**APLIKASI GERBANG LOGIKA BERBASIS ANDROID SEBAGAI SIMULASI IMPLEMENTASI MATA KULIAH LOGIKA INFORMATIKA FAKULTAS TEKNIK UNIVERSITAS MUHAMMADIYAH JAKARTA (FT-UMJ)**

**Yekto Priyandhani1, Popy Meilina2**

1Mahasiswa Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Jakarta

2Dosen Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Jakarta

2016470114@ftumj.ac.id, popy.meilina@ftumj.ac.id

**Abstrak**

Logika Informatika adalah salah satu mata kuliah Teknik Informatika di Fakultas Teknik Universitas Muhammadiyah Jakarta (FT-UMJ). Selama proses pembelajaran mahasiswa diharapkan mampu memahami konsep dasar tentang logika sampai pengoperasian logika terutama logika pada komputer. Logika pada computer memiliki nilai kebenaran dimana jika benar maka nilainya 1 (TRUE) dan jika salah maka nilainya 0 (FALSE). Untuk memahami soal serta menjawab soal dengan input lebih dari dua dan operator lebih dari satu mahasiswa harus membuat table kebenaran yang banyak dan dibutuhkan penalaran yang lebih. Oleh karena itu peneliti mencoba mengimplementasikan hasil pembelajaran dengan membuat sebuah Aplikasi Simulasi Gerbang Logika berbasis android dalam bentuk game. Dengan adanya aplikasi ini pengguna dapat memahami serta menjawab soal tersebut secara langsung dan menarik minat karena disajikan dalam bentuk game. Aplikasi menyediakan informasi penggunaan operator (And, Or, Nand, Nor, Xor, dan Xnor). Aplikasi juga menyediakan pemilihan soal berdasarkan jumlah input atau jenis operator yang digunakan pada soal. Untuk menjawab soal, pengguna hanya tinggal memilih satu dari tiga pilihan jawaban yang disediakan. Jika jawaban benar maka lampu hidup dan jika salah lampu akan mati. Untuk setiap jawaban benar akan diberi nilai sepuluh dan nol untuk jawaban salah. Setelah permainan berakhir aplikasi akan menampilkan perolehan nilai.

**Kata Kunci** : Logika Informatika, Gerbang Logika, *Game*

**Abstract**

*Informatics Logic is one of the courses in Informatics Engineering at the Faculty of Engineering, University of Muhammadiyah Jakarta (FT-UMJ). During the learning process students are expected to be able to understand the basic concepts of logic to the operation of logic, especially logic on computers. Logic on the computer has a truth value where if it is true then the value is 1 (TRUE) and if it is false then the value is 0 (FALSE). To understand the questions and answer questions with more than two inputs and more than one operator, students have to make a lot of truth tables and more reasoning is needed. Therefore, the researchers tried to implement the learning outcomes by making an Android-based Logic Gate Simulation Application in the form of a game. With this application, users can understand and answer the question directly and attract interest because it is presented in the form of a game. Application provides operator usage information (And, Or, Nand, Nor, Xor, and Xnor). The application also provides question selection based on the number of inputs or the type of operator used in the problem. To answer the question, the user only needs to choose one of the three answer options provided. If the answer is correct then the light is on and if it is wrong the light will be off. For each correct answer will be given a value of ten and zero for wrong answers. After the game ends the application will display the score.*

***Keyword****: Informatic Logic, Logic Gate, game.*

**1. Pendahuluan**

Logika Informatika adalah salah satu mata kuliah Teknik Informatika di Fakultas Teknik Universitas Muhammadiyah Jakarta (FT-UMJ). Mata kuliah tersebut berisi tentang kegiatan yang berhubungan dengan penggunaan akal untuk menghasilkan suatu penalaran dengan kebenaran yang dapat dibuktikan secara matematis. Walaupun tanpa perhitungan dengan matemastis atau dengan statistik, tetapi dapat diuji dan masuk akal akan kebenarannya (Rezaldy, Hayadi, & Ropianto, 2021).

 Dalam proses pembelajarannya, dosen pengampu yang berkompeten memberi pemahaman atau pengajaran ke mahasiswa dengan memberikan materi dan menjelaskannya. Selama proses pembelajaran mahasiswa diharapkan mampu memahami konsep logika informatika dimulai dari pengetahuan dasar tentang logika sampai pengoperasian logika terutama logika pada komputer.

