Journal of Digital Business and Technology Innovation (DBESTI)

https://journal.nurulfikri.ac.id/index.php/DBESTI
P-ISSN: 3047-1028 E-ISSN: 3032-775X

PENGEMBANGAN REST API UNTUK EKSTRAKSI INFORMASI DARI *RESUME* MENGGUNAKAN *JAVA SPRING BOOT*

Evry Nazyli Ciptanto[1], Nasrul[2]

[1,2]Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri
Depok, Jawa Barat, Indonesia 16451
evry20045ti@student.nurulfikri.ac.id, nasrul@nurulfikri.ac.id

Abstract

Career platforms like Klob are among the modern recruitment methods. In the Klob career platform context, users face challenges when filling out their profile data. The main challenge is manual filling, which slows the registration and job application process. The problem required development to integrate information extraction from user resumes as an efficient solution. The Java Spring Boot programming language built a resume information extraction backend development system. This integration also utilized Natural Language Processing (NLP) recognition algorithms using OpenNLP to improve understanding of the context of sentences on resumes. One of the OpenNLP methods used was Named Entity Recognition (NER) to identify and extract named entities. The Rule-Based Extraction approach was also specified to extract information from text based on rules. This research was based on the results of testing conducted through the User Acceptance Test (UAT), and it was found that the percentage score given had an excellent interpretation level of 91.1%. This research provides a comprehensive overview of the effectiveness of the proposed solution in overcoming manual filling obstacles, improving user experience, and speeding up registration.

Keywords: *Backend, Information Extraction, Java, Natural Language Processing, Spring Boot*

Abstrak

Platform karier seperti Klob merupakan salah satu rekrutmen modern. Dalam konteks platform karier Klob terdapat kendala yang dihadapi pengguna dalam pengisian data profil. Kendala utama adalah pengisian manual yang memperlambat proses pendaftaran dan melamar pekerjaan. Dari permasalahan tersebut dibutuhkan pengembangan untuk mengintegrasikan ekstraksi informasi dari resume pengguna sebagai solusi efisien. Dalam membangun sistem pengembangan *backend* ekstraksi informasi resume ini menggunakan bahasa pemrograman *Java Spring Boot.* Integrasi juga memanfaatkan algoritma pengenalan *Natural Language Processing* (NLP) menggunakan *OpenNLP* untuk meningkatkan pemahaman konteks kalimat pada resume. Salah satu metode *OpenNLP* yang dimanfaatkan adalah *Named Entity Recognition* (NER) untuk mengidentifikasi dan mengekstraksi entitas bernama. Pendekatan *Rule Based Extraction* juga ditentukan untuk melakukan ekstraksi informasi dari teks berdasarkan aturan-aturan. Penelitian ini dari hasil pengujian yang dilakukan dengan *User Acceptance Test* (UAT) dan didapatkan skor persentase yang diberikan memiliki tingkat interpretasi yang sangat baik sebesar 91,1%. Harapannya, dengan adanya hasil penelitian ini mampu memberikan gambaran keseluruhan mengenai efektivitas solusi yang diusulkan dalam mengatasi kendala pengisian manual, meningkatkan pengalaman pengguna, dan mempercepat proses pendaftaran.

Kata kunci: *Backend*, Ekstraksi Informasi, *Java*, *Natural Language Processing*, *Spring Boot*

1. PENDAHULUAN

Dalam era digital yang terus berkembang, platform karier memainkan peran yang semakin penting dalam dunia rekrutmen modern. Teknologi telah memfasilitasi pertemuan antara pencari kerja yang berbakat dengan perusahaan yang mencari tenaga kerja berkualitas. Salah satu platform karier terkemuka adalah Klob, yang telah menjadi wadah untuk mencocokkan kesempatan kerja dengan kekuatan dan keunikan masing-masing talenta[1]. Namun, seperti banyak platform sejenis lainnya, Klob masih menghadapi tantangan dalam meningkatkan efisiensi dan pengalaman pengguna.

Salah satu masalah yang dihadapi oleh pengguna baru adalah proses pengisian data profil secara manual saat pertama kali menggunakan platform Klob. Hal ini dapat

Diterima 26 Maret 2024, Direvisi 28 Oktober 2024, Diterima untuk publikasi 29 Oktober 2024

menjadi hambatan bagi mereka yang ingin melamar pekerjaan dengan cepat. Oleh karena itu, diperlukan langkah-langkah inovatif yang dapat mengatasi kendala ini dan meningkatkan efisiensi penggunaan platform.

Solusi yang diusulkan adalah menyediakan opsi unggah resume untuk mengisi data profil secara otomatis pada platform. Solusi ini tidak hanya akan menghemat waktu, melainkan juga akan meningkatkan akurasi dan konsistensi data yang dimasukkan. Sehingga data-data profil mereka akan secara otomatis terisi pada platform. Namun, tantangan utama dalam implementasi solusi ini adalah struktur yang berbeda dari setiap resume, ketidakstrukturan konten, adanya data yang tidak valid, serta tantangan teknis seperti pengambilan foto profil dari sebuah resume.

Untuk mengatasi masalah ini, penelitian ini diarahkan untuk merancang pengembangan solusi yang efisien dengan mengintegrasikan ekstraksi informasi dari resume pengguna berbasis bahasa pemrograman *Java*. Hasil dari perancangan ini akan menjadi panduan dalam mengembangkan *backend* untuk ekstraksi resume menggunakan bahasa pemrograman *Java* yang lebih sesuai dengan platform Klob. Dengan demikian, penelitian ini bertujuan untuk memberikan pengguna Klob dengan kemudahan dalam mengisi data profil mereka, meningkatkan pengalaman pengguna, dan mempermudah pencarian dan lamaran pekerjaan.

Namun, penelitian ini memiliki batasan yang jelas. Fokus akan diberikan pada pengembangan *backend* untuk ekstraksi informasi dari resume pengguna. Batasan-batasan ini mencakup format resume, bahasa yang digunakan, dan ruang lingkup ekstraksi informasi yang dibutuhkan. Dengan demikian, batasan-batasan tersebut akan membantu mengarahkan jalannya penelitian dan menyediakan kerangka kerja yang jelas bagi pengembangan solusi.

