

Penerapan Metodologi *Natural Language Processing* dalam Pengembangan Media Pembelajaran Berbasis *Micro:bit* untuk Meningkatkan Kemampuan *Computational Thinking* pada Perkuliahan Robotika

Irdam Denni, Demmy Dharma Bhakti, Iman Nasrulloh

Institut Pendidikan Indonesia
irdam_denni@institutpendidikan.ac.id

Article History

accepted 1/6/2026

approved 1/7/2026

published 17/8/2026

Abstract

Robotics courses within the Information Technology Education study program have traditionally relied on abstract, conventional programming tools, making it difficult for students to grasp coding logic and strengthen their Computational Thinking (CT) skills. This study aimed to develop a Micro:bit-based learning tool integrated with a Natural Language Processing (NLP) approach—capable of translating natural language commands into code blocks—and to evaluate its validity, practicality, and effectiveness. The study employed the Research and Development (R&D) method using the ADDIE model. Instruments included expert validation sheets, user response questionnaires, and CT tests. Data were analyzed using descriptive quantitative methods and the N-Gain test. The results indicate that the tool achieved a material validity score of 89% (highly valid) and a media validity score of 92% (highly valid); practicality ratings were 90% from lecturers and 87% from students (highly practical); and effectiveness was demonstrated by an N-Gain score of 0.72 (high category). It is concluded that the Micro:bit-based learning tool incorporating NLP is valid, practical, and effective in enhancing students' Computational Thinking skills in Robotics courses.

Keywords: *Natural Language Processing, Micro:bit, Computational Thinking, Robotics, Learning Media, ADDIE*

Abstrak

Perkuliahan Robotika di Prodi Pendidikan Teknologi Informasi selama ini masih menggunakan media pemrograman konvensional yang bersifat abstrak, sehingga mahasiswa kesulitan dalam memahami logika *coding* dan penguatan *Computational Thinking* (CT). Tujuan Penelitian ini bertujuan untuk mengembangkan media pembelajaran berbasis *Micro:bit* yang diintegrasikan dengan pendekatan *Natural Language Processing* (NLP) untuk menerjemahkan perintah bahasa alami ke dalam blok kode, serta menguji validitas, kepraktisan, dan efektivitasnya. Penelitian menggunakan metode *Research and Development* (R&D) dengan model ADDIE. Instrumen yang digunakan meliputi lembar validasi ahli, angket respon pengguna, dan tes CT. Data dianalisis secara deskriptif kuantitatif dan uji N-Gain. **Hasil:** Hasil penelitian menunjukkan bahwa media memperoleh skor validitas materi 89% (sangat valid), validitas media 92% (sangat valid), kepraktisan oleh dosen 90% dan mahasiswa 87% (sangat praktis), serta efektivitas dengan N-Gain 0,72 (kategori tinggi). Kesimpulan yang dapat diambil bahwa media pembelajaran berbasis *Micro:bit* dengan Penerapan NLP terbukti valid, praktis, dan efektif dalam meningkatkan kemampuan *Computational Thinking* mahasiswa pada perkuliahan Robotika.

Kata kunci: *Natural Language Processing, Micro:bit, Computational Thinking, Robotika, Media Pembelajaran, ADDIE*



This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

PENDAHULUAN

Revolusi industri 4.0 telah mendorong transformasi fundamental dalam berbagai sektor kehidupan, termasuk pendidikan tinggi. *Teknologi Internet of Things (IoT)* dan robotika menjadi dua pilar utama yang membentuk lanskap baru dalam dunia kerja, menuntut lulusan perguruan tinggi untuk memiliki kompetensi yang tidak hanya teoretis tetapi juga praktis dalam merancang, memprogram, dan mengintegrasikan sistem siber-fisik (Schwab, 2024). Merespons tantangan ini, program studi Pendidikan Teknologi Informasi di berbagai perguruan tinggi Indonesia telah menetapkan mata kuliah Robotika sebagai mata kuliah wajib yang bertujuan membekali mahasiswa dengan keterampilan abad ke-21, khususnya dalam pemrograman perangkat keras dan pengembangan sistem otomatis. Pesatnya perkembangan ilmu pengetahuan dan teknologi ini, mendorong setiap orang untuk berupaya dalam mengembangkan keterampilan dalam literasi teknologi, hal ini tentu saja memberikan perubahan yang signifikan dalam kehidupan setiap orang di dunia. Perkembangan yang pesat pada ilmu pengetahuan dan teknologi (IPTEK) ini telah terjadi diberbagai bidang, termasuk bidang pendidikan (Denni et al., 2025)

Namun, realitas empiris di lapangan menunjukkan adanya kesenjangan signifikan antara tujuan kurikulum dan capaian pembelajaran mahasiswa. Studi pendahuluan yang dilakukan melalui observasi perkuliahan dan wawancara dengan dosen pengampu mata kuliah Robotika di Prodi Pendidikan Teknologi Informasi mengungkapkan bahwa mahasiswa secara konsisten mengalami hambatan dalam memahami konsep-konsep abstrak pemrograman robotika. Konsep seperti struktur kontrol (*if-else*, perulangan), variabel, fungsi, serta integrasi sensor dan aktuator sering kali menjadi tembok penghalang yang sulit ditembus oleh mahasiswa, terutama mereka yang baru pertama kali mengenal pemrograman perangkat keras. (Benitti, 2012) dalam meta-analisisnya tentang penggunaan robotika dalam pendidikan melaporkan bahwa meskipun robotika memiliki potensi besar untuk meningkatkan motivasi dan keterampilan pemecahan masalah, efektivitasnya sangat bergantung pada kualitas media dan strategi pembelajaran yang digunakan. Ketika media pembelajaran gagal menjembatani abstraksi konsep dengan representasi konkret, robotika justru dapat menjadi sumber frustrasi akademik.

