

**Mekanisme Kontrol Kualitas Penguji untuk Pengujian
Penerimaan Pengguna Berbasis Crowdsourcing
menggunakan Algoritma M-X**

Tugas Akhir

diajukan untuk memenuhi salah satu syarat memperoleh gelar sarjana

dari Program Studi S1 Rekayasa Perangkat Lunak

Fakultas Informatika

Universitas Telkom

1302213015

Glorious Satria Dhamang Aji



**Universitas
Telkom**

Program Studi Sarjana Rekayasa Perangkat Lunak

Fakultas Informatika

Universitas Telkom

Bandung

2025

LEMBAR PENGESAHAN

**Mekanisme Kontrol Kualitas Penguji untuk Pengujian
Penerimaan Pengguna Berbasis Crowdsourcing menggunakan
Algoritma M-X
Tester Quality Control Mechanism for Crowdsourcing-Based
User Acceptance Testing using M-X Algorithm**

1302213015

Glorious Satria Dhamang Aji

Tugas akhir ini telah diterima dan disahkan untuk memenuhi sebagai syarat
memperoleh gelar pada Program Studi Sarjana S1 Rekayasa Perangkat Lunak

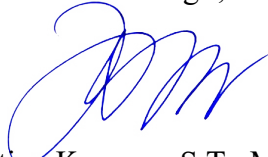
Fakultas Informatika

Universitas Telkom

Bandung, 26 Maret 2025

Menyetujui

Pembimbing I,



Dana Sulistiyo Kusumo, S.T., M.T., Ph. D

NIP: 02780011

Ketua Program Studi

Sarjana Rekayasa Perangkat Lunak,



Dr. Eng. Jati Hiliamsyah Husen, S.T., M.Eng.

NIP: 20920040

LEMBAR ORISINALITAS

Dengan ini saya, Glorious Satria Dhamang Aji, menyatakan sesungguhnya bahwa Tugas Akhir saya dengan judul Mekanisme Kontrol Kualitas Penguji untuk Pengujian Penerimaan Pengguna Berbasis Crowdsourcing menggunakan Algoritma M-X berserta dengan seluruh isinya merupakan hasil karya saya sendiri, dengan tidak melakukan penjiplakan yang tidak sesuai dengan etika keilmuan yang berlaku dengan masyarakat keilmuan, serta produk dari tugas akhir ini bukan merupakan hasil dari *Generative AI*. Saya siap menggugung risiko/sanksi yang diberikan jika di kemudian hari ditemukan pelanggaran terhadap etika keilmuan dalam Laporan Tugas Akhir, atau jika ada klaim dari pihak lain terhadap keaslian karya.

Bandung, 25 Maret 2025

Yang menyatakan

Glorious satria Dhamang Aji

NIM: 1302213015

ABSTRAK

Pengujian penerimaan pengguna berbasis crowdsourcing (Crowdsourced User Acceptance Testing – UAT) sekarang menghadapi tantangan akibat dari variatifnya karakteristik penguji yang dapat memengaruhi konsistensi dari luaran yang dihasilkan. Penelitian ini mengevaluasi penerapan algoritma M-X di dalam mekanisme kontrol kualitas untuk menyaring penguji berdasarkan konsistensi jawaban mereka ketika mengerjakan tugas sehubungan dengan UAT tanpa bergantung pada jawaban yang telah ditentukan. Untuk menunjang penelitian, maka sistem berbasis client-server dikembangkan sebagai alat simulasi. Sistem dikembangkan menggunakan GraphQL dan MongoDB untuk mendukung proses simulasi pengujian. Evaluasi menggunakan Confusion Matrix—dengan pemetaan langsung terhadap pertanyaan penelitian—menunjukkan bahwa algoritma M-X mencapai akurasi 79%, presisi 82%, dan recall 75%. Hasil ini mendemonstrasikan keefektifannya dalam mendeteksi penguji yang konsisten dan memperkuat penerapannya dalam skenario UAT yang bersifat subjektif dan eksploratif.

Kata Kunci: *crowdsourcing, user acceptance testing, kontrol kualitas, algoritma*

M-X, online assessment, client-server.

ABSTRACT

Crowdsourced User Acceptance Testing (UAT) often faces challenges due to the variability of tester characteristics, which can affect the consistency of test outcomes. This study evaluates the use of M-X algorithm within a quality control mechanism for filtering testers based on response consistency without relying on predefined correct answers. A client-server system was developed using GraphQL and MongoDB to support a simulation process conducted over five iterations with an increasing number of testers. The evaluation was carried out. The evaluation was carried out using Confusion Matrix metrics—mapped directly to research questions—which showed the M-X algorithm achieves 79% accuracy, 82% precision, and 75% recall. These results demonstrate its effectiveness in detecting consistent testers and reinforce its applicability to subjective and explorative UAT scenarios.

Keywords: *crowdsourcing, user acceptance testing, kontrol kualitas, algoritma*

M-X, online assessment, client-server.

KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yeshua sang Mesias, atas dasar kasih karunia, dan bimbingan-Nya yang tak terhingga, sehingga Laporan Tugas Akhir dengan judul "Mekanisme Kontrol Kualitas Penguji untuk Pengujian Penerimaan Pengguna Berbasis Crowdsourcing menggunakan Algoritma M-X" ini dapat terselesaikan. Tugas akhir ini adalah puncak dari sebuah perjalanan panjang yang penuh liku, namun juga dihiasi dengan pengalaman-pengalaman luar biasa yang tak pernah terbayangkan sebelumnya.

Penulisan tugas akhir ini bertujuan untuk mengevaluasi efektivitas algoritma M-X sebagai mekanisme kontrol kualitas penguji dalam konteks *User Acceptance Testing* (UAT) berbasis *crowdsourcing*. Latar belakang penelitian ini berangkat dari tantangan variabilitas karakteristik penguji dalam pekerjaan crowdsourcing yang dapat memengaruhi konsistensi hasil pengujian. Oleh karena itu, diperlukan sebuah pendekatan yang mampu menyaring penguji yang konsisten tanpa bergantung pada jawaban yang telah ditentukan sebelumnya.

Terima kasih yang tulus dan mendalam saya sampaikan kepada berbagai pihak yang telah memberikan bantuan, bimbingan, dan dukungan yang relevan selama proses penyusunan tugas akhir ini, serta sepanjang perjalanan studi saya yang luar biasa.

Terutama, saya ingin menghaturkan terima kasih yang sebesar-besarnya kepada kedua orang tua saya tercinta, yang menjadi sumber kekuatan, doa, dan pengorbanan tanpa henti, bahkan di tengah keterbatasan dana yang kerap menjadi ujian berat selama delapan semester. Ketahanan dan semangat yang mereka berikan adalah alasan utama saya mampu bertahan.

Kepada Bapak Dana Sulistiyo Kusumo, S.T., M.T., Ph.D. selaku Pembimbing I saya, saya mengucapkan terima kasih tak terhingga atas bimbingan, arahan, kesabaran, dan wawasan yang sangat berharga. Beliau telah membimbing saya dengan penuh dedikasi.

Apresiasi juga saya sampaikan kepada Bapak Dr Eng. Jati Hiliamsyah Husen, S.T., M.Eng. selaku Ketua Program Studi Sarjana Rekayasa Perangkat Lunak, atas dukungan dan fasilitas yang telah diberikan.

Secara khusus, saya merasa sangat berterima kasih kepada Telkom University yang telah membuka pintu bagi saya untuk merasakan pengalaman-pengalaman internasional yang transformatif. Kesempatan untuk terlibat dalam *Global Internship Program (GIP)*, *Indonesian International Student Mobility Awards (IISMA)* yang mengizinkan saya menjelajahi keindahan perbedaan dan harmoni budaya, dan yang akan datang, partisipasi dalam *International Conference on Software Engineering and Information Technology (ICOSEIT)* di bulan Oktober nanti, adalah anugerah yang tak ternilai harganya. Pengalaman-pengalaman ini tidak hanya memperkaya ilmu dan wawasan saya, tetapi juga membentuk pribadi yang lebih mandiri dan adaptif.

Kepada seluruh dosen, staf, rekan-rekan seperjuangan, sahabat, dan setiap individu yang telah memberikan motivasi, dukungan, dan pengertian, saya ucapkan terima kasih banyak.

Harapan saya, semoga Laporan Tugas Akhir ini dapat memberikan manfaat yang signifikan bagi pengembangan ilmu pengetahuan, khususnya di bidang *crowdsourcing based UAT*, serta dapat menjadi referensi yang berguna bagi penelitian selanjutnya.

Saya menyadari bahwa laporan tugas akhir ini masih jauh dari sempurna. Oleh karena itu, saya dengan rendah hati memohon maaf apabila terdapat kekurangan, kesalahan, atau kekhilafan dalam penulisan maupun isinya. Kritik dan saran yang membangun akan senantiasa saya terima dengan lapang dada demi perbaikan di masa mendatang.

UCAPAN TERIMA KASIH

En la vida, hay un principio y un final. Hay cosas buenas y malas. Hay victorias y derrotas. Hay sonrisas y lágrimas. Hay errores y verdades. Pero nada es casualidad.

Dalam kehidupan, ada awal dan akhir. Ada baik dan buruk. Ada menang dan kalah. Ada senyum dan tangis. Ada kesalahan dan kebenaran. Tapi tidak ada yang kebetulan.

Dan di dalam setiap kejadian, hati saya belajar bersyukur. Hari ini, dengan hati penuh, saya mengucapkan terima kasih yang tulus.

Pertama dan selamanya, kepada Tuhan Yeshua sang Mesias. Setiap langkah dan kesempatan adalah cerminan dari kasih dan tuntunan-Nya. Dia menopang saya di masa sulit dan menerangi jalan saya. *Terima kasih untuk segalanya!*

Berikutnya, dengan kasih sayang yang meluap, kepada orang tua saya tercinta. Anda adalah pahlawan saya. Cinta tanpa syarat dan pengorbanan Anda, terutama saat kesulitan finansial melanda selama delapan semester ini, adalah cahaya bagi saya. Setiap usaha Anda adalah investasi untuk masa depan saya. Ibu, Ayah, kekuatan Anda adalah warisan saya, *cinta Anda adalah tempat berlindung abadi saya!*

Dan untukmu, **almamater** tercinta saya, tempat saya belajar dan berkembang. Anda membuka pintu bagi saya untuk pengetahuan dan banyak pengalaman yang mengubah hidup. Kesempatan menjelajahi berbagai tempat dan mengenal budaya lain adalah hadiah tak ternilai dari Anda. Anda tidak hanya memberi saya ilmu, tapi juga impian dan kemampuan untuk meraihnya. Untuk pendidikan, persahabatan, pelajaran hidup, dan setiap kesempatan, *rasa terima kasih saya akan abadi!*

Pencapaian ini, yang saya rayakan hari ini, bukan hanya milik saya. Ini adalah hasil dari dukungan dan pengorbanan banyak pihak. Saya berharap karya ini dapat memberi sedikit kontribusi pada ilmu pengetahuan dan menginspirasi orang lain untuk mengejar impian dengan semangat dan ketekunan yang sama.

DAFTAR ISI

LEMBAR PENGESAHAN	ii
LEMBAR ORISINALITAS	iii
ABSTRAK	iv
<i>ABSTRACT</i>	v
KATA PENGANTAR.....	vi
UCAPAN TERIMA KASIH	viii
DAFTAR GAMBAR.....	xi
DAFTAR RUMUS	xi
DAFTAR TABEL	xi
BAB 1. PENDAHULUAN.....	8
1.1. Latar Belakang	8
1.2. Perumusan Masalah.....	10
1.3. Tujuan Penelitian.....	10
1.4. Batasan Masalah.....	11
1.5. Metode Penelitian.....	12
1.5.1 Kajian Pustaka	12
1.5.2 Rancangan Penelitian	12
1.5.3 Pengembangan Sistem.....	12
1.5.4 Simulasi dan Uji Coba.....	13
1.5.5 Analisis Hasil Simulasi.....	13
1.5.6 Penyusunan Laporan / Buku TA	13
BAB 2. TINJAUAN PUSTAKA	14

2.1.	Penelitian Terkait	14
2.2.	User Acceptance Testing	15
2.3.	Crowdsourcing Software Testing	16
2.4.	Algoritma M-1	17
2.5.	Algoritma M-X	18
2.6.	Prior-Experience Based Validation	19
2.7.	Confusion Matrix	20
2.8.	Gherkin Syntax	23
BAB 3.	PERANCANGAN SISTEM DAN METODOLOGI	16
3.1.	Tahap Perancangan Simulasi	16
3.1.1	Tahap Perancangan Skenario Simulasi	16
3.1.2	Tahap Perancangan Data Source Simulasi	16
3.1.3	Tahap Perancangan Komponen Penting Simulasi	17
3.1.4	Tahap Perancangan Platform Simulasi	22
3.2	Tahap Eksekusi Simulasi	27
3.3	Validasi Algoritma dan Evaluasi	32
3.4	Tahap Analisis	33
BAB 4.	HASIL PERCOBAAN DAN ANALISIS	34
4.1.	Analisis Empiris Data	34
4.1.1.	Accuracy	35
4.1.2.	Precision	35
4.1.3.	Recall	35
4.1.4.	True Negative Rate (TNR)	35
4.1.5.	F1 Score	35
4.2.	Visualisasi Data	36
4.3.	Menjawab Research Question	39

BAB 5. KESIMPULAN.....	42
DAFTAR PUSTAKA.....	16
LAMPIRAN.....	19

DAFTAR GAMBAR

Gambar 3.1 Flowchart Penyelesaian Masalah	16
Gambar 3.2 Model Desain Task Gherkin	19
Gambar 3.3 Solution Architecture Sistem dengan Implementasi Algoritma M-X.....	22
Gambar 3.4 Dokumentasi Hari Pertama Simulasi	27
Gambar 3.5 Dokumentasi Hari Kedua Simulasi.....	27
Gambar 3.6 Dokumentasi Hari Ketiga Simulasi.....	28
Gambar 3.7 Rasio Perbandingan Actual Eligibility Relawan	32

DAFTAR RUMUS

(2.1) Rumus Algoritma M1	17
(2.2) Rumus Algoritma MX.....	19
(2.3) Rumus Metriks Accuracy	21
(2.4) Rumus Metriks Precision	22
(2.5) Rumus Metriks Recall	22

DAFTAR TABEL

Tabel 1.1 Tabel keterkaitan antara tujuan, pengujian dan kesimpulan	10
Tabel 2.1 Format Gherkins Scenario	24
Tabel 3.1 Tabel Komponen Platform Simulasi	23
Tabel 3.2 Tabel Tahapan Implementasi Algoritma M-X.....	26
Tabel 3.3 Tabel Data Relawan	28
Tabel 3.4 <i>Actual Eligibility</i> Relawan	31

Tabel 4.1 Tabel Hasil Simulasi.....	34
--	-----------

BAB 1. PENDAHULUAN

1.1. Latar Belakang

Software delivery yang sesuai dengan kebutuhan yang telah ditentukan merupakan salah satu tujuan utama dalam pengembangan perangkat lunak komersial [1]. Untuk mencapai kualitas yang sesuai standar, diperlukan serangkaian pengujian terstruktur, dengan User Acceptance Testing (UAT) sebagai tahap krusial terakhir sebelum implementasi [2].

UAT memegang peranan penting sebagai penghubung antara pengembang dan pengguna akhir, memastikan perangkat lunak yang dibangun sesuai dengan kebutuhan pengguna dan tujuan bisnis [3]. Tantangan utama dalam UAT adalah memastikan cakupan pengujian yang luas dan mencakup beragam skenario penggunaan nyata, yang sering kali membutuhkan biaya, waktu, dan sumber daya yang signifikan.

Crowdsourcing menawarkan solusi potensial untuk tantangan tersebut dengan melibatkan banyak pengguna dari berbagai latar belakang dalam proses pengujian, sehingga perusahaan mendapatkan cakupan pengujian yang lebih luas dan realistis dengan alokasi anggaran yang lebih efisien [6]. Namun, pendekatan ini menghadirkan tantangan baru berupa keacakan karakteristik penguji (*workers*).

Keacakan karakteristik *workers* dalam *crowdsourcing* membuka celah kecacatan pengujian, di mana penguji memiliki probabilitas untuk tidak memenuhi standar kebutuhan pengujian. Menurut penelitian Kazai et al. (2011), "keberagaman *workers* dalam *crowdsourcing* dapat menyebabkan hasil yang tidak konsisten akibat perbedaan pengalaman, perhatian, dan pemahaman terhadap persyaratan tugas" [7]. Dalam konteks ini, tidak semua penguji mampu atau bersedia mengikuti standar. Keacakan karakteristik ini dipengaruhi oleh motivasi, latar belakang, keterampilan, dan perilaku *workers*, dengan risiko adanya *workers* yang tidak jujur, asal-asalan, atau malas [8].

Konsekuensinya, keacakan karakteristik *workers* berpotensi memperlambat waktu pengujian, mengganggu proses distribusi perangkat lunak, dan menghasilkan hasil uji yang bias [7]. Diperlukan mekanisme klasifikasi *workers* yang efektif untuk

memastikan proses pengujian berjalan optimal dan menghasilkan data yang andal.

Algoritma M-X yang dikembangkan oleh D. Dang, Y. Liu, X. Zang, dan S. Huang [9]. menawarkan pendekatan menjanjikan untuk menilai kualitas *workers* berdasarkan konsistensi jawaban. Keunggulan algoritma ini terletak pada kemampuannya mengidentifikasi *workers* yang konsisten tanpa bergantung pada jawaban baku, sehingga dapat diterapkan pada skenario pengujian yang bersifat subjektif maupun eksploratif seperti pengujian *usability*, validasi fitur, atau evaluasi berbasis skenario realistis. Namun, tingkat akurasi algoritma ini dalam konteks UAT berbasis *crowdsourcing* belum dikaji secara komprehensif, dan belum jelas sejauh mana algoritma ini dapat mengurangi variabilitas karakteristik penguji untuk memastikan konsistensi kualitas pengujian.

Penelitian ini mengusulkan pendekatan empiris untuk mengevaluasi efektivitas algoritma M-X dalam klasifikasi kelayakan *workers* pada konteks UAT berbasis *crowdsourcing*. Melalui eksperimen kuantitatif, penelitian ini akan mengukur keakuratan algoritma dalam mengidentifikasi pola konsistensi jawaban di antara penguji, serta kemampuannya beradaptasi dengan lingkungan pengujian UAT berbasis *crowdsourcing*.

Untuk mendukung eksperimen tersebut, sistem simulasi berbasis arsitektur client-server akan dikembangkan dengan GraphQL sebagai antarmuka pengiriman data dan MongoDB sebagai basis data. Pemilihan teknologi ini mempertimbangkan kebutuhan fleksibilitas dan skalabilitas sistem [10], kemampuan GraphQL dalam mengelola struktur data kompleks secara efisien [11], serta keunggulan MongoDB dalam menangani data tidak terstruktur dengan performa tinggi [12]. Penelitian ini diharapkan menghasilkan evaluasi komprehensif terhadap efektivitas algoritma M-X sebagai mekanisme kontrol kualitas dalam lingkungan pengujian berbasis *crowdsourcing*.