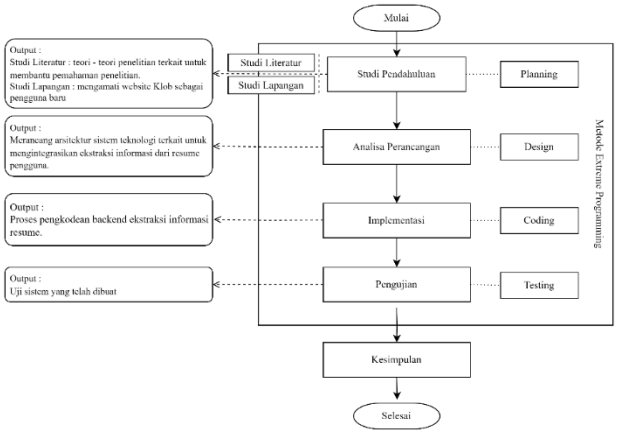
Penelitian ini memiliki batasan yang jelas, di mana fokus akan diberikan pada pengembangan *backend* untuk ekstraksi informasi dari resume pengguna. Batasan-batasan ini meliputi penekanan pada sistem yang dikembangkan untuk pengenalan dan ekstraksi informasi dari resume dalam format PDF, penggunaan model *OpenNLP* dalam bahasa Inggris khususnya untuk meresume teks, pemusatan pada data minimal yang diperlukan untuk melengkapi profil pengguna, serta penekanan pada ekstraksi data dari teks utama dalam resume sambil mengabaikan informasi yang tersembunyi dalam gambar atau data non-teks. Dengan demikian, batasan-batasan tersebut akan membantu mengarahkan jalannya penelitian dan menyediakan kerangka kerja yang jelas bagi pengembangan solusi.

2. METODE PENELITIAN

Dalam bagian ini, terdapat langkah-langkah yang akan dijalani dalam penelitian, termasuk di dalamnya adalah proses pengembangan sistem menggunakan metode *Extreme Programming* (XP). Metode penelitian akan menguraikan terkait tahapan penelitian, pengumpulan data dan metode pengujian.

2.1 Tahapan Penelitian

Dalam penelitian ini, penulis menjalani berbagai tahapan sesuai dengan tujuan penelitian. Proses tahapan penelitian ini melibatkan alur yang diuraikan secara rinci untuk setiap proses yang telah disusun, sehingga penelitian dapat dilaksanakan dengan jelas dan terstruktur.



Gambar 1. Tahapan Penelitian

Pada Gambar 1 tahapan penelitian ini menerapkan metodologi *Extreme Programming* (XP) untuk pengembangan *backend* ekstraksi informasi resume. *Extreme Programming* (XP) adalah suatu proses rekayasa perangkat lunak yang umumnya mengadopsi pendekatan berorientasi objek. Metodologi ini didasarkan pada prinsip-prinsip yang serupa, dengan fokus pada pengembangan sistem dalam waktu singkat yang dapat mengatasi perubahan yang cepat dari pengembang dengan berbagai perubahan yang mungkin terjadi[2]. Metodologi ini cocok digunakan ketika tim dihadapkan pada kebutuhan yang tidak jelas atau perubahan kebutuhan yang cepat.

A. Studi Pendahuluan

Dalam studi pendahuluan ini, dilakukan studi literatur untuk memahami teknologi dan pendekatan terkini dalam ekstraksi informasi dari resume. Fokusnya adalah mengidentifikasi solusi teknologi baru terkait ekstraksi informasi. Selain itu, dilakukan studi lapangan dengan mengamati langsung *website* Klob untuk mendapatkan wawasan praktis dalam konteks penelitian.

B. Analisa Perancangan

Dalam merancang, Tahapan perancangan ini melibatkan merancang arsitektur sistem untuk pengembangan ekstraksi informasi dari resume pengguna. Untuk pengembangan *backend* menggunakan *Java*, *Java* sering diidentifikasi sebagai bahasa pemrograman yang mampu beroperasi di berbagai sistem operasi, selama JVM tersedia di sistem operasi tersebut[3]. Pilihan dan justifikasi penggunaan kerangka kerja *Spring Boot* dalam pengembangan *backend*.

Dengan *Spring Boot* menjadi sangat sederhana karena didukung oleh perpustakaan yang komprehensif, dan *Spring Boot* tergolong ringan dalam penggunaannya[4]. Serta perancangan hasil dalam bentuk JSON setelah pengguna mengunggah resume.

C. Implementasi

Dalam tahap ini, peneliti akan memulai dengan menulis kode dalam bahasa pemrograman *Java* untuk ekstraksi informasi dari resume. Indikator capaian yang menjadi fokus pada tahap ini mencakup konfigurasi proyek *Spring Boot* sesuai kebutuhan, implementasi layanan ekstraksi informasi dari resume, pengembangan algoritma yang mampu mengenali struktur dan konten resume, serta pembuatan API *endpoint* yang dapat menerima unggahan resume dari pengguna.

D. Pengujian

Pengujian dilakukan terhadap sistem yang telah dikembangkan menggunakan resume untuk memastikan akurasi ekstraksi informasi. Pengujian melibatkan verifikasi akurasi ekstraksi informasi dari resume tersebut. Aplikasi memulai serangkaian aktivitas untuk memverifikasi bahwa tujuan penelitian telah tercapai melalui pengujian fungsional.

E. Kesimpulan

Langkah terakhir melibatkan penyusunan kesimpulan dari hasil penelitian dan pengembangan. Peneliti menjelaskan solusi yang telah diimplementasikan untuk mengatasi masalah pengisian data profil. Kesimpulan ini mencakup evaluasi keberhasilan dalam menangani masalah, pencapaian tujuan yang telah ditetapkan.

