

TUGAS AKHIR

**RANCANG BANGUN ALAT PENGUJIAN OTOMATIS API BERBASIS
SPESIFIKASI API DENGAN PENDEKATAN FUZZ TESTING MENGGUNAKAN
METODE EXTREME PROGRAMMING**



DISUSUN OLEH:

UMAR JIHAD

NIM. 22106050029

**STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA**

**PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA
YOGYAKARTA**

2026

LEMBAR PENGESAHAN TUGAS AKHIR



KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Marsda Adisucipto Telp. (0274) 540971 Fax. (0274) 519739 Yogyakarta 55281

PENGESAHAN TUGAS AKHIR

Nomor : B-906/Un.02/DST/PP.00.9/05/2026

Tugas Akhir dengan judul : Rancang Bangun Alat Pengujian Otomatis API Berbasis Spesifikasi API dengan Pendekatan Fuzz Testing Menggunakan Metode Extreme Programming

yang dipersiapkan dan disusun oleh:

Nama : UMAR JIHAD
Nomor Induk Mahasiswa : 22106050029
Telah diujikan pada : Senin, 20 April 2026
Nilai ujian Tugas Akhir : A

dinyatakan telah diterima oleh Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta

TIM UJIAN TUGAS AKHIR



Valid ID: 6a0a8c9333ff2

Ketua Sidang
Dr. Agung Fatwanto, S.Si., M.Kom.
SIGNED



Valid ID: 6a0136cc09d41

Penguji I
Muhammad Mustakim, S.T. M.T.
SIGNED



Valid ID: 6a02ac35599f8

Penguji II
Eko Hadi Gunawan, M.Eng.
SIGNED



Valid ID: 6a0a9ba8e927c

Yogyakarta, 20 April 2026
UIN Sunan Kalijaga
Dekan Fakultas Sains dan Teknologi
Prof. Dr. Dra. Hj. Khurul Wardati, M.Si.
SIGNED

LEMBAR KEASLIAN TUGAS AKHIR

SURAT PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini:

Nama : Umar Jihad
NIM : 22106050029
Program Studi : Informatika
Fakultas : Sains dan Teknologi

Menyatakan dengan sesungguhnya, bahwa skripsi saya yang berjudul **Rancang Bangun Alat Pengujian Otomatis API Berbasis Spesifikasi API dengan Pendekatan *Fuzz Testing* Menggunakan Metode *Extreme Programming*** merupakan hasil pekerjaan penulis sendiri sepanjang pengetahuan penulis, bukan duplikasi atau saduran dari karya orang lain kecuali bagian tertentu yang penulis ambil sebagai bahan acuan. Apabila terbukti pernyataan ini tidak benar, sepenuhnya menjadi tanggung jawab penulis.

Yogyakarta, 6 April 2026



Umar Jihad
NIM. 22106050029

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

LEMBAR PERSETUJUAN TUGAS AKHIR



Universitas Islam Negeri Sunan Kalijaga



FM-UINSK-BM-05-03/R0

SURAT PERSETUJUAN SKRIPSI/TUGAS AKHIR

Hal : Persetujuan Skripsi / Tugas Akhir
Lamp : -

Kepada
Yth. Dekan Fakultas Sains dan Teknologi
UIN Sunan Kalijaga Yogyakarta
di Yogyakarta

Assalamu'alaikum wr. wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengadakan perbaikan seperlunya, maka kami selaku pembimbing berpendapat bahwa skripsi Saudara:

Nama : Umar Jihad
NIM : 22106050029
Judul Skripsi : Rancang Bangun Alat Pengujian Otomatis API Berbasis Spesifikasi API dengan Pendekatan *Fuzz Testing* Menggunakan Metode *Extreme Programming*

sudah dapat diajukan kembali kepada Program Studi Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu dalam Program Studi Informatika.

Dengan ini kami mengharap agar skripsi/tugas akhir Saudara tersebut di atas dapat segera dimunaqasyahkan. Atas perhatiannya kami ucapkan terima kasih.

Wassalamu'alaikum wr. wb.

Yogyakarta, 6 April 2026

Pembimbing

Dr. Agung Fatwanto, S.Si., M.Kom.

NIP. 19770103 200501 1 003

INTISARI

API (*Application Programming Interface*) menjadi elemen penting dalam pembangunan perangkat lunak masa kini karena berperan sebagai jembatan komunikasi antar sistem. Tingginya intensitas pemakaian API mendorong kebutuhan akan pengujian yang memadai guna memverifikasi bahwa setiap *endpoint* bekerja sesuai spesifikasi yang ditetapkan. Sayangnya, pendekatan pengujian manual kerap menyita banyak waktu dan rentan terhadap kekeliruan. Pengembangan ini berupaya merancang dan membangun sebuah alat pengujian API berbasis otomatisasi yang dapat menyusun dan menjalankan skenario pengujian secara mandiri berdasarkan dokumen *OpenAPI Specification* (OAS). Alat ini menganalisis dokumen konfigurasi serta dokumen OAS, lalu secara otomatis menghasilkan beragam skenario. Di samping itu, teknik *fuzz testing* diterapkan untuk menciptakan data uji yang menyimpang dari spesifikasi demi mengukur ketangguhan API dalam menghadapi masukan yang tidak semestinya. Pengembangan ini membuktikan bahwa alat yang dihasilkan mampu mengotomasi seluruh siklus pengujian API sekaligus menyajikan laporan hasil pengujian secara terstruktur.

