

**IMPLEMENTASI CHATBOT UNTUK MENDIAGNOSA  
PENYAKIT KULIT PADA MANUSIA BERBASIS  
APLIKASI ANDROID**



**SKRIPSI**

Diajukan untuk memenuhi salah satu syarat kelulusan Program Sarjana

**DITO IQBAL KUSUMA**  
**NIM : 17200402**

**Program Studi Teknologi Informasi**

**Fakultas Teknik dan Informatika**

**Universitas Bina Sarana Informatika**

**Jakarta**

**2024**

## SURAT PERNYATAAN KEASLIAN SKRIPSI

Yang bertanda tangan dibawah ini:

Nama : Dito Iqbal Kusuma  
NIM : 17200402  
Jenjang : Sarjana (S1)  
Program Studi : Teknologi Informasi  
Fakultas : Fakultas Teknik dan Informatika  
Perguruan Tinggi : Universitas Bina Sarana Informatika

Dengan ini menyatakan bahwa Skripsi yang telah saya buat dengan judul: **“Implementasi Chatbot Untuk Mendiagnosa Penyakit Kulit Pada Manusia Berbasis Aplikasi Android”**, adalah asli (orisinil) atau tidak plagiat (menjiplak) dan belum pernah diterbitkan/dipublikasikan dimanapun dan dalam bentuk apapun.

Demikianlah surat pernyataan ini saya buat dengan sebenar-benarnya tanpa ada paksaan dari pihak manapun juga. Apabila di kemudian hari ternyata saya memberikan keterangan palsu dan atau ada pihak lain yang mengklaim bahwa Skripsi yang telah saya buat adalah hasil karya milik seseorang atau badan tertentu, saya bersedia diproses baik secara pidana maupun perdata dan kelulusan saya dari **Universitas Bina Sarana Informatika** dicabut/dibatalkan.

Dibuat di : Jakarta  
Pada Tanggal : 5 Juli 2024  
Yang menyatakan,



Dito Iqbal Kusuma

## SURAT PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Yang bertanda tangan dibawah ini, saya:

Nama : Dito Iqbal Kusuma  
NIM : 17200402  
Jenjang : Sarjana (S1)  
Program Studi : Teknologi Informasi  
Fakultas : Fakultas Teknik dan Informatika  
Perguruan Tinggi : Universitas Bina Sarana Informatika

Dengan ini menyatakan bahwa seluruh data, informasi, interpretasi serta pernyataan yang terdapat dalam karya ilmiah Penulis dengan judul **“Implementasi Chatbot Untuk Mendiagnosa Penyakit Kulit Pada Manusia Berbasis Aplikasi Android”** ini, kecuali yang disebutkan sumbernya adalah hasil pengamatan, penelitian, pengelolaan, serta pemikiran saya.

Penulis menyetujui untuk memberikan ijin kepada pihak **Universitas Bina Sarana Informatika** untuk mendokumentasikan karya ilmiah saya tersebut secara internal dan terbatas, serta tidak untuk mengunggah karya ilmiah Penulis repository Universitas Bina Sarana Informatika.

Penulis bersedia untuk bertanggung jawab secara pribadi, tanpa melibatkan pihak **Universitas Bina Sarana Informatika**, atas materi/isi karya ilmiah tersebut, termasuk bertanggung jawab atas dampak atau kerugian yang timbul dalam bentuk akibat tindakan yang berkaitan dengan data, informasi, interpretasi serta pernyataan yang terdapat pada karya ilmiah saya ini.

Demikian pernyataan ini saya buat dengan sebenar-benarnya.

Dibuat di : Jakarta  
Pada Tanggal : 5 Juli 2024  
Yang menyatakan,



**Dito Iqbal Kusuma**

## PERSETUJUAN DAN PENGESAHAN SKRIPSI

Skripsi ini diajukan oleh:

Nama : Dito Iqbal Kusuma  
NIM : 17200402  
Jenjang : Sarjana (S1)  
Program Studi : Teknologi Informasi  
Fakultas : Teknik dan Informatika  
Perguruan Tinggi : Universitas Bina Sarana Informatika  
Judul Skripsi : Implementasi Chatbot Untuk Mendiagnosa Penyakit Kulit Pada Manusia Berbasis Aplikasi Android

Telah dipertahankan pada periode 2024-1 dihadapan penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh Sarjana Komputer (S.Kom) pada Program Sarjana (S1) Program Studi Teknologi Informasi di Universitas Bina Sarana Informatika.

Jakarta, 16 Agustus 2024

### PEMBIMBING SKRIPSI

Pembimbing I : Wati Erawati, M.Kom.



### DEWAN PENGUJI

Penguji I : Ahmad Jurnaidi Wahidin, M.Kom



Penguji II : Ellya Verawati, S.Pd.I., M.Pd.



## PEDOMAN PENGGUNAAN HAK CIPTA

Skripsi sarjana yang berjudul **“Implementasi Chatbot Untuk Mendiagnosa Penyakit Kulit Pada Manusia Berbasis Aplikasi Android”** adalah hasil karya tulis asli Dito Iqbal Kusuma dan bukan hasil terbitan sehingga peredaran karya tulis hanya berlaku di lingkungan akademik saja, serta memiliki hak cipta. Oleh karena itu, dilarang keras untuk menggandakan baik sebagian maupun seluruhnya karya tulis ini, tanpa seizin penulis.

Referensi kepustakaan diperkenankan untuk dicatat tetapi pengutipan atau peringkasan isi tulisan hanya dapat dilakukan dengan seizin penulis dan disertai ketentuan pengutipan secara ilmiah dengan menyebutkan sumbernya.

Untuk keperluan perizinan pada pemilik dapat menghubungi informasi yang tertera dibawah ini:

Nama : Dito Iqbal Kusuma  
Alamat : Jalan Kedondong RT 01 RW 15 No. 36, Kota Depok  
No. Telp : 085878812840  
Email : [ditoiqbalkusuma11@gmail.com](mailto:ditoiqbalkusuma11@gmail.com)

**LEMBAR KONSULTASI SKRIPSI****UNIVERSITAS BINA SARANA INFORMATIKA**

NIM : 17200402  
Nama Lengkap : Dito Iqbal Kusuma  
Dosen Pembimbing : Wati Erawati, M.Kom  
Judul Skripsi : Implementasi Chatbot Untuk Mendiagnosa Penyakit Kulit Pada Manusia Berbasis Aplikasi Android

No	Tanggal Bimbingan	Pokok Bahasan	Paraf Dosen Pembimbing
1	4 April 2024	Bimbingan Perdana	<i>[Signature]</i>
2	17 April 2024	Pengajuan Judul dan Bab I	<i>[Signature]</i>
3	25 April 2024	Acc Bab I dan Pengajuan Bab II	<i>[Signature]</i>
4	16 Mei 2024	Acc Bab II	<i>[Signature]</i>
5	22 Mei 2024	Pengajuan Bab III	<i>[Signature]</i>
6	30 Mei 2024	Pengajuan Bab III (Lanjutan)	<i>[Signature]</i>
7	13 Juni 2024	Pengajuan Bab IV	<i>[Signature]</i>
8	20 Juni 2024	Acc Bab IV dan Pengajuan Bab V	<i>[Signature]</i>
9	27 Juni 2024	Acc Keseluruhan	<i>[Signature]</i>

**Catatan untuk Dosen Pembimbing.****Bimbingan Skripsi**

- Dimulai pada tanggal : 4 April 2024
- Diakhiri pada tanggal : 27 Juni 2024
- Jumlah pertemuan bimbingan : 9 Pertemuan

Disetujui oleh,  
Dosen Pembimbing

(Wati Erawati M.Kom)

## PERSEMBAHAN

Dengan mengucapkan puji syukur kepada Allah SWT., Skripsi ini kupersembahkan untuk:

1. Kedua orang tua tercinta yang selalu memberikan segalanya sampai saat ini, baik dukungan materi, motivasi dan doa yang terbaik bagiku untuk meraih kesuksesanku di dunia maupun akhirat.
2. Kakakku, Danu Ivan Gita Perdana yang telah membimbing dan memotivasi dalam pengerjaan Skripsi ini.
3. Adikku, Muhammad Hafy Azizi yang telah memberikan dukungan yang tiada henti.



## KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Esa, sehingga penulis dapat menyelesaikan skripsi dengan judul **“Implementasi Chatbot Untuk Mendiagnosa Penyakit Kulit Pada Manusia Berbasis Aplikasi Android”** sebagai salah satu syarat untuk menempuh sidang Sarjana guna mencapai gelar kesarjanaan pada Jurusan Teknologi Informasi Di Universitas Bina Sarana Informatika.

Penulisan Skripsi ini tidak lepas dari dukungan, bantuan dan bimbingan berbagai pihak. Dalam kesempatan ini penulis mengucapkan terima kasih dan penghargaan yang sebesar-besarnya kepada:

1. Rektor Universitas Bina Sarana Informatika.
2. Dekan Fakultas Teknik dan Informatika Universitas Bina Sarana Informatika.
3. Bapak Hendra Supendar, M.Kom selaku Ketua Program Studi Teknologi Informasi Universitas Bina Sarana Informatika.
4. Ibu Wati Erawati, M.Kom sebagai Dosen Pembimbing yang telah banyak memberikan bimbingan dan pengarahan serta waktunya dalam menyusun laporan Skripsi.
5. Dosen, Karyawan dan Staff Universitas Bina Sarana Informatika.
6. Orang tua saya tercinta yang selalu memberikan dukungan baik secara moril maupun materil sehingga dapat melaksanakan dan menyusun laporan Skripsi.
7. Teman-teman Teknologi Informasi UBSI terutama angkatan 2020 yang telah memberikan dukungan moril maupun spiritual.

8. Semua pihak yang tidak dapat penulis sebutkan satu persatu tanpa mengurangi rasa terima kasih.

Penulis menyadari banyak kekurangan, maka penulis mengharapkan kritik serta sarannya dari pembaca untuk penyempurnaan karya-karya yang akan datang. Harapan dari penulis, semoga laporan Skripsi ini bisa bermanfaat bagi siapa saja yang mempergunakannya.

Depok, 29 Juni 2024

Penulis



Dito Iqbal Kusuma

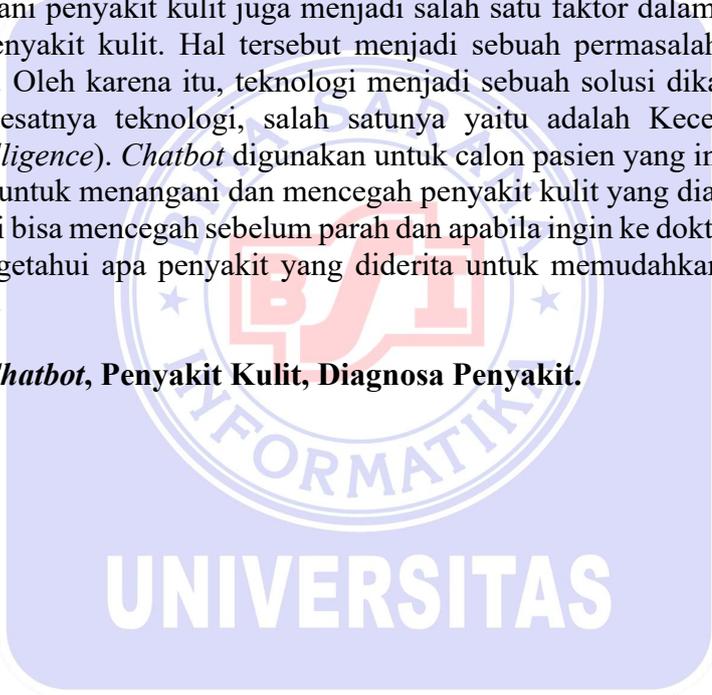


## ABSTRAK

### **Dito Iqbal Kusuma (17200402), Implementasi *Chatbot* Untuk Mendiagnosa Penyakit Kulit Pada Manusia Berbasis Aplikasi *Android*.**

Kulit merupakan organ terbesar pada tubuh manusia yang berada dibagian terluar. Kesehatan kulit yang buruk dapat menyebabkan berbagai macam penyakit kulit. Penyakit kulit rawan terjadi pada daerah tropis termasuk Indonesia. Hal ini dikarenakan keadaan suhu, cuaca, dan kelembaban udara yang berubah-ubah setiap waktu dan tidak menentu. Pendiagnosaan penyakit kulit sangat penting dilakukan secepat mungkin. Sedangkan dalam proses ini dibutuhkannya seorang dokter untuk melakukan diagnosa. Dalam proses ini memerlukan tenaga, waktu, dan uang untuk datang ke rumah sakit. Selain itu, kurangnya dokter spesialis yang memang khusus dalam menangani penyakit kulit juga menjadi salah satu faktor dalam keterlambatan penanganan penyakit kulit. Hal tersebut menjadi sebuah permasalahan besar yang perlu dihadapi. Oleh karena itu, teknologi menjadi sebuah solusi dikarenakan sudah berkembang pesatnya teknologi, salah satunya yaitu adalah Kecerdasan Buatan (*Artificial Intelligence*). *Chatbot* digunakan untuk calon pasien yang ingin melakukan diagnosa awal untuk menangani dan mencegah penyakit kulit yang dialami. Sehingga calon pasien ini bisa mencegah sebelum parah dan apabila ingin ke dokter, calon pasien ini sudah mengetahui apa penyakit yang diderita untuk memudahkan dokter dalam menanganinya.