1.2. Perumusan Masalah

Setelah memahami latar belakang dan konteks penelitian, berikut adalah rumusan masalah yang menjadi fokus penelitian:

1. Bagaimana tingkat akurasi algoritma M-X dalam mengklasifikasikan penguji dalam lingkungan UAT berbasis crowdsourcing?
2. Sejauh mana algoritma M-X dapat mengurangi variabilitas karakteristik penguji dan memastikan konsistensi kualitas pengujian?

Melalui kedua rumusan masalah ini, penelitian bertujuan untuk memberikan pemahaman mendalam tentang kesesuaian algoritma M-X untuk seleksi penguji yang andal dalam konteks modern UAT berbasis *crowdsourcing*.

1.3. Tujuan Penelitian

Tujuan utama dari penelitian ini adalah untuk menjawab rumusan masalah dengan mengimplementasikan penelitian kuantitatif menggunakan sistem simulasi yang dirancang secara metodologis dan menganalisis data metrik kinerja algoritma secara sistematis. Hasil dari penelitian ini diharapkan dapat memberikan landasan empiris bagi pengembangan sistem seleksi penguji yang lebih adaptif dan akurat dalam lingkungan UAT crowdsourcing yang lebih dinamis.

Tabel 1.1 Tabel keterkaitan antara tujuan, pengujian dan kesimpulan

No.	Tujuan	Pengujian	Kesimpulan
1	Mengevaluasi efektivitas algoritma M-X dalam klasifikasi kelayakan penguji pada pengujian UAT berbasis crowdsourcing.	Mengadakan simulasi terkontrol untuk membawa algoritma M-X ke ranah seleksi penguji untuk UAT berbasis Crowdsourcing	Analisis dilakukan untuk menilai sejauh mana algoritma M-X mampu menyaring penguji yang konsisten dan bagaimana performanya berubah seiring peningkatan jumlah penguji dan variasi respons.

1.4. Batasan Masalah

Untuk mencegah meluasnya cakupan dari masalah ini yang berpotensi memudahkan fokus penelitian, maka dibuatlah batasan-batasan permasalahan untuk penelitian dan perancangan sebagai berikut:

1. Penelitian ini mencakup kontrol kualitas penguji dalam konteks UAT crowdsourcing, dan tidak mencakup pengujian pada tahap pengujian perangkat lunak lainnya, seperti contoh *Unit Testing* atau *Integration Testing*.
2. Penelitian ini akan mempertimbangkan keterbatasan waktu, biaya, dan sumber daya dalam proses kontrol kualitas, sehingga pendekatan yang dirancang akan berorientasi pada solusi yang efisien dan praktis.
3. Fokus utamanya adalah pada evaluasi terhadap metode untuk menekan tingkat keacakan dan ketidakseragaman karakteristik penguji, tanpa melibatkan aspek teknis implementasi UAT secara keseluruhan.
4. Simulasi hanya dilakukan pada skenario pilihan ganda, sehingga algoritma M-X tidak diuji untuk soal tipe lain seperti isian atau esai.
5. Penentuan *ground truth* dilakukan secara manual dan terbatas pada skenario pengujian tertentu, sehingga validitas absolut tidak dijamin secara menyeluruh.
6. Evaluasi hanya dilakukan terhadap konsistensi antar jawaban (*peer consistency*), bukan terhadap efektivitas hasil pengujian terhadap fungsionalitas perangkat lunak secara menyeluruh.
7. Sistem yang dikembangkan tidak diuji pada lingkungan produksi nyata, tetapi hanya pada simulasi internal yang dikendalikan.
8. Jumlah penguji dalam simulasi dibatasi maksimal 24 orang, yang mungkin belum mencerminkan kompleksitas crowd dalam skenario industri sebenarnya.
9. Faktor eksternal seperti motivasi, kelelahan, atau kontekstual lingkungan pengujian tidak dikendalikan secara ketat, sehingga dapat mempengaruhi hasil klasifikasi kualitas penguji.

10. Relawan penelitian akan diambil mayoritas dari lingkup Telkom University karena keterbatasan sumber daya dan waktu.

1.5. Metode Penelitian

Penelitian ini menggunakan metode penelitian kuantitatif, lebih tepatnya simulasi kuantitatif, dengan variabel untuk mengamati dan mengukur pengaruh algoritma M-X dalam proses klasifikasi kualitas penguji. Eksperimen dilakukan melalui sistem berbasis web yang dirancang khusus untuk mengimplementasikan algoritma tersebut dalam konteks *User Acceptance Testing* (UAT) berbasis *crowdsourcing*. Data yang dihasilkan dari simulasi bersifat numerik dan dianalisis untuk mengukur efektivitas klasifikasi menggunakan metrik seperti akurasi, presisi, dan *recall*.

1.5.1 Kajian Pustaka

Pada tahap ini dilakukan studi literatur dan pengumpulan referensi jurnal yang sesuai dengan topik yang diangkat pada penelitian ini. Studi literatur ini mencakup *Crowdsourcing Software Testing*, *Usability Testing*, kontrol kualitas *workers*, serta penggunaan algoritma untuk klasifikasi *workers* berdasarkan keterampilan dan kualitas. Informasi yang didapat akan digunakan sebagai landasan teori dan acuan dalam mengembangkan metode kontrol kualitas *workers* pada penelitian ini.

1.5.2 Rancangan Penelitian

Kegiatan rancangan penelitian ini akan terdiri dari perancangan sistematika penyelesaian masalah, perancangan sistem, baik itu desain arsitektur, maupun desain model, dan perancangan skema pengujian penelitian. Penelitian ini berfokus pada implementasi metode kontrol kualitas *workers* pada *User Acceptance Testing* berbasis *Crowdsourcing*. Penelitian ini akan mengimplementasikan algoritma M-X di dalam ranah platform UAT dan melihat hasil dari implementasi algoritma tersebut.

1.5.3 Pengembangan Sistem

Dalam rangka mewujudkan keberhasilan dari penelitian ini, maka dibangun sistem berbasis web yang dapat menyeleksi kandidat *workers* untuk melakukan *User Acceptance Testing* berbasis *crowdsourcing* dengan

mengimplementasikan algoritma M-X. Sistem ini akan mencakup fitur pengelolaan data *workers*, klasifikasi berdasarkan kualitas, dan rekomendasi *workers* terbaik untuk setiap tugas pengujian. Sistem akan dibangun menggunakan kerangka *framework* Next.js untuk antarmuka pengguna dan Nest.js sebagai backend, serta menggunakan database non-relasional seperti mongoDB untuk manajemen data.

1.5.4 Simulasi dan Uji Coba

Penelitian ini menggunakan metode pengumpulan data kuantitatif dan kualitatif. Data kuantitatif didapatkan melalui hasil uji coba sistem oleh pengguna sebagai kandidat penguji. Uji coba yang dijalankan akan menerapkan algoritma M-X terhadap proses kontrol kualitas di sistem. Data kualitatif diperoleh dari hasil simulasi proses pengujian UAT oleh para penguji yang lolos uji coba. Data ini akan dianalisis untuk mengidentifikasi pola perilaku dan karakteristik *workers*.

1.5.5 Analisis Hasil Simulasi

Analisis hasil simulasi dilakukan menggunakan pendekatan evaluasi berbasis Confusion Matrix untuk mengukur performa algoritma M-X dalam mengklasifikasikan penguji. Metrik evaluasi utama yang dihitung meliputi akurasi (*accuracy*), presisi (*precision*), *recall*, *True Negative Rate* (TNR), dan *F1 Score* yang masing-masing memberikan perspektif berbeda terhadap kemampuan klasifikasi algoritma.

1.5.6 Penyusunan Laporan / Buku TA

Semua hasil penelitian, mulai dari kajian pustaka, pengumpulan data, analisis, pengembangan sistem, hingga pengujian, akan didokumentasikan dalam bentuk laporan akhir. Laporan ini akan mencakup kesimpulan dan rekomendasi untuk penelitian lebih lanjut.

BAB 2. TINJAUAN PUSTAKA

2.1. Penelitian Terkait

Pada tahun 2019, Wang et al. [13] meneliti paper berjudul ‘Characterizing Crowds to Better Optimize Worker Recommendation in Crowdsourced Testing’. Masalah utama yang dibahas dalam paper ini adalah inefisiensi dalam pengujian berbasis *crowdsourcing* akibat variabilitas tingkat keterampilan *workers*, yang mengakibatkan laporan bug terduplikasi atau terlewat. Tujuan utama penelitian ini adalah untuk mengembangkan metode baru guna merekomendasikan *workers* berbasis *crowdsourcing* yang sesuai untuk tugas pengujian tertentu, memaksimalkan deteksi bug, sekaligus meminimalkan biaya. Pendekatan yang digunakan adalah algoritma *Multi-Objective Crowd Worker Recommendation* (MOCOM) yang mengintegrasikan kerangka kerja optimasi multi-objektif dengan empat tujuan utama: memaksimalkan probabilitas deteksi bug, meningkatkan relevansi terhadap tugas, memastikan keberagaman *workers*, dan meminimalkan biaya. Algoritma ini menggunakan NSGA-II untuk optimasi multi-objektif, serta *machine learning* untuk memprediksi probabilitas deteksi bug setiap *workers* berdasarkan data historis dan konteks. Hasil penelitian menunjukkan bahwa MOCOM efektif dalam meningkatkan efisiensi deteksi bug dengan pemilihan *workers* yang lebih optimal, meskipun metode ini memerlukan sumber daya komputasi yang tinggi dan sangat bergantung pada data historis.

Pada tahun 2021, Hong et al. [14] menulis *paper* berjudul ‘Crowd Worker Selection with Wide Learning and Narrow Evaluation’. *Paper* ini membahas tentang kesulitan menemukan *workers* yang sesuai dalam sistem *crowdsourcing* tradisional. Studi ini memperkenalkan *Worker Search Model* (WSM) yang menggunakan pendekatan *wide learning* untuk menganalisis atribut *workers* seperti usia, jenis kelamin, lokasi, dan preferensi, serta *Response Limit Model* (RLM) untuk mengelompokkan respons *workers* demi evaluasi yang lebih efisien. Simulasi yang dilakukan menunjukkan peningkatan akurasi dan efisiensi pemilihan *workers*

dibandingkan metode tradisional berbasis LSP. Namun, implementasi sistem ini membutuhkan sumber daya komputasi yang signifikan, serta ketersediaan data demografis yang dapat menimbulkan masalah privasi.

Pada tahun 2021, Yao et al. [15] meneliti paper berjudul ‘A Study on Testers’ Learning Curve in Crowdsourced Software Testing’. Penelitian ini mengeksplorasi proses pembelajaran penguji berbasis *crowdsourcing* dengan menggunakan model kurva pembelajaran hiperbolik dan eksponensial untuk mengevaluasi peningkatan keterampilan tester. Data dari platform pengujian MOOCTEST digunakan untuk melatih dan mengevaluasi model ini. Hasil penelitian menunjukkan bahwa model kurva eksponensial lebih baik dalam menggambarkan peningkatan keterampilan dibandingkan kurva hiperbolik, meskipun terdapat keterbatasan seperti bias terhadap data yang sebagian besar berasal dari mahasiswa. Studi ini memberikan wawasan penting mengenai evaluasi keterampilan tester dalam platform *crowdsourcing*.

2.2. User Acceptance Testing

User Acceptance Testing (UAT) didefinisikan sebagai proses verifikasi yang dilakukan oleh pengguna untuk menentukan apakah perangkat lunak yang dikembangkan sesuai dengan spesifikasi dan kebutuhan yang diharapkan. UAT dilakukan setelah semua pengujian sebelumnya selesai dan perangkat lunak dianggap stabil untuk digunakan di lingkungan operasional. User Acceptance Testing (UAT) berada pada tahap akhir dalam proses pengujian perangkat lunak sebelum dirilis untuk penggunaan operasional. UAT bertujuan untuk memverifikasi kesiapan perangkat lunak berdasarkan kriteria penerimaan yang telah ditentukan dan disepakati oleh pengembang serta pengguna. Kriteria penerimaan mencakup kelengkapan fitur, dokumentasi, serta kemampuan sistem untuk memenuhi kebutuhan dan ekspektasi pengguna akhir. Proses UAT harus didasarkan pada perencanaan pengujian yang jelas, termasuk desain kasus uji, prosedur pengujian, dan dokumen hasil pengujian. Setelah memenuhi kriteria penerimaan, perangkat lunak dapat dianggap layak untuk dioperasikan [1].

Untuk melaksanakan *User Acceptance Testing* (UAT), dibutuhkan input meliputi dokumen spesifikasi perangkat lunak, hasil pengujian sebelumnya (seperti *System Testing*), dan dokumen profil operasional untuk skenario pengujian berdasarkan kondisi aktual. UAT juga memerlukan dan melibatkan calon pengguna sebagai penguji utama untuk menguji sistem dalam kasus penggunaan pengguna asli. UAT memerlukan uji coba di dalam berbagai kondisi penggunaan, termasuk penanganan input yang tidak valid dan skenario *real-world usage* [1]. Hal ini penting karena sistem harus mampu menangani variasi besar dalam cara pengguna berinteraksi dengannya, baik dari segi teknis maupun fungsional. Pengujian ini memastikan perangkat lunak dapat memenuhi kebutuhan pengguna di berbagai situasi, seperti perubahan lingkungan teknis, beban kerja tinggi, atau skenario operasional yang jarang terjadi. Selain itu, pengujian dengan berbagai kondisi membantu mendeteksi dan mencegah kegagalan yang mungkin tidak terlihat dalam pengujian sebelumnya [1].

2.3. Crowdsourcing Software Testing

Crowdsourcing awalnya adalah konsep pengalihdayaan aktivitas korporat ke sejumlah massa independen. Secara kolektif, sejumlah massa tersebut mengambil alih tugas-tugas korporat yang sebelumnya, seperti contoh penyelesaian *Research Question* atau *Pattern Recognition*, dan tugas-tugas tersebut bisa diselesaikan dengan biaya yang lebih murah dibandingkan *Expert Hiring* atau penggunaan mesin [16]. Dalam konteks perangkat lunak, metode crowdsourcing telah digunakan untuk pengujian perangkat lunak (*Software Testing*) dengan istilah Crowdsourcing Software Testing (CST). Dalam CST, perusahaan atau pengembang perangkat lunak memanfaatkan kerumunan (crowd)—sekelompok individu dengan latar belakang, keterampilan, dan pengalaman yang beragam—untuk melakukan berbagai jenis pengujian perangkat lunak. CST memiliki beberapa keunggulan, seperti efisiensi biaya, karena memungkinkan perusahaan mengurangi biaya jika dibandingkan dengan *Expert Hiring* atau melibatkan tim pengujian internal [17], dan keanekaragaman pengujian, karena metode ini *allows various tester backgrounds*, yang memberikan hasil pengujian yang lebih representatif dan

beragam. Proses CST biasanya melibatkan tiga komponen, antara lain adalah *requester*, *tester*, dan *platform*.

Dalam pelaksanaan CST, kualitas *tester* menjadi sangat penting untuk memastikan kualitas pengujian. Terdapat 3 (tiga) faktor utama yang mempengaruhi pemilihan *tester* yang berkualitas. Faktor yang pertama adalah faktor *personality*. Faktor ini mengacu pada kepribadian dan perilaku tester yang dapat mempengaruhi kinerja mereka dalam pengujian. Sebagai contoh S. Alyahya (2020) mengatakan bahwa *tester* dengan kepribadian ekstrovert cenderung lebih baik dalam teknik Black-Box Testing, sedangkan introvert unggul dalam White-Box Testing [17]. Kemudian faktor kedua adalah kapabilitas *tester*. Faktor ini biasanya dievaluasi berdasarkan kinerja terdahulu, atau dengan kata lain bersifat historis. Faktor ketiga adalah Profile. Faktor ini berkaitan juga dengan *domain knowledge* yang dimiliki oleh *tester* yang mempengaruhi performa pengujian. Atribut lain dalam faktor ini adalah ketersediaan waktu, kemampuan bahasa, dan spesifikasi teknis *tester*.

2.4. Algoritma M-1

Algoritma M-1 adalah dasar dari algoritma M-X. Algoritma M-1 dirancang untuk mengevaluasi kualitas *workers* pada masalah pilihan tunggal (single-choice) dan biner[9]. Algoritma ini dirancang untuk mengukur akurasi kualitas *workers* berdasarkan konsistensi jawaban mereka terhadap serangkaian tugas. Algoritma M-1 menghitung akurasi kualitas *workers* berdasarkan kesamaan jawaban antara beberapa *workers* pada tugas yang sama tanpa memerlukan jawaban yang sudah diketahui sebelumnya (*pre-developed answers-question*) [9]. *Pre-developed answer-question* adalah pertanyaan dengan jawaban yang tidak perlu dipersiapkan terlebih dahulu, alih-alih kebenaran dari pertanyaan tersebut ditentukan dari konsistensi jawaban sekelompok orang yang menjawab pertanyaan tersebut. M-X bekerja dengan paling sedikit tiga *workers* untuk setiap tugas.

Secara matematis, algoritma M-1 ditulis sebagai berikut:

$$Q_{ij} = A_i \cdot A_j + \frac{1}{M+1} \cdot (1 - A_i) \cdot (1 - A_j) \cdot (M - 1) \quad (2.1)$$

Di mana:

Q_{ij} : Probabilitas bahwa pasangan workers i dan j memberikan jawaban yang sama untuk sebuah tugas [9]

M : Jumlah opsi pada pertanyaan (biasanya ini adalah jumlah pilihan dalam soal pilihan ganda). [9]

A_i : Probabilitas bahwa worker i memberikan jawaban yang benar. [9]

A_j : Probabilitas bahwa worker j memberikan jawaban yang benar. [9]

Formula ini mendefinisikan dua komponen utama. Komponen pertama, $A_i \cdot A_j$, adalah probabilitas bahwa sepasang *workers* secara bersamaan memberikan jawaban yang benar. Komponen kedua, $\frac{1}{M+1} \cdot (1 - A_i) \cdot (1 - A_j) \cdot (M - 1)$, adalah probabilitas bahwa sepasang *workers* memberikan jawaban yang sama secara acak ketika jawaban mereka tidak benar. Dengan menggunakan Q_{ij} yang dihitung dari data respons *workers*, algoritma ini menyusun sistem persamaan linear yang menyertakan hubungan antar *workers* [9]. Sistem persamaan ini diselesaikan untuk menentukan nilai akurasi A_i setiap *workers*. Selanjutnya, akurasi masing-masing *workers* dihitung sebagai rata-rata dari konsistensi mereka terhadap *workers* lain pada semua tugas yang dikerjakan bersama. Probabilitas Q_{ij} berfokus pada menghitung kesamaan jawaban antara dua *workers* i dan j secara individu [9]. Setiap pasangan *workers* dibandingkan, tetapi algoritma secara keseluruhan membutuhkan minimal 3 *workers* untuk menciptakan korelasi yang cukup untuk menyelesaikan sistem persamaan linear. Dengan lebih banyak *workers*, sistem memiliki redundansi data untuk menghasilkan nilai akurasi yang lebih stabil.

