
**RANCANG BANGUN APLIKASI PENDETEKSI KESALAHAN STRUKTUR, FORMAT DAN EJAAN
KATA NASKAH SKRIPSI**

Jhildie Kolinu¹⁾, Jullia Titaley²⁾, Wisard Kalengkongan³⁾, Eliasta Ketaren⁴⁾

Program Studi Sistem Informasi

Universitas Sam Ratulangi

Jln. Kampus Bahu UNSRAT, Kecamatan Malalayang, Manado 95115, Sulawesi Utara

email: jhildiekolinu106@student.unsrat.ac.id¹⁾, july_titaley@unsrat.ac.id²⁾,

wisard.kalengkongan@unsrat.ac.id³⁾, eliasketaren@unsrat.ac.id⁴⁾

Abstrak

Penelitian ini bertujuan untuk merancang dan membangun aplikasi *web* bernama CekNulis yang dapat mengidentifikasi kesalahan penulisan pada naskah skripsi mahasiswa FMIPA UNSRAT, khususnya dalam hal struktur, format pengetikan, dan ejaan kata, sesuai dengan panduan penulisan FMIPA UNSRAT. Metode yang digunakan dalam pengembangan aplikasi ini adalah *Prototype* yang memanfaatkan teknik *Exact String Matching*, dan didukung oleh berbagai pustaka Python seperti Flask, Python-docx, Regex, dan SQLAlchemy. Hasil penelitian menunjukkan bahwa aplikasi CekNulis dapat mendeteksi kesalahan struktur, format pengetikan, dan ejaan kata pada file .docx menggunakan fitur deteksi skripsi.

Kata Kunci : *Aplikasi Web, Kesalahan Penulisan, Naskah Skripsi, Exact String Matching, Python.*

1. Pendahuluan

Skripsi merupakan dokumen wajib yang berisi rancangan dan hasil penelitian mahasiswa, sebagai salah satu syarat memperoleh gelar Strata satu (S1). Skripsi adalah bentuk tulisan ilmiah yang disusun dengan merujuk pada pemikiran para ahli, lalu dipadukan dengan pandangan penulis sendiri [9]. Penulisan skripsi tetap mengikuti aturan akademik yang berlaku dan pada akhirnya dipresentasikan dalam sidang sebagai bentuk pertanggungjawaban karya tersebut. Skripsi berperan sebagai sarana untuk mengukur seberapa dalam pemahaman teori yang dimiliki mahasiswa, sekaligus menjadi latihan nyata dalam melakukan penelitian dengan menerapkan ilmu yang telah dipelajari selama menjalani perkuliahan [1].

Dalam penyusunan naskah skripsi untuk mahasiswa dari masing-masing fakultas di setiap perguruan tinggi memiliki format dan struktur penulisan yang berbeda-beda, hal ini disebabkan oleh kurikulum dan kebijakan akademik dari masing-masing perguruan tinggi yang merepresentasikan karakteristik dari setiap fakultas. Setiap perguruan tinggi umumnya menyediakan panduan penulisan skripsi yang memuat aturan tentang struktur, format, serta tata cara penulisan yang wajib diikuti oleh mahasiswa saat menyusun naskah skripsi mereka [10].

Panduan penulisan skripsi untuk mahasiswa Fakultas Matematika dan Ilmu Pengetahuan Alam (FMIPA) Universitas Sam Ratulangi (UNSRAT) telah disusun sejak tahun 2012 oleh sejumlah dosen di lingkungan FMIPA. Dokumen panduan tersebut dapat diunduh melalui browser dengan mengetikkan kata kunci “Panduan Penulisan FMIPA UNSRAT” [11].

Dalam penyusunan naskah skripsi, tidak menutup kemungkinan untuk mahasiswa melakukan kesalahan dalam penulisan. Contoh seperti naskah proposal penelitian yang merupakan bagian dari bab satu sampai dengan bab tiga untuk naskah skripsi. Naskah proposal penelitian yang dibuat oleh beberapa mahasiswa FMIPA UNSRAT angkatan 2021 masih ditemukan beberapa kesalahan penulisan seperti struktur dan format pengetikan yang tidak sesuai dengan panduan penulisan FMIPA UNSRAT, serta kesalahan penulisan ejaan kata dalam isi naskah skripsi.

Pengembangan aplikasi saat ini, seperti aplikasi desktop, web dan mobile dapat difasilitasi dengan internet. Aplikasi berbasis web merupakan platform yang disediakan bagi pengguna untuk mengakses dan menggunakan aplikasi lewat internet. Aplikasi web memiliki beberapa kelebihan yang ditawarkan bagi pengguna, seperti kemudahan akses, pembaruan otomatis dan dapat diakses dari berbagai perangkat, seperti smartphone, komputer, laptop dan smartwatch.

Penggunaan aplikasi web melalui browser, selalu disertai dengan penggunaan internet dalam mengakses aplikasi tersebut secara digital (online). Perkembangan internet yang menghubungkan jutaan perangkat telah mengalami peningkatan dalam beberapa tahun terakhir. Pengguna internet diseluruh dunia mencapai jumlah 5,54 miliar pengguna pada awal Juli 2024 atau setara dengan 67,1% dari populasi manusia di dunia, hal ini menunjukkan bahwa internet telah menjadi bagian yang penting dalam kehidupan sehari-hari untuk memfasilitasi komunikasi dan informasi secara luas [3].

Berdasarkan penelitian terdahulu, peneliti mengambil metode *Exact String Matching* untuk mendeteksi dan mencocokkan teks yang tidak valid berdasarkan daftar kata valid yang tersimpan dalam database. Metode *Exact String Matching* sebelumnya telah diterapkan oleh Gunawan dan Sudarsono (2022) dalam penelitian mereka di SMAN 8 Kota Bengkulu, yang bertujuan untuk meningkatkan efisiensi pencarian data pada Sistem Informasi Akademik (SIA). Mereka mengembangkan sistem pencarian untuk mengenali pola teks berdasarkan kata kunci

yang dimasukkan ke dalam search bar. Pengembangan sistem dilakukan menggunakan pendekatan Rapid Application Development (RAD) dan dirancang dengan model Unified Modeling Language (UML). Hasil penelitian menunjukkan bahwa algoritma Exact String Matching dapat diterapkan ke dalam sistem untuk mempercepat proses pencarian data akademik seperti data siswa, guru, mata pelajaran, dan nilai [4].

Kemudian berdasarkan masalah yang ditemukan dari beberapa mahasiswa FMIPA UNSRAT angkatan 2021 mengenai struktur penulisan, format pengetikan, dan penulisan ejaan kata dalam naskah proposal penelitian, memunculkan solusi pembuatan aplikasi *web* yang bernama CekNulis sebagai pendeteksi kesalahan penulisan (struktur penomoran subbab, format pengetikan, dan ejaan kata) dari naskah skripsi, dengan menggunakan metode *Exact String Matching* dan beberapa pustaka (*library*) Python yang mendukung pengelolaan *file* Microsoft Word berformat .docx. Tujuan dari penelitian ini adalah untuk membantu mahasiswa FMIPA UNSRAT dalam proses penyusunan naskah skripsi yang sesuai dengan panduan penulisan yang berlaku di FMIPA UNSRAT. Aplikasi CekNulis diharapkan menjadi solusi mengenai penyusunan naskah skripsi bagi mahasiswa FMIPA UNSRAT, dalam mempersiapkan naskah skripsi.

2. Landasan Teori

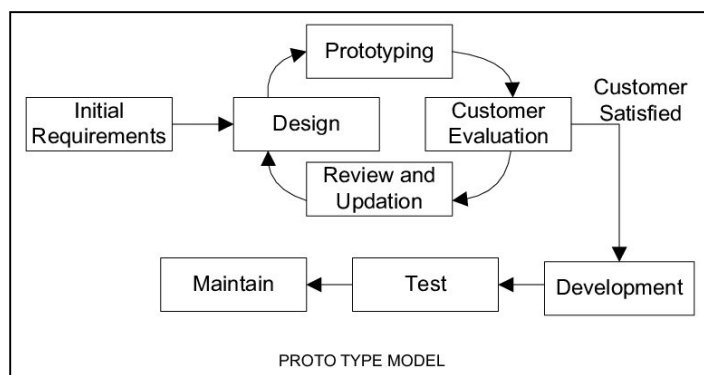
Skripsi

Skripsi merupakan karya ilmiah yang disusun oleh mahasiswa sebagai syarat kelulusan program sarjana (S1) di perguruan tinggi, baik negeri maupun swasta di Indonesia. Istilah "skripsi" berasal dari kata *scriptie* dalam bahasa Belanda, yang merujuk pada hasil penelitian tertulis dalam bidang keilmuan tertentu dengan mengikuti kaidah penulisan ilmiah yang berlaku. Penulisan skripsi bertujuan untuk mengukur kemampuan mahasiswa dalam mengintegrasikan pengetahuan teoritis dan keterampilan berpikir kritis dalam membahas suatu topik sesuai dengan bidang studinya [13].

Berdasarkan penjelasan mengenai skripsi, dapat disimpulkan bahwa skripsi memegang peranan penting dalam pendidikan perguruan tinggi di Indonesia. Skripsi berfungsi sebagai sarana untuk melatih dan menguji pengetahuan serta keterampilan yang telah diperoleh mahasiswa selama masa perkuliahan, sehingga mahasiswa dituntut untuk berpikir kritis dan mampu mengintegrasikan pemahaman teoritis dalam menganalisis suatu permasalahan atau topik penelitian sesuai dengan bidang keilmuannya.

Prototype

Prototype (Purwarupa) adalah representasi awal suatu produk atau sistem yang digunakan untuk menguji konsep desain sebelum implementasi. Dalam pengembangan perangkat lunak dan desain antarmuka, *Prototype* memungkinkan desainer dan pengembang untuk menciptakan versi awal aplikasi yang dapat diuji oleh pengguna, sehingga pengguna dapat memberikan umpan balik (*feedback*). *Prototype* berfungsi metode pengembangan perangkat lunak untuk para desainer dan pengembang dalam memvisualisasikan ide dan fitur-fitur yang akan diterapkan ke dalam aplikasi [8]. Dalam metode pengembangan *Prototype*, terdapat dua jenis *Prototype* yaitu *Prototype Low-Fidelity* dan *High-Fidelity*. *Prototype Low-Fidelity*, seperti sketsa atau *Wireframe*, digunakan pada tahap awal untuk memberikan gambaran dasar terhadap desain aplikasi tanpa terlalu banyak hal yang mendetail. Sementara itu, *Prototype High-Fidelity* bersifat interaktif dan lebih mendekati produk akhir dari aplikasi, sehingga umum digunakan untuk menguji alur kerja sistem dan pengalaman pengguna secara lebih spesifik.



Gambar 1. *Prototype Model* (Letsoin, 2017)

Exact String Matching

String Matching merupakan teknik dasar yang sangat penting dalam berbagai aplikasi komputasi seperti penyuntingan teks, pencarian informasi, hingga analisis urutan biologis. Pencocokan *string* bertujuan untuk menemukan semua posisi dalam sebuah teks di mana suatu pola muncul secara identik. Baik pola maupun teks direpresentasikan sebagai deretan karakter, dan pencocokan dikatakan terjadi apabila seluruh karakter dalam pola sesuai dengan bagian teks pada posisi tertentu. Pendekatan ini dikenal sebagai *Exact String Matching*, yaitu metode yang tidak mentoleransi perbedaan karakter antara pola dan teks. Salah satu metode paling mendasar dalam metode *Exact String Matching* adalah algoritma Brute Force, yang melakukan perbandingan karakter secara berurutan dari kiri ke kanan untuk setiap kemungkinan pergeseran posisi dalam teks. Meskipun memiliki kompleksitas waktu $O((n - m + 1) \times m)$ dengan keterangan $O = \text{Big-O}$ untuk menggambarkan kompleksitas

waktu suatu algoritma, n = panjang teks dan m = panjang pola. Pendekatan Brute Force tetap digunakan sebagai pondasi awal dalam pengembangan algoritma pencocokan *string* lainnya karena kesederhanaan dan kejelasan dari langkah-langkah kerjanya [2].