**Kata kunci:** *Chatbot*, Penyakit Kulit, Diagnosa Penyakit.



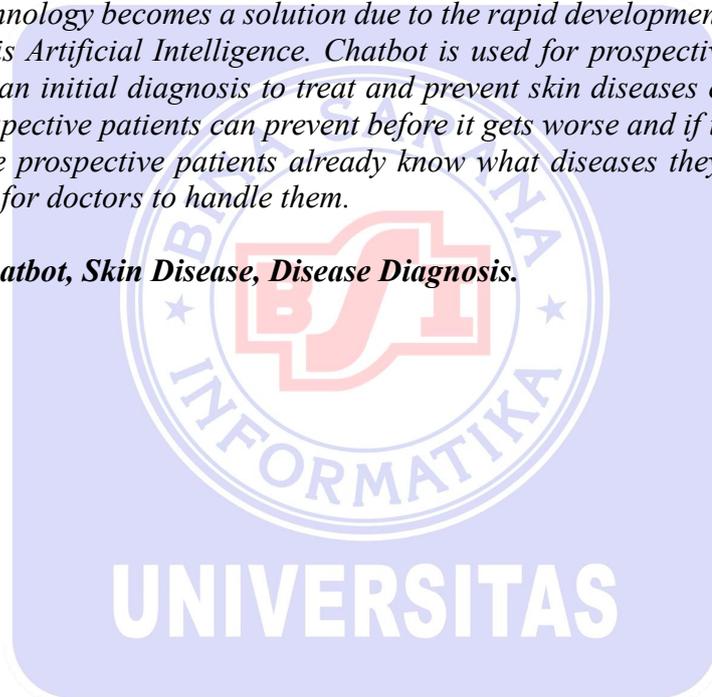
UNIVERSITAS

## ***ABSTRACT***

**Dito Iqbal Kusuma (17200402), *Implementation of Chatbot to Diagnose Skin Diseases in Humans Based on Android Application.***

*Skin is the largest organ in the human body that is located on the outermost part. Poor skin health can lead to various skin diseases. Skin diseases are prone to occur in tropical areas including Indonesia. This is due to the temperature, weather, and humidity conditions that change every time and erratically. It is very important to diagnose skin diseases as quickly as possible. While in this process a doctor is needed to diagnose. This requires energy, time and money to travel to the hospital. In addition, the lack of specialists who are specialized in handling skin diseases is also a factor in the delay in handling skin diseases. This is a big problem that needs to be faced. Therefore, technology becomes a solution due to the rapid development of technology, one of which is Artificial Intelligence. Chatbot is used for prospective patients who want to make an initial diagnosis to treat and prevent skin diseases experienced. So that these prospective patients can prevent before it gets worse and if they want to see a doctor, these prospective patients already know what diseases they suffer from to make it easier for doctors to handle them.*

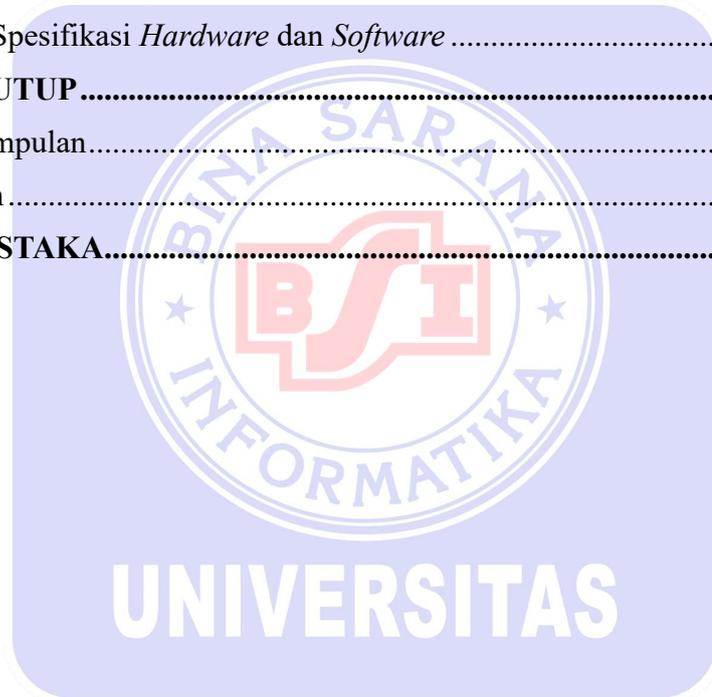
**Keywords: *Chatbot, Skin Disease, Disease Diagnosis.***



# DAFTAR ISI

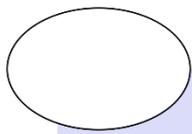
<b>PEDOMAN PENGGUNAAN HAK CIPTA .....</b>	<b>v</b>
<b>PERSEMBAHAN.....</b>	<b>vii</b>
<b>KATA PENGANTAR.....</b>	<b>viii</b>
<b>ABSTRAK .....</b>	<b>x</b>
<b><i>ABSTRACT</i> .....</b>	<b>xi</b>
<b>DAFTAR ISI.....</b>	<b>xii</b>
<b>DAFTAR SIMBOL .....</b>	<b>xiv</b>
<b>DAFTAR GAMBAR.....</b>	<b>xvii</b>
<b>DAFTAR TABEL .....</b>	<b>xviii</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang Masalah.....	1
1.2 Identifikasi Permasalahan.....	3
1.3 Perumusan Masalah.....	4
1.4 Tujuan dan Manfaat.....	4
1.5 Metode Penelitian.....	5
1.5.1 Teknik Pengumpulan Data.....	5
1.5.2 Model Pengembangan Perangkat Lunak.....	6
1.6 Ruang Lingkup.....	8
<b>BAB II LANDASAN TEORI .....</b>	<b>9</b>
2.1 Tinjauan Pustaka.....	9
2.1.1 <i>Artificial Intelligence</i> .....	9
2.1.2 <i>Chatbot</i> .....	10
2.1.3 <i>Natural Language Processing (NLP)</i> .....	12
2.1.4 <i>Unified Modeling Language (UML)</i> .....	14
2.1.5 <i>Android</i> .....	15
2.1.6 <i>Kotlin</i> .....	17
2.1.7 <i>Firebase</i> .....	18
2.2 Penelitian Terkait.....	19
<b>BAB III ANALISIS DAN PERANCANGAN APLIKASI.....</b>	<b>22</b>
3.1 Analisis Aplikasi.....	22
3.1.1 Analisis Masalah .....	22

3.1.2	Analisis Kebutuhan .....	23
3.2	Rancangan Algoritma .....	24
<b>BAB IV</b>	<b>IMPLEMENTASI DAN PENGUJIAN SOFTWARE .....</b>	<b>25</b>
4.1	Desain .....	25
4.1.1	<i>Database</i> .....	25
4.1.2	<i>Software Architecture</i> .....	26
4.1.3	<i>User Interface</i> .....	32
4.2	<i>Code Generation</i> .....	38
4.3	<i>Testing</i> .....	41
4.4	<i>Support</i> .....	43
4.4.1	<i>Publikasi Software</i> .....	43
4.4.2	<i>Spesifikasi Hardware dan Software</i> .....	43
<b>BAB V</b>	<b>PENUTUP .....</b>	<b>45</b>
5.1	Kesimpulan .....	45
5.2	Saran .....	46
<b>DAFTAR PUSTAKA</b>	<b>.....</b>	<b>47</b>

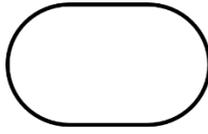


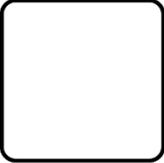
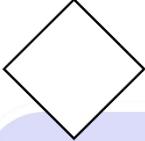
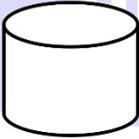
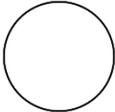
## DAFTAR SIMBOL

### 1. Use Case Diagram

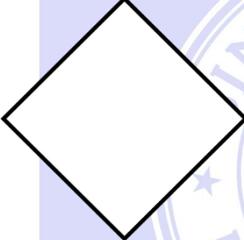
Simbol	Nama	Fungsi
 Pegguna	Aktor	Orang yang berinteraksi dengan sistem
	<i>Use Case</i>	Abstraksi dan interaksi antara sistem dan aktor
	<i>Association</i>	Abstraksi dari penghubung antara aktor dengan use case
	<i>Include</i>	<i>Use case</i> yang ditambahkan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelumnya

### 2. Flowchart

Simbol	Nama	Fungsi
	<i>Terminal</i>	Menunjukkan titik awal atau titik akhir suatu program

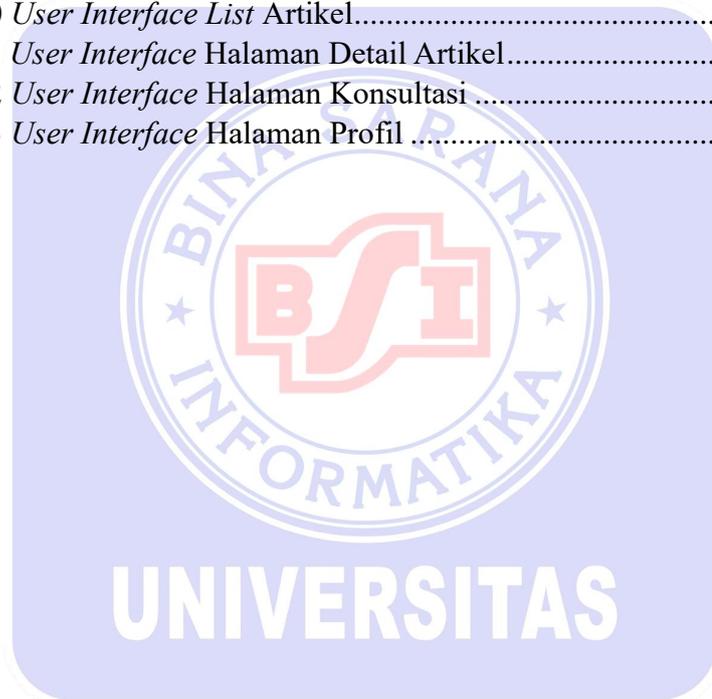
	<p><i>Process</i></p>	<p>Menunjukkan suatu kegiatan proses dari operasi program komputer</p>
	<p><i>Decision</i></p>	<p>Keputusan dalam menunjukkan suatu keputusan yang mempunyai satu atau lebih transisi dengan sebuah kondisi</p>
	<p><i>Input/Output</i></p>	<p>Simbol yang menyatakan proses <i>input</i> atau <i>output</i> tanpa tergantung peralatan.</p>
	<p><i>Database</i></p>	<p>Mewakili data yang disimpan di layanan basis data</p>
	<p><i>Connector</i></p>	<p>Menunjukkan sambungan dari proses ke proses lainnya dalam halaman yang sama</p>
	<p><i>Line Connector</i></p>	<p>Menunjukkan alur dari proses</p>

### 3. Activity Diagram

Simbol	Nama	Fungsi
	<i>Initial State</i>	Menunjukkan awal alur aktivitas
	<i>Final State</i>	Menunjukkan akhir alur aktivitas
	<i>Action</i>	Menunjukkan proses sistem yang sedang melakukan eksekusi aksi
	<i>Line Connector</i>	Menunjukkan alur dari proses
	<i>Decision</i>	Keputusan dalam menunjukkan suatu keputusan yang mempunyai satu atau lebih transisi dengan sebuah kondisi

## DAFTAR GAMBAR

Gambar I. 1 Model Pengembangan Perangkat Lunak <i>Prototype</i> .....	6
Gambar IV. 1 <i>Use Case Diagram</i> .....	26
Gambar IV. 2 <i>Flowchart Login</i> .....	28
Gambar IV. 3 <i>Flowchart Halaman Utama</i> .....	29
Gambar IV. 4 <i>Flowchart Halaman Chatbot</i> .....	29
Gambar IV. 5 <i>Activity Diagram Login</i> .....	30
Gambar IV. 6 <i>Activity Diagram Chatbot</i> .....	31
Gambar IV. 7 <i>Activity Diagram Profil</i> .....	31
Gambar IV. 8 <i>User Interface Halaman Login</i> .....	32
Gambar IV. 9 <i>User Interface Halaman Utama</i> .....	33
Gambar IV. 10 <i>User Interface List Artikel</i> .....	34
Gambar IV. 11 <i>User Interface Halaman Detail Artikel</i> .....	35
Gambar IV. 12 <i>User Interface Halaman Konsultasi</i> .....	36
Gambar IV. 13 <i>User Interface Halaman Profil</i> .....	37



## DAFTAR TABEL

Tabel IV. 1 Tabel <i>Authentication</i> .....	25
Tabel IV. 2 Tabel Artikel .....	25
Tabel IV. 3 Keterangan <i>Use Case Login</i> .....	27
Tabel IV. 4 Keterangan <i>Use Case Diagnosa Menggunakan Chatbot</i> .....	27
Tabel IV. 5 Keterangan <i>Use Case Membaca Artikel Edukasi</i> .....	27
Tabel IV. 6 Keterangan <i>Use Case Profile</i> .....	28
Tabel IV. 7 Pengujian <i>form konsultasi chat</i> .....	42
Tabel IV. 8 Pengujian <i>form konsultasi image &amp; text</i> .....	42
Tabel IV. 9 Pengujian <i>form konsultasi text</i> .....	43
Tabel IV. 10 Spesifikasi <i>Hardware</i> .....	44
Tabel IV. 11 Spesifikasi <i>Android untuk Software</i> .....	44



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang Masalah

Kulit adalah bagian terbesar dari tubuh manusia yang berisi otot-otot dan organ dalam. Kulit memiliki ketebalan, warna, dan tekstur yang berbeda di seluruh tubuh. Struktur kulit manusia memiliki beberapa lapisan, yaitu *epidermis*, *dermis*, dan *hipodermis* (jaringan subkutan). Fungsi utama kulit adalah untuk melindungi tubuh, mengatur suhu tubuh, mengumpulkan informasi sensorik dari lingkungan, serta berpartisipasi dalam sistem kekebalan tubuh untuk melindungi dari penyakit. Kondisi kulit pada umumnya tidak selalu steril, kondisi tidak steril ini dapat mempengaruhi penampilan dan menjadi permasalahan kesehatan yang perlu diperhatikan (Nabillah, 2021). Oleh karena itu, dalam menjaga serta merawat kesehatan kulit menjadi sangat penting. Kesehatan kulit yang buruk dapat menyebabkan berbagai macam penyakit kulit.

“Penyakit kulit merupakan penyakit yang tidak bisa dianggap sepele, karena jika tidak diobati dapat menyebabkan munculnya penyakit lain dan bisa juga menyebabkan kematian jika tidak segera ditangani” (Sitohang, 2023). Tidak menjaga kebersihan dan terpapar zat berbahaya di lingkungan sekitar, alergi dan infeksi bakteri juga dapat menyebabkan penyakit kulit. Penyakit kulit ini rawan terjadi pada daerah tropis termasuk Indonesia. Udara yang lembab dan cuaca panas sepanjang tahun lebih mungkin meningkatkan persentase terkena penyakit kulit seperti yang disebabkan oleh jamur, bakteri, ataupun parasit.

Pendiagnosaan penyakit kulit sangat penting dilakukan secepat mungkin. Sedangkan dalam proses ini dibutuhkannya seorang dokter untuk melakukan diagnosa penyakit yang diderita oleh seorang pasien. Dalam proses ini memerlukan tenaga, waktu, dan uang untuk datang ke rumah sakit. Hal ini sangat kontras dikarenakan kondisi setiap orang itu berbeda-beda. Dimana seseorang harus mengeluarkan biaya yang mungkin tidak sedikit untuk mengobati atau sekedar melakukan pendeteksian, waktu yang tidak semua orang memiliki waktu senggang dalam kesibukannya. Dan juga kurangnya edukasi untuk masyarakat dalam penyakit kulit ini, jadi banyak orang yang menganggap penyakit kulit ini akan sembuh dengan sendirinya jika dibiarkan begitu saja. Selain itu, kurangnya dokter spesialis yang memang khusus dalam menangani penyakit kulit juga menjadi salah satu faktor dalam keterlambatan penanganan penyakit kulit.

Hal tersebut menjadi sebuah permasalahan besar yang perlu dihadapi. Sebagai solusi dari permasalahan, pemanfaatan perkembangan ilmu pengetahuan dan teknologi menjadi sebuah solusi yang memadai dikarenakan sudah berkembang pesatnya teknologi dan muncul teknologi-teknologi yang telah diciptakan oleh para ahli salah satunya yaitu adalah Kecerdasan Buatan (*Artificial Intelligence*). “Kecerdasan Buatan (*Artificial Intelligence*) adalah bagian dari ilmu komputer yang memungkinkan komputer bekerja seperti dan sebaik yang dilakukan oleh manusia” (Jaya dkk., 2019). Perkembangan teknologi kecerdasan buatan ini dapat digunakan dalam berbagai bidang, termasuk salah satunya bidang kesehatan.

Penelitian ini akan menerapkan teknologi *Chatbot* untuk mendeteksi penyakit kulit pada manusia. Dalam dunia komputer *Chat* adalah layanan komunikasi yang menggunakan tulisan sebagai alat berkomunikasi, sedangkan *bot*

merupakan sebuah program yang berisi data yang apabila diberi *input* maka akan menghasilkan *output* sebagai respon. Dengan kata lain, “*chatbot* atau percakapan dengan robot adalah teknologi kecerdasan buatan yang dapat melakukan percakapan antar manusia berdasarkan pengetahuan yang telah disediakan” (Prasetyo dkk., 2021).

