

## Pengembangan Aplikasi Tanyapdf AI (Studi Kasus Institutional Repository Universitas Muhammadiyah Surakarta)

Muhammad Irfan Zahran<sup>1✉</sup>, Widi Widayat<sup>2</sup>

<sup>1,2</sup>Universitas Muhammadiyah Surakarta, Surakarta, Indonesia

✉ Email korespondensi: [fanzahran123@gmail.com](mailto:fanzahran123@gmail.com)

**Abstrak.** *Institutional repository* Universitas Muhammadiyah Surakarta merupakan platform untuk menampilkan berbagai naskah publikasi dari dokumen penelitian mahasiswa. Mayoritas mahasiswa yang sedang atau akan mengambil mata kuliah skripsi mengakses website tersebut untuk mencari referensi. Namun proses membaca dokumen penelitian pada website tersebut sering kali tidak efisien, karena mahasiswa yang hanya ingin mencari satu informasi spesifik dari sebuah dokumen harus membaca keseluruhan isi dokumen terlebih dahulu. Terlebih lagi, dalam proses mencari referensi skripsi, tentunya terdapat banyak dokumen penelitian yang perlu dibaca. Oleh karenanya perlu adanya sistem pembaca dokumen Pdf yang mempermudah pembaca dalam menggali informasi dari dokumen penelitian. Metode pengembangan yang digunakan pada penelitian ini adalah metode waterfall. Metode ini terdiri dari analisis kebutuhan, desain sistem, implementasi, pengujian, dan pemeliharaan. Sistem ini dikembangkan menggunakan teknologi Next.js, Supabase, LangChain, AI SDK, serta menerapkan teknik Retrieval Augmented Generation (RAG) untuk memproses data. Hasil penelitian ini berupa aplikasi website yang terintegrasi dengan AI untuk membaca dan menganalisa dokumen penelitian. Hasil pengujian sistem menggunakan Black Box menunjukkan semua fungsi dan fitur dapat berjalan dengan baik, dan hasil perolehan skor System Usability Scale (SUS) memperoleh skor 71 yang artinya mendapatkan kategori good sehingga sistem layak digunakan.

**Kata kunci:** kecerdasan buatan; aplikasi website; dokumen penelitian; skripsi



## PENDAHULUAN

Dalam lingkungan akademik, universitas sering kali menyediakan platform digital berupa website untuk menampilkan berbagai naskah publikasi dari dokumen penelitian mahasiswa, salah satunya adalah dokumen skripsi. Skripsi merupakan salah satu syarat lulus bagi mahasiswa pada tingkat sarjana dengan jumlah sks (sistem kredit semester) yang tertinggi dibandingkan dengan tugas lainnya, berkisar antara 5 sampai 6 sks informasi adalah penggabungan antara manusia, hardware, dan software yang terintegrasi untuk menyimpan, mengatur, dan menghasilkan informasi yang berguna [1]. Salah satu contohnya adalah sistem informasi website *instutional repository* yang dimiliki oleh Universitas Muhammadiyah Surakarta. *Institutional repository* merupakan serangkaian layanan yang dikembangkan oleh suatu institusi berupa pengelolaan serta penyebarluasan berbagai hasil kegiatan ilmiah sivitas akademika dalam bentuk digital material [2]. Website ini menjadi sumber informasi penting bagi mahasiswa dan peneliti yang ingin membaca dan mempelajari dokumen penelitian.

Namun, membaca dan memahami isi dari dokumen penelitian yang panjang memerlukan waktu yang lama dan sering kali tidak efisien. Proses ini menjadi tantangan, ketika pengguna hanya ingin memperoleh informasi spesifik dengan cepat namun harus membaca seluruh isi dokumen terlebih dahulu. Terlebih lagi mayoritas dokumen penelitian tersebut menggunakan bahasa yang kompleks dan pembahasan yang menyeluruh, sehingga membuat pencarian informasi spesifik menjadi tugas yang melelahkan. Dalam upaya untuk meningkatkan efisiensi akses informasi, teknologi AI dapat digunakan untuk mengembangkan sistem yang mampu menganalisis dokumen penelitian tersebut secara cepat dan akurat serta memberikan informasi spesifik yang dibutuhkan oleh pembaca. Selain itu, AI dapat digunakan untuk mengembangkan mesin pencari cerdas yang dapat dengan cepat dan akurat mengambil informasi relevan dari basis data besar. Teknologi AI, seperti pemrosesan bahasa alami dan visi mesin, memungkinkan perpustakaan mengotomatisasi proses pengatalogan, digitalisasi dokumen, serta mengatur informasi secara efisien [3].

*Language models* (LM) adalah model statistik yang menetapkan probabilitas dalam pemrosesan bahasa alami [4]. *Language Models* yang lebih kompleks dan dilatih dengan data yang berukuran sangat besar disebut dengan *Large Language models* (LLM) [5]. Jumlah data yang besar tersebut mampu meningkatkan kemampuan LM untuk menghasilkan bahasa alami yang fasih.

Salah satu model dari LLM yang cukup populer adalah generative pre-trained transformer (GPT). GPT adalah model bahasa yang telah dilatih dengan data tekstual yang berukuran sangat besar dan dapat melakukan berbagai tugas terkait Bahasa [6].



GPT memiliki kemampuan untuk menghasilkan teks dan konten seperti manusia (gambar, dokumen, dan lainnya) serta mampu menjawab berbagai pertanyaan.

Pada tanggal 18 Juli 2024, Open AI yang merupakan perusahaan penelitian kecerdasan buatan telah merilis model LLM GPT-4o mini, model yang paling hemat biaya. Namun mampu mengerjakan berbagai tugas seperti membaca konteks pertanyaan, riwayat percakapan lengkap, berinteraksi dengan pelanggan secara *realtime*, hingga membaca serta menganalisa data tekstual [7].

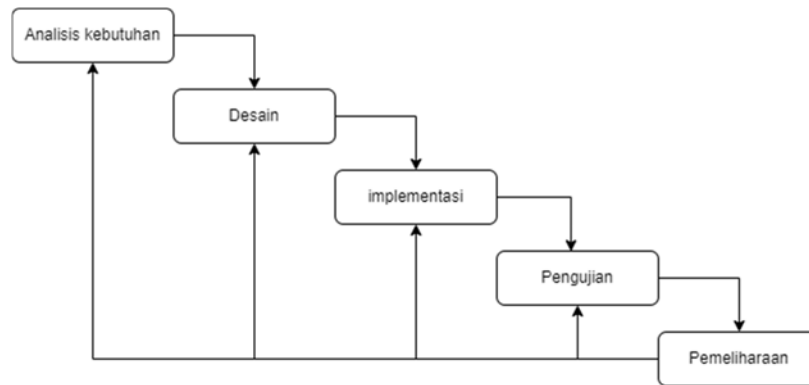
Penelitian ini bertujuan untuk mengembangkan sistem TanyaPDF, sebuah platform website yang memanfaatkan teknologi kecerdasan buatan untuk membantu menyelesaikan permasalahan di atas, yakni membantu pembaca memperoleh informasi spesifik dari dokumen penelitian secara efisien dengan bantuan fitur *chat* interaktif. Sistem TanyaPDF dikembangkan menggunakan teknologi Open AI API dengan model AI GPT-4o mini, Langchain, AI SDK, Next.js, dan supabase serta menggunakan teknik Retrieval Augmented Generation (RAG), yang dikenal untuk meningkatkan akurasi dalam *language models*, sangat efektif dalam kasus-kasus di mana terdapat data yang relevan secara kontekstual [8].