Kata kunci: Pengujian Otomatis, *OpenAPI Specification*, Pengujian API, *Fuzz Testing*.

ABSTRACT

APIs (Application Programming Interfaces) serve as critical building blocks in contemporary software development by facilitating seamless communication across systems. The growing adoption of APIs has heightened the need for rigorous testing to verify that each endpoint behaves in accordance with its defined specifications. However, manual testing approaches are frequently inefficient and susceptible to human error. This study sets out to design and build an automated API testing tool that independently constructs and executes test scenarios derived from OpenAPI Specification (OAS) documents. The tool parses both configuration files and OAS documents, then subsequently generates a comprehensive suite of test scenarios. Furthermore, fuzzing techniques are employed to synthesize non-conforming test data, thereby assessing the API's resilience against unexpected or malformed inputs. The findings confirm that the proposed tool successfully automates the full API testing lifecycle and delivers well-structured testing reports.

Keywords: API testing, OpenAPI Specification, automated testing, fuzzing.

HALAMAN MOTTO

مَا أَصَابَ مِنْ مُصِيبَةٍ فِي الْأَرْضِ وَلَا فِي أَنْفُسِكُمْ إِلَّا فِي كِتَابٍ مِّن قَبْلِ أَنْ نَبْرَأَهَا ۗ إِنَّ ذَلِكَ عَلَى اللَّهِ يَسِيرٌ ۚ ۲۲

“Tidak ada bencana (apa pun) yang menimpa di bumi dan tidak (juga yang menimpa) dirimu, kecuali telah tertulis dalam Kitab (Lauhul Mahfuz) sebelum Kami mewujudkannya. Sesungguhnya hal itu mudah bagi Allah.”

QS. Al-Hadid 57: Ayat 22

لَّكَيْلًا تَأْسَوْا عَلَىٰ مَا فَاتَكُمْ وَلَا تَفْرَحُوا بِمَا آتَاكُمْ ۗ وَاللَّهُ لَا يُحِبُّ كُلَّ مُخْتَالٍ فَخُورٍ ۚ ۲۳

(Yang demikian itu kami tetapkan) agar kamu tidak bersedih terhadap apa yang luput dari kamu dan tidak pula terlalu gembira terhadap apa yang diberikan-Nya kepadamu. Allah tidak menyukai setiap orang yang sombong lagi membanggakan diri.

QS. Al-Hadid 57: Ayat 23

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

HALAMAN PERSEMBAHAN

Dalam mengerjakan tugas akhir ini, hal utama bagi penulis adalah meniatkannya *lillahi ta'ala*. Dengan niat itu pula penulis dapat menyelesaikan tugas akhir ini dan mempersembahkannya khusus untuk:

Kedua Orang Tua Tercinta

Kakak Saya

Almamater Tercinta

Program Studi Informatika

Fakultas Sains dan Teknologi

UIN Sunan Kalijaga Yogyakarta

Semua orang yang pernah membuat Saya dari tidak tahu menjadi tahu



STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

LEMBAR PEDOMAN PENGGUNAAN TUGAS AKHIR

Tugas akhir ini tidak dipublikasikan, tetapi tersedia di perpustakaan dalam lingkungan UIN Sunan Kalijaga, yang diperkenankan dipakai sebagai referensi kepustakaan, tetapi pengutipan harus seizin penyusun, dan harus menyebutkan sumbernya sesuai dengan kebiasaan ilmiah. Dokumen tugas akhir ini merupakan hak milik UIN Sunan Kalijaga Yogyakarta.



KATA PENGANTAR

Alhamdulillah Robbil ‘Alamin, penulis menyampaikan rasa syukur yang mendalam kepada Allah *Subhanahu wata’ala* atas limpahan rahmat-Nya, sehingga penulis dapat menyelesaikan tugas akhir dengan judul “Rancang Bangun Alat Pengujian Otomatis API Berbasis Spesifikasi API dengan Pendekatan *Fuzz Testing* Menggunakan Metode *Extreme Programming*”.

Tugas akhir ini merupakan hasil dari proses yang panjang dan menantang, yang menguji ketekunan, konsistensi, serta semangat belajar menulis. Segala pencapaian dalam proses ini tidak lepas dari bimbingan, doa, dan dukungan banyak pihak yang telah memberikan kontribusi baik secara langsung maupun tidak langsung.

