

Studi Implementasi Logika Digital: Integrasi Pemecahan Masalah Teoretis dan Pengujian Berbasis Simulasi

Munabila*¹, Nandira Kanza Alika Ramdani², Purwanti Rethob Rumlean³, Sara Nabila Putri Sahlin⁴

Informatika, Teknik, Universitas Halu Oleo, Kota Kendari, Indonesia

e-mail: ^{1*}munabila903@gmail.com, ²nandiraramdani@gmail.com,

³rettobpurwanti11@gmail.com, ⁴sarahnabilaputri28@gmail.com

Abstract

Manual digital circuit design carries a high risk of human error, which can hinder design efficiency. This study aims to verify the consistency between digital logic theory and simulation results using Logisim Evolution software. Test results show that the simulation output is identical to the manual theoretical calculations, thus demonstrating the simulator's effectiveness as an accurate verification tool prior to the physical implementation stage.

Keywords: *digital logic, Boolean algebra, Karnaugh map, Logisim Evolution, circuit simulation*

Abstrak

Perancangan sirkuit digital secara manual memiliki risiko tinggi terhadap kesalahan manusia yang dapat menghambat efisiensi desain. Penelitian ini bertujuan untuk memverifikasi konsistensi antara teori logika digital dengan hasil simulasi pada perangkat lunak Logisim Evolution. Hasil pengujian menunjukkan bahwa output simulasi identik dengan kalkulasi teoretis manual, sehingga membuktikan efektivitas simulator sebagai alat verifikasi yang akurat sebelum tahap implementasi fisik.

Kata Kunci: logika digital, aljabar Boolean, peta Karnaugh, Logisim Evolution, simulasi sirkuit

PENDAHULUAN

Perancangan sirkuit digital dimulai dari pemahaman teoretis mengenai aljabar Boolean dan tabel kebenaran yang menjadi dasar logika operasional sebuah sistem. Namun, perancangan di atas kertas seringkali memiliki risiko kesalahan manusia, sehingga memerlukan validasi sebelum diimplementasikan secara fisik. Tujuan utama dari penelitian ini adalah untuk menguji konsistensi antara teori logika digital dengan hasil simulasi pada perangkat lunak Logisim guna memastikan keduanya memberikan hasil yang identik.

Urgensi dari pengujian ini terletak pada efisiensi proses desain, di mana simulasi memungkinkan deteksi kesalahan logika (*bug*)

secara dini tanpa memerlukan biaya komponen fisik. Dengan membandingkan output simulasi terhadap hasil kalkulasi manual, penelitian ini bertujuan untuk menilai akurasi dan reliabilitas desain sirkuit digital. Melalui integrasi ini, diharapkan pemahaman teoretis mengenai gerbang logika dapat terverifikasi secara visual dan interaktif, sehingga menjamin fungsionalitas sistem berjalan sesuai dengan spesifikasi yang direncanakan.

METODE PENELITIAN

1. Fundamental Sistem Digital dan Representasi Biner

Sistem digital didefinisikan sebagai sistem yang menggunakan sinyal-sinyal diskret untuk merepresentasikan informasi dalam bentuk bilangan biner, yaitu angka 0 dan 1. Penggunaan data biner ini memungkinkan sistem untuk memproses informasi secara akurat dan lebih tahan terhadap gangguan (*noise*) dibandingkan dengan sistem analog. Menurut Helmiawan, Herdiana, dan Guntara (2025), keunggulan utama sinyal digital terletak pada presisi data yang tinggi dan kemudahannya untuk disimpan dalam media penyimpanan tanpa kehilangan kualitas.

2. Aljabar Boolean dan Fondasi Operasi Logika

Aljabar Boolean merupakan fondasi matematika utama untuk mengoperasikan nilai-nilai logika yang direpresentasikan sebagai angka 1 (*true*) dan 0 (*false*). Operasi logika dasar seperti NOT, AND, dan OR menjadi instruksi mendasar bagi gerbang logika dalam melakukan pengambilan keputusan di dalam sirkuit digital. Tabel kebenaran digunakan sebagai metode standar untuk mengevaluasi fungsionalitas dari setiap kombinasi input dan output pada gerbang-gerbang tersebut.