Dengan mengembangkan TanyaPDF, diharapkan dapat memberikan satu kontribusi positif dalam meningkatkan efisiensi dan produktivitas pembaca dalam lingkungan akademik, serta memaksimalkan pemanfaatan dokumen penelitian yang tersedia di *institutional repository* Universitas Muhammadiyah Surakarta.

## METODE

Pengembangan sistem ini menggunakan metode *waterfall*. *Waterfall* merupakan pendekatan yang diterapkan untuk mengembangkan sistem secara terstruktur dan bertahap. Penggunaan metode *waterfall* mengikuti pendekatan linier, dimana pengembangan dilakukan dalam tahapan-tahapan yang jelas dan saling terkait [9]. Model *waterfall* lebih sesuai diterapkan untuk sistem atau perangkat lunak yang bersifat generik, yang berarti semua kebutuhan sistem dapat ditentukan sejak awal dengan spesifikasi yang umum [10]. Proyek dipecah menjadi beberapa tugas, dengan tahap-tahap sebagai pengelompokan tingkat tertinggi. Metode *waterfall* yang tepat membutuhkan tahap-tahap yang harus diselesaikan secara berurutan dan memiliki kriteria penyelesaian yang jelas [11]. Tahapan model *waterfall* antara lain analisis kebutuhan, desain antarmuka, implementasi, pengujian, dan pemeliharaan. Setiap tahapan akan dijelaskan secara menyeluruh, bersama dengan berbagai hal yang digunakan. Gambaran proses metode *waterfall* ditampilkan pada gambar 1.





**Gambar 1.** Diagram tahap-tahap dalam metode *waterfall*

## 2.1 Analisis Kebutuhan

Analisis kebutuhan sistem merupakan langkah pertama yang penting dalam merancang sistem untuk menentukan tipe dan cakupan, serta memastikan bahwa seluruh tuntutan pengguna dapat terpenuhi secara optimal dan efektif [12]. Peneliti melakukan proses analisis kebutuhan dengan mewawancarai beberapa mahasiswa yang menggunakan *institutional repository* Universitas Muhammadiyah Surakarta. Kemudian peneliti mengidentifikasi permasalahan dihadapi oleh pengguna, salah satu permasalahan yang ditemukan adalah ketika mahasiswa ingin mencari referensi skripsi, mereka perlu membaca banyak dokumen penelitian dan harus membaca keseluruhan isi dokumen tersebut untuk menggali satu informasi spesifik yang kemudian bisa dijadikan referensi pada penelitiannya. Hal tersebut tentu saja memakan waktu yang tidak sebentar serta memerlukan tenaga, fokus, dan pikiran untuk mampu membaca serta menganalisa setiap dokumen penelitian. Hasil identifikasi ini kemudian dirumuskan menjadi kebutuhan fungsional dan kebutuhan non-fungsional. Kebutuhan fungsional merupakan pernyataan layanan yang wajib diberikan kepada sistem agar mampu memberikan reaksi yang tepat terhadap masukan tertentu serta dalam situasi tertentu. Untuk kebutuhan non-fungsional lebih mengarah kepada batasan layanan atau fungsi yang diberikan sistem [13]. Kebutuhan fungsional pada sistem TanyaPDF seperti fitur utama yang harus dimiliki sistem, sistem harus mampu menghemat waktu pengguna dalam membaca dan menganalisa dokumen penelitian memudahkan pengguna, mampu membuat ringkasan isi dokumen, serta mampu memberikan *insight* atau jawaban spesifik terkait pertanyaan pengguna tentang isi dokumen tersebut. Sedangkan kebutuhan non-fungsionalnya mencakup performa yakni mampu memberikan respon jawaban dengan cepat, desain antarmuka yang responsif, dan memiliki navigasi yang sederhana dan tata letak yang



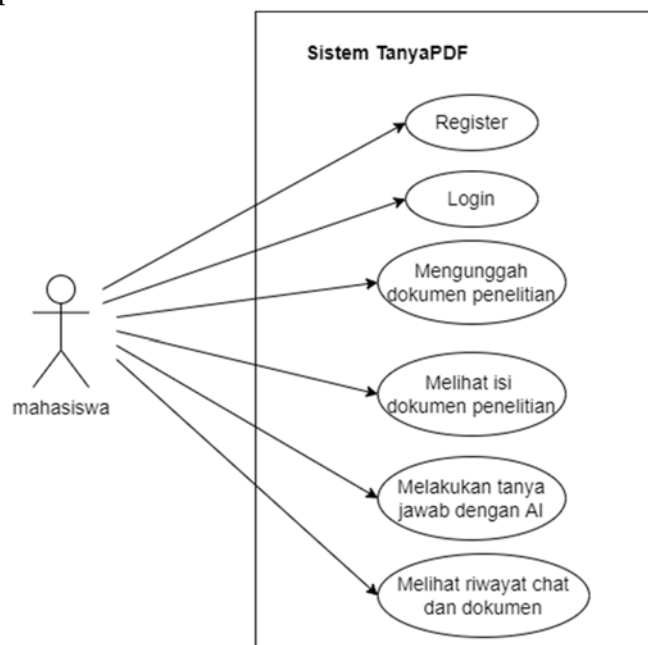
jelas untuk memudahkan pengguna. Seluruh kebutuhan tersebut menjadi dasar untuk pengembangan sistem TanyaPDF.

