

PERANCANGAN RANGKAIAN VIRTUAL JAM CATUR DIGITAL

MAKALAH

Disusun untuk memenuhi UAS mata kuliah
Sistem Digital

Dosen Pengampu:

Bapak Endra Pratama, S.Si., M.Cs.



Disusun oleh:

Kelompok 7

1. Daffa Gathfan Chaidar (L0125077)
2. Gamma Assyafi Fadhillah Ar Rasyad (L0125013)
3. Muhammad Haidar Amru (L0125025)
4. Rosyid Al Ansorri (L0125065)

KELAS A

PROGRAM STUDI INFORMATIKA

**FAKULTAS TEKNOLOGI INFORMASI DAN SAINS DATA
UNIVERSITAS SEBELAS MARET SURAKARTA**

2025

KATA PENGANTAR

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa, karena atas limpahan rahmatnya penyusun dapat menyelesaikan makalah ini tepat waktu tanpa ada halangan yang berarti dan sesuai dengan harapan.

Ucapan terima kasih kami sampaikan kepada Bapak Endra Pratama, S.Si., M.Cs. sebagai dosen pengampu mata kuliah Sistem Digital yang telah membantu memberikan arahan dan pemahaman dalam penyusunan makalah ini.

Kami menyadari bahwa dalam penyusunan makalah ini masih banyak kekurangan karena keterbatasan kami. Maka dari itu penyusun sangat mengharapkan kritik dan saran untuk menyempurnakan makalah ini. Semoga apa yang ditulis dapat bermanfaat bagi semua pihak yang membutuhkan.

Surakarta, 14 Desember 2025

Penulis

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	ii
DAFTAR GAMBAR	iii
DAFTAR TABEL	iv
DAFTAR LAMPIRAN	v
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Tujuan.....	2
BAB II LANDASAN TEORI	3
2.1 Gerbang Logika.....	3
2.2 Flip-flop	4
2.3 Counter.....	5
2.4 Decoder	7
2.5 Seven Segment Display.....	7
2.6 Sinyal Detak (<i>Clock Signal</i>).....	8
2.7 Multiplexer.....	9
2.8 Finite State Machine	10
BAB III PEMBAHASAN	12
3.1 Perancangan Sistem	12
3.2 Perancangan Decoder.....	13
3.3 Perancangan Counter	14
3.4 Perancangan Kontrol.....	15
3.5 Hasil Pengujian dan Simulasi	16
BAB IV PENUTUP	18
4.1 Kesimpulan	18
4.2 Saran.....	18
DAFTAR PUSTAKA	19
LAMPIRAN	20

DAFTAR GAMBAR

Gambar 2.1 Simbol Skematik Flip-flop.....	5
Gambar 2.2 IC 74LS192 sebagai <i>Synchronous Decade Up/Down Counter</i>	6
Gambar 2.3 Diagram Blok IC sebagai Dekoder BCD ke 7-Segmen.....	7
Gambar 2.4 Diagram dua macam 7-Segmen.....	8
Gambar 2.5 Bentuk Gelombang Sinyal Clock Ideal dengan Periode 1 Detik.....	9
Gambar 2.6 Diagram Blok Logika Multiplexer 2-ke-1.....	10
Gambar 2.7 Diagram <i>state</i> FSM pada Perancangan Logika Kontrol Jam Catur.....	10
Gambar 3.1 Diagram Blok Arsitektur Sistem Jam Catur Digital.....	12
Gambar 3.2 Rangkaian Utama Jam Catur Digital.....	12
Gambar 3.3 Rangkaian Logika Kombinasional Internal Decoder BCD ke 7-Segment (Gate Level).....	14
Gambar 3.4 Skema Rangkaian <i>Cascaded Counter</i> (Pencacah Bertingkat) untuk Menit dan Detik pada Sub-sirkuit (MM:SS).....	15
Gambar 3.5 Rangkaian Logika Kontrol Utama (Main Control Logic).....	16
Gambar 3.6 Tampilan Sistem pada Kondisi Awal (Ready/Reset).....	16
Gambar 3.7 Simulasi Giliran Pemain 1 (Magnus).....	17
Gambar 3.8 Simulasi Giliran Pemain 2 (Gamma).....	17

DAFTAR TABEL

Tabel 2.1 Jenis-jenis Gerbang Logika.....	3
---	---

DAFTAR LAMPIRAN

Lampiran 1. Tabel Kebenaran Decoder BCD ke 7-Segment.....	20
Lampiran 2. Rangkaian Sub-Sirkuit (Logisim).....	20
Lampiran 3. Repozitori Github Proyek.....	21

BAB I

PENDAHULUAN

1.1 Latar Belakang

Catur adalah sebuah permainan strategis yang melibatkan dua orang pemain dan sering dianggap sebagai olahraga otak. Dalam permainan ini, terdapat perangkat krusial berupa jam catur yang berfungsi membatasi waktu pikir masing-masing pemain agar turnamen berjalan sesuai jadwal. Seiring dengan perkembangan zaman, jam catur digital hadir sebagai solusi modern menggantikan jam catur mekanik konvensional. Jam catur digital memanfaatkan teknologi elektronika untuk memberikan tingkat keakuratan yang lebih tinggi, kemudahan pemakaian, serta berbagai fitur pendukung lainnya.

