

# Tranformasi Digital dalam Integrasi AI dan Machine Learning menggunakan Automasi Testing

Ragil Nur Iman

Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Pamulang  
Jl. Raya Puspitek, No. 10, Buaran, Serpong, Tangerang Selatan, Banten, Indonesia

e-mail : [dosen03141@unpam.ac.id](mailto:dosen03141@unpam.ac.id)

## Abstrak

Transformasi digital telah menjadi pendorong utama dalam pengembangan teknologi informasi modern, khususnya dalam integrasi Artificial Intelligence (AI) dan Machine Learning (ML). Proses ini menuntut adaptasi sistem yang lebih canggih, cepat, dan andal. Salah satu pendekatan penting dalam mendukung transformasi ini adalah penerapan automasi testing. Automasi testing memainkan peran krusial dalam memastikan kualitas, stabilitas, dan kecepatan implementasi sistem berbasis AI/ML. Melalui automasi, pengujian terhadap model-model AI dan ML dapat dilakukan secara efisien, mulai dari validasi data, evaluasi performa model, hingga integrasi sistem secara menyeluruh. Studi ini membahas bagaimana automasi testing menjadi fondasi utama dalam integrasi teknologi AI/ML dengan sistem bisnis digital yang terus berkembang. Dengan mengurangi kesalahan manual dan mempercepat proses validasi, automasi testing mendorong terciptanya solusi digital yang lebih adaptif dan berbasis data. Hasilnya, organisasi mampu meningkatkan produktivitas, kecepatan inovasi, serta kualitas produk digital secara keseluruhan.

**Kata kunci:** Keywords: Digital Transformation, Automation Testing, Artificial Intelligence, Machine Learning, Software Testing, DevOps, Continuous Integration/Continuous Deployment (CI/CD)

## Abstract

*Digital transformation has become a major driver in the development of modern information technology, especially in the integration of Artificial Intelligence (AI) and Machine Learning (ML). This process demands a more sophisticated, fast, and reliable system adaptation. One important approach in supporting this transformation is the implementation of automated testing. Automation testing plays a crucial role in ensuring the quality, stability, and speed of implementation of AI/ML-based systems. Through automation, testing of AI and ML models can be carried out efficiently, starting from data validation, model performance evaluation, to overall system integration. This study discusses how automated testing is the main foundation in the integration of AI/ML technology with the ever-growing digital business system. By reducing manual errors and accelerating the validation process, automated testing drives the creation of more adaptive and data-driven digital solutions. As a result, organizations are able to increase productivity, speed of innovation, and the overall quality of digital products.*

**Keywords:** Keywords: Digital Transformation, Automation Testing, Artificial Intelligence, Machine Learning, Software Testing, DevOps, Continuous Integration/Continuous Deployment (CI/CD)

## 1. Pendahuluan

Perkembangan teknologi digital saat ini telah membawa dunia memasuki era transformasi digital, di mana proses bisnis, layanan publik, dan aktivitas sehari-hari semakin bergantung pada teknologi cerdas dan otomatis. Transformasi digital bukan sekadar digitalisasi, tetapi juga mencakup perubahan fundamental dalam cara organisasi beroperasi dan menciptakan nilai melalui pemanfaatan

teknologi mutakhir seperti Artificial Intelligence (AI) dan Machine Learning (ML) (Westerman et al., 2014).

AI dan ML telah menjadi fondasi dari berbagai sistem cerdas yang digunakan dalam pengambilan keputusan otomatis, personalisasi layanan, serta analisis data skala besar. Namun, tantangan utama dalam penerapan AI dan ML adalah bagaimana mengintegrasikannya ke dalam ekosistem digital yang kompleks

secara andal dan efisien (Davenport & Ronanki, 2018). Proses integrasi ini tidak lepas dari pengujian sistem yang komprehensif, di mana automasi testing memainkan peran krusial dalam memastikan performa dan stabilitas model AI/ML di lingkungan nyata.