Untuk itu, penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Prof. Noorhaidi, S.Ag., M.A., M.Phil., Ph.D., selaku Rektor UIN Sunan Kalijaga Yogyakarta.
2. Prof. Dr. Dra. Hj. Khurul Wardati, M.Si., selaku Dekan Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta.
3. Dr. Muhammad Mustakim, S.T., M.T., selaku Ketua Program Studi Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta.
4. Agus Mulyanto, S.Si., M.Kom., selaku dosen penasihat akademik yang telah membimbing penulis selama masa studi.
5. Dr. Agung Fatwanto, S.Si., M.Kom., selaku pembimbing tugas akhir yang dengan sabar memberikan arahan, masukan, dan dukungan dalam proses penyusunan tugas akhir ini.
6. Seluruh dosen dan staf Program Studi Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta yang telah memberikan ilmu dan bantuan selama masa perkuliahan.
7. Kedua orang tua tercinta, atas segala doa, kasih sayang, serta dukungan yang tak pernah henti hingga penulis dapat menyelesaikan pendidikan tinggi ini.
8. Rekan-rekan seperjuangan Informatika angkatan 2022 yang telah menjadi bagian dari perjalanan akademik penulis, khususnya dalam memberikan semangat dan kebersamaan selama kuliah.

9. Terakhir, penulis ingin mengucapkan terima kasih kepada penulis yaitu diriku sendiri, Umar Jihad. Terima kasih telah bertahan sejauh ini, dan terus berjalan melewati segala tantangan yang semesta hadirkan.

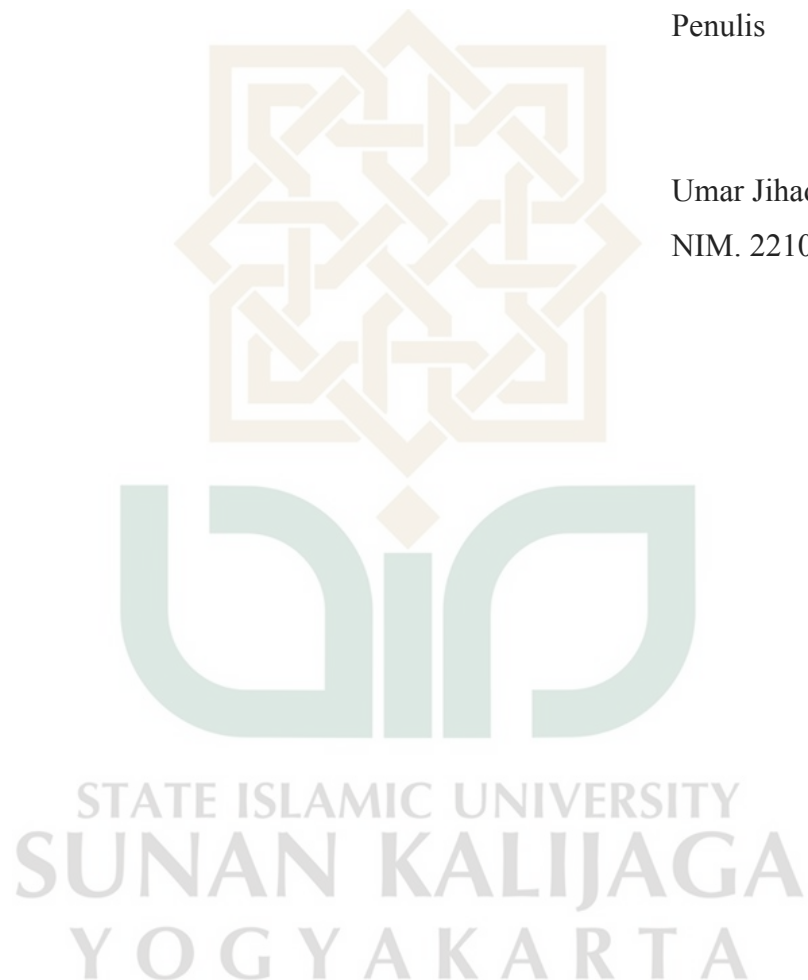
Tugas akhir ini pasti tidak terlepas dari kekurangan dan kesalahan. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun guna perbaikan di masa yang akan datang, Semoga tugas akhir ini dapat memberikan manfaat dan dapat dikembangkan untuk kedepannya.

