

BAB 1 PENDAHULUAN

1.1 Latar Belakang Masalah

Kemampuan berpikir komputasional dan pemrograman menjadi bagian dari kompetensi yang perlu dikembangkan pada jenjang pendidikan menengah. Capaian Pembelajaran Informatika Fase E di Indonesia menempatkan algoritma dan pemrograman sebagai salah satu elemen pembelajaran; peserta didik diharapkan mampu menerapkan konsep pemrograman prosedural dalam suatu bahasa pemrograman serta mengembangkan program yang terstruktur [1]. Jenjang pendidikan menengah atas secara umum mencakup peserta didik berusia sekitar 15–18 tahun [2]. Berdasarkan konteks tersebut, aplikasi pada penelitian ini ditujukan sebagai media latihan mandiri bagi siswa SMA/SMK, terutama peserta didik yang telah memperoleh pengantar algoritma dan pemrograman.

Pembelajaran pemrograman tidak hanya menuntut peserta didik mengingat sintaks, tetapi juga memahami alur eksekusi, tipe data, percabangan, perulangan, metode, objek, serta proses menemukan kesalahan. Kesulitan pada satu konsep dapat memengaruhi pemahaman konsep berikutnya karena materi pemrograman bersifat bertahap. Sistem latihan yang memberikan soal sama kepada seluruh siswa berisiko menghasilkan dua kondisi: soal terlalu mudah sehingga tidak memberi tantangan, atau soal terlalu sulit sehingga siswa tidak memperoleh penguatan pada prasyarat yang belum dikuasai. Penelitian mengenai ITS untuk pembelajaran pemrograman menunjukkan bahwa pelacakan kondisi siswa, bantuan bertahap, dan umpan balik informatif dapat mendukung pengalaman belajar yang lebih personal [3]. Umpan balik yang disajikan segera setelah interaksi juga membantu siswa memahami kesalahan dibandingkan hanya menerima nilai akhir [4].

Kesulitan tersebut juga ditemukan secara empiris pada penelitian mutakhir. Lehtinen dkk. menemukan bahwa sekitar sepertiga mahasiswa pada mata kuliah pemrograman pengantar mengalami kesulitan ketika diminta menjelaskan kode program yang mereka tulis sendiri [5]. Pada konteks sekolah menengah, Shahin dkk. melaporkan bahwa praktik *debugging* dipersepsikan sebagai salah satu bagian paling sulit dalam aktivitas pemrograman kolaboratif oleh peserta didik [6]. Kedua penelitian berasal dari konteks peserta dan lingkungan belajar yang berbeda dari siswa SMA/SMK Indonesia sehingga hasilnya tidak digunakan

sebagai estimasi langsung terhadap populasi sasaran. Namun, temuan tersebut menunjukkan bahwa kesulitan memahami logika program, menjelaskan kode, dan memperbaiki kesalahan tetap muncul pada pembelajaran pemrograman serta memperkuat kebutuhan akan latihan bertahap, *hint*, umpan balik, dan pemantauan kondisi pengguna.

Bahasa Java dipilih karena memiliki karakteristik yang sesuai untuk latihan pemrograman tekstual pada tingkat menengah. Java merupakan bahasa berorientasi objek, bertipe kuat, dan dijalankan melalui Java Virtual Machine sehingga program dapat dijalankan pada berbagai platform yang menyediakan JVM [7]. Penggunaan Java pada pendidikan menengah juga memiliki preseden internasional. Mata pelajaran AP Computer Science A, yang ditujukan untuk siswa sekolah menengah, menggunakan Java untuk mempelajari variabel, struktur kontrol, modularitas, kelas, objek, dan pemecahan masalah [8]. Oleh karena itu, Java dipilih bukan karena merupakan satu-satunya bahasa yang sesuai untuk pemula, melainkan karena dapat menjembatani pembelajaran prosedural menuju konsep berorientasi objek yang relevan bagi siswa SMA/SMK, khususnya program keahlian yang berkaitan dengan pengembangan perangkat lunak.

