LAPORAN PENELITIAN INTERNAL

Pengembangan Sistem Evaluasi Otomatis Berbasis Ekstraksi OCR dan Pencocokan Pola (*Pattern Matching*) untuk Penilaian Latihan Pemrograman Web

Disusun oleh:

Ketua

Ir. Yohanes Sugiarto, Dipl.-Ing., M.Sc. NUPTK. 1258757658130133

PROGRAM STUDI TEKNIK MEKATRONIKA
POLITEKNIK ATMI SURAKARTA
SURAKARTA
2026

LEMBAR PENGESAHAN

1.	Judul Penelitian	:	Pengembangan Sistem Evaluasi Otomatis Berbasis Ekstraksi OCR dan Pencocokan Pola (*Pattern Matching*) untuk Penilaian Latihan Pemrograman Web
2.	Fokus Penelitian	:	Digitalisasi
3.	Tingkat Kesiapterapan Teknologi (TKT)	:	4
4.	Pelaksana Kegiatan		
	Ketua Peneliti		
	• Nama	:	Ir. Yohanes Sugiarto, Dipl.-Ing., M.Sc.
	• NUPTK	:	1258757658130133
	• Jabatan Fungsional	:	Asisten Ahli
	• Program Studi	:	Teknik Mekatronika
	• Email	:	ysugiarto@atmi.ac.id
	Anggota Peneliti		Tidak ada – penelitian dilakukan oleh peneliti tunggal
5.	Durasi Kegiatan	:	1 tahun
6.	Pendanaan		
	• Politeknik	:	-
	• Industri / UMKM	:	-

Surakarta, 24 Juli 2026

Mengetahui
Ketua Program Studi

Ir. Hilarius Prin Pujianto, S.T., M.Eng.
NUPTK. 5446767668130242

Ketua Peneliti

Ir. Yohanes Sugiarto, Dipl.-Ing., M.Sc.
NUPTK. 1258757658130133

Menyetujui
Ketua LPPM

POLITEKNIK ATMI SURAKARTA LPPM

Ir. Adhi Setya Hutama, S.T., M.Sc.
NUPTK. 5340765666130283

ABSTRAK

Penilaian latihan pemrograman web mahasiswa yang dikumpulkan melalui Moodle secara manual menghadapi beban kerja besar, mengingat variasi format submission (HTML/CSS/JavaScript langsung, arsip ZIP/RAR, tautan GitHub, atau dokumen PDF/DOCX berisi cuplikan layar kode) serta risiko inkonsistensi penilaian. Penelitian ini mengembangkan dan mengevaluasi Lab3Evaluator, sistem evaluasi otomatis berbasis Python yang menilai submission HTML, CSS, Bootstrap, JavaScript, PHP, dan Laravel pada skala 0-100 secara konsisten. Metode meliputi pipeline ekstraksi konten multi-format — pembacaan langsung HTML/ZIP/RAR, penelusuran tautan GitHub, serta OCR (PyMuPDF dan Tesseract) untuk submission PDF/DOCX — diikuti penilaian berbasis pencocokan pola (regex) terhadap kriteria berjenjang, dengan tiga mode pencocokan (seluruh pola wajib cocok, satu pola cukup, atau minimal N pola) untuk mengakomodasi variasi gaya kode. Hasil disajikan sebagai CSV, Excel, JSON, dan situs web inkremental. Sistem diuji pada 13 modul evaluator satu semester kurikulum Program Studi Teknologi Rekayasa Mekatronika, dari Git dasar hingga autentikasi Laravel Breeze, pada dataset kelas Latihan_TRMK2026 (53 mahasiswa). Hasil menunjukkan rata-rata nilai keseluruhan 85,4 dari 100 (92,0 dari submission saja), variasi rata-rata antar-modul 62,7-97,9, dan tingkat submission rata-rata 93,0%. Modul dengan pencocokan pola toleran konsisten menghasilkan nilai lebih tinggi dan lebih tahan terhadap variasi implementasi mahasiswa. Penelitian ini menyimpulkan bahwa kombinasi OCR dan pencocokan pola berjenjang layak untuk otomatisasi penilaian tugas pemrograman web skala kelas, dengan syarat kriteria regex dirancang toleran terhadap variasi format dan kesalahan OCR.

Kata kunci: Evaluasi Otomatis, Computer-Aided Assessment, Optical Character Recognition, Pencocokan Pola, Regular Expression, Moodle, Rekayasa Perangkat Lunak

Manual grading of web-programming lab submissions collected through Moodle poses a substantial workload problem, given varied submission formats (direct HTML/CSS/JavaScript files, ZIP/RAR archives, GitHub links, or PDF/DOCX documents with code screenshots) and the risk of grading inconsistency. This study develops and evaluates Lab3Evaluator, a Python-based system that scores web-programming submissions (HTML, CSS, Bootstrap, JavaScript, PHP, and Laravel) on a 0-100 scale. The method comprises a multi-format content-extraction pipeline — direct HTML/ZIP/RAR reading, GitHub link resolution, and OCR (PyMuPDF and Tesseract) for PDF/DOCX — followed by a regex pattern-matching scoring mechanism against tiered per-step criteria, with three matching modes (all-required, any-one, or minimum-N patterns) to accommodate coding-style variation. Results are reported as CSV, Excel, JSON, and an incremental static website. The system was tested across 13 evaluator modules covering a full semester's curriculum for the Mechatronics Engineering Technology program, from basic Git to Laravel Breeze authentication, on the Latihan_TRMK2026 dataset (53 students). Results show an overall average score of 85.4 out of 100 (92.0 from actual submissions), per-module averages ranging from 62.7 to 97.9, and a 93.0% average submission rate. Modules using more tolerant matching modes consistently produced higher scores and were more robust to coding-style variation. The study concludes that combining OCR with tiered pattern matching is a viable approach to automating class-scale web-programming assessment, provided the regex criteria tolerate format variation and OCR error.

Keywords: *Automated Assessment, Computer-Aided Assessment, Optical Character Recognition, Pattern Matching, Regular Expression, Moodle, Software Engineering*

Jl. Mojo No. 1, Karangasem, Laweyan, Surakarta 57145. Phone : +62 271 714466
E-mail: politeknik@atmi.ac.id • Website: http://www.atmi.ac.id
Jawa Tengah, Indonesia.

PRAKATA

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas rahmat dan karunia-Nya sehingga penelitian mengenai Pengembangan Sistem Evaluasi Otomatis Berbasis Ekstraksi OCR dan Pencocokan Pola (*Pattern Matching*) untuk Penilaian Latihan Pemrograman Web ini dapat diselesaikan. Laporan ini disusun sebagai dokumentasi pengembangan dan pengujian sistem, sekaligus bahan evaluasi bagi penerapan otomatisasi penilaian di lingkungan Politeknik ATMI Surakarta.

Penulis mengucapkan terima kasih kepada:

1. Direktur, Ketua LPPM, dan Ketua Program Studi Teknik Mekatronika Politeknik ATMI Surakarta atas dukungan terhadap penelitian ini;
2. Mahasiswa Program Studi Teknik Mekatronika (kelas Latihan_TMK2026) atas *submission* latihan yang menjadi dasar pengujian sistem;
3. Keluarga dan rekan sejawat atas dukungan selama proses penelitian berlangsung.

Penulis menyadari laporan ini masih memiliki keterbatasan. Kritik dan saran yang membangun sangat diharapkan demi penyempurnaan penelitian dan pengembangan sistem selanjutnya.

