

**APLIKASI DETEKSI GAMBAR HASIL SCAN MENGGUNAKAN  
METODE COSINE SIMILARITY BERBASIS MOBILE**

**SKRIPSI**

**DISUSUN OLEH**

**M.RIZKI SAPUTRA SARAGIH**

**NPM. 2009020134**



**UMSU**

Unggul | Cerdas | Terpercaya

**PROGRAM STUDI TEKNOLOGI INFORMASI  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA**

**MEDAN**

**2024**

## LEMBARAN PENGESAHAN

**Judul Skripsi** : APLIKASI DETEKSI GAMBAR HASIL SCAN  
MENGUNAKAN METODE COSINE SIMILARITY  
BERBASIS MOBILE

**Nama Mahasiswa** : M. RIZKI SAPUTRA SARAGIH

**NPM** : 2009020134

**Program Studi** : TEKNOLOGI INFORMASI

Menyetujui Komisi Pembimbing



(Farid Akbar Siregar S.Kom., M.Kom)  
NIDN. 0104049401

Ketua Program Studi



(Fatma Sari Hutagalung, S.Kom, M.Kom)  
NIDN. 0117019301

Dekan



(Dr. Al-Khowarizmi, S.Kom., M.Kom.)  
NIDN. 0127099201

## PERNYATAAN ORISINALITAS

APLIKASI DETEKSI GAMBAR HASIL SCAN MENGGUNAKAN METODE  
COSINE SIMILARITY BERBASIS MOBILE

### SKRIPSI

Saya menyatakan bahwa karya tulis ini adalah hasil karya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya.

Medan, Juli 2024

Yang membuat pernyataan



M. RIZKI SAPUTRA SARAGIH

NPM. 2009020134

**PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH  
UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademika Universitas Muhammadiyah Sumatera Utara, saya bertanda tangan dibawah ini:

Nama : M. Rizki Saputra Saragih  
NPM : 2009020134  
Program Studi : Teknologi Informasi  
Karya Ilmiah : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Muhammadiyah Sumatera Utara Hak Bebas Royalti Non-Eksekutif (*Non-Exclusive Royalty free Right*) atas penelitian skripsi saya yang berjudul:

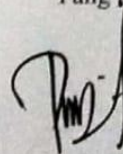
**APLIKASI DETEKSI GAMBAR HASIL SCAN MENGGUNAKAN  
METODE COSINE SIMILARITY BERBASIS MOBILE**

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksekutif ini, Universitas Muhammadiyah Sumatera Utara berhak menyimpan, mengalih media, memformat, mengelola dalam bentuk database, merawat dan mempublikasikan Skripsi saya ini tanpa meminta izin pada saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemegang dan atau sebagai pemilik hak cipta.

Demikian pernyataan ini dibuat dengan sebenarnya.

Medan, Juli 2024

Yang membuat Pernyataan



M. RIZKI SAPUTRA SARAGIH

NPM. 2009020134

## RIWAYAT HIDUP

### DATA PRIBADI

Nama Lengkap : M. Rizki Saputra Saragih  
Tempat dan Tanggal Lahir : Resak Estate, 13 Juni 2002  
Alamat Rumah : Margodadi  
Telepon/Faks/HP : 082387121437  
E-mail : riskisaput7@gmail.com  
Instansi Tempat Kerja : -  
Alamat Kantor : -

### DATA PENDIDIKAN

SD : SD 016 MAHONI  
TAMAT: 2014  
SMP : SMP MAHONI  
TAMAT: 2017  
SMA : SMK SWASTA BHAKTI  
AGRO MANDIRI  
TAMAT: 2020

## KATA PENGANTAR



Puji syukur atas kehadiran Allah SWT, berkat limpahan rahmat, hidayah dan karunianya, penulis bisa menyelesaikan skripsi dengan judul “**APLIKASI DETEKSI GAMBAR HASIL SCAN MENGGUNAKAN METODE COSINE SIMILARITY BERBASIS MOBILE**”. Skripsi ini adalah salah satu pada beberapa persyaratan untuk menyelesaikan pendidikan dan memperoleh gelar sarjana pada program studi S1 Teknologi Informasi di Universitas Muhammadiyah Sumatera Utara.

Penyusunan skripsi ini tidak lepas pada bimbingan, bantuan, arahan dan dukungan pada berbagai pihak terkait. Oleh karena itu pada kesempatan ini penulis menginformasikan terimakasih kepada:

1. Bapak Prof. Dr. Agussani, M.AP., Rektor Universitas Muhammadiyah Sumatera Utara (UMSU)
2. Bapak Dr. Al-Khowarizmi, S.Kom., M.Kom, Dekan Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
3. Ibu Fatma Sari Hutagalung, S.Kom., M.Kom, Ketua Program Studi Teknologi Informasi
4. Bapak Mhd. Basri, S.Si, M.Kom, Sekretaris Program Studi Teknologi Informasi
5. Pembimbing saya Bapak Farid Akbar Siregar, S.Kom,M.Kom
6. Ayahanda Husin Saragih Serta Ibunda Indrayani Panjaitan atas doa dan kasih sayangnya yang tulus dan tak terhingga kepada penulis, terima kasih selalu Sebagai *support sytem* terbaik sepanjang perjalanan hidup penulis yang selalu memberikan kasih sayang tak terhingga dengan penuh cinta, tulus dan selalu memberikan motivasi, semangat serta do'a yang terus mengalir dan tak pernah henti di setiap langkah hingga penulis mampu menyelesaikan studinya sampai sarjana.

7. Kakak-Kakak saya yang telah memberikan motivasi dan perhatiannya.
8. Seseorang yang bernama Tri ayunita selaku seseorang yang tak kalah penting kehadirannya. Yang selalu mendukung, menghibur, mendengarkan keluhan, serta memberikan tenaga dan waktunya kepada penulis hingga saat ini.
9. Kepada teman-teman seperjuangan saya terkhusus teman sekelas saya yang memberikan banyak bantuan, menemani dalam penelitian, memberikan motivasi dan telah menemani perjuangan saya dalam menyelesaikan skripsi ini.
10. Semua Pihak yang terlibat baik langsung maupun tidak langsung dalam pengerjaan Skripsi ini yang tidak penulis sebutkan satu persatu diucapkan terima kasih.

## ABSTRAK

Gambar diartikan suatu alat untuk berhubungan dalam menginformasikan pesan tertentu pada pembuat gambar, maka dari itu gambar dapat digunakan Sebagai bahasa dalam ilmu teknik. Sebuah gambar dapat disalin Sebagai banyak ketika sebuah file gambar masih ada. Akan tetapi ketika file gambar tidak ada misalnya ijazah maka harus melakukan foto ulang ataupun melakukan scan pada gambar yang masih ada dalam bentuk fisik. Masalah yang terjadi adalah gambar yang dibuat melalui foto ulang ataupun scan hasilnya tidak terlalu mirip. Itu lah sebabnya dibutuhkan sebuah cara yang dapat menguji kemiripan gambar maka dari itu mengetahui seberapa mirip gambar yang telah dibuat ulang. Penelitian ini menggunakan metode *Cosine Similarity* untuk deteksi gambar hasil *scan*. Nikmati respons yang lebih pintar, unggah fail dan gambar, dan lainnya.

**Kata Kunci: Aplikasi, Deteksi, Gambar, Scan, Metode Cosine Similarity, Mobile**



## **ABSTRACT**

Images are a tool for communicating to convey certain messages from the image maker, so that images can be used as a language in the technical field. An image can be copied into multiple copies while an image file still exists. However, if the image file is not there, for example a diploma, you have to take another photo or scan the image that is still there in physical form. The problem that occurs is that the results of the images created by re-photographing or scanning are not very similar. Therefore, we need a method that can test the similarity of images so that we know how similar the image that has been recreated is. This research uses the Cosine Similarity method to detect scanned images. With the implementation of the Cosine Similarity method, it can detect similarities and differences in images.

**Keywords: Application, Detection, Image, Scan, Cosine Similarity Method, Mobile**

## DAFTAR ISI

<b>LEMBAR PENGESAHAN</b> .....	<b>i</b>
<b>PERNYATAAN ORISINALITAS</b> .....	<b>ii</b>
<b>PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS</b> .....	<b>iii</b>
<b>RIWAYAT HIDUP</b> .....	<b>iv</b>
<b>KATA PENGANTAR</b> .....	<b>v</b>
<b>ABSTRAK</b> .....	<b>vii</b>
<b>DAFTAR ISI</b> .....	<b>ix</b>
<b>DAFTAR GAMBAR</b> .....	<b>x</b>
<b>DAFTAR TABEL</b> .....	<b>xi</b>
<b>BAB I</b> .....	<b>2</b>
<b>PENDAHULUAN</b> .....	<b>2</b>
1.1 Latar Belakang Masalah .....	2
1.2 Rumusan Masalah .....	4
1.3 Batasan Masalah .....	4
1.4 Tujuan Penelitian .....	4
1.5 Manfaat Penelitian .....	5
<b>BAB II</b> .....	<b>6</b>
<b>LANDASAN TEORI</b> .....	<b>6</b>
2.1 Aplikasi .....	6
2.2 Deteksi .....	6
2.3 Gambar .....	7
2.4 Scan.....	8
2.5 <i>Metode Cosine Similarity</i> .....	9
2.6 Mobile.....	11
2.7 Unified Modeling Language ( <i>UML</i> ).....	20
2.8 Penelitian Terdahulu .....	24
<b>BAB III</b> .....	<b>27</b>
<b>ANALISA DAN DESAIN SISTEM</b> .....	<b>27</b>
3.1 Analisa .....	27
3.2 Penerapan Metode .....	27
3.3 Desain Sistem .....	33
<b>BAB IV</b> .....	<b>43</b>
<b>HASIL DAN UJI COBA</b> .....	<b>43</b>
4.1 Tampilan Hasil .....	43
4.2 Pembahasan .....	45
4.2.1 Uji Coba Program.....	46
4.2.2 Hasil Uji Coba .....	47
4.3 Kesimpulan Hasil Pengujian .....	52
<b>BAB V</b> .....	<b>53</b>
<b>KESIMPULAN DAN SARAN</b> .....	<b>53</b>
5.1 Kesimpulan.....	53
5.2 Saran.....	53
<b>DAFTAR PUSTAKA</b> .....	<b>55</b>

## DAFTAR GAMBAR

Gambar 3. 1 <i>Flowchart</i> Metode <i>Cosine Similarity</i> .....	28
Gambar 3. 2 <i>Flowchart</i> Tahapan Penggunaan Perangkat Lunak.....	29
Gambar 3. 3 <i>Use Case</i> Aplikasi Deteksi Gambar Hasil Scan Menggunakan Metode <i>Cosine Similarity</i> Berbasis Mobile .....	33
Gambar 3. 4 <i>ActivityDiagram</i> Gambar Pertama.....	34
Gambar 3. 5 <i>ActivityDiagram</i> Gambar Kedua .....	35
Gambar 3. 6 <i>ActivityDiagram</i> <i>Cosine Similarity</i> .....	36
Gambar 3. 7 <i>Activity Diagram</i> Hasil .....	37
Gambar 3. 8 <i>Sequence Diagram</i> Gambar Pertama.....	38
Gambar 3. 9 <i>Sequence Diagram</i> Gambar Kedua .....	39
Gambar 3. 10 <i>SequenceDiagram</i> Metode .....	40
Gambar 3. 11 <i>Sequence Diagram</i> Hasil .....	41
Gambar 3. 12 Desain <i>User Interface</i> .....	42
Gambar 4. 1 Tampilan <i>Form</i> Citra Digital .....	43
Gambar 4. 2 Tampilan Upload Gambar Pertama .....	44
Gambar 4. 3 Tampilan Upload Gambar Pertama .....	44
Gambar 4. 4 Tampilan Upload Gambar Pertama .....	45

## DAFTAR TABEL

Tabel 2. 1 Versi Mobile Android .....	11
Tabel 2. 2 Simbol <i>Use Case</i> .....	20
Tabel 2. 3 Simbol <i>Activity Diagram</i> .....	22
Tabel 2. 4 Simbol <i>Sequence Diagram</i> .....	22
Tabel 2. 5 <i>Class Diagram</i> .....	23
Tabel 2. 6 Penelitian Terdahulu.....	24
Tabel 3. 1 Nilai RGB Gambar .....	31
Tabel 4. 1 <i>Blackbox Testing Form</i> Deteksi .....	46
Tabel 4. 2 <i>Blackbox Testing Form</i> Deteksi .....	47

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang Masalah**

Gambar diartikan sebagai media komunikasi yang berfungsi untuk menyampaikan pesan tertentu kepada pencipta gambar. Oleh karena itu, gambar dapat berperan sebagai bahasa dalam bidang teknik. Penyampaian informasi melalui gambar merupakan elemen kunci dalam proses pembuatan gambar (Rahmawati & Rahayu, 2021). Maka, gambar diharapkan mampu menyampaikan data dan informasi dengan jelas serta akurat. Informasi yang disampaikan dalam gambar tidak dapat disampaikan secara lisan, melainkan harus menggunakan simbol atau lambang tertentu. Dengan demikian, kualitas informasi dan data yang ditransmisikan akan bergantung pada keterampilan pembuat gambar. Suatu gambar dapat direproduksi berkali-kali selama file gambar tersebut masih tersedia. Namun, jika file gambar hilang, seperti dalam kasus ijazah, maka diperlukan pemotretan ulang atau pemindaian dari bentuk fisik gambar yang ada. Masalah yang muncul adalah hasil gambar dari pemotretan ulang atau pemindaian sering kali tidak seakurat aslinya. Oleh sebab itu, diperlukan metode untuk mengukur seberapa mirip gambar yang direproduksi.