## 2.2 Desain

Proses tahap ini bertujuan agar kebutuhan yang telah didefinisikan sebelumnya diproses menjadi representasi dengan bentuk “blueprint” perangkat lunak sebelum memasuki proses pengkodean [14]. Pada tahap ini, peneliti menggambarkan alur kerja sistem TanyaPDF secara detail untuk memastikan setiap proses sesuai dengan kebutuhan yang telah diidentifikasi. Penggambaran ini mencakup beberapa elemen penting, seperti:

### a. Use Case Diagram

Peneliti menggunakan *use case diagram* untuk menentukan interaksi pengguna dengan sistem secara umum. *Use case diagram* adalah metodologi yang digunakan dalam analisis sistem untuk mengidentifikasi, mengklarifikasi, dan mengorganisir persyaratan sistem. *Use case diagram* digunakan dalam UML (Unified Modeling Language), sebuah notasi standar untuk pemodelan objek dan sistem dunia nyata [15]. Pada Gambar 2 terlihat bahwa sistem memiliki 1 role user yang dapat mengakses berbagai fitur, yakni melakukan autentikasi (register & login), mengunggah dokumen penelitian, melihat *preview* isi dokumen penelitian serta melakukan tanya jawab dengan AI *assistant* untuk menggali *insight* terkait informasi spesifik yang terdapat di dalam dokumen penelitian.

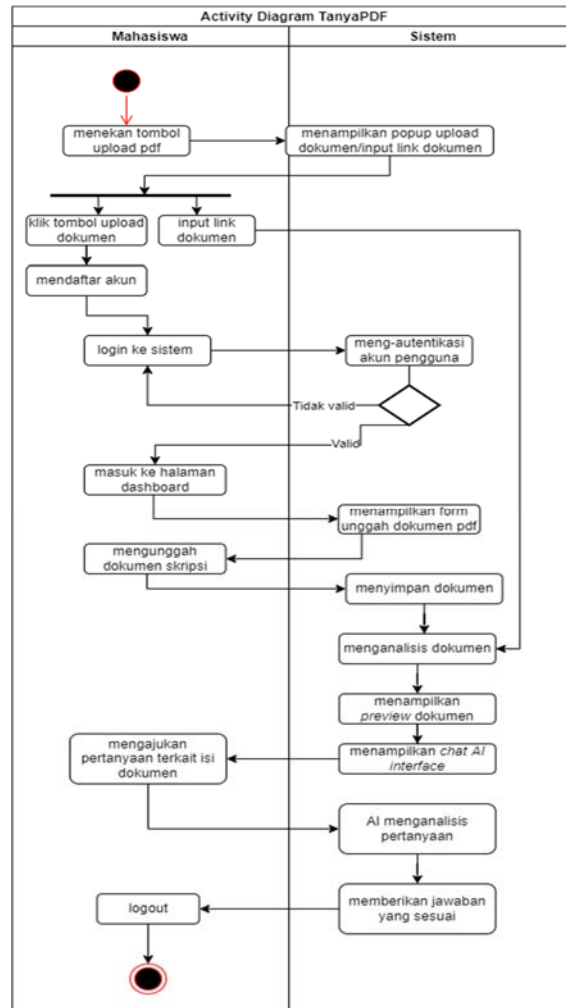


Gambar 2. Use Case Diagram



b. *Activity Diagram*

*Activity diagram* merupakan salah satu yang paling banyak digunakan di antara semua diagram UML. Alur *activity diagram* dapat digunakan untuk mengontrol urutan tindakan yang dieksekusi atau untuk merepresentasikan komunikasi data [16]. Pada gambar 3 terlihat bahwa aktivitas pengguna (mahasiswa) dimulai dengan menekan tombol upload pdf pada *landing page* kemudian ada 2 opsi, pengguna dapat mengunggah dokumen atau memasukkan *link* dokumen, jika memasukkan *link* dokumen maka pengguna akan langsung diarahkan ke halaman *chat* dan dapat langsung berinteraksi dengan *AI assistant* untuk tanya jawab terkait isi dokumen tersebut. Jika pengguna ingin mengunggah dokumen maka pengguna perlu mendaftar akun dan masuk ke akun tersebut, kemudian sistem akan menyimpan dan menganalisa dokumen, sehingga pengguna dapat bertanya kepada *AI assistant* terkait isi dokumen tersebut.

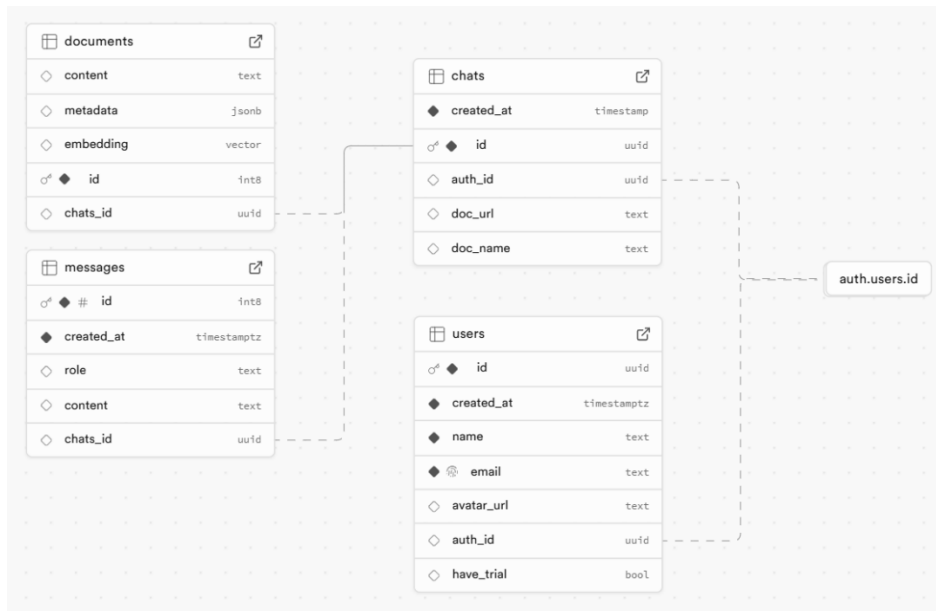


Gambar 3. *Activity Diagram*



### c. Entity Relationship Diagram

Entity Relationship Diagram (ERD) merupakan yang memiliki bentuk simbol visual dalam proses pembuatan basis data, dapat menghubungkan antara data satu dengan yang lain, dapat mengaitkan data yang saling berhubungan satu sama lain [17]. Pada ERD di bawah ini, terdapat 5 entitas, yakni *documents*, *messages*, *chats*, *auth\_users*, dan *users*. Tabel *documents* dan *messages* memiliki relasi *one-to-many* dengan tabel *chats*. Tabel *auth\_users* memiliki relasi *one-to-one* dengan tabel *users*. Detail relasi antar tabel dapat dilihat pada gambar 4.



Gambar 4. Entity Relationship Diagram

## 2.3 Implementasi Sistem

Tahap implementasi sistem ini meliputi pembuatan *database*, proses *coding*, dan implementasi antarmuka aplikasi [18]. Pada tahap ini peneliti mengimplementasikan hasil desain ke dalam kode program. Proses implementasi sistem ini menggunakan *framework* Next.js sebagai *frontend*, *tailwind.css* untuk desain antar muka, Open AI API, serta *supabase* sebagai *backend*.

