

LAPORAN PROYEK AKHIR

PERANCANGAN *BACKEND* PADA SISTEM
TENAGA KERJA MENGGUNAKAN *RESTAPI*
(STUDI KASUS: PT. CITRATAMA INDONESIA)

Riandy Enrisa
NIM. 2055301121

Pembimbing
Dr. Juni Nurma Sari, S.Kom., M.M.T.

PROGRAM STUDI TEKNIK INFORMATIKA
POLITEKNIK CALTEX RIAU
2024

LAPORAN PROYEK AKHIR

**PERANCANGAN *BACKEND* PADA SISTEM
TENAGA KERJA MENGGUNAKAN *RESTAPI*
(STUDI KASUS: PT. CITRATAMA INDONESIA)**

Riandy Enrisa
NIM. 2055301121

Pembimbing
Dr. Juni Nurma Sari, S.Kom., M.M.T.

**PROGRAM STUDI TEKNIK INFORMATIKA
POLITEKNIK CALTEX RIAU
2024**

HALAMAN PENGESAHAN

PERANCANGAN *BACKEND* PADA SISTEM TENAGA
KERJA MENGGUNAKAN REST API
(STUDI KASUS PT. CITRATAMA INDONESIA)

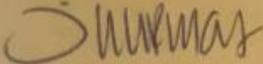
Riandy Enrisa
NIM. 2055301121

Proyek Akhir ini diajukan sebagai salah satu persyaratan untuk
memperoleh gelar Sarjana Terapan Komputer (S.Tr.Kom)
di Politeknik Caltex Riau

Pekanbaru 29 Juli 2024

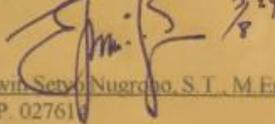
Disetujui oleh:

Pembimbing,

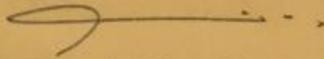


1. Dr. Jumi Nurma Sari, S.Kom.
M.M.T.
NIP. 017218

Penguji

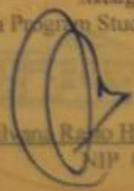
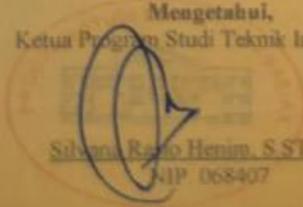


1. Erwin Setyo Nugroho, S.T., M.Eng.
NIP. 02761



2. Warmi Nengsih, S.Kom., M.Kom.
NIP. 088210

Mengetahui,
Ketua Program Studi Teknik Informatika



Solwara Rado Henim, S.ST., M.T.
NIP. 068407

PERNYATAAN

Dengan ini saya menyatakan bahwa dalam proyek akhir yang berjudul :

“Perancangan *Backend* Pada Sistem Tenaga Kerja Menggunakan *RestApi* (Studi Kasus: Pt. Citratama Indonesia)”

Adalah benar hasil karya saya, dan tidak mengandung karya ilmiah atau tulisan yang pernah diajukan di suatu Perguruan Tinggi.

Setiap kata yang dituliskan tidak mengandung plagiat, pernah ditulis maupun diterbitkan orang lain kecuali yang secara tertulis diacu dalam laporan proyek akhir ini dan disebutkan pada daftar pustaka. Saya siap menanggung seluruh akibat apabila terbukti melakukan plagiat.

Pekanbaru, 29 Juli 2024

Riandy Enrisa

ABSTRAK

Penempatan tenaga kerja merupakan proses bisnis yang krusial dalam operasional perusahaan. Di era digital, proses tersebut dapat dioptimalkan melalui digitalisasi. Implementasi sistem berbasis digital membuka peluang bagi PT. Citratama Indonesia untuk mempermudah komunikasi dengan perusahaan-perusahaan mitra. Penelitian ini bertujuan untuk mengembangkan sebuah sistem penempatan tenaga kerja yang digunakan oleh PT. Citra Indonesia sebagai jembatan antara perusahaan mitra dan pelamar kerja. Dalam pengembangan sistem, metode *prototype* digunakan sebagai pendekatan utama. Sistem dibagi menjadi dua bagian yaitu, *frontend* dan *backend*. Penelitian ini berfokus pada pengembangan *backend* dengan menggunakan *framework Laravel* dan database *MySQL*. Arsitektur *Model-View-Controller* (MVC) diterapkan dalam pengembangan sistem ini dengan fokus pada komponen *model* dan *controller* yang menjadi tanggung jawab tim *backend*. Selain itu, juga bertanggung jawab dalam pembuatan API yang akan diakses oleh *frontend* untuk pengiriman dan penerimaan data. Teknologi *chatbot* diterapkan untuk meningkatkan pengalaman pengguna dalam Q&A. Hasil penelitian ini merupakan *REST API* yang dibangun menggunakan *framework Laravel* dan menghasilkan sejumlah *endpoint* yang digunakan oleh *frontend* untuk melakukan interaksi dan mengakses data dari *backend* secara efisien dan aman. Fungsionalitas dari setiap *endpoint* API yang dibangun telah diverifikasi dengan pengujian *blackbox*. Pengujian dengan metode *whitebox* menggunakan *flowgraph* membuktikan bahwa setiap independent path yang menunjukkan alur dari program berjalan dengan baik tanpa kendala.

Kata Kunci: Tenaga kerja, PT. Citratama Indonesia, *back end*, API

ABSTRACT

Employee placement is a crucial business process in company operations. In the digital era, this process can be optimized through digitalization. The implementation of a digital-based system opens up opportunities for PT. Citratama Indonesia to facilitate communication with partner companies. This study aims to develop an employee placement system used by PT. Citratama Indonesia as a bridge between partner companies and job applicants. The prototype method was used as the main approach in the system development. The system is divided into two parts: frontend and backend. This research focuses on backend development using the Laravel framework and MySQL database. The Model-View-Controller (MVC) architecture is applied in this system development, focusing on the model and controller components, which are the responsibility of the backend team. Additionally, the backend team is responsible for creating APIs to be accessed by the frontend for sending and receiving data. Chatbot technology is implemented to enhance user experience in Q&A. The result of this research is a REST API built using the Laravel framework, producing several endpoints used by the front end to interact and access data from the backend efficiently and securely. The functionality of each built API endpoint has been verified with black box testing. Whitebox testing using a flowgraph ensures that every independent path demonstrating the program's flow operates smoothly without issues.

Keywords: *Employee placement, PT. Citratama Indonesia, backend, API*

KATA PENGANTAR

Segala puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat dan barokah-Nya sehingga penulis dapat menyelesaikan proyek akhir yang berjudul “PERANCANGAN *BACKEND* PADA SISTEM TENAGA KERJA MENGGUNAKAN REST API(STUDI KASUS PT.CITRATAMA INDONESIA) ”. Proyek akhir ini disusun sebagai salah satu syarat untuk menyelesaikan jenjang pendidikan Diploma IV pada Program Studi Teknik Informatika Politeknik Caltex Riau.

Pada kesempatan ini, penulis ingin mengucapkan terima kasih kepada pihak yang telah banyak memberikan bantuan dan dukungan yang tiada terhingga baik secara langsung maupun tidak langsung. Ucapan terima kasih tersebut penulis tujukan kepada:

1. Allah *subhanahu wata'ala* yang telah memberikan kesehatan jasmani dan rohani sehingga proyek akhir dapat diselesaikan tepat waktu.
2. Kedua orang tua dan keluarga tercinta yang selalu memberikan motivasi dan tiada henti-hentinya mendoakan dan memberikan semangat.
3. Bapak Dr.Dadang Syarif Sihabudin Sahid, S.Si., M.Sc. selaku Direktur Politeknik Caltex Riau.
4. Ibu Silvana Rasio Henim, S.S.T., M.T. selaku Ketua Program Studi Teknik Informatika yang selalu memberikan semangat dan mengingatkan jadwal sidang.
5. Ibu Meilany Dewi, S.T., M.T. selaku dosen koordinator proyek akhir yang telah membantu, mengarahkan dan mengingatkan untuk selalu mengerjakan proyek akhir.
6. Ibu Dr. Juni Nurma Sari, S.Kom., M.M.T. selaku dosen pembimbing yang telah memberikan sangat banyak bantuan dan bimbingan selama pengerjaan proyek akhir.
7. Bapak Erwin Setyo Nugroho, S.T., M.Eng. selaku penguji 1, dan Ibu Warnia Nengsih, S.Kom., M.Kom. selaku penguji 2 yang telah

menguji dan memberikan arahan untuk penyempurnaan proyek akhir.

8. Seluruh dosen program studi Teknik Informatika yang telah memberikan bekal ilmu kepada penulis dalam menyelesaikan proyek akhir.
9. Sahabat-sahabat senasib seperjuangan yang saling memberikan dukungan dan motivasi sehingga penulis dapat menyelesaikan Proyek Akhir tepat waktu serta teman-teman dan kerabat lainnya yang tidak mungkin disebutkan satu-persatu.

Penulis sangat menyadari sepenuhnya bahwa laporan proyek akhir ini masih jauh dari sempurna, oleh karena itu segala jenis kritik, saran dan masukan yang membangun sangat penulis harapkan agar dapat memberikan wawasan bagi pembaca dan yang paling utama penulis sendiri.

Pekanbaru, 29 Juli 2024

Riandy Enrisa

DAFTAR ISI

HALAMAN JUDUL	ii
HALAMAN PENGESAHAN	Error! Bookmark not defined.
PERNYATAAN	iii
ABSTRAK	v
ABSTRACT	vi
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL	xv
BAB IPENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	4
1.4.1 Tujuan	4
1.4.2 Manfaat	4
1.5 Metodologi Penelitian	4
1.6 Sistematika Penulisan	5
BAB II TINJAUAN PUSTAKA	6
2.1 Penelitian Terdahulu	6
2.2 Landasan Teori	13
2.2.1 Profil PT. Citratama Indonesia	13
2.2.2 Sistem.....	14

2.2.3	Tenaga Kerja	14
2.2.4	Metode <i>Prototype</i>	15
2.2.5	<i>Backend</i>	17
2.2.6	<i>RestAPI</i>	17
2.2.7	<i>Chatbot</i>	18
2.2.8	<i>MySQL</i>	18
2.2.9	<i>Whitebox Testing</i>	18
2.2.10	<i>Blackbox Testing</i>	19
BAB III PERANCANGAN.....		20
3.1	Tahap Pengumpulan Kebutuhan dan Analisis	20
3.1.1	Proses Bisnis	20
3.1.2	Identifikasi Kebutuhan Pengguna.....	2
3.1.3	Identifikasi Aktor (<i>User</i>)	3
3.2	Tahap Perancang Cepat	4
3.2.1	Arsitektur Sistem.....	4
3.2.2	<i>Use Case Diagram</i>	4
3.2.3	<i>Entity Relationship Diagram</i>	5
3.2.4	Perancangan Tabel	6
3.2.5	Keamanan aplikasi	9
3.3	Metode Pengujian	10
3.3.1	<i>Whitebox Testing</i>	10

3.3.2	<i>Blackbox Testing</i>	18
BAB IV	PENGUJIAN DAN ANALISIS	21
4.1	Implementasi <i>REST API</i>	21
4.2	Implementasi dokumen <i>API</i>	25
4.2.1	Dokumentasi <i>API authentication</i>	26
4.2.2	Dokumentasi <i>API admin</i>	28
4.2.3	Dokumentasi <i>API pelamar kerja</i>	39
4.2.4	Dokumentasi <i>API perusahaan mitra</i>	42
4.3	Pengujian	45
4.3.1	Pengujian <i>whitebox</i>	45
4.3.2	Pengujian <i>Blackbox</i>	65
4.4	Analisis Hasil	69
4.4.1	Analisis implementasi <i>REST API</i>	69
4.4.2	Analisis Pengujian <i>Whitebox</i>	69
4.4.3	Analisis Pengujian <i>Blacbox</i>	69
BAB V	PENUTUP	71
5.1	Kesimpulan	71
5.2	Saran	71
DAFTAR PUSTAKA	73
LAMPIRAN A	A-1
LAMPIRAN B	B-1
LAMPIRAN C	C-1
LAMPIRAN D	D-17

DAFTAR GAMBAR

Gambar 2. 1 Metode <i>Prototype</i> (Frederica et al., 2021).....	15
Gambar 3.1 Proses Bisnis Sebelum Sistem	20
Gambar 3.2 Proses Bisnis Sesudah Sistem	1
Gambar 3.3 Arsitektur Sistem	4
Gambar 3.4 <i>Use Case Diagram</i>	5
Gambar 3.5 <i>Entity Relationship Diagram</i>	6
Gambar 3.6 <i>Flowchart</i> PT. Citratama Indonesia	10
Gambar 3.7 <i>Flowgraph</i> PT. Citratama Indonesia	11
Gambar 3.8 <i>Flowchart</i> Perusahaan Mitra.....	13
Gambar 3.9 <i>Flowgraph</i> Perusahaan Mitra.....	14
Gambar 3.10 <i>Flowchart</i> Pelamar Kerja.....	16
Gambar 3.11 <i>Flowgraph</i> Pelamar Kerja.....	16
Gambar 4.1 Implementasi API Registrasi	26
Gambar 4.2 Implementasi API <i>Login</i>	27
Gambar 4.3 Implementasi API <i>Logout</i>	27
Gambar 4.4 Implementasi API Berita.....	28
Gambar 4.5 API Mengambil Data Berita.....	28
Gambar 4.6 API Menambah Berita	29
Gambar 4.7 API Mengubah Data Berita	29
Gambar 4.8 API Menghapus Data Berita	30
Gambar 4.9 Implementasi API Pelamar Kerja.....	30
Gambar 4.10 API Mengambil Data Pelamar Kerja.....	31
Gambar 4.11 API Menambah Data Pelamar Kerja	32

Gambar 4.12 <i>API</i> Mengubah Data Pelamar Kerja	32
Gambar 4.13 Implementasi <i>API</i> Perusahaan Mitra.....	33
Gambar 4.14 <i>API</i> Mengambil Data Perusahaan Mitra.....	33
Gambar 4.15 <i>API</i> Menambah Perusahaan Mitra.....	34
Gambar 4.16 <i>API</i> Mengubah Data Perusahaan Mitra.....	35
Gambar 4.17 Implementasi <i>API Approve Recruitment Admin</i>	35
Gambar 4.18 <i>API Approve Recruitment Admin</i>	36
Gambar 4.19 Implementasi <i>API</i> Kirim Pesan.....	36
Gambar 4.20 <i>API</i> Mengirim Pesan.....	36
Gambar 4.21 <i>API</i> Tambah Pertanyaan	37
Gambar 4.22 <i>API</i> Menampilkan Data <i>Chatbot</i>	37
Gambar 4.23 <i>API</i> Mengubah Data <i>Chatbot</i>	38
Gambar 4.24 Implementasi <i>Chatbot</i>	39
Gambar 4.25 Implementasi <i>API</i> Data Diri.....	39
Gambar 4.26 <i>API</i> Menambahkan Data Diri.....	40
Gambar 4.27 <i>API</i> Mengambil Data Diri	41
Gambar 4.28 <i>API</i> Mengubah Data Diri	42
Gambar 4.29 Implementasi <i>API</i> Pelamar Kerja Tersedia	42
Gambar 4.30 <i>API</i> Mengambil Data Pelamar Kerja.....	43
Gambar 4.31 Implementasi <i>API Recruitment</i>	43
Gambar 4.32 <i>API</i> Mengambil Data <i>Recruitment</i>	44
Gambar 4.33 <i>API</i> Menambahkan Data <i>Recruitment</i>	44
Gambar 4.34 <i>API</i> Konfirmasi <i>Recruitment</i>	45
Gambar 4.35 <i>Flowchart Login Admin</i>	46
Gambar 4.36 <i>Flowgraph Login Admin</i>	47

Gambar 4.37 <i>Flowchart</i> Input Data Pelamar Kerja	49
Gambar 4.38 <i>Flowgraph</i> Input Data Pelamar Kerja	50
Gambar 4.39 <i>Flowchart</i> Proses <i>Recruitment</i>	52
Gambar 4.40 <i>Flowgraph</i> Proses <i>Recruitment</i>	53
Gambar 4.41 <i>Flowchart</i> Tambah Jadwal.....	55
Gambar 4.42 <i>Flowgraph</i> Tambah Jadwal.....	56
Gambar 4.43 <i>Flowchart</i> Input Data Diri	59
Gambar 4.44 <i>Flowgraph</i> Input Data Diri	60
Gambar 4.45 <i>Flowchart</i> <i>Chatbot</i>	62
Gambar 4.46 <i>Flowgraph</i> Input Data Diri	63
Gambar 4.47 Pengujian autentikasi berhasil.....	66
Gambar 4.48 Pengujian autentikasi gagal.....	67
Gambar 4.49 Pengujian chatbot 1 kata kunci	68
Gambar 4.50 Pengujian chatbot 2 kata kunci	68

DAFTAR TABEL

Tabel 2.1 Penelitian Terdahulu.....	8
Tabel 3.1 Dekripsi Aktor	3
Tabel 3.2 Tabel Perusahaan Mitra	6
Tabel 3.3 Tabel Admin.....	7
Tabel 3.4 Tabel Pelamar Kerja	7
Tabel 3.5 Tabel <i>Recruitment</i>	8
Tabel 3.6 Tabel Jadwal.....	8
Tabel 3.7 Tabel Penerima Jadwal	8
Tabel 3.8 Tabel Berita	9
Tabel 3.9 Tabel Faq.....	9
Tabel 3.10 Tabel <i>Notifications</i>	9
Tabel 3.11 <i>Case</i> PT. Citratama Indonesia	11
Tabel 3.12 <i>Case</i> Perusahaan Mitra	14
Tabel 3.13 <i>Case</i> Pelamar Kerja	17
Tabel 3.14 Pengujian <i>Blackbox</i>	18
Table 4.1 Hasil implementasi <i>API</i>	21
Table 4.2 Test case login admin	48
Table 4.3 Test case input data pelamar kerja	50
Table 4.4 Test case proses recruitment	54
Table 4.5 Test case tambah jadwal	57
Table 4.6 Test case input data diri	61
Table 4.7 Test case chatbot	64
Table 4.8 Rekapitulasi hasil whitebox	64

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada era sekarang semua sudah menggunakan teknologi untuk mempermudah manusia dalam melakukan aktivitas maupun bisnis. Salah satu bisnis sekarang adalah penempatan tenaga kerja, tenaga kerja merupakan faktor penting yang berpengaruh terhadap kinerja perusahaan. Penempatan tenaga kerja merupakan proses penting dalam manajemen sumber daya manusia di berbagai organisasi atau perusahaan. Penempatan tenaga kerja ini melibatkan penempatan individu ke peran atau posisi tertentu sesuai dengan kualifikasi, keahlian, dan kebutuhan organisasi.

PT. Citratama Indonesia adalah perusahaan yang berdedikasi dalam bidang penempatan kerja dan tenaga kerja yang berkualitas. PT. Citratama Indonesia menjembatani antara para pelamar kerja yang memiliki potensi dan keterampilan unggul dengan peluang pekerjaan yang sesuai dengan kemampuan mereka, PT. Citratama Indonesia menjadi penyalur tenaga kerja yang siap bekerja dengan perusahaan mitra. PT. Citratama Indonesia telah melakukan perekrutan kurang lebih 50 pelamar yang sudah ditempatkan dan 30 pelamar yang belum di tempatkan, PT. Citratama Indonesia telah bekerjasama dengan 28 perusahaan mitra. PT. Citratama Indonesia memiliki permasalahan yaitu perusahaan tidak memiliki administrasi yang tertata, proses pendaftaran dan penyimpanan data pelamar kerja masih menggunakan google form, PT. Citratama Indonesia melakukan pengiriman berkas ke mitra perusahaan menggunakan email atau melakukan kunjungan langsung.

Dengan uraian masalah yang ada, solusi dari masalah adalah perancangan sistem yang dapat memanajemen perekrutan pelamar kerja dan perusahaan, dengan melakukan pendekatan *prototype*. Dengan dibangunnya sistem ini diharapkan dapat dijadikan sarana penempatan tenaga kerja oleh PT. Citratama Indonesia yang telah berkerja sama dengan perusahaan mitra bagi pelamar kerja yang sudah terdaftar. Pada bagian pelamar kerja menampilkan berita-berita terkait mengenai pekerjaan dan pelamar kerja dapat melakukan unggah data diri dan berkas-berkas yang diperlukan, serta terdapat *chatbot* untuk

tempat pelamar kerja bertanya. Pada bagian perusahaan menampilkan list data dari pelamar kerja, perusahaan mitra dapat membuat list pelamar kerja yang ingin direkrut dan mengirim *list* tersebut ke PT. Citratama Indonesia. Pada bagian PT. Citratama Indonesia menampilkan data perusahaan dan data dari pelamar kerja, sebelumnya PT. Citratama Indonesia perusahaan dalam mendaftarkan pelamar kerja yang sudah ditempatkan di perusahaan mitra, pada bagian halaman PT. Citratama Indonesia juga menampilkan status pelamar kerja yang sudah ditempatkan pada perusahaan mitra. Sistem ini dibagi menjadi *front end* dan *back end*, dengan metode *prototype* ini user dapat berpartisipasi aktif dalam pengembangan sistem, sehingga hasil dari sistem yang dibangun bisa sesuai dengan kebutuhan dan keinginan user.

Penulis sendiri mengerjakan bagian *back end* pada sistem ini, teknologi yang digunakan pada sistem ini yaitu penggunaan framework laravel dan database *MySQL*. Konsep dalam pengembangan ini menggunakan MVC, MVC adalah kepanjangan dari *model*, *view*, *controller* yang merupakan sebuah design arsitektur yang memudahkan pengembangan sebuah sistem. Untuk konsep MVC bagian *model* dan *controller* dikerjakan oleh *back end*. *Model* merupakan bagian yang berguna untuk mengatur, menyiapkan, dan mengorganisasikan data yang ada di database, *controller* merupakan bagian yang menghubungkan *model* dan *view* pada setiap proses request dari user. Pada bagian ini *back end* membuat sebuah *API* yang nantinya dapat diakses oleh *front end* untuk proses pengiriman dan penerimaan data. Pada pengembangan ini juga menggunakan teknologi *chatbot*, yang mana *chatbot* ini digunakan untuk *Q&A* bagi pelamar kerja, sehingga *chatbot* ini bisa menjadi sarana pelamar kerja untuk menanyakan pertanyaan kepada admin tanpa perlu menunggu lama. Pada bagian *back end* ini juga membuat fitur notifikasi WA, fitur notifikasi ini dibuat dengan tujuan agar admin dapat mengirimkan informasi kepada pelamar kerja melalui notifikasi WA. Terdapat dua pengujian pada penelitian ini yaitu *white box testing* dan *black box testing*. *White box testing* menggunakan metode control flow testing, yang mana *control flow testing* menguji alur dari sebuah program, dengan pengujian *control flow* dapat mengetahui apakah alur program sudah berjalan dengan baik atau belum. pengujian *blackbox* menguji fungsionalitas dari sistem yang telah dirancang. Memastikan setiap

endpoint API berfungsi dengan baik, mengirimkan *request* ke setiap *endpoint* dan memeriksa *response* yang diberikan, *response* sesuai dengan yang diharapkan berdasarkan spesifikasi. *REST API* dipilih untuk pengembangan proyek ini karena *REST API* sangat sesuai untuk *frontend* yang terpisah dan berkomunikasi dengan *backend* melalui *endpoint*. *REST API* mudah diimplementasikan dan lebih fleksibilitas dalam format data menggunakan metode *http* standar. *REST API* didukung oleh banyak alat bantu dan framework yang memudahkan pengembangan dan dokumentasi.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan di atas, maka perumusan masalah yang dihadapi oleh PT. Citratama Indonesia adalah:

- 1) Tidak adanya sistem manajemen untuk proses perekrutan pelamar kerja.
- 2) Bagaimana cara merancang dan mengimplementasikan sistem backend yang efektif untuk mengelola dan memproses data secara efisien?

1.3 Batasan Masalah

Adapun batasan masalah dalam penelitian ini adalah sebagai berikut:

- 1) Proyek akhir ini berfokus pada perancangan back end sistem tenaga kerja pada PT. Citratama Indonesia dengan menggunakan *Rest API*.
- 2) Pengembangan sistem ini menggunakan metode *prototype*.
- 3) Membuat sebuah fitur *chatbot* untuk *Q&A* menggunakan *API*.
- 4) Membuat fitur notifikasi *whatsapp* untuk mengirimkan pesan instan.
- 5) Pengujian sistem ini melakukan pengujian *white box testing* dan *black box testing*.