2.5. Algoritma M-X

Algoritma M-X adalah perluasan dari algoritma M-1 yang mampu mengevaluasi kualitas pekerja dalam tugas crowdsourcing dengan tipe *multiple-choice*. Dalam algoritma ini, setiap pertanyaan *multiple choice* dengan M opsi diubah menjadi sejumlah M sub-pertanyaan *single choice*. Setiap sub-pertanyaan ini kemudian diperlakukan sebagai soal dengan dua pilihan: memilih opsi tersebut

atau tidak. Dengan demikian, setiap pekerja dinilai berdasarkan tingkat konsistensi jawabannya untuk masing-masing opsi.

Proses algoritma M-X dilakukan dalam beberapa tahap

1. **Konversi Tugas:** Setiap pertanyaan multiple-choice dibagi menjadi sub-pertanyaan *single choice*, di mana setiap opsi dianggap sebagai pertanyaan terpisah.
2. **Evaluasi Sub-tugas:** Menggunakan algoritma M-1 untuk mengevaluasi kualitas pekerja di setiap sub-tugas.
3. **Penggabungan Akurasi:** Menghitung akurasi komprehensif tiap pekerja dengan mengalikan hasil akurasi dari seluruh opsi yang telah dihitung.

Secara formal, akurasi pekerja pada algoritma M-X dihitung dengan rumus berikut:

$$A_i = \prod_{j=1}^M A_{ij} \quad (2.2)$$

di mana A_{ij} adalah akurasi pekerja i pada sub-pertanyaan atau opsi j [9]. Pendekatan ini memastikan bahwa pekerja harus benar dalam semua opsi yang benar untuk dianggap akurat pada suatu pertanyaan *multiple choice* secara keseluruhan.

2.6. Prior-Experience Based Validation

Prior Experience-Based Validation merupakan pendekatan yang memanfaatkan riwayat performa dan keahlian domain sebagai dasar evaluasi kesesuaian crowdworker. Dalam konteks pengujian UAT berbasis *crowdsourcing*, metode ini mengintegrasikan data historis kinerja penguji, verifikasi keahlian domain, dan konsistensi pola respons untuk membentuk kriteria validasi yang handal [13]. Pemilihan penguji berdasarkan pengalaman terdahulu menawarkan beberapa keunggulan strategis. Pengalaman pengujian yang terdokumentasi memberikan indikator yang kuat mengenai kemampuan penguji dalam menemukan bug, sebagaimana dinyatakan oleh Cui et al. bahwa "pengalaman deteksi bug dari

pekerja yang dipilih menjamin lebih banyak pekerja berpengalaman yang harus dipilih untuk melaksanakan tugas pengujian, yang penting dalam *crowdsourced testing*" [18]. Pekerja dengan pengalaman substansial cenderung lebih mahir dalam mendeteksi permasalahan krusial pada aplikasi.

Selain itu, pendekatan ini memperhitungkan faktor pengetahuan domain sebagai komponen evaluasi penguji. Wang et al. mengidentifikasi bahwa "pekerja akan lebih mungkin untuk mendeteksi bug ketika melakukan tugas yang mereka kenal" [13], menegaskan pentingnya relevansi domain dalam pemilihan penguji. Pengetahuan domain yang sesuai memungkinkan penguji untuk lebih efektif dalam mengeksplorasi fungsionalitas sistem dan mengidentifikasi kelemahan berdasarkan pemahaman mereka tentang konteks penggunaan.

Keragaman karakteristik pengujian yang dapat diakomodasi merupakan keunggulan lain dari metode ini. Dengan mengevaluasi penguji berdasarkan rekam jejak dan keahlian domain, pendekatan ini dapat diterapkan pada berbagai jenis proyek pengujian dengan karakteristik yang berbeda tanpa bergantung pada struktur pengujian yang rigid [18]. Fleksibilitas ini memungkinkan adaptasi terhadap berbagai lingkungan pengujian sambil tetap mempertahankan standar kualitas yang tinggi. Dengan mempertimbangkan kapabilitas penguji dari multi-dimensi yang mencakup konteks pengujian, kapabilitas, dan pengetahuan domain [13], Prior Experience-Based Validation memberikan kerangka kerja yang komprehensif untuk memastikan pemilihan penguji yang optimal dalam lingkungan *crowdsourced testing*.

2.7. Confusion Matrix

Confusion Matrix adalah representasi tabular yang menunjukkan performa model klasifikasi dengan membandingkan hasil prediksi dengan data aktual [19]. Matriks ini menyediakan kerangka kerja yang komprehensif untuk menganalisis seberapa baik sebuah algoritma klasifikasi dapat membedakan antara kelas biner [19]. Dalam konteks evaluasi tester untuk *crowdsourced testing*, *confusion matrix* menjadi alat yang penting untuk mengukur efektivitas algoritma dalam mengklasifikasikan

tester yang memenuhi syarat dan yang tidak memenuhi syarat [20]. *Confusion Matrix* memanfaatkan empat istilah penting dalam evaluasi klasifikasi:

1. **True Positive (TP):** Merupakan jumlah instance yang secara aktual termasuk dalam kelas positif dan berhasil diprediksi dengan benar oleh sistem sebagai kelas positif.
2. **True Negative (TN):** Merupakan jumlah instance yang secara aktual termasuk dalam kelas negatif dan berhasil diprediksi dengan benar oleh sistem sebagai kelas negatif.
3. **False Positive (FP):** Merupakan jumlah instance yang secara aktual termasuk dalam kelas negatif tetapi secara keliru diprediksi oleh sistem sebagai kelas positif.
4. **False Negative (FN):** Merupakan jumlah instance yang secara aktual termasuk dalam kelas positif tetapi secara keliru diprediksi oleh sistem sebagai kelas negatif.

Berdasarkan nilai TP, TN, FP, dan FN ini, beberapa metrik evaluasi utama yang digunakan dalam penelitian ini antara lain adalah:

1. Accuracy

Accuracy merupakan ukuran seberapa tepat keseluruhan prediksi yang dilakukan oleh sebuah model klasifikasi. Metrik ini menunjukkan proporsi dari total prediksi yang benar (baik positif maupun negatif) dibandingkan dengan seluruh jumlah instance yang diuji. Semakin tinggi nilai accuracy, semakin baik model dalam mengklasifikasikan data secara keseluruhan. Rumus accuracy ditulis sebagai berikut:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

(2.3)

2. Precision

Precision mengukur tingkat ketepatan model dalam mengklasifikasikan suatu instance sebagai kelas positif. Dengan kata lain, precision menunjukkan berapa banyak dari semua instance yang diprediksi sebagai positif benar-benar merupakan instance positif. Metrik ini penting dalam konteks di mana kesalahan klasifikasi positif (false positive) perlu diminimalkan. Precision dihitung dengan rumus sebagai berikut:

$$Accuracy = \frac{TP}{TP + FP}$$
(2.4)

3. Recall

Recall (juga dikenal sebagai sensitivitas) mengukur kemampuan model dalam menemukan semua instance yang benar-benar termasuk dalam kelas positif. Metrik ini menunjukkan proporsi instance positif yang berhasil diidentifikasi dengan benar oleh model. Recall sangat penting dalam situasi di mana kegagalan mendeteksi instance positif (false negative) dapat berdampak besar. *Recall* dihitung dengan rumus berikut:

$$Accuracy = \frac{TP}{TP + FN}$$
(2.5)

4. True Negative Rate (TNR)

True Negative Rate, yang juga dikenal sebagai spesifitas, adalah metrics yang mengukur proporsi kasus negatif yang berhasil diidentifikasi dengan benar oleh model. TNR menunjukkan kemampuan model dalam menghindari kesalahan positif palsu. Nilai TNR yang tinggi menunjukkan bahwa model mampu dengan baik mengidentifikasi kasus-kasus negatif dan jarang memberikan peringatan palsu.

$$TNR = \frac{TN}{TN + FP}$$
(2.6)

5. F1 Score

F1 Score adalah rata-rata harmonik dari *precision* dan *recall*, yang memberikan keseimbangan antara kedua metrik tersebut. F1 Score sangat berguna terutama saat dataset tidak seimbang (imbalanced), di mana jumlah data positif dan negatif sangat berbeda.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (2.7)$$

Dengan menggunakan ketiga metrik tersebut, penelitian ini berpotensi secara objektif mengevaluasi sejauh mana algoritma M-X efektif dalam menyaring dan mengklasifikasikan pengujian berdasarkan konsistensi jawaban mereka dalam skenario UAT berbasis *crowdsourcing*. Kombinasi dari *accuracy*, *precision*, dan *recall* memberikan gambaran komprehensif mengenai performa algoritma.

2.8. Gherkin Syntax

Gherkin merupakan *Domain Specific Language* (DSL) untuk menulis spesifikasi perilaku sistem dengan cara yang mudah dibaca oleh manusia. Sintaks Gherkin menggunakan struktur bahasa alami dengan format “*Given-When-Then*” yang memungkinkan pengembang dan *workers*, atau dalam hal ini pengujian untuk memahami dan memverifikasi skenario perilaku sistem dengan konkret [21].

Gherkin dipilih sebagai format penulisan skenario *User Acceptance Testing* (UAT) karena memungkinkan penulisan kasus uji (*test case*) yang eksplisit, terstruktur, dan terverifikasi berdasarkan skenario yang relevan dengan pemahaman pembaca, atau dalam hal ini *workers* sebagai pengujian. Sintaks ini sangat membantu dalam perancangan skenario pengujian yang realistis, karena menyajikan penyusunan *task* yang jelas dan konsisten. [22]

Berikut adalah contoh penulisan kasus uji menggunakan format Gherkin dengan menggunakan operator yang disediakan.

Tabel 2.1 Format Gherkins Scenario

Format Gherkin	Contoh Narasi Skenario
Feature	Validasi fitur pencarian pekerjaan.
Scenario	Pencarian pekerjaan berdasarkan kata kunci.
Given	user membuka halaman daftar pekerjaan.
When	user mengetik "desain grafis" di kolom pencarian.
And	menekan tombol cari
Then	sistem menampilkan daftar pekerjaan yang mengandung kata "desain grafis"

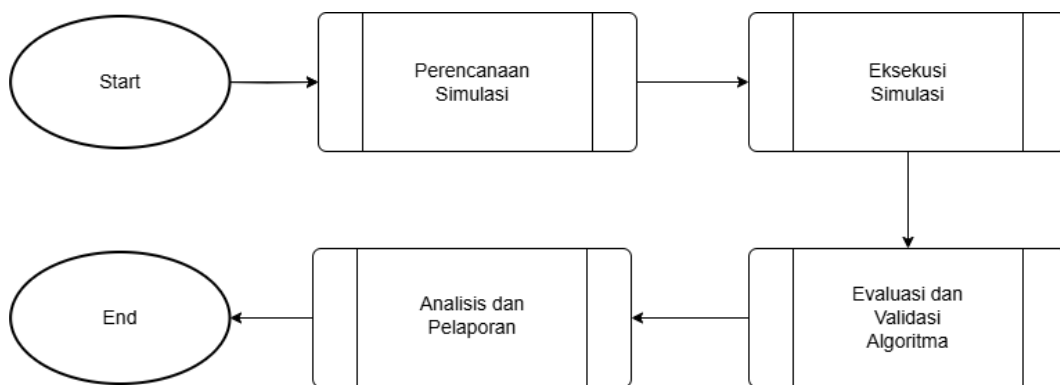
Dimana operator tersebut menyatakan:

- *Feature*: merepresentasikan fungsionalitas spesifik dari perangkat lunak. [22]
- *Scenario*: mengidentifikasi peristiwa yang merealisasikan sebagian dari fitur tersebut. [22]
- *Given*: menentukan prasyarat atau kondisi awal yang dibutuhkan untuk menjalankan skenario. [22]
- *When*: menjelaskan tindakan atau peristiwa yang terjadi selama proses eksekusi atau pengujian. [22]
- *And*: berfungsi sebagai penghubung logis yang digunakan untuk menggabungkan beberapa elemen prasyarat, tindakan, atau hasil yang kompleks, jika elemen tersebut menyatakan kondisi negatif, bisa menggunakan operator *But*. [22]

BAB 3. PERANCANGAN SISTEM DAN METODOLOGI

Sistematika Penyelesaian Masalah

Untuk menyelesaikan masalah dalam penelitian, diperlukan langkah-langkah logis yang terstruktur. Proses penyelesaian masalah ini melibatkan beberapa tahapan utama yang saling terhubung dan berkesinambungan. Tahapan tersebut diurutkan sesuai pada gambar 3.1.



Gambar 3.1 Flowchart Penyelesaian Masalah

3.1. Tahap Perancangan Simulasi

Untuk mencapai hasil simulasi yang optimal dan tepat sasaran, maka rancangan simulasi adalah sebagai berikut:

3.1.1 Tahap Perancangan Skenario Simulasi

Simulasi diadakan secara *hybrid*, untuk mempermudah aksesibilitas [23]. Kemudian akan diadakan 3 hari, dan dibagi menjadi 2 tipe, yaitu asinkronus dan sinkronus. Simulasi diadakan bersama 24 orang relawan. Relawan tersebut diminta untuk mengerjakan task Gherkin yang sudah disesuaikan dengan persyaratan yang ada di algoritma M-X, sehingga algoritma M-X akan berjalan dengan tepat dan andal.

3.1.2 Tahap Perancangan Data Source Simulasi

Sumber data hasil dari simulasi ini yang berkaitan dengan data diri volunteer, *actual eligibility* mereka yang diukur dengan *Prior-Experience Based Validation* dan hasil simulasi luaran dari algoritma M-X

3.1.3 Tahap Perancangan Komponen Penting Simulasi

Gherkins Task Design

Rancangan skenario Task Gherkins dalam simulasi harus selaras dengan ketentuan yang ditetapkan dalam algoritma M-X untuk mengoptimalkan implementasi algoritma M-X, maka dari itu, perlu dilakukan penyesuaian berdasarkan aturan-aturan berikut:

1. Format Pilihan Ganda

Setiap *task* harus memiliki sejumlah pilihan tetap (M opsi), di mana masing-masing pilihan dapat dievaluasi secara independen [9]. Algoritma M-X dirancang untuk mengevaluasi respons pada pertanyaan *multiple-choice*. Format dengan sejumlah pilihan tetap (M opsi) memungkinkan algoritma mengukur konsistensi jawaban antar pengujian dengan cara yang terstruktur.

2. Dekonstruksi Biner

Setiap *task* harus dapat diubah menjadi sejumlah M sub-pertanyaan biner bertipe *closed question* yang terpisah, masing-masing merepresentasikan satu opsi [9]. Hal ini sejalan dengan alur algoritma M-X yang akan mengolah setiap opsi sebagai *closed question* terpisah, menerapkan algoritma M-1 pada masing-masing, lalu menggabungkan keseluruhan hasil untuk mendapatkan skor kualitas pengujian. Pendekatan ini diperlukan karena algoritma dasar M-1 hanya bisa menangani pertanyaan biner, sehingga memungkinkan evaluasi yang lebih teliti terhadap konsistensi jawaban pengujian untuk setiap pilihan.

3. Kemandirian Antar Opsi

Setiap opsi harus berdiri sendiri secara logis, tanpa adanya hubungan hierarkis atau saling meniadakan antar opsi [9]. Kemandirian opsi sangat penting karena algoritma ini

mengevaluasi worker secara terpisah untuk setiap opsi. Jika opsi saling bergantung atau eksklusif (misalnya jika memilih opsi A berarti tidak dapat memilih opsi B), maka asumsi dasar algoritma tentang probabilitas pilihan akan terganggu. Opsi-opsi yang independent memastikan bahwa algoritma mengukur akurasi pekerja pada setiap dimensi secara terpisah, lalu menggabungkannya menjadi skor keseluruhan yang valid dengan persamaan produktif seperti yang dijelaskan dalam rumus $A_i = \prod A_{ij}$.

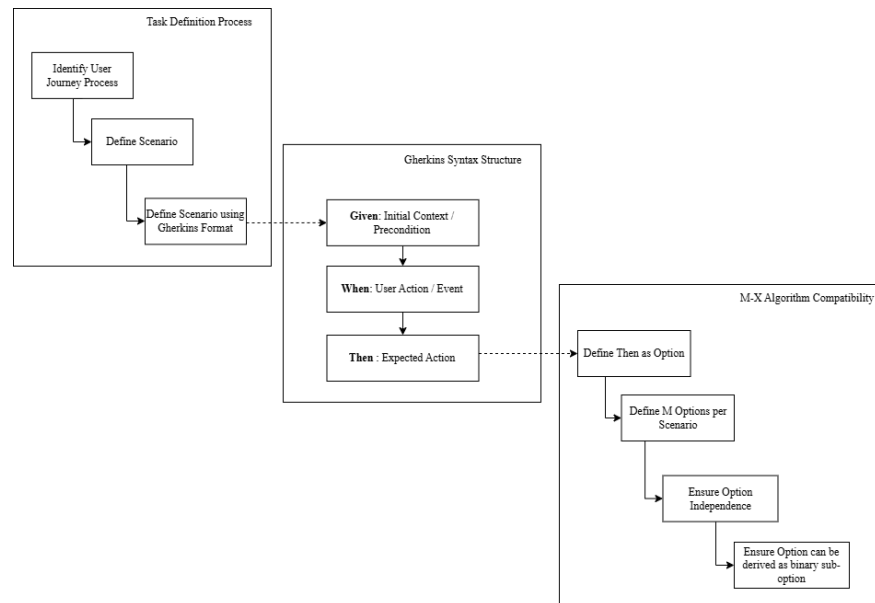
4. Tanpa Kebenaran Mutlak

Algoritma M-X tidak bergantung pada jawaban benar yang telah ditentukan sebelumnya, melainkan mengandalkan pola kesepakatan antar penguji [9]. Hal ini menjadikan algoritma ini sangat berguna untuk kasus UAT yang tidak memiliki "jawaban benar" absolut, tetapi melihat bentuk eksplorasi pengguna / penguji.

5. Distribusi Acak atas Jawaban Salah

Diasumsikan bahwa jika seorang penguji menjawab salah, ia akan memilih secara acak di antara opsi-opsi yang salah, dengan peluang yang sama untuk setiap opsi [9]. Asumsi ini memungkinkan algoritma untuk mengevaluasi konsistensi jawaban tanpa perlu mempertimbangkan kecenderungan atau preferensi penguji terhadap opsi tertentu, sehingga penilaian kualitas penguji lebih objektif dan fokus pada pola kesamaan jawaban antar penguji.