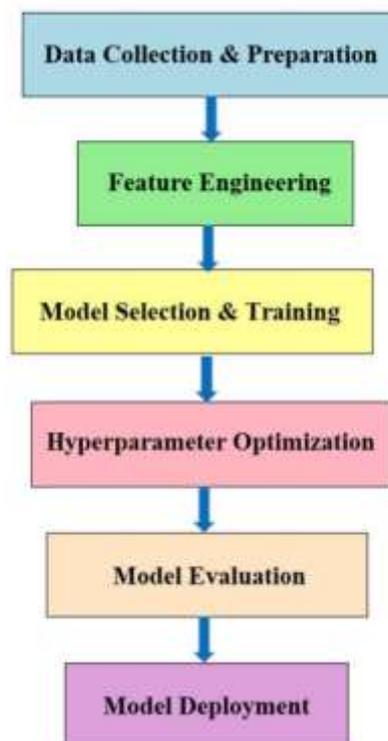
Automasi testing, atau pengujian perangkat lunak secara otomatis menggunakan skrip dan alat bantu, memberikan keuntungan signifikan dibandingkan pengujian manual. Dengan automasi, organisasi dapat mengurangi waktu pengujian, meningkatkan akurasi deteksi kesalahan, serta mempercepat siklus pengembangan perangkat lunak (Garousi et al., 2020). Dalam konteks AI dan ML, automasi testing juga mencakup validasi data, evaluasi model, pengujian integrasi API, dan regresi model secara terus-menerus.

Dengan demikian, integrasi AI dan ML dalam proses transformasi digital membutuhkan dukungan kuat dari automasi testing yang terstruktur dan berkelanjutan. Penelitian ini bertujuan untuk mengkaji peran strategis automasi testing dalam mendukung penerapan teknologi AI/ML di tengah tuntutan transformasi digital yang cepat dan kompetitif.

## 2. Metodologi

Penelitian ini menggunakan pendekatan kualitatif deskriptif dengan elemen studi kasus terapan, guna menggambarkan peran automasi testing dalam proses integrasi sistem berbasis AI/ML dalam konteks transformasi digital. Penelitian ini bertujuan untuk mengidentifikasi tantangan dan solusi teknis yang relevan, serta mengevaluasi dampak automasi terhadap kualitas dan efisiensi pengujian sistem cerdas.

Metode penelitian ini memiliki langkah-langkah terperinci seperti yang disajikan pada Gambar 2.1



Gambar 1. Langkah-langkah Penelitian

### a. Data Collection

Langkah pertama memerlukan pengumpulan data dari mesin industri, yang meliputi perangkat yang akan dipantau; meliputi; getaran, suhu, tekanan, dan kecepatan motor. Sensor ini memberikan nilai riil dari waktu ke waktu dan ini penting dalam mengidentifikasi penyakit yang dapat menyebabkan kerusakan mesin. Kumpulan data yang digunakan dalam penelitian ini terdiri dari data sensor temporal yang diperoleh dari mesin industri selama beberapa bulan dengan label kegagalan. Ini biasanya dalam bentuk fitur seperti getaran, suhu, tekanan, kecepatan motor, dan label kegagalan. Prapemrosesan data merupakan keharusan dalam kasus ini karena data sensor mentah memerlukan prapemrosesan sebelum dimasukkan ke dalam sistem, yang beberapa tantangannya meliputi; nilai yang hilang, outlier, dan noise. Pekerjaan ini menggunakan fitur Missing Completely at Random (MCAR) untuk menggeneralisasi strategi dalam menangani data yang hilang dengan imputasi K-Nearest Neighbor (KNN) untuk mempertahankan hubungan antara pembacaan sensor. Outlier yang mungkin dan mungkin diidentifikasi

menggunakan metode IQR dan diganti atau dihentikan berdasarkan batasan tertentu. Data tersebut kemudian dinormalisasi menggunakan skala Min-Max untuk menyelaraskan nilai antara 0 dan 1 dan memastikan bahwa nilai setiap fitur memainkan peran yang sama selama proses pelatihan. Kumpulan data yang segera dibuat dibagi menjadi set pelatihan dan pengujian (masing-masing dalam rasio 4:1), sehingga memungkinkan untuk memeriksa hasil yang diberikan oleh model yang bekerja dengan data yang tidak terlihat.

