

**PENGEMBANGAN MODUL PENGECEKAN DAFTAR PUSTAKA PADA
APLIKASI SIKAT**

(Skripsi)

Oleh

**RETNO LESTARI
2117051077**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2025**

**PENGEMBANGAN MODUL PENGECEKAN DAFTAR PUSTAKA PADA
APLIKASI SIKAT**

Oleh

RETNO LESTARI

**Skripsi Sebagai Salah Satu Syarat Untuk Mencapai Gelar
SARJANA KOMPUTER**

Pada

**Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Lampung**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2025**

ABSTRAK
**PENGEMBANGAN MODUL PENGECEKAN DAFTAR PUSTAKA PADA
APLIKASI SIKAT**

OLEH
RETNO LESTARI

Ketidaksesuaian antara sitasi di dalam teks dengan referensi pada daftar pustaka sering menjadi masalah krusial dalam penulisan karya ilmiah berformat *American Psychological Association* (APA). Proses verifikasi manual untuk memastikan kesesuaian ini memakan waktu yang lama dan sangat rentan terhadap kesalahan manusia (*human error*), yang pada akhirnya dapat mengurangi kredibilitas tulisan. Penelitian ini bertujuan untuk mengatasi permasalahan tersebut dengan mengembangkan sistem verifikasi otomatis yang diselesaikan menggunakan metode *Waterfall*. Hasil dari penelitian ini adalah sebuah modul pengecekan daftar pustaka yang terintegrasi pada aplikasi Sikat yang mampu memverifikasi kesesuaian antara sitasi dan referensi. Kinerja modul tersebut kemudian dievaluasi melalui pengujian *Data-Driven Testing* (DDT) terhadap 25 dokumen skripsi. Hasil pengujian menunjukkan bahwa modul mencapai tingkat akurasi rata-rata sebesar 80,01% dengan nilai *False Negative* (FN) nol, yang mengindikasikan sistem berhasil mengidentifikasi seluruh kesalahan tanpa ada yang terlewat. Dari segi efisiensi, sistem mampu menyelesaikan pengecekan dengan waktu rata-rata 62,9 detik per dokumen, jauh lebih cepat dibandingkan pengecekan manual yang memerlukan rata-rata 1442,7 detik (sekitar 24 menit). Modul ini terbukti menjadi solusi yang efektif dan efisien dalam meningkatkan akurasi serta mempercepat proses penyusunan daftar pustaka bagi para akademisi.

Kata Kunci: Pengecekan Sitasi, APA, *Data-Driven Testing*, Aplikasi Sikat

ABSTRACT
DEVELOPMENT OF BIBLIOGRAPHY CHECKING MODULE IN SIKAT
APPLICATION

FROM
RETNO LESTARI

Discrepancies between in-text citations and references in the bibliography often become a critical issue in writing scientific papers using the American Psychological Association (APA) format. The manual verification process to ensure this consistency is time-consuming and highly prone to human error, which can ultimately reduce the credibility of the writing. This study aims to address these problems by developing an automated verification system completed using the Waterfall method. The result of this research is a bibliography checking module integrated into the Sikat application capable of verifying the conformity between citations and references. The module's performance was then evaluated through Data-Driven Testing (DDT) on 25 thesis documents. The test results show that the module achieved an average accuracy rate of 80.01% with a False Negative (FN) value of zero, indicating that the system successfully identified all errors without any being missed. In terms of efficiency, the system was able to complete the check with an average time of 62.9 seconds per document, significantly faster than manual checking which required an average of 1442.7 seconds (approximately 24 minutes). This module has proven to be an effective and efficient solution in improving accuracy and accelerating the process of bibliography compilation for academics.

Keywords: Citation Checking, APA, Data-Driven Testing, Sikat Application

Judul Skripsi : PENGEMBANGAN MODUL PENGECEKAN

DAFTAR PUSTAKA PADA APLIKASI SIKAT

Nama Mahasiswa : Retno Lestari

Nomor Pokok Mahasiswa : 2117051077

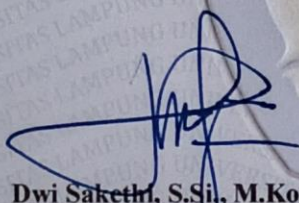
Program Studi : S1 Ilmu Komputer

Jurusan : Ilmu Komputer

Fakultas : Matematika dan Ilmu Pengetahuan Alam

MENYETUJUI

1. Komisi Pembimbing


Dwi Sakethi, S.Si., M.Kom.

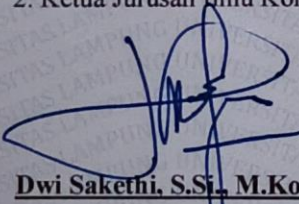
NIP. 196806111998021001


Wartariyus, S.Kom., M.T.I.

NIP. 197301222006041002

2. Ketua Jurusan Ilmu Komputer

3. Ketua Prodi Ilmu Komputer


Dwi Sakethi, S.Si., M.Kom.

NIP. 196806111998021001


Tristivanto, S.Kom., M.I.S., Ph.D.

NIP. 198104142005011001

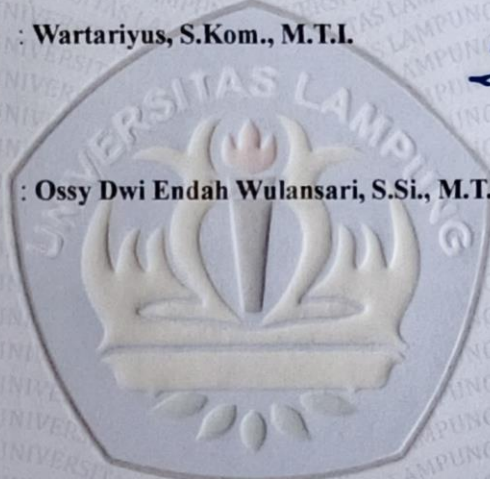
MENGESAHKAN

1. Tim Penguji

Ketua : **Dwi Sakethi, S.Si., M.Kom.**

Sekretaris : **Wartariyus, S.Kom., M.T.I.**

Penguji : **Ossy Dwi Endah Wulansari, S.Si., M.T.**



2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



Dr. Eng. Heri Satria, S.Si., M. Si.
NIP. 197110012005011002

Tanggal Lulus Ujian Skripsi: 19 September 2025

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Retno Lestari

NPM : 2117051056

Menyatakan bahwa skripsi saya yang berjudul **“Pengembangan Modul Pengecekan Daftar Pustaka Pada Aplikasi Sikat”** merupakan karya saya sendiri dan bukan karya orang lain. Semua tulisan yang tertuang di skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti skripsi saya merupakan hasil penjiplakan atau dibuat orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar yang telah saya terima.

Bandar Lampung, 15 Desember 2025

Penulis,



Retno Lestari
NPM. 2117051077

RIWAYAT HIDUP



Lahir di Kab. Lampung Selatan pada hari Senin 17 November 2003. Anak pertama dari empat bersaudara dari Bapak Affan Setiawan dan Ibu Rita Puswati. Menyelesaikan pendidikan dasar di SDN 2 Pasuruan pada Tahun 2015, kemudian menyelesaikan pendidikan menengah pertama di SMPN 1 Penengahan pada Tahun 2018, dan lulus dari pendidikan menengah atas di SMA Kebangsaan pada Tahun 2021.

Pada Tahun 2021, terdaftar menjadi mahasiswa S1 Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung melalui jalur SBMPTN. Kegiatan-kegiatan yang dilakukan selama menjadi mahasiswa yaitu sebagai berikut.

1. Menjadi Anggota Bidang Kajian dan Keumatan UKM-F Rois Fakultas Matematika dan Ilmu Pengetahuan Alam periode 2023/2024.
2. Melaksanakan Kerja Praktik pada bulan Desember – Februari periode 2023/2024 di PT. BPRS Bahtera Artha Jaya (Bank BAJA).
3. Melaksanakan Kuliah Kerja Nyata (KKN) di Desa Labuhan Ratu Kecamatan Pasir Sakti Kabupaten Lampung Timur pada Tahun 2024.

MOTO

"Fabiayyi ala irobbikuma tukadziban"

(QS. Ar-Rahman 55:61)

"It always seems impossible until it's done."

(Nelson Mandela)

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

PERSEMBAHAN

Alhamdulillah ‘ala kulli hal

Puji dan syukur tercurahkan kepada Allah SWT atas segala Rahmat dan Karunia-Nya sehingga saya dapat menyelesaikan skripsi ini. Shalawat serta salam selalu tercurahkan Kepada Nabi Muhammad SAW.

Kupersembahkan karya ini kepada:

Keluargaku Tercinta

Terima kasih yang tak terhingga kuucapkan atas setiap pengorbanan, bimbingan, dan cinta tanpa syarat yang kalian berikan. Kalian telah menjadi pelita yang menerangi jalan, membimbingku, dan memberikan kasih sayang tak terbatas. Kehadiran dan dukungan kalian adalah kekuatan terbesarku dalam menyelesaikan skripsi ini dan meraih gelar sarjana. Perjalanan ini takkan pernah berhasil tanpa setiap tetes keringat, dukungan tulus dan doa yang selalu menyertaiku. Kalian adalah alasan utamaku untuk terus berjuang dan akhirnya mewujudkan mimpi ini.

Terima kasih juga untuk semua pihak yang terlibat dalam pembuatan skripsi ini. Semoga Allah senantiasa membalas setiap kebaikan kalian.

Almamater Tercinta Jurusan Ilmu Komputer FMIPA Universitas Lampung

Setiap pelajaran dan kenangan di sini akan selalu menjadi bekal terbaik untuk melangkah.

SANWACANA

Segala puji dan syukur penulis panjatkan atas kehadiran Allah Subhanahu Wa Ta'ala yang telah memberikan nikmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi ini. Sholawat serta salam penulis sanjung agungkan kepada Baginda Nabi Muhammad Shalallahu 'Alaihi Wasallam yang penulis harapkan syafaatnya di hari akhir kelak.

Skripsi yang berjudul **“Pengembangan Modul Pengecekan Daftar Pustaka Pada Aplikasi Sikat”** ini disusun sebagai salah satu persyaratan untuk memperoleh gelar sarjana di jurusan Ilmu Komputer Universitas Lampung. Penyelesaian skripsi ini tidak lepas dari bantuan, arahan dan dukungan dari berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Allah Subhanahu Wa Ta'ala yang telah memberikan nikmat kesehatan dan kemampuan untuk menyelesaikan skripsi ini.
2. Kedua Orang Tua tercinta Ayahanda Affan Setiawan dan Ibunda Rita Puswati yang selalu memberikan do'a, bimbingan dan dukungan.
3. Ketiga adik tersayang Zidan Albana, Layla Ramadhani, dan M. Dzaki Isam yang menjadi motivasi utama bagi penulis dalam penyelesaian skripsi ini.
4. Keluarga besar terkasih Papa, Ibu, Pakwo, Makwo, Pakde Satiman, Bude Ris, Yuk Icha, Yuk Thata dan sanak saudara lainnya yang tidak dapat penulis sebutkan satu-persatu.
5. Bapak Dr. Eng. Heri Satria, S.Si., M.Si. selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
6. Bapak Dwi Sakethi, M.Kom. selaku Ketua Jurusan Ilmu Komputer FMIPA Universitas Lampung dan Dosen Pembimbing Utama yang telah memberikan bimbingan, arahan, ide, kritik dan saran yang di berikan telah

menjadi kunci keberhasilan penulis dalam menyelesaikan skripsi.

