

IMPLEMENTASI LOGIKA KOMPUTASI DENGAN PARADIGMA PEMROGRAMAN TERSTRUKTUR

Diefa Anasya Asyshura¹, Mhd Rizki Syahputra², Lisa Agustina Nasution³, Mhd Raihan Koansa⁴, Mitgever Pandiangan⁵, Ahmad Bukhori⁶

Universitas Pembangunan Panca Budi, Medan

e-mail: ¹diefaanasyaa05@mail.ac.id, ²kytelkom89@mail.ac.id

³lisaagustinanasution@mail.ac.id, ⁴raihankhoansa@mail.ac.id

⁵pandianganmitgeverpandiangan@mail.ac.id, ⁶ahmadbukhoricbs12@mail.ac.id

Abstract: *The rapid advancement of information technology necessitates computer systems that are effective, efficient, and easy to understand. A key factor in the development of such systems is computational logic, which serves as the foundation for systematically formulating problem-solving steps. This study aims to analyze how computational logic can be constructed using the structured programming paradigm. A literature review method was employed, examining various sources regarding the concepts of computational logic and structured programming. The results indicate that the application of fundamental structures—such as sequence, selection, and iteration—enhances code readability and program efficiency while simplifying debugging and maintenance processes. Furthermore, integrating computational logic with structured programming reduces program complexity and results in more organized software. Thus, the combination of these two elements is crucial for producing high-quality programs.*

Keyword: *Computational Logic; Structured Programming; Algorithms; Programming Paradigm*

Abstrak: Perkembangan teknologi informasi yang pesat menuntut adanya sistem komputer yang efektif, efisien, dan mudah dipahami. Salah satu faktor utama dalam pengembangan sistem tersebut adalah logika komputasi, yang berperan sebagai dasar dalam menyusun langkah-langkah penyelesaian masalah secara sistematis. Penelitian ini bertujuan untuk menganalisis bagaimana logika komputasi dapat dibangun dengan menggunakan paradigma pemrograman terstruktur. Metode yang digunakan adalah studi literatur dengan mengkaji berbagai sumber terkait konsep logika komputasi dan pemrograman terstruktur. Hasil penelitian menunjukkan bahwa penerapan struktur dasar seperti urutan (sequence), percabangan (selection), dan perulangan (iteration) mampu meningkatkan keterbacaan kode, efisiensi program, serta mempermudah proses debugging dan pemeliharaan. Selain itu, integrasi logika komputasi dengan pemrograman terstruktur juga dapat mengurangi kompleksitas program dan menghasilkan perangkat lunak yang lebih terorganisir. Dengan demikian, kombinasi keduanya sangat penting dalam menghasilkan program yang berkualitas tinggi.

Kata kunci: Computational Logic; Structured Programming; Algorithms; Programming Paradigm

PENDAHULUAN

Perkembangan teknologi informasi yang semakin pesat menuntut adanya kemampuan dalam membangun sistem komputer yang efektif, efisien, dan mudah dipahami. Salah satu aspek penting dalam

pengembangan sistem tersebut adalah kemampuan dalam membangun logika komputasi yang baik. Logika komputasi berperan sebagai dasar dalam menyusun langkah-langkah penyelesaian masalah secara sistematis melalui program komputer (Wahyuni & Wadly, 2023).

Dalam praktiknya, banyak permasalahan yang muncul akibat penggunaan logika yang tidak terstruktur, seperti program yang sulit dipahami, sulit diperbaiki, dan rentan terhadap kesalahan. Oleh karena itu, diperlukan suatu pendekatan pemrograman yang mampu menghasilkan kode yang rapi, jelas, dan mudah dikembangkan.

Paradigma pemrograman terstruktur hadir sebagai solusi untuk mengatasi permasalahan tersebut. Pemrograman terstruktur menekankan penggunaan struktur kontrol yang jelas, yaitu urutan (*sequence*), percabangan (*selection*), dan perulangan (*iteration*). Dengan pendekatan ini, logika komputasi dapat dibangun secara sistematis sehingga menghasilkan program yang lebih terorganisir dan efisien (A. S. P. Tarigan et al., 2023).