1.4 Tujuan Penelitian

1.4.1 Tujuan

Adapun tujuan dari penelitian ini adalah merancang dan membangun sebuah sistem perancangan *back end* tenaga kerja pada PT. Citratama Indonesia menggunakan metode *Rest API*.

1.4.2 Manfaat

Manfaat yang diharapkan dari penelitian ini adalah:

- 1) Memudahkan bagian *frontend* untuk memproses data pada sistem tenaga kerja Pt. Citratama Indonesia.
- 2) Meningkatkan efektifitas alur program dari sistem tenaga kerja yang dibangun.

1.5 Metodologi Penelitian

Adapun metode penelitian yang digunakan dalam pembuatan proyek ini adalah:

1) Wawancara

Pada tahapan penelitian ini, dilakukan wawancara dengan pihak Pt. Citratama untuk mengidentifikasi permasalahan dan kebutuhan mereka.

2) Studi Literatur

Metode ini dilakukan dengan membaca jurnal, penelitian terdahulu, *website* yang berhubungan dengan penilaian proyek akhir.

3) Perancangan

Perancangan yang dilakukan pada pembuatan sistem ini yaitu identifikasi masalah, arsitektur sistem, *user requirement*, *use case diagram*, dan *Entity Relationship Diagram* (ERD).

4) Implementasi

Tahapan Implementasi merupakan tahapan melakukan implementasi desain website kedalam bahasa pemrograman *PHP* dan *MySQL* sebagai *database* sesuai dengan pemodelan yang dibangun.

5) Pengujian

Pengujian pada sistem ini menggunakan *black box testing* dan *white box testing*.

6) Analisis dan Evaluasi

Tahapan ini digunakan untuk menganalisis dan mengevaluasi sistem yang sudah dibangun apakah sudah sesuai atau belum berdasarkan pengujian yang dilakukan sebelumnya.

1.6 Sistematika Penulisan

Sistematika penulisan laporan proyek akhir ini secara keseluruhan terdiri dari empat bab, masing-masing terdiri dari beberapa sub bab. Adapun pokok pembahasan dari masing-masing bab tersebut secara garis besar sebagai berikut:

BAB I PENDAHULUAN

Bab ini menyajikan penjelasan secara umum tentang sistem yang akan dibangun. Penjelasan tersebut meliputi latar belakang masalah, perumusan masalah, batasan masalah, tujuan dan manfaat penelitian, metodologi penelitian dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini menyajikan beberapa hasil penelitian terdahulu dan landasan teori yang diperlukan untuk merancang sistem.

BAB III PERANCANGAN

Bab ini menyajikan tentang perancangan sistem terdiri dari arsitektur sistem, metodologi penelitian, dan implementasi.

BAB IV PENGUJIAN DAN ANALISIS

Bab ini menyajikan mengenai pengujian sistem dan analisis hasil. Sistem diuji menggunakan beberapa cara dan setiap metode dianalisis secara rinci.

BAB V KESIMPULAN DAN SARAN

Bab ini menyajikan kesimpulan dan saran setelah melakukan pengujian dan analisa sistem. Saran diberi untuk pengembangan sistem lebih lanjut di masa mendatang.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian terdahulu digunakan sebagai landasan dalam penulisan proyek akhir untuk sebagai masukan dan konsep, yaitu:

Penelitian yang pertama yang pernah dilakukan oleh (Wulandari & Sutiyatno, 2019) dengan judul Sistem Informasi Penyaluran Tenaga Kerja Berbasis Web di CV. Jasa Informasi Kerja Magelang. CV. Jasa Informasi Kerja (JIK) adalah suatu badan usaha yang melakukan penyaluran tenaga kerja. JIK menyediakan lowongan pekerjaan kepada calon tenaga kerja (canaker), dan menyediakan tenaga kerja bagi perusahaan. Canaker datang ke JIK untuk mengisi data diri. JIK akan menghubungi canaker selama kurang lebih dua minggu untuk memberikan informasi lowongan pekerjaan, kemudian canaker yang berminat harus datang kembali ke JIK dengan membawa surat lamaran. Canaker harus mengisi formulir pendaftaran, canaker yang terdaftar harus mengikuti sejumlah rangkaian tes. Hasil tes akan diberitahukan ke canaker kurang lebih dua minggu setelah tes melalui SMS. Berdasarkan proses pendaftaran, seleksi dan hasil ditemukan beberapa kekurangan diantaranya akses informasi lowongan harus menunggu selama dua minggu, dan canaker juga harus melakukan dua tahap yang sama. Penelitian ini bertujuan merancang dan membangun sistem informasi penyaluran tenaga kerja berbasis web yang dapat mengatasi kelemahan yang ada di CV. Jasa Informasi Kerja Magelang. Metodologi penelitian yang digunakan adalah metode rekayasa model *SDLC (System Development Life Circle)*. Hasil Penelitian ini adalah aplikasi sistem informasi penyaluran tenaga kerja berbasis web di CV. Jasa Informasi Kerja Magelang dengan menggunakan bahasa pemrograman *PHP* dan *MySQL* sebagai database.

Penelitian yang kedua yang pernah dilakukan oleh (Abdurrohman & Warih Utami, 2023) dengan judul Rancang Bangun Aplikasi Pemetaan Tenaga Kerja Berbasis Web Sistem Informasi Geografis pada Disnakertrans Jawa Timur Studi Kasus Bidang Penempatan dan Pelatihan Kerja. Dinas Ketenagakerjaan dan Transmigrasi Jawa Timur (Disnakertrans Jatim), khususnya pada Bidang Penempatan dan Pelatihan Kerja membutuhkan suatu

pengkajian pemetaan tenaga kerja serta peningkatan progresif teknologi dalam mengelola, dan menangani tenaga kerja. Untuk mempermudah tugas tersebut proses bisnis melalui pengkajian pemetaan sebaran tenaga kerja dilakukan, oleh karena itu, muncul gagasan untuk mengembangkan sebuah aplikasi berbasis web yang berguna untuk memberikan gambaran berupa peta digital interaktif yang memuat data tenaga kerja dan klasifikasinya. Aplikasi ini akan membantu dalam proses seperti analisis internal, administrasi kerja, dan penghitungan kebutuhan dana pelatihan kerja tahunan di setiap Balai Latihan Kerja (BLK) di daerah. Hasil dari penelitian ini adalah sebagai berikut; 1) Menampilkan peta digital sebaran tenaga kerja Jatim 2 Memiliki keamanan yang cukup 3) Terintegrasi dengan laporan tenaga kerja 4) Membantu penentuan besaran alokasi dana pelatihan tahunan pada UPT (Unit Pelaksana Teknis) BLK (Balai Latihan Kerja) di daerah.

Penelitian yang ketiga yang pernah dilakukan oleh (Dwi Haristya Aryadi et al., 2022) dengan judul Sistem Informasi Manajemen Tenaga Kerja Outsourcing Pada PT. ACM Berbasis Java. PT. ACM yang merupakan perusahaan jasa tenaga kerja outsourcing yang mana pada sistem berjalan saat ini, masih menggunakan cara konvensional seperti pendataan pada kertas, yang mana selain memakan tempat serta rawan rusak dan hilang, juga membuat pencarian informasi memakan waktu yang lama. Tujuan penelitian ini adalah membangun sistem informasi manajemen tenaga kerja outsourcing menggunakan *Java* dan *MySQL* untuk mengatasi masalah pada sistem berjalan perusahaan. Metode penelitian yang digunakan adalah *Research and Development (R&D)*. Penelitian ini menghasilkan sebuah rancangan Sistem Informasi Manajemen Tenaga Kerja Outsourcing yang diharapkan dapat membantu perusahaan dalam mengelola tenaga kerja outsourcing yang lebih baik, serta dapat meminimalisasi kerusakan dan kehilangan data, juga penyajian informasi yang lebih cepat, tepat, dan akurat.

Adapun untuk menelaah perbandingan dari masing-masing penelitian di atas secara lebih spesifik, dapat dilihat pada tabel 2.1 berikut.

Tabel 2.1 Penelitian Terdahulu

Judul Penelitian	Masalah	Tujuan	Metode	Hasil	Pengembangan
Sistem Informasi Penyaluran Tenaga Kerja Berbasis Web di CV. Jasa Informasi Kerja Magelang	Seleksi dan hasil ditemukan beberapa kekurangan diantaranya akses informasi lowongan harus menunggu selama dua minggu, dan canaker juga harus melakukan dua tahap yang sama.	Merancang dan membangun sistem informasi penyaluran tenaga kerja berbasis web yang dapat mengatasi kelemahan yang ada di CV. Jasa Informasi Kerja Magelang.	Sistem Development Life Cycle (SDLC)	Aplikasi sistem informasi penyaluran tenaga kerja berbasis web di CV. Jasa Informasi Kerja Magelang dengan menggunakan bahasa pemrograman PHP dan MySQL sebagai	Pengembangan lebih lanjut dengan sistem dapat menampilkan laporan tenaga kerja dengan memfilter tanggal dan bulan yang diinginkan. Menampilkan hasil rekapitulasi dengan diagram agar hasil laporan rekapitulasi lebih menarik dan kenaikan ataupun penurunan tenaga kerja disalurkan bisa terlihat dengan jelas.

Judul Penelitian	Masalah	Tujuan	Metode	Hasil	Pengembangan
				ai database.	
Rancang Bangun Aplikasi Pemetaan Tenaga Kerja Berbasis Web Sistem Informasi Geografis pada Disnakertr ans Jawa Timur Studi Kasus Bidang Penempatan dan Pelatihan Kerja	Dinas Ketenag akerjaan dan Transmigrasi Jawa Timur (Disnakertr ans Jawa Timur Bidang Penempatan dan Pelatihan Kerja membut uuhkan suatu pengkajian pemetaan tenaga kerja serta peningkatan progresif teknologi dalam mengelola, dan	Aplikasi ini akan membantu dalam proses seperti analisis internal, administrasi kerja, dan penghitungan kebutuhan dana pelatihan kerja tahunan di setiap Balai Latihan Kerja (BLK) di daerah	<i>Waterfall</i>	Menganalisis peta digital sebaran tenaga kerja Jawa Timur 2) Memeriksa keamanan yang cukup 3) Terintegrasi dengan laporan tenaga kerja 4) Membantu penentuan	Fitur-fitur tambahan seperti integrasi dengan system lain, notifikasi real-time, dan lain-lain perlu dikembangkan agar aplikasi memiliki nilai tambah dan memberikan solusi lebih baik bagi pengguna.

Judul Penelitian	Masalah	Tujuan	Metode	Hasil	Pengembangan
	menangani tenaga kerja			besar an aloka si dana pelati han tahun an pada UPT (Unit Pelak sana Tekni s) BLK (Balai Latih an Kerja) di daera h.	
Sistem Informasi Manajemen Tenaga Kerja Outsourcing Pada PT. ACM Berbasis Java	PT. ACM yang merupakan perusahaan jasa tenaga kerja outsourcing yang mana pada sistem	Tujuan peneliti an ini adalah memba ngun sistem informa si manaje men tenaga kerja outsour	Research and Develop ment (R&D)	Ranc an Siste m Infor masi Mana jemen Tena ga Kerja Outso urcin g	Sistem informasi ini dapat dikembangkan untuk kedepannya seiring dengan kebutuhan perusahaan yang akan datang. Sistem perlu peninjauan dan pemeliharaan,

Judul Penelitian	Masalah	Tujuan	Metode	Hasil	Pengembangan
	berjalan saat ini, masih menggunakan cara konvensional seperti pendataan pada kertas, yang mana selain memakan tempat serta rawan rusak dan hilang, juga membuat pencarian informasi memakan waktu yang lama.	cung menggunakan Java dan MySQL untuk mengatasi masalah pada sistem berjalan perusahaan. Metode penelitian yang digunakan adalah Research and Development (R&D).		yang diharapkan dapat membantu perusahaan dalam mengelola tenaga kerja outsourcing yang lebih baik, serta dapat meminimalisasi kerusakan dan kehilangan data, juga penyajian informasi yang lebih	terutama untuk back-up data pada database.

Judul Penelitian	Masalah	Tujuan	Metode	Hasil	Pengembangan
				cepat, tepat, dan akurat.	
Perancangan backend pada sistem tenaga kerja dengan menggunakan rest api (studi kasus PT. Citratam indonesia)	Tidak memiliki administrasi yang tertata, proses pendaftaran dan penyimpanan data pelamar kerja masih menggunakan google form, PT. Citratama Indonesia melakukan pengiriman berkas ke mitra perusahaan menggunakan	Merancang dan membangun sebuah sistem perancangan backend tenaga kerja pada PT. Citratama Indonesia menggunakan metode Rest API	Prototipe	Sistem ini diharapkan dapat dijadikan sarana pempatan tenaga kerja oleh PT. Citratama Indonesia yang telah berkerja sama dengan perusahaan mitra bagi pelam	Sistem ini kedepannya akan melakukan pengembangan tambahan seperti fitur notifikasi disistem, fitur notifikasi WA dan fitur chatbot.

Judul Penelitian	Masalah	Tujuan	Metode	Hasil	Pengembangan
	email atau melakukan kunjungan langsung			ar kerja yang sudah terdaftar.	

Setelah adanya tabel perbandingan diatas maka dapat diambil kesimpulan berdasarkan tiga penelitian terdahulu dapat diambil beberapa pengembangan yang bisa dilakukan seperti menampilkan grafik penempatan tenaga kerja di halaman *dashboard* sehingga dapat melihat jumlah tenaga kerja yang sudah ditempatkan dan sistem memiliki fitur yang dapat memberikan notifikasi. Dari pengembangan tersebut dilakukan pada penelitian ini untuk menampilkan grafik dan fitur notifikasi.

2.2 Landasan Teori

2.2.1 Profil PT. Citratama Indonesia

PT. Citratama Indonesia adalah perusahaan yang berdedikasi dalam bidang penempatan kerja dan rekrutmen tenaga kerja yang berkualitas. Dengan pengalaman dan komitmen yang kuat, PT. Citratama Indonesia menjembatani antara para pencari kerja yang memiliki potensi dan keterampilan unggul dengan peluang pekerjaan yang sesuai dengan kemampuan mereka. PT. Citratama Indonesia menjalin kerjasama yang erat dengan berbagai perusahaan dan organisasi. Hal ini memungkinkan untuk memiliki wawasan mendalam tentang kebutuhan tenaga kerja di berbagai perusahaan dan membantu klien menemukan kandidat terbaik untuk mengisi posisi yang kosong. Selain menghubungkan pencari kerja dengan perusahaan, PT. Citratama Indonesia juga berkomitmen untuk mengembangkan potensi tenaga kerja dengan menyediakan pelatihan dan pengembangan keterampilan guna meningkatkan daya saing pencari kerja di dunia kerja.

PT. Citratama Indonesia menjadi penyalur tenaga kerja yang siap bekerja dengan perusahaan mitra. PT. Citratama Indonesia telah melakukan perekrutan kurang lebih 50 pelamar yang sudah ditempatkan dan 30 pelamar yang belum di tempatkan, PT. Citratama Indonesia telah bekerjasama dengan 28 perusahaan mitra. PT. Citratama Indonesia memiliki beberapa layanan yaitu penempatan kerja sama, penempatan umum, vendor penjualan, vendor marketing. Penempatan kerja sama berfokus pada kerja sama dengan lembaga atau institusi dalam menyalurkan tenaga kerja siap pakai, penempatan umum berfokus pada menjadi penyalur tenaga kerja siap pakai tanpa biaya, vendor penjualan berfokus pada kerja sama dengan perusahaan dan menjadi vendor penjualan berbasis *performance*, vendor marketing berfokus pada kerja sama dengan lembaga atau institusi dan menjadi vendor marketing berbasis *performance*.

2.2.2 Sistem

Kata sistem berasal dari bahasa Yunani, yaitu *systema*, yang artinya himpunan bagian atau komponen yang saling berhubungan secara teratur dan merupakan suatu keseluruhan. Selain itu, bisa diartikan sekelompok elemen yang independen, namun saling terkait sebagai satu kesatuan. Sistem terdiri atas struktur dan proses. Struktur sistem merupakan unsur-unsur yang membentuk sistem tersebut, sedangkan proses sistem menjelaskan cara kerja setiap unsur sistem dalam mencapai tujuan. Setiap sistem merupakan bagian dari sistem lain yang lebih besar dan terdiri atas berbagai sistem yang lebih kecil, yang disebut subsistem. Setiap sistem diciptakan untuk menangani sesuatu yang berulang-ulang atau yang secara rutin terjadi. (Rusdiana et al., 2014).

2.2.3 Tenaga Kerja

Sumber daya manusia (SDM) atau *human resources* mengandung dua pengertian. Pertama, sumber daya manusia mengandung pengertian usaha kerja atau jasa yang dapat diberikan dalam proses produksi. Dalam hal ini SDM mencerminkan kualitas usaha yang diberikan oleh seorang dalam waktu tertentu untuk menghasilkan barang dan jasa. Pengertian kedua dari SDM menyangkut manusia yang mampu bekerja untuk memberikan jasa

atau usaha kerja tersebut. Mampu bekerja berarti mampu melakukan kegiatan yang mempunyai nilai ekonomis, yaitu bahwa kegiatan tersebut menghasilkan barang atau jasa untuk memenuhi kebutuhan masyarakat. Secara fisik, kemampuan bekerja diukur dengan usia. Dengan kata lain, orang dalam usia kerja dianggap mampu bekerja. Kelompok penduduk dalam usia kerja tersebut dinamakan tenaga kerja atau man power. Secara singkat, tenaga kerja didefinisikan sebagai penduduk dalam usia kerja. (Gwijangge et al., n.d., 2018).

2.2.4 Metode *Prototype*

Prototype adalah salah satu versi dari sebuah sistem potensial yang memberikan ide bagi para pengembang dan calon pengguna, bagaimana sistem berfungsi dalam bentuk yang telah selesai. Metode ini merupakan suatu proses pembentukan model untuk memberikan gambaran dasar mengenai alat dalam tahap awal pengujian. (Frederica Rosabel Ramli et al., 2021). Gambar 2.1 dibawah menunjukkan pendekatan *prototype* secara umum dan melibatkan beberapa yang dapat di lihat pada gambar 2.1 berikut.



Gambar 2. 1 Metode *Prototype* (Frederica Rosabel Ramli et al., 2021)

Sesuai dengan metodologi *prototype* berikut adalah tahapan pengembangan aplikasi:

1. Pengumpulan Kebutuhan

Pengguna dan pengembang bersama-sama mendefinisikan format seluruh perangkat lunak, mengidentifikasi semua kebutuhan, dan garis besar sistem yang dibuat, sehingga hasil yang didapat sesuai dengan keinginan dari pengguna.

2. Membangun *Prototyping*

Pada tahap ini melibatkan desain sistem yang memerlukan perbaikan jika terdapat ketidaksesuaian antara pengguna dan pengembang sistem sehingga tingkat partisipasi aktif dari pengguna sangat mempengaruhi pencapaian tujuan yang telah disepakati. Tahapan ini memungkinkan pengguna untuk memberikan umpan balik terhadap sistem dengan mengacu pada dokumentasi kebutuhan *user* yang telah dibuat pada tahap sebelumnya. Hasil dari tahap ini adalah spesifikasi perangkat lunak yang mencakup arsitektur sistem, struktur data, dan elemen-elemen lainnya.

3. Evaluasi *Prototyping*

Evaluasi ini dilakukan oleh pelanggan apakah *prototyping* yang sudah dibangun sudah sesuai dengan keinginan pengguna. Jika sudah sesuai maka langkah 4 akan diambil. Jika tidak *prototyping* direvisi dengan mengulang langkah 1, 2, dan 3 (perubahan desain).

4. Mengkodekan Sistem/Implementasi sistem

Dalam tahap ini *prototyping* yang sudah disepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai dan dilakukan penerapan yaitu memvisualkan desain sistem yang telah disetujui antara pengguna dan pengembang

5. Menguji sistem

Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, sistem dilakukan pengujian terlebih dahulu sebelum digunakan. Untuk memastikan sistem yang dikembangkan berjalan lancar tanpa adanya kesalahan atau error, serta memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan Pengujian ini dilakukan dengan *White Box* dan *Black Box testing*.

6. Evaluasi sistem

Pengguna mengevaluasi apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan. Jika ya, langkah 7 dilakukan; jika tidak, ulangi langkah 4 dan 5.

7. Menggunakan sistem/Pemeliharaan sistem

Perangkat lunak yang telah diuji dan diterima pelanggan siap untuk digunakan.

2.2.5 Backend

Backend adalah tempat di mana proses pada suatu sistem informasi atau aplikasi berjalan, data dapat ditambahkan, diubah maupun dihapus. *Backend* biasanya mengurus segala jenis proses yang tidak berhubungan langsung dengan pengguna, seperti server dan basis data. *Backend* dibutuhkan dalam pengembangan sistem dan manajemen data pada sistem (Pangestika & Dirgahayu, 2020). *Backend* merupakan suatu program yang berjalan pada sisi server (server-side) yang melakukan tugas untuk berinteraksi langsung dengan basis data dalam melakukan manipulasi data ke basis data, sehingga *backend* tidak melakukan interaksi secara langsung kepada pengguna. *Backend* memiliki bagian yang bertanggung jawab dalam mengelola aplikasi dan database agar dapat saling berkomunikasi dengan baik dan guna mendukung antarmuka sistem bekerja sesuai dengan fungsinya.

2.2.6 RestAPI

Representational State Transfer (REST) adalah sebuah gaya arsitektur untuk pendistribusian sistem hypermedia. Arsitektur *REST* adalah arsitektur klien – server dimana klien mengirim request pada server dan server memproses request dan mengembalikan sebuah response (transaksi). Setiap transaksi bersifat independen dan tidak terkait dengan transaksi lainnya (stateless). Hal ini yang membuat aplikasi *REST* sederhana dan ringan. *RESTful* web service adalah sebutan untuk aplikasi web yang menggunakan arsitektur REST. *RESTful* web service menggunakan metode http *GET*, *POST*, *PUT*, dan *DELETE* untuk menerima, membuat, memperbarui dan menghapus resource. *Application Programming Interface* (API) adalah sebuah tools yang membuat beberapa sistem dapat saling terhubung. Pada *API* terdapat dua bagian, yaitu server yang berfungsi

sebagai penyedia dari *API* dan klien yang berbentuk sebuah program yang mengetahui data apa yang tersedia pada *API* dan dapat memanipulasi data tersebut sesuai request pengguna. (Galindra Wardhana et al., 2020).

2.2.7 *Chatbot*

Chatbot adalah perangkat lunak yang memiliki kemampuan untuk melakukan komunikasi dengan manusia secara otomatis dengan format pesan singkat tertulis (chat). *Chatbot* adalah salah satu produk dari bidang ilmu pengolahan bahasa natural yang dikembangkan dengan menggunakan metode kecerdasan buatan agar dapat memproses informasi dan memberikan jawaban yang paling tepat sesuai dengan diharapkan pengguna. *Chatbot* diharapkan dapat menjadikan komputer atau sebuah mesin agar sanggup menggantikan peran manusia sebagai lawan bicara dari manusia yang lain. (Nugraha & Sebastian, 2021).

2.2.8 *MySQL*

MySQL adalah sebuah perangkat lunak sistem manajemen basis data *SQL* atau *DBMS*. *MySQL* mendukung operasi basis data transaksional maupun non-transaksional, pada mode operasi transaksional, *MySQL* dapat dikatakan unggul untuk operasi basis data dibandingkan dengan perangkat lunak kompetitor lainnya, namun pada mode non-transaksional tidak ada jaminan atas reliabilitas terhadap data yang tersimpan, karenanya modus non-transaksional hanya cocok untuk jenis aplikasi yang tidak membutuhkan reliabilitas data seperti aplikasi blogging berbasis web, CMS, dan sejenisnya. (Yunhasnawa et al., 2023).

2.2.9 *Whitebox Testing*

White box testing disebut sebagai pengujian structural. Yang mana perangkat lunak yang diuji merupakan hal transparan kepada penguji. Dalam pengujian, uji dirancang dari perspektif pengembang dikarenakan struktur internal dikenal dengan menguji segala bagian kode yang mampu untuk diuji dengan tujuan untuk menentukan kesalahan logis dari kode sumber perangkat lunak. (Praniffa et al., 2023).

2.2.10 *Blackbox Testing*

Pengujian *black box* testing disebut sebagai pengujian perilaku. Dimana struktur interior, logika perangkat lunak yang diuji tidak diketahui oleh penguji. Penguji didasarkan kepada spesifikasi kebutuhan dan tidak perlu dilakukannya analisis kode. Pengujian *black box* testing pengujian ini dilakukan dari sudut pandang pengguna akhir. (Praniffa et al., 2023).

