

Analisis Gamifikasi *Fill-in-the-Blank* dengan Evaluasi Otomatis *Abstract Syntax Tree* Pada Pembelajaran *Python*

DOI: <http://dx.doi.org/10.35889/jutisi.v14i3.3250>

Creative Commons License 4.0 (CC BY – NC)

Margaretha Violina Putri^{1*}, Usman Nurhasan², Hendra Pradibta³

Sistem Informasi Bisnis, Politeknik Negeri Malang, Malang, Indonesia

*e-mail Corresponding Author: haeretha15@gmail.com

Abstract

This study develops and implements a Fill-in-the-Blank (FIB)-based learning system with gamification and automatic evaluation using Abstract Syntax Tree (AST) in Basic Python Programming learning. The applied methodology combines Research and Development (R&D) to design a prototype and Quasi-Experiment with a pre-test, post-test design on the experimental group (n=26) and control group (n=25). Needs analysis through literature, lecturer interviews, and classroom observations resulted in an interactive system with points, levels, and badges mechanisms. Measurement instruments include AST evaluators, concept understanding tests, motivation questionnaires, and student activity logs. The System Usability Scale (SUS) results averaged 76.47, indicating the system is easy to use and well-accepted. Blackbox testing confirmed that all features function as designed. Quasi-Experiment analysis showed a significant improvement in the experimental group, from Pre-Test 27.45 to Post-Test 92.92 (Paired Sample T-Test, $p = 0.000$) with stable retention in the Delay-Test 81.84 ($p = 0.087$), while the control group only improved limitedly. Independent Sample T-Test also proved a significant difference between the experimental and control groups. These findings demonstrate the effectiveness of the FIB system in improving learning outcomes, comprehension retention, and interactive learning experiences.

Keywords: Abstract Syntax Tree; Fill-in-the-Blank; Gamification; Python Programming; Learning System

Abstrak

Studi ini mengembangkan sistem pembelajaran *Fill-in-the-Blank* (FIB) berbasis gamifikasi dan evaluasi otomatis *Abstract Syntax Tree* (AST) untuk pembelajaran Python dasar. Metode yang digunakan menggabungkan *Research and Development* (R&D) dalam perancangan prototipe dan *Quasi-Experiment* dengan desain *pre-test* dan *post-test* pada kelompok eksperimen (n=26) dan kontrol (n=25). Analisis kebutuhan menghasilkan sistem interaktif dengan fitur poin, level, dan lencana. Instrumen evaluasi meliputi *AST evaluator*, tes konsep, kuesioner motivasi, dan *log aktivitas*. Hasil *System Usability Scale* (SUS) mencapai 76,47, menunjukkan sistem mudah digunakan. Uji *blackbox* memastikan seluruh fitur berfungsi sesuai rancangan. Hasil eksperimen menunjukkan peningkatan signifikan pada kelompok eksperimen dari 27,45 ke 92,92 ($p = 0,000$) dan retensi stabil 81,84 ($p = 0,087$), sementara kelompok kontrol hanya meningkat terbatas. Secara keseluruhan, sistem FIB terbukti efektif meningkatkan hasil belajar, retensi pemahaman, dan pengalaman belajar interaktif.

Kata Kunci: Abstract Syntax Tree; Fill-in-the-Blank; Gamifikasi; Pemrograman Python; Sistem Pembelajaran

1. Pendahuluan

Pemrograman Python saat ini menjadi salah satu kompetensi kunci dalam pendidikan teknologi informasi karena sifatnya yang sederhana, fleksibel, serta banyak diaplikasikan dalam pengembangan perangkat lunak, analisis data, dan kecerdasan buatan [1]. Python bahkan sering dijadikan bahasa pemrograman pertama bagi pemula karena sintaksnya lebih mudah dibandingkan dengan C++ atau Java [2]. Namun, proses pembelajaran Python di tingkat dasar

masih menghadapi berbagai tantangan. Mahasiswa kerap kesulitan memahami struktur program, menerapkan logika ke dalam bentuk kode, serta mempertahankan motivasi belajar ketika pembelajaran lebih banyak bersifat teoritis dan minim interaktivitas [3].

Untuk menjawab tantangan tersebut, telah dikembangkan beberapa platform pembelajaran berbasis *Fill-in-the-Blank Problem* (FIB), seperti C Programming Learning Assistant System (CPLAS) dan Python Programming Learning Assistant System (PYPLAS) [4] [5]. Pendekatan ini efektif bagi pemula karena berfokus pada pengisian potongan kode yang hilang, sehingga mahasiswa dapat memahami alur program tanpa harus menulis keseluruhan kode dari awal [6]. Meskipun demikian, sebagian besar evaluasi jawaban pada *platform* tersebut masih terbatas pada metode *string matching*, yang hanya membandingkan kesesuaian teks. Akibatnya, variasi jawaban dengan sintaks berbeda namun memiliki logika dan hasil eksekusi yang sama sering dianggap salah.

Kesenjangan tersebut menjadi dasar dilakukannya penelitian ini, yaitu perlunya evaluasi otomatis yang lebih akurat dalam pembelajaran pemrograman. Salah satu pendekatan yang potensial adalah pemanfaatan *Abstract Syntax Tree* (AST), yang mampu menganalisis struktur kode berdasarkan makna semantik, bukan sekadar pencocokan string. Berbeda dengan metode *string matching* yang hanya menilai kesamaan karakter atau kata, AST merepresentasikan kode dalam bentuk pohon sintaksis, sehingga dapat mendeteksi kesetaraan logika meskipun sintaks penulisannya bervariasi [7]. Pada Gambar 1 akan ditunjukkan contoh kode program Python.

# Jawaban 1	# Jawaban 2
for i in range(5):	for angka in range(0,5):
print(i)	print(angka)

Gambar 1. Contoh Kode Program Python.

Sebagai ilustrasi, jawaban mahasiswa tersebut akan dianggap berbeda jika dievaluasi menggunakan *string matching*, karena perbedaan nama variabel dan cara penulisan fungsi *range*. Padahal, secara logika, keduanya menghasilkan keluaran yang sama. Dengan pendekatan *Abstract Syntax Tree* (AST), kedua kode tersebut akan direpresentasikan dalam struktur sintaksis yang identik, sehingga dapat dikenali sebagai jawaban yang benar. Hal ini menjadikan AST lebih objektif, fleksibel terhadap variasi gaya penulisan, dan mampu mengurangi *false negative*, yaitu kasus jawaban benar yang salah dinilai akibat perbedaan sintaks. Selain itu, aspek motivasi belajar juga perlu diperhatikan. Penelitian terdahulu menunjukkan bahwa integrasi elemen gamifikasi dapat meningkatkan keterlibatan, motivasi, dan ketekunan mahasiswa dalam pembelajaran. Namun, hingga saat ini, belum terdapat penelitian yang secara khusus mengintegrasikan pendekatan *Fill-in-the-Blank* (FIB), evaluasi otomatis berbasis AST, dan gamifikasi secara terpadu dalam konteks pembelajaran Python [8].

Berdasarkan kesenjangan tersebut, penelitian ini merumuskan pertanyaan utama: bagaimana mengembangkan media pembelajaran berbasis *Fill-in-the-Blank* (FIB) yang terintegrasi dengan gamifikasi serta dilengkapi evaluasi otomatis menggunakan *Abstract Syntax Tree* (AST)? Pertanyaan tersebut dijabarkan menjadi tiga fokus: (1) merancang media pembelajaran Python berbasis FIB dengan elemen gamifikasi, (2) mengimplementasikan evaluasi otomatis berbasis AST untuk menilai jawaban mahasiswa secara objektif, dan (3) menganalisis efektivitas pendekatan ini dalam meningkatkan pemahaman dan keterlibatan mahasiswa.

