berfungsi untuk menyimpan sebuah gambar yang dapat ditampilkan di layar ke dalam suatu media penyimpanan data. Untuk penyimpanan tersebut digunakan format gambar. Setiap format gambar memiliki karakteristik masing-masing. Beberapa format umum saat ini, yaitu bitmap (.bmp), tagged image format (.tif, tiff), portable network graphics (.png), graphics interchange format (.gif), jpeg (.jpg), mpeg (.mpg), dan lain-lain.

1. Format File BMP (Bitmap)

Pada penelitian ini, citra yang akan digunakan adalah citra yang berformat bitmap. Pada format bitmap, citra bitmap adalah suatu format citra yang memiliki kualitas paling tinggi diantara format citra lainnya. Bitmap mampu menyesuaikan gambar asli dengan gambar yang diwujudkan. Dengan kelebihan tersebut, bitmap juga memiliki kekurangan yaitu ukuran file (*size*) yang lebih besar. Format ini dapat menyimpan informasi dengan kualitas tingkat 1 bit sampai 24 bit. Terjemahan bebas bitmap adalah pemetaan bit. Artinya, nilai intensitas piksel di dalam citra dipetakan ke sejumlah bit tertentu. Peta bit yang umum adalah 8, artinya setiap piksel panjangnya 8 bit. Delapan bit ini merepresentasikan nilai intensitas piksel. Dengan demikian ada sebanyak $2^8 = 256$ derajat keabuan, mulai dari 0 sampai 255. Format bitmap, disimpan sebagai suatu matriks dimana

masing-masing elemennya digunakan untuk menyimpan informasi warna untuk setiap piksel. Jumlah warna yang dapat disimpan ditentukan dengan satuan bit per piksel. Semakin besar ukuran bit per piksel dari suatu gambar semakin banyak pula jumlah warna yang dapat disimpan. Format bitmap ini cocok digunakan untuk menyimpan citra digital yang memiliki banyak variasi dalam bentuk maupun warnanya. Contoh citra bitmap atau .bmp dapat dilihat pada Gambar 2.4.



Gambar 2.4: Gambar Citra .bmp
Sumber Wikipedia

2. Format File JPEG (*Joint Photographic Experts Group*)

Format jpeg atau jpg adalah suatu format citra yang memiliki beberapa kelebihan dan kekurangan dibandingkan format-format lainnya. Kelebihan paling umum adalah ukuran file (size) dan kualitas citra yang dihasilkan lebih fleksibel dibandingkan lainnya. Dan kekurangannya adalah kualitas citranya tidak sesuai dengan aslinya. Format JPEG mendukung mode warna RGB, CMYK dan Grayscale, tetapi tidak mampu menampilkan citra dengan latar belakang transparan. Format JPEG menerjemahkan informasi tersebut menjadi komponen luminance (komponen cahaya) dan dua komponen chromatic (komponen perubahan warna dari hijau ke merah dan dari biru ke kuning). Untuk kompresinya format file citra ini menggunakan kompresi JPEG (Prihatini, 2018:45).

Salah satu contoh citra .jpeg dapat dilihat pada Gambar 2.5.



Gambar 2.5: Contoh Citra JPEG

Sumber Wikipedia

2.2.3 Elemen-Elemen Citra Digital

Citra digital mengandung sejumlah elemen-elemen dasar. Elemen-elemen dasar tersebut dimanipulasi dalam pengolahan citra dan dieksploitasi lebih lanjut dalam computer vision. Elemen-elemen dasar yang penting diantaranya adalah :

1. Kecerahan

Kecerahan adalah kata lain untuk intensitas cahaya. Sebagaimana telah dijelaskan pada bagian penerokan, kecerahan pada sebuah titik (*piksel*) di dalam citra bukanlah intensitas yang sebenarnya, tetapi sebenarnya adalah intensitas rata-rata dari suatu area yang melingkupinya. Sistem visual manusia mampu menyesuaikan dirinya dengan tingkat kecerahan (*brightness level*) mulai dari yang paling rendah sampai yang paling tinggi dengan jangkauan sebesar 1010.

2. Kontras

Kontras menyatakan sebaran terang (*lightness*) dan gelap (*darkness*) di dalam sebuah gambar. Citra dengan kontras rendah dicirikan oleh sebagian besar komposisi citranya adalah terang atau sebagian besar gelap. Pada

citra dengan kontras yang baik, komposisi gelap dan terang tersebar secara merata.

3. Kontur (*Contour*)

Kontur adalah keadaan yang ditimbulkan oleh perubahan intensitas pada pixel pixel yang bertetangga. Karena adanya perubahan intensitas inilah mata kita mampu mendeteksi tepi-tepi (*edge*) objek di dalam citra.

4. Warna

Warna adalah persepsi yang dirasakan oleh sistem visual manusia terhadap panjang gelombang cahaya yang dipantulkan oleh objek. Setiap warna mempunyai panjang gelombang (λ) yang berbeda. Warna merah mempunyai panjang gelombang paling tinggi, sedangkan warna ungu (violet) mempunyai panjang gelombang paling rendah. Warna-warna yang diterima oleh mata (sistem visual manusia) merupakan hasil kombinasi cahaya dengan panjang gelombang berbeda. Penelitian memperlihatkan bahwa kombinasi warna yang memberikan rentang warna yang paling lebar adalah red (R), green (G), dan blue (B).

5. Bentuk (*Shape*)

Shape adalah properti intrinsik dari objek tiga dimensi dengan pengertian bahwa shape merupakan properti intrinsik utama untuk sistem visual manusia. Manusia lebih sering mengasosiasikan objek dengan bentuknya ketimbang elemen lainnya (warna misalnya). Pada umumnya, citra yang dibentuk oleh mata merupakan citra dwimatra (2 dimensi), sedangkan objek yang dilihat umumnya berbentuk trimatra (3 dimensi). Informasi bentuk objek dapat diekstraksi dari citra pada permulaan pra-pengolahan dan segmentasi citra. Salah satu tantangan utama pada computer vision adalah merepresentasikan bentuk, atau aspek-aspek penting dari bentuk.

6. Tekstur

Tekstur dicirikan sebagai distribusi spasial dari derajat keabuan di dalam sekumpulan piksel-piksel yang bertetangga. Jadi, tekstur tidak dapat didefinisikan untuk sebuah piksel. Sistem visual manusia pada hakikatnya tidak menerima informasi citra secara independen pada setiap pixel,

melainkan suatu citra dianggap sebagai suatu kesatuan. Resolusi citra yang diamati ditentukan oleh skala pada mana tekstur tersebut dipersepsi (Sutoyo, 2017:67).

2.2.4. Warna

Isi dari sebuah citra digital adalah piksel atau kotak warna. Manusia dapat melihat radiasi elektromagnetik dengan panjang gelombang 400 sampai 700 nanometers (nm) sebagai warna. Hewan juga bisa melihat sisi yang berbeda dari spektrum elektromagnetik dan dapat melihat warna yang berbeda dari apa yang tidak dapat dilihat oleh manusia.

Pengalaman warna secara natural adalah proses kombinasi dari mata dan otak. Mata bertindak sebagai penerima cahaya dan otak menginterpretasikan data dari mata sebagai informasi visual dan menerjemahkan data tersebut sebagai warna. Penglihatan manusia didasarkan atas tiga penerima, satu untuk merah, yang lain untuk hijau, sisanya untuknya biru. Ada banyak representasi warna dari banyak perbedaan lingkup warna, atau model yang biasa memiliki tiga atau empat channel (Santoso, 2018:54).

Sistem yang dipakai untuk mewakili warna yaitu sistem RGB (Red, Green, Blue). Sistem RGB adalah sistem penggabungan antara warna-warna primer (additive primary colours) yaitu merah (Red), Hijau (Green) dan Biru (Blue) untuk memperoleh warna tertentu. Misalnya warna putih diperoleh dari hasil gabungan warna merah = 255, hijau = 255 dan biru = 255. Dalam sistem RGB, warna putih cerah dinyatakan dengan RGB (255, 255, 255). Range nilai dari setiap warna primer adalah 0 sampai 255.

Sehingga kemungkinan warna yang dapat terbentuk dengan sistem RGB adalah $256 \times 256 \times 256$ yakni kurang lebih 16.7 juta warna. Pada tabel dibawah ini diperlihatkan beberapa kode warna hasil gabungan warna RGB (Siregar Arifin, 2016:11). Di bawah ini merupakan contoh tabel warna untuk sistem warna RGB atau Red, Green Blue :

Tabel 2.1 Kode Warna RGB

Colour	Red	Green	Blue
Black	0	0	0
Blue	0	0	255
Green	0	255	0
Cyan (Blue+Green)	0	255	255
Red	255	0	0
Magenta (Red+Blue)	255	0	255
Yellow (Red+Green)	255	255	0
White (Red+Green+Blue)	255	255	255
Gray	128	128	128

2.2.4.1 Model RGB (*Red, Green, Blue*)

Model warna RGB adalah mode 1 warna berdasarkan konsep penambahan kuat cahaya primer yaitu Red, Green dan Blue. Struktur warna RGB dapat dilihat pada gambar 2.4



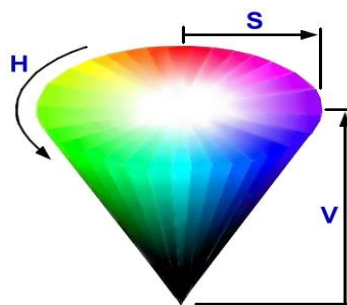
Gambar 2.6: Struktur Warna RGB
Sumber : Wikipedia

Dalam suatu ruang yang sama sekali tidak ada cahaya, maka ruangan tersebut adalah gelap total. Tidak ada signal gelombang cahaya yang diserap oleh mata kita atau RGB (0,0,0). Apabila kita menambahkan cahaya merah pada ruangan tersebut, maka ruangan akan berubah warna menjadi merah misalnya RGB (255,0,0), semua benda dalam ruangan tersebut hanya dapat terlihat berwarna merah.

Demikian apabila cahaya kita ganti dengan hijau atau biru. Berdasar pada tri-stimulus vision theory yang mengatakan bahwa manusia melihat warna dengan cara membandingkan cahaya yang datang dengan sensor sensor peka cahaya pada retina (yang berbentuk kerucut). Sensor-sensor tersebut paling peka terhadap cahaya dengan panjang gelombang 630 nm (merah), 530 nm (hijau) dan 450 nm (biru). Model ini dapat digambarkan dengan kubus dengan sumbu-sumbu R, G dan B (Santoso, 2018:50).

2.2.4.2 Model HSV (*Hue, Saturation, and Value*)

Model warna HSV mendefinisikan warna dalam terminologi Hue, Saturation dan Value. Hue menyatakan warna sebenarnya, seperti merah, violet, dan kuning. Hue digunakan untuk membedakan warna-warna dan menentukan kemerahan (*redness*), kehijauan (*greenness*), dsb, dari cahaya. Hue berasosiasi dengan panjang gelombang cahaya. Saturation menyatakan tingkat kemurnian suatu warna, yaitu mengindikasikan seberapa banyak warna putih diberikan pada warna. Value adalah atribut yang menyatakan banyaknya cahaya yang diterima oleh mata tanpa memperdulikan warna. Model warna HSV dapat dilihat pada gambar 2.5 (Sutoyo, 2017:60).



Gambar 2.7: Model Warna HSV

Sumber : Wikipedia

2.2.4.3 Model HSL (*Hue, saturation , and Lightness*)

Pada dasarnya model warna HSL hampir sama dengan model warna HSV. Model warna HSL terdiri dari 3 komponen yaitu Hue, Saturation , dan Lightness. Hue merupakan karakteristik warna berdasar cahaya yang dipantulkan oleh objek, dalam warna dilihat dari ukurannya mengikuti tingkatan 0 sampai 360. Sebagai contoh, pada tingkat 0 adalah warna Merah, 60 adalah warna Kuning, untuk warna Hijau pada tingkatan 120, sedangkan pada 180 adalah warna Cyan. Untuk tingkat 240 merupakan warna Biru, serta 300 adalah warna Magenta.

Saturation atau Chroma adalah tingkatan warna berdasarkan ketajamannya berfungsi untuk mendefinisikan warna suatu objek cenderung murni atau cenderung kotor (gray). Saturation mengikuti persentase yang berkisar dari 0% sampai 100% sebagai warna paling tajam. Lightness adalah tingkatan warna berdasarkan pencampuran dengan unsur warna Putih sebagai unsur warna yang memunculkan kesan warna terang atau gelap. Nilai koreksi warna pada Lightness berkisar antara 0 untuk warna paling gelap dan 100 untuk warna paling terang (Santoso, 2018:76).

2.2.4.4 Model CMYK (*Cyan Magenta Yellow Key*)

Cyan Magenta Yellow Key (CMYK), atau sering disingkat sebagai CMYK adalah proses pencampuran pigmen yang lazim digunakan percetakan. Tinta process cyan, process magenta, process yellow, process black dicampurkan dengan komposisi tertentu dan akurat sehingga menghasilkan warna tepat seperti yang diinginkan. Bahkan bila suatu saat diperlukan, warna ini dengan mudah bisa dibentuk kembali. Sistem CMYK juga digunakan oleh banyak printer kelas bawah karena keekonomisannya.

CMYK adalah kependekan dari cyan, magenta, yellow-kuning, dan warna utamanya *black* (hitam), dan seringkali dijadikan referensi sebagai suatu proses pewarnaan dengan mempergunakan empat warna) adalah bagian dari model pewarnaan yang sering dipergunakan dalam pencetakan berwarna. Namun ia juga dipergunakan untuk menjelaskan proses pewarnaan itu sendiri. Meskipun berbeda-beda dari setiap tempat pencetakan, operator surat khabar, pabrik surat

kabar dan pihak-pihak yang terkait, tinta untuk proses ini biasanya, diatur berdasarkan urutan dari singkatan tersebut. Model ini, baik sebagian ataupun keseluruhan, biasanya ditampilkan dalam gambar dengan warna latar putih warna ini dipilih, dikarenakan dia dapat menyerap panjang struktur cahaya tertentu. Model seperti ini sering dikenal dengan nama “*subtractive*”, arena warna-warnanya mengurangi warna terang dari warna putih.

Dalam model yang lain “*additive color*”, seperti halnya RGB (Red-Merah, Green-Hijau, Blue-Biru), warna putih menjadi warna tambahan dari kombinasi warna-warna utama, sedangkan warna hitam dapat terjadi tanpa adanya suatu cahaya. Dalam model CMYK, berlaku sebaliknya : warna putih menjadi warna natural dari kertas atau warna latar, sedangkan warna hitam adalah warna kombinasi dari warna-warna utama. Untuk menghemat biaya untuk membeli tinta, dan untuk menghasilkan warna hitam yang lebih gelap, buatlah satu warna hitam khusus yang menggantikan warna kombinasi dari cyan, magenta dan kuning (Santoso, 2018:80).

