

## IMPLEMENTASI KECERDASAN BUATAN DALAM PENGEMBANGAN APLIKASI MOBILE BINARYTALKHUB

Rafif Dhia Yusrana, Yukandri, Nurdin, Viktor Handrianus Pranatawijaya, Nova Noor Kamala Sari

Jurusan Teknik Informatika, Fakultas Teknik, Universitas Palangka Raya  
Jl. Yos Sudarso, Kec. Jekan Raya, Palangka Raya, Kalimantan Tengah, Indonesia  
yukandri01@mhs.eng.upr.ac.id

### ABSTRAK

Penelitian ini mengeksplorasi integrasi kecerdasan buatan atau *Artificial Intelligence* (AI) ke dalam aplikasi mobile dengan mengimplementasikan sistem peringkas teks otomatis menggunakan API *ChatGPT*. Dalam lanskap teknologi yang terus berkembang, aplikasi mobile telah menjadi komponen penting dalam aktivitas sehari-hari. Permintaan akan aplikasi yang cerdas dan adaptif semakin meningkat seiring dengan peningkatan penggunaan perangkat seluler. Salah satu tantangan utama adalah kebutuhan untuk meringkas teks panjang secara cepat dan akurat agar pengguna dapat dengan mudah memahami inti informasi tanpa harus membaca keseluruhan dokumen. Penelitian ini bertujuan untuk memberikan solusi melalui pengembangan sistem otomatis yang mampu memberikan ringkasan dari teks panjang dengan memanfaatkan API *ChatGPT*. Metodologi penelitian meliputi studi literatur, pembuatan antarmuka mobile menggunakan *Flutter*, implementasi API *ChatGPT*, dan pengujian sistem. Tahap implementasi melibatkan perolehan kunci API *ChatGPT*, persiapan program dengan bahasa pemrograman yang sesuai, dan integrasi kunci API untuk mengaktifkan fitur peringkas teks otomatis. Hasil penelitian menunjukkan bahwa sistem berhasil menghasilkan ringkasan yang padat namun tetap menyampaikan inti informasi dari dokumen sumber. Meskipun demikian, aplikasi ini memiliki keterbatasan dalam mengolah file PDF dengan ukuran lebih dari 10 MB yang menyebabkan error dan tidak dapat diproses oleh sistem. Terdapat potensi untuk pengembangan lebih lanjut seperti personalisasi sistem, pengeditan langsung, peningkatan keamanan, pengembangan platform lain, dan pengoptimalan performa. Penelitian ini menunjukkan potensi besar integrasi AI dalam aplikasi mobile untuk meningkatkan fungsionalitas dan pengalaman pengguna.

**Kata kunci :** Aplikasi Mobile, Artificial Intelligence, Peringkas Teks, API ChatGPT, Flutter

### 1. PENDAHULUAN

Dalam lanskap teknologi yang terus berkembang dan melekat erat dengan kehidupan sehari-hari, aplikasi mobile telah menjadi komponen penting dalam aktivitas manusia. Seiring dengan peningkatan penggunaan perangkat seluler, tuntutan akan aplikasi yang cerdas dan adaptif juga semakin meningkat. Salah satu pendekatan yang menarik untuk memenuhi kebutuhan tersebut adalah dengan mengintegrasikan *Artificial Intelligence* (AI) ke dalam aplikasi mobile.

Kecerdasan buatan telah menunjukkan potensi besar dalam meningkatkan fungsionalitas dan pengalaman pengguna aplikasi mobile. Salah satu solusi untuk mengatasi kebutuhan ini adalah dengan mengembangkan sistem otomatis yang mampu memberikan ringkasan dari teks panjang secara cepat dan akurat. Sistem pemendek teks otomatis ini memungkinkan pengguna untuk mendapatkan intisari informasi dari teks yang panjang dalam waktu singkat [1].