 Logika pada *computer* memiliki nilai kebenaran dimana jika benar maka nilainya 1 (TRUE) dan jika salah maka nilainya 0 (*FALSE*) serta memiliki operator logika yaitu AND, OR, NAND, NOR, XOR, dan XNOR. Semua aturan pengoperasian mengikuti aturan yang ada berdasarkan operator yang digunakan. Dalam kasus dua input dan satu operator kita dapat melihatnya pada referensi atau materi dasar kuliah Logika Informatika yang digunakan, akan tetapi jika input lebih dari dua, mahasiswa harus membuat table kebenaran yang banyak. Begitu juga dengan operator lebih dari dua membuat mahasiswa harus menggunakan penalaran lebih. Berdasarkan analisis berdasarkan data yang telah dilakukan, peneliti mencoba mengimplementasikan hasil belajar mata kuliah Logika Informatika dengan membuat sebuah Aplikasi Simulasi Gerbang Logika berbasis android dalam bentuk *game*.

**2. Tinjauan Pustaka**

**2.1 Logika dan Komputer**

Rekayasa kerangka kerja PC terbuat dari sirkuit rasional 1 (valid) dan 0 (palsu) yang digabungkan dengan berbagai pintu rasional AND, NOR, NOT, XOR, XNOR dan NAND. BUKAN pintu, DAN pintu masuk dan juga pintu masuk adalah tiga pintu utama dari pintu masuk dasar yang penting. Pintu lain yang merupakan gabungan dari tiga jalan masuk utama, khususnya Jalan Masuk NAND, Jalan Masuk NOR, Jalan Masuk XOR dan Jalan Masuk XNOR (Sugiartowo & Ambo, 2018).

1. Gerbang NOT

Gerbang NOT sering juga disebut sebagai rangkaian inventer (pembalik). Tugas rangkaian NOT (pembalik) ialah memberikan suatu keluaran yang tidak sama dengan masukan. Simbol logika untuk pembalik (inverter, rangkaian NOT) diperlihatkan pada gambar berikut.



Gambar 1 Gerbang NOT

1. Gerbang AND

Setidaknya ada dua saluran info dan saluran hasil di pintu masuk AND yang merupakan salah satu pintu rasional penting. Hasil ganda bergantung pada status informasi dan kapasitasnya pada pintu masuk AND. Kondisi hasil akan menjadi alasan 1 jika semua jalur informasi adalah alasan 1 yang merupakan standar fungsi dari pintu masuk AND. Bagaimanapun hasilnya akan menjadi alasan 0.



Gambar 2 Gerbang AND

1. Gerbang OR

Pintu OR adalah salah satu jalan masuk dasar pemikiran yang memiliki setidaknya 2 saluran informasi dan saluran hasil. Meskipun jumlah saluran informasi yang dimiliki oleh pintu OR, sebenarnya memiliki pedoman kerja yang sama di mana kondisi hasil akan menjadi alasan 1 jika satu atau seluruh saluran informasi adalah alasan 1. Terlebih lagi, hasilnya adalah alasan 0.



Gambar 3 Gerbang OR

1. Gerbang NAND

Pintu NAND adalah augmentasi dari pintu masuk AND. Pintu ini benar-benar merupakan pintu masuk DAN di mana pintu NOT diperkenalkan.



Gambar 4 Gerbang NAND

1. Gerbang NOR

Pintu NOR adalah augmentasi dari pintu masuk OR. Peningkatan ini melalui pengenalan pintu NOT pada hasil dari pintu masuk OR. Gambar 4 menunjukkan perpaduan ini di samping gambar untuk pintu NOR. Karena pada dasarnya pintu OR yang hasilnya dibalik, tabel realitas adalah sesuatu yang bertentangan dengan tabel realitas dari pintu masuk OR.



Gambar 5 Gerbang NOR

1. Gerbang XOR

Pintu XOR mewakili Selective OR dimana jika informasi tersebut memiliki alasan yang sama, hasilnya akan menjadi alasan 0 dan sebaliknya dengan asumsi informasi tersebut adalah alasan yang berbeda, hasilnya akan menjadi alasan 1.