Salah satu metode yang dapat diterapkan adalah dengan mengembangkan aplikasi untuk menguji kemiripan gambar. Hasil akhir dari penelitian ini adalah menentukan persentase kemiripan hasil pemindaian pada beberapa gambar yang diuji. Aplikasi ini dirancang dengan menggunakan pemrograman Android, yang

memungkinkan pengguna untuk mengunggah dua foto: satu foto asli dan satu foto hasil pemindaian. Aplikasi ini kemudian akan menampilkan hasil pemindaian dan menentukan apakah hasil tersebut berkualitas baik. Namun, diperlukan metode yang tepat untuk mendapatkan nilai kemiripan yang akurat.

Berdasarkan penelitian yang dilakukan oleh (Niko, 2022) tentang Implementasi Metode Levenshtein Distance dan Cosine Similarity untuk Deteksi Kemiripan Gambar, disimpulkan bahwa metode cosine similarity dapat digunakan untuk mendeteksi kesamaan gambar, dengan tingkat kemiripan sebesar 83% pada dua gambar yang dibandingkan. Penelitian oleh (Lindang et al., 2022) mengenai sistem penentuan kesamaan skripsi di perpustakaan menggunakan metode cosine similarity menyimpulkan bahwa metode ini efektif dalam mengukur tingkat kesamaan antara skripsi. Sementara itu, penelitian oleh (AZMI et al., 2020) tentang pengenalan wajah dalam keamanan login dengan menggunakan kombinasi algoritma Viola-Jones dan cosine similarity menunjukkan bahwa penerapan metode tersebut memberikan hasil yang baik dengan akurasi sebesar 77,20%. Dari 500 gambar yang digunakan sebagai sampel, 386 gambar berhasil diidentifikasi dengan benar.

Berdasarkan hasil-hasil penelitian sebelumnya yang telah berhasil memanfaatkan metode cosine similarity dalam berbagai kasus, penelitian ini juga mengadopsi metode tersebut untuk mendeteksi gambar hasil scan. Dengan penerapan metode cosine similarity, penelitian ini bertujuan untuk mengidentifikasi kemiripan dan perbedaan gambar. Berdasarkan pemaparan di latar belakang, peneliti kemudian menetapkan judul penelitian ini yaitu “**Aplikasi**

## **Deteksi Gambar Hasil Scan Menggunakan Metode Cosine Similarity Berbasis Mobile”.**

### **1.2 Rumusan Masalah**

Bagaimana cara mendeteksi kesamaan dan perbedaan pada gambar hasil scan yang terlihat serupa menggunakan metode Cosine Similarity sehingga tidak dapat mengelabui penglihatan manusia.

### **1.3 Batasan Masalah**

Batasan Masalah pada penelitian ini yaitu:

1. Penelitian hanya membahas mengenai deteksi hasil scan.
2. Metode yang digunakan adalah Cosine Similarity.
3. Input aplikasi adalah gambar hasil scan dengan format JPG dan PNG.
4. Output aplikasi adalah persentase hasil deteksi gambar.
5. Platform yang digunakan adalah Visual Basic Berbasis Mobile menggunakan bahasa pemrograman *basic*.
6. Desain aplikasi menggunakan pemodelan *Unified Modeling Language* (UML).

### **1.4 Tujuan Penelitian**

Menggunakan metode cosine similarity untuk mendeteksi kesamaan dan perbedaan pada gambar hasil scan yang tampak mirip tidak akan mengelabui indera penglihatan manusia.

## **1.5 Manfaat Penelitian**

Manfaat pada penelitian ini adalah sebagai berikut:

1. Dapat dengan mudah mengidentifikasi kesamaan dan perbedaan pada gambar yang di pindai.
2. Mengetahui dan memahami hasil penerapan metode cosine similarity untuk mendeteksi kesamaan dan perbedaan dalam gambar yang dipindai.
3. Mendapatkan wawasan dalam pembuatan aplikasi deteksi gambar hasil scan.



## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Aplikasi**

Aplikasi merupakan sebuah program yang siap digunakan dan dirancang untuk menjalankan fungsi tertentu, baik oleh pengguna maupun aplikasi lain, serta dapat diakses oleh target yang dituju. Berikut adalah beberapa pandangan terkait aplikasi: Aplikasi dapat diartikan sebagai penerapan, penyimpanan data, masalah, atau tugas dalam suatu media yang memungkinkan penerapan dalam bentuk baru (Ramadhan et al., 2021). Aplikasi juga adalah program yang dapat langsung menjalankan proses yang dibutuhkan pengguna di komputer. Selain itu, aplikasi didefinisikan sebagai kumpulan file spesifik yang mengandung kode program yang menghubungkan pengguna dengan perangkat keras komputer. Sistem, di sisi lain, didefinisikan sebagai jaringan yang terhubung antar entitas. Sistem informasi adalah representasi akurat dari informasi berbasis komputer. Informasi diibaratkan sebagai darah yang mengalir dalam tubuh manusia, menandakan betapa pentingnya informasi dalam sebuah organisasi (Sihombing & Yanris, 2020).

#### **2.2 Deteksi**

Deteksi merupakan salah satu teknik dalam kecerdasan buatan atau machine learning yang mampu mengenali gambar dengan cepat. Pada proses ini, gambar yang diambil akan diekstraksi untuk mendeteksi bagian wajah, menentukan apakah seluruh bagian wajah teridentifikasi atau tidak. Jika bagian wajah terdeteksi sepenuhnya, maka klasifikasi pengenalan wajah dianggap berhasil.

Penelitian ini bertujuan untuk mendeteksi wajah orang yang menggunakan masker maupun yang tidak menggunakan masker (Suhirja & Jemakmun, 2022).

### **2.3 Gambar**

Gambar diartikan sebagai bentuk media pembelajaran yang berfungsi untuk memperjelas pesan dalam situasi di mana ruang terbatas, seperti ketika objek terlalu besar, peristiwa telah terjadi di masa lalu, atau jaraknya terlalu jauh. Manfaat menggunakan gambar sebagai media pendidikan antara lain meliputi hal-hal berikut:

1. Media visual membantu siswa lebih mudah memahami pelajaran, terutama melalui penyajian gambar yang merepresentasikan kata-kata atau makna secara verbal.
2. Dengan menggunakan media visual, pengalaman anak-anak diperluas, persepsi mereka dipertajam, dan konsep-konsep mereka menjadi lebih komprehensif, sehingga dapat meningkatkan keinginan dan minat baru dalam proses belajar.
3. Gambar memberikan makna yang lebih mendalam terhadap pemahaman atau informasi. Kemampuan berpikir secara abstrak hanya bisa diperoleh melalui latihan dan didasarkan pada pengalaman-pengalaman sebelumnya dengan realitas yang nyata. Ketika seseorang melihat dan mendengar secara bersamaan saat menerima pelajaran, penerangan, atau penyuluhan, keraguan atau kesalahpahaman dapat dihindari secara efektif.

4. Gambar mampu mengatasi keterbatasan ruang dan waktu. Dengan menggunakan gambar, siswa dapat melihat objek yang berada jauh atau peristiwa yang terjadi di masa lalu (Rahmawati & Rahayu, 2021).

Penggunaan gambar sebagai media atau sumber pembelajaran di sekolah dasar belum menjadi praktik umum di kalangan guru. Hal ini disebabkan oleh berbagai tantangan dalam menyiapkan gambar, yang meskipun tidak terlalu rumit, bisa memerlukan waktu dan biaya untuk diperoleh. Seperti pepatah Tiongkok yang mengatakan, "Sebuah gambar bernilai seribu kata," gambar berperan sebagai alat pembelajaran yang membantu menjelaskan informasi, mengatasi keterbatasan ruang yang disebabkan oleh objek yang terlalu besar, atau menggambarkan peristiwa yang terjadi di masa lalu atau yang jaraknya jauh. Dengan gambar, siswa dapat mengubah ide-ide abstrak menjadi lebih konkret (Ali, 2021).

#### **2.4 Scan**

Pemindaian didefinisikan sebagai proses menyalin objek dalam dokumen, seperti surat kuasa, yang kemudian dapat dikirim melalui berbagai saluran untuk mengatasi kendala jarak antara pihak-pihak yang terlibat. Seiring waktu, berbagai tantangan telah muncul, termasuk situasi di mana surat kuasa dapat diragukan jika salah satu pihak berada di luar negeri dan istri memerlukan persetujuan suami untuk meminjam uang dari bank. Dalam kasus seperti ini, perjalanan yang terlalu jauh membuat istri harus mengirimkan surat kuasa yang telah dipindai dan diberi cap melalui email kepada suaminya untuk ditandatangani dan dikembalikan dengan tanda tangan lengkap dari suaminya (Pratiwi et al., 2020; Siregar et al., 2023).

Pemindaian, atau scanning, adalah proses konversi media melalui pemindaian dokumen yang kemudian menghasilkan data dalam bentuk gambar yang dapat disimpan di komputer. Proses ini dilakukan dengan menggunakan perangkat pemindai atau print scanner (Puspitadewi, 2021).

## 2.5 Metode Cosine Similarity

Metode Cosine similarity melibatkan dua konsep utama: "Cosine," yang berkaitan dengan sudut antara dua vektor dengan nilai dimensi, dan "Similarity," yang menunjukkan sejauh mana keduanya mirip satu sama lain (Niko, 2022). Cosine similarity mengukur kemiripan antara dua vektor berdimensi  $n$  dengan menghitung kosinus dari sudut di antara keduanya. Metode ini umum digunakan dalam penambangan teks untuk membandingkan dan menganalisis dokumen (Lindang et al., 2022). Cosine similarity berfungsi untuk menilai sejauh mana dua data saling mirip dengan membandingkan objek-objek tersebut dan menghitung tingkat kesamaannya. Metode ini mirip dengan jarak Euclidean, yang mengukur jarak antara dua data, namun perbedaannya terletak pada cara pengukuran; jarak Euclidean mengukur jarak langsung, sementara cosine similarity lebih fokus pada sudut antara data (AZMI et al., 2020). Cosine similarity digunakan untuk menentukan nilai kemiripan antara dua besaran. Berikut adalah rumus *cosine similarity*:

$$\text{Similarity}(A, B) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \sqrt{\sum_{i=1}^n (B_i)^2}} \dots\dots\dots(2.1)$$

Keterangan :

A = vektor

B = vektor

$A_i$  = bobot term  $i$  dalam blok  $A_i$

$B_i$  = bobot term  $i$  dalam blok  $B_i$

$\|A\|$  = determinan vektor  $A$

$\|B\|$  = determinan vektor  $B$

$i$  = jumlah term dalam kalimat

$n$  = jumlah vektor. (Siswanto & Ceng Giap, 2020).

$A$  merujuk pada bobot masing-masing fitur dalam vektor  $A$ , sedangkan  $B$  merujuk pada bobot fitur dalam vektor  $B$ . Dalam konteks pengambilan informasi,  $A$  adalah bobot setiap istilah dalam dokumen  $A$ , dan  $B$  adalah bobot setiap istilah dalam dokumen  $B$ . Penelitian ini menggunakan cosine similarity karena gambar dianggap sebagai data berdimensi tinggi. Dalam gambar, setiap piksel dianggap sebagai dimensi yang berbeda, dan nilai warna setiap piksel dianggap sebagai nilai dalam setiap dimensi. Pengukuran kemiripan dilakukan dengan membandingkan Dokumen 1 dan Dokumen 2. Sistem kemudian menghitung nilai kesamaan dengan menjumlahkan nilai  $A_i \cdot B_i$  dari istilah  $A$  dan istilah  $B$ , lalu menjumlahkan nilai  $A_i^2$  dari semua istilah dalam dokumen  $A$  dan  $B_i^2$  dari semua istilah dalam dokumen  $B$ , dan mengalikannya dua kali. Cosine similarity adalah ukuran yang digunakan untuk menilai kesamaan antara dua vektor non-nol dalam ruang produk dalam, yang didefinisikan sebagai kosinus dari sudut di antara keduanya dan juga sama dengan produk dalam dari kedua vektor yang dinormalisasi dengan panjang yang sama (Subekti, 2021).