7. Bapak Tristiyanto, S.Kom., M.I.S., Ph.D. selaku Kepala Prodi S1 Ilmu Komputer FMIPA Universitas Lampung dan Dosen Pembimbing Akademik yang telah memberikan saran dan dukungan akademik kepada penulis.
8. Bapak Wartariyus, S.Kom., M.T.I. selaku Dosen Pembimbing II yang memberikan arahan, ide, kritik, dan saran yang menjadi pendorong utama saya dalam menyelesaikan dan memperbaiki skripsi ini dengan baik.
9. Ibu Ossy Dwi Endah Wulansari, S.Si., M.T. selaku Dosen Pembahas yang juga memberikan arahan, saran dan masukan yang bermanfaat dalam perbaikan skripsi ini.
10. Bapak dan Ibu Dosen Jurusan Ilmu Komputer FMIPA Universitas Lampung yang telah memberikan banyak ilmu dan pengalaman semasa perkuliahan.
11. Ibu Ade Nora Maela, Mas Sam dan seluruh staf di Jurusan Ilmu Komputer yang telah membantu segala urusan administrasi penulis di Jurusan Ilmu Komputer.
12. Tersayang "YTTA" Ina, Pira, Shela, Dinda, Revi, Nur, Nurul dan Osmycin yang selalu memberikan dukungan dan motivasi selama perkuliahan. Semoga persahabatan ini tetap terjalin.
13. Tercinta Adhistia Rihhadatul Aisy, Resta Meyliana Puspita dan Ulkiyatul Istiana yang menemani penulis dalam setiap langkah yang penulis ambil.
14. Rekan-rekan satu Jurusan Ilmu Komputer angkatan 2021 yang tidak dapat penulis sebutkan satu persatu semasa menjalani dunia perkuliahan.

Sebagai penutup, penulis menyadari sepenuhnya bahwa skripsi ini masih memiliki banyak kekurangan. Meskipun demikian, besar harapan penulis agar karya ini dapat memberikan manfaat bagi para pembaca. Semoga segala bantuan dan dukungan yang telah diberikan dicatat sebagai amal ibadah oleh Allah SWT. Aamiin.

Bandar Lampung, 15 Desember 2025

Retno Lestari
NPM. 2117051077

DAFTAR ISI

	Hal
ABSTRAK	iii
<i>ABSTRACT</i>	iv
MENGESAHKAN	vi
PERNYATAAN	vii
RIWAYAT HIDUP	viii
MOTO	ix
PERSEMBAHAN	x
SANWACANA	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvi
DAFTAR TABEL	xvii
I. PENDAHULUAN	1
1.1 Latar Belakang & Masalah	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah	4
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	5
II. TINJAUAN PUSTAKA	6
2.1 Daftar Pustaka	6
2.2 Koreksi Kata (Sikat)	7
2.3 <i>American Psychological Association (APA)</i>	7
2.4 <i>Website</i>	8
2.5 <i>HyperText Markup Language (HTML)</i>	9
2.6 <i>Hypertext Preprocessor (PHP)</i>	9
2.7 <i>Entity Relationship Diagram (ERD)</i>	10
2.8 <i>Unified Modeling Language (UML)</i>	11

2.8.1 <i>Use Case Diagram</i>	11
2.8.2 <i>Activity Diagram</i>	14
2.9 Metode <i>Waterfall</i>	16
2.10 <i>Data Driven Testing</i> (DDT).....	17
2.11 Penelitian Terdahulu	18
III. METODOLOGI PENELITIAN	20
3.1 Waktu dan Tempat Penelitian	20
3.2 Alat Pendukung	20
3.2.1 Perangkat Keras (<i>Hardware</i>)	20
3.2.2 Perangkat Lunak (<i>Software</i>)	20
3.3 Tahapan Penelitian.....	21
3.3.1 Identifikasi Masalah.....	22
3.3.2 Studi Pustaka.....	23
3.3.3 Analisis Kebutuhan	24
3.3.4 Desain Sistem	29
3.3.5 Implementasi.....	34
3.3.6 Pengujian	35
3.3.7 Pemeliharaan.....	36
IV. PEMBAHASAN	38
4.1 Implementasi (<i>Coding</i>).....	38
4.1.1 Implementasi Modul Pembacaan Dokumen	38
4.1.2 Implementasi Ekstraksi dan Transformasi Data	43
4.1.3 Implementasi Algoritma Penyesuaian	45
4.1.4 Implementasi Penyajian Hasil dan Persistensi Data	47
4.2 Pengujian	50
4.2.1 <i>Data-Driven Testing</i> (DDT).....	51
4.2.2 Perbandingan Pengecekan Manual dan Sistem	58
V. SIMPULAN DAN SARAN.....	61
5.1 Kesimpulan.....	61

5.2 Saran.....	61
DAFTAR PUSTAKA	62

DAFTAR GAMBAR

Gambar

1. Metode <i>Waterfall</i>	17
2. Tahapan Penelitian	21
3. <i>Use Case Diagram</i>	26
4. Alur Penggunaan Modul	28
5. <i>Activity Diagram</i> Unggah File.....	30
6. <i>Activity Diagram</i> Tampilkan Hasil Pengecekan	31
7. <i>Entity Relationship Diagram</i>	32
8. <i>Wireframe Upload</i>	33
9. <i>Wireframe</i> Hasil.....	34
10. Definisi Kelas dan Inisiasi Objek.....	39
11. Implementasi Logika Pembacaan Konten <i>.docx</i> dan <i>.doc</i>	41
12. Implementasi Metode Utama dan Validasi Tipe File	42
13. Eksekusi Pembacaan dan Pra-pemrosesan Akhir.....	43
14. Implementasi Ekstraksi dan Tranformasi.....	45
15. Implementasi Algoritma Penyesuaian.....	46
16. Penyajian Hasil Analisis pada Antarmuka Pengguna.....	48
17. Tampilan Antarmuka Penyajian Hasil Pengecekan.....	48
18. Implementasi Persistensi Data ke Basis Data	49
19. Tampilan Antarmuka Lihat Hasil	50

DAFTAR TABEL

Tabel

1. Simbol <i>Entity Relationship Diagram</i>	10
2. Simbol <i>Use Case Diagram</i>	12
3. Simbol <i>Activity Diagram</i>	15
4. Penelitian Terdahulu.....	18
5. Spesifikasi Perangkat Keras	20
6. Spesifikasi Perangkat Lunak	21
7. Kebutuhan Fungsional	25
8. Kebutuhan Non-Fungsional	27
9. Empat Komponen.....	52
10. Indeks Persentase	52
11. Pengujian DDT	53
12. Data Uji Skripsi#1	54
13. Hasil Pengujian DDT	56
14. Perbandingan Pengecekan Manual dengan Sistem	59

I. PENDAHULUAN

1.1 Latar Belakang & Masalah

Penulisan daftar pustaka yang akurat dan sesuai dengan format yang ditetapkan, khususnya format *American Psychological Association* (APA), memegang peranan yang sangat penting dalam penulisan karya ilmiah, seperti skripsi, tesis, atau tugas akhir (Suparno & Thamrin, 2021). Daftar pustaka yang tepat dan sesuai tidak hanya berfungsi untuk memberikan informasi terkait sumber yang digunakan dalam penelitian, tetapi juga menunjukkan kredibilitas sumber tersebut. Selain itu, keberadaan daftar pustaka yang benar dan terstruktur dengan baik juga memastikan keabsahan karya ilmiah itu di mata akademik dan menghindari masalah plagiarisme yang dapat mencoreng integritas penelitian. Oleh karena itu, penulisan daftar pustaka yang benar menjadi salah satu elemen penting dalam menghasilkan karya ilmiah yang berkualitas.

Namun, meskipun sudah diatur dengan jelas dalam pedoman penulisan, kenyataannya banyak ditemukan kesalahan dalam penulisan daftar pustaka yang disebabkan oleh ketidaksesuaian antara sitasi yang ada dalam teks dengan referensi yang tercantum dalam daftar pustaka. Kesalahan ini dapat mencakup berbagai hal, seperti penulisan nama penulis yang salah, tahun publikasi yang tidak sesuai, hingga ketidaksesuaian judul atau nomor halaman dalam referensi. Hal ini menunjukkan bahwa pengecekan manual terhadap daftar pustaka sangat rentan terhadap kesalahan, terutama ketika jumlah referensi yang harus diperiksa sangat banyak. Penulisan daftar pustaka yang tidak tepat ini tidak hanya dapat mengurangi kredibilitas karya ilmiah, tetapi juga dapat berdampak pada penilaian akhir yang diberikan oleh pembimbing atau penguji.

Proses pengecekan manual yang dilakukan oleh pengguna untuk memastikan kesesuaian antara sitasi dalam teks dan referensi yang tertera dalam daftar pustaka tentu memakan waktu yang sangat lama. Tidak jarang, pengguna harus memeriksa satu per satu setiap referensi yang pengguna gunakan, yang tentu saja sangat mempengaruhi efisiensi dalam penyusunan dokumen. Selain itu, kesalahan-kesalahan kecil yang tidak terdeteksi selama pengecekan manual dapat berujung pada ketidaksesuaian yang lebih besar, yang akan sangat sulit untuk diperbaiki ketika sudah mendekati waktu deadline.

Seiring dengan pesatnya perkembangan teknologi informasi, penggunaan alat bantu berbasis teknologi menjadi sangat penting untuk mempercepat dan meningkatkan akurasi dalam proses pengecekan daftar pustaka. Aplikasi Sikat, yang sebelumnya telah digunakan untuk pengecekan penulisan kata, memiliki potensi besar untuk dikembangkan lebih lanjut dengan menambahkan modul pengecekan daftar pustaka (Saputra *et al.*, 2021) Integrasi modul ini dengan Sikat dipilih karena aplikasi ini sudah dikenal dan digunakan oleh banyak pengguna, sehingga pengguna dapat dengan mudah beradaptasi dengan fitur baru tanpa perlu mempelajari platform baru. Selain itu, Sikat memiliki infrastruktur teknologi yang mendukung pengembangan fitur tambahan, memungkinkan penambahan modul pengecekan daftar pustaka dengan memanfaatkan sistem yang sudah ada. Pendekatan ini meminimalkan kebutuhan untuk membangun aplikasi baru dari awal, sekaligus memastikan bahwa fitur tersebut sesuai dengan kebutuhan akademik pengguna. Dengan menggunakan aplikasi Sikat, pengguna dapat memverifikasi kesesuaian antara sitasi dan daftar pustaka dengan lebih cepat dan akurat, tanpa harus bergantung pada pengecekan manual yang memakan waktu.

Pengembangan modul pengecekan ini diharapkan dapat menjadi solusi atas permasalahan kesalahan penulisan referensi yang sering kali terjadi akibat kelalaian atau *human error*. Dengan adanya sistem ini, diharapkan pengguna tidak hanya mampu meminimalisir risiko kesalahan, tetapi juga dapat mengalihkan fokus dan waktu pengguna pada aspek yang lebih substansial, seperti pengembangan konten dan analisis data. Solusi pengecekan yang lebih cepat dan akurat ini diharapkan dapat meningkatkan produktivitas pengguna, sehingga waktu yang sebelumnya

digunakan untuk pengecekan manual dapat dimanfaatkan untuk kegiatan penelitian yang lebih mendalam dan produktif.