Surakarta, Juli 2026

Peneliti

DAFTAR ISI

HALAMAN JUDUL i
LEMBAR PENGESAHAN ii
ABSTRAK iii
PRAKATA iv
DAFTAR ISI v
1. Pendahuluan 9
1.1. Rumusan Masalah 10
1.2. Tujuan Penelitian 10
1.3. Batasan Masalah 10
2. Tinjauan Pustaka dan Dasar Teori 11
2.1. Penelitian Terdahulu 11
2.2. Dasar Teori 12
2.2.1. Computer-Aided Assessment dan Automated Grading 12
2.2.2. Optical Character Recognition (OCR) dan Tesseract 12
2.2.3. Regular Expression untuk Pencocokan Pola (Pattern Matching) 12
2.2.4. Static Site Generation untuk Pelaporan Hasil Pembelajaran 13
2.2.5. Karakteristik Submission pada LMS Moodle 13
3. Metodologi Penelitian 14
3.1. Objek Penelitian 14
3.2. Metode Penelitian 14
3.2.1. Arsitektur Sistem 14
3.2.2. Pipeline Ekstraksi Konten Submission 15
3.2.3. Mekanisme Penilaian: Pencocokan Pola Berjenjang 16
3.2.4. Desain 13 Modul Evaluator 16
3.2.5. Pembangkitan Output dan Situs Web Inkremental 17
3.3. Alat dan Bahan Penelitian 17
3.4. Lokasi Penelitian 17
3.5. Diagram Alir 17
3.5.1. Tahap Analisis dan Desain Kriteria 18
3.5.2. Tahap Implementasi dan Uji Coba 18
3.5.3. Variabel Penelitian 18
3.5.4. Waktu Penelitian 19

4. Hasil dan Pembahasan 20
4.1. Implementasi Modul Evaluator 20
4.2. Statistik Hasil Evaluasi — Latihan_TMK2026 20
4.3. Pembahasan per Kelompok Latihan 21
4.3.1. Kelompok Dasar Web (Latihan 01-05) 21
4.3.2. Kelompok JavaScript (Latihan 06-09) 21
4.3.3. Kelompok PHP dan Laravel (Latihan 10-13) 21
4.4. Toleransi Sistem terhadap Variasi Format dan Noise OCR 22
4.5. Kendala dan Solusi dalam Penelitian 23
5. Penutup 24
5.1. Kesimpulan 24
5.2. Saran Pengembangan 24
Lampiran 1. Contoh Lengkap CRITERIA — Evaluator Latihan JavaScript3 (Stopwatch) 27
Lampiran 2. Struktur Direktori Proyek 28
Lampiran 3. Isi config/evaluators.json (Registry Evaluator) 28

DAFTAR GAMBAR

Gambar 3.1 Arsitektur sistem Lab3Evaluator

Gambar 3.2 Siklus pengembangan evaluator per latihan (6 tahap)

Gambar 4.1 Rata-rata nilai dan tingkat submission per latihan — Latihan_TMK2026

Jl. Mojo No. 1, Karangasem, Laweyan, Surakarta 57145. Phone : +62 271 714466
E-mail: politeknik@atmi.ac.id • Website: http://www.atmi.ac.id
Jawa Tengah, Indonesia.

DAFTAR TABEL

Tabel 2.1 Penelitian terdahulu terkait penilaian otomatis dan OCR

Tabel 3.1 Prioritas ekstraksi konten submission per format

Tabel 3.2 Ringkasan 13 modul evaluator

Tabel 3.3 Alat dan bahan penelitian

Tabel 4.1 Statistik hasil evaluasi per latihan — Latihan_TMK2026

Tabel 4.2 Perbandingan kelompok latihan berdasarkan rata-rata nilai

Tabel 4.3 Kendala teknis dan solusi selama penelitian dan pengembangan

1. Pendahuluan

Pembelajaran pemrograman web pada jenjang pendidikan vokasi umumnya disampaikan melalui rangkaian latihan bertahap: dimulai dari kontrol versi (Git), struktur dokumen HTML, tata letak dengan CSS dan framework Bootstrap, interaktivitas dengan JavaScript, hingga pengembangan aplikasi web dinamis berbasis PHP dan framework Laravel. Submission latihan dikumpulkan mahasiswa melalui Learning Management System (LMS) Moodle dalam folder yang menyimpan berkas assignment per mahasiswa. Pada praktiknya, format berkas yang dikumpulkan sangat beragam: sebagian mahasiswa mengunggah berkas kode langsung (.html, .css, .js, .php), sebagian mengemasnya dalam arsip terkompresi (.zip/.rar), sebagian membagikan tautan repositori GitHub, dan sebagian lain — karena keterbatasan atau kebiasaan — mengumpulkan cuplikan layar (screenshot) kode program yang ditempel ke dalam dokumen PDF atau DOCX.

Penilaian manual terhadap submission dengan variasi format tersebut menimbulkan beban kerja yang signifikan bagi pengajar. Untuk satu angkatan dengan puluhan mahasiswa dan belasan latihan sepanjang satu semester, jumlah submission yang perlu diperiksa dapat mencapai ratusan hingga ribuan berkas. Setiap submission harus dibuka, diperiksa terhadap serangkaian kriteria teknis (misalnya: apakah struktur HTML lengkap, apakah CSS eksternal digunakan, apakah terdapat validasi input, apakah framework yang disyaratkan benar-benar dipakai), lalu diberi skor secara konsisten dengan penilaian terhadap submission mahasiswa lain. Proses ini rentan terhadap kelelahan pemeriksa (grader fatigue), inkonsistensi antar-sesi penilaian, dan keterlambatan pemberian umpan balik kepada mahasiswa.

Upaya otomatisasi penilaian tugas pemrograman (automated assessment/automated grading) bukan hal baru dalam penelitian pendidikan komputer, namun sebagian besar pendekatan yang ada berfokus pada eksekusi kode (compile-and-run) atau unit testing terhadap kode sumber yang dapat dijalankan secara langsung [1], [4], [5]. Pendekatan tersebut kurang sesuai untuk konteks latihan HTML/CSS statis (tidak ada “output” yang bisa dieksekusi dan dibandingkan secara langsung) maupun untuk submission yang berupa dokumen cuplikan layar kode, bukan berkas kode itu sendiri. Diperlukan pendekatan penilaian yang menilai *kehadiran dan kebenaran struktural* elemen kode/markup terhadap kriteria rubrik, serta mampu mengekstraksi konten kode dari berbagai format submission termasuk dokumen hasil pindai atau cuplikan layar.

Berdasarkan uraian tersebut, penelitian ini mengembangkan Lab3Evaluator, sebuah sistem evaluasi otomatis berbasis Python yang menggabungkan dua teknik utama: (1) ekstraksi konten multi-format, termasuk Optical Character Recognition (OCR) menggunakan Tesseract untuk submission berbasis PDF/DOCX yang berisi cuplikan layar kode; dan (2) penilaian berbasis pencocokan pola (regular expression pattern matching) terhadap kriteria penilaian berjenjang per langkah kerja, dengan tiga mode pencocokan (seluruh pola wajib cocok, salah satu pola cukup, atau minimal N dari sejumlah pola) untuk mengakomodasi variasi gaya penulisan kode mahasiswa. Sistem menghasilkan skor 0-100 beserta rincian per-langkah (breakdown), serta menyajikan hasil dalam bentuk berkas

CSV/Excel untuk kebutuhan administratif dan situs web statis inkremental untuk eksplorasi hasil oleh pengajar maupun mahasiswa.

1.1. Rumusan Masalah

Berdasarkan latar belakang di atas, rumusan masalah penelitian ini adalah: bagaimana merancang dan mengimplementasikan sistem evaluasi otomatis yang (a) mampu mengekstraksi konten kode dari berbagai format submission Moodle, termasuk dokumen hasil pindai/cuplikan layar; (b) mampu menilai kode HTML/CSS/JavaScript/PHP/Laravel terhadap kriteria rubrik secara konsisten tanpa perlu mengeksekusi kode tersebut; dan (c) menghasilkan nilai yang dapat dipertanggungjawabkan (auditable) melalui rincian per-langkah penilaian.

1.2. Tujuan Penelitian

1. Mengembangkan pipeline ekstraksi konten submission yang menangani berkas HTML/ZIP/RAR/tautan GitHub secara langsung serta submission PDF/DOCX melalui OCR.

2. Mengembangkan mekanisme penilaian berbasis pencocokan pola (pattern matching) dengan kriteria berjenjang yang dapat diterapkan pada 13 jenis latihan pemrograman web yang berbeda kompleksitasnya.

3. Mengevaluasi kinerja sistem — dalam hal rata-rata nilai, tingkat submission, dan ketahanan terhadap variasi format/OCR — pada dataset nyata satu angkatan mahasiswa (Latihan_TMK2026, 53 mahasiswa, 13 latihan).

1.3. Batasan Masalah

1. Penelitian dibatasi pada 13 modul evaluator yang telah dikembangkan dan diuji, mencakup latihan Git, HTML tabel/form/div, Bootstrap, JavaScript (dasar, To-Do List, Stopwatch, integrasi web), PHP dengan basis data, dan Laravel (portofolio, To-Do App, autentikasi Breeze).

2. Mekanisme penilaian menggunakan pencocokan pola (regular expression) terhadap teks/kode hasil ekstraksi, bukan analisis semantik berbasis Large Language Model (LLM) maupun eksekusi/unit testing kode secara langsung.