2.2 Metode Pengumpulan Data

Pada penelitian ini, tahapan pengumpulan data dan informasi melalui sejumlah metode, seperti:

A. *Online* Kuesioner

Penyebaran *online* kuesioner dilakukan dengan maksud untuk mengevaluasi sejauh mana kinerja aplikasi yang dikembangkan dapat berjalan secara optimal dan memenuhi kebutuhan pengguna. Hasil yang diperoleh akan diolah menggunakan skala *likert* guna menilai sejauh mana pengembangan aplikasi telah berhasil atau masih memerlukan perbaikan.

B. Observasi

Observasi dilakukan dengan menganalisis langsung platform Klob pada *website https://www.klob.id/* yang melibatkan pengamatan langsung terhadap objek penelitian terkait permasalahan yang akan dibahas. Ini melibatkan identifikasi proses pada platform Klob sebagai acuan untuk menentukan data yang harus diekstrak dari resume, sehingga dapat digunakan dalam profil biodata di Klob.

C. Studi Kepustakaan

Dalam pengembangan layanan ekstraksi informasi resume di platform Klob, studi kepustakaan memegang peran penting. Studi kepustakaan dilakukan dengan cermat untuk mengidentifikasi dan mengakses sumber-sumber pustaka yang mendukung penelitian ini. Sumber-sumber literatur yang relevan termasuk buku, jurnal, artikel, dan makalah dalam berbagai jenis.

2.3 Metode Pengujian

Pada pengujian ini merupakan suatu mekanisme yang digunakan untuk mengidentifikasi data uji yang mampu menguji perangkat lunak yang dikembangkan. Pengujian dilakukan menggunakan *User Acceptance Testing* (UAT). Menggunakan *User Acceptance Test* (UAT) merupakan sebuah proses pengujian yang dijalankan oleh pengguna dengan melakukan verifikasi terhadap sistem, apakah sistem telah diterima dan memenuhi kebutuhan yang diminta[5].

UAT akan melibatkan pengujian kemampuan aplikasi dalam mengidentifikasi dan mengekstrak data kunci, seperti informasi dasar, riwayat pendidikan, dan keterampilan. Sebelum di tahap UAT, peneliti menggunakan beberapa metode pengujian untuk mendapatkan hasil sesuai dengan penelitian. Adapun metode pengujian yang melibatkan teknik *Black Box testing* dan kuesioner.

A. *Black Box Testing*

Dalam penelitian ini, pendekatan yang diterapkan untuk menguji REST API yang telah dirancang adalah melalui metode *Black Box* dan menggunakan aplikasi Postman. Dalam pengujian ini, penguji hanya mengevaluasi hasil yang dihasilkan oleh perangkat lunak berdasarkan *input* yang diberikan, tanpa perlu mengetahui detail tentang bagaimana atau apa yang terjadi di dalamnya[6].

Uji coba dapat dengan memanfaatkan alat bantu Postman, yang memungkinkan pengujian fungsionalitas REST API melalui penggunaan permintaan *HTTP*. Pengujian *black box* untuk pengembangan REST API bertujuan untuk memastikan fungsionalitas yang benar, keandalan, keamanan, kemudahan penggunaan, serta kompatibilitas dan interoperabilitas dengan hasil yang diharapkan agar API dapat beroperasi sesuai spesifikasi, memberikan respons yang konsisten, menjaga keamanan, menawarkan antarmuka yang intuitif, dan dapat berintegrasi dengan berbagai sistem eksternal.

B. Skala *Likert*

Dalam evaluasi hasil ekstraksi informasi resume melalui kuesioner skala *likert*, responden akan diminta untuk menilai keakuratan, kelengkapan, dan keterbacaan informasi yang diekstrak dari resume. Dalam skala *likert*, terdapat dua jenis pernyataan, yakni pernyataan positif yang digunakan dalam mengevaluasi aspek positif, dan

pernyataan negatif yang digunakan dalam mengevaluasi aspek negatif[7].

Tabel 1. Kriteria Interpretasi Skala *Likert*

Interval Persentase	Nilai	Kualifikasi
0% - 20%	1	Sangat Tidak Setuju
21% - 40%	2	Tidak Setuju
41% - 60%	3	Kurang Setuju
61% - 80%	4	Setuju
81% - 100%	5	Sangat Setuju

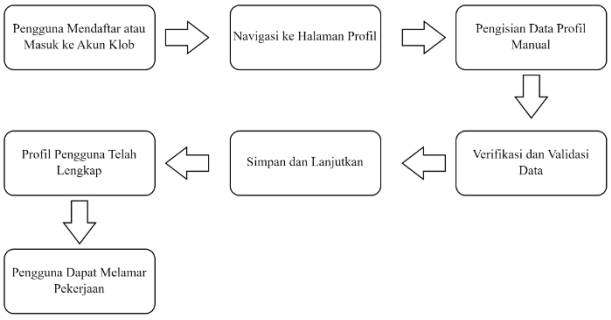
Pada Tabel 1, dijelaskan interval skala *likert* yang akan digunakan, ini berkisar dari "Sangat Tidak Setuju" hingga "Sangat Setuju". Interval skala *likert*, dapat digunakan untuk mengukur tingkat kepuasan atau penilaian responden terhadap berbagai aspek kinerja sistem ekstraksi.

3. HASIL DAN PEMBAHASAN

Pada bagian ini akan dibahas mengenai analisis dan proses perancangan implementasi serta evaluasi kinerja dalam pengembangan *backend* ekstraksi informasi resume

3.1 Analisis Sistem Berjalan

Pada kondisi sistem yang sedang berjalan saat ini proses pengisian profil pada platform Klob dilakukan secara manual oleh pengguna. Pada tahap ini, pengguna diminta untuk meng-*input*-kan berbagai informasi yang meliputi data dasar, ringkasan diri, riwayat pendidikan, dan keahlian secara langsung ke dalam formulir yang disediakan oleh platform.