2.3. Perbaikan Kualitas Citra (*Image Enhancement*)

Perbaikan kualitas citra merupakan salah satu proses awal dalam pengolahan citra. Perbaikan kualitas diperlukan karena seringkali citra yang dijadikan objek pembahasan mempunyai kualitas yang buruk, misalnya citra mengalami derau (noise) pada saat pengiriman melalui saluran transmisi, citra terlalu terang/gelap, citra kurang tajam, kabur, dan sebagainya. Melalui operasi pemrosesan awal inilah kualitas citra diperbaiki sehingga citra dapat digunakan untuk aplikasi lebih lanjut, misalnya untuk aplikasi pengenalan (recognition) objek di dalam citra. Proses-proses yang termasuk ke dalam perbaikan kualitas citra (Sutoyo, 2019:10)

1. Pengubahan kecerahan gambar (*image brightness*)

Pengubahan kecerahan gambar dilakukan untuk membuat citra lebih terang atau lebih gelap. Kecerahan gambar dapat diperbaiki dengan menambahkan (atau mengurangi) sebuah konstanta kepada (atau dari) setiap piksel di dalam citra.

2. Peregangan kontras (*contrast stretching*)

Kontras menyatakan sebaran terang (*lightness*) dan gelap (*darkness*) di dalam sebuah gambar. Citra dapat dikelompokkan ke dalam tiga kategori kontras : citra kontras rendah (*low contrast*), citra kontras bagus (*good contrast/normal contrast*), dan citra kontras tinggi (*high contrast*).

3. Pengubahan histogram citra

Nilai-nilai intensitas di dalam citra diubah sehingga penyebarannya seragam (*uniform*). Tujuan dari perataan histogram adalah untuk memperoleh penyebaran histogram yang merata, sedemikian sehingga setiap derajat keabuan memiliki jumlah piksel yang relatif sama.

4. Pelembutan citra (*image smoothing*)

Pelembutan citra bertujuan untuk menekan gangguan (*noise*) pada citra.

5. Penajaman tepi (*sharpening edge*)

Operasi penajaman citra bertujuan memperjelas tepi objek di dalam citra dengan menghilangkan bagian citra yang lembut.

6. Pengubahan geometric

Koreksi geometrik dilakukan pada citra yang memiliki gangguan yang terjadi pada waktu proses perekaman citra, misalnya pergeseran koordinat citra (*translasi*), perubahan ukuran citra, dan perubahan orientasi koordinat citra (*skew*). Koreksi geometri yang sederhana adalah dengan operasi geometri sederhana, seperti rotasi, translasi, dan penskalaan citra.

7. Pewarnaan semu (*pseudocoloring*)

Pewarnaan semu adalah proses memberi warna tertentu pada nilai-nilai piksel suatu citra berdasarkan kriteria tertentu, misalnya suatu warna tertentu untuk suatu interval derajat keabuan tertentu.

2.4. Content Based Image Retrieval (CBIR)

CBIR adalah aplikasi yang digunakan untuk pengambilan *query image* dari sebuah arsip gambar yang besar, dengan semakin bertambahnya koleksi multimedia, perkembangan alat untuk melakukan pencarian informasi semakin dibutuhkan, pada zaman sekarang ini telah banyak terdapat mesin pencari gambar

yang menggunakan tenamun mesin pencari gambar yang menggunakan intensitas dan warna dari gamks, bar masih sulit ditemukan, fitur dari gambar digital seperti fitur bentuk, warna, dan tekstur dapat digunakan sebagai kunci indeks untuk melakukan pencarian dan pengambilan gambar dari suatu database yang besar, dalam penelitian yang telah dilakukan diharapkan hanya terdapat perbedaan halus dalam pengambilan gambar yang dilakukan oleh pengguna dengan menggunakan teks dan gambar, mengingat bahwa pengambilan gambar dengan menggunakan teks dapat berhasil mengambil dokumen tanpa memahami isi, biasanya tidak mudah bagi pengguna untuk memberikan gambaran tentang apa yang sedang dicari dengan menggunakan teks. (Arwin Halim: 2017: 88).

Di tengah kondisi seperti ini muncul suatu masalah penamaan berkas gambar yang direkam, karena nama berkas gambar yang direkam, disimpan bukanlah berdasarkan situasi atau kejadian dari gambar itu sendiri, melainkan dengan menggunakan kode tanggal, bulan, tahun dan nomor urut dari gambar itu direkam, atau juga menggunakan nama *default* dari pengaturan pemilik alat perekam itu sendiri, dalam kondisi ini hampir dapat di pastikan tidaklah mudah untuk mencari gambar yang diinginkan dengan cepat, apalagi mencari gambar yang sama menggunakan nama dari gambar.

Selain itu dengan perkembangan komputer dan aplikasi-aplikasi pengolahan gambar juga, sangat memudahkan setiap orang untuk merubah ataupun menggandakan berkas-berkas gambar dengan sangat cepat, hal ini membuka peluang untuk terjadi proses duplikasi secara tidak sengaja ataupun disengaja yang tidak penting dan tidak perlu sehingga dapat mengurangi sedikit banyaknya kapasitas media penyimpanan yang ada, selain itu berkas gambar yang terduplikat juga dapat mengganggu struktur *file* dan *folder* yang sudah disusun dengan rapi, dan terkadang dapat membuat bingung untuk mencari lokasi berkas.

Metode Content Based Image Retrieval atau disingkat dengan CBIR adalah sebuah *metode* pencarian berkas gambar, berkas gambar dicari menggunakan unsur-unsur atau konten-konten yang terkandung pada gambar itu sendiri, jadi pencarian tidak lagi menggunakan properti dari berkas gambar seperti nama, tanggal, atau yang lainnya.

Image Retrieval adalah aplikasi pengolahan citra yang dapat membantu pengguna mengambil atau mencari dengan cepat citra pada suatu database citra berdasarkan *query* atau permintaan pengguna (Putra, 2010, chap.1). Dalam penjelasan lain menyebutkan bahwa *Content Based Image Retrieval (CBIR) is considered as the process of retrieving desired images from huge database based on extracted features from the image themselves (without resorting to a key word)* Latha, Y.M., Jinaga, B.C., Reddy, V.S.K., 2007, p.12 dari kedua pengertian diatas sudah dapat disimpulkan bahwa *Content Based Image Retrieval* adalah metode penemuan kembali berkas citra tanpa menggunakan sebuah kata kunci melainkan dengan berbasis ekstraksi fitur.

Ekstraksi fitur atau dengan membuat *query* dari konten-konten citra yaitu mengambil informasi-informasi yang terkandung dalam isi citra, seperti fitur warna, aplitudo, deteksi tepi, wavelet dan lain-lain yang kemudian disimpan ke dalam basis data dan setelah itu dilakukan hal yang sama kepada gambar kunci atau gambar *query*, kemudian membandingkannya dengan yang ada pada basis data. (Insan Taufik: 2018:17)

Contoh penerapan CBIR dalam kehidupan sehari-hari adalah pada bidang medis. Penggunaan CBIR dalam bidang medis pada umumnya diimplementasikan dengan menggabungkan *database* pusat dengan arsitektur sistem distribusi yang cocok untuk database gambar besar seperti dalam pengarsipan gambar dan sistem komunikasi.

Dalam penerapan CBIR untuk fitur warna dan bentuk digunakan *metode Color retrieval* untuk pencarian gambar berdasarkan fitur warnanya yang dilakukan berdasarkan nilai hue dan untuk fitur bentuk dari warnanya digunakan *metode Shape retrieval* yang dilakukan berdasarkan nilai *grayscale* suatu gambar dan dengan menggunakan teknik *clustering k-means*. (Arwin Halim:2017 Vol:15 Hal:84)

Suatu gambar memiliki ciri yang berbeda satu dan yang lainnya bergantung pada karakteristik yang menonjol pada gambar tersebut sebagai contoh, bunga matahari dan bunga melati dibedakan melalui perbedaan warnanya, kain dan kertas dapat dibedakan dari teksturnya, sedangkan gambar lingkaran

dengan gambar kotak dibedakan melalui bentuknya, masing-masing ciri dasar dari gambar ini didapatkan melalui proses ekstraksi ciri yang tidak mudah, karena satu gambar dapat mempunyai *multiple feature*. Proses ekstraksi ciri yang baik menentukan keberhasilan dalam membangun aplikasi *image* CBIR atau *Content Based Image Retrieval* merupakan suatu teknik untuk mencari suatu gambar dengan membandingkan gambar query dengan gambar yang ada di database (*Query By Example*) sistem CBIR secara umum dibangun dengan melihat karakteristik dari suatu gambar atau dengan kata lain dengan melihat ciri dari gambar tersebut. Ciri merupakan suatu tanda yang khas, yang membedakan antara satu gambar dengan gambar yang lain dan pada dasarnya suatu gambar memiliki ciri ciri. (Nana Ramadijanti,2017:50)

Content Based Image Retrieval (CBIR) atau temu kenali citra merupakan suatu metode yang digunakan untuk melakukan pencarian citra digital pada suatu basis data citra. Beberapa konten aktual pada sebuah citra yang meliputi warna, bentuk, tekstur atau informasi lain yang didapatkan dari citra tersebut merupakan objek yang dianalisa dalam proses pencarian *content based*. CBIR merupakan suatu aplikasi dari *computer vision* yang mempunyai teknik pencarian gambar yang diambil dari basis data yang menyediakan gambar sebagai gambar uji. Proses *query* gambar dilakukan dengan mengekstraksi fitur yang meliputi histogram, nilai warna, tekstur, dan deteksi tepi. (Rizky Aemelia Dkk, 2018:26)

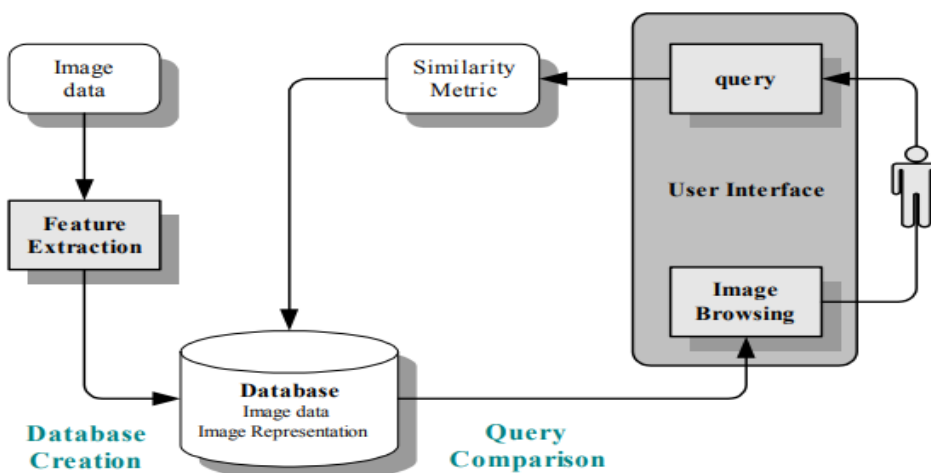
Dalam sistem pencarian data terdiri dari dua kategori yaitu *text base* dan *content based*. *User* biasanya dihadapkan pada model pencarian berdasarkan teks yang mengharuskan pengguna memasukan teks untuk mengambil atau menampilkan data dari database walaupun prosesnya lebih cepat, namun sistem pencarian citra berdasarkan teks membutuhkan ketergantungan terhadap manusia yang sangat tinggi dan sering kali perbedaan persepsi dalam mendeskripsikan citra menyebabkan hasil yang diinginkan tidak sesuai. Pada *content based image retrieval* proses pemanggilan kembali data citra dengan mencocokkan citra yang dicari dan mencocokkan ciri-cirinya terhadap citra yang ada didalam data *base* citra teknik inilah yang disebut dengan *metode content based image retrieval*. (Faradilla Dkk:2017)

Pada *image query* dilakukan proses ekstraksi fitur yang merupakan proses secara umum dari CBIR. Histogram, susunan warna, tekstur, bentuk, tipe spesifikasi dari objek, tipe event tertentu dan nama individu merupakan parameter fitur image yang dapat digunakan untuk proses retrieval.

Dalam CBIR setiap gambar yang disimpan dalam database telah diekstrak dan fiturnya dibandingkan dengan fitur dari citra query, dalam hal ini ada dua langkah untuk melakukan nya yaitu :

- Ekstraksi Fitur : langkah pertama dalam proses ini adalah untuk mengekstrak fitur gambar untuk sebagian dibedakan.
- Matching atau Pencocokan : langkah kedua adalah pencocokan fitur untuk menghasilkan hasil yang secara visual mirip dengan citra *query*.

Proses paling penting pada sistem CBIR adalah ekstraksi fitur, karena hasil dari proses ini akan diketahui perbedaan pada setiap image berdasarkan cirinya seperti bentuk, warna, dan tekstur, teknik analisis komponen utama, histogram warna, *wavelet transform* dan lainnya merupakan beberapa teknik yang dapat digunakan untuk *ekstraksi fitur*, proses pencocokan *image* untuk memperoleh *image* yang mempunyai kemiripan dengan *image query*. Proses *matching* dilakukan dengan menghitung jarak antara *image* yaitu *image query* (acuan) dan *image target* (uji) pada basis data, parameter yang digunakan dalam perhitungan jarak berdasarkan pada hasil *ekstraksi fitur*.



Gambar 2.8 Diagram arsitektur CBIR.

Sumber : Wikipedia

Content-based image retrieval (CBIR) adalah teknik pencarian gambar dari database gambar yang besar dengan menganalisa fitur-fitur dari gambar seperti warna, bentuk, tekstur, atau informasi lainnya yang bisa diekstrak dari gambar, ada 3 tahap fundamental pada *content based image retrieval*, yaitu tahap ekstraksi fitur dari gambar, penyimpanan fitur, dan pencarian gambar, pada *content based image retrieval*, gambar pada data base melewati tahap ekstraksi fitur dan hasilnya disimpan ke database kemudian, gambar *query* juga diekstrak fiturnya kemudian dibandingkan dengan fitur pada database dengan pengukuran kemiripan yang menghasilkan hasil pencarian gambar yang mirip. (Arwin Halim: 2017:16)

Content Based berarti bahwa pencarian akan menganalisis konten dari gambar, bukan metadata seperti kata kunci, tag, atau deskripsi terkait dengan gambar, konten yang dimaksud dapat berupa warna, tekstur, bentuk, tepi, dan lain-lain. *Content Based Image Retrieval* (CBIR) adalah suatu teknik dimana setiap gambar akan diekstraksi fitur-fiturnya, dan pencarian gambar hanya berdasarkan pada fitur-fitur gambar tersebut, dalam sistem pencarian gambar berdasarkan konten, pengguna sistem memasukkan gambar dan sistem harus menemukan gambar yang relevan dengan gambar masukan. (Arwin Halim, 2016:31)

Sistem arsitektur pada CBIR dibagi menjadi dua tahap, tahap pertama merupakan tahap memasukkan gambar untuk pencarian, sedangkan tahap kedua merupakan tahap dimana kumpulan gambar akan diekstraksi fiturnya dan kemudian disimpan ke dalam basis data. Ketika pencarian gambar dilakukan, fitur pada gambar tersebut akan diekstraksi fiturnya. Pada tahap pencarian, fitur dari gambar masukan akan dibandingkan dengan setiap fitur gambar di dalam basis data. Perbandingan fitur antara gambar masukan dan fitur dalam basis data dapat dilakukan dengan menggunakan metode pengukuran kemiripan (*Similarity Measurement*), seperti *Euclidean Distance*, *Chi-Square Distance*, dan *Weighted Euclidean Distance*.