3. Studi kasus statistik hasil menggunakan dataset kelas Latihan_TMK2026 (Program Studi Teknik Mekatronika, 53 mahasiswa); generalisasi ke populasi mahasiswa/program studi lain memerlukan pengujian lanjutan.

4. Akurasi OCR bergantung pada kualitas cuplikan layar/pindai yang dikumpulkan mahasiswa dan tidak dievaluasi terhadap ground-truth transkripsi manual secara menyeluruh; evaluasi akurasi OCR bersifat kualitatif berdasarkan dampaknya terhadap perubahan nilai (lihat sub-bab 4.4).

2. Tinjauan Pustaka dan Dasar Teori

2.1. Penelitian Terdahulu

Penelitian terkait dikelompokkan pada tiga area: penilaian otomatis tugas pemrograman berbasis pencocokan pola/eksekusi kode, penilaian otomatis terintegrasi Learning Management System (LMS), dan Optical Character Recognition (OCR) untuk keperluan penilaian dokumen hasil pindai. Ringkasan penelitian terdahulu disajikan pada Tabel 2.1.

No.	Penelitian	Objek Penelitian	Metode	Hasil Penelitian	Keterbatasan / Gap terhadap Lab3Evaluator
1	D.S. Morris (2002) [1]	Grading tugas Java (Rutgers, 400-600 submission/minggu)	Java reflection + inheritance + regex Perl	Waktu grading turun ke hitungan menit/batch	Berbasis eksekusi kode, khusus Java; bukan ekstraksi dari dokumen/OCR
2	D.S. Morris (2003) [2]	Kelanjutan sistem grading Java interaktif	Suite Perl+Java + basis data, grader-in-the-loop	300-600 tugas/minggu tergrade konsisten	Masih perlu intervensi manual grader untuk kasus ambigu
3	Messer et al. (2024) [3]	Systematic review 121 paper automated grading tools (2017-2021)	PRISMA systematic literature review	Mayoritas tool menilai correctness via unit test	Evaluasi readability/kualitas struktur kode masih jarang diteliti — celah novelty
4	Edwards & Perez-Quinones (2008) [4]	Web-CAT, automated grader open-source lintas bahasa	Web-based submission + test-driven grading	Dipakai luas lintas institusi pendidikan CS	Fokus eksekusi/unit test; tidak menangani submission non-kode (scan/OCR)
5	Rodríguez-del-Pino et al. (2022) [5]	Virtual Programming Lab (VPL) plugin Moodle	Eksekusi/test otomatis submission dari dalam LMS	Mengurangi beban koreksi manual dosen	Plugin server-side terintegrasi Moodle, bukan tool eksternal pemroses folder ekspor
6	Purbaningtyas (2012) [7]	Penilaian otomatis modul Java sederhana	Regex (java.util.regex Pattern/Matcher) + black-box test	Mempercepat waktu pemeriksaan	Terbatas pada modul Java sederhana, tidak multi-bahasa web dan tanpa OCR
7	Smith (2007) [8]	Arsitektur mesin Tesseract OCR	Segmentasi baris/kata + adaptive thresholding + klasifikasi 2-pass	Akurasi kompetitif untuk teks cetak	Akurasi turun pada tulisan tangan/tata letak kompleks — dasar preprocessing OCR Lab3Evaluator
8	Jocović et al. (2023) [11]	Sistem penilaian otomatis ujian kertas (paper test)	Tesseract OCR ekstraksi jawaban + pencocokan kunci jawaban	Efektif untuk teks cetak/screenshot rapi	Akurasi turun pada tulisan tangan tidak rapi — memperkuat justifikasi fokus Lab3Evaluator pada submission cetak/screenshot

Tabel 2.1 Penelitian terdahulu terkait penilaian otomatis dan OCR

Berdasarkan Tabel 2.1, dapat disimpulkan tiga kesenjangan (gap) yang menjadi ruang kontribusi Lab3Evaluator. Pertama, penelitian grading berbasis regex yang ada ([1], [2], [7]) terbatas pada satu bahasa pemrograman (umumnya Java) dan mengasumsikan submission berupa berkas kode langsung, sedangkan Lab3Evaluator menerapkan pencocokan pola berjenjang pada enam kelompok teknologi web (HTML, CSS, Bootstrap, JavaScript, PHP, Laravel) sekaligus menangani submission non-berkas-kode (PDF/DOCX cuplikan layar). Kedua, review sistematis oleh Messer et al. [3] mengonfirmasi bahwa

evaluasi terhadap struktur/kualitas tampilan kode (bukan sekadar lulus/gagal unit test) masih jarang diteliti — inilah yang menjadi fokus kriteria CRITERIA per-langkah pada Lab3Evaluator (misalnya kelengkapan struktur HTML, penggunaan layout CSS, komponen Bootstrap). Ketiga, penelitian OCR untuk keperluan penilaian ([8], [11]) selama ini diterapkan pada ujian kertas/jawaban terstruktur, sedangkan penelitian ini menerapkan OCR pada domain baru: cuplikan layar kode program di dalam dokumen PDF/DOCX.

2.2. Dasar Teori

2.2.1. Computer-Aided Assessment dan Automated Grading

Computer-Aided Assessment (CAA) adalah pemanfaatan sistem komputer untuk membantu atau menggantikan sebagian proses penilaian pembelajaran, mencakup penilaian jawaban terstruktur (seperti pada sistem STACK untuk soal matematika [10]) maupun penilaian tugas pemrograman (automated grading). Pada domain tugas pemrograman, dua pendekatan utama automated grading yang umum digunakan adalah: (1) dynamic analysis — menjalankan kode mahasiswa dan membandingkan keluaran terhadap hasil yang diharapkan (unit testing), sebagaimana diterapkan pada Web-CAT [4] dan Virtual Programming Lab [5]; dan (2) static/pattern-based analysis — memeriksa struktur atau kehadiran elemen tertentu pada kode tanpa mengeksekusinya, sebagaimana diterapkan oleh Morris [1], [2] dan Purbaningtyas [7] menggunakan regular expression. Lab3Evaluator menggunakan pendekatan kedua karena latihan HTML/CSS statis tidak memiliki keluaran eksekusi yang dapat dibandingkan secara langsung, dan pendekatan berbasis pola lebih fleksibel untuk menilai kehadiran elemen struktural (tag, atribut, properti CSS) tanpa memerlukan lingkungan eksekusi/browser otomatis.

2.2.2. Optical Character Recognition (OCR) dan Tesseract

OCR adalah proses konversi gambar berisi teks menjadi representasi teks yang dapat diproses mesin. Tesseract, sebagaimana dijelaskan oleh Smith [8], menggunakan pipeline dua-tahap: segmentasi baris dan kata dari citra, dilanjutkan klasifikasi karakter melalui dua pass (pass pertama untuk adaptasi, pass kedua untuk klasifikasi akhir menggunakan model yang telah disesuaikan). Akurasi Tesseract dilaporkan baik untuk teks cetak (typewritten) dengan tata letak sederhana, namun menurun pada tulisan tangan atau tata letak kompleks [8]. Temuan ini konsisten dengan penerapan OCR untuk penilaian ujian kertas oleh Jocović et al. [11], yang melaporkan akurasi memadai untuk jawaban tercetak/terstruktur namun menurun pada tulisan tangan tidak rapi. Karena cuplikan layar kode program (dari editor kode atau terminal) pada dasarnya adalah teks bertipe monospace yang cetak dan terstruktur, kondisi ini mendekati skenario ideal OCR, sehingga menjadi dasar pemilihan Tesseract pada Lab3Evaluator — dengan tambahan tahap prapemrosesan citra (rendering resolusi tinggi via PyMuPDF dan inversi warna latar gelap) untuk mendekatkan kondisi masukan ke skenario tersebut.