Gambar 2. Alur Pengisian Profil Klob

Pada Gambar 2 alur pengisian profil klob dimulai dengan pengguna yang mendaftar atau masuk ke akun mereka. Setelah itu, pengguna diarahkan ke halaman profil untuk mengisi formulir secara manual, termasuk informasi dasar, ringkasan diri, riwayat pendidikan, dan keahlian. Pengguna kemudian memverifikasi dan memvalidasi data yang di-*input*-kan sebelum menyimpannya ke dalam basis data Klob. Dengan profil yang lengkap, pengguna dapat melanjutkan untuk menelusuri peluang kerja dan melamar posisi yang diinginkan.

Meskipun proses manual ini memberikan pengguna kontrol penuh terhadap informasi yang mereka berikan, tanpa opsi unggah resume, terdapat keterbatasan dalam akses informasi yang lebih rinci. Pengguna perlu meluangkan waktu dan usaha lebih banyak dalam pengisian manual, meningkatkan risiko kesalahan manusia dan dapat memengaruhi kualitas serta keakuratan profil. Keterbatasan ini juga dapat mempengaruhi efisiensi dan kenyamanan pengguna, terutama bagi mereka yang ingin melamar pekerjaan dengan cepat. Oleh karena itu, keberadaan opsi unggah resume dianggap dapat meningkatkan pengalaman pengguna dengan memberikan akses lebih baik terhadap informasi dan meminimalkan beban pengguna dalam mengisi profil secara manual.

3.2 Analisis Permasalahan dan Usulan Perbaikan

Proses manual pengisian profil sering kali menghadapi beberapa permasalahan. Pengguna, terutama yang memiliki kebutuhan untuk melamar pekerjaan dengan cepat, mungkin mengalami keterlambatan dalam mengakses atau merespons peluang pekerjaan. Selain itu, ketidakakuratan data dan kelelahan pengguna saat mengisi formulir juga dapat merugikan pengalaman pengguna secara keseluruhan.

Dalam rangka mengatasi kendala pengisian profil manual, sebuah solusi diusulkan untuk mengimplementasikan layanan ekstraksi informasi dari resume pengguna. Opsi ini memungkinkan pengguna untuk mengunggah resume mereka, dan sistem akan secara otomatis mengekstrak informasi yang diperlukan untuk melengkapi profil pengguna di platform Klob

A. Analisis Permasalahan

Proses manual pengisian profil pada platform Klob memberikan tantangan signifikan, terutama dalam hal efisiensi dan kecepatan akses informasi. Beberapa permasalahan yang diidentifikasi meliputi:

a. Keterlambatan dalam Melamar Pekerjaan
   Pengguna, terutama yang memiliki urgensi dalam melamar pekerjaan, mengalami keterlambatan akibat proses manual yang memakan waktu
b. Keakuratan Data
   Proses manual rentan terhadap kesalahan manusia, sehingga data yang dimasukkan oleh pengguna tidak selalu akurat atau konsisten
c. Kesulitan Pengguna Baru
   Pengguna baru mungkin merasa kesulitan atau terintimidasi oleh proses manual ini, yang dapat mengurangi daya tarik platform

B. Usulan Perbaikan

Sebagai solusi untuk mengatasi permasalahan tersebut, diusulkan implementasi layanan ekstraksi informasi dari resume. Memberikan opsi kepada pengguna untuk mengunggah resume mereka sebagai alternatif pengisian

profil manual. Mengimplementasikan algoritma ekstraksi informasi dari resume untuk secara otomatis mendapatkan data profil pengguna berdasarkan apa yang terdapat dalam resume. Serta Menyertakan panduan pengguna yang jelas untuk memandu pengguna baru dalam menggunakan opsi ekstraksi informasi dari resume.

3.2 Analisis Kebutuhan Sistem

Langkah awal dalam mengembangkan *backend* untuk mengekstraksi informasi dari resume menggunakan *Java Spring Boot* adalah melakukan analisis kebutuhan sistem. Pada tahap ini, dilakukan identifikasi dan pengumpulan spesifikasi fungsional dan non-fungsional dari sistem yang akan dikembangkan.

A. Kebutuhan Fungsional

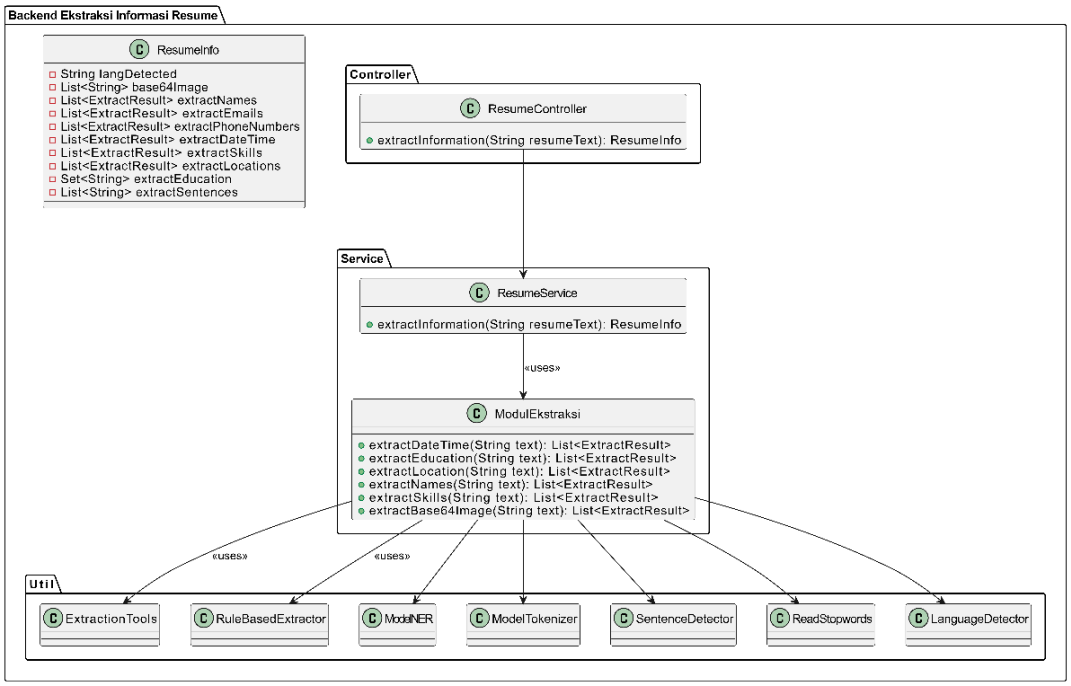
a. Sistem harus mampu melakukan ekstraksi informasi yang komprehensif dari berbagai jenis resume, termasuk informasi dasar, kalimat-kalimat penting, pendidikan, dan keterampilan. Proses ini melibatkan pemrosesan teks untuk mengidentifikasi entitas kunci dan hubungannya
b. Integrasi dengan algoritma pengenalan bahasa natural (NLP) menggunakan *OpenNLP* untuk meningkatkan pemahaman konteks kalimat pada resume. Hal ini akan membantu meningkatkan akurasi dalam ekstraksi informasi dan adaptabilitas terhadap perubahan dalam bahasa atau format resume.
c. Menyediakan API yang bersih dan terdokumentasi dengan baik untuk memfasilitasi integrasi mudah dengan aplikasi pihak ketiga. API harus mendukung operasi seperti ekstraksi data, manajemen pengguna, dan pengelolaan hasil ekstraksi