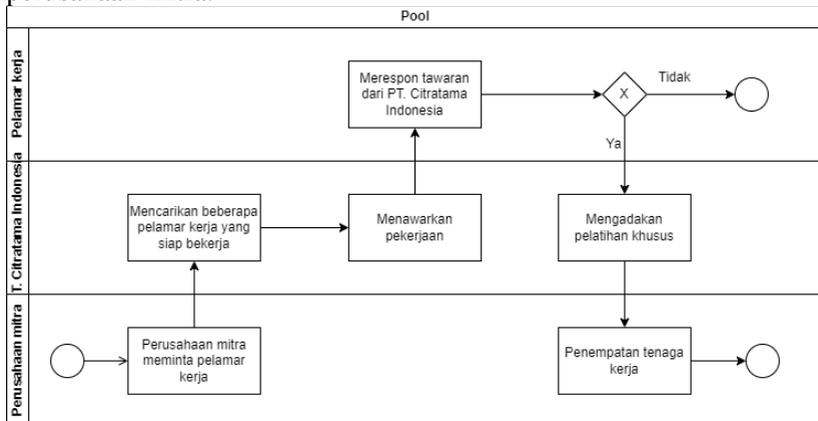
BAB III PERANCANGAN

3.1 Tahap Pengumpulan Kebutuhan dan Analisis

Pada tahap pengumpulan kebutuhan ini dilakukan dengan cara wawancara, wawancara dilakukan dengan *commisioner* dari PT. Citratama Indonesia. Hasil dari wawancara tersebut dianalisa dan mendapatkan beberapa hasil seperti proses bisnis, identifikasi pengguna, dan kebutuhan perancangan sistem.

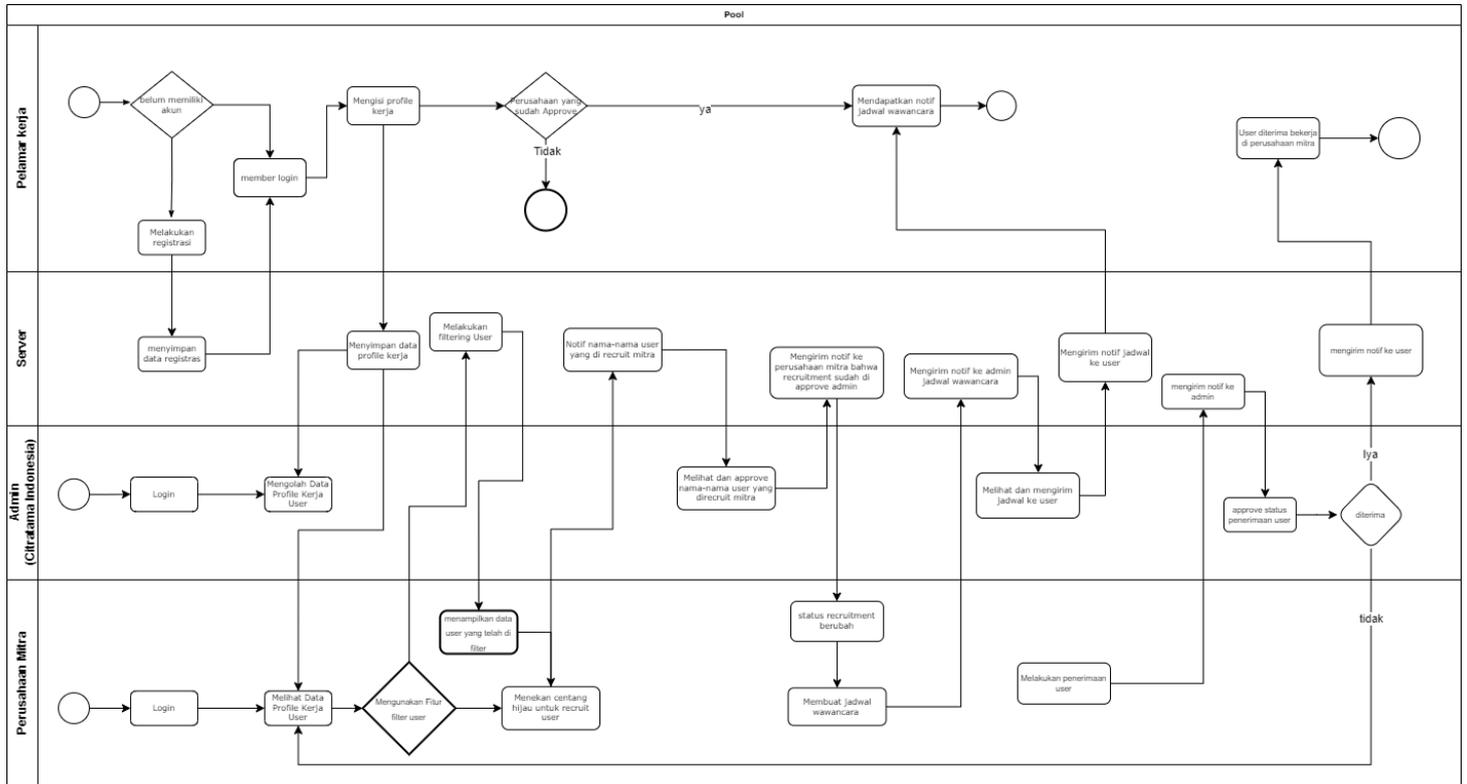
3.1.1 Proses Bisnis

Gambar 3.1 menjelaskan tentang mekanisme proses bisnis yang sedang dijalani oleh PT. Citratama Indonesia. Proses dimulai dari perusahaan mitra membutuhkan tenaga kerja yang siap bekerja, kemudian PT. Citratama Indonesia mencari pelamar kerja dan jika pelamar kerja siap untuk bekerja maka PT. Citratama Indonesia melakukan pelatihan khusus kepada pelamar kerja, setelah itu pelamar kerja yang sudah mendapatkan pelatihan khusus ditempatkan di perusahaan mitra.



Gambar 3.1 Proses Bisnis Sebelum Sistem

Proses bisnis yang di terapkan pada sistem tenaga kerja ini dapat dilihat seperti pada gambar 3.2 berikut.



Gambar 3.2 Proses Bisnis Sesudah Sistem

Proses bisnis dari sistem yang dirancang dimulai dari pelamar kerja jika belum memiliki akun pelamar kerja harus melakukan registrasi terlebih dahulu, data yang telah diregistrasi tersimpan ke database, jika sudah memiliki akun maka memasuki form login, apabila sudah berhasil melakukan login selanjutnya pelamar kerja mengisi form data diri sesuai dengan form yang sudah disediakan, data diri yang sudah diisi oleh pelamar kerja tersimpan ke database, data diri dari pelamar kerja yang sudah tersimpan tampil atau terlihat di halaman PT. Citratama Indonesia dan perusahaan mitra. Perusahaan mitra dapat melakukan login terlebih dahulu jika sudah melakukan registrasi, jika belum registrasi maka dapat melakukan registrasi, pelamar kerja dan perusahaan mitra untuk dapat melakukan registrasi. Setelah perusahaan mitra berhasil login maka masuk ke halaman *dashboard*, di *dashboard* menampilkan data pelamar kerja. Perusahaan mitra dapat melakukan filtering data-data pelamar kerja yang ingin dicari sesuai dengan kriteria perusahaan mitra, apabila dari kualifikasi pelamar kerja tersebut terpenuhi maka perusahaan mitra dapat menekan tombol centang hijau untuk melakukan *recruitment* pada pelamar kerja, setelah melakukan *recruitment* maka memberikan notifikasi ke PT. Citratama Indonesia, *recruitment* yang dilakukan perusahaan mitra akan menunggu *approve* dari PT. Citratama Indonesia, setelah PT. Citratama Indonesia melakukan *approve* dapat dilakukannya penjadwalan oleh perusahaan mitra, jika jadwal sudah dibuat maka PT. Citratama Indonesia dapat melihat jadwal dan mengirim jadwal ke pelamar kerja. Perusahaan mitra akan melakukan penerimaan untuk memilih pelamar kerja yang diterima, setelah itu memberikan notifikasi ke PT. Citratama Indonesia, jika pelamar diterima maka status akan berubah menjadi tidak tersedia kalau tidak diterima status dari pelamar kerja kembali tersedia dan ditampilkan kembali di halaman PT. Citratama Indonesia dan perusahaan mitra. Pelamar kerja juga dapat mengakses chatbot untuk bertanya sesuai dengan pertanyaan yang sudah tersedia seputar dengan pelaksanaan *recruitment* dari perusahaan mitra.

Proses bisnis yang awalnya dilakukan secara manual di PT. Citratama Indonesia telah berubah setelah adanya pembuatan sistem tenaga kerja. Perubahan ini bertujuan untuk meningkatkan efisiensi, transparansi, dan akurasi dalam perekrutan tenaga kerja. Dengan sistem baru, proses *registrasi*, pengisian data diri, dan pencarian

pelamar menjadi lebih cepat dan efisien. Akurasi data meningkat dengan penyimpanan terstruktur dalam *database*, mengurangi risiko kesalahan manusia. Transparansi dan monitoring menjadi lebih baik karena PT. Citratama Indonesia dan perusahaan mitra dapat memantau status pelamar secara *real-time*. Sistem ini memungkinkan kemudahan akses bagi pelamar dan mitra untuk *registrasi* dan pengisian data kapan saja dan di mana saja. Fitur *filtering* membantu perusahaan mitra menemukan kandidat yang tepat dengan cepat. Notifikasi dan penjadwalan memastikan semua pihak terkait selalu mendapatkan informasi terbaru tanpa perlu komunikasi manual. Fitur *chatbot* membantu pelamar kerja untuk bertanya dan mendapatkan jawaban secara cepat. Secara keseluruhan, perubahan ini meningkatkan efisiensi, pengalaman pengguna, dan memberikan keuntungan kompetitif bagi PT. Citratama Indonesia dalam manajemen tenaga kerja.

3.1.2 Identifikasi Kebutuhan Pengguna

Pada tahap ini pengembang melakukan identifikasi kebutuhan terhadap sistem yang dibangun dengan menggunakan metode wawancara. Pada tahap wawancara tersebut diketahui kebutuhan sistem yang di bangun. Adapun hasil wawancara terdapat pada **Lampiran B** dan pengumpulan kebutuhan sebagai berikut:

Berikut merupakan hasil wawancara dalam pengumpulan kebutuhan pada platform website berdasarkan aktor pelamar kerja:

- a) *User* dapat mengakses halaman *dashboard*.
- b) *User* dapat menambahkan, merubah, dan menghapus data profile kerja/CV (*Curriculum Vitae*).
- c) *User* dapat mengakses dan berinteraksi sama *chabot*.
- d) *User* dapat melihat fitur notifikasi.
- e) *User* dapat melihat halaman berita.

Berikut merupakan hasil wawancara dalam pengumpulan kebutuhan pada platform website berdasarkan aktor Perusahaan PT. Citratama Indonesia:

- a) *User* dapat mengakses halaman *dashboard*.
- b) *User* dapat mengedit dan menghapus data pelamar kerja.

- c) *User* dapat menambahkan, mengedit dan menghapus data perusahaan mitra.
- d) *User* dapat mengirim dan melihat notifikasi untuk ke perusahaan mitra dan pelamar.
- e) *User* dapat menambahkan, mengedit dan menghapus pada halaman berita.
- f) *User* dapat melakukan *monitoring* dan *approval recruitment*, jadwal, dan penerimaan.

Berikut merupakan hasil wawancara dalam pengumpulan kebutuhan pada platform website berdasarkan aktor perusahaan mitra:

- a) *User* dapat mengakses halaman *dashboard*.
- b) *User* dapat melakukan *recruitment* pelamar kerja.
- c) *User* dapat melihat fitur notifikasi dari PT. Citratama Indonesia.
- d) *User* dapat menambahkan dan mengedit jadwal.
- e) *User* dapat menambahkan, mengedit dan menghapus notifikasi untuk ke PT. Citratama Indonesia.

3.1.3 Identifikasi Aktor (*User*)

Aktor adalah seseorang yang dapat berinteraksi atau mengakses sebuah sistem. Terdapat 3 aktor pada sistem ini, yaitu *user* (Pelamar kerja), Owner (PT. Citratama Indonesia), dan client (Perusahaan Mitra). Identifikasi aktor dapat dilihat pada tabel berikut.

Tabel 3.1 Deskripsi Aktor

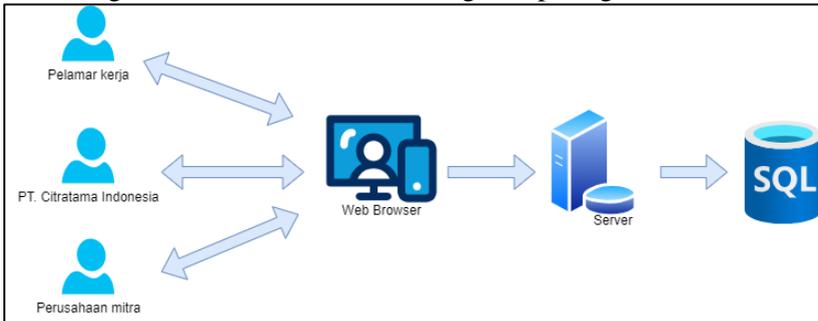
No	Aktor	Deskripsi
1.	Pelamar kerja	Aktor dengan <i>role</i> ini mempunyai wewenang untuk mengelola data diri, <i>Curriculum Vitae</i> (CV), interaksi dengan chat bot
2.	PT. Citratama Indonesia	Aktor dengan <i>role</i> ini mempunyai wewenang untuk mengelola data user, data <i>client</i> , monitoring approval, halaman berita, notifikasi untuk ke user (pelamar) dan Client (perusahaan mitra)

No	Aktor	Deskripsi
3.	Perusahaan mitra	Aktor dengan role ini mempunyai wewenang untuk <i>recruitment</i> user (pelamar kerja), notifikasi ke owner (PT. Citratama Indonesia)

3.2 Tahap Perancang Cepat

3.2.1 Arsitektur Sistem

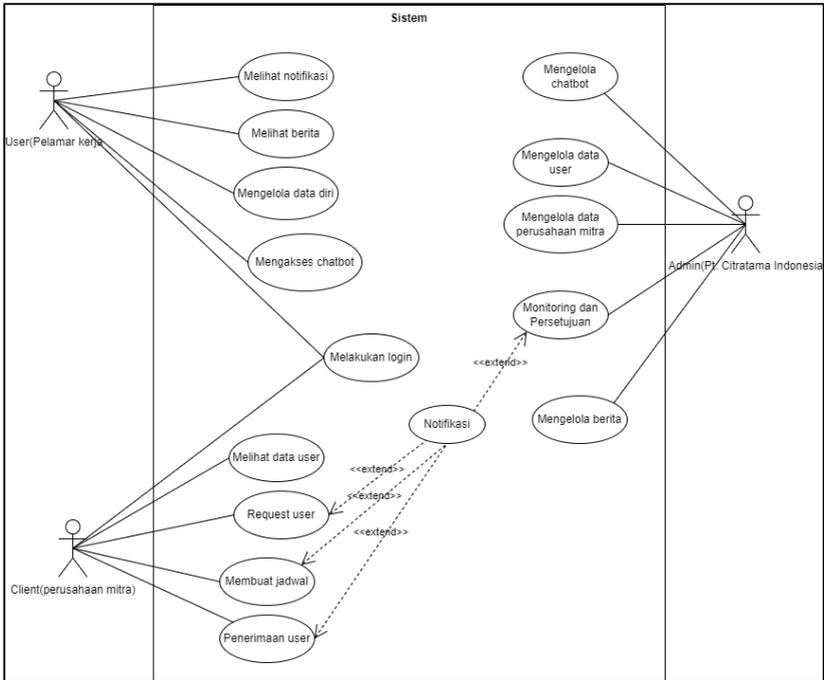
Arsitektur sistem dirancang sesuai dengan kebutuhan sistem yang berfungsi untuk mempermudah proses pengembangan sistem dan dapat mengetahui bagaimana alur atau cara kerja sistem tersebut. Perancangan arsitektur sistem ini dibangun seperti gambar berikut.



Gambar 3.3 Arsitektur Sistem

3.2.2 Use Case Diagram

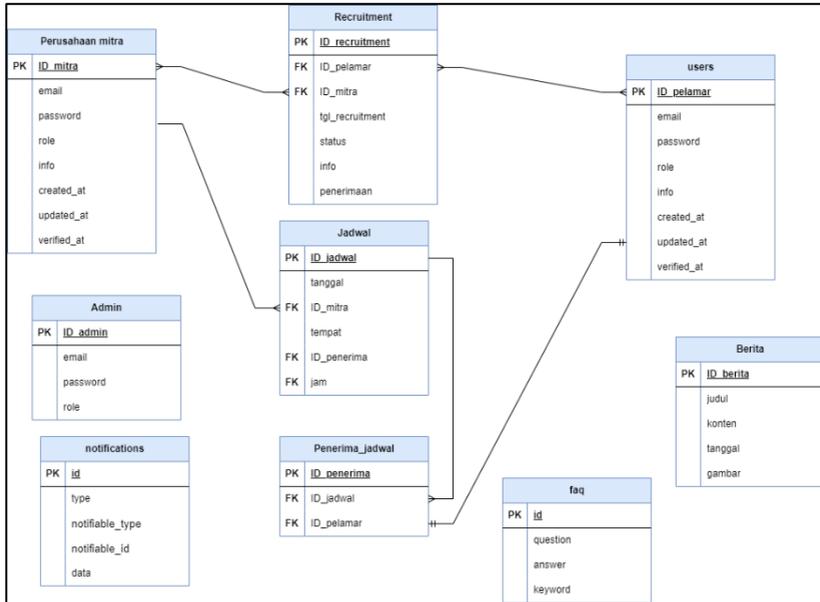
Perancangan *use case diagram* berdasarkan kebutuhan sistem sesuai dengan aktornya pada gambar berikut.



Gambar 3.4 Use Case Diagram

3.2.3 Entity Relationship Diagram

Perancangan *Entity Relationship Diagram* (ERD) dapat dilihat pada gambar berikut.



Gambar 3.5 Entity Relationship Diagram

3.2.4 Perancangan Tabel

Perancangan tabel merupakan tahapan setelah perancangan ERD dimana memberikan detail lengkap terkait struktur data yang telah direpresentasikan pada ERD di gambar 3.5

1. Tabel perusahaan mitra

Tabel 3.2 Tabel Perusahaan Mitra

Atribut	Type data	Deskripsi
ID_mitra	Int(20)	Primary key
Email	Varchar(50)	
Password	Varchar(50)	
Role	Varchar(50)	
Info	LongText	Atribut ini menyimpan beberapa data <i>JSON</i> dalam bentuk string

Atribut	Tipe data	Deskripsi
		seperti nama, nomor telepon, alamat, dll.
Created_at	Timestamp	
Updated_at	Timestamp	
Verified_at	Timestamp	

2. Tabel admin

Tabel 3.3 Tabel Admin

Atribut	Tipe data	Deskripsi
ID_admin	Int(20)	<i>Primary key</i>
Email	Varchar(50)	
Password	Varchar(50)	
Role	Varchar(50)	

3. Tabel pelamar kerja

Tabel 3.4 Tabel Pelamar Kerja

Atribut	Tipe data	Deskripsi
ID_pelamar	Int(20)	<i>Primary key</i>
Email	Varchar(50)	
Password	Varchar(50)	
Role	Varchar(50)	
Info	LongText	Atribut ini menyimpan beberapa data <i>JSON</i> dalam bentuk string seperti nama, umur, nomor telepon, alamat, tempat/tanggal lahir dan pendidikan terakhir.
Created_at	Timestamp	
Updated_at	Timestamp	
Verified_at	Timestamp	

4. Tabel *recruitment*

Tabel 3.5 Tabel *Recruitment*

Atribut	Tipe data	Deskripsi
ID_recruitment	Int(20)	<i>Primary key</i>
ID_pelamar	Int(20)	<i>Foreign key</i>
ID_mitra	Int(20)	<i>Foreign key</i>
Tgl_recruitment	Datetime	
status	Varchar(50)	Atribut ini untuk status recruitment
info	Varchar(50)	Atribut ini untuk status setelah diapprove oleh admin
penerimaan	Varchar(50)	Atribut ini status setelah diapprove yang kedua oleh admin untuk penerimaan pelamar kerja

5. Tabel jadwal

Tabel 3.6 Tabel Jadwal

Atribut	Tipe data	Deskripsi
ID_jadwal	Int(20)	<i>Primary key</i>
ID_penerima	Int(20)	<i>Foreign key</i>
ID_mitra	Int(20)	<i>Foreign key</i>
tanggal	Datetime	
tempat	Varchar(50)	
jam	Varchar(255)	

6. Tabel penerima jadwal

Tabel 3.7 Tabel Penerima Jadwal

Atribut	Tipe data	Deskripsi
ID_penerima	Int(20)	<i>Primary key</i>

Atribut	Tipe data	Deskripsi
ID_jadwal	Int(20)	<i>Foreign key</i>
ID_pelamar	Int(20)	<i>Foreign key</i>

7. Tabel berita

Tabel 3.8 Tabel Berita

Atribut	Tipe data	Deskripsi
ID_berita	Int(20)	<i>Primary key</i>
Judul	Varchar(50)	
konten	Varchar(50)	
gambar	Varchar(50)	
tanggal	datetime	

8. Tabel faq

Tabel ini dirancang untuk menyimpan data dari kata kunci dan jawaban yang digunakan untuk chatbot pada sistem ini.

Tabel 3.9 Tabel Faq

Atribut	Tipe data	Deskripsi
Id	Int(20)	<i>Primary key</i>
Question	Varchar(255)	
Answer	Text	
Keyword	Varchar(255)	

9. Tabel *notifications*

Tabel 3.10 Tabel *Notifications*

Atribut	Tipe data	Deskripsi
Id	Int(20)	<i>Primary key</i>
Type	Varchar(255)	
Notifiable_type	Varchar(255)	
Notifiable_id	Bigint(20)	
data	text	

3.2.5 Keamanan aplikasi

Dalam perancangan keamanan aplikasi *backend REST API* untuk PT. Citratama Indonesia, penggunaan *middleware* di *laravel*

merupakan pendekatan utama. *Middleware auth* berfungsi untuk memastikan bahwa setiap permintaan yang masuk telah diautentikasi dengan benar, sehingga hanya pengguna yang valid yang dapat mengakses sistem. Untuk mengelola hak akses dan memastikan bahwa pengguna hanya dapat mengakses data atau fitur yang sesuai dengan role dari setiap *user*, *middleware* tambahan seperti *can*, *role*, atau *permission* diterapkan.

3.3 Metode Pengujian

3.3.1 Whitebox Testing

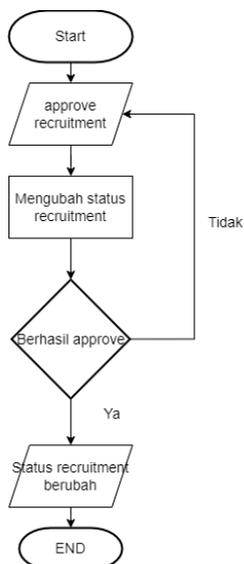
Pengujian ini menguji alur dari program yang dapat dilihat dari penjelasan berikut.

3.3.1.1 PT. Citratama Indonesia

Pengujian sistem untuk PT. Citratama Indonesia dilakukan melalui beberapa tahapan:

1. Pembuatan *flowchart*

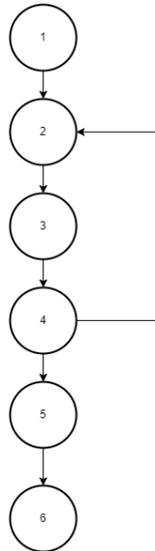
Pada tahapan ini dilakukan pembuatan *flowchart* untuk menggambarkan alur proses sistem bagian PT. Citratama Indonesia.



Gambar 3.6 *Flowchart* PT. Citratama Indonesia

2. Pembuatan *flowgraph*

Pada tahapan selanjutnya dilakukan pembuatan *flowgraph* untuk merepresentasikan alur kontrol program berdasarkan *flowchart* yang telah dibuat.



Gambar 3.7 *Flowgraph* PT. Citratama Indonesia

3. Penentuan jalur

Terdapat 2 jalur yang didapat berdasarkan *flowgraph* yang telah dibuat.

Jalur 1: 1-2-3-4-5-6

Jalur 2: 1-2-3-4-2-3-4-5-6

4. *Case* PT. Citratama Indonesia

Berikut merupakan *case* dari PT. Citratama Indonesia yang dapat dilihat dari tabel 3.11 berikut.