3. Metode Penyederhanaan Logika Digital

Proses penyederhanaan ekspresi Boolean sangat krusial dalam perancangan sirkuit digital untuk meningkatkan efisiensi dan menekan biaya

produksi. Metode simplifikasi yang umum digunakan meliputi penggunaan hukum aljabar Boolean, seperti hukum distributif dan hukum De Morgan, serta teknik grafis menggunakan Peta Karnaugh (K-Map). Melalui K-Map, perancang dapat memetakan kombinasi variabel ke dalam kotak-kotak untuk mengidentifikasi istilah logika yang dapat digabungkan, sehingga jumlah gerbang logika yang dibutuhkan dalam sirkuit akhir menjadi minimal.

4. Karakteristik Rangkaian Kombinasional dan Sekuensial

Rangkaian logika secara umum diklasifikasikan menjadi dua jenis utama, yaitu rangkaian kombinasional dan rangkaian sekuensial. Pada rangkaian kombinasional, output yang dihasilkan murni bergantung pada kondisi input saat ini tanpa melibatkan elemen penyimpanan data. Sebaliknya, rangkaian sekuensial memiliki elemen memori seperti flip-flop (misalnya tipe SR, D, atau JK) yang menyimpan status sebelumnya, sehingga output dipengaruhi oleh input saat ini dan kondisi historis sistem.

5. Peran Simulasi dalam Verifikasi Desain Sistem

Simulasi merupakan tahap krusial dalam perancangan sistem digital untuk memastikan fungsionalitas desain sebelum masuk ke tahap implementasi fisik pada perangkat keras. Proses ini memungkinkan perancang untuk mendeteksi kesalahan logika (*bug*) secara dini, sehingga dapat menghemat biaya produksi dan waktu pengembangan. Menurut Ichsan dan Kurniawan (2017), simulasi memberikan visualisasi interaktif mengenai aliran data, yang sangat membantu dalam proses verifikasi fungsional sirkuit yang kompleks.

6. Arsitektur Modular dan Fitur Utama Perangkat Lunak Logisim

Logisim adalah perangkat lunak simulator *open-source* yang dirancang untuk mendukung pembelajaran dan perancangan rangkaian logika digital secara visual. Salah satu fitur unggulannya adalah sub-sirkuit, yang memungkinkan penerapan desain modular, di mana unit-unit logika kecil dapat dibangun dan diuji secara mandiri sebelum digunakan kembali dalam sistem yang lebih besar. Dokumentasi resmi Logisim menyatakan bahwa antarmuka grafis yang sederhana memudahkan pengguna dalam memantau propagasi sinyal secara *real-time*.

7. Analisis Permasalahan

Permasalahan utama dalam perancangan sirkuit digital adalah adanya risiko kesalahan manusia (*human error*) pada tahap perhitungan manual aljabar Boolean dan pembuatan tabel

kebenaran di atas kertas. Selain itu, abstraksi logika dalam perangkat keras modern sering kali sulit dipahami tanpa adanya visualisasi aliran data yang jelas. Implementasi fisik secara langsung menggunakan komponen *Integrated Circuit* (IC) tanpa tahap verifikasi awal dinilai tidak efisien karena dapat menimbulkan biaya tinggi akibat kerusakan komponen jika terjadi kesalahan logika (*bug*) pada sirkuit. Oleh karena itu, sebagaimana telah diuraikan pada bagian Pendahuluan, diperlukan sebuah metode verifikasi yang mampu memberikan umpan balik instan mengenai keakuratan desain sebelum tahap produksi fisik.

8. Arsitektur dan Rancangan Metode

Penelitian ini menerapkan strategi desain modular (*bottom-up design*) dan verifikasi komparatif multi-level. Secara garis besar, rancangan metode dibagi menjadi empat tahapan utama sebagai berikut.