Yogyakarta, 4 Maret 2026

Penulis

Umar Jihad

NIM. 22106050029



DAFTAR ISI

HALAMAN SAMBUL.....	i
LEMBAR PENGESAHAN TUGAS AKHIR.....	ii
LEMBAR KEASLIAN TUGAS AKHIR.....	iii
LEMBAR PERSETUJUAN TUGAS AKHIR.....	iv
INTISARI.....	v
HALAMAN PERSEMBAHAN.....	vii
LEMBAR PEDOMAN PENGGUNAAN TUGAS AKHIR.....	viii
KATA PENGANTAR.....	ix
DAFTAR ISI.....	xi
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR.....	xiv
DAFTAR LAMPIRAN.....	xv
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	3
1.3. Batasan Masalah.....	3
1.4. Tujuan Pengembangan.....	3
1.5. Manfaat Pengembangan.....	4
BAB II TINJAUAN PUSTAKA.....	5
2.1. Application Programming Interface (API).....	5
2.2. Pengujian API.....	5
2.3. Spesifikasi API.....	6
2.4. Fuzz Testing.....	8
2.5. Extreme Programming (XP).....	9
2.6. Analisis Alat Serupa.....	10
BAB III METODE PENGEMBANGAN SISTEM.....	12
3.1. Alat dan Bahan.....	12
3.2. Metode Pengembangan.....	12
BAB IV PERANCANGAN DAN IMPLEMENTASI SISTEM.....	15
4.1. Perencanaan (Planning).....	15
4.1.1. Analisis Kebutuhan Fungsional.....	15
4.1.2. Analisis Kebutuhan Non-fungsional.....	16
4.2. Perancangan (Design).....	17
4.2.1. Arsitektur Sistem.....	18
4.2.2. Alur Kerja Sistem.....	19
4.2.3. Pembangkitan Skenario Pengujian.....	20
4.2.4. Pembangkitan Fuzz Data.....	21
4.2.5. Format Dokumen OAS dan Dokumen Konfigurasi.....	23
4.2.6. Format Penyajian Laporan.....	25
4.3. Implementasi (Coding).....	26

4.3.1. Iterasi Pertama.....	27
4.3.2. Iterasi Kedua.....	29
4.3.3. Iterasi Ketiga.....	30
4.3.4. Iterasi Keempat.....	31
4.4. Pengujian (Testing).....	33
4.4.1. White Box Testing.....	34
4.4.2. Black Box Testing.....	35
4.4.3. Performance Testing.....	36
BAB V PENUTUP.....	38
5. 1. Kesimpulan.....	38
5. 2. Saran.....	38
DAFTAR PUSTAKA.....	40
LAMPIRAN.....	43



DAFTAR TABEL

Tabel 2.1. Contoh Penulisan Dokumen OAS.....	7
Tabel 2.2. Hasil eksplorasi alat pengujian otomatis API.....	10
Tabel 4.1. Daftar fungsi pada komponen configReader.....	29
Tabel 4.2. Daftar fungsi pada komponen oasReader.....	30
Tabel 4.3. Daftar fungsi pada komponen dataGenerator.....	30
Tabel 4.4. Daftar skenario pengujian yang dapat dibangkitkan oleh komponen scenario.....	31
Tabel 4.5. Daftar fungsi pada komponen runner.....	32
Tabel 4.6. Daftar fungsi pada komponen assertion.....	33
Tabel 4.7. Skenario dan hasil pengujian pembangkitan data valid.....	35
Tabel 4.8. Skenario dan hasil pengujian pembangkitan data invalid.....	35
Tabel 4.9. Skenario dan hasil black box testing.....	36
Tabel 4.10. Skenario dan hasil performance testing.....	37

DAFTAR GAMBAR

Gambar 2.1. Tahapan Fuzz Testing.....	8
Gambar 3.1. Tahapan Metode XP.....	13
Gambar 4.1. Diagram Komponen.....	19
Gambar 4.2. Data Flow Diagram.....	20
Gambar 4.3. Algoritma pembangkitan skenario pengujian.....	21
Gambar 4.4. Algoritma pembangkitan fuzz data.....	22
Gambar 4.5. Contoh isi dokumen konfigurasi.....	25
Gambar 4.6. Contoh skenario yang yang berhasil.....	26
Gambar 4.7. Contoh skenario yang gagal.....	26
Gambar 4.8. Daftar dependensi pada berkas package.json.....	28
Gambar 4.9. Struktur direktori dan berkas sistem.....	28
Gambar 4.10. Hasil white box testing.....	34

DAFTAR LAMPIRAN

Lampiran 1 Ringkasan Tugas Akhir

Lampiran 2 Daftar Singkatan



BAB I

PENDAHULUAN

1.1. Latar Belakang

Pada era saat ini, pengujian perangkat lunak sangat bergantung pada penerapan otomatisasi pengujian. Seiring dengan meningkatnya kompleksitas perangkat lunak, kebutuhan akan proses pengujian yang mampu meminimalkan penggunaan sumber daya juga semakin tinggi. Dari sisi konsumsi sumber daya, kecepatan eksekusi, dan cakupan pengujian, metode pengujian manual dinilai kurang efisien. Permasalahan-permasalahan tersebut merupakan aspek yang dapat diatasi melalui otomatisasi pengujian, yang pada sebagian besar kasus terbukti mampu meningkatkan efektivitas proses pengujian perangkat lunak [1]. Salah satu area pengembangan perangkat lunak yang sangat membutuhkan penerapan otomatisasi pengujian adalah pengembangan API.