Tujuan penelitian ini adalah mengembangkan prototipe media pembelajaran pemrograman Python berbasis gamifikasi dengan mekanisme FIB, mengintegrasikan evaluasi otomatis menggunakan AST, serta mengevaluasi dampaknya terhadap pemahaman dan motivasi mahasiswa. Dengan demikian, penelitian ini memberikan kontribusi pada dua aspek utama: pertama, menghadirkan metode evaluasi baru yang lebih andal dibandingkan *string matching*; kedua, memperkaya literatur pembelajaran pemrograman melalui penerapan strategi gamifikasi yang terbukti meningkatkan keterlibatan dan motivasi belajar.

2. Tinjauan Pustaka

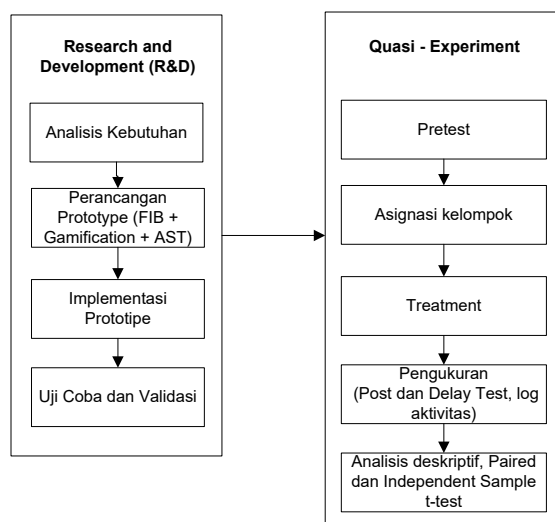
Media pembelajaran berperan penting dalam meningkatkan efektivitas proses belajar karena berfungsi sebagai alat bantu dan sumber belajar yang memudahkan penyampaian materi serta meningkatkan minat dan motivasi siswa [9]. Namun kesulitan memahami konsep dasar pemrograman masih menjadi kendala utama bagi banyak pemula. Penelitian terdahulu

menunjukkan bahwa banyak siswa dikategorikan sebagai *slow learner* karena kesulitan memahami konsep dasar pemrograman yang menuntut kemampuan berpikir logis dan analitis. Multazam mengembangkan model pembelajaran dengan empat komponen utama, yaitu sintaks, prinsip reaksi, sistem sosial, dan sistem pendukung, untuk mengatasi kendala mahasiswa dalam pemrograman web, termasuk kesulitan memahami sintaks (33%), kurangnya mentor (25,6%), dan lemahnya pemahaman logika serta algoritma (20,5%) [10]. Penelitian lain memperkenalkan aplikasi PseudoLearn berbasis *drag-and-drop* untuk membangun pseudocode, yang dilengkapi log data guna merekam aktivitas belajar mahasiswa. Hasil penelitian mereka menunjukkan adanya peningkatan skor post-test pada kelompok eksperimen dibandingkan kelompok kontrol, dengan analisis ANOVA yang mengonfirmasi pengaruh signifikan metode tersebut terhadap pemahaman mahasiswa [11]. Dalam upaya lain, Pgtracer, sebuah alat berbasis web untuk soal *fill-in-the-blank* dengan C++ dan *trace table*, yang tidak hanya menguji sintaksis tetapi juga alur eksekusi program. Fitur analisis log pada alat ini membantu mengidentifikasi kesalahan siswa sekaligus memberikan umpan balik yang bermanfaat, serta terbukti efektif dalam meningkatkan pemahaman konsep dasar pemrograman dan meringankan beban pengajar [12]. Sementara itu, Othman menekankan pentingnya gamifikasi untuk mendukung siswa berprestasi rendah dalam kursus pemrograman. Melalui elemen aturan, tantangan, *progress bar*, poin, dan rencana, gamifikasi terbukti dapat meningkatkan motivasi belajar dan menurunkan tingkat kegagalan [8]. Secara keseluruhan, berbagai penelitian ini menegaskan bahwa inovasi teknologi dalam pembelajaran pemrograman mampu memberikan dampak positif pada peningkatan motivasi, pemahaman, dan hasil belajar siswa. Hasil penelitian Gumilar juga menunjukkan bahwa pemanfaatan sistem pembelajaran berbasis web mampu mendukung fleksibilitas belajar, sehingga peserta didik dapat mengakses materi pembelajaran kapan saja dan di mana saja [13].

Berbeda dari penelitian terdahulu yang umumnya memisahkan aspek gamifikasi dan penilaian otomatis, penelitian ini mengintegrasikan keduanya ke dalam satu sistem pembelajaran berbasis *Fill-in-the-Blank*. Model yang diusulkan memanfaatkan analisis AST untuk mengevaluasi struktur dan makna kode Python secara otomatis, disertai elemen gamifikasi seperti poin, tantangan, dan indikator progres untuk meningkatkan motivasi belajar. Pendekatan ini juga melibatkan analisis log aktivitas siswa guna mengaitkan performa belajar dengan metrik permainan, sehingga menghasilkan sistem evaluasi yang adaptif, semantik, dan berorientasi motivasi—yang menjadi kebaruan utama penelitian ini.

3. Metodologi

Penelitian ini berfokus pada pengembangan media pembelajaran dan pengujian efektivitasnya. Metodologi yang digunakan adalah *Research and Development* (R&D) untuk menghasilkan prototipe, dipadukan dengan *Quasi-Experiment* pada tahap evaluasi guna menilai dampaknya terhadap pemahaman dan motivasi belajar mahasiswa. Pada Gambar 2 akan ditampilkan metodologi yang digunakan dalam penelitian ini.



Gambar 2. Metodologi R&D dan *Quasi Experiment*.

3.1 Research and Development

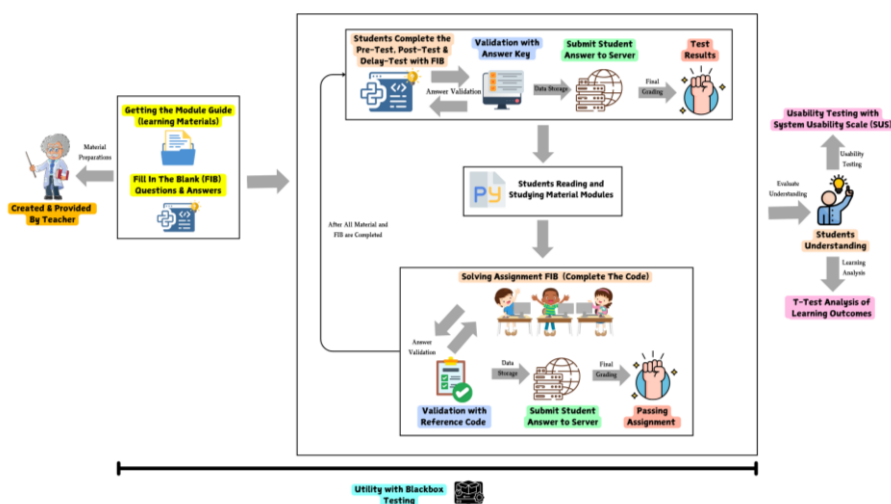
1) Analisis Kebutuhan

Analisis kebutuhan dilakukan melalui studi literatur, wawancara dengan dosen, dan observasi kelas pemrograman untuk mengidentifikasi kesulitan mahasiswa dalam memahami sintaks Python serta keterbatasan media pembelajaran interaktif. Berdasarkan temuan ini, dirancang sistem *Fill-in-the-Blank* (FIB) yang dilengkapi dengan gamifikasi dan evaluasi otomatis berbasis *Abstract Syntax Tree* (AST). Penelitian menggunakan pendekatan *Quasi-Experiment* dengan desain *Pre-Test* dan *Post-Test* pada kelompok eksperimen dan kontrol. Instrumen pengukuran meliputi: AST Evaluator untuk menilai akurasi jawaban kode Python, tes pemahaman konsep Python, kuesioner pengguna, dan log aktivitas belajar pengguna. Pada Gambar 3 ditunjukkan alur sistem dari penelitian yang dilakukan, sedangkan pada Tabel 1 ditunjukkan kebutuhan fungsional dan non fungsional dari sistem yang dibuat.