Walaupun Java relevan untuk pembelajaran, sintaks tekstual dan struktur programnya dapat menimbulkan beban belajar bagi siswa pemula. Aplikasi latihan perlu menyediakan urutan soal, petunjuk, dan umpan balik yang disesuaikan dengan perkembangan pengguna. *Intelligent Tutoring System* (ITS) menawarkan kerangka untuk kebutuhan tersebut melalui integrasi *domain model*, *student model*, *tutoring model*, dan antarmuka pengguna. ITS tidak hanya menyajikan kumpulan soal, tetapi juga menyimpan riwayat interaksi dan menggunakan informasi tersebut untuk menentukan dukungan belajar berikutnya. Fodouop Kouam membahas efektivitas ITS dalam mendukung peserta dengan tingkat pengalaman pemrograman yang berbeda [9]. Lai dan Lin menganalisis perilaku serta hasil belajar pengguna ITS berdasarkan perbedaan tingkat pengetahuan awal [10]. Yang dkk. menunjukkan penerapan tutor berbasis ChatGPT untuk mendukung pembelajaran pemrograman melalui interaksi dan umpan balik [11].

Penelitian ini menggunakan algoritma *Random Forest* untuk memperkirakan peluang keberhasilan pengguna pada kandidat soal. Model dilatih menggunakan data interaksi CodeWorkout Fall 2019 yang dipublikasikan untuk *2nd CSED Data Challenge 2021*. Dataset tersebut memuat riwayat mahasiswa pada 50 soal pemrograman Java dan menyediakan informasi jumlah percobaan, keberhasilan penyelesaian, serta label keberhasilan atau kesulitan [12].

Tinjauan terhadap data CSEDM juga menekankan pentingnya dokumentasi asal data, struktur interaksi, dan keterbatasan dataset ketika digunakan untuk pemodelan pendidikan komputasi [13].

Pemilihan *Random Forest* dilakukan setelah mempertimbangkan beberapa algoritma klasifikasi. *Logistic Regression* memiliki kelebihan berupa proses pelatihan yang sederhana, cepat, dan mudah dijelaskan, sehingga tetap digunakan sebagai model pembanding atau *baseline*. Namun, model tersebut memiliki keterbatasan ketika hubungan antara riwayat siswa dan peluang keberhasilan bersifat nonlinier. Pohon keputusan menawarkan aturan yang mudah dibaca, tetapi satu pohon dapat sensitif terhadap perubahan data dan berisiko mengalami *overfitting*. *Gradient Boosting* sering memberikan kinerja yang kuat pada data tabular, tetapi memerlukan penyetelan parameter yang lebih sensitif dan integrasi penjelasan model yang lebih kompleks. Sementara itu, *Support Vector Machine* mampu menangani batas keputusan yang kompleks, tetapi keluaran probabilitas dan penjelasan kontribusi fitur tidak sepraktis model berbasis pohon untuk kebutuhan aplikasi ini.

Berdasarkan pertimbangan tersebut, *Random Forest* dipilih sebagai model utama karena menggabungkan banyak pohon keputusan, mampu memodelkan hubungan nonlinier pada fitur riwayat siswa yang heterogen, dan menyediakan informasi kepentingan fitur yang dapat digunakan dalam analisis model. Tinjauan Daza dkk. menunjukkan bahwa prediksi kinerja akademik menggunakan beragam teknik *data mining*, sehingga pemilihan model perlu disesuaikan dengan karakteristik data dan tujuan analisis [14]. Survei Batool dkk. juga menekankan perlunya mengevaluasi teknik prediksi, fitur, dan konteks penggunaan pada data pendidikan [15]. Penggunaan *Random Forest* pada prediksi kinerja peserta didik berbasis data tabular turut dilaporkan oleh Pan dkk. [16], sedangkan penelitian prediksi dini menekankan pentingnya membandingkan model dan mengevaluasi fitur yang tersedia sebelum hasil akhir diketahui [17]. Pertimbangan ini digunakan sebagai dasar desain, bukan sebagai eksperimen menyeluruh terhadap seluruh algoritma. Oleh karena itu, implementasi penelitian tetap menyertakan *Logistic Regression* sebagai *baseline* agar kinerja *Random Forest* tidak dinilai tanpa model pembanding.