## 2.6 Mobile

Smartphone, yang sering dikenal dengan nama Android, merupakan sistem operasi yang dirancang untuk smartphone dan tablet. Sistem operasi ini berperan sebagai penghubung antara perangkat dan pengguna, memfasilitasi interaksi pengguna dengan perangkat serta menjalankan berbagai aplikasi yang ada di dalamnya (Budiyanto, 2023). Android adalah generasi terbaru dari platform mobile yang berbasis Linux, mencakup sistem operasi, middleware, dan aplikasi (Kuswanto, 2020). Berikut adalah beberapa versi android:

**Tabel 2. 1 Versi Mobile Android**

No	Nama	Versi	Keterangan
1	Alpha	1.0	Versi Alpha Android 1.0 diperkenalkan pada September 2008 dan tidak ditujukan untuk penggunaan komersial. Meskipun belum diluncurkan secara resmi untuk pasar, versi awal Android ini telah dilengkapi dengan berbagai fitur seperti akses ke browser web, streaming video YouTube, pemutar media, Google Maps, serta integrasi dengan aplikasi Google lainnya.
2	Beta	1.1	Beberapa bulan kemudian, Android 1.1 versi Beta diluncurkan pada 9

			<p>Februari 2009. Seperti versi Alpha sebelumnya, versi ini juga belum dirilis untuk tujuan komersial dan hanya tersedia untuk satu jenis perangkat. Pembaruan ini bertujuan untuk memperbaiki berbagai bug serta meningkatkan fitur-fitur seperti detail lokasi dalam aplikasi peta dan fungsi untuk menyembunyikan atau menampilkan tombol panggilan.</p>
3	Cupcake	1.5	<p>Pada 30 April 2009, Android 1.5 Cupcake akhirnya diluncurkan secara komersial. Versi Cupcake adalah rilis pertama yang tersedia untuk umum dan menandai awal penggunaan nama makanan manis untuk setiap versi Android yang dirilis. Beberapa fitur yang diperkenalkan dengan Android Cupcake termasuk dukungan untuk rotasi layar otomatis, widget, serta keyboard virtual.</p>
4	Donut	1.6	<p>Pembaruan Android selanjutnya adalah versi 1.6 Donut, yang dirilis</p>

			<p>pada 15 September 2009. Versi Donut memperkenalkan berbagai fitur baru, seperti tampilan persentase daya baterai, dukungan untuk gerakan, dan kemampuan pencarian di Android Market, yang sekarang dikenal sebagai Play Store.</p>
5	Éclair	2.0	<p>Satu bulan setelahnya, pada 26 Oktober 2009, Android mengalami pembaruan lagi untuk memperbaiki bug dan menambahkan fitur baru. Pembaruan untuk versi 2.0 ini, yang dikenal sebagai Eclair, menghadirkan berbagai fitur baru, termasuk dukungan multi-touch, wallpaper hidup, perbaikan antarmuka, serta dukungan browser untuk HTML5.</p>
6	Froyo	2.2	<p>Pada 20 Mei 2010, Android merilis versi terbarunya, yaitu 2.2 Froyo. Versi Froyo ini mulai mendapatkan perhatian luas dari produsen ponsel. Pembaruan Android 2.2 menghadirkan beberapa fitur baru, termasuk</p>



			kemampuan untuk memperbesar gambar di galeri dengan gerakan, peningkatan pada fitur USB tethering dan hotspot Wi-Fi, serta dukungan untuk GIF animasi di browser web.
7	Gingerbread	2.3	Pada versi ini, tidak dapat disangkal bahwa Android telah menjadi sistem operasi seluler yang sangat populer. Kolaborasi dengan produsen Samsung dalam pengembangan seri Samsung Galaxy turut memperkuat popularitas Android. Versi 2.3 Gingerbread, yang dirilis pada 6 Desember 2010, menambahkan berbagai fitur baru, seperti dukungan NFC dan kemampuan untuk menyalin teks.
8	Honeycomb	3.0	Versi Android berikutnya adalah 3.0 Honeycomb, yang diluncurkan pada 22 Februari 2011. Versi ini dirancang khusus untuk tablet PC, dengan menambahkan fitur-fitur yang fokus pada kebutuhan komputasi bisnis di perangkat tablet. Android Honeycomb

			memperkenalkan dukungan untuk prosesor multi-core, fitur video chat melalui Google Talk, serta akselerasi dalam beralih antara aplikasi yang sedang berjalan dengan fitur multitasking recent apps.
9	Ice Cream Sandwich	4.0	Pada 19 Oktober 2011, Android meluncurkan versi terbarunya, yaitu 4.0 Ice Cream Sandwich. Versi ini membawa fitur-fitur dari Honeycomb agar kompatibel dengan smartphone, yang sebelumnya hanya tersedia untuk tablet PC. Selain fitur-fitur yang sudah ada di Honeycomb, versi 4.0 juga memperkenalkan beberapa pembaruan baru, termasuk perbaikan antarmuka dan opsi kustomisasi widget.
10	Jelly Bean	4.2	Android 4.1 Jelly Bean dirilis pada 27 Juni 2012. Fokus utama dari pengembangan versi Jelly Bean adalah peningkatan kinerja antarmuka. Fitur-fitur baru dalam versi ini mencakup keyboard yang dapat disesuaikan oleh

			pengguna dan mendukung gerakan, antarmuka yang lebih responsif, dukungan untuk tampilan nirkabel, serta widget yang dapat diatur dan disesuaikan ukurannya.
11	Kitkat	4.4	Android memilih nama KitKat untuk versi terbarunya, yaitu versi 4.4, yang dirilis pada 31 Oktober 2013. Versi ini membawa peningkatan optimasi dan menawarkan berbagai fitur baru. Beberapa fitur yang diperkenalkan dalam Android KitKat termasuk WebViews berbasis Chromium, peningkatan kinerja untuk perangkat dengan spesifikasi rendah, serta dukungan untuk sensor batching dan pelacakan langkah.
12	Lollipop	5.0	Versi Android selanjutnya, yaitu 5.0 Lollipop, dirilis pada 25 Juni 2014. Versi ini memperluas penggunaan Android tidak hanya sebagai sistem operasi untuk smartphone, tetapi juga untuk perangkat mobile lainnya seperti

			Android TV dan Google Fit. Fitur-fitur baru dalam versi ini mencakup antarmuka pengguna yang mengadopsi desain material dari Google serta perlindungan reset pabrik yang bertujuan untuk melindungi perangkat dari risiko reset jika perangkat hilang.
13	Marshmallow	6.0	Android 6.0 Marshmallow diluncurkan pada 5 Mei 2015, memperkenalkan berbagai fitur baru. Versi ini menambahkan dukungan untuk sensor sidik jari, memungkinkan akses yang lebih aman ke smartphone. Selain itu, Marshmallow menyediakan kemampuan untuk menjalankan beberapa aplikasi dalam tampilan layar dengan dukungan multi-jendela, integrasi dengan platform realitas virtual, dan fitur penghemat data yang membantu mengurangi penggunaan bandwidth.
14	Nougat	7.0	Dirilis pada 19 Oktober 2016, Android 7.0 Nougat fokus pada

			<p>penyempurnaan antarmuka pengguna, membuatnya lebih responsif dan mendukung penggunaan beberapa aplikasi secara bersamaan melalui fitur multi-jendela. Versi ini juga memperkenalkan beberapa fitur tambahan, termasuk mode malam atau night light, kemampuan keyboard bawaan untuk mengirim GIF animasi secara langsung, dan dukungan untuk panggilan dengan beberapa peserta.</p>
15	Oreo	8.0	<p>Android memilih nama Oreo untuk versi 8.0 yang diluncurkan pada Agustus 2017. Versi ini menghadirkan antarmuka yang lebih sederhana untuk mempermudah akses ke aplikasi. Android Oreo memperkenalkan berbagai fitur baru, seperti fungsi Isi Otomatis untuk mempermudah pengisian formulir, dukungan untuk tampilan gambar-dalam-gambar, dan peningkatan dalam proses booting untuk mempercepat waktu startup.</p>

16	Pie	9.0	<p>Android 9.0 Pie, yang dirilis pada Agustus 2018, merupakan versi terbaru dari sistem operasi Android. Versi 9.0 Pie menonjol dengan integrasi kecerdasan buatan (AI) yang memungkinkan perangkat untuk secara otomatis menganalisis dan memahami pola penggunaan pengguna. Fitur lainnya termasuk kecerahan adaptif yang menyesuaikan secara otomatis tingkat kecerahan layar, serta dukungan untuk ponsel tanpa bezel.</p>
17	10	10	<p>Android 10 adalah versi terbaru yang diluncurkan tanpa menggunakan nama makanan manis seperti versi sebelumnya. Penamaan ini merayakan dekade keberadaan Android di industri. Android 10 menekankan pada penyempurnaan mode malam atau mode gelap dan memperbaiki fungsi pengatur suara untuk meningkatkan kualitas audio.</p>

## 2.7 Unified Modeling Language (UML)

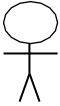
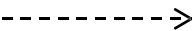
Unified Modeling Language (UML) adalah serangkaian konvensi pemodelan yang dirancang untuk menggambarkan atau mendefinisikan sistem perangkat lunak melalui objek. UML dianggap sebagai kumpulan teknik efektif yang telah terbukti bermanfaat dalam memodelkan sistem yang besar dan kompleks. Selain dalam pemodelan perangkat lunak, UML juga diterapkan di berbagai bidang ilmu yang memerlukan pemodelan. (Andikos, 2019).




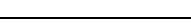

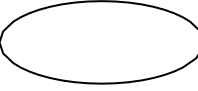


Alat-alat yang digunakan dalam desain berorientasi objek berbasis UML adalah sebagai berikut:

### 1. Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem dengan menekankan pada aspek "apa" yang dilakukan sistem, bukan bagaimana sistem tersebut bekerja. Setiap use case menggambarkan interaksi antara aktor dan sistem. Penjelasan lebih lanjut tentang diagram use case dapat ditemukan pada sumber-sumber yang tercantum di Tabel 2.2.

**Tabel 2. 2 Simbol Use Case**

Gambar	Nama	Keterangan
	<i>Actor</i>	Menetapkan kumpulan peran yang dimiliki pengguna saat berhubungan dengan sebuah use case.
	<i>Dependency</i>	Sebuah hubungan di mana perubahan yang terjadi pada elemen-elemen yang independen mempengaruhi elemen-elemen yang tergantung pada elemen-elemen yang tidak independen.

	<i>Generalization</i>	hubungan di mana objek anak (keturunan) mewarisi perilaku dan struktur data dari objek induk (ancestor).
	<i>Include</i>	Menentukan tujuan sumber tersebut.
	<i>Extend</i>	Menentukan apakah use case target pada titik tertentu memperluas perilaku use case sumber.
	<i>Association</i>	Menghubungkan dua objek.
	<i>System</i>	Mengidentifikasi sebuah paket yang akan menunjukkan sistem dengan cara yang terbatas.
	<i>Use Case</i>	Urutan tindakan yang dilakukan oleh sistem untuk menghasilkan hasil yang diukur bagi seorang aktor.
	<i>Collaboration</i>	Bagaimana aturan dan elemen lainnya bekerja sama untuk menghasilkan perilaku yang lebih baik (sinergi).
	<i>Note</i>	Elemen fisik yang ada saat aplikasi berjalan dan merupakan representasi dari sumber daya komputasi


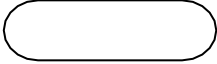



(Sumber: Andikos, 2019)

## 2. Diagram Aktivitas (*Activity Diagram*)

Diagram aktivitas digunakan untuk menggambarkan berbagai alur kegiatan dalam sistem yang sedang dirancang. Diagram ini mengilustrasikan cara setiap alur dimulai, keputusan-keputusan yang mungkin muncul, serta cara alur tersebut diakhiri. Selain itu, diagram ini juga dapat menunjukkan proses-proses paralel yang mungkin terjadi selama eksekusi, seperti yang dijelaskan dalam tabel 2.3.



**Tabel 2. 3 Simbol Activity Diagram**

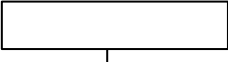
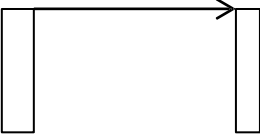
<b>Gambar</b>	<b>Nama</b>	<b>Keterangan</b>
	<i>Activity</i>	Menunjukkan bagaimana setiap kelas antarmuka saling berhubungan satu sama lain.
	<i>Action</i>	Status dalam sistem yang mencerminkan pelaksanaan suatu tindakan.
	<i>Initial Node</i>	Bagaimana objek dibuat atau diinisialisasi
	<i>Activity Final</i>	Cara Objek dibuat dan di hancurkan
	<i>Fork Node</i>	Satu alur yang pada tahap tertentu bercabang menjadi beberapa jalur


(Sumber: Andikos, 2019)

### 3. Diagram Urutan (*Sequence Diagram*)

Diagram urutan menunjukkan bagaimana objek berinteraksi dalam sistem dan di sekitarnya, termasuk pengguna dan antarmuka, melalui pesan yang disampaikan dari waktu ke waktu. Diagram ini dapat digambarkan menggunakan berbagai simbol yang tercantum dalam Tabel 2.4.