Kesalahan umum yang sering terjadi misalnya dalam penggunaan "*et al.*" pada sitasi. Banyak yang langsung menulis (Pratama *et al.*, 2020) pada penyebutan pertama, padahal seharusnya semua nama penulis disebutkan seperti (Pratama, Wijaya, & Santoso, 2020) terlebih dahulu, baru kemudian disingkat pada sitasi berikutnya. Kesalahan lain yang umum ditemukan adalah pada format daftar pustaka, terutama pada kapitalisasi judul artikel dan penggunaan cetak miring. Judul artikel seharusnya ditulis dengan format *sentence case* (hanya huruf pertama di awal kalimat yang kapital) dan tanpa tanda kutip, sementara nama jurnal beserta nomor volumenya harus dicetak miring secara bersamaan, contohnya: *Jurnal Sistem Informasi*, 8(2).

Selain itu, aplikasi Sikat yang telah ada juga dapat terus dikembangkan untuk mengakomodasi berbagai format sitasi yang lebih kompleks. Dengan memperkenalkan modul pengecekan daftar pustaka ini, diharapkan pengguna tidak hanya memperoleh solusi untuk masalah pengecekan referensi dalam format APA, tetapi juga dapat menggunakan alat ini untuk berbagai format sitasi lainnya di masa depan. Hal ini akan meningkatkan fleksibilitas dan kegunaan aplikasi dalam mendukung berbagai kebutuhan akademik pengguna.

Secara keseluruhan, pengembangan modul pengecekan daftar pustaka pada aplikasi Sikat ini bertujuan untuk memberikan solusi praktis bagi pengguna dalam memverifikasi kesesuaian sitasi dan daftar pustaka pengguna dengan lebih efisien. Dengan mengadopsi teknologi yang ada, diharapkan proses pengecekan daftar pustaka dapat dilakukan dengan lebih cepat, lebih akurat, dan lebih efisien. Dengan demikian, kualitas penulisan ilmiah pengguna dapat meningkat, dan pengguna dapat lebih fokus pada pengembangan isi dokumen pengguna tanpa khawatir tentang kesalahan penulisan referensi yang dapat menghambat kelancaran proses akademik.

1.2 Rumusan Masalah

Berdasarkan latar belakang dan masalah yang telah diuraikan, maka rumusan masalah dari penelitian ini yaitu :

1. Mengembangkan modul pengecekan daftar pustaka pada aplikasi Sikat yang dapat memverifikasi kesesuaian antara sitasi dalam teks dengan referensi yang tercantum dalam daftar pustaka secara akurat.
2. Menganalisis kemampuan modul ini dalam membantu proses pengecekan kesesuaian antara sitasi dengan referensi.

1.3 Batasan Masalah

Penelitian ini memiliki beberapa batasan masalah, yaitu sebagai berikut :

1. Pengembangan modul pengecekan daftar pustaka akan difokuskan hanya pada format sitasi APA 6 sesuai dengan panduan karya tulis ilmiah Universitas Lampung tahun 2020.
2. Data yang digunakan untuk pengecekan daftar pustaka hanya terbatas pada dokumen bertipe *.docx* dan *doc*.
3. Modul ini hanya akan memfokuskan pada pengecekan kesesuaian sitasi dan daftar pustaka.

1.4 Tujuan Penelitian

Berikut adalah tujuan penelitian pengembangan modul pengecekan daftar pustaka pada aplikasi Sikat:

1. Mengembangkan modul pengecekan daftar pustaka pada aplikasi Sikat untuk memverifikasi kesesuaian sitasi dan referensi.
2. Menganalisis efisiensi modul pengecekan daftar pustaka dalam membantu memeriksa kesesuaian sitasi dan referensi pada dokumen.

1.5 Manfaat Penelitian

Terdapat beberapa manfaat yang didapatkan dari penelitian ini, yaitu :

1. Meningkatkan efisiensi pengecekan daftar pustaka dengan menyediakan alat praktis yang dapat memverifikasi kesesuaian sitasi dan referensi dalam waktu singkat.
2. Meningkatkan akurasi penulisan referensi dalam karya ilmiah.

II. TINJAUAN PUSTAKA

2.1 Daftar Pustaka

Daftar pustaka memiliki peran yang sangat penting dalam penulisan karya ilmiah, seperti skripsi, tesis, atau disertasi (Arisetywan *et al.*, 2022). Penulisan daftar pustaka yang akurat dan sesuai dengan format yang ditetapkan, seperti format *American Psychological Association* (APA), memastikan kredibilitas dan keabsahan sumber yang digunakan dalam penelitian. Namun, meskipun diatur dengan jelas dalam pedoman penulisan, banyak pengguna yang menghadapi kesulitan dalam menyusun daftar pustaka secara manual, sehingga sering terjadi ketidaksesuaian antara sitasi dalam teks dan referensi yang tertera pada bagian daftar pustaka. Hal ini dapat mengurangi kualitas karya ilmiah dan mempengaruhi penilaian akhir. Oleh karena itu, penting bagi pengguna untuk memahami dan mengikuti aturan penulisan daftar pustaka dengan tepat guna menghindari masalah seperti plagiarisme.

Seiring dengan kemajuan teknologi, berbagai aplikasi telah dikembangkan untuk mempermudah proses pembuatan daftar pustaka, salah satunya adalah Mendeley. Mendeley tidak hanya membantu dalam mengelola referensi dan pencarian literatur, tetapi juga memungkinkan pembuatan sitasi dan daftar pustaka secara otomatis. Aplikasi ini terintegrasi dengan Microsoft Word, sehingga memudahkan pengguna untuk menambahkan sitasi dan menghasilkan daftar pustaka yang sesuai dengan berbagai format, termasuk APA. Penggunaan alat bantu seperti Mendeley terbukti efektif dalam meningkatkan efisiensi dan akurasi penyusunan daftar pustaka, sehingga pengguna dapat lebih fokus pada pengembangan isi karya ilmiah pengguna tanpa khawatir tentang kesalahan dalam penulisan referensi.

2.2 Koreksi Kata (Sikat)

Sikat merupakan sebuah sarana yang dirancang untuk mengoreksi kata dasar dan kata yang berimbuhan dalam dokumen bertipe *.docx*. Aplikasi ini bertujuan untuk mempermudah pengguna dalam mendeteksi dan memperbaiki kesalahan penulisan kata pada dokumen yang menggunakan format Microsoft Word. Fitur utama dari aplikasi ini adalah kemampuannya dalam mendeteksi kata dasar dan kata berimbuhan secara otomatis, sehingga dapat membantu pengguna untuk menghindari kesalahan penulisan yang sering terjadi, baik dalam penulisan akademik maupun sehari-hari. Dengan menggunakan aplikasi ini, pengguna dapat menghemat waktu dan tenaga dalam memeriksa kesalahan penulisan secara manual, serta meningkatkan kualitas dokumen yang pengguna buat.

Pengembangan Sikat ini merupakan bagian dari skripsi Sarjana Komputer yang bertujuan untuk menciptakan sebuah sarana yang dapat mendeteksi kesalahan penulisan kata dalam dokumen bertipe *.docx*. Aplikasi ini, yang dikembangkan oleh Andika Saputra pada tahun 2020, diharapkan dapat memberikan kontribusi positif dalam mempermudah proses pengecekan dan koreksi dokumen, khususnya bagi pengguna, penulis, atau profesional yang sering bekerja dengan dokumen berbasis Microsoft Word. Penggunaannya yang sederhana dan efisien menjadikan Sikat sebagai alat bantu yang efektif untuk memastikan keakuratan dalam penulisan dokumen.

2.3 *American Psychological Association (APA)*

Format *American Psychological Association (APA)* adalah salah satu gaya penulisan yang paling banyak digunakan dalam dunia akademik, terutama di bidang ilmu sosial dan pendidikan. Gaya ini dirancang untuk memastikan kejelasan, konsistensi, dan kredibilitas dalam penyajian referensi dan pengutipan dalam karya ilmiah. Menurut penelitian yang dilakukan oleh (Triana *et al.*, 2025), meskipun sebagian besar pengguna memahami dasar-dasar pengutipan dengan gaya APA, masih banyak ditemukan kesalahan teknis dalam penerapannya, seperti kesalahan format kutipan, ketidaksesuaian informasi daftar pustaka, serta kelalaian dalam mencantumkan halaman atau penulis yang tepat. Kesalahan semacam ini dapat

merugikan kualitas penelitian akademik karena dapat mengurangi kredibilitas dan menyebabkan penolakan proposal penelitian.

Terdapat masalah utama yang ditemukan dalam penerapan gaya APA adalah ketidakkonsistenan dalam menulis nama pengarang, kesalahan urutan sumber berdasarkan tahun terbit, dan penggunaan tanda baca yang salah. Gaya APA juga mengatur dengan ketat bagaimana kutipan langsung dan tidak langsung harus disusun, serta aturan-aturan tentang referensi yang harus diikuti agar tulisan dapat memenuhi standar akademik internasional. Oleh karena itu, penting bagi pengguna dan penulis akademik lainnya untuk memahami dan menerapkan prinsip-prinsip gaya APA dengan benar, guna menghindari masalah yang dapat merugikan integritas akademik dan kualitas penelitian pengguna. Terdapat langkah yang dapat diambil untuk memperbaiki kesalahan ini termasuk penyediaan panduan yang komprehensif, pelatihan berkala, serta pemanfaatan teknologi dan perangkat lunak manajemen referensi seperti Mendeley atau EndNote yang dapat membantu memastikan format kutipan dan referensi sesuai dengan standar APA.

2.4. Website

Website adalah sekumpulan halaman informasi yang dapat diakses melalui internet menggunakan perangkat lunak browser seperti Google Chrome atau Mozilla Firefox (Setyaningrum *et al.*, 2025). Dalam aplikasi Sikat, *website* memainkan peran penting sebagai platform yang memungkinkan pengguna untuk mengakses dan menggunakan fitur-fitur pengecekan secara online. Aplikasi Sikat, yang berbasis web, menyediakan kemudahan bagi pengguna untuk melakukan pengecekan daftar pustaka dan koreksi kata secara otomatis, tanpa perlu menginstal perangkat lunak tambahan pada komputer. Penggunaan *website* sebagai platform utama memungkinkan aplikasi ini lebih fleksibel dan mudah diakses oleh pengguna dari berbagai perangkat yang terhubung ke internet.

2.5. *HyperText Markup Language (HTML)*

HTML (*Hyper Text Markup Language*) adalah bahasa pemrograman yang digunakan untuk membuat dan mendesain halaman web (Mardiati & Saputra, 2025). HTML berfungsi sebagai fondasi utama bagi pembuatan *website* dengan menyusun elemen-elemen dasar yang membentuk struktur halaman web, seperti paragraf, gambar, tautan, dan tabel. Dalam pengembangan *website*, HTML digunakan untuk menentukan struktur dasar dari halaman, di mana setiap elemen diberikan tag tertentu yang akan diinterpretasikan oleh browser untuk ditampilkan kepada pengguna. HTML merupakan bahasa yang bersifat statis, artinya konten yang ditampilkan tidak berubah kecuali ada perubahan yang dilakukan oleh pengembang *website*. Dalam pengembangan aplikasi berbasis web seperti Sikat, HTML digunakan untuk membuat antarmuka pengguna (UI) yang sederhana dan mudah diakses, memungkinkan interaksi pengguna dengan aplikasi secara langsung melalui browser.



