JURNAL INSY PRO Information System and Processing
Volume 11, Nomor 1, Mei 2026 P–ISSN 2597-419X, E-ISSN 2579-468X
Ojs : http://journal.uinalauddin.ac.id/index.php/insypro

SISTEM ASESMEN ADAPTIF BERBASIS AI MENGGUNAKAN Q-LEARNING UNTUK PEMBELAJARAN PEMROGRAMAN BERBASIS OUTCOME-BASED EDUCATION

AI-Based Adaptive Assessment System Using Q-Learning For Programming Learning Based On Outcome-Based Education

Sri Wahyuni[1], Reza Maulana[2], Rahman[2]

[1]Program Studi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Alauddin Makassar
[2]Program Studi Sistem Informasi, Fakultas Sains dan Teknologi, Universitas Islam Negeri Alauddin Makassar

sri.wahyuni@uin-alauddin.ac.id[1]; reza.maulana@uin-alauddin.ac.id[2]; rahman.mallawing@uin-alauddin.ac.id[2]

Abstrak – Evaluasi pembelajaran pemrograman sering menggunakan instrumen yang sama untuk seluruh mahasiswa sehingga informasi yang diperoleh belum selalu cukup granular untuk menggambarkan penguasaan kompetensi individual. Penelitian ini mengembangkan rancangan AI-Based Adaptive Assessment System Using Q-Learning yang mengintegrasikan Outcome-Based Education (OBE), Large Language Model (LLM), Student Model, dan reinforcement learning. Struktur CPL–CPMK–Sub-CPMK–indikator–rubrik digunakan sebagai mekanisme traceability, sedangkan LLM menghasilkan dan mengevaluasi soal secara terkontrol. Student Model merepresentasikan tingkat penguasaan dan confidence pada setiap kompetensi. Q-Learning ditempatkan sebagai pengambil keputusan adaptif untuk memilih target kompetensi, tingkat kesulitan, dan tipe soal berdasarkan state mahasiswa; algoritma ini tidak digunakan sebagai pengganti pengukuran kompetensi. Metodologi menggunakan Research and Development melalui analisis kebutuhan, perancangan arsitektur, pengembangan prototipe, validasi instrumen, dan rancangan eksperimen dengan pembanding non-adaptif. Hasil rancangan menunjukkan alur tertutup dari respons mahasiswa menuju evaluasi AI, pembaruan Student Model, pemberian reward, dan pemilihan soal berikutnya. Artikel ini tidak mengklaim hasil kuantitatif sebelum eksperimen empiris dilakukan. Kontribusi penelitian adalah kerangka adaptasi yang menggabungkan traceability OBE, evaluasi berbasis rubrik dan test case, serta optimasi urutan asesmen berbasis Q-Learning.

Kata kunci: asesmen adaptif, Q-Learning, LLM, pemrograman, outcome-based education

Abstract – This study develops an AI-Based Adaptive Assessment System Using Q-Learning for programming learning within an Outcome-Based Education (OBE) framework. Conventional assessments often expose all learners to the same sequence of items, which may provide limited diagnostic information about individual competency. The proposed system integrates an OBE knowledge base, Large Language Model (LLM), Student Model, assessment engine, and Q-Learning policy. CPL–CPMK–Sub-CPMK–indicator–rubric mappings provide assessment traceability, while the LLM is constrained by competency targets, difficulty, item type, and structured output requirements. The Student Model stores mastery and confidence estimates for each competency. Q-Learning acts as a sequential decision mechanism that selects the next assessment action, such as target competency, difficulty, and item type, rather than directly estimating student ability. The research follows a Research and Development approach covering requirements analysis, architecture design, prototype development, instrument validation, and an experimental evaluation plan against non-adaptive baselines. The resulting design establishes a closed loop from learner response to AI-supported evaluation, Student Model updating, reward calculation, and next-item selection. No empirical performance values are fabricated; quantitative claims are reserved for the planned validation study. The main contribution is a traceable adaptive-assessment architecture that combines OBE, rubric-constrained LLM assessment, executable code testing, and Q-Learning-based sequencing.