**Tabel 2. 4 Simbol Sequence Diagram**

<b>Gambar</b>	<b>Nama</b>	<b>Keterangan</b>
	<i>Lifeline</i>	Objek entitas, antarmuka yang saling berhubungan satu sama lain.
	<i>Message</i>	Spesifikasi untuk komunikasi antar objek yang mencakup informasi mengenai aktivitas yang terjadi.



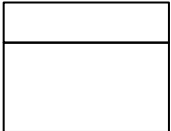

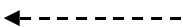
	<i>Message</i>	Spesifikasi komunikasi antar objek yang memuat informasi tentang aktivitas yang berlangsung.
---	----------------	--



(Sumber: Andikos, 2019)

#### 4. *Class Diagram* (Diagram Kelas)

Kelas merupakan spesifikasi yang, ketika diinstansiasi, menghasilkan objek dan dianggap sebagai pusat dari pengembangan serta desain berbasis objek. Diagram kelas memperlihatkan struktur dan deskripsi dari kelas, paket, dan objek, serta hubungan di antara mereka seperti komposisi, pewarisan, dan asosiasi. Diagram kelas dapat digambarkan dengan simbol-simbol yang dijelaskan dalam Tabel 2.5.

**Tabel 2.5 *Class Diagram***

<b>Gambar</b>	<b>Nama</b>	<b>Keterangan</b>
	<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data pada objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
	<i>Nary Association</i>	Upaya untuk menghinpada asosiasi dengan lebih pada 2 objek.
	<i>Class</i>	Himpunan pada objek-objek yang berbagi atribut serta operasi yang sama.
	<i>Collaboration</i>	Deskripsi pada urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu

		objek.
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.

(Sumber: Andikos, 2019)

## 2.8 Penelitian Terdahulu

Penelitian terdahulu berkaitan dengan penulisan skripsi yang penulis buat dan disajikan pada Tabel 2.5

**Tabel 2. 6 Penelitian Terdahulu**

No.	Penulis	Judul	Tahun	Kesimpulan
1.	Atmaja	Implementasi Metode Levensthein Distance Dan Cosine Similarity Untuk Deteksi Kemiripan Gambar	2022	Metode Levenshtein Distance dapat diterapkan untuk mendeteksi kemiripan gambar, dengan hasil pengujian menunjukkan tingkat kesamaan sebesar 70% antara dua gambar. Di sisi lain, metode Cosine Similarity juga efektif dalam mendeteksi kemiripan gambar, dengan hasil pengujian mencapai 83% untuk dua gambar.
2.	Lindang, et al	Sistem Penentuan Kemiripan Antar Skripsi Menggunakan Metode Cosine Similarity Pada Perpustakaan	2022	Hasil penelitian dan pengujian sistem dengan menggunakan metode cosine similarity menunjukkan bahwa setelah melakukan 10

				<p>kali pengujian dengan kata kunci dan 9 kali evaluasi, diperoleh rata-rata akurasi sebesar 97% dan rata-rata waktu komputasi sebesar 0,277154 detik.</p>
3.	Saleh, et al	Face Identification on Login Security Using Algorithm Combination of Viola-Jones and Cosine Similarity	2020	<p>Dari hasil pengujian yang dilakukan, dapat disimpulkan bahwa penerapan metode Viola-Jones, GLCM, dan Cosine Similarity untuk identifikasi wajah dalam sistem keamanan login memberikan hasil yang memuaskan dengan tingkat akurasi mencapai 77,20%. Dari 500 gambar yang digunakan sebagai sampel, sebanyak 386 gambar berhasil diidentifikasi dengan benar.</p>
4.	Siswanto & Giap	Implementasi Algoritma Rabin-Karp Dan Cosine Similarity Untuk Pendeteksi Plagiarisme Pada Dokumen	2020	<p>Hasil uji coba aplikasi yang menggunakan algoritma Rabin-Karp dan metode Cosine Similarity menunjukkan bahwa pemrosesan perbandingan plagiarisme untuk teks dengan total 42.432 karakter memerlukan waktu antara 12,49 hingga 16,17 detik. Durasi ini bervariasi tergantung pada jumlah karakter dalam dokumen yang dianalisis.</p>

5.	Haryono, et al	Rancang Bangun Sistem Ujian Online Menggunakan Algoritma Cosine Similarity Berbasis Web	2021	Penerapan sistem aplikasi ujian online berbasis web telah terbukti efektif dalam mengatasi berbagai tantangan. Bagi para pengajar, sistem ini menyederhanakan proses pelaksanaan ujian online, mempermudah pengelolaan nilai, dan membuat penyusunan soal ujian menjadi lebih efisien. Di sisi lain, siswa dapat dengan mudah mengakses hasil ujian mereka secara langsung melalui sistem ini.
----	-------------------	---	------	--

## **BAB III**

### **ANALISA DAN DESAIN SISTEM**

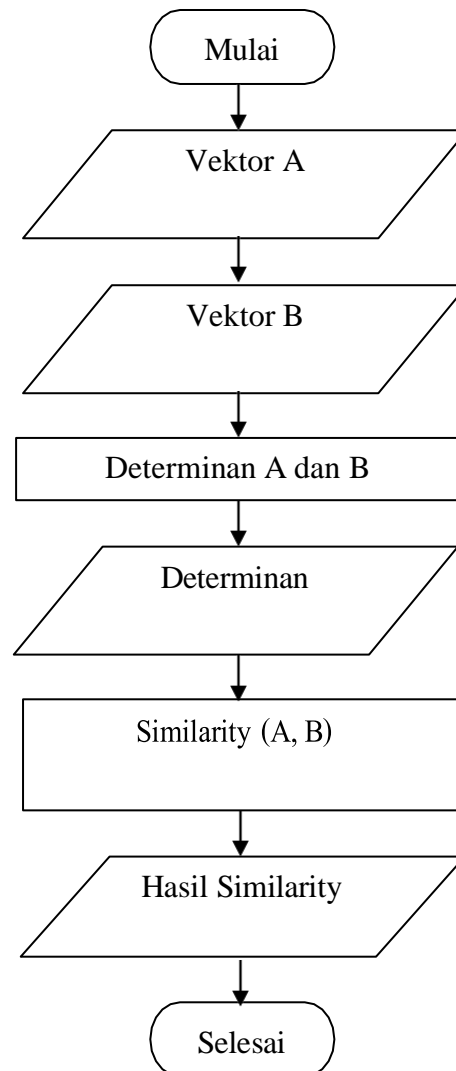
#### **3.1 Analisa**

Gambar dapat disalin sebanyak mungkin selama file gambar tersebut masih tersedia. Namun, ketika file gambar sudah tidak ada, satu-satunya opsi adalah memotret ulang atau memindai gambar yang ada dalam bentuk fisik. Masalah yang sering muncul adalah hasil dari pemotretan ulang atau pemindaian tidak selalu akurat dalam mencerminkan gambar aslinya. Untuk mengatasi hal ini, diperlukan metode yang dapat mengukur kemiripan antara gambar-gambar tersebut. Penelitian ini memanfaatkan metode Cosine Similarity untuk mengidentifikasi kesamaan dan perbedaan antara gambar-gambar tersebut. Implementasi metode ini memungkinkan deteksi yang lebih baik terhadap tingkat kemiripan antara gambar yang dihasilkan.

#### **3.2 Penerapan Metode**

Cosine Similarity adalah metode untuk mengukur kesamaan antara dua vektor dalam ruang berdimensi dengan memanfaatkan nilai cosinus dari sudut yang dibentuk oleh kedua vektor tersebut. Karena cosinus dari sudut 0 derajat adalah 1 dan nilai cosinus untuk sudut lainnya kurang dari 1, dua vektor dianggap mirip jika nilai Cosine Similarity-nya mendekati 1. Metode ini berlaku dalam ruang positif, dengan hasil yang berada dalam rentang antara 0 dan 1. Nilai 0 menunjukkan bahwa dokumen tersebut tidak mirip, sementara nilai 1 menunjukkan tingkat kemiripan yang sangat tinggi. Perlu dicatat bahwa batasan ini berlaku untuk sejumlah dimensi yang terlibat dalam perhitungan.

*Flowchart metode cosine similarity:*



**Gambar 3.1** *Flowchart Metode Cosine Similarity*

**Keterangan:**

Gambar 3.1 menggambarkan proses penerapan metode cosine similarity yang dimulai dengan penentuan vektor A dan B. Langkah berikutnya adalah menghitung nilai determinan untuk masing-masing vektor tersebut. Setelah nilai determinan diperoleh, tahap terakhir adalah menentukan nilai kemiripan antara vektor A dan B.

Vektor A: yang digunakan pada vektor A adalah gambar pertama

Vektor B: yang digunakan pada vektor B adalah gambar kedua

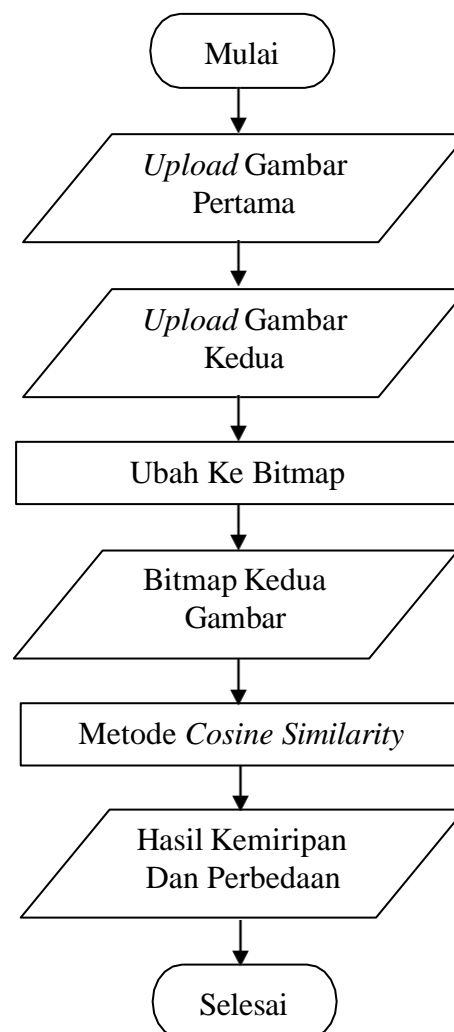
Determinan A dan B: penentuan nilai determinan pada vektor A dan B

Determinan: hasil pada penentuan nilai determinan

Similarity (A, B): Proses pencarian nilai perbedaan gambar

Hasil similarity: Hasil pada pencarian nilai similarity

*Flowchart* tahapan penggunaan perangkat lunak yang akan dibuat:



**Gambar 3. 2** *Flowchart* Tahapan Penggunaan Perangkat Lunak



**Keterangan:**

Gambar 3.2 menguraikan langkah-langkah dalam penggunaan perangkat lunak yang direncanakan. Proses dimulai dengan mengunggah gambar pertama. Setelah gambar pertama berhasil ditampilkan, langkah berikutnya adalah mengunggah gambar kedua. Setelah gambar kedua muncul, kedua gambar akan diproses menggunakan metode cosine similarity. Proses ini akan menghasilkan analisis tentang tingkat kemiripan dan perbedaan antara kedua gambar.

Upload Gambar Pertama: yang digunakan adalah gambar pertama

Upload Gambar Kedua: yang digunakan adalah gambar kedua

Ubah ke Bitmap: mengubah nilai gambar ke bitmap

Bitmap kedua gambar: nilai bitmap pada kedua gambar

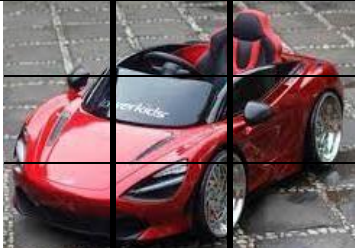
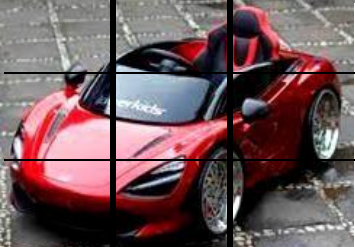
Metode Cosine Similarity: Proses pencarian nilai perbedaan gambar

Hasil kemiripan dan perbedaan gambar: Hasil pada pencarian nilai similarity

Contoh:

Tahapan metode Cosine Similarity dimulai dengan mengunggah gambar pertama. Setelah gambar pertama muncul, langkah berikutnya adalah mengunggah gambar kedua. Setelah gambar kedua ditampilkan, kedua gambar tersebut akan diproses menggunakan metode Cosine Similarity. Proses ini akan menghasilkan analisis mengenai kemiripan dan perbedaan antara kedua gambar.