2.2.3. Regular Expression untuk Pencocokan Pola (Pattern Matching)

Regular expression (regex) adalah notasi formal untuk mendeskripsikan pola string, yang dapat digunakan untuk memeriksa kehadiran, kecocokan, atau ekstraksi substring

tertentu dari suatu teks. Dalam konteks automated grading, regex telah digunakan sejak awal 2000-an untuk memeriksa struktur keluaran atau kode program tanpa memerlukan parser bahasa pemrograman penuh [1], [7]. Pendekatan ini memiliki kelebihan berupa kesederhanaan implementasi dan independensi terhadap bahasa/kerangka kerja, namun memiliki kelemahan berupa kerentanan terhadap variasi gaya penulisan (whitespace, urutan atribut, penamaan variabel) dan false negative jika pola dirancang terlalu ketat. Lab3Evaluator mengatasi kelemahan ini dengan tiga mode pencocokan pada setiap kriteria penilaian — require_all (seluruh pola wajib cocok), pencocokan minimal satu pola, dan min_matches=N (minimal N dari beberapa pola alternatif) — sehingga kriteria dapat dirancang toleran terhadap variasi implementasi mahasiswa yang secara fungsional setara (dibahas lebih lanjut pada sub-bab 3.2.3 dan 4.4).

2.2.4. Static Site Generation untuk Pelaporan Hasil Pembelajaran

Static site generation adalah pendekatan pembangkitan halaman web dari data terstruktur (basis data, berkas JSON, dsb.) menjadi berkas HTML statis yang dapat disajikan tanpa server aplikasi dinamis. Pendekatan ini umum digunakan untuk menyajikan dashboard hasil pembelajaran yang dibangkitkan otomatis dari data terstruktur, sebagaimana ditunjukkan pada sistem dashboard learning analytics multimodal M2LADS [6] yang menyusun visualisasi web dari data heterogen. Lab3Evaluator menerapkan prinsip serupa pada skala lebih sederhana: setiap hasil evaluasi disimpan sebagai berkas JSON (single source of truth), yang kemudian dibangkitkan menjadi situs statis (dashboard per prodi, per latihan, per mahasiswa) secara inkremental — hanya halaman latihan yang baru dievaluasi yang diregenerasi, sementara data latihan lain tetap dipertahankan dari JSON yang sudah ada.

2.2.5. Karakteristik Submission pada LMS Moodle

Moodle adalah salah satu LMS open-source yang paling banyak digunakan di institusi pendidikan, termasuk untuk pengelolaan submission tugas (assignment). Struktur folder ekspor submission Moodle menyertakan identitas mahasiswa dalam nama folder bersama sebuah ID numerik internal. Salah satu tantangan praktis yang ditemukan selama pengembangan Lab3Evaluator adalah ID numerik tersebut **tidak stabil** antar-aktivitas tugas — mahasiswa yang sama dapat memiliki ID berbeda pada latihan yang berbeda — sehingga identitas mahasiswa pada sistem ini digunakan berdasarkan nama hasil penguraian (parsing) nama folder, bukan ID numerik tersebut. Pendekatan integrasi lain seperti Virtual Programming Lab [5] menghindari isu ini dengan berjalan sebagai plugin di dalam Moodle (memiliki akses langsung ke identitas pengguna melalui API internal), sedangkan Lab3Evaluator bekerja pada hasil ekspor folder submission secara offline sehingga harus menangani ketidakstabilan ini pada lapisan aplikasi.

3. Metodologi Penelitian

3.1. Objek Penelitian

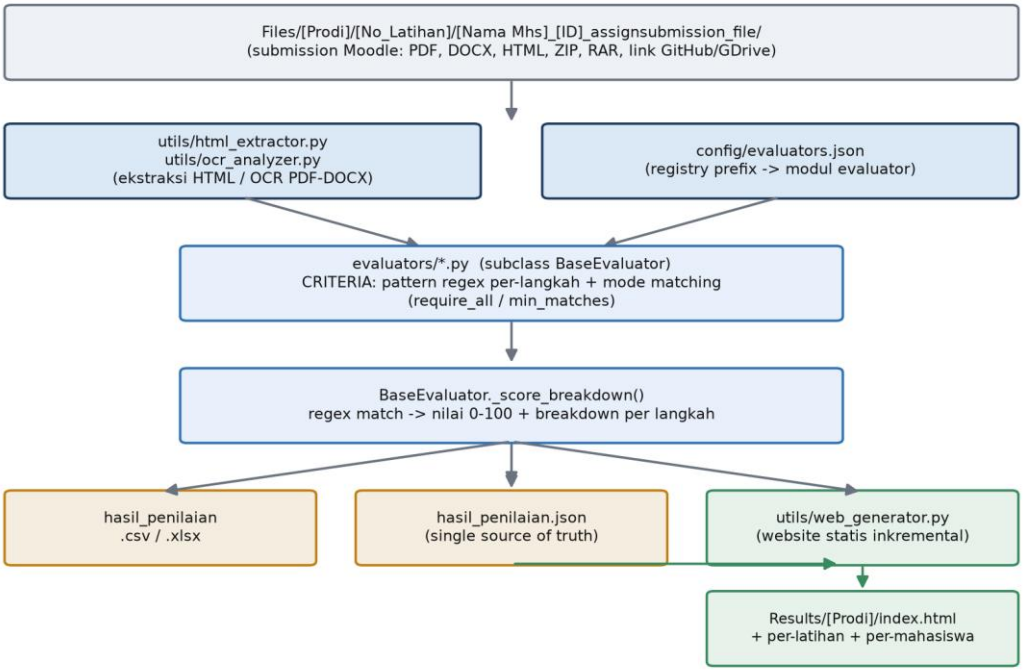
Objek penelitian ini adalah sistem evaluasi otomatis Lab3Evaluator beserta dataset submission latihan pemrograman web mahasiswa Program Studi Teknik Mekatronika, Politeknik ATMI Surakarta, kelas Latihan_TMK2026 (53 mahasiswa, 13 latihan sepanjang satu semester). Sistem dikembangkan sebagai aplikasi command-line berbasis Python 3.13 yang memproses folder hasil ekspor submission Moodle secara batch (offline), tanpa memerlukan integrasi langsung ke server Moodle.

3.2. Metode Penelitian

Penelitian menggunakan pendekatan design and development research dengan siklus iteratif per modul evaluator: setiap latihan dianalisis, dirancang kriteria penilaiannya, diimplementasikan, diuji pada sampel submission nyata, dijalankan penuh, lalu diverifikasi hasilnya pada situs web keluaran — dengan iterasi ulang pada tahap desain/implementasi apabila ditemukan pola regex yang meleset terhadap kasus/variasi baru (Gambar 3.2, sub-bab 3.5).

3.2.1. Arsitektur Sistem

Arsitektur Lab3Evaluator terdiri atas lima lapisan: (1) sumber submission mentah pada struktur folder Moodle; (2) lapisan ekstraksi konten (html_extractor.py dan ocr_analyzer.py); (3) lapisan evaluator berbasis registry prefix nomor folder yang dipetakan ke modul Python melalui config/evaluators.json; (4) mekanisme penilaian pencocokan pola pada BaseEvaluator; dan (5) lapisan keluaran (CSV/Excel/JSON) serta pembangkit situs web statis inkremental. Diagram arsitektur ditunjukkan pada Gambar 3.1.



Gambar 3.1 Arsitektur sistem Lab3Evaluator

3.2.2. Pipeline Ekstraksi Konten Submission

Modul html_extractor.py menerapkan urutan prioritas ekstraksi konten sebagaimana ditunjukkan pada Tabel 3.1. Urutan ini dirancang berdasarkan observasi terhadap variasi format submission mahasiswa selama pengembangan sistem — file HTML langsung diprioritaskan karena tidak memerlukan pemrosesan tambahan, sedangkan OCR ditempatkan sebagai fallback terakhir karena biaya komputasi dan risiko kesalahan bacanya lebih tinggi dibanding pembacaan teks langsung.