Application programming interface (API) adalah mekanisme yang mampu menghubungkan antar sistem untuk dapat saling bertukar data. Open AI memiliki API untuk berbagai model GPT yang mereka miliki dan dapat dimanfaatkan oleh para *software developer* untuk mengembangkan sistem yang ditenagai oleh model GPT tersebut [19].

LangChain adalah sebuah *framework opensource* untuk mengembangkan aplikasi berdasarkan LLM. Salah satu pemanfaatan Langchain adalah mengembangkan aplikasi *chatbot* yang memungkinkan pengguna dapat berkomunikasi dengan AI model.

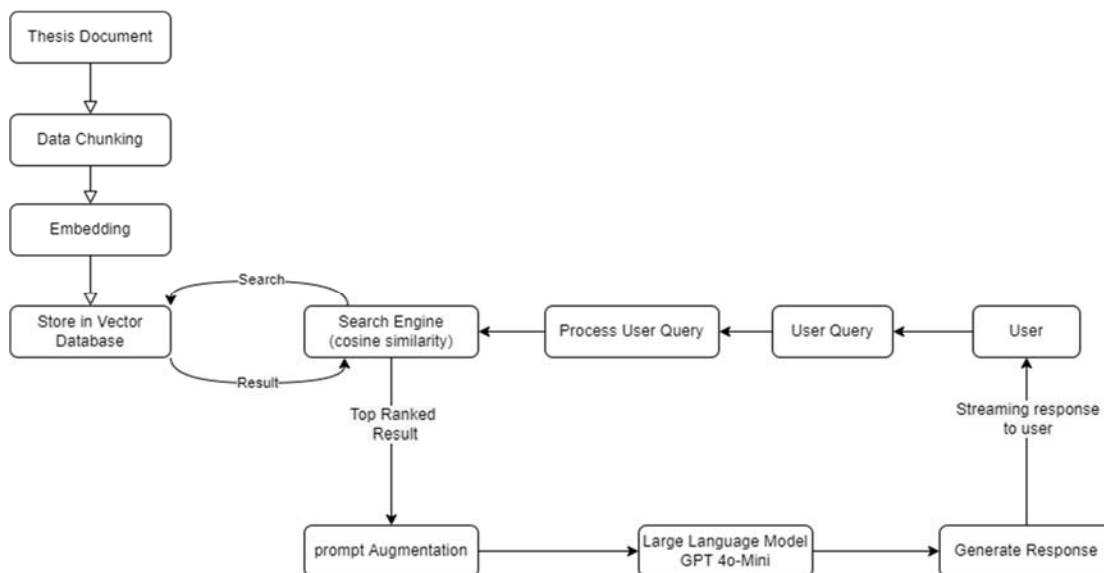


Pengembangan sistem tersebut menggunakan teknik *Retrieval Augmented Generation*(RAG) yakni teknik untuk menambah pengetahuan LLM dengan data tambahan.

AI SDK adalah perangkat untuk membantu pengembang membangun aplikasi bertenaga AI. Next.js adalah kerangka kerja React.js yang dapat digunakan untuk mengembangkan aplikasi website.

Supabase adalah salah satu penyedia layanan database dengan sumber kode terbuka yang menggunakan postgres database dan dapat digunakan untuk mengembangkan aplikasi yang terintegrasi dengan Open AI.

Penelitian ini menggunakan teknik RAG (*Retrieval Augmented Generation*) untuk mengelola dan memproses data. RAG menggabungkan LLM dengan sumber data eksternal. Pendekatan RAG adalah metode yang digunakan dalam pemrosesan bahasa alami dan mesin pembelajaran yang menggabungkan kekuatan *retrieval* dan *generative models* untuk meningkatkan kualitas teks yang dihasilkan [20]. Dibalik layar RAG ini menggunakan teknik similarity search yang mengidentifikasi dan mengambil data dalam *database* yang mirip dengan kueri tertentu, menggunakan ukuran metrik untuk mengevaluasi kedekatan antara elemen-elemen tersebut [21]. Pendekatan ini menggabungkan kemampuan generatif dari LLM seperti GPT-4o mini dengan ketepatan mekanisme pencarian data, sehingga menghasilkan sistem yang dapat menjawab berbagai pertanyaan terkait informasi spesifik dari sebuah dokumen. Sebagai gambaran implementasi teknik tersebut untuk pengembangan sistem TanyaPDF dapat dilihat pada gambar 5.



**Gambar 5.** Diagram alur proses RAG



Tahap-tahapan yang dilalui dari proses tersebut adalah:

1. Pengguna dapat mengunggah dokumen skripsi ke dalam sistem TanyaPDF. Kemudian sistem akan mengekstrak data teks dari dokumen tersebut.
2. Data teks tersebut dipecah ke dalam bagian yang lebih kecil (*chunk*) agar mempermudah dalam memberikan referensi bagian dokumen yang relevan.
3. Setiap bagian dari data teks tersebut kemudian di embed, yakni proses mentransform data teks menjadi data vektor (multidimensional array). Penulis menggunakan model teks embedding dari OpenAI pada proses ini.
4. Semua data multidimensional array tersebut kemudian disimpan ke dalam database vektor. Peneliti menggunakan supabase sebagai database untuk menyimpan data tersebut.
5. Kemudian ketika pengguna melakukan *query* (mengajukan pertanyaan kepada AI), pertanyaan tersebut langsung diproses, sistem mengambil data vektor dari database dan mencari data yang sesuai dengan pertanyaan pengguna. Peneliti menggunakan teknologi Langchain.js dalam proses ini karena terdapat *function* bawaan yang mempermudah implementasi proses ini seperti *similaritySearch()* untuk menghitung seberapa relevan pertanyaan dengan data di database, dan *SupabaseVectorStore* untuk mengelola data vektor di database.
6. Setelah ditemukan data yang paling relevan dengan pertanyaan pengguna, data tersebut kemudian dijadikan sebagai referensi, konteks tambahan bagi model LLM (peneliti menggunakan model LLM GPT-4o-mini). Sehingga mampu memberikan jawaban yang sesuai dengan pertanyaan pengguna.