### ***b. Feature Engineering***

Hal ini khususnya penting untuk data deret waktu seperti sensor. Rekayasa fitur merupakan proses yang sangat penting dalam pemilihan model saat bekerja dengan model kloning. Tujuannya adalah untuk menghasilkan fitur yang lebih baik yang menangkap fitur yang lebih bermakna dalam upaya untuk menandai kegagalan mesin. Dalam studi ini, pertamanya kami menghasilkan fitur berbasis jendela geser yang menggambarkan karakteristik statistik besaran data sensor dari waktu ke waktu. Fitur-fitur ini adalah rata-rata dari berbagai orde, simpangan baku geser, dan momen orde lebih tinggi lainnya seperti kemiringan dan kurtosis. Misalnya, menggunakan rata-rata bergerak getaran lima periode dapat mengurangi faktor kebisingan yang dapat menyembunyikan perubahan yang lebih dramatis dalam perilaku mesin yang menyebabkan kegagalan.

### ***c. Model Selection and Training***

Untuk penelitian ini, XGBoost yang merupakan singkatan dari extreme gradient boosting dipilih sebagai model karena berbagai alasan; Pertama, model ini bagus dalam menangani big data. Kedua, jenis model ini sesuai untuk data terstruktur/tabular yang berisi data numerik (fitur dengan nilai kontinu) dan kategorikal (fitur dengan nilai diskrit). XGBoost adalah algoritma dari keluarga gradient boosting yang memiliki kinerja waktu dan klasifikasi yang hebat. Pertama, kita mulai dengan data pelatihan, yang digunakan untuk membangun model

XGBoost dengan hiperparameter asli yang ditetapkan ke nilai default. Ia menggunakan pohon keputusan untuk mengambil satu langkah pada satu waktu sehingga mengurangi kesalahan residual yang terjadi dalam prediksi di setiap langkah. Setiap pohon dibangun dengan serangkaian fitur tertentu, dan keputusan fitur didasarkan pada kontribusinya terhadap nilai fungsi kerugian terkecil.

### ***d. Hyperparameter Optimization.***

Namun, untuk menyempurnakan model lebih jauh, kami menggunakan metode pengoptimalan hiperparameter. Pertama, hiperparameter meliputi laju pembelajaran, jumlah pohon ( $n\_estimator$ ), kedalaman maksimum pohon, dan laju subsampel minimum yang disetel dari pencarian grid menggunakan validasi silang atau pencarian acak. Ini adalah hiperparameter penting lainnya yang menentukan kapasitas model dan menangani pembelajaran berlebihan. Misalnya, laju pembelajaran rendah yang berkisar dari 0,01 hingga 0,05 dikombinasikan dengan pembuatan lebih banyak pohon untuk memungkinkan pembelajaran yang baik. Lebih jauh, jumlah iterasi yang pencarian optimumnya dilakukan untuk parameter kedalaman maksimum harus optimal untuk menghindari underfitting maupun overfitting dan dapat bervariasi antara 6 dan 15.

### ***e. Model Evaluation***

Akhirnya setelah penyetelan model, keakuratan model diuji dengan daftar kriteria penilaian yang telah ditetapkan sebelumnya. Ukuran kinerja untuk tugas ini adalah keakuratan, presisi, penarikan kembali, dan skor F1, tetapi penarikan kembali memainkan peran penting dalam prediksi kegagalan. Karena pemeliharaan prediktif bertujuan untuk menghindari kegagalan, tingkat penarikan kembali yang tinggi berarti sebagian besar kegagalan yang mungkin akan didiagnosis dengan benar sehingga kecil kemungkinan untuk tidak mendeteksi kegagalan. Karena kinerja model dalam mengklasifikasikan kejadian menjadi kegagalan dan non-kegagalan merupakan fokus utama

makalah ini, matriks kebingungan dianalisis untuk menetapkan seberapa efektif model melakukan klasifikasi. Metrik evaluasi lain yang ditetapkan untuk menilai kemampuan model dalam membedakan antara dua kelas (kegagalan dan tidak ada kegagalan) berada di bawah skor ROC-AUC (Karakteristik Operasi Penerima - Area di Bawah Kurva).