Tabel 3.11 *Case* PT. Citratama Indonesia

Path	1
Jalur	1-2-3-4-5-6
Skenario	1. <i>Start</i> 2. Approve <i>recruitment</i> 3. Mengubah <i>status recruitment</i> 4. Berhasil approve

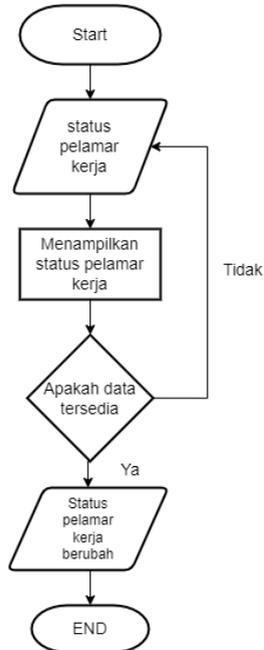
	5. Status recruitment berubah 6. End
Hasil Pengujian	
Path	2
Jalur	1-2-3-4-2-3-4-5-6
Skenario	1. <i>Start</i> 2. <i>Approve recruitment</i> 3. Mengubah status <i>recruitment</i> 4. <i>Gagal approve</i> 5. <i>Approve recruitment</i> 6. Mengubah status <i>recruitment</i> 7. <i>Berhasil approve</i> 8. Status <i>recruitment</i> berubah 9. <i>End</i>
Hasil Pengujian	

3.3.1.2 Perusahaan Mitra

Pengujian sistem untuk PT. Citratama Indonesia dilakukan melalui beberapa tahapan:

1. Pembuatan *flowchart*

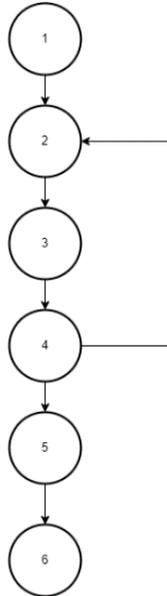
Pada tahapan ini dilakukan pembuatan *flowchart* untuk menggambarkan alur proses sistem bagian perusahaan mitra.



Gambar 3.8 *Flowchart* Perusahaan Mitra

2. Pembuatan *flowgraph*

Pada tahapan selanjutnya dilakukan pembuatan *flowgraph* untuk merepresentasikan alur kontrol program berdasarkan *flowchart* yang telah dibuat.



Gambar 3.9 *Flowgraph* Perusahaan Mitra

3. Penentuan jalur

Terdapat 2 jalur yang didapat berdasarkan *flowgraph* yang telah dibuat.

Jalur 1: 1-2-3-4-5-6

Jalur 2: 1-2-3-4-2-3-4-5-6

4. *Case* perusahaan mitra

Berikut merupakan *case* dari perusahaan mitra yang dapat dilihat dari tabel 3.12 berikut.

Tabel 3.12 *Case* Perusahaan Mitra

Path	1
Jalur	1-2-3-4-5-6
Skenario	1. <i>Start</i> 2. Status pelamar kerja 3. Menampilkan status pelamar kerja 4. Data tersedia

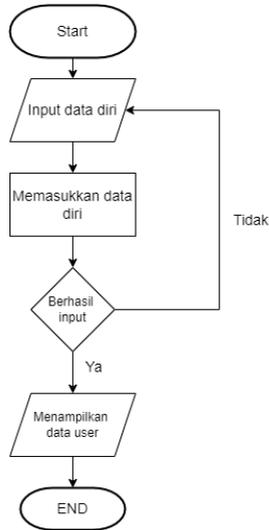
	5. Status pelamar kerja berubah 6. <i>End</i>
Hasil Pengujian	
Path	2
Jalur	1-2-3-4-2-3-4-5-6
Skenario	1. <i>Start</i> 2. Status pelamar kerja 3. Menampilkan status pelamar kerja 4. Data tidak tersedia 5. Status pelamar kerja 6. Menampilkan status pelamar kerja 7. Data tersedia 8. Status pelamar kerja berubah 9. <i>End</i>
Hasil Pengujian	

3.3.1.3 Pelamar kerja

Pengujian sistem untuk pelamar kerja dilakukan melalui beberapa tahapan:

1. Pembuatan *flowchart*

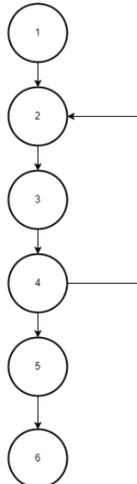
Pada tahapan ini dilakukan pembuatan *flowchart* untuk menggambarkan alur proses sistem bagian pelamar kerja.



Gambar 3.10 *Flowchart* Pelamar Kerja

2. Pembuatan *flowgraph*

Pada tahapan selanjutnya dilakukan pembuatan *flowgraph* untuk merepresentasikan alur kontrol program berdasarkan *flowchart* yang telah dibuat.



Gambar 3.11 *Flowgraph* Pelamar Kerja

3. Penentuan jalur
Terdapat 2 jalur yang didapat berdasarkan *flowgraph* yang telah dibuat.
Jalur 1: 1-2-3-4-5-6
Jalur 2: 1-2-3-4-2-3-4-5-6
4. *Case* pelamar kerja
Berikut merupakan *case* dari perusahaan mitra yang dapat dilihat dari tabel 3.13 berikut.

Tabel 3.13 *Case* Pelamar Kerja

Path	1
Jalur	1-2-3-4-5-6
Skenario	<ol style="list-style-type: none"> 1. <i>Start</i> 2. Input data diri 3. Memasukkan data diri 4. Berhasil input 5. Menampilkan data user 6. <i>End</i>
Hasil Pengujian	
Path	2
Jalur	1-2-3-4-2-3-4-5-6
Skenario	<ol style="list-style-type: none"> 1. <i>Start</i> 2. Input data diri 3. Memasukkan data diri 4. Gagal input 5. Input data diri 6. Memasukkan data diri 7. Berhasil input 8. Menampilkan data user 9. <i>End</i>
Hasil Pengujian	

3.3.2 Blackbox Testing

Pengujian sistem merupakan tahapan pengujian software atau perangkat lunak untuk mengevaluasi sebuah sistem apakah sudah berjalan dengan baik atau belum, pengujian *blackbox* menguji fungsionalitas dari sistem yang telah dirancang. Berikut merupakan tabel perancangan pengujian dengan menggunakan *blackbox*.

Tabel 3.14 Pengujian *Blackbox*

Kode	Use Case		Tanggal Pengujian	Hasil Pengujian		Catatan Pengujian
	Nama Uji			[] Berhasil	[] Gagal	
1	Nama Uji	Login		[] Berhasil	[] Gagal	
	Kasus Uji	Email: riandy@gmail.com Password: 12345678				
	Hasil Yang Diharapkan	Jika berhasil maka akan mengenerate token Jika gagal maka akan menampilkan pesan gagal				
2	Nama Uji	Lihat data pelamar kerja		[] Berhasil	[] Gagal	
	Kasus Uji	-				
	Hasil Yang Diharapkan	Dapat menampilkan detail informasi dari pelamar kerja				
3	Nama Uji	Tambah data diri		[]	[]	

Kode	Use Case		Tanggal Pengujian	Hasil Pengujian		Catatan Pengujian
				Berhasil	Gagal	
	Kasus Uji	Mengisi form data diri		Berhasil	Gagal	
	Hasil Yang Diharapkan	Dapat menambahkan data diri				
		Jika berhasil maka akan menampilkan pesan berhasil				
		Jika gagal maka akan menampilkan pesan gagal				
4	Nama Uji	Edit data diri		[] Berhasil	[] Gagal	
	Kasus Uji	Mengisi form data diri				
	Hasil Yang Diharapkan	Dapat mengubah data diri				
		Jika berhasil maka akan menampilkan pesan berhasil				
	Jika gagal maka akan menampilkan pesan gagal.					
5	Nama Uji	Chatbot		[] Berhasil	[] Gagal	
	Kasus Uji	-				

Kode	<i>Use Case</i>		Tanggal Pengujian	Hasil Pengujian		Catatan Pengujian
	Hasil Yang Diharapkan	Dapat memberikan pertanyaan dan bot memberikan jawaban				
6	Nama Uji Kasus Uji	Lihat notifikasi -		[] Berhasil	[] Gagal	
	Hasil Yang Diharapkan	Dapat menampilkan detail informasi dari notifikasi				

BAB IV PENGUJIAN DAN ANALISIS

4.1 Implementasi *REST API*

Teknologi yang digunakan dalam membangun *API* ini menggunakan *laravel*. Proses implementasi *REST API* dengan menggunakan *laravel*. Pertama, membuat *model-model* yang diperlukan untuk masing-masing basis data, seperti *MySQL* untuk data umum aplikasi. Autentikasi diimplementasikan menggunakan *bearer token* selama pembuatan *API* ini, sehingga pengguna yang telah diotentikasi hanya dapat mengakses layanan yang tersedia. Setelah menyelesaikan konfigurasi dan implementasi dasar, membangun *endpoint-endpoint REST API* menggunakan fitur-fitur *laravel* itu sendiri, seperti *routing*, *middleware*, dan validasi permintaan. Setiap *endpoint* dirancang untuk memenuhi kebutuhan spesifik dari aplikasi, dengan operasi (*create*, *update*, *delete*) standar dan operasi khusus lainnya yang diperlukan. Menggunakan basis data *MySQL* dalam membangun aplikasi ini memberikan fleksibilitas dalam manajemen pengguna untuk memastikan keamanan dan pengelolaan yang optimal. Selain itu, setiap permintaan yang berhasil akan mendapatkan *response code* 200 OK, menandakan bahwa *response code* berhasil dan permintaan diproses dengan baik.

Table 4.1 Hasil implementasi *API*

No	Kandidat <i>API</i>		Proses	<i>Response Code</i>	<i>Message Body</i>
	Method	URI			
1	POST	/login	Melakukan login dengan autentikasi	200 OK	Berhasil login
2	POST	/regiter	Melakukan registrasi	200 OK	Berhasil registasi
3	POST	/logout	Melakukan logout	200 OK	Berhasil logout
4	POST	/datadiri	Menambahkan data diri baru	200 OK	Berhasil menambahkan data

No	Kandidat API		Proses	Response Code	Message Body
	Method	URI			
5	GET	/show-profile	Menampilkan data diri pelamar kerja	200 OK	Data berhasil ditemukan
6	POST	/update-profile/{id}	Mengedit data berdasarkan id pelamar kerja	200 OK	Berhasil update data
7	POST	/admin/tambah-pelamar	Menambahkan data pelamar kerja	200 OK	Berhasil menambahkan data
8	POST	/admin/update-pelamar	Mengedit data pelamar kerja	200 OK	Berhasil melakukan update data
9	GET	/admin/pelamar	Menampilkan daftar semua pelamar kerja	200 OK	Data Berhasil Ditemukan
10	DELETE	/admin/pelamar/{id}	Menghapus pelamar kerja berdasarkan id	200 OK	Berhasil melakukan delete data
11	POST	/admin/tambah-mitra	Menambahkan data perusahaan mitra	200 OK	Berhasil Menambahkan Data
12	POST	/admin/update-mitra	Mengedit data perusahaan mitra	200 OK	Berhasil Melakukan Update Data

No	Kandidat API		Proses	Response Code	Message Body
	Method	URI			
13	GET	/admin/mitra	Menampilkan data perusahaan mitra	200 OK	Data berhasil ditemukan
14	DELETE	/admin/mitra/{id}	Menghapus perusahaan mitra berdasarkan id	200 OK	Berhasil melakukan update
15	POST	/admin/tambah-berita	Menambahkan data berita	200 OK	Berhasil menambahkan data
16	POST	/admin/update-berita/{id}	Mengedit data berita berdasarkan id berita	200 OK	Berhasil melakukan update data
17	GET	/admin/berita	Menampilkan data berita	200 OK	Data berhasil ditemukan
18	<i>DELETE</i>	/admin/berita/{id}	Menghapus berita berdasarkan id berita	200 OK	Berhasil Melakukan <i>Delete</i> Data
19	POST	/admin/approve/{recruitment}	Melakukan approve berdasarkan id recruitment	200 OK	Berhasil melakukan apporve
20	POST	/admin/sendWA/{jadwal}	Melakukan kirim pesan berdasarkan id jadwal	200 OK	Berhasil melakukan kirim pesan
21	GET	/userTersedia	Menampilkan data pelamar	200 OK	Data berhasil ditemukan

No	Kandidat API		Proses	Response Code	Message Body
	Method	URI			
			kerja yang tersedia		
22	GET	/userStatus	Menampilkan semua pelamar kerja yang direcrutment	200 OK	Data berhasil ditemukan
23	GET	/recruitments	Menampilkan recruitmen t pelamar kerja berdasarkan perusahaan mitra	200 OK	Data berhasil ditemukan
24	POST	/recruit	Menambahkan data recruitmen t	200 OK	Berhasil menambahkan data
25	POST	/recruit/confirm/{id}	Melakukan approve recruitmen t	200 OK	Berhasil melakukan approve
26	POST	/jadwal/add	Menambahkan data jadwal	200 OK	Berhasil menambahkan data
27	GET	/jadwal	Menampilkan semua data jadwal	200 OK	Data berhasil ditemukan
28	POST	/jadwal/update/{id}	Mengedit data jadwal berdasarkan data jadwal	200 OK	Berhasil melakukan update data
29	DELETE	/jadwal/delete/{id}	Menghapus data jadwal	200 OK	Berhasil melakukan

No	Kandidat <i>API</i>		Proses	<i>Response Code</i>	<i>Message Body</i>
	Method	URI			
			berdasarkan id jadwal		n delete data
30	GET	/faq	Menampilkan data faq	200 OK	Data berhasil ditemukan
31	POST	/faq/add	Menambahkan data faq	200 OK	Berhasil menambahkan data
32	POST	/faq/update/{id}	Mengedit data faq berdasarkan id faq	200 OK	Berhasil melakukan update data
33	DELETE	/faq/delete/{id}	Menghapus data faq berdasarkan id faq	200 OK	Berhasil melakukan delete data

4.2 Implementasi dokumen *API*

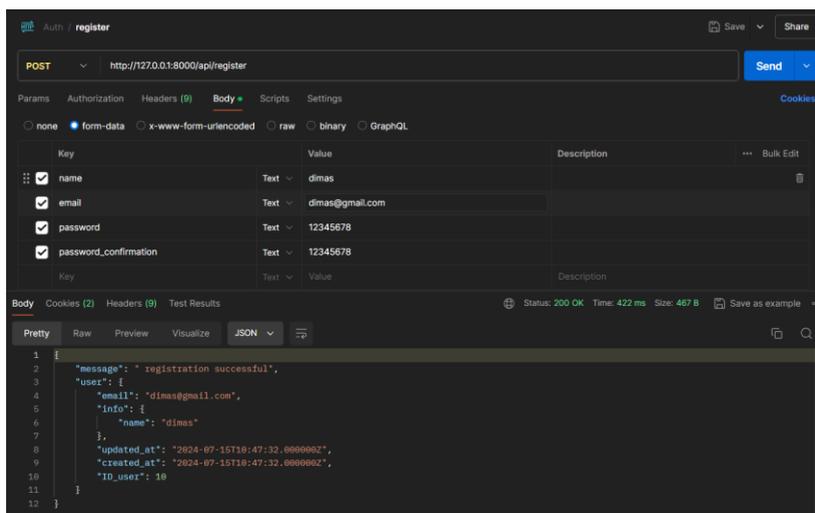
Dokumentasi pengujian *API* penting untuk memastikan fungsionalitas yang diharapkan dari setiap *endpoint*. Prosesnya mencakup pengujian dengan input yang bervariasi dan respon yang dihasilkan. Meskipun setiap *API* memiliki *endpoint* dan parameter yang berbeda, prinsip dasarnya serupa. Dokumentasi ini dibuat pada *API* yang sudah dibuat dengan jenis permintaan (seperti *GET*, *POST*, dan *DELETE*). Dengan demikian, dokumentasi ini dapat diterapkan secara efisien pada berbagai jenis *API*, seperti perusahaan mitra, admin, berita, dan jadwal, karena proses pengujian dasarnya hampir sama untuk semua *API* tersebut.

Keamanan aplikasi *backend REST API* untuk PT. Citratama Indonesia menggunakan *middleware*. *Middleware auth* memastikan *otentikasi* yang tepat, sedangkan *middleware can, role*, atau *permission* mengelola hak akses pengguna. Penggunaan *middleware throttle* membatasi jumlah permintaan ke *API* untuk mencegah serangan *DDoS*. *HTTPS* diterapkan untuk mengamankan transmisi data.

4.2.1 Dokumentasi *API authentication*

4.2.1.1 *API* registrasi

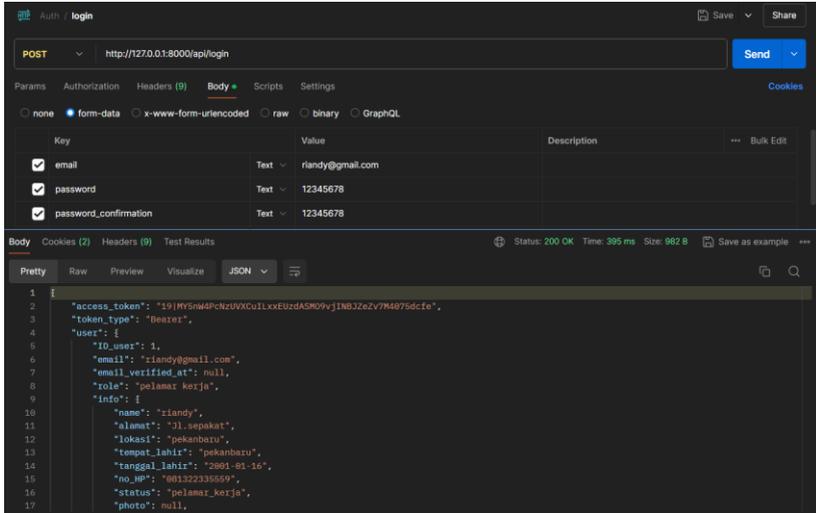
Pada *API* ini digunakan untuk *user* dalam melakukan proses *registrasi*. Dalam *API* ini menerima request dengan method *POST* dari form-data dan memberikan response sesuai dengan gambar 4.1 sehingga *user* mendapatkan akses untuk melakukan login menggunakan akun yang telah didaftarkan.



Gambar 4.1 Implementasi *API* Registrasi

4.2.1.2 *API* Login

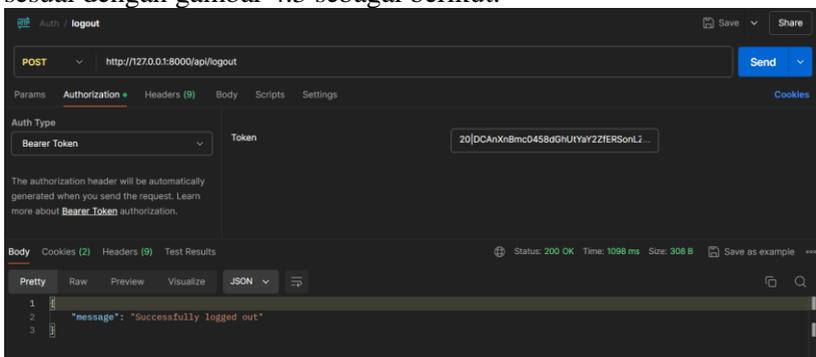
Pada *API* ini digunakan untuk *user* dalam melakukan proses login. Dalam *API* ini menerima *request* dengan *method* *POST* dari form-data dan memberikan *response* sesuai dengan gambar 4.2 sehingga *user* dapat melakukan login ke dalam sistem dan memberikan token untuk hak akses dalam penggunaan fitur-fitur yang ada di dalam sistem.



Gambar 4.2 Implementasi API Login

4.2.1.3 API Logout

Pada API ini digunakan untuk *user* dalam melakukan proses logout. Dalam API ini menerima *request* dengan *method* POST dengan *type* *bearer token* yang didapatkan saat melakukan *login*. API ini memastikan bahwa token akses tidak lagi valid sehingga tidak bisa digunakan lagi untuk mengakses sistem dan mendapatkan *response* sesuai dengan gambar 4.3 sebagai berikut.

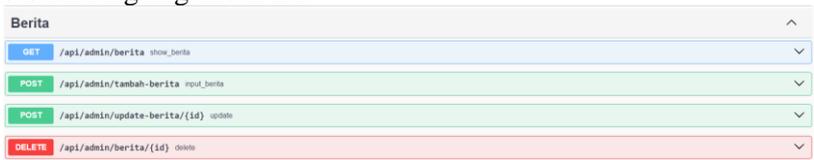


Gambar 4.3 Implementasi API Logout

4.2.2 Dokumentasi API admin

4.2.2.1 API berita

API ini digunakan oleh admin untuk mengelola data berita. Pada API ini dapat melakukan beberapa jenis permintaan seperti *GET*, *POST*, dan *DELETE* sehingga sistem nantinya dapat melakukan fungsionalitas lengkap untuk operasi CRUD (*Create, Read, Update, Delete*) pada data berita. Berikut merupakan implementasi API berita sesuai dengan gambar 4.4.

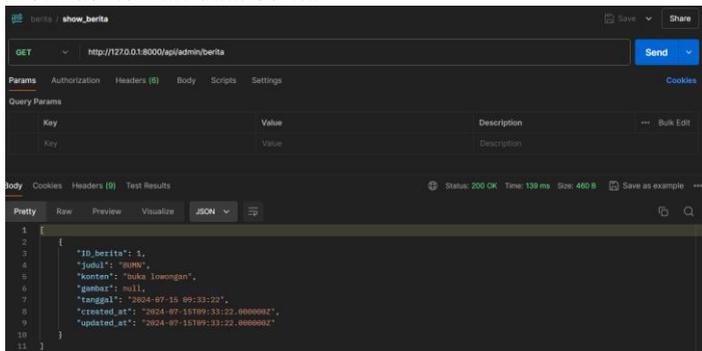


Berita	
GET	/api/admin/berita show_berita
POST	/api/admin/tambah-berita input_berita
POST	/api/admin/update-berita/{id} update
DELETE	/api/admin/berita/{id} delete

Gambar 4.4 Implementasi API Berita

a) API mengambil data berita

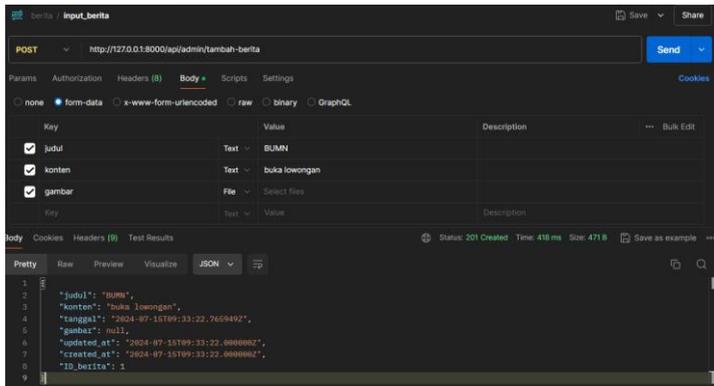
Pada API ini digunakan untuk *user* dalam melakukan proses pengambilan data berita yang tersedia. Dalam API ini menerima *request* dengan *method GET* dan memberikan *response* sesuai dengan gambar 4.5 yaitu mengambil informasi terkait data berita.



Gambar 4.5 API Mengambil Data Berita

b) API menambahkan berita

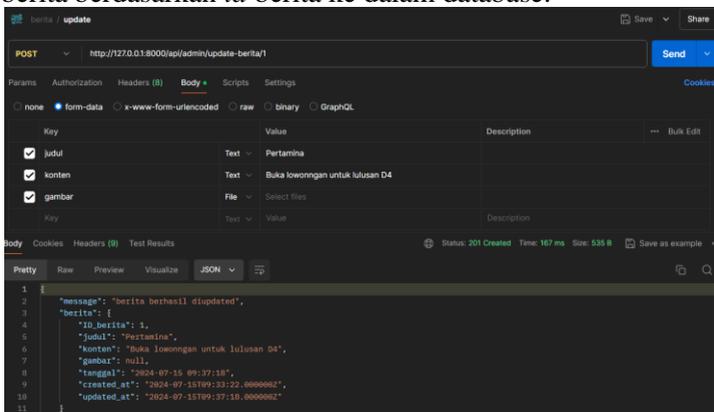
Pada API ini digunakan untuk *user* dalam melakukan proses penambahan data berita. Dalam API ini menerima *request* dengan *method POST* dari form-data dan memberikan *response* sesuai dengan gambar 4.6 yaitu menambahkan ke dalam database.