Dalam beberapa tahun terakhir, terjadi peningkatan yang sangat pesat dalam ketersediaan API yang memungkinkan akses terhadap berbagai layanan melalui antarmuka aplikasi. Popularitas ini didorong oleh sejumlah tren industri, seperti berkembangnya *cloud computing* serta adopsi arsitektur *microservices* secara luas [2]. Sebagian besar API modern menerapkan gaya arsitektur *REpresentational State Transfer* (REST) dimana API menggunakan aturan *client-server*. Teknik ini menjadi pendekatan terstandarisasi yang digunakan secara luas dalam penyediaan layanan melalui jaringan Internet [4, 5]. Tingginya tingkat adopsi REST API tercermin dari tersedianya ribuan API yang terdaftar dalam berbagai direktori publik.

Saat membangun sistem *backend* pada sebuah aplikasi, pengujian API menjadi semakin krusial karena adanya cacat atau kesalahan pada API dapat menimbulkan dampak yang signifikan, baik terhadap layanan internal yang bergantung pada API tersebut maupun terhadap aplikasi pihak ketiga serta pengguna akhir [1]. Oleh sebab itu, kualitas dan keandalan API harus dijaga melalui proses pengujian yang sistematis dan berkelanjutan. Penerapan pengujian otomatis pada API juga berperan penting dalam mendeteksi kesalahan sejak dini serta meminimalkan risiko gangguan layanan pada tahap implementasi maupun pemeliharaan sistem.

Meskipun demikian, proses verifikasi terhadap kebenaran dan fungsionalitas API merupakan tantangan yang cukup kompleks. Seorang penguji tidak hanya

dituntut untuk membangun dan mengirimkan pesan jaringan ke API, misalnya melalui protokol HTTP di atas TCP, tetapi juga perlu menyiapkan data yang sesuai pada basis data serta, dalam beberapa kasus, melakukan *mocking* terhadap interaksi dengan layanan eksternal lainnya [3]. Kompleksitas tersebut menyebabkan proses pengujian API menjadi lebih memakan waktu dan rentan terhadap kesalahan karena dilakukan secara manual. Oleh karena itu, diperlukan pendekatan pengujian yang lebih terstruktur dan otomatis untuk meningkatkan efisiensi serta konsistensi hasil pengujian.

Pengujian API juga sering terkendala karena kurangnya pemanfaatan spesifikasi API secara optimal. *OpenAPI Specification* (OAS) merupakan format yang paling banyak digunakan dalam pendokumentasian API [6]. OAS mendefinisikan standar antarmuka dan deskripsi layanan yang bersifat umum serta dapat diterapkan pada berbagai bahasa pemrograman. OAS memungkinkan baik pengguna maupun sistem otomatis untuk memahami kapabilitas suatu layanan tanpa perlu mengakses kode sumber. Dengan adanya deskripsi layanan yang disusun secara tepat menggunakan OAS, pengguna API dapat memahami fungsi dan karakteristik setiap operasi yang tersedia serta berinteraksi dengan API tersebut tanpa harus mengetahui detail logika implementasinya. Secara teknis, setiap operasi direpresentasikan sebagai sebuah *endpoint* berupa *Uniform Resource Identifier* (URI) yang dipadukan dengan metode HTTP tertentu [7]. Setiap kombinasi *endpoint* dan metode HTTP memiliki beragam kemungkinan *test case* yang perlu diuji. Untuk dapat menghasilkan data uji yang bervariasi dan sesuai dengan karakteristik setiap *test case* tersebut, diperlukan suatu pendekatan pengujian yang mampu membangkitkan berbagai variasi input secara otomatis. Salah satu teknik yang dapat diimplementasikan untuk tujuan tersebut adalah *fuzz testing*. Teknik ini diterapkan dengan menghasilkan berbagai variasi input pada setiap *endpoint* dan metode HTTP, seperti parameter dan *payload*. Prinsip utama dari *fuzz testing* adalah membangkitkan sejumlah besar *test case* secara otomatis menggunakan data masukan yang tidak valid untuk mengungkap potensi kesalahan yang sulit terdeteksi melalui metode pengujian tradisional, di mana setiap skenario uji umumnya dirancang secara manual oleh penguji [8].