Berikut ini adalah langkah-langkah kerja dari algoritma Brute Force untuk melakukan pencocokan *string*:

1. Proses pencocokan dimulai dengan teknik Brute Force di bagian paling kiri teks.
Dimulai dari kiri ke kanan, karakter dalam pola dibandingkan satu per satu dengan karakter dalam teks hingga salah satu keadaan berikut terwujud:
 - a. Karakter pola dan karakter teks tidak sesuai (*mismatch*).
 - b. Pencocokan dianggap berhasil atau sesuai jika setiap karakter dalam pola cocok dengan bagian teks.
2. Algoritma *Brute Force* mengulang langkah kedua hingga lokasi pola mencapai akhir teks setelah menggeser pola satu tempat ke kanan setelah pencocokan di satu posisi selesai.



Gambar 2. Proses Pencocokan *String* dengan Algoritma *Brute Force*

Berdasarkan Gambar 2, berikut adalah perhitungan menggunakan kompleksitas waktu $O((n - m + 1) \times m)$, untuk mencari tahu berapa banyak operasi perbandingan karakter yang terjadi, selama dilakukan proses pencarian pola yang cocok dengan teks:

Diketahui:

1. Teks (T): G C A T C G C A G A G A G T A T A C A G T A C G Panjang $n = 24$
2. Pola (P): G C A G A G A G Panjang $m = 8$

Maka, jumlah pergeseran yang terjadi untuk mencari pola yang cocok dengan teks adalah sebanyak:

$$n - m + 1 = 24 - 8 + 1 = 17 \text{ kali}$$

Kemudian, untuk menghitung maksimal perbandingan karakter adalah sebagai berikut:

$$O((n - m + 1) \times m) = (24 - 8 + 1) \times 8 = 17 \times 8 = 136 \text{ perbandingan karakter}$$

Penggunaan teknik *Exact String Matching* penelitian ini bertujuan untuk membandingkan seluruh teks pada naskah skripsi dengan teks valid yang terdapat dalam *database* yang berisikan KBBI (Kamus Besar Bahasa Indonesia) dan OPTED (*Online Plain Text English Dictionary*). Langkah ini dilakukan untuk mengidentifikasi kata-kata yang tidak valid, baik dalam bahasa Indonesia maupun bahasa Inggris.

Flask

Flask merupakan kerangka kerja mikro (*micro-framework*) berbasis *web* dalam bahasa Python. Flask diciptakan untuk memudahkan pengembangan aplikasi *web* tanpa pustaka tambahan dan tetap mendukung aplikasi berskala besar, contoh seperti aplikasi Pinterest dan LinkedIn melalui berbagai ekstensi, seperti integrasi basis data, validasi formulir, dan penanganan unggahan [12].

Flask merupakan *framework web* berbasis Python yang tergolong sebagai *micro-framework* karena dirancang untuk memiliki inti yang sederhana namun tetap fleksibel. Istilah "*micro*" tidak berarti Flask memiliki keterbatasan fungsionalitas, melainkan memberikan kebebasan bagi pengembang untuk menentukan sendiri teknologi yang ingin digunakan, seperti sistem basis data atau validasi formulir. Flask tidak menyertakan abstraksi *database* atau sistem validasi secara *default*, namun memungkinkan penggunaan dengan berbagai ekstensi dapat diintegrasikan seolah-olah merupakan bagian asli dari

framework. Secara konvensional, struktur proyek Flask menyimpan *template* dan *file* statis di dalam *directory templates* dan *static*. Selain itu, Flask dirancang untuk mendukung pengembangan aplikasi dari skala kecil hingga besar, dengan menyediakan *hooks* dan *class* yang dapat dimodifikasi sesuai kebutuhan proyek. Salah satu fitur teknis penting dalam Flask adalah penggunaan *thread-local* yang memungkinkan pengelolaan objek dalam konteks permintaan (*request context*) tanpa perlu meneruskan objek antar fungsi, yang mendukung pengembangan aplikasi yang aman dan efisien. Meski Flask telah terlindungi dari serangan umum seperti XSS (*Cross-Site Scripting*) melalui mesin templating Jinja2, Ronacher juga menekankan pentingnya kesadaran terhadap aspek keamanan secara keseluruhan saat membangun aplikasi *web*, karena berbagai bentuk serangan tetap menjadi ancaman potensial, bahkan untuk aplikasi kecil [7].

Berdasarkan penjelasan tersebut, dapat disimpulkan bahwa Flask merupakan *framework web* berbasis Python yang mengutamakan kesederhanaan dan fleksibilitas dalam pengembangannya. Meskipun digolongkan sebagai *micro-framework*, Flask tetap mampu membangun aplikasi skala besar dengan bantuan berbagai ekstensi yang dapat diintegrasikan sesuai kebutuhan. Keunggulan utama Flask terletak pada kemudahan penggunaannya, struktur proyek yang konvensional, serta fitur teknis seperti *thread-local* dan dukungan terhadap keamanan dasar melalui templating Jinja2. Dengan memberikan kebebasan kepada pengembang dalam memilih teknologi yang digunakan, Flask menjadi pilihan yang ideal untuk membangun aplikasi *web* secara modular dan efisien, baik untuk proyek kecil maupun besar.

Black Box

Pengujian *Black Box* adalah metode pengujian perangkat lunak yang menilai fungsionalitas sistem berdasarkan *input* dan *output*, tanpa memeriksa struktur internal atau *source code* dari perangkat lunak. Dalam metode *Black Box*, penguji hanya melihat *input* yang diberikan dan *output* yang dihasilkan oleh perangkat lunak, sehingga penguji merancang skenario uji untuk menguji berbagai fungsi aplikasi, menjalankan pengujian, dan mengevaluasi hasil pengujian guna memastikan perangkat lunak sesuai dengan spesifikasi kebutuhan pengguna [6].

Metode pengujian *Black Box* merupakan pendekatan pengujian perangkat lunak yang dirancang tanpa melihat struktur internal dari kode program, melainkan berfokus pada spesifikasi dan fungsionalitas yang terlihat oleh pengguna. Metode ini dapat diterapkan pada berbagai tingkat pengujian dalam siklus hidup pengembangan perangkat lunak, mulai dari *Unit Testing*, *Integration Testing*, *System Testing*, hingga *Acceptance* dan *Maintenance Testing*. Tujuan utama dari *Black Box testing* adalah untuk memastikan bahwa perangkat lunak memenuhi kebutuhan pengguna berdasarkan masukan dan keluaran (*input/output*) yang telah ditentukan dalam dokumentasi spesifikasi. Beberapa teknik yang umum digunakan dalam *Black Box Testing* antara lain *Equivalence Partitioning*, *Boundary Value Analysis*, *Syntax Testing*, dan *Use Case Testing*. Meskipun dirancang untuk pengujian dinamis, metode *Black Box* juga dapat digunakan dalam konteks pengujian statis untuk memvalidasi spesifikasi *input* dan *output*. Metode ini sering dikombinasikan dengan pendekatan *Grey-Box Testing*, yang menggunakan informasi tambahan tentang *source code* untuk meningkatkan akurasi dan efektivitas. Dengan demikian, metode *Black Box* memberikan fleksibilitas tinggi dalam mendeteksi kegagalan sistem dari sudut pandang pengguna tanpa perlu terlibat langsung dengan struktur internal perangkat lunak [5].

Berdasarkan penjelasan tersebut, dapat disimpulkan bahwa pengujian *Black Box* merupakan metode yang efektif untuk menilai kesesuaian fungsi perangkat lunak terhadap kebutuhan pengguna tanpa harus memahami atau mengevaluasi struktur internal dari kode program. Dengan berfokus pada *input* dan *output*, metode *Black Box* memungkinkan penguji untuk merancang skenario pengujian secara objektif dan menyeluruh berdasarkan spesifikasi fungsional. Selain itu, fleksibilitasnya dalam diterapkan pada berbagai level pengujian serta kemampuannya untuk dikombinasikan dengan pendekatan lain seperti *Grey-Box*, menjadikan *Black Box Testing* sebagai salah satu metode penting dalam memastikan kualitas dan keandalan perangkat lunak dari perspektif pengguna.

3. Metode Penelitian

Lokasi dan Waktu Penelitian

Penelitian dilakukan di Fakultas Matematika dan Ilmu Pengetahuan Alam (FMIPA), Universitas Sam Ratulangi (UNSRAT). Lokasi ini dipilih karena beberapa mahasiswa FMIPA UNSRAT dalam membuat naskah proposal penelitian yang merupakan bagian dari bab satu sampai dengan bab tiga untuk naskah skripsi, masih saja ditemukan beberapa kesalahan dalam penulisan naskah, seperti struktur penulisan, format pengetikan, dan penulisan ejaan kata. Sehingga dibutuhkan solusi berupa aplikasi yang dapat membantu mahasiswa FMIPA UNSRAT dalam penulisan naskah skripsi.

Penelitian dimulai pada bulan Oktober 2024, diawali dengan tahapan identifikasi masalah yang mencakup pengumpulan informasi terkait kesalahan struktur, format dan ejaan kata dalam penulisan naskah proposal penelitian mahasiswa FMIPA UNSRAT angkatan 2021 yang sudah melakukan ujian seminar usul penelitian.

Data Penelitian

Data yang digunakan dalam penelitian ini adalah data primer (cetakan naskah proposal penelitian) dan data sekunder (dokumen panduan penulisan FMIPA UNSRAT tahun 2012, Kamus Besar Bahasa Indonesia dan Online Plain Text English Dictionary). Data primer diperoleh dari empat mahasiswa tingkat akhir program studi

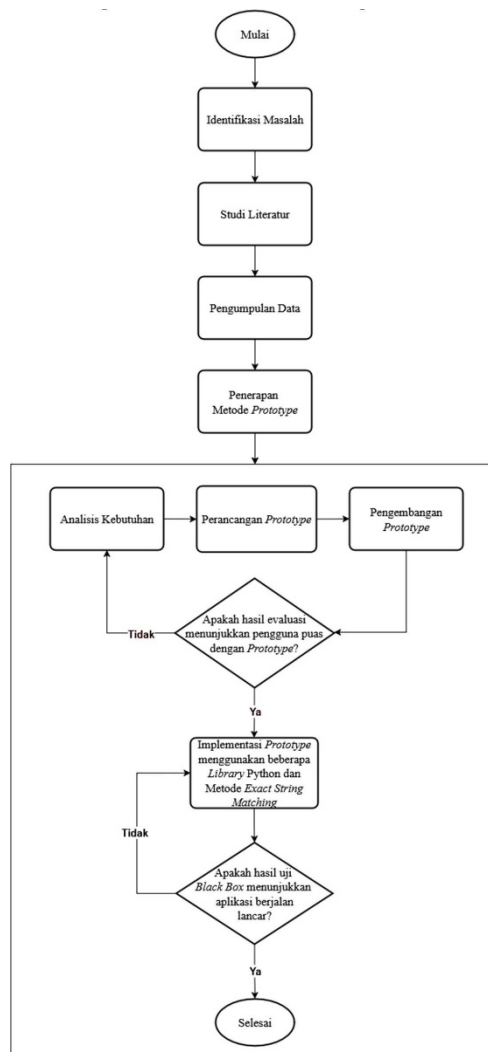
Sistem Informasi FMIPA UNSRAT angkatan 2021 yang telah melakukan ujian seminar usul penelitian dalam bentuk hardcopy (cetakan naskah proposal penelitian), sehingga dapat dilakukan identifikasi dan pengumpulan informasi terkait kesalahan struktur penomoran subbab, format pengetikan, dan ejaan kata dalam penulisan naskah proposal penelitian. Data sekunder yang digunakan pada penelitian ini, yaitu dokumen panduan penulisan FMIPA UNSRAT tahun 2012, dokumen Kamus Besar Bahasa Indonesia (KBBI), dan dokumen Online Plain Text English Dictionary (OPTED). Dokumen panduan penulisan FMIPA UNSRAT tahun 2012, berisikan informasi mengenai persyaratan akademik dan administrasi, penilaian, format penulisan, aturan pengetikan dan informasi seputar panduan lainnya; dokumen KBBI berisi ribuan kata baku dalam bahasa Indonesia yang ditampung ke dalam file berformat .csv; dokumen OPTED berisi ribuan kata baku dalam bahasa Inggris yang ditampung ke dalam file berformat .csv.