Untuk mencapai tujuan pemendek teks otomatis, ada dua teknik utama yang dapat digunakan. Pertama, teknik ekstraktif, di mana sistem mengambil kalimat-kalimat penting dari teks asli tanpa mengubah struktur kalimatnya. Kalimat-kalimat tersebut kemudian disusun serupa dengan dokumen aslinya. Kedua, teknik abstraktif, yang tidak hanya mengambil kalimat-kalimat penting, tetapi juga memahami inti informasi dan menyusun kalimat-kalimat baru dengan menggunakan bahasa yang lebih ringkas. Teknik ini

melibatkan pemahaman semantik dan kompresi informasi untuk menghasilkan ringkasan yang padat tetapi tetap menyampaikan inti informasi. Kedua teknik ini diperlukan dalam pengembangan sistem pemendek teks otomatis yang efektif dan kompresi informasi untuk menghasilkan ringkasan yang padat tetapi tetap menyampaikan inti informasi. Kedua teknik ini diperlukan dalam pengembangan sistem pemendek teks otomatis yang efektif [2].

Dalam penelitian ini, kami akan menggunakan API *ChatGPT* untuk melakukan peringkas teks dan mengintegrasikannya ke dalam aplikasi mobile.

### 2. TINJAUAN PUSTAKA

#### 2.1. Peringkas Teks

Peringkas teks adalah teknologi yang bertujuan untuk menghasilkan versi singkat dari sebuah dokumen panjang dengan tetap mempertahankan informasi penting dan makna utamanya. Terdapat dua pendekatan utama dalam peringkas teks:

##### a. Metode ekstraktif

Metode ini memilih dan mengekstrak kalimat-kalimat penting dari teks asli tanpa mengubah struktur kalimatnya. Kalimat-kalimat yang diekstrak kemudian disusun untuk membentuk ringkasan. Dhimas Anjar Prabowo, Muhammad Fhadli, Mochammad Ainun Najib, Handika Agus Fauzi, dan Imam Cholissodin mengembangkan metode peringkas teks ekstraktif menggunakan TF-IDF yang ditingkatkan dengan algoritma

genetika. Penelitian ini menunjukkan peningkatan akurasi dalam menghasilkan ringkasan dibandingkan dengan metode TF-IDF konvensional[1]

#### b. Metode abstraktif

Pendekatan ini melibatkan pemahaman semantik teks dan menghasilkan kalimat-kalimat baru yang merangkum isi dokumen. Metode ini dapat menghasilkan ringkasan yang lebih koheren dan alami, tetapi lebih kompleks untuk diimplementasikan. Kasyfi Ivanedra dan Metty Mustikasari mengimplementasikan metode Recurrent Neural Network (RNN) untuk peringkasan teks dengan teknik abstraktif. Hasil penelitian menunjukkan bahwa pendekatan ini mampu menghasilkan ringkasan yang lebih alami dan koheren [2]

Peringkasan teks otomatis memiliki berbagai aplikasi praktis, termasuk membantu pengguna memahami inti dokumen dengan cepat, meningkatkan efisiensi dalam pencarian informasi, dan mendukung pengambilan keputusan berbasis data.

Penelitian-penelitian ini memberikan landasan penting untuk pengembangan aplikasi peringkasan berbasis kecerdasan buatan, seperti yang diimplementasikan dalam penelitian ini menggunakan API ChatGPT. Dengan memanfaatkan kemajuan dalam teknologi AI dan pembelajaran mesin, peringkasan teks terus berkembang menjadi alat yang semakin canggih dan berguna dalam mengelola informasi di era digital.

## 2.2. Flutter

*Flutter* adalah sebuah kerangka kerja (*framework*) sumber terbuka yang dikembangkan oleh Google untuk membangun aplikasi mobile multiplatform. Keunggulan utama *Flutter* terletak pada kemampuannya untuk mengembangkan aplikasi iOS dan Android dengan menggunakan satu basis kode yang sama. *Flutter* dirancang untuk menghasilkan aplikasi yang responsif dan memiliki performa tinggi, sehingga pengalaman pengguna terasa mulus dan alami di berbagai perangkat. Dengan menggunakan *Flutter*, para pengembang dapat menghemat waktu dan upaya dalam pengembangan, sekaligus menyajikan antarmuka pengguna yang konsisten di berbagai platform [3].