Berdasarkan latar belakang diatas yang telah dijabarkan, maka diangkatlah judul skripsi “**IMPLEMENTASI CHATBOT UNTUK MENDIAGNOSA PENYAKIT KULIT PADA MANUSIA BERBASIS APLIKASI ANDROID**”.

## 1.2 Identifikasi Permasalahan

Berdasarkan dari latar belakang masalah yang telah diuraikan diatas, maka dapat disimpulkan beberapa permasalahan sebagai berikut:

1. Terbatasnya dokter spesialis yang khusus menangani penyakit kulit pada manusia.
2. Pasien pergi ke rumah sakit dan harus mengantri untuk berkonsultasi dengan dokter.
3. Keterbatasan waktu dan uang yang dimiliki pasien untuk berkonsultasi dengan dokter secara langsung.
4. Kurangnya pengetahuan masyarakat tentang penyakit kulit serta cara pencegahannya.

### 1.3 Perumusan Masalah

Berdasarkan deskripsi yang telah dijabarkan pada latar belakang masalah diatas, maka diperoleh perumusan masalah dalam penelitian sebagai berikut:

1. Bagaimana mendiagnosa penyakit kulit pada manusia menggunakan teknologi kecerdasan buatan?
2. Bagaimana merancang sistem *chatbot* berbasis aplikasi *Android* untuk melakukan diagnosa terhadap penyakit kulit pada manusia?
3. Bagaimana efektifitas sistem *chatbot* dalam memberikan solusi atas diagnosa yang telah dilakukan oleh sistem terkait penyakit kulit pada manusia?

### 1.4 Tujuan dan Manfaat

Berdasarkan rumusan masalah penelitian diatas, maka tujuan dari penelitian ini yaitu sebagai berikut:

1. Merancang dan membangun aplikasi untuk mendiagnosa penyakit kulit pada manusia yang berbasis aplikasi *Android*.
2. Merancang sistem *chatbot* untuk melakukan konsultasi serta mendiagnosa dari gejala penyakit kulit pasien.
3. Menyediakan fasilitas konsultasi untuk masyarakat yang dapat diakses dengan mudah untuk mendapatkan diagnosa tentang penyakit kulit.

Sedangkan manfaat penulisan skripsi adalah sebagai berikut:

1. Manfaat untuk penulis
  - a. Memenuhi salah satu syarat kelulusan Program Sarjana pada Program Studi Teknologi Informasi Universitas Bina Sarana Informatika.

- b. Mengembangkan keahlian dalam pengembangan aplikasi *mobile* berbasis *Android*.
  - c. Meningkatkan pemahaman tentang penyakit kulit serta cara pencegahan dan penanganannya.
2. Manfaat untuk objek penelitian
- a. Menyediakan fasilitas konsultasi serta edukasi untuk masyarakat tentang penyakit kulit.
  - b. Membantu meningkatkan efisiensi dalam membantu dokter melakukan pendiagnosaan awal.
  - c. Membantu dalam efektifitas dan kemudahan konsultasi melalui *smartphone* tanpa harus mengantri di rumah sakit.
3. Manfaat untuk pembaca
- a. Memberikan pemahaman mengenai pembuatan aplikasi diagnosa menggunakan chatbot yang berbasis aplikasi *Android*.
  - b. Memberikan pengetahuan yang lebih tentang penggunaan teknologi kecerdasan buatan dalam bidang kesehatan terutama pada penyakit kulit.

## 1.5 Metode Penelitian

### 1.5.1 Teknik Pengumpulan Data

Penulis menggunakan beberapa metode berikut dalam mengumpulkan data-data yang digunakan dalam penelitian, yaitu:

#### 1. Observasi

Penulis melakukan observasi secara langsung ke berbagai rumah sakit ataupun klinik yang memiliki dokter spesialis kulit dan melakukan pengamatan dengan kegiatan yang berhubungan dengan permasalahan yang diambil.

## 2. Wawancara

Penulis melakukan wawancara dengan dokter spesialis kulit untuk mendapatkan data-data yang akurat dan relevan sesuai dengan topik permasalahan yang diteliti.

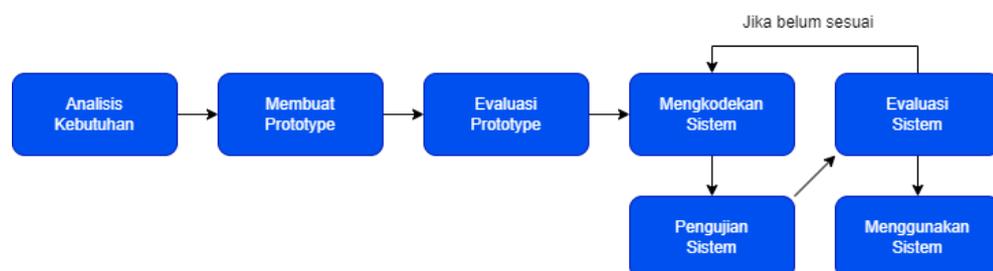
## 3. Studi Pustaka

Selain melakukan observasi dan wawancara, penulis juga melakukan studi pustaka untuk mengumpulkan data-data dan referensi-referensi dari berbagai sumber seperti jurnal, media internet serta buku maupun buku elektronik.

### 1.5.2 Model Pengembangan Perangkat Lunak

Pada penelitian ini, penulis menerapkan strategi *Prototype* sebagai proses pengembangan perangkat lunak dalam perancangan program ini. “Metode *prototype* adalah metode pembuatan aplikasi dengan cara memberikan contoh rancangan kepada calon pengguna aplikasi dan memberikan analisis sebelum dilakukan penulisan sintaks” (Meilinda dkk., 2021).

“Model ini memungkinkan pengguna untuk mendapatkan pratinjau pertama dari perangkat lunak yang akan dikembangkan dan pengguna dapat menguji dari awal pengembangan sebelum perangkat lunak dirilis” (Paksi dkk., 2023).



Sumber: Paksi dkk., (2023)

**Gambar I. 1 Model Pengembangan Perangkat Lunak *Prototype***

Dalam model *prototype* memiliki beberapa tahapan yang biasanya harus dilalui atau dilakukan oleh perancang, yaitu:

#### 1. Analisis Kebutuhan

Model ini dimulai dari tahap analisis kebutuhan, dimana pengembang menganalisa kebutuhan pengguna untuk mendefinisikan semua kebutuhan perangkat lunak yang akan dikembangkan.

#### 2. Membuat *Prototype*

Langkah lain yang dilakukan setelah analisa, yaitu merancang dan membuat *prototype* dari rancangan perangkat lunak yang berfokus pada penerapan perangkat lunak dan metode serta algoritma yang digunakan.

#### 3. Evaluasi *Prototype*

Setelah *prototype* berhasil diselesaikan, maka dilanjutkan dengan mengevaluasi *prototype* dan menentukan apakah *prototype* telah mendefinisikan produk akhir dan memiliki alur kerja yang sesuai.

#### 4. Mengkodekan Sistem

Jika hasil dianggap sudah sesuai dan memuaskan, langkah selanjutnya adalah melakukan pengkodean sistem menggunakan bahasa pemrograman yang sesuai.

#### 5. Pengujian Sistem

Tahap selanjutnya adalah melakukan pengujian terhadap sistem yang telah dikembangkan. Pengujian sistem ini menguji kinerja perangkat lunak berdasarkan tugas-tugas yang dibuat. Pengujian ini sering dilakukan dengan menggunakan metode, seperti *whitebox testing*, *blackbox testing*, ataupun metode pengujian lainnya.

## 6. Evaluasi Sistem

Setelah pengujian dilakukan dan terdapat beberapa fungsi yang tidak berjalan sesuai rencana maka langkah selanjutnya akan melakukan pengkodean ulang untuk memperbaiki fungsi-fungsi sistem sesuai dengan rencana.

## 7. Menggunakan Sistem

Tahap ini adalah tahapan terakhir pada model *prototype*. Pada tahap ini perangkat lunak sudah dilakukan uji coba dan tidak terdapat kesalahan pada sistem maka perangkat lunak siap digunakan oleh pengguna.

### 1.6 Ruang Lingkup

Pada penelitian ini ruang lingkupnya hanya berfokus pada hal-hal yang berkaitan dengan penyakit kulit manusia, seperti gejala-gejala yang dialami, cara penanganan, cara pencegahan, obat untuk mengobati penyakit kulit yang menggunakan fungsi dan fitur *chatbot*.



## BAB II

### LANDASAN TEORI

#### 2.1 Tinjauan Pustaka

Penulis menggunakan teori-teori berikut sebagai acuan saat penulisan skripsi ini, di antaranya:

##### 2.1.1 *Artificial Intelligence*

*Artificial Intelligence* atau kecerdasan buatan adalah teknologi yang dirancang untuk membuat sistem komputer untuk belajar, mengenali, mengambil keputusan, dan melakukan tugas-tugas yang kompleks, yang dilatih untuk meniru kemampuan kecerdasan manusia.

Kecerdasan buatan (*Artificial Intelligence*) merupakan cabang dari penelitian, aplikasi dan instruksi yang berkaitan dengan pemrograman komputer untuk melakukan hal-hal yang menurut manusia adalah cerdas (Efrian & Latifa, 2022).

Jenis-jenis kecerdasan buatan (*Artificial Intelligence*) berdasarkan kemampuan dan fungsinya:

1. AI Lemah (*Weak AI*) yaitu kecerdasan buatan yang memiliki kemampuan terbatas untuk melakukan tugas tertentu.
2. AI Kuat (*Strong AI*) yaitu kecerdasan buatan yang memiliki kemampuan hampir atau setara dengan kecerdasan manusia.
3. AI Sempit (*Narrow AI*) yaitu kecerdasan buatan yang dirancang untuk menyelesaikan tugas-tugas yang lebih spesifik.

4. AI Luas (*Broad AI*) yaitu kecerdasan buatan yang dirancang untuk melakukan berbagai tugas dan belajar dari pengalaman-pengalaman.

Kecerdasan buatan sudah sangat berkembang sekali pada saat ini, banyak dampak besar yang telah diciptakan dari perkembangan teknologi kecerdasan buatan ini. Dampak-dampaknya mempengaruhi dalam berbagai aspek kehidupan, seperti di bidang kesehatan, transportasi dan masih banyak lagi.

Kecerdasan buatan memiliki fungsi yang sangat beragam, karena sering dilatih dan diciptakan untuk menyerupai kemampuan manusia dalam menyelesaikan tugas. Kecerdasan buatan dapat berfungsi dan diterapkan kedalam beberapa jenis teknologi seperti berikut :

1. Otomasi
2. *Machine Learning*
3. *Chatbot*
4. *Natural Language Processing*
5. *Machine Vision*
6. Robotika

### 2.1.2 *Chatbot*

*Chatbot* adalah sistem yang dirancang untuk memungkinkan orang berkomunikasi dengan tulisan, ucapan atau gambar. *Chatbot* dibuat menggunakan perintah tertentu untuk merespons *input* atau pertanyaan pengguna. *Chatbot* dibuat berdasarkan topik yang dilatih dalam *database*. Banyak *chatbot* dibangun berdasarkan hal yang ingin diselesaikan dalam kebutuhan pribadi atau profesional. Sebagian besar *Chatbot* sering disebut-sebut sebagai salah satu dari representasi interaksi manusia dengan mesin yang paling canggih dan menjanjikan (Haryanto & Saefurrahman, 2024).

Model *Artificial Intelligence* yang dikembangkan oleh berbagai perusahaan besar dan salah satunya adalah *Google*. *Google* telah mengembangkan model kecerdasan buatan yang bernama Gemini. Gemini merupakan sistem kecerdasan buatan yang inovatif. Gemini AI, yang diciptakan oleh *Google Deepmind* memanfaatkan kekuatan teknologi GenAI multi-model. Pendekatan ini melibatkan pelatihan AI tidak hanya pada data teks, tetapi juga pada berbagai jenis data lainnya, seperti gambar, audio, dan lainnya, sehingga memungkinkannya untuk unggul dalam tugas-tugas yang membutuhkan pemahaman dan integrasi lintas modal (Perera & Lankathilake, 2023). Model kecerdasan buatan Gemini ini disebut sebagai model “Multimodal” yang memungkinkan untuk melakukan tugas-tugas yang melebihi kemampuan model bahasa tradisional.

*Google* Gemini memiliki tiga versi yang berbeda diantaranya adalah :

1. Gemini Ultra

Gemini Ultra adalah model terbesar dan paling canggih yang dibuat untuk tugas-tugas yang sangat kompleks seperti penelitian ilmiah dan pengembangan Large Language Model (LLM).

2. Gemini Pro

Gemini Pro adalah model yang lebih rendah dari Gemini Ultra dan paling canggih untuk tugas-tugas yang rumit dan memastikan respons dan penanganan *query* yang cepat.

### 3. Gemini Nano

Gemini Nano adalah model Gemini versi terkecil dan paling ringan. Model ini memungkinkan kinerja *offline* seperti chat dan ringkasan teks pada perangkat seluler.

#### 2.1.3 *Natural Language Processing* (NLP)

“*Natural Language Processing* (NLP) bidang kecerdasan buatan yang mempelajari interaksi manusia dan mesin menggunakan bahasa alami. Jenis komputasi ini berguna untuk memfasilitasi komunikasi antara manusia dan mesin yang mencari informasi agar keduanya dapat berkomunikasi” (Furqan dkk., 2023). Beberapa metode *preprocessing* yang digunakan dalam pemrosesan bahasa alami adalah sebagai berikut:

##### 1. *Case Folding*

“*Case Folding* merupakan langkah dalam proses mengubah kata menjadi satu bentuk yang sama. Tujuan dari metode *Case Folding* adalah untuk mengembalikan setiap kata dalam bentuk yang lebih kecil, sehingga data teks terstruktur dalam bentuk yang sama” (Albab dkk., 2023).

##### 2. Tokenisasi

“Tokenisasi adalah proses mengidentifikasi karakter tertentu yang mungkin atau mungkin tidak dianggap sebagai jeda kata. Metode tokenisasi membagi urutan karakter dalam teks menjadi kata-kata terpisah” (Albab dkk., 2023).

### 3. *Part-of-speech Tagging*

“*Part-of-speech Tagging* adalah penugasan tag POS ke frasa atau kata. Untuk menganalisis teks, setiap kata dapat diberikan informasi tentang bentuknya, seperti kata kerja, kata benda, atau kata sifat” (Furqan dkk., 2023).

### 4. *Syntatic Parsing*

“*Syntatic Parsing* adalah proses menganalisis sintaksis kalimat berdasarkan beberapa konsep tata bahasa tertentu. Proses screening secara umum dapat dibagi menjadi dua bagian, yaitu kelompok ketergantungan dan kelompok daya tarik” (Furqan dkk., 2023).

### 5. *Stemming*

“*Stemming* adalah proses pemetaan dan analisis berbagai jenis kata dalam bentuk dasarnya. Metode segmentasi dan proyeksi digunakan untuk menemukan kata dasar dari sebuah kata yang mengalami imbuhan dengan menghilangkan atau menghapus imbuhan tersebut” (Albab dkk., 2023).

*Chatbot* meniru percakapan manusia melalui internet, memungkinkan interaksi kapanpun dan dimanapun. Analisis entitas, bagian penting dalam NLP, membantu *chatbot* memahami maksud dan inti dari percakapan dengan pengguna.