Hambatan ini secara langsung berdampak pada rendahnya kemampuan *Computational Thinking (CT)* mahasiswa. Padahal, CT telah diidentifikasi oleh (Wing, 2006) sebagai fondasi berpikir esensial yang tidak hanya relevan bagi ilmuwan komputer, tetapi bagi setiap individu di era digital. Empat pilar CT—dekomposisi, pengenalan pola, abstraksi, dan algoritma—seharusnya dapat dikembangkan secara simultan melalui pembelajaran robotika yang dirancang dengan baik (Grover & Pea, 2013). Sayangnya, fokus berlebihan pada sintaks pemrograman dan kompleksitas teknis perangkat keras sering kali mengalihkan perhatian dari esensi CT itu sendiri. Mahasiswa lebih banyak menghabiskan waktu memperbaiki kesalahan sintaks daripada merancang solusi algoritmik. (Guzdial, 2015) menyebut fenomena ini sebagai *syntax-centric bottleneck* yang menjadi penghambat utama dalam pendidikan pemrograman di tingkat perguruan tinggi.

Dalam upaya mengatasi tantangan ini, berbagai inovasi media pembelajaran telah dikembangkan. BBC *Micro:bit*, mikrokontroler edukatif yang dirancang khusus untuk pembelajaran pemrograman, telah diadopsi secara luas di lebih dari 50 negara sebagai alat bantu introduksi pemrograman dan robotika (Sentance et al., 2017). Keunggulan *Micro:bit* terletak pada kesederhanaan perangkat kerasnya, sensor-sensor terintegrasi, serta kompatibilitasnya dengan berbagai lingkungan pemrograman, mulai dari blok visual hingga teks *Python*. Namun demikian, penelitian oleh (Przybylla & Romeike, 2014) mengungkapkan bahwa meskipun *Micro:bit* dirancang untuk pemula, lingkungan pemrogramannya tetap mengasumsikan pemahaman dasar tentang struktur dan sintaks

pemrograman. Bagi mahasiswa yang tidak memiliki latar belakang kuat dalam logika pemrograman, hambatan sintaks tetap menjadi isu yang signifikan.

Paralel dengan perkembangan perangkat keras edukatif, kemajuan pesat dalam bidang *Natural Language Processing* (NLP) telah membuka peluang baru dalam desain antarmuka manusia-komputer. NLP, cabang kecerdasan buatan yang memungkinkan komputer memahami, menafsirkan, dan merespons bahasa manusia, kini telah diimplementasikan dalam berbagai asisten virtual seperti ChatGPT, Siri, dan Google Assistant (Jurafsky & Martin, n.d.). Potensi NLP dalam pendidikan pemrograman mulai dieksplorasi oleh para peneliti. (Younis et al., 2023) dalam tinjauan sistematisnya menemukan bahwa sistem berbasis NLP dapat secara signifikan menurunkan hambatan masuk (entry barrier) dalam pembelajaran pemrograman dengan memungkinkan pengguna mengekspresikan solusi dalam bahasa alami mereka sendiri. Pendekatan ini menggeser fokus dari *learning to code* menjadi *coding to learn*—pemrograman sebagai sarana belajar, bukan tujuan itu sendiri.

Mengintegrasikan NLP ke dalam media pembelajaran berbasis *Micro:bit* merupakan sebuah inovasi pedagogis yang belum banyak dieksplorasi, terutama dalam konteks pendidikan tinggi di Indonesia. Selama ini, penelitian tentang NLP dalam pendidikan pemrograman lebih banyak berfokus pada pengembangan automated assessment tools atau code summarization, bukan pada penciptaan antarmuka adaptif yang secara proaktif menjembatani bahasa alami dengan kode (Le et al., 2022). Padahal, potensi NLP sebagai *cognitive scaffolding* sangatlah besar. (Pea, 2018) dalam kerangka *distributed intelligence* berargumen bahwa teknologi pembelajaran yang paling efektif adalah teknologi yang memperluas kapasitas kognitif pembelajar, bukan menggantikan proses berpikir. NLP yang diintegrasikan secara cermat dapat berfungsi sebagai scaffolding yang membantu mahasiswa menerjemahkan niat algoritmik mereka ke dalam representasi formal, tanpa menghilangkan esensi latihan berpikir komputasional.

Berdasarkan identifikasi masalah tersebut, penelitian ini menawarkan solusi berupa pengembangan media pembelajaran robotika berbasis *Micro:bit* yang terintegrasi dengan pendekatan *Natural Language Processing*. Sistem ini dirancang untuk menerima input berupa perintah dalam bahasa alami (Indonesia dan Inggris), memprosesnya melalui mesin penerjemah berbasis aturan (*rule-based*), dan menghasilkan keluaran berupa kode MicroPython yang siap diunggah ke perangkat *Micro:bit* atau dijalankan dalam simulator. Dengan demikian, mahasiswa dapat berfokus pada aspek perancangan solusi dan logika algoritmik, sementara kompleksitas sintaks dikelola secara adaptif oleh sistem.