## 2.3. Dart

*Dart* merupakan bahasa pemrograman lintas platform yang dirancang oleh Google untuk memenuhi berbagai kebutuhan pengembangan aplikasi. *Dart* menganut paradigma pemrograman berorientasi objek (OOP), di mana struktur kode diorganisir dalam kelas yang berisi metode dan variabel. Sintaks *Dart*

menyerupai gaya bahasa C, sehingga memiliki kemiripan dengan bahasa seperti Java, JavaScript, dan Swift. Meskipun awalnya diperkenalkan untuk pengembangan aplikasi mobile, *Dart* kini dapat digunakan untuk mengembangkan berbagai jenis aplikasi, termasuk aplikasi Android, antarmuka web, aplikasi *Internet of Things* (IoT), aplikasi *back-end* (CLI), dan bahkan game. Dengan dukungan multiplatform yang kuat, *Dart* menawarkan fleksibilitas dan efisiensi bagi para pengembang dalam membangun solusi perangkat lunak yang beragam [4].

## 2.4. Aplikasi mobile

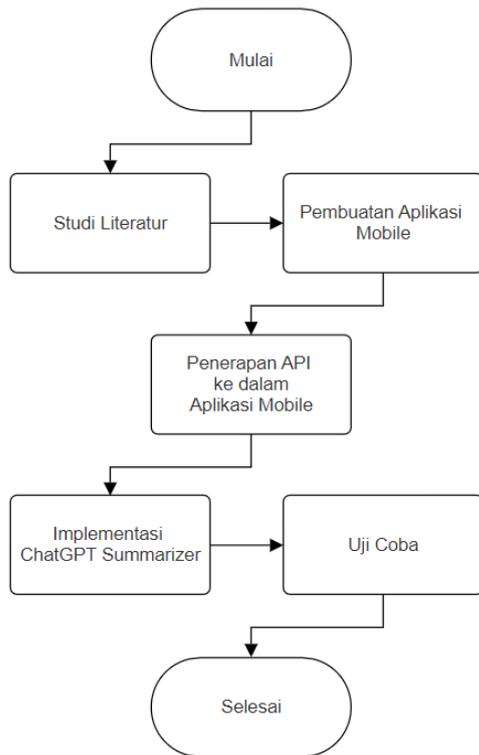
Aplikasi mobile adalah perangkat lunak yang dirancang untuk digunakan pada perangkat bergerak seperti ponsel pintar (*smartphone*) dan tablet. Keunggulan utama aplikasi seluler terletak pada portabilitasnya, yang memungkinkan pengguna untuk mengakses dan memanfaatkan aplikasi tersebut di mana pun dan kapan pun mereka berada. Tidak terikat pada lokasi tertentu, aplikasi mobile memberikan fleksibilitas bagi penggunanya untuk memperoleh informasi, melakukan tugas, atau mengakses layanan tertentu dengan mudah melalui perangkat yang selalu mereka bawa. Dengan kemampuan untuk diakses secara mobile, aplikasi ini menawarkan kenyamanan dan kemudahan dalam menunjang aktivitas sehari-hari di era digital saat ini [5].

## 2.5. API (Application Programming Interface)

API (*Application Programming Interface*) merupakan sebuah perantara yang memfasilitasi komunikasi dan pertukaran data antara dua atau lebih aplikasi perangkat lunak secara terstruktur. API mendefinisikan sekumpulan aturan, protokol, dan spesifikasi yang menjelaskan bagaimana sebuah aplikasi harus berinteraksi dengan aplikasi lain atau dengan sistem operasi secara aman dan efisien. Dengan adanya API, pengembangan dapat mengakses dan memanfaatkan fungsi, data, atau layanan dari aplikasi lain tanpa perlu memahami secara detail bagaimana semuanya bekerja di balik layar. API menyediakan antarmuka yang terdokumentasi dengan baik, sehingga memudahkan integrasi dan pengembangan aplikasi yang saling terhubung dalam suatu ekosistem perangkat lunak yang kompleks [6].