Keywords: adaptive assessment, Q-Learning, large language model, programming education, outcome-based education

PENDAHULUAN

Pembelajaran pemrograman menuntut mahasiswa tidak hanya mengingat sintaks, tetapi juga memahami konsep, menyusun algoritma, menelusuri alur program, menulis kode, memperbaiki kesalahan, dan menerapkan solusi pada konteks baru. Karakteristik tersebut membuat asesmen pemrograman lebih kompleks dibandingkan tes pengetahuan deklaratif.

Pada asesmen konvensional, mahasiswa sering menerima paket soal dan urutan item yang sama. Strategi tersebut sederhana dan mudah dikelola, tetapi kurang mampu mengeksplorasi perbedaan penguasaan kompetensi antarmahasiswa secara efisien.

Pendekatan Outcome-Based Education (OBE) menempatkan capaian pembelajaran sebagai pusat perancangan kurikulum dan asesmen. Keterlacakan antara capaian pembelajaran dan bukti asesmen menjadi penting agar skor dapat diinterpretasikan sebagai bukti ketercapaian kompetensi, bukan sekadar nilai ujian. Studi OBE pada pendidikan tinggi menekankan hubungan antara learning outcomes, aktivitas pembelajaran, dan assessment (Tho et al., 2021). Kerangka quality assurance yang lebih baru juga menekankan indikator dan rubrik untuk mengukur attainment secara terstruktur.

Perkembangan generative AI memperluas kemungkinan otomasi asesmen. LLM telah diteliti untuk menghasilkan latihan pemrograman, solusi contoh, dan penjelasan kode. Sarsa et al. (2022) menunjukkan bahwa model generatif dapat menghasilkan latihan yang beragam dan relevan ketika diberi kendala konsep dan konteks, walaupun pengawasan manusia tetap diperlukan. Pada sisi evaluasi, penelitian tugas pemrograman menunjukkan bahwa LLM dapat menghasilkan skor dan umpan balik yang berkorelasi dengan penilaian manusia, tetapi performanya bergantung pada model, jenis tugas, dan strategi prompting (Mohamed et al., 2025).

Temuan mutakhir memperkuat alasan untuk tidak menjadikan LLM sebagai satu-satunya penilai. Jukiewicz (2026) membandingkan 18 LLM pada lebih dari 6.000 solusi mahasiswa dan menemukan perbedaan sistematis dalam perilaku penilaian antarmodel, sementara kesesuaian dengan penilai manusia masih moderat. Sistem asesmen berbasis LLM karena itu memerlukan rubrik eksplisit, audit trail, validasi, dan human-in-the-loop, terutama ketika hasil digunakan untuk keputusan akademik.

Permasalahan berikutnya adalah pemilihan item setelah suatu respons diterima. Computerized adaptive testing memilih item berdasarkan estimasi kemampuan, sedangkan reinforcement learning memungkinkan kebijakan memilih tindakan secara berurutan dengan mempertimbangkan konsekuensi jangka panjang. Li et al. (2023) memformulasikan adaptive learning sebagai Markov decision process dan menunjukkan potensi deep reinforcement learning untuk menemukan kebijakan pembelajaran individual. Penelitian formative adaptive assessment pada mahasiswa pemrograman juga menunjukkan manfaat mekanisme adaptif terhadap performa dan engagement dibandingkan asesmen non-adaptif.

Atas dasar masalah tersebut, penelitian ini merancang AI-Based Adaptive Assessment System Using Q-Learning. Kebaruan utama tidak diletakkan pada penggunaan LLM atau Q-Learning secara terpisah, melainkan pada pemisahan fungsi: OBE menentukan apa yang harus diukur, LLM membantu menghasilkan dan menilai artefak asesmen, Student Model merepresentasikan kondisi kompetensi, dan Q-Learning mengoptimalkan keputusan tentang asesmen berikutnya. Dengan desain tersebut, sistem diharapkan menghasilkan asesmen yang adaptif sekaligus tetap dapat ditelusuri ke indikator capaian pembelajaran.