Gambar 4.6 API Menambah Berita

c) API mengubah data berita

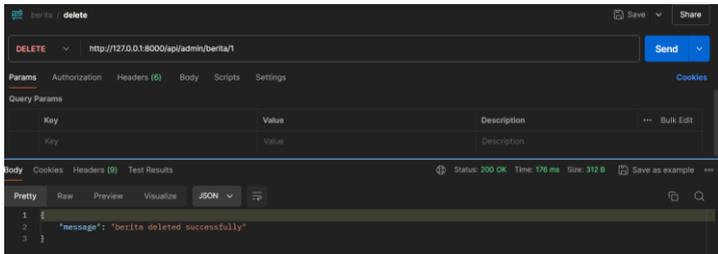
Pada API ini digunakan untuk *user* dalam melakukan proses perubahan pada data berita. Dalam API ini menerima *request* dengan *method POST* dari form-data dan memberikan *response* sesuai dengan gambar 4.7 yaitu mengubah data berita berdasarkan *id* berita ke dalam database.



Gambar 4.7 API Mengubah Data Berita

d) *API* menghapus data berita

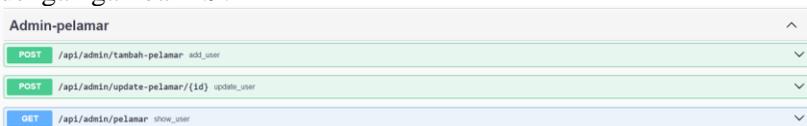
Pada *API* ini digunakan untuk *user* dalam melakukan proses penghapusan pada data berita berdasarkan *id* berita. Dalam *API* ini menerima *request* dengan *method DELETE* yang akan menghapus data dari server dan memberikan *response* sesuai dengan gambar 4.8 yaitu menghapus data berita berdasarkan *id* berita di database.



Gambar 4.8 *API* Menghapus Data Berita

4.2.2.2 *API* pelamar kerja

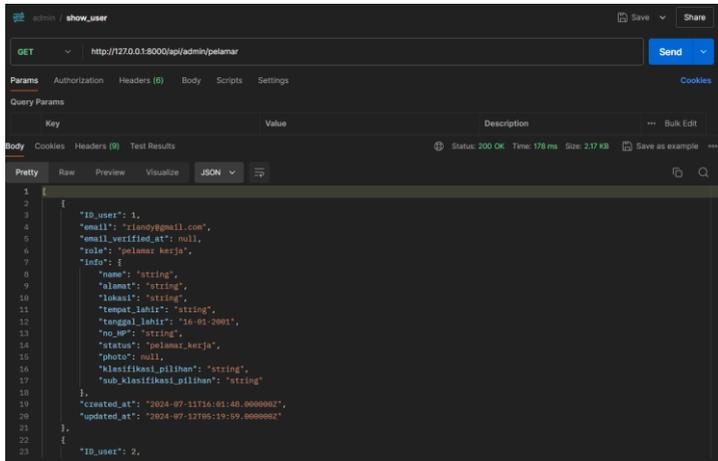
API ini digunakan oleh admin untuk mengelola data pelamar kerja. Pada *API* ini dapat melakukan beberapa jenis permintaan seperti *GET*, *POST* sehingga sistem nantinya dapat melakukan fungsionalitas lengkap untuk operasi CRU (*Create, Read, Update*) pada data pelamar kerja. Berikut merupakan implementasi *API* pelamar kerja sesuai dengan gambar 4.9.



Gambar 4.9 Implementasi *API* Pelamar Kerja

a) *API* mengambil data pelamar kerja

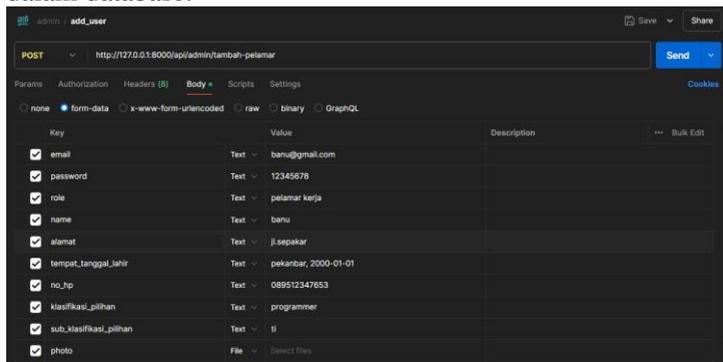
Pada *API* ini digunakan untuk *user* dalam melakukan proses pengambilan data pelamar kerja yang tersedia. Dalam *API* ini menerima *request* dengan *method GET* dan memberikan *response* sesuai dengan gambar 4.10 yaitu mengambil informasi terkait data pelamar kerja.



Gambar 4.10 API Mengambil Data Pelamar Kerja

b) API menambahkan pelamar kerja

Pada API ini digunakan untuk *user* dalam melakukan proses penambahan data pelamar kerja. Dalam API ini menerima *request* dengan *method POST* dari form-data dan memberikan *response* sesuai dengan gambar 4.11 yaitu menambahkan ke dalam database.



```

1 {
2   "message": "data berhasil ditambahkan",
3   "user": {
4     "email": "banu@gmail.com",
5     "info": {
6       "name": "banu",
7       "alamat": "j.l.depakas",
8       "tempat_lahir": null,
9       "tanggal_lahir": null,
10      "no_hp": "089512345678",
11      "photo": null,
12      "klasifikasi_pilihan": "programmer",
13      "sub_klasifikasi_pilihan": "ti"
14    }
15  },
16  "updated_at": "2024-07-15T09:43:47.000000Z",
17  "created_at": "2024-07-15T09:43:47.000000Z",
18  "ID_user": 8
19 }

```

Gambar 4.11 API Menambah Data Pelamar Kerja

c) API mengubah data pelamar kerja

Pada API ini digunakan untuk *user* dalam melakukan proses perubahan pada data pelamar kerja. Dalam API ini menerima *request* dengan *method POST* dari *form-data* dan memberikan *response* sesuai dengan gambar 4.12 yaitu mengubah data pelamar kerja berdasarkan *id* pelamar kerja ke dalam database.

POST http://127.0.0.1:8000/api/admin/update-pelamar/8

Key	Value	Description
email	banu@gmail.com	
name	banu	
alamat	j.sepakar	
tempat_lahir	pekanbaru	
tanggal_lahir	2000-01-02	
no_hp	081322335559	
klasifikasi_pilihan	programmer	
sub_klasifikasi_pilihan	ti	
photo	Select files	

```

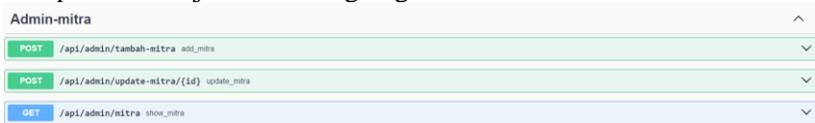
1 {
2   "message": "data user berhasil diperbarui",
3   "user": {
4     "ID_user": 8,
5     "email": "banu@gmail.com",
6     "email_verified_at": null,
7     "role": "pelamar kerja",
8     "info": {
9       "name": "banu",
10      "alamat": "j.l.sepakar",
11      "tempat_lahir": "pekanbaru",
12      "tanggal_lahir": "2000-01-02",
13      "no_hp": "089512345678",
14      "photo": null,
15      "klasifikasi_pilihan": "programmer",
16      "sub_klasifikasi_pilihan": "ti",
17      "no_hp": "081322335559"
18    }
19  },
20  "created_at": "2024-07-15T09:43:47.000000Z",
21  "updated_at": "2024-07-15T09:47:15.000000Z"
22 }

```

Gambar 4.12 API Mengubah Data Pelamar Kerja

4.2.2.3 API perusahaan mitra

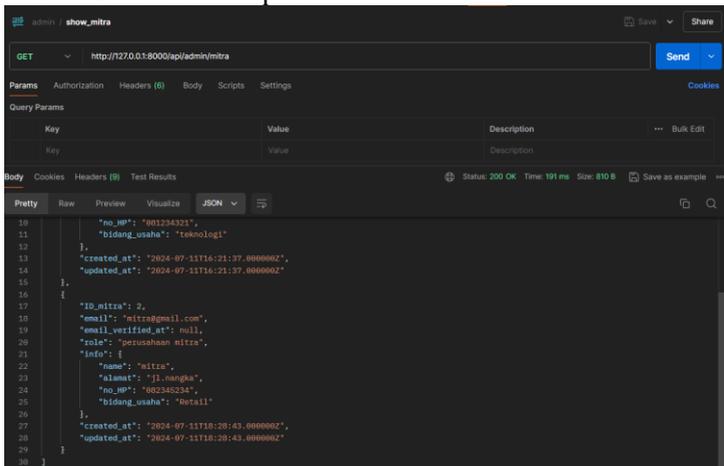
API ini digunakan oleh admin untuk mengelola data perusahaan mitra. Pada API ini dapat melakukan beberapa jenis permintaan seperti *GET*, *POST* sehingga sistem nantinya dapat melakukan fungsionalitas lengkap untuk operasi CRU (*Create, Read, Update*) pada data pelamar kerja. Berikut merupakan implementasi API pelamar kerja sesuai dengan gambar 4.13.



Gambar 4.13 Implementasi API Perusahaan Mitra

a) API mengambil data perusahaan mitra

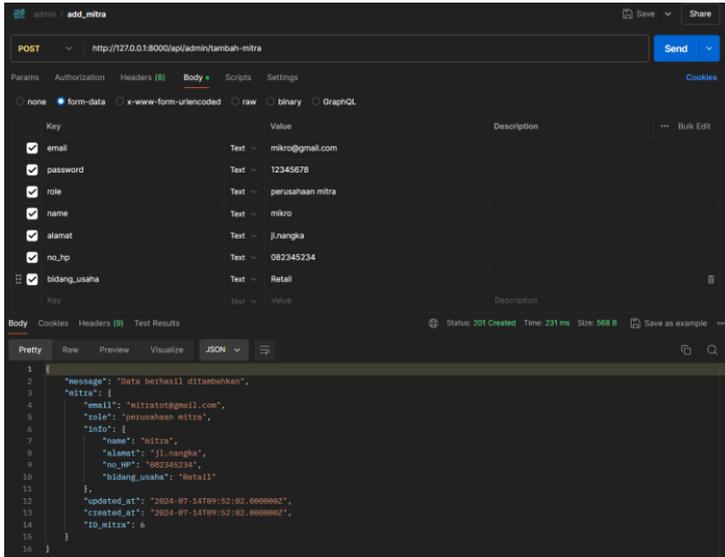
Pada API ini digunakan untuk *user* dalam melakukan proses pengambilan data perusahaan mitra yang tersedia. Dalam API ini menerima *request* dengan *method GET* dan memberikan *response* sesuai dengan gambar 4.14 yaitu mengambil informasi terkait data perusahaan mitra.



Gambar 4.14 API Mengambil Data Perusahaan Mitra

b) API menambahkan perusahaan mitra

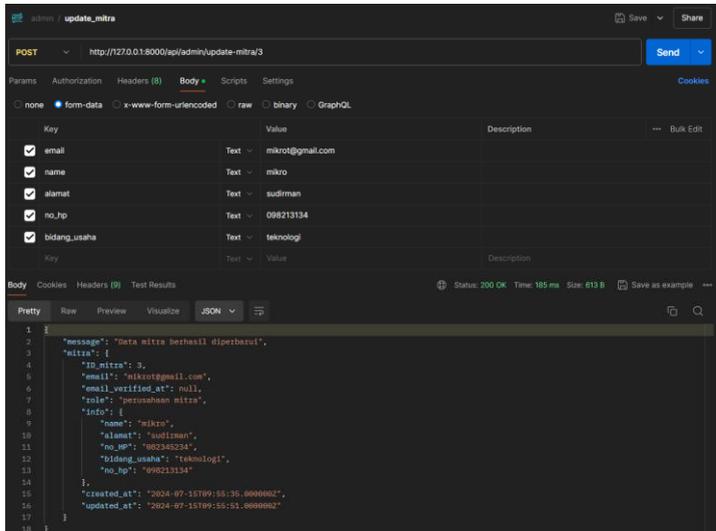
Pada API ini digunakan untuk *user* dalam melakukan proses penambahan data perusahaan mitra. Dalam API ini menerima *request* dengan *method POST* dari *form-data* dan memberikan *response* sesuai dengan gambar 4.15 yaitu menambahkan ke dalam database.



Gambar 4.15 API Menambah Perusahaan Mitra

c) API mengubah data perusahaan mitra

Pada API ini digunakan untuk *user* dalam melakukan proses perubahan pada data perusahaan mitra. Dalam API ini menerima *request* dengan *method POST* dari *form-data* dan memberikan *response* sesuai dengan gambar 4.16 yaitu mengubah data pelamar kerja berdasarkan id perusahaan mitra ke dalam database.



Gambar 4.16 API Mengubah Data Perusahaan Mitra

4.2.2.4 API approve admin

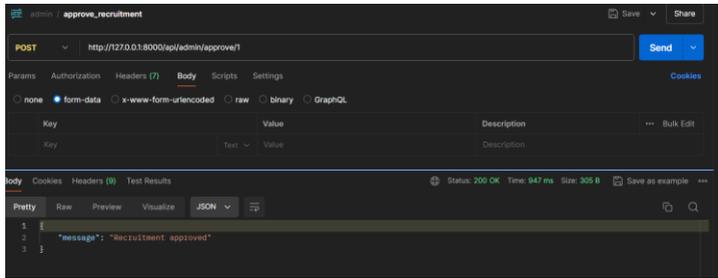
API ini dibuat untuk admin melakukan *approve recruitment*, pada API ini melakukan jenis permintaan dengan method *POST*.



Gambar 4.17 Implementasi API Approve Recruitment Admin

a) API approve recruitment admin

Pada API ini digunakan untuk *user* dalam melakukan proses *approve recruitment*. Dalam API ini menerima *request* dengan method *POST* dari form-data dan memberikan *response* sesuai dengan gambar 4.18.



Gambar 4.18 API Approve Recruitment Admin

4.2.2.5 API mengirim pesan

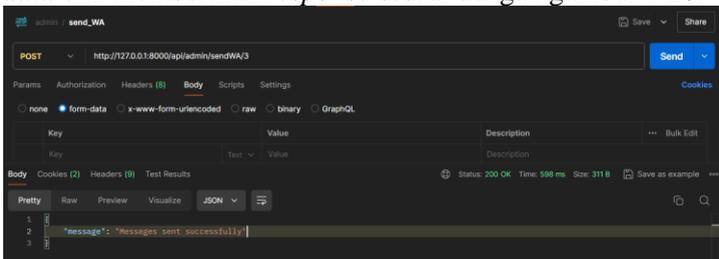
API ini dibuat untuk admin melakukan kirim pesan, pada API ini melakukan jenis permintaan dengan method *POST*.



Gambar 4.19 Implementasi API Kirim Pesan

a) API mengirim pesan

Pada API ini digunakan untuk *user* dalam melakukan proses pengiriman pesan mengenai jadwal dengan *WhatsApp*. Dalam API ini menerima *request* dengan *method POST* dari *form-data* dan memberikan *response* sesuai dengan gambar 4.20.



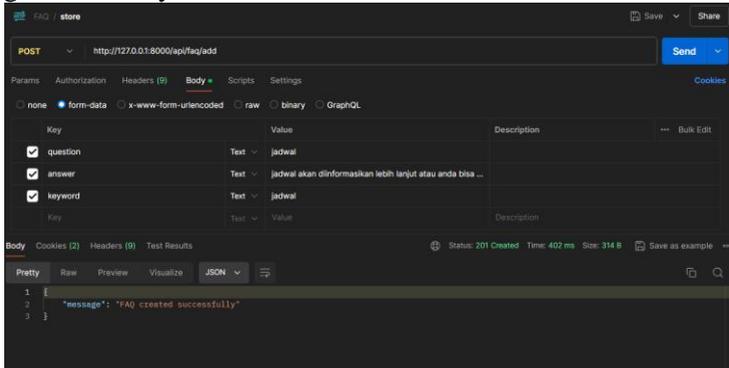
Gambar 4.20 API Mengirim Pesan

4.2.2.6 API chatbot

API ini dibuat untuk pelamar kerja dapat mengelola data diri, pada API ini melakukan beberapa jenis permintaan seperti *GET* dan *POST*.

a) API menambahkan pertanyaan

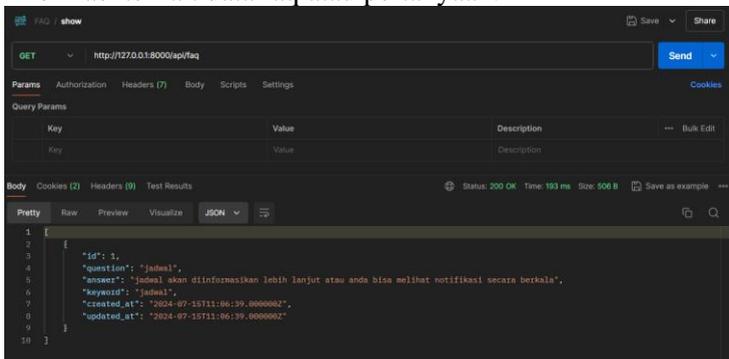
Pada *API* ini digunakan untuk *user* dalam melakukan proses penambahan data pertanyaan sehingga dapat dikenali oleh bot. Dalam *API* ini menerima *request* dengan *method POST* dari form-data dan memberikan *response* sesuai dengan gambar 4.21 yaitu menambahkan ke dalam database.



Gambar 4.21 *API* Tambah Pertanyaan

b) *API* menampilkan

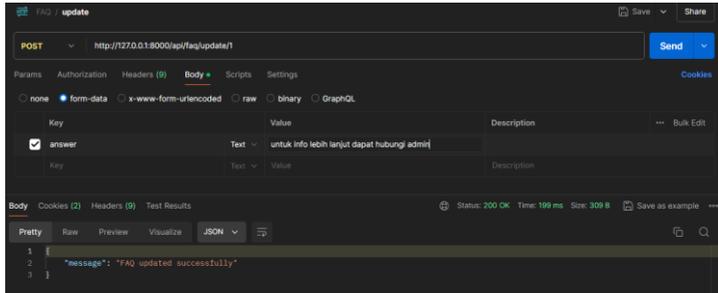
Pada *API* ini digunakan untuk *user* dalam melakukan proses pengambilan data pertanyaan yang tersedia. Dalam *API* ini menerima *request* dengan *method GET* dan memberikan *response* sesuai dengan gambar 4.22 yaitu mengambil informasi terkait data faq atau pertanyaan.



Gambar 4.22 *API* Menampilkan Data *Chatbot*

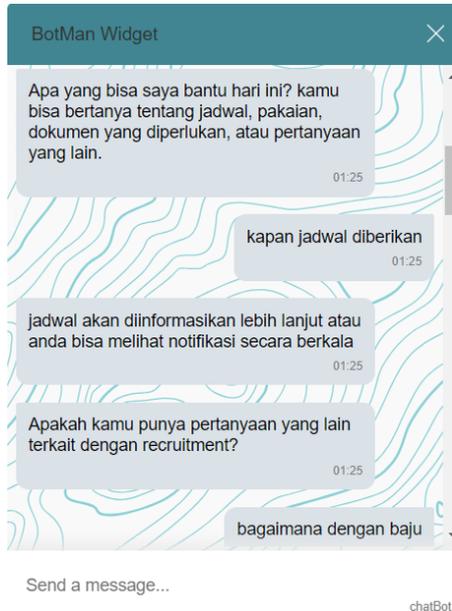
c) *API* mengubah data *chatbot*

Pada *API* ini digunakan untuk *user* dalam melakukan proses perubahan pada data *chatbot* seperti pertanyaan atau jawaban. Dalam *API* ini menerima *request* dengan *method POST* dari form-data dan memberikan *response* sesuai dengan gambar 4.23.



Gambar 4.23 *API* Mengubah Data *Chatbot*

Dalam proyek ini, menggunakan *botman* untuk mengembangkan *chatbot* yang dapat berkomunikasi dengan pengguna secara efektif. *Botman* mempermudah integrasi *chatbot* dengan berbagai platform yang digunakan dalam proyek, memungkinkan pengguna untuk berinteraksi didalam sistem. *Botman* memfasilitasi pengaturan logika percakapan dan respons *chatbot* dengan cara yang terorganisir, sehingga mempercepat proses pengembangan dan mengurangi kebutuhan untuk menangani rincian teknis dari setiap platform. Dengan *Botman*, *chatbot* dalam proyek ini mampu meningkatkan pengalaman pengguna dalam menggunakan sistem.



Gambar 4.24 Implementasi *Chatbot*

4.2.3 Dokumentasi *API* pelamar kerja

4.2.3.1 *API* data diri

API ini digunakan oleh admin untuk mengelola data diri. Pada *API* ini dapat melakukan beberapa jenis permintaan seperti *GET* dan *POST* sehingga sistem nantinya dapat melakukan fungsionalitas lengkap untuk operasi CRU (*Create, Read, Update*) pada data diri. Berikut merupakan implementasi *API* data diri sesuai dengan gambar 4.25.

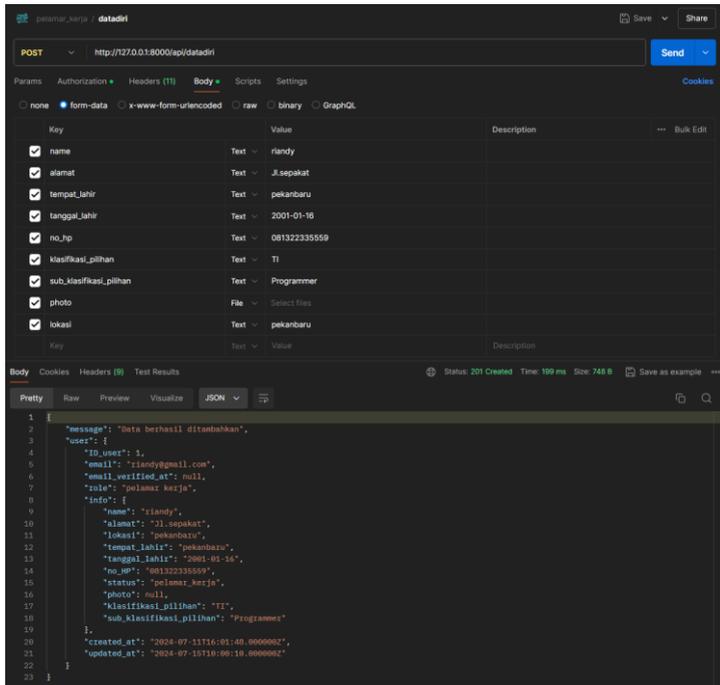


Gambar 4.25 Implementasi *API* Data Diri

a) *API* menambahkan data diri

Pada *API* ini digunakan untuk *user* dalam melakukan proses penambahan data diri. Dalam *API* ini menerima *request* dengan *method POST* dari form-data dan memberikan

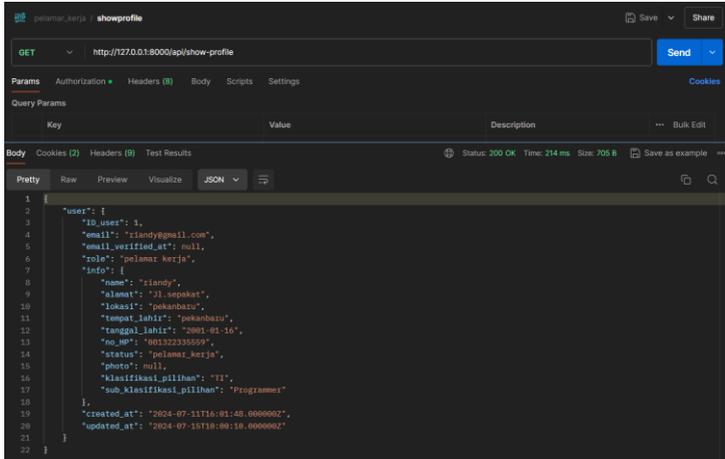
response sesuai dengan gambar 4.26 yaitu menambahkan ke dalam database.



Gambar 4.26 API Menambahkan Data Diri

b) *API* mengambil data diri

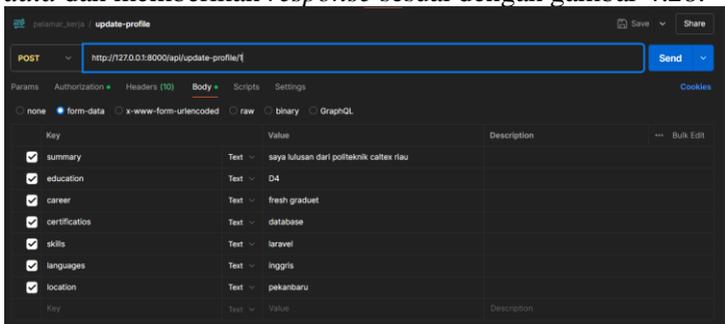
Pada *API* ini digunakan untuk *user* dalam melakukan proses pengambilan data diri. Dalam *API* ini menerima *request* dengan *method GET* dan memberikan *response* sesuai dengan gambar 4.27.