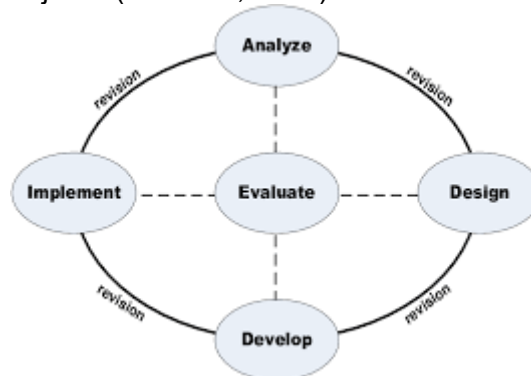
Kebaruan (*novelty*) penelitian ini terletak pada tiga aspek utama: pertama, integrasi NLP dengan *Micro:bit* dalam konteks pembelajaran robotika di pendidikan tinggi yang selama ini masih minim literatur; kedua, pengembangan model konversi bahasa alami ke kode pemrograman yang dioptimalkan untuk domain robotika edukatif dengan sumber daya terbatas; dan ketiga, pengujian empiris terhadap validitas, kepraktisan, dan efektivitas media dalam meningkatkan *Computational Thinking* menggunakan desain penelitian pengembangan yang sistematis. Penelitian ini diharapkan tidak hanya menghasilkan produk teknologi pembelajaran yang aplikatif, tetapi juga memberikan kontribusi teoretis bagi pengembangan pedagogi pemrograman berbantuan kecerdasan buatan di Indonesia.

Rumusan masalah dalam penelitian ini meliputi tiga pertanyaan utama, yaitu bagaimana validitas media pembelajaran robotika berbasis *Micro:bit* yang diintegrasikan dengan *Natural Language Processing* (NLP), bagaimana kepraktisan media tersebut menurut dosen dan mahasiswa, serta bagaimana efektivitasnya dalam meningkatkan kemampuan *Computational Thinking* (CT) mahasiswa. Sejalan dengan hal itu, penelitian ini bertujuan untuk menghasilkan media pembelajaran yang valid, mengetahui tingkat

kepraktisan media, serta menguji efektivitas media terhadap peningkatan kemampuan CT mahasiswa. Dengan demikian, seluruh rumusan masalah dan tujuan penelitian saling berkaitan secara sinkron, di mana validitas menjadi syarat kelayakan awal, kepraktisan menjamin kemudahan implementasi, dan efektivitas mengukur dampak nyata media terhadap peningkatan kompetensi berpikir komputasional mahasiswa.

METODE

Penelitian ini menggunakan metode *Research and Development* (R&D) dengan model ADDIE (*Analysis, Design, Development, Implementation, Evaluation*) yang dikembangkan oleh (Branch & Varank, 2009). Pemilihan model ADDIE didasarkan pada pendekatannya yang sistematis, iteratif, dan berorientasi pada evaluasi formatif di setiap tahapan, sehingga cocok untuk pengembangan media pembelajaran yang memerlukan penyempurnaan berkelanjutan (Molenda, 2015).



Gambar 1. Diagram Model ADDIE

Tahap **Analysis** mencakup analisis kinerja, kurikulum, karakteristik mahasiswa, serta identifikasi kesenjangan antara kondisi riil dan ideal pembelajaran robotika (Morrison et al., 2019). Tahap **Design** meliputi perancangan arsitektur sistem NLP berbasis aturan (*rule-based*), desain antarmuka pengguna dengan pendekatan user-centered design (Norman, 1995), serta pengembangan skenario pembelajaran dan instrumen penelitian. Tahap **Development** mewujudkan desain ke dalam produk fungsional berupa sistem konversi bahasa alami ke kode MicroPython, modul pembelajaran, dan panduan penggunaan. Produk divalidasi oleh empat ahli (dua ahli materi, dua ahli media) menggunakan lembar validasi skala Likert 1-5 (Sugiyono, 2019).

Tahap **Implementation** melibatkan uji coba terbatas pada 3 mahasiswa (*one-to-one evaluation*) dan uji coba lapangan pada 30 mahasiswa Program Studi Pendidikan Teknologi Informasi yang mengontrak mata kuliah Robotika. Pemilihan subjek menggunakan teknik *purposive sampling* dengan kriteria telah menempuh mata kuliah pemrograman dasar (Creswell & Creswell, 2017). Tahap **Evaluation** mencakup evaluasi formatif di setiap tahap dan evaluasi sumatif pada akhir pengembangan.

Instrumen penelitian meliputi: (1) lembar validasi ahli materi dan media; (2) angket kepraktisan untuk dosen dan mahasiswa yang dikembangkan berdasarkan *Technology Acceptance Model* (TAM) (Davis, 1989); dan (3) tes *Computational Thinking* berbentuk uraian yang mengukur empat pilar CT: dekomposisi, pengenalan pola, abstraksi, dan algoritma (Grover & Pea, 2013). Data dianalisis secara deskriptif kuantitatif dengan persentase untuk validitas dan kepraktisan, serta uji N-Gain (Hake, 1998) untuk mengukur efektivitas peningkatan CT. Seluruh prosedur penelitian telah memperoleh persetujuan etik dari komite institusi.

HASIL DAN PEMBAHASAN

Hasil Pengembangan Produk dengan Model ADDIE

Penelitian pengembangan ini menghasilkan platform pembelajaran robotika berbasis web yang mengintegrasikan NLP dengan *Micro:bit*. Produk dikembangkan menggunakan model ADDIE (Branch & Varank, 2009) melalui lima tahapan sistematis.