## 2.4 Pengujian Sistem

Tahapan selanjutnya adalah pengujian sistem yang telah dikembangkan. Pengujian ini bertujuan untuk mengukur kesesuaian dan kinerja sistem dengan kebutuhan yang ada. Berdasarkan hasil pengujian ini, dapat dilakukan evaluasi untuk meningkatkan kualitas sistem. Pengujian ini menggunakan metode black box. Metode pengujian ini merupakan pengujian yang menilai hasil penggunaan aplikasi berdasarkan input yang diterima (data pengujian) untuk memastikan fungsi aplikasi telah memenuhi kriteria yang ditetapkan (requirement). Fokus dari pengujian ini adalah pada antarmuka dan pengujian fungsional yang terdapat pada aplikasi, beserta kecocokan dengan rangkaian fungsi yang diperlukan oleh pengguna [22]. Selanjutnya, penulis menggunakan metode System Usability Scale (SUS). Metode ini adalah salah satu metode uji pengguna yang menitikberatkan pada pembuatan sistem yang mudah dipelajari dan digunakan [23]. Penulis melakukan penyebaran kuesioner melalui tautan google *form* kepada pengguna untuk memberikan penilaian kelayakan



sistem TanyaPDF. Sehingga, dengan 2 metode pengujian tersebut dapat disimpulkan tingkat kelayakan dari sistem TanyaPDF.

## 2.5 Pemeliharaan Sistem

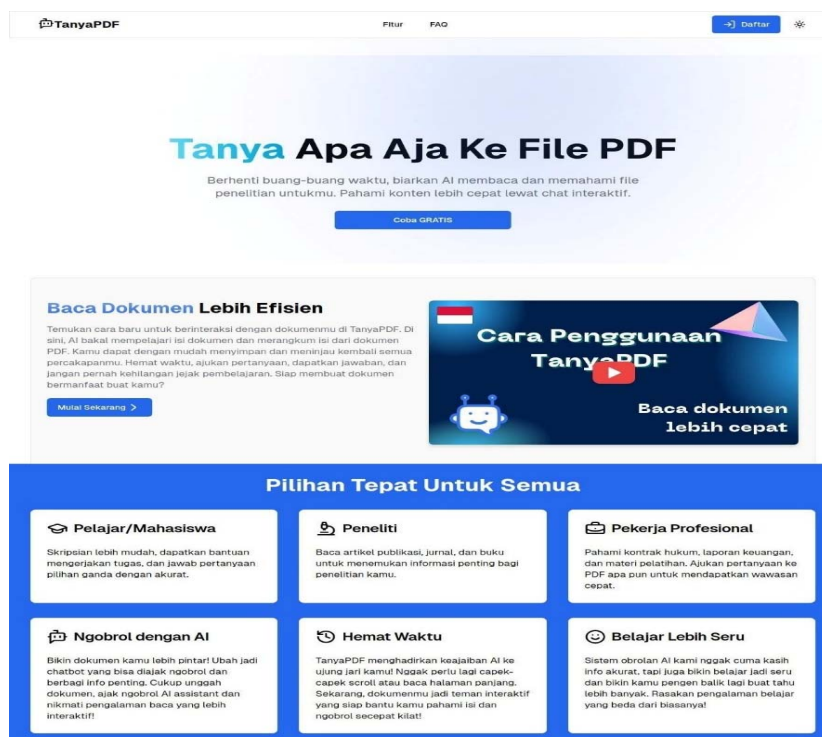
Setelah dilakukannya sejumlah pengujian program, tahap selanjutnya adalah implementasi dan pemeliharaan sistem TanyaPDF dan diharapkan mampu membantu mahasiswa menjadi lebih efisien dalam membaca dokumen penelitian di institutional repository Universitas Muhammadiyah Surakarta. Pemeliharaan sistem juga dapat dilakukan untuk meningkatkan dan menyesuaikan sistem jika diperlukan.

## HASIL

Hasil dari penelitian ini berupa sistem TanyaPDF yang membantu mahasiswa membaca dan mencari informasi spesifik dari sebuah dokumen penelitian di *institutional repository* Universitas Muhammadiyah Surakarta dengan lebih efisien. Hasil ini merupakan capaian dari perancangan sistem yang telah dilakukan sebelumnya.

### 3.1 Halaman Landing Page

Halaman landing page ini dapat dilihat pada gambar 6, ini adalah halaman yang berisi gambaran singkat mengenai sistem TanyaPDF, penjelasan fitur, hingga cara penggunaannya.

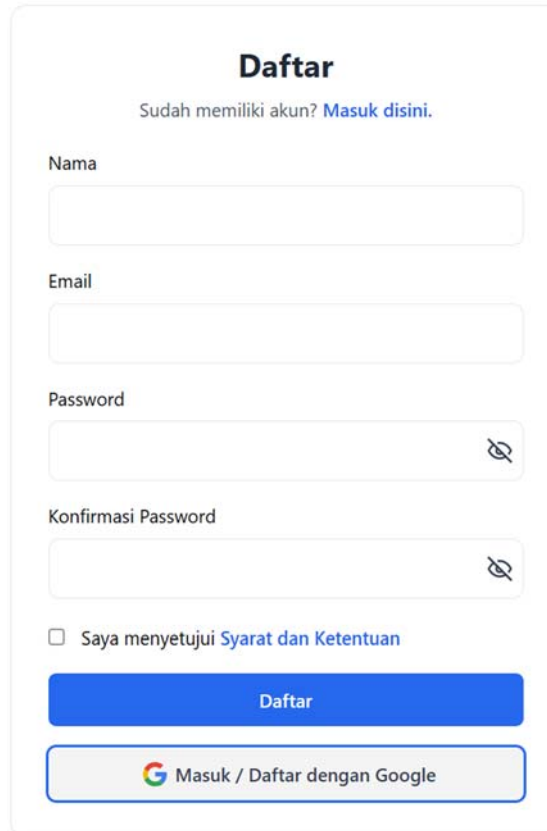


Gambar 6. Landing page



### 3.2 Halaman Daftar

Pada halaman ini pengguna dapat mendaftar akun menggunakan *email & password* atau menggunakan google. Apabila pengguna memilih untuk mendaftar menggunakan *email & password* maka sistem akan otomatis mengirim tautan ke *email* pengguna untuk memverifikasi email tersebut.