Fokus penelitian bukan menjadikan *Random Forest* sebagai satu-satunya pengendali sistem. Model memprediksi peluang keberhasilan, sedangkan mekanisme adaptif menggabungkan prediksi tersebut dengan *mastery* topik, riwayat soal, penggunaan *hint*, batas tingkat kesulitan sesi, dan kebijakan pedagogis.

Dengan demikian, adaptasi yang diterapkan merupakan adaptasi hibrida berbasis data dan aturan. Ouyang dan Jiao menjelaskan paradigma AI dalam pendidikan yang menempatkan hubungan peserta didik, pengajar, dan sistem sebagai bagian dari pengambilan keputusan pembelajaran [18]. Villegas-Ch dkk. membahas ITS adaptif yang menggunakan personalisasi dan umpan balik untuk menyesuaikan dukungan terhadap kebutuhan pengguna [19]. Perpindahan level setelah satu sesi ditentukan oleh hasil 20 soal, sedangkan pemilihan soal di dalam sesi mempertimbangkan hasil prediksi model dan kondisi pengguna.

Aplikasi yang dibangun berbasis web menggunakan Python dan Streamlit. Aplikasi memiliki autentikasi pengguna, penyimpanan progres, *student model*, *adaptive engine*, *hint* bertahap, pemeriksaan jawaban, dan ringkasan sesi. Bank soal terakhir terdiri atas 5.500 soal, 11 topik, 5 tingkat kesulitan, dan 5 jenis soal. Setiap sesi berisi 20 soal yang dipilih satu per satu tanpa pengulangan dari kandidat yang sesuai dengan topik dan tingkat kesulitan sesi. Angka 20 merupakan batas desain sesi, bukan jumlah soal yang dipelajari oleh model dan bukan 20 soal akhir pada dataset CSEDm.

Apabila perbedaan kemampuan tersebut tidak ditangani, siswa yang belum menguasai konsep prasyarat dapat terus menerima soal yang berada di luar kesiapan belajarnya, sedangkan siswa yang telah menguasai materi dapat menerima latihan yang kurang menantang. Kondisi tersebut berpotensi mengurangi fungsi latihan sebagai sarana penguatan konsep dan menyulitkan siswa maupun guru dalam membaca perkembangan kemampuan secara bertahap. Berdasarkan masalah tersebut, penelitian ini membangun aplikasi *Intelligent Tutoring System* pembelajaran Java yang mengintegrasikan bank soal, *student model*, *hint* progresif, umpan balik, penyimpanan progres, dan pemilihan soal adaptif. *Random Forest* digunakan untuk memperkirakan peluang keberhasilan pengguna pada kandidat soal, sedangkan keputusan akhir juga mempertimbangkan *mastery*, riwayat interaksi, tingkat kesulitan sesi, dan aturan pedagogis. Dengan demikian, kontribusi penelitian terletak pada integrasi model prediksi dan aturan pedagogis sebagai pendukung latihan Java adaptif bagi siswa SMA/SMK. Judul penelitian disesuaikan dengan fokus tersebut menjadi “Rancang Bangun Aplikasi Intelligent Tutoring System Pembelajaran Java dengan Random Forest sebagai Pendukung Pemilihan Soal Adaptif”.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, rumusan masalah penelitian ini adalah sebagai berikut.

1. Bagaimana merancang dan membangun aplikasi *Intelligent Tutoring System* pembelajaran Java bagi siswa SMA/SMK yang mengintegrasikan bank soal, *student model*, *hint*, umpan balik, penyimpanan progres, dan pemilihan soal adaptif?
2. Bagaimana mengevaluasi aplikasi yang dibangun melalui pengujian fungsional, pengujian model *Random Forest*, pengujian mekanisme adaptif, *explainable AI*, dan respons pengguna?

1.3 Batasan Masalah

Batasan masalah penelitian ini adalah sebagai berikut.