B. Kebutuhan Non-Fungsional

a. Sistem harus responsif dengan waktu eksekusi yang cepat, terutama dalam menangani permintaan ekstraksi dari resume yang besar. Pengoptimalan kode dan penyesuaian kapasitas server perlu diperhatikan
b. Keamanan data harus diutamakan dengan menerapkan enkripsi pada data yang disimpan dan melibatkan protokol otentikasi yang kuat. Proteksi terhadap serangan sisi server dan sisi pengguna perlu diimplementasikan secara cermat.
c. Arsitektur sistem harus dirancang untuk mudah ditingkatkan kapasitasnya agar dapat menangani peningkatan volume data dan jumlah pengguna. Evaluasi perlu dilakukan terhadap pemilihan teknologi yang mendukung otomatisasi peningkatan kapasitas dan manajemen beban

3.2 *Class Diagram* Sistem

Class diagram adalah representasi grafis dari susunan dan hubungan antar kelas dalam sistem. Untuk ekstraksi informasi resume, *class diagram* digunakan untuk menggambarkan bagaimana berbagai komponen atau kelas dalam sistem saling berinteraksi



Gambar 3. *Class Diagram* Sistem

Dalam gambar 3 *class diagram* disini terdapat kelas *ResumeController* yang bertanggung jawab untuk menerima permintaan dan berkomunikasi dengan kelas *ResumeService* yang berisi logika bisnis untuk mengekstrak informasi dari teks resume dibantu oleh kelas

ModulEkstraksi menggunakan layanan ekstraksi dan utilitas untuk mengekstrak informasi dari resume.

Package Util terdapat beberapa tools fungsi untuk membantu mengekstrak informasi, dalam Kelas *ModelNER* dan *ModelTokenizer* memanfaatkan *OpenNLP* untuk melakukan *tokenization* dan pengenalan entitas. Kemudian, *RuleBasedExtractor* adalah kelas yang menggunakan aturan tertentu untuk mengekstrak informasi tambahan seperti nomor telepon dan alamat email.

3.3 Perancangan Sistem

Perancangan sistem antarmuka pengguna diimplementasikan sebagai layanan RESTful API yang menyediakan *endpoint* untuk unggah resume, pengambilan hasil ekstraksi, dan operasi lainnya. Dokumentasi API dibuat untuk memfasilitasi integrasi dengan aplikasi klien eksternal

Status	*Response*
200 OK	{"code": "OK", "message": "Data has been successfully get.", "result": []}
400 - Request Body Error	{"code":"REQUEST_BODY_ERROR", "message": "The mandatory request body is missing"}
400 - API Validation Error	{"code":"API_VALIDATION_ERROR", "message": "There is an issue with the information you provided."}
400 - Invalid *Input*	{"code":"INVALID_*INPUT*", "message": " The request cannot be processed due to invalid *input*." }
404 - Not Found	{"code": "NOT_FOUND", "message": " The requested resource could not be found on the server."}
500 - Internal Server Error	{"code": "SERVER_ERROR", "message": " We're encountering a problem on our server." }

Perancangan *response* API di atas merinci perancangan antarmuka pengguna (API *response*) dengan kode status dan *respons body* API dalam format *content-type application/json* untuk berbagai kebutuhan. Pada setiap *endpoint*, informasi khusus terstruktur dalam format JSON, mencakup status keberhasilan, pesan informatif, dan data terkait ekstraksi atau validasi. Penggunaan *content-type application/json* mempermudah integrasi dengan aplikasi dan sistem lainnya, meningkatkan efisiensi komunikasi antara klien dan server.

3.4 Implementasi Model *OpenNLP*

Dalam proses implementasi sistem ekstraksi informasi resume, di awal pengembangan dibutuhkan data model. Data model diambil dari pustaka *Apache OpenNLP* yang memberikan berbagai layanan untuk memproses teks dalam bahasa alami, termasuk berbagai tugas pemrosesan *tokenization*, *segments sentences*, *tags Part-Of-Speech* (POS), *recognizes and extracts named entity*, *provides coreference resolution*, dan sebagainya[8].

