

RESOLVER PROPERTI EFEKTIF UNTUK PEMERIKSAAN FORMAT SKRIPSI BERBASIS ATURAN DENGAN EVALUASI INJEKSI KESALAHAN

EFFECTIVE PROPERTY RESOLVER FOR RULE-BASED THESIS FORMAT CHECKING WITH ERROR INJECTION EVALUATION

**Bayu Charisma Putra^{1*}, Gaguk Triono², Dhofirul Yahya³, Mokhammad Fathoni
Rokhman⁴**

Program Studi Teknik Informatika, Fakultas Teknik, Universitas Maarif Hasyim Latif,
Sidoarjo, Indonesia^{1,2,3,4}

bayu_charisma@dosen.umaha.ac.id

ABSTRACT

Thesis formatting compliance is commonly assessed manually by supervisors, examiners, and academic programs, resulting in inconsistent evaluations and allowing formatting violations to remain undetected until approval. Existing checking tools often rely on directly assigned formatting or style definitions, overlooking the layered inheritance mechanisms that determine the effective properties displayed in Microsoft Word. This study develops a rule-based thesis format checker that directly processes Microsoft Word (.docx) files using Office Open XML. The system integrates an effective-property resolver that follows Word's formatting inheritance hierarchy, an intermediate document representation, 44 rules organized into nine categories, and a configurable profile that separates institutional requirements from application code. A web-based interface presents supporting evidence for each detected violation. The evaluation was conducted in the Informatics Engineering Study Program at Universitas Maarif Hasyim Latif through two procedures: verification of evidence cards by the program head using 17 approved theses and controlled fault injection into two reference documents, producing 164 mutants with expected rule sets established before execution. Following one verified revision cycle, deterministic rules achieved a precision of 1.000 (90/90) on actual theses. For injected faults, the system achieved 0.985 precision, 0.985 recall, and an F1-score of 0.985. Removing the resolver reduced recall for direct-formatting mutants from 1.000 to 0.773 and specificity from 1.000 to 0.693. Residual violations, representing a lower-bound estimate, were identified in 15.9% of deterministic document–rule pairs among theses previously approved by supervisors and examiners. These findings demonstrate that resolving effective formatting properties is essential for accurate automated checking and that evidence-based evaluation improves the reproducibility of accuracy assessments.

Keywords: *automated format checking; fault injection; Office Open XML; rule-based system*

Abstrak

Kepatuhan terhadap format skripsi umumnya diperiksa secara manual oleh dosen pembimbing, dosen penguji, dan program studi. Proses ini dapat menghasilkan penilaian yang tidak konsisten serta menyebabkan pelanggaran format tidak terdeteksi hingga skripsi disetujui. Perangkat pemeriksa yang tersedia umumnya hanya mengandalkan format yang ditetapkan secara langsung atau definisi gaya (*style*), tanpa memperhitungkan mekanisme pewarisan berlapis yang menentukan properti format efektif sebagaimana ditampilkan dalam Microsoft Word. Penelitian ini mengembangkan sistem pemeriksa format skripsi berbasis aturan yang memproses berkas Microsoft Word (.docx) secara langsung menggunakan Office Open XML. Sistem ini mengintegrasikan resolver properti efektif yang mengikuti hierarki pewarisan format Microsoft Word, representasi dokumen perantara, 44 aturan yang dikelompokkan ke dalam sembilan kategori, serta profil konfigurasi yang memisahkan nilai persyaratan institusi dari kode aplikasi. Antarmuka berbasis web menyajikan bukti pendukung untuk setiap pelanggaran yang terdeteksi. Evaluasi dilakukan di Program Studi Teknik Informatika Universitas Maarif Hasyim Latif melalui dua prosedur, yaitu verifikasi kartu bukti oleh ketua program studi terhadap 17 skripsi yang telah disetujui serta injeksi kesalahan secara terkendali pada dua dokumen acuan yang menghasilkan 164 dokumen mutan. Kumpulan aturan yang diharapkan untuk setiap mutan ditetapkan sebelum sistem dijalankan. Setelah satu siklus revisi yang telah diverifikasi, aturan deterministik mencapai presisi 1,000 (90/90) pada skripsi aktual. Pada pengujian dengan injeksi kesalahan, sistem mencapai presisi 0,985, *recall* 0,985, dan skor F1 sebesar 0,985. Penghilangan resolver menurunkan *recall* pada mutan dengan format yang ditetapkan secara langsung dari 1,000 menjadi 0,773 serta menurunkan spesifisitas dari 1,000 menjadi 0,693. Pelanggaran format yang masih tersisa, yang merepresentasikan estimasi batas bawah, ditemukan pada 15,9% pasangan dokumen–aturan deterministik dalam skripsi yang sebelumnya telah melalui pemeriksaan dosen pembimbing dan penguji. Temuan ini menunjukkan bahwa resolusi properti format efektif merupakan faktor penting dalam meningkatkan akurasi

pemeriksaan otomatis, sedangkan evaluasi berbasis bukti memungkinkan klaim akurasi diperiksa dan direproduksi secara lebih transparan.

Kata kunci: pemeriksaan format otomatis; injeksi kesalahan; Office Open XML; sistem berbasis aturan

PENDAHULUAN

Perkembangan teknologi informasi telah mendorong pemanfaatan otomatisasi dalam berbagai proses akademik, termasuk pemeriksaan format dokumen ilmiah. Skripsi sebagai salah satu bentuk karya ilmiah mahasiswa harus memenuhi ketentuan penulisan yang ditetapkan oleh perguruan tinggi, seperti ukuran kertas, margin halaman, jenis dan ukuran huruf, spasi, penomoran, struktur bab, serta format tabel, gambar, dan daftar pustaka; kesalahan pada aspek-aspek tersebut termasuk yang paling sering ditemukan dalam naskah skripsi (Putra et al., 2026). Kepatuhan terhadap ketentuan tersebut diperlukan untuk menjaga konsistensi penyajian dokumen dan memenuhi standar administrasi akademik. Namun, pemeriksaan format skripsi masih sering dilakukan secara manual menggunakan aplikasi pengolah kata. Proses ini membutuhkan waktu, bergantung pada ketelitian pemeriksa, dan berpotensi menghasilkan penilaian yang tidak konsisten, terutama ketika jumlah dokumen yang diperiksa cukup besar dan aturan format yang diterapkan beragam.

Salah satu pendekatan yang dapat digunakan untuk mengatasi permasalahan tersebut adalah pemeriksaan dokumen berbasis aturan (*rule-based document checking*). Pendekatan ini memungkinkan ketentuan format diterjemahkan menjadi aturan yang dapat dievaluasi secara otomatis berdasarkan karakteristik dokumen (Solihin & Eastman, 2015). Dalam konteks dokumen Microsoft Word, format dokumen disimpan dalam struktur Office Open XML (OOXML), yang memuat informasi mengenai paragraf, *run* teks, *style*, bagian dokumen, dan berbagai properti pemformatan lainnya (Ecma International, 2016). Meskipun struktur tersebut menyediakan informasi yang diperlukan untuk pemeriksaan otomatis,

penentuan format aktual suatu elemen tidak selalu dapat dilakukan dengan membaca satu properti secara langsung. Sebagian properti ditetapkan melalui pemformatan langsung pada elemen teks atau paragraf, sedangkan properti lainnya diwariskan dari *style* yang diterapkan maupun dari definisi pemformatan bawaan dokumen. Akibatnya, pemeriksa yang hanya membaca nilai pada satu tingkat struktur berpotensi menghasilkan keputusan yang tidak sesuai dengan format efektif yang digunakan oleh dokumen.

Permasalahan tersebut menunjukkan pentingnya mekanisme *resolver properti efektif*, yaitu komponen yang menentukan nilai pemformatan akhir suatu elemen dengan mempertimbangkan sumber properti dan mekanisme pewarisannya. Resolver diperlukan agar pemeriksaan format tidak hanya bergantung pada nilai yang ditetapkan secara langsung, tetapi juga memperhitungkan nilai yang berasal dari hierarki *style* dan properti bawaan dokumen sesuai dengan aturan pemformatan OOXML. Dengan mekanisme ini, sistem diharapkan mampu membedakan dokumen yang benar-benar melanggar ketentuan dari dokumen yang hanya memiliki perbedaan pada lokasi penyimpanan properti pemformatannya. Ketepatan resolusi properti juga menjadi dasar penting untuk menghasilkan temuan yang dapat ditelusuri melalui informasi mengenai lokasi pelanggaran, nilai yang diharapkan, nilai aktual, serta sumber nilai pemformatan yang digunakan dalam proses evaluasi.

Meskipun pemeriksaan berbasis aturan dapat menghasilkan keputusan yang terstruktur dan konsisten, kinerja sistem tetap perlu dievaluasi menggunakan data yang memungkinkan kebenaran hasilnya diketahui. Evaluasi yang hanya menggunakan dokumen nyata yang telah disetujui belum cukup untuk mengukur

kemampuan sistem dalam mendeteksi seluruh jenis pelanggaran, karena jumlah kesalahan yang tidak terdeteksi pada dokumen tersebut umumnya tidak diketahui. Sebaliknya, evaluasi melalui injeksi kesalahan (*fault injection*) memungkinkan pelanggaran dibuat secara terkendali berdasarkan definisi aturan yang telah ditetapkan (Arocena et al., 2015; Barr et al., 2015). Kesalahan dapat ditanam melalui pemformatan langsung maupun melalui *style*, dengan variasi tingkat penyimpangan dan proporsi elemen yang tidak patuh. Dengan menetapkan himpunan pelanggaran yang diharapkan sebelum sistem dijalankan, hasil pemeriksaan dapat dibandingkan dengan kondisi yang telah diketahui untuk menghitung presisi (*precision*), *recall*, F1, dan spesifisitas. Pendekatan ini juga memungkinkan pengujian terhadap batas toleransi dan ambang kepatuhan yang digunakan oleh sistem.