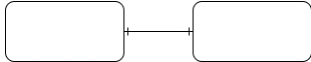
2.6 *Hypertext Preprocessor (PHP)*

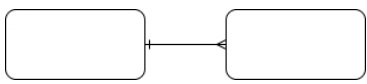
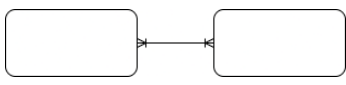
PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman yang digunakan untuk membuat aplikasi web dinamis dan interaktif (Arista Kusuma, 2025). PHP bekerja pada sisi server dan digunakan untuk mengelola berbagai proses, seperti pengolahan data, pemrosesan form, dan komunikasi dengan basis data. Dalam konteks pengembangan aplikasi berbasis web seperti Sikat, PHP memungkinkan pembuatan fitur dinamis yang mendukung pengolahan data secara langsung pada server dan menghasilkan output yang dikirimkan ke browser pengguna. Sebagai bahasa pemrograman yang sangat fleksibel dan efisien, PHP sering digunakan bersama dengan HTML untuk membangun *website* atau aplikasi berbasis web yang dapat berinteraksi dengan pengguna. PHP dapat mempermudah pengembangan aplikasi web dengan menyediakan berbagai fitur bawaan untuk mengelola sesi pengguna, bekerja dengan basis data MySQL, serta mengirimkan data secara aman melalui HTTP.

2.7 Entity Relationship Diagram (ERD)

Entity-Relationship Diagram (ERD) adalah alat bantu yang digunakan dalam perancangan basis data untuk menggambarkan hubungan antar entitas dalam sistem informasi (Afiifah *et al.*, 2022). ERD memanfaatkan simbol grafis untuk menggambarkan entitas, atribut, dan relasi di antara entitas dalam basis data. Tiga elemen dasar dalam ERD adalah entitas, yang mewakili objek yang relevan dengan data yang dibutuhkan; atribut, yang menggambarkan informasi yang terkait dengan entitas; dan relasi, yang menggambarkan hubungan antara entitas. Dalam pengembangan aplikasi seperti Sikat, ERD digunakan untuk merancang struktur basis data yang efisien, memetakan bagaimana data akan saling berinteraksi, dan memastikan bahwa setiap entitas dan relasi terdefinisi dengan jelas untuk menghindari kesalahan desain yang dapat mengganggu kinerja aplikasi. Berikut adalah daftar simbol ERD menurut (Afiifah *et al.*, 2022):

Tabel 1. Simbol *Entity Relationship Diagram*

Bentuk	Nama Simbol	Fungsi Simbol
	Entitas	Entitas adalah sebuah objek berwujud nyata yang dapat dibedakan dengan objek lainnya.
	Atribut	Field atau disebut sebagai atribut. Setiap entitas memiliki atribut untuk mendeskripsikan karakteristik dari suatu entitas
	Relasi <i>One to One</i>	Setiap entitas hanya memiliki hubungan dengan satu entitas lain.
Bentuk	Nama Simbol	Fungsi Simbol

	Relasi <i>One to Many</i>	Satu entitas dapat memiliki hubungan dengan banyak entitas lain
	Relasi <i>Many to Many</i>	Relasi di mana setiap lebih dari satu baris data dari Entitas pertama berhubungan dengan lebih dari satu baris data pada entitas kedua

2.8 Unified Modeling Language (UML)

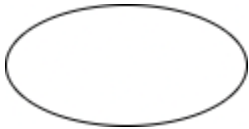


Unified Modeling Language (UML) adalah sebuah bahasa standar yang digunakan untuk menggambarkan, merancang, dan mendokumentasikan sistem perangkat lunak berbasis objek. UML menyediakan berbagai jenis diagram, seperti *Use Case* Diagram, dan *Activity* Diagram, yang digunakan untuk menggambarkan berbagai aspek sistem secara visual dan mudah dipahami. Penggunaan UML memudahkan pengembang dan tim proyek untuk berkomunikasi tentang desain sistem dengan cara yang terstruktur, serta memberikan gambaran yang jelas mengenai alur kerja, hubungan antar objek, dan proses yang terjadi dalam sistem. Dalam pengembangan aplikasi berbasis web seperti Sikat, UML digunakan untuk merancang arsitektur sistem, menggambarkan interaksi antar komponen, serta mendokumentasikan alur kerja aplikasi, sehingga mempermudah pengembangan dan pemeliharaan perangkat lunak.


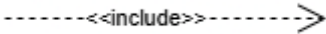
2.8.1 Use Case Diagram

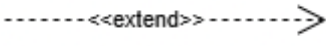
Use Case diagram merupakan model yang dihasilkan dari proses analisis dan perancangan sistem yang bertujuan untuk menggambarkan kebutuhan sistem secara visual. Kebutuhan-kebutuhan ini akan diimplementasikan oleh pengguna, sehingga rancangan sistem dapat tergambarkan dengan jelas. *Use Case* diagram pada dasarnya adalah representasi abstrak dari interaksi yang terjadi antara sistem dan aktor. Karena itu, pemilihan tingkat abstraksi yang tepat dan sesuai dengan

kebutuhan sistem menjadi sangat penting. Cara kerja *Use Case* diagram adalah dengan mendeskripsikan pola-pola interaksi yang khas antara admin dan pengguna dalam sebuah sistem melalui diagram yang menunjukkan bagaimana sistem tersebut digunakan dalam praktiknya. Berikut daftar simbol *Use Case* diagram menurut (Ramdany *et al.*, 2024).

Tabel 2. Simbol *Use Case* Diagram

Bentuk	Nama Simbol	Fungsi Simbol
	<i>Use Case</i>	Fungsionalitas yang disediakan sistem sebagai unit –unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>Use Case</i> .
	Asosiasi	Komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.
	Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor



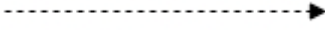


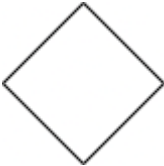
		<p>belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor. Aktor merupakan peran yang dimainkan oleh pemakai ketika berinteraksi dengan system</p>
	<i>Generalization</i>	<p>Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah <i>Use Case</i> di mana fungsi yang satu adalah fungsi yang lebih umum dari lainnya</p>
	<i>Include</i>	<p><i>Include</i> menunjukkan bahwa <i>Use Case</i> yang ditambahkan akan selalu dipanggil ketika <i>Use Case</i> tambahan dijalankan, sedangkan <i>Include</i> memiliki makna bahwa <i>Use Case</i> tambahan akan selalu melakukan verifikasi apakah <i>Use Case</i> yang ditambahkan telah dieksekusi sebelum <i>Use Case</i> tambahan itu sendiri dijalankan, sehingga membentuk suatu hubungan ketergantungan</p>

		yang terstruktur dalam alur proses sistem.
	<i>Extend</i>	Relasi <i>Use Case</i> tambahan ke sebuah <i>Use Case</i> dan <i>Use Case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>Use Case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>Use Case</i> yang ditambahkan.

2.8.2 Activity Diagram

Activity diagram mengilustrasikan alur fungsionalitas dalam sistem informasi. Secara komprehensif, *Activity* diagram menetapkan titik awal dan akhir dari suatu workflow, berbagai aktivitas yang berlangsung selama proses workflow tersebut, serta menjelaskan urutan kronologis dari rangkaian aktivitas yang terjadi. *Activity* diagram juga menawarkan metode untuk memodelkan proses yang berjalan secara paralel. Bagi individu yang sudah familiar dengan analisis dan perancangan struktur tradisional, diagram ini mengintegrasikan konsep-konsep fundamental yang menjadi dasar dari diagram alir data dan diagram alur sistem. Berikut simbol *Activity* diagram menurut (Suharni *et al.*, 2023).

Tabel 3. Simbol *Activity* Diagram

Bentuk Simbol	Nama Simbol	Fungsi Simbol
	<i>Activity</i>	Menggambarkan mekanisme interaksi antar kelas antarmuka dalam sistem, menjelaskan hubungan dan komunikasi yang terjadi di antara pengguna.
	<i>Control Flow</i>	Merepresentasikan urutan pelaksanaan proses dalam sistem secara kronologis.
	<i>Object Flow</i>	Menunjukkan aliran objek dari suatu action atau <i>Activity</i> ke action berikutnya.
	<i>Start Point</i>	Menyatakan bahwa suatu objek atau proses dimulai.
	<i>End Point</i>	Menyatakan bahwa suatu objek atau proses selesai.
	<i>Decesion</i>	Mengindikasikan adanya keputusan atau tindakan yang perlu dilakukan berdasarkan kondisi atau kriteria tertentu yang telah ditentukan.

2.9 Metode *Waterfall*

Metode *Waterfall* adalah salah satu pendekatan dalam pengembangan perangkat lunak yang mengedepankan proses yang sistematis dan berurutan, di mana setiap tahap harus diselesaikan sebelum melanjutkan ke tahap berikutnya (Alif et al., 2023). Berikut adalah tahapan yang akan di lakukan dalam penelitian ini :

1. Analisis Kebutuhan (*Requirement Analysis*)

Pada tahap awal, dilakukan identifikasi dan perincian kebutuhan aplikasi Sikat secara mendetail. Hal ini mencakup penentuan fungsi utama, fitur yang dibutuhkan, serta batasan teknis dan tujuan aplikasi. Semua kebutuhan ini didokumentasikan secara jelas sebagai dasar pengembangan selanjutnya.

2. Pemodelan Sistem

Setelah kebutuhan dianalisis, dibuat desain dan model sistem aplikasi. Tahap ini meliputi pembuatan diagram alur kerja, skema basis data, dan rancangan antarmuka pengguna agar struktur dan proses kerja aplikasi dapat divisualisasikan secara menyeluruh.

3. Pengkodean (*Implementasi*)

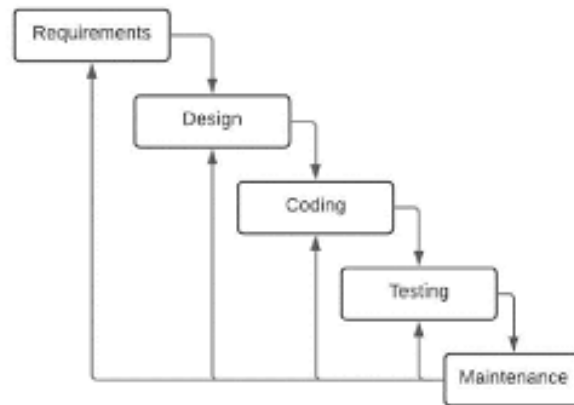
Berdasarkan model dan desain yang telah dibuat, dilakukan proses penulisan kode program aplikasi secara mandiri, dengan memastikan setiap fungsi dan fitur berjalan sesuai spesifikasi yang telah direncanakan.

4. Pengujian (*Testing*)

Setelah pengkodean selesai, dilakukan pengujian menyeluruh untuk memastikan aplikasi bebas dari kesalahan, fitur berfungsi dengan baik, dan sistem berjalan stabil saat digunakan.

5. Pemeliharaan (*Maintenance*)

Setelah aplikasi siap digunakan, tahap pemeliharaan dilakukan dengan memperbaiki bug yang ditemukan serta melakukan pembaruan atau penyempurnaan agar aplikasi tetap optimal dan sesuai kebutuhan yang berkembang.



Gambar 1. Metode *Waterfall*.

