

## BAB IV

### PERANCANGAN

Dalam sebuah penelitian proses perancangan aplikasi dilakukan sebagai tahap awal dalam pengembangan aplikasi. Pembuatan rancangan yang baik menjadi kunci keberhasilan pengembangan aplikasi, sehingga tahap perancangan memiliki peranan yang sangat penting. Pada bab ini akan akan dijelaskan tentang tahapan-tahapan yang dilakukan dalam merancang sebuah sistem atau aplikasi yang efektif dan efisien.

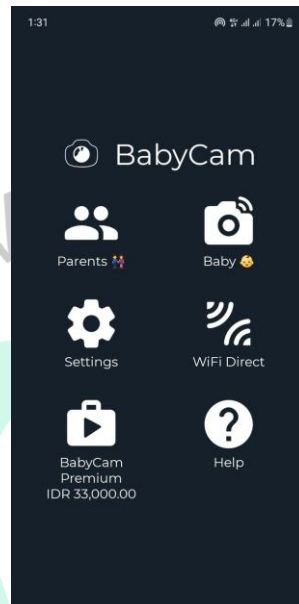
#### 4.1. Analisis Sistem Terdahulu

Pada penelitian ini dilakukan analisa terhadap dua sistem terdahulu sebagai pembandingan, yaitu Aplikasi BabyCam yang dapat diunduh melalui *Application Store* dan aplikasi hasil dari penelitian yang dilakukan pada tahun 2020 oleh Muchamad Fajri Amirul Nasrullah, Yulinda Sigalingging, Fandy Neta, dkk yang berjudul “Deteksi Pergerakan Bayi Berbasis Color Filtering Menggunakan Aforge.NET”.

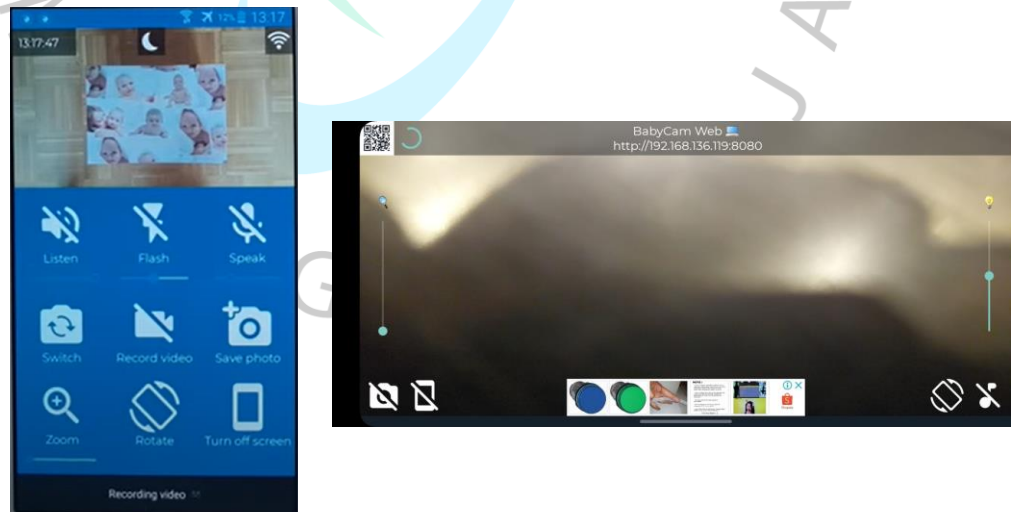
Pada aplikasi BabyCam, fitur yang tersedia hanya fitur untuk memantau bayi secara daring dari dua perangkat ponsel yang terhubung menggunakan *tethering* atau koneksi *wi-fi* dari salah satu perangkat. Fitur lainnya dalam aplikasi ini memungkinkan orang tua untuk melakukan aktivitas, seperti mendengarkan suara dari perangkat bayi, berbicara kepada perangkat bayi, menyalakan sebuah lagu, dan sebagainya. Dibandingkan dengan aplikasi yang dirancang dari penelitian ini, fitur pendeteksi pergerakan bayi akan lebih berguna demi keselamatan dan kenyamanan bayi.

Pada aplikasi hasil dari penelitian “Deteksi Pergerakan Bayi Berbasis Color Filtering Menggunakan Aforge.NET” menggunakan metode *color filtering* dalam pengimplementasiannya. Menurut Sharma K, dkk. *Color filtering* adalah teknik yang memungkinkan identifikasi objek berdasarkan warnanya. Dengan demikian, pemisahan objek dari latar belakang menjadi lebih mudah. Metode *color filtering* memiliki kekurangan dan kelebihan, kekurangannya yaitu kurangnya sensitif terhadap perubahan dalam citra dan kelebihan teknik *color filtering* lebih cepat

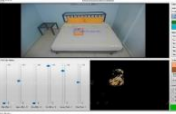
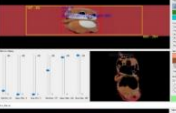
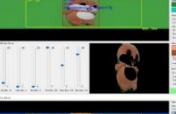

dalam pemrosesan citra, karena hanya membutuhkan satu operasi pem-*filter*-an untuk mengisolasi objek berdasarkan warna. Namun dengan metode *color filtering* jika warna objek yang ingin diisolasi terlalu mirip dengan warna latar belakang dapat menghasilkan banyak kesalahan pada luaran.



Gambar 4.1 Antarmuka Aplikasi BabyCam



Gambar 4.2 Antarmuka Aplikasi BabyCam Saat Memonitoring Bayi

Warna	Terlacak?	Keterangan	Gambar
Latar Putih	Ya	Sukses Terlacak	
Latar Merah	Ya	Sukses Terlacak	
Latar Hijau	Ya	Sukses Terlacak	
Latar Biru	Ya	Sukses Terlacak	
Error	Tidak	Gagal Terlacak	

Gambar 4.3 Hasil Pengujian Berbagai Warna Latar Tempat Tidur Bayi

## 4.2. Spesifikasi Kebutuhan Sistem Baru

Dalam upaya memudahkan proses pengembangan aplikasi serta memenuhi kebutuhan aplikasi yang akan dibangun, diperlukan spesifikasi aplikasi yang lengkap dan terperinci. Komponen yang tercakup dalam spesifikasi aplikasi meliputi proses aplikasi, kebutuhan perangkat keras dan lunak, serta data yang digunakan.

### 4.2.1 Spesifikasi Proses Aplikasi

Spesifikasi Proses Aplikasi merupakan penjelasan terperinci tentang tahapan yang dilakukan dalam penggunaan aplikasi yang dibangun agar proses penyelesaian masalah dapat dilakukan dengan baik, benar dan konsisten. Dari mulai proses masukan data sampai keluarnya hasil dari aplikasi yang dikembangkan. Spesifikasi proses pada aplikasi melibatkan serangkaian tahapan proses yang harus dilakukan secara terurut, sebagaimana yang dijelaskan pada Gambar 4.4.



Gambar 4.4 Tahapan Proses Aplikasi

- 1) Tahap Pemberian Input: Pada tahap ini, pengguna harus dapat memasukan masukan (*input*) berupa video rekaman objek bayi.
- 2) Tahap Konfigurasi: Pada tahap ini, pengguna juga harus mengkonfigurasi beberapa variabel seperti batas yang harus ditetapkan.
- 3) Hasil luaran: Pada tahap ini, aplikasi akan memberikan luaran berupa notifikasi pesan dan suara peringatan yang menandakan bahwa objek telah melewati batas yang ditetapkan dan memberikan tanda.