Tabel 1. Kebutuhan Fungsional dan Non Fungsional

Kode	Jenis Kebutuhan	Deskripsi Kebutuhan Sistem	Tujuan / Hasil yang Diharapkan
F1	Fungsional	Sistem dapat menampilkan soal <i>Fill-in-the-Blank</i> berisi potongan kode Python yang harus diisi pengguna.	Mahasiswa dapat berlatih melengkapi sintaks Python dengan konteks yang benar.
F2	Fungsional	Sistem melakukan evaluasi otomatis terhadap jawaban pengguna menggunakan Abstract Syntax Tree (AST) untuk membandingkan struktur kode dengan jawaban kunci.	Penilaian lebih akurat secara semantik, bukan hanya berbasis output.
F3	Fungsional	Sistem mencatat log aktivitas pengguna , seperti waktu pengerjaan, jumlah percobaan, dan hasil evaluasi tiap percobaan.	Data log dapat dianalisis untuk memahami pola belajar mahasiswa.
F4	Fungsional	Sistem memberikan umpan balik otomatis berdasarkan hasil evaluasi AST (misalnya kesalahan sintaks, urutan, atau struktur).	Pengguna memahami letak kesalahan dan dapat memperbaikinya secara mandiri.
F5	Fungsional	Sistem mengimplementasikan elemen gamifikasi seperti poin, tantangan, lencana, dan <i>progress bar</i> .	Meningkatkan motivasi dan keterlibatan mahasiswa dalam pembelajaran.
F6	Fungsional	Sistem menampilkan laporan hasil belajar meliputi nilai, tingkat keberhasilan, serta durasi pengerjaan.	Dosen dan mahasiswa dapat memantau perkembangan belajar secara terukur.
F7	Fungsional	Admin/dosen dapat menambah, mengubah, atau menghapus soal dan data latihan pada sistem.	Menjamin fleksibilitas dalam pengelolaan konten pembelajaran.
NF1	Nonfungsional	Sistem memiliki antarmuka yang responsif dan mudah digunakan di perangkat desktop maupun mobile.	Meningkatkan kenyamanan dan aksesibilitas pengguna.
NF2	Nonfungsional	Waktu respon evaluasi AST tidak lebih dari 3 detik untuk setiap pengujian kode.	Menjamin performa sistem yang efisien.
NF3	Nonfungsional	Sistem mampu menyimpan data secara konsisten dan mencegah kehilangan hasil evaluasi/log aktivitas.	Menjaga integritas dan keandalan data pengguna.

Alur eksperimen dimulai dari penyusunan modul dan pertanyaan FIB beserta jawaban referensinya, dilanjutkan dengan *Pre-Test*, pembelajaran mandiri, dan penyelesaian tugas FIB yang divalidasi otomatis. Setelah itu, dilakukan *Post-Test* dan *Delay-Test* untuk mengevaluasi peningkatan pemahaman. Evaluasi sistem mencakup: (1) Efektivitas pembelajaran, dianalisis menggunakan *paired sample T-Test*; (2) *Usability*, diuji dengan *System Usability Scale* (SUS); (3) Fungsionalitas, diuji melalui *blackbox testing* untuk memastikan kesesuaian sistem dengan spesifikasi. Hasil analisis ini menjadi dasar untuk rekomendasi pengembangan sistem yang meningkatkan efektivitas dan pengalaman belajar interaktif.



Gambar 3. Alur Eksperimen dan Penelitian.

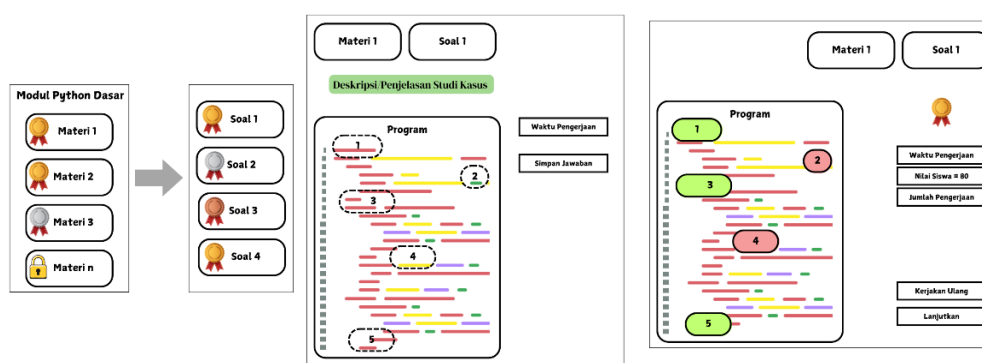
Pada Tabel 2 Ditunjukkan kompetensi yang akan digunakan dalam fitur pembelajaran berbasis *Fill-in-the-Blank*.

Table 2. Kompetensi Materi *Fill-in-the-Blank*

Materi	Judul	Kode Unit	Kompetensi Inti
1	Pengenalan Python & Interpreter	J.620100.010.02, J.620100.011.01	Memahami konsep dasar pemrograman, instalasi tools, sintaks dasar & eksekusi kode.
2	Tipe Data & Operasi Dasar	J.620100.022.02	Menggunakan tipe data, operasi dasar, manipulasi string, dan ekspresi Python.
3	Kontrol Flow (Percabangan)	J.620100.017.02	Membuat program dengan percabangan if, if-else, if-elif-else, dan nested if.
4	Control Flow (Perulangan)	J.620100.017.02	Menggunakan perulangan for, while, break, continue, dan nested loop.
5	Struktur Data (List & Tuple)	J.620100.004.01	Menggunakan List & Tuple serta membedakan karakteristik keduanya.
6	Struktur Data Lanjut (Set & Dictionary)	J.620100.004.01	Menggunakan Set untuk data unik & Dictionary dengan konsep key-value.
7	Fungsi	J.620100.022.02	Membuat & memanggil fungsi dengan parameter dan return value.
8	OOP Dasar (Classes)	J.620100.018.02	Membuat class, object, inheritance & overriding pada Python.
9	Error Handling & Debugging	J.620100.025.02	Menangani error dengan try-except & melakukan debugging.

2) Perancangan Prototipe

Prototipe sistem dirancang untuk meningkatkan keterlibatan dan pemahaman Python melalui soal *Fill-in-the-Blank* (FIB) yang memungkinkan mahasiswa mengisi bagian kode yang kosong. Mahasiswa memilih *Course* Pemrograman Dasar Python, mempelajari materi yang disiapkan pengajar, dan menyelesaikan soal FIB yang divalidasi otomatis menggunakan *Abstract Syntax Tree* (AST) dengan bobot 70% struktur kode dan 30% *output*. Setiap soal dapat dikerjakan berulang kali, dengan pencatatan waktu pengerjaan, skor, dan jumlah percobaan untuk mendukung analisis perilaku belajar. Sistem juga menyertakan mekanisme gamifikasi berupa poin, *level*, dan *badges* (emas, perak, perunggu) untuk meningkatkan motivasi mahasiswa. Setelah memenuhi kriteria modul, mahasiswa melanjutkan ke *Post-Test* FIB. Prototipe menyajikan *real-time feedback* berupa skor dan indikator bagian kode yang benar atau salah, serta studi kasus kontekstual untuk setiap modul. Integrasi ketiga komponen utama FIB, gamifikasi, dan AST Evaluator menjamin pengalaman belajar yang interaktif, terstruktur, dan terukur. Pada Gambar 4 ditunjukkan Struktur tampilan interaksi soal pada fitur *Fill-in-the-Blank*.