**Daftar**

Sudah memiliki akun? [Masuk disini.](#)

Nama


Email

Password

Konfirmasi Password

Saya menyetujui [Syarat dan Ketentuan](#)

**Daftar**

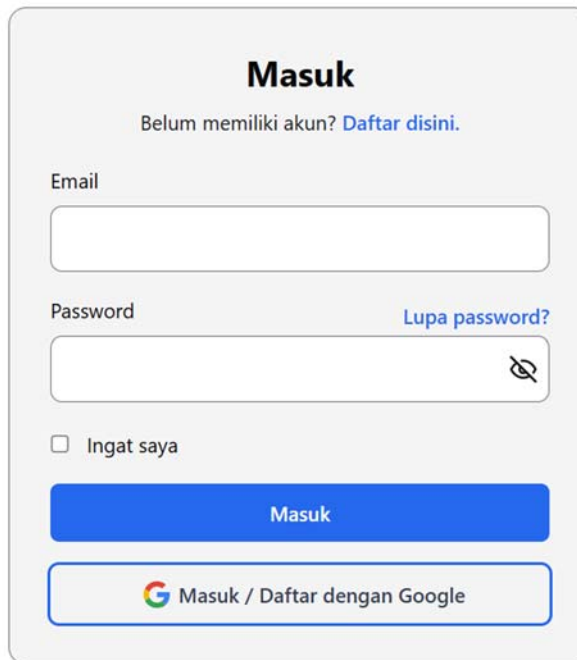
 [Masuk / Daftar dengan Google](#)

**Gambar 7.** Halaman daftar

### 3.3 Halaman Masuk

Halaman ini digunakan untuk melakukan autentikasi akun pengguna. Pengguna dapat masuk menggunakan *email & password* maupun akun google. Pengguna juga dapat menyimpan data akunnya dengan fitur "ingat saya". Apabila pengguna lupa dengan kata sandi yang dimilikinya, dapat menekan tombol "lupa password".





Gambar 8. Halaman masok

### 3.4 Halaman Chat

Pada halaman ini, pengguna dapat mengunggah dokumen penelitian yang ingin dianalisa. Kemudian sistem TanyaPDF akan memproses, menganalisa, serta menyimpan dokumen tersebut ke dalam *database*. Selanjutnya pengguna dapat melakukan tanya jawab dengan asisten AI untuk mencari informasi tertentu dari sebuah dokumen penelitian.



Gambar 9. Halaman Chat



### 3.5 Pengujian *Black Box*

Pengujian sistem TanyaPDF menggunakan metode *black box*. Metode ini digunakan untuk memverifikasi setiap fitur berfungsi dengan baik. Hasil pengujian *black box* disajikan pada Tabel 1.

**Tabel 1.** Hasil pengujian *black box* daftar dan masuk

Halaman	Skenario	Aksi	Hasil yang diharapkan	Hasil
	Membuka halaman "daftar"	Klik tombol "daftar" di navbar <i>landing page</i>	Menampilkan halaman "daftar"	valid
Daftar	Melakukan pendaftaran akun melalui <i>email</i> dan <i>password</i>	Mengisi <i>form</i> nama, <i>email</i> , <i>password</i> , konfirmasi <i>password</i> , <i>checklist</i> syarat dan ketentuan, klik tombol "daftar"	Pengguna menerima <i>email</i> konfirmasi pendaftaran	valid
	Konfirmasi pendaftaran akun	Klik tombol "konfirmasi pendaftaran akun"	Pengguna diarahkan kembali ke sistem TanyaPDF dan pendaftaran akun berhasil	valid
	Melakukan pendaftaran menggunakan akun google	Klik tombol "Masuk/Daftar dengan Google"	Pengguna berhasil mendaftarkan akun	Valid
	Mendaftar akun tanpa mengisi email	Klik tombol daftar tanpa mengisi form email	Muncul pesan error "silahkan isi email yang valid"	Valid
Masuk	Masuk akun menggunakan <i>email</i> dan <i>password</i>	Mengisi data <i>email</i> dan <i>password</i> selanjutnya klik tombol "masuk"	Pengguna berhasil masuk ke dalam sistem	valid



Halaman	Skenario	Aksi	Hasil yang diharapkan	Hasil
	Masuk menggunakan akun google	Klik tombol "Masuk/Daftar dengan Google"	Pengguna masuk dengan akun google	valid
	Masuk menggunakan email/password salah	Mengisi data <i>email</i> dan <i>password</i> selanjutnya klik tombol "masuk"	Muncul pesan error "email/password salah"	valid
	Mengunggah dokumen penelitian	<i>Drag and drop</i> dokumen penelitian dengan format pdf ke bagian unggah file.	Dokumen penelitian berhasil diunggah, diproses, dan sistem membuat <i>chat</i> baru.	valid
Chat	Mengakses <i>chat</i>	Klik salah satu chat di <i>sidebar</i>	Menampilkan detail isi <i>chat</i> dan <i>preview</i> dokumen	valid
	Berpindah halaman pada <i>preview</i> dokumen	Klik tombol <i>previous</i> dan <i>next page</i>	Berhasil pindah halaman	valid
	Mengajukan pertanyaan	Melakukan <i>input</i> pertanyaan terhadap isi dokumen dan klik tombol "kirim"	Mengirim pesan pertanyaan dan AI menjawab pertanyaan tersebut	valid
	Menghapus <i>chat</i>	Klik tombol "hapus" pada <i>sidebar chat</i>	Menghapus dokumen, <i>chat</i> , dan riwayat pesan	valid



Halaman	Skenario	Aksi	Hasil yang diharapkan	Hasil
	Mengubah tampilan tema website	Klik tombol "switch theme"	Mengganti tema webiste (terang/gelap)	valid
	Menampilkan profil pengguna	Klik tombol "profile"	Menampilkan <i>popup</i> nama dan <i>email</i> pengguna serta tombol keluar	valid
	Menampilkan <i>modal</i> keluar dari sistem	klik tombol "keluar" pada <i>popup profile</i>	Menampilkan <i>modal</i> konfirmasi keluar dari sistem	valid
	Keluar dari sistem	klik tombol "keluar" pada <i>modal</i>	Berhasil keluar akun dan <i>redirect</i> ke halaman <i>landing page</i>	
	Menyembunyikan <i>sidebar chat</i>	Klik tombol "hide sidebar"	Berhasil menyembunyikan <i>sidebar</i>	valid

### 3.6 Survei *System Usability Scale*

Pengujian ini berisi 10 pernyataan yang diberikan kepada pengguna setelah menggunakan sistem yang ditunjukkan pada Tabel 2. Pengguna kemudian diminta untuk menjawab antara sangat setuju, setuju, ragu-ragu, atau sangat tidak setuju, dengan pernyataan yang disediakan. Nomor 1-5 digunakan untuk menampilkan tingkat respons pengguna.

**Tabel 2.** Pengujian *System Usability Scale* (SUS)

Pernyataan
Saya tertarik untuk menggunakan sistem ini lagi.
Saya berfikir sistem ini tidak praktis jika digunakan.
Saya berfikir sistem ini mudah untuk digunakan.
Saya perlu meminta bantuan orang teknis, saat menggunakan sistem.
Saya berfikir bahwa fitur atau tombol pada sistem berjalan dengan semestinya.
Saya melihat banyak ketidakserasian pada sistem.
Saya yakin pengguna lain akan mudah memahami cara menggunakan sistem ini.
Saya merasa bingung selama menggunakan sistem.



**Pernyataan**

Saya sangat yakin bisa menggunakan sistem.

Saya perlu belajar dan membiasakan diri, sebelum menggunakan sistem.

