

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Mata kuliah pengantar pemrograman cenderung memiliki tingkat kelulusan yang lebih rendah dibandingkan dengan mata kuliah pengantar lainnya di bidang STEM [1]. Sebuah penelitian yang menganalisis data dari 17 universitas di 8 negara pada tahun 2014 hingga 2018 menemukan bahwa rata-rata tingkat kelulusan kuliah ini hanya mencapai 75% [1]. Penelitian yang sama juga menyimpulkan bahwa angka tersebut menunjukkan masih adanya peluang besar untuk meningkatkan kualitas pengajaran dan pembelajaran di mata kuliah pengantar pemrograman [1]. Salah satu upaya peningkatan yang dapat dilakukan adalah dengan membantu mahasiswa menghadapi tantangan utama yang mereka alami. Tantangan tersebut bukan terletak pada pemahaman konsep dasar, melainkan pada penerapan konsep tersebut [2]. Mahasiswa merasa bahwa metode pembelajaran praktis merupakan pendekatan yang paling membantu mereka untuk menghadapi masalah tersebut [2]. Salah satu bentuk utama dari pembelajaran praktis adalah melalui latihan soal. Latihan terbukti efektif dalam membantu mahasiswa memahami konsep secara lebih mendalam serta meningkatkan performa mereka dalam ujian [3], [4]. Selain itu, latihan juga berperan penting dalam membangun kepercayaan diri mahasiswa dan menurunkan tingkat plagiarisme [4], [5].

Mata kuliah Algoritma Pemrograman merupakan salah satu mata kuliah pengantar pemrograman di S1 Informatika Universitas Telkom. Dalam mata kuliah yang ditujukan untuk mahasiswa semester dua tersebut, latihan soal dilakukan menggunakan *plug-in* CodeRunner yang terintegrasi dengan *learning management system* (LMS) berbasis Moodle. Melalui CodeRunner, mahasiswa dapat mengetik dan memvalidasi kode program dengan umpan balik instan sehingga memberikan pengalaman latihan mandiri yang interaktif [6], [7]. Keinteraktifan dan kemudahan penggunaan CodeRunner menjadikannya sangat disukai oleh para mahasiswa [6], [7]. Meskipun demikian, pemberian soal dalam latihan hanya dapat dilakukan secara seragam atau acak. Hal ini disebabkan oleh

keterbatasan CodeRunner dan Moodle, yang belum dilengkapi kemampuan adaptif untuk memberikan soal latihan yang terpersonalisasi berdasarkan kemampuan masing-masing mahasiswa. Meskipun terdapat penelitian sebelumnya terkait personalisasi Moodle, seperti yang dilakukan pada [8], [9], [10], dan [11], penelitian-penelitian tersebut hanya berfokus pada personalisasi jalur pembelajaran dan bahan ajar, tanpa mencakup soal latihan.

Latihan yang pemberian soalnya tidak mempertimbangkan kemampuan mahasiswa berpotensi menyebabkan ketimpangan antara tingkat kesulitan soal dan kemampuan mahasiswa. Ketimpangan ini dapat memicu rasa frustrasi, tingginya tingkat *dropout* (menyerah), serta rendahnya tingkat keberhasilan mahasiswa dalam menyelesaikan soal latihan [12]. [13]. Mengingat besarnya manfaat yang dapat diperoleh dari latihan soal dalam pembelajaran pemrograman, sangat disayangkan jika mahasiswa kehilangan motivasi untuk mengerjakan latihan akibat rasa frustrasi tersebut [13], [14]. Dengan demikian, personalisasi latihan pemrograman penting dilakukan untuk mencegah dampak negatif dari ketimpangan tingkat kesulitan soal terhadap kemampuan mahasiswa. Salah satu solusi untuk menerapkan personalisasi tersebut adalah dengan menggunakan sistem latihan adaptif yang terbukti mampu memberikan soal sesuai dengan tingkat kemampuan maupun usaha mahasiswa [12], [15], [16], [17], [18].

Bukti tersebut didapatkan dari berbagai penelitian sebelumnya yang mencoba membuat sistem latihan adaptif untuk pemrograman menggunakan sistem rekomendasi di luar sistem Moodle dan CodeRunner. Penelitian yang dilakukan Hosseini dan Brusilovsky [15] serta Cardenas-Cobo [16] mengembangkan sistem rekomendasi yang menggunakan kemampuan mahasiswa sebagai faktor utama dalam menentukan soal yang direkomendasikan. Di sisi lain, Pereira et al. [12] dan Toledo et al. [17], [18] mengembangkan sistem yang tidak hanya menggunakan kemampuan mahasiswa tetapi juga usaha sebagai faktor penentu rekomendasi soal. Namun, penelitian [12] belum mampu menyesuaikan tingkat kesulitan soal secara bertahap. Di sisi lain, penelitian [17] dan [18] menggunakan teknik yang

dinilai kurang sesuai untuk rekomendasi soal latihan [19]. Selain itu, kelima penelitian tersebut memiliki kesamaan dalam tidak mempertimbangkan kemampuan yang telah dimiliki mahasiswa sebelum menggunakan sistem yang dikembangkan. Terakhir, dari seluruh penelitian yang telah dicantumkan, hanya penelitian Hosseini dan Brusilovsky [15], Cardenas-Cobo [16], dan Pereira et al. [12] yang memungkinkan pengguna memberikan umpan balik mengenai kepuasan mereka terhadap soal yang direkomendasikan.