1. **Tahap Analisis Teoretis.** Proses dimulai dengan merumuskan fungsi logika yang diinginkan ke dalam bentuk ekspresi Boolean. Selanjutnya, dilakukan pembuatan tabel kebenaran untuk mencakup semua kemungkinan kombinasi input dan output. Penyederhanaan ekspresi dilakukan menggunakan hukum aljabar Boolean atau teknik Peta Karnaugh (K-Map) untuk meminimalkan jumlah gerbang logika yang dibutuhkan sehingga meningkatkan efisiensi desain.
2. **Tahap Perancangan Sistem (Arsitektur Modular).** Sirkuit dibangun di dalam Logisim dengan pendekatan modular menggunakan fitur sub-sirkuit. Setiap unit logika (seperti Adder, Arithmetic Unit, atau Logic Unit) didesain dan diuji secara independen sebelum digabungkan ke dalam sirkuit utama. Hal ini memudahkan proses debugging dan memastikan keterbacaan aliran data pada sistem yang kompleks seperti CPU 8-bit.
3. **Tahap Simulasi dan Pengujian.** Simulasi dijalankan pada Logisim untuk mengamati propagasi sinyal secara *real-time*. Pengujian dilakukan dengan memasukkan berbagai kombinasi angka biner melalui pin input dan memantau status pin output. Untuk memastikan akurasi pada ruang input yang luas, penelitian ini juga memanfaatkan fitur Test Vector, di mana berkas teks berisi input dan output yang diharapkan dimuat untuk melakukan pengujian otomatis terhadap fungsionalitas sirkuit.
4. **Tahap Verifikasi Komparatif dan Analisis.** Hasil akhir simulasi dibandingkan secara langsung dengan hasil kalkulasi teoretis

manual. Indikator keberhasilan ditetapkan apabila sirkuit memberikan output yang identik dengan tabel kebenaran di setiap skenario uji. Jika terdapat perbedaan (data error), dilakukan analisis pada jalur pengabelan (wiring) atau logika gerbang di dalam sub-sirkuit untuk dilakukan perbaikan iteratif hingga desain tervalidasi sempurna.

9. Studi Kasus

Untuk menjalankan tahap verifikasi komparatif yang dijelaskan pada bagian sebelumnya, penelitian ini mengambil dua soal representatif dari buku referensi rangkaian digital sebagai bahan uji. Kedua soal dipilih karena mewakili dua kompetensi dasar yang berbeda, yaitu membaca rangkaian gerbang logika untuk mendapatkan ekspresi Boolean, dan menyederhanakan fungsi Boolean menggunakan Peta Karnaugh.

9.1 Studi Kasus 1: Penurunan Ekspresi Boolean dari Diagram Logika

Studi kasus pertama diambil dari soal latihan (Practice Exercise 2.4) yang meminta penentuan ekspresi Boolean dari sebuah diagram logika dengan dua input, x dan y. Pada rangkaian tersebut, x dan y masing-masing dibalik oleh gerbang NOT sehingga menghasilkan x' dan y' . Sinyal-sinyal ini kemudian dipasangkan ke dalam dua gerbang AND, satu menghasilkan xy' dan satu lagi $x'y$, sebelum keduanya digabungkan melalui gerbang OR menjadi keluaran F, seperti ditunjukkan pada Gambar 1.

Practice Exercise 2.4

What Boolean expression is implemented by the following logic diagram?

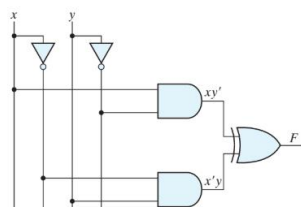


FIGURE PE2.4

Answer:

$$F = (x'y + xy')' = (x'y)'(xy')' = (x + y')(x' + y) = xx' + xy + y'x' + yy' = xy + x'y'$$

Gambar 1 Rangkaian gerbang logika dan penyelesaian manual studi kasus 1

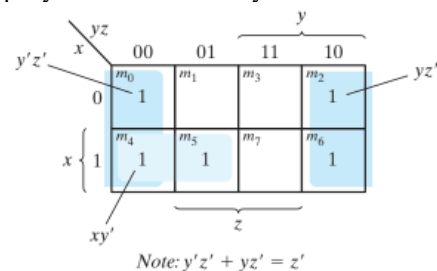
Penyelesaian manual pada buku sumber menunjukkan proses penyederhanaan sebagai berikut: $F = (x'y + xy')' = (x'y)'(xy')' = (x + y')(x' + y) = xx' + xy + y'x' + yy' = xy + x'y'$. Hasil akhir ini merupakan fungsi XNOR dua-input, sebuah bentuk yang cukup sering dijumpai pada rangkaian pembandingan (*comparator*). Soal ini dipakai untuk menguji kemampuan menelusuri jalur sinyal pada diagram gerbang, sebelum rangkaian yang sama