#### **f. Linear Regression**

Model XGBoost yang disetel digunakan dalam estimasi waktu nyata dan pengambilan keputusan dalam sistem tertanam dalam lingkungan industri. Langkah terakhir adalah membuat alur kerja penyebaran model yang ditransformasikan, dengan memindahkan model ke perangkat tepi dan menggunakan teknik seperti pemangkasan model dan kuantisasi untuk meningkatkan kinerjanya dalam hal seberapa banyak data yang dapat ditangani atau seberapa cepat eksekusinya. Misalnya, pohon keputusan menggunakan pemangkasan untuk menyeimbangkan di mana hanya cabang yang kurang penting yang dihilangkan; dalam kuantisasi, presisi bobot model dikurangi untuk menghemat ruang di mana informasi yang tidak berguna tidak dipertahankan.

Setelah penerapan, sistem ini terus menganalisis data sensor waktu nyata yang dimasukkan ke dalamnya dari mesin dalam bentuk getaran, suhu, tekanan, serta kecepatan motor, dan menawarkan prediksi kegagalan dalam waktu inferensi terbatas (misalnya, 10 ms per prediksi). Ketika model menunjukkan bahwa kegagalan kemungkinan akan terjadi, alarm akan dikeluarkan sehingga staf pemeliharaan dapat melakukan tindakan pencegahan sebelum kegagalan terjadi. Ini adalah sistem berbasis tepi waktu nyata, dan ini meminimalkan waktu yang diperlukan untuk melakukan intervensi dan sangat menurunkan risiko pemadaman yang tidak terjadwal sehingga memangkas biaya operasi.

### **3. Hasil dan Pembahasan**

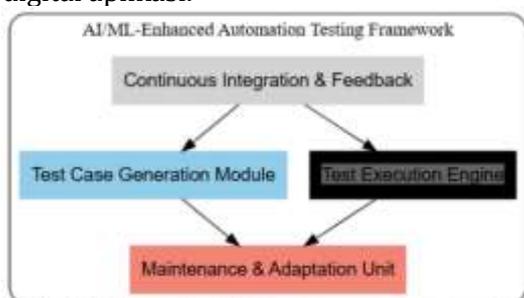
Bagian ini menyajikan Pembelajaran Terbimbing untuk Pengambilan Keputusan Prediktif yang tampaknya menggambarkan hasil pada set data Pemeliharaan Prediktif. Set data ini berisi data mentah dari sensor yang dipasang di mesin industri dan tujuannya adalah untuk meramalkan apakah mesin akan gagal dalam waktu yang ditentukan berikutnya. Ada atribut yang melibatkan getaran, suhu, tekanan, dan kecepatan putaran motor. Tujuannya adalah untuk meramalkan hasil biner: Ditetapkan dengan '1' untuk menunjukkan kegagalan atau '0' untuk menandakan tidak adanya kegagalan. Berikut ini hasil pada kinerja model yang lebih tinggi dengan metode canggih yang diterapkan untuk meningkatkan aplikasi pemeliharaan prediktif dalam industri. Ini akan mengasumsikan bahwa kami telah menerapkan Mesin Pendorong Gradien yang umumnya dikenal sebagai GBM; khususnya XGBoost, sehingga menghasilkan hasil yang lebih baik terkait dengan akurasi, presisi, penarikan kembali di antara indikator kinerja perusahaan lainnya. XGBoost paling cocok untuk masalah data terstruktur/tabular dan digunakan dalam kompetisi

Pembelajaran Mesin berisiko tinggi.

Dalam kasus ini, kami menggunakan dataset Predictive Maintenance yang terdiri dari data dari sensor mesin. Fitur-fiturnya meliputi: Data mentah: Getaran, Suhu, Tekanan, Kecepatan Motor, Label Kegagalan - Label Kegagalan dapat berupa 0 = tidak ada kegagalan dan 1 = kegagalan. Data yang dikumpulkan memiliki 15000 baris yang dikumpulkan dari beberapa bulan. Di mana setiap baris sesuai dengan pembacaan dari sensor mesin beserta label kegagalan, apakah mesin gagal dalam waktu 30 hari setelah pembacaan. Berikut adalah karakteristik setelah persiapan:

Saat aplikasi berubah seiring waktu, unit pemeliharaan dan adaptasi memastikan bahwa kerangka pengujian terus berjalan dengan baik seiring waktu. Unit ini terus menggunakan pembelajaran penguatan (RL) dan pemantauan berkelanjutan untuk memperbarui skrip pengujian: ☐