Dengan demikian, berikut adalah model desain perancangan Gherkins Task yang mengikuti aturan di atas.



Gambar 3.2 Model Desain Task Gherkin

Dari model tersebut, diturunkan penjelasan sebagai berikut:

1. **Identifikasi Perjalanan Pengguna (*User Journey Identification*)**
Pola interaksi inti pengguna dengan perangkat lunak dipetakan untuk menentukan ruang lingkup skenario pengujian.
2. **Pendefinisian Skenario (*Scenario Definition*)**
Setiap pola interaksi disusun sebagai skenario pengujian yang terpisah.
3. **Pemformatan Gherkin (*Gherkin Formatting*)**
Skenario-skenario tersebut diformat menggunakan struktur Gherkin “*Given-When-Then*” sebagai berikut:
 - **Given:** Menetapkan konteks awal atau prasyarat skenario
 - **When:** Menjelaskan aksi atau kejadian spesifik yang memicu skenario
 - **Then:** Menyatakan hasil atau keluaran yang diharapkan
4. **Integrasi Kesesuaian dengan M-X (*M-X Compatibility Integration*):**
Pernyataan “Then” diubah menjadi bentuk evaluasi berbasis opsi:

1. Setiap pernyataan “Then” didefinisikan sebagai sebuah opsi yang dapat dipilih oleh penguji
2. Opsi sejumlah M ditentukan untuk setiap skenario guna memenuhi kebutuhan algoritma
3. Kemandirian antar opsi diverifikasi untuk memastikan kompatibilitas dengan dekomposisi biner
4. Setiap opsi dirumuskan agar dapat dievaluasi sebagai sub-opsi biner yang memuat nilai benar atau salah.

Implementasi Praktis Gherkin Task

Agar desain Gherkin Task dapat digunakan dengan optimal, berikut adalah aturan praktis yang dapat diikuti oleh *volunteer* simulasi:

1. Setiap skenario UAT disajikan kepada para penguji dengan deskripsi lengkap dalam format Gherkin (*Given-When-Then*).
2. Untuk komponen “*Then*”, para penguji diberikan M opsi yang merepresentasikan kemungkinan hasil dari skenario tersebut.
3. Penguji dapat memilih satu atau lebih opsi yang mereka anggap benar untuk setiap skenario.
4. Setiap skenario dievaluasi oleh minimal tiga penguji untuk memenuhi syarat minimum yang diperlukan oleh algoritma $M-X$.
5. Jawaban dicatat dalam bentuk vektor biner (1 = dipilih, 0 = tidak dipilih) untuk setiap opsi.
6. Tidak ada jawaban acuan yang diberikan kepada penguji, sehingga menjaga independensi algoritma terhadap kebenaran yang telah ditentukan sebelumnya.

Penentuan Threshold Kelayakan (Eligibility Threshold)

Dalam penelitian ini, digunakan *Rule-Based Validation* untuk menentukan threshold yang memberi nilai batas yang membedakan *eligible worker* dan tidak. *Rule-Based Validation* merupakan pendekatan yang memungkinkan pengguna untuk mendefinisikan dan menyesuaikan threshold secara manual, memberikan fleksibilitas dalam menentukan klasifikasi kelayakan workers. Berikut adalah Kelebihan *Rule-Based Validation*

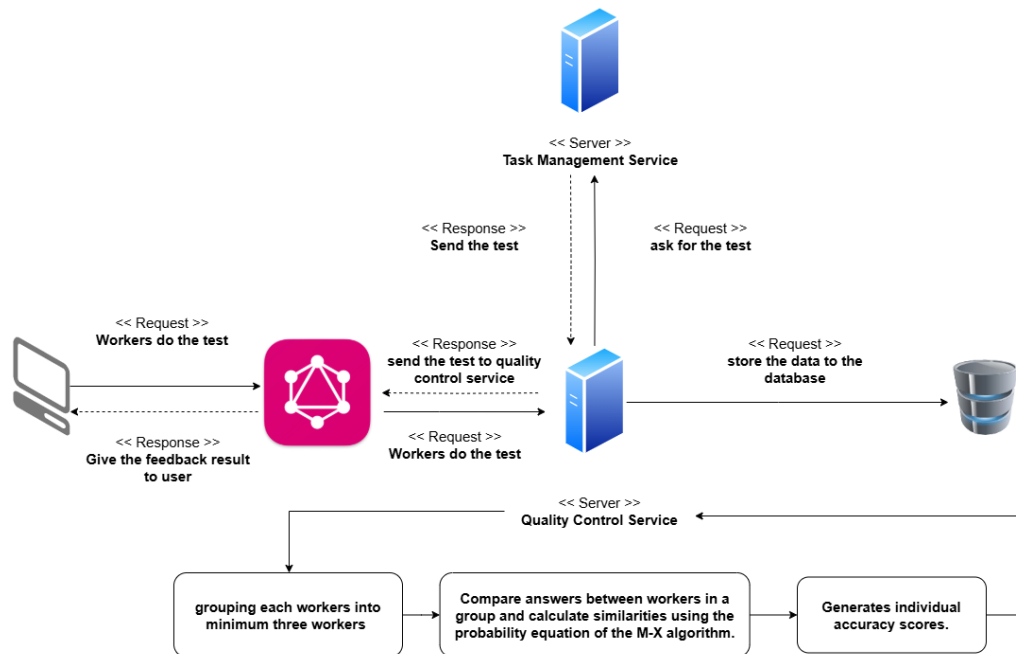
1. Memungkinkan penetapan nilai threshold secara manual.
2. Lebih fleksibel dibanding threshold tetap atau otomatis.
3. Bisa disesuaikan dengan karakteristik proyek atau kebutuhan pengujian tertentu [24].

Dibandingkan dengan pendekatan threshold tetap atau otomatis, *Rule-Based Validation* memberikan kontrol lebih besar kepada pengguna untuk menyesuaikan kriteria kelayakan sesuai dengan konteks pengujian. Berikut adalah implementasi dari penggunaan *threshold* ini:

1. menginput nilai threshold melalui antarmuka.
2. Sistem membandingkan nilai akurasi dari algoritma M-X dengan nilai threshold yang ditentukan.

Kemudian, akan dihitung *eligibility* dari rata-rata nilai eligibilitas setiap tugas. Tujuannya untuk menilai konsistensi performa worker secara menyeluruh. Pendekatan ini diharapkan membuat hasil evaluasi lebih adil, objektif, dan tahan terhadap variasi antar tugas dalam skenario UAT berbasis *crowdsourcing*.

3.1.4 Tahap Perancangan Platform Simulasi



Gambar 3.3 Solution Architecture Sistem dengan Implementasi Algoritma M-X

Website platform dirancang sebagai berikut:

- Komunikasi Client-Server:** Client (antarmuka pengguna) mengakses sistem melalui GraphQL *gateway*, yang berfungsi sebagai titik masuk tunggal untuk semua permintaan data. Pendekatan ini memungkinkan client untuk mendapatkan data yang dibutuhkan secara spesifik dan efisien dalam satu permintaan, mengurangi overhead jaringan dan meningkatkan responsivitas aplikasi.
- Implementasi GraphQL:** Platform menggunakan Apollo GraphQL sebagai library implementasi, yang menyediakan ekosistem lengkap untuk mengelola data, termasuk manajemen state, caching, dan penanganan error. Apollo GraphQL dipilih

karena kemampuannya menangani operasi kompleks dan mendukung pengembangan yang skalabel.

- c) **Mekanisme Komunikasi:** Arsitektur ini memungkinkan komunikasi dua antara *client* dan *server*. *Client* dapat mengirimkan *queries* (permintaan data) dan *mutations* (perubahan data) melalui GraphQL, sementara *server* mengembalikan respons terstruktur sesuai dengan skema yang telah didefinisikan. Pendekatan ini memudahkan pengembangan fitur-fitur interaktif dan *real-time* yang dibutuhkan dalam platform simulasi.
- d) **Penempatan Algoritma:** Algoritma M-X diimplementasikan pada *Quality Control Service* di sisi server. Penempatan ini memastikan bahwa proses komputasi dilakukan di *server*, sementara client hanya bertugas menampilkan hasil. Algoritma ini diintegrasikan secara langsung dengan lapisan servis untuk mengakses data dari MongoDB dan mengembalikan hasil pemrosesan melalui GraphQL *resolver*.

Komponen Sistem

Dalam rangka menunjang berjalan dengan optimalnya simulasi yang akan diadakan, maka platform simulasi harus dengan menyeluruh menyediakan wadah yang dapat dengan konsisten menyimpan dan mengolah data relawan sebagai pengguna platform tersebut. Berikut adalah tabel yang mendefinisikan komponen yang diperlukan:

Tabel 3.1 Tabel Komponen Platform Simulasi

Komponen	Fungsi
Worker Management Service	Menangani pendaftaran relawan, manajemen profil, dan pelacakan kualifikasi

Authentication and Authorization Service	Mengelola proses otentikasi dan otorisasi
Task Management Service	Mengontrol penugasan tugas, pengiriman, dan validasi
Quality Control Service	Menerapkan algoritma M-X untuk penilaian dan klasifikasi kualitas pekerja
GraphQL Gateway	memfasilitasi pertukaran data yang efisien antara komponen klien dan server
MongoDB Database	Menyimpan data relawan, response, dan hasil klasifikasi

Input Data

Sistem menerima data berupa respons dari *workers* terhadap skenario pengujian yang telah disediakan oleh *Task Management Service*. Setiap *worker* diberi tugas yang sama, dan jawaban mereka direkam dalam MongoDB melalui GraphQL Gateway. Data ini mencakup:

- a. identitas tugas (*TaskId*),
- b. identitas *workers* (*WorkerId*), dan
- c. jawaban yang diberikan beserta id-nya.

Implementasi Algoritma

Proses implementasi algoritma dilakukan oleh *Quality Control Service*, di mana algoritma M-X digunakan untuk menilai tingkat konsistensi jawaban antar *workers* yang mengerjakan tugas yang sama. Algoritma M-X menghitung probabilitas konsistensi dengan membandingkan setiap pasangan *workers*. Jika jawaban dari dua *workers* pada suatu tugas sama, maka algoritma mencatatnya sebagai konsisten; jika tidak, dianggap tidak konsisten. Data yang diproses ini digunakan untuk menghitung skor akurasi individual setiap *worker* berdasarkan persamaan probabilitas yang ditentukan.

Perancangan Pseudocode Algoritma M-X

Secara algoritmik, algoritma M-X dapat dituliskan sebagai berikut:

Input:

- N multiple-choice tasks (each with M answer options)
- Set of Workers $W = \{w_1, w_2, \dots, w_n\}$, where $n \geq 3$
- Worker responses denoted as S_{ikj} , where $S_{ikj} = 1$ if worker w_i selected option j for task k , and 0 otherwise

Output: Accuracy rate A_i of each worker w_i

1. **For** each task k from 1 to N **do**:
2. **For** each answer option j from 1 to M **do**:
3. **Create** binary sub-task based on answer option j (selected or not selected).
4. **For** each pair of workers (w_i, w_j), where $i \neq j$ **do**:
5. **Calculate** $X_{ij}(k,j) = 1$ if S_{ikj} equals S_{jkj} ; otherwise, $X_{ij}(k,j) = 0$.
6. **End for**
7. **End for**
8. **End for**
9. **For** each pair of workers (w_i, w_j), where $i \neq j$ **do**:
10. **Compute** Q_{ij} using the following formula:

$$Q_{ij} = (\text{Total number of matching answers between } w_i \text{ and } w_j) / (N \times M)$$
11. **End for**
12. **Use** the following set of equations to **calculate** the **accuracy** A_i of each worker:

$$Q_{ij} = A_i \times A_j + [(1 - A_i)(1 - A_j)] / (M - 1)$$
13. **Solve** this system of equations using a **numerical method** (e.g., iterative methods or least squares) to **determine** each A_i .
14. **Calculate** the **overall accuracy** for each worker on the multiple-choice tasks by:

$$A_i = \Pi (A_{ij} \text{ for all options } j \text{ from } 1 \text{ to } M)$$
15. **Return** the **accuracy** A_i for each worker.

Tabel 3.2 Tabel Tahapan Implementasi Algoritma M-X

Baris	Tahapan Algoritma	Penjelasan
1-8	Konversi Tugas ke Sub-Tugas Biner	Setiap opsi pada tugas multiple-choice diubah menjadi sub-tugas biner (1 = dipilih, 0 = tidak dipilih).
	Perhitungan Kesamaan Jawaban Antar Pekerja	Untuk setiap sub-tugas, sistem menghitung kesamaan jawaban antar pasangan pekerja. Jika sama (dua-duanya memilih atau tidak memilih), nilai = 1; jika tidak, 0.
9-11	Perhitungan Probabilitas Kesamaan (Q_{ij})	Nilai Q_{ij} dihitung dari total kesamaan jawaban antar pasangan pekerja, dinormalisasi dengan jumlah tugas dan jumlah opsi.
12-13	Penyelesaian Sistem Persamaan Non-Linear	Nilai akurasi pekerja (A_i) dihitung dari Q_{ij} dengan menyelesaikan persamaan non-linear menggunakan metode numerik (iteratif atau least squares).
14-15	Penggabungan Nilai Akurasi Sub-Tugas menjadi Akurasi Final	Nilai akurasi akhir diperoleh dengan mengalikan akurasi dari setiap sub-tugas. Hasil akhir digunakan untuk menentukan apakah pekerja dianggap <i>eligible</i> .

3.2 Tahap Eksekusi Simulasi

Simulasi dilakukan selama 3 hari dengan bertahap. Relawan dari simulasi ini didominasi oleh mahasiswa Telkom University dengan dan tanpa pengalaman di User Acceptance Testing.



Gambar 3.4 Dokumentasi Hari Pertama Simulasi



Gambar 3.5 Dokumentasi Hari Kedua Simulasi



Gambar 3.6 Dokumentasi Hari Ketiga Simulasi

Berdasarkan *Prior-Experience Based Validation*, persebaran *eligibility* relawan dilihat dari pengalaman dan pengetahuan terdahulu mengenai pengujian perangkat lunak, lebih spesifiknya adalah *User Acceptance Testing*.

Selaras dengan batasan masalah, relawan mayoritas adalah mahasiswa Telkom University dan berumur rentang 20-24 tahun. Dari 24 relawan, 12 memiliki pengalaman dalam User Acceptance Testing, sementara 12 lainnya tidak memiliki pengalaman tersebut. Sebagian besar relawan (17 orang) berasal dari program studi terkait teknologi informasi seperti Rekayasa Perangkat Lunak, Informatika, dan Sistem Informasi, dengan 7 orang yang memiliki latar belakang spesifik dari program studi Rekayasa Perangkat Lunak.

Tabel 3.3 Tabel Data Relawan

ID	Nama Relawan	Biografi	Pengalaman
R1	Tio Haidar Hanif	Mahasiswa, 22 tahun	Mahasiswa RPL, pernah melakukan User Acceptance Testing, pernah mengambil mata kuliah Pengujian Perangkat Lunak.
R2	Dede Rahmat Fitriansyah	Mahasiswa, 23 tahun	Mahasiswa IF, belum pernah melakukan User Acceptance Testing, belum pernah mengambil mata kuliah Pengujian Perangkat Lunak, berfokus di <i>front-end development</i> .
R3	Muhammad Ramdhan Fitra Hidayat	Mahasiswa, 22 tahun	Mahasiswa IF, berfokus di <i>competitive programming</i> , belum pernah ada pengalaman pengujian perangkat lunak.
R4	Rizqi Firdaus	Mahasiswa, 23 tahun	Mahasiswa RPL, pernah melakukan User Acceptance Testing, pernah mengambil mata kuliah Pengujian Perangkat Lunak.
R5	Matthew Theodore	Mahasiswa, 21 tahun	Mahasiswa RPL, pernah melakukan User Acceptance Testing, pernah mengambil mata kuliah Pengujian Perangkat Lunak.
R6	Rafidhia Haikalpasya	Cyber Security Specialist, 22 tahun	Cyber Security Specialist, berfokus di <i>white hat cyber security</i> .
R7	Jaatsiyah Dina Astrianto	Mahasiswi, 22 tahun	Mahasiswi RPL, pernah melakukan User Acceptance Testing, pernah mengambil mata kuliah Pengujian Perangkat Lunak.
R8	Vidya Ratmayanti	Mahasiswi, 22 tahun	Mahasiswi RPL, pernah melakukan User Acceptance Testing, pernah mengambil mata kuliah Pengujian Perangkat Lunak.
R9	Sultan Alexander	Mahasiswa, 21 tahun	Mahasiswa IF, berfokus di <i>front end development</i> , belum pernah ada pengalaman pengujian perangkat lunak.
R10	Regina Maharani	Mahasiswi, 22 tahun	Mahasiswa sisfo, pernah melakukan <i>user acceptance testing</i> tapi bukan dari background <i>software engineering</i> .

R11	Alif Taufiqurrahman	Mahasiswa, 22 tahun	Mahasiswa RPL, pernah melakukan User Acceptance Testing, pernah mengambil mata kuliah Pengujian Perangkat Lunak.
R12	I Made Rai Michael Timonuli	Mahasiswa, 22 tahun	Mahasiswa Sisfo, berfokus di Data and Business Analytics, belum pernah ada pengalaman pengujian perangkat lunak.
R13	Anthony Febrian	Mahasiswa, 22 tahun	Mahasiswa TI, berfokus di Cyber Security, sudah pernah ada pengalaman pengujian perangkat lunak, terutama di User Acceptance Testing
R14	Mohammad William	Mahasiswa, 22 tahun	Mahasiswa IT, belum pernah melakukan User Acceptance Testing, berfokus di <i>computer hardware</i> .
R15	Aji Pandoyo	Mahasiswa, 22 tahun	Mahasiswa Sisfo, berfokus di UI/UX Design, belum pernah ada pengalaman pengujian perangkat lunak.
R16	Putri Auliya Rahmah	Mahasiswi, 22 tahun	Mahasiswa RPL, pernah melakukan <i>User Acceptance Testing</i> , pernah mengambil mata kuliah Pengujian Perangkat Lunak.
R17	Agung Hadi Winoto	Mahasiswa, 22 tahun	Mahasiswa DS, berfokus di Data and AI, belum pernah ada pengalaman pengujian perangkat lunak.
R18	Adam Syamsudin	Mahasiswa, 22 tahun	Mahasiswa IT, berfokus di Business Analytics, belum pernah ada pengalaman pengujian perangkat lunak.
R19	Intan Silvia Dewi	Mahasiswi, 22 tahun	Mahasiswa IT, berfokus di UI/UX <i>Design</i> , sudah pernah ada pengalaman di pengujian perangkat lunak, pernah pengujian <i>User Acceptance Testing</i> .
R20	Valent Reza	Mahasiswa. 21 tahun	Mahasiswa IT, berfokus di Data Analytics, belum pernah ada pengalaman di pengujian perangkat lunak.