Gambar 4.27 API Mengambil Data Diri

c) API mengubah data diri

Pada API ini digunakan untuk *user* dalam melakukan proses perubahan pada data diri berdasarkan *id* pelamar kerja. Dalam API ini menerima *request* dengan *method POST* dari *form-data* dan memberikan *response* sesuai dengan gambar 4.28.



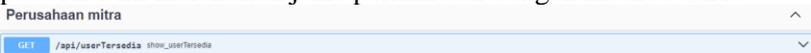
```
1 {
2   "message": "Data user berhasil diperbarui",
3   "user": {
4     "ID_user": 1,
5     "email": "riandy@gmail.com",
6     "email_verified_at": null,
7     "role": "pelamar kerja",
8     "info": {
9       "name": "riandy",
10      "alamat": "Jl.sepakat",
11      "lokasi": "pakanbaru",
12      "tanggal_lahir": "pakanbaru",
13      "tanggal_lahir": "2003-01-16",
14      "no_HP": "981122335566",
15      "status": "pelamar kerja",
16      "photo": null,
17      "klasifikasi_pilihan": "TI",
18      "sub_klasifikasi_pilihan": "Programmer",
19      "summary": "sapa lulusan dari politeknik caltex riau",
20      "education": "SM",
21      "career": "fresh graduet",
22      "certification": "database",
23      "skills": "laravel",
24      "languages": "loggis",
25      "location": "pakanbaru"
26    }
27   }
28 }
```

Gambar 4.28 API Mengubah Data Diri

4.2.4 Dokumentasi API perusahaan mitra

4.2.4.1 API pelamar kerja tersedia

API ini dibuat untuk pelamar kerja dapat mengelola data diri, pada API ini melakukan jenis permintaan dengan method GET.



Gambar 4.29 Implementasi API Pelamar Kerja Tersedia

a) API mengambil data pelamar kerja tersedia

Pada API ini digunakan untuk user dalam melakukan proses pengambilan data pelamar kerja yang tersedia. Dalam API ini menerima request dengan method GET dan memberikan response sesuai dengan gambar 4.30 yang berisikan informasi.

```

46  {
47    "ID_user": 6,
48    "email": "waji@gmail.com",
49    "email_verified_at": null,
50    "role": "pelamar kerja",
51    "info": {
52      "name": "waji"
53    },
54    "created_at": "2024-07-14T12:03:11.000000Z",
55    "updated_at": "2024-07-14T12:03:11.000000Z"
56  },
57  {
58    "ID_user": 8,
59    "email": "banu@gmail.com",
60    "email_verified_at": null,
61    "role": "pelamar kerja",
62    "info": {
63      "name": "banu",
64      "alamat": "jl. sukabatu",
65      "tempat_lahir": "pekambatu",
66      "tanggal_lahir": "2000-01-02",
67      "no_hp": "80952347653",
68      "photo": null,
69      "klasifikasi_pilihan": "programmer",
70      "sub_klasifikasi_pilihan": "ti",
71      "no_hp": "801322335559"
72    },

```

Gambar 4.30 API Mengambil Data Pelamar Kerja

4.2.4.2 API recruitment

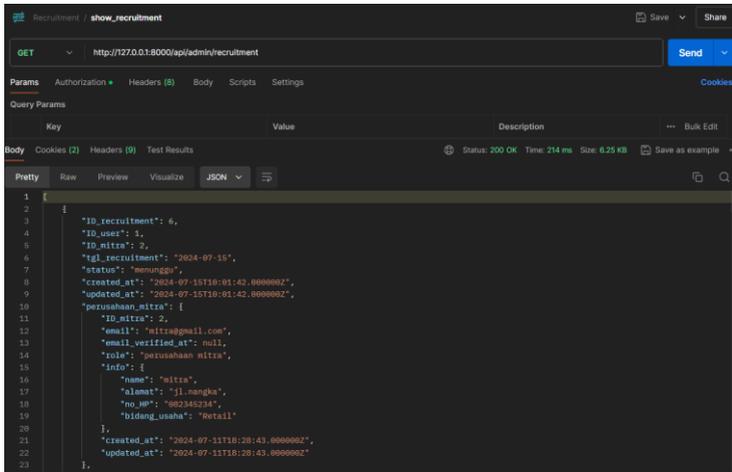
API ini digunakan oleh admin untuk mengelola data *recruitment*. Pada API ini dapat melakukan beberapa jenis permintaan seperti *GET* dan *POST* sehingga sistem nantinya dapat melakukan fungsionalitas lengkap untuk operasi CRU (*Create, Read, Update*) pada data *recruitment*. Berikut merupakan implementasi API *recruitment* sesuai dengan gambar 4.31.

Recruitment	
POST	/api/recruit recruitment
POST	/api/recruit/confirm(id) confirm_recruitment
GET	/api/recruitments show_recruitment
GET	/api/userStatus Show_AllUser

Gambar 4.31 Implementasi API Recruitment

a) API mengambil data *recruitment*

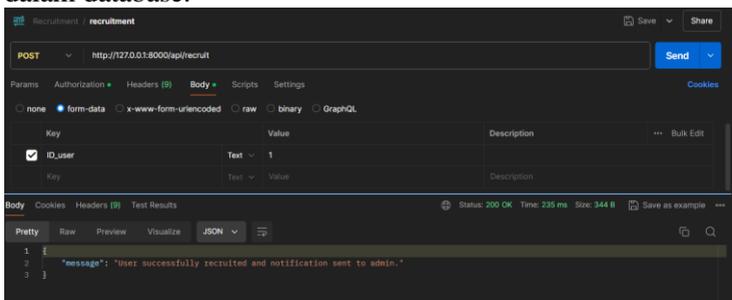
Pada API ini digunakan untuk *user* dalam melakukan proses pengambilan data *recruitment*. Dalam API ini menerima *request* dengan *method GET* dan memberikan *response* sesuai dengan gambar 4.32 yang berisikan informasi.



Gambar 4.32 API Mengambil Data Recruitment

b) API menambahkan recruitment

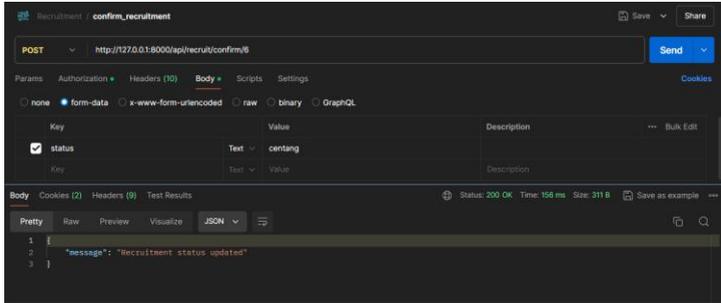
Pada API ini digunakan untuk user dalam melakukan proses penambahan data recruitment. Dalam API ini menerima request dengan method POST dari form-data dan memberikan response sesuai dengan gambar 4.26 yaitu menambahkan ke dalam database.



Gambar 4.33 API Menambahkan Data Recruitment

c) API konfirmasi recruitment

Pada API ini digunakan untuk user dalam melakukan proses konfirmasi recruitment. Dalam API ini menerima request dengan method POST dari form-data dan memberikan response sesuai dengan gambar 4.18.



Gambar 4.34 API Konfirmasi Recruitment

4.3 Pengujian

Berikut adalah beberapa pengujian yang dilakukan terhadap sistem tenaga kerja bagian *backend* berbasis API yang telah dibangun. Pengujian dilakukan dengan dua cara yaitu pengujian dengan pengujian *blackbox* dan pengujian *whitebox*.

4.3.1 Pengujian *whitebox*

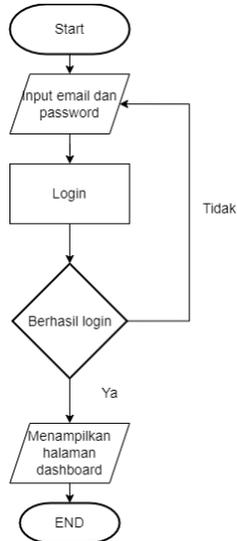
Pengujian *whitebox* berfokus pada pengujian alur logika (*Flowgraph Testing*) yang digunakan dalam sistem. Tujuannya adalah untuk memastikan bahwa alur logika dalam sistem berfungsi dengan benar, menggunakan *flowgraph* untuk mengidentifikasi setiap jalur dalam kode dan menguji masing-masing jalur.

4.3.1.1 Admin (PT. Citratama Indonesia)

Pengujian ini menguji alur dari program yang dapat dilihat dari *flowgraph* berikut.

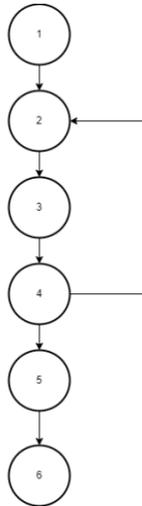
1. Login

Pada proses *login* memastikan hanya pengguna yang dapat mengakses fitur admin dimana alur proses *login* yang direpresentasikan secara visual melalui *flowchart* pada gambar 4.35 berikut



Gambar 4.35 *Flowchart Login Admin*

Berdasarkan *flowchart* pada gambar 4.35 maka disajikan *flowgraph* untuk proses *login* yang mengilustrasikan alur kontrol dari awal proses *login* hingga gagal yang dapat dilihat pada gambar 4.36 sehingga dapat membantu dalam analisis dan pengidentifikasian jalur.



Gambar 4.36 *Flowgraph Login Admin*

Dari *flowgraph* diatas dapat disimpulkan bahwa terdapat 6 *Node* dan 6 *Edge*. *Node* adalah urutan dari simbol proses dan simbol keputusan. Sedangkan *Edge* adalah anak panah yang menggambarkan aliran dari control sesuai dengan diagram alur. Jika dihitung dengan *Cyclometric Complexity* maka didapatkan

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 6 - 6 + 2 \\
 &= 2
 \end{aligned}$$

Dibutuhkan Maksimal 2 *Test Cases* untuk menguji efektifitas dan efisiensi yang terletak di alur atau aktivitas. Dari perhitungan *Cyclometric Complexity* diatas dapat dijumlahkan $V(G)$ keseluruhan jika dijumlahkan maka hasil dari *test cases* yang dapat dilakukan adalah 2 test cases berdasarkan uji *whitebox* testing. Lalu didapatkan *independent path*:

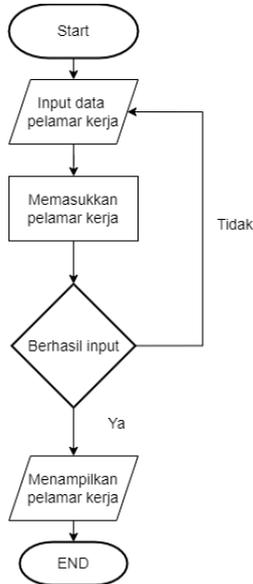
- Jalur 1: 1-2-3-4-5-6
- Jalur 2: 1-2-3-4-2-3-4-5-6

Table 4.2 Test case login admin

Path	1
Jalur	1-2-3-4-5-6
Skenario	<ol style="list-style-type: none"> 1. <i>Start</i> 2. <i>Input email dan password</i> 3. <i>Login</i> 4. <i>Berhasil login</i> 5. <i>Menampilkan dashboard</i> 6. <i>End</i>
Hasil Pengujian	Berhasil
Path	2
Jalur	1-2-3-4-2-3-4-5-6
Skenario	<ol style="list-style-type: none"> 1. <i>Start</i> 2. <i>Input email dan password</i> 3. <i>Login</i> 4. <i>Gagal login</i> 5. <i>Input email dan password</i> 6. <i>Login</i> 7. <i>Berhasil login</i> 8. <i>Menampilkan dashboard</i> 9. <i>End</i>
Hasil Pengujian	Berhasil

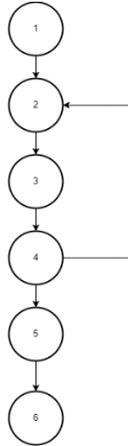
2. Proses input data pelamar kerja

Pada proses input data pelamar kerja mengelola data pelamar kerja dengan menambahkan data pelamar kerja dimana alur proses yang direpresentasikan secara visual melalui *flowchart* pada gambar 4.37 berikut.



Gambar 4.37 *Flowchart* Input Data Pelamar Kerja

Berdasarkan *flowchart* pada gambar 4.37 maka disajikan *flowgraph* yang mengilustrasikan alur kontrol dari awal proses input data pelamar kerja hingga gagal yang dapat dilihat pada gambar 4.38 sehingga dapat membantu dalam analisis dan pengidentifikasian jalur.



Gambar 4.38 *Flowgraph* Input Data Pelamar Kerja

Dari *flowgraph* diatas dapat disimpulkan bahwa terdapat 6 *Node* dan 6 *Edge*. *Node* adalah urutan dari simbol proses dan simbol keputusan. Sedangkan *Edge* adalah anak panah yang menggambarkan aliran dari kontrol sesuai dengan diagram alur. Jika dihitung dengan *Cyclometric Complexity* maka didapatkan

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 6 - 6 + 2 \\
 &= 2
 \end{aligned}$$

Dibutuhkan Maksimal 2 *Test Cases* untuk menguji efektifitas dan efisiensi yang terletak di alur atau aktivitas. Dari perhitungan *Cyclometric Complexity* diatas dapat dijumlahkan $V(G)$ keseluruhan jika dijumlahkan maka hasil dari test cases yang dapat dilakukan adalah 2 *test cases* berdasarkan uji *whitebox* testing. Lalu didapatkan *independent path*:

- Jalur 1: 1-2-3-4-5-6
- Jalur 2: 1-2-3-4-2-3-4-5-6

Table 4.3 Test case input data pelamar kerja

Path	1
Jalur	1-2-3-4-5-6

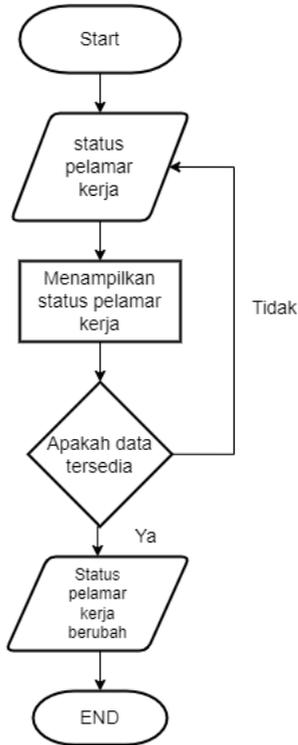
Skenario	<ol style="list-style-type: none"> 1. Start 2. Input data pelamar kerja 3. Memasukkan data pelamar kerja 4. Berhasil input 5. Menampilkan data pelamar kerja 6. End
Hasil Pengujian	Berhasil
Path	2
Jalur	1-2-3-4-2-3-4-5-6
Skenario	<ol style="list-style-type: none"> 1. Start 2. Input data pelamar kerja 3. Memasukkan data pelamar kerja 4. Gagal input 5. Input data pelamar kerja 6. Memasukkan data pelamar kerja 7. Berhasil input 8. Menampilkan data pelamar kerja 9. End
Hasil Pengujian	Berhasil

4.3.1.2 Perusahaan mitra

Pengujian ini menguji alur dari program yang dapat dilihat dari flowgraph berikut.

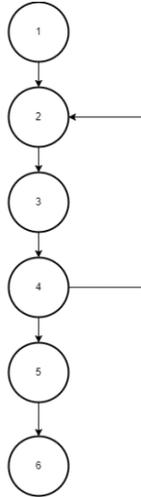
1. Proses *recruitment*

Pada proses *recruitment* mengelola data *recruitment* dimana alur proses yang direpresentasikan secara visual melalui *flowchart* pada gambar 4.39 berikut.



Gambar 4.39 *Flowchart* Proses *Recruitment*

Berdasarkan *flowchart* pada gambar 4.39 maka disajikan *flowgraph* yang mengilustrasikan alur kontrol dari awal proses *recruitment* hingga gagal yang dapat dilihat pada gambar 4.40 sehingga dapat membantu dalam analisis dan pengidentifikasian jalur.



Gambar 4.40 *Flowgraph* Proses *Recruitment*

Dari *flowgraph* diatas dapat disimpulkan bahwa terdapat 6 *Node* dan 6 *Edge*. *Node* adalah urutan dari simbol proses dan simbol keputusan. Sedangkan *Edge* adalah anak panah yang menggambarkan aliran dari kontrol sesuai dengan diagram alur. Jika dihitung dengan *Cyclometric Complexity* maka didapatkan

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 6 - 6 + 2 \\ &= 2 \end{aligned}$$

Dibutuhkan maksimal 2 *Test Cases* untuk menguji efektifitas dan efisiensi yang terletak di alur atau aktivitas. Dari perhitungan *Cyclometric Complexity* diatas dapat dijumlahkan $V(G)$ keseluruhan jika dijumlahkan maka hasil dari *test cases* yang dapat dilakukan adalah 2 *test cases* berdasarkan uji *whitebox* testing. Lalu didapatkan *independent path*:

- Jalur 1: 1-2-3-4-5-6
- Jalur 2: 1-2-3-4-2-3-4-5-6

Table 4.4 Test case proses recruitment

Path	1
Jalur	1-2-3-4-5-6
Skenario	<ol style="list-style-type: none"> 1. Start 2. Status pelamar kerja 3. Menampilkan status pelamar kerja 4. Data tersedia 5. Status pelamar kerja berubah 6. End
Hasil Pengujian	Berhasil
Path	2
Jalur	1-2-3-4-2-3-4-5-6
Skenario	<ol style="list-style-type: none"> 1. Start 2. Status pelamar kerja 3. Menampilkan status pelamar kerja 4. Data tidak tersedia 5. Status pelamar kerja 6. Menampilkan status pelamar kerja 7. Data tersedia 8. Status pelamar kerja berubah 9. End
Hasil Pengujian	Berhasil

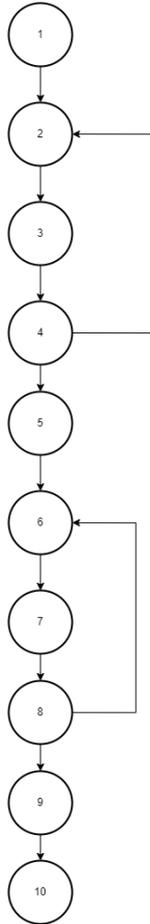
2. Proses tambah jadwal

Pada proses tambah jadwal mengelola data jadwal dimana alur proses yang direpresentasikan secara visual melalui *flowchart* pada gambar 4.40 berikut.



Gambar 4.41 *Flowchart* Tambah Jadwal

Berdasarkan *flowchart* pada gambar 4.41 maka disajikan *flowgraph* yang mengilustrasikan alur kontrol dari awal proses tambah jadwal hingga gagal yang dapat dilihat pada gambar 4.42 sehingga dapat membantu dalam analisis dan pengidentifikasian jalur.



Gambar 4.42 *Flowgraph* Tambah Jadwal

Dari *flowgraph* diatas dapat disimpulkan bahwa terdapat 10 *Node* dan 10 *Edge*. *Node* adalah urutan dari simbol proses dan simbol keputusan. Sedangkan *Edge* adalah anak panah yang menggambarkan aliran dari kontrol sesuai dengan diagram alur. Jika dihitung dengan *Cyclometric Complexity* maka didapatkan

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 11 - 10 + 2
 \end{aligned}$$

= 3

Dibutuhkan Maksimal 2 *Test Cases* untuk menguji efektifitas dan efisiensi yang terletak di alur atau aktivitas. Dari perhitungan *Cyclometric Complexity* diatas dapat dijumlahkan $V(G)$ keseluruhan jika dijumlahkan maka hasil dari test cases yang dapat dilakukan adalah 3 *test cases* berdasarkan uji *whitebox* testing. Lalu didapatkan *independent path*:

Jalur 1: 1-2-3-4-5-6-7-8-9-10

Jalur 2: 1-2-3-4-2-3-4-5-6-7-8-9-10

Jalur 3: 1-2-3-4-5-6-7-8-6-7-8-9-10

Table 4.5 Test case tambah jadwal

Path	1
Jalur	1-2-3-4-5-6-7-8-9-10
Skenario	<ol style="list-style-type: none">1. Start2. Input jadwal3. Memasukkan data jadwal4. Berhasil input data jadwal5. Menampilkan data jadwal6. Assign pelamar kerja7. Pelamar kerja diassign untuk mendapatkan jadwal8. Berhasil assign9. Jadwal dan penerima jadwal (pelamar kerja)10. End
Hasil Pengujian	Berhasil
Path	2
Jalur	1-2-3-4-2-3-4-5-6-7-8-9-10
Skenario	<ol style="list-style-type: none">1. Start2. Input data jadwal3. Memasukkan data jadwal4. Gagal input data jadwal5. input data jadwal6. Memasukkan data jadwal

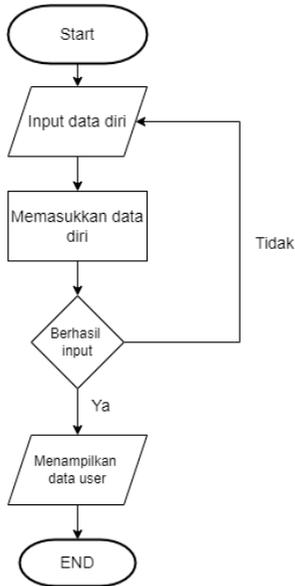
	7. Berhasil input data jadwal 8. Menampilkan data jadwal 9. Assign pelamar kerja 10. Pelamar kerja diassign untuk mendapatkan jadwal 11. Berhasil assign 12. Jadwal dan penerima jadwal (pelamar kerja) 13. End
Hasil Pengujian	Berhasil
Path	3
Jalur	1-2-3-4-5-6-7-8-6-7-8-9-10
Skenario	1. Start 2. Input jadwal 3. Memasukkan data jadwal 4. Berhasil input data jadwal 5. Menampilkan data jadwal 6. Assign pelamar kerja 7. Pelamar kerja diassign untuk mendapatkan jadwal 8. Gagal assign 9. Assign pelamar kerja 10. Pelamar kerja diassign untuk mendapatkan jadwal 11. Berhasil assign 12. Jadwal dan penerima jadwal 13. End
Hasil	Berhasil

4.3.1.3 Pelamar kerja

Pengujian ini menguji alur dari program yang dapat dilihat dari *flowgraph* berikut.

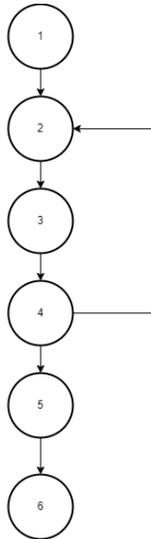
1. Proses input data diri

Pada proses input data diri dimana alur proses yang direpresentasikan secara visual melalui *flowchart* pada gambar 4.43 berikut.



Gambar 4.43 *Flowchart* Input Data Diri

Berdasarkan *flowchart* pada gambar 4.43 maka disajikan *flowgraph* yang mengilustrasikan alur kontrol dari awal proses input data diri hingga gagal yang dapat dilihat pada gambar 4.44 sehingga dapat membantu dalam analisis dan pengidentifikasian jalur.



Gambar 4.44 *Flowgraph* Input Data Diri

Dari *flowgraph* diatas dapat disimpulkan bahwa terdapat 6 *Node* dan 6 *Edge*. *Node* adalah urutan dari simbol proses dan simbol keputusan. Sedangkan *Edge* adalah anak panah yang menggambarkan aliran dari kontrol sesuai dengan diagram alur. Jika dihitung dengan *Cyclometric Complexity* maka didapatkan

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 6 - 6 + 2 \\
 &= 2
 \end{aligned}$$

Dibutuhkan Maksimal 2 *Test Cases* untuk menguji efektifitas dan efisiensi yang terletak di alur atau aktivitas. Dari perhitungan *Cyclometric Complexity* diatas dapat dijumlahkan $V(G)$ keseluruhan jika dijumlahkan maka hasil dari test cases yang dapat dilakukan adalah 2 *test cases* berdasarkan uji *whitebox* testing. Lalu didapatkan *independent path*:

Jalur 1: 1-2-3-4-5-6

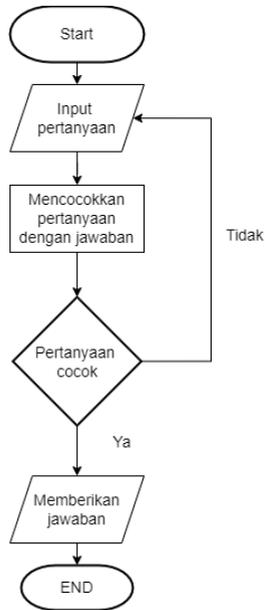
Jalur 2: 1-2-3-4-2-3-4-5-6

Table 4.6 Test case input data diri

Path	1
Jalur	1-2-3-4-5-6
Skenario	<ol style="list-style-type: none"> 1. Start 2. Input data diri 3. Memasukkan data diri 4. Berhasil input 5. Menampilkan data user 6. End
Hasil Pengujian	Berhasil
Path	2
Jalur	1-2-3-4-2-3-4-5-6
Skenario	<ol style="list-style-type: none"> 1. Start 2. Input data diri 3. Memasukkan data diri 4. Gagal input 5. Input data diri 6. Memasukkan data diri 7. Berhasil input 8. Menampilkan data user 9. End
Hasil Pengujian	Berhasil

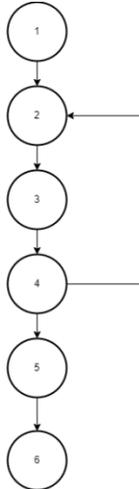
2. Chatbot

Pada proses *chatbot* dimana alur proses yang direpresentasikan secara visual melalui *flowchart* pada gambar 4.45 berikut.