Pemantauan Berkelanjutan: Bagian ini terus memeriksa aplikasi untuk setiap perubahan pembaruan, termasuk fitur baru, perubahan kode, dan perubahan tren penggunaan. Unit mengumpulkan data waktu nyata tentang status aplikasi yang terus berkembang dengan bantuan sistem kontrol versi dan lingkungan penerapan.  $\square$  Pembelajaran Penguatan untuk Pembaruan Dinamis: Skrip pengujian diubah menggunakan pembelajaran penguatan berdasarkan status aplikasi saat ini. Misalnya, model RL dapat memprioritaskan pengembangan skrip pengujian baru atau mengubah skrip yang sudah ada untuk lebih baik mengatasi perubahan aplikasi yang rentan terhadap cacat.  $\square$  Pemeliharaan Skrip Otomatis: Unit dapat secara otomatis memperbarui atau membuat skrip pengujian untuk mencerminkan fungsionalitas aplikasi baru saat perubahan terdeteksi. Hal ini menurunkan risiko pengujian yang sudah ketinggalan zaman yang mungkin tidak dapat dilakukan pada aplikasi saat ini dengan meminimalkan intervensi manual dan memastikan pengujian selalu mutakhir. Organisasi dapat membangun kerangka kerja otomatisasi yang kuat, fleksibel, dan efektif dengan menggabungkan ketiga komponen ini: Mesin Eksekusi Pengujian, dan Unit Pemeliharaan & Adaptasi. Selain meningkatkan cakupan pengujian dan efisiensi pelaksanaan, kerangka kerja ini juga memastikan bahwa praktik pengujian terus ditingkatkan dengan transformasi digital aplikasi.



**Gambar 2.** Arsitektur Kerangka Kerja Pengujian Otomatisasi Yang Ditingkatkan AI/ML.

### A. Pengumpulan dan Praproses Data

Kualitas dan organisasi data pelatihan sangat penting saat membuat model

pembelajaran mesin yang kuat untuk pengujian otomatis dalam inisiatif transformasi digital. Data dari jalur integrasi berkelanjutan (CI) yang mencakup beberapa proyek dikumpulkan dalam konteks ini. Log cacat, modifikasi kode, dan hasil kasus uji dari masa lalu semuanya disertakan dalam pengumpulan data yang bervariasi ini; masing-masing menawarkan wawasan penting tentang siklus hidup pengembangan perangkat lunak.

### Praproses: Pembersihan dan Penataan Data

Sebelum memberikan data ini ke dalam model ML, praproses menyeluruh dilakukan untuk memastikan kualitasnya tetap terjaga. Jalur praproses biasanya mencakup langkah-langkah berikut:

#### Normalisasi Data

Proses normalisasi memerlukan konversi nilai yang diukur pada beberapa skala ke skala umum. Dengan menjamin bahwa setiap area berkontribusi secara merata, normalisasi mengurangi kemungkinan skala yang lebih besar dapat memengaruhi model secara tidak proporsional.

#### Pemilihan Fitur

Pemilihan Fitur adalah menemukan dan menyimpan hanya fitur-fitur yang memengaruhi model secara signifikan. Subset dari modal yang dipilih adalah menyederhanakan yang mengurangi kemungkinan overfitting dan meningkatkan generalisasinya.

Pengurangan Dimensionalitas Jumlah fitur tetap tinggi, yang akan mengakibatkan kutukan dimensionalitas, di mana kinerja model dapat menurun. Ini meningkatkan visualisasi dan interpretabilitas model.

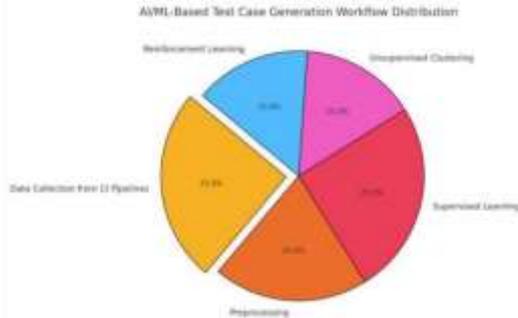
#### Meningkatkan Kinerja Model

Secara keseluruhan, proses praproses ini menjamin bahwa data bersih digunakan untuk melatih model ML. Fondasi data yang kuat ini sangat penting dalam mencapai jaminan kualitas yang andal dan terukur dalam proyek transformasi digital.