**Tabel 3. 1 Nilai RGB Gambar**

No.	Gambar	RGB
1.		R:200 R:121 R:180
		G:211 G:181 G:122
		B:175 B:111 B:221
		R:112 R:200 R:214
		G:161 G:198 G:100
		B:113 B:211 B:241
		R:211 R:200 R:102
		G:178 G:189 G:200
		B:210 B:200 B:198
2.		R:200 R:127 R:198
		G:211 G:181 G:120
		B:175 B:120 B:220
		R:200 R:205 R:214
		G:161 G:198 G:100
		B:113 B:211 B:241
		R:211 R:200 R:102
		G:171 G:143 G:117
		B:230 B:150 B:200

Pada kedua gambar maka diperoleh data sebagai berikut:

Red:

200: pada gambar pertama ada 1, gambar kedua ada 1.

121: pada gambar pertama ada 1, gambar kedua ada 0.

180: pada gambar pertama ada 1, gambar kedua ada 0.

112: pada gambar pertama ada 1 gambar kedua ada 0.

200: pada gambar pertama ada 1, gambar kedua ada 0.

214: pada gambar pertama ada 1 gambar kedua ada 1.

211: pada gambar pertama ada 1, gambar kedua ada 1.

200: pada gambar pertama ada 1 gambar kedua ada 1.

102: pada gambar pertama ada 1, gambar kedua ada 1.

Green:

211: pada gambar pertama ada 1, gambar kedua ada 1.

181: pada gambar pertama ada 1, gambar kedua ada 1.

122: pada gambar pertama ada 1, gambar kedua ada 0.

161: pada gambar pertama ada 1, gambar kedua ada 0.

198: pada gambar pertama ada 1, gambar kedua ada 1.

100: pada gambar pertama ada 1, gambar kedua ada 1.

178: pada gambar pertama ada 1, gambar kedua ada 0.

189: pada gambar pertama ada 1, gambar kedua ada 0.

200: pada gambar pertama ada 1, gambar kedua ada 0.

Blue:

175: pada gambar pertama ada 1, gambar kedua ada 1.

111: pada gambar pertama ada 1 gambar kedua ada 0.

221: pada gambar pertama ada 1, gambar kedua ada 0.

113: pada gambar pertama ada 1, gambar kedua ada 1.

211: pada gambar pertama ada 1, gambar kedua ada 1.

241: pada gambar pertama ada 1, gambar kedua ada 1.

210: pada gambar pertama ada 1, gambar kedua ada 0.

200: pada gambar pertama ada 1, gambar kedua ada 0.

198: pada gambar pertama ada 1, gambar kedua ada 0.

$$\text{Similarity (A, B)} = \frac{\sum_{i=1}^n a_i \cdot b_i}{\sqrt{\sum_{i=1}^n (a_i)^2} \sqrt{\sum_{i=1}^n (b_i)^2}}$$

$$\text{Similarity (R)} = \frac{(1 * 1) + (1 * 0) + (1 * 0) + (1 * 0) + (1 * 0) + (1 * 1) + (1 * 1) + (1 * 1) + (1 * 1)}{\sqrt{1^2 + 1^2 + 1^2 + 1^2 + 1^2 + 1^2 + 1^2 + 1^2} \sqrt{1^2 + 0^2 + 0^2 + 0^2 + 0^2 + 1^2 + 1^2 + 1^2}}$$

$$\text{Similarity (R)} = \frac{5}{\sqrt{9} * \sqrt{5}} = 0.75$$

$$\text{Similarity (G)} = \frac{(1 * 1) + (1 * 1) + (1 * 0) + (1 * 0) + (1 * 1) + (1 * 1) + (1 * 0) + (1 * 0) + (1 * 0)}{\sqrt{1^2 + 1^2 + 1^2 + 1^2 + 1^2 + 1^2 + 1^2 + 1^2} \sqrt{1^2 + 1^2 + 0^2 + 0^2 + 1^2 + 1^2 + 0^2 + 0^2}}$$

$$\text{Similarity (G)} = \frac{4}{\sqrt{9} * \sqrt{4}} = 0.6666667$$

$$\text{Similarity (B)} = \frac{(1 * 1) + (1 * 0) + (1 * 0) + (1 * 1) + (1 * 1) + (1 * 1) + (1 * 0) + (1 * 0) + (1 * 0)}{\sqrt{1^2 + 1^2 + 1^2 + 1^2 + 1^2 + 1^2 + 1^2 + 1^2 * \sqrt{1^2 + 0^2 + 0^2 + 1^2 + 1^2 + 0^2 + 0^2}}}$$

$$\text{Similarity (R)} = \frac{4}{\sqrt{9 * 4}} = 0.6666667$$

$$\text{Total} = \frac{\text{Similarity(R)} + \text{Similarity(G)} + \text{Similarity(B)} * 3}{0.75 + 0.6666667 + 0.6666667}$$

$$\text{Dist(A, B)} = \frac{3}{3}$$

$$\text{Dist(A, B)} = 0.69$$

$$\text{Persentase} = 0.69 * 100\%$$

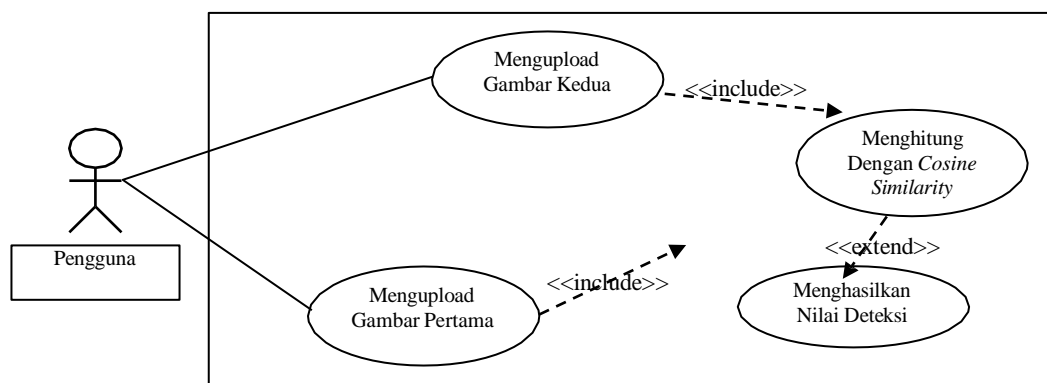
$$\text{Dist(A, B)} = 69\% \text{ mirip}$$

### 3.3 Desain Sistem

Bentuk desain sistem yang penulis buat menggunakan beberapa bentuk diagram pada *Unified Modeling Language* 3 yaitu *Use Case Diagram*, *Activity Diagram* dan *Sequence Diagram*.

#### 3.3.1. Use Case Diagram

Perancangan dimulai pada identifikasi aktor dan bagaimana hubungan antara aktor dan use case didalam perangkat lunak. Perancangan *Use Case Diagram* dapat dilihat pada gambar 3.3.



**Gambar 3.3 Use Case Aplikasi Deteksi Gambar Hasil Scan Menggunakan Metode Cosine Similarity Berbasis Mobile**

**Keterangan:**

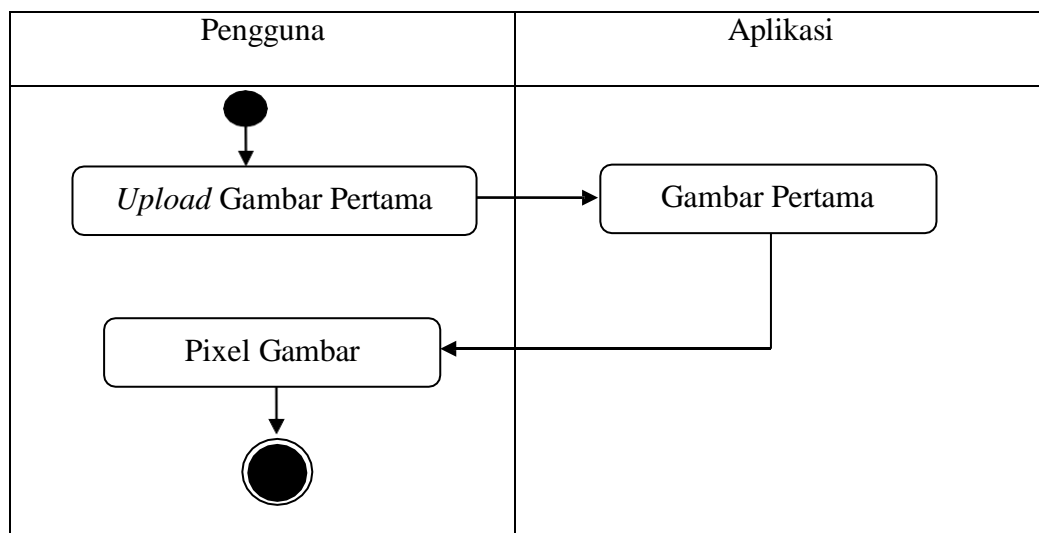
Gambar 3.3 menunjukkan langkah-langkah penggunaan aplikasi, yang dimulai dengan mengunggah gambar pertama. Setelah gambar pertama berhasil dimuat, langkah berikutnya adalah mengunggah gambar kedua. Setelah gambar kedua ditampilkan, kedua gambar tersebut akan dianalisis menggunakan metode Cosine Similarity. Proses ini akan menampilkan hasil analisis mengenai kemiripan dan perbedaan antara kedua citra gambar.

**3.3.2. ActivityDiagram**

Serangkaian kerja sistem melakukan deteksi persamaan gambar dapat terlihat sebagai berikut:

1. *Activity Diagram* Gambar Pertama

Gambar 3.4 menjelaskan pengguna aplikasi yang melakukan *Upload* Gambar Pertama untuk deteksi kemiripan dan perbedaancitra gambar:



**Gambar 3. 4 ActivityDiagram Gambar Pertama**

**Keterangan:**

Gambar 3.4 menjelaskan tahapan *activity diagram* gambar pertama yang dimulai pada pengguna mengupload gambar pertama maka dari itu sistem menyajikan pixel pada gambar pertama.

Upload Gambar Pertama: Proses melakukan upload gambar pertama

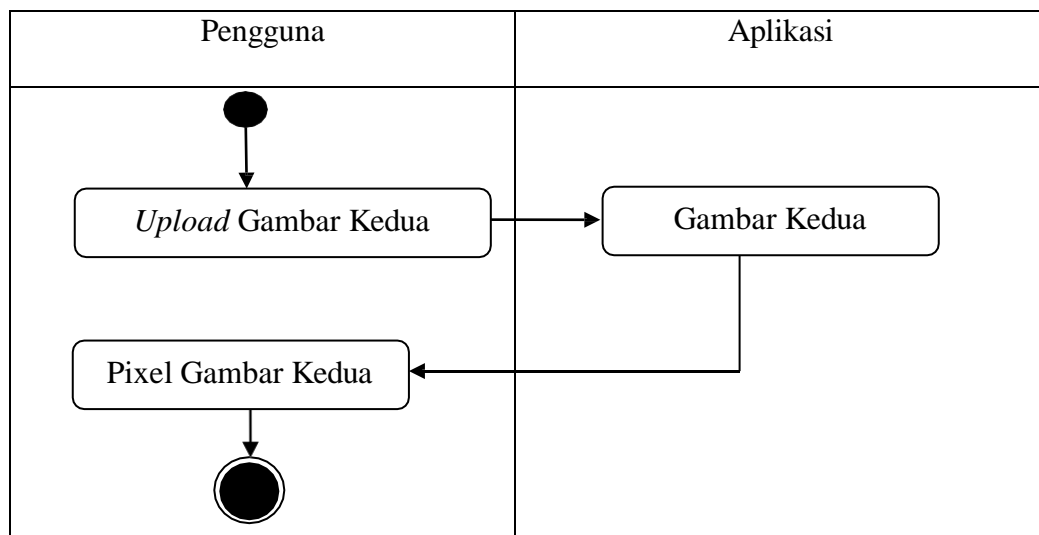
Gambar Pertama: Memperlihatkan gambar pertama

Pixel Gambar Pertama: Memperlihatkan pixel gambar pertama

## 2. Activity Diagram Gambar Kedua

Gambar 3.5 menjelaskan Pengguna aplikasi yang melakukan *Upload*

Gambar Kedua untuk deteksi kemiripan dan perbedaan citra gambar:



**Gambar 3.5 ActivityDiagram Gambar Kedua**

**Keterangan:**

Gambar 3.5 menggambarkan proses dalam activity diagram untuk gambar kedua. Tahapan dimulai ketika pengguna mengunggah gambar kedua, kemudian aplikasi akan menampilkan piksel dari gambar tersebut.