2.10 Data Driven Testing (DDT)

Data-Driven Testing (DDT) adalah sebuah metodologi pengujian perangkat lunak di mana logika dari skrip pengujian dipisahkan dari data yang digunakan untuk pengujian itu sendiri (Rininda *et al.*, 2023). Dalam pendekatan ini, skrip tes dirancang untuk membaca berbagai macam input beserta hasil yang sesuai (*expected output*) dari sumber data eksternal, seperti *spreadsheet*, file CSV, atau *basis data*. Keunggulan utama dari metode ini adalah kemampuannya untuk menjalankan satu skrip tes secara berulang dengan ratusan atau ribuan variasi data yang berbeda, sehingga meningkatkan cakupan pengujian secara signifikan tanpa perlu menulis ulang kode untuk setiap skenario pengujian. Tujuan DDT adalah untuk memvalidasi fungsionalitas dan ketangguhan (*robustness*) sistem terhadap berbagai kondisi data secara efisien.

2.11 Penelitian Terdahulu

Berikut beberapa penelitian terdahulu yang selaras dengan penelitian saat ini.

Tabel 4. Penelitian Terdahulu

Nama (Tahun)	Judul Penelitian	Hasil Penelitian
Saputra (2021)	Sistem Pendeteksi Penulisan Kata pada Dokumen Berbasis Web	Penelitian ini mengembangkan utility untuk mendeteksi kesalahan penulisan kata pada dokumen Word (.docx) berbasis web. Hasil penelitian menunjukkan bahwa sistem dapat mendeteksi kesalahan penulisan kata dasar dan kata berimbuhan dengan mengandalkan kamus sebagai referensi
Salim (2020)	Perangkat Lunak Koreksi Penggunaan Sumber Kutipan dan Daftar Pustaka pada Jurnal Pengguna	Penelitian ini menghasilkan perangkat lunak berbasis desktop yang mampu mengoreksi sumber kutipan dan daftar pustaka secara otomatis. Hasil pengujian menunjukkan bahwa aplikasi ini dapat membantu editor dalam mempercepat pengecekan dan mengurangi kesalahan dalam sumber kutipan dan daftar pustaka
Sakethi (2022)	Utility Klasifikasi Kata Berdasarkan Kamus pada Dokumen Bertipe	Penelitian ini mengembangkan sistem untuk mengklasifikasikan kata dalam dokumen Word (.docx) berbasis web. Hasil

	Word (.docx) Berbasis Web	penelitian menunjukkan tingkat kepuasan pengguna sebesar 88,35%, yang menunjukkan sistem ini sangat efektif dalam mengklasifikasikan kata
--	---------------------------	---

III. METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian

Waktu pengerjaan penelitian ini dimulai pada semester genap tahun 2025 dan dilakukan di lingkungan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.

3.2 Alat Pendukung

Berikut adalah alat yang digunakan dalam penelitian ini.

3.2.1 Perangkat Keras (*Hardware*)

Dalam penelitian ini dibutuhkan perangkat keras yang mampu menunjang selama pengembangan yaitu laptop ASUS, untuk spesifikasinya dapat dilihat pada tabel 5.

Tabel 5. Spesifikasi Perangkat Keras

Nama Perangkat	Spesifikasi
Processor	Intel(R) Core(TM) i3-4
RAM	12GB
CPU	1.70Ghz

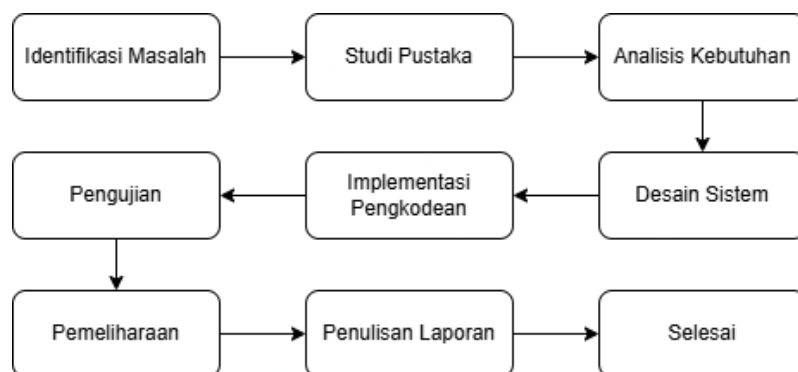
3.2.2 Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan dalam penelitian dapat dilihat pada tabel 6.

Tabel 6. Spesifikasi Perangkat Lunak

Nama Software	Kegunaan
Visual Studio Code	Digunakan sebagai text editor untuk menulis dan mengedit kode sumber aplikasi
XAMPP	Berfungsi sebagai server lokal yang memungkinkan pengembangan dan pengujian aplikasi web
Google Chrome	Digunakan sebagai browser untuk mengakses dan menguji aplikasi web, serta memverifikasi tampilan dan fungsionalitas sistem pada berbagai perangkat dan resolusi layar.

3.3 Tahapan Penelitian

**Gambar 2.** Tahapan Penelitian.

Proses penelitian dan pengembangan modul pengecekan sitasi Anda dimulai dari identifikasi masalah terkait kesulitan penulisan format APA 6, yang dilanjutkan dengan studi pustaka untuk mengumpulkan landasan teori dan analisis kebutuhan untuk mendefinisikan fitur-fitur sistem. Berdasarkan hasil analisis, Anda melakukan desain sistem untuk merancang arsitektur dan antarmuka, yang kemudian diwujudkan melalui implementasi pengkodean. Setelah modul selesai dibuat, dilakukan pengujian secara menyeluruh untuk memastikan fungsionalitas dan akurasi, lalu masuk ke tahap pemeliharaan untuk perbaikan dan pengembangan berkelanjutan. Seluruh rangkaian proses ini pada akhirnya

didokumentasikan secara sistematis dalam penulisan laporan hingga proyek dianggap selesai.

3.3.1 Identifikasi Masalah

Dalam penulisan karya ilmiah seperti skripsi, tesis, atau tugas akhir, penyusunan daftar pustaka yang sesuai dengan format yang ditetapkan merupakan salah satu elemen penting yang mendukung kredibilitas karya tersebut. Salah satu format yang paling umum digunakan dalam penulisan akademik adalah format *American Psychological Association* (APA). Meskipun telah ada pedoman yang jelas mengenai penulisan daftar pustaka, banyak ditemukan kesalahan yang terjadi dalam proses penulisan referensi ini.

Terdapat tiga masalah utama yang sering terjadi antara lain:

1. Ketidaksesuaian antara sitasi dalam teks dan referensi dalam daftar pustaka. Pengguna seringkali mengalami kesulitan dalam memastikan bahwa setiap sitasi yang disebutkan dalam teks telah tercantum dengan benar dalam daftar pustaka, baik dalam hal nama penulis, tahun publikasi, maupun informasi lainnya.
2. Kesalahan format dalam penulisan referensi. Meskipun terdapat panduan tentang format APA, banyak ditemukan kesalahan teknis seperti penulisan nama penulis yang tidak sesuai, penempatan tanda baca yang salah, serta ketidaksesuaian urutan informasi referensi yang memengaruhi kesesuaian dengan standar APA.
3. Proses pengecekan yang memakan waktu dan rentan kesalahan. Pengecekan manual yang dilakukan oleh pengguna untuk memverifikasi kesesuaian antara sitasi dan daftar pustaka memerlukan waktu yang lama, terutama ketika jumlah referensi sangat banyak. Proses ini tidak hanya mengganggu efisiensi, tetapi juga sering kali tidak terhindar dari kesalahan manusia, yang pada akhirnya dapat mempengaruhi kualitas penelitian.

Masalah-masalah tersebut menunjukkan bahwa penyusunan daftar pustaka yang akurat memerlukan perhatian yang lebih besar, baik dari segi teknis maupun waktu. Untuk itu, diperlukan solusi yang dapat membantu pengguna dalam mengatasi masalah ini, salah satunya adalah dengan pengembangan modul pengecekan daftar pustaka pada aplikasi Sikat. Modul ini diharapkan dapat memverifikasi kesesuaian antara sitasi dalam teks dengan referensi dalam daftar pustaka secara cepat, akurat, dan efisien, mengurangi kesalahan manusia yang sering terjadi dalam pengecekan manual.

3.3.2 Studi Pustaka

Beberapa penelitian terdahulu telah dilakukan untuk mengembangkan sistem yang dapat mendeteksi kesalahan dalam penulisan dokumen, terutama yang berkaitan dengan penulisan kata dan pengecekan referensi. Salah satu penelitian yang relevan adalah penelitian yang dilakukan oleh Saputra (2021), yang mengembangkan sebuah sistem berbasis web untuk mendeteksi kesalahan penulisan kata pada dokumen Word (.docx). Sistem ini memanfaatkan kamus sebagai referensi untuk mendeteksi kesalahan pada kata dasar dan kata berimbuhan, yang menunjukkan efektivitasnya dalam memperbaiki kesalahan penulisan secara otomatis.

Selain itu, (Salim *et al.*, 2020) mengembangkan perangkat lunak berbasis desktop yang dirancang untuk mengoreksi penggunaan sumber kutipan dan daftar pustaka pada jurnal pengguna. Hasil pengujian dari perangkat lunak ini menunjukkan bahwa aplikasi tersebut dapat mempercepat proses pengecekan sumber kutipan dan daftar pustaka, serta mengurangi kesalahan yang sering terjadi dalam penyusunan referensi. Ini menunjukkan adanya potensi untuk mengembangkan alat bantu yang dapat meningkatkan efisiensi dan akurasi dalam penulisan daftar pustaka.

Penelitian lainnya yang relevan adalah oleh (Sakethi & Kartika, 2022), yang mengembangkan sebuah sistem berbasis web untuk mengklasifikasikan kata-kata dalam dokumen Word (.docx). Sistem ini menunjukkan hasil yang sangat memuaskan, dengan tingkat kepuasan pengguna mencapai 88,35%. Penelitian ini membuktikan bahwa sistem berbasis web dapat memberikan solusi yang efektif dalam mempermudah proses klasifikasi kata, yang memiliki potensi untuk diterapkan dalam pengecekan penulisan kata dan referensi.

Penelitian-penelitian tersebut memberikan dasar yang kuat bagi pengembangan aplikasi serupa yang dapat membantu pengguna dalam memverifikasi kesesuaian sitasi dan daftar pustaka. Dengan adanya sistem yang dapat mengoreksi dan memverifikasi penulisan referensi secara lebih cepat dan akurat, kualitas penulisan ilmiah dapat meningkat, mengurangi kesalahan manusia yang sering terjadi dalam pengecekan manual.

3.3.3 Analisis Kebutuhan

Pengguna utama dari sistem ini adalah civitas akademika yang sedang mengerjakan karya ilmiah seperti skripsi atau tugas akhir. Tujuan utama pengguna adalah untuk mendapatkan solusi praktis yang dapat memverifikasi kesesuaian antara sitasi di dalam teks dengan referensi di daftar pustaka secara efisien, cepat, dan akurat, guna menghindari kesalahan yang sering terjadi pada pengecekan manual. Untuk mewujudkan solusi tersebut, pada tahap analisis kebutuhan, terdapat dua kategori utama yang perlu diperhatikan: kebutuhan fungsional dan kebutuhan nonfungsional.

Kebutuhan fungsional secara spesifik akan mencakup fitur dan kemampuan yang harus ada dalam sistem untuk mencapai tujuan utama tersebut, seperti kemampuan memverifikasi kesesuaian sitasi dan referensi sesuai format APA 6. Sementara itu, kebutuhan nonfungsional akan meliputi aspek-aspek yang berhubungan dengan kualitas dan kinerja sistem seperti kecepatan pengecekan dan kemudahan penggunaan antarmuka yang secara langsung mendukung tujuan agar solusi yang dihasilkan terasa efisien dan cepat bagi pengguna. Pada akhirnya, kedua jenis kebutuhan ini saling melengkapi untuk memastikan bahwa sistem yang dikembangkan tidak hanya berfungsi dengan baik, tetapi juga mampu memberikan pengalaman pengguna yang optimal serta memenuhi standar kualitas yang diinginkan.

