

DAFTAR ISI

HALAMAN PENGESAHAN	2
PERNYATAAN	3
PRAKATA	4
DAFTAR ISI	5
DAFTAR TABEL	9
DAFTAR GAMBAR	10
DAFTAR SIMBOL	11
INTISARI	12
BAB I	1
1.1. Latar Belakang	1
1.2. Perumusan Produk	4
1.3. Batasan Produk	5
1.4. Tujuan Penelitian	5
1.5. Manfaat Penelitian	6
BAB II	7
2.1. Identifikasi Masalah	7
2.1.1. Hambatan Aksesibilitas dan Kompleksitas Operasional Alat Fuzzing Akademik	7
2.1.2. Kesenjangan Integrasi dalam Alur Kerja Industri (CI/CD)	8
2.1.3. Beban Kognitif dalam Interpretasi dan Manajemen Hasil Pengujian	8
2.2. Pengembangan Ide Inovatif	9
2.2.1. Demokratisasi Pengujian API Cerdas melalui Antarmuka Web	9
2.2.2. Integrasi Rekomendasi Proaktif dalam Alur Kerja CI/CD	9
2.2.3. Arsitektur Terintegrasi untuk Operasional Skala Industri	10
2.2.4. Aksesibilitas dan Visualisasi Hasil sebagai Fokus Desain	10
BAB III	11
3.1. REST API	11
3.1.1. Arsitektur Representational State Transfer (REST)	11
3.1.2. OpenAPI Specification (OAS)	11
3.2. Fuzzing REST API	12
3.2.1. Konsep Fuzz Testing	12
3.2.2. Tantangan Pengujian API Stateful	12
3.3. Semantic Operation Dependency Graph (SODG)	12
3.3.1. Pemodelan Dependensi Operasi	12
3.3.2. Ekstraksi Semantik dari Dokumentasi API	13

3.4. Multi-Agent Reinforcement Learning (MARL)	13
3.4.1. Reinforcement Learning (RL) dalam Pengujian	13
3.4.2. Agen Terspesialisasi dalam AutoRestTest	13
3.4.3. Parameter Optimasi RL	14
3.5. Large Language Models (LLM) dalam Otomasi Pengujian	14
3.5.1. Pemahaman Konteks Semantik	15
3.5.2. Integrasi LLM Engine	15
3.6. Jenis Pengujian Spesifikasi REST API	15
3.6.1. Pengujian REST API Berbasis Spesifikasi	15
3.6.2. Pengujian REST API Stateful dan Berbasis Model	16
3.6.3. Pengujian REST API Berbasis Reinforcement Learning (RL)	17
3.7. AutoRestTest	18
3.7.1. Integrasi SODG, MARL, dan LLM	18
3.7.2. Perbandingan dengan Alat Pengujian Tradisional	19
3.8. Sintesis dan Posisi Penelitian	19
3.9. Arsitektur Aplikasi Web dan Komunikasi Asinkron	20
3.9.1. Framework Next.js dan Server-Side Rendering (SSR)	20
3.9.2. FastAPI dan Python Backend	20
3.9.3. Orkestrasi Task (Trigger.dev)	20
3.10. Persistensi Data dan Optimasi Performa	21
3.10.1. PostgreSQL dan Basis Data Relasional	21
3.10.2. Prisma ORM	21
3.10.3. Sistem Caching dan In-Memory Database (Upstash Redis)	21
3.11. Continuous Integration dan Continuous Deployment (CI/CD)	22
3.11.1. Siklus Umpan Balik Kontinu	22
3.11.2. Alur Kerja GitHub Actions	22
3.12. Evaluasi Kualitas Perangkat Lunak	22
3.12.1. Task-Based Usability Testing	23
3.12.2. Metrik Google Lighthouse	23
BAB IV	24
4.1. Deskripsi Umum	24
4.2. Alur Pengembangan Produk	25
4.3. Analisis Solusi Terkait dan Posisi Strategis	26
4.4. Requirements Engineering	28
4.4.1. Functional Requirements	28
4.4.2. Non-Functional Requirements	29
4.5. Perancangan Desain Antarmuka Pengguna (UI/UX)	30