Gambar 4. Struktur Tampilan Interaksi Soal Pada Fitur Fill-in-the-Blank

3) Implementasi Prototipe

Prototipe sistem dikembangkan berbasis *website* untuk meningkatkan aksesibilitas, dengan fitur utama berupa penyajian materi, soal interaktif *Fill-in-the-Blank* (FIB), umpan balik otomatis, serta pencatatan aktivitas mahasiswa mencakup waktu pengerjaan, skor, dan jumlah percobaan. Setelah tahap pengembangan selesai, sistem diimplementasikan dengan melibatkan 51 mahasiswa Jurusan Teknologi Informasi Politeknik Negeri Malang. Tahap implementasi ini bertujuan untuk memastikan aplikasi dapat diakses dan dioperasikan dengan baik, sekaligus memverifikasi kesesuaian fungsionalitas serta performa sistem terhadap ekspektasi pengguna.

4) Ujicoba dan Validasi

Ujicoba dan validasi sistem dilakukan dengan mengacu pada Standar Kompetensi Kerja Nasional Indonesia (SKKNI) sebagai kerangka penilaian pembelajaran vokasi. SKKNI memuat aspek pengetahuan, keterampilan, dan sikap kerja yang relevan dengan pendidikan vokasi, pelatihan kerja, serta sertifikasi kompetensi. Standar ini digunakan sebagai acuan dalam penyusunan instrumen pengujian dan penilaian, sehingga hasil implementasi prototipe dapat dievaluasi secara objektif berdasarkan indikator kompetensi yang berlaku.

Data yang digunakan untuk proses validasi terdiri atas tiga kategori utama. Pertama, data modul pembelajaran berupa tingkat kesulitan materi (mudah, sedang, sulit) yang dianalisis untuk menilai pengaruh kompleksitas materi terhadap pemahaman mahasiswa dalam mengerjakan soal *Fill-in-the-Blank*. Kedua, data hasil belajar yang meliputi nilai *Pre-Test*, *Post-Test*, dan *Delay-Test*, skor per soal dalam setiap modul, waktu pengerjaan, serta frekuensi percobaan. Data ini bersifat numerik dan menjadi indikator utama dalam mengukur efektivitas pembelajaran, efisiensi waktu, serta ketekunan mahasiswa dalam menyelesaikan soal. Ketiga, data mahasiswa, yang mencakup informasi demografis seperti tingkat kelas, pengalaman pemrograman, serta latar belakang akademik, serta data pembandingan antara kelompok kontrol (pembelajaran konvensional) dan kelompok eksperimen (dengan fitur *Fill-in-the-Blank*).

Hasil ujicoba dianalisis menggunakan *Paired Sample T-Test* untuk mengukur perbedaan hasil belajar sebelum dan sesudah intervensi, serta uji komparatif antara kelompok kontrol dan eksperimen untuk menilai signifikansi pengaruh fitur *Fill-in-the-Blank*. Validasi sistem juga mencakup aspek fungsionalitas dan performa agar sesuai dengan kebutuhan pengguna, serta mengacu pada SKKNI sebagai standar kompetensi nasional. Pada Tabel 3 ditunjukkan data yang akan digunakan dalam fitur pembelajaran berbasis *Fill in the Blank*.

Table 3. Pemahaman Data

Sumber Data	Data yang Diambil	Keterangan
Modul Pembelajaran	Tingkat kesulitan materi (<i>Mudah, Sedang, Sulit</i>)	Kategorikal – Data berupa kategori tingkat kesulitan. Digunakan untuk menilai pengaruh kompleksitas materi terhadap pemahaman dan hasil belajar.
Hasil Belajar	Nilai mahasiswa (<i>Pre-Test, Post-Test, Delay-Test</i>)	Numerik (0–100) – Data skor hasil belajar. Digunakan untuk mengukur peningkatan hasil belajar melalui evaluasi berbasis FIB.
Hasil Belajar	Skor setiap soal per modul	Numerik (0–100) – Data skor detail per soal. Digunakan untuk evaluasi terperinci dengan validasi AST (struktur & output).

Sumber Data	Data yang Diambil	Keterangan
Hasil Belajar	Waktu pengerjaan	Numerik (detik/menit) – Data berupa lama pengerjaan soal. Digunakan untuk analisis efisiensi mahasiswa dalam menyelesaikan soal.
Hasil Belajar	Jumlah pengerjaan	Numerik (frekuensi) – Data berupa jumlah percobaan. Digunakan untuk mencatat intensitas mahasiswa dalam menyelesaikan soal.
Data Mahasiswa	Data demografis (kelas, pengalaman, latar belakang akademik)	Kategorikal – Data berupa klasifikasi demografis mahasiswa. Digunakan untuk identifikasi profil dan analisis korelatif.
Data Mahasiswa	Data kelompok kontrol dan eksperimen	Kategorikal – Data berupa pembagian kelompok pembelajaran. Digunakan untuk membandingkan efektivitas pembelajaran konvensional dan berbasis FIB.

3.2 Quasi Experiment

1) Pre Test

Pre-Test dilaksanakan untuk mengukur kemampuan awal mahasiswa pada materi pemrograman dasar Python. Dari total 51 peserta, mayoritas memperoleh skor rendah (0), menandakan keterbatasan pemahaman awal. Namun, beberapa mahasiswa mencapai skor tinggi hingga 100, yang menunjukkan adanya variasi tingkat kesiapan. Distribusi nilai terbagi ke dalam tiga *level*, yaitu :

- Low*: mayoritas dengan skor 0;
- Medium*: rentang 65,88–70, menandakan pemahaman dasar mulai terbentuk.
- High*: skor $\geq 97,94$ hingga 100, menunjukkan penguasaan materi yang sangat baik.

Hasil ini menegaskan adanya heterogenitas kemampuan awal, sehingga stratifikasi perlu dipertimbangkan pada analisis berikutnya untuk menjaga validitas eksperimen.

Tabel 4. Hasil Pre Test Mahasiswa

Strata	Rentang Nilai	Jumlah Mahasiswa	Persentase
Low	0 – 30	36	70,60%
Medium	65,88 – 70	8	15,70%
High	$\geq 97,94$ – 100	7	13,70%
Total	-	51	100%

2) Asignasi Kelompok

Mahasiswa dibagi menjadi dua kelompok, yaitu kontrol (25 mahasiswa) dan eksperimen (26 mahasiswa), dengan mempertimbangkan tingkat kemampuan awal yang dikategorikan ke dalam dua stratum: *high* dan *low*. Pembagian ini bertujuan untuk menjaga keseimbangan karakteristik awal antar kelompok sehingga hasil eksperimen dapat dianalisis secara valid tanpa bias kemampuan awal [14]. Tabel 5 menyajikan distribusi mahasiswa berdasarkan stratum dan kelompok. Pada stratum *high*, terdapat 8 mahasiswa di kelompok kontrol dan 7 mahasiswa di kelompok eksperimen. Sedangkan pada stratum *low*, kelompok kontrol terdiri dari 17 mahasiswa dan kelompok eksperimen terdiri dari 19 mahasiswa. Dominasi jumlah mahasiswa pada stratum *low* menunjukkan bahwa sebagian besar peserta penelitian memiliki tingkat pemahaman awal yang masih rendah terhadap pemrograman Python. Kondisi ini mencerminkan profil nyata mahasiswa pada mata kuliah pemrograman dasar, sehingga hasil intervensi sistem pembelajaran dapat lebih menggambarkan efektivitasnya terhadap mayoritas pengguna.