#### 4.2.2 Spesifikasi Kebutuhan Perangkat Lunak

Untuk mendukung pengembangan dan pengoperasian aplikasi, diperlukan berbagai *tools* dan teknologi dalam bentuk perangkat lunak, seperti bahasa pemrograman, perangkat lunak berupa IDE (*Integrated Development Environment*), dan lain sebagainya. Yang dirincikan dalam Tabel 4.1.:

Tabel 4.1. Spesifikasi Kebutuhan Perangkat Lunak

No	Kategori	Perangkat Lunak
1	Bahasa Pemograman	Python
2	IDE ( <i>Integrated Development Environment</i> )	IntelliJ
3	Pustaka	matplotlib dan numpy

#### 4.2.3 Spesifikasi Kebutuhan Perangkat Keras

Selain kebutuhan akan perangkat lunak, spesifikasi tertentu atas kebutuhan perangkat keras untuk menunjang pengembangan dan pengoperasian aplikasi. Berikut rincian spesifikasi kebutuhan perangkat keras dalam Tabel 4.2.:

Tabel 4.2. Spesifikasi Kebutuhan Perangkat Keras

No	Perangkat Keras	Spesifikasi
1	Laptop	<b>Prosesor:</b> 8th Gen Intel Core i7-8750H <b>Memory:</b> 16GB DDR4 2666Mhz <b>Sistem Operasi:</b> Windows 11 Home

#### 4.2.4 Spesifikasi Data (Input Video)

Penelitian ini menggunakan data dengan format video sebagai data yang akan diolah oleh aplikasi. Video yang dibutuhkan untuk aplikasi dalam penelitian ini juga memiliki beberapa kriteria, antara lain:

1) Statis

Video yang digunakan harus statis atau tidak bergerak, sehingga objek yang bergerak dapat terdeteksi dengan jelas.

2) Kamera yang stabil

Kamera yang digunakan untuk merekam video harus stabil dan tidak goyang, karena pergerakan kamera dapat mempengaruhi hasil pengolahan video.

3) Posisi Kamera dan Objek

Posisi pengambilan video harus tepat di atas objek agar posisi objek dapat terlihat dengan jelas di dalam video dengan jarak yang tidak terlalu jauh dan tidak terlalu dekat.

4) Kualitas video yang baik

Video yang digunakan harus memiliki kualitas yang baik dan jelas, sehingga objek yang bergerak dapat terdeteksi dengan tepat.

5) Cahaya yang cukup stabil

Video harus direkam dengan pencahayaan yang baik, sehingga objek yang bergerak dapat terdeteksi dengan jelas.

6) Warna yang kontras





Sistem mendeteksi perbedaan warna pada tiap piksel, maka warna objek yang bergerak dan warna latar belakang yang digunakan harus berbeda atau kontras, sehingga aplikasi dapat melakukan deteksi dengan akurat.

7) Ukuran video yang sesuai

Ukuran video yang digunakan harus sesuai dengan kebutuhan aplikasi, tidak terlalu besar atau terlalu kecil untuk meminimalisir waktu komputasi.

Untuk memperjelas penjelasan mengenai syarat data yang dibutuhkan untuk aplikasi pada penelitian ini, berikut klasifikasi video berdasarkan kriteria yang dibutuhkan pada Tabel 4.3

Tabel 4.3. Pengklasifikasian Data Video

No	Tangkapan Layar	Penjelasan
1		<b>Video tidak dapat digunakan</b> karena tidak memenuhi aspek: Kualitas video yang baik dan Cahaya yang cukup stabil
2		<b>Video tidak dapat digunakan</b> karena tidak memenuhi aspek: Kualitas video yang baik, Cahaya yang cukup stabil, Latar belakang yang kontras dan Warna yang kontras
3		<b>Video dapat digunakan</b> karena memenuhi semua aspek yang dibutuhkan
4		<b>Video dapat digunakan</b> karena memenuhi semua aspek yang dibutuhkan

### 4.3. Perancangan sistem

Dalam upaya menjelaskan rancangan sistem atau aplikasi yang dikembangkan, sub bab ini akan membahas secara detail mengenai diagram alir aplikasi, use case, *activity diagram*, *sequence diagram*, serta rancangan tampilan antarmuka yang diterapkan. Diharapkan dengan adanya bab perancangan aplikasi ini, peneliti dapat menjelaskan secara komprehensif mengenai bagaimana aplikasi yang dirancang bekerja dalam melakukan tugas-tugasnya.



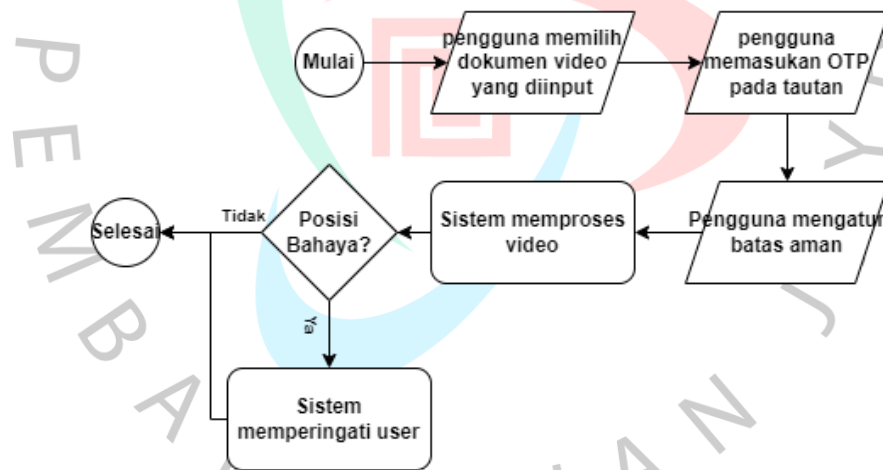
Gambar 4.5 Diagram Alir Proses Perancangan

Gambar 4.5 merupakan diagram alir yang menggambarkan proses perancangan. Dimulai dari mengidentifikasi kebutuhan dan tujuan sistem yang akan

dibangun, menganalisis sistem yang ada atau proses yang akan digantikan untuk mengidentifikasi kelemahan dan peluang, merancang konsep sistem secara keseluruhan dengan membuat model konseptual yang menggambarkan fungsi dan interaksi dalam sistem, dan merinci desain sistem dengan membuat spesifikasi teknis untuk setiap komponen, antarmuka pengguna, desain basis data, dan algoritma pemrosesan.

#### 4.3.1 Diagram Alir Aplikasi

Diagram Alir merupakan salah satu jenis diagram yang dapat digunakan untuk merepresentasikan alur kerja dari suatu aplikasi atau prosedur dengan jelas secara visual. Diagram ini sering dipergunakan untuk memvisualisasikan suatu algoritma atau prosedur dengan cara yang mudah dipahami dan jelas, sehingga dapat membantu memudahkan pemahaman konsep dan kerangka kerja dari suatu aplikasi atau prosedur.