2.5. Metode Pengukuran Kemiripan (*Similarity Measure*)

Proses CBIR terdapat sub proses matching (pencocokan) dan pengukuran kemiripan (*similarity measure*) merupakan salah satu proses penting yang harus diperhatikan. Pengukuran derajat kesamaan atau kecocokan pada dua atau lebih citra. hal itu dilakukan dengan menghitung kemiripan (*similarity*) untuk mencari nilai-nilai kemiripan dari suatu citra dengan citra lainnya berdasarkan *distance* (jarak) vektor. Jarak digunakan untuk menentukan kesamaan (*similarity degree*) atau ketidaksamaan (*similarity degree*) dua vektor fitur. Tingkat kesamaan berupa suatu nilai (*score*) dan berdasarkan nilai tersebut dua vektor fitur akan dikatakan mirip atau tidak. Semakin besar nilai *distance* (mendekati satu), maka kedua citra tersebut semakin berbeda, sebaliknya semakin kecil nilai *distance* (mendekati nol), maka semakin mirip kedua citra tersebut (Putra, 2019:47). Adapun beberapa metode yang dapat digunakan untuk mengukur tingkat kemiripan dua buah vektor fitur yaitu:

1. *Euclidean Distance*

Metrika yang paling sering digunakan untuk menghitung kesamaan dua vektor *euclidean distance* menghitung akar dari kuadrat perbedaan dua vektor (*root of square differences between two vectors*). Semakin kecil nilai jarak euclidean maka semakin mirip dua vektor tersebut dan sebaliknya semakin besar nilai jarak euclidean maka semakin tidak mirip kedua vektor tersebut (Zhang, 2020:4). Adapun persamaan *euclidean distance* sebagai berikut:

$$d(A,B) = \sqrt{\sum_{j=1}^n (H_j^A - H_j^B)^2} \dots\dots\dots (3)$$

Keterangan :

A : Vektor A

B : Vektor B

d (A,B) : Jarak Euclidean antara vektor A dan vektor B

n : Jumlah elemen vektor

j : Indeks elemen vektor

H : Elemen vektor

2. *City Block Distance*

City block distance biasanya disebut sebagai manhattan distance atau boxcar distance atau absolute value distance. *City block distance* menghitung nilai mutlak perbedaan dari 2 vektor (Putra, 2019:48). Adapun persamaan *city block distance* sebagai berikut

$$D_1(X_2, X_1) = \sum_{f=1}^p |X_{2f} - X_{1f}| \quad (4)$$

Keterangan :

Dt : Jarak City Block

X2 : Vektor kedua

X1 : Vektor Pertama

f : Indeks elemen vektor

p : Jumlah elemen vektor

3. *Chebyshev Distance*

Chebyshev distance disebut juga *maximum value distance* yang mengukur jarak berdasarkan nilai mutlak atau sebuah magnitudo absolut perbedaan 2 vektor. Dari masing – masing nilai perbedaan akan dipilih nilai paling besar untuk dijadikan chebyshev distance) (Putra, 2019:74). Adapun persamaan *chebyshev distance* sebagai :

$$d_{i,j} = \max_k (|X_{ik} - X_{jk}|) \quad (5)$$

Keterangan :

$d_{i,j}$: Jarak *Chebyshev*

X_i : Vektor pertama

X_j : Vektor kedua

k : Indeks Matriks

4. *Minkowski Distance*

Minkowski distance dengan ordo λ ini menggeneralisasikan beberapa metrika sebelumnya, dimana $\lambda=1$ dinyatakan sebagai city block distance, $\lambda=2$ dinyatakan dengan euclidean distance dan $\lambda=\infty$ (tak terhingga) dinyatakan dengan Chebyshev distance) (Putra, 2019:80).

Adapun persamaan *minkowski distance* sebagai berikut :

$$d_p(Q, T) = \left(\sum_{i=0}^{N-1} (Q_i - T_i)^\lambda \right)^{\frac{1}{\lambda}} \dots\dots\dots (6)$$

Keterangan :

Dp : Jarak Minkowski

Q : Vektor Q

T : Vektor T

N : Jumlah elemen vektor

i : Indeks elemen vektor

5. Canberra Distance

Dalam canberra distance, untuk setiap nilai dari 2 vektor yang akan dicocokkan, canberra distance membagi absolute selisih 2 nilai dengan jumlah dari absolute 2 nilai tersebut. Hasil dari setiap dua nilai dicocokkan lalu dijumlahkan untuk mendapatkan canberra distance. Jika kedua koordinat nol – nol kita memberikan definisi dengan $0/0 = 0$. Canberra distance ini sangat peka terhadap sedikit perubahan dengan kedua koordinat mendekati nol

Adapun persamaan canberra distance sebagai berikut :

$$d_{i,j} = \sum_k^n = 1 \frac{|X_{ik} - X_{jk}|}{|X_{ik}| + |X_{jk}|} \dots\dots\dots (7)$$

Keterangan :

di,j : Jarak Canberra

Xi : Vektor i

Xj : Vektor j

k : Indeks vektor

n : Jumlah elemen vektor

2.6. Algoritma

Algoritma adalah kumpulan instruksi atau perintah yang dibuat secara jelas dan sistematis berdasarkan urutan yang logis (logika) untuk penyelesaian suatu masalah. Sejumlah konsep yang mempunyai relevansi dengan masalah rancangan program yaitu kemampuan komputer, kesulitan dan ketepatan.

Untuk keperluan matematika dan program komputer metode yang sering digunakan yaitu:

1. Diagram Alir (Flowchart)
2. Kode Semu (Pseudocode)
3. Algoritma Fundamental.

Komponen utama dalam algoritma yaitu *finiteness*, *definiteness*, *input*, *output*, dan *effectiveness*. Sehingga dalam merancang sebuah algoritma ada 3 komponen yang harus ada yaitu:

1. Komponen masukan (*input*)

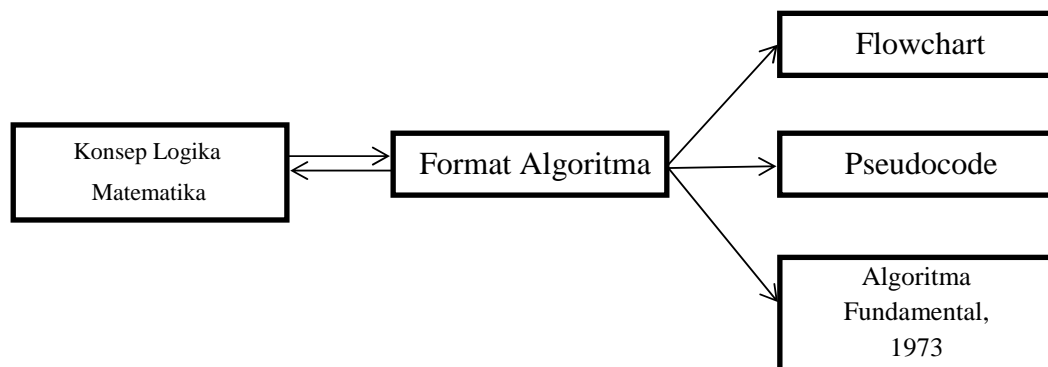
Komponen ini biasanya terdiri dari pemilihan variabel, jenis variabel, tipe variabel, konstanta dan parameter (dalam fungsi).

2. Komponen keluaran (*output*)

Komponen ini merupakan tujuan dari perancangan algoritma dan program. Permasalahan yang diselesaikan dalam algoritma dan program harus ditampilkan dalam komponen keluaran. Karakteristik keluaran yang baik adalah benar (menjawab) permasalahan dan tampilan yang ramah (*user friendly*).

3. Komponen Proses (*Procesing*)

Komponen ini merupakan bagian utama dan terpenting dalam merancang sebuah algoritma. Dalam bagian ini terdapat logika masalah, logika algoritma (sintaksis dan semantik), rumusan, metode (rekursi, perbandingan, penggabungan, pengurangan dan lain-lain).



Gambar 2.9. Struktur Hubungan dan Jenis Algoritma

2.6.1. Definisi Analisis Algoritma

Algoritma tidak selalu memberikan hasil terbaik yang mungkin diperoleh, maka diharapkan adanya suatu evaluasi mutu hasil dari algoritma tersebut, Sekali sebuah algoritma diberikan kepada sebuah permasalahan dan dijamin akan memberikan hasil yang diharapkan, maka langkah penting selanjutnya adalah menentukan besar biaya yang diperlukan algoritma tersebut untuk memperoleh hasil itu. Proses inilah yang disebut dengan analisis algoritma.

Ukuran biaya eksekusi suatu algoritma yang paling sering digunakan adalah lamanya waktu diperlukan, Namun juga masih ada ukuran-ukuran lainnya, misalnya besarnya memori yang diperlukan untuk mengeksekusi algoritma tersebut, Maksud dilakukannya analisis algoritma adalah untuk :

1. Memenuhi aktivitas intelektual
2. Meramalkan suatu hal yang akan terjadi atau yang akan didapat algoritma tersebut.
3. Mengetahui efektifitas suatu algoritma dibanding dengan algoritma yang lain untuk persoalan yang sama.

2.6.2. Algoritma Shingling

Pada penelitian yang berjudul "*Identifying and Filtering Near-Duplicate Documents*", dibahas tentang masalah mendeteksi kemiripan halaman web satu dengan web yang lain, pada proses web *indexing* skala besar yang dilakukan oleh *Alta Vista search engine*. Andrei Broder mengajukan algoritma untuk menyaring *near-duplicate documents*, dan sukses diimplementasikan selama kurang lebih

tiga tahun pada *Alta Vista*. Dokumen yang dibandingkan disederhanakan dalam bentuk set, melalui proses *shingling*.

Pada penelitian tersebut, satu *shingle* terdiri dari 4 kata, sehingga disebut *4-shingling* setiap *shingle* dibangkitkan sebuah *fingerprint* untuk kemudian dibandingkan dengan nilai *fingerprint* dari *shingle* pada dokumen kedua, Algoritma yang digunakan untuk membangkitkan nilai *fingerprint* adalah *Rabin's fingerprint*. *Rabin's* membangkitkan nilai *fingerprint* sebesar 64 bit (8 byte). (Sumber: Frendy Juniarto Baba, 2016:2).

Penelitian yang dilakukan membahas tentang *near-duplicate* dan *plagiarism*. *Plagiarism* secara sederhana adalah merepresentasikan ide orang lain sebagai ide milik sendiri, baik terjadi secara sengaja maupun secara kebetulan. *Near-duplicate* adalah kondisi ketika dua dokumen memiliki isi yang sama berbeda pada beberapa kata atau susunan kata.

Algoritma *Shingling* merupakan algoritma yang ditemukan oleh Andrei Broder, algoritma ini bekerja dengan cara membuat sebuah *shingle* yang berisi beberapa kata dengan jumlah yang tetap. Angka yang menentukan jumlah kata dalam satu *shingle* ini disebut gram. Pada tiap *shingle* dibangkitkan nilai *fingerprint*. Proses *shingling* ini akan menghasilkan himpunan yang berisi sejumlah *fingerprint*. Himpunan ini kemudian dibandingkan dengan himpunan yang dihasilkan dari dokumen kedua, nilai kemiripan diperoleh dengan cara membagi jumlah *fingerprint* yang sama (*intersection*) dari dua dokumen dengan jumlah *fingerprint* gabungan (*union*).

Proses perhitungan tersebut didefinisikan sebagai berikut:

$$r(A,B) = \frac{S_{A \cap B}}{S_{A \cup B}}$$

Persamaan 1 Rumus Nilai Kemiripan

Algoritma *Shingling* dilakukan melalui beberapa langkah berikut :

1. Hilangkan tanda baca pada dokumen.
2. Dimulai dengan kata pertama, buat satu *shingle* berisi kata pertama tersebut sampai 3 kata berikutnya.

3. Pindah ke kata kedua, buat *shingle* berisi kata kedua dan 3 kata berikutnya.
4. Lakukan pembentukan *shingle* sampai dengan 4 kata terakhir dari dokumen tersebut.
5. Untuk tiap *shingle*, bangkitkan nilai *fingerprint*.
6. Lakukan langkah 1 sampai dengan 5 untuk dokumen kedua.
7. Gunakan rumus nilai kemiripan dokumen (Persamaan 1) untuk menghitung nilai kemiripan dokumen. (Muhamad Yusuf Adiansyah, 2019:4)

Hingga pada waktu penelitian ini ditulis, belum ada penelitian sebelumnya yang membahas penggunaan algoritma *Shingling* dengan *metode Content Based Image Retrieval*, Berdasarkan penelitian - penelitian yang telah dilakukan tentang *Content Based Image Retrieval* pada citra digital, maka dilakukan penelitian yang bertujuan untuk merancang dan mengimplementasikan aplikasi *Content Based Image Retrieval*, Pada *image* berupa *Format PNG* dengan algoritma *Shingling*, sehingga diperoleh *stego image*.

Pada penelitian ini digabungkan antara *Metode Content Based Image Retrieval* dengan Algoritma *Shingling*, Kelebihan dari *Metode Content Based Image Retrieval* merupakan suatu aplikasi dari komputer vision yang mempunyai teknik pencarian gambar yang diambil dari basis data yang menyediakan gambar sebagai gambar uji, *Shingling* berfungsi untuk memiliki lokasi penyisipan terbaik dengan cara mencari lokasi piksel yang memiliki bit paling mirip dengan bit pesan. (Muhamad Yusuf Adiansyah, 2019:4).

2.7. Visual Basic .Net

Microsoft Visual Studio adalah sebuah *Integrated Development Environment* buatan *Microsoft Corporation*, *Microsoft Visual Studio* dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan di atas *Windows*) ataupun *managed code* (dalam bentuk *Microsoft Intermediate Language* di atas *.NET Framework*). Selain itu, *Visual Studio* juga dapat digunakan untuk mengembangkan aplikasi *Silverlight*,

aplikasi *Windows Mobile* (yang berjalan di atas *.NET Compact Framework*), *Visual Basic* mencakup sebuah kode editor yang didukung oleh fitur *intellisense* atau yang disebut dengan *code refactoring*. *Debugger* telah terintegrasi bekerja pada *level source level debugger* dan *level debugger mesin*. *Toll built in* mencakup *form desainer* untuk membangun sebuah aplikasi *GUI*, *web desainer*, *class desainer* dan *database schema desainer*.