Tabel 5. Distribusi Stratum Kontrol dan Eksperimen

Strata	Kontrol	Eksperimen	Total
High	8	7	15
Low	17	19	36
Total	25	26	51

3) *Treatment*

Pada tahap *treatment*, masing-masing kelompok diberikan perlakuan berbeda untuk melihat pengaruh intervensi:

a. Kelompok eksperimen (26 mahasiswa)

Menggunakan sistem pembelajaran berbasis web yang dikembangkan dengan fitur *Fill-in-the-Blank* (FIB), evaluasi otomatis melalui *Abstract Syntax Tree* (AST Evaluator), serta pencatatan aktivitas mahasiswa. Sistem ini dirancang untuk memberikan umpan balik instan, meningkatkan interaktivitas, dan mendorong keterlibatan aktif dalam memahami konsep pemrograman Python.

b. Kelompok kontrol (25 mahasiswa)

Mendapatkan pembelajaran dengan metode konvensional melalui perkuliahan tatap muka dan latihan soal tanpa dukungan sistem otomatis.

Durasi perlakuan dilakukan selama beberapa sesi perkuliahan dengan cakupan materi pemrograman dasar Python. Tahap ini bertujuan untuk menguji apakah penggunaan sistem dengan evaluasi otomatis memberikan dampak signifikan terhadap peningkatan pemahaman, motivasi, serta efektivitas pembelajaran jika dibandingkan dengan metode tradisional.

4) Pengukuran *Post Test*, *Delay Test*, dan Log Aktivitas

Pengukuran dilakukan melalui beberapa instrumen untuk menilai efektivitas perlakuan. *Post-Test* diberikan untuk mengukur peningkatan pemahaman konsep Python setelah intervensi. Selain itu, *delay-test* digunakan untuk mengevaluasi retensi pengetahuan mahasiswa setelah jangka waktu tertentu [15]. Aspek motivasi mahasiswa diamati melalui kuesioner motivasi, yang menilai dimensi motivasi intrinsik, ekstrinsik, dan keterlibatan [16]. Sementara itu, log aktivitas sistem dikumpulkan untuk menganalisis intensitas dan pola keterlibatan mahasiswa selama proses pembelajaran. Pada Tabel 6 ditunjukkan metode analisis materi yang akan digunakan untuk pengukuran *Post-Test*, *Delay-Test* dan log aktivitas mahasiswa.

Tabel 6. Pengukuran *Post-Test*, *Delay-Test*, dan Log Aktivitas

Instrumen	Indikator Utama	Metode Analisis
<i>Pre-Test</i>	Skor pemahaman dasar sebelum perlakuan	Statistik deskriptif, uji kesetaraan kelompok
<i>Post-Test</i>	Skor pemahaman konsep setelah perlakuan	Paired sample t-test, uji perbedaan kelompok
<i>Delay test</i>	Skor konsistensi pemahaman setelah interval waktu	Analisis komparatif (<i>Post-Test</i> vs <i>delay test</i>)
Kuesioner motivasi	Aspek motivasi intrinsik, ekstrinsik, dan keterlibatan	Analisis deskriptif, uji reliabilitas (Cronbach's Alpha)
Log aktivitas	Frekuensi interaksi, durasi penggunaan, tingkat keberhasilan	Analisis learning analytics, visualisasi pola aktivitas

5) Analisis *T-Test*

Analisis data dilakukan untuk mengevaluasi efektivitas prototipe melalui beberapa tahapan. Tahapan analisis data dalam penelitian ini mencakup tiga prosedur utama, yaitu Analisis Deskriptif, Paired Sample T-Test, dan Independent Sample t-Test. Analisis deskriptif digunakan untuk menggambarkan distribusi skor dan kecenderungan hasil belajar peserta. Selanjutnya, *Paired Sample T-Test* diterapkan untuk menguji perbedaan signifikan antara skor *Pre-Test* dan *Post-Test* dalam kelompok yang sama [17]. Sementara itu, *Independent Sample T-Test* digunakan untuk membandingkan hasil belajar antara kelompok eksperimen dan kontrol [18]. Langkah pertama dalam analisis adalah menentukan hipotesis, dengan hipotesis nol (H_0) menyatakan bahwa kedua varian sama, sedangkan hipotesis alternatif (H_a) menyatakan bahwa kedua varian berbeda. Selanjutnya, kriteria pengujian ditetapkan berdasarkan probabilitas, di mana H_0 diterima jika p-value > 0,05 dan H_a diterima jika p-value < 0,05 [19]. Kesimpulan penelitian kemudian diambil berdasarkan kriteria pengujian tersebut. Dengan penerapan yang tepat, uji T (*T-Test*) menjadi alat yang sangat berguna dalam analisis data penelitian untuk

mendukung pengambilan keputusan berdasarkan signifikansi statistik. Untuk lebih memperjelas, pada Tabel 7 ditunjukkan tahapan analisis yang digunakan dalam penelitian ini.

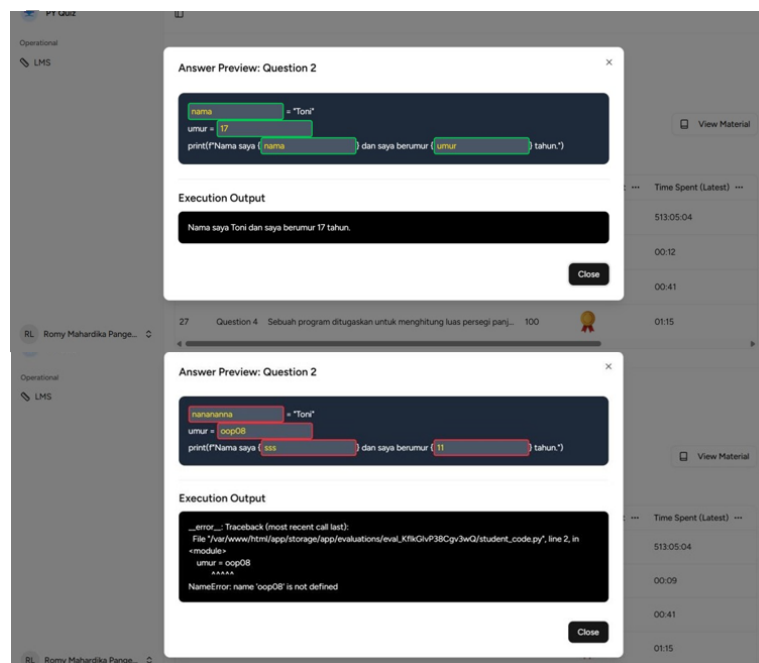
Tabel 7. Tahap Analisis dan Instrumen Penelitian

Tahapan Analisis	Instrumen/Data yang Digunakan	Metode Uji Statistik & Rumus
Analisis Deskriptif	Nilai <i>Pre-Test</i> , <i>Post-Test</i> , kuesioner motivasi, log	Mean, standar deviasi, distribusi frekuensi
Paired Sample T-Test	Nilai <i>Pre-Test</i> dan <i>Post-Test</i> mahasiswa	$t = \frac{\bar{D}}{\frac{SD}{\sqrt{n}}}$
Independent Sample T-Test	Nilai <i>Post-Test</i> kedua kelompok	$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$

4. Hasil dan Pembahasan

4.1 Research and Development

Antarmuka pada fitur Answer Preview dan Execution Output yang disediakan oleh sistem pembelajaran interaktif Fill-in-the-Blank (FIB) memungkinkan mahasiswa untuk melihat kembali jawaban yang telah diisiikan beserta hasil eksekusinya, baik yang berhasil maupun yang mengandung kesalahan. Fitur ini sangat membantu siswa dalam memahami letak kesalahan mereka, karena sistem tidak hanya menampilkan output akhir tetapi juga mengaitkannya langsung dengan bagian kode yang bermasalah. Dengan adanya tampilan perbandingan antara jawaban siswa dan jawaban kunci, siswa dapat melakukan evaluasi diri secara langsung dan memperbaiki pemahamannya terhadap sintaks maupun logika program.