Gambar 4.6 Diagram Alir Penggunaan Aplikasi

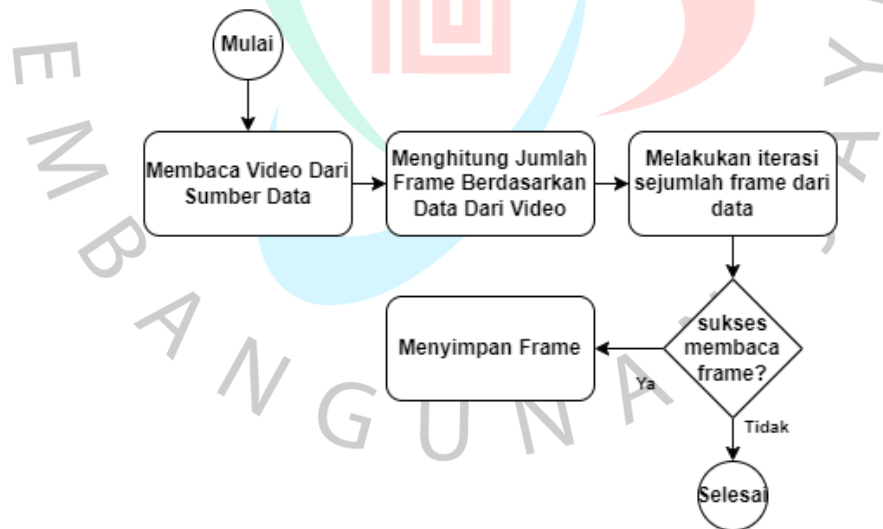
Gambar 4.6 merupakan diagram alir penggunaan dari aplikasi pendeteksi posisi bayi. Dimulai dengan pengguna memilih dokumen video serta mengatur batas aman dari daerah yang ingin ditetapkan. Setelah itu, aplikasi akan membaca video tersebut dan memprosesnya. Hasil dari proses aplikasi memungkinkan terdapat dua bentuk luaran, yakni hasil proses aplikasi menunjukkan posisi bayi dalam posisi bahaya atau dalam posisi aman. Apabila posisi bayi terdeteksi dalam posisi bahaya, sistem akan memberikan sebuah peringatan kepada pengguna.

### 4.3.2 Diagram Alir Pembuatan Model



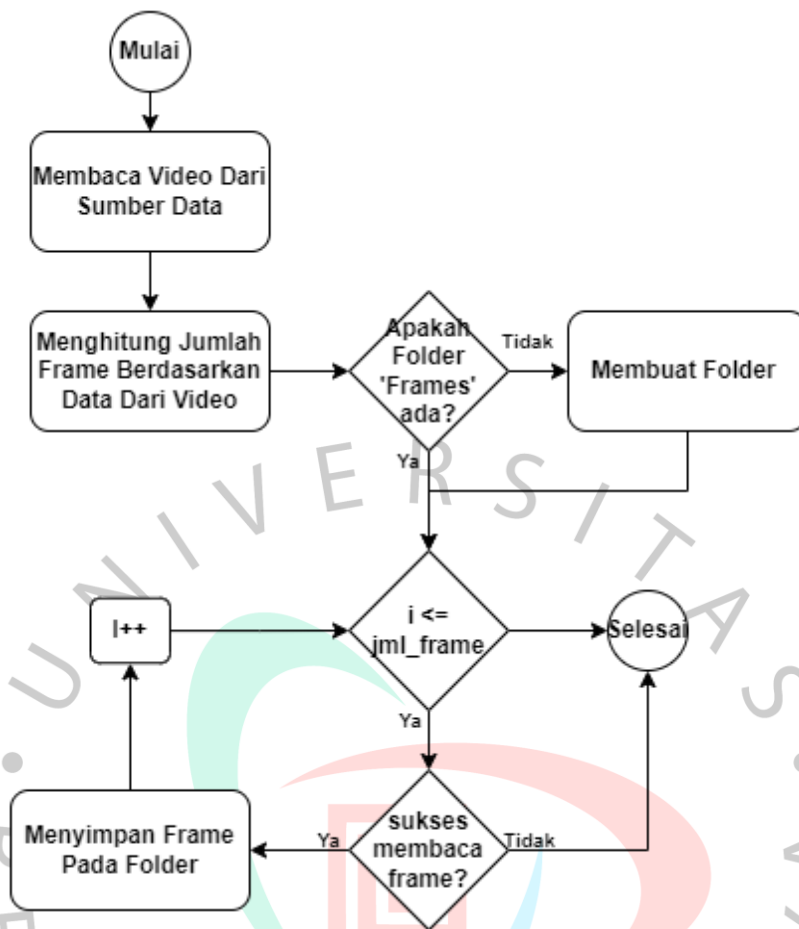
Gambar 4.7 Diagram Alir Model Aplikasi

Dalam penelitian ini, seperti yang digambarkan dalam Gambar 4.7 proses pembuatan model aplikasi meliputi pencacahan video, pendeteksian gerak, serta melakukan pemindaian untuk mendeteksi posisi objek yang bergerak di daerah tertentu dan deteksi bahaya.



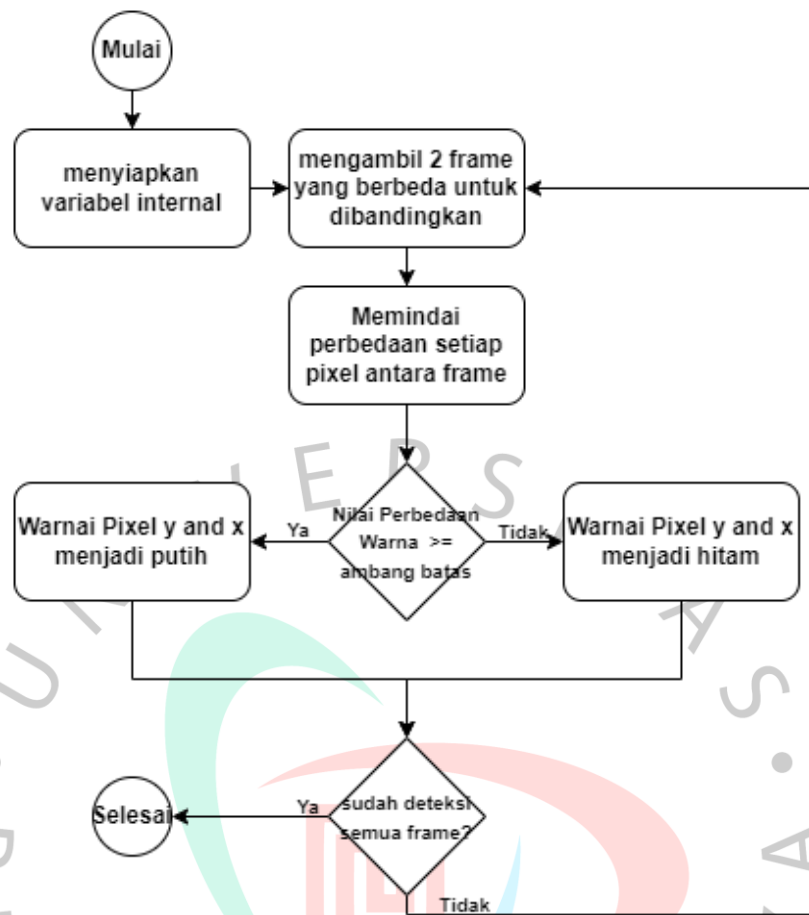
Gambar 4.8 Diagram Alir Proses Pencacahan Video