Pertanyaan penelitian adalah: (1) bagaimana mengintegrasikan OBE, LLM, Student Model, dan Q-Learning dalam satu siklus asesmen adaptif pemrograman; (2) bagaimana memastikan soal dan penilaian AI tetap selaras dengan indikator serta rubrik; dan (3) apakah kebijakan adaptif Q-Learning dapat mengurangi jumlah item atau waktu asesmen tanpa menurunkan kualitas diagnosis kompetensi dibandingkan strategi non-adaptif.

METODOLOGI PENELITIAN

Desain Penelitian

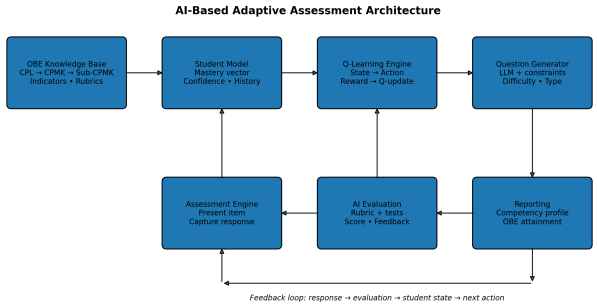
Penelitian menggunakan pendekatan Research and Development (R&D) melalui analisis kebutuhan, pemodelan OBE, perancangan arsitektur, pengembangan prototipe, validasi instrumen, pengujian komponen AI, dan eksperimen terkontrol. Tahap desain dalam artikel ini menghasilkan spesifikasi sistem yang dapat diimplementasikan dan diuji pada mahasiswa mata kuliah pemrograman dasar. Karena data mahasiswa nyata belum digunakan pada naskah ini, tidak ada angka performa eksperimen yang diperlakukan sebagai hasil empiris.

Unit analisis utama adalah interaksi asesmen. Satu interaksi terdiri atas state mahasiswa, action pemilihan item, soal, respons mahasiswa, hasil evaluasi, perubahan Student Model, reward, dan Q-value setelah pembaruan. Struktur ini memungkinkan setiap keputusan adaptif diaudit dan direkonstruksi.

Arsitektur Sistem

Arsitektur yang dirancang memisahkan knowledge layer, assessment layer, learner-model layer, dan decision layer. OBE Knowledge Base menyimpan

CPL, CPMK, Sub-CPMK, indikator, materi/prasyarat, rubrik, dan metadata soal. Question Generator menggunakan LLM dengan constraint tersebut sehingga generasi tidak hanya berbasis prompt bebas. Assessment Engine mengelola sesi dan terminasi. AI Evaluation menggabungkan rubrik, analisis LLM, dan bila memungkinkan eksekusi kode serta test case. Student Model memperbarui estimasi penguasaan, sedangkan Q-Learning membaca state baru dan memilih action berikutnya.



Gambar 1. Arsitektur AI-Based Adaptive Assessment berbasis OBE, LLM, Student Model, dan Q-Learning.

Model OBE dan Student Model

Struktur kompetensi direpresentasikan sebagai graf hierarkis CPL → CPMK → Sub-CPMK → indikator. Setiap indikator memiliki rubrik dan dapat memiliki item konseptual, tracing, coding, atau debugging. Dengan demikian, setiap respons dapat ditelusuri dari item hingga capaian yang diukur. Student Model menyimpan vektor mastery $m = [m_1, m_2, ..., m_n]$ dan confidence $c = [c_1, c_2, ..., c_n]$ untuk n indikator. Nilai mastery diperlakukan sebagai estimasi yang diperbarui berdasarkan kualitas respons, tingkat kesulitan item, dan reliabilitas bukti.