Gambar 5. Tampilan Penilaian Jawaban Siswa

Pengujian *System Usability Scale* (SUS) juga dilakukan untuk menilai kenyamanan dan kemudahan penggunaan sistem dari perspektif pengguna [20]. Kuesioner 10 pernyataan diberikan kepada 51 mahasiswa setelah *Post-Test*, dengan skor dikonversi sesuai prosedur standar SUS. Hasil pengolahan menunjukkan skor rata-rata 76,47, termasuk kategori “Good” (kelas B) dan lebih tinggi dari ambang rata-rata umum 68. Hal ini menandakan sistem pembelajaran berbasis *Fill-in-the-Blank* (FIB) mudah digunakan dan diterima baik oleh mahasiswa. Distribusi skor SUS mahasiswa ditampilkan dalam Tabel 8.

Tabel 8. Distribusi Skor SUS

Rentang Skor SUS	Jumlah Responden	Skor Responden
40–49	1	45
50–59	6	50, 52.5, 55, 55, 57.5, 57.5
60–69	3	65, 65, 67.5
70–79	7	70, 70, 72.5, 75, 75, 77.5, 77.5
80–89	10	80, 80, 82.5, 82.5, 82.5, 85, 85, 85, 87.5, 87.5
90–100	8	90, 90, 90, 92.5, 92.5, 95, 95, 100, 100, 100

Pengujian *blackbox* dilakukan untuk mengevaluasi fungsionalitas sistem secara keseluruhan tanpa mempertimbangkan struktur internal kode atau mekanisme implementasi di baliknya. Pendekatan ini menekankan aplikabilitas sistem dari perspektif pengguna akhir, sehingga dapat menilai apakah sistem memenuhi spesifikasi dan kebutuhan fungsional yang telah ditetapkan [21]. Dalam penelitian ini, pengujian *blackbox* dilakukan pada tiga *level* pengguna utama, yaitu: Mahasiswa, Pengajar, dan Admin. Setiap *level* pengguna memiliki skenario penggunaan yang berbeda, sesuai dengan peran dan akses yang dimiliki. Tujuan pengujian adalah untuk memastikan integritas fungsional sistem, kehandalan antarmuka, dan konsistensi output pada setiap *level* pengguna. Hasil pengujian disajikan secara ringkas untuk masing-masing *level* pengguna, menggambarkan kesesuaian sistem dengan spesifikasi fungsional yang diharapkan, serta mengidentifikasi area yang memerlukan perbaikan atau penyesuaian lebih lanjut. Dengan demikian, pengujian *blackbox* pada Tabel 9 ini menjadi tolak ukur awal efektivitas sistem dalam mendukung proses pembelajaran dan administrasi secara nyata.

Tabel 9. Hasil Pengujian *Blackbox*

Pengguna	Fitur Diuji	Hasil
Mahasiswa	Course, Modul, FIB, Tes, Hasil	Valid
Pengajar	Manajemen Konten & Soal FIB Otomatis	Valid
Admin	Manajemen Data & Laporan	Valid

Hasil SUS menunjukkan tingkat *usability* baik dan diterima mahasiswa, sementara pengujian *blackbox* menegaskan bahwa sistem berfungsi sesuai desain untuk semua pengguna. Fitur pembuatan soal FIB otomatis memudahkan pengajar, sedangkan mahasiswa mendapatkan feedback akurat. Distribusi skor SUS yang sebagian besar berada di atas 68 mengindikasikan penerimaan sistem yang luas. Kombinasi hasil ini menegaskan bahwa sistem pembelajaran berbasis FIB yang dikembangkan siap digunakan secara efektif dan mendukung proses belajar-mengajar.

4.2 Quasi Experiment

Analisis statistik dilakukan untuk mengevaluasi efektivitas sistem pembelajaran berbasis *Fill-in-the-Blank* (FIB) terhadap hasil belajar dan retensi pemahaman siswa. Pendekatan yang diterapkan bersifat parametrik, meliputi analisis deskriptif untuk menyajikan ringkasan nilai *Pre-Test*, *Post-Test*, dan *Delay-Test*, serta analisis inferensial untuk menilai perbedaan antar kelompok dan perubahan dalam satu kelompok. *Independent Sample T-Test* digunakan untuk membandingkan rata-rata antar kelompok yang tidak saling berhubungan, sedangkan *Paired Sample T-Test* digunakan untuk mengukur perubahan hasil belajar dalam satu kelompok sebelum dan sesudah intervensi. Pendekatan ini memberikan dasar empiris yang kuat untuk menilai efektivitas sistem FIB terhadap performa akademik dan daya ingat jangka menengah siswa.

Hasil analisis deskriptif menunjukkan bahwa rata-rata *Pre-Test* kedua kelompok relatif setara, menegaskan kesetaraan kondisi awal sebelum perlakuan. *Post-Test* dan *Delay-Test* kelompok eksperimen menunjukkan peningkatan yang signifikan, sedangkan kelompok kontrol hanya mengalami peningkatan terbatas dengan distribusi nilai yang lebih variatif. Hal ini menegaskan peran sistem FIB dalam meningkatkan hasil belajar secara konsisten dan mempertahankan retensi pemahaman dalam jangka waktu tertentu.

Tabel 10. Ringkasan Analisis Statistik dengan Indikator Signifkansi

Kelompok	Tes	N	Mean	t	p	Jenis Uji
Eksperimen	Pre-Test	26	27,45	-	0,963	Independent Sample T-Test
Kontrol	Pre-Test	25	27,99	0,046		
Eksperimen	Post-Test	26	92,92	4,488	0	Independent Sample T-Test
Kontrol	Post-Test	25	49,68			
Eksperimen	Delay-Test	26	81,84	3,387	0,001	Independent Sample T-Test
Kontrol	Delay-Test	25	49,52			
Eksperimen	Pre → Post	26	27,45 → 92,92	7,342	0	Paired Sample T-Test
Eksperimen	Post → Delay	26	92,92 → 81,84	1,78	0,087	Paired Sample T-Test
Kontrol	Pre → Post	25	27,99 → 49,68	2,019	0,055	Paired Sample T-Test
Kontrol	Post → Delay	25	49,68 → 49,52	0,014	0,989	Paired Sample T-Test

Pada kelompok eksperimen, nilai rata-rata *Pre-Test* sebesar 27,45 meningkat secara signifikan menjadi 92,92 pada *Post-Test*, disertai penurunan simpangan baku dari 39,48 menjadi 13,20. Penurunan simpangan baku ini menunjukkan bahwa intervensi melalui sistem FIB tidak hanya meningkatkan performa belajar secara umum, tetapi juga meratakan distribusi hasil belajar antar siswa, sehingga variasi pemahaman menjadi lebih homogen. *Delay-Test* satu minggu setelah *Post-Test* menunjukkan rata-rata 81,84, menandakan retensi pemahaman tetap baik meskipun sedikit menurun. Sebaliknya, kelompok kontrol menunjukkan pola yang berbeda. Nilai rata-rata *Pre-Test* sebesar 27,99 hanya meningkat terbatas menjadi 49,68 pada *Post-Test*, dengan simpangan baku yang tetap tinggi. Nilai *Delay-Test* tercatat 49,52, menandakan retensi pemahaman yang relatif rendah. Hal ini mengindikasikan bahwa metode pembelajaran konvensional memiliki kemampuan terbatas dalam meningkatkan hasil belajar siswa dan tidak mampu menstabilkan distribusi nilai antar peserta, sehingga variasi pemahaman tetap tinggi.