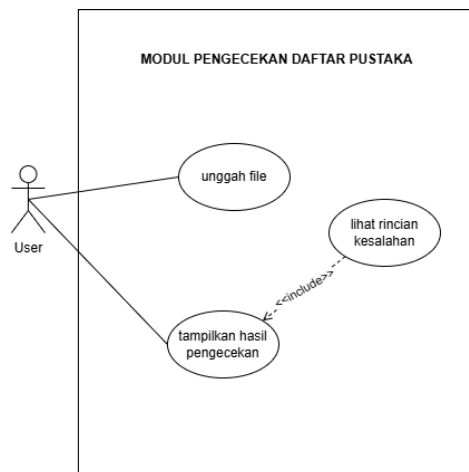
3.3.3.1 Kebutuhan Fungsional

Kebutuhan fungsional dalam pengembangan modul pengecekan daftar pustaka pada aplikasi ini mencakup berbagai fitur utama yang harus ada untuk memenuhi tujuan sistem. Berikut adalah kebutuhan fungsional dalam bentuk tabel.

Tabel 7. Kebutuhan Fungsional

Kebutuhan Fungsional	Deskripsi
Pengunggahan File	Pengguna dapat mengunggah file dokumen dalam format <i>.docx</i> yang akan diperiksa.
Verifikasi Sitasi dan Daftar Pustaka	Sistem memverifikasi kesesuaian antara sitasi dalam teks dan referensi dalam daftar pustaka sesuai format APA 6.
Tombol Cek	Setelah mengunggah file, pengguna dapat menekan tombol "cek" untuk memulai proses pengecekan.
Halaman Hasil	Setelah pengecekan selesai, hasil verifikasi ditampilkan pada halaman baru, yang menunjukkan kesesuaian dan kesalahan.
Penyajian Hasil	Hasil pengecekan disajikan secara jelas dan mudah dipahami, dengan detail kesalahan yang ditemukan

Kebutuhan-kebutuhan fungsional yang telah diuraikan pada tabel di atas merupakan landasan bagi fitur utama aplikasi. Untuk menggambarkan bagaimana pengguna berinteraksi dengan fitur-fitur tersebut, kebutuhan ini dimodelkan ke dalam sebuah alur kerja yang terstruktur. Alur kerja interaksi antara pengguna dan sistem ini kemudian direpresentasikan secara visual melalui diagram *Use Case* yang disajikan pada Gambar 3.



Gambar 3. *Use Case Diagram.*

Dalam gambar 3, terdapat *Use Case* yang menggambarkan fungsionalitas utama aplikasi. Pertama, Pengguna akan melakukan unggah file, yang merupakan langkah pertama dalam proses pengecekan. Setelah file diunggah, sistem akan memproses file tersebut dan menampilkan hasil pengecekan melalui tampilkan hasil pengecekan. Hasil ini menunjukkan apakah sitasi dan daftar pustaka sesuai dengan format yang telah ditetapkan. Selain itu, jika terdapat kesalahan dalam referensi, Pengguna dapat memilih untuk lihat rincian kesalahan, yang memberikan penjelasan lebih lanjut tentang kesalahan yang ditemukan dalam file tersebut, seperti ketidaksesuaian nama penulis, tahun publikasi, atau elemen referensi lainnya.

Terdapat relasi include antara tampilkan hasil pengecekan dan lihat rincian kesalahan, yang menunjukkan bahwa menampilkan hasil pengecekan termasuk dalam proses melihat rincian kesalahan. Artinya, setelah hasil pengecekan ditampilkan, pengguna dapat langsung melihat kesalahan yang ditemukan dalam referensi. Secara keseluruhan, *Use Case Diagram* ini menggambarkan proses yang sederhana namun jelas mengenai bagaimana Pengguna berinteraksi dengan sistem

untuk mengunggah file, memverifikasi kesesuaian referensi, dan melihat rincian kesalahan yang terdeteksi dalam daftar pustaka. Diagram ini memberikan gambaran yang komprehensif mengenai alur interaksi dan fungsionalitas utama dari aplikasi pengecekan daftar pustaka yang akan dikembangkan.

3.3.3.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional dalam pengembangan modul pengecekan daftar pustaka ini mencakup aspek-aspek yang mendukung kualitas dan kinerja sistem secara keseluruhan. Beberapa kebutuhan non-fungsional tersebut antara lain dapat dilihat pada tabel berikut.

Tabel 8. Kebutuhan Non-Fungsional

Kebutuhan Non-Fungsional	Deskripsi
Kemudahan Penggunaan	Antarmuka pengguna (UI) harus sederhana dan intuitif agar mudah dipahami dan digunakan oleh pengguna tanpa membutuhkan pelatihan khusus.
Kompatibilitas	Aplikasi harus dapat dijalankan di berbagai perangkat dengan sistem operasi yang berbeda, seperti Windows, macOS, dan Linux.
Skalabilitas	Sistem harus dapat menangani berbagai ukuran file dan jumlah pengguna tanpa penurunan kinerja yang signifikan.
Responsivitas Tampilan	Tampilan aplikasi harus responsif, memastikan bahwa aplikasi dapat digunakan dengan baik di berbagai ukuran layar dan perangkat (desktop, tablet, ponsel).

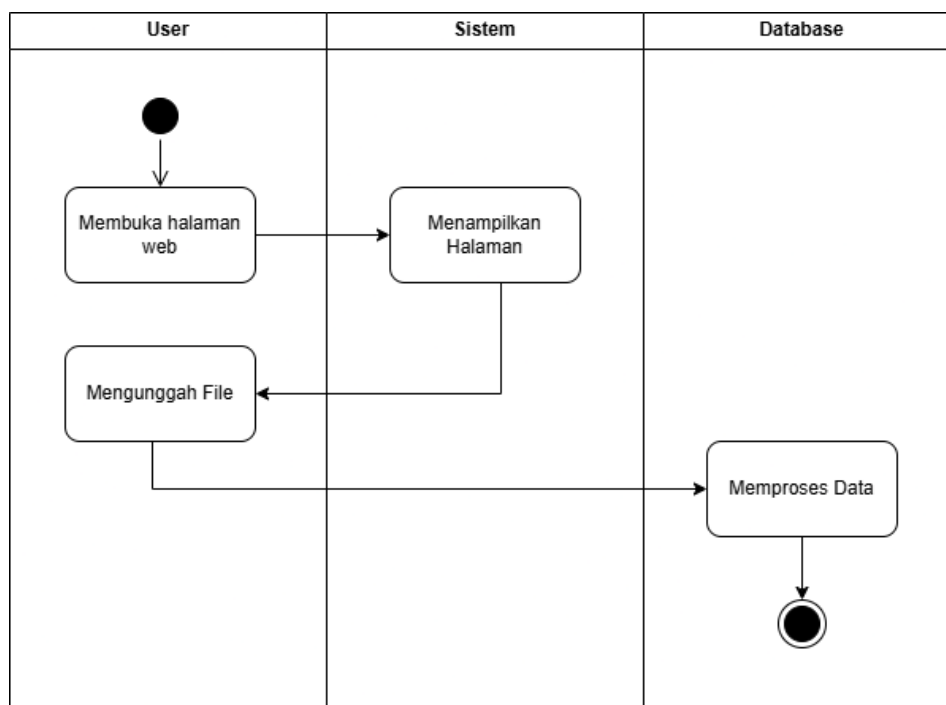
3.3.4 Desain Sistem

Pada tahap desain sistem, diagram akan dibuat untuk menggambarkan struktur dan alur kerja dari aplikasi pengecekan daftar pustaka. Desain sistem ini bertujuan untuk merinci bagaimana setiap komponen dan fitur dalam aplikasi akan saling berinteraksi serta bagaimana data akan diproses dan disimpan. Tiga diagram utama yang akan digunakan dalam desain sistem ini adalah *Entity-Relationship Diagram* (ERD), *Use Case Diagram*, dan *Activity Diagram*. ERD akan digunakan untuk menggambarkan hubungan antar entitas dalam basis data aplikasi, seperti entitas pengguna, file, dan hasil pengecekan. *Use Case Diagram* akan menggambarkan interaksi antara pengguna dan sistem, serta fungsionalitas utama yang dapat diakses oleh pengguna. Sedangkan *Activity Diagram* akan menggambarkan alur proses yang terjadi selama penggunaan aplikasi, mulai dari pengunggahan file hingga penampilan hasil pengecekan. Diagram-diagram ini diharapkan dapat memberikan gambaran yang jelas mengenai struktur dan alur kerja sistem, serta mempermudah pengembangan dan pemeliharaan aplikasi.

3.3.4.1 *Activity Diagram*

Dalam penelitian ini, terdapat dua aktivitas utama yang digambarkan melalui *Activity Diagram*, yaitu unggah file dan tampilkan hasil pengecekan. Kedua aktivitas ini merupakan langkah-langkah penting dalam alur kerja aplikasi pengecekan daftar pustaka. Aktivitas pertama, unggah file, melibatkan pengguna yang mengunggah file dokumen untuk diperiksa, sementara aktivitas kedua, tampilkan hasil pengecekan, menunjukkan proses setelah file selesai diproses oleh sistem, di mana hasil pengecekan ditampilkan kepada pengguna. Kedua aktivitas ini berperan dalam memastikan bahwa sistem dapat dengan efisien memverifikasi kesesuaian sitasi dan daftar pustaka, serta memberikan hasil yang jelas dan terperinci kepada pengguna.

1. Activity Diagram Unggah File

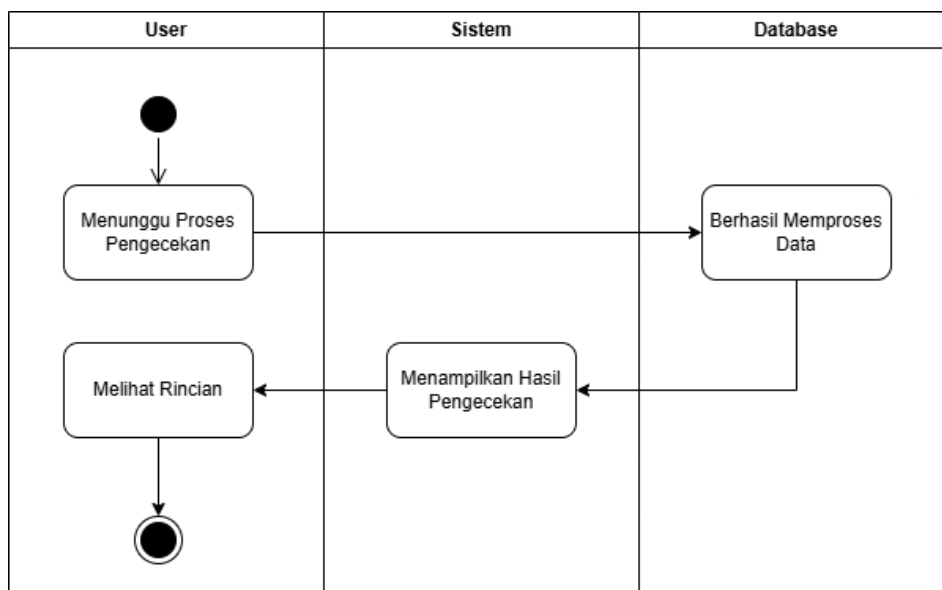


Gambar 6. Activity Diagram Unggah File.