## 3. METODE PENELITIAN

Berdasarkan diagram alir yang disajikan pada Gambar 1, proses penelitian diawali dengan tahapan studi literatur. Pada tahap ini, peneliti melakukan pencarian dan analisis terhadap berbagai sumber literatur yang relevan untuk mengidentifikasi landasan teori serta referensi pendukung yang dapat digunakan sebagai dasar dalam melakukan penelitian yang sedang dikerjakan [7].



Gambar 1. Diagram alir

Tahap selanjutnya adalah pembuatan antarmuka mobile menggunakan *Flutter*. Pada tahap ini, peneliti merancang dan mengembangkan antarmuka pengguna yang intuitif dan ramah pengguna untuk aplikasi mobile peringkasan teks. Kemudian, peneliti melakukan implementasi API *ChatGPT* ke dalam aplikasi mobile. Tahap ini meliputi proses memperoleh kunci API *ChatGPT*, mempersiapkan program dengan bahasa pemrograman yang sesuai, dan mengintegrasikan kunci API ke dalam program untuk mengaktifkan fitur peringkasan teks otomatis.

Setelah tahap implementasi selesai, peneliti melakukan pengujian terhadap sistem yang telah dikembangkan. Pada tahap ini, dilakukan evaluasi terhadap kinerja dan keakuratan sistem dalam menghasilkan ringkasan teks. Setelah semua tahapan selesai, peneliti dapat menarik kesimpulan dari hasil penelitian dan memberikan saran untuk pengembangan lebih lanjut.

**3.1. Pembuatan Antar Muka Mobile**

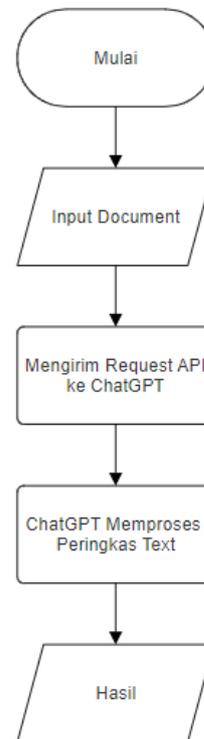
Aplikasi mobile ini dilengkapi dengan antarmuka pengguna yang dirancang secara intuitif dan ramah pengguna, dengan tujuan untuk memfasilitasi interaksi yang mudah dan nyaman. Pengguna dapat mengunggah file teks atau memasukkan teks secara langsung melalui editor teks yang disediakan dalam aplikasi. Selain itu, terdapat fitur pengaturan yang memungkinkan pengguna untuk memilih panjang ringkasan yang diinginkan, sehingga mereka dapat menyesuaikan output ringkasan sesuai dengan kebutuhan dan preferensi masing-masing.

**3.2. Penerapan API ChatGPT**

Dalam penelitian ini, kebutuhan dan tujuan spesifik dari implementasi *ChatGPT* berbasis mobile adalah untuk mengembangkan sebuah fitur peringkasan teks. Untuk mencapai tujuan tersebut, langkah pertama yang perlu dilakukan adalah membuat akun di *ChatOpenAI* untuk memperoleh kunci API *ChatGPT* [8].

Setelah mendapatkan kunci API, tahap selanjutnya adalah mempersiapkan program yang memadai dengan bahasa pemrograman pilihan. Dengan mengintegrasikan kunci API *ChatGPT* ke dalam program, fitur peringkasan teks pada aplikasi mobile dapat diimplementasikan secara efektif, memungkinkan pengguna untuk memasukkan teks panjang dan mendapatkan ringkasan yang padat namun tetap menyampaikan inti informasi.