Berdasarkan permasalahan tersebut, diperlukan suatu alat pengujian otomatis yang mampu memanfaatkan spesifikasi API sebagai dasar dalam menghasilkan dan mengeksekusi pengujian secara sistematis. Pendekatan *fuzz testing* berbasis spesifikasi API diharapkan dapat membantu mengotomatisasi proses pembangunan

data uji yang beragam pada setiap endpoint API, sehingga pengujian dapat dilakukan secara lebih menyeluruh, efisien, dan konsisten. Dengan adanya alat pengujian otomatis ini, proses verifikasi fungsionalitas API dapat dilakukan secara lebih efektif untuk mendukung peningkatan kualitas API yang dikembangkan.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang diuraikan, maka rumusan masalah yang akan menjadi fokus penelitian antara lain:

1. Bagaimana merancang dan membangun alat pengujian otomatis API yang mampu melakukan pengujian secara dinamis berdasarkan spesifikasi API?
2. Bagaimana menerapkan pendekatan *fuzz testing* untuk menghasilkan data uji yang digunakan dalam pengujian otomatis API?
3. Bagaimana menerapkan metode *extreme programming* dalam merancang dan membangun alat pengujian otomatis API?

1.3. Batasan Masalah

Agar pembahasan dalam penelitian ini tetap terfokus dan sesuai dengan tujuan yang ingin dicapai, maka ditetapkan beberapa batasan masalah sebagai berikut:

1. API yang menjadi objek pengujian dalam penelitian ini dibatasi pada API yang menerapkan gaya arsitektur REpresentational State Transfer (REST).
2. Dokumen spesifikasi API yang digunakan sebagai dasar pengujian dibatasi pada format OpenAPI Specification (OAS).
3. Pengujian dilakukan berdasarkan daftar respons dari pasangan endpoint dan metode HTTP yang didefinisikan dalam dokumen OAS.

1.4. Tujuan Pengembangan

Berdasarkan rumusan masalah di atas, maka tujuan yang ingin diperoleh dari pelaksanaan penelitian ini adalah merancang dan membangun alat pengujian otomatis API berbasis spesifikasi API dengan pendekatan *fuzz testing* menggunakan metode *extreme programming* yang dapat mempermudah proses pengujian API.

1.5. Manfaat Pengembangan

Manfaat dari pengembangan ini ada dua hal, yaitu sebagai berikut:

1. Bagi Pengembang
 - a. Meningkatkan pemahaman dan penguasaan terhadap konsep pengujian perangkat lunak, khususnya pengujian otomatis pada API.
 - b. Mengembangkan kemampuan dalam menganalisis permasalahan pengujian API serta merancang solusi berbasis spesifikasi API.
2. Bagi Pengguna
 - a. Memberikan kemudahan bagi pengembang perangkat lunak dalam melakukan pengujian API secara otomatis.
 - b. Membantu dalam mendeteksi potensi kesalahan, kegagalan, atau perilaku tidak terduga pada API melalui alat pengujian otomatis.
 - c. Mengurangi waktu dan upaya yang diperlukan dalam proses pengujian API, sehingga dapat meningkatkan efisiensi pengembangan dan kualitas layanan yang dihasilkan.

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan serangkaian proses perancangan dan pembangunan yang telah dijalankan, penulis berhasil mewujudkan sebuah alat pengujian otomatis untuk API yang berlandaskan dokumen *OpenAPI Specification* (OAS). Alat tersebut mampu membaca dokumen konfigurasi dan OAS guna mengenali berbagai komponen API. Dari informasi yang diperoleh, sistem secara otomatis menyusun bermacam skenario pengujian, dengan mengandalkan teknik *fuzz testing* untuk memproduksi variasi data uji yang tidak sesuai spesifikasi.

Teknik *fuzz testing* berhasil diintegrasikan ke dalam proses pembangkitan data uji dengan menghasilkan variasi input tidak valid, baik pada parameter maupun request body. Penerapan teknik ini memungkinkan sistem menjangkau berbagai potensi kesalahan yang umumnya sulit terdeteksi melalui pengujian manual, sehingga cakupan dan kedalaman pengujian menjadi jauh lebih komprehensif.

Di sisi pengembangan, metode Extreme Programming (XP) berhasil diterapkan secara efektif melalui sejumlah iterasi yang mencakup tahap perencanaan, perancangan, implementasi, dan evaluasi. Pendekatan ini memastikan proses pembangunan sistem berjalan secara bertahap dan terstruktur, mulai dari pembacaan dokumen konfigurasi dan OAS, penyusunan skenario pengujian, hingga eksekusi dan penyajian laporan hasil pengujian. Evaluasi menggunakan white box testing, black box testing, dan performance testing membuktikan bahwa seluruh fungsi utama sistem beroperasi sesuai rancangan dan mampu menjalankan skenario pengujian dengan efisien. Dengan demikian, alat yang dikembangkan terbukti efektif dalam menjadikan proses pengujian API lebih sistematis dan konsisten, sekaligus menekan ketergantungan pada pendekatan pengujian manual.