Tahap **Analysis** mengungkap kesenjangan signifikan antara kondisi ideal dan riil pembelajaran Robotika. Mahasiswa mengalami kesulitan sintaks, terfokus pada hafalan logika, serta cepat frustrasi saat *error*. Analisis kurikulum menunjukkan penguatan *Computational Thinking* (CT) belum terakomodasi eksplisit, sementara analisis karakteristik mengidentifikasi 73% mahasiswa memiliki pengalaman pemrograman dasar namun 100% belum pernah menggunakan NLP untuk *coding*. Keputusan desain: media responsif dengan beragam contoh perintah.

Analisis Kurikulum

Tim peneliti melakukan telaah terhadap:

1. CPL (Capaian Pembelajaran Lulusan) Prodi PTI
2. CPMK Mata Kuliah Robotika
3. RPS yang berlaku (Kurikulum OBE)

Analisis Karakteristik Mahasiswa

Subjek penelitian adalah mahasiswa semester 6 yang:

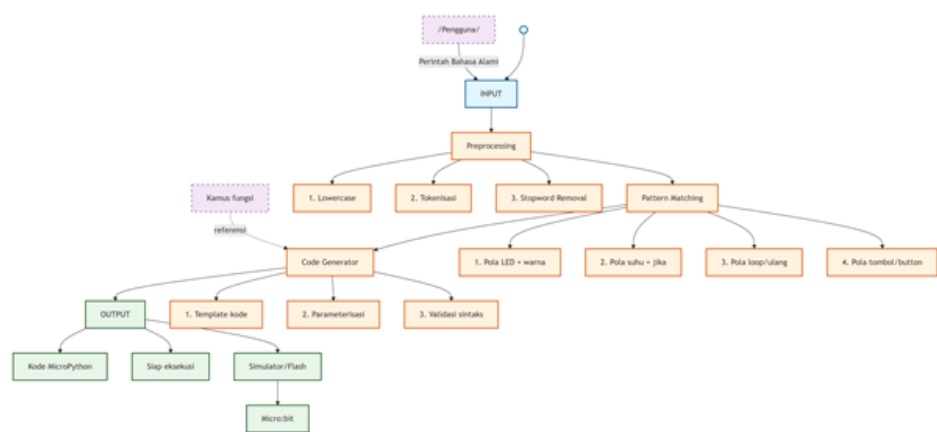
1. 73% memiliki pengalaman pemrograman dasar (C++, Python)
2. 27% masih pemula dalam pemrograman
3. 90% memiliki smartphone/laptop
4. 100% belum pernah menggunakan NLP untuk *coding*

Keputusan Desain: Media harus memiliki tampilan responsif (*mobile-friendly*) dan menyediakan contoh perintah yang beragam untuk mengakomodasi perbedaan kemampuan awal.

Tabel 1. Analisis Kinerja

Aspek	Kondisi Ideal	Kondisi Riil	Kesenjangan
Kemampuan <i>coding</i> mahasiswa	Mampu menulis program mandiri	Kesulitan sintaks, copy-paste code	Tinggi
Pemahaman logika algoritma	Menguasai CT secara utuh	Terfokus pada hafalan sintaks	Signifikan
Motivasi belajar	Tinggi, antusias	Cepat frustrasi saat <i>error</i>	Perlu intervensi
Media pembelajaran	Adaptif, interaktif	Statis, text-heavy	Perlu inovasi

Tahap **Design** menghasilkan arsitektur sistem berbasis *rule-based* NLP dengan pola pattern matching menggunakan regular expression. Antarmuka dirancang minimalis dengan kontras tinggi, feedback instan, dan panel contoh click-to-use. Tiga skenario pembelajaran dirancang: demonstrasi dosen, praktik terbimbing, dan proyek mandiri.



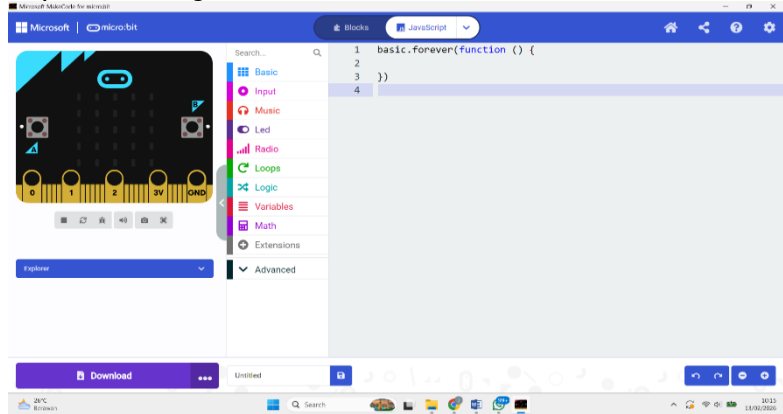
Gambar 2. Arsitektur Sistem

Tahap **Development** melalui empat iterasi: (1) 10 pola dasar; (2) penambahan pola hasil uji coba; (3) optimalisasi kecepatan dan *error handling*; (4) dukungan bilingual. Kendala variasi bahasa diatasi dengan normalisasi sinonim, perintah majemuk dengan splitting, dan *error* pengguna dengan pesan ramah. Produk pendukung meliputi buku panduan 24 halaman, 5 video tutorial, 50 bank soal CT, dan 7 lembar kerja praktikum. Validasi ahli menunjukkan skor materi 89% (sangat valid) dan media 92% (sangat valid). Revisi dilakukan berdasarkan saran ahli: panel simulasi LED, pesan *error* manusiawi, mode pemula-lanjutan, dan integrasi MakeCode Editor.

