

**“PERANCANGAN SISTEM APLIKASI *GENERATE* SKRIPSI KE
JURNAL BERBASIS WEB”**

SKRIPSI

**Di ajukan sebagai Salah Satu Syarat untuk Mendapatkan
Gelara Sarjana Komputer (S.Kom) Program Studi Informatika**



OLEH:

YULISTIAH NENGSIH

105841107020

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MUHAMMADIYAH MAKASSAR

2024

**“PERANCANGAN SISTEM APLIKASI *GENERATE* SKRIPSI KE
JURNAL BERBASIS WEB”**

**Di ajukan sebagai Salah Satu Syarat untuk Mendapatkan
Gelar Sarjana Komputer (S.Kom) Program Studi Informatika**



PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MUHAMMADIYAH MAKASSAR

2024



بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

PENGESAHAN

Skripsi atas nama Yulistiah Nengsih dengan nomor induk Mahasiswa 105 84 11070 20, dinyatakan diterima dan disahkan oleh Panitia Ujian Tugas Akhir/Skripsi sesuai dengan Surat Keputusan Dekan Fakultas Teknik Universitas Muhammadiyah Makassar Nomor : 143/05/A.5-VI/V/45/2024, sebagai salah satu syarat guna memperoleh gelar Sarjana Komputer pada Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar pada hari Sabtu tanggal 31 Agustus 2024.

Panitia Ujian :

Makassar, 26 Safar 1446 H
31 Agustus 2024 M

1. Pengawas Umum

a. Rektor Universitas Muhammadiyah Makassar

Dr. Ir. H. Abd. Rakhim Nanda, ST., MT., JPU

b. Dekan Fakultas Teknik Universitas Hasanuddin

Prof. Dr. Eng. Muhammad Isran Ramli, ST., MT.

2. Penguji

a. Ketua : Dr. Ir. Zahir Zainuddin, M.Sc.

b. Sekretaris : Lukman, S.Kom., M.T.

3. Anggota : 1. Titin Wahyuni, S.Pd., M.T.

2. Lukman Anas, S.Kom., M.T.

3. Desi Anggreani, S.Kom., M.T.

Mengetahui :

Pembimbing I

Rizki Yusliana Bakti, S.T., M.T

Pembimbing II

Muhyiddin A.M Hayat, S.Kom., M.T

Dekan



Dr. Hj. Nurmawaty, ST., MT., IPM.

NBM : 795 108



UNIVERSITAS MUHAMMADIYAH MAKASSAR

FAKULTAS TEKNIK

GEDUNG MENARA IQRA LT. 3

Jl. Sultan Alauddin No. 259 Telp. (0411) 866 972 Fax (0411) 865 588 Makassar 90221

Website: www.unismuh.ac.id, e_mail: unismuh@gmail.com

Website: <http://teknik.unismuh.makassar.ac.id>

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

HALAMAN PENGESAHAN

Tugas Akhir ini diajukan untuk memenuhi syarat ujian guna memperoleh gelar Sarjana Komputer (S.Kom) Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar.

Judul Skripsi : **PERANCANGAN APLIKASI GENERATE SKRIPSI KE JURNAL BERBASIS WEB**

Nama : Yulistiah Nengsih

Stambuk : 105 84 11070 20

Makassar, 31 Agustus 2024

Telah Diperiksa dan Disetujui
Oleh Dosen Pembimbing;

Pembimbing I

Pembimbing II

Rizki Yusliana Bakti, S.T., MT.

Muhyiddin A.M Hayat, S.Kom., M.T

Mengetahui,

Ketua Program Studi Arsitektur



Muhyiddin A.M Hayat, S.Kom., M.T

NBM : 504 577

MOTTO

“Sesungguhnya sesudah kesulitan ada kemudahan. Maka apabila kamu telah selesai (dari satu urusan), kerjakanlah dengan sungguh-sungguh (urusan) yang lain. Dan hanya kepada ALLAH tempatmu berharap”

(Q.S Al-Insyirah:5-8)

“Setiap orang punya gilirannya masing-masing. Bersabar dan tungguilah.”

(Gol D. Roger)



PERSEMBAHAN

Karya ini merupakan wujud rasa syukur kepada sang pencipta Allah Subhanahu Wata'ala atas segala nikmat yang tak terhingga, sehingga penulis dapat menyelesaikan skripsi ini.

Karya ini juga saya persembahkan sebagai tanda bukti sayang dan cinta yang tiada terhingga kepada kedua Orang Tua saya, Bapak **Muhammad Dahru** dan yang sangat teristimewa Ibu **Rusna** yang selalu menjadi *support system* terbaik. Dengan semangat yang tidak kenal lelah, kerja keras, dukungan, harapan serta doa-doa tulus yang selalu dilangitkan, membuat penulis dapat menyelesaikan studinya sampai menjadi seorang Sarjana. Anak saya **Dyah Azzahrah**, duniaku dan cinta matiku, alasan terbesar bagi penulis bertahan sampai detik ini, terima kasih sudah menjadikan aku Ibu dan terima kasih sudah menerima aku sebagai Ibu. Saudara saya **Al-Fandi Dahru, S.H.**, yang turut memberikan dukungan, motivasi dan doa.



ABSTRAK

YULISTIAH NENGSIH, Perancangan Aplikasi Generate Skripsi ke Jurnal Berbasis Web (dibimbing oleh Rizki Yusliana Bakti, ST., MT dan Muhyiddin A.M. Hayat, S.Kom., M.T).

Penelitian ini bertujuan untuk merancang sistem aplikasi berbasis web yang dapat mengonversi skripsi menjadi artikel jurnal secara otomatis. Sistem ini diharapkan dapat membantu mahasiswa dalam mempublikasikan hasil penelitiannya dengan lebih mudah dan cepat. Aplikasi ini dikembangkan menggunakan beberapa pustaka *Python* untuk menganalisis dan mengekstraksi informasi penting dari skripsi, seperti pendahuluan, metode, hasil, dan kesimpulan, serta menyusunnya dalam format jurnal sesuai standar yang ditentukan. Pengujian dilakukan menggunakan metode *Black Box Testing* untuk memastikan sistem berjalan dengan baik sesuai dengan *input* dan *output* yang diharapkan. Hasil pengujian menunjukkan bahwa aplikasi ini berhasil mengonversi skripsi menjadi artikel jurnal dengan tingkat akurasi 100% kode berhasil mengekstraksi bagian-bagian penting dari skripsi, dan dapat menyesuaikannya dengan format template jurnal.

Kata Kunci: Perancangan Aplikasi, *Generate Skripsi ke Jurnal*, Template Jurnal.

ABSTRACT

YULISTIAH NENGSIH, *Design of Generate Thesis Applications for Web-Based Journals* (supervised by Rizki Yusliana Bakti, ST., MT and Muhyiddin A.M. Hayat, S.Kom., M.T).

This research aims to design a web-based application system that can convert thesis into journal articles automatically. This system is expected to help students in publishing their research results more easily and quickly. This application was developed using several Python libraries to analyze and extract important information from the thesis, such as introductions, methods, results, and conclusions, and compile it in a journal format according to the specified standards. Testing is carried out using the Black Box Testing method to ensure the system runs properly according to the expected inputs and outputs. The test results show that this application successfully converts the thesis into a journal article with a 100% accuracy level of code, successfully extracts important parts from the thesis, and can adjust it to the journal template format.

Keywords: *Application Design, Generate Thesis for Journal, Journal Template.*

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Assalamu'alaikum Warahmatullahi Wabarakatuh

Alhamdulillah hirabbil 'alamin senantiasa penulis panjatkan rasa syukur kepada sang pencipta Allah Subhanahu Wata'ala atas rahmat, hidayah dan karuniaNya sehingga penulis mampu menyusun Skripsi yang berjudul **“PERANCANGAN SISTEM APLIKASI *GENERATE* SKRIPSI KE JURNAL BERBASIS WEB”** dapat diselesaikan. Shalawat serta salam tak lupa penulis kirimkan kepada baginda tercinta Rasulullah, Nabi Muhammad Shallallahu alayhi wasallam sebagai uswatun hasanah dan rahmatan lil alamin.

Dalam penyusunan Skripsi ini penulis banyak menerima bimbingan, arahan, motivasi, serta bantuan oleh berbagai pihak, baik secara langsung maupun tidak langsung. Penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Diri sendiri, terima kasih telah bertahan sejauh ini, berusaha keras dan berjuang sejauh ini, terima kasih sudah menjadi pribadi yang mampu mengendalikan diri dari berbagai kesulitan, terima kasih karna tidak menyerah dan selalu semangat untuk menghadapi proses penulisan skripsi ini sampai skripsi ini bisa diselesaikan sebagaimana semestinya.
2. Kedua orang tua terhebat saya, cinta pertama dan panutan penulis, yaitu Bapak **Muhammad Dahru** dan Ibu **Rusna** yang selalu menjadi *support system* bagi penulis. Beliau memang tidak merasakan pendidikan di bangku perkuliahan, namun dengan semangat yang tidak pernah kenal lelah, kerja keras, doa – doa tulus, harapan, dukungan dan didikan yang diberikan, membuat penulis dapat menyelesaikan studinya sampai menjadi seorang Sarjana.
3. Anak kandung saya **Dyah Azzahrah**, dunia dan cinta mati penulis, yang selalu menjadi *support system* bagi penulis, alasan terbesar bagi penulis bertahan sampai detik ini

4. Saudara kandung saya **Al-Fandi Dahru, S.H.**, yang turut memberikan dukungan, motivasi serta doa.
5. Bapak **Dr. Ir. Abd. Rakhim Nanda, S.T., M.T., I.P.U.**, sebagai rektor perguruan tinggi Universitas Muhammadiyah Makassar.
6. Ibu **Dr. Ir. Hj Nurnawati, S.T., M.T., I.P.M.**, selaku Dekan Fakultas Teknik Universitas Muhammadiyah Makassar.
7. Bapak **Muhyiddin AM Hayat S.Kom., M.T.**, selaku Ketua Program Studi Informatika Universitas Muhammadiyah Makassar dan selaku Dosen Pembimbing 2 yang telah memberi arahan dan bimbingan serta saran yang sangat berarti dalam penyusunan skripsi ini.
8. Ibu **Rizki Yusliana Bakti S.T., MT.**, selaku Dosen Pembimbing 1 yang telah memberikan arahan dan bimbingan serta saran yang sangat berarti dalam penyusunan skripsi ini
9. Ibu **Titin Wahyuni S.Pd., M.T.**, selaku Dosen Pembimbing Akademik yang selama ini turut serta membantu dan memberikan arahan.
10. Kakak **Andi Agung Dwi Arya B S.Kom.**, selaku Dosen Informatika yang selama ini turut serta membantu dan memberikan arahan.
11. Seluruh Dosen Fakultas Teknik Program Studi Informatika Universitas Muhammadiyah Makassar yang telah memberikan ilmu dan bantuannya serta dorongannya dalam penulisan skripsi ini.
12. Sahabat penulis Wilda Yanti, Rina Permatasari, Amelia Safitri dan Wanda Tyrana Dewi MJ, terimakasih atas *support* dan dedikasinya selama ini membersamai penulis hingga detik ini.
13. Teman – teman saya khususnya Program Studi Informatika dan seluruh Angkatan 2020 Fakultas Teknik Universitas Muhammadiyah Makassar, terima kasih yang sebesar – besarnya atas semua dukungan, doa dan sudah membersamai penulis selama kurang lebih 4 tahun belakangan ini.
14. Kepada semua pihak yang tidak bisa penulis tuliskan satu persatu, penulis mengucapkan terima kasih yang sedalam – dalamnya dan sebesar – besarnya.

Semoga kebaikan menjadi Amal Sholeh dan di balas dengan kebaikan pula oleh Allah Subhanahu Wata'ala Aamiin ya Rabbal Alamin. Demikian laporan Skripsi ini, dan penulis sadar bahwa masih banyak kekurangan didalam laporan ini oleh karna itu penulis mengharapkan kritik dan saran yang sifatnya membangun dari pembaca atas laporan ini. Akhir kata penulis ucapkan terima kasih

Billahi fisabilhaq, fastabiqul khairat.

Wassalamualaikum Warahmatullahi Wabarakatuh.

Makassar, 17 Agustus 2024

Penulis

Yulistiah Nengsih



DAFTAR ISI

PENGESAHAN	iii
HALAMAN PENGESAHAN	iv
MOTTO	v
PERSEMBAHAN	vi
ABSTRAK	vii
ABSTRACT	viii
KATA PENGANTAR	ix
DAFTAR ISI	xii
DAFTAR TABEL	xiv
DAFTAR GAMBAR	xv
DAFTAR LAMPIRAN	xvi
DAFTAR ISTILAH	xvii
BAB I PENDAHULUAN	1
A. Latar Belakang	1
B. Rumusan Masalah	3
C. Tujuan Penelitian	3
D. Manfaat Penelitian	3
E. Ruang Lingkup Penelitian.....	4
F. Sistematika Penulisan	4
BAB II TINJAUAN PUSTAKA	6
A. Landasan Teori.....	6
B. Penelitian Terkait	9
C. Kerangka Berfikir.....	12

BAB III METODE PENELITIAN	13
A. Tempat dan Waktu Penelitian	13
B. Alat dan Bahan	14
C. Perancangan Sistem	14
D. Teknik Pengujian Sistem	16
E. Teknik Analisis Data	19
BAB IV HASIL DAN PEMBAHASAN	21
A. Perancangan Aplikasi	21
B. Proses Generate Skripsi ke Jurnal	26
C. Pengujian Sistem	27
BAB V KESIMPULAN DAN SARAN	37
A. Kesimpulan	37
B. Saran	37
DAFTAR PUSTAKA	38
LAMPIRAN	43

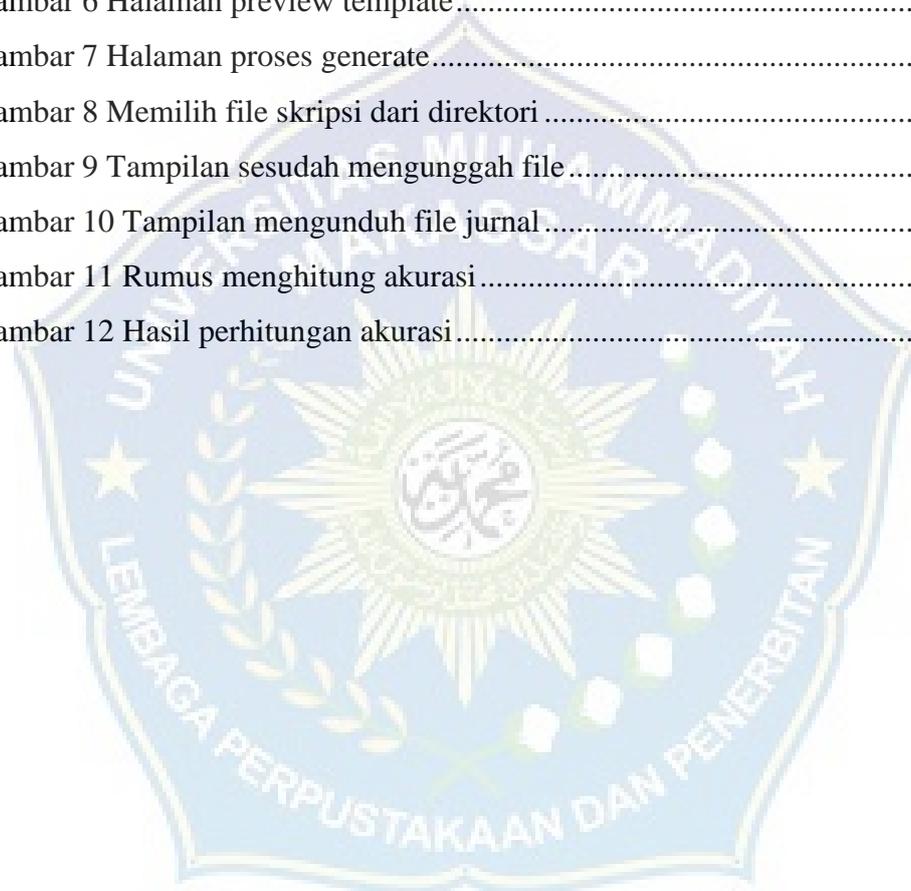
DAFTAR TABEL

Tabel 1 Penelitian Terkait	9
Tabel 2 Jadwal Kegiatan	13
Tabel 3 Black Box Testing	28
Tabel 4 Perbandingan waktu data uji coba	35