5.2. Saran

Melewati proses pengembangan yang tidak mudah, penulis akhirnya berhasil merancang dan membangun alat pengujian otomatis API berbasis spesifikasi API dengan pendekatan *fuzz testing*. Selama pelaksanaan proses pengembangan sampai terciptanya hasil akhir berupa perangkat lunak alat pengujian, masih ditemukan

berbagai kekurangan dan kelemahan. Oleh karena itu, untuk kebaikan pengembangan sistem selanjutnya, maka penulis memberikan beberapa saran yang ditujukan bagi pengembang yang ingin mengembangkan alat ini lebih lanjut, yaitu:

1. Menambahkan dukungan terhadap tipe *request body* menjadi lebih banyak agar cakupan pengujian menjadi lebih luas.
2. Mengembangkan dukungan terhadap skema autentikasi yang lebih kompleks, seperti OAuth 2.0, sehingga alat dapat digunakan pada sistem yang memiliki kebutuhan keamanan yang lebih tinggi.
3. Menambahkan mekanisme *dependency handling* antar skenario pengujian, seperti pengambilan nilai dari respons sebelumnya untuk digunakan pada *endpoint* lain. Dengan adanya fitur ini, alat akan mampu menguji alur proses API yang lebih kompleks dan mendekati kondisi penggunaan nyata.

DAFTAR PUSTAKA

- [1] Naavainhna Mudiyansele, I. (2024). AI-Driven REST API Testing API Automation and Testing. *Lappeenranta–Lahti University of Technology LUT Master’s Programme in Software Engineering and Digital Transformation*.
- [2] Kim, M., Xin, Q., Sinha, S., & Orso, A. (2022). Automated test generation for REST APIs: no time to rest yet. *ISSTA 2022: Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis*, 22, 289–301. <https://doi.org/10.1145/3533767.3534401>
- [3] Golmohammadi, A., & Zhang, M. (2023). Testing RESTful APIs: A Survey. *ACM Transactions on Software Engineering and Methodology*, 33(1). <https://doi.org/10.1145/3617175>
- [4] Ehsan, A., Abuhaliqa, M. A. M. E., Catal, C., & Mishra, D. (2022). RESTful API Testing Methodologies: Rationale, Challenges, and Solution Directions. *Applied Sciences* 2022, 12(9). <https://doi.org/10.3390/APP12094369>
- [5] Marculescu, B., Zhang, M., & Arcuri, A. (2022). On the faults found in REST APIs by Automated Test Generation. *Automated Test Generation. ACM Trans. Softw. Eng. Methodol.*, 31(3), 41. <https://doi.org/10.1145/3491038>
- [6] Lazar, K., Vetzler, M., Kate, K., Tsay, J., Boaz, D., Gupta, H., Shinnar, A., Vallam, R. D., Amid, D., Goldbraich, E., Uziel, G., Laredo, J., & Tavor, A. A. (2025). Generating OpenAPI Specifications from Online API Documentation with Large Language Models. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 237–253. <https://doi.org/10.18653/V1/2025.ACL-INDUSTRY.18>
- [7] Muhamad, W., Suhardi, & Bandung, Y. (2022). Transforming OpenAPI Specification 3.0 documents into RDF-based semantic web services. *Journal of Big Data*, 9(1), 55-. <https://doi.org/10.1186/S40537-022-00600-8>
- [8] Tokos, A. (2023). Evaluating fuzzing tools for automated testing of REST APIs using OpenAPI specification. *Department of Information Technologies Faculty of Science and Engineering Abo Akademi University*. <https://www.doria.fi/handle/10024/187395>
- [9] Gupta, D., Shrivastava, A., Jaggi, I., & Katoch, N. (2021). A Systematic Review on Extreme Programming. *Journal of Physics: Conference Series*, 1969. <https://doi.org/10.1088/1742-6596/1969/1/012046>
- [10] Ardiansah, T., Rahmanto, Y., & Amir, Z. (2023). Penerapan Extreme Programming Dalam Sistem Informasi Akademik SDN Kuala Teladas. *Journal of Information Technology, Software Engineering and Computer Science (ITSECS)*, 1, 44–51. <https://ejournal.techcart-press.com/index.php/itsecs/article/view/25/26>
- [11] Mukti, A., Widayanto, Mukmin, C., & Kasih, E. (2022). Perancangan Smart Home Menggunakan Konsep Internet of Things (IOT) Berbasis Microcontroller. *Jurnal JUPITER*, 14(2), 516–522. <https://jurnal.polsri.ac.id/index.php/jupiter/article/view/5600/2214>
- [12] Supriyanto, K., Aziiz, D., & Permatasari, H. (2023). Pengujian White Box Sistem Informasi Kasir Toko A Berbasis Web. *Prosiding Seminar Nasional Teknologi*