Penelitian terdahulu telah membahas otomatisasi pemeriksaan format dokumen akademik (Aulia & Haviana, 2026; Azzahra et al., 2025; Bloch et al., 2025; Martsinkevich et al., 2023; Ramadhan, 2025; Svintsytska et al., 2025), validasi dokumen teknis (Hlukhov & Sydorko, 2023; Sydorko & Hlukhov, 2024; Verbovyi & Saiapina, 2025), analisis tata letak (Binmakhashen & Mahmoud, 2019), penilaian dokumen akademik berbasis model bahasa dan umpan balik penulisan (Fröhlich & Schlippe, 2025; Knight et al., 2020; Zhang et al., 2025), serta penggunaan pengujian berbasis mutasi untuk mengevaluasi perangkat lunak (Andrews et al., 2005; DeMillo et al., 1978; Jia & Harman, 2011; Just et al., 2014; Papadakis et al., 2019). Namun, masih diperlukan evaluasi yang secara khusus menghubungkan resolusi properti pemformatan efektif dengan ketepatan pemeriksaan aturan pada dokumen skripsi. Selain itu, pengujian perlu membedakan kemampuan sistem dalam mengenali pelanggaran yang benar-benar ditanam dari kecenderungannya melaporkan dokumen

yang sebenarnya telah memenuhi ketentuan. Perbedaan ini penting karena peningkatan sensitivitas deteksi tidak selalu diikuti oleh peningkatan ketepatan, terutama ketika sistem menggunakan ambang kepatuhan atau toleransi pemformatan.

Berdasarkan permasalahan tersebut, penelitian ini bertujuan mengembangkan dan mengevaluasi pemeriksa format skripsi berbasis aturan yang dilengkapi *resolver* properti efektif untuk dokumen Microsoft Word berformat .docx. Evaluasi dilakukan melalui dua pendekatan, yaitu pemeriksaan terhadap skripsi nyata dan pengujian injeksi kesalahan pada dokumen acuan yang telah diverifikasi. Penelitian ini dipandu empat pertanyaan penelitian: seberapa akurat sistem, dibedakan antara aturan deterministik dan heuristik (RQ1); seberapa besar kontribusi *resolver* properti efektif dibandingkan mode pemeriksaan naif yang hanya membaca definisi *style* (RQ2); seberapa sensitif hasil terhadap perubahan ambang kepatuhan dan toleransi pemformatan (RQ3); dan seberapa umum pelanggaran format pada skripsi yang sudah dinyatakan lulus (RQ4). Hasil penelitian diharapkan memberikan kontribusi teknis dalam pengembangan pemeriksa dokumen akademik yang lebih akurat, transparan, dan dapat direproduksi, sekaligus menyediakan dasar evaluasi yang lebih terukur bagi penerapan otomatisasi pemeriksaan format skripsi di lingkungan perguruan tinggi.

METODE

Penelitian ini menggunakan pendekatan rancang bangun sistem yang dilanjutkan dengan evaluasi empiris. Tahap pengembangan dilakukan terlebih dahulu, kemudian versi mesin pemeriksa dan profil konfigurasinya ditetapkan sebagai versi tetap sebelum pelaksanaan pengujian melalui dua komponen evaluasi yang saling melengkapi (Tabel 1). Pengujian pertama (Uji 1) dilaksanakan dalam dua iterasi. Temuan berupa kartu bukti yang dinyatakan tidak tepat pada iterasi awal

digunakan sebagai dasar penyempurnaan mesin pemeriksa. Setelah perbaikan dilakukan, versi mesin ditetapkan kembali dan Uji 1 diulang sebagaimana dijelaskan pada subbab Data Penelitian dan Prosedur Evaluasi.

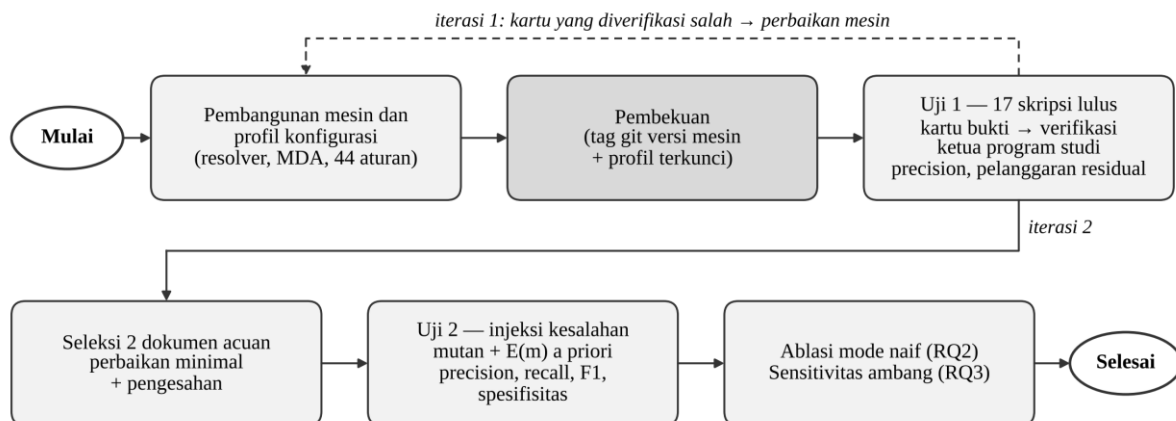
Uji 1 bertujuan mengukur presisi sistem pada dokumen skripsi aktual. Namun, pengujian ini belum dapat digunakan untuk menghitung *recall* karena pelanggaran yang tidak terdeteksi atau tidak dilaporkan oleh sistem tidak tersedia sebagai temuan yang dapat diverifikasi.

Keterbatasan tersebut diatasi melalui Uji 2, yaitu pengujian menggunakan data semi-sintetis yang dibentuk melalui injeksi kesalahan dengan kebenaran acuan yang telah ditentukan sejak tahap pembuatannya. Melalui pengujian ini, kinerja sistem dievaluasi berdasarkan presisi, *recall*, skor F1, dan spesifisitas (*specificity*). Secara keseluruhan, alur pelaksanaan penelitian, mulai dari pengembangan sistem hingga kedua tahap evaluasi, disajikan pada Gambar 1.

Tabel 1. Komponen Evaluasi Penelitian

Komponen Evaluasi	Data Pengujian	Sumber Kebenaran Acuan	Parameter yang Diukur
Uji 1: Evaluasi pada dokumen aktual	17 skripsi yang telah dinyatakan lulus	Verifikasi kartu bukti oleh ketua program studi	Presisi dan pelanggaran format residual
Uji 2: Evaluasi melalui injeksi kesalahan	2 dokumen acuan dan 164 dokumen mutan	Manifes mutasi yang ditetapkan secara <i>a priori</i> sebelum pengujian	Presisi, <i>recall</i> , skor F1, spesifisitas (<i>specificity</i>), analisis ablasi, dan analisis sensitivitas

Sumber: Hasil rancangan dan evaluasi penelitian.



Gambar 1. Alur penelitian.

Mesin Pemeriksa dan Resolver Properti Efektif

Mesin pemeriksa dirancang untuk memproses dokumen berformat .docx melalui enam tahapan utama, yaitu validasi berkas, pembacaan XML secara aman, pembentukan Model Dokumen Antara (MDA), segmentasi dokumen, eksekusi aturan pemeriksaan, dan penyusunan laporan hasil. MDA merepresentasikan

elemen-elemen dokumen yang relevan, meliputi paragraf, *run*, seksi, *header* dan *footer* efektif, tabel, gambar, serta bab. Setiap paragraf dilengkapi dengan informasi properti format efektif yang diperoleh melalui mekanisme resolusi. Pada tahap segmentasi, dokumen dikelompokkan menjadi bagian awal, isi utama, dan bagian akhir. Sistem juga mengidentifikasi wilayah daftar isi agar

entri di dalamnya tidak keliru diinterpretasikan sebagai judul bab.

Resolver properti efektif menentukan nilai format pada elemen *run* dengan menerapkan hierarki pencarian bertingkat. Proses pencarian dimulai dari (1) pemformatan langsung pada *run*; (2) *style* karakter beserta rantai pewarisan *w:basedOn*; (3) *style* paragraf atau *style* paragraf bawaan apabila paragraf tidak memiliki deklarasi *style*; (4) nilai format bawaan dokumen melalui *w:docDefaults*; dan (5) font tema apabila referensi jenis huruf menggunakan mekanisme tema. Pencarian dihentikan ketika nilai properti ditemukan pada tingkat yang relevan sesuai mekanisme resolusi yang diterapkan. Selain nilai akhirnya, *resolver* menyimpan informasi mengenai sumber atau tingkat pewarisan properti tersebut. Informasi ini digunakan sebagai bukti pendukung pada kartu temuan sehingga pengguna dapat menelusuri dasar penetapan hasil pemeriksaan. Untuk mengevaluasi kontribusi *resolver*, sistem juga menyediakan mode pemeriksaan naif yang hanya membaca properti dari definisi *style* tanpa mempertimbangkan pemformatan langsung.

Selain mekanisme pewarisan format, sistem mengakomodasi sejumlah perilaku Microsoft Word yang teridentifikasi pada korpus pengujian. Implementasinya dicocokkan dengan dokumen PDF yang dihasilkan melalui ekspor dari Microsoft Word 2021. Perilaku tersebut mencakup penggunaan *style* Normal pada paragraf tanpa deklarasi *w:pStyle*, prioritas referensi font tema, simulasi penomoran otomatis termasuk label *legal* melalui *w:isLgl*, serta pengenalan nomor halaman yang direpresentasikan sebagai *field* PAGE maupun angka yang dimasukkan secara manual. Sistem juga menangani penggunaan *header* dan *footer* khusus pada halaman pertama setiap seksi serta mengabaikan teks tersembunyi dan teks berwarna putih sesuai kebijakan pemeriksaan yang diterapkan.