DAFTAR GAMBAR

Gambar 1 Kerangka Pikir Penelitian.....	12
Gambar 2 Flowchart.....	15
Gambar 3 Template Jurnal.....	27
Gambar 4 Kode menjalankan program.....	30
Gambar 5 Tampilan awal aplikasi.....	30
Gambar 6 Halaman preview template.....	31
Gambar 7 Halaman proses generate.....	32
Gambar 8 Memilih file skripsi dari direktori.....	33
Gambar 9 Tampilan sesudah mengunggah file.....	33
Gambar 10 Tampilan mengunduh file jurnal.....	34
Gambar 11 Rumus menghitung akurasi.....	36
Gambar 12 Hasil perhitungan akurasi.....	36

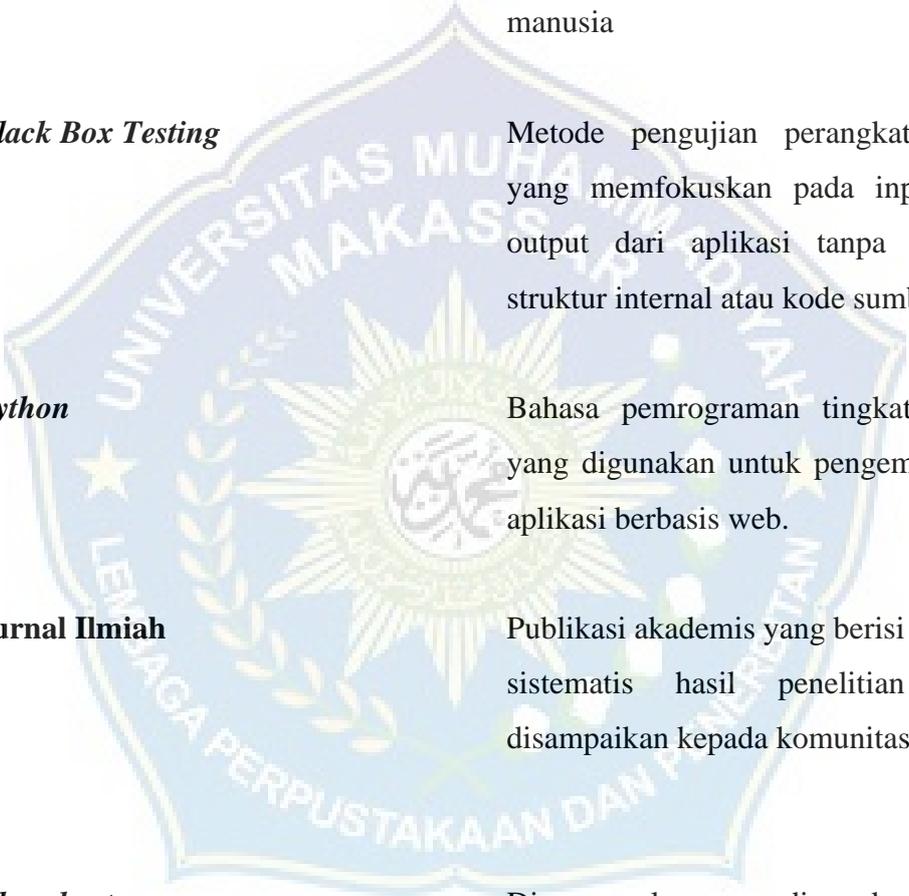


DAFTAR LAMPIRAN

Lampiran 1. Source Code.....	43
Lampiran 2 Data Skripsi Uji Coba.....	66
Lampiran 3 Data Jurnal Manual.....	66
Lampiran 4 Jurnal Hasil Generate.....	66
Lampiran 5 Link Github File Data.....	66
Lampiran 6 Hasil Uji Plagiat.....	67



DAFTAR ISTILAH



<i>Artificial Intelligence (AI)</i>	Kecerdasan yang ditunjukkan oleh mesin atau perangkat lunak yang mampu melakukan tugas-tugas yang biasanya membutuhkan kecerdasan manusia
<i>Black Box Testing</i>	Metode pengujian perangkat lunak yang memfokuskan pada input dan output dari aplikasi tanpa melihat struktur internal atau kode sumbernya.
<i>Python</i>	Bahasa pemrograman tingkat tinggi yang digunakan untuk pengembangan aplikasi berbasis web.
Jurnal Ilmiah	Publikasi akademis yang berisi laporan sistematis hasil penelitian yang disampaikan kepada komunitas ilmiah
<i>Flowchart</i>	Diagram alur yang digunakan untuk menggambarkan langkah-langkah atau proses dalam suatu sistem.
Ekstraksi Informasi	Proses pengambilan data penting dari teks untuk diolah atau digunakan lebih lanjut dalam aplikasi.

Pengujian *Impaired*

Pengujian yang dilakukan untuk membandingkan kinerja atau hasil aplikasi pada dua kelompok pengguna atau dua kondisi yang berbeda.

Behavioral Testing

Pengujian perangkat lunak yang dilakukan untuk mengamati hasil input dan output dari suatu aplikasi tanpa memperhatikan kode sumber.

Visual Studio Code

Editor kode sumber yang digunakan dalam pengembangan aplikasi berbasis Python.

Skripsi

Penelitian yang disusun oleh mahasiswa tingkat akhir sebagai persyaratan wajib untuk memperoleh gelar sarjana.

Template Jurnal

Format standar yang digunakan dalam penulisan artikel ilmiah untuk publikasi dalam jurnal.

Konversi Skripsi

Proses mengubah format skripsi menjadi format yang sesuai dengan jurnal ilmiah.

Ekstraksi Teks

Proses pengambilan teks penting dari dokumen untuk digunakan lebih lanjut dalam aplikasi.

BAB I

PENDAHULUAN

A. Latar Belakang

Peran dunia pendidikan dalam melestarikan budaya menulis karya ilmiah merupakan tuntutan dalam era globalisasi untuk menopang akreditasi perguruan tinggi serta mengasah kemampuan menulis mahasiswa. Wadah yang mampu melahirkan karya ilmiah yang berkualitas dapat dihasilkan oleh dosen maupun mahasiswa yang berkecimpung di dunia akademik (Riana et al., 2024).

Dalam konteks pendidikan tinggi, skripsi merupakan salah satu indikator kualitas pendidikan yang diberikan oleh perguruan tinggi. Skripsi adalah penelitian yang disusun oleh mahasiswa tingkat akhir dengan menerapkan kaidah ilmiah sesuai bidang keilmuan program studi atau jurusan sebagai persyaratan wajib untuk memperoleh gelar sarjana (Darmalaksana, 2022).

Pentingnya skripsi tidak hanya terbatas pada aspek akademis, tetapi juga pada kontribusinya terhadap pengembangan ilmu pengetahuan. Hasil penelitian dari skripsi dapat memberikan kontribusi yang berarti dalam menyelesaikan permasalahan yang ada, serta menjadi dasar bagi penelitian-penelitian selanjutnya. Oleh karena itu, institusi pendidikan harus memastikan bahwa proses penulisan skripsi berjalan dengan baik dan menghasilkan karya ilmiah yang berkualitas tinggi.

Sesuai edaran Kementerian Pendidikan dan Kebudayaan tahun 2012 bahwa mahasiswa wajib menulis tugas akhir dalam bentuk skripsi, tesis, dan disertasi serta menerbitkan tulisannya di jurnal ilmiah (Alunaza & Mentari, 2024).

Struktur penulisan artikel ilmiah umumnya meliputi pendahuluan, metode penelitian, hasil dan pembahasan, dan simpulan. Jurnal ilmiah rata-rata menerapkan kualifikasi kelayakan artikel terbit mencakup teknis dan substansi (Darmalaksana & Hambali, 2021). Sayangnya kemampuan menulis skripsi atau karya ilmiah masih dianggap sebagai kendala sehingga menyebabkan mahasiswa kesulitan dalam penyelesaian studinya (Bakri et al., 2023).

Publikasi skripsi ke jurnal ilmiah adalah langkah penting untuk berbagi temuan penelitian dengan komunitas akademik yang lebih luas. Proses ini melibatkan beberapa tahapan yang harus dilakukan dengan cermat. Proses publikasi skripsi ke jurnal ilmiah membutuhkan waktu dan usaha yang tidak sedikit.

Teknologi digital adalah teknologi yang pengoperasiannya tidak lagi membutuhkan banyak tenaga manusia dan bertujuan untuk menggunakan sistem otomatis dengan sistem komputer. Kata "digital" berasal dari bahasa Latin *digitus*, jari, dan mengacu pada salah satu alat komputer tertua. Ketika informasi disimpan, ditransmisikan dalam bentuk digital lalu diubah menjadi angka pada tingkat mesin yang paling sederhana (Hendri et al., 2023).

Perkembangan ilmu pengetahuan dan teknologi yang kian pesat semakin membawa perubahan dalam aktivitas kehidupan manusia sehari-hari dewasa ini. Salah satu kecanggihan teknologi yang saat ini menjadi isu menarik dan hangat adalah kehadiran kecanggihan teknologi berupa *Artificial Intelligence* atau kecerdasan buatan (selanjutnya disebut "AI") (Aras Samsithawrati et al., 2023).

Perancangan aplikasi generate skripsi ke jurnal dapat membantu mahasiswa dalam proses konversi skripsi ke jurnal secara otomatis. Aplikasi ini menggunakan beberapa *library python* untuk menganalisis teks skripsi dan mengekstrak informasi penting seperti pendahuluan, tinjauan pustaka, metodologi, hasil dan pembahasan, dan kesimpulan. Informasi yang diekstrak kemudian diformat sesuai dengan template jurnal.

Berdasarkan latar belakang diatas perancangan aplikasi ini sangat diharapkan dapat membantu mahasiswa dalam menghemat waktu dan tenaga dalam proses konversi skripsi ke jurnal. Selain itu, aplikasi ini juga diharapkan dapat meningkatkan kualitas jurnal yang dihasilkan oleh mahasiswa.

B. Rumusan Masalah

Berdasarkan latar belakang diatas, penulis dapat merumuskan masalah sebagai berikut:

1. Bagaimana merancang sistem aplikasi yang dapat mengubah teks skripsi menjadi jurnal yang sesuai dengan format template jurnal?
2. Bagaimana tingkat akurasi dan kualitas teks yang dihasilkan oleh sistem aplikasi tersebut dalam memenuhi standar jurnal AINET: Jurnal Informatika?

C. Tujuan Penelitian

Berdasarkan dari rumusan masalah di atas maka tujuan penelitian:

1. Melakukan penelitian terhadap proses perancangan sistem aplikasi yang dapat mengubah skripsi menjadi jurnal ilmiah yang sesuai dengan format template jurnal.
2. Menguji tingkat akurasi dan kualitas teks yang dihasilkan oleh sistem aplikasi dalam memenuhi standar jurnal AINET: Jurnal Informatika.

D. Manfaat Penelitian

Penelitian tentang Perancangan Aplikasi Generate Skripsi ke Jurnal ini diharapkan dapat bermanfaat baik secara teoritis maupun praktis.

1. Secara Teoris
 - a. Untuk pengembangan ilmu pengetahuan, terutama bagi dunia Teknik Informatika dan Ilmu Komputer.
2. Secara Praktis
 - a. Bagi Peneliti
 - 1) Sebagai portofolio untuk penelitian yang berguna pada masa yang akan datang
 - b. Bagi Mahasiswa
 - 1) Mempermudah dalam membantu mahasiswa dalam proses konversi skripsi ke jurnal secara otomatis.

c. Bagi Universitas

1) Sebagai bahan referensi untuk penelitian yang akan datang

E. Ruang Lingkup Penelitian

Ruang lingkup penelitian adalah batasan yang di sajikan fokus penelitian. Dari analisis rumusan masalah di atas dapat dirumuskan beberapa batasan masalah yaitu:

1. Penelitian ini lebih berfokus pada format template jurnal, penelitian ini tidak berfokus kepada isi konten dari skripsi.
2. Penelitian ini hanya fokus pada generate jurnal dari skripsi dengan format teks, penelitian ini tidak mencakup generate jurnal dari skripsi dengan format multimedia (contoh: gambar, tabel) dan format PDF.
3. Data yang digunakan yaitu skripsi/jurnal prodi Informatika.
4. Template jurnal yang di gunakan adalah template AINET: Jurnal Informatika.

F. Sistematika Penulisan

Secara garis besar penulisan laporan tugas akhir ini terbagi menjadi beberapa bagian bab yang tersusun sebagai berikut:

BAB I PENDAHULUAN

Bab ini menerangkan secara singkat, padat dan jelas mengenai latar belakang umum yaitu gambaran secara garis besar tentang isi laporan, rumusan masalah, tujuan dan manfaat, batasan permasalahan, metodologi yang di gunakan dan sistematika penulisan laporan tugas akhir.

BAB II TINJAUAN PUSTAKA

Bab ini berisi tentang teori – teori yang saling berhubungan dengan penelitian seperti metode dan algoritma yang melandasi penulis dalam melaksanakan skripsi.

BAB III METODE PENELITIAN

Bab ini membahas tentang metode penelitian dan alat yang digunakan untuk perancangan sistem.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi pemaparan hasil serta pembahasan terhadap perancangan pada bab sebelumnya, bagaimana suatu sistem dirancang berdasarkan algoritma yang digunakan dengan mengimplementasikan model *Natural Language Processing*. Serta dilakukan pula uji coba sistem untuk mendapatkan hasil yang sesuai dengan tujuan awal penelitian

BAB V KESIMPULAN

Bab ini merupakan bab terakhir yang berisi kesimpulan dan saran dari hasil penelitian, serta merupakan garis besar dari metode penelitian yang telah dilakukan. Kesimpulan adalah hasil akhir dari penelitian yang dilakukan, sedangkan Saran berisi tentang rekomendasi sesuai dengan keterbatasan yang ada pada sistem.

BAB II

TINJAUAN PUSTAKA

A. Landasan Teori

1. *Artificial Intelligence*

Kecerdasan buatan atau *Artificial Intelligence* merupakan salah satu teknologi yang cukup ramai diperbincangkan saat ini. Kecerdasan buatan sendiri dapat diartikan sebagai suatu teknologi mutakhir yang memungkinkan komputer melakukan pekerjaan selayaknya manusia. Hampir seluruh bidang di dunia telah berhasil memanfaatkan keberadaan dari teknologi satu yang ini. Salah satunya ialah di bidang pendidikan. Beberapa tahun terakhir, pendidikan tinggi telah banyak yang mulai memanfaatkan fungsi dari teknologi *Artificial Intelligence* ini. Contohnya para mahasiswa yang menggunakan teknologi kecerdasan buatan ini untuk menunjang kegiatan perkuliahan mereka, seperti dalam mengerjakan tugas, mencari informasi, mencari materi, dan lain sebagainya (Agustya et al., 2023).

Teknologi kecerdasan buatan memiliki banyak keuntungan yang dapat meningkatkan kehidupan manusia. Salah satu keuntungan terbesar adalah kemampuannya untuk mengambil keputusan dengan cepat dan akurat berdasarkan data. Hal ini dapat membantu mengurangi kesalahan manusia dan mempercepat waktu respon dalam situasi kritis. Selain itu, teknologi kecerdasan buatan dapat membantu meningkatkan efisiensi di berbagai sektor (Misnawati, 2023).