Microsoft Visual Studio didukung bahasa pemrograman yang berbeda adapun bahasa pemrograman yang didukung oleh *Visual Basic Studio* adalah bahasa pemrograman *C++*, *Visual Basic*, *Visual C#*, *Visual Studio* juga dapat mendukung bahasa pemrograman lain seperti *M*, *python* dan *ruby* yang semuanya itu terdapat pada pack extra yang terpisah dari *visual studio*. (Sumber : *Edy Winarno ST, M.Eng, Ali Zaki, SmitDev Community (2010:8)*, dalam buku "*Web Programming dengan Visual Basic 2010*").

2.8. MySQL

Sebuah basis data mempunyai fungsi untuk mengoleksi banyak data sedangkan data didefinisikan sebagai deskripsi tentang benda, kejadian, aktivitas dan transaksi, yang tidak mempunyai makna atau tidak berpengaruh secara langsung kepada pemakainya, sedangkan menurut *McFadden*, dkk dalam kadir, mendefinisikan informasi sebagai data yang telah diproses sedemikian rupa sehingga meningkatkan pengetahuan seseorang yang menggunakan data tersebut.

Saat ini sudah menjadi pengetahuan umum bahwa banyak sekali aplikasi pada organisasi *small medium business (SMB)* yang di bangun dengan menggunakan sistem basis data, karena faktor pentingnya data dan informasi tersebut, maka diperlukan suatu mekanisme untuk mereplikasi atau menggandakan sistem basis data sehingga jika terjadi suatu keadaan yang termasuk ke dalam *force majeure*, organisasi tersebut tidak akan mengalami kehilangan data, replikasi basis data ini melibatkan paling sedikit 2 sistem basis data yang terhubung dalam suatu jaringan komputer.

MySQL adalah sebuah database manajemen system (DBMS) populer yang memiliki fungsi sebagai relational database manajemen system (RDBMS), Selain itu *MySQL software* merupakan suatu aplikasi yang sifatnya open source serta server basis data *MySQL* memiliki kinerja sangat cepat, reliable, dan mudah untuk digunakan serta bekerja dengan arsitektur *client server* atau *embedded systems*. Dikarenakan faktor *open source* dan populer tersebut maka cocok untuk mendemonstrasikan proses replikasi basis data.

2.9. Mengenal XAMPP

XAMPP dari *Apache*, *MySQL*, *PHP* dan *Perl* adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kumpulan dari beberapa program XAMPP memiliki arti sebagai berikut:

1. Huruf X mengandung arti bahwa perangkat lunak pemrograman ini dapat dijalankan di banyak sistem operasi seperti windows, linux, mac os dan solaris.
2. Huruf A merupakan singkatan dari *Apache* merupakan suatu perangkat lunak aplikasi *webserver*. Tugas utama *apache* adalah menghasilkan halaman *web* yang benar kepada *user* berdasarkan kode *PHP* yang dituliskan oleh pembuat halaman web. *Webserver* ini juga memiliki kemampuan untuk melakukan pengolahan database (input, proses dan output) *MySQL* dengan berdasarkan kode *PHP* yang diketikan oleh pembuatnya.
3. Huruf P merupakan singkatan dari *PHP* (<http://id.wikipedia.org/wiki/php>). Pada awalnya *PHP* merupakan singkatan dari *Personal Home Page* yang pertama kali dibuat oleh Ramus Lerdorf pada tahun 1995. Pada waktu itu *PHP* masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari web. Kemudian pada tahun 1977, sebuah perusahaan bernama zend menulis ulang skrip ini menjadi lebih bersih, lebih baik dan lebih cepat. Pada Juni 1998, perusahaan tersebut merilis interpreter baru untuk *PHP* dan meresmikan rilis tersebut sebagai *HPH 3.0* dan singkatan *PHP* diubah

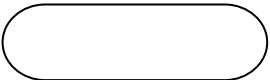
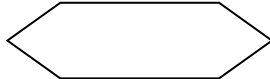

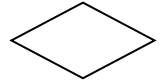
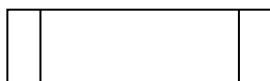
menjadi akronim berulang PHP Hypertext Preprocessing. Pada Juni 2004, zend kembali merilis PHP 5.0 yang merupakan penyempurnaan dari PHP 4.0 yang dirilis pada tahun 1999. Dalam versi ini inti dari interpreter PHP mengalami perubahan besar. Versi ini juga memasukan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan Bahasa pemrograman ke arah paradig berorientasi objek.



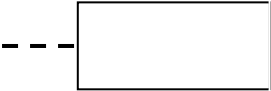
2.10. Flowchart

Sistem flowchart merupakan alat yang banyak digunakan untuk menggambarkan sistem secara fisik, simbol-simbol yang digunakan dalam sistem flowchart, Berikut adalah simbol-simbol program flowchart menurut ANSI (*American National Standard Institute*), yakni:

Tabel 2.2. Flowchart

Sumber : Jogiyanto; 2016: 95

	Terminal Untuk menunjukkan awal dan akhir program.
	Persiapan (<i>preparation</i>) Untuk memberikan nilai awal pada suatu variabel atau <i>counter</i> .
	Pengolahan (<i>processing</i>) Untuk pengolahan aritmatika dan pemindahan data.
	Keputusan (<i>decision</i>) Untuk mewakili operasi perbandingan logika.
	Proses terdefinisi (<i>predefined process</i>) Untuk proses yang detailnya dijelaskan terpisah, misalnya dalam bentuk subroutine.

	<p>Penghubung (<i>connector</i>)</p> <p>Untuk menunjukkan hubungan arus proses yang terputus masih dalam halaman yang sama.</p>
	<p>Penghubung halaman lain (<i>off page connector</i>)</p> <p>Untuk menunjukkan hubungan arus proses yang terputus masih dalam halaman yang sama.</p>
	<p>Penjelasan (<i>annotation flag</i>)</p> <p>Untuk memberikan keterangan-keterangan guna memperjelas simbol-simbol yang lain.</p>

BAB III

METODE PENELITIAN

3.1. Waktu dan Tempat Penelitian

Waktu yang digunakan dalam penelitian ini yaitu dalam kurun waktu Delapan bulan terhitung mulai bulan Juli 2020 sampai Maret 2020. Tempat yang digunakan dalam penelitian ini adalah Laboratorium Komputer Fakultas Sains dan Teknologi Universitas Islam Negeri Sumatera Utara yang berlokasi di Jalan IAIN No. 1, Gaharu, Kec. Medan Timur, Kota Medan, Sumatera Utara 20235.

3.2. Alat dan Bahan Penelitian

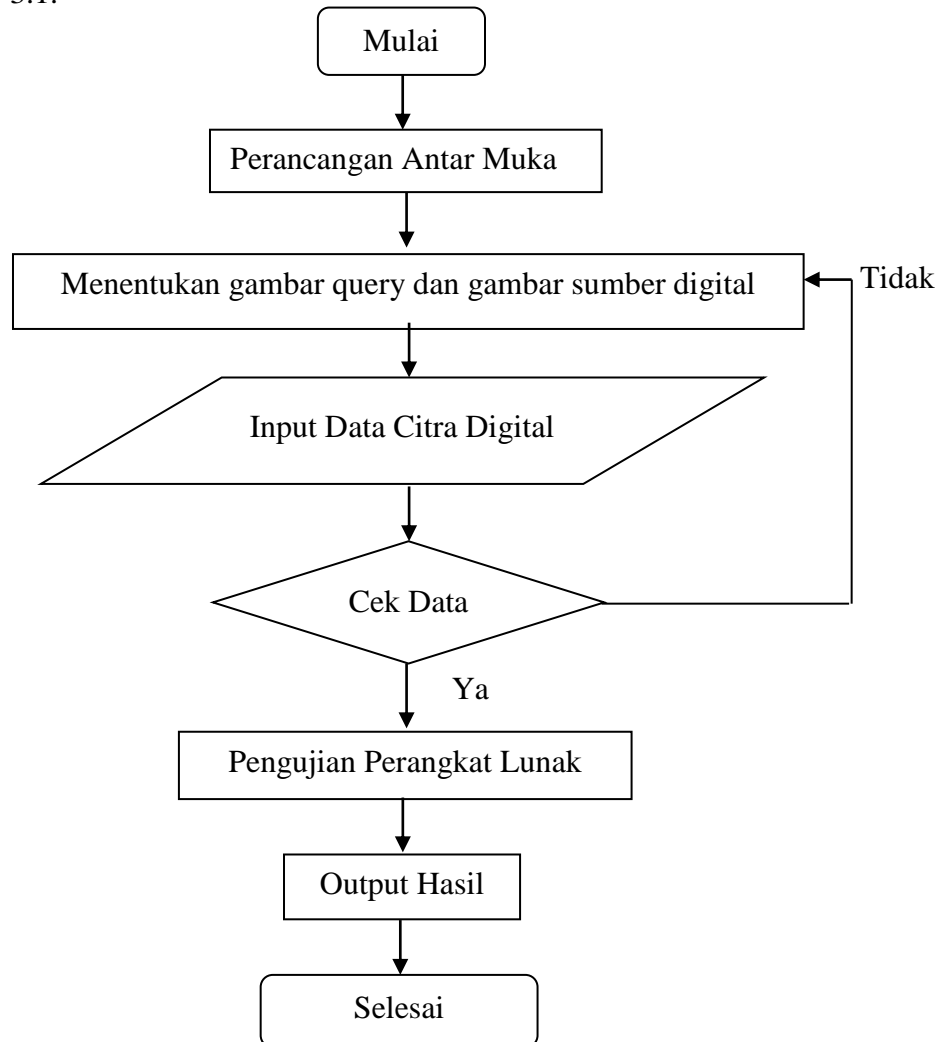
Alat yang digunakan dalam penelitian ini yaitu seperangkat komputer dan perangkat lunak visual studio 2010 dan XAMPP. Bahan yang akan digunakan dalam penelitian ini yaitu gambar berextensi BMP yang terdiri dari 24 bit, 256 color dan 16 color dan untuk tahapan pengujian di laboratorium UIN menggunakan gambar query dan gambar sumber sebagai perbandingan kemiripan gambar.

3.3. Prosedur Penelitian

Penelitian dilakukan dengan menyiapkan gambar atau citra bitmap yang terdiri dari 24 bit, 256 color dan 16 color. Selanjutnya gambar tersebut dicari proses dengan mengimplementasi dari pengolahan citra yang mempunyai kemiripan karakteristik atau *content* dari sekumpulan gambar. Secara umum proses yang dilakukan dengan proses ekstraksi fitur. Parameter fitur yang dapat digunakan untuk retrieval dapat berupa histogram, susunan warna, tekstur, bentuk, tipe spesifik dari objek tersebut.

Perangkat lunak sistem kemiripan gambar dirancang menggunakan gambar query dan gambar sumber dengan metode CBIR (*Content Based Image Retrieval*) dan algoritma yang digunakan Shingling. Data yang telah diuji dengan perangkat lunak dibandingkan untuk mencari nilai kemiripan dan sisa dari perbandingan gambar tersebut.

Prosedur penelitian yang akan dilakukan digambar dengan diagram pada Gambar 3.1.



Gambar 3.1. Flowchart Prosedur Penelitian

Dari Gambar 3.1 dapat dilihat gambaran dari sistem temu kembali citra ini terdapat 3 proses pada sistem. Pertama fitur citra *query* baik fitur warna maupun fitur tekstur diekstraksi dimana fitur citra diekstraksi dengan metode *Colour Histogram* dan fitur tekstur diekstraksi dengan metode *Grey Level Co-Occurrence Matrices*. Lalu yang kedua vektor fitur citra *query* dihitung jaraknya dengan semua vektor fitur citra yang ada di dalam database menggunakan algoritma *shingling*. Setelah itu yang ketiga jarak hasil perhitungan tersebut diurutkan dari yang paling kecil ke yang paling besar (*ascending*), Citra yang ada di dalam

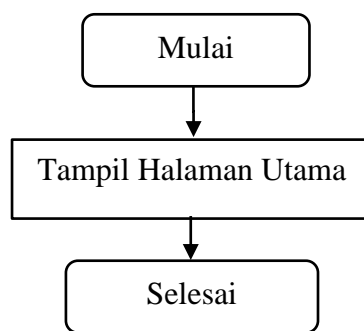
database yang memiliki nilai jarak vektor fitur yang paling kecil dengan vektor fitur citra *query* merupakan citra yang paling.

3.4. Desain Sistem

Untuk melakukan perancangan sistem yang akan di bangun peneliti degan menggunakan flowchart.

1. Flowchart Halaman Utama

Diagram alir halaman utama yang dirancang dapat dilihat pada flowchart di bawah ini :

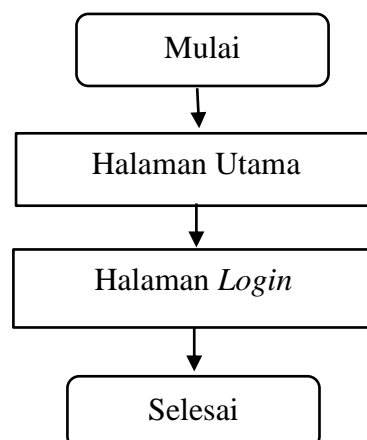


Gambar 3.2 Flowchart Halaman Utama

Dari Gambar 3.2 dapat dilihat gambaran dari kemiripan gambar, pertama kali aplikasi dijalankan langsung yang tampil adalah halaman utama, jika halaman utama ini tidak dapat tampil maka aplikasi yang dijalankan gagal menuju halaman berikutnya.

2. Flowchart Masuk ke Halaman *Login*

Diagram alir halaman *login* yang dirancang dapat dilihat pada flowchart di bawah ini :

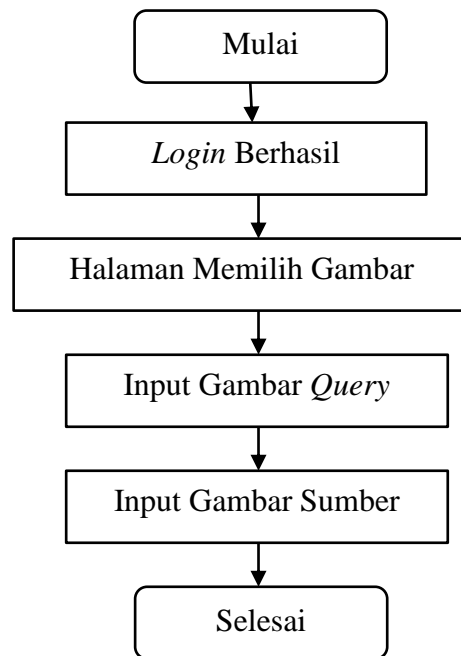


Gambar 3.3. Flowchart Masuk ke Halaman *Login*

Dari Gambar 3.3 dapat dilihat gambaran dari kemiripan gambar untuk masuk ke halaman *login*, pertama kali aplikasi dijalankan langsung yang tampil halaman utama, di halaman utama tersebut sudah disediakan halaman *login* dan *user* harus menggunakan inputan yang ada di halaman *login* untuk masuk ke halaman berikutnya.