- Informasi Dan Bisnis (SENATIB)*, 519–522.
<https://ojs.uadb.ac.id/Senatib/article/view/3234/2641>
- [13] Shadiq, J., Safei, A., & Loly, R. W. R. (2021). Pengujian Aplikasi Peminjaman Kendaraan Operasional Kantor Menggunakan BlackBox Testing. *INFORMATION MANAGEMENT FOR EDUCATORS AND PROFESSIONALS*, 5(2), 97.
<https://doi.org/10.51211/IMBI.V5I2.1561>
- [14] Hasanuddin, Asgar, H., & Hartono, B. (2022). RANCANG BANGUN REST API APLIKASI WESHARE SEBAGAI UPAYA MEMPERMUDAH PELAYANAN DONASI KEMANUSIAAN. *JINTEKS (Jurnal Informatika Teknologi Dan Sains)*, 4(1), 8–14. <https://jurnal.uts.ac.id/JINTEKS/article/view/1474/882>.
- [15] Rahmat, K. C., & Yung, S. (2025). Penerapan API Berbasis REST Guna Mengefisiensi Pendistribusian Data dan Aksebilitas Data di BPS Provinsi Jawa Barat. *Jurnal Ilmu Komputer*, 3(4), 171–190.
https://jurnal.ilmubersama.com/index.php/hello_world/article/view/636/427
- [16] Baniaş, O., Florea, D., Gyalai, R., & Curiac, D.-I. (2021). Automated Specification-Based Testing of REST APIs. *Selected Papers from the International Symposium on Electronics and Telecommunications ISETC 2020*, 21(16).
<https://doi.org/10.3390/S21165375>.
- [17] S, A. (2025). Penerapan Metode Extreme Programming dalam Pengembangan Aplikasi Point of Sale pada Usaha Phi Phi Thai. *Bridge : Jurnal Publikasi Sistem Informasi dan Telekomunikasi*. <https://doi.org/10.62951/bridge.v3i3.613>.
- [18] Ramadhan, C., Senubekti, M., & Amalia, D. (2025). Penerapan Metodologi Agile dalam Pengembangan Perangkat Lunak. *Router : Jurnal Teknik Informatika dan Terapan*. <https://doi.org/10.62951/router.v3i2.411>.
- [19] Zhu, X., Wen, S., Çamtepe, S., & Xiang, Y. (2022). Fuzzing: A Survey for Roadmap. *ACM Computing Surveys (CSUR)*, 54, 1 - 36.
<https://doi.org/10.1145/3512345>.
- [20] Manès, V., Han, H., Han, C., Cha, S., Egele, M., Schwartz, E., & Woo, M. (2018). The Art, Science, and Engineering of Fuzzing: A Survey. *IEEE Transactions on Software Engineering*, 47, 2312-2331. <https://doi.org/10.1109/tse.2019.2946563>.
- [21] Yu, X., Lv, L., Zhang, Z., Lan, Y., & Liu, L. (2025). Response relation model-driven approach for automated fuzz testing of RESTful Web APIs. , 13637, 1363702 - 1363702-6. <https://doi.org/10.1117/12.3056906>.
- [22] Atlidakis, V., Godefroid, P., & Polishchuk, M. (2019). RESTler: Stateful REST API Fuzzing. 2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE), 748-758. <https://doi.org/10.1109/icse.2019.00083>.
- [23] Martin-Lopez, A., Segura, S., & Ruiz-Cortés, A. (2021). RESTest: automated black-box testing of RESTful web APIs. *Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis*.
<https://doi.org/10.1145/3460319.3469082>.
- [24] harjono, W., & Tute, K. J. (2022). Perancangan Sistem Informasi Perpustakaan Berbasis Web Menggunakan Metode Waterfall. *SATESI: Jurnal Sains Teknologi Dan Sistem Informasi*, 2(1), 47–51. <https://doi.org/10.54259/satesi.v2i1.773>

- [25] Aldo, D., Rahman Habibie, D., & Susie. (2021). Metode FAST Untuk Pembangunan Sistem Inventory. *JURNAL INOVTEK POLBENG - SERI INFORMATIKA*, 6(2), 211–221. <https://doi.org/https://doi.org/10.35314/isi.v6i2>
- [26] Abdilah, R., Hermawan, R., Hermawansyah, W., Adkha, I., & Arifin, H. (2024). Pengujian Perangkat Lunak Sistem Informasi Inventori pada Usaha Jasa Pengiriman Paket. *Polygon : Jurnal Ilmu Komputer Dan Ilmu Pengetahuan Alam*, 2(4), 166–175. <https://doi.org/https://doi.org/10.62383/polygon.v2i4.199>
- [27] Nurfauziah, H., & Jamaliyah, I. (2022). *PERBANDINGAN METODE TESTING ANTARA BLACKBOX DENGAN WHITEBOX PADA SEBUAH SISTEM INFORMASI*. 8(2), 103–115. <https://jurnas.saintekmu.ac.id/index.php/visualika/en/article/view/24/19>
- [28] Srivastava, N., Kumar, U., Singh, P., & Singh, P. (2021). Software and Performance Testing Tools. *Journal of Infor-Matics Electrical and Electronics Engineering*, 02(001), 1–12. <https://doi.org/10.54060/JIEEE/002.01.001>