Gambar 4.9 Diagram Alir Representasi Kode Pencacah Video

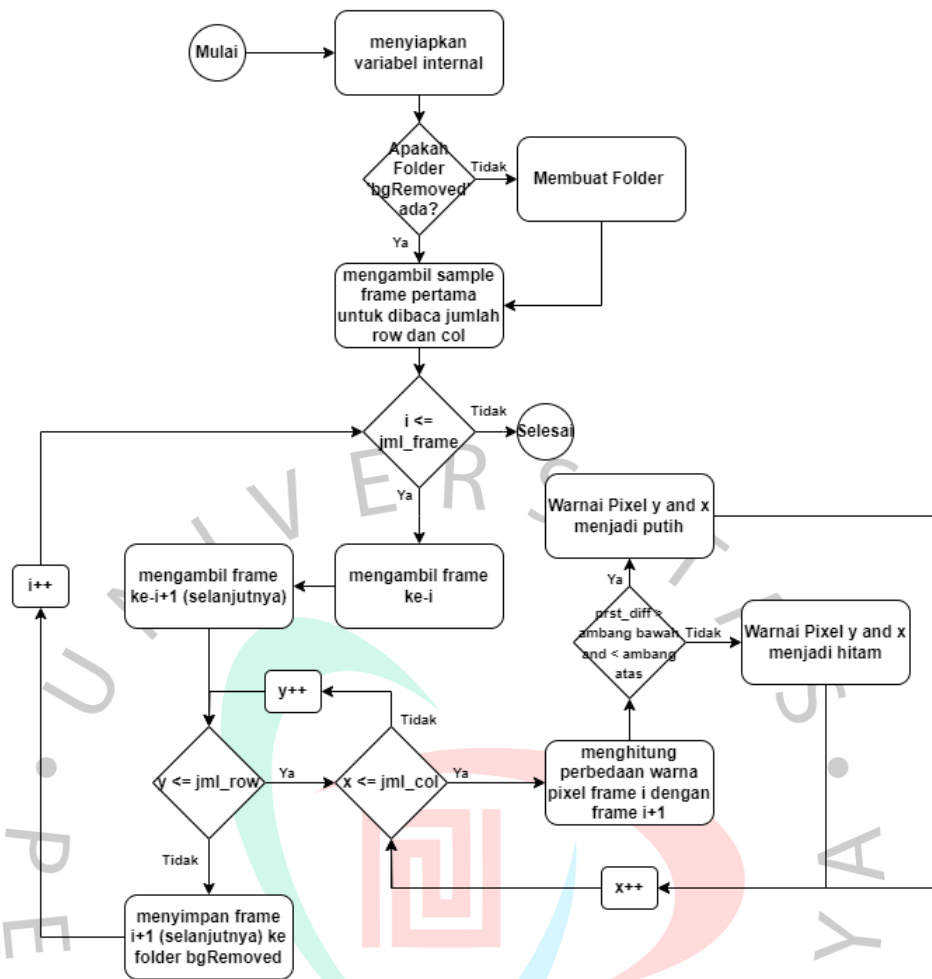
Langkah awal dalam pemrosesan pada aplikasi dalam penelitian ini ditunjukkan pada Gambar 4.8 yaitu proses pencacahan video. Video yang sudah dibaca berapa banyak *frame* dalam rentang waktu tertentu akan melalui proses pengulangan sampai banyaknya *frame* untuk dibaca lalu disimpan ke dalam suatu direktori dengan nama 'Frames' dan format jpg.



Gambar 4.10 Diagram Alir Proses Pendeteksian gerak

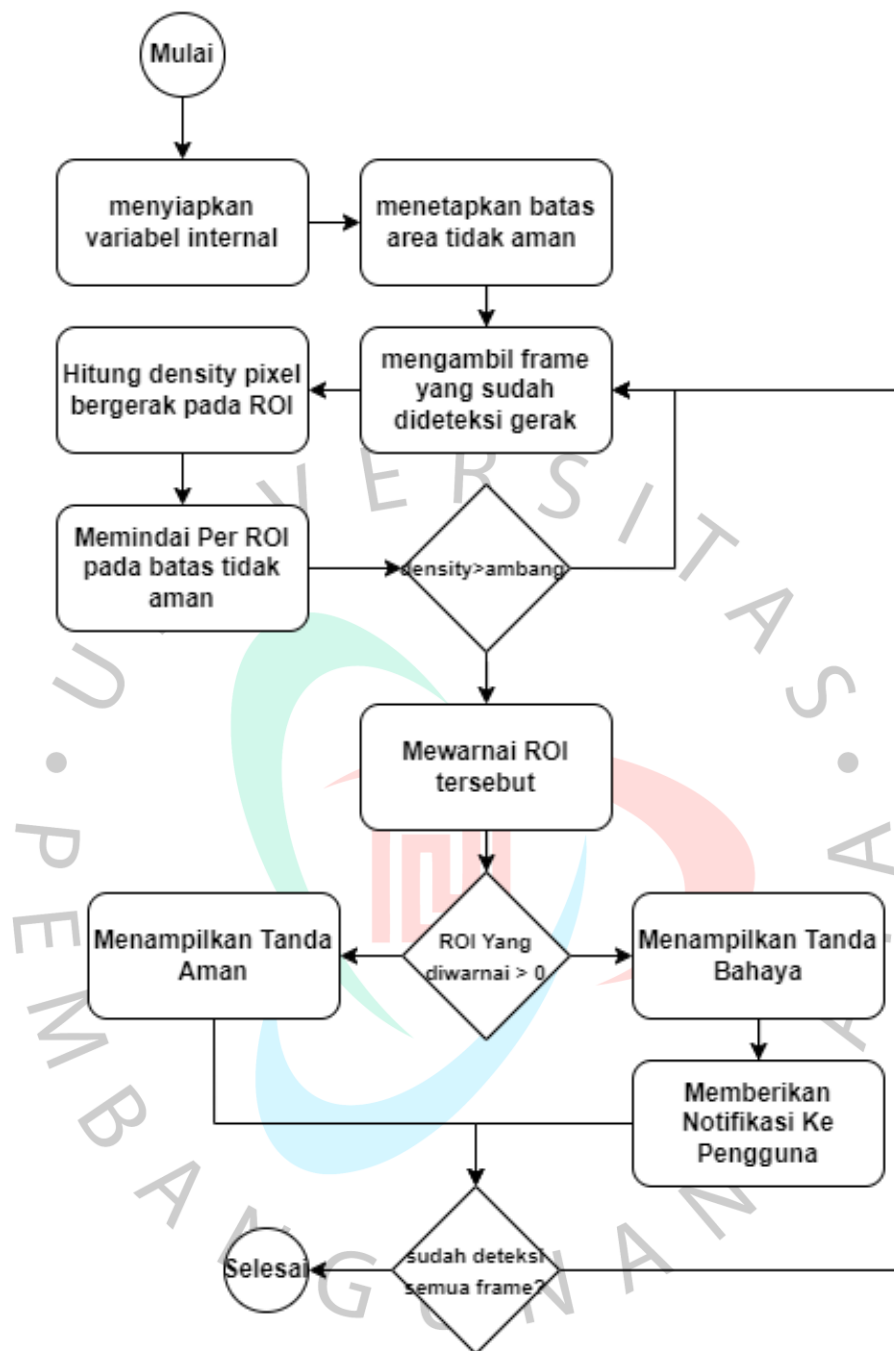
Dalam proses pendeteksian gerak, warna pada setiap piksel suatu frame dibandingkan dengan piksel yang sesuai pada frame berikutnya. Perbedaan warna antara dua piksel tersebut dianggap sebagai indikasi terjadinya gerakan. Jika terdeteksi adanya perbedaan yang signifikan antara dua piksel, frame tersebut akan dianggap sebagai frame yang mengalami gerakan.

Setelah proses pendeteksian gerak dilakukan, frame yang mengalami gerakan atau perubahan warna akan disimpan dalam sebuah direktori atau folder khusus. Hal ini memungkinkan pengguna atau sistem untuk mengakses dan menganalisis frame-frame tersebut secara terpisah untuk keperluan lebih lanjut, seperti identifikasi objek yang bergerak atau pemrosesan lanjutan.