### 2.1.4 Unified Modeling Language (UML)

*Unified Modeling Language* adalah sebuah bahasa yang sudah menjadi standar untuk membuat rancangan perangkat lunak yang komprehensif. Layaknya cetak biru (*blueprint*) yang digunakan arsitek untuk membuat sebuah bangunan, seorang *programmer* juga membutuhkan cetak biru (*blueprint*) ini yaitu UML untuk membuat dan merancang rancangan sistem suatu program.

*Unified Modeling Language* (UML) juga digunakan untuk mengembangkan sistem rekayasa perangkat lunak, yang merupakan bahasa visual untuk mendefinisikan dan mendokumentasikan sistem. Persyaratan dalam skenario yang mengekspresikan bagaimana pengguna menggunakan sebuah sistem ditunjukkan dengan UML. Batasan-batasan dari sebuah sistem juga ditunjukkan dengan UML (Barjakly dkk., 2021). Terdapat sejumlah jenis-jenis diagram yang ada pada UML, yaitu:

#### 1. *Use Case Diagram*

*Use case diagram* merupakan diagram yang bekerja dengan cara mendeskripsikan tipikal interaksi antara pengguna dengan sebuah sistem itu sendiri. Diagram ini terdiri dari sebuah aktor dengan interaksi yang dilakukannya. Aktor tersebut dapat berupa manusia, perangkat keras, sistem, ataupun yang berinteraksi dengan sistem.

#### 2. *Activity Diagram*

“*Activity diagram* adalah diagram yang menggambarkan alur kerja atau aktivitas dari sebuah sistem yang ada pada perangkat lunak. Diagram aktivitas

menggambarkan aktivitas sistem bukan apa yang dilakukan oleh aktor” (Musthofa & Adiguna, 2022).

### 2.1.5 *Android*

Sistem operasi *mobile* yang dijalankan berdasarkan *kernel Linux* yang dibangun untuk perangkat seluler seperti ponsel pintar. Keterbukaan *Android* sebagai sebuah *platform* memberikan kebebasan kepada para pengembang untuk berinovasi dan meningkatkan *platform* ini untuk membangun aplikasi yang kompatibel dengan berbagai perangkat.

Menurut Hermawan mendefinisikan “*Android* adalah sistem operasi *mobile* yang muncul sebagai solusi inovatif ditengah keterbatasan sistem operasi lain, seperti halnya dengan *Windows Mobile*, *iOS*, *Symbian*, dan masih banyak lagi yang berfokus membangun aplikasi bawaan sendiri tanpa melihat potensi yang cukup besar dari aplikasi pihak ketiga. Oleh karena itu, *Android* memungkinkan aplikasi pihak ketiga untuk mengakses data, berkomunikasi, serta kebebasan dalam pendistribusian aplikasi pihak ketiga di *platform Android*.” (Alexandra, 2022).

#### 1. Sejarah Perkembangan *Android*

Sejak awal tahun 2003, *Android* dikembangkan oleh perusahaan bernama *Android Inc.* Awalnya dimaksudkan untuk menjadi sistem operasi untuk kamera digital. Namun, pada tahun 2004, rencananya mengalami perubahan yaitu *Android* sebagai sistem operasi untuk ponsel pintar dan *Google* membeli *Android Inc.* pada tahun 2005.

Dari awal dimulai hingga kini *Android* sudah berkembang dan sudah banyak mengeluarkan sistem operasi dengan versi terbaru. Versi-versi pada *Android* ini memiliki penamaan menggunakan nama makanan manis, seperti *Cupcake*, *Oreo*, *Kitkat* dan sebagainya. Hingga saat ini, versi sistem operasi *Android* yang telah rilis adalah sebagai berikut:

- a. *Android 1.0 dan 1.1 (Astro-Alpha dan Bender-Beta)*
  - b. *Android 1.5 (Cupcake)*
  - c. *Android 1.6 (Donut)*
  - d. *Android 2.0 dan 2.1 (Éclair)*
  - e. *Android 2.2 (Froyo)*
  - f. *Android 2.3 (Gingerbread)*
  - g. *Android 3.0 (Honeycomb)*
  - h. *Android 4.0 (Ice Cream Sandwich)*
  - i. *Android 4.1 – 4.3 (Jellybean)*
  - j. *Android 4.4 (Kitkat)*
  - k. *Android 6.0 (Marshmallow)*
  - l. *Android 8.0 dan 8.1 (Oreo)*
  - m. *Android 9.0 (Pie)*
  - n. *Android 10*
  - o. *Android 11*
  - p. *Android 12*
  - q. *Android 13*
2. *Android Software Development Kit (SDK)*

Pengembang yang ingin mengembangkan aplikasi berbasis *Android* dapat menggunakan *Android SDK* yang memiliki banyak alat untuk pengembangan yang luas dan terdiri dari *debugger*, *libraries*, *handset emulator*, dokumentasi, kode sampel dan tutorial.

Salah satu SDK *Android* yang paling umum untuk mengembangkan aplikasi *Android* yaitu *Java SE Development Kit*. Disamping itu ada berbagai

bahasa yang ditetapkan *Google* pada tahun 2017 lalu salah satunya itu adalah *Kotlin*.

### 2.1.6 *Kotlin*

*Kotlin* adalah bahasa pemrograman tingkat tinggi yang memiliki kemampuan *multiplatform*, diketik secara statis, bertujuan umum, dan memiliki paralelisme. *Kotlin* dirancang untuk dapat beroperasi dengan *Java*, dan versi *Java Virtual Machine* dari library standar *Kotlin* bergantung pada *Java Class Library*, tetapi versi *inline* memungkinkan sintaks menjadi lebih ringkas. *Kotlin* merupakan bahasa pemrograman *open source*, yang dikembangkan *JetBrains* untuk berbagai *platform*, akan tetapi popularitasnya bahasa pemrograman ini semakin meningkat dibuat untuk pengembangan aplikasi *mobile* berbasis *Android*.

#### 1. Sejarah Perkembangan *Kotlin*

*Kotlin* pertama kali dikembangkan pada tahun 2011 oleh *Jetbrains*. Pada awalnya, pimpinan *Jetbrains* yaitu *Dimitry Jemerov* menyadari bahwa banyak bahasa pemrograman, seperti *Java* memiliki aturan pengkodean yang kompleks. Karena *Java* memiliki banyak keunggulan terutama di lingkungannya yang besar, ide untuk membuat bahasa pemrograman yaitu *Kotlin* muncul yang mana memiliki keunggulan yang lebih ringkas, tetapi bisa menggunakan seluruh lingkungan *Java*.

*Kotlin* menjadi *open source* dibawah lisensi *Apache 2* pada tahun 2012 oleh *Jetbrains*. Yang berarti penggunaan *Kotlin* secara bebas. *Kotlin* telah berkembang pesat dari waktu ke waktu, terutama setelah mendapatkan dukungan kelas pertama (*First-class Support*) oleh *Google* pada tahun 2017. Bahkan pada tahun 2019, *Google* memprioritaskan *Kotlin* sebagai bahasa pemrograman utama untuk *Android* dibanding bahasa pemrograman lainnya.

## 2. Kelebihan *Kotlin*

Tiap bahasa pemrograman pasti memiliki kelebihan dan kekurangannya masing-masing seperti *Java* yang memiliki kekurangan yaitu aturan pengkodean yang rumit, begitu pula dengan *Kotlin*. Berikut ini kelebihan dari bahasa pemrograman *Kotlin*:

- a. Ekspresif dan ringkas.
- b. Kode yang lebih aman.
- c. Bisa digunakan bersama dengan *Java*.
- d. Serentak dan terstruktur.

## 3. Kekurangan *Kotlin*

Semua bahasa pemrograman pasti memiliki sebuah kekurangan dan begitu pula dengan *Kotlin*. Berikut ini kekurangan dari *Kotlin*:

- a. Kecepatan *compile* yang tidak stabil.
- b. Minim komunitas dan pengembang.

### 2.1.7 *Firebase*

*Firebase* bersifat *Backend as a Service* (BaaS) yang merupakan layanan *cloud* dari *Google* yang menyediakan semua infrastruktur dan alat yang dibutuhkan untuk membantu pengembang aplikasi dalam mengembangkan aplikasinya.

*Firebase* didirikan pertama kali pada tahun 2011. Produk yang pertama kali ditawarkan adalah *Realtime Database*. *Realtime Database* digunakan *developer*

untuk menyimpan data dan mensinkronisasi ke banyak pengguna. Pada bulan Oktober 2014, *Firebase* diakuisisi oleh *Google*.

*Firebase* memiliki fitur yang sangat banyak dan beragam. Berikut ini jenis-jenis fitur *Firebase* yang disediakan:

1. *Firebase Analytics*
2. *Firebase Cloud Messaging and Notifications* (FCM)
3. *Firebase Authentication*
4. *Firebase Cloud Firestore*
5. *Firebase Realtime Database*
6. *Firebase Hosting*

## 2.2 Penelitian Terkait

Penulis mendapatkan inspirasi dari berbagai penelitian yang telah dilakukan sebelumnya dan berkaitan dengan situasi permasalahan yang dibahas dalam skripsi ini. Penelitian terdahulu yang berkaitan dengan skripsi ini, meliputi:

1. Penelitian yang dilakukan oleh (Haryanto & Saefurrahman, 2024) yang berjudul “Implementasi *Chatbot* Kesehatan Kucing Melalui *Dialogflow* dan *Telegram* untuk Pemberian Informasi Penyakit dan Perawatan”. Pada penelitian ini bertujuan untuk mengetahui sistematika pembuatan dan penggunaan *chatbot* untuk mendiagnosa penyakit serta perawatan.
2. Penelitian yang dilakukan oleh (Furqan dkk., 2023) yang berjudul “*Chatbot Telegram* menggunakan *Natural Language Processing*”. Pada penelitian ini mengimplementasikan metode algoritma *Natural Language Processing* pada *Chatbot*.

3. Penelitian yang dilakukan oleh (Prasetyo dkk., 2021) yang berjudul “Implementasi *Natural Language Processing* dalam Pembuatan *Chatbot* pada Program *Information Technology* Universitas Surabaya”. Di penelitian ini mengimplementasikan metode *Natural Language Processing* dan cara kerja algoritma tersebut didalam *chatbot*.
4. Penelitian yang dilakukan oleh (Efrian & Latifa, 2022) yang berjudul “*Image Recognition* Berbasis *Convolutional Neural Network* (CNN) Untuk Mendeteksi Penyakit Kulit Pada Manusia”. Di penelitian ini menerapkan kemampuan *image recognition* yaitu teknologi yang mampu mendefinisikan gambar yang diunggah pengguna untuk didiagnosa dan diteliti menggunakan kecerdasan buatan dan mengirim respon atas diagnose yang telah dilakukan oleh kecerdasan buatan tersebut.
5. Penelitian yang dilakukan oleh (Sitohang, 2023) yang berjudul “Sistem Pakar Diagnosa Penyakit Kulit Pada Manusia Menggunakan Metode *Hybrid*”. Pada penelitian ini menjelaskan apa saja gejala-gejala penyakit kulit dan dari gejala tersebut dipergunakan untuk sistem pakar yang menggunakan metode *hybrid* yaitu *Forward Chaining* dan *Certainty Factor* untuk melakukan Analisa penyakit.

## BAB III

### ANALISIS DAN PERANCANGAN APLIKASI

#### 3.1 Analisis Aplikasi

Proses identifikasi dan evaluasi kebutuhan aplikasi yang akan diimplementasikan itu merupakan sebuah tahap dalam menganalisa aplikasi. Setelah dilakukan identifikasi, analisa, serta evaluasi, langkah selanjutnya dapat melakukan perencanaan serta perancangan aplikasi.

##### 3.1.1 Analisis Masalah

Melalui pengamatan mendalam di beberapa tempat, penelitian ini mengidentifikasi sebuah masalah. Berikut adalah uraian dan gambaran masalah penelitian dalam penelitian ini:

1. Belum adanya sebuah aplikasi berbasis *mobile* yang menyediakan informasi dan melakukan diagnosa menggunakan *chatbot* terhadap penyakit kulit pada manusia.
2. Kurangnya jumlah dokter spesialis di Indonesia, terutama dokter spesialis kulit. Berdasarkan data dari Konsil Kedokteran Indonesia, Dokter dengan bergelar Sp.DV yang merupakan dokter spesialis kulit hanya berjumlah 459 di Provinsi DKI Jakarta.
3. Mahalnya harga untuk membuat perjanjian diagnosa penyakit kulit dengan dokter spesialis.

### 3.1.2 Analisis Kebutuhan

#### 1. Analisis Kebutuhan Fungsional

Fitur-fitur fungsional dirancang untuk memenuhi kebutuhan pengguna dari masalah yang sudah diuraikan. Terdapat fitur-fitur fungsional yang akan dirancang untuk sistem yang akan dibangun, yaitu:

- a. *Chatbot* untuk melakukan diagnosa penyakit kulit yang diderita pengguna.
- b. *Image Recognition* untuk mendiagnosa melalui gambar yang di input oleh pengguna.
- c. Artikel-artikel berisi informasi tentang kulit sebagai media edukasi untuk pengguna.

#### 2. Analisis Kebutuhan Non-fungsional

Dalam proses perancangan dan pengembangan sistem, diperlukan berbagai alat bantu non-fungsional. Berikut adalah uraian persyaratan kebutuhan non-fungsional untuk proses pengembangan sistem yang akan dibuat:

##### a. Spesifikasi Perangkat lunak (*Software*)

Adapun untuk spesifikasi perangkat lunak yang digunakan untuk menjalankan program aplikasi ini adalah sebagai berikut:

- 1) Sistem operasi : Windows 11 Home 64-bit
- 2) Bahasa Pemrograman : *Kotlin*
- 3) Program : *Android Studio, Gemini API, Firebase, Phone emulator* (opsional)

##### b. Spesifikasi Perangkat Keras (*Hardware*)

Adapun spesifikasi minimal yang dibutuhkan untuk merancang program aplikasi ini adalah sebagai berikut:

- 1) *Processor* : AMD Ryzen 5 atau Intel I5

- 2) *Memory* : 16 GB
- 3) *Harddisk* : 250 GB
- 4) *VGA* : 2 GB

### 3.2 Rancangan Algoritma

“*Deep Learning* adalah metode dalam kecerdasan buatan (*artificial intelligence*) yang mengajarkan komputer untuk memproses data dengan cara yang terinspirasi oleh otak manusia.” (Ananto dkk., 2023). Metode ini dapat mendeteksi pola yang kompleks dalam teks, gambar, audio, dan data lainnya untuk menghasilkan informasi yang akurat.

Dalam *deep learning* terdapat dua algoritma yang biasanya sering digunakan, yaitu:

1. “*Convolutional Neural Network (CNN)* adalah algoritma yang sering digunakan untuk melatih data besar dengan jutaan parameter” (Ananto dkk., 2023).
2. *Recurrent Neural Networks (RNN)* suatu jenis jaringan umpan balik pada *neural network* dimana *loop* berfungsi sebagai koneksi umpan balik dalam jaringan. Jaringan RNN adalah jaringan yang memungkinkan *output* jaringan diterima sebagai *input* yang kemudian digunakan untuk menghasilkan *output* baru dalam jaringan tersebut. (Tarkus dkk., 2020).