Gambar 6 Gerbang XOR

1. Gerbang XNOR

Pintu XNOR mewakili Selective NOT OR adalah sesuatu yang bertentangan dengan pintu masuk XOR dimana jika informasi adalah alasan yang sama, hasilnya akan menjadi alasan 1 dan sebaliknya dengan asumsi informasi adalah alasan yang beragam, hasilnya akan menjadi alasan 0.



Gambar 7 Gerbang XNOR

**2.2 Waterfall Model**

Menurut (Pressman, 2015) model cascade adalah kemajuan pemrograman menggunakan model gaya lama yang metodis dan berurutan. Nama model ini benar-benar "Model Berturut-turut Lurus". Model ini sering disinggung sebagai "siklus hidup teladan" atau teknik kaskade. Model ini mendapat tempat dengan model pemrograman komputer konvensional dan pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970-an dengan 7 (tujuh) tahapan dan mengalami perbaikan serta perubahan hingga menjadi 5 (lima) tahapan sehingga model ini yang paling banyak dipakai dalam Software Engineering (SE). Metode ini mengambil pendekatan yang sistematis dan berurutan. Disebut Waterfall karena prosesnya bertahap.



Gambar 8 Waterfall Model

1. *Communication* (*Project Initiation & Requirements Gathering*)

Sebelum memulai pekerjaan khusus, sangat penting untuk berbicara dengan klien untuk memahami dan mencapai tujuan yang ingin dicapai. Hasil dari korespondensi ini adalah pernyataan usaha, misalnya, menyelidiki masalah yang dialami dan mengumpulkan informasi mendasar, serta membantu menentukan fitur dan elemen produk. Berbagai informasi tambahan juga dapat diambil dari buku harian, artikel, dan *internet*.

1. *Planning* (*Estimating, Scheduling, Tracking*)

Tahap selanjutnya adalah tahap penyusunan yang menjelaskan penilaian terhadap usaha khusus yang akan dilakukan, bahaya yang dapat terjadi, aset yang diharapkan untuk membuat kerangka, item pekerjaan yang akan diserahkan, perencanaan pekerjaan yang akan diselesaikan, dan tindak lanjut. dari ukuran kerja kerangka kerja.

1. *Modelling* (*Analysis & Design*)

Tahap ini adalah tahap rencana dan demonstrasi dari rekayasa kerangka kerja yang berpusat di sekitar rencana struktur informasi, model pemrograman, tampilan antarmuka, dan perhitungan program. Tujuannya adalah untuk lebih mudah memahami pandangan berkali-kali dari apa yang akan selesai.

1. *Construction* (*Code & Test*)

Tahap pengembangan ini adalah cara untuk membuat interpretasi dari struktur rencana menjadi kode atau struktur/bahasa yang dapat diuraikan oleh mesin. Setelah pengkodean selesai, dilakukan percobaan pada framework dan selanjutnya kode yang telah dibuat. Tujuannya adalah untuk menemukan kesalahan yang mungkin terjadi untuk diperbaiki kemudian.

1. *Deployment* (*Delivery, Support, feedback*)

Tahap *Arrangement/Deployment* adalah tahap pelaksanaan pemrograman kepada klien, pemeliharaan pemrograman intermiten, perbaikan pemrograman, penilaian pemrograman, dan pengembangan pemrograman tergantung pada kritik yang diberikan dengan tujuan agar framework dapat terus berjalan dan berkreasi sesuai dengan kapasitasnya.

**3. Metode Penelitian**

Langkah-langkah yang dilakukan pada penelitian ini adalah sebagai berikut:

a. Pengumpulan Data

 Data yang dikumpulkan berupa:

1. Metode Observasi
2. Metode *Study* Kepustakaan
3. Pengumpulan data yang dilakukan dengan wawancara dengan pihak yang terkait. Di penulisan ini Ibu Retnani Latifa sebagai dosen pengampu mata kuliah Logika Informatika Fakultas Teknik Universitas Muhammadiyah Jakarta dan mahasiswa Teknik Informatika Fakultas Teknik Universitas Muhammadiyah Jakarta.

b. Analisis Data

 Pada tahap ini penulis menganalisi terhadap data-data yang sudah diperoleh pada tahap sebelumnya.