**3.3. Implementasi**



Gambar 2. Implementasi ChatGPT

Dalam implementasi fitur peringkasan teks ini, API *ChatGPT* bertindak sebagai komponen inti yang menjalankan proses peringkasan teks secara otomatis. Aplikasi yang dikembangkan berfungsi sebagai antarmuka pengguna dan penghubung antara pengguna dengan API *ChatGPT* [9].

Dengan memanfaatkan kemampuan peringkasan teks yang dimiliki oleh API *ChatGPT*, pengguna dapat dengan mudah mendapatkan ringkasan dari dokumen panjang hanya dengan mengunggah atau memasukkan dokumen tersebut ke dalam aplikasi. Hal ini dapat menghemat waktu dan upaya pengguna dalam memahami inti informasi dari berbagai dokumen panjang.

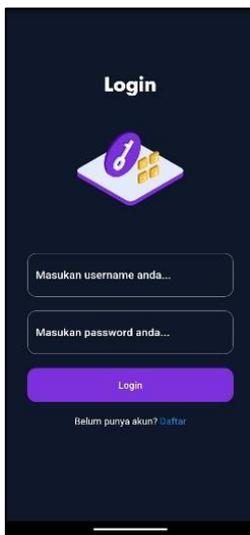
#### 4. HASIL DAN PEMBAHASAN

##### 4.1. Interface Aplikasi Mobile



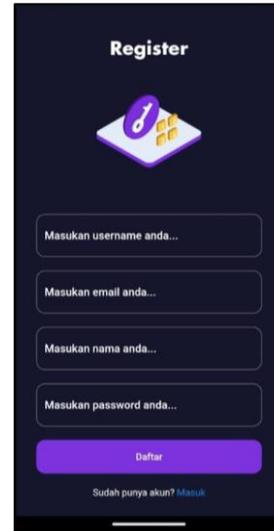
Gambar 3. Interface awal atau halaman pembuka

Gambar 3 menunjukkan antarmuka awal atau halaman pembuka saat pengguna membuka aplikasi mobile untuk peringkas teks PDF menggunakan *ChatGPT*. Antarmuka awal ini berfungsi untuk menyambut pengguna, memberikan gambaran umum tentang aplikasi, dan membangun ekspektasi tentang pengalaman yang akan diperoleh dalam menggunakan fitur peringkas teks berbasis *ChatGPT* ini.



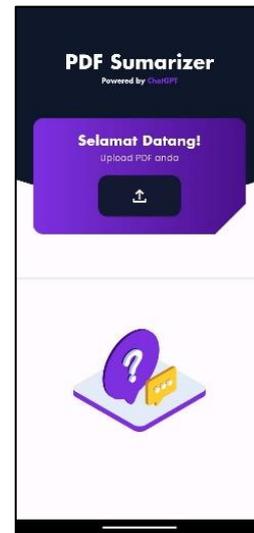
Gambar 4. Interface login

Gambar 4 menunjukkan halaman login aplikasi mobile untuk mengakses fitur peringkas teks PDF menggunakan *ChatGPT*. Pada halaman ini, pengguna diminta untuk memasukkan kredensial akun mereka sebelum dapat menggunakan aplikasi. Halaman login ini memberikan antarmuka yang sederhana dan intuitif bagi pengguna untuk memasukkan kredensial akun mereka sebelum dapat mengakses fitur utama aplikasi peringkas teks PDF.



Gambar 5. Interface register

Gambar 5 menunjukkan antarmuka registrasi dirancang untuk memungkinkan pengguna baru mendaftar dan membuat akun sebelum menggunakan fitur utama aplikasi. Halaman registrasi ini merupakan bagian penting dari aplikasi karena menyediakan pintu masuk bagi pengguna baru untuk mulai menggunakan aplikasi.