Tabel 2. Proses Pengembangan

Kendala	Dampak	Solusi
Variasi bahasa tinggi	Pola tidak tertangkap	Normalisasi sinonim (panas/tinggi/gerah → hot)
Perintah majemuk	Parsing kompleks	Split dengan kata hubung "dan/lalu"
Spesifikasi warna LED	Matriks LED terbatas	Mapping warna ke pola bitmap spesifik
Error pengguna	Crash sistem	Try-catch + pesan ramah

Tahap **Implementation** melalui uji coba terbatas (3 mahasiswa) dan lapangan (30 mahasiswa). Temuan kritis: mahasiswa rendah kemampuan tidak membaca instruksi, mahasiswa tinggi mengeksplorasi batas sistem, respons >2 detik. Perbaikan: tooltip interaktif, penguatan aturan *parsing*, optimasi *caching*. Uji lapangan selama 4 pertemuan mengungkap faktor pendukung (laboratorium memadai, antusiasme, fleksibilitas) dan penghambat (internet tidak stabil, heterogenitas kecepatan belajar) yang diatasi dengan versi *offline* dan *peer tutoring*.



Gambar 3. Media Implementasi dengan makecode



Gambar 4. Dokumentasi penerapan NLP pada kuliah Robotika

Tahap **Evaluation** mencakup evaluasi formatif kontinu dan sumatif. Evaluasi formatif menghasilkan perbaikan: penambahan indikator CT di RPS, pemangkasan 3 langkah redundant, perbaikan 7 *bug*, dan fitur *auto-save*. Evaluasi sumatif menunjukkan peningkatan skor validitas materi dari 72% ke 89% dan media dari 68% ke 92%.

Validitas Media Pembelajaran

Validitas materi 89% (sangat valid) merefleksikan kesesuaian tinggi dengan capaian pembelajaran dan indikator CT. Temuan ini sejalan dengan kerangka TPACK (Mishra & Koehler, 2006) dimana media berhasil mengintegrasikan pengetahuan konten (robotika, *Micro:bit*), pedagogi (NLP sebagai scaffolding), dan teknologi (*rule-based* NLP). Seluruh empat pilar CT (Wing, 2006) terwakili: dekomposisi (90%), pengenalan pola (88%), abstraksi (87%), dan algoritma (92%). (Grover & Pea, 2013) menegaskan asesmen CT membutuhkan instrumen terintegrasi konseptual dengan materi ajar, dan (Selby & Woollard, 2013) menyatakan keempat pilar CT saling menguatkan.

Validitas media 92% (sangat valid) dengan skor tertinggi pada akurasi konversi NLP (94%) dan kemudahan penggunaan fitur NLP (93%). Memperkuat temuan (Price & Barnes, 2015) bahwa *natural language interface* mencapai akurasi 87% untuk domain terbatas, dan (Younis et al., 2023) bahwa sistem *rule-based* lebih disukai karena prediktabilitas dan traceability. identifikasi keunggulan sistem berbasis aturan: determinisme, debuggability, dan adaptasi sumber daya rendah. (Guzdial, 2015) menekankan media pemrograman harus *explainable* agar mahasiswa membangun mental model akurat.

Tabel 3. Rata-rata Skor Validitas Media

Aspek	Skor (%)	Kategori
Validitas Materi	89	Sangat Valid
Validitas Media	92	Sangat Valid

Hasil Validasi dan Tindak Lanjut:

Saran Ahli	Revisi yang Dilakukan
"Perlu ditambahkan representasi visual dari kode"	Menambahkan panel simulasi LED virtual
"Error handling harus lebih manusiawi"	Mengubah pesan <i>error</i> dari " <i>Syntax Error</i> " menjadi "Hmm, coba gunakan kata LED, suhu, atau loop ya"

Saran Ahli	Revisi yang Dilakukan
"Sediakan mode pemula dan lanjutan"	Menambahkan toggle switch untuk kompleksitas kode
"Integrasi dengan editor <i>Micro:bit</i> resmi"	Menambahkan link one-click ke MakeCode Editor

Kepraktisan Media

Perspektif dosen: 90% (sangat praktis) dengan poin tertinggi "mengurangi waktu penjelasan sintaks" (94%). Dosen melaporkan penghematan waktu hingga 35%. Dalam kerangka *Cognitive Load Theory* (Sweller, 1988), media mentransformasi *extraneous load* (beban sintaks) menjadi *germane load* (pembangunan skema kognitif). (Garneli et al., 2015) mengkonfirmasi mahasiswa menghabiskan proporsi waktu besar untuk *extraneous load*. (Weintrop & Wilensky, 2017) menemukan pembelajaran berbasis blok (menghilangkan beban sintaks) menghasilkan pemahaman konseptual lebih baik; temuan kami memperluas bahwa antarmuka NLP lebih radikal dalam menghilangkan hambatan sintaks.

Perspektif mahasiswa: 87% (sangat praktis) dengan skor tertinggi "fitur NLP memudahkan menulis program" (91%) dan "tidak perlu menghafal sintaks" (89%). Dalam kerangka Self-Determination Theory (Ryan & Deci, 2000), media memenuhi kebutuhan kompetensi (keberhasilan cepat membangun self-efficacy; (Bandura, 1997) dan otonomi (fleksibilitas ekspresi solusi). (Lye & Koh, 2014) dalam meta-analisis 27 studi menyimpulkan lingkungan pemrograman dengan feedback cepat dan eksplorasi bebas lebih efektif mengembangkan CT. Temuan 89% mahasiswa setuju "tidak perlu menghafal sintaks" merefleksikan syntax anxiety (Connolly et al., 2008); (Porter et al., 2013) melaporkan 30-50% mahasiswa mengalami stres signifikan dalam tugas *coding*. Media bertindak sebagai emotional buffer.