4.5.1. Pembentukan Persona Arketipe	30
4.5.2. Pembentukan User Flow	30
4.5.3. Arsitektur Informasi dan Tata Letak Antarmuka	34
4.6. Perancangan Arsitektur Sistem	35
4.6.1. Arsitektur Informasi dan Tata Letak Antarmuka	35
4.6.2. Alur Data dan Orkestrasi Asinkron	37
4.7. Perancangan Basis Data	40
4.7.1. Entity Relationship Diagram (ERD)	40
4.7.2. Definisi Skema Data	41
4.8. Strategi Evaluasi dan Validasi	42
4.8.1. Pengujian Fungsional	42
4.8.2. Pengujian Non-Fungsional	42
4.8.3. Pengujian Usabilitas (Task-Based Usability Testing)	43
4.8.4. Kuesioner Persepsi Subjektif	44
4.8.5. Matriks Penilaian dan Mekanisme Penskalaan	44
4.8.6. Audit Kualitas Web (Google Lighthouse)	44
BAB V	46
5.1. Lingkungan Implementasi	46
5.1.1. Spesifikasi Perangkat Keras	46
5.1.2. Spesifikasi Perangkat Lunak	46
5.2. Implementasi Basis Data	47
5.2.1. Skema Data Prisma	48
5.2.2. Detail Struktur Tabel dan Relasi	49
5.2.3. Transformasi ke PostgreSQL	49
5.3. Implementasi Backend dan Orkestrasi Tugas	50
5.3.1. Implementasi Task Runner (Trigger.dev)	50
5.3.2. Implementasi API Wrapper (FastAPI)	52
5.4. Implementasi Antarmuka Pengguna (Frontend)	53
5.4.1. Formulir Konfigurasi Pengujian (TestForm.tsx)	54
5.4.2. Dasbor Riwayat dan Pemantauan (JobHistory.tsx)	57
5.4.3. Visualisasi Hasil Pengujian (JobDetailsDataExplorer.tsx)	58
5.5. Implementasi Integrasi CI/CD	61
5.5.1. Otomasi Setup Workflow (Octokit)	61
5.5.2. Mekanisme Triggering CI/CD	63
5.6. Implementasi Keamanan dan Autentikasi	63
5.6.1. Manajemen Autentikasi Pengguna (Clerk)	63
5.6.2. Perlindungan Rute dengan Middleware	64

5.6.3. Keamanan Komunikasi Antar-Layanan (API Key)	65
5.7. Akses Layanan dan Repositori	66
BAB VI	67
6.1. Hasil Pengujian Fungsional	67
6.1.1. Hasil Unit Testing	67
6.1.2. Hasil Validasi Fitur Utama	69
6.1.3. Hasil Validasi Integrasi CI/CD	70
6.2. Hasil Pengujian Non-Fungsional	71
6.3.1. Analisis Performa API (Postman)	72
6.3.2. Audit Kualitas Web (Google Lighthouse)	73
6.3. Analisis Task-Based Usability Testing	74
6.3.1. Profil Demografis Partisipan	74
6.3.2. Hasil Observasi dan Tingkat Penyelesaian Tugas	75
6.3.3. Proses Normalisasi Skor Perilaku	76
6.3.4. Analisis Skor Komponen Berbot	77
6.3.5. Kalkulasi Skor Akhir Usabilitas	77
6.3.6. Analisis Masukan Peserta dan Perbaikan Fitur	78
6.4. Analisis Kuesioner Pasca-Sesi	80
6.4.1. Interpretasi Data Persepsi Subjektif	81
6.4.2. Sintesis Masukan dan Sentimen Kualitatif	82
6.5. Pembahasan Hasil dan Dampak Penelitian	83
6.5.1. Penjembatanan Teori dan Praktik Melalui Antarmuka Web	83
6.5.2. Peran Integrasi CI/CD dalam Siklus Penjaminan Kualitas	83
6.5.3. Kualitas Teknis dan Keberterimaan Pengguna	84
6.6. Kesimpulan dan Saran	85
6.6.1. Kesimpulan	85
6.6.2. Saran	86
DAFTAR PUSTAKA	87
LAMPIRAN	89