Studi Implementasi Logika Digital....

dibangun ulang di Logisim Evolution untuk keperluan verifikasi.

9.2 Studi Kasus 2: Penyederhanaan Fungsi dengan Peta Karnaugh

Studi kasus kedua diambil dari Contoh 3.3 pada buku yang sama, berupa fungsi Boolean tiga variabel $F(x, y, z) = \Sigma(0, 2, 4, 5, 6)$. Penyederhanaan dilakukan dengan memetakan keenam minterm tersebut ke dalam Peta Karnaugh 3 variabel, lalu mengelompokkan sel-sel yang berdekatan, seperti ditunjukkan pada Gambar 2. Empat sel pada kolom pertama dan terakhir dikelompokkan menjadi satu suku literal tunggal z' , sedangkan sel yang mewakili minterm 5 digabungkan dengan sel yang sudah terpakai sebelumnya untuk membentuk suku dua-literal xy' . Hasil penyederhanaan akhirnya adalah $F = z' + xy'$.



Gambar 2 Peta Karnaugh 3 variabel studi kasus 2

Soal ini dipilih karena melibatkan proses pengelompokan minterm yang tumpang tindih, sesuatu yang menuntut ketelitian saat dikerjakan secara manual, sehingga cocok dijadikan tolok ukur untuk menilai apakah fitur minimisasi otomatis pada Logisim Evolution memberikan hasil yang sama.

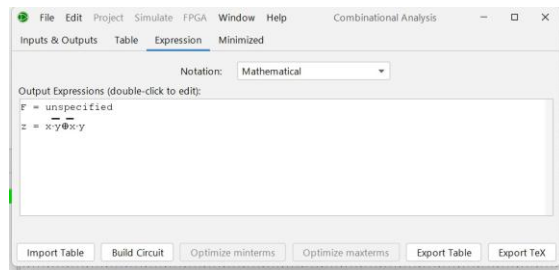
Kedua ekspresi hasil analisis manual di atas selanjutnya dimasukkan ke dalam fitur Combinational Analysis pada Logisim Evolution. Proses dan hasil yang diperoleh dipaparkan pada bagian Hasil dan Pembahasan.

HASIL DAN PEMBAHASAN

Bagian ini memaparkan hasil verifikasi terhadap dua studi kasus yang telah dijelaskan pada bagian Metode Penelitian, menggunakan fitur Combinational Analysis yang tersedia pada Logisim Evolution. Fitur ini memungkinkan variabel input dan output didaftarkan terlebih dahulu, tabel kebenaran dibangun, kemudian ekspresi hasil minimalisasi diperoleh secara otomatis dalam bentuk Sum of Products.

Hasil Studi Kasus 1

Fitur Combinational Analysis menampilkan ekspresi keluaran untuk studi kasus pertama sebagaimana ditunjukkan pada Gambar 3.



Gambar 3 Hasil Combinational Analysis studi kasus 1 pada Logisim Evolution

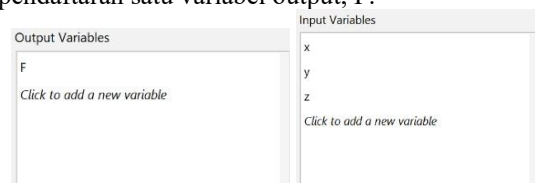
Selain F yang masih berstatus *unspecified*, muncul satu ekspresi tambahan dengan label z, yaitu $z = x \cdot y \oplus \bar{x} \cdot \bar{y}$.