Kecerdasan buatan (AI) adalah kecerdasan yang ditunjukkan oleh mesin atau perangkat lunak. Ini adalah bidang studi akademis yang mempelajari tujuan menciptakan kecerdasan. Peneliti AI dan buku teks mendefinisikan bidang AI ini sebagai "studi dan desain agen cerdas", di mana agen cerdas adalah sistem yang memahami lingkungannya dan mengambil tindakan yang memaksimalkan peluang keberhasilannya (Gitakarma & Tjahyani, 2022).

Artificial Intelligence atau kecerdasan buatan adalah sistem komputer yang mampu melakukan tugas-tugas yang biasanya membutuhkan kecerdasan manusia. Proses yang terjadi dalam *Artificial Intelligence* mencakup learning, reasoning, dan self-correction. Proses ini mirip dengan manusia yang melakukan analisis sebelum memberikan keputusan (Zein, 2021).

Chatbot adalah salah satu sistem cerdas yang dihasilkan dari Pemrosesan Bahasa Alami atau *Natural Language Processing* (NLP) yang merupakan salah satu cabang dari Kecerdasan Buatan atau *Artificial Intelligence* (AI). *Chatbot* memungkinkan manusia dapat berkomunikasi dengan mesin menggunakan perantara bahasa alami (Aditama, 2020).

2. Skripsi

Skripsi adalah penelitian yang disusun oleh mahasiswa tingkat akhir dengan menerapkan kaidah ilmiah sesuai bidang keilmuan program studi atau jurusan sebagai persyaratan wajib untuk memperoleh gelar sarjana. Penulisan skripsi ditempuh melalui pengajuan proposal penelitian kepada dosen pembimbing, mengikuti seminar proposal, mendapat Surat Keputusan (SK) tentang topik skripsi, melakukan penelitian dan penulisan hasil penelitian melalui proses bimbingan, dan pelaksanaan sidang skripsi yang dikenal dengan istilah munaqasyah. Hasil penulisan skripsi wajib disebarluaskan kepada khalayak luas minimal diunggah di dalam repositori digital library pendidikan tinggi (Darmalaksana, 2022).

Skripsi adalah karya ilmiah dari pelaksanaan penelitian yang dilakukan secara mandiri oleh mahasiswa program sarjana, seperti yang di jelaskan oleh (Kasim et al., 2020). Bahwa skripsi ditulis oleh mahasiswa mengikuti kaedah dan etika keilmuan sebagai satu syarat untuk memperoleh gelar akademik (sarjana) sesuai dengan karakteristik keilmuan program studi.

Skripsi merupakan karangan ilmiah yang wajib ditulis oleh mahasiswa sebagai bagian dari persyaratan akhir pendidikan akademisnya. Skripsi bertujuan agar mahasiswa mampu menyusun dan menulisa suatu

karya ilmiah, sesuai dengan bidang ilmunya. Mahasiswa yang mampu menulis skripsi dianggap mampu memadukan pengetahuan dan keterampilannya dalam memahami dan menganalisis, menggambarkan, dan menjelaskan masalah yang berhubungan dengan bidang keilmuan yang diambilnya. Skripsi juga merupakan persyaratan untuk mendapatkan status sarjana (S1) di setiap Perguruan Tinggi (Ramdhan & Nufriana, 2019).

3. Jurnal Ilmiah

Sesuai edaran Kementerian Pendidikan dan Kebudayaan tahun 2012 bahwa mahasiswa wajib menulis tugas akhir dalam bentuk skripsi, tesis, dan disertasi serta menerbitkan tulisannya di jurnal ilmiah (Alunaza & Mentari, 2024).

Karya tulis ilmiah adalah suatu karya yang memuat dan mengkaji suatu masalah tertentu dengan menggunakan kaidah-kaidah keilmuan. Kaidah keilmuan yang dimaksud bahwa karya ilmiah menggunakan metode ilmiah di dalam membahas permasalahan, menyajikan kajian dengan menggunakan bahasa baku dan tata tulis ilmiah, serta menggunakan prinsip-prinsip keilmuan yakni bersifat objektif, logis, empiris, sistematis, lugas, jelas dan konsisten (Sakaria et al., 2023).

Menulis artikel merupakan kegiatan mengubah hasil penelitian skripsi, tesis, dan disertasi ke dalam bentuk yang lebih ringkas. Bentuk ringkas ini disebut sebagai artikel hasil penelitian. Artikel hasil penelitian akan lebih bermanfaat jika dipublikasikan pada jurnal nasional. Tujuan pengabdian ini adalah membekali mahasiswa dalam hal menulis artikel untuk dipublikasikan pada jurnal nasional terakreditasi dan mengirim melalui *open journal system* (Hasanudin et al., 2021).

Struktur penulisan artikel ilmiah umumnya meliputi pendahuluan, metode penelitian, hasil dan pembahasan, dan simpulan. Jurnal ilmiah rata-rata menerapkan kualifikasi kelayakan artikel terbit mencakup teknis dan substansi (Darmalaksana & Hambali, 2021).

Artikel ilmiah dalam bahasa Inggris disebut *research article*. Artikel ilmiah tertulis berisi laporan sistematis hasil penelitian atau hasil penelitian,

laporan tersebut disampaikan kepada komunitas ilmiah tertentu, yang merupakan khalayak khusus, dengan tujuan menyebarluaskan hasil penelitian dan kontribusi penulis penelitian. Mereka menggunakan pemikiran, komentar dan diskusi, termasuk bentuk lisan dan tulisan (Ma'rif & Fitria, 2021).

Publikasi hasil penelitian dimaksudkan agar berbagai masalah yang ditemukan dan inovasi yang dihasilkan dalam penelitian tersebut dapat menjadi referensi bagi guru-guru di tempat lain. Dalam hal ini, salah satu cara yang efektif untuk mempresentasikan hasil penelitian adalah dengan menulis dan mempublikasikannya dalam bentuk artikel ilmiah pada jurnal ilmiah (Udil, 2021).

B. Penelitian Terkait

Peneliti menggali banyak inspirasi dan referensi untuk menyusun skripsi ini melalui penelitian sebelumnya yang terkait dengan latar belakang masalah. Penelitian sebelumnya yang menjadi fokus mencakup:

Tabel 1 Penelitian Terkait

Penelitian	Tujuan/Kasus	Metode/Algoritma	Hasil
Darmanto, Novi Indah Pradasari, Eka Wahyudi (2023)	Pengembangan sistem deteksi plagiarisme untuk tugas akhir mahasiswa berbasis <i>Natural Language Processing</i> (NLP).	<i>Natural Language Processing</i> menggunakan algoritma <i>Jaro-Winkler</i> dan <i>TF-IDF</i> (<i>Term Frequency-Inverse Document Frequency</i>)	Sistem deteksi plagiarisme yang dibangun mampu mendeteksi kesalahan penulisan dan kemiripan dokumen dengan tingkat akurasi yang tinggi.

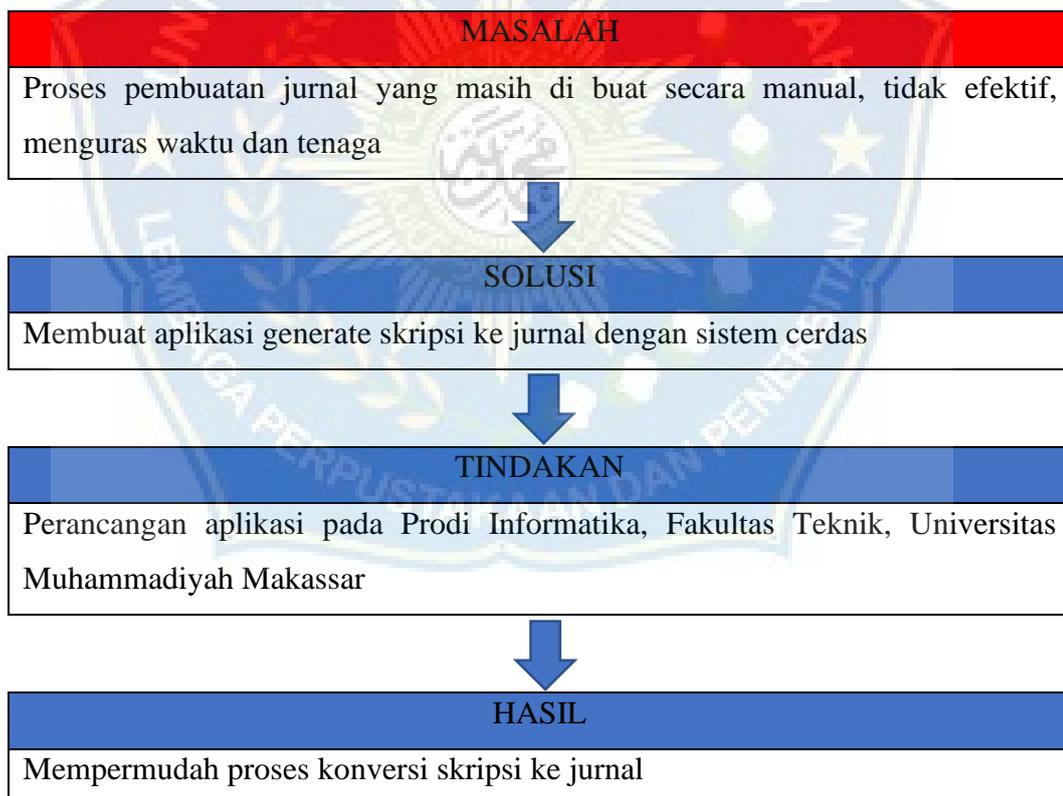
Elya (2021)	Wita	Mengidentifikasi kalimat ambigu pada surat kabar untuk mempermudah pembaca dalam memahami makna dari judul surat kabar yang ambigu.	Penerapan <i>Natural Language Processing</i> dengan metode <i>Shift Reduce Parsing</i>	Kalimat ambigu dapat diidentifikasi dengan proses parsing. <i>Natural Language Processing</i> dapat diterapkan untuk mengidentifikasi kalimat ambigu dengan metode <i>Shift Reduce Parsing</i> , dan ketiadaan tanda baca memperbesar peluang terjadinya kalimat ambigu
Humuntal Rumapea (2021)	Untuk mendeteksi kemiripan artikel melalui penggunaan kata kunci	Metode <i>Fuzzy String Matcing</i>	Artikel yang memiliki jumlah penggunaan kata kunci yang sama atau mirip dapat dikategorikan sebagai artikel dengan konten yang serupa. Menghasilkan persentase kemiripan yang dapat membantu pengguna dalam	

			mengetahui isi artikel tanpa harus membaca seluruh dokumen.
Robby Ananda, Hilda Ratna Dewi, Aulia Ramadina Salsabila (2020)	Pengembangan aplikasi "Cek-Aja," untuk memperbaiki dan meningkatkan kualitas karya tulis berbahasa Indonesia.	<i>Natural Language Processing (NLP)</i>	Aplikasi Cek-Aja merupakan inovasi teknologi yang bertujuan untuk mengatasi masalah <i>writing anxiety</i> dan meningkatkan minat menulis di kalangan masyarakat, memanfaatkan kombinasi kecerdasan buatan, NLP, dan <i>cloud computing</i> untuk memberikan umpan balik dan revisi otomatis terhadap tulisan.
Migunani, Kevin Aditama (2020)	Pemanfaatan <i>Natural Language Processing (NLP)</i> dan <i>pattern matching</i> dalam	<i>Natural Language Processing (NLP)</i> dan <i>Pattern matching</i>	Sistem guru virtual yang memanfaatkan <i>Natural Language Processing (NLP)</i> dan <i>pattern matching</i> ini

pembelajaran
melalui guru
virtual.

efektif membantu
siswa belajar
mandiri dengan
menyediakan
materi pelajaran,
membuat soal, dan
menjawab
pertanyaan sesuai
kurikulum
pendidikan.

C. Kerangka Berfikir



Gambar 1 Kerangka Pikir Penelitian

BAB III METODE PENELITIAN

A. Tempat dan Waktu Penelitian

Tempat Penelitian adalah suatu tempat atau objek yang akan dilakukan suatu penelitian. Penentuan lokasi penelitian merupakan langkah penting dalam proses penelitian karena memudahkan peneliti untuk melakukan penelitian. Lokasi penelitian yang dipilih oleh penulis adalah di Universitas Muhammadiyah Makassar. Tempat atau wilayah tersebut dipilih oleh penulis dengan alasan karena tempat penelitian atau lebih tepatnya Fakultas Teknik Universitas Muhammadiyah Makassar pada proses pembuatan jurnal ilmiah masih dilakukan dengan cara manual sehingga seringkali menguras banyak waktu dan tenaga mahasiswa, oleh karena itu peneliti memilih tempat penelitian tersebut dikarenakan masalah yang diangkat oleh peneliti sehaluan dengan tempat tersebut.

Waktu penelitian ini akan dilakukan dalam jangka waktu kurang lebih 2 bulan, yaitu dimulai pada bulan Juni 2024 sampai dengan Juli 2024.

Tabel 2 Jadwal Kegiatan

NO	KEGIATAN	MEI		JUNI				JULI				
		III	IV	I	II	III	IV	I	II	III	IV	
1	Studi Literatur											
2	Analisis Sistem											
3	Desain Sistem											
4	Implementasi Sistem											
5	Pengujian Sistem											
6	Penulisan Laporan											

B. Alat dan Bahan

Adapun alat dan bahan yang mendukung penelitian ini:

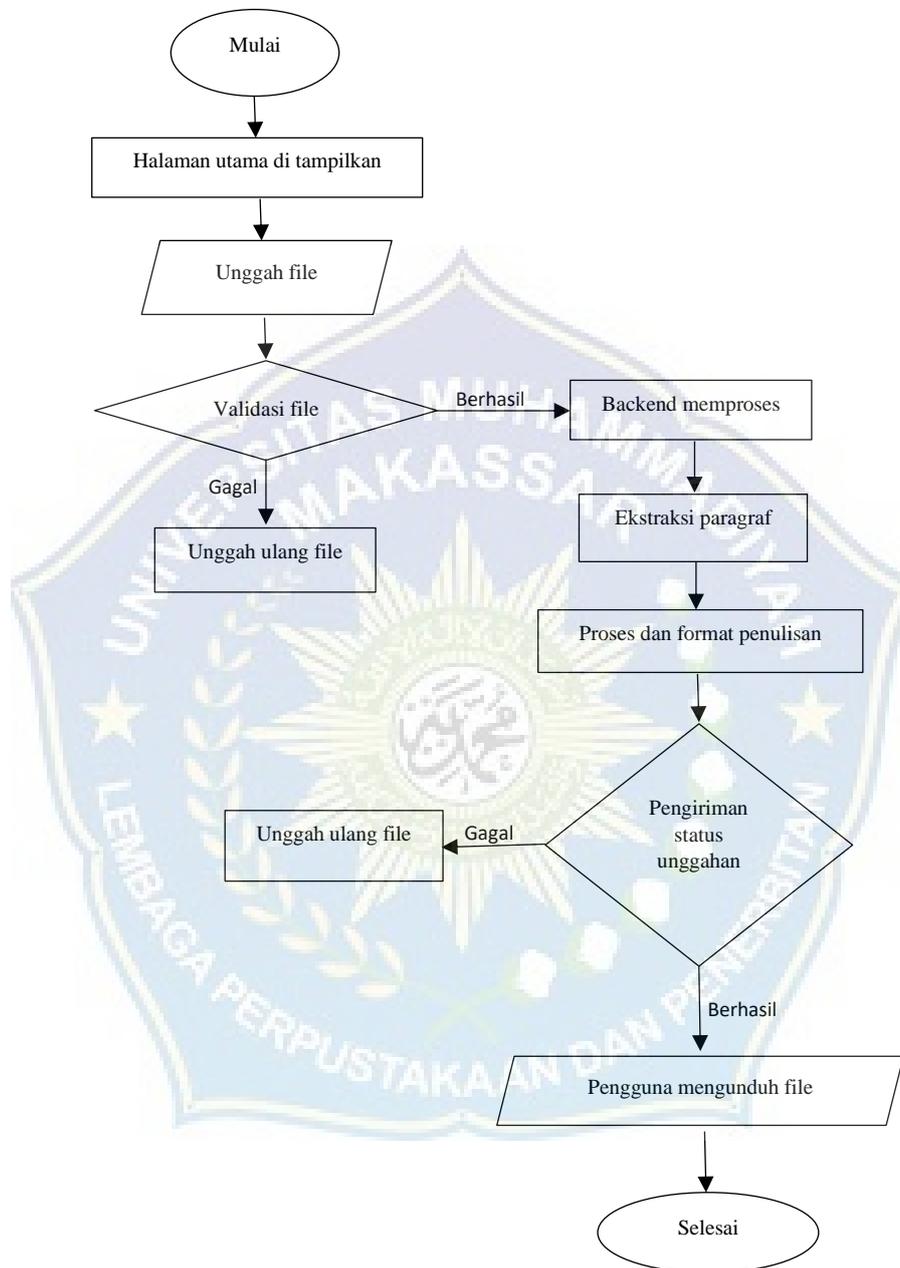
1. Hardware (Perangkat Keras)
 - a. Laptop ASUS A455L
 - b. RAM 4,00 GB
 - c. Intel Core i5–5200U 2.7GHz
2. Software (Perangkat Lunak)
 - a. *Visual Studio Code*
 - b. *Python*
 - c. *Java Script*

C. Perancangan Sistem

Perancangan sistem merupakan langkah-langkah yang terlibat dalam pemodelan, perancangan, dan konstruksi suatu sistem untuk memenuhi kebutuhan khusus dan memecahkan masalah tertentu pada penelitian ini. Tahapan ini melibatkan pemilihan teknologi yang sesuai, penentuan arsitektur sistem, desain antarmuka pengguna, pemilihan metode dan algoritma pemrograman, serta pengujian menyeluruh untuk memastikan kinerja optimal. Proses ini bertujuan untuk menghasilkan solusi yang efisien dan efektif sesuai dengan tujuan dan persyaratan yang telah ditetapkan.