Penentuan tingkat judul (*heading*) dilakukan dengan menggabungkan beberapa indikator secara berurutan, yaitu nilai *w:outlineLvl*, nama *style*, pola teks, dan penomoran otomatis. Apabila suatu judul hanya dapat dikenali berdasarkan pola teks, tingkat kepastian identifikasinya dianggap lebih rendah. Oleh sebab itu, aturan yang bergantung pada identifikasi tersebut tidak langsung menghasilkan status PELANGGARAN, tetapi diturunkan menjadi PERINGATAN. Strategi ini diterapkan untuk mengurangi risiko kesalahan deteksi akibat identifikasi judul yang belum didukung oleh struktur dokumen yang memadai.

Katalog Aturan dan Profil Konfigurasi

Katalog pemeriksaan terdiri atas 44 aturan yang dikelompokkan ke dalam sembilan kategori sebagaimana disajikan pada Tabel 2. Berdasarkan mekanisme implementasinya, aturan dibedakan menjadi aturan berbasis Office Open XML (OOXML) sebanyak 19 aturan, aturan berbasis teks sebanyak 15 aturan, dan aturan heuristik sebanyak 10 aturan. Aturan OOXML dan aturan berbasis teks bersifat deterministik serta dapat menghasilkan status PELANGGARAN apabila kondisi ketidaksesuaian terpenuhi. Sementara itu, aturan heuristik dibatasi pada pemberian status PERINGATAN karena hasil evaluasinya bergantung pada indikator atau pola yang tidak selalu memberikan kepastian penuh.

Sistem menyediakan empat status hasil pemeriksaan, yaitu LULUS, PELANGGARAN, PERINGATAN, dan TIDAK_BERLAKU. Dalam profil yang digunakan untuk evaluasi, aturan ELM-05 yang memeriksa batas abstrak maksimal satu halaman dinonaktifkan. Penonaktifan ini dilakukan karena pedoman yang menjadi acuan menetapkan panjang abstrak berdasarkan jumlah kata, bukan jumlah halaman. Dengan demikian, dari keseluruhan 44 aturan yang tersedia, terdapat 43 aturan aktif dalam evaluasi, yang terdiri atas 34 aturan deterministik dan 9 aturan heuristik.

Tabel 2. Katalog Aturan Pemeriksaan Berdasarkan Kelompok

No.	Kelompok Aturan	Kode Aturan	OOXML TEKS HEUR		
1	A. Halaman Sampul	SMP-01–SMP-05	1	0	4
2	B. Lembar Pengesahan	PGS-01–PGS-03	0	1	2
3	C. Halaman dan Margin	HAL-01–HAL-04	4	0	0
4	D. Huruf dan Paragraf	FNT-01–FNT-04, PAR-01–PAR-04	7	0	1
5	E. Penomoran	NOM-01–NOM-05	3	2	0
6	F. Struktur dan Heading	STR-01–STR-05	1	4	0
7	G. Tabel dan Gambar	TBG-01–TBG-04	2	2	0
8	H. Elemen Wajib	ELM-01–ELM-05	0	4	1
9	I. Daftar Pustaka	PUS-01–PUS-05	1	2	2
Total 9 Kelompok		44 Aturan	19	15	10

Keterangan: OOXML = aturan berbasis Office Open XML; TEKS = aturan berbasis teks; HEUR = aturan heuristik. Sumber: Hasil pengembangan katalog aturan penelitian.

Catatan: Pada profil evaluasi, aturan ELM-05 dinonaktifkan sehingga jumlah aturan aktif menjadi 43, terdiri atas 19 aturan OOXML, 15 aturan TEKS, dan 9 aturan HEUR.

Seluruh parameter persyaratan format dikelola melalui profil konfigurasi yang terpisah dari kode sumber aplikasi. Profil evaluasi disusun berdasarkan *Buku Pedoman Proposal dan Tugas Akhir Program Studi Teknik Informatika UMAHA* (2021), dengan mempertimbangkan hasil konfirmasi kepada ketua program studi mengenai ketentuan yang berlaku pada tahun penelitian. Untuk aspek yang belum diatur secara eksplisit dalam pedoman, parameter ditetapkan dengan mengacu pada templat yang telah disetujui oleh ketua program studi.

Ketentuan format yang diterapkan mencakup penggunaan kertas A4; margin atas, bawah, kiri, dan kanan masing-masing sebesar 4, 3, 4, dan 3 cm; penggunaan Times New Roman berukuran 12 pt untuk teks isi dan 14 pt untuk judul bab; spasi 1,5; serta indentasi baris pertama sebesar 1,27 cm. Penomoran bab menggunakan angka Romawi. Nomor halaman pada bagian awal ditulis menggunakan angka Romawi kecil dan ditempatkan di tengah bawah, sedangkan bagian isi menggunakan angka Arab yang diletakkan di kanan atas, kecuali halaman pertama setiap bab yang menempatkan nomor halaman di tengah bawah. Selain itu, judul tabel diletakkan di

atas tabel, sementara judul gambar ditempatkan di bawah gambar. Panjang abstrak dibatasi maksimal 350 kata, sedangkan daftar pustaka mengikuti gaya APA dengan format indentasi menggantung (*hanging indent*) dan urutan alfabetis.

Pemeriksaan numerik dilakukan dengan menerapkan toleransi sebesar 0,05 cm untuk margin, 0,10 cm untuk ukuran kertas, 0,05 untuk spasi, 0,15 cm untuk indentasi, dan 0 pt untuk ukuran huruf. Khusus aturan berbasis paragraf, yaitu FNT-01, FNT-02, PAR-01 hingga PAR-04, serta PUS-02, PUS-03, dan PUS-05, status lulus diberikan apabila sekurang-kurangnya 90% paragraf yang dievaluasi memenuhi ketentuan. Adapun aturan lainnya dinyatakan lulus apabila setiap unit pemeriksaan yang relevan telah sesuai dengan persyaratan yang ditetapkan.

Data Penelitian dan Prosedur Evaluasi

Data penelitian pada Uji 1 terdiri atas 17 dokumen skripsi yang telah dinyatakan lulus di Program Studi Teknik Informatika Universitas Maarif Hasyim Latif (UMAHA), dengan pelaksanaan sidang hingga tahun 2026. Seluruh dokumen berformat .docx dan diberi kode anonim D01–D17. Panjang dokumen berkisar

antara 68 hingga 131 halaman, dengan jumlah seksi antara 6 dan 104 pada setiap dokumen. Empat dokumen, yaitu D13, D14, D15, dan D17, menunjukkan karakteristik yang mengindikasikan bahwa dokumen tersebut berasal dari konversi PDF ke Word. Karakteristik ini antara lain berupa margin bawah sekitar 0,5 cm pada sebagian besar seksi dan penggunaan kotak teks dalam jumlah besar. Keempat dokumen tetap dipertahankan tanpa perubahan untuk merepresentasikan kondisi dokumen yang ditemukan dalam praktik. Seluruh skripsi disusun ketika pedoman tahun 2021 masih berlaku sehingga penilaian ketidaksesuaian dilakukan berdasarkan ketentuan yang berlaku pada saat masing-masing skripsi disusun. Penggunaan data telah memperoleh izin dari program studi. Untuk menjaga kerahasiaan identitas, anonimisasi diterapkan pada nama berkas, tabel, dan tangkapan layar, sedangkan bagian yang memuat nama mahasiswa, nomor induk mahasiswa (NIM), dan nama dosen pembimbing diburamkan. Isi berkas .docx tidak diubah karena menjadi objek pemeriksaan sistem. Sebagai acuan visual untuk menentukan lokasi halaman, setiap dokumen diekspor ke format PDF menggunakan Microsoft Word 2021.

Sebelum evaluasi dilaksanakan, versi kode sumber mesin pemeriksa diberi penanda (*tag*) Git dan profil konfigurasi dikunci untuk menjaga konsistensi pengujian. Perbaikan terhadap sistem setelah pengujian hanya diperbolehkan apabila seluruh rangkaian evaluasi dijalankan kembali serta hasil sebelum dan sesudah perbaikan dilaporkan secara transparan. Prosedur ini diterapkan untuk mencegah pemanfaatan informasi dari data uji dalam proses pengembangan yang tidak terdokumentasi (Kapoor & Narayanan, 2023). Pembekuan sistem dilakukan dalam dua tahap. Tahap pertama, dengan identitas beku-uji-2026-10-11, digunakan untuk menghasilkan hasil Uji 1 iterasi pertama. Kartu bukti yang dinilai tidak tepat pada iterasi tersebut kemudian dianalisis sebagai

dasar perbaikan mesin tanpa mengubah profil konfigurasi. Setelah perbaikan selesai, sistem dibekukan kembali dengan identitas beku-uji-2026-10-13b. Hasil iterasi kedua digunakan sebagai dasar penyajian seluruh hasil utama, termasuk Uji 2, sedangkan hasil iterasi pertama disajikan sebagai pembanding pada subbab Presisi dan Pelanggaran Residual pada Skripsi Nyata. Karena perbaikan dilakukan dengan menggunakan 17 skripsi yang sama, nilai presisi pada iterasi kedua perlu ditafsirkan secara hati-hati karena berpotensi memberikan estimasi kinerja yang optimistis ketika diterapkan pada dokumen yang sepenuhnya independen.

Evaluasi pada Uji 1 menggunakan pasangan dokumen dan kode aturan (*document–rule pair*) sebagai unit analisis dengan penilaian biner, sedangkan pasangan berstatus TIDAK_BERLAKU dikeluarkan dari evaluasi. Untuk setiap pasangan yang dilaporkan oleh sistem, perangkat evaluasi menghasilkan kartu bukti yang memuat kode aturan, lokasi temuan berupa bagian, bab, seksi, dan kutipan teks, nomor halaman tercetak dan halaman PDF, nilai yang diharapkan beserta toleransinya, nilai yang ditemukan, tingkat atau sumber pewarisan properti yang disertai potongan XML, serta tangkapan halaman PDF hasil ekspor Microsoft Word dengan penanda lokasi temuan. Nomor halaman ditentukan melalui pencocokan kutipan temuan dengan teks pada setiap halaman PDF secara berurutan karena berkas .docx tidak menyimpan hasil paginasi akhir. Untuk setiap pasangan dokumen–aturan, sistem memilih paling banyak tiga kartu bukti secara acak menggunakan *seed* yang dicatat agar proses pemilihan dapat direproduksi. Dalam pelaksanaannya, ketua program studi yang tidak terlibat dalam pengembangan sistem memverifikasi satu kartu untuk setiap pasangan, yaitu kartu pertama yang terpilih secara acak. Pasangan dinyatakan benar apabila kartu yang diverifikasi terbukti sesuai dengan kondisi dokumen.