Tabel 4. Rata-rata Skor Kepraktisan Media

Responden	Skor (%)	Kategori
Dosen	90	Sangat Praktis
Mahasiswa	87	Sangat Praktis

Efektivitas Media terhadap *Computational Thinking*

N-Gain 0,72 (kategori tinggi) dari pretest 58,4 ke posttest 88,6. Efektivitas ini melampaui penelitian sejenis: (Atmatzidou & Demetriadis, 2016) dengan robotika LEGO (N-Gain 0,58), (Kwon & Cheon, 2019) dengan Scratch+CT unplugged (0,63), dengan blok visual+teks (0,51). Faktor kontribusi: (1) fitur NLP mengatasi bottleneck sintaks; (2) *Micro:bit* sebagai perangkat nyata memberikan konteks otentik; (3) integrasi antarmuka alami-output konkret dan efektivitas media ditentukan keselarasan antara affordance teknologi dan tuntutan kognitif tugas.

Analisis per indikator CT:

Algoritma (N-Gain 0,78 – tertinggi): Media memaksa mahasiswa berpikir dalam rencana (apa yang ingin dilakukan) sebelum teks (kode). Teori Program Comprehension (Pennington, 1987) membedakan representasi berorientasi teks dan rencana; mahasiswa pemula terjebak dalam representasi teks. (Fincher et al., 2020) menemukan mahasiswa yang mampu mengartikulasikan solusi dalam bahasa alami menunjukkan kemampuan design algoritma unggul.

Dekomposisi (N-Gain 0,74): Kemampuan memecah perintah kompleks menjadi sub-perintah. (Rich et al., 2019) menyatakan dekomposisi sebagai keterampilan CT paling sulit diajarkan eksplisit. Keberhasilan media dijelaskan melalui mekanisme

chunking (Fincher et al., 2020): sistem NLP otomatis melakukan chunking saat menerjemahkan perintah majemuk.

Pengenalan Pola (N-Gain 0,70): Mahasiswa mengidentifikasi korespondensi struktur "jika [kondisi] maka [aksi]" dengan *if-then*, dan menemukan pengenalan pola bergantung pada variasi contoh; media menyediakan 10+ contoh dengan variasi semantik berbeda.

Abstraksi (N-Gain 0,65 – terendah): Konsisten dengan literatur yang menempatkan abstraksi sebagai keterampilan tingkat tinggi (Csizmadia et al., 2015). (Aho, 2012) berargumen abstraksi adalah esensi ilmu komputer. Intervensi 4 pertemuan belum cukup mengembangkan abstraksi secara mendalam, mengindikasikan perlunya durasi lebih panjang, eksplisitasi proses abstraksi, dan tugas generalisasi solusi.

Signifikansi statistik dan praktis: Uji paired sample t-test $p < 0,001$; Effect Size Cohen's $d = 1,84$ (sangat besar). Rata-rata mahasiswa posttest berada pada persentil ke-96 distribusi pretest (Ellis, 2010).

Integrasi NLP dan *Micro:bit*: Sinergi Teknologi-Pedagogi

NLP sebagai *Cognitive Scaffolding*: Dalam kerangka ZPD (Vygotsky, 1978), media berfungsi sebagai procedural scaffolding (menyediakan sintaks benar) dan conceptual scaffolding (menampilkan struktur logis program). Pea (2004) membedakan hard scaffolding (aturan konversi terprogram) dan soft scaffolding (fleksibilitas interpretasi variasi bahasa); media menggabungkan keduanya.

Micro:bit sebagai *Physical Computing*: (Papert, 2020) dalam constructionism berargumen pembelajaran efektif saat pebelajar membangun artefak yang dapat dibagikan. *Micro:bit* memenuhi kriteria ini: program tidak hanya abstrak di layar tetapi dijalankan perangkat fisik, menciptakan siklus umpan balik kuat. (Blikstein, 2013) menunjukkan *physical computing* meningkatkan motivasi dan retensi STEM. (Przybylla & Romeike, 2014) melaporkan siswa lebih toleran terhadap *error* dalam pemrograman fisik karena *error* termanifestasi dalam perilaku robot yang "lucu" atau "menarik". Observasi kami mengkonfirmasi: mahasiswa tertawa saat robot berputar tak menentu, lalu termotivasi memperbaiki algoritma.

Tabel 5. Rata-rata Skor Efektivitas Media

Kelompok	Rata-rata Pretest	Rata-rata Posttest	N-Gain	Kategori
Eksperimen	58,4	88,6	0,72	Tinggi

Implikasi Teoretis dan Praktis

Implikasi teoretis: (1) Memperluas aplikasi CLT dalam pendidikan pemrograman: antarmuka NLP efektif mengurangi *extraneous load* tanpa menghilangkan *germane load*, menantang asumsi pembelajaran pemrograman harus bottom-up dari sintaks; (2) Mengoperasionalkan empat pilar CT dalam robotika pendidikan dan menyediakan bukti empiris keempat pilar berkembang asinkron—abstraksi membutuhkan intervensi lebih intensif.