1. Target utama aplikasi adalah siswa SMA/SMK, terutama peserta didik berusia sekitar 15–18 tahun yang telah memperoleh pengantar algoritma dan pemrograman. Aplikasi tidak ditujukan untuk anak usia dini dan tidak menggantikan proses pembelajaran di kelas.
2. Materi dibatasi pada 11 topik Java yang tersedia di bank soal, yaitu variabel, tipe primitif, *string*, operator, percabangan, perulangan, *array*, metode, kelas dan objek, *debugging*, serta penelusuran kode.
3. Model *Random Forest* dilatih menggunakan CodeWorkout Fall 2019. Dataset berasal dari mata kuliah pengantar pemrograman tingkat universitas di Amerika Serikat, sehingga terdapat perbedaan domain dengan target pengguna SMA/SMK di Indonesia.
4. Target model adalah label keberhasilan atau kesulitan siswa pada suatu soal, bukan label tingkat kesulitan 1–5. Hasil model digunakan sebagai sinyal pendukung pemilihan kandidat soal.
5. Penyesuaian tingkat kesulitan antarsesi menggunakan aturan pedagogis berdasarkan jumlah jawaban benar, sedangkan pemilihan soal di dalam sesi menggunakan kombinasi prediksi model, kondisi *student model*, dan filter sistem.

6. Evaluasi pengguna bersifat formatif dan deskriptif. Data dari 40 responden digunakan untuk menilai persepsi terhadap prototipe, bukan untuk menyimpulkan efektivitas pembelajaran pada seluruh populasi siswa SMA/SMK.
7. Aplikasi belum menjalankan kode Java di dalam *sandbox*; pemeriksaan jawaban menggunakan jawaban benar, variasi jawaban yang diterima, dan pola kata kunci yang telah disiapkan.

1.4 Tujuan Penelitian

Tujuan penelitian ini adalah sebagai berikut.

1. Merancang dan membangun aplikasi *Intelligent Tutoring System* pembelajaran Java bagi siswa SMA/SMK yang mengintegrasikan bank soal, *student model*, *hint*, umpan balik, penyimpanan progres, dan pemilihan soal adaptif dengan *Random Forest* sebagai komponen pendukung.
2. Mengevaluasi hasil rancang bangun melalui pengujian fungsional, kinerja model *Random Forest*, mekanisme adaptif, penjelasan fitur model, dan respons pengguna.

1.5 Manfaat Penelitian

Manfaat yang diharapkan adalah sebagai berikut.

1. Bagi siswa SMA/SMK, aplikasi dapat menjadi media latihan mandiri yang memberikan soal, petunjuk, umpan balik, dan progres belajar secara bertahap.
2. Bagi guru dan pengembang media pembelajaran, penelitian ini memberikan contoh integrasi komponen ITS, model prediksi, aturan pedagogis, dan penyimpanan progres dalam aplikasi web.
3. Bagi peneliti berikutnya, penelitian ini menyediakan dasar untuk menguji transfer model dari dataset pendidikan pemrograman ke konteks sekolah menengah, memperbaiki kualitas bank soal, dan mengukur dampak pembelajaran secara eksperimental.

1.6 Sistematika Penulisan

Sistematika penulisan laporan adalah sebagai berikut.

1. Bab 1 menjelaskan urgensi, alasan pemilihan Java, target pengguna, rumusan masalah, batasan, tujuan, dan manfaat.
2. Bab 2 membahas ITS, pembelajaran adaptif, *student model*, *Random Forest*, dataset, XAI, dan evaluasi.
3. Bab 3 menjelaskan *Waterfall*, kebutuhan, dataset, pelatihan model, pemilihan 20 soal, arsitektur, antarmuka, dan rancangan pengujian.
4. Bab 4 menyajikan hasil implementasi, evaluasi model, audit *leakage*, pengujian adaptif dan fungsional, serta respons pengguna.
5. Bab 5 menyajikan simpulan yang menjawab dua tujuan penelitian dan saran pengembangan lanjutan.