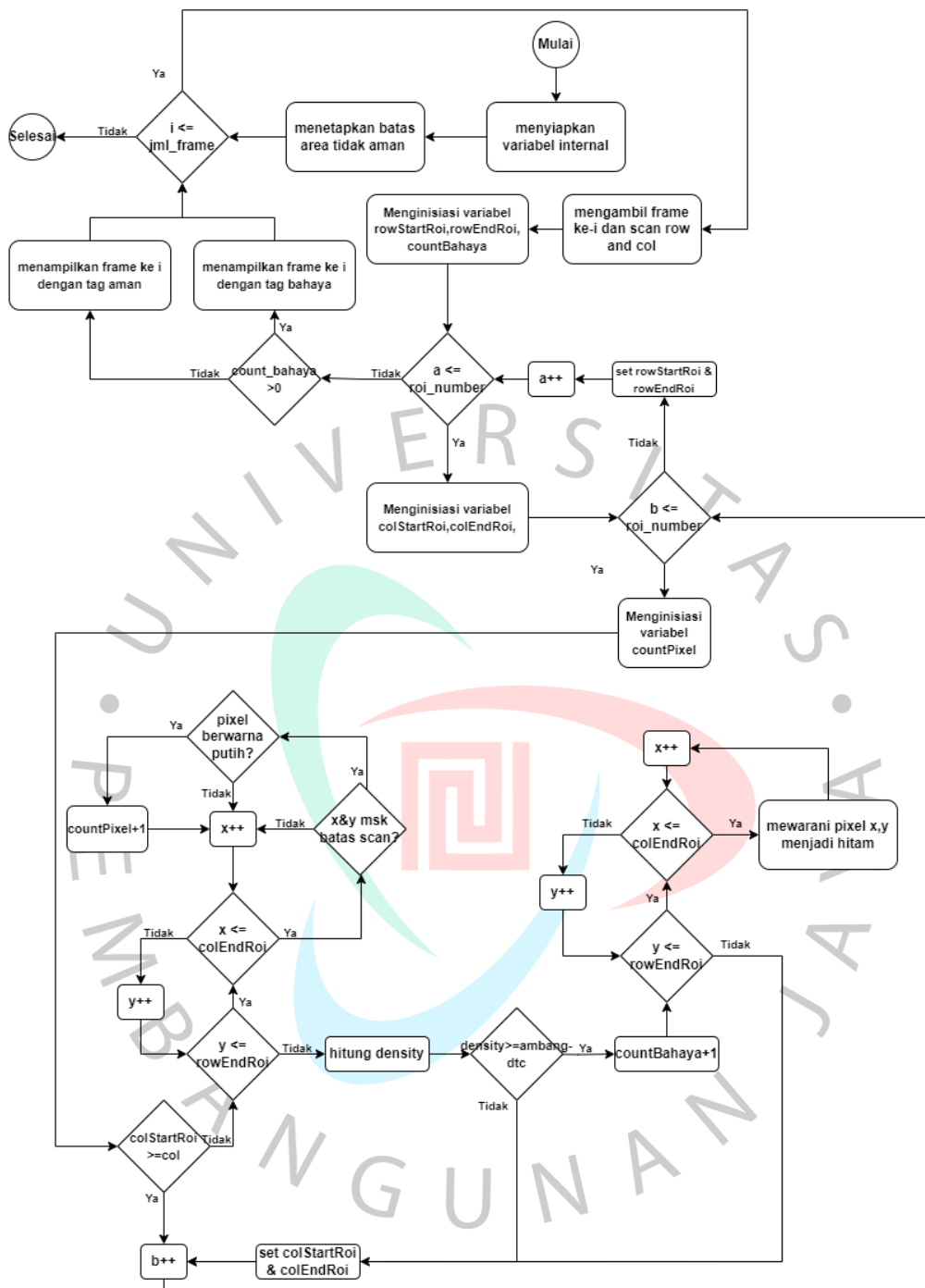
Gambar 4.11 Diagram Alir Representasi Kode Pendeteksian gerak

Dapat dilihat pada Gambar 4.10 aplikasi akan melakukan proses pengulangan untuk mengambil *frame* yang sudah disimpan pada proses pencacahan video. Setelah itu, pada *frame* dengan *frame* berikutnya akan dibandingkan pada setiap piksel-nya, apabila terjadi perubahan warna menggunakan ukuran persentase terhadap ambang batas dan ambang bawah yang sudah ditetapkan, maka piksel tersebut akan diubah warnanya menjadi putih. Jika tidak terjadi perubahan warna, maka piksel tersebut akan diubah menjadi hitam, setelah proses perbandingan dan pendeteksian gerak, *frame* tersebut akan disimpan dalam direktori yang berbeda dengan direktori sebelumnya dengan nama 'bgRemoved' dalam format jpg.



Gambar 4.12 Diagram Alir Proses Deteksi Posisi / Deteksi Bahaya

Setelah proses pendeteksian gerak dan data dari *frame* yang sudah berhasil dideteksi piksel yang memenuhi kriteria bergerak dan disimpan dalam sebuah direktori, seperti penggambaran pada Gambar 4.12 maka akan dilakukan pendeteksian kembali dari *frame* yang sudah dideteksi gerak.



Gambar 4.13 Diagram Alir Representasi Kode Deteksi Posisi / Deteksi Bahaya

Alur proses pada tahapan dimulai dengan membaca *frame* yang sudah disimpan pada proses sebelumnya. Selanjutnya, dilakukan pembuatan batas-batas area tertentu pada gambar berdasarkan masukan yang telah ditentukan oleh pengguna. Setelah itu, sistem akan menghitung area yang harus di-pindai

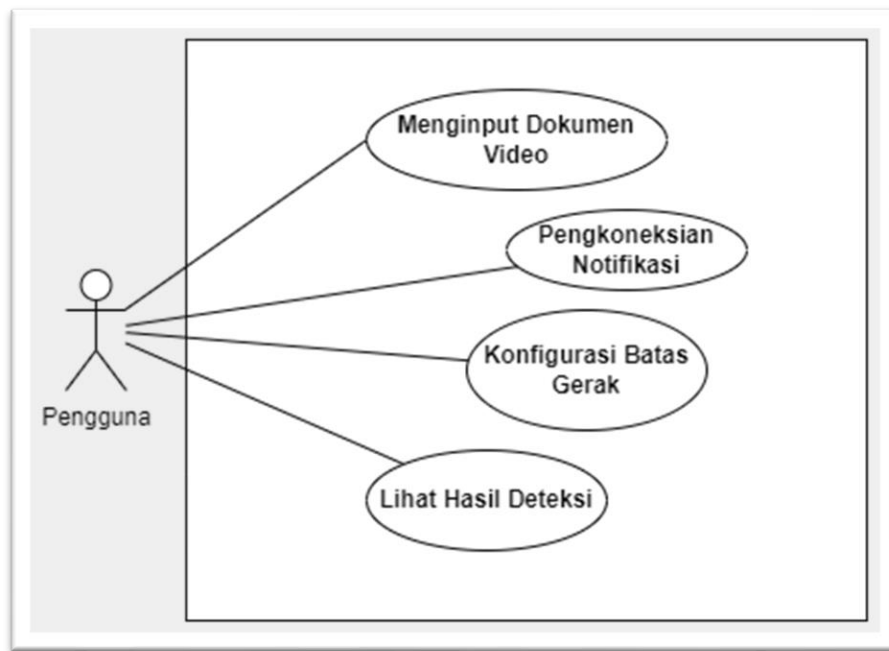
berdasarkan persentase yang telah ditentukan dari jarak area batas yang dimasukkan oleh pengguna. Kemudian, dilakukan pemindaian pada area yang telah ditentukan atau dapat disebut sebagai *region of interest* (ROI) pada *frame* tersebut. ROI pada *frame* akan diproses dengan menggunakan perulangan. Batas deteksi pada ROI ditentukan menggunakan parameter batas atas, bawah, kiri, dan kanan berdasarkan perhitungan persentase pindai yang didapatkan.