R21	Andika Rama	Mahasiswa, 22 tahun	Mahasiswa IT, berfokus di Data Analytics, belum pernah ada pengalaman di pengujian perangkat lunak.
R22	Yosef Baptista De Morin	Mahasiswa, 22 tahun	Mahasiswa IT, berfokus di Front End Development, sudah pernah ada pengalaman di pengujian perangkat lunak, pernah pengujian User Acceptance Testing
R23	Eveline Natalie	Mahasiswi, 22 tahun	Mahasiswa IT, berfokus di Front End Development, sudah pernah ada pengalaman di pengujian perangkat lunak, pernah pengujian User Acceptance Testing
R24	Yekonyah Sadhu Ditho	Siswa, 17 tahun	Siswa SMK RPL, sudah pernah ada pengalaman pengujian perangkat lunak, pernah melakukan User Acceptance Testing.

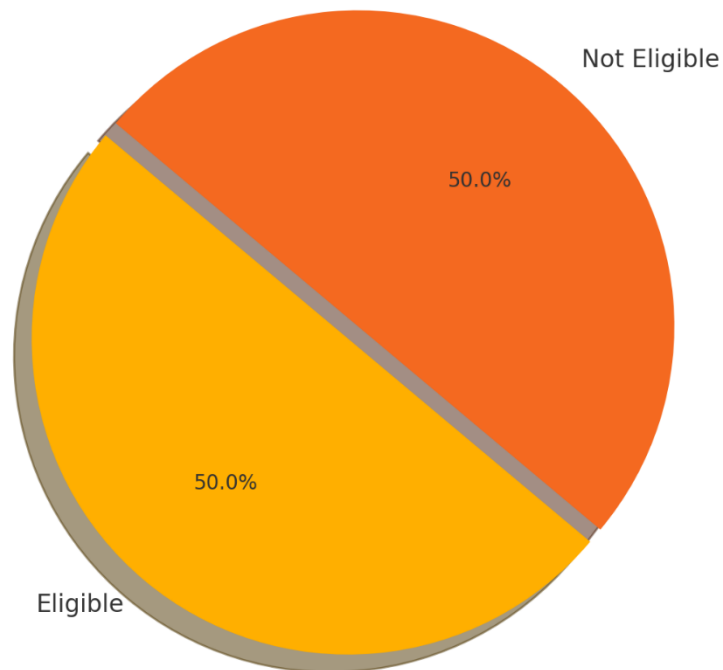
Sedemikian sehingga dapat didefinisikan *actual eligibility* sebagai berikut:

Tabel 3.4 Actual Eligibility Relawan

Status	Himpunan ID
Eligible	[R1, R4, R5, R7, R8, R10, R11, R13, R16, R19, R22, R23]
Not Eligible	[R2, R3, R6, R9, R12, R14, R15, R17, R18, R20, R21, R24]

Perbandingan yang seimbang antara relawan eligible dan non-eligible (12:12) dirancang untuk memfasilitasi evaluasi yang seimbang terhadap kemampuan algoritma M-X dalam mengklasifikasikan penguji. Dengan demikian, rasio yang terbentuk dari pengambilan data actual secara manual memiliki keseimbangan antara data relawan eligible dan relawan tidak eligible.

Rasio Actual Eligible vs Not Eligible (Ground Truth)



Gambar 3.7 Rasio Perbandingan Actual Eligibility Relawan

3.3 Validasi Algoritma dan Evaluasi

Algoritma dievaluasi menggunakan Confusion Matrix untuk menganalisis kinerja klasifikasi M-X secara komprehensif. Pendekatan evaluasi ini diimplementasikan dengan menghitung sejumlah metrik yang memungkinkan pengukuran terhadap efektivitas algoritma.

Untuk menjawab rumusan masalah pertama, penelitian ini mengadopsi pendekatan evaluasi kuantitatif berbasis metrik berikut:

1. **Accuracy (ACC):** Mengevaluasi proporsi total klasifikasi yang benar ($TP + TN$) terhadap seluruh populasi ($TP + TN + FP + FN$), memberikan indikasi performa global algoritma dalam lingkungan pengujian.

2. **Precision (P)**: Mengkuantifikasi reliabilitas positif prediktif algoritma dengan rasio $TP/(TP + FP)$, mengukur kemampuan algoritma untuk meminimalisasi false positive dalam identifikasi penguji eligible.
3. **Recall (R)**: Menilai sensitivitas algoritma melalui rasio $TP/(TP + FN)$, mengindikasikan kapabilitas dalam mendeteksi seluruh penguji eligible dari populasi.
4. **F1 Score**: Menghitung rata-rata harmonik precision dan recall $(2 \times P \times R)/(P + R)$, menyediakan metrik tunggal yang menyeimbangkan kedua aspek kritis tersebut.

Untuk menjawab rumusan masalah kedua, penelitian ini menerapkan metodologi analisis sebagai berikut:

1. **True Negative Rate (TNR)**: Mengukur spesifisitas algoritma melalui rasio $TN/(TN + FP)$, mengindikasikan efektivitas dalam eliminasi penguji non-eligible yang berpotensi meningkatkan variabilitas.
2. **Analisis Distribusi Variabilitas**: Membandingkan koefisien variasi respons dalam populasi sebelum dan sesudah klasifikasi oleh algoritma M-X.

Metrik-metrik ini dianalisis melalui satu siklus simulasi untuk mengevaluasi akurasi algoritma dalam mengklasifikasikan penguji dalam konteks pengujian UAT berbasis crowdsourcing.

3.4 Tahap Analisis

Untuk menganalisis data secara deskriptif dan bermakna, akan dilakukan visualisasi data, seperti *heatmap*, *bar chart*, *pie chart*, dan lain-lain untuk mempermudah pemahaman dan meminimalisir kesalahpahaman [25]. Diharapkan visualisasi data juga memberikan *insights* yang akan menjadi dasar penelitian yang akan datang.

BAB 4. HASIL PERCOBAAN DAN ANALISIS

4.1. Analisis Empiris Data

Berdasarkan hasil pencocokan antara nama-nama relawan dan data sistem, diketahui bahwa seluruh 24 relawan berhasil teridentifikasi dengan benar. Data actual menunjukkan bahwa sebanyak 12 orang relawan tergolong *eligible*, dan 12 lainnya *not eligible*. Hal ini menciptakan distribusi yang seimbang (50:50).

Sistem memprediksi eligibility relawan dengan hasil sebagai berikut:

- **True Positive (TP):** 9 relawan benar diprediksi *eligible*
- **True Negative (TN):** 10 relawan benar diprediksi *not eligible*
- **False Positive (FP):** 2 relawan salah diprediksi *eligible*
- **False Negative (FN):** 3 relawan salah diprediksi *not eligible*

Secara detail akan digambarkan dengan tabel di bawah ini.

Tabel 4.1 Tabel Hasil Simulasi

Full Name	Actual	Predicted	Classification Result
Tio Haidar Hanif	Eligible	Eligible	True Positive
Dede Rahmat Fitriansyah	Not Eligible	Not Eligible	True Negative
Muhammad Ramdhan Fitra Hidayat	Not Eligible	Not Eligible	True Negative
rizqi firdaus	Eligible	Eligible	True Positive
Matthew Theodore	Eligible	Eligible	True Positive
Haikal P	Not Eligible	Not Eligible	True Negative
jaatsiyah astrianto	Eligible	Not Eligible	False Negative
Vidya Ratmayanti	Eligible	Eligible	True Positive
Sultan Alexander	Not Eligible	Not Eligible	True Negative
Regina Maharani	Eligible	Eligible	True Positive
Alif T	Eligible	Not Eligible	False Negative
I Made Rai Michael Timonuli	Not Eligible	Not Eligible	True Negative
Anthony Febrian	Eligible	Eligible	True Positive

mohamad william	Not Eligible	Not Eligible	True Negative
Aji Pandoyo	Not Eligible	Not Eligible	True Negative
Putri Auliya Rahmah	Eligible	Not Eligible	False Negative
Agung Hadi Winoto	Not Eligible	Not Eligible	True Negative
Adam Syam Syamsudin	Not Eligible	Not Eligible	True Negative
Intan Silvia	Eligible	Eligible	True Positive
Valent Reza	Not Eligible	Eligible	False Positive
Andika Rama	Not Eligible	Not Eligible	True Negative
Yosef Baptista De Morin Dasman	Eligible	Eligible	True Positive
eveline natalie	Eligible	Eligible	True Positive
Yekonyah Satria Sadhu	Not Eligible	Eligible	False Positive

Sedemikian sehingga, dapat dihitung metrik yang terkait. Metrik akan secara detail dihitung sebagai berikut.

4.1.1. Accuracy

$$\frac{(TP + TN)}{(TP + TN + FP + FN)} = (9 + 10) / (9 + 10 + 2 + 3) = 0.79$$

4.1.2. Precision

$$TP / (TP + FP) = 9 / (9 + 2) = 0.82$$

4.1.3. Recall

$$TP / (TP + FN) = 9 / (9 + 3) = 0.75$$

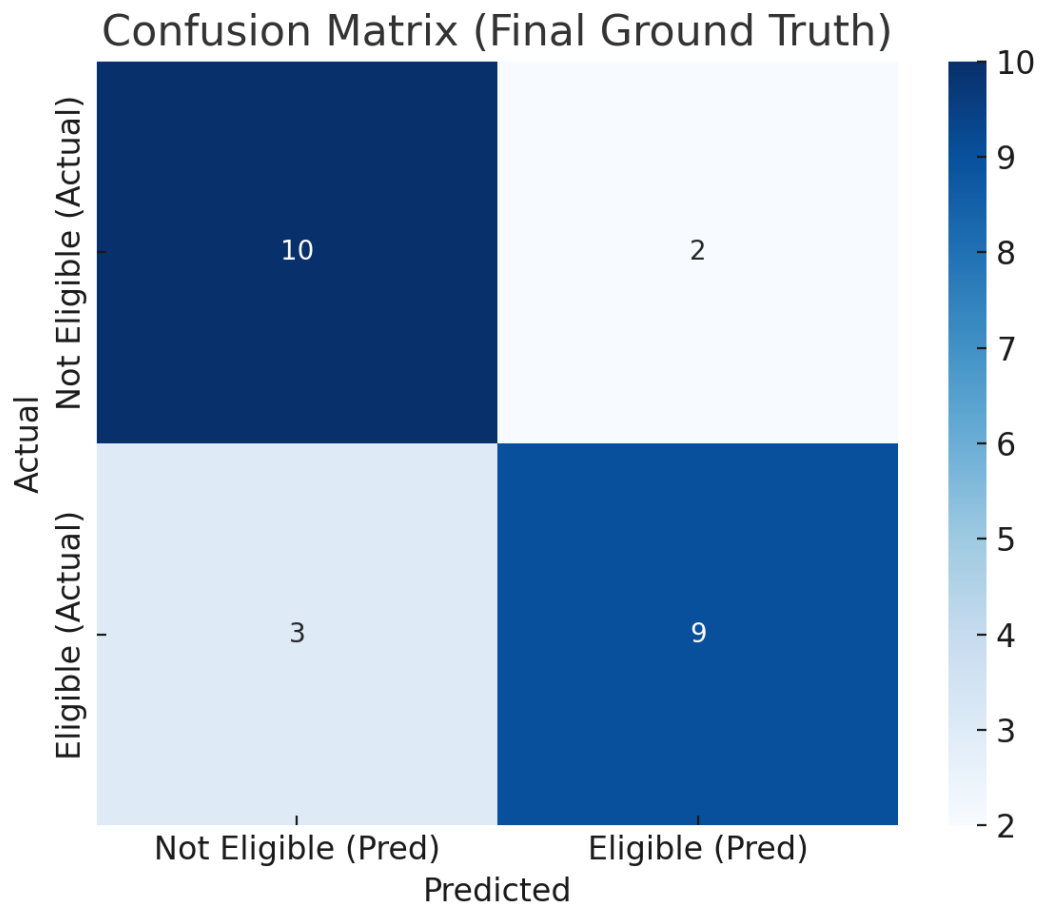
4.1.4. True Negative Rate (TNR)

$$TN / (TN + FP) = 10 / (10 + 2) = 0.83$$

4.1.5. F1 Score

$$\begin{aligned} & 2 \times (Precision \times Recall) / (Precision + Recall) \\ & = 2 \times (0.82 \times 0.75) / (0.82 + 0.75) = 0.78 \end{aligned}$$

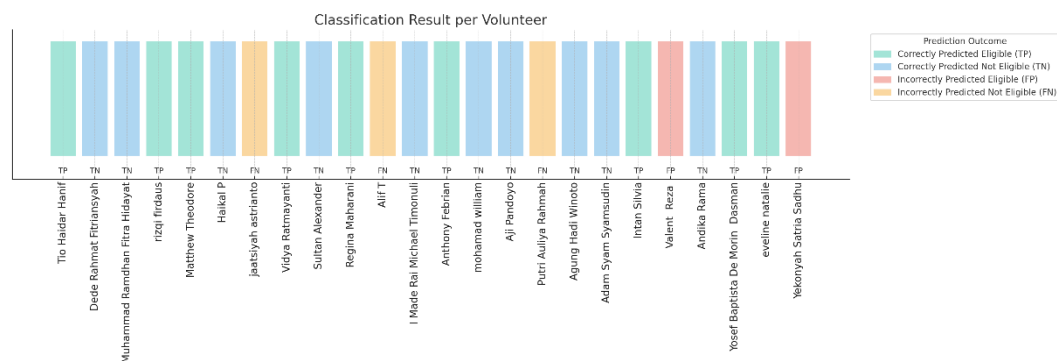
4.2. Visualisasi Data



Gambar 4.4 Heatmap Confusion Matrix

Heatmap Confusion Matrix memvisualisasikan kinerja klasifikasi algoritma M-X dalam membedakan antara penguji yang *eligible* dan yang tidak. Sel yang lebih gelap di sepanjang diagonal (Positif Benar dan Negatif Benar) menunjukkan klasifikasi yang benar, sedangkan sel yang lebih terang (Positif Palsu dan Negatif Palsu) menunjukkan kesalahan klasifikasi. Dengan 9 positif benar, 10 negatif benar, 2 positif palsu, dan 3 negatif palsu, matriks menunjukkan kinerja algoritma yang relatif kuat, dengan sebagian besar prediksi selaras dengan status *actual eligibility*. Distribusi sel yang lebih gelap yang seimbang dalam kategori yang memenuhi syarat dan yang tidak memenuhi syarat menunjukkan bahwa algoritma memiliki kinerja yang cukup baik baik untuk kedua kelas, meskipun dengan spesifisitas yang

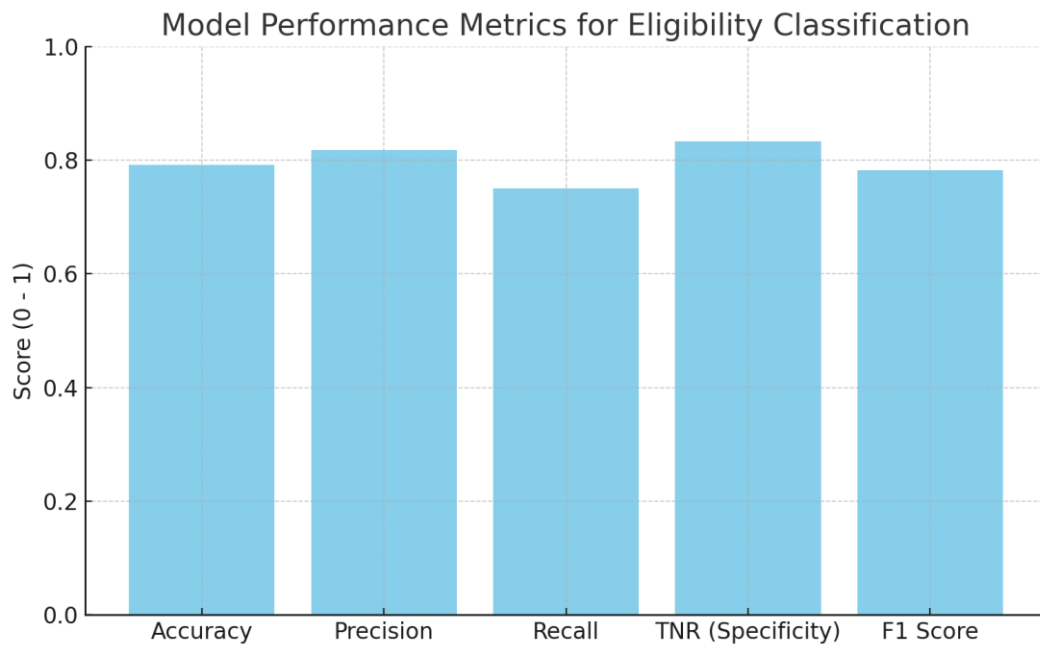
sedikit lebih baik (mengidentifikasi penguji yang tidak *eligible*) daripada sensitivitas (mengidentifikasi penguji yang memenuhi syarat).



Gambar 4.2 Classification Result

Diagram batang ini mengilustrasikan distribusi hasil klasifikasi yang dilakukan terhadap 24 relawan dalam empat kategori *outcome* berdasarkan confusion matrix. Visualisasi menunjukkan bahwa dari keseluruhan relawan, terdapat 9 True Positives (relawan *eligible* yang benar diklasifikasikan *eligible*), 10 True Negatives (relawan non-*eligible* yang benar diklasifikasikan non-*eligible*), 2 False Positives (relawan non-*eligible* yang salah diklasifikasikan *eligible*), dan 3 False Negatives (relawan *eligible* yang salah diklasifikasikan non-*eligible*).

Diagram ini dengan jelas memperlihatkan bahwa mayoritas relawan (19 dari 24 atau 79%) berhasil diklasifikasikan dengan tepat oleh algoritma M-X, yang terdiri dari 9 relawan *eligible* dan 10 relawan non-*eligible*. Kesalahan klasifikasi yang terjadi relatif kecil, dengan total 5 dari 24 relawan (21%) yang mengalami kesalahan prediksi. Dari kesalahan tersebut, algoritma lebih cenderung melakukan kesalahan *False Negative* (3 relawan) dibandingkan dengan *False Positive* (2 relawan), menunjukkan sifat algoritma yang sedikit lebih konservatif dalam mengklasifikasikan relawan sebagai *eligible*.



Gambar 4.3 Performance Metrics

Diagram batang ini menampilkan lima metrik kinerja utama yang dihitung dari confusion matrix dalam bentuk visualisasi yang komprehensif dan mudah dibandingkan. Metrik-metrik tersebut adalah *Accuracy* (0.79), *Precision* (0.82), *Recall* (0.75), *True Negative Rate* (0.83), dan *F1 Score* (0.78). Visualisasi ini memungkinkan perbandingan langsung antara kelima metrik, menunjukkan distribusi nilai yang relatif tinggi dan merata (semua di atas 0.75), yang mengindikasikan performa yang seimbang dari algoritma M-X.