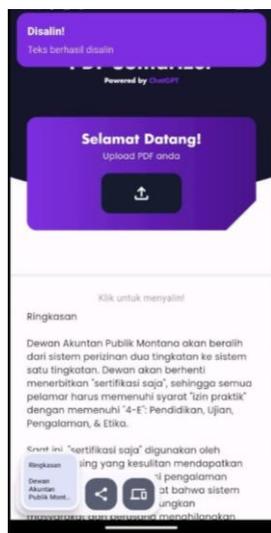
Gambar 6. Interface utama aplikasi

Gambar 6 menunjukkan antarmuka halaman utama dari sebuah aplikasi. Aplikasi ini memungkinkan pengguna untuk mengunggah file PDF dan menghasilkan ringkasan dari isinya dengan menggunakan teknologi *ChatGPT*. Antarmuka ini terlihat sederhana, intuitif, dan berfokus pada tujuan utama aplikasi yaitu meringkas file PDF dengan menggunakan kemampuan *ChatGPT*.



Gambar 7. Interface hasil ringkasan

Gambar 7 menampilkan antarmuka yang menunjukkan hasil ringkasan teks yang telah diproses oleh aplikasi. Pada antarmuka ini, pengguna dapat melihat ringkasan dari teks panjang yang mereka masukkan atau unggah sebelumnya. Teks hasil ringkasan ditampilkan secara jelas dengan tata letak yang rapi dan mudah dibaca, memungkinkan pengguna untuk dengan cepat memahami inti informasi dari dokumen yang diringkas.



Gambar 8. Interface menyalin ringkasan

Gambar 8 menunjukkan antarmuka yang memungkinkan pengguna untuk menyalin ringkasan teks yang telah dihasilkan oleh aplikasi. Pada antarmuka ini, terdapat tombol atau fitur yang memudahkan pengguna untuk menyalin teks ringkasan ke clipboard perangkat mereka. Dengan fitur ini, pengguna dapat dengan mudah menggunakan ringkasan tersebut di aplikasi lain atau untuk keperluan lain tanpa harus mengetik ulang.

## 4.2. Penerapan API ChatGPT pada Aplikasi Mobile

```
Future<String?> uploadFile(String filePath) async {
  updateLoading(true);
  String? authToken = await getToken();
  final Uri apiUrl =
    Uri.parse('https://binarytalkhub.pomob.cloud/api/sumarize/pdf');

  final file = File(filePath);
  if (!file.existsSync()) {
    print('File not found');
    return 'File not found';
  }

  var request = http.MultipartRequest('POST', apiUrl)
    ..headers.addAll({
      'Authorization': 'Bearer $authToken',
    })
    ..files.add(await http.MultipartFile.fromPath('pdfFile', filePath));

  try {
    final streamedResponse = await request.send();
    final responseString = await http.Response.fromStream(streamedResponse);
    final Map<String, dynamic> response = jsonDecode(responseString.body);

    if (responseString.statusCode == 200) {
      print(response['data']['message']);
      updateLoading(false);
      return response['data']['message'];
    } else {
      print(responseString.body);
      print(response['errors']);
      updateLoading(false);
      Get.snackbar('Upload Status', response['errors'],
        snackPosition: SnackPosition.TOP,
        colorText: Colors.white,
        backgroundColor: Colors(0xFF738E1));
    }
  } catch (e) {
    print('Error uploading file: $e');
    isLoading.value = false;
    return 'Error uploading file: $e';
  }
}
```

Gambar 9. Pengunggahan PDF dengan pengambilan ringkasan

Fungsi ini digunakan untuk mengunggah file PDF ke server menggunakan metode POST multipart. Fungsi ini mengambil token autentikasi, membuat request POST dengan file PDF, mengirim request ke server, dan menangani respons dari server. Jika berhasil, fungsi mengembalikan pesan sukses. Jika gagal, fungsi menampilkan pesan error.