METODE

Penelitian ini menggunakan pendekatan kualitatif dengan metode studi literatur (*literature review*). Pendekatan ini dipilih untuk memperoleh pemahaman yang komprehensif mengenai implementasi logika komputasi dalam paradigma pemrograman terstruktur melalui kajian sistematis terhadap berbagai sumber ilmiah yang relevan. Metode studi literatur memungkinkan peneliti mengidentifikasi, menganalisis, dan mensintesis berbagai konsep, teori, serta hasil penelitian terdahulu yang berkaitan dengan topik penelitian. Data yang digunakan merupakan data sekunder yang diperoleh dari jurnal ilmiah nasional dan internasional, buku referensi, prosiding, serta artikel ilmiah yang membahas logika komputasi, algoritma, computational thinking, dan paradigma pemrograman terstruktur. Pemilihan sumber pustaka dilakukan berdasarkan tingkat relevansi dengan topik penelitian, kredibilitas penerbit, serta kebaruan publikasi agar informasi yang diperoleh memiliki validitas akademik yang memadai. Tahapan penelitian meliputi

identifikasi permasalahan, penelusuran dan pengumpulan literatur, seleksi sumber berdasarkan kriteria inklusi, analisis isi (*content analysis*), serta sintesis hasil kajian. Pada tahap analisis, setiap referensi dikaji untuk mengidentifikasi konsep utama, karakteristik pemrograman terstruktur, serta hubungan antara logika komputasi dan implementasinya dalam pengembangan perangkat lunak (Wahyuni et al., 2023; Walmarwah et al., 2024). Analisis data dilakukan secara deskriptif kualitatif melalui proses reduksi data, pengelompokan informasi berdasarkan tema, interpretasi terhadap hasil kajian, serta penyusunan kesimpulan. Pendekatan ini bertujuan menghasilkan uraian yang sistematis mengenai penerapan logika komputasi dalam paradigma pemrograman terstruktur sehingga dapat memberikan gambaran konseptual yang mendalam serta menjadi referensi dalam pengembangan perangkat lunak yang efektif, efisien, dan mudah dipelihara (A. D. Tarigan et al., 2022; Wibowo et al., 2017).

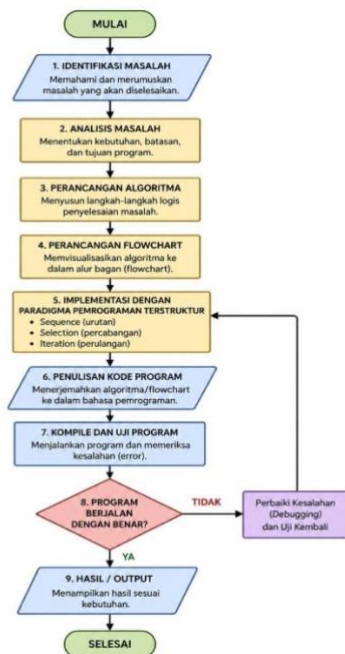
HASIL DAN PEMBAHASAN

Berdasarkan hasil kajian dari berbagai literatur, logika komputasi memiliki peran penting dalam proses pengembangan perangkat lunak. Logika komputasi digunakan sebagai dasar untuk menyusun langkah-langkah penyelesaian masalah secara sistematis melalui algoritma. Dengan adanya logika yang baik, proses pembuatan program menjadi lebih terarah dan tujuan yang ingin dicapai dapat diwujudkan dengan lebih efektif.