Kemudian peneliti menghitung nilai rata-rata SUS berdasarkan skor nilai SUS yang diperoleh. Dengan menghitung rata-rata data skor SUS, kita dapat memperoleh wawasan tentang kecenderungan titik tengah data, yang memberikan indikasi yang jelas tentang kegunaan sistem secara keseluruhan [24]. Data skor SUS dapat dilihat pada tabel 3.

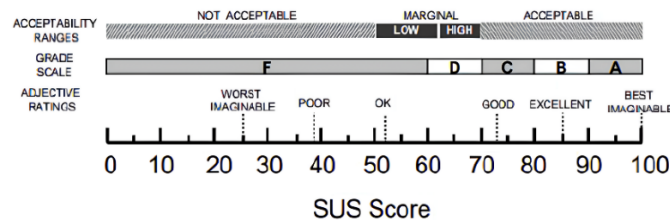
**Tabel 3.** Hasil Pengujian *System Usability Scale* (SUS)

Responden	Hasil skor setiap pertanyaan										Jumlah	Nilai (Jumlah x 2,5)
	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10		
1	4	1	4	3	3	2	4	2	4	3	28	70
2	5	1	5	2	5	2	5	4	5	5	31	77.5
3	3	4	3	4	3	3	3	3	2	4	16	40
4	4	3	3	4	3	3	3	2	4	3	22	55
5	5	2	5	1	4	2	5	1	5	1	37	92.5
6	4	1	4	2	5	2	4	1	4	2	33	82.5
7	5	1	5	2	5	1	5	1	5	1	39	97.5
8	4	3	4	2	3	3	3	2	4	2	26	65
9	5	2	5	5	5	5	5	3	5	5	25	62.5
10	4	2	4	2	4	2	4	2	5	2	31	77.5
11	4	4	4	4	5	4	3	4	3	4	19	47.5
12	5	2	4	3	4	2	4	2	4	3	29	72.5
13	5	1	4	2	4	3	5	2	4	5	29	72.5
14	3	3	3	3	3	3	3	3	3	3	20	50
15	4	2	4	2	4	2	4	2	4	3	29	72.5
16	4	1	5	1	4	1	5	1	5	1	38	95
17	4	1	5	2	4	2	4	4	4	3	29	72.5
18	4	1	5	2	5	3	4	2	4	4	30	75
19	3	3	3	3	3	4	4	4	4	3	20	50
20	5	1	4	3	3	3	3	2	5	1	30	75
21	4	3	3	2	3	3	3	3	3	3	22	55
22	3	2	3	3	3	3	3	3	3	3	21	52.5
23	4	3	3	4	4	3	3	2	3	4	21	52.5
24	4	2	4	2	4	2	4	2	4	3	29	72.5
25	4	2	4	2	3	3	3	2	4	3	26	65
26	4	2	5	2	5	2	5	1	5	2	35	87.5
27	5	1	5	1	5	1	5	2	5	1	39	97.5
28	3	3	3	3	4	3	5	3	3	3	23	57.5



29	5	2	5	2	5	2	5	2	5	2	35	87.5
30	5	3	3	3	3	3	3	3	3	3	22	55
31	5	1	5	1	5	2	5	2	5	1	38	95
32	3	4	1	5	1	5	2	4	1	5	37	92.5
<b>Jumlah</b>											<b>909</b>	<b>2272.5</b>
<b>Rata-rata hasil akhir (Nilai/Jumlah responden)</b>											<b>71.01</b>	

Tabel 3 menyajikan hasil perhitungan menggunakan *System Usability Scale* (SUS) dengan nilai rata-rata 71. Hasil tersebut kemudian diukur kelayakannya menggunakan diagram nilai yang telah ditentukan, seperti yang ditunjukkan pada gambar 10.



**Gambar 10.** Penilaian *System Usability Scale* (SUS)

Berdasarkan Gambar 10, rata-rata nilai hasil pengujian SUS pada sistem TanyaPDF ini berada di angka 71, sehingga termasuk dalam kategori "Acceptable" dengan peringkat "C" dan penilaian "GOOD". Sistem TanyaPDF ini tergolong dalam kategori "Acceptable" menandakan bahwa dapat diterima oleh pengguna.

## PEMBAHASAN

Pada bagian ini, semua hasil penelitian dianalisis dan dijabarkan lebih lanjut untuk memahami pengaruhnya terhadap tujuan penelitian, serta membahas dampak dari penerapan sistem secara menyeluruh.

Hasil penelitian menunjukkan bahwa sistem TanyaPDF yang dikembangkan berhasil memenuhi tujuan utama penelitian, yaitu membantu mahasiswa dalam membaca dan mencari informasi spesifik dari dokumen penelitian secara lebih efisien melalui *institutional repository* Universitas Muhammadiyah Surakarta. Setiap fitur yang telah dirancang mampu berjalan sesuai fungsinya, terlihat dari hasil pengujian *black box* yang seluruhnya menunjukkan status valid. Hal ini membuktikan bahwa alur pendaftaran, autentikasi, unggah dokumen, hingga interaksi melalui antarmuka *chat* dapat digunakan tanpa mengalami kendala fungsional yang berarti.

Keberhasilan ini tidak hanya berhenti pada sisi teknis, namun juga mencerminkan tingkat kegunaan yang cukup baik. Hal ini didukung oleh hasil survei *System Usability Scale* (SUS) dengan nilai rata-rata 71,01 yang termasuk dalam kategori Acceptable dengan peringkat C dan penilaian "GOOD". Nilai tersebut mengindikasikan



bahwa pengguna merasa sistem ini cukup mudah dipahami dan dioperasikan, walaupun masih terdapat ruang untuk pengembangan lebih lanjut agar pengalaman pengguna semakin optimal.

## KESIMPULAN

Penelitian ini berhasil merancang dan mengimplementasikan sistem TanyaPDF yang mampu membantu mahasiswa dalam membaca serta mencari informasi spesifik dari dokumen penelitian di *institutional repository* Universitas Muhammadiyah Surakarta secara lebih efisien. Melalui serangkaian pengujian *black box*, seluruh fitur utama telah terbukti berjalan dengan baik. Hasil uji System Usability Scale (SUS) menunjukkan nilai rata-rata 71,01 yang termasuk kategori Acceptable dengan penilaian "GOOD", sehingga sistem dapat diterima oleh pengguna dan dinilai cukup mudah digunakan.

## UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada staff perpustakaan Universitas Muhammadiyah Surakarta yang telah memberikan fasilitas dan dukungan dalam pengumpulan data serta pengujian sistem TanyaPDF. Selain itu, penulis mengucapkan terima kasih kepada pak Widi Widayat, S.Kom., M.Eng., selaku dosen pembimbing, atas segala arahan, saran, dan masukan yang sangat berarti selama proses penelitian ini berlangsung. Penulis juga menyampaikan banyak terima kasih kepada kedua orang tua, teman, dan saudara atas dukungan yang diberikan selama penelitian ini berlangsung.