Diagram menunjukkan bahwa True Negative Rate memiliki nilai tertinggi (0.83), menandakan kemampuan algoritma dalam mengidentifikasi penguji non-eligible dengan sangat baik. *Precision* (0.82) juga menunjukkan nilai yang tinggi, yang berarti keandalan algoritma dalam mengklasifikasikan penguji sebagai eligible. *Recall* (0.75) meskipun masih menunjukkan performa yang baik, memiliki nilai sedikit lebih rendah dibandingkan metrik lainnya, mengindikasikan adanya ruang untuk peningkatan dalam hal kemampuan algoritma mendeteksi seluruh penguji eligible. *F1 Score* (0.78) menunjukkan keseimbangan yang baik antara *precision* dan *recall*.

4.3. Menjawab Research Question

Berdasarkan evaluasi terhadap algoritma M-X yang telah dilakukan, berikut adalah jawaban yang akan menjawab *Research Question* (Perumusan Masalah).

Q1: Bagaimana tingkat akurasi algoritma M-X dalam mengklasifikasikan penguji dalam lingkungan UAT berbasis crowdsourcing?

Berdasarkan hasil empiris penelitian ini, algoritma M-X menunjukkan akurasi yang menjanjikan sebesar 79% dalam mengklasifikasikan penguji pada konteks UAT berbasis crowdsourcing. Angka ini mengindikasikan bahwa algoritma mampu dengan tepat mengklasifikasikan hampir empat dari lima penguji sesuai dengan status eligibility aktual mereka, yang merupakan tingkat keandalan yang cukup tinggi untuk konteks seleksi penguji.

Precision sebesar 82% menunjukkan bahwa dari seluruh penguji yang diklasifikasikan sebagai eligible oleh algoritma, 82% di antaranya memang benar-benar eligible berdasarkan prior-experience based validation. Nilai ini sangat penting karena menunjukkan reliabilitas algoritma dalam mengidentifikasi penguji berkualitas tinggi, sehingga meminimalisir risiko melibatkan penguji yang tidak memenuhi kualifikasi dalam proses UAT.

Recall sebesar 75% mengindikasikan bahwa algoritma berhasil mengidentifikasi tiga perempat dari seluruh penguji eligible yang ada dalam populasi sampel. Meskipun nilai ini sedikit lebih rendah daripada precision, hal ini masih menunjukkan sensitivitas yang baik dalam mendeteksi penguji berkualitas. Terdapat tiga penguji eligible (false negatives) yang tidak terdeteksi oleh algoritma, yang dapat menjadi fokus pengembangan lebih lanjut untuk meningkatkan kapabilitas algoritma.

F1 Score sebesar 0.78 memvalidasi keseimbangan yang baik antara precision dan recall, sehingga menunjukkan bahwa algoritma M-X tidak terlalu condong pada salah satu aspek klasifikasi. Nilai ini juga

mengkonfirmasi reliabilitas algoritma secara keseluruhan dalam konteks UAT berbasis crowdsourcing.

True Negative Rate yang tinggi sebesar 83% mendemonstrasikan kemampuan algoritma dalam mengidentifikasi penguji non-eligible dengan akurat, yang sama pentingnya dengan mengidentifikasi penguji eligible. Kemampuan ini sangat krusial untuk menyaring penguji yang mungkin memberikan kontribusi suboptimal dalam proses pengujian.

Secara keseluruhan, metrik-metrik yang diperoleh dari simulasi menunjukkan bahwa algoritma M-X memiliki tingkat akurasi yang baik dalam mengklasifikasikan penguji dalam lingkungan UAT berbasis crowdsourcing, dengan performa yang seimbang dalam mengidentifikasi kedua kelas penguji.

Q2: Sejauh mana algoritma M-X dapat mengurangi variabilitas karakteristik penguji dan memastikan konsistensi kualitas pengujian?

Berdasarkan hasil simulasi dan analisis empiris, algoritma M-X menunjukkan kapabilitas yang substansial dalam mengurangi variabilitas karakteristik penguji pada UAT berbasis crowdsourcing. Dengan True Negative Rate (TNR) sebesar 83%, algoritma secara efektif mampu menyaring mayoritas penguji non-eligible yang berpotensi meningkatkan variabilitas dalam hasil pengujian. Kemampuan algoritma untuk mengidentifikasi dan mengeliminasi penguji yang menunjukkan pola jawaban inkonsisten sangat penting dalam menjaga homogenitas karakteristik kelompok penguji yang terlibat.

Aspek penting dari algoritma M-X adalah independensinya terhadap kebenaran yang telah ditentukan sebelumnya (predefined ground truth). Algoritma hanya mengandalkan pola konsistensi antar jawaban penguji, yang membuatnya sangat fleksibel untuk diterapkan dalam berbagai konteks pengujian yang berbeda. Pendekatan ini memungkinkan algoritma untuk mengidentifikasi penguji berkualitas berdasarkan konsistensi respons mereka, terlepas dari tipe pengujian atau skenario UAT yang diterapkan.

Dalam simulasi yang dilakukan, algoritma M-X berhasil memfilter sebagian besar penguji *not eligible* (10 dari 12), sekaligus mempertahankan mayoritas penguji *eligible* (9 dari 12). Dengan mengaplikasikan pendekatan rule-based validation untuk menentukan threshold kelayakan, algoritma dapat disesuaikan dengan kebutuhan spesifik proyek pengujian, meningkatkan fleksibilitasnya dalam beradaptasi dengan variasi karakteristik pengujian.

Meskipun terdapat beberapa false positives (2 penguji non-*eligible* yang terklasifikasi sebagai *eligible*), jumlah ini relatif kecil dan tidak signifikan mempengaruhi konsistensi kualitas keseluruhan. Kemampuan algoritma untuk menjaga precision yang tinggi (82%) menunjukkan bahwa pool penguji yang terpilih akan didominasi oleh individu dengan pola respons yang konsisten, sehingga meminimalisir variabilitas dalam hasil pengujian.

Dengan demikian, algoritma M-X telah menunjukkan kapabilitas yang menjanjikan dalam mengurangi variabilitas karakteristik penguji dan memastikan konsistensi kualitas pengujian dalam konteks UAT berbasis crowdsourcing. Penerapan algoritma ini berpotensi meningkatkan efisiensi dan reliabilitas proses pengujian dengan menghilangkan penguji yang menunjukkan pola inkonsistensi, sehingga menghasilkan hasil pengujian yang lebih representatif dan dapat diandalkan.

BAB 5. KESIMPULAN

Penelitian ini berhasil membuktikan efektivitas algoritma M-X sebagai mekanisme kontrol kualitas penguji dalam konteks User Acceptance Testing berbasis crowdsourcing. Melalui simulasi yang melibatkan 24 relawan, algoritma M-X menunjukkan kemampuan klasifikasi yang sangat baik dengan tingkat akurasi 79%, precision 82%, recall 75%, dan F1 score 78%. Kinerja ini mengkonfirmasi bahwa algoritma M-X dapat secara efektif membedakan antara penguji yang eligible dan non-eligible berdasarkan konsistensi pola jawaban mereka, tanpa bergantung pada jawaban yang telah ditentukan sebelumnya.

Kelebihan signifikan dari algoritma M-X terletak pada kemampuannya untuk mengurangi variabilitas karakteristik penguji, yang ditunjukkan oleh True Negative Rate sebesar 83%. Kemampuan ini sangat penting dalam memastikan homogenitas dan konsistensi pool penguji yang terlibat dalam proses UAT. Fleksibilitas algoritma M-X juga menjadi keunggulan utama, memungkinkan penerapannya dalam berbagai skenario pengujian dan penyesuaian threshold melalui rule-based validation sesuai dengan kebutuhan spesifik proyek.

Secara keseluruhan, penelitian ini memberikan landasan empiris yang kuat bahwa algoritma M-X merupakan solusi yang menjanjikan untuk meningkatkan kualitas pengujian dalam UAT berbasis crowdsourcing. Implementasi algoritma ini dapat meningkatkan efisiensi dan efektivitas proses pengujian melalui seleksi penguji berkualitas tinggi, sehingga menghasilkan hasil UAT yang lebih representatif dan dapat diandalkan.

Keterbatasan dan Penelitian Selanjutnya

Penelitian ini memiliki beberapa keterbatasan yang perlu diperhatikan. Simulasi terbatas pada 24 relawan yang mayoritas berasal dari Telkom University dengan rentang usia 20-24 tahun, sehingga kurang merepresentasikan keberagaman demografi penguji dalam situasi nyata. Pengujian dilakukan dalam lingkungan simulasi terkontrol, bukan dalam lingkungan produksi yang memiliki kompleksitas

lebih tinggi. Selain itu, simulasi hanya mencakup skenario pilihan ganda dan tidak mengontrol secara ketat faktor eksternal seperti motivasi dan kelelahan penguji. Evaluasi juga terbatas pada konsistensi antar jawaban, bukan pada efektivitas hasil pengujian terhadap fungsionalitas perangkat lunak secara menyeluruh.

Untuk penelitian selanjutnya, disarankan untuk memperluas populasi sampel dengan melibatkan relawan dari berbagai latar belakang demografis, pendidikan, dan pengalaman untuk menguji keandalan algoritma dalam populasi yang lebih heterogen. Implementasi algoritma M-X dalam proyek UAT nyata juga penting untuk mengevaluasi performa dalam kondisi produksi aktual. Pengembangan varian algoritma yang dapat menangani berbagai jenis tugas pengujian dan studi mendalam tentang penentuan threshold optimal untuk berbagai jenis proyek juga menjadi arah penelitian yang potensial. Selain itu, kombinasi algoritma M-X dengan metode kontrol kualitas lain seperti penugasan berbasis reputasi atau evaluasi berbasis kinerja dapat meningkatkan akurasi klasifikasi. Analisis longitudinal juga direkomendasikan untuk mengevaluasi konsistensi kinerja algoritma dan evolusi karakteristik penguji dari waktu ke waktu.

DAFTAR PUSTAKA

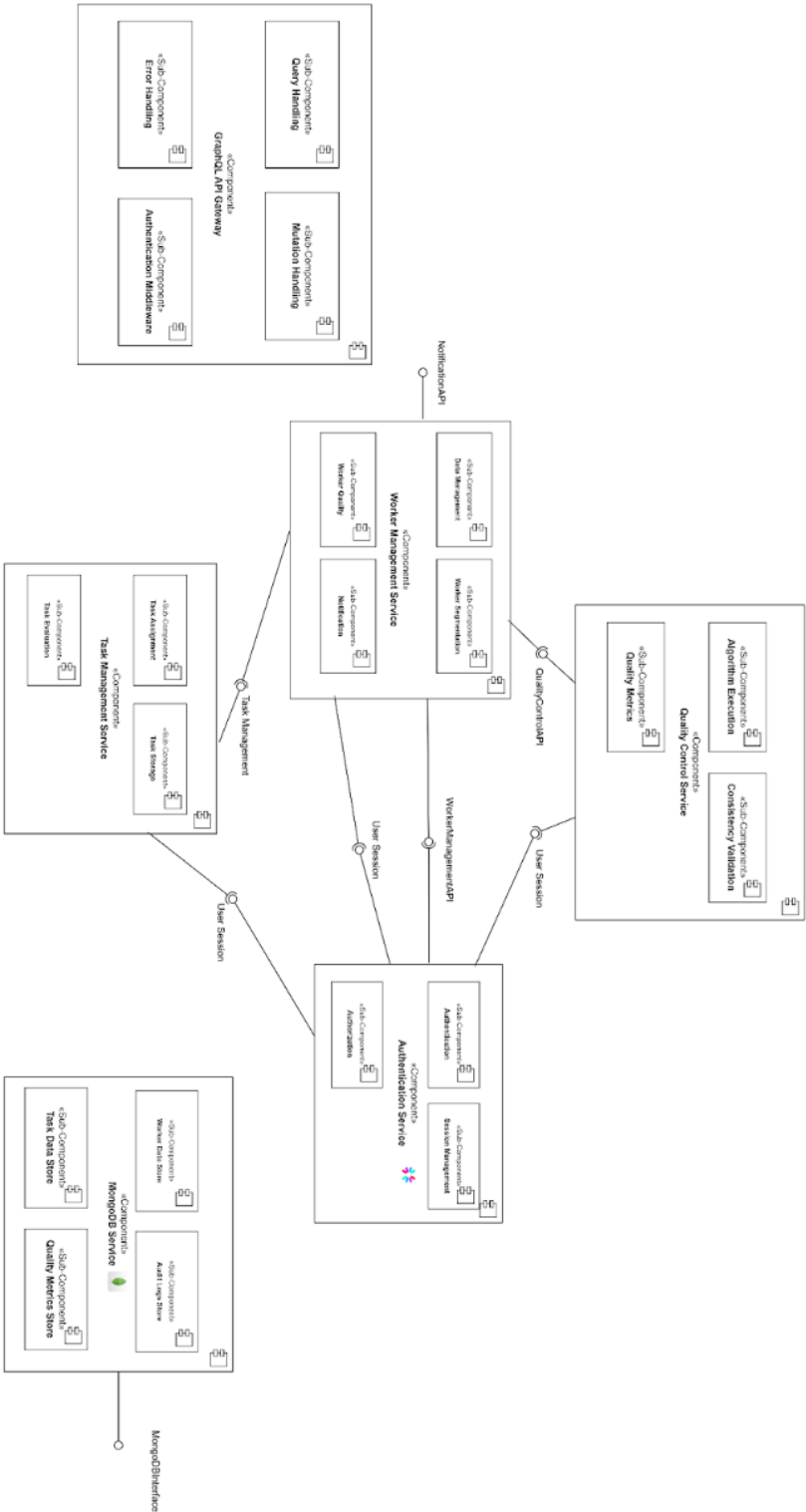
- [1] H. K. N. & W. P. W. L. Leung, "A study of user acceptance tests," *Software Quality Journal*, pp. 137-149, 1997.
- [2] I. Sommerville, *Software Engineering 9th*, Boston: Addison-Wesley, 2011.
- [3] D. H. A. H. Y. & F. G. Febrianti, "Rancang Bangun Sistem Informasi Quality Control Berbasis User Acceptance Testing (UAT) Untuk Project Digital Pada PT ARG Solusi Teknologi," *Jurnal Vocational Teknik Elektronika dan Informatika*, vol. 12, no. 2, pp. 226-234, 2024.
- [4] I. & D. O. Otaduy, "User Acceptance Testing for Agile-developed Web-based applications: empowering customers through wikis and mind maps," *The Journal of Systems and Software*, 2017.
- [5] P. & T. S. Pandit, "AgileUAT: A Framework for User Acceptance Testing based on User Stories and Acceptance Criteria," *International Journal of Computer Applications*, vol. 120, no. 10, pp. 16-21, 2015.
- [6] M. & A. S. Alsayyari, "Supporting Coordination in Crowdsourced Software Testing Services," in *2018 IEEE Symposium on Service-Oriented System Engineering (SOSE)*, Bamberg, Germany, 2018.
- [7] G. K. J. & M.-F. N. Kazai, "Worker Types and Personality Traits in Crowdsourcing," in *Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM)*, Glasgow, Scotland, UK, 2011.
- [8] J. M. & K. A. Rzeszotarski, "Instrumenting the Crowd: Using Implicit Behavioral Measures to Predict Task Performance," in *Proceedings of the 24th*

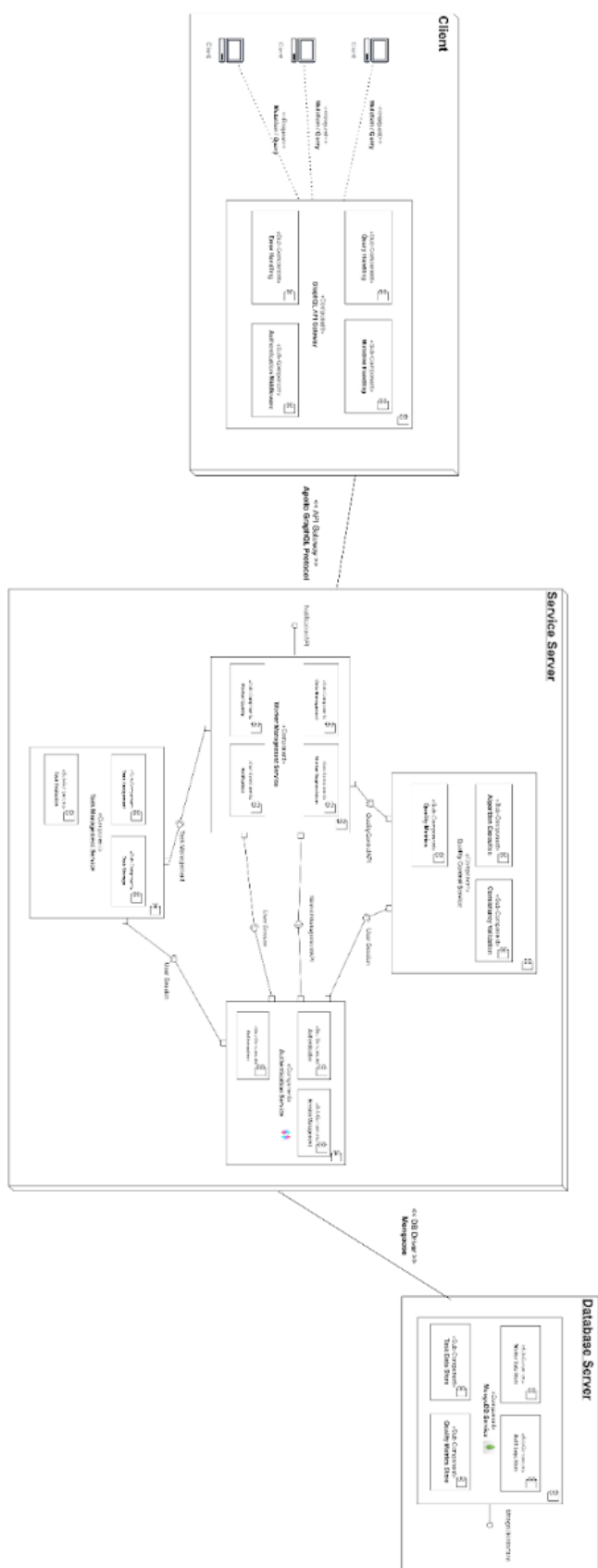
Annual ACM Symposium on User Interface Software and Technology (UIST), Santa Barbara, California, USA, 2011.