Activity Diagram pada gambar 5 menggambarkan alur proses pengunggahan file dalam aplikasi pengecekan daftar pustaka. Proses dimulai ketika Pengguna membuka halaman web aplikasi. Setelah halaman berhasil ditampilkan, pengguna kemudian mengunggah file dokumen yang ingin diperiksa. File yang diunggah ini, biasanya dalam format *.docx*, berisi teks dan daftar pustaka yang perlu diverifikasi kesesuaiannya.

Begitu file diunggah, Sistem menerima file tersebut dan memulai pemrosesan data yang ada di dalamnya. Proses ini mencakup pengecekan untuk memastikan bahwa file tersebut valid dan siap untuk dianalisis lebih lanjut. Selanjutnya, basis data akan memproses data file yang telah diunggah, menyimpan informasi tentang file, serta mungkin mencatat hasil pengecekan atau kesalahan yang ditemukan.

2. Activity Diagram Tampilkan Hasil Pengecekan



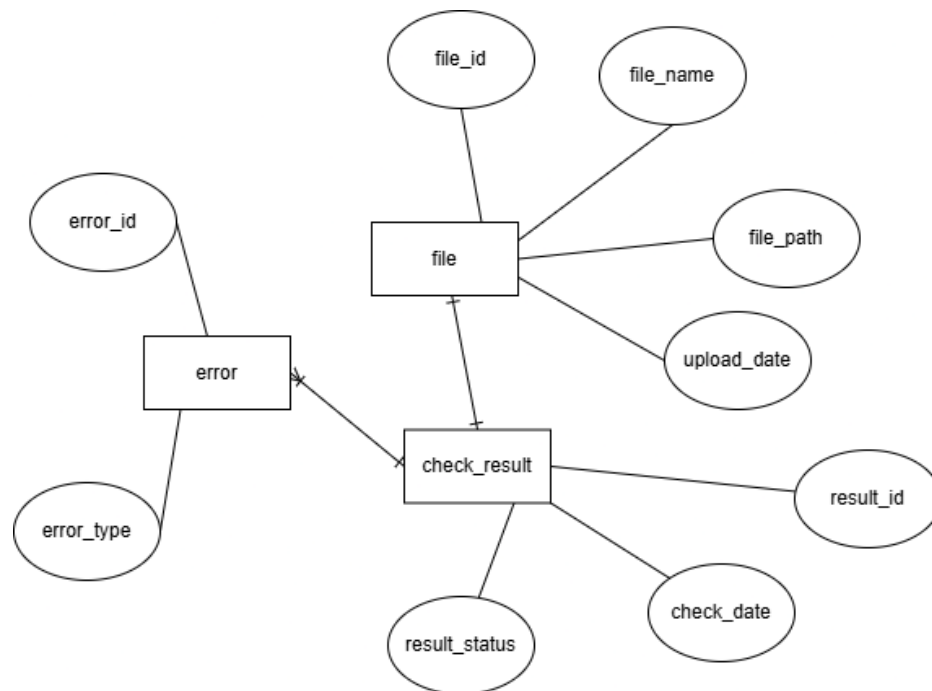
Gambar 7. Activity Diagram Tampilkan Hasil Pengecekan.

Activity Diagram pada gambar 6 menggambarkan alur proses setelah file diunggah dan diproses dalam aplikasi pengecekan daftar pustaka, khususnya pada tahap menampilkan hasil pengecekan. Proses dimulai dengan Pengguna yang menunggu proses pengecekan selesai. Setelah file diproses oleh sistem, pengguna harus menunggu hingga proses pengecekan selesai dan hasilnya siap untuk ditampilkan.

Setelah sistem selesai memproses data, Basis data akan mencatat hasil dari proses pengecekan tersebut. Kemudian, Sistem akan menampilkan hasil pengecekan kepada pengguna. Hasil ini akan menunjukkan apakah sitasi dan referensi dalam file tersebut sesuai dengan format yang diinginkan, serta mencantumkan kesalahan yang ditemukan jika ada.

Selanjutnya, Pengguna memiliki dapat melihat rincian kesalahan yang ditemukan dalam pengecekan tersebut. Ini memungkinkan pengguna untuk memeriksa kesalahan yang terdeteksi secara lebih detail, seperti kesalahan nama penulis, tahun publikasi, atau elemen lainnya dalam daftar pustaka. Diagram ini menunjukkan bagaimana pengguna dapat dengan mudah mengakses hasil pengecekan dan rincian kesalahan yang ditemukan, memberikan gambaran yang jelas mengenai status file pengguna.

3.3.4.2 ERD



Gambar 8. *Entity Relationship Diagram.*

Diagram *Entity-Relationship Diagram* (ERD) pada gambar 7 menggambarkan struktur basis data untuk aplikasi pengecekan daftar pustaka yang akan dikembangkan. Terdapat tiga entitas utama dalam sistem ini: *File*, *Check Result*, dan *Error*.

Struktur basis data untuk aplikasi ini dirancang secara fisik menggunakan *Entity-Relationship Diagram* (ERD) yang terdiri dari tiga tabel utama: *file*, *check_result*, dan *error*. Tabel *file* merepresentasikan data master untuk setiap dokumen yang diunggah, dengan atribut meliputi *file_id* sebagai *primary key*, *file_name*, *file_path*, dan *upload_date*. Selanjutnya, tabel *check_result* berfungsi untuk menyimpan riwayat dan status dari setiap proses pengecekan. Tabel ini memiliki *result_id* sebagai *primary key* dan *file_id* sebagai *foreign key*. Terakhir, rincian setiap kesalahan yang ditemukan akan disimpan dalam tabel *error*, yang memiliki *error_id* sebagai *primary key* dan *result_id* sebagai *foreign key*.

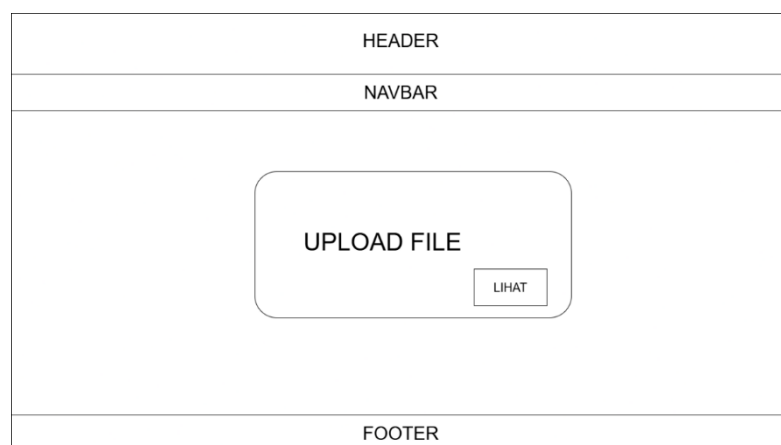
Relasi antar tabel tersebut dirancang untuk mengakomodasi alur kerja sistem. Hubungan antara tabel *file* dan *check_result* bersifat *one-to-many*, sebuah desain yang memungkinkan satu file yang sama dapat diperiksa beberapa kali, misalnya

setelah pengguna melakukan revisi, sehingga seluruh riwayat pengecekannya dapat tersimpan. Demikian pula, relasi antara tabel *check_result* dan error juga bersifat *one-to-many*, karena satu proses pengecekan sangat mungkin menemukan lebih dari satu jenis kesalahan dalam dokumen yang sedang divalidasi.

3.3.4.3 Wireframe

Wireframe adalah kerangka dasar visual (rancangan visual fidelitas rendah) yang merepresentasikan struktur tata letak sebuah antarmuka pengguna (UI). Tujuan dari pembuatan *wireframe* adalah untuk merencanakan penempatan elemen-elemen penting, seperti tombol, area teks, dan navigasi, tanpa terfokus pada detail desain grafis seperti warna atau tipografi. Berikut adalah rancangan *wireframe* untuk halaman-halaman utama pada aplikasi pengecekan daftar pustaka ini.

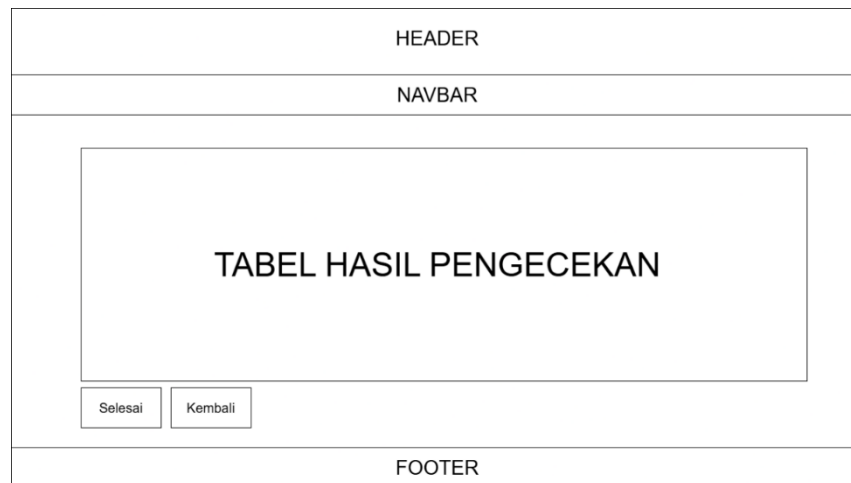
1. Wireframe Upload



Gambar 9. *Wireframe Upload.*

Wireframe ini memiliki struktur halaman yang standar, terdiri dari *Header*, *Navbar* untuk navigasi, dan *Footer*. Pada bagian konten utama, terdapat satu area fungsional yang besar untuk proses "*UPLOAD FILE*", di mana pengguna dapat memilih dan mengunggah dokumen berformat *.docx* yang akan diperiksa. Setelah file dipilih, pengguna menekan tombol "*LIHAT*" untuk melanjutkan ke proses verifikasi dan melihat hasilnya.

2. Wireframe Hasil



Gambar 10. Wireframe Hasil.

Halaman ini mempertahankan konsistensi tata letak dengan adanya *Header*, *Navbar*, dan *Footer*. Area konten utama didedikasikan untuk menampilkan "TABEL HASIL PENGECEKAN" yang akan berisi rincian kesesuaian antara sitasi dalam naskah dengan daftar pustaka. Di bagian bawah area konten, terdapat dua tombol navigasi: tombol "Selesai" untuk mengakhiri sesi pengecekan dan tombol "Kembali" yang memungkinkan pengguna untuk kembali ke halaman unggah sebelumnya.

3.3.5 Implementasi

Pada tahap implementasi atau pengkodean, nantinya modul pengecekan daftar pustaka ini akan dibangun menggunakan *framework* Codeigniter 4 (CI4). Codeigniter 4 dipilih karena merupakan *framework* PHP yang ringan, efisien, dan mudah digunakan, serta mendukung pengembangan aplikasi web yang cepat dan fleksibel. *Framework* ini mengikuti arsitektur *Model-View-Controller* (MVC), yang memungkinkan pemisahan logika aplikasi, tampilan antarmuka, dan pengelolaan data secara jelas dan terstruktur.