OpenNLP menyediakan algoritma dan alat untuk melaksanakan *Named Entity Recognition* (NER). Beberapa pendekatan dari NER adalah *sequence tagging*, *Part of Speech* (POS) *tagging*, dan *chunking*[9]. *List* data model *OpenNLP* yang digunakan dalam tabel berikut:

No	Nama Model	Komponen	Keterangan
1	en-token.bin	*Tokenizer*	Memisahkan teks menjadi unit-token seperti kata atau frase.
2	en-sent.bin	*Sentence Detector*	Pemisahan teks menjadi kalimat-kalimat terpisah.
3	en-pos-maxent.bin	*POS Tagger*	Menandai jenis kata.
4	en-ner-date.bin	*Name Finder*	Pengenalan entitas tanggal.
5	en-ner-time.bin	*Name Finder*	Pengenalan entitas waktu.
6	en-ner-location.bin	*Name Finder*	Pengenalan entitas lokasi
7	en-ner-organization.bin	*Name Finder*	Pengenalan entitas organisasi
8	en-ner-person.bin	*Name Finder*	Pengenalan entitas nama orang
9	langdetect-183.bin	*Language Detector*	Mendeteksi bahasa

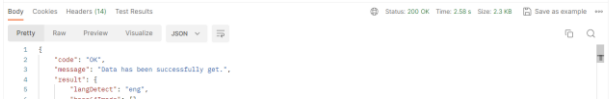
Penggunaan data model tersebut digunakan untuk membantu dalam pengolahan bahasa alami dengan mendukung kegiatan seperti memecah teks menjadi token, menandai jenis kata, mengenali entitas bernama, dan melakukan analisis sintaksis.

3.5 Implementasi Antarmuka

Selama proses implementasi antarmuka sistem, peneliti akan menampilkan sistem dalam bentuk layanan *RESTful API* untuk ekstraksi informasi dari resume, dengan fokus pada penjelasan *respons* yang diberikan saat pengguna mengunggah resume. API adalah sistem yang memfasilitasi

komunikasi antara satu aplikasi dengan yang lain, dengan model API yang didasarkan pada *RESTful Web Service* [10].

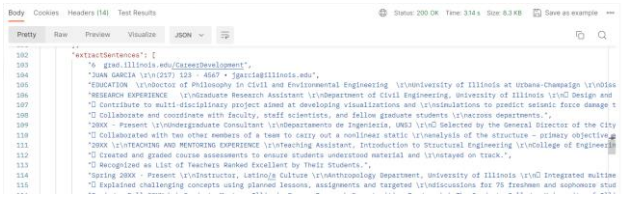
A. Tampilan *Response* Deteksi Bahasa



Gambar 4. Tampilan *Response* Deteksi Bahasa

Pada *response* JSON di gambar 4, terdapat *property* "*langDetect*" ini mengembalikan kode "eng" yang berarti resume yang diupload dalam bentuk bahasa Inggris. Jika ternyata *response* bukan "eng" maka sistem akan mengembalikan pesan kesalahan yang menyatakan bahwa saat ini bahasa yang didukung adalah bahasa Inggris. Model *OpenNLP* yang dipakai untuk deteksi bahasa adalah "*langdetect-183.bin*".

B. Tampilan Hasil Ekstraksi Kalimat



Gambar 5. Tampilan Hasil Ekstraksi Kalimat

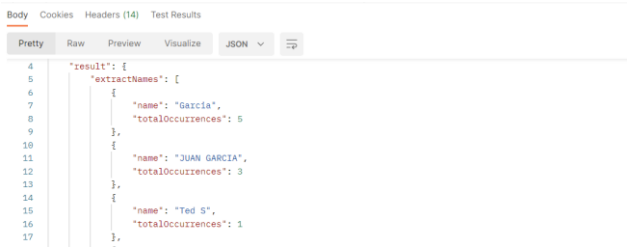
Pada gambar 5, ekstraksi deteksi kalimat ini berbentuk kumpulan kalimat. Dalam ekstraksi deteksi kalimat menggunakan model *en-sent.bin* dari pustakan *OpenNLP*. Data deteksi kalimat disimpan dalam property objek *extractSentences*.

C. Tampilan Hasil Ekstraksi Data Gambar

Gambar 6. Tampilan Hasil Ekstraksi Data Gambar

Hasil ekstraksi data gambar 6 berbentuk *list* dalam *Base64*. Gambar *Base64* merupakan representasi teks dari data biner gambar yang dienkripsi dengan menggunakan skema *encoding Base64*. Untuk mendapatkan gambar asli, pada proses integrasi diperlukan proses *decoding* kembali.

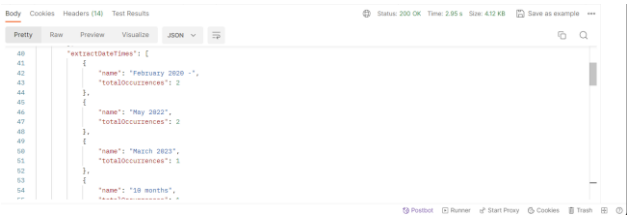
D. Tampilan Hasil Ekstraksi Data Nama



Gambar 7. Tampilan Hasil Ekstraksi Data Nama

Pada gambar 7, yaitu tampilan hasil ekstraksi data nama, *response* API memberikan sekumpulan data nama(*extractName*) dalam bentuk *array of objects.* Setiap objek memiliki dua properti yakni "*name*" yang menyimpan kategori informasi nama dan "*totalOccurrences*" yang menyimpan jumlah kemunculan kategori tersebut. Dengan demikian, *array* ini memberikan gambaran terstruktur tentang sejumlah kategori informasi dan frekuensi kemunculannya. Model *OpenNLP* yang digunakan adalah "en-ner-person.bin".

E. Tampilan Hasil Ekstraksi Tanggal dan Waktu



Gambar 8. Tampilan Hasil Ekstraksi Tanggal dan Waktu

Sama seperti hasil ekstraksi sebelumnya, pada gambar 8 *response extractDateTimes* menampilkan data yang berkaitan dengan tanggal dan waktu. Untuk ekstraksi data ini menggunakan model bernama *en-ner-date.bin* dan *en-ner-time.bin*

F. Tampilan Hasil Ekstraksi Email

Gambar 9. Tampilan Hasil Ekstraksi Email

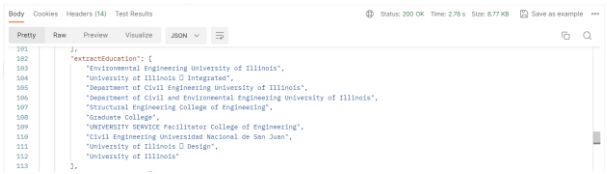
Pada gambar 9, dalam *response extractEmails* terdapat hasil dalam bentuk *array of objects.* Untuk object "*name*" yang menyimpan kategori informasi email. Dalam ekstraksi email menggunakan *rule base extraction* metode *regular expressions*. *Regular Expressions* adalah pola aturan yang digunakan untuk mencocokkan dan mengekstraksi teks yang cocok dengan pola yang telah ditentukan[11].