Prioritas	Format / Kondisi	Penanganan
1	HTML/HTM langsung di root folder	Dibaca langsung sebagai teks
2	Arsip ZIP	Diekstraksi, cari HTML terbaik (prioritas index.html, path terpendek)
3	Arsip RAR	Diekstraksi via rarfile / 7-Zip (evaluator Laravel)
4	TXT/MD berisi URL GitHub	Unduh repo (coba branch main -> master -> semua branch via API), simpan sebagai audit trail
5	TXT/MD berisi URL Google Drive	Unduh via gdown, ekstrak jika berupa arsip
6	TXT berisi markup HTML/CSS	Dibaca langsung sebagai teks (dengan deteksi heuristik penanda HTML)
7	HTML di subfolder (rekursif)	Pencarian rekursif, termasuk folder source/ hasil preprocessing
8	PDF (fallback)	Render per halaman via PyMuPDF -> deteksi gambar embedded -> OCR Tesseract
9	DOCX (fallback)	Ekstraksi teks via python-docx + OCR gambar embedded pada word/media/

Tabel 3.1 Prioritas ekstraksi konten submission per format

3.2.3. Mekanisme Penilaian: Pencocokan Pola Berjenjang

Setiap modul evaluator mewarisi kelas BaseEvaluator dan mendefinisikan atribut CRITERIA berupa daftar kriteria penilaian per langkah kerja. Setiap kriteria memiliki bobot poin dan satu atau lebih pola regular expression, dengan salah satu dari tiga mode pencocokan: seluruh pola wajib cocok (require_all=True), cukup satu pola yang cocok (require_all=False), atau minimal N dari beberapa pola alternatif harus cocok (min_matches=N). Mode ketiga secara khusus ditambahkan selama pengembangan untuk mengakomodasi kriteria yang memiliki beberapa cara implementasi yang sama-sama valid (misalnya: layout dapat dibuat dengan flexbox, grid, float, atau tabel CSS). Cuplikan inti mekanisme penilaian ditunjukkan pada Kode Program 3.1.

```
def _score_breakdown(self, text: str) -> tuple[int, list[dict]]:
    flags = re.IGNORECASE | re.MULTILINE
    total = 0
    breakdown = []
    for crit in self.CRITERIA:
        patterns = crit["patterns"]
        min_matches = crit.get("min_matches")
        if min_matches is not None:
            matched = [p for p in patterns if re.search(p, text, flags)]
            passed = len(matched) >= min_matches
        elif crit.get("require_all"):
            missing = [p for p in patterns if not re.search(p, text, flags)]
            passed = len(missing) == 0
        else:
            passed = any(re.search(p, text, flags) for p in patterns)
        earned = crit["points"] if passed else 0
        total += earned
        breakdown.append({"step": crit["step"], "max_points": crit["points"],
"earned": earned})
    return total, breakdown
```

Kode Program 3.1 Mekanisme pencocokan pola berjenjang (evaluators/base.py, disederhanakan)

3.2.4. Desain 13 Modul Evaluator

Tiga belas modul evaluator dikembangkan secara bertahap mengikuti urutan kurikulum satu semester, dari latihan penggunaan Git dasar hingga autentikasi Laravel Breeze. Ringkasan tiap modul — jumlah langkah penilaian, total poin, dan format submission yang didukung — disajikan pada Tabel 3.2.

No.	Nama Latihan	Jml. Langkah	Format Input Didukung
01	RPL - Latihan Github	12	PDF/DOCX (OCR)
02	LatihanTabel (Web01)	11	.html/.zip/.rar/.txt, GitHub, PDF/DOCX OCR
03	LatihanForm (Web02)	8	sda + CSS eksternal rekursif
04	LatihanDiv (Web03)	10	sda + CSS dalam .txt
05	LatihanBootstrap (Web04)	10	sda + folder source/ hasil preprocessing
06	LatihanJavaScript1	10	.html + <script> inline / .js terpisah
07	LatihanJavaScript2 (To-Do List)	10	sda
08	LatihanJavaScript3 (Stopwatch)	5	sda

09	LatihanJavaScriptWeb	5	HTML+CSS+JS folder Bootstrap, deteksi chart library
10	Final Web PHP	7 (6 fungsional + 1 bonus)	.php/.html/.js/.sql folder
11	Laravel 1 - Portofolio	4	ZIP/RAR (7-Zip), folder Laravel
12	Laravel 2 - ToDo App	4	sda
13	Laravel 3 - Breeze Auth	4	sda

Tabel 3.2 Ringkasan 13 modul evaluator (seluruhnya berskala 0-100 poin)

3.2.5. Pembangkitan Output dan Situs Web Inkremental

Setiap eksekusi evaluator menghasilkan tiga berkas: CSV dan XLSX (untuk kebutuhan administratif dan impor ke spreadsheet), serta JSON yang menjadi single source of truth berisi nilai beserta rincian breakdown per langkah. Modul web_generator.py membaca seluruh berkas JSON yang tersedia dan membangkitkan situs statis berisi (a) halaman ringkasan seluruh mahasiswa x seluruh latihan, (b) halaman per latihan dengan rincian breakdown dan tautan unduh CSV/XLSX, dan (c) halaman per mahasiswa yang merekap seluruh latihan. Proses ini bersifat inkremental: menjalankan evaluasi untuk satu latihan hanya memperbarui halaman latihan tersebut, sementara halaman latihan lain tetap dipertahankan dari JSON yang sudah ada — sehingga rincian breakdown tidak pernah hilang akibat evaluasi ulang latihan lain.

3.3. Alat dan Bahan Penelitian

No.	Alat/Bahan	Fungsi
1	Python 3.13.1	Bahasa pemrograman utama sistem
2	PyMuPDF (fitz)	Rendering halaman PDF menjadi citra untuk OCR
3	Tesseract OCR	Mesin OCR (mode --psm 6 --oem 3) untuk ekstraksi teks dari citra
4	Pillow	Prapemrosesan citra (inversi latar gelap) sebelum OCR
5	python-docx	Ekstraksi teks dan gambar embedded dari berkas DOCX
6	openpyxl	Pembangkitan keluaran berformat Excel (.xlsx)
7	rarfile / 7-Zip	Ekstraksi arsip RAR
8	gdown	Pengunduhan berkas/folder dari Google Drive
9	Dataset Files/Latihan_TMK2026	53 mahasiswa x 13 latihan submission Moodle (studi kasus)

Tabel 3.3 Alat dan bahan penelitian

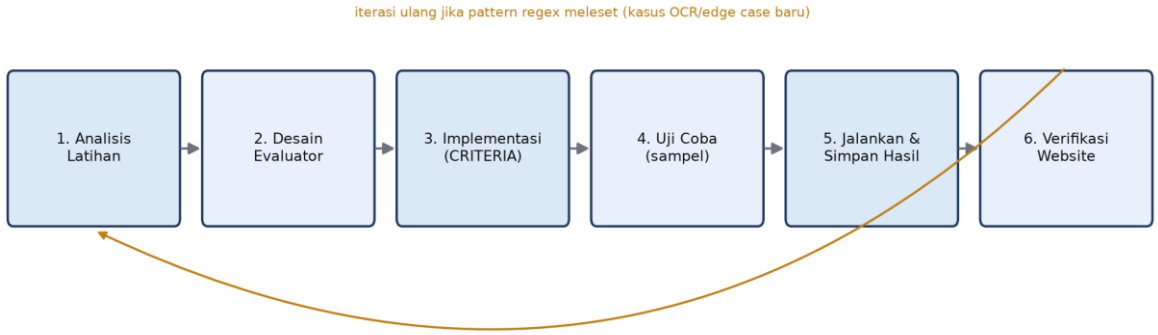
3.4. Lokasi Penelitian

Penelitian dan pengembangan sistem dilakukan di lingkungan Program Studi Teknik Mekatronika, Politeknik ATMI Surakarta. Dataset submission yang digunakan sebagai studi kasus berasal dari aktivitas pembelajaran pemrograman web pada program studi yang sama.

3.5. Diagram Alir

Setiap modul evaluator dikembangkan mengikuti siklus enam tahap yang ditunjukkan pada Gambar 3.2: analisis latihan, desain kriteria evaluator, implementasi, uji coba pada sampel, eksekusi penuh dan penyimpanan hasil, serta verifikasi pada situs web keluaran. Apabila pada tahap uji coba maupun setelah eksekusi penuh ditemukan pola regex

yang tidak menangkap variasi implementasi mahasiswa yang valid, siklus kembali ke tahap desain untuk perbaikan pola.



Gambar 3.2 Siklus pengembangan evaluator per latihan (6 tahap)

3.5.1. Tahap Analisis dan Desain Kriteria

Tahap ini diawali dengan pemeriksaan sampel submission latihan yang bersangkutan untuk memahami format berkas yang dominan digunakan mahasiswa (kode langsung, arsip, tautan repositori, atau dokumen cuplikan layar) serta variasi gaya implementasi yang mungkin muncul. Kriteria penilaian kemudian dirumuskan sebagai daftar langkah kerja dengan bobot poin yang menjumlah 100, mengikuti instruksi latihan yang diberikan kepada mahasiswa.