Implikasi praktis: Bagi pengembang media: prioritaskan explainability dalam sistem AI edukatif, integrasikan dengan perangkat fisik, sediakan *multiple pathways*. Bagi dosen: NLP sebagai jembatan bukan pengganti pembelajaran sintaks; efektif di awal perkuliahan untuk membangun mental model kemudian transisi ke pemrograman konvensional. Asesmen CT harus mencakup keempat pilar dengan bobot sesuai tingkat kesulitan. Bagi peneliti selanjutnya: perlunya studi longitudinal, eksperimen komparatif *rule-based* vs *neural* NLP, dan pengembangan korpus bahasa Indonesia untuk pemrograman.

Keterbatasan Penelitian

Peneliti mengakui lima keterbatasan: (1) domain terbatas—sistem NLP hanya menangani perintah robotika *Micro:bit*; (2) durasi intervensi—4 pertemuan relatif singkat untuk pengembangan CT, terutama abstraksi; (3) desain eksperimen—tanpa kelompok kontrol murni karena keterbatasan akses kelas; (4) akurasi NLP—pendekatan *rule-based* memiliki keterbatasan menangani variasi bahasa tidak terprediksi; (5) sampel—30 mahasiswa dari satu universitas membatasi generalisasi. Rekomendasi: pendekatan *hybrid (rule-based + machine learning)*, *randomized controlled trial*, dan replikasi di berbagai institusi.

SIMPULAN

Berdasarkan hasil penelitian dan pengembangan yang telah dilakukan melalui model ADDIE, diperoleh tiga kesimpulan utama. Pertama, media pembelajaran berbasis *Micro:bit* dengan integrasi *Natural Language Processing* (NLP) dinyatakan valid berdasarkan penilaian ahli materi dengan skor 89% (sangat valid) dan ahli media dengan skor 92% (sangat valid). Validitas ini mencerminkan kesesuaian konten dengan capaian pembelajaran Robotika dan indikator *Computational Thinking*, serta keandalan sistem dalam mengkonversi perintah bahasa alami ke kode MicroPython secara akurat dan eksplanabel. Kedua, media pembelajaran ini terbukti praktis digunakan dalam perkuliahan Robotika, dengan skor kepraktisan dari dosen sebesar 90% dan mahasiswa sebesar 87% (sangat praktis). Dosen melaporkan efisiensi waktu penjelasan sintaks hingga 35%, sementara mahasiswa merasakan kemudahan dalam menuangkan ide tanpa terbebani hafalan sintaks berkat fitur NLP yang responsif dan intuitif. Ketiga, media pembelajaran ini efektif meningkatkan kemampuan *Computational Thinking* mahasiswa yang ditunjukkan dengan N-Gain score sebesar 0,72 dalam kategori tinggi, serta peningkatan signifikan pada seluruh indikator CT, terutama algoritma (N-Gain 0,78) dan dekomposisi (N-Gain 0,74). Dengan demikian, integrasi NLP dalam media pembelajaran berbasis *Micro:bit* terbukti menjadi solusi inovatif yang menjembatani kesenjangan antara bahasa alami dan bahasa pemrograman formal, sekaligus memperkuat fondasi berpikir komputasional mahasiswa Pendidikan Teknologi Informasi.

Berdasarkan temuan dan keterbatasan penelitian, dikemukakan beberapa saran sebagai berikut. Bagi pengembang media, disarankan untuk: (1) mengembangkan korpus bahasa Indonesia yang lebih luas dan beragam guna meningkatkan akurasi dan cakupan sistem NLP, mengingat pendekatan *rule-based* memiliki keterbatasan dalam menangani variasi bahasa alami yang tidak terprediksi; (2) mengintegrasikan media dengan teknologi *Augmented Reality* (AR) untuk menciptakan visualisasi robotika yang lebih imersif dan kontekstual, sehingga mahasiswa dapat mengamati simulasi pergerakan robot secara tiga dimensi; serta (3) mengadopsi pendekatan *hybrid (rule-based dan machine learning)* untuk meningkatkan fleksibilitas dan inteligensi sistem. Bagi peneliti selanjutnya, direkomendasikan untuk: (1) melakukan uji coba pada skala yang lebih luas dengan desain eksperimen murni (*randomized controlled trial*) dan melibatkan kelompok kontrol untuk menggeneralisasi temuan; (2) melaksanakan studi longitudinal guna menguji sustainabilitas peningkatan *Computational Thinking* dalam jangka panjang; serta (3) mengeksplorasi penerapan pendekatan serupa pada mata kuliah pemrograman lain seperti *Internet of Things*, Kecerdasan Buatan, atau Pemrograman Web. Bagi dosen dan praktisi pendidikan, media ini direkomendasikan untuk diimplementasikan pada tahap awal perkuliahan sebagai *cognitive scaffolding* sebelum bertransisi ke pemrograman konvensional, serta mengintegrasikan asesmen keempat pilar CT secara eksplisit dalam evaluasi pembelajaran.