Penggunaan *flowchart*, sangat bermanfaat dalam pembuatan dan pengembangan sistem karena menyediakan ilustrasi visual yang jelas terhadap alur kerja. *Flowchart* membantu dalam mengidentifikasi potensi masalah, memungkinkan perbaikan sebelum implementasi, dan menjadi alat komunikasi yang efektif. Selain itu, *flowchart* juga membantu dalam pemilihan algoritma, memandu proses pengujian, serta berfungsi sebagai bentuk dokumentasi yang berguna untuk menjelaskan konsep sistem kepada berbagai pemangku kepentingan. Dengan demikian, penggunaan *flowchart* membantu menyederhanakan kompleksitas dan meningkatkan pemahaman terhadap alur kerja sistem secara keseluruhan.

Berikut *flowchart* perancangan aplikasi generate skripsi ke jurnal berbasis web:



Gambar 2 Flowchart

1. Mulai: Pengguna menjalan aplikasi
2. Halaman utama di tampilkan: Menampilkan halaman utama aplikasi generate skripsi ke jurnal
3. Unggah file: Pengguna mengunggah file skripsi dengan format docx
4. Validasi file: Rute untuk mengunggah file skripsi jika berhasil dia lanjut ke tahap berikutnya, jika gagal unggah file kembali
5. Backend memproses: Mengekstraksi bagian-bagian penting dari dokumen skripsi seperti Judul, Abstrak, Pendahuluan, dll.
6. Ekstraksi paragraf: Mengisi template jurnal dengan informasi yang diekstraksi dari skripsi. Setiap bagian dalam template digantikan dengan informasi yang sesuai dari skripsi.
7. Proses dan format penulisan: Fungsi utama yang menggabungkan semua langkah sebelumnya untuk mengubah skripsi menjadi jurnal. Fungsi ini membaca dokumen skripsi, mengekstraksi informasi penting, menyesuaikan template jurnal, dan menyimpan hasilnya.
8. Pengguna mengunduh file: Rute yang memungkinkan pengguna untuk mengunduh file jurnal yang telah diproses.
9. Selesai: Skripsi berhasil generate sesuai dengan format template jurnal.

Flowchart ini secara keseluruhan menjelaskan bagaimana sebuah sistem otomatisasi dapat mengolah sebuah skripsi menjadi format jurnal melalui beberapa tahap yang melibatkan pengguna, *preprocessing* teks, ekstraksi informasi, pemrosesan, dan akhirnya menghasilkan dokumen dalam format yang diinginkan.

D. Teknik Pengujian Sistem

1. Black Box Testing

Teknik pengujian yang akan diterapkan adalah *black box testing*. *Black box testing* adalah metode pengujian perangkat lunak yang memfokuskan pada *input* dan *output* dari aplikasi tanpa melihat struktur internal atau kode sumbernya.

Pengujian *black box* hanya melakukan penilaian dari apakah sistem dapat memberikan *output* sesuai dengan *input*-an. Jika hasilnya sesuai, maka sistem dinyatakan dapat berfungsi dengan baik. Jika sistem gagal menjalankan prosedur yang diminta, maka dinyatakan membutuhkan perbaikan.

Metode pengujian *black box* juga dikenal sebagai *behavioral testing* yang dilakukan untuk mengamati hasil *input* dan *output* dari suatu aplikasi. Biasanya, pengujian *black box* dilakukan pada tahap akhir pengembangan aplikasi untuk mengetahui apakah aplikasi mampu berfungsi dengan baik.

Adapun teknik-teknik pengujian *black box* sebagai berikut:

a. Teknik Partisi Ekuivalensi

Teknik ini membagi data input menjadi dua bagian, yaitu nilai valid dan invalid, dan dilakukan pengujian secara terpisah. Kedua bagian tersebut harus menunjukkan perilaku yang sama untuk menyatakan pengujian berhasil.

b. Teknik Nilai Batas

Teknik nilai batas digunakan untuk menguji error pada nilai batas atas dan bawah suatu variabel. Teknik ini digunakan karena banyak aplikasi yang memiliki masalah pada nilai batas, terutama batas atas.

c. Teknik Pengujian Semua Pasangan

Teknik ini menguji seluruh kemungkinan kombinasi data diskrit. Data diskrit adalah data dengan nilai yang pasti dan terbatas, biasanya dalam bentuk bilangan bulat atau ganjil. Metode kombinasi ini digunakan untuk menguji aplikasi yang menggunakan input kotak centang, input tombol radio, kotak daftar, kotak teks, dan lain-lain.

d. Teknik Tabel Keputusan

Teknik ini dilakukan dengan pendekatan sistematis, di mana kombinasi input dirangkum dalam sebuah tabel. Teknik tabel

keputusan cocok untuk menguji fungsi yang memiliki hubungan logistik antara dua atau lebih input.

e. Teknik Transisi Keadaan

Teknik state transisi digunakan untuk mengetahui apakah fungsi software dapat bekerja jika mendapatkan nilai input yang berbeda. Hal ini karena keadaan output sistem bisa berubah tergantung pada kondisi atau peristiwa. Teknik transisi keadaan digunakan pada jenis aplikasi yang memberikan sejumlah percobaan tertentu untuk mengakses aplikasi.

f. Teknik Sebab-Akibat

Teknik sebab-akibat menggunakan grafik untuk menggambarkan hubungan antara penyebab error dan efeknya.

g. Teknik Menebak Kesalahan

Pada teknik ini, identifikasi error pada aplikasi didasarkan pada pengalaman dan pengetahuan penguji.

h. Teknik Kasus Penggunaan

Teknik ini menguji setiap fungsi software dengan cara menjalankan sistem dari tahap awal hingga akhir.

2. Pengujian *Impired*

Pengujian *impaired* (atau *unpaired*) dalam konteks pengujian sebuah aplikasi biasanya mengacu pada pengujian yang dilakukan untuk membandingkan kinerja atau hasil aplikasi pada dua kelompok pengguna atau dua kondisi yang berbeda yang tidak saling berhubungan. Berikut adalah cara bagaimana pengujian *impaired* bisa diterapkan dalam pengujian sebuah aplikasi:

Pengujian *impaired* untuk aplikasi otomatisasi yang membandingkan penggunaan aplikasi dengan proses manual bertujuan untuk melihat apakah aplikasi otomatisasi tersebut lebih efisien atau efektif dibandingkan dengan melakukan tugas secara manual. Berikut adalah cara untuk melakukan pengujian ini:

a. Efisiensi Waktu

Mengukur apakah penggunaan aplikasi otomatisasi mengurangi waktu yang dibutuhkan untuk menyelesaikan tugas dibandingkan dengan proses manual.

b. Interpretasi Hasil Akurasi

Akurasi biasanya dihitung berdasarkan jumlah keluaran yang benar atau sesuai dengan kriteria dibandingkan dengan total output. Jika semua jurnal yang dihasilkan oleh aplikasi otomatisasi dianggap benar atau sesuai

Berdasarkan hasil pengujian ini, peneliti dapat menentukan apakah aplikasi otomatisasi tersebut layak digunakan secara luas atau memerlukan perbaikan. Jika hasilnya positif, aplikasi dapat diimplementasikan secara permanen untuk menggantikan proses manual. Jika hasilnya tidak signifikan, evaluasi lebih lanjut mungkin diperlukan untuk mengidentifikasi area perbaikan pada aplikasi.

Pengujian ini membantu memastikan bahwa aplikasi otomatisasi benar-benar memberikan manfaat praktis bagi pengguna dan tidak hanya sekadar menambahkan kompleksitas tanpa keuntungan yang berarti.

E. Teknik Analisis Data

Teknik analisis data merupakan proses yang dilakukan dalam pengumpulan, pembersihan, transformasi, pengolahan data untuk mendapatkan informasi yang berguna dan bermakna. Tujuan utama dari teknik analisis data adalah untuk mengidentifikasi pola, tren, dan hubungan dalam data sehingga dapat digunakan untuk pengambilan keputusan dan memberikan wawasan yang lebih baik.

Kegiatan analisis data merupakan kegiatan yang tidak dapat diabaikan dalam proses penelitian karena pentingnya ketajaman dan kecermatan dalam penerapan instrumen analisis dalam menentukan keakuratan hasil.

Kesalahan dalam menentukan alat analisis dapat berakibat fatal pada kesimpulan yang dicapai, dan akan berdampak lebih besar pada pemanfaatan dan penerapan temuan penelitian.

Langkah – langkah analisis data untuk penelitian ini adalah sebagai berikut:

1. Pengumpulan Data

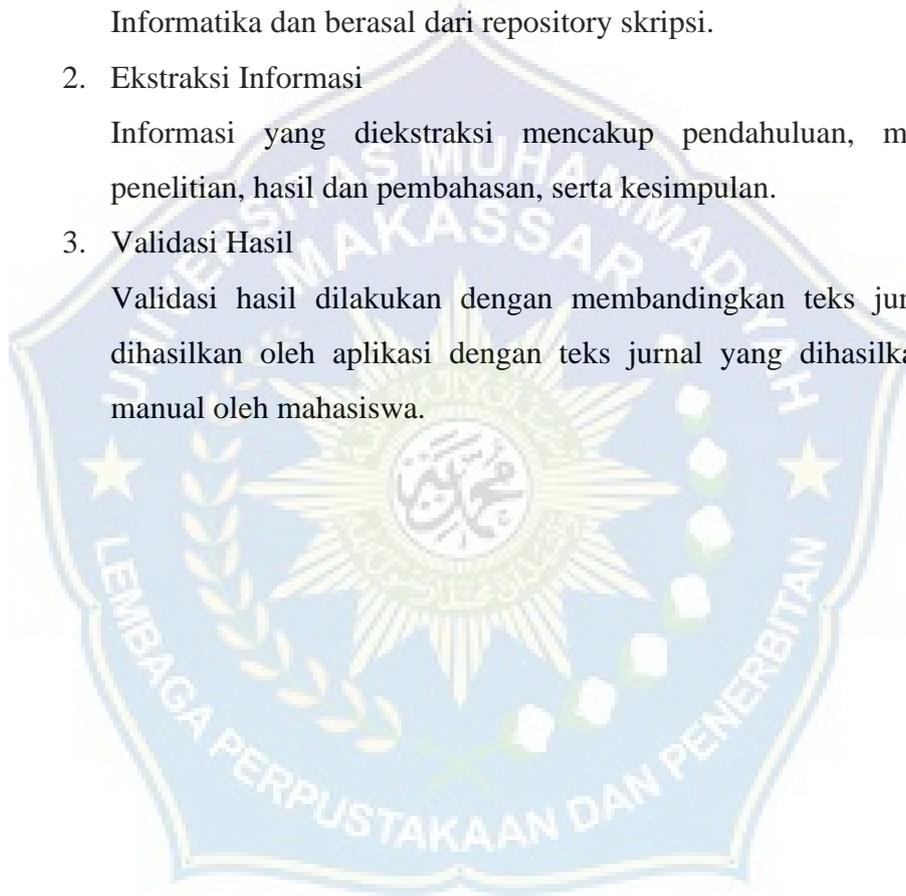
Data yang akan dianalisis berasal dari teks skripsi mahasiswa prodi Informatika dan berasal dari repository skripsi.

2. Ekstraksi Informasi

Informasi yang diekstraksi mencakup pendahuluan, metodologi penelitian, hasil dan pembahasan, serta kesimpulan.

3. Validasi Hasil

Validasi hasil dilakukan dengan membandingkan teks jurnal yang dihasilkan oleh aplikasi dengan teks jurnal yang dihasilkan secara manual oleh mahasiswa.



BAB IV HASIL DAN PEMBAHASAN

A. Perancangan Aplikasi

1. *Impor Library* dan Inisialisasi Aplikasi Web

```
from flask import Flask, request, send_file, jsonify,
render_template, redirect, url_for,
send_from_directory
from werkzeug.utils import secure_filename
import os
from docx import Document
from docx.shared import Pt, Inches
from docx.enum.text import WD_PARAGRAPH_ALIGNMENT

app = Flask(__name__)

UPLOAD_FOLDER = 'uploads'
PROCESSED_FOLDER = 'processed'
if not os.path.exists(UPLOAD_FOLDER):
    os.makedirs(UPLOAD_FOLDER)
if not os.path.exists(PROCESSED_FOLDER):
    os.makedirs(PROCESSED_FOLDER)
```

- a. **Flask:** Digunakan untuk membuat aplikasi web sederhana.
- b. **Werkzeug:** Berfungsi untuk membantu mengelola file yang diunggah.
- c. **os:** Digunakan untuk berinteraksi dengan sistem file.
- d. **python-docx:** Digunakan untuk membaca dan memanipulasi dokumen *Word (.docx)*.
- e. **os.makedirs:** Digunakan untuk membuat folder `'uploads'` dan `'processed'` jika belum ada

2. Mengunggah File Skripsi

```
#Fungsi untuk mengupload file skripsi dan
mengkonversinya menjadi jurnal
@app.route('/upload', methods=['POST'])
def upload_file():
    if 'file' not in request.files:
        return jsonify({"error": "No file part"}),
400
    file = request.files['file']
    if file.filename == '':
        return jsonify({"error": "No selected
file"}), 400
    if file:
        filename = secure_filename(file.filename)
        skripsi_path = os.path.join(UPLOAD_FOLDER,
filename)
        file.save(skripsi_path)
        template_path =
"template_jurnal/Template.docx"
        output_path = os.path.join(PROCESSED_FOLDER,
f"processed_{filename}")
        konversi_skripsi_ke_jurnal(skripsi_path,
template_path, output_path)
        return jsonify({"message": "File uploaded
and processed successfully", "download_url":
f"/download/{filename}"}), 200
```

def upload_file() adalah rute untuk mengunggah file skripsi. Jika file valid, ia akan disimpan ke folder ``uploads``, dan kemudian akan diproses menggunakan fungsi **konversi_skripsi_ke_jurnal**.