Teknik Pengumpulan Data

Teknik pengumpulan data dalam penelitian ini dilakukan dengan memperoleh salinan cetak (*hardcopy*) naskah proposal penelitian dari mahasiswa tingkat akhir Program Studi Sistem Informasi FMIPA UNSRAT angkatan 2021 yang telah menyelesaikan ujian seminar usulan penelitian. Selain itu, data juga diperoleh dari Kamus Besar Bahasa Indonesia (KBBI) melalui situs resmi Kemdikbud (kemdikbud.go.id), serta dari *Online Plain Text English Dictionary* (OPTED) yang diunduh melalui situs Kaggle (kaggle.com).

Tahapan Penelitian

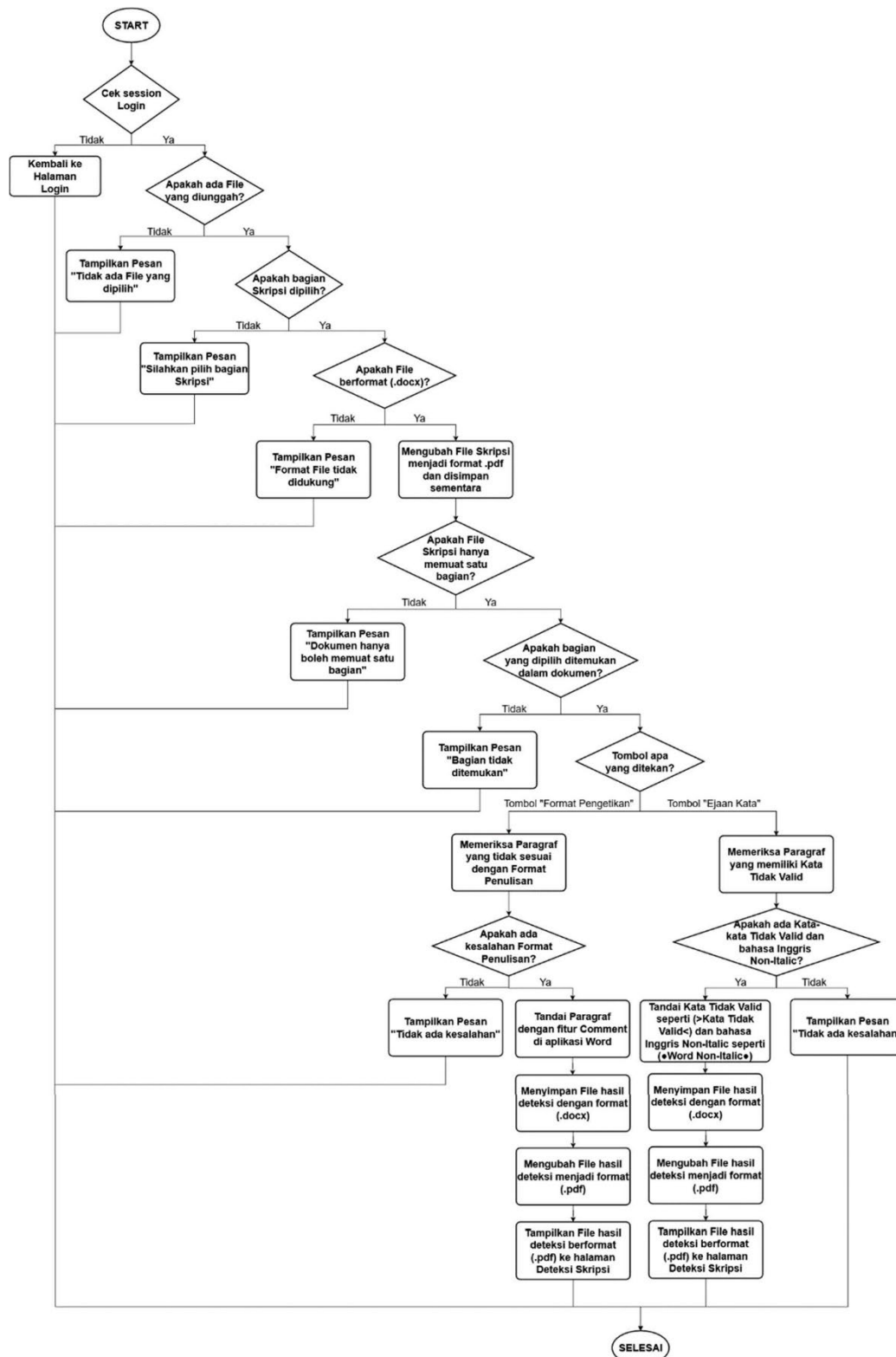
Berikut adalah tahapan penelitian yang dilakukan selama perancangan, serta pembuatan aplikasi pendeteksi kesalahan penulisan dalam naskah skripsi FMIPA UNSRAT, bisa dilihat pada Gambar 3.



Gambar 3. Diagram Alir Tahapan Penelitian

Diagram Alir (Flowchart)

Flowchart adalah diagram yang menggambarkan langkah-langkah kerja dalam suatu proses atau sistem dengan menggunakan simbol-simbol standar dari pembuatan *Flowchart*. Berikut adalah gambaran alur kerja dari sistem aplikasi CekNulis ketika sedang mendeteksi kesalahan penulisan dari *file* skripsi yang diunggah:



Gambar 4. Flowchart Sistem Deteksi Skripsi

Diagram Kasus Penggunaan (Use Case Diagram)

Use Case Diagram adalah ilustrasi visual tentang bagaimana pengguna (aktor) berinteraksi dengan sistem secara fungsional. Berikut adalah gambaran *Use Case Diagram* untuk aplikasi CekNulis:



Gambar 5. Use Case Diagram untuk Aplikasi CekNuklis

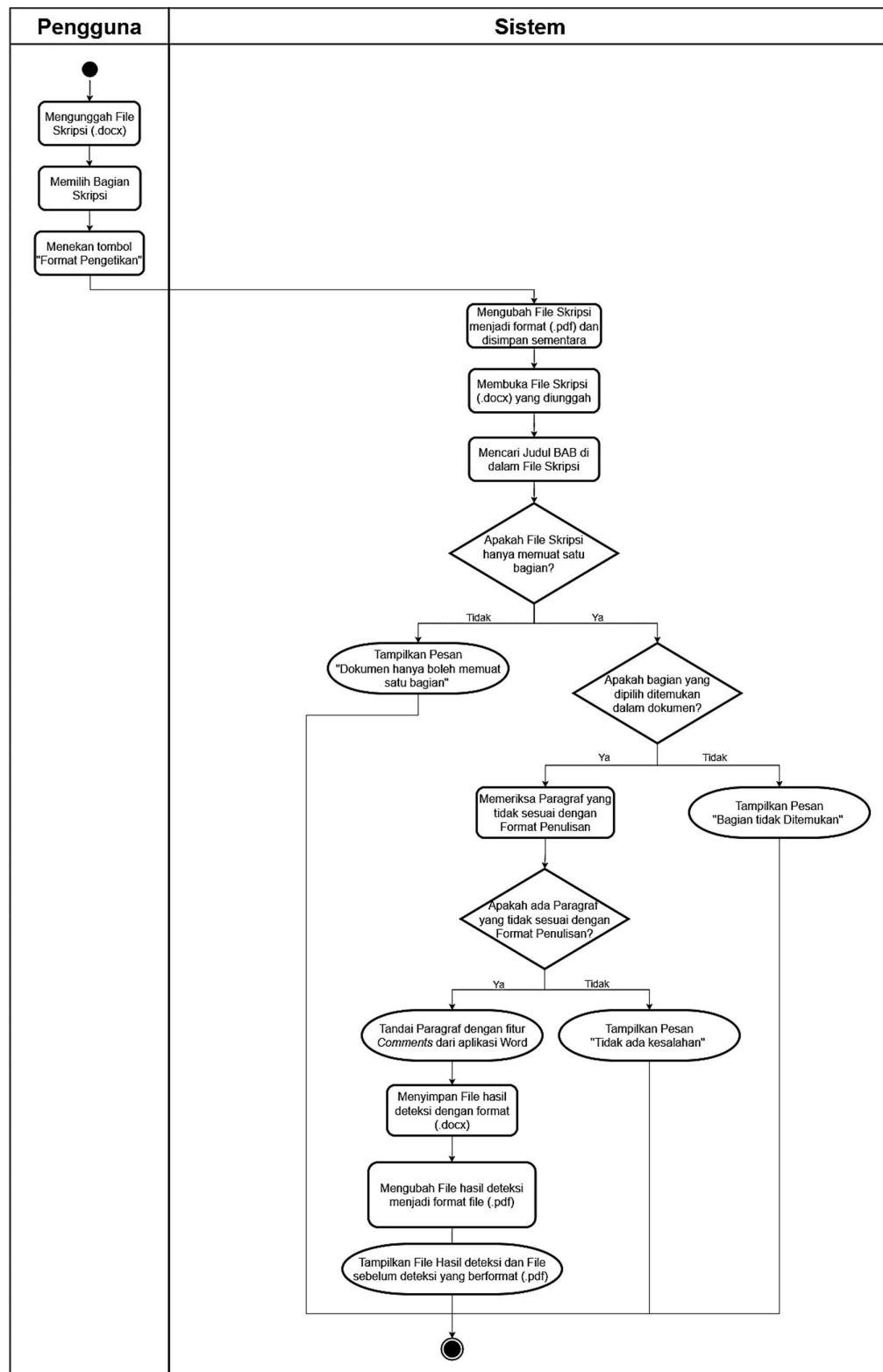
Diagram Aktivitas (Activity Diagram)

Activity Diagram adalah diagram dalam *Unified Modeling Language* yang digunakan untuk menggambarkan urutan aktivitas dalam suatu sistem atau proses. Berikut adalah *Activity Diagram* dari sistem deteksi kesalahan format pengetikan dan ejaan kata pada aplikasi CekNulis:

1. Diagram Aktivitas Deteksi Kesalahan Format Pengetikan

Berikut adalah diagram aktivitas yang menunjukkan proses pengguna saat mendeteksi kesalahan format pengetikan dalam *file* skripsi pada aplikasi CekNulis:

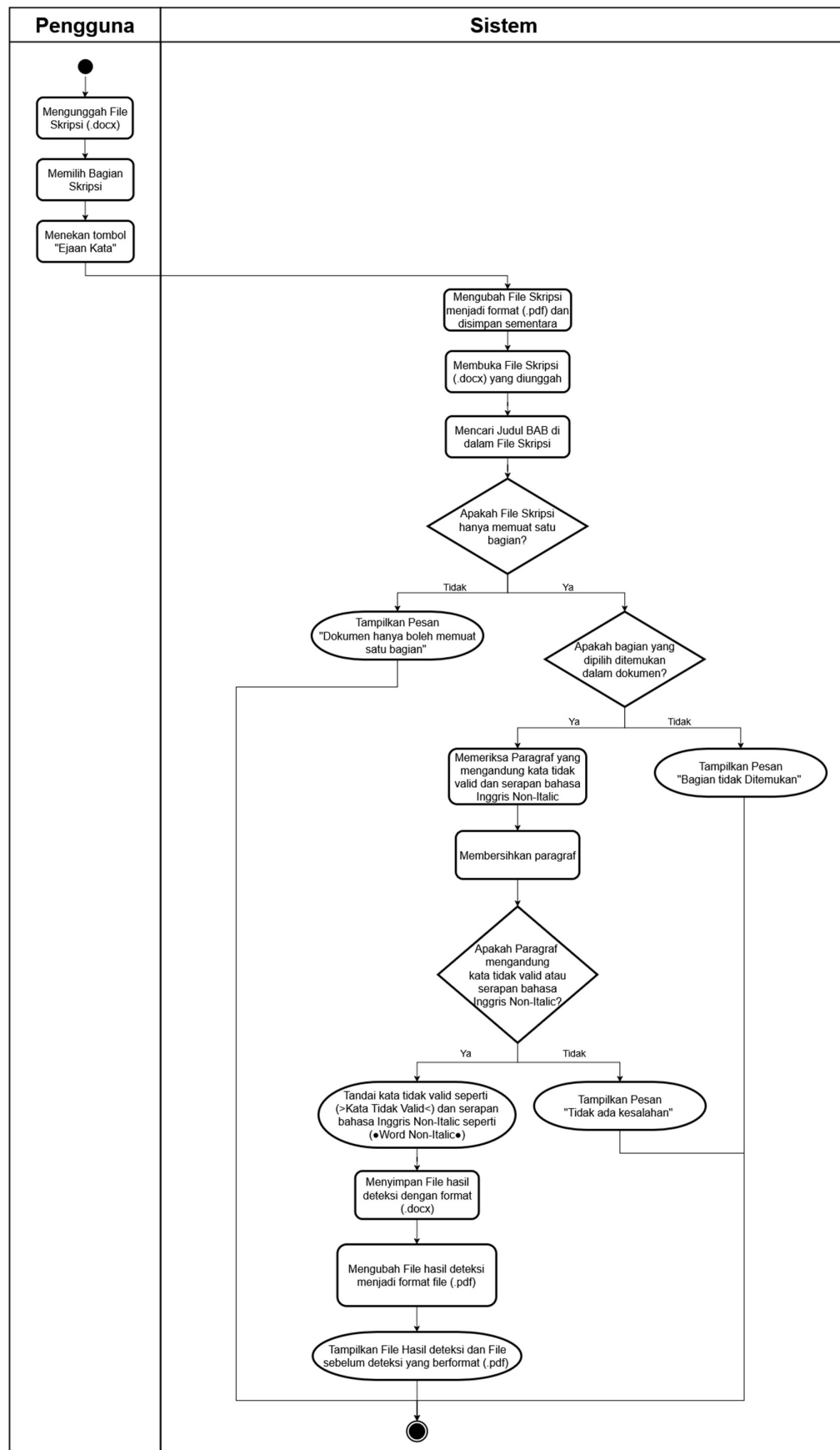
Tabel berikut ini adalah penerapan metode out by spirally.



Gambar 6. Activity Diagram untuk Mendeteksi Kesalahan Format Pengetikan

2. Diagram Aktivitas Deteksi Kata Tidak Valid

Berikut adalah diagram aktivitas yang menunjukkan proses pengguna saat mendeteksi kata tidak valid dalam file skripsi pada aplikasi CekNulis:



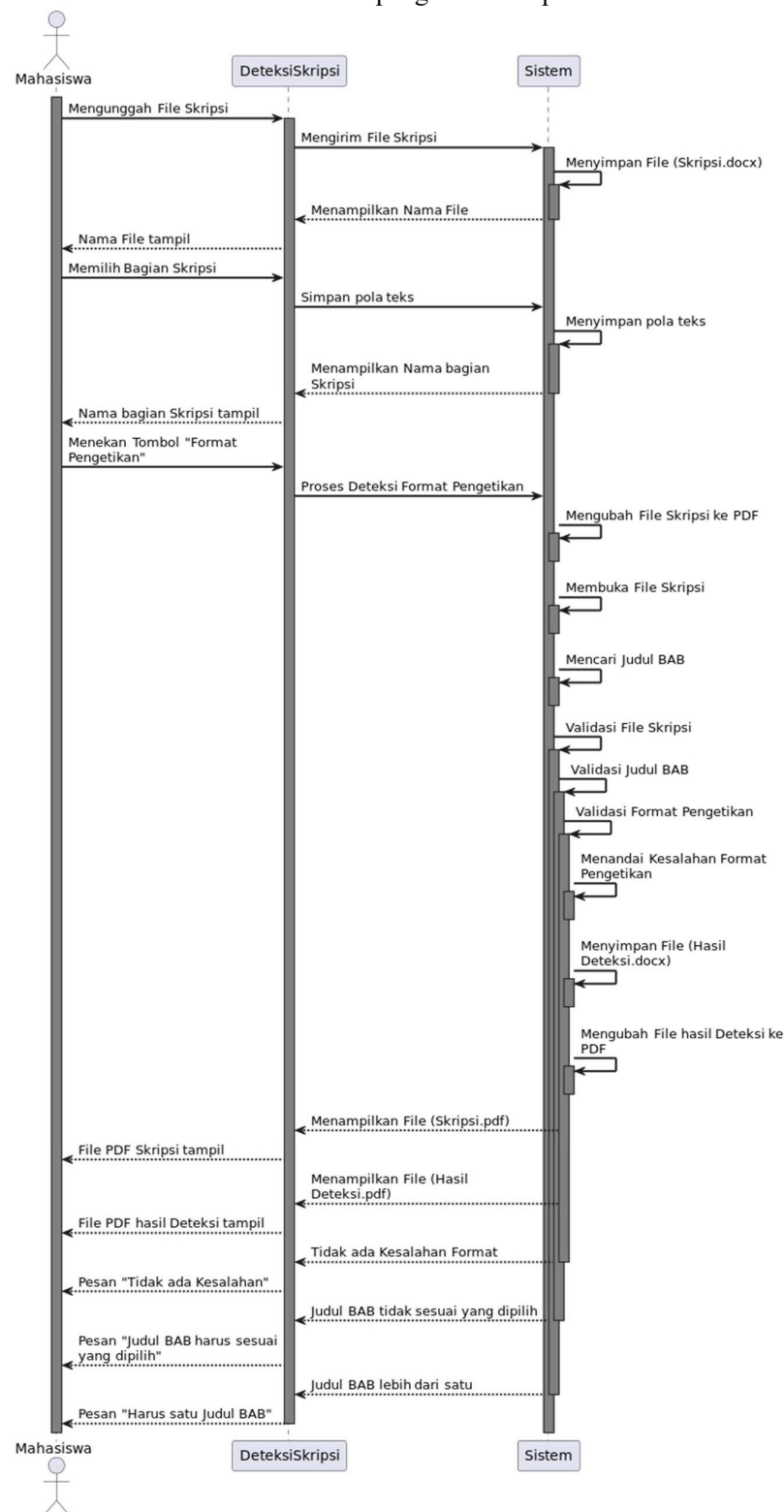
Gambar 7. Activity Diagram untuk Mendeteksi Kata Tidak Valid

Diagram Urutan (Sequence Diagram)

Sequence Diagram adalah diagram dalam *Unified Modeling Language* yang digunakan untuk menggambarkan interaksi antar objek dalam suatu sistem, yang disusun berdasarkan urutan waktu terjadinya proses pengiriman pesan dan perintah antar objek. Berikut adalah beberapa *Sequence Diagram* dari sistem deteksi kesalahan format pengetikan dan ejaan kata pada aplikasi CekNulis:

1. Diagram Urutan Deteksi Kesalahan Format Pengetikan

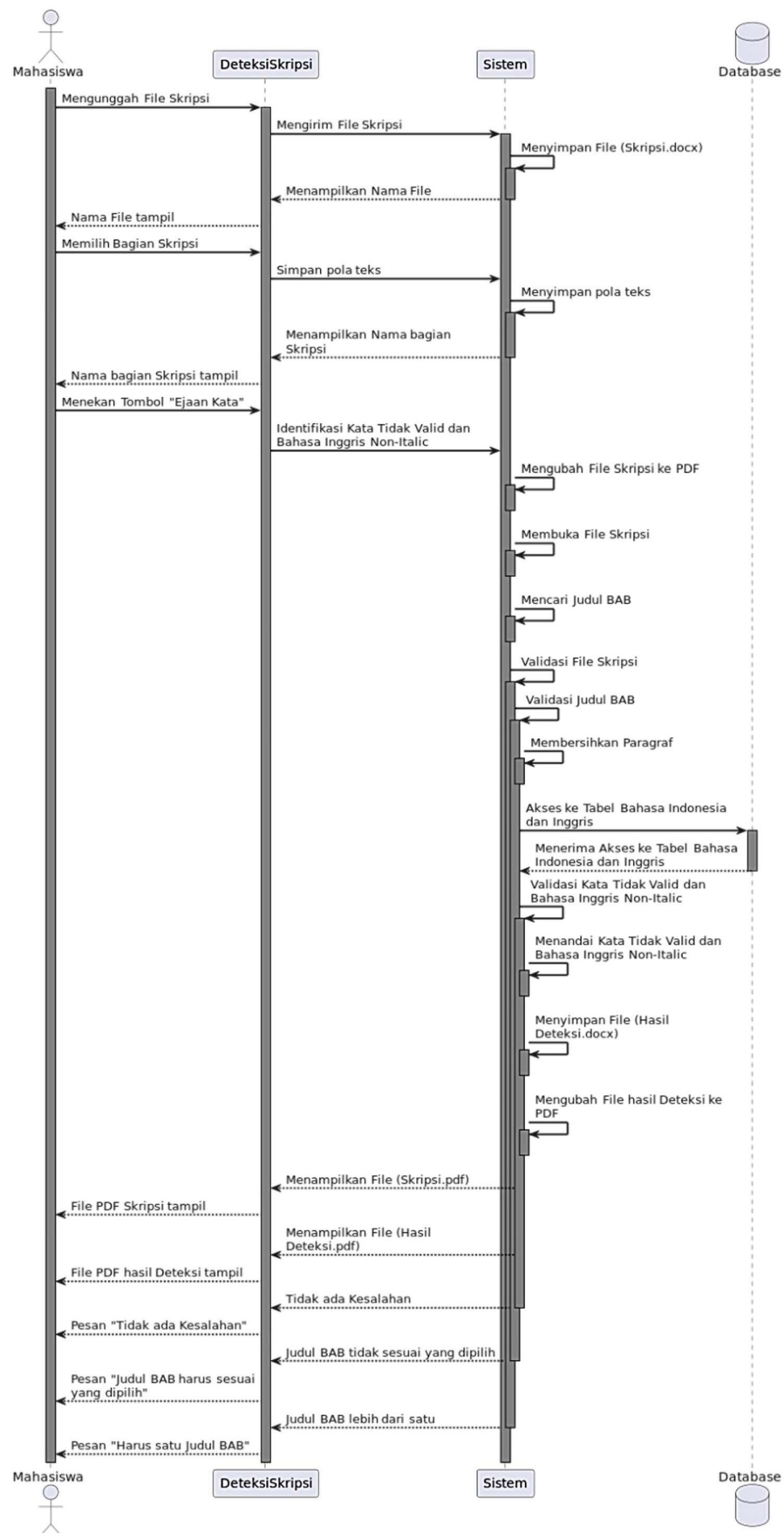
Berikut adalah diagram urutan yang menggambarkan serangkaian interaksi antara pengguna dan sistem selama proses mendeteksi kesalahan format pengetikan skripsi:



Gambar 8. *Sequence Diagram* untuk Mendeteksi Kesalahan Format Pengetikan

2. Diagram Urutan Deteksi Kesalahan Ejaan Kata

Berikut adalah diagram urutan yang menggambarkan serangkaian interaksi antara pengguna dan sistem selama proses mendeteksi kata tidak valid pada dokumen skripsi:



Gambar 9. Sequence Diagram untuk Mendeteksi Kata Tidak Valid

4. Hasil Penelitian

Analisis Kebutuhan Pengguna

Berdasarkan masalah yang ada di FMIPA UNSRAT mengenai kesalahan penulisan (struktur penomoran subbab, format pengetikan, dan ejaan kata) dari beberapa mahasiswa tingkat akhir angkatan 2021 yang telah menyelesaikan seminar usul penelitian. Peneliti menganalisis kebutuhan pengguna dari data yang telah dikumpulkan, berupa *hardcopy* atau cetakan dokumen proposal penelitian. Dari hasil analisis peneliti mengenai kebutuhan pengguna, ditetapkan beberapa fitur utama untuk aplikasi CekNulis:

1. Login, berfungsi untuk memverifikasi akun pengguna dan mengakses halaman deteksi skripsi.
2. Pendaftaran akun, berfungsi untuk pengguna melakukan pendaftaran akun baru.
3. Deteksi skripsi, berfungsi untuk mendeteksi kesalahan penulisan yang ada pada file skripsi.

Setelah peneliti menentukan fitur-fitur utama pada aplikasi CekNulis, terlebih dahulu peneliti merancang dan membuat aplikasi *prototype* CekNulis berdasarkan fitur-fitur utama tersebut. Peneliti melakukan evaluasi terhadap pengguna setelah aplikasi *prototype* selesai dibuat. Evaluasi ini dilakukan kepada mahasiswa angkatan 2021 sebanyak empat kali. Berikut adalah fitur-fitur yang didapatkan dari hasil evaluasi sebanyak empat kali, yang kemudian ditambahkan ke dalam aplikasi *prototype* CekNulis:

1. Admin, berfungsi untuk memantau, serta mengelola akun pengguna yang terdaftar.
2. Lupa *password*, berfungsi untuk memverifikasi keamanan akun sebelum pengguna mengatur ulang katasandi.
3. *Password* baru, berfungsi agar pengguna dapat mengatur ulang *password*, setelah melewati tahap verifikasi keamanan akun.
4. *Download*, berfungsi agar pengguna dapat mengunduh file hasil deteksi dengan format .docx.
5. *Refresh*, berfungsi untuk pengguna mengulang kembali halaman deteksi skripsi.
6. Catatan, berfungsi untuk memberikan informasi kepada pengguna mengenai ketentuan pendeteksian dokumen skripsi.
7. Tampilan file .pdf sebelum deteksi, berfungsi untuk menampilkan file .pdf yang diunggah ke halaman deteksi skripsi.
8. Verifikasi kriteria deteksi skripsi, berfungsi untuk pengguna yang baru pertama kali login. Pengguna harus membaca dan memahami terlebih dahulu kriteria deteksi skripsi yang ada sebelum mengakses halaman deteksi skripsi.
9. *Loading screen*, berfungsi untuk memberikan visual dari suatu proses yang sedang berjalan.
10. Mengubah data pengguna, berfungsi agar admin dapat mengelola data dari pengguna seperti NIM, nama, dan *password*.
11. Menerima akun, berfungsi agar admin dapat memberikan hak akses pengguna dalam mengakses halaman deteksi skripsi.
12. Menghapus akun, berfungsi agar admin dapat menghapus akun pengguna dari daftar.
13. Tambah kata valid, berfungsi agar admin dapat menambahkan beberapa kata valid ke dalam daftar kata valid.
14. Cari kata valid, berfungsi agar admin dapat mencari kata-kata valid yang relevan dengan kata yang dimasukkan.
15. Menghapus kata valid, berfungsi agar admin dapat menghapus salah satu kata valid dari daftar.

Tampilan Antarmuka Aplikasi CekNulis

Berikut merupakan hasil implementasi tampilan antarmuka aplikasi CekNulis dari bentuk *wireframe* yang dibagi ke dalam beberapa halaman, yaitu sebagai berikut:

1. Tampilan Antarmuka Halaman Depan

Berikut merupakan tampilan antarmuka pada aplikasi CekNulis untuk halaman depan:



Gambar 10. Tampilan Antarmuka Halaman Depan

Pada Gambar 10 merupakan tampilan awal aplikasi CekNulis yang menyambut pengguna saat pertama kali mengakses *website* tersebut. Halaman depan menampilkan ikon ilustratif yang mewakili fitur utama, yaitu pemeriksaan dokumen skripsi, serta judul aplikasi dan slogan “Biar Gak Sakit Mata” yang menjelaskan tujuan penggunaannya. Sebuah tombol aksi “Mulai Sekarang” disediakan untuk pengguna dan admin agar bisa langsung masuk ke halaman login.

2. Tampilan Antarmuka Halaman Pendaftaran Akun

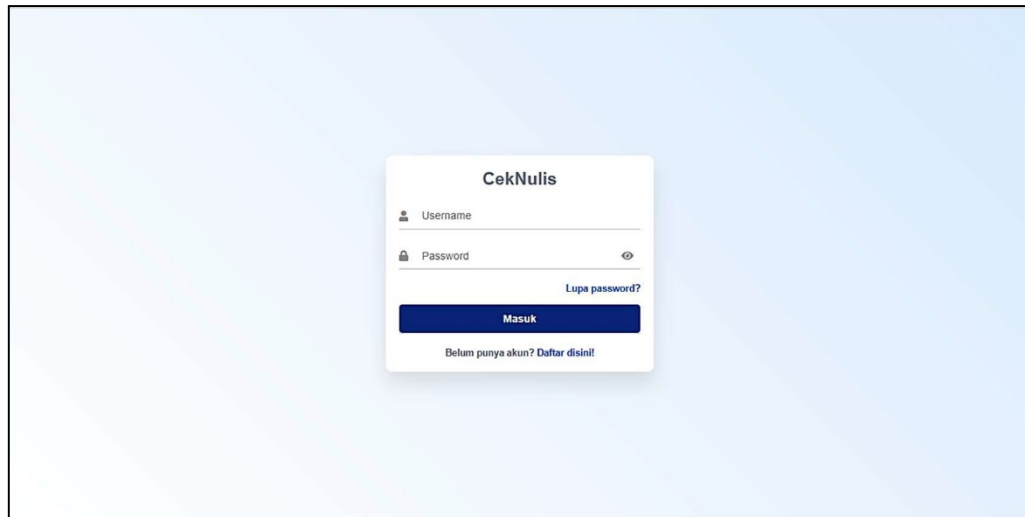
Berikut merupakan tampilan antarmuka pada aplikasi CekNulis untuk halaman pendaftaran akun:

Gambar 11. Tampilan Antarmuka Halaman Pendaftaran Akun

Pada Gambar 11 merupakan tampilan antarmuka halaman pendaftaran akun pada aplikasi CekNulis. Pengguna dapat membuat akun baru dengan mengisi data sesuai dengan yang telah ditentukan oleh sistem. Tombol “Daftar”, berfungsi untuk mengirimkan data pendaftaran akun ke dalam sistem. Data yang diterima oleh sistem, kemudian akan divalidasi dan didaftarkan ke dalam database. Tombol “Silahkan masuk”, berfungsi untuk pengguna kembali ke halaman login dan mencoba akun yang baru dibuat, untuk mengakses halaman deteksi skripsi.

3. Tampilan Antarmuka Halaman Login

Berikut merupakan tampilan antarmuka pada aplikasi CekNulis untuk halaman login:



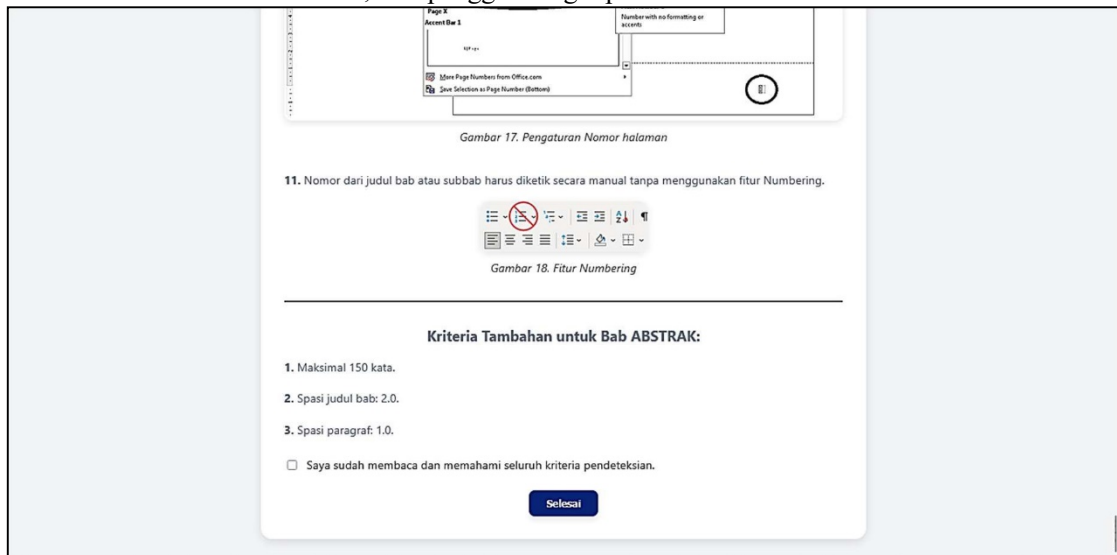
Gambar 12. Tampilan Antarmuka Halaman Login

Pada Gambar 12 merupakan tampilan antarmuka halaman login pada aplikasi CekNulis. *Text box* “Username”, berfungsi untuk pengguna memasukkan NIM, lalu *text box* “Password”, berfungsi untuk pengguna memasukkan *password* yang dibuat saat melakukan pendaftaran akun. Halaman login menyediakan akses ke halaman lain, seperti halaman pendaftaran pada tombol “Daftar disini”, dan halaman lupa *password* pada tombol “Lupa password”.