Gambar 4.45 *Flowchart Chatbot*

Berdasarkan *flowchart* pada gambar 4.45 maka disajikan *flowgraph* yang mengilustrasikan alur kontrol dari awal proses input data diri hingga gagal yang dapat dilihat pada gambar 4.46 sehingga dapat membantu dalam analisis dan pengidentifikasian jalur.



Gambar 4.46 *Flowgraph* Input Data Diri

Dari *flowgraph* diatas dapat disimpulkan bahwa ada 6 *Node* dan 6 *Edge*. *Node* adalah urutan dari simbol proses dan simbol keputusan. Sedangkan *Edge* adalah anak panah yang menggambarkan aliran dari kontrol sesuai dengan diagram alur. Jika dihitung dengan *Cyclometric Complexity* maka didapatkan

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 6 - 6 + 2 \\
 &= 2
 \end{aligned}$$

Dibutuhkan Maksimal 2 *Test Cases* untuk menguji efektifitas dan efisiensi yang terletak di alur atau aktivitas. Dari perhitungan *Cyclometric Complexity* diatas dapat dijumlahkan $V(G)$ keseluruhan jika dijumlahkan maka hasil dari test cases yang dapat dilakukan adalah 2 *test cases* berdasarkan uji *whitebox* testing. Lalu didapatkan *independent path*:

Jalur 1: 1-2-3-4-5-6

Jalur 2: 1-2-3-4-2-3-4-5-6

Table 4.7 Test case chatbot

Path	1
Jalur	1-2-3-4-5-6
Skenario	<ol style="list-style-type: none"> 1. Start 2. Input pertanyaan 3. Memcocokkan pertanyaan dengan data di database 4. Pertanyaan cocok 5. Memberikan jawaban 6. End
Hasil Pengujian	Berhasil
Path	2
Jalur	1-2-3-4-2-3-4-5-6
Skenario	<ol style="list-style-type: none"> 1. Start 2. Input pertanyaan 3. Mencocokkan pertanyaan dengan data di database 4. Pertanyaan tidak cocok 5. Input pertanyaan 6. Mencocokkan pertanyaan dengan data di database 7. Pertanyaan cocok 8. Memberikan pertanyaan 9. End
Hasil Pengujian	Berhasil

Detail mengenai hasil pengujian whitebox dapat ditemukan pada **Lampiran C**. Pengujian *whitebox* yang telah didapatkan maka dilakukan rekapitulasi hasil whitebox yang disajikan pada tabel 4.8.

Table 4.8 Rekapitulasi hasil whitebox

<i>Flowgraph</i>	<i>Cyclometric Complexity</i>	<i>independent path</i>
Login Admin	2	2

<i>Flowgraph</i>	<i>Cyclometric Complexity</i>	<i>independent path</i>
Proses input data pelamar kerja	2	2
Proses input data perusahaan mitra	2	2
Proses input berita	2	2
Proses approve recruitment	2	2
Proses mengirim jadwal	2	2
Proses recruitment	2	2
Proses penerimaan pelamar	2	2
Proses tambah jadwal	3	3
Proses input data diri	2	2
Chatbot	2	2
Jumlah	23	23

Pada tabel 4.8 diatas, berdasarkan hasil pengujian yang didapatkan bernilai sama untuk jumlah *Cyclomatic Complexity* = 23 dan *independent path* = 23, maka ditarik kesimpulan dari hasil pengujian tersebut menunjukkan bahwa alur logika sistem tenaga kerja ini sudah benar dan sesuai dengan logika yang ada pada kode program, serta bebas dari kesalahan.

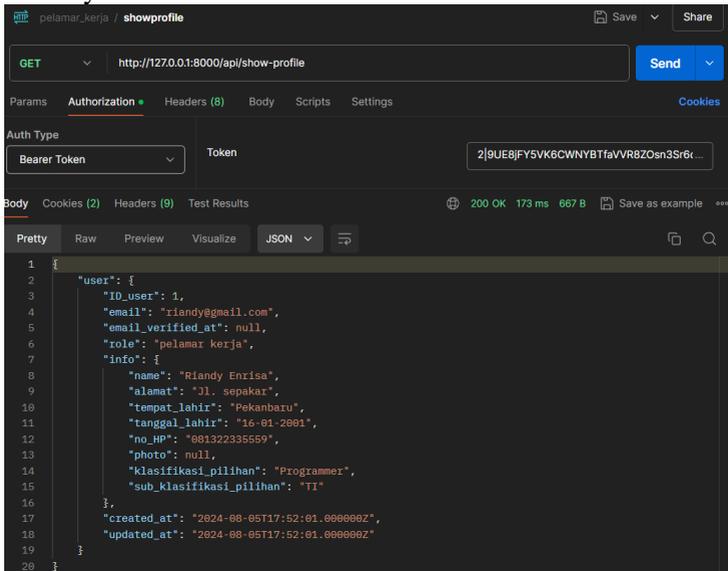
4.3.2 Pengujian *Blackbox*

Pengujian sistem merupakan tahapan pengujian *software* atau perangkat lunak untuk mengevaluasi sebuah sistem apakah sudah berjalan dengan baik atau belum, pengujian *blackbox* menguji fungsionalitas dari sistem yang telah dirancang. Pengujian ini dilakukan dengan menggunakan *postman* untuk memastikan setiap *endpoint* API berfungsi dengan baik, mengirimkan request ke setiap *endpoint* dan memeriksa *response* yang diberikan, *response* sesuai dengan yang diharapkan berdasarkan spesifikasi. Untuk melihat detail hasil pengujian *blackbox* yang telah dibangun dapat ditemukan pada **Lampiran D**. Berdasarkan hasil evaluasi dengan pengujian *blackbox* yang telah dilakukan sebanyak 1 kali kepada penulis sendiri,

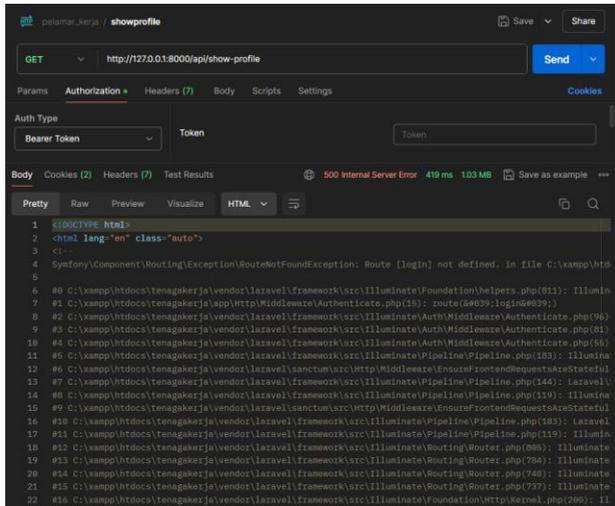
ditemukan 35 butir unit pengujian yang diujikan telah berhasil dilaksanakan tanpa masalah.

4.3.2.1 Pengujian autentikasi

Pada pengujian ini akan menguji bahwa hanya user memiliki token autentikasi yang dapat mengakses *endpoint API*. Untuk pengujiannya dilakukan menggunakan *postman* yang dapat dilihat pada gambar 4.47 untuk user yang memiliki token autentikasi dan pada gambar 4.48 untuk user yang tidak memiliki token autentikasi maka akses nya ditolak.



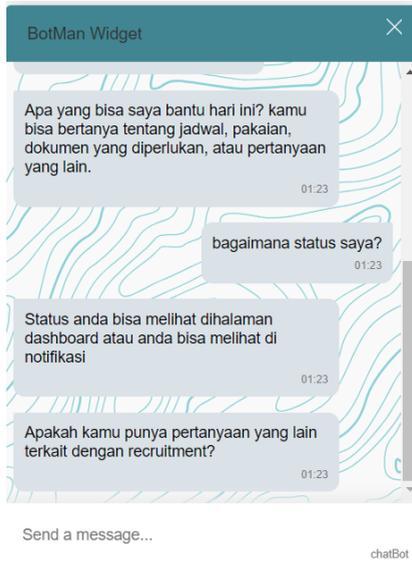
Gambar 4.47 Pengujian autentikasi berhasil



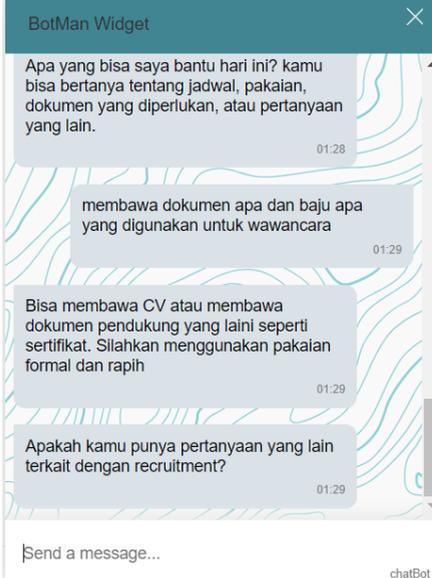
Gambar 4.48 Pengujian autentikasi gagal

4.3.2.2 Pengujian chatbot

Pengujian ini dilakukan untuk melihat respon yang diberikan oleh bot setelah adanya pertanyaan yang dimasukkan oleh *user*. Pengujian ini melihat respon jika diberikan satu kata kunci dan respon dari dua kata kunci. Didalam database terdapat 5 kata kunci, yaitu dokumen, jadwal, baju, *profile*, dan status. Pengujian untuk melihat respon dari satu kata kunci menggunakan kata kunci status yang dapat dilihat pada gambar 4.49. Pengujian untuk melihat respon dari dua kata kunci akan menggunakan kata kunci dokumen dan baju yang dapat dilihat pada gambar 4.50



Gambar 4.49 Pengujian chatbot 1 kata kunci



Gambar 4.50 Pengujian chatbot 2 kata kunci

4.4 Analisis Hasil

4.4.1 Analisis implementasi REST API

Implementasi RESTAPI dalam proyek ini menggunakan *Laravel*, yang memfasilitasi pengembangan dengan fitur-fitur seperti *routing*, *middleware*, dan *validasi*. Dengan konfigurasi yang tepat, API ini berhasil dikembangkan untuk mendukung pengelolaan data menggunakan MySQL secara efektif. *Endpoint-endpoint* API dirancang dengan baik untuk memenuhi berbagai kebutuhan aplikasi, mencakup operasi standar seperti *create*, *read*, *update*, dan *delete*, serta operasi khusus sesuai kebutuhan spesifik proyek. Dengan dokumentasi yang jelas dan pengujian menyeluruh, API ini terbukti berfungsi dengan baik dalam berbagai kondisi dan input, menjadikannya komponen yang andal dan efisien dalam keseluruhan aplikasi.

4.4.2 Analisis Pengujian *Whitebox*

Pengujian *white box* dalam proyek ini menerapkan metode *Flowgraph Testing* untuk memeriksa alur logika di dalam sistem. Tujuan utama dari pengujian ini adalah untuk memastikan bahwa alur kode program berjalan dengan benar. Dengan menggunakan *flowgraph*, setiap jalur eksekusi dalam kode diidentifikasi dan diuji secara mendetail. Pengujian ini dilakukan terhadap 3 role yaitu admin, perusahaan mitra, dan pelamar kerja, untuk memastikan bahwa alur logika yang spesifik untuk setiap role berfungsi sesuai dengan yang diharapkan. Pendekatan ini memastikan bahwa semua jalur kode yang mungkin terlewat selama proses eksekusi telah diuji, sehingga mengidentifikasi dan memperbaiki potensi masalah dalam alur kode program.

4.4.3 Analisis Pengujian *Blacbox*

Pengujian *blackbox* dalam proyek ini dilakukan untuk mengevaluasi fungsionalitas sistem secara menyeluruh. Pengujian dilakukan dengan mengirimkan *request* ke setiap *endpoint API* dan memeriksa respons yang diterima untuk memastikan bahwa respons tersebut sesuai dengan spesifikasi yang telah ditetapkan, dengan pengujian tersebut juga dapat membuktikan keamanan dari penggunaan *middleware* dapat berfungsi dengan baik sehingga *user* yang tidak memiliki token autentikasi tidak dapat mengakses *endpoint*

API. Pendekatan ini, dapat memastikan bahwa semua *endpoint* berfungsi dengan baik. Pengujian *blackbox* ini, yang dilakukan sebanyak satu kali, mencakup 35 unit pengujian yang berhasil diselesaikan tanpa ditemukan masalah. Pendekatan ini memberikan keyakinan bahwa sistem bekerja secara efektif dan memenuhi standar fungsional yang diinginkan.

Pengujian chatbot dalam proyek ini dilakukan dengan memasukkan pertanyaan yang mengandung satu kata kunci dan dua kata kunci. Hasil pengujian menunjukkan bahwa respon yang diberikan oleh chatbot untuk satu kata kunci sesuai dengan kata kunci yang diberikan. Untuk pertanyaan yang mengandung dua kata kunci, chatbot mampu mengenali kedua kata kunci tersebut dan memberikan jawaban yang komprehensif, chatbot merespon dengan dua bagian jawaban yang mencakup kedua aspek yang ditanyakan.

BAB V PENUTUP

5.1 Kesimpulan

Berdasarkan tujuan dan manfaat yang telah dirumuskan, penelitian ini berhasil merancang dan membangun sistem *backend* untuk tenaga kerja pada PT. Citratama Indonesia menggunakan metode REST API. Beberapa kesimpulan yang dapat diambil adalah sebagai berikut:

1. Perancangan dan pembangunan sistem tenaga kerja bagian backend yang dirancang menggunakan *REST API* berhasil diimplementasikan dengan baik. Sistem ini mampu mengelola data pelamar kerja dan perusahaan mitra secara efisien, memastikan bahwa data dapat diakses, diperbarui, dan dikelola dengan mudah.
2. *Endpoint API* yang dikembangkan dalam sistem ini berfungsi dengan baik. *Endpoint-endpoint* tersebut bekerja sesuai dengan yang diharapkan dan memberikan respons yang tepat, sehingga dapat digunakan secara efektif oleh frontend untuk berbagai keperluan aplikasi
3. Sistem ini telah diuji menggunakan metode *whitebox* dan *blackbox* untuk memastikan bahwa semua fungsi bekerja dengan benar dan sesuai dengan spesifikasi yang ditetapkan. Pengujian *whitebox* memastikan bahwa logika internal sistem dan jalur eksekusi berjalan dengan benar, sementara pengujian *blackbox* memastikan bahwa *endpoint API* yang sudah dibangun dapat berfungsi dengan benar.
4. Sistem ini memiliki keamanan yang baik dan sudah diuji, keamanan menggunakan *middleware* autentikasi menunjukkan bahwa hanya user yang memiliki token autentikasi yang bisa mengakses *endpoint API* dan jika user tidak memiliki token autentikasi akan ditolak permintaan akses *endpoint API*.

5.2 Saran

Adapun saran yang dapat dijadikan pertimbangan untuk pengembangan penelitian selanjutnya adalah

1. Meskipun sistem telah menggunakan middleware untuk autentikasi dan otorisasi, langkah-langkah ini mungkin belum cukup untuk melindungi dari ancaman keamanan yang lebih kompleks. Disarankan untuk menerapkan langkah-langkah keamanan tambahan seperti autentikasi dua faktor (2FA), enkripsi data sensitif, serta melakukan penilaian kerentanan dan pengujian penetrasi secara rutin. Langkah-langkah ini akan membantu dalam mengidentifikasi dan mengatasi celah keamanan yang mungkin tidak terdeteksi dengan pengamanan dasar saja.
2. Dalam meningkatkan kenyamanan dan interaktivitas pengguna, sistem ini dapat ditambahkan fitur chat *real-time*. Fitur ini memungkinkan pelamar kerja dan PT. Citratama Indonesia untuk berkomunikasi secara langsung dan cepat di sistem. Dengan adanya fitur ini, proses tanya jawab dapat dilakukan dengan lebih efisien, mengurangi waktu tunggu, dan meningkatkan kepuasan pengguna terhadap sistem.

DAFTAR PUSTAKA

- Abdurrohman, F., & Warih Utami, A. (2023). Rancang bangun aplikasi pemetaan tenaga kerja berbasis web sistem informasi geografis pada Disnakertrans Jawa Timur studi kasus bidang penempatan dan pelatihan kerja. (*Journal of Emerging Information Systems and Business Intelligence*).
- Dwi Haristya Aryadi, S., Paramita, A., Raya Temgang No, J., Gedong, K., Rebo, P., & Timur, J. (2022). *Sistem informasi manajemen tenaga kerja outsourcing pada PT. ACM berbasis java*.
- Frederica Rosabel Ramli, Fikri Hakim, & Ria Anggelina Hutabarat. (2021a). Perancangan Web Design Aplikasi E-Learning dengan Metode Prototype pada Tingkat SMA. *Majalah Ilmiah UPI YPTK*, 13–18. <https://doi.org/10.35134/jmi.v28i1.62>
- Frederica Rosabel Ramli, Fikri Hakim, & Ria Anggelina Hutabarat. (2021b). Perancangan Web Design Aplikasi E-Learning dengan Metode Prototype pada Tingkat SMA. *Majalah Ilmiah UPI YPTK*, 13–18. <https://doi.org/10.35134/jmi.v28i1.62>
- Galindra Wardhana, W., Arwani, I., & Rahayudi, B. (2020). *Implementasi Teknologi Restful Web Service Dalam Pengembangan Sistem Informasi Perekaman Prestasi Mahasiswa Berbasis Website (Studi Kasus: Fakultas Teknologi Pertanian Universitas Brawijaya)* (Vol. 4, Issue 2). <http://j-ptiik.ub.ac.id>

- Gwijangge, L., Kawung, G. M. V, Siwu, H., Pembangunan, J. E., Ekonomi, F., & Bisnis, D. (n.d.). *Pengaruh investasi dan tenaga kerja terhadap pertumbuhan ekonomi provinsi Papua*.
- Nugraha, K. A., & Sebastian, D. (2021). Chatbot Layanan Akademik Menggunakan K-Nearest Neighbor. *Jurnal Sains Dan Informatika*, 7(1), 11–19. <https://doi.org/10.34128/jsi.v7i1.285>
- Pangestika, R., & Dirgahayu, R. T. (n.d.). *Pengembangan Back-end Sistem Informasi Pendataan Sekolah Desa Komunitas Pendar Foundation Yogyakarta*.
- Praniffa, A. C., Syahri, A., Sandes, F., Fariha, U., Giansyah, Q. A., & Hamzah, M. L. (n.d.). Pengujian black box dan white box sistem informasi parkir berbasis web black box and white box testing of web-based parking system. In *Jurnal Testing dan Implementasi Sistem Informasi* (Vol. 1, Issue 1).
- Rusdiana, H. A., Moch, M. M., Irfan, S. T., Kom, M., & Ramdhadi, H. M. A. (2014). *Sistem Informasi Manajemen Sistem Informasi Manajemen Pustaka Setia Pengantar: Penerbit PUSTAKA SETIA Bandung*.
- Wahyudi, I., & Alameka, F. (2023). Analisis blackbox testing dan user acceptance testing terhadap sistem informasi solusimedsosku. *Jurnal Teknosains Kodepena* /, 04, 1–9.
- Wulandari, N., & Sutiyatno, S. (2019). Sistem informasi penyaluran tenaga kerja berbasis web di CV. Jasa informasi kerja magelang. In *Jurnal TRANSFORMASI* (Vol. 15, Issue 1).

Yunhasnawa, Y., Aprilianto, S., Allam, M. F., Arbi, D., Himawan, S.,
& Informasi, J. T. (n.d.). *Pengembangan restful api untuk
membantu pembudidaya ikan lele dalam sistem manajemen
rantai pasok “panen-panen.”*

LAMPIRAN A

REQUIREMENT PLANNING

1. Formulir *Requirement Planning*

Formulir Pengumpulan Kebutuhan Pengguna

Pengguna : Elvi Srienti

Jabatan : Direktur PT. Citratama Indonesia

No	Pertanyaan	Jawaban
1	Apa latar belakang dari Pt. Citratama Indonesia	Citratama Indonesia menjembatani antara kebutuhan perusahaan dengan tenaga kerja yang sudah terlatih dan memiliki potensi serta keterampilan sehingga mampu bersaing dalam dunia usaha dan dunia industri. Sementara itu di sisi dunia usaha dan industri citratama indonesia memiliki wawasan mendalam tentang kebutuhan tenaga kerja di berbagai perusahaan sehingga dapat membantu perusahaan menemukan kandidat terbaik untuk mengisi posisi yang kosong
2	Bagaimana dengan jam operasional di Pt. Citratama Indonesia	Dari pukul 08.00 s.d 16.00 dari hari senin sampai dengan hari jumat

3	<p>Bisnis apa yang dijalankan di Pt. Citratama Indonesia</p>	<p>Citratama Indonesia mempunyai 6 Unit bisnis, di antaranya:</p> <ol style="list-style-type: none"> 1. Sales Training 2. Training Softskill 3. Vendor Penjualan 4. Jasa Rekrutment 5. Jasa Head Hunter 6. Outbond Training
4	<p>Sudah berapa banyak melakukan perekrutan pelamar kerja untuk perusahaan</p>	<p>Kurang lebih 50 pelamar yang sudah ditempatkan dan 30 pelamar yang belum di tempatkan, 28 perusahaan mitra yang telah bekerjasama</p>
5	<p>Apa kendala/masalah yang sedang dihadapi</p>	<p>Proses administrasi yang ada di citratama indonesi masih menggunakan pola yang kurang tertata sehingga data kandidat dan data perusahaan relasi masih belum terorganisir dengan baik</p>
6	<p>Kenapa butuh pembuatan sistem ini</p>	<p>Karena, tidak punya administrasi yang tertata, daftar dengan menggunakan link yang sudah disiapkan yaitu link gform, upload berkas dan CV dilakukan rekrutment dengan menggunakan Gdrive, Citratama mengirim berkasnya ke mitra perusahaan dengan melalui email, melakukan kunjungan langsung sehingga dibutuhkan sistem untuk semua</p>

7	Fitur apa saja yang diinginkan di sistem ini	Fitur notifikasi, perekrutan pelamar kerja dan list pelamar kerja.
8	Tahun berapakah PT. Citratama Indonesia berdiri?	2022 agustus berdirinya, legalitasnya keluar pada bulan maret tahun 2023
9	Pada Pt. Citratama Indonesia apa posisi/jabatan bapak?	Direktur

Pekanbaru, 14 November 2023

Elvi Srienti
Direktur PT. Citratama Indonesia

2. BAP Wawancara

Berita Acara Wawancara

Pada hari Senin tanggal 19 November 2023 dengan ini menerangkan yang bertanda tangan di bawah ini:

Nama : Riandy Enrisa
NIM : 2055301121
Jurusan : Teknik Informatika
Instansi : Politeknik Caltex Riau

Telah melakukan wawancara dengan:

Nama : Dhea
Umur : 23 tahun
Pekerjaan : IT Consultant

Dalam rangka penyusunan Proyek Akhir dengan judul "Perancangan *backend* pada sistem tenaga kerja menggunakan REST API"

Pekanbaru, 19 November 2023

Narasumber	Pewawancara
	
Dhea	Riandy Enrisa

LAMPIRAN B
HASIL WAWANCARA

Pengguna : Elvi Srienti

Jabatan : Direktur PT. Citratama Indonesia

No	Pertanyaan	Jawaban
1	Apa latar belakang dari Pt. Citratama Indonesia	Citratama Indonesia menjembatani antara kebutuhan perusahaan dengan tenaga kerja yang sudah terlatih dan memiliki potensi serta keterampilan sehingga mampu bersaing dalam dunia usaha dan dunia industri. Sementara itu di sisi dunia usaha dan industri citratama indonesia memiliki wawasan mendalam tentang kebutuhan tenaga kerja di berbagai perusahaan sehingga dapat membantu perusahaan menemukan kandidat terbaik untuk mengisi posisi yang kosong
2	Bagaimana dengan jam operasional di Pt. Citratama Indonesia	Dari pukul 08.00 s.d 16.00 dari hari senin sampai dengan hari jumat