Aplikasi dapat mengkatagorikan objek dapat dibidang memasuki area berbahaya dengan menghitung jumlah piksel yang berwarna putih pada area yang sedang diproses. Setiap ROI kemudian dihitung densitasnya, dan jika nilai densitas melebihi ambang batas yang telah ditentukan, maka ROI tersebut dianggap berbahaya dan dilakukan pewarnaan tertentu pada ROI tersebut. Selanjutnya, proses tersebut akan diulangi hingga seluruh *frame* pada direktori tersebut telah diproses.

#### 4.3.3 Use Case

Pemanfaatan *use case* diagram dalam perancangan aplikasi merupakan salah satu cara yang berguna dalam menggambarkan interaksi antara aktor dengan aplikasi, serta berbagai fungsi atau fitur yang tersedia pada aplikasi yang dibangun. Dalam konteks penelitian ini, *use case* dipergunakan untuk mengilustrasikan berbagai aktivitas atau tindakan yang dapat dilakukan oleh pengguna ketika berinteraksi dengan aplikasi yang telah dibangun. Dengan menggunakan *use case* diagram, pengguna dapat memahami secara rinci dan jelas mengenai fungsionalitas aplikasi dan bagaimana interaksi antara admin dengan aplikasi dapat berlangsung secara efektif.

*Use case* diagram menggunakan notasi-notasi khusus untuk memodelkan berbagai kasus penggunaan atau skenario yang mungkin terjadi. Diagram ini menggambarkan aktor-aktor yang terlibat, tindakan-tindakan yang dapat dilakukan oleh aktor tersebut, serta respons yang diharapkan dari aplikasi.



Gambar 4.14 Use Case Aplikasi

Pada Gambar 4.14 digambarkan bahwa pengguna aplikasi pendeteksi posisi bayi dapat memanfaatkan fitur yang tersedia, seperti mengatur masukan video, mengkonfigurasi batas aman yang ditetapkan sesuai kehendaknya, dan melihat hasil luaran deteksi dari sistem yang dapat menunjukkan bahwa bayi dalam posisi aman atau tidak.

Pemanfaatan skenario *use case* dalam perancangan aplikasi merupakan alat untuk menggambarkan lebih rinci prosedur dari sebuah *use case* agar dapat lebih mendalami interaksi antara pengguna dan aplikasi. Berikut, akan dijelaskan secara terperinci diuraikan setiap skenario yang telah dirancang.

Tabel 4.4. Skenario Use Case Menginput Dokumen Video

<b>Nama Use Case</b>	Menginput Dokumen Video	
<b>Deskripsi</b>	Pengguna dapat memasukan input berupa file video untuk dianalisa dalam proses deteksi.	
<b>Aktor</b>	Pengguna	
<b>Skenario</b>	<b>Aksi Aktor</b>	<b>Response sistem</b>
<b>Skenario Normal</b>	<i>Step 1:</i> Pengguna menjalankan aplikasi.	
		<i>Step 2:</i> Aplikasi menampilkan tampilan utama aplikasi.

	<p><i>Step 3:</i> Pegguna memasukan dokumen dengan format video.</p>	
		<p><i>Step 4:</i> Aplikasi akan menampilkan halaman pengkoneksian notifikasi.</p>
<b>Skenario Alternatif</b>	<p><i>Step 1:</i> Pegguna menjalankan aplikasi.</p>	
		<p><i>Step 2:</i> Aplikasi akan menampilkan tampilan utama aplikasi.</p>
	<p><i>Step 3:</i> Pegguna memasukan dokumen yang bukan format video.</p>	
		<p><i>Step 4:</i> Aplikasi akan menampilkan peringatan dan notifikasi <i>popup</i> bahwa pegguna tidak dapat memasukan dokumen selain format video pengkoneksian notifikasi selain seperti ".mp4", ".avi", ".mkv".</p>

Tabel 4.5. Skenario Use Case Pengkoneksian Notifikasi

<b>Nama Use Case</b>	Pengkoneksian Notifikasi	
<b>Deskripsi</b>	Pegguna dapat memasukan OTP pada tautan yang ada pada halaman pengkoneksian notifikasi yang berbentuk <i>barcode</i> .	
<b>Aktor</b>	Pegguna	
<b>Skenario</b>	<b>Aksi Aktor</b>	<b>Response sistem</b>
<b>Skenario Normal</b>	<p><i>Step 1:</i> Pegguna akan memindai <i>barcode</i> yang ada dalam halaman pekoneksian notifikasi dengan <i>smartphone</i> pegguna, pegguna akan dialihkan menuju halaman <i>channel</i> telegram, pegguna mengirimkan OTP yang tertera pada aplikasi</p>	



	kedalam <i>channel</i> telegram tersebut.	
		<i>Step 2:</i> Aplikasi akan mengalihkan pengguna ke halaman konfigurasi batas gerak.
<b>Skenario Alternatif</b>	<i>Step 1:</i> Pengguna tidak melakukan pemindaian pada barcode yang disediakan dan pengguna tidak mengirimkan kode OTP ke tautan yang disediakan sampai waktu tertentu.	
		<i>Step 2:</i> Pengguna akan tetap dihalaman pengkoneksian sampai waktu pengkoneksian habis, dan jika waktu pengkoneksian habis, aplikasi akan memunculkan notifikasi berupa <i>alert pop-up</i> yang menandakan waktu pengkoneksian habis, dan aplikasi akan berhenti.

Tabel 4.6. Skenario Use Case Konfigurasi Batas Gerak

<b>Nama Use Case</b>	Konfigurasi Batas Gerak	
<b>Deskripsi</b>	Pengguna dapat menginput persentase batas gerak berdasarkan tepi yang diinginkan agar menjadi batas area pergerakan	
<b>Aktor</b>	Pengguna	
<b>Skenario</b>	<b>Aksi Aktor</b>	<b>Response sistem</b>
<b>Skenario Normal</b>	<i>Step 1:</i> Pengguna memasukan angka persentase pada kotak <i>input</i> pada halaman konfigurasi, lalu menekan tombol 'Ubah ukuran;	
		<i>Step 2:</i> Aplikasi menampilkan video yang sudah dimasukan dengan batas yang ukurannya telah ditentukan.
	<i>Step 3:</i>	

	Pengguna dapat mengganti ukuran batas sampai batas yang diinginkan. Jika sudah, dapat menekan tombol 'Scan'	
		<i>Step 4:</i> Aplikasi akan mengalihkan pengguna ke halaman Hasil Deteksi.
<b>Skenario Alternatif</b>	<i>Step 1:</i> Pengguna tidak memasukkan angka presentase ukuran batas yang diinginkan pada kotak <i>input</i> dan pengguna menekan tombol 'Scan'	
		<i>Step 2:</i> Aplikasi akan menampilkan peringatan dan notifikasi <i>popup</i> bahwa bahwa pengguna tidak menetapkan batas