## BAB IV

### IMPLEMENTASI DAN PENGUJIAN SOFTWARE

#### 4.1 Desain

##### 4.1.1 Database

Aplikasi ini menggunakan *Firebase* sebagai media penyimpanan basis data. Berikut adalah desain rancangan basis data dari Aplikasi Diagnosa Penyakit Kulit Menggunakan *Chatbot*:

##### 1. Tabel *Authentication*

**Tabel IV. 1 Tabel *Authentication***

Authentication Field
Identifier
Providers
Created
Signed In
User UID

Sumber: Hasil Penelitian

##### 2. Tabel Artikel

**Tabel IV. 2 Tabel Artikel**

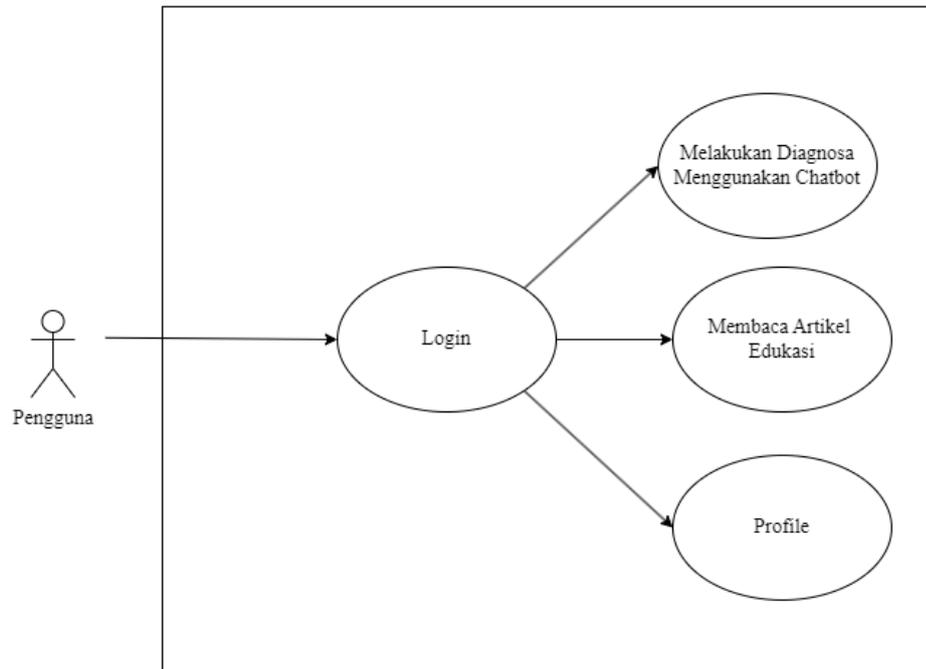
Artikel	
Field	Tipe Data
Id	<i>Integer</i>
imageUrl	<i>String</i>
title	<i>String</i>
date	<i>String</i>
content	<i>String</i>

Sumber: Hasil Penelitian

### 4.1.2 Software Architecture

Berikut adalah rancangan arsitektur perangkat lunak dari Aplikasi Diagnosa Penyakit Kulit Menggunakan *Chatbot*:

#### 1. Use Case Diagram



Sumber: Hasil Penelitian

**Gambar IV. 1 Use Case Diagram**

Pada gambar 4.1 adalah *use case* diagram dari aplikasi diagnosa penyakit kulit menggunakan *chatbot*. Setiap *use case* menggambarkan operasi yang berbeda, penjelasannya seperti pada tabel dibawah ini:

a. *Use Case Login***Tabel IV. 3 Keterangan *Use Case Login***

Nama	<i>Login</i>
Aktor	Pengguna
Deskripsi	Melakukan proses <i>login</i> ke dalam aplikasi
Tujuan	Memverifikasi akun dan mengalihkan ke halaman utama

Sumber: Hasil Penelitian

b. *Use Case Diagnosa Menggunakan Chatbot***Tabel IV. 4 Keterangan *Use Case Diagnosa Menggunakan Chatbot***

Nama	<i>Register</i>
Aktor	Pengguna
Deskripsi	Melakukan proses daftar akun
Tujuan	Membuat akun dan menyimpan data ke <i>database</i>

Sumber: Hasil Penelitian

c. *Use Case Membaca Artikel Edukasi***Tabel IV. 5 Keterangan *Use Case Membaca Artikel Edukasi***

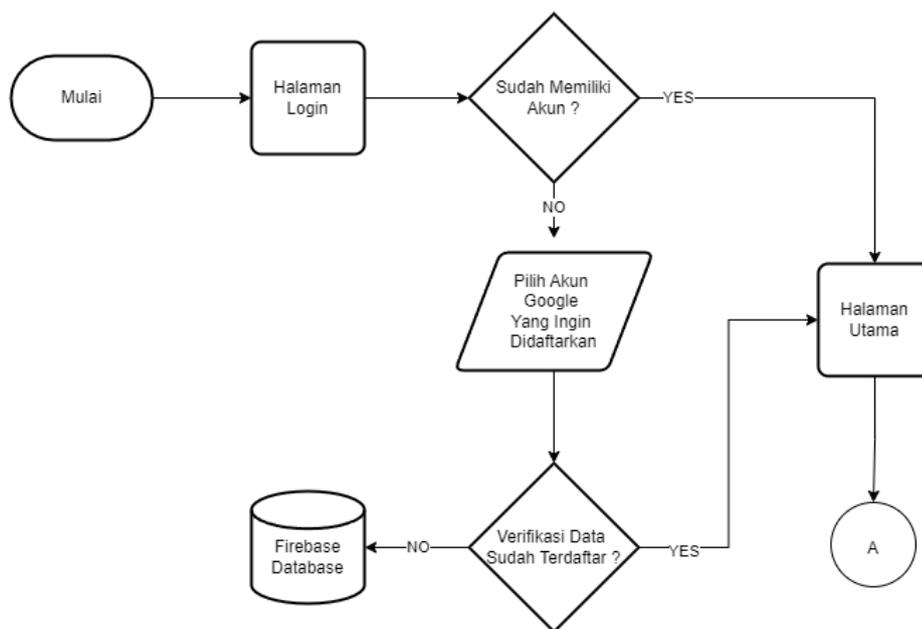
Nama	Membaca Artikel Edukasi
Aktor	Pengguna
Deskripsi	Membaca artikel edukasi
Tujuan	Menampilkan daftar artikel

Sumber: Hasil Penelitian

d. *Use Case Profile***Tabel IV. 6** Keterangan Use Case *Profile*

Nama	<i>Profile</i>
Aktor	Pengguna
Deskripsi	Melihat detail informasi pengguna
Tujuan	Menampilkan informasi pengguna

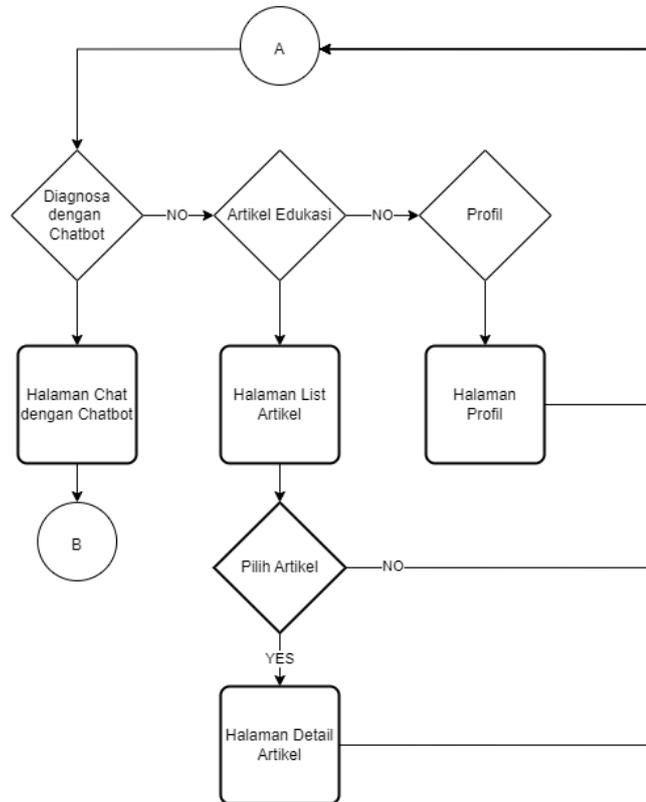
Sumber: Hasil Penelitian

2. *Flowchart*a. *Flowchart Login*

Sumber: Hasil Penelitian

**Gambar IV. 2** *Flowchart Login*

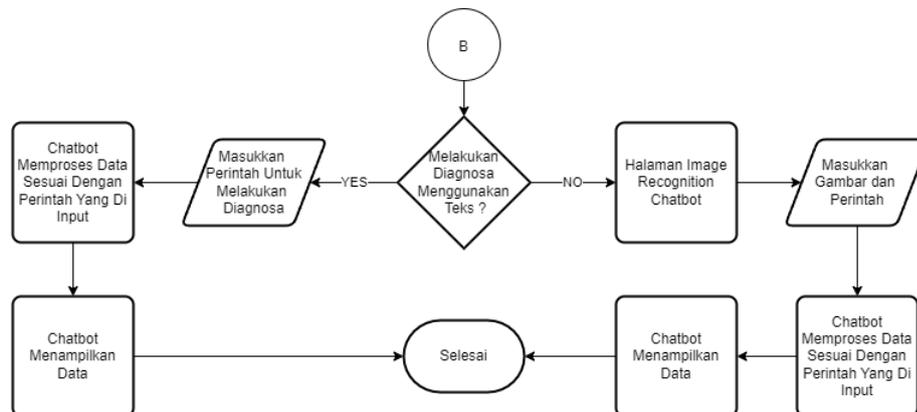
b. *Flowchart* Halaman Utama



Sumber: Hasil Penelitian

**Gambar IV.3** *Flowchart* Halaman Utama

c. *Flowchart* Halaman Chatbot

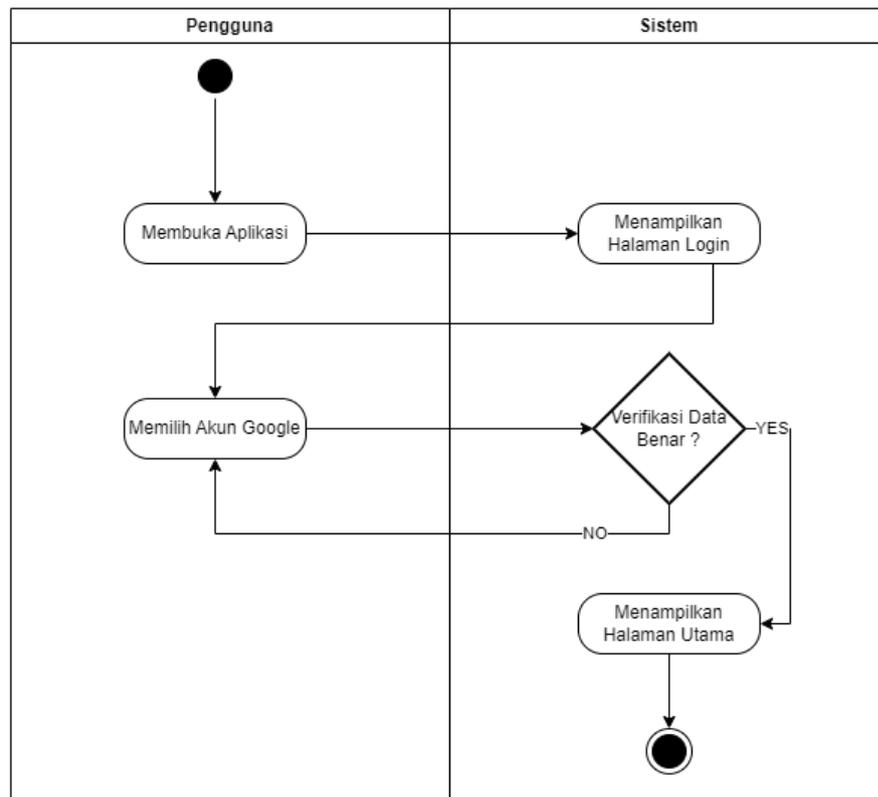


Sumber: Hasil Penelitian

**Gambar IV.4** *Flowchart* Halaman Chatbot

### 3. Activity Diagram

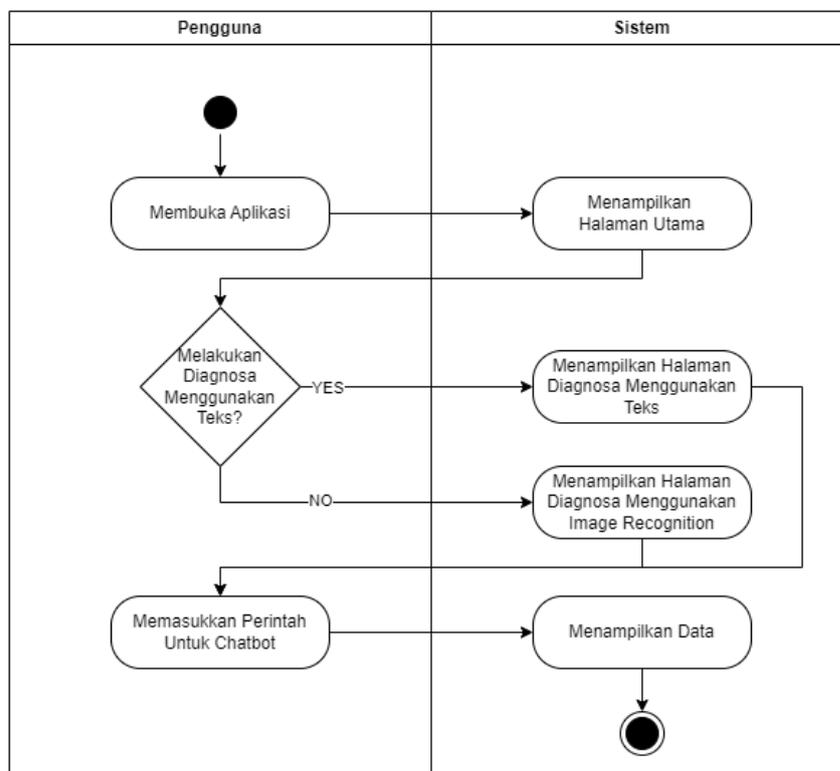
#### a. Activity Diagram Login



Sumber: Hasil Penelitian

**Gambar IV. 5 Activity Diagram Login**

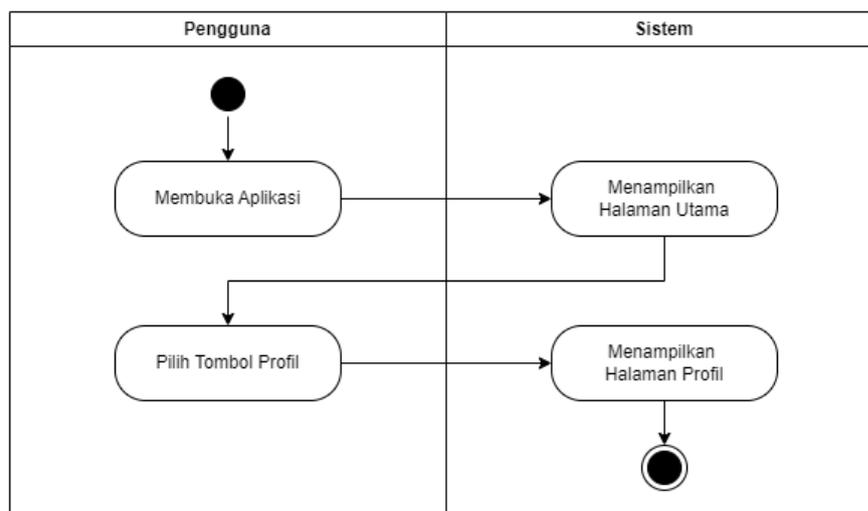
b. *Activity Diagram Chatbot*



Sumber: Hasil Penelitian

**Gambar IV. 6 Activity Diagram Chatbot**

c. *Activity Diagram Profil*



Sumber: Hasil Penelitian

**Gambar IV. 7 Activity Diagram Profil**

### 4.1.3 User Interface

Berikut adalah rancangan antarmuka pengguna dari Aplikasi Diagnosa Penyakit Kulit Menggunakan *Chatbot*:

#### 1. Halaman *Login*

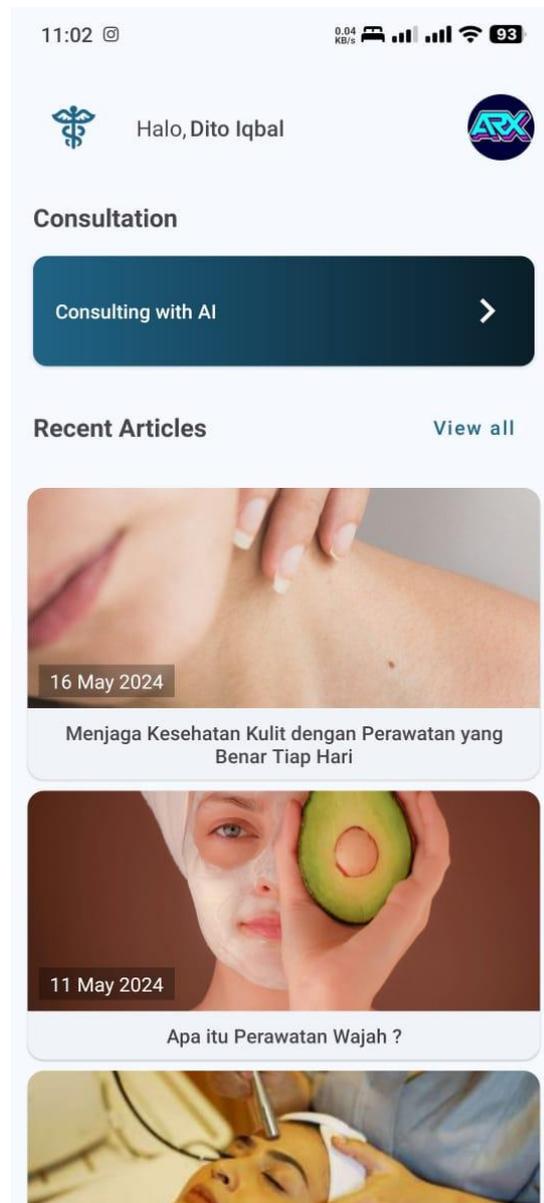


Sumber: Hasil Penelitian

#### **Gambar IV. 8 User Interface Halaman *Login***

Didalam tampilan Halaman *Login* ini terdapat logo aplikasi, pesan teks, dan sebuah tombol untuk masuk menggunakan akun *Google*.

## 2. Halaman Utama

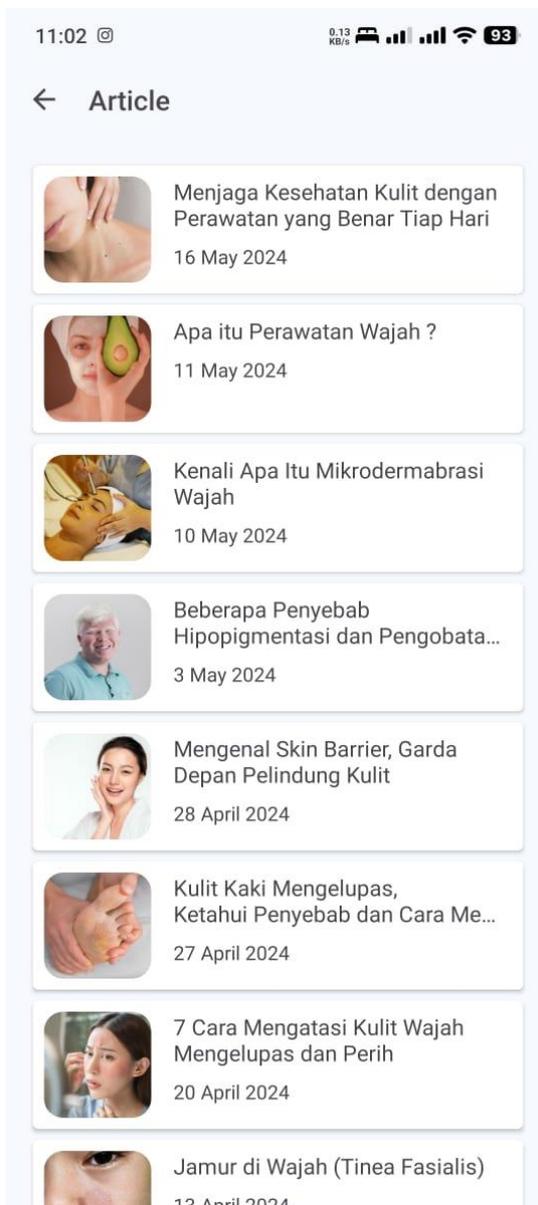


Sumber: Hasil Penelitian

**Gambar IV. 9** *User Interface* Halaman Utama

Didalam tampilan Halaman Utama ini terdapat tombol-tombol seperti tombol profil yang bergambar foto profil akun pengguna, tombol konsultasi untuk menavigasi ke halaman konsultasi, *list* artikel dan tombol *View all* untuk melihat semua *list* artikel.

### 3. Halaman *List* Artikel



Sumber: Hasil Penelitian

**Gambar IV. 10 *User Interface List* Artikel**

Didalam tampilan halaman *list* artikel terdapat *list-list* artikel yang lebih lengkap dari halaman utama, apabila satu artikel diklik akan dinavigasikan ke halaman detail artikel, dan terdapat tombol kembali untuk ke halaman sebelumnya.

#### 4. Halaman Detail Artikel

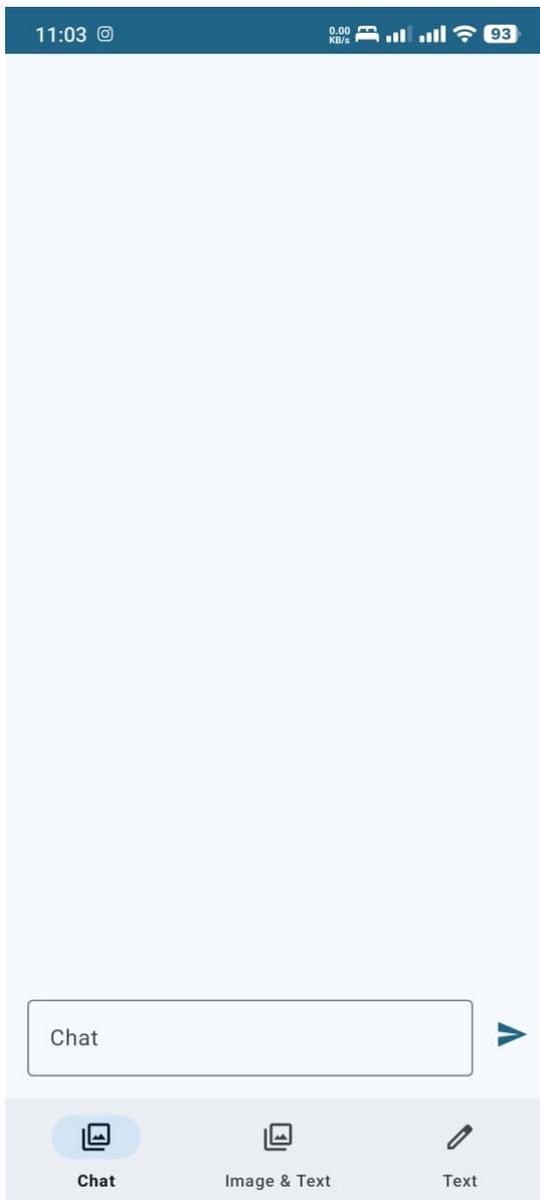


Sumber: Hasil Penelitian

#### **Gambar IV. 11 User Interface Halaman Detail Artikel**

Didalam tampilan halaman detail artikel ini terdapat isi detail dari sebuah artikel yang diklik oleh pengguna. Yang terdapat judul artikel, foto, tanggal dan isi artikel.

## 5. Halaman Konsultasi



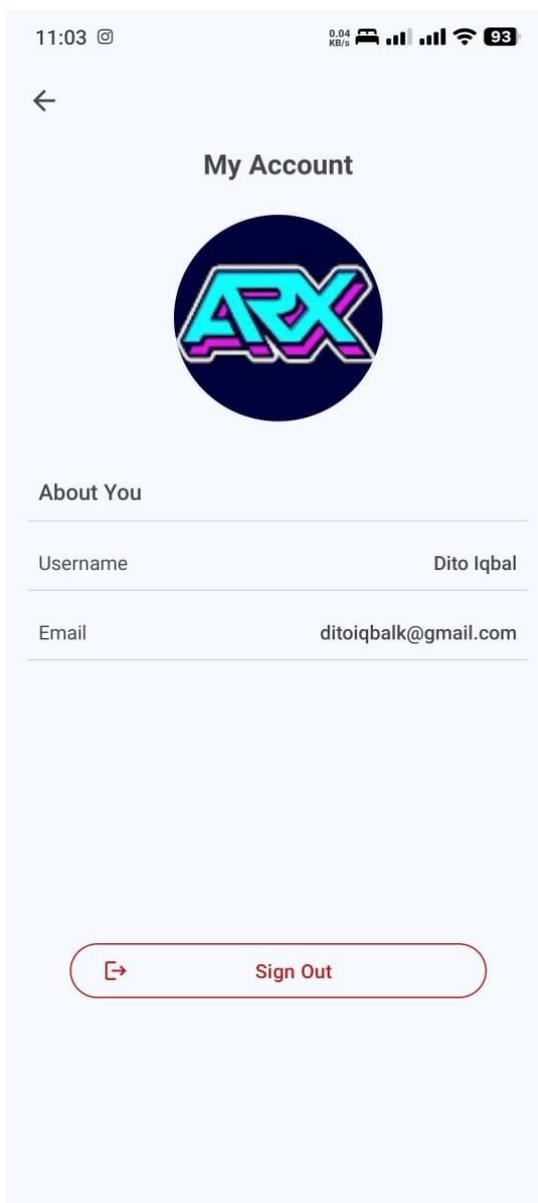
Sumber: Hasil Penelitian

**Gambar IV. 12 *User Interface* Halaman Konsultasi**

Didalam halaman konsultasi ini terdapat *input form* untuk pengguna melakukan konsultasi dengan *bot* dan tombol kirim untuk mengirim *input* ke *server*. Dan juga terdapat 3 navigasi bawah yaitu *Chat*, *Image & Text* dan *Text* untuk

melakukan konsultasi sesuai kebutuhan seperti konsultasi berkepanjangan seperti *chat*, *Image & Text* untuk berkonsultasi menggunakan *input* gambar dan teks. *Text* untuk berkonsultasi hanya dengan sekali *input* tidak seperti *chat*.

## 6. Halaman Profil



Sumber: Hasil Penelitian

**Gambar IV. 13** *User Interface* Halaman Profil

Didalam halaman profil ini terdapat foto profil dari akun pengguna, detail dari akun pengguna seperti *username* dan *email*. Dan juga terdapat sebuah tombol *Sign Out* untuk keluar dari aplikasi.

## 4.2 Code Generation

### 1. Gemini Module

```

@Module InstallIn(SingletonComponent::class)
object GeminiModule {
    private val harassment = SafetySetting(HarmCategory.HARASSMENT, BlockThreshold.ONLY_HIGH)
    private val hateSpeech = SafetySetting(HarmCategory.HATE_SPEECH,
BlockPrivateValMEDA00gANDg80V6)ionConfig {
        temperature = 0.99f
        topK = 50
        topP = 0.99f
    }
    @Provides Singleton GeminiPro]
    fun provideGemini(): GenerativeModel {
        return GenerativeModel(
            modelName = "gemini-pro",
            apiKey = API_KEY,
            safetySettings = listOf(
                harassment, hateSpeech
            ),
            generationConfig = config
        )
    }
    @Provides Singleton GeminiProVision]
    fun provideGeminiVision(): GenerativeModel {
        return GenerativeModel(
            modelName = "gemini-1.5-flash",
            apiKey = API_KEY,
            safetySettings = listOf(
                harassment,
                hateSpeech,
                SafetySetting(HarmCategory.DANGEROUS_CONTENT, BlockThreshold.ONLY_HIGH),
                SafetySetting(HarmCategory.SEXUALLY_EXPLICIT, BlockThreshold.MEDIUM_AND_ABOVE)
            ),
        )
    }
    private const val API_KEY = "YOUR API KEY"
}
@Qualifier
@Retention(AnnotationRetention.BINARY)
annotation class GeminiPro
@Qualifier
@Retention(AnnotationRetention.BINARY)
annotation class GeminiProVision
return go(f, seed, [])
}

```

Gambar IV. 14 Code Generation Gemini Module

## 2. Konsultasi *Chat*

```

@AndroidEntryPoint
class ChatFragment : Fragment() {
    private var _binding : FragmentChatBinding? = null
    private val binding : FragmentChatBinding get() = _binding!!
    private val viewModel by viewModels<ChatViewModel>()
    private val chatAdapter = ChatAdapter()
    private val messageList = mutableListOf<Pair<String, Int>>()
    private var retryCount = 0
    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View {
        _binding = FragmentChatBinding.inflate(inflater, container, false)
        return binding.root
    }
    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        setAdapter()
        sendMessage()
        observe()
    }
    private fun setAdapter(){
        binding.chatRv.adapter = chatAdapter
    }
    private fun sendMessage(){
        binding.chatSend.setOnClickListener {
            val userMessage = binding.chatPromptTextEt.text.toString().trim()
            if (userMessage.isEmpty()) {
                Toast.makeText(context, "Please enter a message prompt", Toast.LENGTH_SHORT).show()
                return@setOnClickListener
            }
            try {
                viewModel.geminiChat(userMessage)
                messageList.add(Pair(userMessage, ChatAdapter.VIEW_TYPE_USER))
                chatAdapter.setMessages(messageList)
                scrollPosition()
                binding.chatPromptTextEt.setText("")
                binding.chatPromptProgress.visibility = View.VISIBLE
                binding.chatSend.visibility = View.GONE
                val inputMethodManager = requireContext().getSystemService(Context.INPUT_METHOD_SERVICE)
                as InputMethodManager
                inputMethodManager.hideSoftInputFromWindow(it.windowToken, 0)
            } catch (e: Exception) {
                e.printStackTrace()
                if (e.message?.contains("overload") == true || e.message?.contains("server busy") ==
                true) {
                    // Handle server overload error
                    if (retryCount < 2) { // Retry up to 2 times
                        retryCount++
                        val handler = Handler(Looper.getMainLooper())
                        handler.postDelayed({
                            viewModel.geminiChat(userMessage)
                        }, 1000) // Retry after 1 second
                    } else {
                        Toast.makeText(context, "Server overload. Please try again later.",
                        Toast.LENGTH_SHORT).show()
                        // Handle max retries reached (optional)
                    }
                } else if (e.message?.contains("SAFETY") == true) {
                    // Handle safety violation error (existing code)
                    Toast.makeText(context, "Your prompt violates safety guidelines. Please rephrase.",
                    Toast.LENGTH_SHORT).show()
                } else {
                    Log.e("AI Error", e.message.toString())
                    Toast.makeText(context, "An error occurred. Please try again later.",
                    Toast.LENGTH_SHORT).show()
                }
                binding.chatPromptProgress.visibility = View.GONE
                binding.chatSend.visibility = View.VISIBLE
            }
        }
    }
    private fun observe(){
        viewModel.messageResponse.observe(viewLifecycleOwner){content ->
            binding.chatPromptProgress.visibility = View.GONE
            binding.chatSend.visibility = View.VISIBLE
            content.text?.let {
                messageList.add(Pair(it, ChatAdapter.VIEW_TYPE_GEMINI))
                chatAdapter.setMessages(messageList)
                scrollPosition()
            }
        }
    }
    private fun scrollPosition(){
        binding.chatRv.smoothScrollToPosition(chatAdapter.itemCount - 1)
    }
}