c. Perancangan

 Pada tahap ini data-data yang telah dikumpulkan dan dianalisis lalu dilakukan perancangan system model yang akan digunakan dalam pembuatan aplikasi. *Tools* yang digunakan *Unified Modeling* *Language* (UML) seperti usecase diagram, *activity* *diagram*, *Hierarchy* *plus* *Input*-*Output* (HIPO), dan *user* *interface* (UI) *Design Tools* menggunakan *Balsamiq* *Mockup*

d. Pembuatan Aplikasi

 Pada tahap ini, dilakukan pembuatan aplikasi setelah tahap perancangan system selesai. Pembuatan aplikasi menggunakan Android Studio dengan spesifikasi aplikasi mengikuti kebutuhan system yang telah diperoleh dari analisis sebelumnya.

e. Pengujian Aplikasi

 Proses pengujian dalam penulisan ini yaitu dengan menggunakan pengujian *black box testing* dengan mengevaluasi dari sisi fungsional berdasarkan input dan *output.*

Kebutuhan fungsional yang diperlukan oleh Aplikasi *game* Gerbang Logika yang dianalisis dari system berjalan, pada aplikasi terdapat satu actor. Berikut adalah *Use Case diagram* system usulan aplikasi yang dapat menjelaskan interaksi apa saja yang dapat dilakukan oleh actor dengan system yang dibuat.



Gambar 8 *Use case* Sistem Usulan Aplikasi *game* Gerbang Logika

Pada Gambar 8 dijelaskan bahwa *actor* Mahasiswa dapat melakukan *Tutorial* dan Bermain. Selanjutnya adalah rancangan *activity diagram* berdasarkan *Use case* Sistem usulan.



Gambar 9 *Activity Diagram Tutorial*

******

Gambar 10 *Activity Diagram* Mulai Bermain

Pada tahap selanjutnya adalah pembuatan *Sequence Diagram* system usulan.

******

Gambar 11 *Sequence Diagram Tutorial*

******

Gambar 12 *Sequence Diagram* Mulai Bermain

Setelah itu adalah perancangan HIPO Diagram untuk aplikasi



Gambar 13 *HIPO Diagram* Aplikasi Game Gerbang Logika

**4. Hasil dan Pembahasan**

Implementasi menggunakan bahasa pemrograman *javascript* dengan *Android* *Studio.* Berikut adalah hasil dari implementasi. Pada halaman utama Gambar 14, pengguna akan disajikan menu Latihan/*Tutorial* dan Mulai.

.

Gambar 14 Tampilan Halaman Utama

Pada menu Latihan, aplikasi akan menampilkan penjelasan mengenai penggunaan operator logika And, Or, Nand, Nor, Xor, Xnor lengkap dengan deskripsi, symbol pada rangkaian, dan *true table-*nyaa masing-masing Gambar 15.



Gambar 15 Tampilan Halaman Latihan/*Tutorial*

Setelah Latihan, pada tombol Mulai jika kita pilih maka akan menampilkan dua penggolongan kategori yaitu kategori **kesulitan** berdasarkan tingkat kesulitannya atau berdasarkan jumlah penggunaan input yang digunakan pada rangkaian soal. Mudah (dua input dengan satu operator logika), Sedang (tiga input dengan satu sampai dua operator logika), dan Sulit (empat input dengan satu sampai tiga operator). Jumlah soal pada kategori ini adalah sepuluh soal. Selanjutnya pengkategorian soal **kategorikal** berdasarkan operator yang digunakan pada rangkaian soal yaitu ada (And, Or, Nand, Nor, Xor, Xnor) dengan distribusi dua soal Mudah, empat soal Sedang, dan empat soal Sulit.



Gambar 16 Tampilan Halaman Kategori

Setiap kategori soal terdiri dari lima model soal yang terdiri dari sepuluh soal yang diambil dari *bank* soal. Total soal pada *bank* soal adalah 90 soal dengan 18 soal Mudah (And, Or, Nand, Nor, Xor, Xnor masing-masing 3 soal), 36 soal Sedang (And, Or, Nand, Nor, Xor, Xnor masing-masing 6 soal), 36 soal Sulit (And, Or, Nand, Nor, Xor, Xnor masing-masing 6 soal). Setiap kategori yang dipilih akan memanggil satu model soal-nya secara acak menggunakan algoritma random pada program yaitu

|  |
| --- |
| random\_int = (int)Math.floor(Math.random()\*(max-min+1)+min); |

Gambar 17 Kode Random

Setelah diacak maka aplikasi akan menampilkan modul sesuai dengan hasil pengacakan tadi. Berikut adalah tampilan soal.