Pada prototipe awal, pembaruan mastery dapat menggunakan pembobotan skor indikator: $m_i(t+1) = (1-\lambda)m_i(t) + \lambda s_i(t)$, dengan $s_i(t)$ sebagai skor terstandar dan λ sebagai laju pembaruan. Formulasi ini sengaja sederhana agar sistem dapat diuji lebih dahulu. Pada penelitian lanjutan, model tersebut dapat dibandingkan dengan Item Response Theory atau Bayesian Knowledge Tracing sehingga ketidakpastian kemampuan dapat dimodelkan secara lebih formal.

Formulasi Q-Learning

Q-Learning digunakan sebagai mekanisme pemilihan strategi asesmen. State s_t dibentuk dari ringkasan Student Model, misalnya kategori mastery rendah, sedang, dan tinggi untuk setiap Sub-CPMK, ditambah confidence dan riwayat item. Action a_t dapat berupa kombinasi target kompetensi, tingkat kesulitan, dan tipe soal. Pembaruan Q-value mengikuti:

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha\,[r_t + \gamma\, max_a\, Q(s_{t+1},a) - Q(s_t,a_t)]$$

α adalah learning rate, γ adalah discount factor, dan r_t adalah reward yang mencerminkan kualitas keputusan asesmen. Strategi ε-greedy dapat digunakan pada fase eksplorasi. Pada implementasi awal, state didiskretisasi agar Q-table tetap terkelola; jika dimensi kompetensi membesar, deep Q-learning dapat dipertimbangkan sebagaimana arah penelitian adaptive learning berbasis reinforcement learning (Li et al., 2023).

Desain Reward

Reward tidak boleh hanya mengoptimalkan skor jawaban. Reward perlu merepresentasikan tujuan pedagogis dan efisiensi diagnosis. Rancangan awal menggunakan komponen peningkatan mastery (+), kesesuaian tingkat kesulitan (+), informasi diagnostik (+), item terlalu mudah atau terlalu sulit (−), serta tidak adanya perubahan setelah beberapa item (−). Nilai numerik final harus dikalibrasi menggunakan data pilot dan penilaian ahli.

Kondisi	Reward awal — Tujuan
Mastery meningkat	+10 — Mendorong action yang memberi bukti peningkatan
Level sesuai kemampuan	+5 — Menghargai item yang informatif
Item terlalu mudah	−1 — Mengurangi item berinformasi rendah
Item terlalu sulit	−2 — Menghindari lompatan kesulitan
Tidak ada progres	−3 — Mendorong remediasi atau perubahan tipe soal

Generasi dan Evaluasi Soal oleh LLM

Generator menerima konteks mata kuliah, Sub-CPMK, indikator, materi/prasyarat, level kognitif, tingkat kesulitan, tipe soal, profil kompetensi, dan format output. Output minimum terdiri atas prompt soal, indikator target, expected answer atau solution outline, rubrik, tingkat kesulitan, dan validation notes. Constraint ini penting karena penelitian generasi latihan pemrograman menunjukkan bahwa LLM mampu menghasilkan artefak yang berguna, tetapi masih memerlukan oversight manusia (Sarsa et al., 2022).

Untuk jawaban kode, evaluasi dirancang sebagai hybrid assessment. Unit test atau test case digunakan untuk bukti correctness yang dapat dieksekusi, sedangkan LLM digunakan untuk aspek seperti penjelasan konsep, kualitas algoritmik, struktur solusi,

dan feedback. Pendekatan hybrid lebih defensible daripada meminta LLM menghasilkan satu skor total tanpa bukti. Temuan Mohamed et al. (2025) dan Jukiewicz (2026) mendukung perlunya pengujian agreement dengan penilai manusia dan pelaporan perilaku model secara transparan.