```

Gambar IV. 15 *Code Generation* Konsultasi *Chat*

3. Konsultasi *Image & Text*

```

class ImageFragment : Fragment() {
    private var _binding: FragmentImageBinding? = null
    private val binding: FragmentImageBinding get() = _binding!!
    private val viewModel by viewModels<ImageViewModel>()
    private var pickBitmap = mutableListOf<Bitmap?>()
    private var fullResponse : String = ""
    private val imageAdapter = ImageAdapter()
    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?, savedInstanceState: Bundle?): View {
        _binding = FragmentImageBinding.inflate(inflater, container, false)
        return binding.root }
    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState) sendPrompt() observer()
        binding.imageRv.adapter = imageAdapter
        binding.imageBtn.setOnClickListener {
            pickPhoto() binding.imagePromptResponse.text = ""
        }
        binding.btnDelete.setOnClickListener {
            pickBitmap.clear() imageAdapter.setImageList(pickBitmap)
            binding.imagePromptResponse.text = ""
            binding.imagePromptTextEt.setText("")
        }
        private fun sendPrompt() {
            binding.imageSendPrompt.setOnClickListener { fullResponse = ""
                binding.imagePromptResponse.text = ""
                if (pickBitmap.isEmpty() || binding.imagePromptTextEt.text.toString().isEmpty()) {
                    Toast.makeText(context, "Please input a photo and input prompt", Toast.LENGTH_SHORT)
                        .show()
                }
                return@setOnClickListener } else {
                    CoroutineScope(Dispatchers.IO).launch {
                        val inputContent = content { pickBitmap.forEach { it?.let {image(compressBitmap(it))}
                            } }text(binding.imagePromptTextEt.text.toString())
                        }viewModel.geminiPromptResponse(inputContent)
                        }binding.imagePromptProgress.visibility = View.VISIBLE}}
        private fun compressBitmap(bitmap: Bitmap): Bitmap {
            val scaledBitmap = Bitmap.createScaledBitmap(bitmap, 1024, 1024, true)
            val byteArrayOutputStream = ByteArrayOutputStream()
            scaledBitmap.compress(Bitmap.CompressFormat.JPEG, 80, byteArrayOutputStream)
            return BitmapFactory.decodeByteArray(byteArrayOutputStream.toByteArray(), 0,
                byteArrayOutputStream.size())
        }
        private fun observer() {
            viewModel.promptResponse.observe(viewLifecycleOwner) { response ->
                binding.imagePromptProgress.visibility = View.GONE
                if (response.text.isNullOrEmpty()) {
                    Toast.makeText(requireContext(), "Error: No response from Gemini AI",
                        Toast.LENGTH_SHORT).show()return@observe}fullResponse += response.text
                binding.imagePromptResponse.text = fullResponse}}
        private fun pickPhoto() {
            if (ContextCompat.checkSelfPermission(requireContext(), Manifest.permission.READ_EXTERNAL_STORAGE
                )!= PackageManager.PERMISSION_GRANTED) {
                galleryLauncher.launch("image/*")
                galleryPermission.launch(Manifest.permission.READ_EXTERNAL_STORAGE)
            } else {galleryIntent() galleryLauncher.launch("image/*")}}
        private fun galleryIntent() {
            startActivity(Intent(Intent.ACTION_PICK, MediaStore.Images.Media.EXTERNAL_CONTENT_URI))
            @Suppress("DEPRECATION")
            private val galleryLauncher = registerForActivityResult( ActivityResultContracts.GetContent(),
                ) { uri -> uri?.let { if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.P) {
                    val source = ImageDecoder.createSource(requireContext().contentResolver, uri)
                    ImageDecoder.decodeBitmap(source)
                } else {MediaStore.Images.Media.getBitmap(requireContext().contentResolver, it)
                } }.also { bitmap -> pickBitmap.add(bitmap)imageAdapter.setImageList(pickBitmap)}}
        private val galleryPermission = registerForActivityResult(
            ActivityResultContracts.RequestPermission()
        ) { isGranted: Boolean -> if (isGranted) {galleryIntent()}}
    }
}

```

Gambar IV. 16 *Code Generation* Konsultasi *Image & Text*

#### 4. Konsultasi Teks

```

@AndroidEntryPoint
class TextFragment : Fragment() {
    private var _binding : FragmentTextBinding? = null
    private val binding : FragmentTextBinding get() = _binding!!
    private val viewModel by viewModels<TextViewModel>()

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View {
        _binding = FragmentTextBinding.inflate(inflater,container,false)
        return binding.root
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)
        sendPrompt()
        observe()
    }

    private fun sendPrompt(){
        binding.textSendPrompt.setOnClickListener {
            val prompt = binding.textPromptTextEt.text.toString().trim()
            if (prompt.isNotEmpty()) {
                viewModel.geminiTextPrompt(binding.textPromptTextEt.text.toString().trim())
                binding.textPromptProgress.visibility = View.VISIBLE
            } else {
                Toast.makeText(context, "Please enter a message prompt",
                    Toast.LENGTH}SHORT).show()
            }
        }
    }

    private fun observe(){
        viewModel.promptResponse.observe(viewLifecycleOwner){content ->
            content.text?.let {
                binding.textPromptResponse.text = it
                binding.textPromptProgress.visibility = View.GONE
            }
            println("Candidates ${content.candidates}")
            println("PromptFeedback ${content.promptFeedback}")
        }
    }
}

```

**Gambar IV. 17 Code Generation Konsultasi Text**

#### 4.3 Testing

Pada tahap pengujian ini, aplikasi yang sudah dibuat dilakukan uji coba apakah aplikasi sudah optimal atau tidak. Pengujian ini dilakukan dengan Teknik *Blackbox Testing* yang lebih memfokuskan pengujian pada fungsionalitas di aplikasi tersebut.

1. Pengujian *form* konsultasi *chat*Tabel IV. 7 Pengujian *form* konsultasi *chat*

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Keterangan
1	<i>Input field prompt chat</i> dikosongkan lalu tekan tombol kirim	<i>Input chat</i> kosong	Sistem menampilkan peringatan " <i>Please enter a message prompt</i> "	Valid

Sumber: Hasil Penelitian

2. Pengujian *form* konsultasi *image & text*Tabel IV. 8 Pengujian *form* konsultasi *image & text*

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Keterangan
1	<i>Input field prompt</i> gambar dan teks dikosongkan lalu tekan tombol kirim	Gambar: Kosong  <i>Form</i> : Kosong	Sistem menampilkan peringatan " <i>Please input a photo and input prompt</i> "	Valid
2	<i>Input field prompt</i> gambar dikosongkan lalu tekan tombol kirim	Gambar: Kosong  <i>Form</i> : isi	Sistem menampilkan peringatan " <i>Please input a photo and input prompt</i> "	Valid
3	<i>Input field prompt</i> teks dikosongkan lalu tekan tombol kirim	Gambar: isi  <i>Form</i> : kosong	Sistem menampilkan peringatan " <i>Please input a photo and input prompt</i> "	Valid

Sumber: Hasil Penelitian

### 3. Pengujian *form* konsultasi *text*

**Tabel IV. 9 Pengujian *form* konsultasi *text***

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Keterangan
1	<i>Input field prompt</i> dikosongkan lalu tekan tombol kirim	<i>Input text</i> kosong	Sistem menampilkan peringatan “ <i>Please enter a message prompt</i> ”	Valid

Sumber: Hasil Penelitian

## 4.4 *Support*

### 4.4.1 *Publikasi Software*

Untuk publikasi *software* aplikasi *Android* itu membutuhkan akun *Developer Google Play Console* untuk memulai publikasi di *Google Play Store*. Untuk membuat akunnya memerlukan biaya 25 USD atau sekitar Rp 409.306,25.

Setelah sudah memiliki akun, aplikasi yang sudah dibuat di-*build* menjadi sebuah *Bundle* yang sudah menjadi standarisasi *Google* dan harus memenuhi persyaratan pengujian khusus sebelum aplikasinya dapat tersedia di *Google Play Store*.

### 4.4.2 *Spesifikasi Hardware dan Software*

Berikut adalah spesifikasi yang penulis gunakan untuk membuat aplikasi mulai dari spesifikasi perangkat keras sampai perangkat lunak untuk menjalankan aplikasi.

Tabel IV. 10 Spesifikasi *Hardware*

<b>Kebutuhan</b>	<b>Keterangan</b>
Sistem Operasi	Windows 11 Home 64-bit
<i>Processor</i>	AMD Ryzen 5 5600U with Radeon Graphics 2.30 GHz
RAM	16GB
Hardisk	512GB
VGA	Nvidia Geforce RTX 3050
Software	<i>Android Studio, Figma, Firebase</i>

Sumber: Hasil Penelitian

Tabel IV. 11 Spesifikasi *Android untuk Software*

<b>Kebutuhan</b>	<b>Keterangan</b>
Sistem Operasi	<i>Android 14</i>
CPU	Snapdragon 685 Octa-core 2.80 GHz
RAM	8GB
Hardisk	256GB
Internet	<i>Mobile data</i> atau WiFi

Sumber: Hasil Penelitian

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Penulis mengembangkan aplikasi diagnosa penyakit kulit berbasis *Android* untuk memudahkan para pasien yang ingin melakukan diagnosa dan pencegahan dini sebelum penyakitnya bertambah buruk. Aplikasi ini dikembangkan dikarenakan dokter ahli atau dokter spesialis yang menangani kulit terbatas dan juga biaya untuk membuat perjanjian dan melakukan diagnosa tidaklah murah yang bisa dikeluarkan oleh calon pasien.

Penelitian ini bertujuan untuk mendiagnosa penyakit kulit pada manusia berbasis aplikasi *Android*. Berdasarkan hasil penelitian yang telah dilakukan, maka penulis menyimpulkan bahwa:

1. Melalui aplikasi diagnosa penyakit kulit ini, pengguna dapat lebih mudah mengkonsultasikan dan mendiagnosa gejala-gejala penyakit kulit yang dideritanya.
2. Sistem yang digunakan dalam mendiagnosa menggunakan teknologi kecerdasan buatan berupa *Chatbot* dan menggunakan *chatbot* buatan Google yang bernama Gemini AI.
3. Aplikasi ini memberikan solusi yang efektif bagi pengguna maupun dokter dalam melakukan pendiagnosaan awal tentang penyakit kulit.
4. Memberikan kemudahan untuk pengguna dalam konsultasi tanpa harus membayar dan tanpa harus mengantri secara langsung di rumah sakit.

## 5.2 Saran

Penulis memiliki beberapa saran untuk pengembangan aplikasi diagnosa penyakit kulit ini agar dapat dikembangkan lebih optimal lagi pada penelitian selanjutnya. Berikut adalah saran yang dapat penulis berikan:

1. Menggunakan *chatbot* yang dibuat dan dilatih sendiri agar ruang lingkup yang bisa dibahas oleh *chatbot* nya ini hanya sebatas tentang penyakit kulit manusia dan cara pencegahan serta penanganan.
2. Menambahkan fitur lain seperti membeli obat atau rumah sakit terdekat yang memiliki dokter spesialis kulit.
3. Menambahkan opsi login lainnya seperti menggunakan email yang didaftarkan melalui aplikasi.



## DAFTAR PUSTAKA

- Albab, M. U., Karuniawati P, Y., & Fawaiq, M. N. (2023). Optimization of the Stemming Technique on Text preprocessing President 3 Periods Topic. *Jurnal Transformatika*, 20(2), 1–10.  
<https://journals.usm.ac.id/index.php/transformatika/page1>
- Alexandra, W. (2022). Penerapan Teknologi Augmented Reality Berbasis Android Untuk Pembelajaran Rantai Makanan Pada Hewan. *Jurnal Informatika dan Rekayasa Perangkat Lunak*, 3(1), 107–116.  
<https://doi.org/10.33365/jatika.v3i1.1864>
- Ananto, D. T., Duta Mahardewantoro, D., Mustafa, F., Ardianto, M. G., Rafi, M. M., Zein, R. A., Saputra, O. E., Mujiastuti, R., Rosanti, N., & Adharani, Y. (2023). *Prosiding Seminar Nasional LPPM UMJ Website: Edukasi dan Pelatihan Pengenalan Machine Learning dan Computer Vision Untuk Mengeksplorasi Potensi Visual*. <http://jurnal.umj.ac.id/index.php/semnaskat> E-ISSN: 2714-6286
- Barjakly, Y., Koç, H., Erdoğan, A. M., & Peker, S. (2021). *UML Diagrams in Software Engineering Research: A Systematic Literature Review*. 13.  
<https://doi.org/10.3390/proceedings2021074013>
- Efrian, M. R., & Latifa, U. (2022). Image Recognition Berbasis Convolutional Neural Network (Cnn) Untuk Mendeteksi Penyakit Kulit Pada Manusia. *Power Elektronik : Jurnal Orang Elektro*, 11(2), 276.  
<https://doi.org/10.30591/polektro.v12i1.3874>

- Furqan, M., Sriani, S., & Shidqi, M. N. (2023). Chatbot Telegram Menggunakan Natural Language Processing. *Walisongo Journal of Information Technology*, 5(1), 15–26. <https://doi.org/10.21580/wjit.2023.5.1.14793>
- Haryanto, I. D., & Saefurrahman, S. (2024). Implementasi Chatbot Kesehatan Kucing Melalui Dialogflow dan Telegram untuk Pemberian Informasi Penyakit dan Perawatan. *JTIM : Jurnal Teknologi Informasi dan Multimedia*, 5(4), 365–376. <https://doi.org/10.35746/jtim.v5i4.484>
- Jaya, H., Sabran, D., Pd, M., Ma, M., Djawad, Y. A., Sc, M., Ilham, A., Ahmar, A. S., Si, S., & Sc, M. (2019). Kecerdasan Buatan. In *Journal of Chemical Information and Modeling* (Vol. 53, Nomor 9).
- Meilinda, E., Sabaruddin, R., & Fitriani, D. (2021). Model Prototype Sebagai Metode Pengembangan Perangkat Lunak Pada Sistem Informasi Pengaduan Umum (Studi Kasus : Dinas Perhubungan Provinsi Kalimantan Barat). *Jurnal Khatulistiwa Informatika*, 9(2), 86–91. <https://doi.org/10.31294/jki.v9i2.11753>
- Musthofa, N., & Adiguna, M. A. (2022). Perancangan Aplikasi E-Commerce Spare-Part Komputer Berbasis Web Menggunakan CodeIgniter Pada Dhamar Putra Computer Kota Tangerang. *Jurnal Ilmu Komputer dan Science*, 1(03), 199–207. <https://journal.mediapublikasi.id/index.php/oktal>
- Nabillah, R. (2021). Prevalensi Dermatitis Seboroik Di Poli Kulit Dan Kelamin RSUD Meuraxa Kota Banda Aceh Periode Tahun 2016-2019. *Jurnal Health Sains*, 2(1), 112–119. <https://doi.org/10.46799/jhs.v2i1.77>
- Paksi, A. B., Hafidhoh, N., & Bimonugroho, S. K. (2023). Perbandingan Model Pengembangan Perangkat Lunak Untuk Proyek Tugas Akhir Program Vokasi.