3. Flowchart Masuk ke Halaman Memilih Gambar

Diagram alir halaman memilih gambar dari database yang dirancang dapat dilihat pada flowchart di bawah ini :

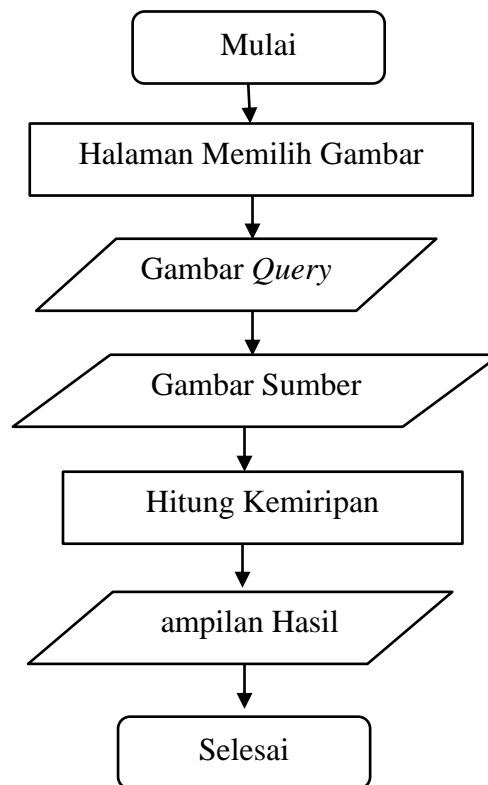


Gambar 3.4Flowchart Memilih Gambar

Dari gambar 3.5 dapat dilihat gambaran dari memilih gambar untuk masuk ke halaman *login*, jika halaman *login* telah berhasil maka masuk halaman memilih gambar, di halaman memilih gambar terdapat dua komponen yaitu *query* dan sumber, disini *user* menginputkan gambar *query* dan dibandingkan dengan gambar di sumber.

4. Flowchart Masuk ke Halaman Menghitung Kemiripan Gambar

Diagram alir halaman memilih gambar dari database yang dirancang dapat dilihat pada flowchart di bawah ini :



Gambar 3.5. Flowchart Halaman Menghitung Kemiripan Gambar

Dari Gambar 3.6 dapat dilihat gambaran dari halaman menghitung kemiripan gambar, jika gambar telah berhasil memasukan gambar baik itu di *query* ataupun di sumber maka langkah terakhir adalah melihat hasil dari kemiripan gambar tersebut dengan cara melakukan eksekusi dari hitung kemiripan.

3.5. Perancangan Antarmuka

Salah satu aspek penting dalam pembuatan perangkat lunak adalah perancangan antarmuka, karena perancangan antar muka yang baik berbanding lurus dengan tingkat *user friendly* sebuah perangkat lunak artinya sistemanya dirancang sedemikian rupa agar pemakai dapat beradaptasi dengan mudah dalam pemakaian perangkat lunak tersebut. Perancangan antar muka dalam penelitian ini

dibuat menggunakan GUI vb.net. Perancangan antar muka dalam penelitian ini terdiri dari perancangan form halaman judul perangkat lunak, perancangan form *login*, perancangan Perbandingan kemiripan gambar, perancangan memilih gambar, perancangan login dan perancangan pengujian kemiripan gambar perangkat lunak.

1. Perancangan Form Judul Perangkat Lunak

Perancangan antar muka form halaman judul perangkat lunak digambarkan pada gambar 3.10.

Form Judul	
Judul Aplikasi Perangkat Lunak	Gambar Utama

Gambar 3.6. Form Judul Perangkat Lunak

2. Perancangan Form *Login* Perangkat Lunak

Perancangan antar muka form halaman *login* perangkat lunak digambarkan pada gambar 3.7.

Form <i>Login</i>	
<p><i>Username:</i> <input type="text"/></p> <p><i>Password:</i> <input type="text"/></p> <p><input type="button" value="Login"/> <input type="button" value="Keluar"/></p>	Gambar

Gambar 3.7. Form *Login* Perangkat Lunak

3. Perancangan Form Perbandingan Kemiripan Gambar

Perancangan antar muka form perbandingan kemiripan gambar, digambarkan pada gambar 3.9.

Form Gambar

Gambar A (*Query*)

Gambar B (Sumber)

Gambar 3.8. Form Perbandingan Kemiripan Gambar

4. Perancangan Form Memilih Gambar

Perancangan antarmuka form memilih gambar, digambarkan pada gambar 3.9.

Form Gambar

Gambar A (*Query*)

Button Memilih

Gambar B (Sumber)

Button Memilih

Gambar 3.9. Form Memilih Gambar

5. Perancangan Form Pengujian Kemiripan Gambar

Perancangan antarmuka form pengujian kemiripan gambar, digambarkan pada gambar 3.10.

Form Pengujian	
<div style="border: 1px solid black; width: 150px; height: 100px; margin: 0 auto;"></div> <p style="text-align: center;">Gambar A (<i>Query</i>)</p> <div style="border: 1px solid black; width: 100px; height: 20px; margin: 0 auto;"></div> <p style="text-align: center;">Label Pixel Label File</p>	<div style="border: 1px solid black; width: 150px; height: 100px; margin: 0 auto;"></div> <p style="text-align: center;">Gambar B (Sumber)</p> <div style="border: 1px solid black; width: 100px; height: 20px; margin: 0 auto;"></div>
<div style="border: 1px solid black; width: 100px; height: 20px; margin: 0 auto;"></div> <p style="text-align: center;">Label Persentasi Label Sisa</p>	

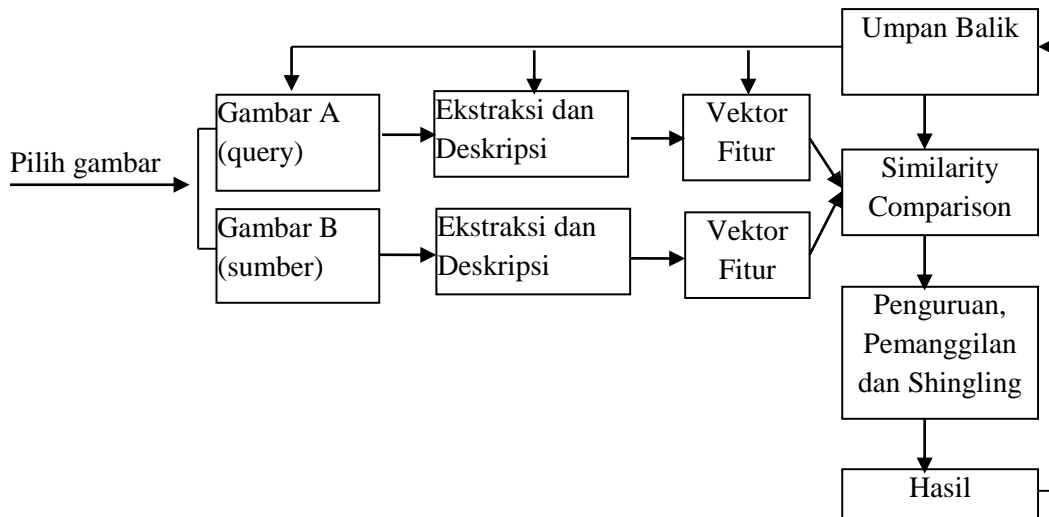
Gambar 3.10. Form Pengujian Kemiripan Gambar

4. Persiapan Gambar Digital

Sampel gambar yang digunakan yaitu gambar 24 bit, 256 color, dan 16 color yang diperoleh dari hasil foto. Gambar yang berasal dari foto digital merupakan data primer yang didapatkan dari sumber tersebut. Proses gambar digital menggunakan foto digital dilakukan untuk mendapatkan gambar asli atau gambar query berdasarkan gambar kamera digital dalam hasil bentuk file extension bitmap atau BMP. File extension ini dapat dikelola dengan aplikasi adobe photoshop atau paint brush. Pada proses perubahan dengan photoshop atau paint brush dapat dibandingkan dengan foto asli dari kamera digital dengan *height* dan *width* pixel tersebut.

5. Pengolahan CIBR

Gambar diolah menggunakan teknik pengolahan CBIR dan algoritma. Proses pengolahan gambar CBIR dapat diterangkan pada gambar 3.11.



Gambar 3.11. Pengolahan CBIR

6. Pengujian

Pengujian perangkat lunak akan dilakukan dengan metode CBIR dan algoritma yang digunakan shingling. Hasil pembacaan gambar query akan dibandingkan dengan data gambar digital yang sudah diidentifikasi pola sinyalnya.

Mengetahui apakah program kemiripan gambar dapat mengidentifikasi dari kedua gambar, maka diperlukan pengujian program. Terdapat proses pengujian yaitu pengujian dengan CBIR dan pengujian dengan Shingling dan tahapan identifikasi akhir. Pada pengujian dari kedua tersebut, banyaknya neuron masukan, neuron tersembunyi dan neuron keluaran telah ditetapkan sebelumnya. Banyak neuron masukan pada gambar *query* dan gambar sumber.

Aplikasi dengan presentasi kemiripan suatu gambar dalam bentuk *file* gambar dengan algoritma *shingling* yang telah dirancang agar layak dan sesuai dengan hasil yang diharapkan.

BAB IV

HASIL DAN PEMBAHASAN

4.1. Hasil

Hasil ini adalah untuk menjelaskan sistem yang dirancang, dan penjabaran akan dibagi menjadi tiga bagian, yaitu perhitungan manual, gambaran flowchart dan tampilan hasil antarmuka pemakai.

4.1.1. Perhitungan Manual

Proses perhitungan dari prosedur kemiripan gambar dengan metode CIBR dengan algoritma shingling dapat dijabarkan sebagai berikut:

1. Proses pembentukan *shingle*

Proses ini dilakukan sebelum proses embedding dilakukan. Proses ini bertujuan untuk mencari lokasi penyisipan yang memiliki bit-bit paling mirip dengan bit pesan yang akan disisipkan. Proses pembentukan *shingle* dilakukan dengan urutan:

- a. Panjang Shingle adalah 8 bit maka shingle terbentuk 8 byte.
- b. Pixel berukuran 8x4 piksel, maka total piksel adalah $8 \times 4 = 32$ piksel. Di dalam 32 piksel terdapat 32×3 warna = 96 warna, diberi nomor urut dari 0 sampai dengan 95.
- c. Pembentukan Shingle

Nomor Urut	Bit
Shingle 0	{0, 1, 2, 3, 4, 5, 6, 7}
Shingle 1	{ 1, 2, 3, 4, 5, 6, 7, 8}
Shingle 2	{ 2, 3, 4, 5, 6, 7, 8, 9}
Shingle 3	{ 3, 4, 5, 6, 7, 8, 9, 10}
Shingle 4	{ 4, 5, 6,7, 8, 9, 10, 11}
Shingle 5	{ 5, 6,7, 8, 9, 10, 11, 12}
Shingle 6	{ 6,7, 8, 9, 10, 11, 12, 13}

Shingle 7	{ 7, 8, 9, 10, 11, 12, 13, 14}
Begitu seterusnya sampai shingle 95	
Shingle 94	{ 94, 95, 1, 2, 3, 4, 5, 6}
Shingle 95	{ 95, 1, 2, 3, 4, 5, 6, 7}

Pada akhir proses, akan terbentuk sejumlah S *shingle*. Nilai S bergantung pada jumlah warna pada gambar terdapat 96 warna, maka di akhir proses terbentuk 96 *shingle*, masing-masing berisi 8 warna yaitu { 95, 1, 2, 3, 4, 5, 6, 7}.

2. Proses pemilihan citra pada gambar B

Proses pencarian lokasi penyisipan terbaik dilakukan dengan cara membandingkan tiap *shingle* dengan bit pesan.

1. Baca *shingle* ke-0.
2. Baca tiap-tiap warna di dalam *shingle* tersebut.

Shingle 0 berisi 255, 10, 20, 255, 160, 155, 200, 160

Maka dapat dirincikan sebagai berikut

Warna	Nilai	Biner
R	255	11111111
G	10	00001010
B	20	00010100
R	255	11111111
G	160	10100000
B	155	10011011
R	200	11001000
G	160	10100000
B	155	10011011
R	200	11001000
G	0	00000000
B	0	00000000
R	100	01100100
G	0	00000000
B	255	11111111

3. Proses ekstraksi, diketahui potongan piksel pada suatu gambar hasil penyisipan adalah:

G 200

B 200

R 100

G 1

B 255

R 255

G 160

B 154

R 200

G 160

B 155

R 200

G 1

B 1

R 255

G 10

B 20

Warna	Nilai	Biner	Posisi
R	255	11111111	0
G	10	00001010	1
B	20	00010100	2
R	255	11111111	3
G	160	10100000	4
B	155	10011011	5
R	200	11001000	6
G	160	10100000	7
B	155	10011011	8
R	200	11001000	9

G	0	00000000	10
B	0	00000000	11
R	100	01100100	12
G	0	00000000	13
B	255	11111111	14

Berdasarkan kode posisi 9, panjang pesan 8, baca warna mulai posisi ke 9, 10, dan seterusnya sampai terkumpul 8 *bit*. Hasilnya adalah:

Warna	Nilai	Biner	Posisi	Penyisipan
R	255	11111111	0	1
G	10	00001010	1	0
B	20	00010100	2	
R	255	11111111	3	
G	160	10100000	4	
B	155	10011011	5	
R	200	11001000	6	
G	160	10100000	7	
B	155	10011011	8	
R	200	11001000	9	0
G	0	00000000	10	0
B	0	00000000	11	0
R	100	01100100	12	0
G	0	00000000	13	0
B	255	11111111	14	1

Maka yang diperoleh nilainya adalah 10000001

1. Input gambar query di gambar A



Gambar 4.1 Gambar Pembentukan *Shingle* Rika Amalia.bmp

2. Baca warna elemen RGB dari setiap piksel query gambar A pada file gambar Rika.bmp.

217,15,46	239,106,49	60,45,2	114,120,244	186,146,138
95,182,178	209,136,13	142,253,82	113,4,164	14,126,153
248,24,181	246,233,108	171,100,106	190,73,242	238,150,84
151,192,14				

3. Baris pertama (baris paling atas) akan digunakan untuk menyimpan jumlah karakter pada *ciphertext*. Jumlah karakter dibatasi maksimal 255 karakter, yang berarti bit panjang ciphertext yang akan disisipkan adalah sebesar 8 bit. Panjang shingle = 4 karakter.