Paradigma pemrograman terstruktur mendukung penerapan logika komputasi melalui tiga struktur dasar, yaitu *sequence* (urutan), *selection* (percabangan), dan *iteration* (perulangan). Ketiga struktur tersebut membantu programmer dalam menyusun alur program secara runtut sehingga kode yang dihasilkan lebih mudah dipahami, diuji, dan dikembangkan. Selain itu,

penggunaan struktur yang jelas juga dapat mengurangi kesalahan dalam proses penulisan program.

Hasil kajian juga menunjukkan bahwa penyusunan algoritma sebelum proses coding merupakan tahapan yang sangat penting. Algoritma yang disusun dengan baik akan mempermudah proses implementasi ke dalam bahasa pemrograman. Sebaliknya, apabila logika yang digunakan kurang tepat, program yang dihasilkan akan sulit dipahami dan berpotensi menghasilkan keluaran yang tidak sesuai dengan kebutuhan. Oleh karena itu, penyusunan algoritma menjadi bagian yang tidak dapat dipisahkan dari pemrograman terstruktur.



Gambar 1 Bagan Alur FLOWchart

Beberapa penelitian terdahulu juga mendukung hasil kajian ini. Lye dan Koh (2021) menjelaskan bahwa kemampuan computational thinking dapat membantu seseorang dalam menyelesaikan masalah secara logis dan sistematis melalui proses pemrograman. Sementara itu, Grover dan Pea (2021) menyatakan bahwa logika komputasi merupakan salah satu kompetensi penting dalam pembelajaran informatika karena dapat meningkatkan kemampuan berpikir analitis dan pemecahan masalah. Temuan tersebut

sejalan dengan pembahasan pada penelitian ini, yaitu bahwa penerapan paradigma pemrograman terstruktur mampu mendukung implementasi logika komputasi sehingga menghasilkan program yang lebih terorganisir, mudah dipelihara, dan lebih efisien.

Secara keseluruhan, hasil kajian menunjukkan bahwa logika komputasi dan paradigma pemrograman terstruktur saling melengkapi. Logika komputasi berfungsi sebagai dasar dalam menentukan langkah penyelesaian masalah, sedangkan pemrograman terstruktur menjadi cara untuk mengimplementasikan langkah tersebut ke dalam bentuk program. Kombinasi keduanya akan menghasilkan perangkat lunak yang memiliki alur yang jelas, mudah dipahami, dan lebih mudah dikembangkan sesuai kebutuhan.

Hubungan Logika Komputasi dengan Pemrograman Terstruktur .

Logika komputasi dan pemrograman terstruktur merupakan dua konsep yang tidak dapat dipisahkan dalam proses pengembangan perangkat lunak. Keduanya saling melengkapi, di mana logika komputasi berfungsi sebagai dasar pemikiran dalam menyusun solusi, sedangkan pemrograman terstruktur menjadi metode untuk mengimplementasikan solusi tersebut secara sistematis dan terorganisir.

Hubungan keduanya dapat dijelaskan sebagai berikut:

1. Logika komputasi → merancang solusi (what to do)
2. Pemrograman terstruktur → mengatur Langkah-langkah Membangun Logika Komputasi

Membangun logika komputasi tidak dapat dilakukan secara langsung tanpa tahapan yang sistematis. Diperlukan proses yang terstruktur agar solusi yang dihasilkan benar, efisien, dan dapat diimplementasikan dengan baik ke dalam program.

Berikut adalah tahapan-tahapan:

Memahami Permasalahan (Problem

Understanding)

Tahap ini merupakan langkah awal yang sangat penting karena kesalahan dalam memahami masalah akan berdampak pada keseluruhan solusi.