4. Tampilan Antarmuka Halaman Catatan Kriteria Deteksi

Tampilan antarmuka untuk tampilan halaman catatan kriteria deteksi pada aplikasi CekNulis memiliki dua jenis, yaitu:

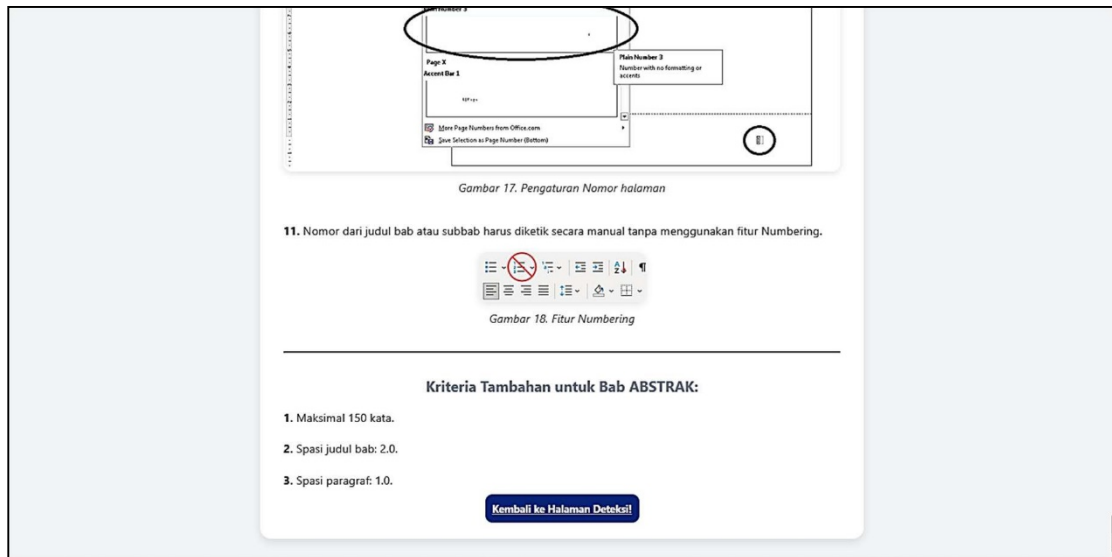
- a. Halaman catatan kriteria deteksi, saat pengguna login pertama kali.



Gambar 13. Tampilan Antarmuka Halaman Catatan Kriteria Deteksi saat Pengguna Login Pertama Kali

Pada Gambar 13 merupakan tampilan antarmuka halaman catatan kriteria deteksi, ketika pengguna login pertama kali di aplikasi CekNulis. Halaman catatan kriteria deteksi berisi informasi mengenai pendeteksian dokumen skripsi, seperti persiapan pengguna sebelum melakukan deteksi skripsi; kriteria pendeteksian format pengetikan skripsi; dan kriteria tambahan untuk pendeteksian bab abstrak. Tombol *checkboxlist* “Saya sudah membaca dan memahami seluruh kriteria pendeteksian”, berfungsi untuk memastikan pengguna telah membaca dan memahami kriteria pendeteksian dokumen skripsi. Tombol “Selesai”, berfungsi untuk mengakhiri proses pengguna dalam membaca kriteria pendeteksian dokumen dan mengakses halaman deteksi skripsi.

- b. Halaman catatan kriteria deteksi, yang diakses melalui tombol “Catatan” di halaman deteksi skripsi.

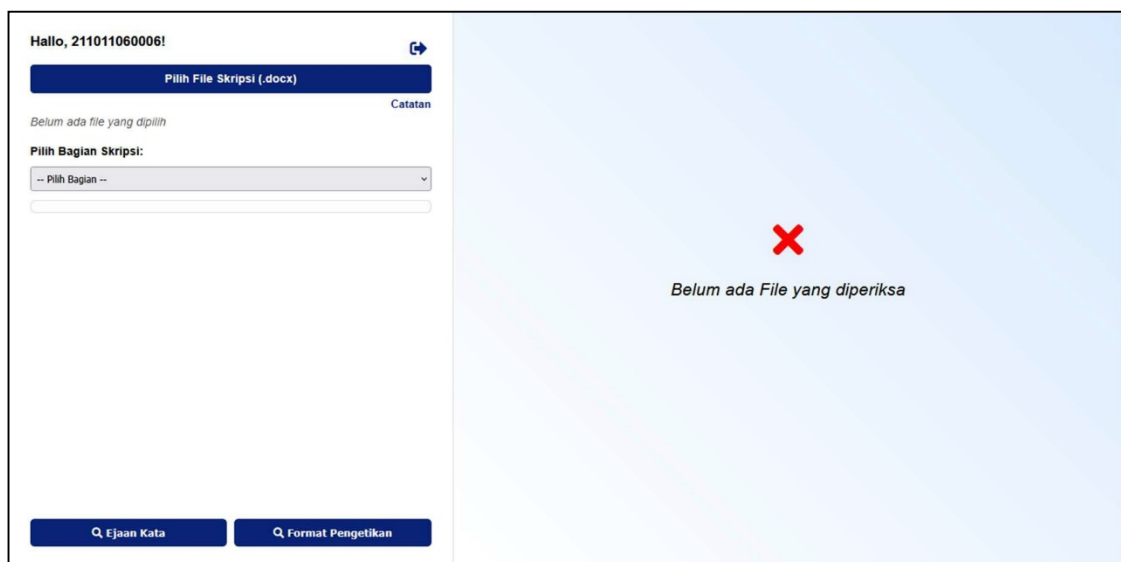


Gambar 14. Tampilan Antarmuka Halaman Catatan Kriteria Deteksi yang Diakses melalui Tombol “Catatan”

Pada Gambar 14 merupakan tampilan antarmuka halaman catatan kriteria deteksi, ketika pengguna login pertama kali di aplikasi CekNulis. Halaman catatan kriteria deteksi berisi informasi mengenai pendeteksian dokumen skripsi, seperti persiapan pengguna sebelum melakukan deteksi skripsi; kriteria pendeteksian format pengetikan skripsi; dan kriteria tambahan untuk pendeteksian bab abstrak. Tombol *checkbox* “Saya sudah membaca dan memahami seluruh kriteria pendeteksian”, berfungsi untuk memastikan pengguna telah membaca dan memahami kriteria pendeteksian dokumen skripsi. Tombol “Selesai”, berfungsi untuk mengakhiri proses pengguna dalam membaca kriteria pendeteksian dokumen dan mengakses halaman deteksi skripsi.

5. Tampilan Antarmuka Halaman Deteksi Skripsi

Berikut merupakan tampilan antarmuka pada aplikasi CekNulis untuk halaman deteksi skripsi:



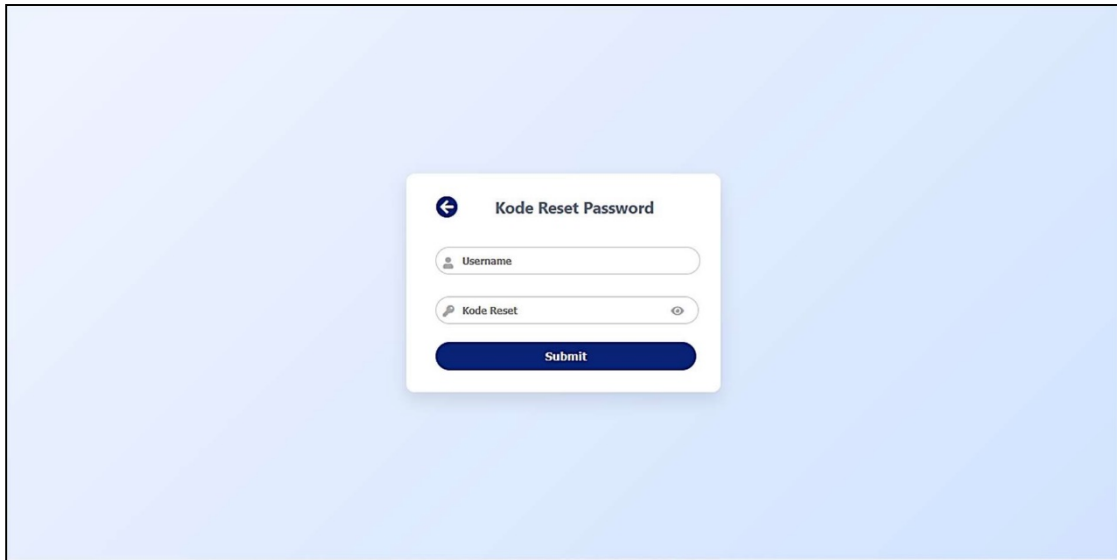
Gambar 15. Tampilan Antarmuka Halaman Deteksi Skripsi

Pada Gambar 15 merupakan tampilan antarmuka halaman deteksi skripsi pada aplikasi CekNulis. Tombol “Pilih file Skripsi (.docx)”, berfungsi untuk pengguna mengunggah file skripsi yang berformat .docx. Tombol “Catatan”, berfungsi untuk melihat kembali berapa informasi mengenai ketentuan pendeteksian dokumen skripsi. *Drop-down list* “Pilih Bagian”, berfungsi untuk pengguna memilih salah satu bagian skripsi dari daftar bagian yang telah ditentukan oleh sistem. Tombol “Ejaan Kata”, berfungsi untuk melakukan deteksi kesalahan ejaan kata pada file skripsi yang telah diunggah oleh pengguna. Tombol “Format Pengetikan”, berfungsi untuk melakukan deteksi kesalahan penulisan format dan struktur dokumen skripsi. Tombol ikon “logout”, berfungsi untuk mengeluarkan akun pengguna dari halaman deteksi skripsi

dan kembali ke halaman login.

6. Tampilan Antarmuka Halaman Lupa *Password*

Berikut merupakan tampilan antarmuka pada aplikasi CekNulis untuk halaman lupa *password*:

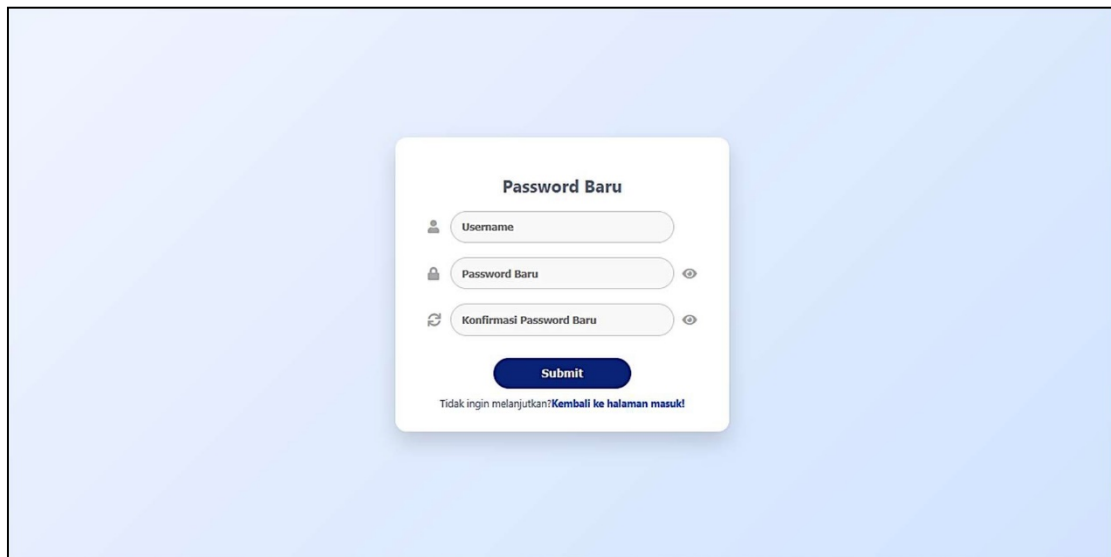


Gambar 16. Implementasi Tampilan Antarmuka Halaman Lupa *Password*

Pada Gambar 16 merupakan tampilan antarmuka halaman lupa *password* pada aplikasi CekNulis. *Text box* “Username”, berfungsi untuk pengguna memasukkan NIM, lalu *text box* “Kode Reset”, berfungsi untuk pengguna memasukkan kode reset yang didapatkan saat melakukan pendaftaran akun. Tombol “Submit”, berfungsi untuk mengirimkan NIM dan kode reset ke dalam sistem untuk diperiksa; Jika data yang dimasukkan lulus dari pemeriksaan, maka pengguna akan mengakses halaman *password* baru.