Nilai ini akan dikonversikan ke biner menjadi 0000 0100.

Sisipkan setiap bit ke dalam elemen warna RGB dari setiap piksel.

Piksel (1, 1)

$$217 \quad = 1101\ 1001 \quad = 11011001 \quad = \mathbf{217}$$

$$15 \quad = 0000\ 1111 \quad = 0000\ 1100 \quad = \mathbf{15}$$

$$46 \quad = 00101110 \quad = 00101110 \quad = \mathbf{46}$$

Piksel (1, 2)

$$238 = 11101110 = 11101111 = \mathbf{239}$$

$$107 = 01101011 = 01101010 = \mathbf{106}$$

$$49 = 00110001 = 00110001 = \mathbf{49}$$

Piksel (1, 3)

$$6 = 00111101 = 00111100 = \mathbf{60}$$

$$45 = 00101101 = 00101101 = \mathbf{45}$$

$$2 = 0000\ 0010 = 2$$

4. Proses pembentukan shingle dimulai dari nomor urut 0. Dari langkah 1 diketahui bahwa panjang shingle adalah 8, maka *shingle* yang pertama berisi warna-warna dari nomor urut 0 sampai dengan 7.

$$\text{Shingle } 0 = \{ 0, 1, 2, 3, 4, 5, 6, 7 \}$$

Bit yang akan disisipkan:

00001010000101001010000010011011

Piksel (2, 1)

$$95 = 01011110 = 01011111 = \mathbf{95}$$

$$182 = 10110110 = 10110110 = \mathbf{182}$$

$$178 = 10110010 = 10110010 = \mathbf{178}$$

Piksel (2, 2)

$$209 = 11010001 = 11010001 = \mathbf{209}$$

$$137 = 10001000 = 10001000 = \mathbf{136}$$

$$13 = 0000\ 1101 = 0000\ 1101 = \mathbf{13}$$

Piksel (2, 3)

$$141 = 10001101 = 10001110 = \mathbf{142}$$

$$252 = 11111100 = 11111101 = \mathbf{253}$$

$$83 = 01010011 = 01010010 = \mathbf{82}$$

Piksel (2, 4)

$$113 = 01110001 = 01110001 = \mathbf{113}$$

$$4 = 0000\ 0100 = 0000\ 0100 = \mathbf{4}$$

$$165 = 10100101 = 10100100 = \mathbf{164}$$

Piksel (2, 5)

$$15 = 0000\ 1111 = 0000\ 111\mathbf{0} = \mathbf{14}$$

$$127 = 01111111 = 0111111\mathbf{0} = \mathbf{126}$$

$$152 = 10011000 = 1001100\mathbf{1} = \mathbf{153}$$

Piksel (3, 1)

$$249 = 11111001 = 1111100\mathbf{0} = \mathbf{248}$$

$$25 = 00011001 = 0001100\mathbf{0} = \mathbf{24}$$

$$181 = 10110101 = 1011010\mathbf{1} = \mathbf{181}$$

Piksel (3, 2)

$$246 = 11110110 = 1111011\mathbf{0} = \mathbf{246}$$

$$232 = 11101000 = 1110100\mathbf{1} = \mathbf{233}$$

$$108 = 01101100 = 0110110\mathbf{0} = \mathbf{108}$$

Piksel (3, 3)

$$171 = 10101011 = 1010101\mathbf{1} = \mathbf{171}$$

$$101 = 01100101 = 0110010\mathbf{0} = \mathbf{100}$$

$$106 = 01101010 = 0110101\mathbf{0} = \mathbf{106}$$

Piksel (3, 4)

$$191 = 10111111 = 1011111\mathbf{0} = \mathbf{190}$$

$$72 = 01001000 = 0100100\mathbf{1} = \mathbf{73}$$

$$242 = 11110010 = 1111001\mathbf{0} = \mathbf{242}$$

Piksel (3, 5)

$$239 = 11101111 = 1110111\mathbf{0} = \mathbf{238}$$

$$151 = 10010111 = 1001011\mathbf{0} = \mathbf{150}$$

$$85 = 01010101 = 0101010\mathbf{0} = \mathbf{84}$$

Piksel (4, 1)

$$151 = 10010111 = 1001011\mathbf{1} = \mathbf{151}$$

$$193 = 11000001 = 1100000\mathbf{0} = \mathbf{192}$$

$$14 = 0000\ 1110 = 0000\ 1110 = 14$$

5. Tampilkan citra Rika bitmap di gambar sumber atau gambar B yang diperoleh

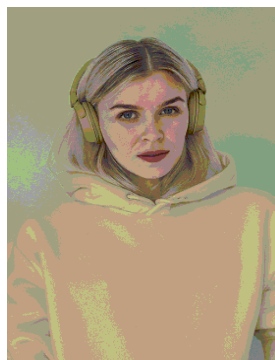


Gambar 4.2 Citra Pemilihan Shingle Rika Amalia

217,15,46	239,106,49	60,45,2	114,120,244	186,146,138
95,182,178	209,136,13	142,253,82	113,4,164	14,126,153
248,24,181	246,233,108	171,100,106	190,73,242	238,150,84
151,192,14				

Setelah diperoleh citra pemilihan Rika Amalia, maka file citra Rika ini akan menjadi perbandingan. Kemudian, perbandingan akan mengekstraksi data dari citra Rika tersebut. Proses kerja dari ekstraksi data dari citra Rika dapat dirincikan sebagai berikut:

1. Input citra pembentukan Gambar B



Gambar 4.3 Citra Pembentukan Shingle Gambar_2.bmp

181,167,132	66,56,51	104,67,65	133,100,68	100,100,69
132,155,132	140,132,68	169,132,132	100,100,4	192,160,128
192,192,128	64,32,0	128,128,64	32,32,64	160,160,128
96,96,64				

2. Ekstraksi panjang *ciphertext* dari baris 1 elemen warna RGB piksel citra stego.

Piksel (1, 1)

181 = 1101 1000

167 = 0000 1100

132 = 0010 1110

Piksel (1, 2)

66 = 1110 1110

56 = 0110 1010

51 = 0011 0001

Piksel (1, 3)

104 = 0011 1100

67 = 0010 1100

65 =

Bit terekstrak = 0000 0100 = 4, berarti panjang shingle adalah 4 karakter. Hal ini berarti bahwa harus diekstrak $4 * 8 = 32$ bit dari citra Gambar_2.bmp.

3. Ekstrak bit shingle mulai dari baris 2 elemen warna RGB piksel citra Gambar_2.bmp.

Piksel (2, 1)

94 = 0101 1110

183 = 1011 0111

178 = 1011 0010

Piksel (2, 2)

208 = 1101 0000

136 = 1000 1000

12 = 0000 1100

Piksel (2, 3)

141 = 1000 1101

253 = 1111 1101

82 = 0101 0010

Piksel (2, 4)

113 = 0111 0001

4 = 0000 0100

164 = 1010 0100

Piksel (2, 5)

14 = 0000 1110

126 = 0111 1110

153 = 1001 1001

Piksel (3, 1)

248 = 1111 1000

24 = 0001 1000

181 = 1011 0101

Piksel (3, 2)

246 = 1111 0110

233 = 1110 1001

108 = 0110 1100

Piksel (3, 3)

171 = 1010 1011

100 = 0110 0100

106 = 0110 1010

Piksel (3, 4)

190 = 1011 1110

73 = 0100 1001

242 = 1111 0010

Piksel (3, 5)

238 = 1110 1110

150 = 1001 0110

84 = 0101 0100

Piksel (4, 1)

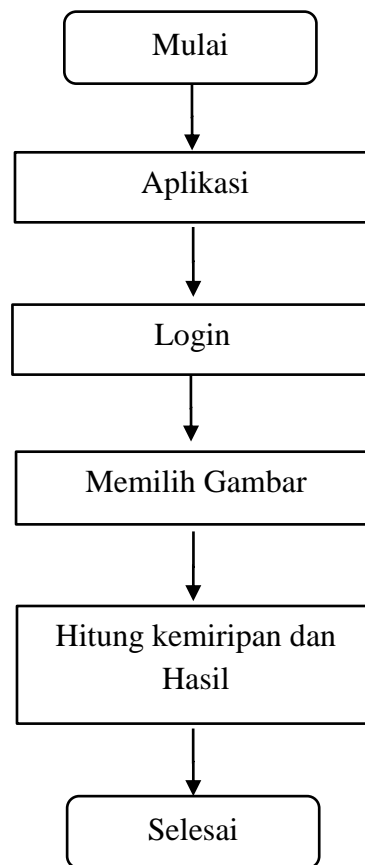
96 = 1001 0111

96 = 1100 0000

Bit terekstrak: 0000011101010010010001

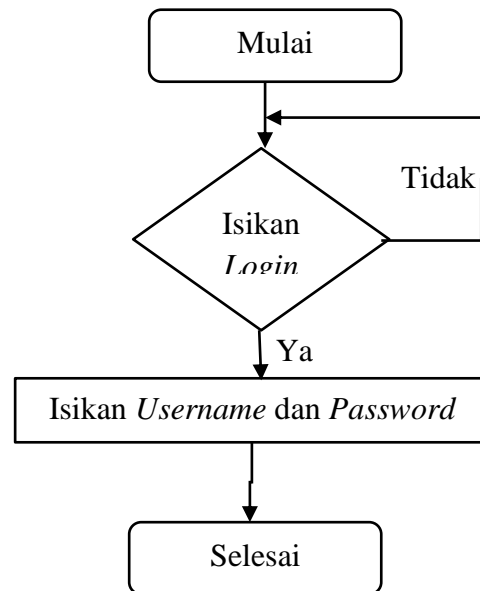
4.1.2. Flowchart

Proses kerja dari aplikasi dapat digambarkan dalam bentuk flowchart seperti terlihat pada gambar berikut:



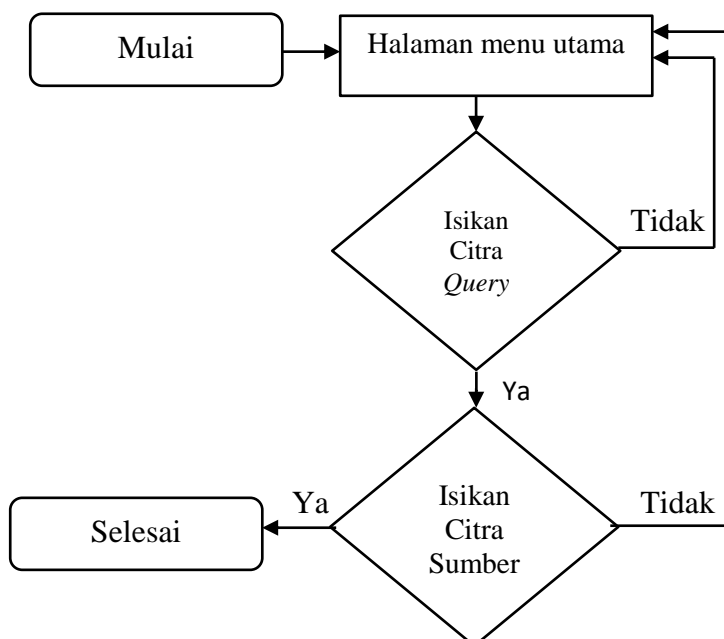
Gambar 4.4Flowchart Proses Kerja

Sementara itu, proses kerja dari prosedur login dapat digambarkan dalam bentuk *flowchart* seperti terlihat pada gambar berikut



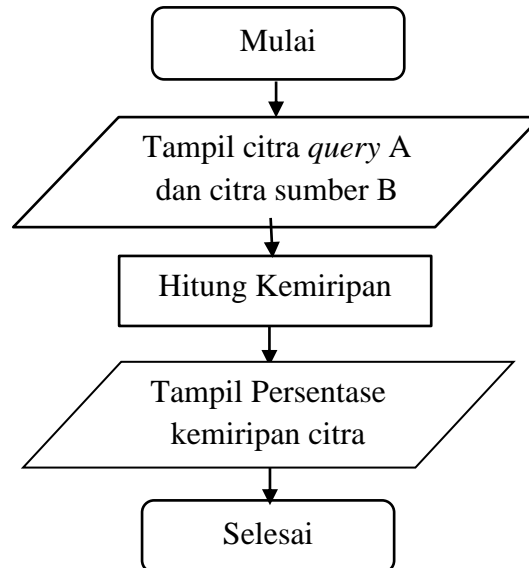
Gambar 4.5 Tampilan Flowchart Proses Login

Proses kerja memasukan gambar di halaman kemiripan citra, bentuk flowchart seperti terlihat pada gambar berikut :



Gambar 4.6 Tampilan Flowchart Proses Memasukan Gambar

Proses kerja dari proses hitung kemiripan gambar dengan metode CBIR dengan algoritma *shingling*, dapat digambarkan dalam bentuk flowchart seperti terlihat pada gambar berikut



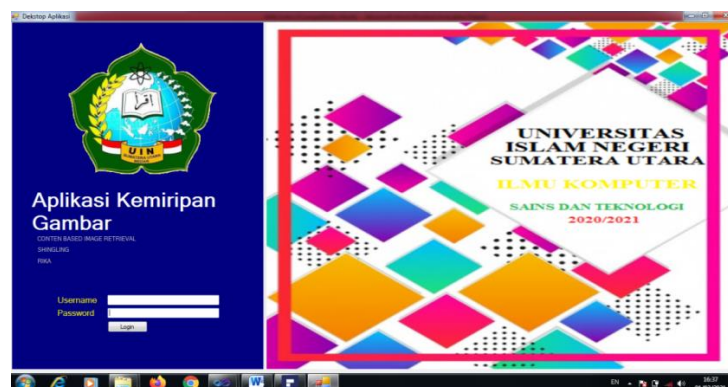
Gambar 4.7 Tampilan Flowchart Proses Hitung Kemiripan Citra

4.2. Tampilan Hasil Antarmuka Pemakai

Berikut adalah tampilan hasil dan pembahasan dari sistem kemiripan gambar metode CBIR menggunakan algoritma *shingling*. Dalam aplikasi ini pengguna dapat lebih mudah memahaminya perbedaan dari kedua gambar.

a. Tampilan Halaman Utama Kemiripan Gambar

Tampilan halaman utama ini adalah awal pemula yang tampil dari aplikasi sistem membandingkan kemiripan gambar, untuk lebih jelasnya dapat dilihat pada gambar 4.1.



Gambar 4.8 Tampilan Halaman Utama Kemiripan Gambar

Tampilan halaman utama pada gambar 4.1 ini adalah menampilkan desktop aplikasi kemiripan gambar metode CBIR algoritma *shingling*, dengan adanya tampilan desktop ini maka pengguna dapat melanjutkan ke halaman berikutnya.

b. Tampilan Halaman *Login*

Tampilan halaman *login* ini adalah awal pemula yang tampil dari aplikasi sistem membandingkan kemiripan gambar, untuk lebih jelasnya dapat dilihat pada gambar 4.2.