## DAFTAR PUSTAKA

- [1] R. Saifur Robbi and E. Sudarmilah, "Perancangan Sistem Informasi Perpustakaan Berbasis Web Pada SDN Pabelan 2 Kartasura," *Infotronik: Jurnal Teknologi Informasi dan Elektronika*, vol. 9, no. 1, pp. 45–58, Jun. 2024, doi: 10.32897/infotronik.2024.9.1.3338.
- [2] F. Reza, I. K. D. Indah, and M. Ropianto, "Perancangan Dan Implementasi Institutional Repository Dengan Metadata Dublin Core," *Jurnal KomtekInfo*, pp. 125–132, Dec. 2022, doi: 10.35134/komtekinfo.v9i4.318.
- [3] K. P. Sari, A. Masruri, and D. R. Rosalia, "Optimalisasi Temu Kembali Informasi Dengan Teknologi Kecerdasan Buatan di Perpustakaan," *JIPi (Jurnal Ilmu Perpustakaan dan Informasi)*, vol. 8, no. 2, p. 349, Nov. 2023, doi: 10.30829/jipi.v8i2.17775.



- [4] N. Carlini *et al.*, *Extracting Training Data from Large Language Models*. 2021. [Online]. Available: <https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting>
- [5] W. X. Zhao *et al.*, "A Survey of Large Language Models," Mar. 2023, [Online]. Available: <http://arxiv.org/abs/2303.18223>
- [6] Q. Zhu and J. Luo, "Generative Pre-Trained Transformer for Design Concept Generation: An Exploration," in *Proceedings of the Design Society*, Cambridge University Press, May 2022, pp. 1825–1834. doi: 10.1017/pds.2022.185.
- [7] OpenAI, "GPT-4o mini: advancing cost-efficient intelligence." Accessed: Apr. 14, 2025. [Online]. Available: <https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/>
- [8] A. Balaguer *et al.*, "RAG vs Fine-tuning: Pipelines, Tradeoffs, and a Case Study on Agriculture," Jan. 2024, [Online]. Available: <http://arxiv.org/abs/2401.08406>
- [9] A. Kusnayat, E. Suryadi, R. Andoyo, E. Mardawati, M. N. Lani, and R. Muhammad, "Enhancing Manufacturing Efficiency at Telkom University: Advanced Automation of Bucket Conveyor and Hammer Mill Machine with PLC and HMI Trough the Waterfall Method," *Journal of Advanced Research in Applied Sciences and Engineering Technology*, vol. 47, no. 2, pp. 223–236, May 2025, doi: 10.37934/ARASET.47.2.223236.
- [10] R. Susanto and A. D. Andriana, "Perbandingan Model Waterfall Dan Prototyping Untuk Pengembangan Sistem Informasi," 2024.
- [11] D. Haq Jayadinata, A. Tirta Pratama, and T. Dewi Sofianti, "Development of Performance Monitoring Dashboard for Product Packaging Manufacturer by using Waterfall Methodology," *Jurnal Ilmiah Teknik Industri*, pp. 151–161, Jun. 2024, doi: 10.23917/jiti.v23i1.2551.
- [12] Y. Sri Rahayu *et al.*, "Implementasi Metode Waterfall Pada Pengembangan Sistem Informasi Mobile E-disarpus," 2024.
- [13] S. Kosasi and S. Margaretha Kuway, "Studi Analisis Persyaratan Kebutuhan Sistem Dalam Menghasilkan Perangkat Lunak Yang Berkualitas," 2012.
- [14] D. Fitria Lestari and E. Sudarmilah, "Rancang Bangun Aplikasi Kesehatan Mental Berbasis Android untuk Penyandang Disabilitas," *The Indonesian Journal of Computer Science*, vol. 14, no. 1, Feb. 2025, doi: 10.33022/ijcs.v14i1.4656.
- [15] A. Y. Aleryani, "Comparative Study between Data Flow Diagram and Use Case Diagram," *International Journal of Scientific and Research Publications*, vol. 6, no. 3, p. 124, 2016, [Online]. Available: [www.ijsrp.org](http://www.ijsrp.org)



- [16] L. Lima, A. Tavares, and S. C. Nogueira, "A framework for verifying deadlock and nondeterminism in UML activity diagrams based on CSP," *Brazilian Symposium on Formal Methods (SBMF)*, pp. 1–29, Oct. 2019.
- [17] K. ' Afiifah, Z. Fira Azzahra, and A. D. Anggoro, "Analisis Teknik Entity-Relationship Diagram dalam Perancangan Database," *Jurnal Intech*, vol. 3, no. 2, pp. 18–22, 2022.
- [18] F. Diatullah Guna Darma, I. Rofni Wulandari, K. Kusumaningtyas, and N. Tri Hartanti, "Implementasi Metode Waterfall Pada Perancangan Website Seminar Nasional KOMA," *Information Technology Journal*, vol. 6, no. 1, 2024.
- [19] G. Brockman, M. Murati, and P. Welinder, "OpenAI API." Accessed: Feb. 22, 2025. [Online]. Available: <https://openai.com/index/openai-api/>
- [20] J. Miao, C. Thongprayoon, S. Suppadungsuk, O. A. Garcia Valencia, and W. Cheungpasitporn, "Integrating Retrieval-Augmented Generation with Large Language Models in Nephrology: Advancing Practical Applications," Mar. 01, 2024, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/medicina60030445.
- [21] J. Bertin, "Advancing Similarity Search with GenAI: A Retrieval Augmented Generation Approach," Dec. 2024, [Online]. Available: <http://arxiv.org/abs/2501.04006>
- [22] M. Mintarsih, "Pengujian Black Box Dengan Teknik Transition Pada Sistem Informasi Perpustakaan Berbasis Web Dengan Metode Waterfall Pada SMC Foundation," *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 5, no. 1, pp. 33–35, Feb. 2023, doi: 10.47233/jteksis.v5i1.727.
- [23] A. Wibowo Soejono, A. Setyanto, and A. Fatah Sofyan, "Evaluasi Usability Website UNRIYO Menggunakan System Usability Scale (Studi Kasus: Website UNRIYO)," 2018, [Online]. Available: [www.respati.ac.id](http://www.respati.ac.id)
- [24] O. Suria, "A Statistical Analysis of System Usability Scale (SUS) Evaluations in Online Learning Platform," *Journal of Information Systems and Informatics*, vol. 6, no. 2, pp. 992–1007, Jun. 2024, doi: 10.51519/journalisi.v6i2.750.