Sekilas bentuk ini terlihat berbeda dari hasil manual pada bagian Metode Penelitian, yang menuliskan F sebagai $xy + x'y'$. Namun, apabila ditelusuri lebih jauh, keduanya sebenarnya sama persis. xy dan $x'y'$ tidak pernah bernilai 1 secara bersamaan; apabila x dan y sama-sama bernilai 1, maka $x'y'$ otomatis bernilai 0, begitu pula sebaliknya. Karena kedua suku ini saling eksklusif, operasi OR dan XOR di antara keduanya akan selalu menghasilkan nilai yang sama. Dengan kata lain, $z = xy \oplus x'y'$ dan $F = xy + x'y'$ hanyalah dua cara penulisan berbeda untuk fungsi yang identik, yaitu XNOR dua-input.

Temuan ini memperkuat hasil studi kasus 1: ekspresi yang diturunkan secara manual dari diagram gerbang logika terbukti benar, dan Logisim Evolution, meski menyajikannya dalam notasi XOR alih-alih OR, tetap sampai pada kesimpulan logika yang sama persis.

Hasil Studi Kasus 2

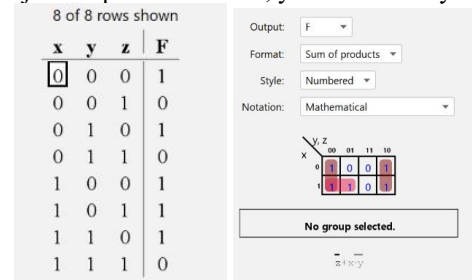
Studi kasus kedua diverifikasi secara lebih menyeluruh, mengikuti alur kerja penuh pada fitur Combinational Analysis, sebagaimana ditunjukkan pada Gambar 4. Variabel input x , y , z didaftarkan lebih dulu pada tab Inputs & Outputs, diikuti pendaftaran satu variabel output, F.



Gambar 4 Pendaftaran variabel input dan output studi kasus 2

Selanjutnya, kedelapan baris tabel kebenaran diisi sesuai fungsi $F(x, y, z) = \Sigma(0, 2, 4, 5, 6)$, sehingga baris dengan kombinasi 000, 010, 100, 101, dan 110 diberi nilai keluaran 1, sedangkan baris 001, 011, dan 111 diberi nilai 0. Tabel kebenaran yang dihasilkan sesuai persis dengan minterm yang disebutkan pada soal, tanpa ada baris yang meleset. Pada tahap berikutnya, Logisim Evolution menjalankan minimisasi otomatis dan

menampilkan Peta Karnaugh 3 variabel berikut ekspresi hasil penyederhanaannya, sebagaimana ditunjukkan pada Gambar 5, yaitu $F = z' + xy'$.



Gambar 5 Peta Karnaugh dan hasil minimisasi otomatis studi kasus 2 pada Logisim Evolution

Ekspresi ini sama persis dengan hasil perhitungan manual pada Contoh 3.3, baik dari jumlah suku maupun literal yang digunakan. Kesesuaian antara tabel kebenaran, hasil minimisasi otomatis, dan perhitungan manual pada studi kasus kedua ini menunjukkan bahwa metode Peta Karnaugh yang dipelajari secara teoretis dapat direproduksi dengan tepat oleh perangkat lunak simulasi. Temuan ini memperkuat argumen yang diajukan pada bagian Pendahuluan, bahwa Logisim dapat dipakai sebagai sarana verifikasi yang mengurangi risiko kesalahan manusia pada tahap perhitungan manual, sekaligus memberikan umpan balik visual ketika ekspresi hasil minimisasi ditampilkan berdampingan dengan peta Karnaugh interaktifnya.

Secara keseluruhan, dua studi kasus di atas mewakili dua cakupan verifikasi yang berbeda. Studi kasus pertama menegaskan bahwa ekspresi Boolean yang ditelusuri secara manual dari sebuah diagram gerbang sudah sesuai sebelum masuk ke tahap simulasi, sedangkan studi kasus kedua membuktikan bahwa proses penyederhanaan dengan Peta Karnaugh, termasuk pengelompokan minterm yang tumpang tindih, dapat direplikasi secara otomatis oleh Logisim Evolution dengan hasil yang identik. Kedua temuan ini bersama-sama mendukung tujuan penelitian yang disampaikan pada bagian Pendahuluan, yaitu menguji konsistensi antara teori logika digital dengan hasil simulasi.