3. Ekstraksi Bagian – Bagian Penting Pada Skripsi

```
def ekstrak_bagian(dokumen):
    bagian = {
        "Judul": ekstrak_judul(dokumen),
        "Nama Penulis": ekstrak_nama(dokumen),
        "Institusi": ekstrak_institusi(dokumen),
        "Abstrak": [],
        "Abstract": [],
        "Pendahuluan": [],
        "Metode Penelitian": [],
        "Hasil dan Pembahasan": [],
        "Kesimpulan": [],
        "Referensi": [],
        "Keywords": ekstrak_keywords(dokumen),
        "Kata Kunci": ekstrak_kata_kunci(dokumen),
    }

    section_flags = {
        "dalam_abstrak": False,
        "dalam_abstract": False,
        "dalam_pendahuluan": False,
        "dalam_metode_penelitian": False,
        "dalam_hasil_dan_pembahasan": False,
        "dalam_kesimpulan": False,
        "dalam_referensi": False
    }
```

```

current_section = None
current_list_pendahuluan = []
current_list_metode_penelitian = []
current_list_hasil_dan_pembahasan = []
current_list_kesimpulan = []
current_list_referensi = []

#Fungsi ini digunakan untuk mengatur ulang semua
flags menjadi False
def reset_flags(flags):
    for key in flags:
        flags[key] = False

for para in dokumen.paragraphs:
    text = para.text.strip()
    if not text:
        continue

#Bagian ini digunakan untuk mengabaikan bagian
yang tidak perlu diambil
    if any(keyword.lower() in text.lower() for
keyword in ["keywords", "keyword", "kata kunci",
"ucapan terima kasih", "lampiran"]):
        reset_flags(section_flags)
        current_section = None
        continue

```

ekstrak_bagian (dokumen) berfungsi mengekstraksi bagian-bagian penting dari dokumen skripsi seperti Judul, Abstrak, Pendahuluan, Metode Penelitian, Hasil dan pembahasan, Kesimpulan dan Referensi.

4. Penyesuaian Template Jurnal

```
#Fungsi ini digunakan untuk mengganti bagian yang ada
di template dan menyesuaikan dengan format jurnal
def      sesuaikan_dengan_template(dokumen_template,
bagian):
```

```
    for para in dokumen_template.paragraphs:
```

sesuaikan_dengan_template(dokumen_template, bagian) digunakan untuk mengisi template jurnal dengan informasi yang diekstraksi dari skripsi. Setiap bagian dalam template digantikan dengan informasi yang sesuai dari skripsi.

5. Konversi ke Jurnal

```
def konversi_skripsi_ke_jurnal(path_skripsi,
path_template, path_output):
    dokumen_skripsi = Document(path_skripsi)
    dokumen_template = baca_template(path_template)
    bagian = ekstrak_bagian(dokumen_skripsi)
    sesuaikan_dengan_template(dokumen_template,
bagian)
    simpan_dokumen_baru(dokumen_template,
path_output)
```

konversi_skripsi_ke_jurnal(path_skripsi, path template, path output) adalah fungsi utama yang menggabungkan semua langkah sebelumnya untuk mengubah skripsi menjadi jurnal. Fungsi ini membaca dokumen skripsi, mengekstraksi informasi penting, menyesuaikan template jurnal, dan menyimpan hasilnya sebagai dokumen baru.

6. Mengunduh File Jurnal

```
#Fungsi ini digunakan untuk mendownload file jurnal
yang telah dihasilkan
@app.route('/download/<filename>', methods=['GET'])
def download_file(filename):
```

```

        processed_path = os.path.join(PROCESSED_FOLDER,
f"processed_{filename}")
        if os.path.exists(processed_path):
            return send_file(processed_path,
as_attachment=True)
        else:
            return jsonify({"error": "File not found"}),
404

```

download_file (file name) adalah rute yang memungkinkan pengguna untuk mengunduh file jurnal yang telah diproses. File jurnal disimpan di folder `processed` dan dapat diunduh melalui URL yang disediakan.

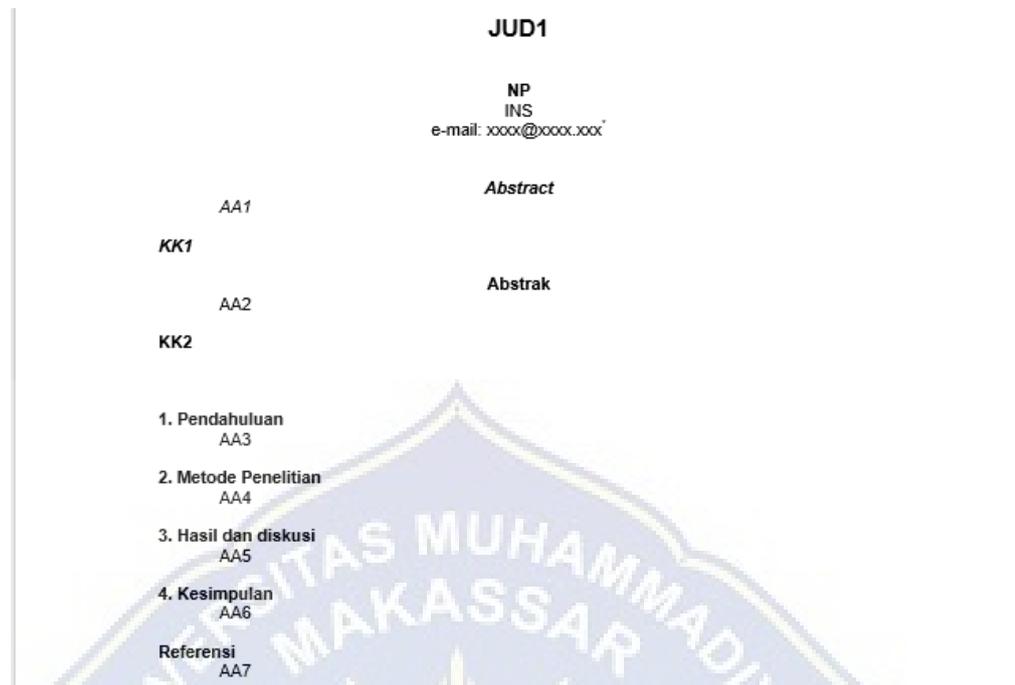
B. Proses Generate Skripsi ke Jurnal

1. Ekstraksi Bagian-Bagian Skripsi

Ekstraksi bagian-bagian skripsi melibatkan identifikasi dan pengambilan konten dari judul, abstrak, *abstract*, pendahuluan, metode penelitian, hasil dan pembahasan, kesimpulan, serta daftar pustaka. Setiap bagian diekstraksi berdasarkan deteksi kata kunci yang umum digunakan dalam skripsi. Proses ini menggunakan struktur kondisional untuk mengenali bagian-bagian dokumen berdasarkan kata kunci seperti 'ABSTRAK', 'ABSTRACT', 'PENDAHULUAN', 'METODE PENELITIAN', 'HASIL DAN PEMBAHASAN', 'KESIMPULAN', dan 'REFERENSI'.

2. Inisialisasi Template Jurnal

Setelah bagian-bagian dari skripsi berhasil diekstraksi, tahap berikutnya adalah memasukkan konten tersebut ke dalam template jurnal. Proses ini melibatkan penggantian *placeholder* dalam template dengan teks yang sesuai dari hasil ekstraksi.



Gambar 3 Template Jurnal

Placeholder seperti 'JUD1' untuk mengekstraksi bagian judul pada skripsi, 'NP' untuk mengekstraksi nama penulis, 'INS' untuk mengekstraksi Instansi atau Universitas, 'AA1' untuk mengekstraksi bagian *abstract*, 'KK1' untuk mengekstraksi *keywords* pada *abstract*, 'AA2' untuk mengekstraksi bagian abstrak, 'KK2' untuk mengekstraksi kata kunci pada abstrak, 'AA3' untuk mengekstraksi bagian pendahuluan, 'AA4' untuk mengekstraksi bagian metode penelitian, 'AA5' untuk mengekstraksi bagian hasil dan pembahasan, 'AA6' untuk mengekstraksi bagian kesimpulan dan 'AA7' referensi untuk mengekstraksi bagian daftar pustaka pada skripsi. Penyesuaian ini juga mencakup format teks, seperti ukuran *font*, perataan teks, dan indentasi paragraf, yang disesuaikan dengan standar penulisan jurnal.

C. Pengujian Sistem

Pengujian sistem dilakukan dengan metode *Black box testing*. Uji coba ini berdasarkan pada detail aplikasi seperti tampilan aplikasi dan kesesuaian alur fungsi dengan proses yang diinginkan dengan memperlihatkan rancangan yang di buat. Pengujian ini tidak menguji dan melihat *source code* program.

1. Pengujian *black box* dengan teknik Tabel Keputusan

Teknik ini dilakukan dengan pendekatan sistematis, di mana kombinasi input dirangkum dalam sebuah tabel. Teknik tabel keputusan cocok untuk menguji fungsi yang memiliki hubungan logistik antara dua atau lebih input.

Berikut adalah hasil pengujian *black box* dengan teknik tabel keputusan:

Tabel 3 Black Box Testing

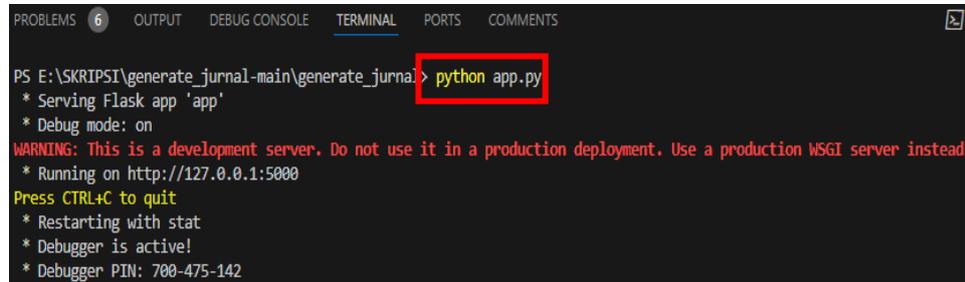
No	Test Case	Skenario Pengujian	Hasil yang di Harapkan	Hasil	
1.	Menjalankan aplikasi	Pengguna menjalankan aplikasi pada terminal menggunakan perintah <code>python app.py</code>	Terminal menunjukkan proses <i>running</i> perintah <code>python `app.py`</code>	[v]Berhasil []Gagal	
2.	Memulai aplikasi	Pengguna mengklik lokal yang sedang berjalan	Sistem beralih ke halaman utama aplikasi <i>generate</i> skripsi yang sedang berjalan pada <code>port http://127.0.0.1:5000</code>	[v]Berhasil []Gagal	
3.	Mimilih template	Pengguna mengklik template dengan kebutuhan	Pilih sesuai dengan kebutuhan	Sistem berhasil berpindah ke halaman template jurnal	[v]Berhasil []Gagal
4.	Memilih template AINET	Pengguna mengklik AINET	jurnal	Sistem berhasil menampilkan	[v]Berhasil []Gagal

			preview penulisan format jurnal Ainet	
5.	<i>Generate</i> jurnal	Pengguna mengklik Lanjutkan ke <i>generate</i> jurnal	Sistem berpindah ke halaman pengonversian jurnal	[v]Berhasil []Gagal
6.	Mengunggah file dengan format yang benar	Pengguna memilih skripsi dengan format docx	Sistem berhasil membaca skripsi dengan format docx. dan skripsi berhasil di generate menjadi jurnal	[v]Berhasil []Gagal
7.	Mengunggah file dengan format salah	Pengguna memilih skripsi dengan format PDF	Sistem tidak bisa membaca skripsi dengan format PDF	[v]Berhasil []Gagal
8.	Mengunduh Jurnal	Pengguna mengklik button Download Jurnal Anda	Sistem memproses skripsi dan mengduh jurnal yang berhasil di generate	[v]Berhasil []Gagal

2. Pengujian *Black Box* dengan teknik Kasus Pengguna

Pengujian *Use Case* atau biasa disebut pengujian kasus pengguna adalah teknik pengujian *black box* yang menggunakan *use case* untuk mengidentifikasi kasus pengujian. Teknik ini menguji setiap fungsi *software* dengan cara menjalankan sistem dari tahap awal hingga akhir.

a. Menjalankan program



```
PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
PS E:\SKRIPSI\generate_jurnal-main\generate_jurnal > python app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 700-475-142
```

Gambar 4 Kode menjalankan program

Ini adalah tampilan terminal yang menunjukkan bagaimana cara menjalankan aplikasi pada *Viscod* menggunakan perintah **python app.py**. Ini memastikan bahwa pengguna atau pengembang dapat dengan mudah memulai aplikasi. Terminal menunjukkan proses running aplikasi *Flask* (**app.py**). Perintah **python `app.py`** dijalankan untuk memulai server lokal pada **127.0.0.1:5000**. Aplikasi sedang dalam mode debug dan sedang memantau perubahan pada file **app.py**.

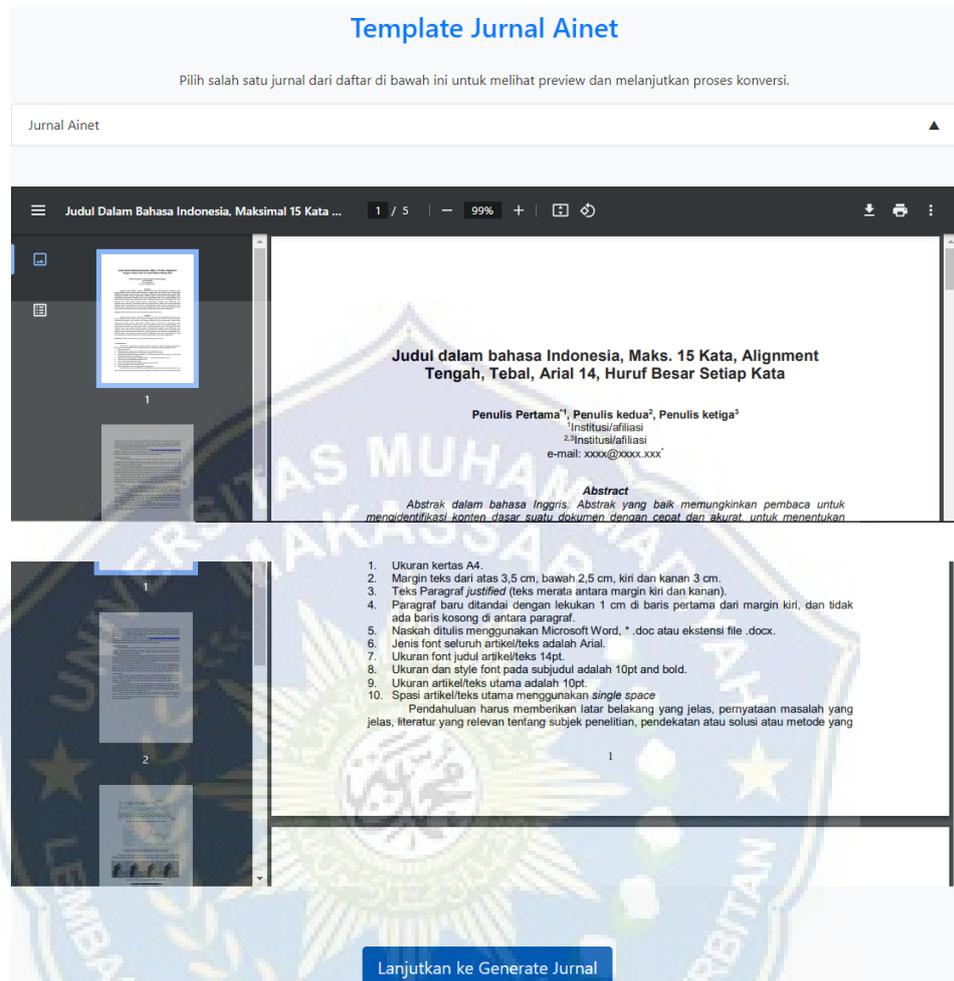
b. Tampilan awal



Gambar 5 Tampilan awal aplikasi

Ini adalah tampilan awal aplikasi "Generate Skripsi ke Jurnal". Pengguna di minta memilih format template jurnal berdasarkan kebutuhannya. Pada halaman utama juga terdapat penjelasan tentang bagaimana cara menggunakan aplikasi pengonversi skripsi ke jurnal.