Presisi pada Uji 1 diperlakukan sebagai batas bawah berdasarkan prosedur verifikasi tersebut. Hal ini disebabkan pasangan yang sebenarnya mengandung pelanggaran, tetapi kartu pertamanya tidak tepat, tetap dihitung sebagai temuan yang salah. Selain kategori benar dan salah, verifikator dapat menetapkan status “tidak terlihat di cetakan” apabila klaim pada kartu tidak dapat dipastikan berdasarkan tampilan dokumen, misalnya ketika pemeriksaan berkaitan dengan penggunaan tab untuk membentuk indentasi atau perbedaan numerik yang terlalu kecil untuk dibedakan secara visual. Kartu dengan status tersebut dikeluarkan dari perhitungan presisi dan dilaporkan secara terpisah. Secara matematis, presisi Uji 1 dihitung sebagai rasio jumlah pasangan yang terverifikasi benar $|B|$ terhadap jumlah pasangan yang dilaporkan dan dinilai $|L|$, yaitu $\text{Presisi} = |B| / |L|$. Sementara itu, pelanggaran residual dihitung dengan membandingkan jumlah pasangan yang terverifikasi benar $|B|$ terhadap seluruh pasangan dokumen–aturan yang berlaku $|T|$, yaitu $\text{Pelanggaran Residual} = |B| / |T|$. Nilai pelanggaran residual merupakan batas bawah karena pelanggaran yang tidak dilaporkan oleh sistem tidak tercakup dalam pembilang. Mengingat seluruh dokumen telah melewati pemeriksaan dosen pembimbing dan dosen penguji, hasil ini menggambarkan pelanggaran format yang masih ditemukan pada skripsi yang telah dinyatakan lulus. Namun, Uji 1 tidak digunakan untuk menghitung tingkat deteksi pemeriksaan manual karena jumlah pelanggaran sebelum proses pemeriksaan manual dilakukan tidak diketahui.

Uji 2 dilaksanakan melalui injeksi kesalahan terkontrol (*controlled fault injection*) untuk mengevaluasi kemampuan sistem mendeteksi pelanggaran yang telah ditentukan sebelumnya. Pemilihan dokumen acuan didasarkan pada hasil verifikasi iterasi kedua dengan empat kriteria berurutan, yaitu jumlah pasangan deterministik yang terverifikasi benar paling sedikit, jumlah kemunculan temuan

paling sedikit apabila terdapat nilai yang sama, keberlakuan seluruh aturan yang diperlukan agar setiap operator mutasi dapat diterapkan, serta dominasi format langsung pada salah satu dokumen untuk menguji kontribusi *resolver* properti efektif. Berdasarkan kriteria tersebut, dipilih dua dokumen acuan. Dokumen D12 ditetapkan sebagai G1 karena memiliki empat pasangan terverifikasi benar dan 14 kemunculan temuan, jumlah paling sedikit di antara 17 dokumen yang diuji. Dokumen D16 ditetapkan sebagai G2 karena memiliki empat pasangan terverifikasi benar dan 102 kemunculan temuan; sekitar 81% teksnya menggunakan ukuran atau jenis huruf yang ditetapkan secara langsung pada elemen *run*.

Kedua dokumen acuan diperbaiki secara minimal menggunakan skrip, terbatas pada kemunculan yang dilaporkan oleh mesin pemeriksa. Proses tersebut menghasilkan 26 perubahan pada G1 dan 101 perubahan pada G2, dengan setiap perubahan dicatat secara individual. Modifikasi isi teks hanya dilakukan apabila diperlukan oleh aturan pemeriksaan, misalnya untuk memperbaiki nomor subbab atau nomor gambar yang digunakan secara berulang. Selanjutnya, kedua dokumen diperiksa secara visual melalui PDF hasil ekspor Microsoft Word dan disahkan oleh ketua program studi. Temuan yang masih dilaporkan sistem pada dokumen acuan yang telah disahkan dikategorikan sebagai positif palsu (*false positive*).

Pembentukan dokumen mutan dilakukan dengan menyisipkan satu kesalahan terkontrol melalui skrip pada tingkat Office Open XML (OOXML). Setiap aturan aktif memiliki satu operator mutasi sehingga terdapat 43 operator yang diterapkan pada kedua dokumen acuan. Pengujian mencakup tiga jenis variasi. Pertama, variasi mekanisme, yaitu injeksi kesalahan melalui perubahan definisi *style* dengan mengembalikan paragraf lain yang mewarisi *style* tersebut ke nilai sebelumnya, atau melalui perubahan format

langsung. Kedua, variasi besaran, yang mencakup mutan dengan penyimpangan yang jelas, mutan pada batas luar yang seharusnya terdeteksi, dan mutan pada batas dalam yang seharusnya tidak terdeteksi, sebagaimana dirangkum dalam Tabel 3. Ketiga, variasi proporsi, yaitu pengaturan jumlah paragraf yang dimutasi

sehingga tingkat kepatuhan akhir berada pada nilai ambang dikurangi 0,05, nilai ambang ditambah 0,05, atau mendekati 0,5. Rancangan ini memungkinkan evaluasi dilakukan terhadap kemampuan deteksi sistem pada berbagai mekanisme pemformatan, tingkat penyimpangan, dan proporsi ketidaksesuaian dokumen.

Tabel 3. Variasi Besaran terhadap Nilai Acuan

Aturan	Toleransi	Jauh	Batas Luar	Batas Dalam
HAL-01 Kertas	0,10 cm	Letter	+0,15 cm	+0,05 cm
HAL-02 Margin	0,05 cm	-0,50 cm	-0,08 cm	-0,03 cm
PAR-01 Spasi	0,05	Spasi 1,0	-0,08	-0,03
PAR-02 Indentasi	0,15 cm	-1,00 cm	-0,20 cm	-0,10 cm
FNT-02-04 Ukuran huruf	0 pt	-2 pt	-0,5 pt	—
ELM-03 Panjang abstrak	350 kata	+150 kata	+1 kata	Tepat 350 kata

Catatan: Toleransi dan nilai penyimpangan mengikuti parameter pada rancangan pengujian. Tanda “—” menunjukkan bahwa variasi batas dalam tidak diterapkan pada aturan tersebut.

Setiap dokumen mutan m memiliki himpunan aturan yang diharapkan terpicu, yang dinotasikan sebagai $E(m)$. Penetapan himpunan tersebut dilakukan sebelum sistem dijalankan, berdasarkan definisi masing-masing aturan dan struktur dokumen, sesuai dengan prinsip preregistrasi (Nosek et al., 2018). Jika memungkinkan, setiap kesalahan diisolasi agar tidak memicu aturan lain yang tidak menjadi sasaran pengujian. Sebagai contoh, ketika judul bab diubah, entri yang bersesuaian dalam daftar isi juga disesuaikan sehingga aturan konsistensi daftar isi tidak terpicu. Pada dokumen acuan yang telah memenuhi ketentuan dan mutan dengan penyimpangan batas dalam, himpunan aturan yang diharapkan terpicu bernilai kosong, yaitu $E(m) = \emptyset$. Sebelum evaluasi utama dilakukan, konsistensi generator mutan diuji menggunakan dokumen buatan yang telah memenuhi seluruh ketentuan. Setiap operator mutasi harus menghasilkan pemicu yang tepat sesuai dengan himpunan $E(m)$ yang telah ditentukan.

Evaluasi pada tingkat pasangan mutan dan kode aturan menggunakan empat kategori klasifikasi. *True positive*

(TP) diberikan apabila suatu aturan termasuk dalam $E(m)$ dan berhasil dilaporkan oleh sistem. *False negative* (FN) terjadi ketika aturan termasuk dalam $E(m)$, tetapi tidak dilaporkan. Sebaliknya, *false positive* (FP) ditetapkan apabila sistem melaporkan aturan yang tidak termasuk dalam $E(m)$, sedangkan *true negative* (TN) mencakup seluruh pasangan lainnya. Berdasarkan klasifikasi tersebut, metrik presisi, *recall*, skor F1, dan spesifisitas dihitung secara terpisah untuk kelompok aturan deterministik dan heuristik. Interval kepercayaan 95% untuk setiap skor dihitung menggunakan metode Wilson (Brown et al., 2001; Wilson, 1927).

Untuk menjawab pertanyaan penelitian kedua (RQ2), dilakukan analisis ablasi dengan menjalankan kembali Uji 2 menggunakan mode naif yang tidak menerapkan *resolver* properti efektif. Sementara itu, pertanyaan penelitian ketiga (RQ3) dikaji melalui analisis sensitivitas dengan menguji ambang kepatuhan pada rentang 0,5–1,0. Sensitivitas terhadap toleransi diperiksa menggunakan mutan batas luar dan batas dalam, yang masing-masing merepresentasikan penyimpangan tepat di luar dan tepat di dalam batas

toleransi. Nilai presisi dari Uji 1 dan *recall* dari Uji 2 tidak digabungkan untuk menghasilkan satu nilai F1 karena kedua metrik tersebut diperoleh dari kumpulan data dan prosedur evaluasi yang berbeda. Sebagai target kinerja indikatif yang ditetapkan dalam proposal penelitian, nilai presisi, *recall*, dan skor F1 untuk aturan deterministik diharapkan masing-masing mencapai sekurang-kurangnya 0,85.