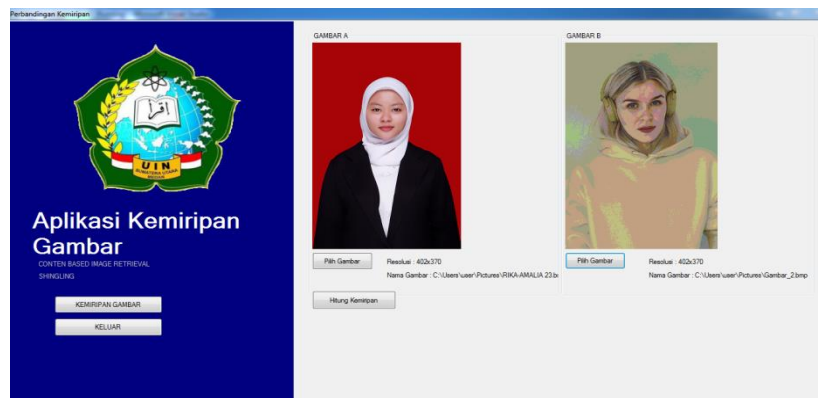


Gambar 4.9 Tampilan Halaman *Login*

Tampilan halaman login pada gambar 4.2 ini adalah menampilkan dalam pengisian *username* dan *password*, jika sudah di masukan *username* dan *password* maka selanjutnya user klik tombol *login*.

c. Tampilan Halaman Kemiripan Kedua Gambar

Tampilan halaman perbandingan ini untuk memasukan gambar ke gambar A dan ke gambar B, untuk lebih jelasnya dapat dilihat pada gambar 4.3.



Gambar 4.10. Tampilan Halaman Kemiripan Kedua Gambar

Tampilan gambar 4.3 menampilkan halaman perbandingan dari kedua gambar yaitu memilih gambar di dalam group gambar A dan memilih gambar di dalam group gambar B. Setelah gambar di masukan ke masing-masing group yang sudah disediakan maka user dapat melihat informasi resolusi gambar dan nama file gambar beserta lokasi diambil gambar tersebut.

d. Tampilan Halaman Hitung Kemiripan Gambar

Tampilan halaman hitung kemiripan gambar terdapat dalam satu form yaitu form perbandingan kemiripan atau disebut dengan form2. Untuk lebih jelasnya bentuk jalur tersebut dapat dilihat pada pada gambar 4.4.



Gambar 4.11 Tampilan Halaman Hitung Kemiripan Gambar

Menampilkan hasil dari perhitungan kemiripan gambar Setelah user mengklik tombol hitung kemiripan gambar maka algoritma shingling akan mengkalkulasinya dan pengguna dapat melihat berapa persen kemiripan gambar dan berapa selisih dari kemiripan gambar.

4.3. Pencocokan Citra Untuk Menentukan Kematangan

Dalam Penelitian ini, pencocokan citra acuan dengan citra uji menggunakan metode CBIR algoritma shingling untuk mencari ketidaksamaan citra dari gambar sumber terhadap gambar query.

Tabel 4.1. Tabel Kemiripan Gambar

No.	Gambar A	Gambar B	Kemiripan	Selisih
1.	Rika Amalia.bmp	Gambar_2.bmp	100%	0%
2.	Rika Amalia.bmp	Rika.bmp	6 %	94%

4.1.3 Pembahasan

Sistem aplikasi kemiripan gambar ini menggunakan aplikasi visual studio 2010 menggunakan bahasa pemrograman vb.net, semua desain aplikasi dibuat di dalam aplikasi ini dengan menggunakan tool yang tersedia di visual studio 2010 yaitu komponen textbox, label, opendialog, picturebox, groupbox, panel. button.

Hasil aplikasi Sistem kemiripan gambar ini bertujuan untuk membandingkan dari kedua gambar yang terdapat di dalam picture A dan picture yang sudah disediakan dalam memilih gambar.

Penelitian ini penulis terfokus dalam pengujian gambar dalam tipe gambar BMP karena mencocokkan atau mencari kemiripan 24 bit, 128 bit dan 16 bit. Dari gambar tersebut maka nantinya didapat berapa hitungan algoritma shingling kemiripan dan selisih kemiripan gambar tersebut.

Hasil dari penelitian penulis memilih gambar yang bertipe 24 bit dan memilih gambar 16 bit dengan nama file Rika.bmp maka didapat hasil perhitungan metode cibr menggunakan algoritma shingling yaitu dengan kemiripan 37.1 % dan selisih dari kemiripan kedua gambar dengan nilai 62.9 persen.

4.1.4 Kelebihan dan Kekurangan Aplikasi yang Dirancang

Aplikasi yang dirancang mempunyai beberapa kelebihan dan kekurangan ketika diterapkan diantaranya :

a. Kelebihan dari sistem yang dirancang :

- 1) Aplikasi kemiripan gambar ini yang dirancang khusus melakukan pengujian kemiripan dari kedua gambar khusus dengan resolusi bitmap (BMP).
- 2) Aplikasi ini dilengkapi dengan sistem metode CBIR dengan menggunakan algoritma *shingling* untuk mencari kemiripan gambar dan selisih kemiripan gambar.

b. Kekurangan dari sistem yang dirancang :

- 1) Gambar yang dibandingkan hanya resolusi BMP yaitu 24 bit, 128 bit dan 16 bit
- 2) Aplikasi ini tidak disediakan pengaturan zoom dalam pengaturan gambar.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Dari uraian secara teoritis pada sistem kemiripan gambar, maka penulis akan mencoba menarik kesimpulan dan akan memberikan saran-saran semoga bermanfaat:

1. Fitur tekstur yang diekstraksi dari citra menggunakan *Grey Level Co-Occurrence Matrices* dapat digunakan untuk *Content Based Image Retrieval*.
2. Gambar yang dapat dicari kemiripan berupa data resolusi bitmap BMP 24 bit, 128 bit dan 16 bit.
3. Proses temu kembali menggunakan fitur tekstur diekstraksi dengan shingling lebih cepat dibandingkan proses temu kembali menggunakan fitur warna diekstraksi dengan *Colour Histogram*, 1,723

5.2. Saran

Untuk menyempurnakan sistem yang telah dibuat ini penulis memberikan saran:

1. Untuk penelitian selanjutnya dapat dilakukan kombinasi antara fitur warna dengan fitur tekstur untuk mendapatkan efektifitas temu kembali yang lebih baik.
2. Penelitian ini hanya dapat melakukan temu kembali citra digital yang memiliki format BMP sehingga untuk pengembangan selanjutnya diharapkan dapat melakukan temu kembali citra digital untuk format file citra digital lainnya seperti JPG, GIF, atau PNG.
3. Penelitian ini hanya menggunakan algoritma shingling, penelitian selanjutnya dapat dilakukan dengan beberapa algoritma yang lain.

DAFTAR PUSTAKA

- Ardiansyah, M, Y., 2019 “Cara kerja algoritma *shingling*” dalam jurnal Perancangan dan Implementasi Aplikasi Deteksi Kemiripan Dokumen Menggunakan Algoritma Shingling dan MD5 Fingerprint, Universitas Kristen Satya Wacana Salatiga
- Baba, F. J., 2016 “*Algoritma Shingling*” dalam jurnal Perancangan dan Implementasi Aplikasi Steganografi pada Image PNG dengan Algoritma Shingling Vol 1, No 1, Universitas Kristen Satya Wacana Salatiga
- Dahlan, Ahmad., “*MySQL*” dalam jurnal Perancangan Replikasi basis data MySQL Dengan Mekanisme Pengamanan Menggunakan SSL Encryption, Vol 8, No1, Universitas Dahlan.
- Faradilla Dkk, 2017 “Sistem pencarian data terdiri dari dua kategori yaitu *text base* dan *content based*” dalam jurnal analisis dan simulasi *Content Based Image Retrieval* (CBIR) berdasarkan ciri tekstur menggunakan metode walvelet: Teknik Telekomunikasi, Universitas Telkom
- Furqan Mhd, Halim, A., H., R.H, 2002 “ *Digital Image Enhancement Using The Method Of Multiscale Retinex And Median Filter*” .
- Halim, Arwin., dkk, 2016, “pemahaman aplikasi *Content Based Image Retrieval* yang digunakan untuk pengambilan query image dari sebuah arsip gambar yang besar” dalam jurnal Aplikasi *Content Based Image Retrieval* Dengan Fitur Warna Dan Bentuk Vol 15, No 2. Medan: Program Studi Teknik Informatika, STMIK Mikroskil.
- Halim, Arwin., dkk, 2017, “contoh penerapan *Content Based Image Retrieval* dalam kehidupan sehari-hari pada bidang medis” dalam jurnal Aplikasi *Content Based Image Retrieval* Dengan Fitur Warna Dan Bentuk Vol 15, No 2 Medan.
- Harsono, D, H., 2018 “Citra (image) adalah representasi optis dari sebuah objek yang disinari oleh sebuah sumber radiasi” dalam jurnal Pengembangan aplikasi untuk perbaikan citra digital film radiografi ISSN 1978-0176 Yogyakarta
- <http://digital.cs.usu.edu/~xqi/Teaching/REU07/Notes/CBIR.pdf>), diakses tanggal 20 Februari 2021
- Jogiyanto, 2016 “Sistem flowchart merupakan alat yang banyak digunakan untuk menggambarkan sistem secara fisik” dalam jurnal analisis dan desain sistem informasi pendekatan terstruktur teori dan praktek, Andi Offset, Yogyakarta.

- Mulyanta, S. E., 2016 “Digital image merupakan objek nyata yang direpresentasikan secara elektronis” Dalam buku Mulyanta Edi S. Sri Special Workshop: Teknik Airbrush Menggunakan Adobe Photoshop CS2/Edi S Mulyanta – Ed I. Yogyakarta: Andi, ISBN: 979-763-038-2
- Putra, D. 2019. Pengolahan Citra Digital. Yogyakarta: Penerbit ANDI
- Putri, R. A., 2016 “Image processing atau pengolahan citra merupakan suatu metode atau teknik yang dapat digunakan untuk memproses citra atau gambar” dalam jurnal pengolahan citra dengan menggunakan web cap pada kendaraan bergerak di jalan raya, Vol 1, No 1, Tulungagung
- R. Munir., 2020. Pengolahan Citra Digital dengan Pendekatan Algoritmik. Informatika: Bandung.
- Rahman, A. 2020. Sistem Temu Balik Citra Menggunakan Jarak Histogram Dalam Model Warna YIQ. *Seminar Nasional Aplikasi Teknologi Informasi 2020*. (Online) <http://journal.uii.ac.id/index.php/Snati/article/download/1564/1337> (19 Desember 2020)
- Ramadijanti, N., 2017 “*Content Based Image Retrieval* merupakan teknik untuk mencari suatu gambar dengan membandingkan gambar query dengan gambar yang ada di data base, CBIR secara umum dibangun dengan karakteristik” dalam jurnal *Content Based Image Retrieval* berdasarkan citra tekstur menggunakan wavelet : Politeknik elektronika negri surabaya.
- Sutoyo, T. & Mulyanto, E. 2021. Teori Pengolahan Citra Digital. Yogyakarta : Penerbit ANDI.
- Taufik, I., 2018 “*Content Based Image Retrieval* aplikasi pengolahan citra yang dapat membantu penggunaan mengambil atau mencari dengan cepat” dalam jurnal *Metode content based image retrieval (CBIR)* untuk pencarian gambar yang sama menggunakan perbandingan histogram warna RGB. Vol 17, No 1 Medan: Program Studi Teknik Informatika, STMIK Pelita Nusantara.
- Winamo, E., ST, M. Zak, A., *SmitDev Community 2010* “Visual Basic” dalam buku “*Web Programming dengan Visual Basic 2010*”
- Yusuf R, N., Furqan, M., Sinaga., 2020 “Implementasi Steganografi Menggunakan Metode *Spread Spectrum* Dalam Pengamanan Data Teks Pada Citra Digital.
- Zhang, Y. 2020. On the use of CBIR in Image Mosaic Generation. (Online). <http://webdocs.cs.ualberta.ca/~zaiane/postscript/thesis/YueZhang2002.pdf> (20 Desember 2014)

Lampiran

Kode Program yang digunakan untuk membuat Penerapan *Content Based Image Retrieval* (CBIR) Dengan Algoritma Shingling pada Aplikasi Kemiripan Gambar

```
Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
    dr = openDR("SELECT * FROM tabel_login WHERE
username='" & TextBox1.Text & "' AND password='" &
TextBox2.Text & "'")
    If dr.Read Then
        user = dr("username")
        Form2.Visible = True
        Me.Visible = False
        ' Me.Close()
    Else
        msgError("Salah kombinasi username dan
password")
        TextBox1.Focus()
    End If
End Sub
End Class

Public Class Form2

    Private GambarA As Class1
    Private GambarB As Class1
    Private namaFileSumber, namafileQuery As String

    Private selisihG() As Double
    Private selisihN(,,) As Double
    Private nilaiBulat As Double

    Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
```

```

        Panel2.Visible = True
    End Sub

    Private Sub Button3_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button3.Click
        Dim OpenFileDialog1 As New OpenFileDialog

        'Setup the open dialog.
        OpenFileDialog1.FileName = ""
        OpenFileDialog1.Title = "Pilih Gambar BMP"
        OpenFileDialog1.InitialDirectory = "C:\"
        OpenFileDialog1.Filter = "Picture Files
()|*.bmp"

        If OpenFileDialog1.ShowDialog =
Windows.Forms.DialogResult.OK Or
Windows.Forms.DialogResult.Yes Then

            GambarA = New Class1(New
Bitmap(OpenFileDialog1.FileName))
            GambarA>NamaGambar =
OpenFileDialog1.SafeFileName
            namafileQuery = OpenFileDialog1.FileName
            PictureBox2.Image = GambarA.Gambar

            Label4.Text = "Resolusi : " &
PictureBox2.Width & "x" & PictureBox2.Height
            Label6.Text = "Nama Gambar : " &
OpenFileDialog1.FileName

        End If

    End Sub

    Public Function BulatkanSelisih(ByVal
histogramSelisih(,,) As Double) As Double
        Dim nilai As Double = 0

```

```

        For i = 0 To 6
            For j = 0 To 3
                For k = 0 To 3
                    nilai += histogramSelisih(i, j,
k)

                Next
            Next
        Next

        Return nilai * 100 / 200

    End Function

    Public Function CariSelisihHistogramGrafik(ByVal
histogramA() As Double, ByVal histogramB() As Double)
As Double()
        Dim histogramSelisih() As Double = histogramA
        For i As Integer = 0 To UBound(histogramA)
            histogramSelisih(i) =
Math.Abs(histogramA(i) - histogramB(i))
        Next
        Return histogramSelisih
    End Function

    Public Function
CariSelisihHistogramNormalisasi(ByVal histogramA(,,)
As Double, ByVal histogramB(,,) As Double) As
Double(,,)
        Dim histogramSelisih(6, 3, 3) As Double
        For i = 0 To 6
            For j = 0 To 3
                For k = 0 To 3
                    histogramSelisih(i, j, k) =
Math.Abs(histogramA(i, j, k) - histogramB(i, j, k))
                Next
            Next
        Next
        Return histogramSelisih
    End Function