Pada hasil *Independent Sample T-Test* menunjukkan tidak adanya perbedaan signifikan pada nilai *Pre-Test* ($p = 0,963$), yang menegaskan bahwa kondisi awal kedua kelompok setara. Namun, analisis *Post-Test* ($p = 0,000$) dan *Delay-Test* ($p = 0,001$) menunjukkan perbedaan signifikan antara kelompok eksperimen dan kontrol, menegaskan efektivitas sistem FIB dalam meningkatkan hasil belajar sekaligus mempertahankan pemahaman siswa dibandingkan metode konvensional. Analisis *Paired Sample T-Test* pada kelompok eksperimen menunjukkan peningkatan signifikan dari *Pre-Test* ke *Post-Test* ($p = 0,000$), sedangkan perbandingan *Post-Test* ke *Delay-Test* tidak signifikan ($p = 0,087$). Temuan ini menegaskan bahwa materi yang diberikan melalui sistem FIB memiliki retensi yang stabil dalam jangka menengah. Pada kelompok kontrol, tidak terdapat peningkatan signifikan baik dari *Pre-Test* ke *Post-Test* ($p = 0,055$) maupun *Post-Test* ke *Delay-Test* ($p = 0,989$) yang mengindikasikan efektivitas metode konvensional terbatas dalam meningkatkan performa dan retensi pemahaman siswa.

4.3 Pembahasan

Hasil *Quasi Experiment* menunjukkan bahwa sistem pembelajaran berbasis *Fill-in-the-Blank* (FIB) secara signifikan meningkatkan hasil belajar, retensi pemahaman, dan konsistensi performa siswa dibandingkan metode konvensional. Temuan ini mendukung efektivitas pendekatan pembelajaran berbasis gamifikasi yang diintegrasikan dengan evaluasi otomatis menggunakan Abstract Syntax Tree (AST), serta memperlihatkan kesesuaian implementasi sistem terhadap Standar Kompetensi Kerja Nasional Indonesia (SKKNI). SKKNI memuat tiga dimensi utama kompetensi, yaitu pengetahuan (knowledge), keterampilan (skills), dan sikap kerja (work attitude), yang menjadi dasar penilaian dalam pendidikan vokasi dan pelatihan berbasis kompetensi [12]. Dalam konteks penelitian ini, sistem FIB dikembangkan dengan mempertimbangkan ketiga aspek tersebut sebagai berikut:

- 1) Aspek Pengetahuan – diuji melalui nilai *Pre-Test*, *Post-Test*, dan *Delay-Test* untuk menilai tingkat pemahaman sintaks dan logika pemrograman Python.
- 2) Aspek Keterampilan – diukur menggunakan evaluasi *Abstract Syntax Tree* (AST) yang menilai struktur kode secara otomatis, memastikan kesesuaian logika dan sintaks.

- 3) Aspek Sikap Kerja – ditinjau dari log aktivitas mahasiswa, termasuk waktu pengerjaan, frekuensi percobaan, serta konsistensi latihan, yang mencerminkan kedisiplinan dan ketekunan dalam proses belajar.

Keterkaitan antara hasil uji FIB dan aspek kompetensi SKKNI dirangkum dalam Tabel 11 berikut.

Tabel 11. Keterkaitan Hasil Uji dan Aspek Kompetensi SKKNI

Aspek SKKNI	Indikator Pengukuran / Data Analisis	Hasil Uji / Temuan Statistik	Interpretasi terhadap Kompetensi Vokasi
Pengetahuan (Knowledge)	Nilai <i>Pre-Test</i> , <i>Post-Test</i> , dan <i>Delay-Test</i>	Peningkatan signifikan pada kelompok eksperimen ($p = 0,000$) dibanding kontrol ($p > 0,05$). Rata-rata meningkat dari 27,45 → 92,92 → 81,84.	Menunjukkan penguasaan konsep sintaks dan logika Python meningkat signifikan setelah menggunakan sistem FIB. Kompetensi kognitif mahasiswa meningkat sesuai standar SKKNI bidang pemrograman dasar.
Keterampilan (Skills)	Evaluasi <i>Abstract Syntax Tree (AST)</i> terhadap struktur kode	Tingkat ketepatan sintaks dan logika kode mahasiswa meningkat dengan simpangan baku yang lebih kecil pada kelompok eksperimen.	Menggambarkan kemampuan menerapkan teori ke praktik. Evaluasi AST memastikan keterampilan teknis sesuai standar industri dan SKKNI untuk kemampuan menulis program terstruktur.
Sikap Kerja (Work Attitude)	Waktu pengerjaan, jumlah percobaan, dan konsistensi log aktivitas	Rata-rata waktu penyelesaian lebih efisien dan jumlah percobaan lebih stabil pada kelompok eksperimen.	Menunjukkan peningkatan disiplin, ketekunan, dan tanggung jawab belajar mandiri — sikap kerja yang sesuai dengan indikator SKKNI untuk pelatihan vokasi TI.
Kompetensi Terpadu (Integrative Competence)	Hasil <i>Paired Sample T-Test</i> dan <i>Independent Sample T-Test</i>	Terdapat perbedaan signifikan antara kelompok eksperimen dan kontrol pada <i>Post-Test</i> ($p = 0,000$) dan <i>Delay-Test</i> ($p = 0,001$).	Menunjukkan bahwa integrasi FIB dengan gamifikasi dan evaluasi AST efektif meningkatkan kompetensi vokasional secara menyeluruh — mencakup kognitif, teknis, dan afektif.

Temuan ini memperlihatkan bahwa sistem FIB tidak hanya meningkatkan hasil belajar secara statistik, tetapi juga mendukung penguatan kompetensi vokasional secara holistik. Peningkatan signifikan pada nilai *Post-Test* dan *Delay-Test* menandakan adanya retensi pemahaman jangka panjang, sedangkan konsistensi hasil evaluasi AST mencerminkan kemampuan teknis yang lebih stabil. Dari sisi perilaku belajar, efisiensi waktu dan konsistensi latihan yang lebih baik mengindikasikan pembentukan sikap kerja profesional, selaras dengan standar SKKNI yang menekankan kedisiplinan, tanggung jawab, dan ketekunan. Dengan demikian, sistem FIB dapat dikatakan relevan sebagai instrumen penilaian berbasis kompetensi yang mampu mengukur dimensi pengetahuan, keterampilan, dan sikap kerja secara terintegrasi. Secara keseluruhan, hasil penelitian ini menegaskan bahwa implementasi sistem FIB selaras dengan tujuan SKKNI untuk menghasilkan lulusan yang kompeten secara teori, terampil secara praktik, dan profesional secara sikap kerja. Integrasi evaluasi otomatis berbasis AST juga memperkuat objektivitas penilaian dan mendukung standarisasi asesmen vokasional di bidang teknologi informasi.

5. Simpulan

Berdasarkan hasil penelitian dan pembahasan, dapat disimpulkan bahwa sistem pembelajaran berbasis *Fill-in-the-Blank* (FIB) menunjukkan tingkat usability yang baik dengan skor rata-rata 76,47 pada *System Usability Scale* (SUS), serta diterima dengan baik oleh mahasiswa. Pengujian *blackbox* menegaskan bahwa seluruh fitur sistem berfungsi sesuai desain pada berbagai *level* pengguna, mulai dari Mahasiswa, Pengajar, hingga Admin. Analisis *Quasi-Experiment* menunjukkan bahwa sistem FIB secara signifikan meningkatkan hasil belajar mahasiswa, dengan peningkatan rata-rata nilai *Pre-Test* 27,45 menjadi *Post-Test* 92,92 pada kelompok eksperimen (*Paired Sample T-Test*, $t = 7,342$, $p = 0,000$), serta retensi pemahaman

yang stabil pada *Delay-Test* 81,84 ($t = 1,78$, $p = 0,087$), dibandingkan kelompok kontrol yang hanya meningkat dari 27,99 menjadi 49,68 (*Post-Test*) dengan retensi 49,52 (*Delay-Test*). Hasil *Independent Sample T-Test* juga menunjukkan perbedaan signifikan antara kelompok eksperimen dan kontrol pada *Post-Test* ($t = 4,488$, $p = 0,000$) dan *Delay-Test* ($t = 3,387$, $p = 0,001$), menegaskan efektivitas sistem FIB dibandingkan metode konvensional.