c. Halaman preview template jurnal AINET



Gambar 6 Halaman preview template

Ini adalah halaman setelah pengguna mengklik button Pilih Template Jurnal Anda. Sistem akan berpindah ke halaman preview template jurnal, halaman ini berisi contoh format penulisan jurnal AINET. Pada halaman ini pengguna mengklik Lanjutkan ke Generate Jurnal untuk proses selanjutnya.

d. Halaman proses Generate

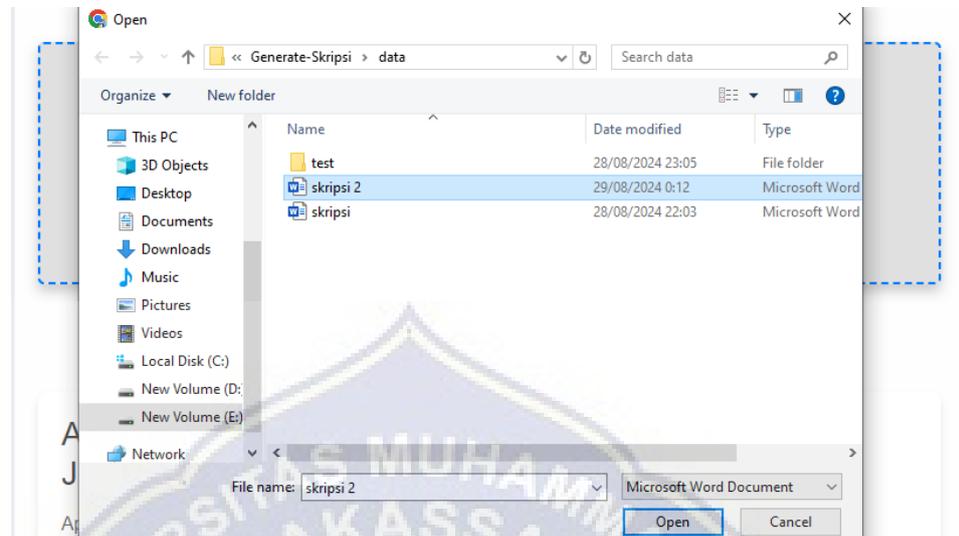


Gambar 7 Halaman proses generate

Pengguna diminta untuk mengunggah *file .docx* melalui area *drag and drop* yang disediakan atau dengan mengklik untuk memilih *file*. Tampilan ini berfungsi sebagai antarmuka pengguna yang sederhana untuk memulai proses konversi skripsi ke jurnal.

Pada halaman ini juga berisi beberapa penjelasan tentang pengonversian jurnal, bagaimana proses dari *backend* bekerja serta panduan penggunaan aplikasi.

e. Mengunggah *file* skripsi



Gambar 8 Memilih file skripsi dari direktori

Tampilan ini menunjukkan proses memilih *file* `skripsi.docx` dari direktori pengguna. Ini adalah bagian penting dalam proses pengujian untuk memastikan bahwa pengguna dapat dengan mudah memilih file yang akan diunggah.

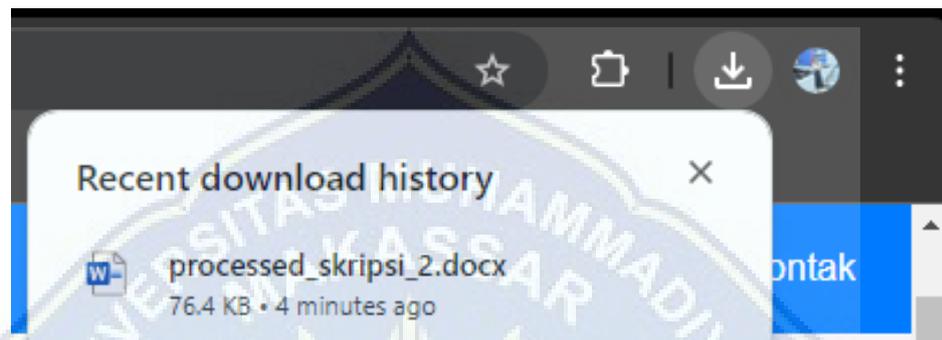
f. Tampilan setelah file di unggah



Gambar 9 Tampilan sesudah mengunggah file

Setelah file `skripsi 2.docx` berhasil diunggah, aplikasi memberikan opsi untuk mengunduh file jurnal yang telah diproses. Ada pesan konfirmasi yang menyatakan bahwa file telah berhasil diubah menjadi jurnal. Pengujian ini memastikan bahwa aplikasi berhasil mengonversi file dan memberikan umpan balik yang jelas kepada pengguna.

g. Mengunduh file jurnal



Gambar 10 Tampilan mengunduh file jurnal

Tampilan ini menunjukkan bahwa file jurnal yang berhasil diunduh disimpan dengan nama `processed_skripsi_2.docx` dan telah berhasil diunduh oleh pengguna. Ini adalah tahap akhir dalam proses pengujian, memastikan bahwa file yang telah dikonversi dapat diakses dan diunduh dengan benar.

3. Pengujian *Impaired*

Dalam konteks aplikasi otomatisasi, *impaired test* merujuk pada pengujian yang dilakukan untuk menilai kinerja atau fungsionalitas sistem otomatisasi. Tujuan dari pengujian ini adalah untuk mengidentifikasi bagaimana sistem atau perangkat lunak berfungsi.

Pengujian data terdiri dari 10 skripsi yang berasal dari skripsi mahasiswa Program Studi Informatika Universitas Muhammadiyah Makassar yang akan digunakan sebagai objek uji coba dalam proses pengembangan dan evaluasi aplikasi otomatisasi pembuatan jurnal ilmiah.

Berikut adalah hasil pengujian yang telah dilakukan:

a. Menghitung Efisiensi Waktu

Tabel 4 Perbandingan waktu data uji coba

Skripsi Uji Coba	Lama Waktu Pembuatan Jurnal	
	Otomatis	Manual
Skripsi 1	1 Menit 20 Detik	1 Jam 30 Menit
Skripsi 2	1 Menit 30 Detik	1 Jam 50 Menit
Skripsi 3	1 Menit 30 Detik	1 Jam 50 Menit
Skripsi 4	1 Menit 20 Detik	1 Jam 30 Menit
Skripsi 5	1 Menit 30 Detik	1 Jam 50 Menit
Skripsi 6	1 Menit 30 Detik	1 Jam 50 Menit
Skripsi 7	1 Menit 10 Detik	1 Jam 30 Menit
Skripsi 8	1 Menit 30 Detik	1 Jam 50 Menit
Skripsi 9	1 Menit 30 Detik	1 Jam 50 Menit
Skripsi 10	1 Menit 20 Detik	1 Jam 30 Menit
Jumlah	10 Menit	10 Jam

Pada tabel di atas dapat disimpulkan aplikasi generate skripsi berhasil mengkonversi 10 skripsi dalam waktu 10 menit, artinya 1 skripsi di ubah menjadi jurnal dalam waktu kurang dari 2 menit. Sedangkan penulisan jurnal dengan proses manual membutuhkan waktu 17 jam (1.020 menit), artinya 1 jurnal selesai dalam waktu kurang dari 2 jam. Jadi efisiensi waktu otomatisasi jauh lebih tinggi (70 kali lebih cepat).

b. Interpretasi Hasil Akurasi

Pengujian data yang terdiri dari 10 skripsi yang berasal dari skripsi mahasiswa Program Studi Informatika Universitas Muhammadiyah Makassar yang akan digunakan sebagai objek uji coba dalam proses pengembangan dan evaluasi aplikasi otomatisasi pembuatan jurnal ilmiah.

Skripsi-skripsi tersebut akan diolah oleh aplikasi otomatisasi dengan tujuan untuk menghasilkan jurnal format AINET secara otomatis. Hasil jurnal yang dihasilkan oleh aplikasi ini kemudian akan dibandingkan dengan 10 jurnal lain yang disusun secara manual. Perbandingan tersebut bertujuan untuk menilai sejauh mana akurasi, konsistensi, dan efektivitas aplikasi otomatisasi dalam menghasilkan jurnal yang sesuai dengan standar ilmiah.

Menghitung akurasi dari sebuah aplikasi otomatisasi yang mengonversi skripsi menjadi jurnal dapat dilakukan dengan membandingkan hasil yang dihasilkan oleh aplikasi dengan hasil yang diharapkan atau standar. Untuk menghitung akurasi menggunakan rumus:

$$\text{Akurasi} = \left(\frac{\text{Jumlah jurnal yang berhasil di konversi}}{\text{Jumlah total skripsi yang di uji coba}} \right) \times 100\%$$

Gambar 11 Rumus menghitung akurasi

Dimana:

- Jumlah jurnal yang berhasil di konversi adalah 10 (skripsi yang berhasil dikonversi dengan benar sesuai dengan template).
- Jumlah total data uji adalah 10 (skripsi yang diuji).

Menghitung menggunakan rumus di atas:

$$\text{Akurasi} = \left(\frac{10}{10} \right) \times 100\% = 100\%$$

Gambar 12 Hasil perhitungan akurasi

Dengan demikian dapat di simpulkan , tingkat akurasi dari aplikasi konversi skripsi menjadi jurnal pada uji coba ini adalah 100%. Ini berarti bahwa aplikasi berhasil mengonversi 10 skripsi menjadi 10 jurnal secara benar sesuai dengan template yang ditetapkan.

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan pengujian yang telah dilakukan dalam penelitian, beberapa kesimpulan dapat ditarik oleh penulis:

1. Perancangan sistem aplikasi yang dibangun dapat berjalan seperti yang diharapkan. Kode yang dikembangkan dalam perancangan sistem aplikasi ini berhasil menggenerate skripsi menjadi jurnal ilmiah yang sesuai dengan format template jurnal.
2. Dari rangkaian penelitian yang telah diuraikan dalam Perancangan Sistem Aplikasi Generate Skripsi ke Jurnal Berbasis Web. Tingkat akurasi dari aplikasi konversi skripsi menjadi jurnal pada uji coba ini adalah 100%. Dapat disimpulkan bahwa kode ini berhasil mengekstraksi bagian-bagian penting dari skripsi, dan dapat menyesuaikan dengan format template jurnal.

B. Saran

Dari hasil penelitian yang telah dilakukan maka dapat disampaikan beberapa saran, yaitu:

1. Untuk pengembangan lebih lanjut, diharapkan untuk meningkatkan kemampuan kode agar aplikasi generate skripsi dalam mengenali struktur dan konteks dapat lebih kompleks, serta menambahkan fitur yang dapat menangani berbagai format dokumen yang lebih bervariasi.
2. Disarankan juga untuk melakukan pengujian lebih lanjut pada berbagai dokumen skripsi dengan struktur yang berbeda-beda untuk memastikan fleksibilitas dan keandalan dari sistem yang dikembangkan.

DAFTAR PUSTAKA

- Aditama, K. (2020). PEMANFAATAN NATURAL LANGUAGE PROCESSING DAN PATTERN MATCHING DALAM PEMBELAJARAN MELALUI GURU VIRTUAL. *ELKOM*, 13(1), 121–133. <http://ejurnal.stekom.ac.id/index.php/home□page121>
- Agustya, K., Salsabilla, Z., Diva, T., Hadi, F., Pratiwi, W., & Mukarromah, S. (2023). *PENGARUH PENGGUNAAN KECERDASAN BUATAN TERHADAP MAHASISWA DI PERGURUAN TINGGI*.
- Akbar Nasution, M., Fitri, A., Sabila Rizwine, K., Styphen Silaban, V., Khoirani, F., Iskandar Ps, W. V, Baru, K., & Percut Sei Tuan, K. (2024). *Implementasi NLP Dalam Pembuatan Chatbot Customer Service Publisher Jurnal Studi Kasus LARISMA* (Vol. 1, Issue 1).
- Alunaza, H., & Mentari. (2024). Edukasi Penulisan Artikel Jurnal dan Strategi Publikasi Ilmiah Bagi Mahasiswa Jurusan Ilmu Administrasi Universitas Tanjungpura. *I-Com: Indonesian Community Journal*, 4(1), 125–135. <https://doi.org/10.33379/icom.v4i1.3954>
- Ananda, R., Ratna Dewi, H., & Ramadina Salsabila, A. (2020). *CEK-AJA: APLIKASI ASISTEN DIGITAL BERBASIS NATURAL LANGUAGE PROCESSING GUNA MEMPERBAIKI DAN MENINGKATKAN KUALITAS KARYA TULIS BERBAHASA INDONESIA*.
- Aras Samsithawrati, P., Kata kunci, A., Buatan, K., Kreatif, K., Intelektual, K., & Hukum, S. (2023). Artificial Intelligence dan Kreatifitas Digital: Subyek Hukum dan Sarananya Dalam Perspektif Kekayaan Intelektual. *Jurnal Kertha Patrika*, 45(3), 2023. <https://doi.org/10.24843/KP.2023.v45.i03.p03>
- Azhar, R., Surahman, A., & Juliane, C. (2022). Analisis Sentimen Terhadap Cryptocurrency Berbasis Python TextBlob Menggunakan Algoritma

Naïve Bayes. In *Jurnal Sains Komputer & Informatika (J-SAKTI)* (Vol. 6, Issue 1).

Bakri, S., Rahayu, S., & Wardani, T. (2023). *Pengabdian Kepada Masyarakat Pembekalan Penulisan Skripsi dan Karya Ilmiah Bagi Mahasiswa Fakultas Ekonomi dan Bisnis Institut Bakti Nusantara.*

Darmalaksana, W. (2022). *PANDUAN PENULISAN SKRIPSI DAN TUGAS AKHIR.*

Darmalaksana, W., & Hambali, A. Y. R. (2021). *Penulisan Tugas Akhir Skripsi dalam Bentuk Artikel Ilmiah: Studi Kasus Fakultas Ushuluddin UIN Sunan Gunung Djati Bandung.*

Eka Rosyadi, H., Amrullah, F., David Marcus, R., & Rahman Affandi, R. (2020). Rancang Bangun Chatbot Informasi Lowongan Pekerjaan Berbasis Whatsapp dengan Metode NLP (Natural Language Processing). *BRILIANT: Jurnal Riset Dan Konseptual*, 5. <https://doi.org/10.28926/briliant>

Gitakarma, S. M., & Tjahyani, S. A. P. L. (2022). PERANAN INTERNET OF THINGS DAN KECERDASAN BUATAN DALAM TEKNOLOGI SAAT INI. In *Jurnal Komputer dan Teknologi Sains (KOMTEKS)* (Vol. 1, Issue 1).

Hasanudin, C., Fitriainingsih, A., Rosyida, F., & Noeruddin, A. (2021). PELATIHAN MENULIS ARTIKEL UNTUK JURNAL NASIONAL TERAKREDITASI DAN MENGIRIM MELALUI OPEN JOURNAL SYSTEM (OJS). *Indonesian Journal Of Community Service*, 1.