Prosedur Adaptasi

Pada awal sesi, sistem memuat Student Model dan menentukan state awal. Q-Learning memilih action berdasarkan policy yang sedang dipelajari. Question Generator menghasilkan atau mengambil item yang memenuhi constraint action. Mahasiswa menjawab, kemudian Assessment Engine meneruskan respons ke evaluator. Hasil evaluasi diperiksa terhadap rubrik dan test case; skor indikator dan confidence memperbarui Student Model. Selanjutnya reward dihitung dan Q-value diperbarui. Siklus berulang sampai target kompetensi memenuhi kriteria mastery atau batas item/waktu tercapai.

Terminasi tidak hanya didasarkan pada jumlah soal. Sistem dapat berhenti ketika confidence telah cukup tinggi pada target indikator, ketika tambahan item diperkirakan memberi informasi rendah, atau ketika batas waktu tercapai. Desain ini membuat jumlah item menjadi variabel adaptif yang nantinya dapat dibandingkan dengan baseline.

HASIL DAN PEMBAHASAN

Hasil Rancangan Sistem

Hasil utama tahap ini adalah spesifikasi prototipe yang membentuk closed-loop assessment. Gambar 1 menunjukkan bahwa OBE Knowledge Base menjadi sumber constraint pedagogis, bukan sekadar repositori metadata. Setiap action Q-Learning harus merujuk pada kompetensi yang sah, sehingga policy tidak dapat memilih item yang berada di luar target pembelajaran. Student Model menjadi antarmuka antara pengukuran dan optimasi keputusan: evaluator menghasilkan bukti kompetensi, sedangkan Q-Learning memanfaatkan bukti tersebut untuk memilih tindakan selanjutnya.

Desain ini memperjelas posisi LLM. LLM bukan pengambil keputusan pedagogis utama dan bukan satu-satunya sumber kebenaran penilaian. Perannya dibatasi pada generasi artefak dan interpretasi respons berdasarkan konteks yang telah ditentukan. Batasan tersebut penting karena studi evaluasi LLM menunjukkan adanya variasi antar model dan tingkat kesesuaian yang tidak selalu setara dengan penilai manusia (Mohamed et al., 2025; Jukiewicz, 2026).

Keunggulan lain adalah traceability. Sebuah nilai mastery pada Sub-CPMK dapat ditelusuri ke indikator, item, respons, rubric criterion, evidence dari test case, model LLM, prompt, dan keputusan policy. Audit trail tersebut dapat digunakan untuk debugging sistem, audit akademik, serta analisis kesalahan. Dalam konteks OBE, keluaran akhirnya bukan hanya skor total, tetapi profil attainment yang menunjukkan indikator mana yang telah mastered, developing, atau membutuhkan remediation.

Skenario Uji

Sebagai contoh, target adalah Sub-CPMK “mahasiswa mampu menerapkan struktur perulangan dalam penyelesaian masalah sederhana”. Student Model awal dapat menunjukkan mastery 0,45. Policy memilih tracing tingkat sedang karena action tersebut diperkirakan informatif. Jika respons benar dan confidence meningkat, policy dapat memilih coding tingkat sedang. Jika mahasiswa gagal pada coding tetapi berhasil pada tracing, sistem tidak harus langsung menurunkan difficulty; Q-Learning dapat memilih debugging atau conceptual item untuk memperoleh bukti diagnostik yang lebih spesifik. Setelah respons berikutnya, state diperbarui dan policy belajar dari reward. Contoh ini menggambarkan adaptasi berbasis bukti, bukan aturan threshold tunggal.

Rancangan Evaluasi Empiris

Validasi empiris dirancang menggunakan mahasiswa yang mengikuti mata kuliah pemrograman dasar. Partisipan dibagi ke kelompok sistem adaptif dan baseline, dengan desain yang mempertimbangkan kesetaraan target kompetensi. Baseline minimal terdiri atas urutan soal tetap dan soal acak terstratifikasi berdasarkan kesulitan. Baseline rule-based dapat ditambahkan untuk membedakan manfaat Q-Learning dari sekadar adaptasi berbasis threshold.