HASIL DAN PEMBAHASAN

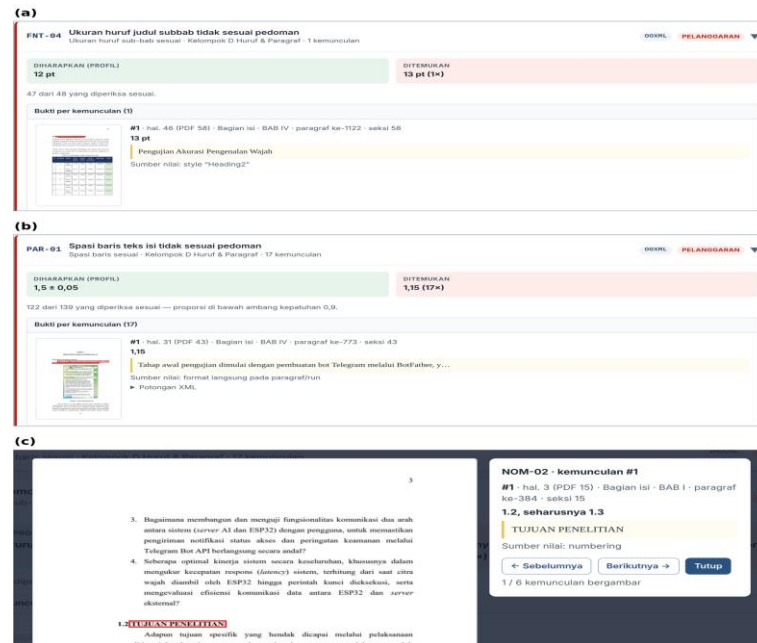
Implementasi Sistem dan Antarmuka

Mesin pemeriksa dikembangkan menggunakan Python sebagai paket independen dari kerangka kerja web, sehingga dapat dijalankan melalui antarmuka baris perintah maupun diintegrasikan ke aplikasi berbasis Django 5.2, PostgreSQL, dan Docker Compose. Sistem memeriksa dokumen .docx, sedangkan PDF hasil ekspor melalui Microsoft Word digunakan sebagai bukti visual, bukan sebagai dasar penilaian format. Hasil pemeriksaan mencakup skor kepatuhan berdasarkan aturan yang berlaku serta temuan berupa pelanggaran dan peringatan. Pada dokumen D17 yang terdiri atas 2.184 paragraf dan 99 seksi, pemeriksaan berlangsung selama 0,6 detik dan menghasilkan empat pelanggaran, satu peringatan, serta skor kepatuhan 88%. Gambar 2 dan Gambar 3 merupakan tangkapan langsung dari aplikasi dengan pemotongan gambar tanpa penyuntingan isi.

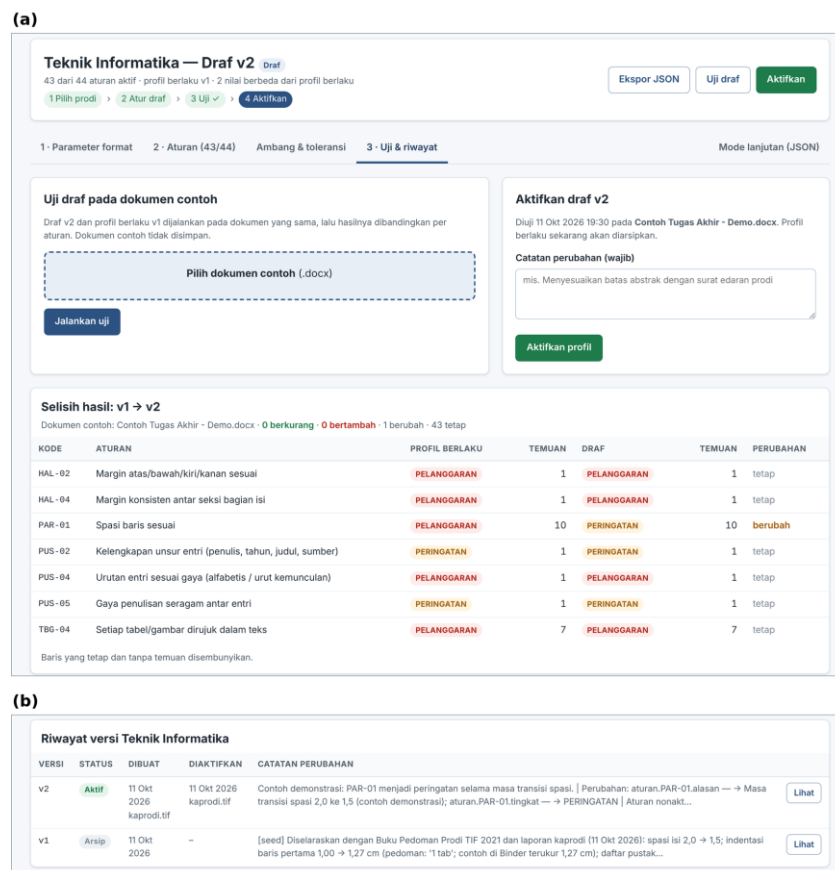
Laporan pemeriksaan disajikan dalam dua tingkat. Tingkat pertama merangkum aturan yang tidak terpenuhi,

nilai acuan beserta toleransinya, dan nilai aktual. Tingkat kedua merinci setiap kemunculan temuan melalui lokasi, kutipan teks, sumber properti, cuplikan XML, dan, jika tersedia, potongan halaman PDF dengan penanda merah. Karena .docx tidak menyimpan paginasi akhir, lokasi temuan pada PDF dipetakan melalui mekanisme pencocokan di lapisan aplikasi. Contoh pada Gambar 2 mencakup ukuran huruf judul subbab 13 pt yang diwarisi dari *style* Heading 2 (FNT-04), 17 paragraf dengan spasi 1,15 (PAR-01), serta penomoran subbab otomatis yang berulang (NOM-02). Temuan heuristik dipisahkan dalam bagian “Perlu diperiksa kembali” karena memerlukan verifikasi lebih lanjut.

Pengelolaan aturan dilakukan melalui tahapan penyusunan draf, pengujian, dan aktivasi profil konfigurasi berversi. Formulir parameter dibentuk berdasarkan metadata aturan agar konsisten dengan mesin pemeriksa. Administrator dapat menonaktifkan aturan dengan alasan tertulis, mengubah ambang dan toleransi, serta menurunkan aturan deterministik menjadi peringatan, tetapi tidak dapat meningkatkan aturan heuristik menjadi pelanggaran. Sebelum diaktifkan, profil draf diuji dengan membandingkan hasilnya terhadap profil yang berlaku pada dokumen contoh yang sama (Gambar 3a). Riwayat versi mencatat administrator, waktu, dan rincian perubahan (Gambar 3b). Profil yang dikunci oleh administrator universitas tidak dapat digantikan melalui mekanisme administrasi biasa, sehingga konsistensi konfigurasi dan pembekuan profil selama evaluasi penelitian dapat dipertahankan.



Gambar 2. Laporan dua tingkat pada D17: (a) FNT-04, satu kemunculan, nilai dari style, dengan potongan halaman PDF; (b) PAR-01, daftar kemunculan (1 dari 17 ditampilkan); (c) perbesaran halaman untuk NOM-02, kemunculan 1 dari 6.



Gambar 3. Pengelolaan profil konfigurasi: (a) pengujian profil draf pada dokumen contoh dan perbandingan hasilnya dengan profil yang sedang berlaku; (b) riwayat versi profil setelah diaktifkan.

Presisi dan Pelanggaran Residual pada Skripsi Nyata (RQ1 dan RQ4)

Mesin pemeriksa menyelesaikan proses analisis setiap dokumen dalam rentang waktu 0,5–1,6 detik. Pada iterasi kedua, sistem mengevaluasi 731 pasangan dokumen–aturan dan mengidentifikasi 107 pasangan yang dilaporkan, terdiri atas 92 pasangan aturan deterministik dan 15

pasangan aturan heuristik. Temuan tersebut mencakup 1.989 kemunculan pelanggaran atau ketidaksesuaian yang direpresentasikan melalui 244 kartu bukti. Perbandingan hasil antara iterasi pertama dan iterasi kedua disajikan pada Tabel 4 untuk menunjukkan perubahan kinerja sistem setelah proses perbaikan mesin dilakukan.

Tabel 4. Hasil Uji 1 pada Dua Iterasi Pembekuan (Interval Wilson 95%)

Iterasi	Pasangan Dilaporkan	Benar / Salah / Tidak Terlihat	Presisi Deterministik	Presisi Heuristik
Pertama (beku- uji-2026-10-11)	160	91 / 65 / 4	0,591 (78/132) [0,506–0,671]	0,542 (13/24) [0,351–0,721]
Kedua (beku- uji-2026-10-13b)	107	103 / 0 / 4	1,000 (90/90) [0,959–1,000]	1,000 (13/13) [0,772–1,000]

Catatan: Presisi dihitung berdasarkan jumlah pasangan yang terverifikasi benar dibandingkan dengan jumlah pasangan yang dinilai, dengan mengecualikan temuan berstatus “tidak terlihat”. Interval kepercayaan 95% dihitung menggunakan metode Wilson.

Pada iterasi pertama, nilai presisi aturan deterministik sebesar 0,591 masih jauh di bawah target. Kesalahan kartu bukti terutama berasal dari tiga penyebab, yaitu salah mengenali objek aturan, mengabaikan cara pemformatan alternatif yang secara visual tetap sesuai, serta perbedaan urutan XML dan tampilan dokumen hasil konversi PDF. Masalah urutan XML pada D15 menyebabkan 12 dari 17 pasangan dinilai salah. Setelah ketiga masalah tersebut diperbaiki, sebanyak 52 dari 65 pasangan salah tidak lagi dilaporkan, sedangkan 13 pasangan lainnya tetap muncul dengan kemunculan yang berbeda dan terverifikasi benar. Namun, dua pasangan yang sebelumnya benar, yaitu D13 HAL-02 dan D15 PUS-02, ikut hilang, sementara satu pasangan benar baru muncul pada D09 TBG-03.