*Jurnal Masyarakat Informatika*, 14(1), 70–79.

<https://doi.org/10.14710/jmasif.14.1.52752>

Perera, P., & Lankathilake, M. (2023). Preparing to Revolutionize Education with the Multi-Model GenAI Tool Google Gemini? A Journey towards Effective Policy Making. *Journal of Advances in Education and Philosophy*, 7(08), 246–253.

<https://doi.org/10.36348/jaep.2023.v07i08.001>

Prasetyo, V. R., Benarkah, N., & Chrisintha, V. J. (2021). Implementasi Natural Language Processing Dalam Pembuatan Chatbot Pada Program Information Technology Universitas Surabaya. *Teknika*, 10(2), 114–121.

<https://doi.org/10.34148/teknika.v10i2.370>

Sitohang, N. (2023). Sistem Pakar Diagnosa Penyakit Kulit Pada Manusia Menggunakan Metode Hybrid. *Penerapan Data Mining Untuk Peringatan Dini Banjir Menggunakan Metode Klustering K-Means*, 2(1), 16–20.

Tarkus, E. D., Sompie, S. R. U. A., & Jacobus, A. (2020). Implementasi Metode Recurrent Neural Network pada Pengklasifikasian Kualitas Telur Puyuh. *Jurnal Teknik Informatika*, 15(2), 137–144.

<https://ejournal.unsrat.ac.id/v3/index.php/informatika/article/view/29552>

## DAFTAR RIWAYAT HIDUP

### I. Biodata Mahasiswa

NIM : 17200402  
Nama Lengkap : Dito Iqbal Kusuma  
Tempat/Tanggal Lahir : Depok, 02 November 2002  
Alamat Lengkap : Jalan Kedondong RT. 01 RW. 15 No. 36,  
Kelurahan Kemiri Muka, Kecamatan Beji,  
Kota Depok, Provinsi Jawa Barat. Kode Pos  
16423

### II. Pendidikan

1. SD Negeri Beji Timur 3 Depok, lulus tahun 2014
2. SMP Negeri 5 Depok, lulus tahun 2017
3. SMK Bakti 17 Jakarta Selatan, lulus tahun 2020

### III. Riwayat Pengalaman Berorganisasi

1. Anggota Organisasi Intra Sekolah (OSIS) SMK Bakti 17 Jakarta Selatan, tahun 2017 s.d. tahun 2019



Depok, 5 Juli 2024

Dito Iqbal Kusuma

**SURAT PERNYATAAN KEBENARAN/KEABSAHAN DATA  
HASIL RISET UNTUK KARYA ILMIAH**

Yang bertanda tangan dibawah ini, saya:

Nama : Dito Iqbal Kusuma  
NIM : 17200402  
Jenjang : Sarjana (S1)  
Program Studi : Teknologi Informasi  
Fakultas : Fakultas Teknik dan Informatika  
Perguruan Tinggi : Universitas Bina Sarana Informatika

Dengan ini menyatakan bahwa data dan atau informasi yang saya gunakan dalam penulisan karya ilmiah Penulis dengan judul **"Implementasi Chatbot Untuk Mendiagnosa Penyakit Kulit Pada Manusia Berbasis Aplikasi Android"** merupakan data dan atau informasi yang saya peroleh berdasarkan hasil PKL/Riset Secara Daring (Online) pada:

Nama Perusahaan : Klinik Medika Occupational Health Center (OHC)  
Alamat Perusahaan : Jl. Batu Ampar 3 No.55, RT.11/RW.2, Batu Ampar, Kec. Kramat jati, Kota Jakarta Timur, Daerah Khusus Ibukota Jakarta 13520

Melalui Alamat Website : <https://medika-ohc.co.id/>

Saya bersedia untuk bertanggung jawab secara pribadi, tanpa melibatkan pihak **Universitas Bina Sarana Informatika**, atas materi/isi karya ilmiah tersebut, termasuk bertanggung jawab atas dampak atau kerugian yang timbul dalam bentuk akibat tindakan yang berkaitan dengan data dan atau informasi yang terdapat pada karya ilmiah saya ini.

Demikian pernyataan ini saya buat dengan sebenarnya.

Mengetahui,

Dosen Pembimbing



Wati Erawati, M.Kom

Dibuat di : Jakarta  
Pada Tanggal : 5 Juli 2024  
Yang menyatakan,



Dito Iqbal Kusuma

## BAB 1 - 5 (35).docx

### ORIGINALITY REPORT

**24%**

SIMILARITY INDEX

**21%**

INTERNET SOURCES

**12%**

PUBLICATIONS

**11%**

STUDENT PAPERS

### PRIMARY SOURCES

<b>1</b>	<b>files.osf.io</b> Internet Source	<b>1%</b>
<b>2</b>	<b>Submitted to Universitas Maritim Raja Ali Haji</b> Student Paper	<b>1%</b>
<b>3</b>	<b>Submitted to Universitas Putera Batam</b> Student Paper	<b>1%</b>
<b>4</b>	<b>repository.nusamandiri.ac.id</b> Internet Source	<b>1%</b>
<b>5</b>	<b>media.neliti.com</b> Internet Source	<b>1%</b>
<b>6</b>	<b>upgraded.id</b> Internet Source	<b>1%</b>
<b>7</b>	<b>docplayer.info</b> Internet Source	<b>1%</b>
<b>8</b>	<b>repository.atmaluhur.ac.id</b> Internet Source	<b>1%</b>
<b>9</b>	<b>Submitted to Universitas Amikom</b> Student Paper	<b>1%</b>

10	<a href="http://www.karyailmiah.trisakti.ac.id">www.karyailmiah.trisakti.ac.id</a> Internet Source	1 %
11	<a href="http://repository.radenintan.ac.id">repository.radenintan.ac.id</a> Internet Source	1 %
12	<a href="http://tugasdenny.wordpress.com">tugasdenny.wordpress.com</a> Internet Source	1 %
13	Eva Meilinda, Raja Sabaruddin, Dewi Fitriani. "MODEL PROTOTYPE SEBAGAI METODE PENGEMBANGAN PERANGKAT LUNAK PADA SISTEM INFORMASI PENGADUAN UMUM (STUDI KASUS : DINAS PERHUBUNGAN PROVINSI KALIMANTAN BARAT)", Jurnal Khatulistiwa Informatika, 2021 Publication	1 %
14	<a href="http://repository.ittelkom-pwt.ac.id">repository.ittelkom-pwt.ac.id</a> Internet Source	<1 %
15	Anna Anna, Nurmalasari Nurmalasari, Yayuk Rohayani. "Penerapan Metode Waterfall Dalam Perancangan Sistem Informasi Akuntansi Pengiriman Barang", Jurnal Sistem Informasi Akuntansi, 2021 Publication	<1 %
16	<a href="http://doaj.org">doaj.org</a> Internet Source	<1 %
17	<a href="http://repositori.unsil.ac.id">repositori.unsil.ac.id</a> Internet Source	<1 %

18	<a href="http://sipora.polije.ac.id">sipora.polije.ac.id</a> Internet Source	<1 %
19	<a href="http://www.softwareseni.co.id">www.softwareseni.co.id</a> Internet Source	<1 %
20	<a href="http://123dok.com">123dok.com</a> Internet Source	<1 %
21	<a href="http://eprints.bsi.ac.id">eprints.bsi.ac.id</a> Internet Source	<1 %
22	<a href="http://repository.uir.ac.id">repository.uir.ac.id</a> Internet Source	<1 %
23	<a href="http://repository.universitasbumigora.ac.id">repository.universitasbumigora.ac.id</a> Internet Source	<1 %
24	<a href="http://repository.unugha.ac.id">repository.unugha.ac.id</a> Internet Source	<1 %
25	<a href="http://text-id.123dok.com">text-id.123dok.com</a> Internet Source	<1 %
26	Submitted to Universitas Pendidikan Indonesia Student Paper	<1 %
27	<a href="http://repository.ubaya.ac.id">repository.ubaya.ac.id</a> Internet Source	<1 %
28	Joko Dwi Mulyanto, Fabriyan Fandi Dwi Imaniawan, Mustofa Mustofa, Ibnu Alfarobi. "Implementasi Metode Prototype pada	<1 %

Sistem Informasi Pemesanan Kaos Sablon CV  
Jiyó'g Berbasis Website", Bianglala  
Informatika, 2023

Publication

29	<a href="http://ekajogja.com">ekajogja.com</a> Internet Source	<1 %
30	<a href="http://journal.pancabudi.ac.id">journal.pancabudi.ac.id</a> Internet Source	<1 %
31	<a href="http://journal.sekawan-org.id">journal.sekawan-org.id</a> Internet Source	<1 %
32	<a href="http://www.unisbank.ac.id">www.unisbank.ac.id</a> Internet Source	<1 %
33	Yayan Hendrian. "Aplikasi Laporan Harian Teknisi CV. Cipta Multi Solution Berbasis Website", Jurnal Infortech, 2023 Publication	<1 %
34	<a href="http://binus.ac.id">binus.ac.id</a> Internet Source	<1 %
35	<a href="http://bpkpenabur.or.id">bpkpenabur.or.id</a> Internet Source	<1 %
36	<a href="http://forum.detik.com">forum.detik.com</a> Internet Source	<1 %
37	Hilman Septian, Eka Wahyu Hidayat, Alam Rahmatulloh. "Aplikasi Pengenalan Bahasa Arab dan Inggris untuk Anak-Anak Berbasis Android", Jurnal Online Informatika, 2018	<1 %

Publication

38	Submitted to Morgan Park High School Student Paper	<1 %
39	Submitted to Universitas Bina Darma Student Paper	<1 %
40	Submitted to Universitas Muria Kudus Student Paper	<1 %
41	<a href="http://ojs.uma.ac.id">ojs.uma.ac.id</a> Internet Source	<1 %
42	<a href="http://repository.uin-suska.ac.id">repository.uin-suska.ac.id</a> Internet Source	<1 %
43	<a href="http://www.linux.org">www.linux.org</a> Internet Source	<1 %
44	<a href="http://fr.scribd.com">fr.scribd.com</a> Internet Source	<1 %
45	<a href="http://repository.unama.ac.id">repository.unama.ac.id</a> Internet Source	<1 %
46	Yulia Yulia, Nanda Diaz Arizona, Tasya Anisa. "Sistem Informasi Akuntansi Pendapatan Jasa Dan Penjualan Suku Cadang Pada Bengkel Podomoro Motor Pontianak Berbasis Web", Jurnal Sistem Informasi Akuntansi, 2021 Publication	<1 %
47	<a href="http://elib.uni-stuttgart.de">elib.uni-stuttgart.de</a> Internet Source	<1 %

48	<a href="https://repositori.usu.ac.id">repositori.usu.ac.id</a> Internet Source	<1 %
49	<a href="https://ristaayustri.blogspot.com">ristaayustri.blogspot.com</a> Internet Source	<1 %
50	<a href="https://www.pinterpandai.com">www.pinterpandai.com</a> Internet Source	<1 %
51	<a href="https://www.rajamobil.com">www.rajamobil.com</a> Internet Source	<1 %
52	Dian, Purnawansyah, Herdianti Darwis, Lilis Nurhayati. "Klasifikasi Penyakit Bawang Merah Menggunakan Naïve Bayes dan Convolutional Neural Network", Indonesian Journal of Computer Science, 2023 Publication	<1 %
53	<a href="https://elibrary.unikom.ac.id">elibrary.unikom.ac.id</a> Internet Source	<1 %
54	<a href="https://repositori.uin-alauddin.ac.id">repositori.uin-alauddin.ac.id</a> Internet Source	<1 %
55	<a href="https://repository.potensi-utama.ac.id">repository.potensi-utama.ac.id</a> Internet Source	<1 %
56	<a href="https://www.coursehero.com">www.coursehero.com</a> Internet Source	<1 %
57	Khoirunnisa devita Sari, Ade Eviyanti. "Expert System for Diagnosing Human Skin Diseases Using Web-Based Naïve Bayes Method",	<1 %

Procedia of Engineering and Life Science,  
2021  
Publication

---

58	Lucky Febrian, Buhori Muslim, Debi Gusmaliza. "Aplikasi Penilaian Lomba Burung Murai Batu Berbasis Android", Jurnal Nasional Ilmu Komputer, 2021 Publication	<1 %
59	beritati.blogspot.co.id Internet Source	<1 %
60	digilib.uinsby.ac.id Internet Source	<1 %
61	edoc.pub Internet Source	<1 %
62	eprints.umm.ac.id Internet Source	<1 %
63	jutif.if.unsoed.ac.id Internet Source	<1 %
64	ojs.amikom.ac.id Internet Source	<1 %
65	pdfcookie.com Internet Source	<1 %
66	pickybest.id Internet Source	<1 %
67	repository.dinamika.ac.id Internet Source	<1 %

---

		<1 %
68	<a href="https://www.slidetodoc.com">slidetodoc.com</a> Internet Source	<1 %
69	<a href="https://smart.stmikplk.ac.id">smart.stmikplk.ac.id</a> Internet Source	<1 %
70	<a href="https://www.kolu.web.id">www.kolu.web.id</a> Internet Source	<1 %
71	<a href="https://www.scilit.net">www.scilit.net</a> Internet Source	<1 %
72	<a href="https://www.scribd.com">www.scribd.com</a> Internet Source	<1 %
73	Ade Fitria Lestari. "Implementasi Extreme Programming Pada Perancangan Sistem Informasi Penjualan Buku Menggunakan Java", JAIS - Journal of Accounting Information System, 2023 Publication	<1 %
74	Mulia Rahmayu, Rosi Kusuma Serli, Mochamad Farhan Aditya. "Pengolahan Data Infrastruktur Suku Dinas Pemuda Dan Olahraga Jakarta Barat Berbasis Web", Journal of Information and Technology, 2021 Publication	<1 %
75	<a href="https://www2.slideshare.net">www2.slideshare.net</a> Internet Source	<1 %

76 Nurmalasari Nurmalasari, Falentina Lidya Mega. "Aplikasi Penjualan Tunai Berbasis Web Pada CV. Damar Abadi Pontianak Pontianak", *Jurnal Sistem Informasi Akuntansi*, 2020 <1%  
Publication

---

77 Sri Watmah, Hanggoro Aji Al Kautsar, Suryanto Suryanto. "Perancangan Sistem Keamanan Brankas Menggunakan Metode Finger Print Berbasis SMS Gateway", *INSANtek*, 2021 <1%  
Publication

---

Exclude quotes  Off

Exclude matches  Off

Exclude bibliography  Off