Meskipun demikian, penelitian ini memiliki beberapa limitasi. Pertama, uji coba sistem terbatas pada mahasiswa satu mata kuliah tertentu, sehingga generalisasi hasil untuk konteks pembelajaran yang lebih luas perlu kehati-hatian. Kedua, fitur sistem masih bersifat dasar dan belum mengintegrasikan analisis performa individual atau rekomendasi adaptif, yang berpotensi meningkatkan efektivitas pembelajaran dan personalisasi interaksi pengguna. Ketiga, penelitian ini fokus pada hasil belajar dan retensi jangka menengah, sehingga evaluasi jangka panjang serta integrasi *learning analytics* berbasis AST dapat menjadi arah pengembangan penelitian berikutnya. Dengan memperhatikan limitasi ini, sistem FIB tetap terbukti memberikan kontribusi positif terhadap efektivitas dan pengalaman belajar interaktif mahasiswa.

Daftar Referensi

- [1] H. C. Ling, K. L. Hsiao, dan W. C. Hsu, "Can Students' Computer Programming Learning Motivation and Effectiveness Be Enhanced by Learning Python Language? A Multi-Group Analysis," *Front Psychol*, vol. 11, no. January, pp. 1-7, 2021, doi: 10.3389/fpsyg.2020.600814.
- [2] A. Fauziah, R. Silpiana, J. P. Matematika, U. Bakti, dan I. Banyuwangi, "Berbasis Python Di Smk Puspa Bangsa Banyuwangi Making a Simple Conversion Application Using Python-Based Gui At," vol. 3, no. 4, pp. 304–308, 2024, doi: 10.58184/mestaka.v3i4.378.
- [3] D. N. Abiyyu dan M. Y. Efendi, "Upaya Peningkatan Pemrograman Pembelajaran Berbasis Game dengan Scratch pada Peserta Didik SMP Dharma Karya," *Seminar nasional dan publikasi ilmiah 2024 FIP UMJ*, no. 2007, pp. 2486–2493, 2024.
- [4] H. H. S. Kyaw, N. Funabiki, S. L. Aung, N. K. Dim, dan W. C. Kao, "A study of element fill-in-blank problems for c programming learning assistant system," *International Journal of Information and Education Technology*, vol. 11, no. 6, pp. 255–261, Jun 2021, doi: 10.18178/ijiet.2021.11.6.1520.
- [5] H. W. Hnin dan K. K. Zaw, "Element Fill-in-Blank Problems in Python Programming Learning Assistant System," dalam *Proceedings of the 4th International Conference on Advanced Information Technologies, ICAIT 2020*, Institute of Electrical and Electronics Engineers Inc., Nov 2020, pp. 88–93. doi: 10.1109/ICAIT51105.2020.9261778.
- [6] S. Matsumoto, A. Ohno, dan H. Shigematsu, "Development of a Diagnostic Tool to Verify the Adequacy of Multiple-Choice Fill-in-the-Blank Programming Learning Tasks and the Appropriateness of Their Difficulty Level," *Information Engineering Express*, vol. 8, no. 2, p. 1, 2022, doi: 10.52731/iee.v8.i2.623.
- [7] A.-T. P. Nguyen dan V.-D. Hoang, "Development of Code Evaluation System based on Abstract Syntax Tree," *Journal of Technical Education Science*, vol. 19, no. 1, pp. 15–24, 2024, doi: 10.54644/jte.2024.1514.
- [8] M. Othman, A. Osman, S. Z. Ahmad, dan N. Abdullah, "Integrating Segmenting and Gamification Principles in the Design of Interactive Gamified Programming Assessments for Low Achievers," *Turkish Online Journal of Distance Education*, vol. 25, no. 3, pp. 167–183, 2024, doi: 10.17718/TOJDE.1315427.
- [9] M. N. Maskuri, H. W. Herwanto, R. Firdaus, R. D. Jayanti, dan H. D. Ratnasari, "Pengembangan Media Pembelajaran Berbasis Augmented Reality Pada Materi Mikrokontroler Mata Pelajaran Informatika di SMA Negeri 9 Malang," *Jutisi: Jurnal Ilmiah Teknik Informatika dan Sistem Informasi*, vol. 13, no. 3, pp. 2102–2113, 2024, doi: 10.35889/jutisi.v13i3.2356.
- [10] M. Multazam, Z. Syahrial, dan Rusmono, "Development of Learning Models in Web Programming Courses With Computer-Based Learning Tutorials," *Turkish Online Journal of Distance Education*, vol. 24, no. 2, pp. 232–244, 2023, doi: 10.17718/tojde.1081507.
- [11] V. A. Lestari, F. A. Pribadi, E. L. Amalia, A. L. Sari, dan E. L. Maulidina, *Preliminary Analysis of Experimental Results Using Drag-and-Drop and Log Data on PseudoLearn in Learning Logic Programming*. Atlantis Press International BV, 2023. doi: 10.2991/978-94-6463-358-0_9.

-
- [12] T. Kakeshita dan M. Murata, "Analysis of Student ' s Learning Log Data in Fill -in-the- Blank Programming Questions," vol. 5, no. 1, pp. 1–17, 2022.
- [13] M.D. Gumilar, F. Sembiring, & A. Erfina, "Implementasi progressive web app pada sistem informasi e-learning untuk pembelajaran bahasa pemrograman python". *Jutisi: Jurnal Ilmiah Teknik Informatika dan Sistem Informasi*, vol. 10, no. 2, pp. 309-318, 2021.
- [14] W. Ruimei, N. N. Zulkifli, dan A. F. M. Ayub, "The impact of the stratified cognitive apprenticeship model on mathematical motivation in high school students," *Environment and Social Psychology*, vol. 9, no. 8, pp. 1-18, 2024, doi: 10.59429/esp.v9i8.2819.
- [15] B. N. Laili, M. Maryani, dan T. Prihandono, "Implementasi Model Problem Based Learning Terhadap Keterampilan Generik Sains Pada Materi Sifat-Sifat Cahaya," *Paedagoria : Jurnal Kajian, Penelitian dan Pengembangan Kependidikan*, vol. 13, no. 1, p. 66, 2022, doi: 10.31764/paedagoria.v13i1.8148.
- [16] A. Manzano-León *dkk.*, "Between level up and game over: A systematic literature review of gamification in education," *Sustainability (Switzerland)*, vol. 13, no. 4, pp. 1–14, 2021, doi: 10.3390/su13042247.
- [17] L. Afifah dan N. K. Umam, "Pengaruh Model Pembelajaran Analogi terhadap Keterampilan Berpikir Kreatif pada Mata Pelajaran Bahasa Indonesia Siswa Kelas III," *JlIP - Jurnal Ilmiah Ilmu Pendidikan*, vol. 6, no. 2, pp. 1370–1375, 2023, doi: 10.54371/jiip.v6i2.1735.
- [18] A. Suciani, D. Ruhiat, dan S. D. Rahayu Septiani, "Komparasi hasil analisis beda rata-rata menggunakan metode statistik parametrik dan nonparametrik," *Jurnal Riset Matematika dan Sains Terapan*, vol. 2, no. 2, pp. 76–91, 2022.
- [19] P. Mishra, U. Singh, dan C. M. Pandey, "Application of Student ' s t - test , Analysis of Variance , and Covariance," pp. 407–411, 2019, doi: 10.4103/aca.ACA.
- [20] N. Huda, F. Habrizons, A. Satriawan, M. Iranda, dan T. Pramuda, "Analisis Usability Testing Menggunakan Metode SUS (System Usability Scale) Terhadap Kepuasan Pengguna Aplikasi Shopee," *Simkom*, vol. 8, no. 2, pp. 208–220, 2023, doi: 10.51717/simkom.v8i2.158.
- [21] E. Pelivani dan B. Cico, "A comparative study of automation testing tools for web applications," *2021 10th Mediterranean Conference on Embedded Computing, MECO 2021*, pp. 7–10, 2021, doi: 10.1109/MECO52532.2021.9460242.