Pada iterasi kedua, ketua program studi memverifikasi 103 kartu yang dapat dinilai dari tampilan cetak dan menyatakan seluruhnya benar. Empat kartu lainnya tidak dapat dipastikan secara visual, meliputi temuan margin bawah D06 dengan selisih kurang dari satu baris serta indentasi berbasis tab pada D01 dan D05 yang

tampak serupa dengan indentasi baris pertama. Temuan semacam ini memerlukan pemeriksaan langsung terhadap berkas. Karena perbaikan mesin menggunakan korpus 17 skripsi yang sama, nilai presisi sebesar 1,000 berpotensi terlalu optimistis untuk dokumen di luar korpus. Kemampuan sistem pada data dengan kebenaran yang ditetapkan secara independen dievaluasi melalui Uji 2 pada subbab Akurasi pada Injeksi Kesalahan.

Pelanggaran residual mencakup 90 dari 567 pasangan deterministik yang berlaku (0,159; IK 95%: 0,131–0,191) dan 13 dari 153 pasangan heuristik (0,085; IK 95%: 0,050–0,140). Nilai tersebut merupakan batas bawah karena pelanggaran yang tidak terdeteksi sistem tidak diperhitungkan. Pada kelompok deterministik, proporsi pelanggaran residual tertinggi ditemukan pada tabel dan gambar (0,312; 20/64), penomoran (0,224; 19/85), daftar pustaka (0,216; 11/51), serta huruf dan paragraf (0,193; 23/119). Sementara itu, halaman sampul dan lembar pengesahan tidak menunjukkan pelanggaran deterministik.

Salah satu kartu yang terverifikasi benar menunjukkan bahwa judul subbab “2.6 ARDUINO IDE” pada D03 menggunakan ukuran huruf 11 pt. Properti tersebut tidak ditetapkan secara langsung pada paragraf maupun *style*, melainkan diwarisi dari nilai bawaan dokumen (w:docDefaults). Temuan ini memperlihatkan bahwa pustaka yang hanya membaca pemformatan langsung berpotensi gagal memperoleh nilai format yang sebenarnya (python-openxml, n.d.).

Tabel 5 menyajikan hasil Uji 1 untuk seluruh 44 aturan, diurutkan berdasarkan jumlah dokumen yang terindikasi melanggar. Gambar 4 menampilkan distribusi pelanggaran yang sama pada setiap dokumen. Kolom “Melanggar” menunjukkan rasio jumlah dokumen dengan pasangan yang terverifikasi benar terhadap jumlah dokumen tempat aturan tersebut berlaku, sedangkan identitas dokumen yang mengalami pelanggaran ditunjukkan pada Gambar 4.

Tabel 5. Hasil Uji 1 per Aturan pada Iterasi Kedua, Diurutkan dari yang Paling Banyak Dilanggar

No.	Kode	Aturan	Ketentuan	Melanggar
1	TBG-04	Rujukan tabel dan gambar	Disebut dalam narasi	14/17
2	PAR-02	Indentasi baris pertama	1,27 cm $\pm$ 0,15 cm	13/17
3	NOM-04	Nomor halaman bagian awal	Romawi kecil, tengah bawah	9/17
4	PUS-04	Urutan entri pustaka	Alfabetis	9/17
5	NOM-02	Nomor subbab	1.1.1, berurutan per bab	8/17
6	PGS-03	Kolom tanda tangan pengesahan (H)	Kaprodi, pembimbing, NIDN	5/17
7	TBG-03	Nomor tabel dan gambar	Per bab, berurutan	5/17
8	HAL-02	Margin halaman	4/3/4/3 cm $\pm$ 0,05 cm	4/14
9	FNT-03	Ukuran huruf judul bab	14 pt	4/17
10	FNT-04	Ukuran huruf judul subbab	12 pt	4/17
11	STR-01	Kelengkapan bab wajib	Lima bab, dengan alias	4/17
12	PUS-05	Keseragaman gaya entri pustaka (H)	Penulis (Tahun), tanpa nomor	4/17
13	HAL-04	Konsistensi margin antarseksi	Sama di semua seksi isi	3/14
14	STR-05	Konsistensi judul bab	Daftar isi sama dengan isi	3/17
15	PUS-02	Kelengkapan unsur entri pustaka (H)	Penulis, tahun, judul, sumber	3/17
16	PAR-01	Spasi baris teks isi	1,5 $\pm$ 0,05	2/17
17	NOM-05	Nomor halaman pertama bab	Tengah bawah	2/17
18	PUS-03	Indentasi menggantung pustaka	1,27 cm	2/17
19	PGS-02	Kelengkapan lembar pengesahan (H)	Judul, NIM, frasa wajib	1/17
20	HAL-01	Ukuran kertas	A4 $\pm$ 0,10 cm	1/17
21	TBG-01	Letak judul tabel	Di atas tabel	1/17

22	ELM-01	Kelengkapan elemen wajib	Sampul sampai daftar pustaka	1/17
23	ELM-02	Urutan elemen	Sesuai urutan pedoman	1/17
24	PAR-04	Cara membuat indentasi (H)	Pengaturan paragraf, bukan tab	0/17
25	SMP-01	Halaman sampul (H)	Halaman pertama berunsur sampul	0/17
26	SMP-02	Judul pada sampul	Kapital, rata tengah	0/17
27	SMP-03	Logo institusi pada sampul (H)	Ada	0/17
28	SMP-04	Nama dan NIM pada sampul (H)	Ada	0/17
29	SMP-05	Identitas institusi pada sampul (H)	Prodi, fakultas, universitas, tahun	0/17
30	PGS-01	Lembar pengesahan	Ada	0/17
31	HAL-03	Orientasi halaman	Tegak	0/17
32	FNT-01	Jenis huruf teks isi	Times New Roman	0/17
33	FNT-02	Ukuran huruf teks isi	12 pt	0/17
34	PAR-03	Perataan teks isi	Rata kiri-kanan	0/17
35	NOM-01	Nomor bab	Angka Romawi	0/17
36	NOM-03	Nomor halaman bagian isi	Angka Arab, kanan atas	0/17
37	STR-02	Urutan bab	Sesuai urutan bab wajib	0/17
38	STR-03	Format judul bab	Kapital, tebal, rata tengah	0/17
39	STR-04	Kedalaman <i>heading</i>	Paling banyak 4 tingkat	0/17
40	TBG-02	Letak judul gambar	Di bawah gambar	0/13
41	ELM-03	Panjang abstrak	Paling banyak 350 kata	0/16
42	ELM-04	Daftar tabel dan daftar gambar	Ada bila terdapat tabel/gambar	0/17
43	PUS-01	Daftar pustaka	Ada	0/17
44	ELM-05	Abstrak paling banyak satu halaman (H)	Nonaktif (lihat ELM-03)	—

Catatan: (H) menandai aturan heuristik. Kolom Melanggar menunjukkan jumlah dokumen dengan pelanggaran yang terverifikasi dibandingkan dengan jumlah dokumen tempat aturan berlaku. Tanda “—” menunjukkan aturan tidak diaktifkan dalam evaluasi.

Keterangan: (H) menunjukkan aturan heuristik. Kartu yang tidak terlihat di cetakan tidak dihitung sebagai pelanggaran, yaitu D06 pada HAL-02 dan HAL-04 serta D01 dan D05 pada PAR-04. Penyebut kurang dari 17 menunjukkan aturan tidak berlaku pada sebagian dokumen. Margin HAL-02 ditulis dalam urutan atas/bawah/kiri/kanan. ELM-05 dinonaktifkan karena batas abstrak diperiksa melalui jumlah kata pada ELM-

03. Distribusi dokumen yang melanggar disajikan pada Gambar 4.

Lima aturan paling sering dilanggar mencerminkan pola ketidakpatuhan yang konsisten. TBG-04 dilanggar pada 14 dari 17 dokumen karena tabel atau gambar tidak dirujuk dalam narasi. PAR-02 dilanggar pada 13 dokumen, umumnya karena indentasi 1,00 cm, lebih kecil dari ketentuan 1,27 cm. NOM-04 dan PUS-04 masing-masing dilanggar pada 9 dokumen,

terkait penomoran halaman awal dan urutan alfabetis pustaka. NOM-02 dilanggar pada 8 dokumen akibat nomor subbab yang melompat atau berulang. Setiap dokumen melanggar 4–10 aturan, dengan median 6; D06 memiliki pelanggaran terbanyak (10), sedangkan D01, D07, D11, dan D16 paling sedikit (4).

Sebanyak 19 aturan tidak pernah dilaporkan pada korpus ini (Tabel 5, nomor

25–43). Aturan tersebut mencakup orientasi halaman, format teks dan judul bab, penomoran, serta keberadaan elemen wajib. Kepatuhan ini kemungkinan terbantu oleh penggunaan templat mahasiswa. Namun, kemampuan sistem mendeteksi pelanggaran pada aturan-aturan tersebut baru dapat dinilai melalui Uji 2.



Gambar 4. Matriks pelanggaran Uji 1 per dokumen dan aturan pada iterasi kedua.

Gambar 4 menyajikan matriks pelanggaran pada Uji 1 iterasi kedua untuk 24 aturan yang setidaknya pernah dilaporkan. Angka di sisi kanan menunjukkan jumlah dokumen yang melanggar dibandingkan dengan jumlah dokumen tempat aturan berlaku, sedangkan baris paling bawah menunjukkan jumlah aturan yang dilanggar pada setiap dokumen.

Akurasi pada Injeksi Kesalahan (RQ1)

Generator menghasilkan 164 mutan, terdiri atas 79 mutan dari G1 dan 85 mutan dari G2, serta dua dokumen acuan sebagai kontrol. Seluruh 43 aturan aktif menjadi sasaran pada setidaknya satu mutan, termasuk 19 aturan yang tidak pernah dilaporkan pada Uji 1. Sebanyak 18 varian tidak dapat diterapkan, terutama varian lewat *style* untuk aturan paragraf, karena

kedua dokumen acuan mengatur spasi, indentasi, dan perataan paragraf secara

langsung, bukan melalui *style*. Hasil pengujian tersebut disajikan pada Tabel 6.