## 4.3. Hasil Tabel Pengujian

Pengujian aplikasi "BinaryTalkHub" dilakukan untuk mengevaluasi kinerja sistem dalam menghasilkan ringkasan teks dari dokumen PDF. Tabel berikut menunjukkan hasil pengujian yang dilakukan:

Tabel 1. Ringkasan Hasil Pengujian Dokumen

No	Judul Dokumen	Panjang Dokumen (Kata)	Panjang Ringkasan (Kata)
1	Artikel Jurnal	3271	140
2	Naskah Publikasi Tugas Akhir	4241	131
3	Makalah	1819	111
4	Materi Pembelajaran	5332	173
5	Manual Book Aplikasi	4502	135

Hasil pengujian menunjukkan bahwa aplikasi "BinaryTalkHub" mampu menghasilkan ringkasan teks yang padat dan informatif dari berbagai jenis dokumen panjang. Keberhasilan aplikasi ini terletak pada penggunaan API ChatGPT yang dapat mengekstraksi informasi penting dari dokumen dan menyusunnya dalam format yang lebih ringkas.

Efisiensi sistem ini memungkinkan pengguna untuk mendapatkan intisari informasi dengan cepat, yang sangat berguna dalam konteks akademik, pendidikan, dan teknis.

## 5. KESIMPULAN DAN SARAN

Penelitian ini mengimplementasikan kecerdasan buatan atau *Artificial Intelligence* (AI) dalam pengembangan aplikasi mobile "BinaryTalkHub" untuk peringkasan teks PDF menggunakan API *ChatGPT*. Hasil pengujian menunjukkan bahwa sistem mampu memberikan ringkasan teks yang akurat dan padat, memungkinkan pengguna untuk dengan cepat mendapatkan inti informasi dari dokumen panjang. Antarmuka yang intuitif dan ramah pengguna serta integrasi dengan API *ChatGPT* menjadi poin kunci dalam kesuksesan aplikasi ini. Namun, aplikasi ini memiliki keterbatasan dalam mengolah file PDF dengan ukuran lebih dari 10 MB, yang menyebabkan error dan tidak dapat diproses oleh sistem.

Untuk pengembangan lebih lanjut, perlu dipertimbangkan penambahan fitur personalisasi yang memungkinkan pengguna menyesuaikan preferensi ringkasan teks sesuai kebutuhan masing-masing, serta fitur pengeditan langsung yang memudahkan pengguna dalam membuat penyesuaian cepat. Selain itu, peningkatan aspek keamanan penting untuk melindungi data pengguna dan dokumen yang diunggah dari potensi ancaman dan kebocoran data. Pengembangan aplikasi untuk platform selain mobile, seperti desktop atau web, dapat memperluas jangkauan pengguna dan memberikan fleksibilitas dalam akses. Pengoptimalan performa juga diperlukan untuk meningkatkan kecepatan dan efisiensi sistem dalam memproses dokumen dan menghasilkan ringkasan.

Penelitian ini menunjukkan potensi besar integrasi AI dalam aplikasi mobile untuk meningkatkan fungsionalitas dan pengalaman pengguna. Dengan berbagai pengembangan lebih lanjut yang disebutkan di atas, aplikasi "BinaryTalkHub" dapat menjadi alat yang lebih efektif dan serbaguna dalam membantu pengguna mengelola informasi dari dokumen panjang.