Hendri, S., Wahyuddin, W., Angga, S., Permana, A., Sembiring, S., Jurnaidi, A., Jatmiko, W., Nugroho, W., Rahajeng, E., Kurnaedi, D., Taufik, R., Bau, R. L., Adhicandra, I., Tubagus, Y., & Rivanthio, R. (2023). *TEKNOLOGI DIGITAL DI ERA MODERN.* www.globaleksekitifteknologi.co.id

- Ismail, & Elihami. (2019). Pelatihan Penyusunan Artikel Publikasi Ilmiah bagi Mahasiswa Perguruan Tinggi STKIP Muhammadiyah Enrekang. *Copyright@2019-Maspul Journal of Community Empowerment*, 1(1), 12–20. <https://doi.org/10.33487/Copyright@2019>
- Kasim, A., Hadjaratie, L., & Dai, R. H. (2020). Rancang Bangun Sistem Informasi Skripsi dan Kerja Praktik Berbasis Web. *Jambura Journal of Informatics*, 2(2), 95–107. <https://doi.org/10.37905/jji.v2i2.5331>
- Laksmi Maitri, A., & Sutopo, J. (2019). *RANCANG BANGUN CHATBOT SEBAGAI PUSAT INFORMASI LEMBAGA KURSUS DAN PELATIHAN MENGGUNAKAN PENDEKATAN NATURAL LANGUAGE PROCESSING*.
- Ma'ruf, H. M., & Fitria, N. T. (2021). *Pelatihan Penulisan Artikel Ilmiah dari Skripsi dan Tesis Untuk Mahasiswa Serta Cara Publikasinya ke Jurnal Nasional*. <https://www.researchgate.net/publication/355473502>
- Misnawati. (2023). ChatGPT: Keuntungan, Risiko, Dan Penggunaan Bijak Dalam Era Kecerdasan Buatan. In *Jurnal Prosiding Mateandrau* (Vol. 2, Issue 1).
- Pasek, P., Mahawardana, O., Sasmita, G. A., Agus, P., & Pratama, E. (2022). Analisis Sentimen Berdasarkan Opini dari Media Sosial Twitter terhadap “Figure Pemimpin” Menggunakan Python. In *JITTER-Jurnal Ilmiah Teknologi dan Komputer* (Vol. 3, Issue 1).
- Prasetyo, V. R., Benarkah, N., & Chrisintha, V. J. (2021). Implementasi Natural Language Processing Dalam Pembuatan Chatbot Pada Program Information Technology Universitas Surabaya. *Teknika*, 10(2), 114–121. <https://doi.org/10.34148/teknika.v10i2.370>
- Ramdhan, A. N., & Nufriana, A. D. (2019). *RANCANG BANGUN DAN IMPLEMENTASI SISTEM INFORMASI SKRIPSIONLINE*

BERBASIS WEB-INTECH. *Jurnal Ilmiah INTECH Information Technology Journal of UMUS*.

Riana, Lase, F., Hatawa, N., Hura, D., Zega, I., & Lase, P. B. (2024). *STRATEGI EFEKTIF UNTUK MENGUBAH SKRIPSI MENJADI ARTIKEL ILMIAH PANDUAN PELATIHAN KONVERSI YANG SUKSES*. 12(1). <https://doi.org/10.37081/ed.v12i1.5716>

Rifano, E. J., Fauzan, Abd. C., Makhi, A., Nadya, E., Nasikin, Z., & Putra, F. N. (2020). Text Summarization Menggunakan Library Natural Language Toolkit (NLTK) Berbasis Pemrograman Python. *ILKOMNIKA: Journal of Computer Science and Applied Informatics*, 2(1), 8–17. <https://doi.org/10.28926/ilkomnika.v2i1.32>

Rumapea, H. (2021). *METHOMIKA: Jurnal Manajemen Informatika & Komputerisasi Akuntansi DETEKSI KEMIRIPAN ARTIKEL MELALUI KEYWORDS DENGAN METODE FUZZY STRING MATCHING DALAM NATURAL LANGUAGE PROCESSING*. 5(1). <https://doi.org/10.46880/jmika.Vol5No1.pp60-66>

Udil, P. A. (2021). Pelatihan Penulisan Artikel Ilmiah Penelitian Tindakan Kelas untuk Publikasi pada Jurnal Ilmiah. In *Jurnal Nasional Pengabdian Masyarakat* (Vol. 2, Issue 1).

Usmany, R. (2022). PENGEMBANGAN CHATBOT PENGADUAN DAN TROUBLESHOOTING TEKNOLOGI INFORMASI DENGAN PENDEKATAN NLP (STUDI KASUS POLITEKNIK NEGERI AMBON). *JURNAL SIMETRIK*, 12(2).

Wita, E. (2021). *Penerapan Natural Language Processing Untuk Mengidentifikasi Kalimat Ambigu pada Surat Kabar Menerapkan Metode Shift Reduce Parsing*. 2(2), 63–66.

Yunefri, Y., Ersan Fadrial, Y., & Sutejo. (2021). CHATBOT PADA SMART COOPERATIVE ORIENTED PROBLEM MENGGUNAKAN

NATURAL LANGUAGE PROCESSING DAN NAIVE BAYES CLASSIFIER CHATBOT ON SMART COOPERATIVE ORIENTED PROBLEMS USING NATURAL LANGUAGE PROCESSING AND NAIVE BAYES CLASSIFIER. *Journal of Information Technology and Computer Science (INTECOMS)*, 4(2).

Zein, A. (2021). *Kecerdasan Buatan Dalam Hal Otomatisasi Layanan*.



LAMPIRAN

Lampiran 1. *Source Code*

```
from flask import Flask, request, send_file, jsonify,
render_template, redirect, url_for, send_from_directory
from werkzeug.utils import secure_filename
import os
from docx import Document
from docx.shared import Pt, Inches
from docx.enum.text import WD_PARAGRAPH_ALIGNMENT

app = Flask(__name__)

UPLOAD_FOLDER = 'uploads'
PROCESSED_FOLDER = 'processed'
if not os.path.exists(UPLOAD_FOLDER):
    os.makedirs(UPLOAD_FOLDER)
if not os.path.exists(PROCESSED_FOLDER):
    os.makedirs(PROCESSED_FOLDER)

def baca_template(path_template):
    return Document(path_template)

def ekstrak_judul(dokumen):
    for para in dokumen.paragraphs:
        if para.text.strip():
            return para.text.strip()
    return "Judul Tidak Ditemukan"

def ekstrak_nama(dokumen):
```

```

nama = None
nomor_mahasiswa = None
paragraf_sebelum_nomor = None

for para in dokumen.paragraphs:
    teks = para.text.strip()

    # Cek apakah teks berupa angka (biasanya nomor
mahasiswa)
    if teks.isdigit() and len(teks) > 7:
        nomor_mahasiswa = teks
        # Nama kemungkinan ada di paragraf sebelumnya
        paragraf_sebelum_nomor =
para._element.getprevious()
        if paragraf_sebelum_nomor is not None:
            nama =
paragraf_sebelum_nomor.text.strip()
            break

if nama:
    return nama
else:
    return "Nama Tidak Ditemukan"

```

```

def ekstrak_institusi(dokumen):
    institusi = None

    # Loop untuk memeriksa paragraf pada halaman pertama
saja
    for para in dokumen.paragraphs:
        teks = para.text.strip()

```

```

        # Jika menemukan kata "Universitas", ambil teks
tersebut
        if teks.lower().startswith("universitas"):
            institusi = teks
            break

# Jika menemukan indikator halaman baru, berhenti
loop

        if "Page Break" in para._element.xml:
            break

        if institusi:
            return institusi
        else:
            return "Institusi Tidak Ditemukan"

def ekstrak_keywords(dokumen):
    keywords = None

    for para in dokumen.paragraphs:
        teks = para.text.strip()

        # Mencari pola "Keywords:" (dengan case-
insensitive)
        if teks.lower().startswith("keywords:") or
teks.lower().startswith("keyword:") or
teks.lower().startswith("keywords      :") or
teks.lower().startswith("keyword :"):
            keywords = teks[len("Keywords:"):].strip()
            break

```

```

if keywords:
    return keywords
else:
    return "Keywords Tidak Ditemukan"

def ekstrak_kata_kunci(dokumen):
    kata_kunci = None

    for para in dokumen.paragraphs:
        teks = para.text.strip()

        # Mencari pola "Kata kunci:" (dengan case-
insensitive)
        if teks.lower().startswith("kata kunci:") or
teks.lower().startswith("keyword:") or
teks.lower().startswith("kata kunci :") or
teks.lower().startswith("keyword :"):
            kata_kunci = teks[len("Kata
kunci:"):].strip()
            break

    if kata_kunci:
        return kata_kunci
    else:
        return "Kata Kunci Tidak Ditemukan"

def ekstrak_bagian(dokumen):
    bagian = {
        "Judul": ekstrak_judul(dokumen),

```

```
"Nama Penulis": ekstrak_nama(dokumen),
"Institusi": ekstrak_institusi(dokumen),
"Abstrak": [],
"Abstract": [],
"Pendahuluan": [],
"Metode Penelitian": [],
"Hasil dan Pembahasan": [],
"Kesimpulan": [],
"Referensi": [],
"Keywords": ekstrak_keywords(dokumen),
"Kata Kunci": ekstrak_kata_kunci(dokumen),
}

section_flags = {
    "dalam_abstrak": False,
    "dalam_abstract": False,
    "dalam_pendahuluan": False,
    "dalam_metode_penelitian": False,
    "dalam_hasil_dan_pembahasan": False,
    "dalam_kesimpulan": False,
    "dalam_referensi": False
}

current_section = None
current_list_pendahuluan = []
current_list_metode_penelitian = []
current_list_hasil_dan_pembahasan = []
current_list_kesimpulan = []
current_list_referensi = []
```

```

    #Fungsi ini digunakan untuk mengatur ulang semua
    flags menjadi False

    def reset_flags(flags):
        for key in flags:
            flags[key] = False

    for para in dokumen.paragraphs:
        text = para.text.strip()
        if not text:
            continue

        #Bagian ini digunakan untuk mengabaikan bagian
        yang tidak perlu diambil
        if any(keyword.lower() in text.lower() for
        keyword in ["keywords", "keyword", "kata kunci", "ucapan
        terima kasih", "lampiran"]):
            reset_flags(section_flags)
            current_section = None
            continue

        #Bagian ini digunakan untuk menentukan bagian
        yang sedang diambil berdasarkan kata kunci yang ada di
        dokumen

        if "ABSTRACT" in text:
            current_section = "Abstract"
            reset_flags(section_flags)
            section_flags["dalam_abstract"] = True
        elif "ABSTRAK" in text:
            current_section = "Abstrak"
            reset_flags(section_flags)
            section_flags["dalam_abstrak"] = True

```

```

        elif "PENDAHULUAN" in text and
(para.style.name.startswith("Heading") or
para.style.name.startswith("SUB")):
    current_section = "Pendahuluan"
    reset_flags(section_flags)
    section_flags["dalam_pendahuluan"] = True
    elif "METODE PENELITIAN" in text and
(para.style.name.startswith("Heading") or
para.style.name.startswith("SUB")):
    current_section = "Metode Penelitian"
    reset_flags(section_flags)
    section_flags["dalam_metode_penelitian"] =
True
    elif "HASIL DAN DISKUSI" in text or "HASIL DAN
PEMBAHASAN" in text and
(para.style.name.startswith("Heading") or
para.style.name.startswith("SUB")):
    current_section = "Hasil dan Pembahasan"
    reset_flags(section_flags)
    section_flags["dalam_hasil_dan_pembahasan"]
= True
    elif "KESIMPULAN" in text or "PENUTUP" in text
and (para.style.name.startswith("Heading") or
para.style.name.startswith("SUB")):
    current_section = "Kesimpulan"
    reset_flags(section_flags)
    section_flags["dalam_kesimpulan"] = True
    elif "REFERENSI" in text or "DAFTAR PUSTAKA" in
text and (para.style.name.startswith("Heading") or
para.style.name.startswith("SUB")):
    current_section = "Referensi"

```

```

        reset_flags(section_flags)
        section_flags["dalam_referensi"] = True
# Abaikan Ekstraksi Seluruh Bagian BAB
elif "BAB" in text:
    reset_flags(section_flags)
    current_section = None
    continue
    elif para.style.name.startswith("Heading") or
para.style.name.startswith("SUB"):
        continue
    elif "Tabel" in text or "Gambar" in text:
        continue
    else:
        #Jika bagian yang sedang diambil adalah
bagian yang memiliki list, maka list tersebut akan
diambil sebagai satu item dalam list
        if section_flags["dalam_pendahuluan"]:
            if para.style.name.startswith("List"):
                current_list_pendahuluan.append(text)
            else:
                if current_list_pendahuluan:
                    bagian["Pendahuluan"].append(cu
rrent_list_pendahuluan)
                    current_list_pendahuluan = []
                    bagian["Pendahuluan"].append(text)
                elif
section_flags["dalam_metode_penelitian"]:
                    if para.style.name.startswith("List"):
                        current_list_metode_penelitian.appe
nd(text)
                    else:

```

```

        if current_list_metode_penelitian:
            bagian["Metode
Penelitian"].append(current_list_metode_penelitian)
            current_list_metode_penelitian
= []

            bagian["Metode
Penelitian"].append(text)

        elif
section_flags["dalam_hasil_dan_pembahasan"]:
            if para.style.name.startswith("List"):
                current_list_hasil_dan_pembahasan.a
ppend(text)
            else:
                if current_list_hasil_dan_pembahasan:
                    bagian["Hasil dan
Pembahasan"].append(current_list_hasil_dan_pembahasan)
                    current_list_hasil_dan_pembahasan
= []

                    bagian["Hasil dan
Pembahasan"].append(text)
                elif section_flags["dalam_kesimpulan"]:
                    if para.style.name.startswith("List"):
                        current_list_kesimpulan.append(text)
                    else:
                        if current_list_kesimpulan:
                            bagian["Kesimpulan"].append(cur
rent_list_kesimpulan)

                            current_list_kesimpulan = []
                            bagian["Kesimpulan"].append(text)
                elif section_flags["dalam_referensi"]:
                    if para.style.name.startswith("List"):

```

```

        current_list_referensi.append(text)
    else:
        if current_list_referensi:
            bagian["Referensi"].append(current_list_referensi)

            current_list_referensi = []
            bagian["Referensi"].append(text)

        else:
            if current_section:
                bagian[current_section].append(text)

            #Bagian ini digunakan untuk menambahkan list yang
            belum terakomodasi ke dalam bagian yang sesuai sebelum
            dijadikan satu string
            if current_list_pendahuluan:
                bagian["Pendahuluan"].append(current_list_pendahuluan)
            if current_list_metode_penelitian:
                bagian["Metode Penelitian"].append(current_list_metode_penelitian)
            if current_list_hasil_dan_pembahasan:
                bagian["Hasil dan Pembahasan"].append(current_list_hasil_dan_pembahasan)
            if current_list_kesimpulan:
                bagian["Kesimpulan"].append(current_list_kesimpulan)
            if current_list_referensi:
                bagian["Referensi"].append(current_list_referensi)

```

```

#Memeriksa apakah ada bagian yang kosong
for key, value in bagian.items():
    if not value:
        return f"Bagian {key} tidak ditemukan."

return bagian

#Fungsi ini digunakan untuk mengganti bagian yang ada di
template dan menyesuaikan dengan format jurnal
def        sesuaikan_dengan_template(dokumen_template,
bagian):
    for para in dokumen_template.paragraphs:

        #Bagian ini digunakan untuk mengganti bagian
yang ada di template sesuai dengan bagian yang ada di
dokumen dengan menggunakan kata kunci yang ada di
dokumen, di sertai format penulisannya sesuai dengan
format jurnal
        if "JUD1" in para.text:
            para.clear()
            judul = bagian.get('Judul', 'Judul Tidak
Ditemukan').upper()
            if len(judul.split()) > 15:
                judul = ' '.join(judul.split()[:15])
            p_judul = para.insert_paragraph_before(judul)
            run_judul = p_judul.runs[0]
            run_judul.font.size = Pt(14)
            run_judul.font.name = 'Arial'
            run_judul.bold = True

            p_judul.alignment =
WD_PARAGRAPH_ALIGNMENT.CENTER