Pada tahap ini dilakukan:

1. Mengidentifikasi input (data yang dibutuhkan)
2. Menentukan output (hasil yang diharapkan)
3. Memahami proses yang diperlukan untuk mengolah input menjadi output
4. Menentukan batasan (constraint) dari masalah Contoh:

Jika ingin menentukan bilangan genap/ganjil:

1. Input: angka
2. Proses: pembagian dengan 2
3. Output: “Genap” atau “Ganjil”

Menyusun Algoritma (Algorithm Design)

Setelah masalah dipahami, langkah selanjutnya adalah menyusun algoritma sebagai solusi logis. Algoritma berisi urutan langkah-langkah yang sistematis dan harus:

1. Logis (masuk akal)
2. Tidak ambigu (tidak bermakna ganda)
3. Memiliki urutan yang jelas

Algoritma yang baik biasanya disusun dari langkah sederhana ke kompleks dan mengikuti pola input → proses → output. Tujuan tahap ini adalah memastikan solusi sudah benar secara logika sebelum ditulis ke dalam kode program.

Membuat Representasi Algoritma (Flowchart atau Pseudocode)

Agar algoritma lebih mudah dipahami, maka perlu direpresentasikan dalam bentuk visual atau semikode.

Flowchart: menggunakan simbol diagram alir untuk menggambarkan proses

Pseudocode: menggunakan bahasa sederhana yang menyerupai kode program

Manfaat tahap ini:

1. Mempermudah pemahaman alur

logika

2. Mengurangi kesalahan sebelum coding
3. Memudahkan komunikasi antar programmer

Implementasi Program (Coding/Programming)

Pada tahap ini, algoritma yang telah dibuat diterjemahkan ke dalam bahasa pemrograman seperti C, Python, atau Java.

Hal yang perlu diperhatikan:

1. Menggunakan struktur dasar (sequence, selection, iteration)
2. Menjaga kerapian dan keterbacaan kode
3. Menghindari penulisan kode yang tidak efisien .

Tahap ini merupakan proses realisasi dari logika komputasi menjadi program yang dapat dijalankan oleh komputer.

Pengujian dan Evaluasi (Testing dan Debugging)

Setelah program dibuat, perlu dilakukan pengujian untuk memastikan bahwa program berjalan sesuai dengan tujuan.

Kegiatan dalam tahap ini meliputi:

1. Testing: menjalankan program dengan berbagai input
2. Debugging: mencari dan memperbaiki kesalahan (error)
3. Validasi: memastikan output sesuai dengan yang diharapkan Jenis kesalahan yang biasanya ditemukan:
4. Syntax error (kesalahan penulisan kode)
5. Logical error (kesalahan logika)
6. Runtime error (kesalahan saat program dijalankan)

Dokumentasi dan Pengembangan

Tahap ini sering diabaikan, padahal sangat penting terutama untuk proyek besar.

Dokumentasi meliputi:

1. Penjelasan fungsi program
2. Cara penggunaan

3. Struktur kode

Pengembangan dilakukan jika:

1. Ada fitur baru
2. Ada perbaikan sistem
3. Ada peningkatan performa

SIMPULAN

Penelitian ini menunjukkan bahwa implementasi logika komputasi melalui paradigma pemrograman terstruktur mampu menghasilkan penyelesaian masalah yang sistematis, terorganisasi, dan mudah dipahami. Penerapan struktur dasar pemrograman, yaitu sekuensial, seleksi, dan perulangan, memungkinkan proses komputasi berjalan secara efisien serta memudahkan pengembangan algoritma yang memiliki alur logika yang jelas. Selain meningkatkan keterbacaan (*readability*) dan kemudahan pemeliharaan (*maintainability*) kode program, paradigma pemrograman terstruktur juga berkontribusi terhadap pengurangan kesalahan logika (*logical error*) melalui pembagian program ke dalam modul-modul yang terstruktur.

Hasil implementasi menunjukkan bahwa paradigma pemrograman terstruktur tetap relevan sebagai fondasi dalam pembelajaran logika komputasi dan pengembangan perangkat lunak, terutama bagi mahasiswa dan pengembang pemula yang memerlukan pemahaman mendalam mengenai proses penyusunan algoritma. Pendekatan ini memberikan dasar yang kuat sebelum mempelajari paradigma pemrograman yang lebih kompleks, seperti pemrograman berorientasi objek maupun pemrograman fungsional. Dengan demikian, penelitian ini menegaskan bahwa integrasi logika komputasi dan paradigma pemrograman terstruktur merupakan pendekatan yang efektif dalam meningkatkan kemampuan analisis algoritmik, kualitas implementasi program, dan efisiensi proses pengembangan perangkat lunak.