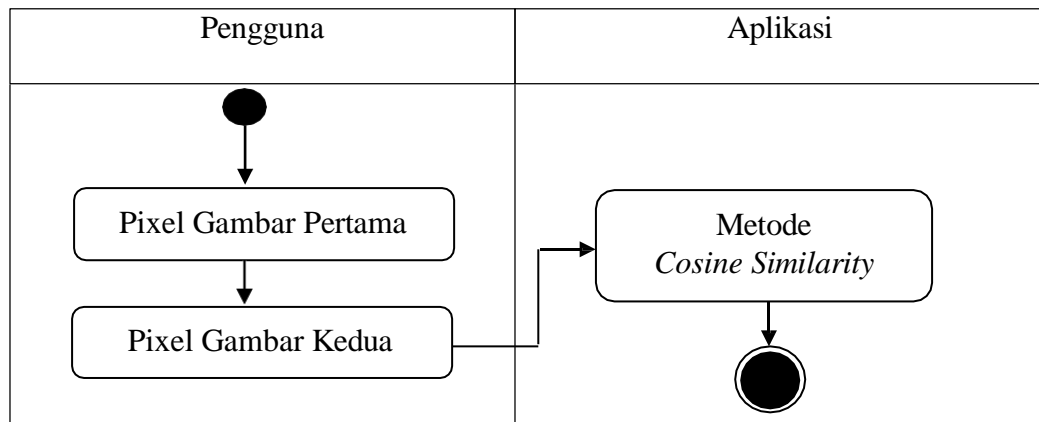
Upload Gambar Kedua: Proses melakukan upload gambar kedua

Gambar Kedua: Memperlihatkan gambar kedua

Pixel Gambar Kedua: Memerlihatkan pixel gambar kedua

### 3. Activity Diagram Cosine Similarity

Gambar 3.6 menjelaskan Pengguna aplikasi yang menggunakan metode *Cosine Similarity* untuk deteksi kemiripan dan perbedaan citra gambar:



**Gambar 3. 6 ActivityDiagram Cosine Similarity**

#### **Keterangan:**

Gambar 3.6 menjelaskan tahapan *activity diagram cosine similarity* dimulai pada pengguna yang menggunakan pixel gambar pertama dan pixel gambar kedua kemudian menggunakan metode *cosine similarity*.

Pixel Gambar Pertama: menginputkan pixel gambar pertama

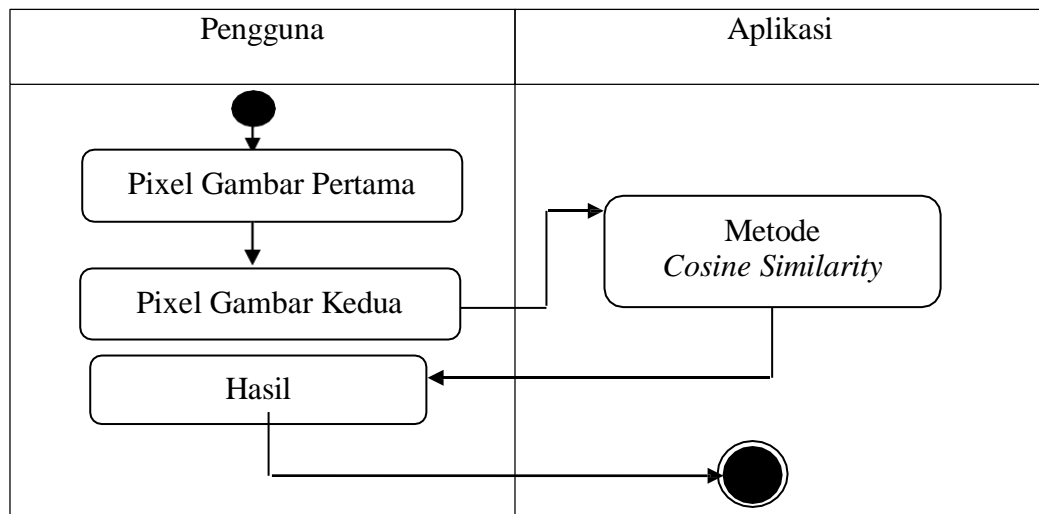
Pixel Gambar Kedua: menginputkan pixel gambar kedua

Metode Cosine Similarity: memproses metode cosine similarity

### 4. Activity Diagram Hasil

Gambar 3.7 menjelaskan Pengguna aplikasi yang mendapatkan hasil pada deteksi kemiripan dan perbedaan citra gambar:

\



**Gambar 3. 7 Activity Diagram Hasil**

**Keterangan:**

Gambar 3.7 menunjukkan langkah-langkah dalam activity diagram hasil. Proses dimulai ketika pengguna memanfaatkan piksel dari gambar pertama dan kedua, kemudian menerapkan metode cosine similarity. Hasil dari proses ini akan menunjukkan tingkat kemiripan dan perbedaan antara kedua gambar.

Pixel Gambar Pertama: menginputkan pixel gambar pertama

Pixel Gambar Kedua: menginputkan pixel gambar kedua

Metode Cosine Similarity: melakukan proses metode cosine similarity

Hasil: memperlihatkan hasil metode cosine similarity

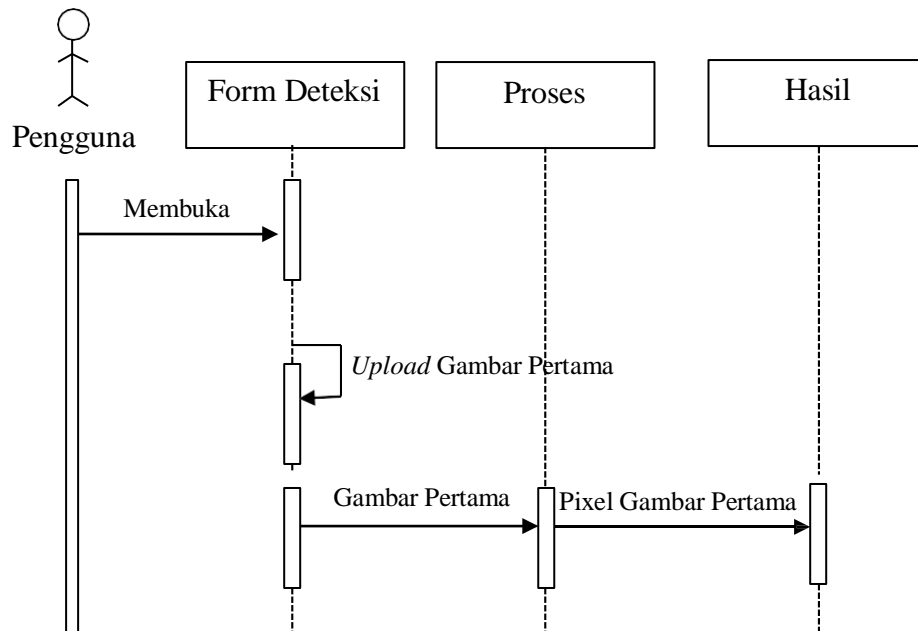
### 3.3.3. Sequence Diagram

Urutan kerja yang dilakukan untuk melakukan deteksi kemiripan dan perbedaan citra gambar dapat dilihat sebagaiberikut:

#### 1. Sequence Diagram Gambar Pertama

Gambar 3.8 menjelaskan Pengguna aplikasi yang mengupload Gambar Pertama untuk deteksi kemiripan dan perbedaan citra gambar:





**Gambar 3.8 Sequence Diagram Gambar Pertama**

**Keterangan:**

Gambar 3.8 menggambarkan tahapan dalam sequence diagram untuk gambar pertama. Proses dimulai ketika pengguna mengunggah gambar pertama, yang kemudian diikuti dengan aplikasi yang menampilkan piksel dari gambar tersebut.

Aktor Pengguna: melakukan beberapa aktifitas pada sequence diagram

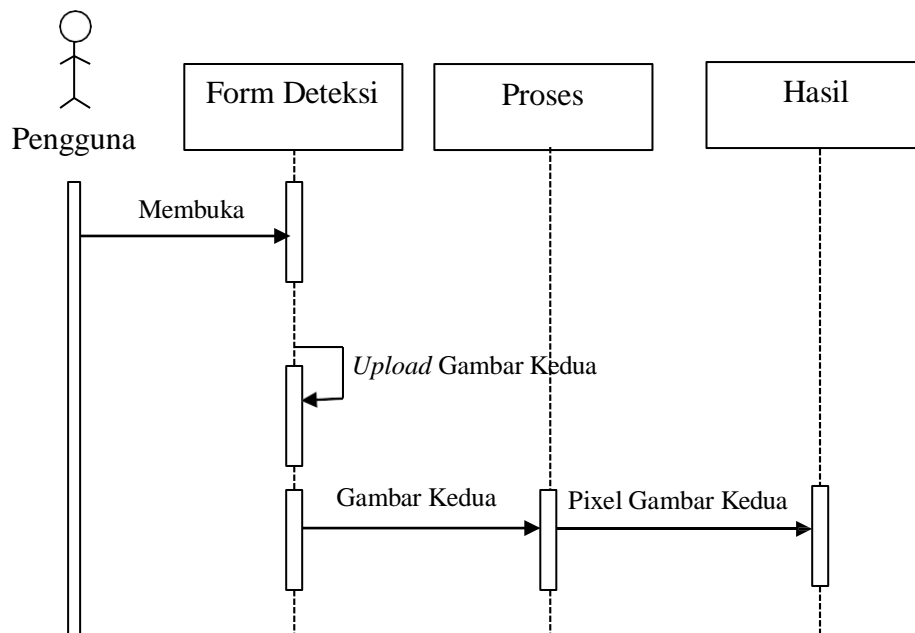
Form Deteksi: kesediaan form deteksi

Proses: proses setelah form deteksi yaitu upload gambar

Hasil: hasil setelah proses yaitu pixel gambar

## 2. Sequence Diagram Gambar Kedua

Gambar 3.9 menjelaskan Pengguna aplikasi yang mengupload Gambar Kedua untuk deteksi kemiripan dan perbedaan citra gambar:



**Gambar 3.9 Sequence Diagram Gambar Kedua**

**Keterangan:**

Gambar 3.9 menggambarkan tahapan dalam diagram urutan untuk gambar kedua, yang dimulai dengan pengguna mengunggah gambar kedua. Setelah itu, aplikasi akan menampilkan piksel dari gambar kedua.

Aktor Pengguna: melakukan beberapa aktifitas pada sequence diagram

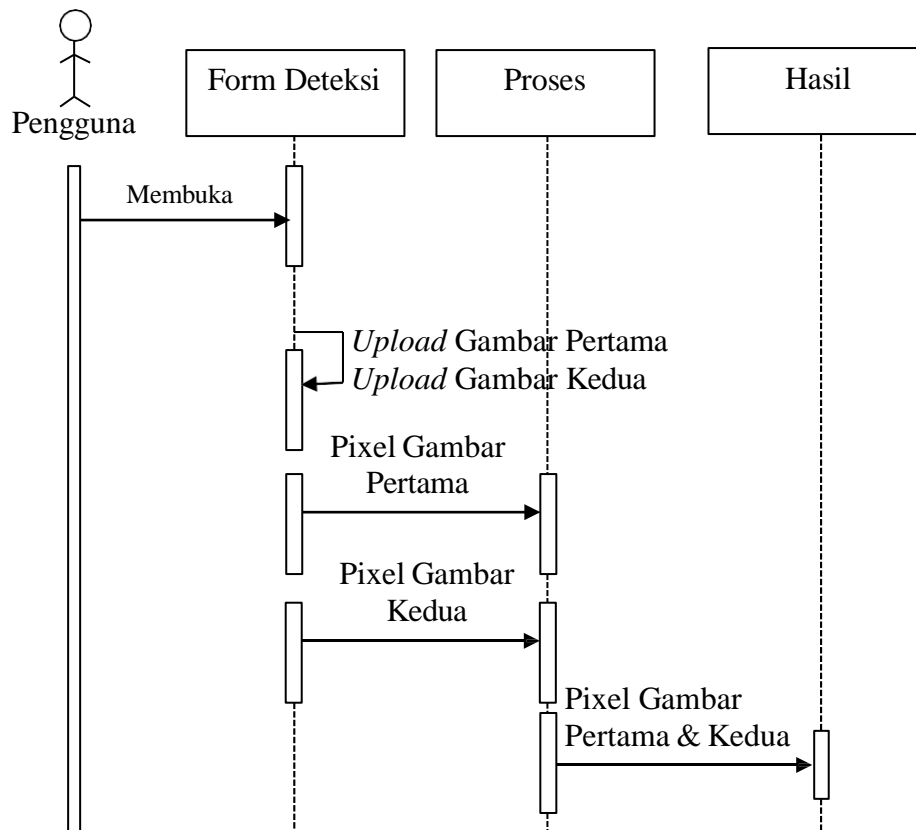
Form Deteksi: kesediaan form deteksi

Proses: proses setelah form deteksi yaitu upload gambar

Hasil: hasil setelah proses yaitu pixel gambar

### 3. Sequence Diagram Metode

Gambar 3.10 menjelaskan Pengguna aplikasi yang mengupload Gambar Pertama dan Gambar Kedua untuk deteksi kemiripan dan perbedaan citra gambar:



**Gambar 3. 10 SequenceDiagram Metode**

**Keterangan:**

Gambar 3.10 menguraikan tahapan dalam diagram urutan untuk metode cosine similarity, yang dimulai dengan pengguna memanfaatkan piksel dari gambar pertama dan gambar kedua, diikuti dengan penerapan metode cosine similarity.