3.5.2. Tahap Implementasi dan Uji Coba

Kriteria yang telah dirancang diimplementasikan sebagai daftar CRITERIA pada modul evaluator baru, kemudian diuji terhadap beberapa sampel submission representatif (kasus normal, kasus tepi/edge case, dan format tidak standar). Pola regex yang tidak menangkap variasi implementasi yang valid secara fungsional — misalnya perbedaan bahasa label (Indonesia/Inggris), perbedaan teknik CSS layout, atau kesalahan baca OCR yang sistematis — diperbaiki secara iteratif pada tahap ini (lihat sub-bab 4.4 dan 4.5 untuk contoh konkret).

3.5.3. Variabel Penelitian

Variabel bebas: Desain pola regex dan mode pencocokan (require_all / min_matches) per kriteria pada tiap modul evaluator; jenis pipeline ekstraksi konten yang digunakan (langsung, arsip, GitHub, atau OCR).

Variabel kontrol: Struktur kontrak BaseEvaluator, algoritma _score_breakdown(), skala penilaian 0-100 per latihan, serta dataset submission yang sama (Latihan_TMK2026) untuk seluruh modul evaluator.

Variabel respons: Nilai 0-100 per mahasiswa per latihan beserta rincian breakdown per langkah, status submission (submitted / not_submitted / pdf_ocr / docx_ocr / no_file / dsb.), dan tingkat submission per latihan.

Jl. Mojo No. 1, Karangasem, Laweyan, Surakarta 57145. Phone : +62 271 714466
E-mail: politeknik@atmi.ac.id • Website: http://www.atmi.ac.id
Jawa Tengah, Indonesia.

3.5.4. Waktu Penelitian

Pengembangan dan pengujian sistem dilakukan sepanjang periode [diisi rentang tanggal/bulan pengembangan], mengikuti jadwal penyampaian latihan pada semester berjalan.

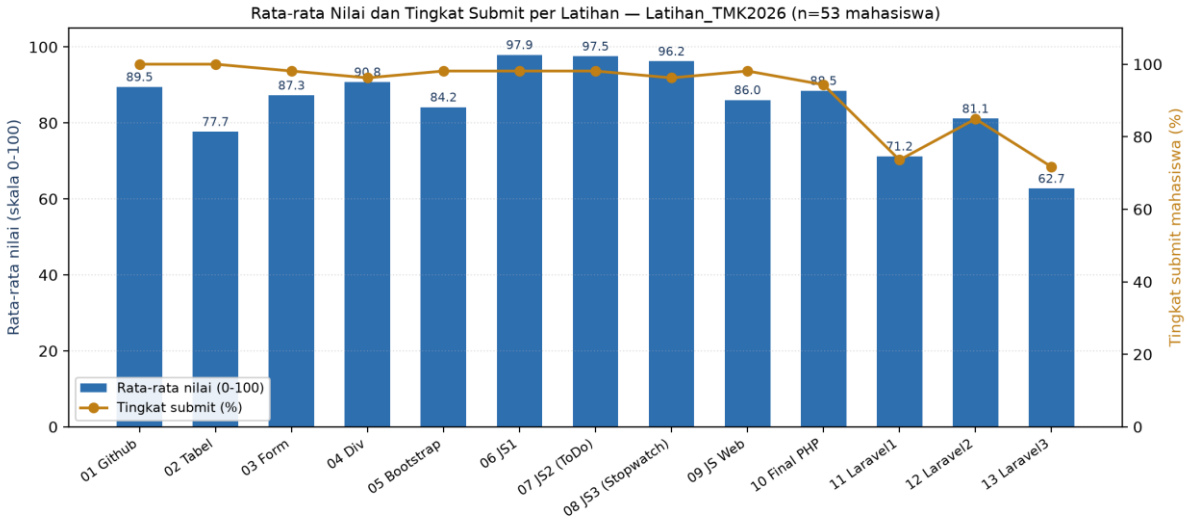
4. Hasil dan Pembahasan

4.1. Implementasi Modul Evaluator

Seluruh 13 modul evaluator yang dirancang pada sub-bab 3.2.4 berhasil diimplementasikan dan diuji hingga selesai (Tabel 3.2). Basis kode sistem terdiri atas sekitar 4.900 baris kode Python, mencakup 13 modul evaluator (rata-rata 245 baris per modul, dengan modul Laravel yang paling kompleks berkisar 400-510 baris karena menggabungkan pembacaan migration, model, controller, routes, dan arsip SQL), lima modul utilitas bersama (ekstraksi HTML, OCR, penulisan hasil, pembangkit situs web, penguraian folder Moodle), serta dua skrip command-line (preprocessing dan entry point evaluasi).

4.2. Statistik Hasil Evaluasi — Latihan_TMK2026

Sistem dijalankan penuh terhadap dataset kelas Latihan_TMK2026 (53 mahasiswa). Grafik rata-rata nilai dan tingkat submission per latihan ditunjukkan pada Gambar 4.1, dengan data lengkap pada Tabel 4.1.



Gambar 4.1 Rata-rata nilai dan tingkat submission per latihan — Latihan_TMK2026

No.	Latihan	N Mhs.	Submit	Avg (semua)	Avg (submit saja)
01	RPL - Latihan Github	53	53	89.5	89.5
02	LatihanTabel	53	53	77.7	77.7
03	LatihanForm	53	52	87.3	88.9
04	LatihanDiv	53	51	90.8	94.3
05	LatihanBootstrap	53	52	84.2	85.8
06	JavaScript1	53	52	97.9	99.8
07	JavaScript2 (ToDo)	53	52	97.5	99.4
08	JavaScript3 (Stopwatch)	53	51	96.2	100.0
09	JavaScriptWeb	53	52	86.0	87.7
10	Final Web PHP	53	50	88.5	93.8
11	Laravel1 - Portofolio	53	39	71.2	96.8
12	Laravel2 - ToDo App	53	45	81.1	95.6
13	Laravel3 - Breeze Auth	53	38	62.7	87.5

Tabel 4.1 Statistik hasil evaluasi per latihan — Latihan_TMK2026 (n=53 mahasiswa)

Kolom “Avg (semua)” menghitung rata-rata dari seluruh 53 mahasiswa termasuk yang tidak submit (bernilai 0), sedangkan “Avg (submit saja)” menghitung rata-rata hanya dari mahasiswa yang mengumpulkan submission. Secara keseluruhan (689 titik data mahasiswa x latihan), rata-rata nilai gabungan seluruh latihan adalah 85,4 (semua mahasiswa) dan 92,0 (submission saja), dengan tingkat submission rata-rata 93,0%. Latihan dengan rata-rata tertinggi adalah JavaScript1 (97,9) dan terendah adalah Laravel3 - Breeze Auth (62,7).

4.3. Pembahasan per Kelompok Latihan

4.3.1. Kelompok Dasar Web (Latihan 01-05)

Kelompok latihan dasar web (Git, Tabel, Form, Div, Bootstrap) menghasilkan rata-rata nilai antara 77,7 hingga 90,8. LatihanTabel (02) memiliki rata-rata terendah pada kelompok ini (77,7), yang secara kualitatif konsisten dengan cakupan kriterianya yang terluas (11 langkah) — meliputi elemen struktural HTML5 semantik (header, nav, section/main, aside, footer) yang lebih banyak dilupakan mahasiswa dibanding elemen dasar seperti tabel dan gaya latar. Sebaliknya, LatihanDiv (04) yang memiliki jumlah kriteria lebih sedikit (10 langkah) namun kriteria layout dan warnanya dirancang toleran terhadap banyak teknik implementasi (flexbox, grid, dan float sekaligus; CSS class maupun inline style) mencapai rata-rata 90,8 — mengindikasikan bahwa toleransi pola regex berkontribusi signifikan terhadap rata-rata nilai yang dihasilkan sistem, terlepas dari kompleksitas materi latihan itu sendiri.

4.3.2. Kelompok JavaScript (Latihan 06-09)

Kelompok latihan JavaScript mencatat rata-rata tertinggi di antara seluruh kelompok (86,0 hingga 97,9) dengan tingkat submission konsisten di atas 96%. Tingginya rata-rata ini berkaitan dengan desain kriteria yang secara eksplisit mengakomodasi variasi implementasi fungsional yang setara — misalnya kriteria tombol kontrol pada evaluator Stopwatch (08) menerima label Indonesia ("mulai", "jeda", "ulang") maupun Inggris, dan kriteria timer menerima setInterval maupun requestAnimationFrame sebagai implementasi yang sama-sama valid. Hal ini menunjukkan bahwa toleransi pencocokan pola terhadap variasi implementasi fungsional yang sama, dan bukan semata kompleksitas materi, adalah faktor dominan yang memengaruhi rata-rata nilai yang dihasilkan sistem.