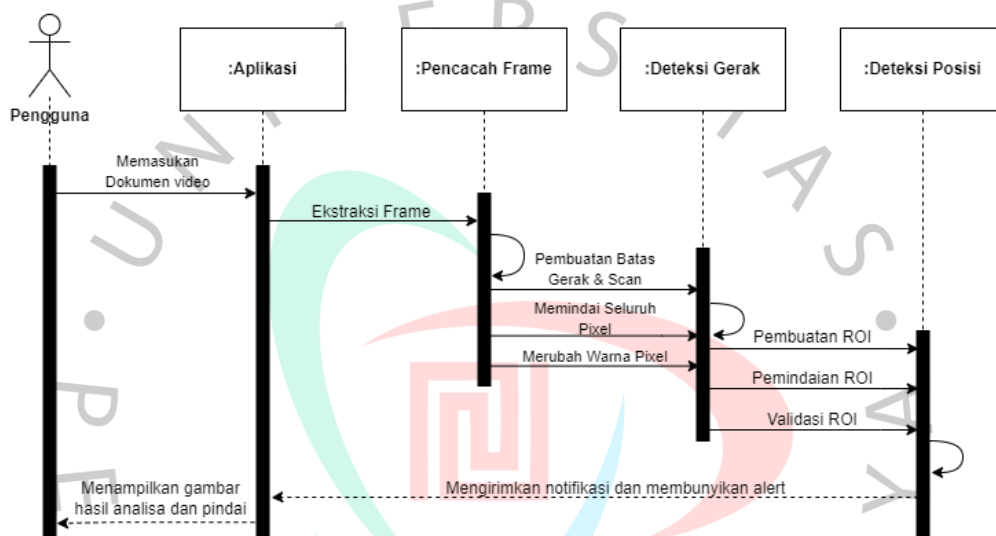
Tabel 4.7. Skenario Use Case Lihat Hasil Deteksi

<b>Nama Use Case</b>	Lihat Hasil Deteksi	
<b>Deskripsi</b>	Pengguna dapat melihat hasil dari analisa dan proses aplikasi	
<b>Aktor</b>	Pengguna	
<b>Skenario</b>	<b>Aksi Aktor</b>	<b>Response sistem</b>
<b>Skenario Normal</b>		<i>Step 1:</i> Aplikasi mendapatkan kriteria aman pada hasil pindai atau objek bayi tidak melewati batas yang telah ditentukan.
		<i>Step 2:</i> Aplikasi akan terus memindai
<b>Skenario Alternatif</b>		<i>Step 1:</i> Aplikasi mendapatkan kriteria bahaya pada hasil pindai atau objek bayi melewati batas yang telah ditentukan.
		<i>Step 2:</i> Aplikasi akan membunyikan <i>alert</i> peringatan dari perangkat

		pengguna dan mengirimkan pesan teks melalui <i>channel</i> telegram
--	--	---

#### 4.3.4 Sequence Diagram

Sequence diagram merupakan representasi visual yang menggambarkan interaksi antara objek-objek yang terlibat dalam sebuah aplikasi. Diagram ini memberikan penjelasan dan tampilan yang jelas tentang bagaimana objek-objek tersebut berinteraksi satu sama lain selama proses berjalannya sistem.

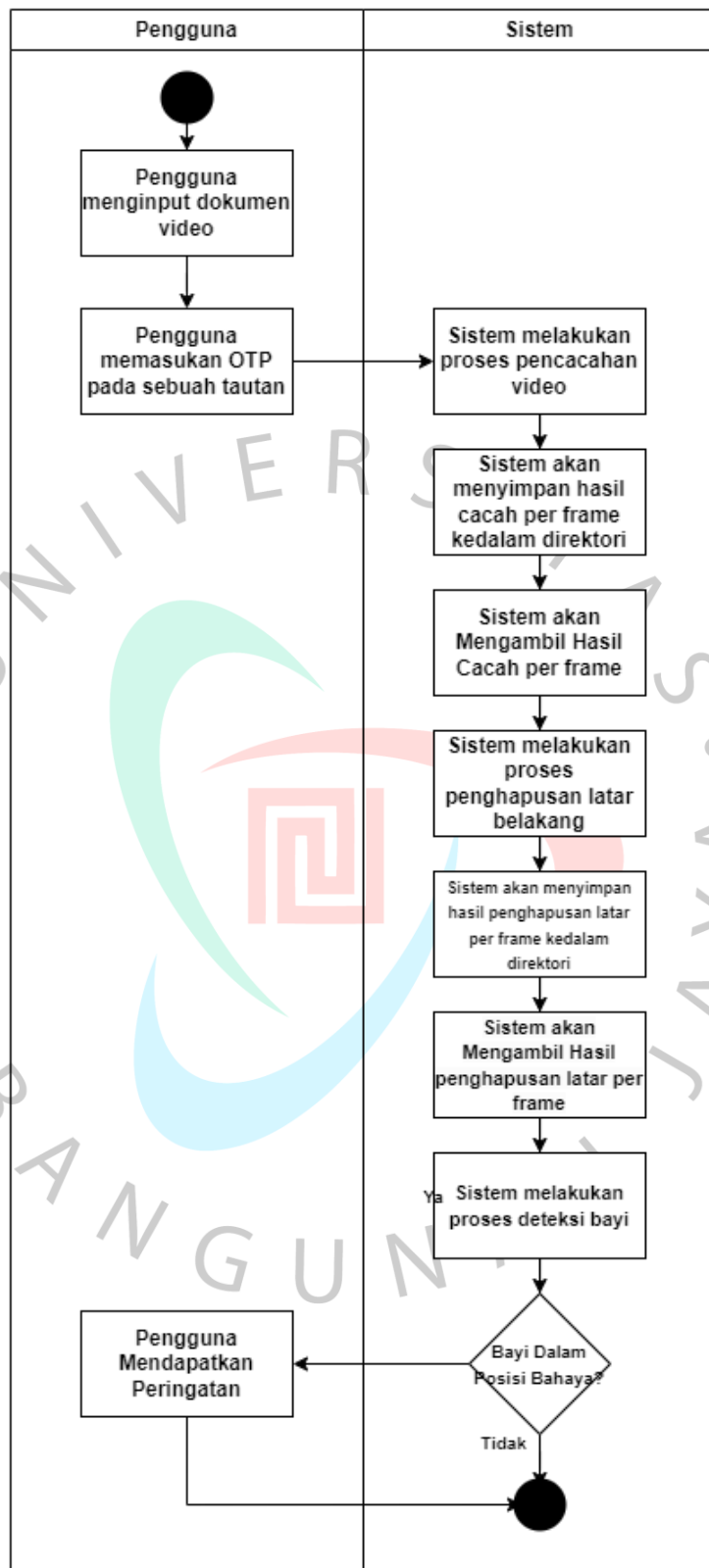


Gambar 4.15 Sequence Diagram Aplikasi

#### 4.3.5 Activity Diagram

*Activity diagram* adalah salah satu jenis diagram yang digunakan dalam pemodelan proses pengguna dan sistem. Diagram ini digunakan untuk merepresentasikan aliran aktivitas atau tindakan dalam suatu proses, sehingga dapat membantu pengguna untuk memahami lebih jelas bagaimana sebuah sistem bekerja atau bagaimana sebuah bisnis menjalankan operasinya.