Aktor Pengguna: melakukan beberapa aktifitas pada sequence diagram

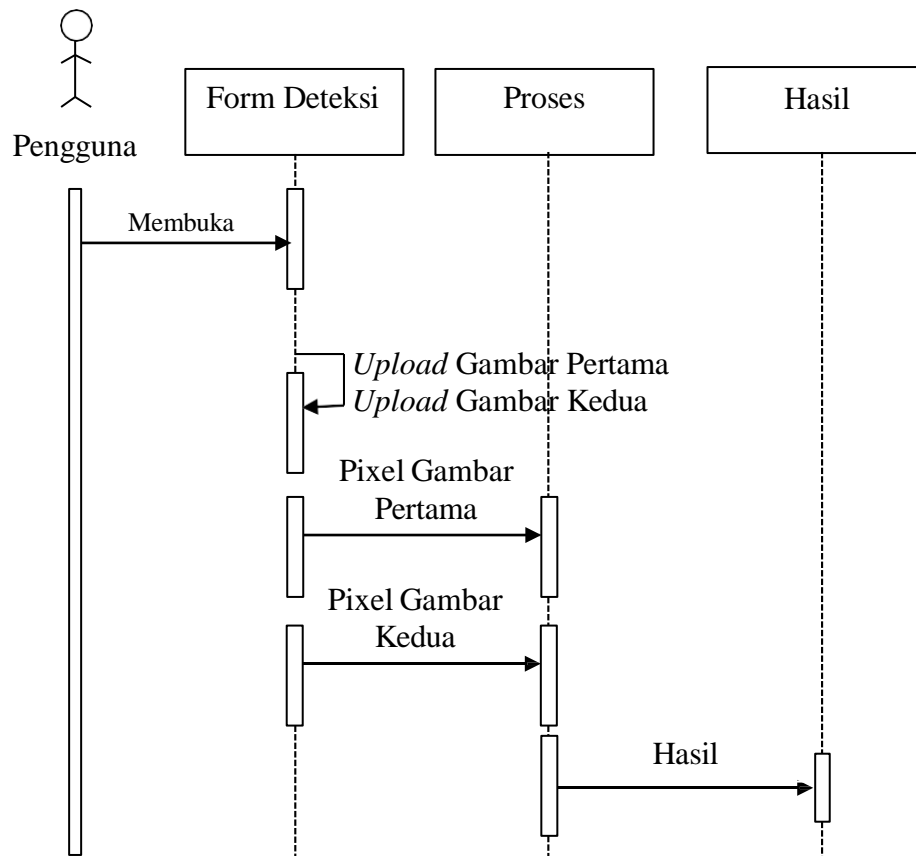
Form Deteksi: kesediaan form deteksi

Proses: proses setelah form deteksi yaitu upload gambar

Hasil: hasil setelah proses yaitu pixel gambar

**4. Sequence Diagram Hasil**

Gambar 3.11 menjelaskan Pengguna aplikasi yang mengupload Gambar Pertama dan Gambar Kedua untuk deteksi kemiripan dan perbedaan citra gambar:



**Gambar 3.11** *Sequence Diagram Hasil*

**Keterangan:**

Gambar 3.11 menggambarkan tahapan dalam diagram urutan untuk proses hasil, yang dimulai dengan pengguna yang memanfaatkan piksel dari gambar pertama dan gambar kedua. Setelah itu, metode cosine similarity diterapkan untuk menghasilkan perbandingan kemiripan dan perbedaan pada gambar hasil pemindaian.

Aktor Pengguna: melakukan beberapa aktifitas pada sequence diagram

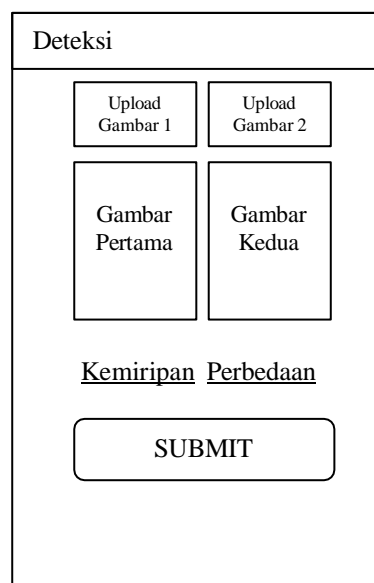
Form Deteksi: kesediaan form deteksi

Proses: proses setelah form deteksi yaitu upload gambar

Hasil: hasil setelah proses yaitu hasil metode

### 3.4. Desain User Interface

Desain user interface Aplikasi Deteksi Gambar Hasil Scan Menggunakan Metode Cosine Similarity Berbasis Mobile sebagai berikut:



The image shows a mobile application interface for image detection. It is titled "Deteksi" at the top. Below the title, there are two columns of buttons. The left column has a button labeled "Upload Gambar 1" above a placeholder box labeled "Gambar Pertama". The right column has a button labeled "Upload Gambar 2" above a placeholder box labeled "Gambar Kedua". Below these two columns, there are two underlined text labels: "Kemiripan" and "Perbedaan". At the bottom of the interface is a large rounded rectangular button labeled "SUBMIT".

**Gambar 3. 12 Desain User Interface**

#### **Keterangan:**

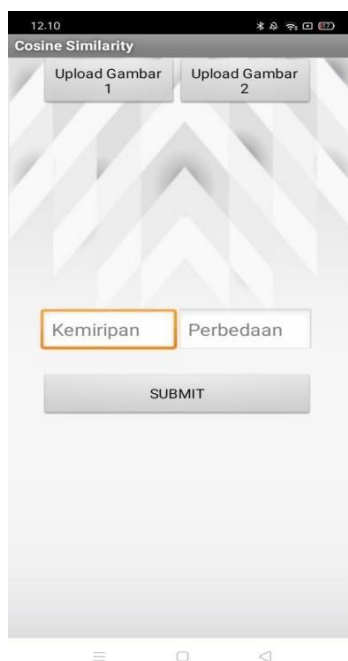
Gambar 3.12 menunjukkan proses desain antarmuka pengguna yang dimulai dengan pengguna mengunggah gambar pertama melalui tombol "upload pertama" dan gambar kedua melalui tombol "upload kedua". Selanjutnya, metode cosine similarity diterapkan dengan mengklik tombol metode, yang kemudian menampilkan hasil perbandingan kemiripan dan perbedaan antara gambar-gambar tersebut.

## BAB IV

### HASIL DAN UJI COBA

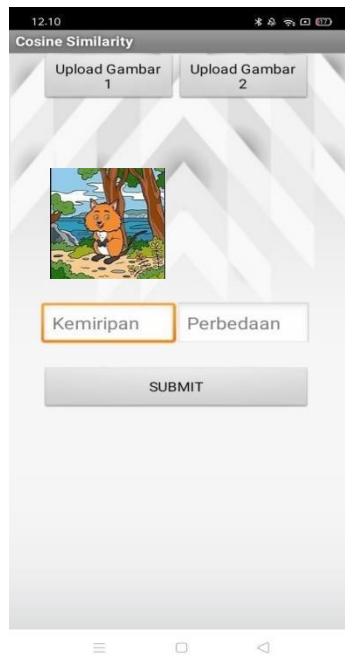
#### 4.1 Tampilan Hasil

Tampilan hasil pada Aplikasi Deteksi Gambar Hasil Scan Menggunakan Metode Cosine Similarity Berbasis Mobile dapat dilihat sebagai berikut:



**Gambar 4. 1 Tampilan *Form* Citra Digital**

Lakukan upload gambar pertama sebagai gambar asli untuk menentukan gambar kedua yaitu kemiripan dan perbedaan gambar dan dapat dilihat sebagai berikut:



**Gambar 4. 2 Tampilan Upload Gambar Pertama**

Lakukan upload gambar kedua sebagai gambar uji untuk menentukan gambar kedua yaitu kemiripan dan perbedaan gambar dan dapat dilihat sebagai berikut:



**Gambar 4. 3 Tampilan Upload Gambar Pertama**

Setelah melakukan upload gambar pertama dan upload gambar kedua maka akan memperlihatkan hasil kemiripan dan perbedaan sebagai berikut:



**Gambar 4. 4 Tampilan Upload Gambar Pertama**

#### **4.2. Pembahasan**

Untuk mengembangkan Aplikasi Deteksi Gambar Hasil Scan dengan Metode Cosine Similarity berbasis mobile, diperlukan baik perangkat keras maupun perangkat lunak yang mendukung agar aplikasi dapat berfungsi dengan optimal. Berikut adalah persyaratan perangkat keras dan perangkat lunak yang diperlukan untuk pembuatan aplikasi tersebut:

1. Satu buah laptop dengan spesifikasi berikut:
  - a. *processor* minimal *Core 2 Duo*
  - b. RAM minimal 8 Gb
  - c. *Hardisk* minimal 80 Gb



2. Perangkat lunak dengan spesifikasi berikut:

- a. Sistem operasi *Windows*
- b. *Eclipse*

#### 4.2.1. Uji Coba Program

Pengujian sistem bertujuan untuk memastikan bahwa sistem dalam kondisi siap pakai. Instrumen yang digunakan untuk melakukan uji ini adalah *Blackbox Testing*:

**Tabel 4. 1 *Blackbox Testing Form Deteksi***





No	Form Deteksi	Keterangan	Validitas
1.	Klik Tombol Upload Gambar Pertama	Aplikasi memperlihatkan pemilihan file gambar	Valid
2.	Klik Tombol Upload Gambar Kedua	Aplikasi memperlihatkan pemilihan file gambar	Valid
3.	Klik Tombol Submit	Aplikasi memperlihatkan hasil perhitungan metode <i>Cosine Similarity</i>	Valid









#### 4.2.2. Hasil Uji Coba







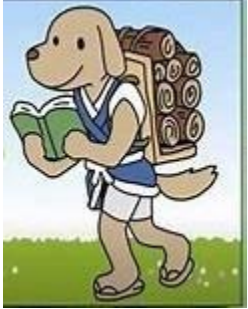



Setelah melakukan uji coba terhadap sistem, maka dapat disimpulkan hasil yang didapatkan yaitu:









1. *Interface* rancangan telah sesuai dengan *Interface* hasil.
2. Metode *Cosine Similarity* telah diterapkan pada aplikasi yang dibuat.
3. *Interface* aplikasi bersifat *user friendly* maka dari itu pengguna dapat menggunakannya dengan mudah.
4. Aplikasi yang telah dibuat berjalan dengan baik.
5. Aplikasi yang telah dibuat tidak memiliki kesalahan logika.











**Tabel 4. 2 Blackbox Testing Form Deteksi**

No	Gambar 1	Gambar 2	Kemiripan	Perbedaan
1.			83%	17%
2.			93%	7%

3.			93%	7%
4.			93%	7%
5.			95%	5%
6.			81%	19%

7.			81%	19%
8.			82%	18%
9.			85%	15%
10.			90%	10%
11.			87%	13%

12.			85%	15%
13.			83%	17%
14.			90%	10%
15.			82%	18%

16.			83%	17%
17.			93%	7%
18.			91%	9%
19.			94%	6%
20.			80%	20%

Hasil pengujian terhadap 20 gambar menunjukkan bahwa tingkat kemiripan tertinggi mencapai 95%, sedangkan tingkat kemiripan terendah adalah 80%. Rata-rata kemiripan yang diperoleh sebesar 80%, sementara perbedaan rata-ratanya mencapai 20%.