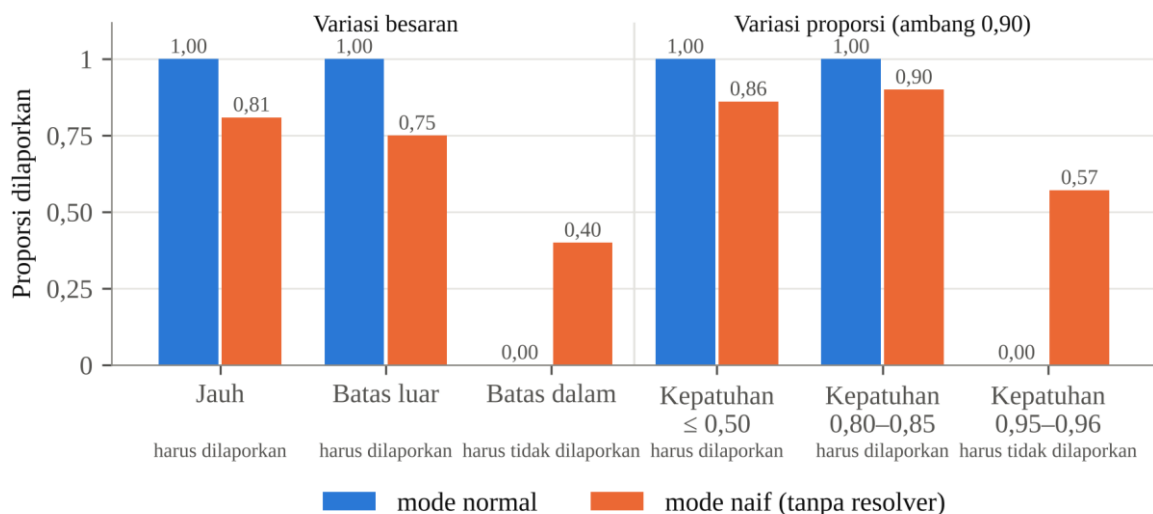
Tabel 6. Kinerja Uji 2 (Interval Wilson 95%)

Golongan	Presisi	Recall	F1	Spesifisitas
Deterministik	0,985 [0,946–0,996]	0,985 [0,946–0,996]	0,985	1,000 [0,999–1,000]
Heuristik	0,926 [0,766–0,979]	0,962 [0,811–0,993]	0,943	0,999 [0,995–1,000]

Catatan: Interval kepercayaan 95% menggunakan metode Wilson. Nilai F1 disajikan sebagai skor tunggal.

Aturan deterministik menghasilkan 129 TP, 2 FN, 2 FP, dan 5.506 TN, sedangkan aturan heuristik menghasilkan 25 TP, 1 FN, 2 FP, dan 1.457 TN. Ketiga metrik deterministik memenuhi target indikatif $\geq 0,85$, dan kedua dokumen acuan tidak menghasilkan temuan. Seluruh mutan

jauh (21/21) dan batas luar (16/16) terdeteksi, sedangkan mutan batas dalam (0/10) dan mutan dengan kepatuhan 0,95–0,96 di atas ambang (0/14) tidak dilaporkan (Gambar 5). Hasil ini menunjukkan bahwa toleransi dan ambang pada profil berfungsi sebagai batas keputusan sistem.



Gambar 5. Proporsi pasangan sasaran yang dilaporkan berdasarkan variasi besaran dan proporsi pada mode normal dan naif.

Tujuh pasangan salah tetap dihitung karena himpunan harapan tidak diubah setelah pengujian. Dua FN pada SMP-01 dan ELM-01 terjadi ketika mutan menghapus halaman sampul G1, tetapi D12 ternyata memiliki dua sampul identik sehingga satu sampul masih tersisa. Kesalahan ini berasal dari asumsi operator, bukan sistem. Satu FN pada PAR-02 terjadi karena generator memperkirakan kepatuhan 0,894, di bawah ambang 0,9,

sedangkan mesin menghitung posisi tab dan memperoleh 0,948.

Empat FP merupakan kesalahan sistem yang nyata. Pada G1, satu entri pustaka yang dipisahkan menjadi dua paragraf digabungkan kembali berdasarkan indentasi. Setelah indentasi menggantung dihapus, baris kedua terdeteksi sebagai entri baru tanpa tahun (PUS-02) dan tidak alfabetis (PUS-04). Temuan ini menunjukkan bahwa pemisahan entri pustaka masih bergantung pada tata letak,

sejalan dengan temuan bahwa identifikasi dan penguraian entri pustaka merupakan tantangan tersendiri (Lei et al., 2016; Tkaczyk et al., 2018) dan satu kesalahan format dapat memicu pelanggaran aturan lain yang tidak berkaitan secara langsung.

Tabel 7. Recall Aturan Sasaran pada Mode Normal dan Mode Naif per Mekanisme Mutasi

Mekanisme	Normal	Naif	Selisih
Lewat format langsung	1,000 (66/66)	0,773 (51/66)	-0,227
Lewat <i>style</i>	1,000 (9/9)	1,000 (9/9)	0,000

Tanpa *resolver*, 15 dari 66 mutan format langsung tidak terdeteksi, meskipun seluruh mutan berbasis *style* berhasil dikenali. Penurunan *recall* terutama terjadi pada aturan jenis dan ukuran huruf serta format judul. Spesifisitas deterministik juga turun dari 1,000 menjadi 0,693 [0,680–0,705], sedangkan presisi merosot menjadi 0,063 karena dokumen yang sesuai ikut dinyatakan melanggar. Selain itu, mutan yang seharusnya lolos tetap dilaporkan pada 4 dari 10 kasus batas dalam dan 8 dari 14 kasus di atas ambang (Gambar 5). Temuan ini menunjukkan bahwa pemeriksaan berbasis *style* saja tidak memadai karena proporsi format huruf yang diatur langsung pada *run* berkisar 4–100% (median 36%), sementara spasi dan

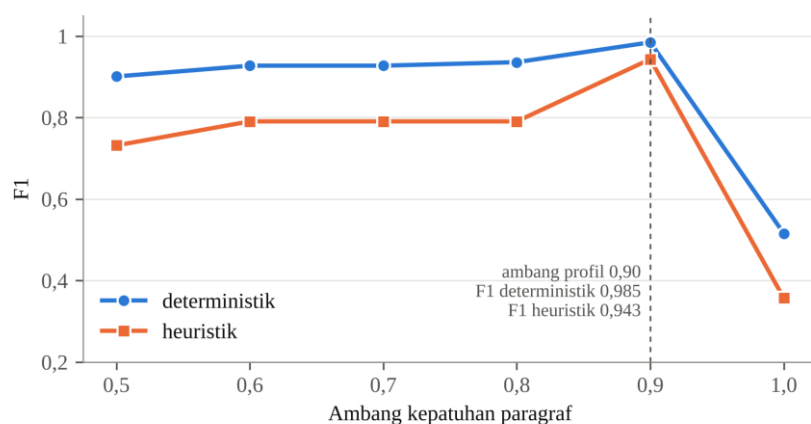
Kontribusi Resolver (RQ2)

Tabel 7 membandingkan *recall* mode normal dengan mode naif, yang hanya membaca properti dari definisi *style*, untuk menilai kontribusi *resolver* dalam mendeteksi pelanggaran format.

indentasi pada kedua dokumen acuan hampir seluruhnya diatur secara langsung.

Sensitivitas Ambang (RQ3)

Gambar 6 menunjukkan bahwa F1 deterministik tertinggi, yaitu 0,985, diperoleh pada ambang kepatuhan paragraf 0,9. Pada ambang 0,5–0,8, F1 berkisar 0,901–0,936, sedangkan pada ambang 1,0 turun menjadi 0,516 akibat meningkatnya *false positive* hingga 243 kasus dan presisi menjadi 0,349. Karena himpunan harapan tetap mengacu pada ambang profil, hasil ini menunjukkan dampak perubahan ambang, bukan penentuan ambang optimal berdasarkan data uji. Sensitivitas terhadap toleransi dievaluasi melalui mutan batas luar dan batas dalam pada subbab Akurasi pada Injeksi Kesalahan.



Gambar 6. Skor F1 berdasarkan variasi ambang kepatuhan paragraf 0,5–1,0, dengan ambang profil 0,9 ditandai sebagai acuan.

Perbandingan dan Keterbatasan

Penelitian ini berbeda dari studi terdahulu yang berfokus pada penilaian

pakar, akurasi agregat, atau pengujian oleh pengembang (Aulia & Haviana, 2026; Azzahra et al., 2025; Ramadhan, 2025;

Svintsytska et al., 2025) karena menyediakan evaluasi yang dapat ditelusuri melalui kartu bukti setiap temuan dan himpunan harapan mutan yang ditetapkan sebelum pengujian. Pendekatan ini menunjukkan peningkatan presisi dari 0,591 pada iterasi pertama menjadi 1,000 pada iterasi kedua setelah perbaikan berdasarkan temuan yang terverifikasi. Evaluasi juga mengungkap tantangan dokumen hasil konversi PDF yang memiliki perbedaan antara struktur OOXML dan tampilan visual, serta pentingnya membedakan kepatuhan visual dan struktural.

Penelitian ini memiliki beberapa keterbatasan. Pertama, penggunaan 17 skripsi yang sama untuk perbaikan sistem dan evaluasi Uji 1 berpotensi menghasilkan estimasi presisi yang optimistis. Kedua, verifikasi hanya melibatkan satu penilai dan satu kartu untuk setiap pasangan dokumen–aturan, sehingga kesepakatan antarpemilai belum dapat diukur, padahal penilaian manusia dapat bervariasi antarpemilai (Voorhees, 2000). Ketiga, Uji 2 menggunakan mutan terstruktur dari hanya dua dokumen acuan yang belum sepenuhnya merepresentasikan kesalahan nyata. Keempat, *recall* pada skripsi nyata belum dapat dihitung karena jumlah pelanggaran yang terlewat dalam pemeriksaan manual tidak diketahui. Selain itu, terdapat temuan benar pada iterasi pertama yang tidak lagi terdeteksi pada iterasi kedua. Terakhir, evaluasi terbatas pada satu program studi dengan profil aturan yang mengacu pada pedoman dan templat setempat. Oleh karena itu, penelitian selanjutnya perlu menggunakan korpus skripsi baru yang lebih beragam, melibatkan beberapa penilai, serta menguji sistem pada program studi lain untuk menilai generalisasi dan ketahanannya terhadap variasi format dokumen.