Dengan demikian, penelitian-penelitian sebelumnya masing-masing memiliki keterbatasan, baik dalam mempertimbangkan kemampuan awal pengguna, menyesuaikan tingkat kesulitan soal secara bertahap, maupun melibatkan umpan balik pengguna dalam sistem yang dikembangkan. Tidak hanya itu, tidak ada satu pun dari penelitian sebelumnya yang mengembangkan sistem yang terintegrasi dengan LMS Moodle dan CodeRunner. Selain itu, menurut direktori Plugin Moodle, belum terdapat *plugin* atau ekstensi Moodle yang mendukung latihan adaptif untuk pemrograman. Keterbatasan ini penting untuk diatasi dalam sebuah sistem baru yang dapat digunakan pada Moodle dan CodeRunner guna menciptakan pengalaman latihan yang lebih baik. Ini karena, sistem yang mampu mempertimbangkan pengetahuan awal mahasiswa terbukti dapat meningkatkan performa mereka dalam proses pembelajaran [20]. Sementara itu, penyesuaian tingkat kesulitan soal, khususnya melalui peningkatan secara bertahap, memastikan mahasiswa mendapatkan tantangan konstruktif yang mendorong mereka untuk bereksplorasi dan menemukan pemahaman melalui proses pembelajaran yang aktif [21]. Penyesuaian tingkat kesulitan soal secara bertahap juga sejalan dengan pendekatan *Mastery Learning* yang menekankan pentingnya penguasaan satu konsep sebelum melanjutkan ke konsep yang lebih sulit [22]. Terakhir, melibatkan mahasiswa dalam memberikan umpan balik akan meningkatkan kinerja sistem adaptif dan memberikan dampak positif pada kepuasan belajar mereka [21].

Kebutuhan akan sistem baru yang mampu mengatasi tiga keterbatasan sebelumnya, serta belum adanya integrasi sistem latihan adaptif pada Moodle dan

CodeRunner yang secara khusus dirancang untuk konteks pembelajaran pemrograman, menjadi dasar dari Tugas Akhir ini. Penelitian ini bertujuan untuk mengembangkan sistem latihan adaptif untuk pemrograman pada Moodle dan CodeRunner. Sistem ini memungkinkan rekomendasi soal latihan yang mempertimbangkan kemampuan awal dan usaha mahasiswa, menyesuaikan tingkat kesulitan soal secara bertahap sesuai pendekatan *Mastery Learning*, serta menerima dan melibatkan umpan balik mahasiswa dalam memberikan rekomendasi soal latihan. Kemampuan awal mahasiswa dalam konteks ini didasarkan pada nilai mata kuliah prasyarat. Sementara itu, sesuai metode yang digunakan pada penelitian [12], data log dari usaha mahasiswa digunakan untuk menentukan tingkat kesulitan soal. Berdasarkan hasil penelitian yang mengembangkan sistem serupa [12], sistem ini dapat meningkatkan kepuasan latihan mahasiswa. Peningkatan kepuasan ini dapat memicu peningkatan motivasi belajar, yang mendorong mahasiswa agar lebih giat dalam berlatih [4]. Ini dapat meningkatkan performa mereka dalam memahami dan mengimplementasikan konsep-konsep dasar pemrograman, yang tercermin melalui peningkatan hasil ujian mahasiswa [3], [4].

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, permasalahan dalam Tugas Akhir ini dapat dirumuskan sebagai berikut:

1. Bagaimana mengembangkan sistem latihan adaptif untuk mata kuliah pengenalan pemrograman yang terintegrasi dengan LMS Moodle, mampu mempertimbangkan kemampuan awal dan usaha mahasiswa, menyesuaikan tingkat kesulitan soal secara bertahap sesuai dengan pendekatan *Mastery Learning*, serta menerima umpan balik dari mahasiswa?
2. Bagaimana pengaruh dari penggunaan sistem latihan adaptif tersebut terhadap performa dan motivasi latihan mahasiswa?

1.3 Tujuan

Berdasarkan rumusan masalah di atas, berikut adalah tujuan dalam Tugas Akhir ini:

1. Mengembangkan sistem latihan adaptif untuk mata kuliah pengenalan pemrograman yang terintegrasi dengan LMS Moodle, mampu mempertimbangkan kemampuan awal dan usaha mahasiswa, menyesuaikan tingkat kesulitan soal secara bertahap sesuai dengan pendekatan *Mastery Learning*, serta menerima umpan balik dari mahasiswa.
2. Mengetahui pengaruh dari penggunaan sistem latihan adaptif tersebut terhadap performa dan motivasi latihan mahasiswa.