Aspek	Metrik dan tujuan
Kualitas soal	Validitas ahli, relevansi indikator, duplicate rate — Memastikan item layak
Kualitas penilaian	MAE/RMSE, weighted κ atau ICC — Menguji agreement AI–ahli
Efektivitas adaptasi	Learning gain, mastery attainment — Menguji perubahan kompetensi
Efisiensi	Jumlah item, waktu asesmen — Mengukur biaya evaluasi
Policy	Cumulative reward, convergence — Menguji pembelajaran agent

Pengalaman pengguna	Usability dan perceived usefulness — Menilai penerimaan sistem

Hipotesis yang akan diuji adalah: H1, sistem Q-Learning menghasilkan learning gain lebih tinggi daripada baseline non-adaptif; H2, sistem mencapai mastery target dengan jumlah item lebih sedikit; H3, penilaian LLM berbasis rubrik memiliki agreement yang dapat diterima dengan penilai ahli; dan H4, mahasiswa menilai sistem lebih relevan dan personal. Uji statistik dipilih setelah pemeriksaan distribusi dan desain data, sedangkan effect size wajib dilaporkan bersama p-value.

Pembahasan

Secara konseptual, kontribusi penelitian terletak pada pemisahan empat fungsi yang sering tercampur dalam sistem AI pendidikan. OBE menentukan target, LLM mengolah artefak asesmen, Student Model mengestimasi kondisi peserta, dan Q-Learning mengoptimalkan urutan tindakan. Pemisahan ini membuat arsitektur lebih dapat dijelaskan dibandingkan sistem yang hanya menghasilkan soal berdasarkan skor sebelumnya.

Dari sisi reinforcement learning, penggunaan Q-Learning lebih tepat dipandang sebagai masalah sequential decision-making. State tidak identik dengan nilai ujian; state adalah representasi ringkas mengenai kompetensi dan ketidakpastian. Reward juga harus diarahkan pada tujuan pedagogis, karena reward yang salah dapat membuat agent mengejar perilaku yang secara matematis menguntungkan tetapi tidak relevan bagi pembelajaran. Untuk ruang kompetensi yang semakin besar, Q-table dapat mengalami ledakan state-action sehingga perlu dipertimbangkan abstraksi state, function approximation, atau deep Q-learning.

Dari sisi LLM, risiko terbesar adalah validitas dan konsistensi. Soal yang tampak benar secara bahasa belum tentu tepat secara pedagogis. Kode yang menghasilkan output benar pada satu test case belum tentu memenuhi indikator algoritmik. Karena itu, validasi sebaiknya bersifat multi-layer: pemeriksaan struktur output, constraint kompetensi, duplicate detection, execution-based testing, rubric scoring, dan audit manusia untuk item atau respons ber-confidence rendah. Temuan terbaru tentang perbedaan perilaku penilaian antar-LLM menunjukkan bahwa pemilihan model harus menjadi variabel eksperimen, bukan keputusan implementasi yang tidak terdokumentasi.

Jika berhasil divalidasi, sistem ini berpotensi mengubah asesmen dari model one-size-fits-all menjadi evidence-driven assessment. Namun, klaim efektivitas harus menunggu hasil eksperimen. Pada tahap sekarang, hasil yang dapat dipertanggungjawabkan adalah rancangan arsitektur, formulasi adaptasi, mekanisme validasi, dan protokol evaluasi; bukan peningkatan skor yang belum diukur.

Keterbatasan Penelitian

Artikel ini belum melaporkan eksperimen mahasiswa sehingga belum dapat menyimpulkan efektivitas Q-Learning secara empiris. Student Model awal juga belum dibandingkan secara langsung dengan IRT atau Bayesian Knowledge Tracing. Kualitas policy bergantung pada desain reward dan jumlah interaksi. Perilaku LLM dapat berubah antarversi dan antarvendor sehingga reproducibility membutuhkan pencatatan model, prompt, parameter inference, dan timestamp. Penggunaan AI dalam keputusan akademik juga menuntut pengamanan data mahasiswa, auditability, dan human oversight.