3	Bisnis apa yang dijalankan di Pt. Citratama Indonesia	Citratama Indonesia mempunyai 6 Unit bisnis, di antaranya: <ol style="list-style-type: none"> 1. Sales Training 2. Training Softskill 3. Vendor Penjualan 4. Jasa Rekrutment 5. Jasa Head Hunter 6. Outbond Training
4	Sudah berapa banyak melakukan perekrutan pelamar kerja untuk perusahaan	Kurang lebih 50 pelamar yang sudah ditempatkan dan 30 pelamar yang belum di tempatkan, 28 perusahaan mitra yang telah bekerjasama
5	Apa kendala/masalah yang sedang dihadapi	Proses administrasi yang ada di citratama indonesi masih menggunakan pola yang kurang tertata sehingga data kandidat dan data perusahaan relasi masih belum terorganisir dengan baik
6	Kenapa butuh pembuatan sistem ini	Karena, tidak punya administrasi yang tertata, daftar dengan menggunakan link yang sudah disiapkan yaitu link gform, upload berkas dan CV dilakukan rekrutment dengan menggunakan Gdrive, Citratama mengirim berkasnya ke mitra perusahaan dengan

		melalui email, melakukan kunjungan langsung sehingga dibutuhkan sistem untuk semua
7	Fitur apa saja yang diinginkan di sistem ini	Fitur notifikasi, perekrutan pelamar kerja dan list pelamar kerja.
8	Tahun berapakah PT. Citratama Indonesia berdiri?	2022 agustus berdirinya, legalitasnya keluar pada bulan maret tahun 2023
9	Pada Pt. Citratama Indonesia apa posisi/jabatan bapak?	Direktur

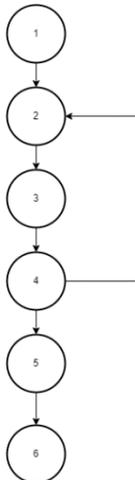
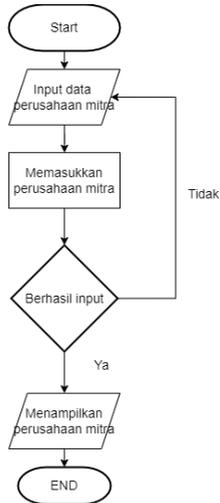
LAMPIRAN C

WHITEBOX TESTING

Melihat Data Anggota (Admin)

Kasus Uji Role: Admin

Proses input data perusahaan mitra



$$V(G) = E - N + 2$$

$$= 6 - 6 + 2$$

$$= 2$$

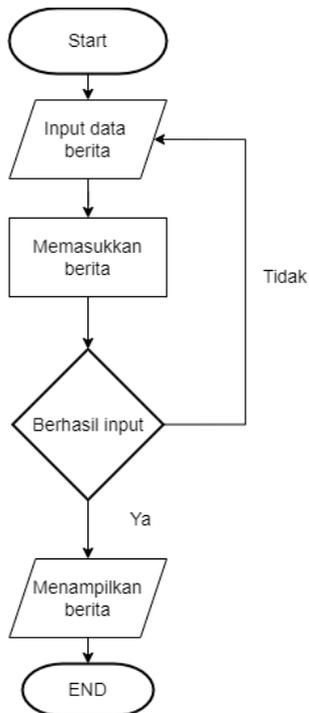
Jalur 1: 1-2-3-4-5-6

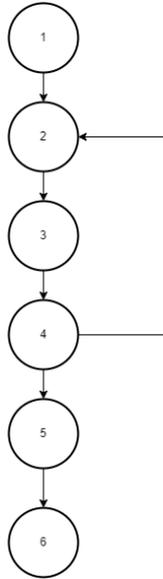
Jalur 2: 1-2-3-4-2-3-4-5-6

Path	1
Jalur	1-2-3-4-5-6
Skenario	<ol style="list-style-type: none"> 1. Start 2. Input data perusahaan mitra 3. Memasukkan data perusahaan mitra 4. Berhasil input 5. Menampilkan data perusahaan mitra 6. End
Hasil Pengujian	Berhasil
Path	2
Jalur	1-2-3-4-2-3-4-5-6
Skenario	<ol style="list-style-type: none"> 1. Start 2. Input data perusahaan mitra 3. Memasukkan data perusahaan mitra 4. Gagal input

	<p>5. Input data perusahaan mitra</p> <p>6. Memasukkan data perusahaan mitra</p> <p>7. Berhasil input</p> <p>8. Menampilkan data perusahaan mitra</p> <p>9. End</p>
Hasil Pengujian	Berhasil

Proses input berita





$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 6 - 6 + 2 \\
 &= 2
 \end{aligned}$$

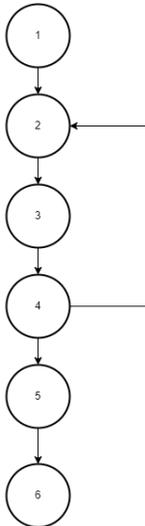
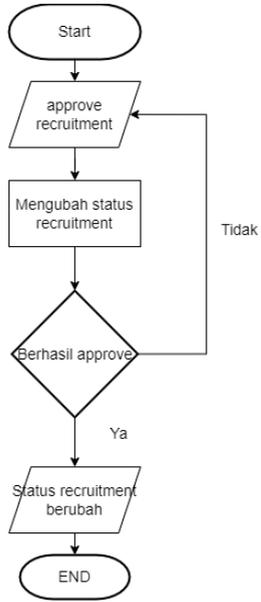
Jalur 1: 1-2-3-4-5-6

Jalur 2: 1-2-3-4-2-3-4-5-6

Path	1
Jalur	1-2-3-4-5-6
Skenario	<ol style="list-style-type: none"> 1. Start 2. Input data berita 3. Memasukkan data berita 4. Berhasil input 5. Menampilkan data berita 6. End

Hasil Pengujian	Berhasil
Path	2
Jalur	1-2-3-4-2-3-4-5-6
Skenario	<ol style="list-style-type: none"> 1. Start 2. Input data berita 3. Memasukkan data berita 4. Gagal input 5. Input data berita 6. Memasukkan data berita 7. Berhasil input 8. Menampilkan data berita 9. End
Hasil Pengujian	Berhasil

Proses approve recruitment



$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 6 - 6 + 2
 \end{aligned}$$

= 2

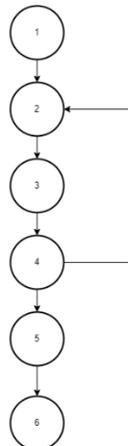
Jalur 1: 1-2-3-4-5-6

Jalur 2: 1-2-3-4-2-3-4-5-6

Path	1
Jalur	1-2-3-4-5-6
Skenario	<ol style="list-style-type: none">1. Start2. Approve recruitment3. Mengubah status recruitment4. Berhasil approve5. Status recruitment berubah6. End
Hasil Pengujian	Berhasil
Path	2
Jalur	1-2-3-4-2-3-4-5-6
Skenario	<ol style="list-style-type: none">1. Start2. Approve recruitment3. Mengubah status recruitment4. Gagal approve5. Approve recruitment6. Mengubah status recruitment7. Berhasil approve8. Status recruitment berubah9. End

Hasil Pengujian	Berhasil
-----------------	----------

Proses pengiriman jadwal



$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 6 - 6 + 2 \\
 &= 2
 \end{aligned}$$

Jalur 1: 1-2-3-4-5-6

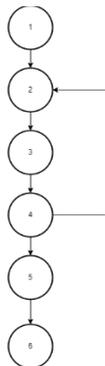
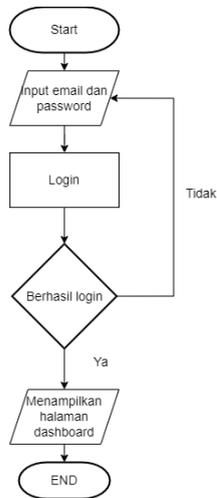
Jalur 2: 1-2-3-4-2-3-4-5-6

Path	1
Jalur	1-2-3-4-5-6
Skenario	<ol style="list-style-type: none"> 1. Start 2. Kirim pesan ke pelamar kerja 3. Mengirim pesan melalui WA 4. Berhasil Mengirim 5. Pesan terkirim ke pelamar kerja 6. End
Hasil Pengujian	Berhasil
Path	2
Jalur	1-2-3-4-2-3-4-5-6
Skenario	<ol style="list-style-type: none"> 1. Start 2. Kirim pesan ke pelamar kerja 3. Mengirim pesan melalui WA 4. Gagal mengirim 5. Kirim pesan ke pelemar kerja 6. Mengirim pesan melalui WA 7. Berhasil mengirim

	8. Pesan terkirim ke pelamar kerja 9. End
Hasil Pengujian	Berhasil

Kasus Uji Role: Perusahaan mitra

Login



$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 6 - 6 + 2 \\
 &= 2
 \end{aligned}$$

Jalur 1: 1-2-3-4-5-6

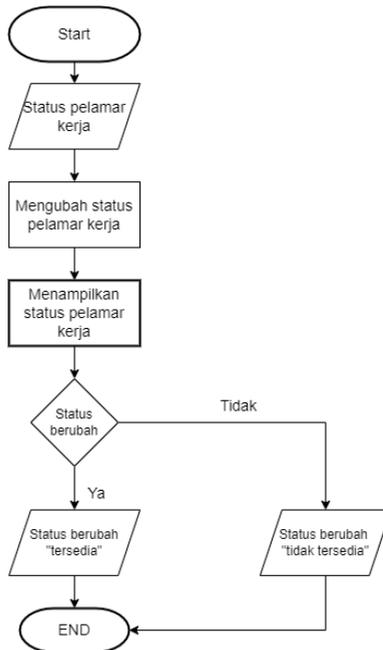
Jalur 2: 1-2-3-4-2-3-4-5-6

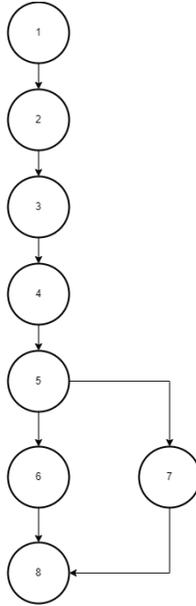
Tabel 1 Case Perusahaan Mitra

Path	1
Jalur	1-2-3-4-5-6
Skenario	<ol style="list-style-type: none"> 1. Start 2. Input email dan password 3. Login 4. Berhasil login 5. Menampilkan dashboard 6. End
Hasil Pengujian	Berhasil
Path	2
Jalur	1-2-3-4-2-3-4-5-6
Skenario	<ol style="list-style-type: none"> 1. Start 2. Input email dan password 3. Login 4. Gagal login 5. Input email dan password 6. Login 7. Berhasil login

	8. Menampilkan dashboard 9. End
Hasil Pengujian	Berhasil

Proses approve recruitment





Gambar 2 *Flowgraph* Perusahaan Mitra

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 8 - 8 + 2 \\
 &= 2
 \end{aligned}$$

Jalur 1: 1-2-3-4-5-6-7-8

Jalur 2: 1-2-3-4-5-7-8

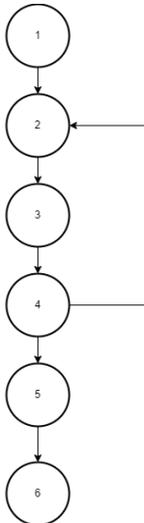
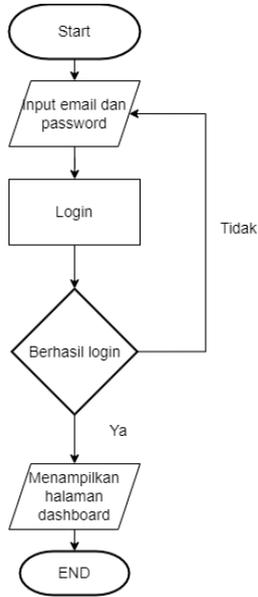
Tabel 1 Case Approve recruitment

Path	1
Jalur	1-2-3-4-5-6-7-8
Skenario	1. Start 2. Status pelamar kerja 3. Mengubah status pelamar kerja 4. Menampilkan status pelamar kerja

	5. Berhasil ubah status 6. Status berubah “tersedia” 7. End
Hasil Pengujian	Berhasil
Path	2
Jalur	1-2-3-4-5-7-8
Skenario	1. Start 2. Status pelamar kerja 3. Mengubah status pelamar kerja 4. Menampilkan status pelamar kerja 5. Berhasil ubah status 6. Status berubah “tidak tersedia” 7. End
Hasil Pengujian	Berhasil

Kasus Uji Role: Pelamar kerja

Login



$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 6 - 6 + 2
 \end{aligned}$$

= 2

Jalur 1: 1-2-3-4-5-6

Jalur 2: 1-2-3-4-2-3-4-5-6

Path	1
Jalur	1-2-3-4-5-6
Skenario	1. Start 2. Input email dan password 3. Login 4. Berhasil login 5. Menampilkan dashboard 6. End
Hasil Pengujian	Berhasil
Path	2
Jalur	1-2-3-4-2-3-4-5-6
Skenario	1. Start 2. Input email dan password 3. Login 4. Gagal login 5. Input email dan password 6. Login 7. Berhasil login 8. Menampilkan dashboard 9. End
Hasil Pengujian	Berhasil

LAMPIRAN D

BLACKBOX TESTING

DOKUMEN PENGUJIAN BLACKBOX

**PERANCANGAN BACKEND PADA SISTEM TENAGA KERJA
MENGUNAKAN REST API
(STUDI KASUS: PT. CITRATAMA INDONESIA)**

PoliTeknik Caltex Riau	PENGUJIAN Perancangan Backend Pada Sistem Tenaga Kerja Menggunakan RestApi(Studi Kasus Pt. Citratama Indonesia)	Hal:
-------------------------------	---	------

PoliTeknik Caltex Riau	Nama	Riandy Enrisa
	NIM	2055301121
	Tanggal: 16 / 0 7 / 24	
	Tempat :	

LEMBAR PENGESAHAN

PENGEMBANG WEBSITE	PEMERIKSA
Diperiapkan Oleh:	Diperiksa dan Dsetujui oleh:
Nama : Riandy Enrisa	Nama : Riandy Enrisa
Posisi : Pengembang website	Posisi : Pengembang website
Tanda Tangan : 	Tanda Tangan : 

Catatan Akhir :

PENGUJIAN		
Pelaksana Catur Riu	Perancangan Backend Pada Sistem Tenaga Kerja Menggunakan RestApi(Studi Kasus Pt. Citirama Indonesia)	Hal:

IDENTIFIKASI DAN HASIL PENGUJIAN

Adapun identifikasi dan hasil pengujian adalah sebagai berikut:

Kasus Uji Role: Admin

Kode	Use Case	Tanggal Pengujian	Hasil Pengujian	Catatan Pengujian
1	Nama Uji Kasus Uji - Email: admin@gmail.com Password: admin	16/07/25	M Berhasil	L Gagal
	Hasil Yang Diharapkan			
	Jika berhasil akan mengenerate token Jika gagal akan menampilkan pesan error			
2	Nama Uji Kasus Uji - Lihat data pelamar kerja	16/07/25	M Berhasil	L Gagal
	Hasil Yang Diharapkan			
	Dapat menampilkan data pelamar kerja			
3	Nama Uji Kasus Uji - Tambah data pelamar kerja	16/07/25	M Berhasil	L Gagal
	Hasil Yang Diharapkan			
	Dapat menambahkan data pelamar kerja Jika berhasil maka data pelamar kerja bertambah akan menampilkan pesan berhasil. Jika gagal maka akan menampilkan pesan gagal.			
4	Nama Uji Kasus Uji - Edit data pelamar kerja	16/07/25	M Berhasil	L Gagal
	Hasil Yang Diharapkan			
	Dapat mengubah data pelamar kerja Jika berhasil maka data pelamar kerja berubah dan akan menampilkan pesan berhasil.			

PENGUJIAN		
Pelaksana Catur Riu	Perancangan Backend Pada Sistem Tenaga Kerja Menggunakan RestApi(Studi Kasus Pt. Citirama Indonesia)	Hal:

Kode	Use Case	Tanggal Pengujian	Hasil Pengujian	Catatan Pengujian
5	Nama Uji Kasus Uji - Hapus data pelamar kerja	16/07/25	M Berhasil	L Gagal
	Hasil Yang Diharapkan			
	Data berhasil di hapus dan hilang dari tabel			
6	Nama Uji Kasus Uji - Lihat data perusahaan mitra	16/07/25	M Berhasil	L Gagal
	Hasil Yang Diharapkan			
	Dapat menampilkan detail informasi dari perusahaan mitra			
7	Nama Uji Kasus Uji - Tambah data perusahaan mitra	16/07/25	M Berhasil	L Gagal
	Hasil Yang Diharapkan			
	Dapat menambahkan data perusahaan mitra Jika berhasil maka data perusahaan mitra bertambah dan akan menampilkan pesan berhasil. Jika gagal maka akan menampilkan pesan gagal			
8	Nama Uji Kasus Uji - Edit data perusahaan mitra	16/07/25	M Berhasil	L Gagal
	Hasil Yang Diharapkan			
	Dapat mengubah data pelamar kerja Jika berhasil maka data pelamar kerja berubah dan akan menampilkan pesan berhasil. Jika gagal maka akan menampilkan pesan gagal.			

PENGUJIAN		
Pelaksana Catur Riu	Perancangan Backend Pada Sistem Tenaga Kerja Menggunakan RestApi(Studi Kasus Pt. Citirama Indonesia)	Hal:

Kode	Use Case	Tanggal Pengujian	Hasil Pengujian	Catatan Pengujian
9	Nama Uji Kasus Uji - Hapus data perusahaan mitra	16/07/25	M Berhasil	L Gagal
	Hasil Yang Diharapkan			
	Data berhasil di hapus dan hilang dari tabel			
10	Nama Uji Kasus Uji - Lihat data berita	16/07/25	M Berhasil	L Gagal
	Hasil Yang Diharapkan			
	Dapat menampilkan detail informasi dari berita			
11	Nama Uji Kasus Uji - Tambah data berita	16/07/25	M Berhasil	L Gagal
	Hasil Yang Diharapkan			
	Dapat menambahkan data berita Jika berhasil maka data berita bertambah dan akan menampilkan pesan berhasil. Jika gagal maka akan menampilkan pesan gagal.			
12	Nama Uji Kasus Uji - Edit data berita	16/07/25	M Berhasil	L Gagal
	Hasil Yang Diharapkan			
	Dapat mengubah data berita Jika berhasil maka data berita berubah dan akan menampilkan pesan berhasil. Jika gagal maka akan menampilkan pesan gagal.			
13	Nama Uji Kasus Uji - Hapus data berita	16/07/25	M Berhasil	L Gagal
	Hasil Yang Diharapkan			
	Data berhasil di hapus dan hilang dari tabel			
14	Nama Uji Kasus Uji - Lihat data FAQ	16/07/25	M Berhasil	L Gagal
	Hasil Yang Diharapkan			
	Dapat menampilkan detail informasi dari FAQ			

Polihefik Catter Ritu	PENGUJIAN Perancangan Backend Pada Sistem Tenaga Kerja Menggunakan RestApi(Studi Kasus Pt. Citrama Indonesia)	Hal:

Kode	Use Case	Tanggal Pengujian	Hasil Pengujian	Catatan Pengujian
15	Hasil Yang Diharapkan	16/07/24	M Berhasil	L Gagal
	Nama Uji			
	Kasus Uji			
16	Hasil Yang Diharapkan	16/07/24	M Berhasil	L Gagal
	Nama Uji			
	Kasus Uji			
17	Hasil Yang Diharapkan	16/07/24	M Berhasil	L Gagal
	Nama Uji			
	Kasus Uji			
18	Hasil Yang Diharapkan	16/07/24	M Berhasil	L Gagal
	Nama Uji			
	Kasus Uji			
19	Hasil Yang Diharapkan	16/07/24	M Berhasil	L Gagal
	Nama Uji			
	Kasus Uji			

Polihefik Catter Ritu	PENGUJIAN Perancangan Backend Pada Sistem Tenaga Kerja Menggunakan RestApi(Studi Kasus Pt. Citrama Indonesia)	Hal:

Kode	Use Case	Tanggal Pengujian	Hasil Pengujian	Catatan Pengujian
20	Hasil Yang Diharapkan	16/07/24	M Berhasil	L Gagal
	Nama Uji			
	Kasus Uji			
21	Hasil Yang Diharapkan	16/07/24	M Berhasil	L Gagal
	Nama Uji			
	Kasus Uji			
22	Hasil Yang Diharapkan	16/07/24	M Berhasil	L Gagal
	Nama Uji			
	Kasus Uji			

Kasus Uji Role: Perusahaan mitra

Kode	Use Case	Tanggal Pengujian	Hasil Pengujian	Catatan Pengujian
1	Nama Uji Login Kasus Uji Email: p1rma@ptcitra1.com Password: 12345678	16/07/24	M Berhasil	L Gagal

Polihefik Catter Ritu	PENGUJIAN Perancangan Backend Pada Sistem Tenaga Kerja Menggunakan RestApi(Studi Kasus Pt. Citrama Indonesia)	Hal:

Kode	Use Case	Tanggal Pengujian	Hasil Pengujian	Catatan Pengujian
2	Hasil Yang Diharapkan	16/07/24	M Berhasil	L Gagal
	Nama Uji			
	Kasus Uji			
3	Hasil Yang Diharapkan	16/07/24	M Berhasil	L Gagal
	Nama Uji			
	Kasus Uji			
4	Hasil Yang Diharapkan	16/07/24	M Berhasil	L Gagal
	Nama Uji			
	Kasus Uji			
5	Hasil Yang Diharapkan	16/07/24	M Berhasil	L Gagal
	Nama Uji			
	Kasus Uji			
6	Hasil Yang Diharapkan	16/07/24	M Berhasil	L Gagal
	Nama Uji			
	Kasus Uji			

Polihealk Caltex Riau	PENGUJIAN	Hal:
	Perancangan Backend Pada Sistem Tenaga Kerja Menggunakan RestApi(Studi Kasus Pt. Citrama Indonesia)	

Kode	Use Case	Tanggal Pengujian	Hasil Pengujian	Catatan Pengujian
	Kasus Uji - Hasil Yang Diharapkan Dapat menampilkan detail informasi dari pelamar kerja yang sudah direvisi			
7	Nama Uji Kasus Uji Hasil Yang Diharapkan Penerimaan pelamar kerja Menekan tombol centang atau silang Dapat mengubah status pelamar kerja menjadi "tidak tersedia" atau "tersedia."	16/07/20	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	

Kasus Uji Role: Pelamar kerja

Kode	Use Case	Tanggal Pengujian	Hasil Pengujian	Catatan Pengujian
	Nama Uji Kasus Uji Hasil Yang Diharapkan Login Email: mudi@igmail.com Password: 12345678 Jika berhasil maka akan mengenerate token Jika gagal maka akan menampilkan pesan gagal	16/07/20	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	
	Nama Uji Kasus Uji Hasil Yang Diharapkan Lihat data pelamar kerja	16/07/20	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	
	Nama Uji Kasus Uji Hasil Yang Diharapkan Dapat menampilkan detail informasi dari pelamar kerja			
	Nama Uji Kasus Uji Hasil Yang Diharapkan Tambah data diri Mengisi form data diri	16/07/20	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	
	Nama Uji Kasus Uji Hasil Yang Diharapkan Dapat menambahkan data diri Jika berhasil maka akan menampilkan pesan berhasil			

Polihealk Caltex Riau	PENGUJIAN	Hal:
	Perancangan Backend Pada Sistem Tenaga Kerja Menggunakan RestApi(Studi Kasus Pt. Citrama Indonesia)	

Kode	Use Case	Tanggal Pengujian	Hasil Pengujian	Catatan Pengujian
	Kasus Uji - Hasil Yang Diharapkan Jika gagal maka akan menampilkan pesan gagal Jika berhasil maka akan menampilkan pesan berhasil			
4	Nama Uji Kasus Uji Hasil Yang Diharapkan Mengisi form data diri Dapat mengubah data diri Jika berhasil maka akan menampilkan pesan berhasil Jika gagal maka akan menampilkan pesan gagal	16/07/20	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	
5	Nama Uji Kasus Uji Hasil Yang Diharapkan Chatbot Dapat memberikan pertanyaan dan bot memberikan jawaban	16/07/20	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	
6	Nama Uji Kasus Uji Hasil Yang Diharapkan Lihat notifikasi Dapat menampilkan detail informasi dari notifikasi	16/07/20	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	