*Activity diagram* membantu dalam menggambarkan urutan aktivitas atau tindakan yang terjadi dalam suatu proses atau alur kerja. Aktivitas dalam diagram ini mewakili langkah-langkah yang dilakukan oleh pengguna atau sistem dalam mencapai tujuan tertentu. Aliran aktivitas dalam diagram tersebut ditunjukkan oleh panah yang menghubungkan satu aktivitas dengan aktivitas berikutnya, menunjukkan urutan eksekusi atau ketergantungan antar aktivitas.



Gambar 4.16 Activity Diagram Aplikasi

#### 4.3.6 Perancangan Desain Antarmuka

Dalam perancangan antarmuka, dilakukan proses merancang dan mengembangkan antarmuka pengguna yang efektif dan mudah digunakan untuk sistem yang sedang dibangun, dengan tujuan memastikan pengguna dapat berinteraksi dengan sistem secara mudah dan memahami informasi yang disajikan dengan jelas. Pada Gambar 4.17 terdapat contoh perancangan antarmuka yang dapat digunakan oleh pengguna.



Gambar 4.17 Tampilan Awal Aplikasi

Pada tampilan awal aplikasi sebagaimana pada Gambar 4.17, pengguna akan mendapat pesan instruksi untuk memasukan dokumen dengan format video. Apabila tombol diklik, maka pengguna akan diarahkan komputer untuk memilih dokumen yang ingin dianalisa dari direktori tertentu. Lalu aplikasi akan menampilkan halaman pengkoneksian notifikasi.



Gambar 4.18 Tampilan Pengkoneksian Notifikasi

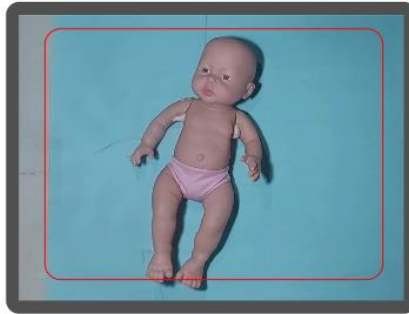
Pada tampilan pengkoneksian notifikasi seperti pada Gambar 4.18, pengguna akan mendapatkan tampilan nomer OTP (*One Time Password*) dan *barcode* yang berisi alamat tautan dari sebuah *channel* di telegram. Pengguna diarahkan untuk mengirimkan OTP ke *channel* tersebut untuk pengkoneksian notifikasi agar *channel* tersebut dapat memberikan notifikasi ke akun pengguna.



Gambar 4.19 Tampilan Pengguna Menentukan Batas Aman

Pada tampilan penetapan batas sebagaimana Gambar 4.19, pengguna dapat mengatur ukuran batas gerak bayi.

## Aplikasi Pendeteksi Bayi

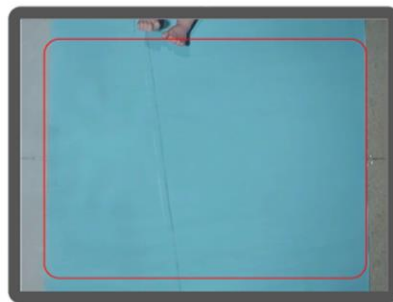


Aman

Gambar 4.20 Tampilan ketika Posisi Bayi dalam area aman

Setelah batas aman telah ditetapkan, maka sistem akan memproses deteksi objek untuk memberikan notifikasi status letak objek. Apabila objek terdeteksi dalam keadaan aman, maka sistem akan menampilkan pesan “aman” berwarna hijau seperti Gambar 4.20.

## Aplikasi Pendeteksi Bayi



Bahaya!

Gambar 4.21 Tampilan ketika Posisi Bayi bahaya

Ketika objek terdeteksi melewati batas aman yang telah ditetapkan, maka sistem akan menampilkan pesan “Bahaya!” dalam warna merah sebagaimana Gambar 4.21.

#### 4.3.7 Perancangan Pengujian Kotak Putih

Pengujian kotak putih merupakan salah satu jenis pengujian perangkat lunak yang dilakukan untuk memeriksa kualitas implementasi teknis dari fitur yang telah dikembangkan dalam sebuah aplikasi. Metode ini memfokuskan pada pengujian struktur internal, arsitektur, dan logika pemrograman yang digunakan dalam implementasi fitur tersebut, selain juga menguji fungsionalitasnya. Tujuannya adalah untuk memastikan bahwa kode program telah dibangun dengan benar, mengikuti standar pemrograman yang baik, dan tidak mengandung bug atau kesalahan yang dapat mempengaruhi kinerja atau keamanan aplikasi secara keseluruhan.

Tabel 4.8. Perancangan Pengujian Kotak Putih

No	Algoritma	Ekspektasi
1	Algoritma pencacah frame dari data video	Dapat mencacah frame per frame dari hasil masukan dokumen video
2	Algoritma pendeteksi gerakan	Dapat mendeteksi perbedaan warna setiap piksel pada perbandingan 2 frame dan menandai piksel tersebut sesuai kriteria
3	Algoritma deteksi posisi bayi berdasarkan daerah yang dibatasi	Dapat mendeteksi dan menandakan setiap ROI yang memiliki nilai kepadatan piksel bergerak sesuai kriteria
4	Algoritma notifikasi ke pengguna	Dapat mengirimkan notifikasi pesan ke akun telegram pengguna melalui suatu akun <i>channel</i> dan audio yang akan berbunyi diperangkat pengguna

#### 4.3.8 Perancangan Pengujian Kotak Hitam

Pengujian kotak hitam merupakan salah satu jenis pengujian perangkat lunak yang dilakukan untuk memeriksa kualitas fitur yang telah dikembangkan pada sebuah aplikasi. Metode ini memfokuskan pada pengujian fungsionalitas fitur aplikasi, tanpa memeriksa atau memperhatikan bagaimana fitur tersebut diimplementasikan secara teknis. Tujuannya adalah untuk memastikan bahwa fitur aplikasi berjalan sesuai dengan spesifikasi yang telah ditetapkan dan memenuhi kebutuhan pengguna. Hal ini dilakukan dengan tujuan untuk memastikan bahwa



fitur tidak memiliki bug atau kesalahan yang dapat mempengaruhi fungsinya atau bahkan merusak sistem secara keseluruhan.

Tabel 4.9. Perencanaan Pengujian Kotak Hitam

No	Halaman	Uji Kasus	Ekspektasi
1	Halaman Utama	Pengguna dapat memilih dokumen video untuk jadi input	Pengguna akan pindah pengkoneksian notifikasi
		Pengguna memilih dokumen selain format video	Pengguna akan mendapatkan notifikasi bahwa format dokumen yang dipilih salah
2	Halaman Pengkoneksian Notifikasi	Pengguna mengirimkan kode OTP yang benar ke tautan yang disediakan	Pengguna akan pindah kehalaman penetapan batas gerak
		Pengguna tidak mengirimkan kode OTP ke tautan yang disediakan	Pengguna akan tetap dihalaman pengkoneksian sampai waktu koneksi habis
3	Halaman Penentuan batas gerak	Pengguna menetapkan batas dengan ukuran yang tidak melewati frame tampilan dari input video	Pengguna akan pindah pemantauan posisi bayi
		Pengguna tidak menetapkan batas dengan ukuran yang melewati frame tampilan dari input video	Pengguna mendapatkan notifikasi bahwa pengguna tidak menetapkan batas
4	Halaman Pemantauan Posisi Bayi	Bayi tidak melewati batas yang telah ditentukan	Tidak terjadi sesuatu
		Bayi melewati batas yang telah ditentukan	Pengguna mendapatkan notifikasi pesan bahwa bayi dalam posisi bahaya dan perangkat pengguna akan berbunyi.