KESIMPULAN

Penelitian ini menghasilkan mesin pemeriksa format skripsi berbasis aturan dengan *resolver* properti efektif yang

dievaluasi pada Program Studi Teknik Informatika UMAHA. Setelah satu siklus perbaikan terverifikasi, aturan deterministik mencapai presisi 1,000 pada 17 skripsi nyata serta presisi, *recall*, dan F1 sebesar 0,985 pada 164 mutan, sedangkan aturan heuristik mencapai F1 0,943 (RQ1). Tanpa *resolver*, *recall* mutan format langsung turun dari 1,000 menjadi 0,773 dan spesifisitas dari 1,000 menjadi 0,693, yang menegaskan pentingnya pembacaan properti efektif (RQ2). F1 deterministik tertinggi dicapai pada ambang 0,9, sedangkan perubahan ambang menghasilkan penurunan kinerja (RQ3). Pelanggaran residual sebagai batas bawah mencapai 15,9% pada pasangan aturan deterministik, terutama pada rujukan tabel dan gambar dalam narasi serta indentasi baris pertama (RQ4). Namun, penggunaan korpus yang sama untuk perbaikan dan evaluasi, pengujian mutan yang terbatas pada dua dokumen acuan, penanganan dokumen hasil konversi PDF, dan cakupan satu program studi membatasi generalisasi hasil. Penelitian selanjutnya perlu menguji skripsi di luar korpus, mengembangkan profil lintas program studi, meningkatkan penanganan dokumen hasil konversi PDF, dan memperbaiki pemisahan entri pustaka agar tidak bergantung pada tata letak.

DAFTAR PUSTAKA

- Andrews, J. H., Briand, L. C., & Labiche, Y. (2005). Is mutation an appropriate tool for testing experiments? In *Proceedings of the 27th International Conference on Software Engineering* (pp. 402–411). <https://doi.org/10.1145/1062455.1062530>
- Arocena, P. C., Glavic, B., Mecca, G., Miller, R. J., Papotti, P., & Santoro, D. (2015). Messing up with BART: Error generation for evaluating data-cleaning algorithms. *Proceedings of the VLDB Endowment*, 9(2), 36–47.
- Aulia, D. S., & Haviana, S. F. C. (2026). Sistem pengecekan format tugas akhir menggunakan rule-based

- system dan similarity judul. *Jurnal Rekayasa Sistem Informasi dan Teknologi*, 3(4), 685–696. <https://doi.org/10.70248/jrsit.v3i4.3822>
- Azzahra, F., Anggraeni, D., & Hayat, M. A. (2025). Penerapan natural language processing dan regular expressions dalam validasi artikel ilmiah. *Jurnal Muhammadiyah's Applied Technology*, 4(3), 335–342. <https://doi.org/10.26618/dkjh7933>
- Barr, E. T., Harman, M., McMin, P., Shahbaz, M., & Yoo, S. (2015). The oracle problem in software testing: A survey. *IEEE Transactions on Software Engineering*, 41(5), 507–525. <https://doi.org/10.1109/TSE.2014.2372785>
- Binmakhashen, G. M., & Mahmoud, S. A. (2019). Document layout analysis: A comprehensive survey. *ACM Computing Surveys*, 52(6), 1–36. <https://doi.org/10.1145/3355610>
- Bloch, L., Rückert, J., & Friedrich, C. M. (2025). Towards automatic formal feedback on scientific documents. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)* (pp. 334–344). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2025.bea-1.26>
- Brown, L. D., Cai, T. T., & DasGupta, A. (2001). Interval estimation for a binomial proportion. *Statistical Science*, 16(2), 101–133. <https://doi.org/10.1214/ss/1009213286>
- DeMillo, R. A., Lipton, R. J., & Sayward, F. G. (1978). Hints on test data selection: Help for the practicing programmer. *Computer*, 11(4), 34–41. <https://doi.org/10.1109/C-M.1978.218136>
- Ecma International. (2016). *ECMA-376: Office Open XML file formats—Part 1: Fundamentals and markup language reference* (5th ed.).
- Fröhlich, T., & Schlippe, T. (2025). *RubiSCoT: A framework for AI-supported academic assessment* (arXiv:2510.17309). arXiv. <https://doi.org/10.48550/arXiv.2510.17309>
- Hlukhov, V. S., & Sydorko, D. S. (2023). Algorithms and software for verification of scientific and technical text documents. *Applied Aspects of Information Technology*, 6(3), 304–317. <https://doi.org/10.15276/aait.06.2023.21>
- Jia, Y., & Harman, M. (2011). An analysis and survey of the development of mutation testing. *IEEE Transactions on Software Engineering*, 37(5), 649–678. <https://doi.org/10.1109/TSE.2010.62>
- Just, R., Jalali, D., Inozemtseva, L., Ernst, M. D., Holmes, R., & Fraser, G. (2014). Are mutants a valid substitute for real faults in software testing? In *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering* (pp. 654–665). <https://doi.org/10.1145/2635868.2635929>
- Kapoor, S., & Narayanan, A. (2023). Leakage and the reproducibility crisis in machine-learning-based science. *Patterns*, 4(9), Article 100804. <https://doi.org/10.1016/j.patter.2023.100804>
- Knight, S., Shibani, A., Abel, S., Gibson, A., Ryan, P., Sutton, N., Wight, R., Lucas, C., Sándor, Á., Kitto, K., Liu, M., Mogarkar, R. V., & Buckingham Shum, S. (2020). AcaWriter: A learning analytics tool for formative feedback on academic writing. *Journal of Writing Research*, 12(1), 141–186. <https://doi.org/10.17239/jowr-2020.12.01.06>

- Lei, Y., Tian, Y., Li, N., Liang, Q., & Zhao, W. (2016). Identification methods for bibliographic reference entries in editable scientific and technical documents. In *Proceedings of the 2016 International Conference on Computer Science and Electronic Technology (CSET 2016)* (pp. 172–178). Atlantis Press. <https://doi.org/10.2991/cset-16.2016.42>
- Martsinkevich, V., Tereshchenko, V., Larionova, G., Berezhkov, A., Kobets, E., Nasyrov, N., & Gorlushkina, N. (2023). Web tool for automated document formatting verification. In *Proceedings of the 33rd Conference of Open Innovations Association FRUCT* (pp. 375–381). FRUCT Oy. <https://fruct.org/publications/volume-33/acm33/files/Mar.pdf>
- Nosek, B. A., Ebersole, C. R., DeHaven, A. C., & Mellor, D. T. (2018). The preregistration revolution. *Proceedings of the National Academy of Sciences*, 115(11), 2600–2606. <https://doi.org/10.1073/pnas.1708274114>
- Papadakis, M., Kintis, M., Zhang, J., Jia, Y., Le Traon, Y., & Harman, M. (2019). Mutation testing advances: An analysis and survey. *Advances in Computers*, 112, 275–378. <https://doi.org/10.1016/bs.adcom.2018.03.015>
- Program Studi Teknik Informatika UMAHA. (2021). *Buku pedoman proposal dan tugas akhir*. Universitas Maarif Hasyim Latif.
- Putra, A. D., Pambudhi, T., Muhsom, & Afwan, B. (2026). Pemetaan tipologi kesalahan teknis dalam naskah skripsi mahasiswa program studi Pendidikan Guru Sekolah Dasar (PGSD). *Paedagogie*, 21(1), 1463–1471. <https://doi.org/10.31603/paedagogie.v21i1.16306>
- python-xml. (n.d.). *Text-related objects*. python-docx documentation. Retrieved October 11, 2026, from <https://python-docx.readthedocs.io/en/latest/api/text.html>
- Ramadhan, H. F. (2025). Development and optimization of an automated thesis format checking system for undergraduate students at Universitas Negeri Makassar. *Journal of Embedded Systems Security and Intelligent Systems*, 6(2). <https://doi.org/10.59562/jessi.v6i2.9211>
- Solihin, W., & Eastman, C. (2015). Classification of rules for automated BIM rule checking development. *Automation in Construction*, 53, 69–82. <https://doi.org/10.1016/j.autcon.2015.03.003>
- Svintsytska, O., Korotun, O., Marchuk, D., & Zhelizko, V. (2025). Algorithms for automated verification of academic document parameters. *Infocommunication and Computer Technologies*, 2(10), 119–127. <https://doi.org/10.36994/2788-5518-2025-02-10-15>
- Sydorko, D., & Hlukhov, V. S. (2024). Program implementation of methods for analysis and verification of technical reports. *Computing Systems and Networks*, 6(2), 208–221. <https://doi.org/10.23939/csn2024.02.208>
- Tkaczyk, D., Collins, A., Sheridan, P., & Beel, J. (2018). Machine learning vs. rules and out-of-the-box vs. retrained: An evaluation of open-source bibliographic reference and citation parsers. In *Proceedings of the 18th ACM/IEEE Joint Conference on Digital Libraries* (pp. 99–108). <https://doi.org/10.1145/3197026.3197048>
- Verbovyi, D. S., & Saiapina, I. O. (2025). Method for automatic document verification for compliance with

- regulatory requirements. *Systems and Technologies*, 3(158), 181–187.
<https://doi.org/10.34185/1562-9945-3-158-2025-18>
- Voorhees, E. M. (2000). Variations in relevance judgments and the measurement of retrieval effectiveness. *Information Processing & Management*, 36(5), 697–716.
[https://doi.org/10.1016/S0306-4573\(00\)00010-8](https://doi.org/10.1016/S0306-4573(00)00010-8)
- Wilson, E. B. (1927). Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158), 209–212.
<https://doi.org/10.1080/01621459.1927.10502953>
- Zhang, J., Sun, Q., Wang, Q., Zhang, W., Xiao, Z., Zhang, X., Ren, J., & Liu, J. (2025). *PEMUTA: Pedagogically-enriched multi-granular undergraduate thesis assessment* (arXiv:2507.19556). arXiv.
<https://doi.org/10.48550/arXiv.2507.19556>