- [9] D. Dang, Y. Liu, X. Zhang and S. Huang, "A Crowdsourcing Worker Quality Evaluation Algorithm on MapReduce for Big Data Applications," in *IEEE Transactions on Parallel and Distributed Systems*, 2015.
- [10] G. Blinowski, A. Ojdowska and A. Przybyłek, "Monolithic vs. Microservice Architecture: A Performance and Scalability Evaluation," in *IEEE Access*, 2022.
- [11] P. Margański and B. Pańczyk, "REST and GraphQL comparative analysis," *Journal of Computer Sciences Institute*, vol. 19, pp. 89-94, 2021.
- [12] N. Chaudhary and N. Mittal, "Leveraging Mongo DB for Efficient Data storage in MERN," in *International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions)*, Noida, India, 2024.
- [13] S. W. J. C. T. M. Q. C. M. X. Q. W. Junjie Wang, "Characterizing Crowds to Better Optimize Worker Recommendation in Crowdsourced Testing," *IEEE Transactions on Software Engineering*, 2019.
- [14] Y.-Y. L. J.-H. K. U.-M. K. Jeon-Pyo Hong, "Crowd Worker Selection with Wide Learning and Narrow Evaluation," in *15th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, Seoul, Korea, 2021.
- [15] S. H. C. Z. E. L. a. N. C. Yongming Yao, "A Study on Testers' Learning Curve in Crowdsourced Software Testing," *IEEE Access*, vol. 9, pp. 77127-77137, 2021.

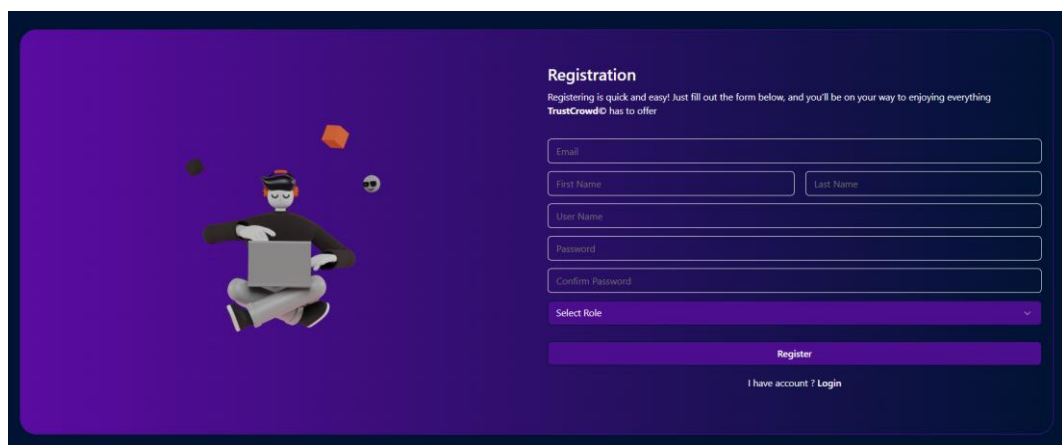
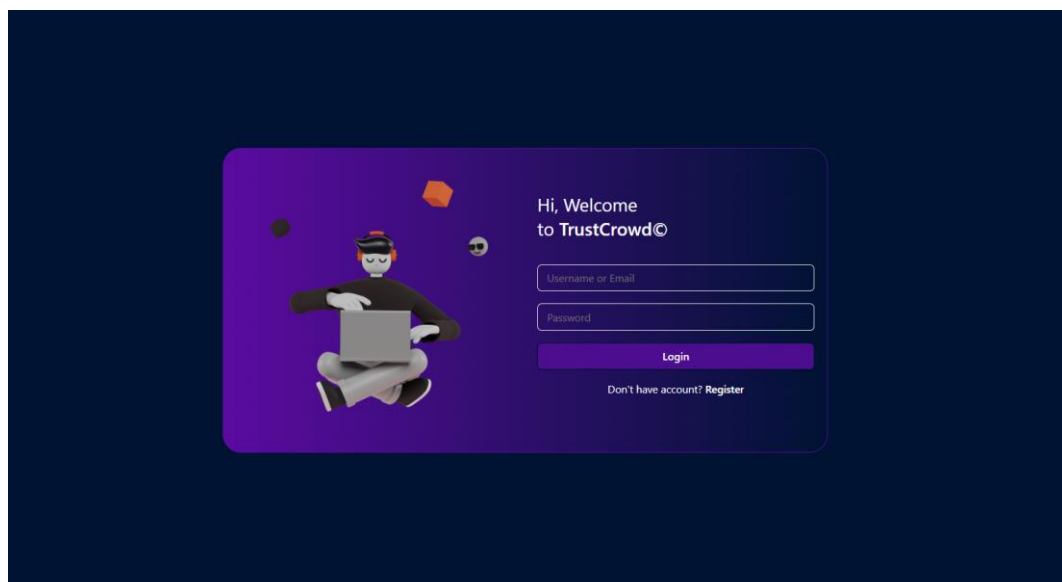
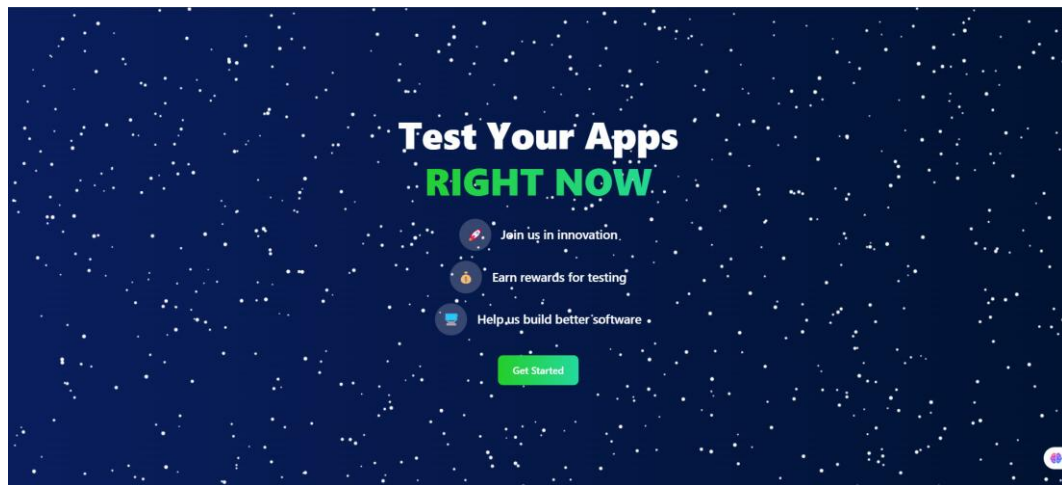
- [16] J. Howe, *Crowdsourcing: Why the Power of the Crowd is Driving the Future of Business*, New York: Crown Business, Crown Business, New York, NY, 2008.
- [17] S. Alyahya, "Crowdsourced Software Testing: A Systematic Literature Review," *Information and Software Technology*, 2020.
- [18] S. W. J. W. Y. H. Q. W. M. L. Qiang Cui, "Multi-Objective Crowd Worker Selection in Crowdsourced Testing," in *29th International Conference on Software Engineering and Knowledge Engineering (SEKE)*, 2017.
- [19] S. Oludare, "Confusion-Matrix Based Performance Evaluation Metrics," in *Proceedings of the 5th International Conference on Applied Information Technology*, Lagos, 2021.
- [20] Q. C. Q. W. S. W. Junjie Wang, "Towards Effectively Test Report Classification to Assist Crowdsourced Testing," in *Proceedings of the ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, Ciudad Real, Spain, 2016.
- [21] T. R. Silva, "Towards a Domain-Specific Language for Behaviour-Driven Development," in *2023 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, Odense, Denmark, 2023.
- [22] R. O. S. W. A. Y. Rosa Reska Riskiana, "FlowForge: A Prototype for Generating User Stories and Gherkin Test Cases from BPMN with DMN Integration and Pattern Matching," in *International Journal on ICT*, Bandung, Indonesia, 2024.

LAMPIRAN






User Interface Trustcrowd© - Platform Simulasi



Welcome to Your Dashboard
Hello, Gaurav Satish Chavhaning Ap

Your Eligibility Status
 **Not Eligible**
 Complete more tasks to become eligible
[Go to taskcard complete more to improve your eligibility!](#)

UAT Reporting Dashboard Locked
 Access comprehensive analytics and reporting tools for the user acceptance testing process
[Become eligible to access](#)

UAT Testing Platform Locked
 Access the main testing platform to conduct user acceptance tests on new features
[Become eligible to access](#)

Your Progress


Task Completed 17 <small>100% of all tasks</small>	Available Tests 17 View available tasks	Active Tests 19 View your results
--	---	---

What to Do Next

Complete Tasks
 Work on available tasks to improve your eligibility status

Check Results
 Review your past submissions and accuracy metrics

Welcome to Your Dashboard
Hello, Gaurav Satish Chavhaning Ap

Your Eligibility Status
 **Eligible**
 You can access all UAT resources

UAT Reporting Dashboard Locked
 Access comprehensive analytics and reporting tools for the user acceptance testing process
[Dashboard & Analytics](#)

UAT Testing Platform Locked
 Access the main testing platform to conduct user acceptance tests on new features
[Test & Document](#)

Your Progress

Task Completed 17 <small>100% of all tasks</small>	Available Tests 17 View available tasks	Active Tests 19 View your results
--	---	---

What to Do Next

Complete Tasks
 Work on available tasks to improve your eligibility status

Check Results
 Review your past submissions and accuracy metrics

[← Back to Dashboard](#)

Available Tasks

Form Validation Failure Check Gherkin Type of Question Scenarios Form validation failure during job posting.	Missing Email Field Registration form without email input Scenarios User registration form submitted without an email address.	Invalid Password Format Check password format enforcement Scenarios User attempts to register with a weak password.
Incomplete Registration Form Missing mandatory fields in registration Scenarios User submits a registration form with missing mandatory fields.	Error in Payment Processing Failure during payment transaction Scenarios Payment processing fails during checkout.	User Login Failure Login attempt with incorrect credentials Scenarios User attempts to log in with invalid credentials.

[Previous](#)
[Page 1 of 3](#)
[Next](#)

Scenario: Form validation failure during job posting.

Given: The user has entered the job title and budget fields.

When: The user clicks the "Submit" button without selecting a job category.

Then: What do you expect the system to do in this case?

- The system immediately displays an inline error message right below the job category field, such as "Job category is required."
- The system prevents the form from being submitted by disabling the "Submit" button until a job category is selected, possibly highlighting the field with a red border.
- Upon clicking "Submit," the system shows a modal dialog alerting the user that a required field (job category) is missing.
- The system displays a tooltip or popover near the job category field that explains the requirement (e.g., "Please select a job category before submitting") while also marking the field with an error indicator

Submit Answer

crowd-control-interface.vercel.app says

Answer submitted successfully!

OK

Scenario: Form validation failure during job posting.

Given: The user has entered the job title and budget fields.

When: The user clicks the "Submit" button without selecting a job category.

Then: What do you expect the system to do in this case?

- The system immediately displays an inline error message right below the job category field, such as "Job category is required."
- The system prevents the form from being submitted by disabling the "Submit" button until a job category is selected, possibly highlighting the field with a red border.
- Upon clicking "Submit," the system shows a modal dialog alerting the user that a required field (job category) is missing.
- The system displays a tooltip or popover near the job category field that explains the requirement (e.g., "Please select a job category before submitting") while also marking the field with an error indicator

Submit Answer

[← Back to Dashboard](#)

Available Tasks

Form Validation Failure
Check Gherkin Type of Question
Scenario: Form validation failure during job posting.
Answer: The system prevents the form from being submitted by disabling the "Submit" button until a job category is selected, possibly highlighting the field with a red border.

Missing Email Field
Registration form without email input
Scenario: User registration form submitted without an email address.
Answer: Show a modal alert that the email field cannot be left blank.

Invalid Password Format
Check password format enforcement
Scenario: User attempts to register with a weak password.
Answer: Highlight the password field with an error indicator and provide guidelines for a strong password.

Incomplete Registration Form
Missing mandatory fields in registration
Scenario: User submits a registration form with missing mandatory fields.

Error in Payment Processing
Failure during payment transaction
Scenario: Payment processing fails during checkout.

User Login Failure
Login attempt with incorrect credentials
Scenario: User attempts to log in with invalid credentials.

[Previous](#) [Page 1 of 3](#) [Next](#)

Test Results

Eligibility Status

Not Eligible
Based on your test performance, you're currently not eligible. Keep taking tests to improve your status!

Completed Tasks

Task
Scenario: Form validation failure during job posting.
Your Answer: The system prevents the form from being submitted by disabling the "Submit" button until a job category is selected, possibly highlighting the field with a red border.

Task
Scenario: User registration form submitted without an email address.
Your Answer: Show a modal alert that the email field cannot be left blank.

Task
Scenario: User attempts to register with a weak password.
Your Answer: Highlight the password field with an error indicator and provide guidelines for a strong password.

[Back to Dashboard](#)

Test Results

Eligibility Status

Eligible

Completed Tasks

Task
Scenario: Form validation failure during job posting.
Your Answer: The system prevents the form from being submitted by disabling the "Submit" button until a job category is selected, possibly highlighting the field with a red border.

Task
Scenario: User registration form submitted without an email address.
Your Answer: Show a modal alert that the email field cannot be left blank.

Task
Scenario: User attempts to register with a weak password.
Your Answer: Highlight the password field with an error indicator and provide guidelines for a strong password.

[Back to Dashboard](#)

Test Results

Eligibility Status

Pending Evaluation

Your eligibility is being evaluated based on your test performance.
Continue completing tasks to receive a full evaluation.

Completed Tasks

Task
Scenario: Form validation failure during job posting.
Your Answer: The system prevents the form from being submitted by disabling the "Submit" button until a job category is selected, possibly highlighting the field with a red border.

Task
Scenario: User registration form submitted without an email address.
Your Answer: Show a modal alert that the email field cannot be left blank.

Task
Scenario: User attempts to register with a weak password.
Your Answer: Highlight the password field with an error indicator and provide guidelines for a strong password.

[Back to Dashboard](#)

Task Management

[Create Task](#)

Title	Description	Status	Action
Form Validation Failure	Check Gherkin Type of Question	Valid	View Delete
Missing Email Field	Registration form without email input	Valid	View Delete
Invalid Password Format	Check password format enforcement	Valid	View Delete
Incomplete Registration Form	Missing mandatory fields in registration	Valid	View Delete
Error in Payment Processing	Failure during payment transaction	Valid	View Delete

A list of tasks (20 total)

[Previous](#) Page 1 of 4 [Next](#)

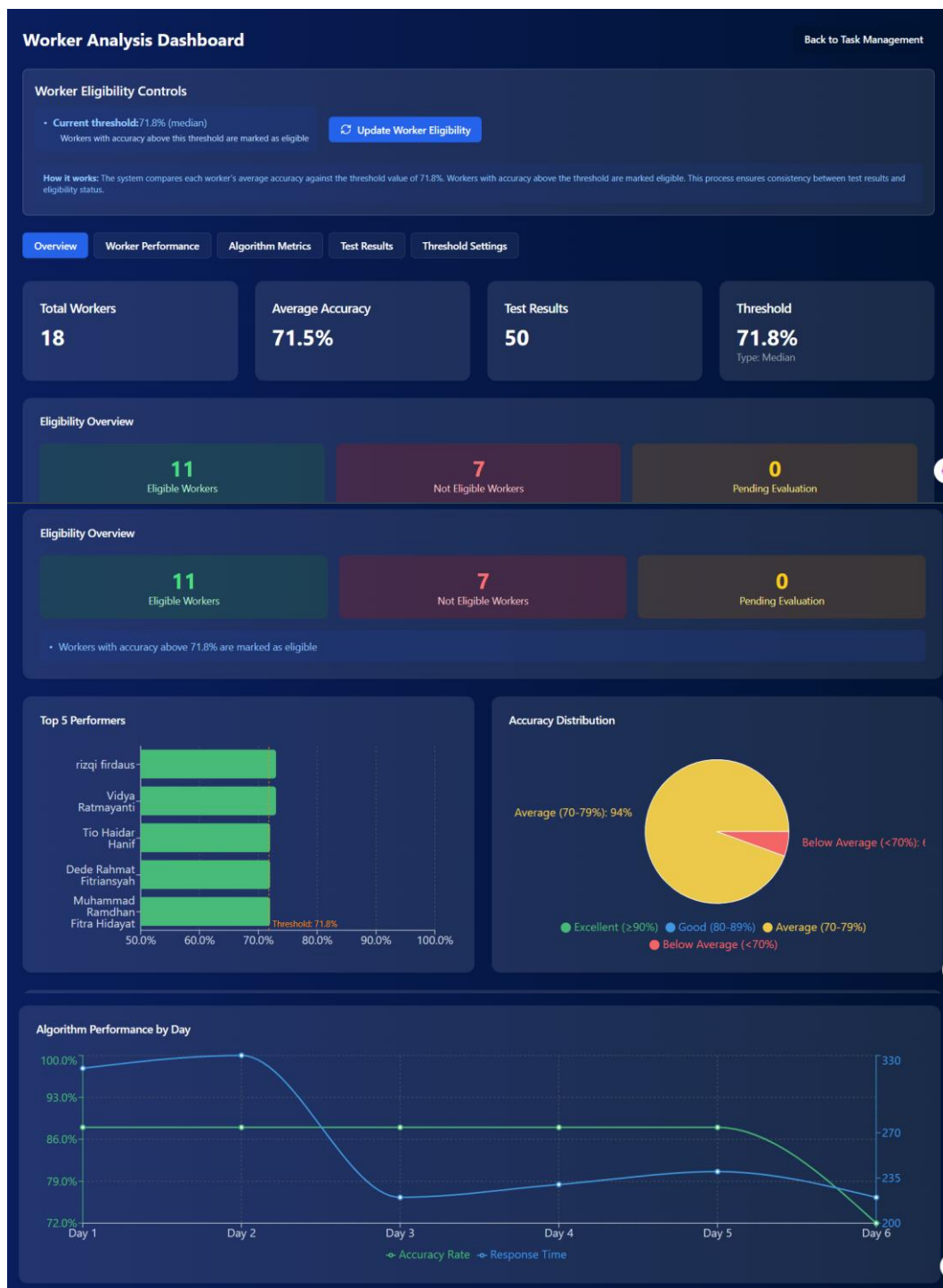
User Management

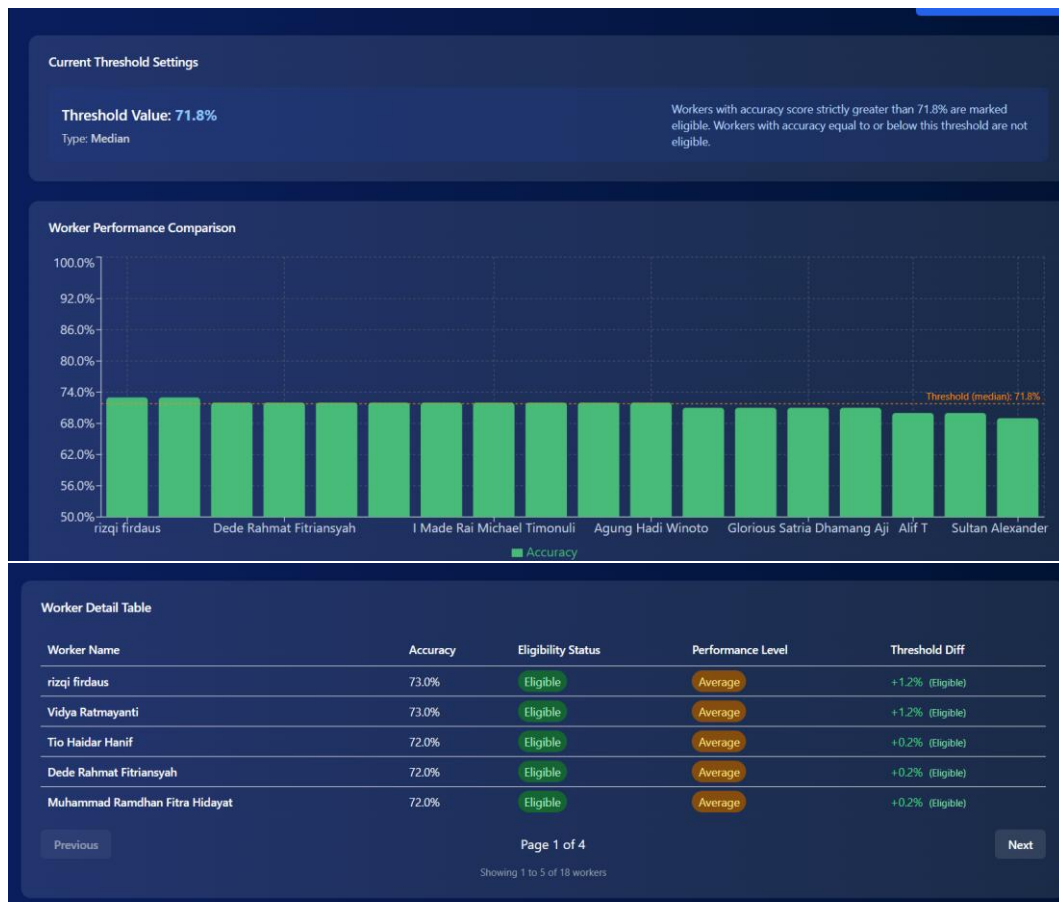
[Create User](#)