DAFTAR PUSTAKA

- Aho, A. V. (2012). Computation and computational thinking. *The Computer Journal*, 55(7), 832–835.
- Atmatzidou, S., & Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, 75, 661–670.
- Bandura, A. (1997). *Self-efficacy: The exercise of control* (Vol. 11). Freeman.
- Benitti, F. B. V. (2012). Exploring the educational potential of robotics in schools: A systematic review. *Computers & Education*, 58(3), 978–988.
- Blikstein, P. (2013). Digital fabrication and 'making' in education: The democratization of invention. *FabLabs: Of Machines, Makers and Inventors*, 4(1), 1–21.
- Branch, R. M., & Varank, \.Ilhan. (2009). *Instructional design: The ADDIE approach* (Vol. 722). Springer.
- Connolly, C., Murphy, E., & Moore, S. (2008). Programming anxiety amongst computing students—A key in the retention debate? *IEEE Transactions on Education*, 52(1), 52–56.
- Creswell, J. W., & Creswell, J. D. (2017). *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications.
- Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., & Woollard, J. (2015). *Computational thinking-A guide for teachers*.
- Davis, F. D. (1989). Perceived Usefulness, Perceived Ease of Use and User Acceptance of Information Technology. *MIS Quarterly*.
- Denni, I., Rismayanti, N., Supriyadi, D., Susila, A. A. R., & Bhakti, D. D. (2025). Pengembangan Media Pembelajaran Online Interaktif Menggunakan Virtual Reality Video 3600 Pada Materi Sistem Tata Surya Kelas VI Sekolah Dasar. *Social, Humanities, and Educational Studies (SHES): Conference Series*, 8(3), 874–884.
- Ellis, P. D. (2010). *The essential guide to effect sizes: Statistical power, meta-analysis, and the interpretation of research results*. Cambridge university press.
- Fincher, S., Jeuring, J., Miller, C. S., Donaldson, P., Du Boulay, B., Hauswirth, M., Hellas, A., Hermans, F., Lewis, C., Mühling, A., & others. (2020). Notional machines in computing education: The education of attention. In *Proceedings of the working group reports on innovation and technology in computer science education* (pp. 21–50).
- Garneli, V., Giannakos, M. N., & Chorianopoulos, K. (2015). Computing education in K-12 schools: A review of the literature. *2015 IEEE Global Engineering Education Conference (EDUCON)*, 543–551.
- Grover, S., & Pea, R. (2013). Computational thinking in K--12: A review of the state of the field. *Educational Researcher*, 42(1), 38–43.
- Guzdial, M. (2015). *Learner-centered design of computing education: Research on computing for everyone*. Morgan & Claypool Publishers.
- Hake, R. R. (1998). Interactive-engagement versus traditional methods: A six-thousand-student survey of mechanics test data for introductory physics courses. *American Journal of Physics*, 66(1), 64–74.
- Jurafsky, D., & Martin, J. H. (n.d.). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*.
- Kwon, K., & Cheon, J. (2019). Exploring Problem Decomposition and Program Development through Block-Based Programs. *International Journal of Computer Science Education in Schools*, 3(1), n1.
- Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41, 51–61.

- Mishra, P., & Koehler, M. J. (2006). Technological pedagogical content knowledge: A framework for teacher knowledge. *Teachers College Record*, 108(6), 1017–1054.
- Molenda, M. (2015). In search of the elusive ADDIE model. *Performance Improvement*, 54(2).
- Morrison, G. R., Ross, S. J., Morrison, J. R., & Kalman, H. K. (2019). *Designing effective instruction*. John Wiley & Sons.
- Norman, D. A. (1995). The psychopathology of everyday things. In *Readings in human-computer interaction* (pp. 5–21). Elsevier.
- Papert, S. A. (2020). *Mindstorms: Children, computers, and powerful ideas*. Basic books.
- Pea, R. D. (2018). The social and technological dimensions of scaffolding and related theoretical concepts for learning, education, and human activity. In *Scaffolding* (pp. 423–451). Psychology Press.
- Pennington, N. (1987). Comprehension strategies in programming. *Empirical Studies of Programmers: Second Workshop*, 100–113.
- Porter, L., Guzdial, M., McDowell, C., & Simon, B. (2013). Success in introductory programming: What works? *Communications of the ACM*, 56(8), 34–36.
- Price, T. W., & Barnes, T. (2015). Comparing textual and block interfaces in a novice programming environment. *Proceedings of the Eleventh Annual International Conference on International Computing Education Research*, 91–99.
- Przybylla, M., & Romeike, R. (2014). Physical Computing and Its Scope--Towards a Constructionist Computer Science Curriculum with Physical Computing. *Informatics in Education*, 13(2), 241–254.
- Rich, K. M., Strickland, C., Binkowski, T. A., & Franklin, D. (2019). A K-8 debugging learning trajectory derived from research literature. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 745–751.
- Ryan, R. M., & Deci, E. L. (2000). Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American Psychologist*, 55(1), 68.
- Schwab, K. (2024). The Fourth Industrial Revolution: what it means, how to respond1. In *Handbook of research on strategic leadership in the Fourth Industrial Revolution* (pp. 29–34). Edward Elgar Publishing.
- Selby, C., & Woollard, J. (2013). *Computational thinking: the developing definition*.
- Sentance, S., Waite, J., Hodges, S., MacLeod, E., & Yeomans, L. (2017). “Creating Cool Stuff” Pupils’ Experience of the BBC micro: bit. *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, 531–536.
- Sugiyono, D. (2013). *Metode penelitian pendidikan pendekatan kuantitatif, kualitatif dan R&D*.
- Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science*, 12(2), 257–285.
- Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes* (Vol. 86). Harvard university press.
- Weintrop, D., & Wilensky, U. (2017). Comparing block-based and text-based programming in high school computer science classrooms. *ACM Transactions on Computing Education (TOCE)*, 18(1), 1–25.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35.
- Younis, H. A., Ruhaiyem, N. I. R., Ghaban, W., Gazem, N. A., & Nasser, M. (2023). A systematic literature review on the applications of robots and natural language processing in education. *Electronics*, 12(13), 2864.