7. Tampilan Antarmuka Halaman *Password* Baru

Berikut merupakan tampilan antarmuka pada aplikasi CekNulis untuk halaman *password* baru:

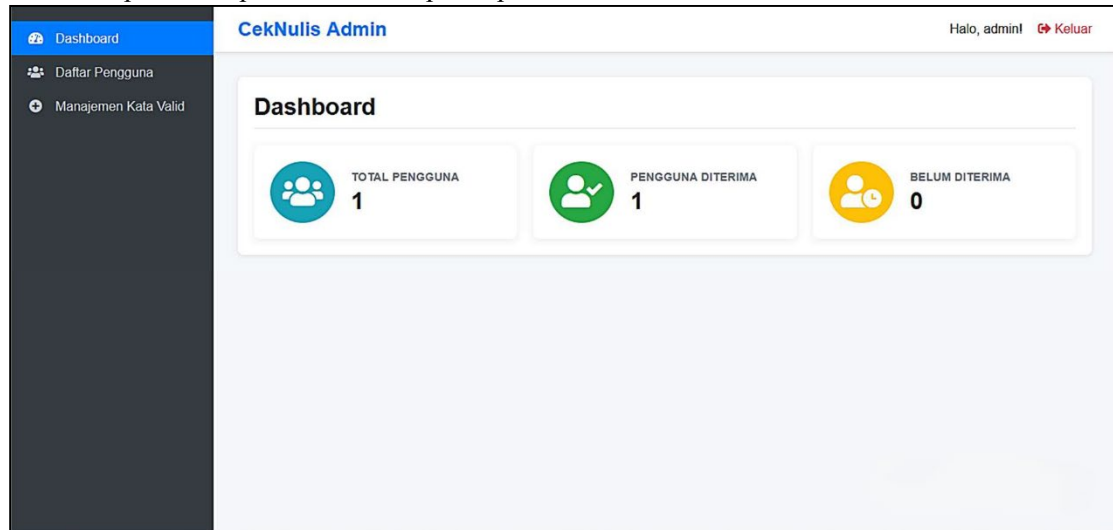


Gambar 17. Tampilan Antarmuka Halaman *Password* Baru

Pada Gambar 17 merupakan tampilan antarmuka halaman *password* baru pada aplikasi CekNulis, setelah pengguna mengkonfirmasi NIM dan kode reset pada halaman lupa *password*. *Text box* “Username”, berfungsi untuk pengguna memasukkan NIM. *Text box* “*Password* Baru”, berfungsi untuk pengguna memasukkan *password* yang baru, menggantikan *password* yang lama. *Text box* “*Konfirmasi Password* Baru”, berfungsi untuk menghindari pengguna dari kesalahan pengetikan *password*. Tombol “Submit”, berfungsi untuk mengirimkan NIM, *password* baru, dan konfirmasi *password* baru ke dalam sistem untuk diperiksa; Jika data yang dimasukkan lulus dari pemeriksaan, maka sistem akan mengubah *password* dari akun pengguna menjadi *password* yang baru.

8. Tampilan Antarmuka Halaman *Dashboard* Admin

Berikut merupakan tampilan antarmuka pada aplikasi CekNulis untuk halaman *dashboard* admin:

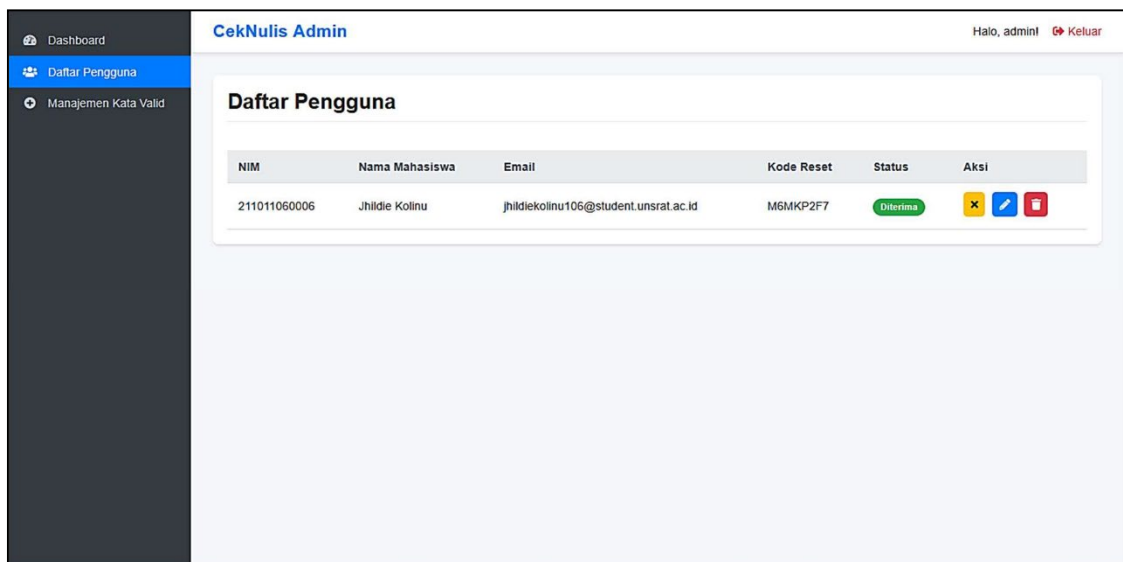


Gambar 18. Tampilan Antarmuka Halaman *Dashboard* Admin

Pada Gambar 18 merupakan tampilan antarmuka halaman *dashboard* admin pada aplikasi CekNulis. Statistik “Total Pengguna”, berfungsi untuk memperlihatkan jumlah akun yang ada pada tabel daftar pengguna. Statistik “Pengguna Diterima”, berfungsi untuk memperlihatkan jumlah akun yang telah diterima oleh admin. Statistik “Belum Diterima”, berfungsi untuk memperlihatkan jumlah akun yang belum diterima oleh admin.

9. Tampilan Antarmuka Halaman Daftar Pengguna

Berikut merupakan tampilan antarmuka pada aplikasi CekNulis untuk halaman daftar pengguna:

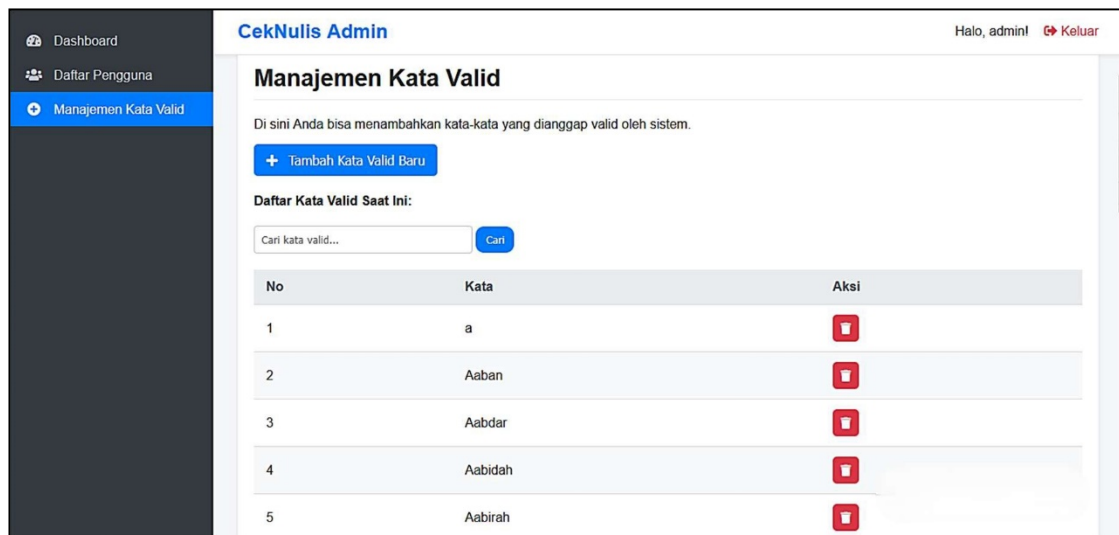


Gambar 19. Tampilan Antarmuka Halaman Daftar Pengguna

Pada Gambar 19 merupakan tampilan antarmuka halaman daftar pengguna pada aplikasi CekNulis. Beberapa tombol yang ada pada kolom “Aksi”, berfungsi untuk admin mengelola akun pengguna. Tombol ikon “Centang”, berfungsi untuk admin menerima akun pengguna, agar bisa mengakses halaman deteksi skripsi. Tombol ikon “Pencil”, berfungsi untuk admin mengubah data akun pengguna. Tombol ikon “Tong Sampah”, berfungsi untuk admin menghapus akun pengguna.

10. Tampilan Antarmuka Halaman Manajemen Kata Valid

Berikut merupakan tampilan antarmuka pada aplikasi CekNulis untuk halaman manajemen kata valid:



Gambar 20. Tampilan Antarmuka Halaman Manajemen Kata Valid

Pada Gambar 20 merupakan tampilan antarmuka halaman manajemen kata valid pada aplikasi CekNulis. Tombol “Tambah Kata Valid Baru”, berfungsi untuk admin menambahkan beberapa kata valid baru ke dalam daftar kata valid. *Text box* “Cari kata valid”, berfungsi untuk admin memasukkan kata valid yang ingin dicari pada daftar kata valid. Tombol “Cari”, berfungsi untuk memulai proses pencarian kata valid yang dimasukkan admin ke dalam *text box* “Cari kata valid”. Tombol ikon “Tong Sampah”, berfungsi untuk admin menghapus salah satu kata valid dari daftar.