Gambar 18 Tampilan Soal

Setelah jawaban di *input* maka system akan menampilkan *true table* serta hasil koreksi jawaban. Lampu akan hidup jika menerima sinyal 1 (jawaban benar) dan lampu akan tetap mati jika menerima sinyal 0 (jawaban salah). Jika pengguna sudah mengerjakan sepuluh soal maka system akan menampilkan *score* (Gambar 19). Jumlah *score* yang diperoleh berdasarkan jumlah jawaban yang benar yaitu penambahan sepuluh poin untuk setiap jawaban yang benar. Berikut adalah tampilannya.



Gambar 19 Tampilan Halaman *Score*

Pada Gambar 19 terdapat *button Continue* dan *exit.* *Continue* berfungsi untuk mengembalikan pengguna ke halaman kategori jika pengguna ingin memainkannya lagi dan *Exit* berfungsi untuk menyudahi permainan.

Setelah itu dilakukan pengujian terhadap akurasi jawaban. Berikut adalah table hasil uji yang dilakukan dalam dua kali permainan.

Tabel 1 Hasil Uji Akurasi Jawaban

|  |  |  |  |
| --- | --- | --- | --- |
| Jumlah Soal (JS) | Jumlah Soal Yang Sesuai (JSYS) | Jumlah Soal Yang Tidak Sesuai (JSYTS) | Tingkat Akurasi(JSYS/JS)x100% |
| 20 | 20 | 0 | 100% |

**5. Kesimpulan**

Berdasarkan hasil analisis dan pembahasan maka didapatkan kesimpulan sebagai berikut:

1. Operator yang digunakan sebanyak enam yaitu And, Or, Nand, Nor, Xor, dan Xnor.
2. Input yang digunakan yaitu dari dua input sampai empat input pada rangkaian.
3. Kategori soal digolongkan menjadi dua yaitu Kesulitan (mudah, sedang, dan sulit) dan Kategorikal (And, Or, Nand, Nor, Xor, dan Xnor).
4. Pada bank soal terdapat total sembilan puluh soal yang terdiri dari delapan belas soal mudah (And, Or, Nand, Nor, Xor, dan Xnor masing-masing tiga soal), tiga puluh enam soal sedang (And, Or, Nand, Nor, Xor, dan Xnor masing-masing enam soal), dan tiga puluh enam soal sulit (And, Or, Nand, Nor, Xor, dan Xnor masing-masing enam soal).
5. True Table akan muncul setelah jawaban di submit, sehingga proses pembelajaran akan tersampaikan.
6. Setiap kategori soal terdiri dari lima modul yang masing-masing modul terdiri dari sepuluh soal.
7. Pemanggilan acak terletak pada pemanggilan modul menggunakan **random\_int = (int)Math.floor(Math.random()\*(max-min+1)+min).**
8. Proses koreksi jawaban dilakukan secara otomatis oleh system dengan mencocokan jawaban yang dipilih dengan kunci jawaban.

**6. Daftar Pustaka**

Pressman. (2015). *Rekayasa Perangkat Lunak: Pendekatan Praktisi Buku I.* Yogyakarta: Andi.

Rezaldy, A., Hayadi, B. H., & Ropianto, M. (2021, Maret 21). *SEJARAH SINGKAT DAN MACAM-MACAM LOGIKA INFORMATIKA*. Retrieved from Academia: academia.edu

Sari, E. L., HD, N. S., & Arnandi, W. (2018). RANGKAIAN SISTEM KELISTRIKAN LAMPU PENERANGAN, LAMPU REM, DAN KLAKSON PADA KENDARAAN MITSUBISHI L300. *RIDTEM (Riset Diploma Teknik Mesin). 1 (1)*, 1-5.

Sugiartowo, & Ambo, S. N. (2018). SIMULASI RANGKAIAN KOMBINASIONAL SEBAGAI MEDIA PEMBELAJARAN SISTEM DIGITAL PADA FAKULTAS TEKNIK UNIVERSITAS MUHAMMADIYAH JAKARTA.

Sukamto, R. A., & Shalahudin, M. (2015). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek.* Bandung: Informatika.