## DAFTAR PUSTAKA

- [1] Dhimas Anjar Prabowo, Muhammad Fhadli, Mochammad Ainun Najib, Handika Agus Fauzi, and Imam Cholissodin, "TF-IDF-Enhanced Genetic Algorithm Untuk Extractive Automatic Text Summarization," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, vol. 3, no. 3, pp. 208–215, Sep. 2016, Accessed: Jun. 07, 2024. [Online]. Available: <https://jtiik.ub.ac.id/index.php/jtiik/article/view/217>
- [2] Kasyfi Ivanedra and Metty Mustikasari, "Implementasi Metode Recurrent Neural Network pada Text Summarization dengan Teknik Abstraktif," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, vol. 6, no. 4, pp. 377–382, Jul. 2019, doi: 10.25126/jtiik.201961067.
- [3] Gerry Surya Chandra and Suhatati Tjandra, "Pemanfaatan Flutter dan Electron Framework pada Aplikasi Inventori dan Pengaturan Pengiriman Barang," *Journal of Information System, Graphics, Hospitality and Technology*, vol. 2, no. 2, pp. 76–81, Dec. 2020, doi: <https://doi.org/10.37823/insight.v2i02.109>.
- [4] Taryana Suryana, "Belajar Bahasa Pemrograman Dart," *UNIKOM Repository*, May 2021, Accessed: Jun. 07, 2024. [Online]. Available: <https://repository.unikom.ac.id/68459/>
- [5] Muhammad Iqbal Maliki, Suaidah, and Parjito, "Rancang Bangun Aplikasi Penjualan Grosir Sembako Pada Toko LA-RIS," *Jurnal Informatika dan Rekayasa Perangkat Lunak (JATIKA)*, vol. 2, no. 3, pp. 304–311, 2021, doi: <https://doi.org/10.33365/jatika.v2i3.1222>.
- [6] Grinaldi Wisnu Tri Prasetyo, Fajar Pradana, and Bondan Sapta Prakoso, "Pengembangan Aplikasi Point of Sales Warung dan UMKM 'WarunkQu' menggunakan Framework Flutter," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, vol. 6, no. 10, pp. 2548–964, Sep. 2022, Accessed: Jun. 07, 2024. [Online]. Available: <https://jtiik.ub.ac.id/index.php/jtiik/article/view/11665>
- [7] Nur Hidayati, "Penggunaan Metode Waterfall Dalam Rancang Bangun Sistem Informasi Penjualan," *Generation Journal*, vol. 3, no. 1, pp. 2580–4952, Jan. 2019, doi: <https://doi.org/10.29407/gj.v3i1.12642>.
- [8] Farida Gultom, Wita Clarisa Ginting, and M. Riswan, "Perancangan Web Dengan Chat GPT Untuk Meningkatkan Kualitas Akademik (Studi Kasus : Universitas Efarina)," *Innovative: Journal Of Social Science Research*, vol. 4, no. 1, pp. 5589–5600, Jan. 2024, doi: <https://doi.org/10.31004/innovative.v4i1.8507>.
- [9] Zheheng Luo, Qianqian Xie, and Sophia Ananiadou, "ChatGPT as a Factual Inconsistency Evaluator for Text Summarization," *Cornell University*, Apr. 2023, doi: <https://doi.org/10.48550/arXiv.2303.15621>.
- [10] Farah Raihanunnisa, Muhammad Arhami, and Rahmad Hidayat, "PENDEKATAN HYBRID PADA SISTEM PERINGKAS TEKS ARTIKEL BERITA BAHASA INGGRIS MENGGUNAKAN NATURAL LANGUAGE PROCESSING," *TELEMATIKA MKOM*, vol. 15, no. 2, pp. 86–92, 2023, Accessed: Jun. 07, 2024. [Online]. Available: <https://journal.budiluhur.ac.id/index.php/telematika/article/view/2679>
- [11] Muhammad Adib zamzam, Cahyo Crysdian, and Khadijah Fahmi Hayati Holle, "Sistem

Automatic Text Summarization Menggunakan Algoritma Textrank,” *MATICS: Journal of Computer Science and Information Technology*, vol. 12, no. 2, pp. 111–116, Sep. 2020, doi: <https://doi.org/10.18860/mat.v12i2.8372>.

- [12] Bagus Fatwan Alfiat, Puspa Eosina, and Safaruddin Hidayat Al Ikhsan, “Perancangan Aplikasi Chatbot Menggunakan Wit.Ai pada Sistem SPP-IRT Berbasis Web,” *Jurnal Informatika Universitas Pamulang*, vol. 6, no. 4, pp. 2622–4615, Feb. 2021, Accessed: Jun. 07, 2024. [Online]. Available: <https://openjournal.unpam.ac.id/index.php/informatika/article/view/13327>