```



```

        Return histogramSelisih
    End Function

    Private Sub Button4_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button4.Click
        Dim OpenFileDialog1 As New OpenFileDialog

        'Setup the open dialog.
        OpenFileDialog1.FileName = ""
        OpenFileDialog1.Title = "Pilih Gambar BMP"
        OpenFileDialog1.InitialDirectory = "C:\"
        OpenFileDialog1.Filter = "Picture Files
()|*.bmp"

        If OpenFileDialog1.ShowDialog =
Windows.Forms.DialogResult.OK Or
Windows.Forms.DialogResult.Yes Then

            PictureBox3.Image =
Image.FromFile(OpenFileDialog1.FileName) '
            PictureBox3.Name =
OpenFileDialog1.SafeFileName

            GambarB = New Class1(New
Bitmap(OpenFileDialog1.FileName))
            namaFileSumber = OpenFileDialog1.FileName
            GambarB>NamaGambar =
OpenFileDialog1.SafeFileName
            PictureBox3.Image = GambarB.Gambar

            Label5.Text = "Resolusi : " &
PictureBox2.Width & "x" & PictureBox2.Height
            Label8.Text = "Nama Gambar : " &
OpenFileDialog1.FileName

        End If

```

```

End Sub

Private Sub Button2_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button2.Click

    selisihG = CariG(GambarA.AmbilHistogramGraph,
GambarB.AmbilHistogramGraph)
    selisihN = CariN(GambarA.AmbilHistogram2,
GambarB.AmbilHistogram2)
    nilaiBulat = BulatkanSelisih(selisihN)

    Label7.Text = "Persentase Kemiripan Adalah "
    If nilaiBulat <> 0 Then
        Label10.Text = "" & 100 -
Math.Round(nilaiBulat, 0) & "%, dengan selisih " &
Math.Round(nilaiBulat, 0) & "%" & " Total : 100%"
    Else
        Label10.Text = "" & 100 -
Math.Round(nilaiBulat, 0) & "%"
    End If
End Sub

Private Sub Button5_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button5.Click

    Form1.Visible = True
    Me.Close()
End Sub
End Class
Public Class Class1
    Private vGambar As Bitmap
    Dim hsvarray(,) As HSV
    Private histogram As Class2
    Private infoGambar As Info

    Sub New(ByVal imgRGB As Bitmap)
        vGambar = imgRGB

```

```

        buatHSV()
        histogram = New Class2(hsvarray)
End Sub

ReadOnly Property AmbilHSV()
    Get
        Return hsvarray
    End Get
End Property

Public Sub buatHSV()
    ReDim Preserve hsvarray(vGambar.Width,
vGambar.Height)
    For x = 0 To vGambar.Width - 1
        For y = 0 To vGambar.Height - 1
            hsvarray(x, y) = rgb2hsv(ambilRGB(x,
y, vGambar))
        Next
    Next
End Sub

Public Function rgb2hsv(ByVal cin As RGB) As HSV

    'sebagai r,g,b aksen
    Dim _R, _G, _B As Double

    'sebagai warna max, warna min, dan delta
    Dim cMax, cMin, delta As Double

    _R = cin.R / 255
    _G = cin.G / 255
    _B = cin.B / 255

    cMax = Math.Max(_R, _G)
    cMax = Math.Max(cMax, _B)

    cMin = Math.Min(_R, _G)
    cMin = Math.Min(cMin, _B)

```

```

        delta = cMax - cMin
        If cMax = _R Then
            rgb2hsv.H = Math.Round(Math.Abs(60 *
modulusku(((G - B) / delta), 6)), 0)
        ElseIf cMax = _G Then
            rgb2hsv.H = Math.Round(Math.Abs(60 *
((B - R) / delta) + 2)), 0)
        ElseIf cMax = _B Then
            rgb2hsv.H = Math.Round(Math.Abs(60 *
((R - G) / delta) + 4)), 0)
        End If

        If Double.IsNaN(rgb2hsv.H) = True Then
            rgb2hsv.H = 0
        End If

        If cMax = 0 Then
            rgb2hsv.S = 0
        Else
            rgb2hsv.S = Math.Round(delta / cMax, 3) *
100
        End If

        rgb2hsv.V = Math.Round(cMax, 3) * 100
    End Function

    Private Function modulusku(ByVal number As
Double, ByVal modNumber As Double) As Double
        If Math.Abs(number) < Math.Abs(modNumber)
Then
            modulusku = Math.Abs(modNumber) -
Math.Abs(number)
        Else
            modulusku = number Mod modNumber
        End If
    End Function

```

```

        ReadOnly Property AmbilHistogramGraph() As
Double()
            Get
                Return histogram.AmbilHistogramGraph
            End Get
        End Property
        ReadOnly Property AmbilHistogram1()
            Get
                Return histogram.AmbilHistogram
            End Get
        End Property

        ReadOnly Property AmbilHistogram2()
            Get
                Return
histogram.AmbilHistogramNormalisasi
            End Get
        End Property

        ReadOnly Property Gambar()
            Get
                Return vGambar
            End Get
        End Property

        Function BacaInfoGambar() As Info
            infoGambar.Lebar = vGambar.Width
            infoGambar.Tinggi = vGambar.Height
            BacaInfoGambar = infoGambar
        End Function

        WriteOnly Property NamaGambar() As String
            Set(ByVal value As String)
                infoGambar>NamaGambar = value
            End Set
        End Property
    End Class
Public Class Class2

```

```

Private imgKuantisasi(,) As HSV
Private imgHistogram(6, 3, 3) As Integer
Private imgHistogramNormalisasi(6, 3, 3) As
Double
Private HistogramGraph(112) As Double

Sub New(ByVal gambarHSV(,) As HSV)
    Kuantisasi(gambarHSV)
    buatHistogramDanNormalisasi()
End Sub

Private Sub buatHistogramDanNormalisasi()
    Dim x As Integer = UBound(imgKuantisasi, 1)
    Dim y As Integer = UBound(imgKuantisasi, 2)
    Dim resolusi As Integer = x * y

    Dim n As Integer = 0
    For i As Integer = 0 To 6
        For j As Integer = 0 To 3
            For k As Integer = 0 To 3
                imgHistogramNormalisasi(i, j, k)
= imgHistogram(i, j, k) * 100 / resolusi
                HistogramGraph(n) =
imgHistogramNormalisasi(i, j, k)
                n += 1
            Next
        Next
    Next
End Sub

Private Sub Kuantisasi(ByVal hsvimg(,) As HSV)
    ReDim Preserve imgKuantisasi(UBound(hsvimg,
1), UBound(hsvimg, 2))
    For x As Integer = 0 To UBound(hsvimg, 1) - 1
        For y As Integer = 0 To UBound(hsvimg, 2)
- 1
            imgKuantisasi(x, y) =
KuantisasiPiksel(hsvimg(x, y))

```

```

        With imgKuantisasi(x, y)
            imgHistogram(.H, .S, .V) += 1
        End With
    Next
Next
End Sub

ReadOnly Property AmbilHistogramGraph() As
Double()
    Get
        Return HistogramGraph
    End Get
End Property

ReadOnly Property AmbilHistogram() As Integer(,,)
    Get
        Return imgHistogram
    End Get
End Property

ReadOnly Property AmbilHistogramNormalisasi() As
Double(,,)
    Get
        Return imgHistogramNormalisasi
    End Get
End Property
End Class

Public Class Class3
    Private vGambar As Bitmap
    Private mGambar(,) As RGB

    Sub New(ByVal gambar As Bitmap)
        vGambar = gambar
        buatRGB()
    End Sub

    Public Structure RGB
        Dim R As Double

```

```

        Dim G As Double
        Dim B As Double
    End Structure

    Public Structure Info
        Dim Lebar As String
        Dim Tinggi As String
        Dim NamaGambar As String
    End Structure

    Public Sub buatRGB()
        ReDim Preserve mGambar(vGambar.Width,
vGambar.Height)
        For x As Integer = 0 To vGambar.Width - 1
            For y As Integer = 0 To vGambar.Height -
1
                mGambar(x, y).R = vGambar.GetPixel(x,
y).R
                mGambar(x, y).G = vGambar.GetPixel(x,
y).G
                mGambar(x, y).B = vGambar.GetPixel(x,
y).B
            Next
        Next
    End Sub

    Property setGambar() As Bitmap
        Get
            Return vGambar
        End Get
        Set(ByVal value As Bitmap)
            vGambar = value
        End Set
    End Property

    ReadOnly Property AmbilRGB() As RGB(,)
        Get
            Return mGambar

```



```

        End Get
    End Property

    Function InfoGambar() As Info
        InfoGambar.Lebar = vGambar.Width
        InfoGambar.Tinggi = vGambar.Height
        InfoGambar>NamaGambar = ""
    End Function
End Class
Public Module Class4
    Public Structure RGB
        Dim R As Double
        Dim G As Double
        Dim B As Double
    End Structure
    Public Structure HSV
        Dim H As Double
        Dim S As Double
        Dim V As Double
    End Structure

    Public Structure Info
        Dim Lebar As String
        Dim Tinggi As String
        Dim>NamaGambar As String
    End Structure

Module Class5
    Public Function ambilRGB(ByVal x As Integer,
ByVal y As Integer, ByVal Vgambar As Bitmap) As RGB
        ambilRGB.R = Vgambar.GetPixel(x, y).R
        ambilRGB.G = Vgambar.GetPixel(x, y).G
        ambilRGB.B = Vgambar.GetPixel(x, y).B
    End Function

    Public Function ambilHSV(ByVal x As Integer,
ByVal y As Integer, ByVal hsvArray(,) As HSV) As HSV
        ambilHSV = hsvArray(x, y)
    End Function

```

```

End Function

Public Function KuantisasiPiksel(ByVal nilaiHSV
As HSV) As HSV
    KuantisasiPiksel.H = Math.Round(nilaiHSV.H /
360 * 6, 0)
    KuantisasiPiksel.S = Math.Round(nilaiHSV.S /
100 * 3, 0)
    KuantisasiPiksel.V = Math.Round(nilaiHSV.V /
100 * 3, 0)
End Function

Public Function CariG(ByVal histogramA() As
Double, ByVal histogramB() As Double) As Double()
    Dim histogramSelisih() As Double = histogramA
    For i As Integer = 0 To UBound(histogramA)
        histogramSelisih(i) =
Math.Abs(histogramA(i) - histogramB(i))
    Next
    Return histogramSelisih
End Function

Public Function CariN(ByVal histogramA(,,) As
Double, ByVal histogramB(,,) As Double) As Double(,,)
    Dim histogramSelisih(6, 3, 3) As Double
    For i = 0 To 6
        For j = 0 To 3
            For k = 0 To 3
                histogramSelisih(i, j, k) =
Math.Abs(histogramA(i, j, k) - histogramB(i, j, k))
            Next
        Next
    Next
    Return histogramSelisih
End Function

Public Function BulatkanSelisih(ByVal
histogramSelisih(,,) As Double) As Double
    Dim nilai As Double = 0

```

```
For i = 0 To 6
    For j = 0 To 3
        For k = 0 To 3
            nilai += histogramSelisih(i, j,
k)
        Next
    Next
Next
Return nilai * 100 / 200
'Return nilai
End Function
End Module
End Module
```

DAFTAR RIWAYAT HIDUP
(*CURRICULUM VITE*)



Nama : Rika Amalia
Nim : 0701162040
Tempat, Tanggal Lahir : Perbaungan, 08 Januari 1998
Jenis Kelamin : Perempuan
Alamat : Dusun Lau Tepu A.
Kel/Desa : Lau Tepu A
Kecamatan : Salapian
Kabupaten : Langkat Sumatera Utara
Kode Pos : 20773
Agama : Islam
Status Nikah : Belum Menikah
Nama Orang Tua
 Ayah : Budi Hartono
 Ibu : Sri Murni



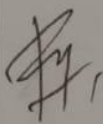


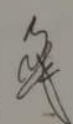




PENDIDIKAN FORMAL








2004-2010 : SD NEGRI 057961 Perk, Tambunan
2013-2016 : SMK Laksamana Martadinata Medan
2016-2021 : Universitas Islam Negeri Sumatera Utara Medan

KARTU BIMBINGAN SKRIPSI

Semester Gasal/Genap Tahun Akademik _____ / _____

Nama : <u>Rika Amalia</u>	Pembimbing I : <u>Dr. Mhd Fuzqan, S.Si., M.Comp. &</u>
NIM : <u>0701162040</u>	Pembimbing II : <u>Yusuf Ramadhan Nst, M.Kom</u>
Prog. Studi : <u>Ilmu Komputer</u>	SK Pembimbing :
Judul Skripsi : <u>Penerapan Metode Content Based Image Retrieval</u> <u>Dengan Algoritma Shingling Pada Aplikasi</u> <u>Kemiripan Gambar</u>	

P E R T	PEMBIMBING I			PEMBIMBING II		
	Tgl.	Materi Bimbingan	Tanda Tangan	Tgl.	Materi Bimbingan	Tanda Tangan
I	<u>4/ Juni 2020</u>	<u>Revisi BAB I</u>		<u>26/ Juni 2020</u>	<u>Bimbingan BAB I (Latar belakang)</u>	
II	<u>22/ Mei 2020</u>	<u>Revisi BAB II</u>		<u>10/ Mei 2020</u>	<u>Bimbingan BAB II (Tinjauan pustaka)</u>	
III	<u>8/ Okt</u>	<u>Revisi BAB II</u>		<u>6/ Okt 2020</u>	<u>Bimbingan BAB II (Cara kerja, Perancangan dan Perencanaan)</u>	
IV	<u>4/ des 2020</u>	<u>Pengecekan proposal skripsi</u>		<u>3/ des 2020</u>	<u>Pengecekan proposal skripsi</u>	
V	<u>9/ des 2020</u>	<u>ACC proposal skripsi</u>		<u>3/ des 2020</u>	<u>ACC proposal skripsi</u>	

VI	22/mar 2021	Bimbingan BAB IV		23/mar 2021	Revisi BAB IV BAB V	
VII	2/mar 2021	Pengecepan daftar pustaka		09/mar 2021	Acc. Pidang	
VIII	27/mar 2021	BAB IV, BAB V				
IX	27/mar 2021	ACC BAB IV, BAB V				
X	29/mar 2021	ACC BAB IV, BAB V sidang				

Medan,20.....
 An. Dekan
 Ketua Jurusan/Program Studi

NIP.

Catatan: Pada saat bimbingan, kartu ini harus diisi dan ditandatangani oleh pembimbing