4.3.3. Kelompok PHP dan Laravel (Latihan 10-13)

Kelompok latihan PHP dan Laravel menunjukkan pola yang berbeda: rata-rata nilai dari submission yang masuk tetap tinggi (87,5 hingga 96,8), namun tingkat submission menurun tajam, khususnya pada Laravel1 (39/53, 73,6%) dan Laravel3 (38/53, 71,7%). Akibatnya rata-rata gabungan (termasuk yang tidak submit) pada Laravel3 turun menjadi 62,7 — rata-rata terendah di antara seluruh latihan. Penurunan tingkat submission pada kelompok ini kemungkinan berkaitan dengan kompleksitas teknis penyiapan proyek Laravel (instalasi dependency, konfigurasi basis data, ukuran arsip proyek yang besar) yang menjadi hambatan tersendiri di luar kemampuan pemrograman itu sendiri, bukan indikasi bahwa sistem evaluasi gagal mengenali submission yang valid. Pada Laravel3 khususnya, di antara

mahasiswa yang submit, ditemukan pola sistematis nilai 75/100 yang berkaitan dengan kriteria penambahan kolom data pengguna tambahan (NIM, nama lengkap, tempat dan tanggal lahir) pada tabel users — sebuah kebutuhan fungsional latihan yang relatif sering terlewat oleh mahasiswa, bukan keterbatasan pencocokan pola sistem.

Kelompok Latihan	Rata-rata Nilai (semua mhs.)	Rata-rata Tingkat Submission
Dasar Web (01-05)	84.0	94.7%
JavaScript (06-09)	94.4	97.6%
PHP & Laravel (10-13)	75.9	82.5%

Tabel 4.2 Perbandingan kelompok latihan berdasarkan rata-rata nilai dan tingkat submission

4.4. Toleransi Sistem terhadap Variasi Format dan Noise OCR

Selama pengembangan, ditemukan bahwa sebagian besar peningkatan akurasi penilaian berasal dari perluasan toleransi pola regex terhadap variasi implementasi mahasiswa yang secara fungsional benar, bukan dari perbaikan logika inti algoritma penilaian. Beberapa temuan representatif yang teramati selama pengujian iteratif dirangkum berikut ini (identitas mahasiswa dianonimkan).

Deteksi gambar embedded pada PDF. Sebagian submission PDF berisi dua lapis konten: teks instruksi (native text layer) dan cuplikan layar terminal (embedded image) pada halaman yang sama. Ekstraksi yang hanya membaca native text layer melewatkan seluruh konten cuplikan layar. Setelah ditambahkan deteksi otomatis (jika halaman mengandung gambar embedded, selalu lakukan OCR penuh), rata-rata nilai evaluator Latihan Github naik dari sekitar 79 menjadi 89,4.

DOCX berisi gambar tanpa teks. Sejumlah submission DOCX hanya berisi cuplikan layar tertempel, sehingga ekstraksi teks standar python-docx menghasilkan nol karakter. Penambahan ekstraksi gambar dari arsip internal DOCX (word/media/) diikuti OCR mengatasi kasus ini sepenuhnya.

Variasi shell terminal. Mahasiswa menggunakan Git Bash, Command Prompt, maupun PowerShell yang masing-masing memiliki format prompt berbeda. Pola navigasi direktori dan deteksi path diperluas untuk mengenali ketiga format tersebut.

Kesalahan baca OCR yang sistematis. OCR pada cuplikan layar kode HTML kerap salah membaca karakter serupa, misalnya tag <h1> terbaca sebagai <hil> (angka 1 terbaca sebagai huruf i/l). Pola regex kriteria terkait disesuaikan menjadi toleran terhadap kesalahan baca yang sistematis ini (misalnya <h[1il]{1,2}[\s>]) tanpa mengorbankan ketelitian pemeriksaan.

Label formulir berbahasa Indonesia. Kriteria yang semula hanya mengenali label berbahasa Inggris (Name, Address, Postcode) diperluas untuk turut menerima padanan bahasa Indonesia (Nama, Alamat, Kode Pos), karena sebagian mahasiswa menulis label formulir dalam bahasa Indonesia yang sebetulnya valid secara fungsional.

4.5. Kendala dan Solusi dalam Penelitian

Kendala	Deskripsi	Solusi
Encoding karakter non-ASCII pada Windows	Output OCR memuat karakter non-ASCII yang menyebabkan error encoding pada konsol PowerShell default	Menetapkan variabel lingkungan PYTHONIOENCODING=utf-8 sebelum menjalankan skrip evaluasi/OCR
ID Moodle tidak stabil antar-latihan	Mahasiswa yang sama memiliki ID numerik submission berbeda pada tiap latihan, sehingga tidak dapat dijadikan kunci identitas lintas-latihan	Identitas mahasiswa ditentukan dari hasil penguraian nama pada nama folder submission, bukan ID numerik
Ekstraksi arsip RAR pada proyek Laravel besar	Pustaka rarfile Python memerlukan binary UnRAR eksternal yang tidak selalu tersedia/konsisten	Ekstraksi RAR pada evaluator Laravel dialihkan memanfaatkan 7-Zip melalui pemanggilan subprocess
Arsip proyek Laravel bersarang (nested)	Sebagian submission berupa ZIP di dalam ZIP, atau ZIP berisi RAR, dengan ukuran mencapai puluhan hingga ratusan megabyte (termasuk folder vendor/)	Ekstraksi selektif yang melewati folder besar tidak relevan (vendor/, node_modules/, .git/, storage/logs/) dan pencarian rekursif root proyek Laravel (penanda berkas artisan)
Nama file HTML ganda pada root submission	Ketika folder submission memiliki beberapa berkas .html, pemilihan berkas pertama secara alfabetis dapat memilih berkas yang bukan submission utama	Pengurutan pemilihan berkas HTML memprioritaskan nama index.html sebelum mengurutkan alfabetis

Tabel 4.3 Kendala teknis dan solusi selama penelitian dan pengembangan

5. Penutup

Berdasarkan pengembangan dan pengujian yang telah dilakukan, berikut disampaikan kesimpulan dan saran pengembangan untuk penelitian selanjutnya yang berkaitan dengan sistem evaluasi otomatis Lab3Evaluator.

5.1. Kesimpulan

1. Mengembangkan pipeline ekstraksi konten submission multi-format.

Pipeline ekstraksi berhasil dikembangkan dan menangani sembilan jalur masukan berbeda (HTML langsung, ZIP, RAR, tautan GitHub, tautan Google Drive, teks berisi markup, pencarian rekursif subfolder, OCR PDF, dan OCR DOCX) sebagaimana dirangkum pada Tabel 3.1, sehingga sistem dapat memproses variasi format submission mahasiswa tanpa memerlukan keseragaman format pengumpulan.

2. Mengembangkan mekanisme penilaian berbasis pencocokan pola berjenjang.

Mekanisme penilaian dengan tiga mode pencocokan (require_all, pencocokan satu pola, dan min_matches=N) berhasil diterapkan pada 13 modul evaluator yang mencakup HTML, CSS, Bootstrap, JavaScript, PHP, dan Laravel. Analisis pada sub-bab 4.3 dan 4.4 menunjukkan bahwa modul dengan kriteria yang dirancang toleran terhadap variasi implementasi fungsional yang setara secara konsisten menghasilkan rata-rata nilai lebih tinggi dan lebih tahan terhadap variasi gaya penulisan kode mahasiswa.

3. Mengevaluasi kinerja sistem pada dataset nyata.

Pengujian pada dataset Latihan_TMK2026 (53 mahasiswa, 13 latihan, 689 titik data) menghasilkan rata-rata nilai gabungan 85,4 (seluruh mahasiswa) dan 92,0 (submission saja), dengan tingkat submission rata-rata 93,0%. Kelompok latihan JavaScript mencatat rata-rata tertinggi (94,4) berkat desain kriteria yang toleran, sedangkan kelompok Laravel mencatat rata-rata terendah (75,9) yang terutama disebabkan oleh penurunan tingkat submission akibat kompleksitas penyiapan proyek, bukan kegagalan sistem mengenali submission yang valid.

5.2. Saran Pengembangan

1. Menambahkan lapisan analisis semantik berbasis Large Language Model (LLM) sebagai pelengkap (bukan pengganti) pencocokan pola, khususnya untuk kasus di mana implementasi mahasiswa valid secara fungsional namun tidak tercakup oleh pola regex yang telah dirancang.