```

```

elif "NP" in para.text:
    para.clear()
    penulis = bagian.get('Nama Penulis', [])
    penulis_str = ''.join(penulis)

    run_penulis = para.add_run(penulis_str)
    run_penulis.font.size = Pt(10)
    run_penulis.font.name = 'Arial'
    run_penulis.bold = True
    para.alignment =
WD_PARAGRAPH_ALIGNMENT.CENTER

elif "INS" in para.text:
    para.clear()
    institusi = bagian.get('Institusi', [])
    institusi_str = ''.join(institusi)

    run_institusi = para.add_run(institusi_str)
    run_institusi.font.size = Pt(10)
    run_institusi.font.name = 'Arial'
    para.alignment =
WD_PARAGRAPH_ALIGNMENT.CENTER

elif "AA1" in para.text:
    para.clear()
    for paragraph in bagian.get('Abstract',
['Tidak ada abstrak ditemukan.']):
        p_abstract =
para.insert_paragraph_before(paragraph)
        run = p_abstract.runs[0]

```

```

run.font.size = Pt(10)
run.font.name = 'Arial'
run.italic = True

p_abstract.alignment =
WD_PARAGRAPH_ALIGNMENT.JUSTIFY
p_abstract.paragraph_format.space_before
= Pt(3)
p_abstract.paragraph_format.space_after
= Pt(3)
p_abstract.paragraph_format.line_spacing
= 1.15
p_abstract.paragraph_format.first_line_
indent = Inches(0.5)

elif "AA2" in para.text:
para.clear()
for paragraph in bagian.get('Abstrak',
['Tidak ada abstrak ditemukan.']):
p_abstrak =
para.insert_paragraph_before(paragraph)
p_abstrak.style.font.size = Pt(10)
p_abstrak.style.font.name = 'Arial'
p_abstrak.alignment =
WD_PARAGRAPH_ALIGNMENT.JUSTIFY
p_abstrak.paragraph_format.space_before
= Pt(3)
p_abstrak.paragraph_format.space_after
= Pt(3)
p_abstrak.paragraph_format.line_spacing
= 1.15

```

```

        p_abstrak.paragraph_format.first_line_in
ndent = Inches(0.5)

        elif "AA3" in para.text:
            para.clear()

                for item_pendahuluan in
bagian.get('Pendahuluan', ['Tidak ada pendahuluan
ditemukan.']):
                    if isinstance(item_pendahuluan, list):
                        for i, list_item in
enumerate(item_pendahuluan, 1):
                            p_pendahuluan =
para.insert_paragraph_before(f"{i}. {list_item}")
                            p_pendahuluan.style.font.size =
Pt(10)
                            p_pendahuluan.style.font.name =
'Arial'
                            p_pendahuluan.paragraph_format.
left_indent = Inches(0.5)
                            p_pendahuluan.paragraph_format.
first_line_indent = Inches(-0.15)
                            p_pendahuluan.paragraph_format.
space_before = Pt(3)
                            p_pendahuluan.paragraph_format.
space_after = Pt(3)
                            p_pendahuluan.paragraph_format.
line_spacing = 1.15
                            p_pendahuluan.alignment =
WD_PARAGRAPH_ALIGNMENT.LEFT
                    else:

```

```

p_pendahuluan =
para.insert_paragraph_before(item_pendahuluan)
p_pendahuluan.style.font.size =
Pt(10)
p_pendahuluan.style.font.name =
'Arial'
p_pendahuluan.alignment =
WD_PARAGRAPH_ALIGNMENT.JUSTIFY
p_pendahuluan.paragraph_format.spac
e_before = Pt(3)
p_pendahuluan.paragraph_format.spac
e_after = Pt(3)
p_pendahuluan.paragraph_format.line
_spacing = 1.15
p_pendahuluan.paragraph_format.firs
t_line_indent = Inches(0.5)
elif "AA4" in para.text:
para.clear()
for item_metode_penelitian in
bagian.get('Metode Penelitian', ['Tidak ada metode
penelitian ditemukan.']):
if isinstance(item_metode_penelitian,
list):
for i, list_item in
enumerate(item_metode_penelitian, 1):
p_metode_penelitian =
para.insert_paragraph_before(f"{i}. {list_item}")
p_metode_penelitian.style.font.
size = Pt(10)

```

```

        p_metode_penelitian.style.font.
name = 'Arial'

        p_metode_penelitian.paragraph_f
ormat.left_indent = Inches(0.5)

        p_metode_penelitian.paragraph_f
ormat.first_line_indent = Inches(-0.15)

        p_metode_penelitian.paragraph_f
ormat.space_before = Pt(3)

        p_metode_penelitian.paragraph_f
ormat.space_after = Pt(3)

        p_metode_penelitian.paragraph_f
ormat.line_spacing = 1.15

        p_metode_penelitian.alignment =
WD_PARAGRAPH_ALIGNMENT.LEFT
    else:
        p_metode_penelitian =
para.insert_paragraph_before(item_metode_penelitian)
        p_metode_penelitian.style.font.size
= Pt(10)
        p_metode_penelitian.style.font.name
= 'Arial'
        p_metode_penelitian.alignment =
WD_PARAGRAPH_ALIGNMENT.JUSTIFY
        p_metode_penelitian.paragraph_forma
t.space_before = Pt(3)
        p_metode_penelitian.paragraph_forma
t.space_after = Pt(3)
        p_metode_penelitian.paragraph_forma
t.line_spacing = 1.15
        p_metode_penelitian.paragraph_forma
t.first_line_indent = Inches(0.5)

```

```

elif "AA5" in para.text:
    para.clear()
    for item_hasil in bagian.get('Hasil dan
Pembahasan', ['Tidak ada hasil dan pembahasan
ditemukan.']):
        if isinstance(item_hasil, list):
            for i, list_item in
enumerate(item_hasil, 1):
                p_hasil =
para.insert_paragraph_before(f"{i}. {list_item}")
                p_hasil.style.font.size = Pt(10)
                p_hasil.style.font.name = 'Arial'
                p_hasil.paragraph_format.left_i
ndent = Inches(0.5)
                p_hasil.paragraph_format.first_
line_indent = Inches(-0.15)
                p_hasil.paragraph_format.space_
before = Pt(3)
                p_hasil.paragraph_format.space_
after = Pt(3)
                p_hasil.paragraph_format.line_s
pacing = 1.15
                p_hasil.alignment =
WD_PARAGRAPH_ALIGNMENT.LEFT
            else:
                p_hasil =
para.insert_paragraph_before(item_hasil)
                p_hasil.style.font.size = Pt(10)
                p_hasil.style.font.name = 'Arial'

```

```

p_hasil.alignment =
WD_PARAGRAPH_ALIGNMENT.JUSTIFY
p_hasil.paragraph_format.space_before
= Pt(3)
p_hasil.paragraph_format.space_after
= Pt(3)
p_hasil.paragraph_format.line_spacing
= 1.15
p_hasil.paragraph_format.first_line
_indent = Inches(0.5)
elif "AA6" in para.text:
para.clear()
for item_kesimpulan in
bagian.get('Kesimpulan', ['Tidak ada kesimpulan
ditemukan.']):
if isinstance(item_kesimpulan, list):
for i, list_item in
enumerate(item_kesimpulan, 1):
p_kesimpulan =
para.insert_paragraph_before(f"{i}. {list_item}")
p_kesimpulan.style.font.size =
Pt(10)
p_kesimpulan.style.font.name =
'Arial'
p_kesimpulan.paragraph_format.l
eft_indent = Inches(0.5)
p_kesimpulan.paragraph_format.f
irst_line_indent = Inches(-0.15)
p_kesimpulan.paragraph_format.s
pace_before = Pt(3)

```

```

        p_kesimpulan.paragraph_format.space
pace_after = Pt(3)
        p_kesimpulan.paragraph_format.line
ine_spacing = 1.15
        p_kesimpulan.alignment =
WD_PARAGRAPH_ALIGNMENT.LEFT
    else:
        p_kesimpulan =
para.insert_paragraph_before(item_kesimpulan)
        p_kesimpulan.style.font.size = Pt(10)
        p_kesimpulan.style.font.name =
'Arial'
        p_kesimpulan.alignment =
WD_PARAGRAPH_ALIGNMENT.JUSTIFY
        p_kesimpulan.paragraph_format.space
_before = Pt(3)
        p_kesimpulan.paragraph_format.space
_after = Pt(3)
        p_kesimpulan.paragraph_format.line
spacing = 1.15
        p_kesimpulan.paragraph_format.first
_line_indent = Inches(0.5)

elif "AA7" in para.text:
    para.clear()
    referensi_paragraf = bagian.get('Referensi',
['Tidak ada referensi ditemukan.'])

# Inisiasi counter untuk penomoran IEEE
nomor_referensi = 1

```

```

# Iterasi setiap paragraf dalam daftar
referensi

for item_referensi in referensi_paragraf:
    if isinstance(item_referensi, list):
        for sub_item in item_referensi:
            # Buat paragraf baru dengan
            penomoran IEEE [1], [2], dst.

            p_referensi =
para.insert_paragraph_before(f"[{nomor_referensi}]
{sub_item}")
            p_referensi.style.font.size =
Pt(10)
            p_referensi.style.font.name =
'Arial'
            p_referensi.paragraph_format.le
ft_indent = Inches(0.22)
            p_referensi.paragraph_format.fi
rst_line_indent = Inches(-0.22)
            p_referensi.paragraph_format.sp
ace_before = Pt(3)
            p_referensi.paragraph_format.sp
ace_after = Pt(3)
            p_referensi.paragraph_format.li
ne_spacing = 1.15

            p_referensi.alignment =
WD_PARAGRAPH_ALIGNMENT.LEFT

            # Increment nomor referensi
            nomor_referensi += 1

        else:

```

```

        # Sama seperti di atas, namun untuk
item tunggal

        p_referensi =
para.insert_paragraph_before(f"[{nomor_referensi}]
{item_referensi}")

        p_referensi.style.font.size = Pt(10)
p_referensi.style.font.name = 'Arial'
        p_referensi.alignment =
WD_PARAGRAPH_ALIGNMENT.JUSTIFY
        p_referensi.paragraph_format.left_i
ndent = Inches(0.22)
        p_referensi.paragraph_format.first_
line_indent = Inches(-0.22)
        p_referensi.paragraph_format.space_
before = Pt(3)
        p_referensi.paragraph_format.space_
after = Pt(3)
        p_referensi.paragraph_format.line_s
pacing = 1.15

        # Increment nomor referensi
        nomor_referensi += 1

elif "KK1" in para.text:
    para.clear()
        run = para.add_run(f"Keywords:
{bagian.get('Keywords', 'Tidak ada keywords
ditemukan.')}")
        run.font.size = Pt(10)
        run.font.name = 'Arial'
        run.italic = True

```

```

        para.alignment = WD_PARAGRAPH_ALIGNMENT.LEFT

        elif "KK2" in para.text:
            para.clear()
            run = para.add_run(f"Kata Kunci:
{bagian.get('Kata Kunci', 'Tidak ada kata kunci
ditemukan.')}")
            run.font.size = Pt(10)
            run.font.name = 'Arial'
            para.alignment = WD_PARAGRAPH_ALIGNMENT.LEFT

def simpan_dokumen_baru(dokumen_template, path_baru):
    dokumen_template.save(path_baru)

def konversi_skripsi_ke_jurnal(path_skripsi,
path_template, path_output):
    dokumen_skripsi = Document(path_skripsi)
    dokumen_template = baca_template(path_template)
    bagian = ekstrak_bagian(dokumen_skripsi)
    sesuaikan_dengan_template(dokumen_template, bagian)
    simpan_dokumen_baru(dokumen_template, path_output)

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/template-ainet')
def template_ainet():
    return render_template('template_ainet.html')

@app.route('/proses-generate')
def proses_generate():

```

```

        return render_template('proses_generate.html')

@app.route('/upload', methods=['POST'])
def upload_file():
    if 'file' not in request.files:
        return jsonify({"error": "No file part"}), 400

    file = request.files['file']
    if file.filename == '':
        return jsonify({"error": "No selected file"}),
400

    if file:
        filename = secure_filename(file.filename)
        skripsi_path = os.path.join(UPLOAD_FOLDER,
filename)
        file.save(skripsi_path)

        template_path = "template_jurnal/Template.docx"
        output_path = os.path.join(PROCESSED_FOLDER,
f"processed_{filename}")

        konversi_skripsi_ke_jurnal(skripsi_path,
template_path, output_path)

        return jsonify({"message": "File uploaded and
processed successfully", "download_url":
f"/download/{filename}"}), 200

```

#Fungsi ini digunakan untuk mendownload file jurnal yang telah dihasilkan

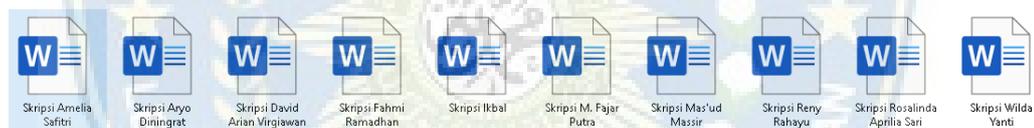
```

@app.route('/download/<filename>', methods=['GET'])
def download_file(filename):
    processed_path = os.path.join(PROCESSED_FOLDER,
f"processed_{filename}")
    if os.path.exists(processed_path):
        return send_file(processed_path,
as_attachment=True)
    else:
        return jsonify({"error": "File not found"}), 404

if __name__ == '__main__':
    app.run(debug=True)
app.run(debug=True)

```

Lampiran 2 Data Skripsi Uji Coba



Lampiran 3 Data Jurnal Manual



Lampiran 4 Jurnal Hasil Generate



Lampiran 5 Link Github File Data

<https://github.com/yulistiahnengsih/Generate-Skripsi>

Lampiran 6 Hasil Uji Plagiat



**MAJELIS PENDIDIKAN TINGGI PIMPINAN PUSAT MUHAMMADIYAH
UNIVERSITAS MUHAMMADIYAH MAKASSAR
UPT PERPUSTAKAAN DAN PENERBITAN**

Alamat kantor: Jl.Sultan Alauddin No.259 Makassar 90221 Tlp.(0411) 866972,881593, Fax.(0411) 865588

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

SURAT KETERANGAN BEBAS PLAGIAT

**UPT Perpustakaan dan Penerbitan Universitas Muhammadiyah Makassar,
Menerangkan bahwa mahasiswa yang tersebut namanya di bawah ini:**

Nama : Yulistiah Nengsih
Nim : 105841107020
Program Studi : Teknik Informatika
Dengan nilai:

No	Bab	Nilai	Ambang Batas
1	Bab 1	3 %	10 %
2	Bab 2	7 %	25 %
3	Bab 3	8 %	10 %
4	Bab 4	2 %	10 %
5	Bab 5	5 %	5%

Dinyatakan telah lulus cek plagiat yang diadakan oleh UPT- Perpustakaan dan Penerbitan Universitas Muhammadiyah Makassar Menggunakan Aplikasi Turnitin.

Demikian surat keterangan ini diberikan kepada yang bersangkutan untuk dipergunakan seperlunya.

Makassar, 30 Agustus 2024
Mengetahui
Kepala UPT- Perpustakaan dan Penerbitan,


Marsinah, S.Hum., M.I.P.
NPM. 964 591



Jl. Sultan Alauddin no 259 makassar 90222
Telepon (0411)866972,881 593,fax (0411)865 588
Website: www.library.unismuh.ac.id
E-mail : perpustakaan@unismuh.ac.id