Dalam implementasi ini, Model akan digunakan untuk mengelola data yang terkait dengan file yang diunggah, hasil pengecekan, dan rincian kesalahan yang ditemukan. View akan menangani antarmuka pengguna, menampilkan halaman unggah file, hasil pengecekan, dan rincian kesalahan dengan desain yang responsif.

Controller akan berfungsi sebagai penghubung antara model dan view, mengontrol alur data antara keduanya serta mengatur proses pengecekan dan tampilan hasilnya.

Penggunaan CI4 juga memungkinkan integrasi dengan berbagai alat dan pustaka lainnya yang diperlukan dalam aplikasi ini, seperti library untuk memproses file *.docx*, serta basis data untuk menyimpan data hasil pengecekan dan rincian kesalahan. Dengan CI4, proses pengkodean dapat dilakukan secara modular, memastikan bahwa setiap komponen aplikasi dapat dikembangkan dan dipelihara dengan mudah. Implementasi ini akan memastikan bahwa aplikasi dapat berfungsi dengan efisien dan memberikan pengalaman pengguna yang optimal.

3.3.6 Pengujian

Tahap pengujian merupakan fase krusial untuk mengukur performa dan akurasi fungsional dari modul pengecekan daftar pustaka yang dikembangkan. Pada penelitian ini, metodologi pengujian yang diterapkan adalah *Data-Driven Testing* (DDT), sebuah pendekatan yang secara efektif memisahkan logika skrip pengujian dari data uji. Secara praktis, pengujian akan memanfaatkan 25 dokumen skripsi sebagai sumber data eksternal yang beragam. Setiap dokumen akan diproses oleh sistem, dan hasilnya akan dibandingkan secara sistematis dengan 'kunci jawaban' (*ground truth*) yang telah disiapkan melalui verifikasi manual untuk mengevaluasi kemampuan sistem dalam mengklasifikasikan setiap sitasi dan referensi sebagai "sesuai" atau "bermasalah".

Untuk mengevaluasi kinerja sistem secara kuantitatif, penelitian ini mengadopsi *Confusion Matrix* sebagai kerangka pengukuran. Hasil perbandingan antara deteksi sistem dan *ground truth* akan diklasifikasikan ke dalam empat kategori: *True Positive* (TP), di mana sistem benar mendeteksi masalah; *True Negative* (TN), di mana sistem benar mengidentifikasi data yang sesuai; *False Positive* (FP), saat sistem keliru menandai data yang benar sebagai masalah; dan *False Negative* (FN), saat sistem gagal mendeteksi masalah yang seharusnya ada. Dari keempat komponen ini, tingkat akurasi sistem akan dihitung menggunakan rumus: Akurasi = $(TP + TN) / \text{Total Item}$.

Perlu dicatat bahwa ruang lingkup pengujian pada penelitian ini sengaja tidak mencakup *User Acceptance Testing* (UAT). Keputusan ini diambil karena fokus utama penelitian adalah untuk memvalidasi akurasi dan fungsionalitas inti (*core functionality*) dari algoritma pengecekan, bukan pada aspek pengalaman pengguna (*user experience*) atau kegunaan antarmuka (*usability*). Sementara UAT lebih berorientasi pada evaluasi subjektif dari sisi pengguna akhir, pendekatan DDT dengan *Confusion Matrix* memberikan tolok ukur yang objektif dan terukur secara kuantitatif terhadap kapabilitas sistem. Dengan demikian, pengujian ini diprioritaskan untuk membuktikan bahwa sistem dapat menjalankan tugas utamanya dengan andal.

3.3.7 Pemeliharaan

Pada tahap pemeliharaan, aspek penting akan dilakukan untuk memastikan bahwa aplikasi pengecekan daftar pustaka tetap berfungsi dengan baik dan memberikan pengalaman pengguna yang optimal. Salah satu aspek utama adalah perbaikan bug dan masalah teknis yang mungkin muncul setelah aplikasi digunakan. Meskipun aplikasi telah diuji sebelumnya, kemungkinan adanya masalah teknis atau bug tetap ada, terutama setelah aplikasi digunakan dalam situasi yang lebih luas. Oleh karena itu, perbaikan bug secara berkala akan dilakukan untuk menjaga agar aplikasi tetap stabil dan berfungsi dengan baik.

Selain itu, peningkatan kinerja juga akan menjadi bagian penting dari pemeliharaan. Seiring berjalannya waktu, dan dengan meningkatnya jumlah pengguna serta file yang diproses, sistem perlu dioptimalkan untuk menjaga agar waktu pemrosesan tetap cepat dan efisien. Ini termasuk melakukan penyesuaian untuk memastikan bahwa aplikasi dapat menangani volume data yang lebih besar tanpa menurunkan kualitas atau kecepatan layanan.

Pemeliharaan juga akan mencakup pemantauan sistem secara berkala untuk memastikan bahwa aplikasi tetap berjalan dengan lancar. Pemantauan ini melibatkan pengecekan terhadap ketersediaan server, pemeliharaan basis data, serta identifikasi dan penanganan kesalahan sistem secara real-time. Dengan melakukan

pemantauan yang tepat, masalah dapat dideteksi lebih awal, dan solusi dapat diterapkan dengan cepat sebelum mempengaruhi pengalaman pengguna.

V. SIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian, dapat disimpulkan bahwa modul pengecekan daftar pustaka telah berhasil dikembangkan dan mampu memverifikasi kesesuaian sitasi dengan baik. Melalui pengujian fungsionalitas menggunakan metode *Data-Driven Testing* (DDT) terhadap 25 dokumen, modul ini menunjukkan performa yang andal dengan mencapai rata-rata akurasi sebesar 80,01%. Salah satu temuan terpenting adalah kemampuan sistem untuk mencapai nilai *False Negative* (FN) nol, yang membuktikan bahwa modul berhasil mengidentifikasi seluruh kesalahan referensi yang ada pada data uji tanpa ada yang terlewat.

Selain itu, analisis efisiensi juga menyimpulkan bahwa modul yang dikembangkan secara signifikan lebih unggul dibandingkan proses manual. Hasil pengujian perbandingan waktu menunjukkan bahwa pengecekan manual membutuhkan rata-rata 1442,7 detik (sekitar 24 menit) per dokumen, sementara modul mampu menyelesaikan tugas yang sama dalam waktu rata-rata hanya 62,9 detik (sekitar 1 menit). Perbedaan yang drastis ini membuktikan bahwa modul sangat efisien dan berhasil memenuhi tujuannya dalam membantu mempercepat proses pemeriksaan kesesuaian sitasi dan referensi secara efektif.

5.2 Saran

Berdasarkan hasil penelitian terdapat saran untuk pengembangan di masa depan yakni penambahan dukungan untuk gaya sitasi lain. Disarankan untuk pengembangan selanjutnya, fungsionalitas modul dapat diperluas untuk mendukung gaya sitasi lain yang umum digunakan di dunia akademik, seperti Chicago, MLA, atau IEEE.

DAFTAR PUSTAKA

- Afiifah, K. ', Fira Azzahra, Z., Anggoro, A. D., Redaksi, D., Akhir, R., & Online, D. (2022). Analisis Teknik Entity-Relationship Diagram dalam Perancangan Database: Sebuah Literature Review. *JURNAL INTECH*, 3(1), 8–11.
- Alif Ramadhan, J., Tresya Haniva, D., & Suharso, A. (2023). Systematic Literature Review Penggunaan Metodologi Pengembangan Sistem Informasi Waterfall, Agile, dan Hybrid. In *Journal Information Engineering and Educational Technology* (Vol. 07).
- Arisetywan, K., Fajar Prastiwi, L., Lika, E., Kiha, E. K., Rado, G., Studi, P., Ekonomi, I., Ekonomi, F., Bisnis, D., Surabaya, U. N., Program,), Pembangunan, S. E., Malang, U. N., Timor, U., & Manajemen, S. (2022). PELATIHAN PEMBUATAN DAFTAR PUSTAKA PADA KARYA ILMIAH MAHASISWA MENGGUNAKAN MS. WORD DAN MENDELEY. *Communnity Development Journal*, 3(2).
- Arista Kusuma, L. (2025). PERANCANGAN SISTEM LAYANAN PENGADUAN MASYARAKAT PADA KANTOR PERTANAHAN KABUPATEN ASAHAN. *Jurnal Riset Multidisplin*, 2. <https://journal.hasbaedukasi.co.id/index.php/jurmie>
- Mardiati, D., & Saputra, Y. (2025). IMPLEMENTASI SISTEM INFORMASI MANAJEMEN KLINIK MENGGUNAKAN METODE BLACK BOX TESTING. *Jurnal Informatika Dan Teknik Elektro Terapan*, 13(1). <https://doi.org/10.23960/jitet.v13i1.6015>
- Ramdany, S. W., Aulia Kaidar, S., Aguchino, B., Amelia, C., Putri, A., & Anggie, R. (2024). Penerapan UML Class Diagram dalam Perancangan Sistem Informasi Perpustakaan Berbasis Web. *Journal of Industrial and Engineering System*, 5(1), 30–41.
- Rininda, G., Santi, I. H., & Kirom, S. (2023). PENERAPAN SVM DALAM ANALISIS SENTIMEN PADA EDLINK MENGGUNAKAN PENGUJIAN CONFUSION MATRIX. In *Jurnal Mahasiswa Teknik Informatika* (Vol. 7, Issue 5).
- Sakethi, D., & Kartika, R. (2022). Utility Klasifikasi Kata Berdasarkan Kamus pada Dokumen Bertipe Word (.docx) Berbasis Web. *Jurnal Pepadun*, 3, 390–398. <https://doi.org/10.23960/pepadun.v3i3.137>

- Salim, I., Muzakir, A., & Syaputra, H. (2020). Perangkat Lunak Koreksi Penggunaan Sumberkutipan Dan Daftar Pustaka Pada Jurnal Mahasiswa. In *Jurnal Pengembangan Sistem Informasi dan Informatika* (Vol. 1, Issue 2). <http://journal.jis-institute.org/index.php/jpsii/index>
- Saputra, A., Sakethi, D., Tri Utami, Y., Ilmu Komputer FMIPA Universitas Lampung Jalan Sumantri Brojonegoro No, J., & Lampung, B. (2021). SISTEM PENDETEKSI PENULISAN KATA PADA DOKUMEN BERBASIS WEB. In *Jurnal Komputasi* (Vol. 9, Issue 1).
- Setyaningrum, M., Setyawati, E., Ary Setyawan, A., & Tinggi Ilmu Komputer Yos Sudarso Purwokerto, S. (2025). SISTEM INFORMASI PENJUALAN BERBASIS WEBSITE MENGGUNAKAN FRAMEWORK CODEIGNITER (STUDI KASUS: MSGLOW BANYUMAS). *Journal in Management and Entrepreneurship*, 4(2), 99–112.
- Suharni, Susilowati, E., & Pakusadewa, F. (2023). PERANCANGAN WEBSITE RUMAH MAKAN NINIK SEBAGAI MEDIA PROMOSI MENGGUNAKAN UNIFIED MODELLING LANGUAGE. *Jurnal Rekayasa Informasi*.
- Suparno, D., & Thamrin, H. (2021). *Pengembangan Keterampilan Penulisan Artikel Ilmiah Program Studi Tarjamah*.
- Triana, M. D., Naibaho, S. A., Sihite, S. H., & Lawolo, C. L. (2025). *Analisis Kritis Kesalahan Pengutipan Langsung “APA” Dalam Proposal Student Grant Mahasiswa: Identifikasi Masalah Dan Strategi Perbaikan*.