G. Tampilan Hasil Ekstraksi Nomor Telepon

Gambar 10. Tampilan Hasil Ekstraksi Nomor Telepon

Dalam hasil *response* dari fungsi *extractPhoneNumbers* pada gambar 10, terdapat data yang disajikan dalam bentuk *arrays of objects*. Objek tersebut mencakup atribut "*name*" yang menyimpan kategori informasi nomor telepon, serta atribut "*totalOccurrences*" yang menunjukkan total frekuensinya. Ekstraksi nomor telepon juga menggunakan *rule base extraction method* untuk mendapatkan data yang sesuai dengan format nomor telepon.

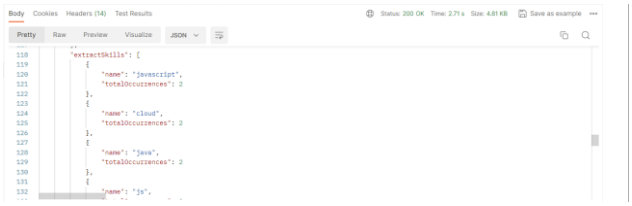
H. Tampilan Hasil Ekstraksi Pendidikan



Gambar 11. Tampilan Hasil Ekstraksi Pendidikan

Gambar 11 menampilkan hasil *response* data ekstraksi pendidikan menggunakan data bernama "en-ner-organization.bin". Model ini untuk untuk mengidentifikasi *Named Entity Recognition* (NER) terhadap kategori entitas organisasi dalam teks berbahasa Inggris.

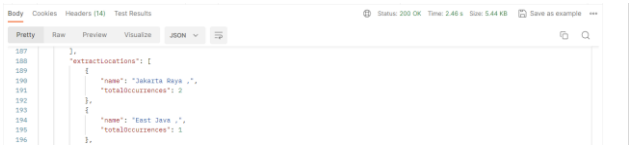
I. Tampilan Hasil Ekstraksi Keterampilan



Gambar 12. Tampilan Hasil Ekstraksi Keterampilan

Ekstraksi keterampilan diperlukan *dataset* yang terdapat pada basis referensi untuk mengidentifikasi dan menghitung keterampilan yang ada dalam teks *input*. Tampilan hasil ekstraksi keterampilan dapat dilihat pada gambar 12. Kemudian *dataset* keterampilan diolah menggunakan algoritma pemrosesan bahasa alami. Hasil kemudian ditampilkan dalam bentuk daftar keterampilan beserta jumlah kemunculannya.

J. Tampilan Hasil Ekstraksi Lokasi



Gambar 13. Tampilan Hasil Ekstraksi Lokasi

Model *OpenNLP* yang digunakan untuk ekstraksi lokasi adalah model "*en-ner-location.bin*". Model ini berfokus pada entitas lokasi atau tempat dalam teks berbahasa Inggris. Ini dapat mendeteksi lokasi atau tempat yang disebutkan dalam teks, seperti nama kota, negara, jalan, atau tempat geografis lainnya seperti yang terlihat pada gambar 13.

3.6 Hasil Pengujian

Evaluasi sistem pengujian adalah langkah-langkah untuk menguji dan menilai sistem yang telah dirancang dengan maksud memastikan bahwa aplikasi beroperasi berdasarkan dengan persyaratan yang ditetapkan

A. *Blackbox Testing*

Pada penelitian ini, uji *black box* dilaksanakan dengan melakukan uji coba kinerja sistem dari spesifikasi yang dirancang dan membandingkannya berdasarkan hasil yang tercapai, fokus pada pengujian fungsionalitas fitur. Rincian skenario pengujian dan uji diperoleh dari perancangan pengujian *black box* yang telah disusun sebelumnya. Berikutnya, disajikan hasil uji coba menggunakan pendekatan *black box*.

Uji *black box* untuk *backend* ekstraksi informasi resume melibatkan 21 butir pengujian yang dijalankan oleh *developer*, kemudian dapat dihitung tingkat keberhasilannya:

$$Persentase = \frac{\text{skenario uji yang berhasil}}{\text{Total skenario uji}} \times 100\%$$

$$Persentase = \frac{21}{21} \times 100\%$$

$$Persentase = 100\%$$

Jadi, kesimpulan bisa diambil bahwa seluruh skenario pengujian berhasil diselesaikan oleh sistem. Dari hasil uji *black box* pada 21 skenario, disimpulkan bahwa seluruh skenario uji (100%) berfungsi dengan baik seta sesuai standar yang diharapkan.

B. *User Acceptance Test* (UAT)

Pengujian UAT adalah serangkaian uji coba yang dijalankan oleh pengguna dengan maksud untuk menghasilkan

dokumen sebagai bukti penerimaan atau penolakan sistem yang telah dibuat. Jika hasil pengujian dianggap memenuhi kebutuhan pengguna, aplikasi dapat diterapkan. Berikut merupakan hasil perhitungan dari respons yang diberikan oleh para peserta dalam kuesioner telah disiapkan

a. Jawaban Sangat Setuju = 57 × 5 = 285
b. Jawaban Setuju = 26 × 4 = 104
c. Jawaban Kurang Setuju = 7 × 3 = 21
d. Jawaban Tidak Setuju = 0 × 2 = 0
e. Jawaban Sangat Tidak Setuju = 0 × 1 = 0

$$Total\ Skor = SS + S + KS + TS + STS$$

$$Total\ Skor = 285 + 104 + 21 + 0 + 0$$

$$Total\ Skor = 410$$

Hasil skala *likert* mendapatkan nilai 91,1% pada uji penerimaan pengguna (UAT) diartikan sebagai penilaian yang sangat positif. Dengan demikian, dapat ditarik kesimpulan bahwa pernyataan tentang fungsionalitas aplikasi sesuai dengan kebutuhan yang diharapkan pengguna.