### **4.3 Kesimpulan Hasil Pengujian**

Kesimpulan hasil pengujian aplikasi terhadap gambar untuk mengetahui kemiripan dan perbedaan yaitu:

1. Aplikasi yang telah dibuat menggunakan metode *Cosine Similarity* untuk mengetahui persamaan dan perbedaan dua gambar.
2. Aplikasi yang telah dibuat dapat membandingkan persamaan dan perbedaan dua buah gambar.
3. Proses pelaksanaan tidak memakan waktu lama.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Berdasarkan pembahasan pada bab-bab sebelumnya yang telah dilakukan maka dapat diambil beberapa kesimpulan sebagai berikut :

1. Aplikasi yang menerapkan metode Cosine Similarity untuk mendeteksi kemiripan dan perbedaan pada citra gambar hasil scan berbasis Android memungkinkan identifikasi yang akurat terhadap gambar-gambar yang terlihat serupa, sehingga dapat menghindari kesalahan persepsi oleh mata manusia.
2. Dengan memanfaatkan nilai warna gambar dalam format RGB, metode Cosine Similarity dapat diterapkan untuk mengidentifikasi kemiripan dan perbedaan pada citra gambar hasil scan.
3. Dengan memanfaatkan pemrograman Android, sebuah aplikasi untuk mendeteksi gambar hasil scan dengan menggunakan metode Cosine Similarity dapat dikembangkan berbasis Android.

#### **5.2 Saran**

Untuk pengembangan lebih lanjut dari aplikasi deteksi gambar hasil scan yang menggunakan metode Cosine Similarity berbasis Android, beberapa saran berikut dapat dipertimbangkan:

1. Sebaiknya, aplikasi yang telah dikembangkan dapat memanfaatkan dua metode berbeda untuk memastikan hasil perbandingan yang lebih akurat.



2. Disarankan agar aplikasi yang telah dikembangkan mampu menggunakan berbagai gambar untuk menganalisis kemiripan.
3. Sebaiknya aplikasi yang telah dikembangkan menyertakan informasi mengenai gambar asli serta gambar yang sedang dianalisis.

## DAFTAR PUSTAKA

- Aditya Fajar Ramadhan, Ade Dwi Putra, & Ade Surahman. (2021). Aplikasi Pengenalan Perangkat Keras Komputer Berbasis Android Menggunakanaugmented Reality (Ar). *Jurnal Teknologi Dan Sistem Informasi (JTSI)*, 2(2), 24–31.
- Ali, M. (2021). Peningkatkan Kemampuan Membaca Dan Menulis Permulaan Dengan Media Gambar Untuk Kelas 2 Pada Sdn 93 Palembang. *PERNIK : Jurnal Pendidikan Anak Usia Dini*, 4(1), 43–51. <https://doi.org/10.31851/pernik.v4i1.6796>
- Andikos, A. F. (2019). Perancangan Aplikasi Multimedia Interaktif Sebagai Media Pembelajaran Pengenalan Hewan Pada Tk Islam Bakti 113 Koto Salak. (*Indonesia Jurnal Sakinah*) *Jurnal Pendidikan Dan Sosial Islam*, 1(1), 34–49.
- AZMI, F., Saleh, A., & Dharshinni, N. P. (2020). Face Identification on Login Security Using Algorithm Combination of Viola-Jones and Cosine Similarity. *Journal of Informatics and Telecommunication Engineering*, 4(1), 203–211. <https://doi.org/10.31289/jite.v4i1.3885>
- Budiyanto, A. (2023). *Perancangan Aplikasi Pembukuan Keuangan Warung Sembako Jakarta Timur Berbasis Manajemen Keuangan dengan Android*. 7(1), 90–94.
- Kuswanto, J. (2020). Media Pembelajaran Berbasis Android Mata Pelajaran Desain Grafis Kelas X. *Edutic - Scientific Journal of Informatics Education*, 6(2), 78–84. <https://doi.org/10.21107/edutic.v6i2.7073>
- Lindang, D. N., Muniar, A. Y., Halid, A., Muhajirin, M., & Amiruddin, A. (2022). Sistem Penentuan Kemiripan Antar Skripsi Menggunakan Metode Cosine Similarity Pada Perpustakaan. *Sntei*, 321–324.
- Niko, N. S. A. (2022). Implementasi Metode Levensthein Distance Dan Cosine Similarity Untuk Deteksi Kemiripan Gambar. *RJOCS (Riau Journal of Computer Science)*, 8(2), 85–93. <https://doi.org/10.30606/rjocs.v8i2.1415>
- Pratiwi, F., Sukadana, I. K., & Seputra, I. P. G. (2020). Penggunaan Materai yang Di Scan pada Surat Kuasa di Bawah Tangan dalam suatu Perjanjian. *Jurnal Konstruksi Hukum*, 1(1), 98–102. <https://doi.org/10.22225/jkh.1.1.2312.98-102>
- Puspitadewi, G. C. (2021). Sadar Arsip Dimulai Dari Pengelolaan Arsip Pribadi. *LibTech: Library and Information Science Journal*, 1(2). <https://doi.org/10.18860/libtech.v1i2.11923>
- Rahmawati, E., & Rahayu, G. D. S. (2021). Implementasi Model Discovery Learning Berbasis Media Gambar Dalam Meningkatkan Kecerdasan Ekologis Siswa Sekolah Dasar. *Journal of Elementary Education*, 04(02), 240–248.
- Sihombing, V., & Yanris, G. J. (2020). Penerapan Aplikasi Dalam Mengolah Aset Desa (Studi Kasus : Kepenghuluan Sri Kayangan) 1). *Terakreditasi DIKTI*, 4(1), 12–15.

- Siregar, F. A., Hutagalung, F. S., & Riza, F. (2023). Perancangan Sistem Arsip Berbasis Online Pada BPP Medan Krio. *Jurnal Abdimas Budi Darma* , 3(2), 33–38. <https://ejurnal.stmik-budidarma.ac.id/index.php/abdimas>
- Siswanto, E., & Ceng Giap, Y. (2020). Implementasi Algoritma Rabin-Karp dan Cosine Similarity untuk Pendeteksi Plagiarisme Pada Dokumen. *Jurnal Algor*, 1(2), 16–22.
- Subekti, Z. M. (2021). *Rancang Bangun Sistem Ujian Online Menggunakan*. 2, 163–168.
- Suhirja, R., & Jemakmun, J. (2022). Sistem Deteksi Masker Menggunakan Algoritma Haar Cascade dalam Menghadapi Era New Normal. *Smatika Jurnal*, 12(02), 222–232. <https://doi.org/10.32664/smatika.v12i02.702>

## LISTING PROGRAM

### 1. Main.java

```
package cosine.similarity;

/*Librari*/
import android.app.Activity;
import android.os.Bundle;
import android.content.Intent;
import android.database.Cursor;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Color;
import android.graphics.drawable.BitmapDrawable;
import android.net.Uri;
import android.provider.MediaStore;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.EditText;

public class main extends Activity {
    /*Pembuatan nama komponen*/
    private static int LOAD_IMAGE_RESULT=1;
    private Button button;
    private ImageView image;
    private ImageView image2;
    private Button button2;
    private Button buttonA;
    private EditText edittext1;
    private EditText edittext2;
    private Bitmap bitmap;
    private Bitmap bitmaps;
    public double sim=0;
    public double sims=0;
    public String isi="";

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        /*Menghubungkan java dengan xml*/
        button=(Button)findViewById(R.id.button);
        button2=(Button)findViewById(R.id.button2);
        buttonA=(Button)findViewById(R.id.buttonA);
```

```

image=(ImageView)findViewById(R.id.image);
image2=(ImageView)findViewById(R.id.image2);
edittext1=(EditText)findViewById(R.id.editText1);
edittext2=(EditText)findViewById(R.id.editText2);
/* Aksi Klik*/
button.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        isi="button";
        Intent i=new Intent(Intent.ACTION_PICK,
android.provider.MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
        startActivityForResult(i, LOAD_IMAGE_RESULT);
    }
});
/* Aksi Klik*/
button2.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        isi="button2";
        Intent i=new Intent(Intent.ACTION_PICK,
android.provider.MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
        startActivityForResult(i, LOAD_IMAGE_RESULT);
    }
});
/* Aksi Klik*/
buttonA.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        try {
            /*mengatur lokasi gambar pertama*/
            BitmapDrawable bmd =
(BitmapDrawable) image.getDrawable();
            bitmap = bmd.getBitmap();
            /*mengatur lokasi gambar kedua*/
            BitmapDrawable bmd2 =
(BitmapDrawable) image2.getDrawable();
            bitmaps = bmd2.getBitmap();
            int uR = 0;
            int uG = 0;
            int uB = 0;
            int uTtl = 0;

            int red=0;
            int reds=0;
            int green=0;
            int greens=0;
            int blue=0;
            int blues=0;

            double simR = 0;
            double simG = 0;

```

```

double simB = 0;
double bR = 0;
double bG = 0;
double bB = 0;
double distance = 0;
double dist = 0;
/*mengubah gambar Sebagai kotak pixel*/
for (int i = 0; i < bitmap.getWidth(); i++)
{
    for (int j = 0; j < bitmap.getHeight(); j++) {
        int p = bitmap.getPixel(i , j);
        int q = bitmaps.getPixel(i , j);
        red=Color.red(p);
        green=Color.green(p);
        blue=Color.blue(p);
        reds=Color.red(q);
        greens=Color.green(q);
        blues=Color.blue(q);

        uTtl++;
        if(red!=reds){
            uR=uR+0;
        }
        else{
            uR=uR+1;
        }
        if(green!=greens){
            uG=uG+0;
        }
        else{
            uG=uG+1;
        }
        if(blue!=blues){
            uB=uB+0;
        }
        else{
            uB=uB+1;
        }
    }
}
bR=Math.sqrt(uTtl)*(Math.sqrt(uR));
bG=Math.sqrt(uTtl)*(Math.sqrt(uG));
bB=Math.sqrt(uTtl)*(Math.sqrt(uB));

```

```

/*edittext1.setText(edittext1.getText()+"\nAtas:"+String.valueOf(uR)+"\n
Bawah"+String.valueOf(bR));*/

        simR=uR/bR;
        simG=uG/bG;
        simB=uB/bB;

distance=Math.ceil(((simR+simG+simB)/3)*100)-1;
        dist=100-distance;

edittext1.setText(String.valueOf(distance)+"%");

edittext2.setText(String.valueOf(dist)+"%");
        } catch (Exception e) {

        }

    }

    private int applyDegree(int red) {
        // TODO Auto-generated method stub
        return 0;
    }

});
}

/*Upload Gambar*/
protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
    super.onActivityResult(requestCode, resultCode, data);

if(requestCode==LOAD_IMAGE_RESULT&&resultCode==RESULT_OK&&d
ata!=null){
    Uri pickedImage=data.getData();
    /*Buka media gambar di HP*/
    String[] filePath={MediaStore.Images.Media.DATA};
    /*Lokasi gambar di HP*/
    Cursor cursor=getContentResolver().query(pickedImage, filePath, null,
null, null);
    cursor.moveToFirst();
    /*Penamaan*/
    String imagePath=cursor.getString(cursor.getColumnIndex(filePath[0]));
    /*Jika klik tombol upload gambar 1*/
    if(isi=="button"){
        image.setImageBitmap(BitmapFactory.decodeFile(imagePath));
    }
}
}

```

```

    /*Jika klik tombol upload gambar 2*/
    if(isi=="button2"){
        image2.setImageBitmap(BitmapFactory.decodeFile(imagePath));
    }
    cursor.close();

    }
    }

}

```

## 2. Main.xml

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/waw"
    tools:context="${relativePackage}.${activityClass}" >

```

```

<ImageView
    android:id="@+id/image"
    android:layout_width="125dp"
    android:layout_height="200dp"
    android:layout_alignParentLeft="true"
    android:layout_below="@+id/button"
    android:layout_marginLeft="30dp"
    android:layout_marginTop="16dp" />

```

```

<ImageView
    android:id="@+id/image2"
    android:layout_width="125dp"
    android:layout_height="200dp"
    android:layout_alignTop="@+id/image"
    android:layout_toRightOf="@+id/image" />

```

```

<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/image"
    android:layout_alignParentTop="true"
    android:layout_alignRight="@+id/image"
    android:text="Upload Gambar 1" />

```

```

<Button
    android:id="@+id/button2"

```



```

    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBottom="@+id/button"
    android:layout_alignLeft="@+id/image2"
    android:layout_alignRight="@+id/image2"
    android:text="Upload Gambar 2" />

```

```

<EditText
    android:id="@+id/editText1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/image"
    android:layout_below="@+id/image"
    android:layout_toLeftOf="@+id/image2"
    android:ems="10"
    android:hint="Kemiripan" >

```

```

    <requestFocus />

```

```

</EditText>

```

```

<EditText
    android:id="@+id/editText2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBaseline="@+id/editText1"
    android:layout_alignBottom="@+id/editText1"
    android:layout_alignLeft="@+id/image2"
    android:layout_alignRight="@+id/image2"
    android:ems="10"
    android:hint="Perbedaan" />

```

```

<Button
    android:id="@+id/buttonA"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/editText1"
    android:layout_alignRight="@+id/editText2"
    android:layout_below="@+id/editText1"
    android:layout_marginTop="21dp"
    android:text="SUBMIT" />

```

```

</RelativeLayout>

```