1.4 Batasan Masalah

Mengingat keterbatasan waktu pengerjaan Tugas Akhir yang hanya berlangsung selama satu semester dan untuk memastikan penelitian yang terarah, maka ditetapkan batasan masalah pada Tugas Akhir ini:

1. Pengembangan model, sistem rekomendasi, dan integrasinya menjadi sebuah sistem latihan adaptif dibatasi pada materi array dan struct yang diajarkan pada mata kuliah Algoritma Pemrograman 2 di Universitas Telkom. Mata kuliah ini dipilih karena merupakan mata kuliah yang termasuk mata kuliah pengenalan pemrograman yang sudah menggunakan Moodle dan CodeRunner untuk latihan mandiri. Namun, saat ini latihan pada mata kuliah tersebut hanya memberikan soal latihan secara seragam atau acak.

1.5 Hipotesis

Berdasarkan latar belakang, rumusan masalah, dan tujuan yang telah ditulis, hipotesis mengenai Tugas Akhir ini dapat dirumuskan sebagai berikut:

1. H_1 : Sistem latihan adaptif yang terintegrasi pada Moodle dan Coderunner mampu memberikan soal yang sesuai dengan kemampuan awal mahasiswa.

H₀: Sistem latihan adaptif yang terintegrasi pada Moodle dan Coderunner tidak mampu memberikan soal yang sesuai dengan kemampuan awal mahasiswa.

2. H₁: Sistem latihan adaptif yang terintegrasi pada Moodle dan Coderunner mampu menyesuaikan tingkat kesulitan soal secara bertahap sesuai dengan pendekatan *Mastery Learning*.

H₀: Sistem latihan adaptif yang terintegrasi pada Moodle dan Coderunner tidak mampu menyesuaikan tingkat kesulitan soal secara bertahap sesuai dengan pendekatan *Mastery Learning*.

3. H₁: Sistem latihan adaptif yang terintegrasi pada Moodle dan Coderunner mampu menerima umpan balik dari mahasiswa untuk meningkatkan relevansi rekomendasi soal selanjutnya.

H₀: Sistem latihan adaptif yang terintegrasi pada Moodle dan Coderunner tidak mampu menerima umpan balik dari mahasiswa untuk meningkatkan relevansi rekomendasi soal selanjutnya.

4. H₁: Terdapat pengaruh positif dari penggunaan sistem latihan adaptif terhadap performa latihan mahasiswa.

H₀: Tidak terdapat pengaruh positif dari penggunaan sistem rekomendasi soal latihan terhadap performa latihan mahasiswa.

5. H₁: Terdapat pengaruh positif dari penggunaan sistem latihan adaptif terhadap motivasi latihan mahasiswa.

H₀: Tidak terdapat pengaruh positif dari penggunaan sistem latihan adaptif terhadap motivasi latihan mahasiswa.

1.6 Metode Penelitian

Tahapan metode penelitian yang dilakukan pada Tugas Akhir ini terbagi menjadi tujuh tahapan utama, yakni sebagai berikut:

1. Studi literatur

Pada tahapan ini, penulis melakukan kajian terhadap penelitian yang berkaitan dengan sistem latihan adaptif untuk pembelajaran pemrograman.

2. Identifikasi masalah

Pada tahapan ini, penulis mengidentifikasi permasalahan yang masih belum terselesaikan dari penelitian-penelitian sebelumnya beserta permasalahan yang ada pada sistem latihan pemrograman yang digunakan di Universitas Telkom.

3. Pengembangan sistem

Pada tahapan ini, penulis merancang dan melakukan implementasi sistem, mulai dari domain model, student model, sistem rekomendasi, dan integrasinya dengan LMS Moodle. Penulis juga melakukan pengumpulan dan prapemrosesan soal latihan.

4. Percobaan

Pada tahapan ini, sistem yang telah berhasil dikembangkan akan dicoba oleh objek penelitian.

5. Analisis hasil percobaan

Setelah percobaan selesai dilakukan, penulis menganalisis hasil percobaan tersebut.

6. Penyusunan Buku TA

Pada tahapan ini, laporan disusun berdasarkan seluruh tahapan yang telah dilakukan sebelumnya. Laporan disusun sesuai dengan format dan ketentuan TA.

1.7 Jadwal Pelaksanaan

Berikut adalah jadwal pelaksanaan dan pengerjaan Tugas Akhir berikut.

Tabel 1.1 Jadwal Pelaksanaan Tugas Akhir.

No.	Deskripsi Tahapan	Bulan 1	Bulan 2	Bulan 3	Bulan 4	Bulan 5	Bulan 6
1	Studi literatur						
2	Identifikasi masalah						
3	Perancangan dan implementasi sistem						
4	Percobaan						
5	Analisis hasil percobaan						
6	Penyusunan Buku TA						