## B. Implementasi Algoritma AI/ML

Untuk mengatasi berbagai tantangan dalam manajemen kasus pengujian, metode ini menggunakan tiga teknik dasar - pembelajaran terbimbing, pengelompokan tak terbimbing, dan pembelajaran penguatan.

Buka di Google Terje



**Gambar 3.** Diagram lingkaran ini menggambarkan distribusi alur kerja pembuatan kasus uji berbasis AI/ML

## Pembelajaran Terbimbing untuk Klasifikasi dan Prioritas

Pembelajaran terbimbing merupakan inti dari kemampuan modul untuk memprediksi kemungkinan cacat dan memprioritaskan kasus uji secara efektif. Modul ini terdiri dari data uji historis, yang terdiri dari perubahan kode dan log cacat. Model mesin dilatih dengan data berlabel ini. Seiring berjalannya waktu, model dilatih ulang, sehingga tetap relevan seiring pengujian otomatisasi berkembang.

## Pengelompokan Tanpa Pengawasan untuk Deteksi Anomali

Teknik pengelompokan tanpa pengawasan melengkapi pembelajaran terbimbing dengan mengungkap pola dan anomali tersembunyi dalam hasil pengujian. Data hasil pengujian dikenakan metode seperti DBSCAN, k-means, dan pengelompokan hierarkis. Metode ini mengklasifikasikan kasus uji terkait berdasarkan fitur seperti perilaku eksekusi dan metrik kinerja. Wawasan yang diperoleh dari pengelompokan menginformasikan pembuatan kasus uji lebih lanjut. Pengelompokan anomali memicu pembuatan kasus uji tambahan atau khusus, yang memastikan cakupan komprehensif dari skenario yang tidak biasa. Pembelajaran Penguatan untuk

## Adaptasi Dinamis

Pembelajaran Penguatan memainkan peran penting dalam memastikan bahwa kasus pengujian tetap efektif dan relevan saat aplikasi berubah. Untuk berkomunikasi dengan lingkungan pengujian secara terus-menerus, agen RL dimasukkan ke dalam kerangka pengujian. Agen mempelajari cara terbaik untuk mengubah pengujian berdasarkan umpan balik yang diterimanya. Dengan mengubah strateginya, agen RL memastikan bahwa rangkaian pengujian diperbarui seiring dengan fitur dan kompleksitas aplikasi.

## C. Pengaturan Eksperimen

Prototipe kerangka kerja yang diusulkan diimplementasikan dan dievaluasi pada proyek transformasi digital berukuran sedang di domain layanan keuangan. Pengaturan eksperimental membandingkan alat pengujian otomatisasi tradisional dengan pendekatan AI/ML kami yang disempurnakan di beberapa indikator kinerja utama (KPI), termasuk cakupan pengujian, tingkat deteksi kesalahan, dan waktu eksekusi.

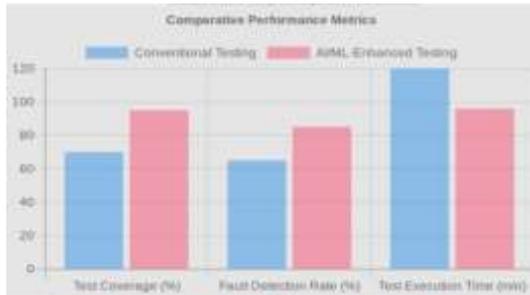
## D. Metrik Kinerja

Eksperimen kami menghasilkan pengamatan berikut:

**Peningkatan Cakupan Pengujian:** Modul pembuatan kasus uji otomatis meningkatkan cakupan sebesar 25% dibandingkan dengan rangkaian pengujian yang disimpan secara manual.

**Peningkatan Deteksi Kesalahan:** Integrasi AI menghasilkan peningkatan deteksi cacat dini sebesar 30%, khususnya dalam siklus pengujian regresi.

**Pengurangan Waktu Eksekusi:** Penjadwalan dan penentuan prioritas pengujian yang dioptimalkan menghasilkan pengurangan 20% dalam waktu eksekusi pengujian secara keseluruhan.