KESIMPULAN

Penelitian ini menghasilkan rancangan AI-Based Adaptive Assessment System Using Q-Learning untuk asesmen pembelajaran pemrograman berbasis OBE. Sistem mengintegrasikan struktur CPL–CPMK–Sub-CPMK–indikator sebagai traceability layer, LLM sebagai generator dan evaluator yang dikendalikan rubrik, Student Model sebagai representasi mastery dan confidence, serta Q-Learning sebagai pengambil keputusan urutan asesmen. Kontribusi utama terletak pada pemisahan fungsi pengukuran dan optimasi keputusan sehingga Q-Learning tidak diposisikan sebagai pengganti mekanisme psikometri. Rancangan juga memasukkan hybrid evaluation untuk kode, audit trail, human-in-the-loop, serta baseline eksperimen yang memungkinkan pengujian learning gain dan efisiensi. Tahap berikutnya adalah implementasi prototipe, validasi ahli, pilot test, dan eksperimen terkontrol. Hasil kuantitatif harus ditentukan dari data aktual, bukan diasumsikan dari desain.

Saran Pengembangan

Pengembangan berikutnya disarankan memperluas Student Model dengan IRT atau Bayesian Knowledge Tracing, menguji beberapa LLM dan model open-weight, menggunakan knowledge graph untuk memperkuat constraint generasi soal, mengembangkan reward berbasis information gain, dan menerapkan human-in-the-loop untuk kasus confidence rendah. Pengujian pada beberapa mata kuliah dan cohort diperlukan untuk menilai generalisasi sistem.

DAFTAR PUSTAKA

Chrysafiadi, K., Virvou, M., & Tsihrintzis, G. A. (2023). Evaluating the user's experience, adaptivity and learning outcomes of a fuzzy-based intelligent tutoring system for computer programming for academic students in Greece. Education and Information Technologies, 28, 6453–6483. https://doi.org/10.1007/s10639-022-11444-3

Jukiewicz, M. (2026). A systematic comparison of large language models for automated assignment assessment in programming education: Exploring the importance of architecture and vendor. Computers and Education Open, 10, 100364. https://doi.org/10.1016/j.caeo.2026.100364

Li, X., Xu, H., Zhang, J., & Chang, H.-H. (2023). Deep reinforcement learning for adaptive learning systems. Journal of Educational and Behavioral Statistics, 48(2), 220–243. https://doi.org/10.3102/10769986221129847

Mohamed, K., Yousef, M., Medhat, W., Mohamed, E. H., Khoriba, G., & Arafa, T. (2025). Hands-on analysis of using large language models for the auto evaluation of programming assignments. Information Systems, 128, 102473. https://doi.org/10.1016/j.is.2024.102473

Novak, M., & Kermek, D. (2024). Assessment automation of complex student programming assignments. Education Sciences, 14(1), 54. https://doi.org/10.3390/educsci14010054

Sarsa, S., Denny, P., Hellas, A., & Leinonen, J. (2022). Automatic generation of programming exercises and code explanations using large language models. arXiv:2206.11861. https://arxiv.org/abs/2206.11861

Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction (2nd ed.). MIT Press.

Tho, M. A., Tho, V. N., & Hong, B. V. (2021). Outcome-based education for developing study programme in higher education: A case study at Nong Lam University–Ho Chi Minh City. Journal of Technical Education Science, 63, 1–10. https://doi.org/10.54644/jte.63.2021.80

Tio, S., Li, D., & Varakantham, P. (2024). EduQate: Generating adaptive curricula through RMABs in education settings. arXiv:2406.14122. https://arxiv.org/abs/2406.14122

Yousef, M., Mohamed, K., Medhat, W., Mohamed, E. H., Khoriba, G., & Arafa, T. (2025). BeGrading: Large language models for enhanced feedback in programming education. Neural Computing and Applications. https://doi.org/10.1007/s00521-024-10449-y