Name	Email	Role	Actions
Admin TrustCrowd	admin@admin.com	admin	View Delete
Validator La Incondicional	validator@validator.com	question_validator	View Delete
Tio Haidar Hanif	atohaidar@student.telkomuniversity.ac.id	worker	View Delete
Dede Rahmat Fitrianyah	ryunelhaimu@gmail.com	worker	View Delete
Muhammad Ramdhan Fitra Hidayat	fitrahidayat@gmail.com	worker	View Delete

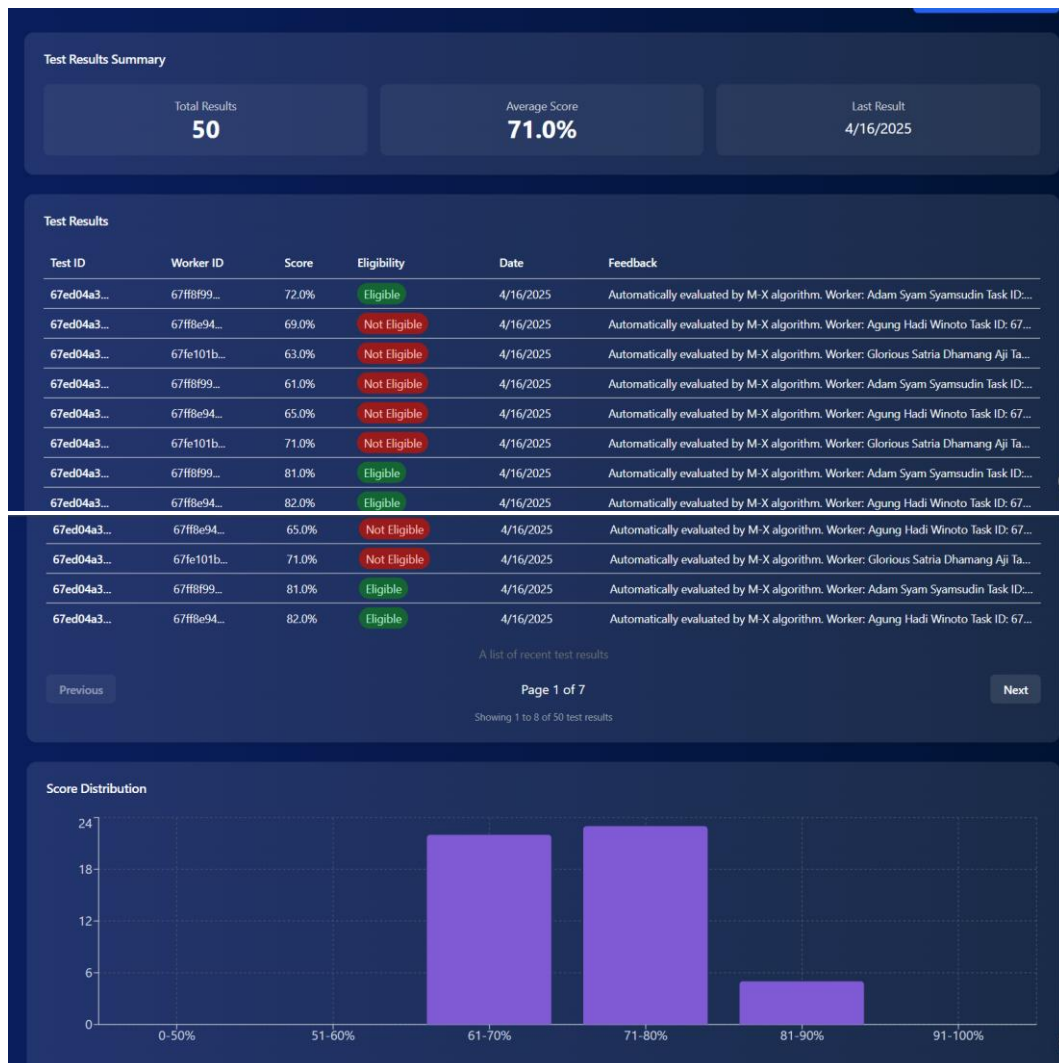
All registered users in the system

[Previous](#) Page 1 of 5 [Next](#)









Worker Eligibility Threshold Configuration

Current Settings

Threshold Type: **Median**
Current Value: **71.78%**
Last updated: 4/17/2025, 11:28:08 PM

Threshold Calculation Method

Median (Middle value of all worker accuracies) ▾

This method determines how the eligibility threshold is calculated.

Median Calculation

The system will calculate the median accuracy value from all workers. Workers with accuracy above this median will be considered eligible.
The exact threshold value will be calculated server-side based on worker data.

Save Configuration

Manually Update Worker Eligibility

Note: Changing the threshold type or value will automatically update all worker eligibility statuses. This process may take a moment to complete.

Median (Middle value of all worker accuracies) ▾

Median (Middle value of all worker accuracies) ✓

Mean (Average of all worker accuracies)

Custom (Set a specific value)

Validate Task Question

Task Scenario:
A user attempts to update a resource without sufficient permissions.

Update Validation Status

Task Scenario:
A file upload operation fails during the process.

Update Validation Status

Task Scenario:
User session expires after a period of inactivity.

Update Validation Status

Prev Page 1 of 1 Next

Validate Task Question

Task Scenario:
A user attempts to update a resource without sufficient permissions.

Update Validation Status

Task Scenario:
User session expires after a period of inactivity.

Update Validation Status

Task Scenario:
A file upload operation fails during the process.

Update Validation Status

Prev Next

Update Validation

Scenario:
A user attempts to update a resource without sufficient permissions.

Description:
Insufficient permissions to update resource

Answers:
Display an error message indicating insufficient permissions.
Prevent the update operation and log the attempt.
Highlight the restricted area and notify the user.
Show a modal dialog with details on required permissions.

Validate Close

Tio Haidar Hanif (Presenting)

Scenario: User registration form submitted without an email address.
 Given: The user has filled out all fields except the email field.
 When: The user submits the registration form.
 Then: What error message should be displayed regarding the missing email?
 Display an inline error message stating "Email is required."
 Disable the submit button until the email field is completed.
 Highlight the email field with a red border and prompt the user to enter a valid email address.
 Show a modal alert that the email field cannot be left blank.

11:05 AM | gtv-jhxx-kzx

Muhammad Ramdhan Fitra Hidayat (Presenting)

Scenario: User submits a registration form with missing mandatory fields.
 Given: The user leaves some required fields blank.
 When: The user clicks the submit button.
 Then: What feedback should the system provide to indicate the missing information?
 Highlight each missing field with an error message indicating that the field is required.
 Prevent form submission and display a summary of all missing fields.
 Disable the submit button until all mandatory fields are filled out.
 Show a modal alert listing the fields that need to be completed.

11:07 AM | gtv-jhxx-kzx

Dede Rahmat Fitriansyah (Presenting)

Scenario: User submits a registration form with missing mandatory fields.
 Given: The user leaves some required fields blank.
 When: The user clicks the submit button.
 Then: What feedback should the system provide to indicate the missing information?
 Highlight each missing field with an error message indicating that the field is required.
 Prevent form submission and display a summary of all missing fields.
 Disable the submit button until all mandatory fields are filled out.
 Show a modal alert listing the fields that need to be completed.

11:08 AM | gtv-jhxx-kzx

Matthew Theodore (Presenting)

Scenario: User registration form submitted without an email address.
Given: The user has filled out all fields except the email field.
When: The user submits the registration form.
Then: What error message should be displayed regarding the missing email?

Display an inline error message stating "Email is required."

Disable the submit button until the email field is completed.

Highlight the email field with a red border and prompt the user to enter a valid email address.

Show a modal alert that the email field cannot be left blank.

Submit Answer

Matthew Theodore, rizqi firdaus, Jastiyah Maulidina, Rafidhia Huska Pasya, Vidya Ananda Ratmayanti, glorious satria

rizqi firdaus (Presenting)

Scenario: User registration form submitted without an email address.
Given: The user has filled out all fields except the email field.
When: The user submits the registration form.
Then: What error message should be displayed regarding the missing email?

Display an inline error message stating "Email is required."

Disable the submit button until the email field is completed.

Highlight the email field with a red border and prompt the user to enter a valid email address.

Show a modal alert that the email field cannot be left blank.

Submit Answer

Matthew Theodore, rizqi firdaus, Jastiyah Maulidina, Rafidhia Huska Pasya, Vidya Ananda Ratmayanti, glorious satria

Vidya Ananda Ratmayanti (Presenting)

Scenario: User registration form submitted without an email address.
Given: The user has filled out all fields except the email field.
When: The user submits the registration form.
Then: What error message should be displayed regarding the missing email?

Display an inline error message stating "Email is required."

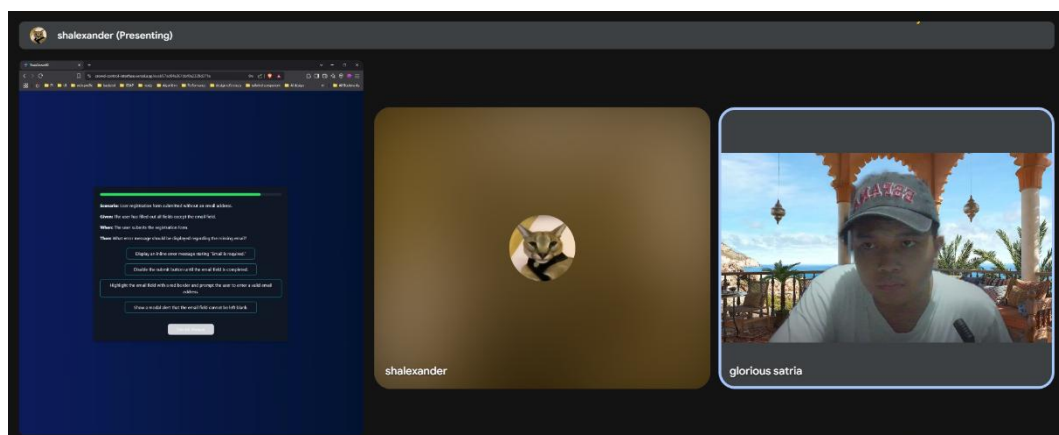
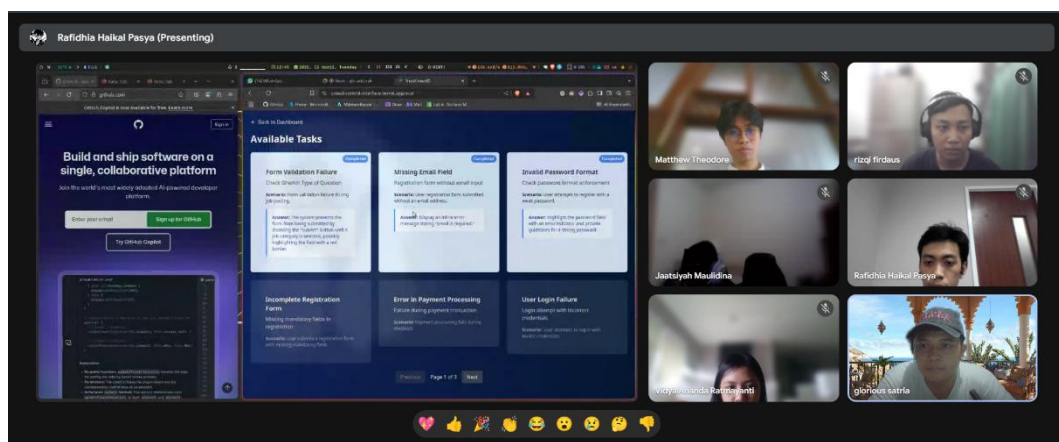
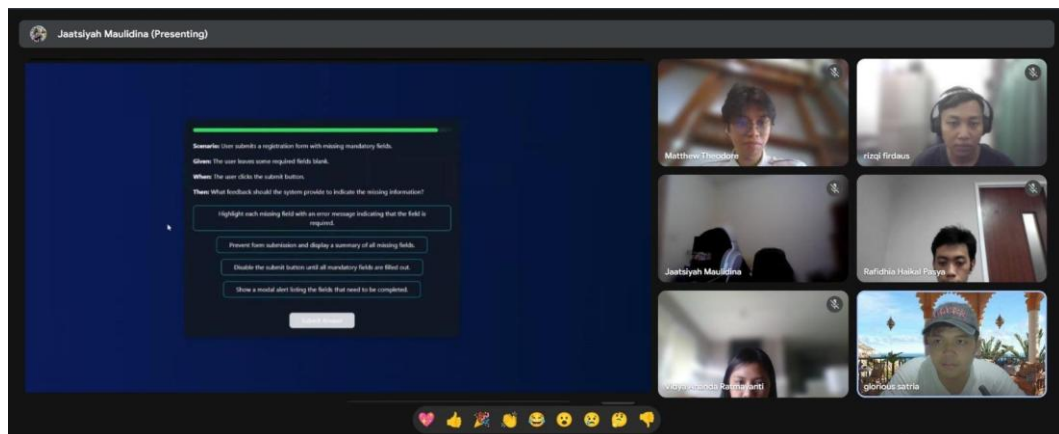
Disable the submit button until the email field is completed.

Highlight the email field with a red border and prompt the user to enter a valid email address.

Show a modal alert that the email field cannot be left blank.

Submit Answer

Matthew Theodore, rizqi firdaus, Jastiyah Maulidina, Rafidhia Huska Pasya, glorious satria



Scenario: User registration form validation without an email address.

Given: The user has filled out all fields except the email field.

When: The user submits the registration form.

Then: What error message should be displayed regarding the missing email?

- Display an inline error message stating "Email is required".
- Disable the submit button until the email field is completed.
- Highlight the email field with a red border and prompt the user to enter a valid email address.
- Show a modal alert that the email field cannot be left blank.

2:14 PM | vrg-ognh-xbs

Scenario: User submits a registration form with missing mandatory fields.

Given: The user leaves some required fields blank.

When: The user clicks the submit button.

Then: What feedback should the system provide to indicate the missing information?

- Highlight each missing field with an error message indicating that the field is required.
- Prevent form submission and display a summary of all missing fields.
- Disable the submit button until all mandatory fields are filled out.
- Show a modal alert listing the fields that need to be completed.

2:16 PM | vrg-ognh-xbs

Available Tasks

- Data Fetching Timeout**
Handling slow network responses
Scenario: A data fetching operation times out due to slow network.
- Server Response Delay**
Handling delays in server responses
Scenario: The server takes too long to respond.
- Unexpected System Crash**
System crashes unexpectedly during operation
Scenario: The system crashes unexpectedly during user interaction.
- Database Connection Error**
Issue connecting to the database
Scenario: The application fails to connect to the database.
- API Endpoint Not Found**
Handling 404 errors for API endpoints
Scenario: The API endpoint requested by the client does not exist.
- Token Authentication Error**
Issue with validating user token during authentication
Scenario: Token authentication fails during the login process.

2:16 PM | vrg-ognh-xbs

The screenshot shows a web application interface with a dark blue background. A central white box contains the following text:

Scenario: User attempts to log in with invalid credentials.
Given: The user has entered an incorrect username or password.
When: The user clicks the login button.
Then: What should the system do to handle the failed login attempt?

Below the text are four buttons with rounded corners:

- Display an inline error message indicating the credentials are invalid.
- Lock the account after several unsuccessful attempts.
- Provide a password recovery option.
- Show a modal alert suggesting the user check their credentials.

At the bottom of the white box is a "Submit Answer" button.

To the right of the web application is a video conference interface showing a grid of participants. The participants are:

- Adam Syam Nursal
- ALIF TAUFIQUEURRA
- Putri Auliyah Rahmah
- Anthony Febrina A...
- Regina Xylia Mihar...
- Anggeng Hudi Winoto
- Michael Timon...
- Rhehi
- glorious satria

The screenshot shows a presentation slide titled "Available Tasks" with a dark blue background. The slide contains six task cards:

- Form Validation Failure**
Check given type of Question.
Scenario: Form validation failure during job posting.
Answer: When clicking "Submit", the system shows a modal dialog alerting the user that a required field (job category) is missing.
- Missing Email Field**
Registration form without email input.
Scenario: User registration form submitted without an email address.
Answer: Display an inline error message stating "Email is required".
- Invalid Password Format**
Check password format and content.
Scenario: User attempts to register with a weak password.
- Incomplete Registration Form**
Missing mandatory fields in registration.
Scenario: User submits a registration form with missing mandatory fields.
- Error in Payment Processing**
Failure during payment transaction.
Scenario: Payment processing fails during checkout.
- User Login Failure**
Login attempt with incorrect credentials.
More options to log in with email credentials.

At the bottom of the slide are buttons for "Previous", "Page 1 of 3", and "Next".

To the right of the presentation is a video conference interface showing a grid of participants. The participants are:

- Adam Syam Nursal
- Anggeng Hudi Winoto
- Rhehi
- ALIF TAUFIQUEURRA
- Putri Auliyah Rahmah
- Anthony Febrina Arie Sena
- 3 others
- glorious satria