**Gambar 4.** Merangkum Metrik Kinerja Komparatif

#### 4. Kesimpulan

Revolusi digital datang dengan cepat dan dahsyat. Tim perangkat lunak kini berlomba-lomba untuk memberikan hasil lebih cepat, mengatasi bug lebih cepat, dan mempertahankan kualitas yang sempurna. Pengujian konvensional tidak mampu mengejar ketertinggalan. Ia bergerak terlalu lambat. Terlalu tidak fleksibel. Oleh karena itu, kami menciptakan kerangka kerja yang lebih cerdas yang digerakkan oleh AI dan ML dan yang melakukan lebih dari sekadar mengotomatisasi; ia juga belajar. Ia memeriksa log cacat, mengikuti perubahan kode, dan melihat data uji dari masa lalu. Setelah itu, ia membuat pilihan yang sebenarnya. Ia pertama-tama mengevaluasi prediksi tentang apa yang paling mungkin gagal. Untuk memastikan bahwa wilayah berdampak tinggi dievaluasi lebih awal—sebelum masalah mencapai produksi—algoritma pembelajaran yang diawasi menilai kasus uji menurut risiko. Hasilnya? siklus rilis yang lebih cepat, lebih sedikit cacat yang diabaikan, dan umpan balik yang lebih cepat kepada pengembang. Namun, itu tidak berakhir di sana.

#### Daftar Pustaka

- A. Kumar and R. Patel, –Machine learning approaches in software testing: A survey,|| IEEE Trans. Software Eng., vol. 45, no. 4, pp. 456–478, Apr. 2020. DOI:10.21275/SR24617015431
- Abu Bakar, N. S. A. (2024). Machine Learning Implementation in Automated Software Testing: A Review. *Journal of Data Analytics and*

*Artificial Intelligence Applications.*

<https://doi.org/10.26650/d3ai.00>

- Afrizal Zein, 2022. Evaluasi Keamanan Wireless LAN Menggunakan Issaf (Information System Security Assessment Framework), *Sainstech: Jurnal Penelitian dan Pengkajian Sains dan Teknologi: Vol. 32 No. 2:* DOI:

<https://doi.org/10.37277/stch.v32i2>

- Baqar, M., & Khanda, R. (2024). The Future of Software Testing: AI-Powered Test Case Generation and Validation. *arXiv preprint arXiv:2409.05808.*

<https://arxiv.org/abs/2409.05808>

- L. Zhang, –Digital transformation: A conceptual framework,|| in Proc. Int. Conf. Digit. Innov., Berlin, Germany, 2019, pp. 112–117. DOI:10.3846/cibmee.2019.073

- Ramadan, A., Yasin, H., & Pektas, B. (2024). The Role of Artificial Intelligence and Machine Learning in Software Testing. *arXiv preprint arXiv:2409.02693.*

<https://arxiv.org/abs/2409.02693>

- Ricca, F., Marchetto, A., & Stocco, A. (2024). A Multi-Year Grey Literature Review on AI-assisted Test Automation. *arXiv preprint arXiv:2408.06224.*

<https://arxiv.org/abs/2408.06224>

- ng and P. Reddy, –Automated test generation using AI techniques,|| *IEEE Software*, vol. 37, no. 2, pp. 62–69, Mar. 2021

- Samsu Supriyatna, Salman Farizy, (2024) Perancangan dan Implementasi Aplikasi Monitoring Berkas Pencairan Dana Berbasis Web Menggunakan Metode Rapid Application Development, *Sainstech: Jurnal Penelitian dan Pengkajian Sains Dan Teknologi: Vol. 34 No. 3:* pp.DOI:

<https://doi.org/10.37277/stch.v34i3.2078>

- S. Chakraborty and D. Roy, –AI-powered automation testing: A case study on efficiency and effectiveness,|| in Proc. IEEE Int. Conf. AI Appl., New York, NY, USA, 2022, pp. 79–85.

Tranformasi Digital dalam Integrasi AI dan Machine Learning menggunakan Automasi Testing  
Author : Ragil Nur Iman- Sainstech Vol. 35 No. 2 (Juni 2025): 34-41  
DOI: <https://doi.org/10.37277/stch.v35i2.2344>

DOI:10.13140/RG.2.2.22649.89448