2. Mengembangkan mekanisme umpan balik otomatis kepada mahasiswa (misalnya notifikasi email atau integrasi kembali ke Moodle) berdasarkan rincian breakdown per langkah yang sudah tersedia pada sistem, sehingga mahasiswa dapat memperbaiki submission sebelum tenggat akhir.

Jl. Mojo No. 1, Karangasem, Laweyan, Surakarta 57145. Phone : +62 271 714466
E-mail: politeknik@atmi.ac.id • Website: http://www.atmi.ac.id
Jawa Tengah, Indonesia.

3. Melakukan validasi akurasi OCR secara kuantitatif terhadap ground-truth transkripsi manual pada sampel submission, untuk mengukur tingkat kesalahan baca OCR secara lebih presisi dibanding pendekatan kualitatif (dampak terhadap perubahan nilai) yang digunakan pada penelitian ini.

4. Memperluas cakupan pengujian ke angkatan dan program studi lain di luar Latihan_TMK2026 untuk menguji generalisasi kinerja sistem pada populasi mahasiswa yang lebih beragam.

5. Mengembangkan deteksi kemiripan/plagiarisme antar-submission sebagai modul tambahan, mengingat sistem saat ini berfokus sepenuhnya pada penilaian kesesuaian kriteria dan belum memeriksa orisinalitas karya.

6. Mengevaluasi kelayakan integrasi langsung sebagai plugin Moodle (mengikuti pola arsitektur Virtual Programming Lab [5]) untuk menghindari ketidakstabilan ID submission dan mempercepat siklus evaluasi-umpan balik.

DAFTAR PUSTAKA

[1] D. S. Morris, "Automatically grading Java programming assignments via reflection, inheritance, and regular expressions," in Proc. 32nd Annu. Frontiers in Education Conf. (FIE 2002), 2002, doi: 10.1109/FIE.2002.1157985.

[2] D. S. Morris, "Automatic grading of student's programming assignments: an interactive process and suite of programs," in Proc. 33rd Annu. Frontiers in Education Conf. (FIE 2003), 2003, doi: 10.1109/FIE.2003.1265998.

[3] M. Messer, N. C. C. Brown, M. Kölling, and M. Shi, "Automated Grading and Feedback Tools for Programming Education: A Systematic Review," ACM Trans. Comput. Educ., vol. 24, no. 1, 2024, doi: 10.1145/3636515.

[4] S. H. Edwards and M. A. Perez-Quinones, "Web-CAT: automatically grading programming assignments," ACM SIGCSE Bulletin, vol. 40, no. 3, p. 328, 2008, doi: 10.1145/1597849.1384371.

[5] J. C. Rodríguez-del-Pino, Z. J. Hernández-Figueroa, J. D. González Domínguez, and J. Skalka, "Virtual Programming Lab for Moodle: Automatic Program Assessment in a First-year University Course," in Microlearning, Cham: Springer, 2022, doi: 10.1007/978-3-031-13359-6_12.

[6] Á. Becerra, R. Daza, R. Cobos, A. Morales, M. Cukurova, and J. Fierrez, "M2LADS: A System for Generating MultiModal Learning Analytics Dashboards in Open Education," arXiv:2305.12561, 2023.

[7] R. Purbaningtyas, "Sistem Penilaian Otomatis Modul Pemrograman Java Sederhana," Jurnal Ilmiah Inovasi, vol. 12, no. 1, 2012, doi: 10.25047/jii.v12i1.328.

[8] R. Smith, "An Overview of the Tesseract OCR Engine," in Proc. 9th Int. Conf. Document Analysis and Recognition (ICDAR 2007), vol. 2, 2007, pp. 629-633, doi: 10.1109/ICDAR.2007.4376991.

[9] A. Agrawal and B. Reed, "A Survey on Grading Format of Automated Grading Tools for Programming Assignments," in Proc. 15th Int. Conf. Education, Research and Innovation (ICERI2022), 2022, doi: 10.21125/iceri.2022.1912.

[10] C. Sangwin, "Computer Aided Assessment of Mathematics Using STACK," in Selected Regular Lectures from the 12th International Congress on Mathematical Education, S. Cho, Ed. Cham: Springer, 2015, doi: 10.1007/978-3-319-17187-6_39.

[11] V. Jocović, B. Nikolić, and N. Bacanin, "Software System for Automatic Grading of Paper Tests," Electronics, vol. 12, no. 19, art. 4080, 2023, doi: 10.3390/electronics12194080.

Lampiran

Lampiran 1. Contoh Lengkap CRITERIA — Evaluator Latihan JavaScript3 (Stopwatch)

Kode berikut menunjukkan definisi kriteria penilaian lengkap satu modul evaluator (5 langkah x 20 poin), termasuk penggunaan ketiga mode pencocokan pola.

```
CRITERIA = [
    {
        "step": 1,
        "description": "Struktur HTML dasar + tag <script> atau file .js eksternal",
        "points": 20,
        "patterns": [
            r"<!doctype\s+html|<html[\s>]",
            r"<body[\s>]|<head[\s>]",
            r"<script[\s>]|src\s*=\s*['\"][^'\"]*\.js",
        ],
        "min_matches": 2,
    },
    {
        "step": 2,
        "description": "Tiga tombol kontrol: Start, Stop/Pause, Reset",
        "points": 20,
        "patterns": [r"\bstart\b|\bmulai\b", r"\bstop\b|\bpause\b|\bjeda\b",
r"\breset\b|\bulang\b"],
        "min_matches": 3,
    },
    {
        "step": 3,
        "description": "Timer realtime: setInterval(fn, 1000/100) atau
requestAnimationFrame",
        "points": 20,
        "patterns": [r"setInterval\b[\s\S]{0,500},\s*1000\b",
                     r"setInterval\b[\s\S]{0,200},\s*100\b",
                     r"requestAnimationFrame\s*\("],
        "require_all": False,
    },
    {
        "step": 4,
        "description": "Update tampilan realtime (innerHTML/textContent/innerText)",
        "points": 20,
        "patterns": [r"\.innerHTML\s*=", r"\.textContent\s*=", r"\.innerText\s*="],
        "require_all": False,
    },
    {
        "step": 5,
        "description": "Stop timer (clearInterval/cancelAnimationFrame) + reset nilai ke
0/null",
        "points": 20,
        "patterns": [r"clearInterval\s*\(|cancelAnimationFrame\s*\(",
r"\w\s*=\s*(?:0\b|null\b)"],
        "require_all": True,
    },
]
```

Lampiran 2. Struktur Direktori Proyek

```
Lab3Evaluator/
|-- config/evaluators.json        # registry prefix nomor -> modul evaluator
|-- evaluators/
|   |-- base.py                   # BaseEvaluator, _score_breakdown()
|   |-- latihan_git.py ... latihan_laravel3.py   # 13 modul evaluator
|-- utils/
|   |-- folder_parser.py          # parsing folder Moodle, master list
|   |-- html_extractor.py         # ekstraksi HTML multi-format + GitHub
|   |-- ocr_analyzer.py           # OCR PDF/DOCX via PyMuPDF + Tesseract
|   |-- result_writer.py          # tulis CSV/Excel/JSON
|   |-- web_generator.py          # situs statis inkremental
|-- Files/[Prodi]/[No_Latihan]/[Nama Mhs]_[ID]_assignsubmission_file/
|-- Results/[Prodi]/[No_Latihan]/{hasil_penilaian.csv, .xlsx, .json, index.html}
|-- preprocess.py                 # CLI: ekstrak arsip/GitHub/GDrive ke source/
|-- run_evaluator.py              # entry point CLI evaluasi
```

Lampiran 3. Isi config/evaluators.json (Registry Evaluator)

```
{
  "01": "evaluators.latihan_git",
  "02": "evaluators.latihan_tabel",
  "03": "evaluators.latihan_form",
  "04": "evaluators.latihan_div",
  "05": "evaluators.latihan_bootstrap",
  "06": "evaluators.latihan_javascript1",
  "07": "evaluators.latihan_javascript2",
  "08": "evaluators.latihan_javascript3",
  "09": "evaluators.latihan_javascript_web",
  "10": "evaluators.latihan_final_web",
  "11": "evaluators.latihan_laravel1",
  "12": "evaluators.latihan_laravel2",
  "13": "evaluators.latihan_laravel3"
}
```