KESIMPULAN DAN SARAN

Penelitian ini menyimpulkan bahwa terdapat konsistensi penuh antara teori aljabar Boolean dan Peta Karnaugh dengan hasil reproduksi digital pada Logisim Evolution. Untuk pengembangan selanjutnya, disarankan menerapkan strategi desain modular dan memanfaatkan fitur pengujian otomatis (*test vector*) guna menjamin reliabilitas pada sistem digital yang lebih kompleks.

UCAPAN TERIMA KASIH

Terima kasih kepada para penyedia referensi fundamental dan pengembang perangkat lunak simulasi yang telah mendukung kelancaran proses verifikasi sirkuit dalam penelitian ini.

DAFTAR PUSTAKA

- Beltran, M. J. (2026). 8-bit CPU design: A Beltran, M. J. (2026). *8-bit CPU designed and implemented in Logisim, featuring a custom instruction set, ALU, registers, control unit, and memory interface*. GitHub. <https://github.com/migueljbeltran/8bit-cpu>
- Burch, C. (2012). *The guide to being a Logisim user*. <http://www.cburch.com/logisim/docs/2.5.0/en/guide/index.html>
- DigiSim.io. (2026, May 7). DigiSim vs Logisim Evolution: An honest comparison. *DigiSim Blog*. <https://digisim.io/blog/digisim-vs-logisim-evolution/>
- Fauziyyah, H. (2025). *Pengenalan aplikasi Logisim Evolution dan pembuatan gerbang logika dasar serta rangkaian kombinasional* (Laporan Praktikum). Universitas Bhayangkara Jakarta Raya.
- Floyd, T. L. (2013). *Digital fundamentals*. Pearson.
- Harris, D., & Harris, S. (2012). *Digital design and computer architecture*. Morgan Kaufmann.
- Helmiawan, M. A., Herdiana, D., & Guntara, A. (2025). *Pengantar sistem digital*. PT Penerbit Naga Pustaka. <https://nagapustaka.store/>
- Hidayat, R. (2014). *Implementasi perangkat lunak Logisim 2.7.1 untuk meningkatkan kompetensi siswa tentang rangkaian gerbang logika di SMK Negeri Situraja* (Skripsi). Universitas Pendidikan Indonesia. <http://repository.upi.edu/>
- Ichsan, M. H., & Kurniawan, W. (2017). Design and implementation 8 bit CPU architecture on Logisim for undergraduate learning support. *International Conference on Sustainable Information Engineering and Technology (SIET)*.
- Iswantoro, E. D., Ichsan, M. H., & Kurniawan, W. (2019). Desain arithmetic logic unit 8bit untuk central processing unit 8bit. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 3(1), 892-898. <http://j-ptiik.ub.ac.id>
- Javaid, W. (2025). Design and simulation of a vending machine in Logisim-Evolution. *WiredWhite*. <https://wiredwhite.com/vending-machine-design-simulation-logisim/>
- Mano, M. M., & Ciletti, M. D. (2017). *Digital design*. Pearson.
- Marhguy, T. (2026). *8-bit hybrid ALU: Discrete transistors + 74HC logic*. GitHub. <https://github.com/tmarhguy/8bit-discrete-transistor-alu>
- Self, G. (2019). *Logisim-Evolution lab manual* (Edition 4.0).
- Sianipar, S. R. R., Ichsan, M. H., & Setiawan, E. (2020). Desain datapath arsitektur komputer MIC-1 8 bit menggunakan IC74XX. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 4(6), 1591-1600. <http://j-ptiik.ub.ac.id>
- Wakerly, J. F. (2006). *Digital design: Principles and practices*. Prentice Hall.

Munabila, Nandira Kanza Alika Ramdani, Purwanti Rethob Rumlean, Sara Nabila Putri Sahlin

Submitted: **01/01/2026**; Revised: **20/01/2026**; Accepted: **02/03/2026**; Published: **30/04/2026**