Berdasarkan pembahasan yang telah diuraikan, dapat disimpulkan bahwa

logika komputasi merupakan fondasi utama dalam proses pengembangan perangkat lunak, karena berperan dalam menyusun langkah-langkah penyelesaian masalah secara sistematis melalui algoritma. Tanpa logika yang baik, program yang dihasilkan berpotensi tidak sesuai dengan tujuan dan sulit untuk dikembangkan.

Paradigma pemrograman terstruktur hadir sebagai pendekatan yang efektif untuk mengimplementasikan logika tersebut ke dalam bentuk program yang terorganisir. Dengan memanfaatkan struktur dasar seperti urutan (*sequence*), percabangan (*selection*), dan perulangan (*iteration*), alur program menjadi lebih jelas, terarah, dan mudah dipahami. Dengan demikian, kombinasi antara logika komputasi yang tepat dan penerapan pemrograman terstruktur yang baik akan menghasilkan program yang efisien, minim kesalahan, serta mudah dalam proses pengujian, pemeliharaan, dan pengembangan di masa yang akan datang.

DAFTAR PUSTAKA

- Hermawan, H., & Setiyani, H. (2019). Implementasi Algoritma A-Star Pada Permainan Komputer Roguelike Berbasis Unity. *Jurnal Algoritma, Logika Dan Komputasi*, 2(1).
- Journal, I., & Science, C. (2022). *EXTREME PROGRAMMING VS SCRUM: A COMPARISON OF AGILE MODELS*. 2(2), 80–96.
- Saelan, A. (2009). Logika Fuzzy. *Struktur Diskrit*, 1(13508029), 1–5.
- Tarigan, A. D., Wibowo, P., & Tarigan, A. S. (2022). Perancangan Otoped Listrik Menggunakan Panel Surya Sebagai Media Transportasi. *RELE (Rekayasa Elektrikal Dan Energi): Jurnal Teknik Elektro*, 5(1), 62–65.
- Tarigan, A. S. P., Syahputra, M. R., & Fahreza, M. (2023). KELAYAKAN INSTALASI LISTRIK (Studi Kasus: Rumah Tangga Desa Klambir Lima Kecamatan Hamparan Perak

-
- Kabupaten Deli Serdang). *Penerbit Tahta Media*.
- Wahyuni, S., Akbar, A., Khaliq, A., & Akbar, A. (2023). WEB-BASED APPLICATION FOR SEA PRODUCTS TRADING TO INCREASE FISHERMEN'S INCOME IN SECANGGAN VILLAGE. *Prosiding Universitas Dharmawangsa*, 3(1), 736–745.
- Wahyuni, S., & Wadly, F. (2023). Application Of Inventory And Service Transactions On Web-Based Cv Medan Teknik using the Agile Kanban Method. *International Journal Of Computer Sciences and Mathematics Engineering*, 2(1).
- Walmarwah, M. L. A., Lubis, A., & Wahyuni, S. (2024). Implementasi Cloud Computing Menggunakan Platform AWS Pada Website Rumah Kue Havivya Medan. *Tekmapro*, 19(1), 134–147.
- Wibowo, P., Lubis, S. A., & Hamdani, Z. T. (2017). Smart home security system design sensor based on pir and microcontroller. *International Journal of Global Sustainability*, 1(1), 67–73.
- Zai, F., Tanti, L., Riza, B. S., Studi, P., Informasi, S., Teknik, F., & Utama, U. P. (2021). *Rancang Bangun Sistem Informasi Penjualan Pakan Ternak Berbasis Mobile*. 9(1), 90–99