Pengujian *Black Box* Aplikasi CekNulis

Pengujian dilakukan menggunakan pendekatan *Black Box*, yang mengevaluasi fungsionalitas sistem dari sudut pandang pengguna tanpa memahami cara kerja internal perangkat lunak. Tujuan pengujian *Black Box* adalah untuk memverifikasi bahwa setiap fitur dalam aplikasi CekNulis berfungsi sebagaimana mestinya. Berikut adalah hasil pengujian menggunakan pendekatan *Black Box* pada fitur-fitur utama aplikasi CekNulis:

1. Pengujian Fitur Pendaftaran Akun Aplikasi CekNulis

Berikut merupakan hasil pengujian menggunakan pendekatan *Black Box* pada aplikasi CekNulis untuk fitur pendaftaran akun:

Tabel 1. Hasil Pengujian *Black Box* pada Fitur Pendaftaran Akun

No.	Skenario Pengujian	Data Uji	Hasil yang Diharapkan	Hasil Aktual	Status
1.	Daftar dengan data yang valid	Nama Mahasiswa: Jhildie Kolinu NIM: 211011060006 Password: *** Konfirmasi Password: *** Kode Reset: Y6Y88FSQ	Tampilan pesan “Pendaftaran akun berhasil, silahkan hubungi Kaprodi untuk memverifikasi akun”	Sesuai	Berhasil
2.	Daftar dengan NIM kurang dari 12 digit	NIM: 21101106000	Tampilan pesan “NIM harus 12 digit”.	Sesuai	Berhasil
3.	Daftar dengan NIM lebih dari 12 digit	NIM: 2110110600006	Tampilan pesan “NIM harus 12 digit”.	Sesuai	Berhasil
4.	Daftar dengan NIM yang berupa huruf	NIM: ABCDEFGHJL KL	Tampilan pesan “NIM harus berupa angka”.	Sesuai	Berhasil
5.	Daftar dengan NIM yang sudah terdaftar	NIM: 211011060006	Tampilan pesan “NIM sudah terdaftar, Silahkan gunakan NIM lain”.	Sesuai	Berhasil
6.	Daftar dengan domain yang bukan dari UNSRAT	Email jhildiekolinu106@gmail.com	Tampilan pesan “Email harus menggunakan domain @student.unsrat.ac.id”.	Sesuai	Berhasil
7.	Daftar dengan konfirmasi <i>password</i> yang tidak cocok.	Konfirmasi Password: salah 123	Tampilan pesan “Konfirmasi <i>password</i> tidak cocok”.	Sesuai	Berhasil

2. Pengujian Fitur Login Aplikasi CekNulis

Berikut merupakan hasil pengujian menggunakan pendekatan *Black Box* pada aplikasi CekNulis untuk fitur login:

No.	Skenario Pengujian	Data Uji	Hasil yang Diharapkan	Hasil Aktual	Status
1.	Daftar dengan data yang valid	Nama Mahasiswa: Jhildie Kolinu NIM: 211011060006 Password: *** Konfirmasi Password: *** Kode Reset: Y6Y88FSQ	Tampilan pesan “Pendaftaran akun berhasil, silahkan hubungi Kaprodi untuk memverifikasi akun”	Sesuai	Berhasil
2.	Daftar dengan NIM kurang dari 12 digit	NIM: 21101106000	Tampilan pesan “NIM harus 12 digit”.	Sesuai	Berhasil
3.	Daftar dengan NIM lebih dari 12 digit	NIM: 2110110600006	Tampilan pesan “NIM harus 12 digit”.	Sesuai	Berhasil
4.	Daftar dengan NIM yang berupa huruf	NIM: ABCDEFGH	Tampilan pesan “NIM harus berupa angka”.	Sesuai	Berhasil
5.	Daftar dengan NIM yang sudah terdaftar	NIM: 211011060006	Tampilan pesan “NIM sudah terdaftar, Silahkan gunakan NIM lain”.	Sesuai	Berhasil
6.	Daftar dengan domain yang bukan dari UNSRAT	Email jhildiekolinu106@gmail.com	Tampilan pesan “Email harus menggunakan domain @student.unsrat.ac.id”.	Sesuai	Berhasil
7.	Daftar dengan konfirmasi <i>password</i> yang tidak cocok.	Konfirmasi Password: salah 123	Tampilan pesan “Konfirmasi <i>password</i> tidak cocok”.	Sesuai	Berhasil

3. Pengujian Fitur Deteksi Skripsi Aplikasi CekNulis

Berikut merupakan hasil pengujian menggunakan pendekatan *Black Box* pada aplikasi CekNulis untuk fitur deteksi skripsi:

Tabel 3. Hasil Pengujian *Black Box* pada Fitur Deteksi Skripsi

No.	Skenario Pengujian	Data Uji	Hasil yang Diharapkan	Hasil Aktual	Status
1.	Tekan salah satu tombol deteksi skripsi, tanpa mengunggah file.	Tidak ada file	Tampilan pesan “Tidak ada file yang dipilih”.	Sesuai	Berhasil
2.	Tekan salah satu tombol deteksi skripsi, tanpa memilih bagian skripsi.	<i>file</i> (TES.docx)	Tampilan pesan “Silakan pilih bagian skripsi yang ingin diperiksa”.	Sesuai	Berhasil
3.	Unggah file dengan format .pdf.	<i>file</i> (TES.pdf)	Tampilan pesan “Format file tidak didukung, harap unggah file .docx”.	Sesuai	Berhasil
4.	Unggah <i>file</i> skripsi (.docx) yang lebih dari satu judul bab.	<i>file</i> (TES.docx)	Tampilan pesan “Pendeteksian hanya dapat dilakukan jika dokumen hanya memuat bagian skripsi yang dipilih”.	Sesuai	Berhasil
5.	Pilih bagian skripsi yang tidak sesuai dengan judul bab yang ada pada <i>file</i> skripsi.	<i>file</i> (TES.docx)	Tampilan pesan “Bagian yang dipilih tidak ditemukan dalam dokumen, silahkan periksa kembali teks pada judul BAB, termasuk gaya dan ukuran font yang digunakan”.	Sesuai	Berhasil
6.	Deteksi Kata Tidak Valid pada <i>file</i> skripsi (.docx), sesuai dengan ketentuan.	<i>file</i> (TES.docx)	Tampilan <i>file</i> hasil deteksi skripsi dalam format .pdf pada halaman deteksi skripsi.	Sesuai	Berhasil
7.	Deteksi Format Pengetikan pada <i>file</i> skripsi (.docx), sesuai dengan ketentuan.	<i>file</i> (TES.docx)	Tampilan <i>file</i> hasil deteksi skripsi dalam format .pdf pada halaman deteksi skripsi.	Sesuai	Berhasil
8.	Tekan tombol “Catatan”.	Tidak ada <i>file</i>	Masuk ke halaman catatan kriteria deteksi.	Sesuai	Berhasil
9.	Tekan tombol ikon <i>Download file</i> (.docx) hasil deteksi skripsi.	<i>file</i> (TES.docx)	Berhasil mengunduh <i>file</i> (.docx) hasil deteksi skripsi.	Sesuai	Berhasil
10.	Tekan tombol ikon <i>Refresh</i> .	<i>file</i> (TES.docx)	Memuat ulang halaman deteksi skripsi.	Sesuai	Berhasil

5. Kesimpulan

Dari hasil pengembangan dan pengujian aplikasi CekNulis, dapat disimpulkan bahwa sistem deteksi kesalahan penulisan skripsi telah berhasil dirancang dan diimplementasikan untuk memeriksa kesesuaian strukturnomor subbab, format pengetikan, dan ejaan kata pada dokumen skripsi.

Hasil pengujian dengan metode *Black Box* menunjukkan bahwa fitur pendaftaran akun, login, dan deteksi skripsi beroperasi dengan benar sesuai dengan skenario pengujian, tanpa terdeteksi adanya kesalahan fungsional. Deteksi skripsi yang terdiri dari dua bagian, yaitu deteksi kata tidak valid dan deteksi format pengetikan skripsi, memberikan hasil keluaran (*output*) yang sesuai dengan harapan. Sistem dapat menandai kesalahan struktur penomoran subbab dan format pengetikan menggunakan fitur (*comments*) pada aplikasi Word, dan dapat menandai kesalahan ejaan kata dengan format (>Kata Tidak Valid<), serta menandai serapan bahasa Inggris yang tidak dibuat miring (*italic*) dengan format (●Word Non-Italic●).

Dengan demikian, aplikasi CekNulis berfungsi sebagai alat bantu penulisan skripsi, khususnya membantu mahasiswa FMIPA UNSRAT untuk memastikan bahwa naskah yang ditulisnya telah sesuai dengan pedoman penulisan FMIPA yang berlaku dan terhindar dari kesalahan ejaan kata.

6. Daftar Pustaka

- [1] H. Sholihin, H. Latipa Sari, and H. Aspriyono, "Implementasi Kriptografi Klasik Untuk Pengamanan Database Berbasis Web," Apr. 2022. [Online]. Available: <https://kriptografihidayat.my.id/>
- [2] Akrom, F. Marwati, and A. Astofa, "Pentingnya Edukasi Cyber Security Untuk Menjaga Keamanan Data Pribadi dari Serangan Cyber Phishing Bagi Siswa/Siswi PKBM INTAN Tangerang Selatan," *AMMA : Jurnal Pengabdian Masyarakat*, vol. 2, no. 12, pp. 1508–1514, Jan. 2024.
- [3] A. Azhara *et al.*, "Analisis Kebocoran Data NPWP dalam Sistem e-Government: Tinjauan Keamanan Informasi dan Kepercayaan Publik," *QISTINA : Jurnal Multidisiplin Indonesia*, vol. 4, no. 1, pp. 981–996, Jun. 2025.
- [4] S. Azura *et al.*, "Penerapan Kemanan Data Text menggunakan Metode Kriptografi Vigenere Chiper Berbasis Web," *Digital Transformation Technology (Digitech) | e*, vol. 3, no. 1, pp. 20–28, Mar. 2023, doi: 10.47709/digitech.v3i1.2311.
- [5] V. G. Ramengkomole, E. Ketaren, and D. T. Salaki, "Implementasi Metode Substitusi Out By Diagonals Pada Pengamanan File Teks," *Jurnal Times Technology Informatics & Computer System*, vol. 14, no. 1, pp. 162–174, Jun. 2025, [Online]. Available: <http://ejournal.stmik-time.ac.id>
- [6] N. Siregar, Suriati, and A. Usman, "Penerapan Algoritma Kriptografi Hybrid Substitusi dan Transposisi Spiral Dalam Mengamankan Data Teks," *ALGORITMA: Jurnal Ilmu Komputer dan Informatika*, vol. 05, no. 01, pp. 81–90, Apr. 2021.
- [7] N. Sephiana, M. Tahir, S. D. Wulandari, F. R. Rahmansyah, R. N. Nuvitasari, and R. A. Prakasa, "Analisis Perbandingan Algoritma Monoalphabetic Cipher dan Polyalphabetic Substitution Cipher Pada Sistem Keamanan Data," *Jurnal Keilmuan dan Aplikasi Teknik Informatika*, pp. 16–21, Jun. 2023, doi: 10.35891/explorit.
- [8] R. Syahputra, "Implementasi Algoritma Adaptive Huffman Code Dalam Kompresi File Teks Terenkripsi Algoritma Loki97," Nov. 2021.
- [9] H. Patiung and A. D. Wowor, "Perancangan Algoritma Square Transposisi dengan Skema Spiral," *JURNAL INOVTEK POLBENG - SERI INFORMATIKA*, vol. 9, no. 2, pp. 878–889, 2024.
- [10] E. Setyawati, C. E. Widjayanti, R. R. Siraiz, and H. Wijoyo, "Pengujian Keamanan Komputer Kriptografi pada Surat Elektronik Berbasis Website dengan Enkripsi Metode MD5," *Jurnal Manajemen Informatika Jayakarta*, vol. 1, no. 1, pp. 56–67, Feb. 2021, doi: 10.52362/jmijayakarta.v1i1.367.
- [11] D. F. Ningtyas and N. Setiyawati, "Implementasi Flask Framework pada Pembangunan Aplikasi Purchasing Approval Request," *Jurnal Janitra Informatika dan Sistem Informasi*, vol. 1, no. 1, pp. 19–34, Apr. 2021, doi: 10.25008/janitra.v1i1.120.
- [12] R. Parlita, S. I. Pradika, A. M. Hakim, and R. N. Kholilul, "Bot Whatsapp Sebagai pemberi Data Statistik Covid-19 Menggunakan PHP, Flask, dan MySQL," Jun. 2020. [Online]. Available: <https://api.kawalc Corona.com/>