4. KESIMPULAN

Berdasarkan hasil penelitian pengembangan *backend* layanan ekstraksi informasi resume, maka dapat disimpulkan sebagai berikut.

1. Rancangan pengembangan *backend* untuk ekstraksi informasi dari resume pengguna mencakup serangkaian tahap yang terinci, melibatkan analisis sistem yang berjalan, identifikasi permasalahan, analisis kebutuhan, perancangan arsitektur API *backend*, dan pemilihan teknologi yang tepat. Penggunaan Model UML digunakan untuk menggambarkan secara visual struktur dan hubungan antar-komponen dalam sistem. Selanjutnya, implementasi dari rancangan ini dilaksanakan dalam pengembangan *backend*, memanfaatkan bahasa pemrograman *Java* dengan dukungan dari kerangka kerja *Spring Boot*. Sebagai langkah akhir, tahap pengujian diatur dengan pembuatan rancangan pengujian, mencakup *Black Box Testing* dan *User Acceptance Test* (UAT).
2. Berdasarkan *User Acceptance Test* (UAT) didapatkan persentase hasil 91,1% dan interpretasi skor sangat baik, ini menunjukkan pengembangan *backend* untuk ekstraksi informasi resume dinyatakan sesuai dengan harapan.

Saran dari penelitian tentang pengembangan *backend* untuk ekstraksi informasi dari resume menyoroti beberapa masukan kunci untuk peningkatan selanjutnya. Pertama, pengembangan saat ini terfokus pada bahasa Inggris, yang menyarankan perlunya diversifikasi dalam bahasa dan jenis data. Langkah kedua adalah menerapkan strategi pelatihan model NLP yang khusus untuk bahasa dan jenis data yang relevan, yang akan meningkatkan akurasi ekstraksi. Selanjutnya, penelitian tentang pemanfaatan *machine learning* dalam sistem ekstraksi informasi dari resume juga diperlukan.

DAFTAR PUSTAKA

[1] Klob, “Tentang kami - Klob,” https://klob.id/about-us. Accessed: Nov. 03, 2023. [Online]. Available: https://klob.id/about-us

[2] M. R. Hanif, Nasrul, and K. Panji, “Analisis dan Perancangan Sistem Informasi Pembayaran Sekolah berbasis Extreme Programming menggunakan Framework MVC,” *Jurnal Informatika Terpadu*, vol. 9, no. 1, pp. 60–67, Mar. 2023, doi: https://doi.org/10.54914/jit.v9i1.639.

[3] M. F. Setiawan, M. N. Witama, and R. Hikmah, “Perancangan Sistem Pengolahan Data Produksi Konveksi Berbasis Java Pada CV Nirwana Bunga Abadi,” *Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI)*, vol. 3, no. 3, pp. 202–208, Dec. 2020, doi: 10.32672/jnkti.v3i3.2435.

[4] W. C. Umbu Dagha, “Web Event, Spring Boot, Java Pembangunan Aplikasi Web Event menggunakan Framework Spring Boot di PT XYZ,” *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, vol. 8, no. 3, pp. 1457–1469, Sep. 2021, doi: 10.35957/jatisi.v8i3.1052.

[5] M. Asqia and T. Nabarian, “Pemanfaatan Google Sheets dan Google Form untuk Layanan Administrasi Mahasiswa Menggunakan Konsep Electronic Service Quality,” *Jurnal Teknologi Terpadu*, vol. 7, no. 1, pp. 15–22, Jul. 2021, doi: 10.54914/jtt.v7i1.339.

[6] S. Munir *et al.*, “Analisis Dan Rancang Bangun Prototype Web Marketplace UMKM Juara,” *Jurnal Teknologi Terpadu*, vol. 6, no. 2, pp. 66–71, Dec. 2020, doi: https://doi.org/10.54914/jtt.v6i2.265.

[7] V. H. Pranatawijaya, W. Widiatry, R. Priskila, and P. B. A. A. Putra, “Penerapan Skala Likert dan Skala Dikotomi Pada Kuesioner Online,” *Jurnal Sains dan Informatika*, vol. 5, no. 2, pp. 128–137, Dec. 2019, doi: 10.34128/jsi.v5i2.185.

[8] K. Taha, R. Davuluri, P. Yoo, and J. Spencer, “Personizing the prediction of future susceptibility to a specific disease,” *PLoS One*, vol. 16, no. 1, p. e0243127, Jan. 2021, doi: 10.1371/journal.pone.0243127.

[9] A. Roy, “Recent Trends in Named Entity Recognition (NER),” *CoRR*, vol. abs/2101.11420, Jan. 2021, doi: https://doi.org/10.48550/arXiv.2101.11420.

[10] S. A. Achsan and Y. A. Susetyo, "RESTFUL WEB SERVICE IMPLEMENTATION USING SPRING FRAMEWORK IN ROOM ASSETS MANAGEMENT SYSTEM," *Jurnal Teknik Informatika (JUTIF)*, vol. 3, no. 2, pp. 395–303, 2022, doi: 10.20884/1.jutif.2022.3.2.213.

[11] J. M. Bintang, M. F. Ashshidiq, and H. F. Dzakwan, "Penerapan Algoritma String Matching dan Regular Expression pada Aplikasi Kamus Besar Bahasa Indonesia (KBBI)," *BIOS : Jurnal Teknologi Informasi dan Rekayasa Komputer*, vol. 4, no. 1, pp. 34–41, Mar. 2023, doi: 10.37148/bios.v4i1.57.