Dari sudut pandang sistem digital, jam catur merupakan aplikasi yang menarik karena menggabungkan berbagai konsep dasar elektronika digital. Perancangan jam catur digital melibatkan penggunaan gerbang logika untuk operasi dasar, seperti *flip-flop* untuk penyimpanan data, *counter* untuk pencacahan waktu, *decoder* untuk konversi kode, *seven segment display* untuk menampilkan angka, dan *finite state machine* (FSM) untuk mengatur logika kontrol sistem.

Selain itu, perancangan jam catur juga mengajarkan pentingnya sinkronisasi waktu, manajemen *state*, dan desain antarmuka pengguna yang intuitif. Melalui proyek ini, pemahaman komprehensif mengenai integrasi komponen sistem digital dapat tercapai. Oleh karena itu, makalah ini disusun untuk memberikan pemahaman mendalam mengenai perancangan dan implementasi jam catur digital ditinjau dari perspektif sistem digital.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, rumusan masalah dalam makalah ini adalah sebagai berikut:

1. Bagaimana merancang arsitektur sistem jam catur digital yang mengintegrasikan logika kombinasional dan sekuensial.

2. Bagaimana mekanisme kerja *finite state machine* (FSM) beserta komponen pendukung (*counter* dan *decoder*) dalam mengetur transisi *state* dan kontrol waktu?
3. Bagaimana unjuk kerja dan akurasi sistem jam catur digital berdasarkan hasil pengujian simulasi?

1.3 Tujuan

Tujuan dari penyusunan makalah ini adalah sebagai berikut:

1. Merancang sistem jam catur digital menggunakan komponen dasar sistem digital, seperti gerbang logika, *flip-flop*, dan *multiplexer*.
2. Menjelaskan implementasi logika kontrol sistem, mencakup peran FSM dalam manajemen *state* dan sinkronisasi antara pencacah waktu dan tampilan.
3. Memvalidasi fungsionalitas sistem melalui simulasi untuk memastikan jam catur bekerja sesuai dengan spesifikasi yang dirancang.

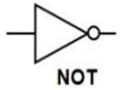
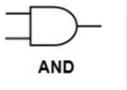
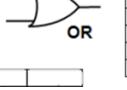
BAB II

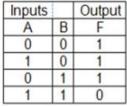
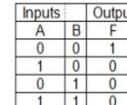
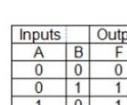
LANDASAN TEORI

2.1 Gerbang Logika

Gerbang logika adalah komponen dasar dalam sistem elektronika digital yang berfungsi untuk melakukan operasi logika berdasarkan prinsip aljabar Boolean. Komponen ini beroperasi pada dua level tegangan yang merepresentasikan logika biner, yaitu logika *HIGH* (1) dan logika *LOW* (0). Pada perancangan jam catur digital, gerbang logika digunakan secara luas untuk menyusun logika kontrol dan pemrosesan sinyal.

Terdapat beberapa jenis gerbang logika dasar yang memiliki karakteristik *input* dan *output* berbeda. Karakteristik kerja masing-masing gerbang logika dirangkum dalam Tabel 2.1 berikut:

No.	Gambar	Nama Gerbang	Simbol Aljabar	Prinsip Kerja																	
1.	  <table border="1"> <tr> <th>Input</th> <th>Output</th> </tr> <tr> <td>I</td> <td>F</td> </tr> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </table>	Input	Output	I	F	0	1	1	0	NOT	$F = \bar{A}$	Menghasilkan <i>output</i> yang merupakan kebalikan dari <i>input</i> . Jika <i>input</i> bernilai 1, maka <i>output</i> bernilai 0, dan sebaliknya.									
Input	Output																				
I	F																				
0	1																				
1	0																				
2.	  <table border="1"> <tr> <th>Inputs</th> <th>Output</th> </tr> <tr> <td>A</td> <td>B</td> <td>F</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </table>	Inputs	Output	A	B	F	0	0	0	1	0	0	0	1	0	1	1	1	AND	$F = A \cdot B$	Menghasilkan <i>output</i> bernilai 1 hanya jika seluruh <i>input</i> bernilai 1. Jika salah satu <i>input</i> bernilai 0, maka <i>output</i> bernilai 0.
Inputs	Output																				
A	B	F																			
0	0	0																			
1	0	0																			
0	1	0																			
1	1	1																			
3.	  <table border="1"> <tr> <th>Inputs</th> <th>Output</th> </tr> <tr> <td>A</td> <td>B</td> <td>F</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </table>	Inputs	Output	A	B	F	0	0	0	1	0	1	0	1	1	1	1	1	OR	$F = A + B$	Menghasilkan <i>output</i> bernilai 1 jika salah satu atau seluruh <i>input</i> bernilai 1. <i>Output</i> bernilai 0 hanya jika semua <i>input</i> bernilai 0.
Inputs	Output																				
A	B	F																			
0	0	0																			
1	0	1																			
0	1	1																			
1	1	1																			

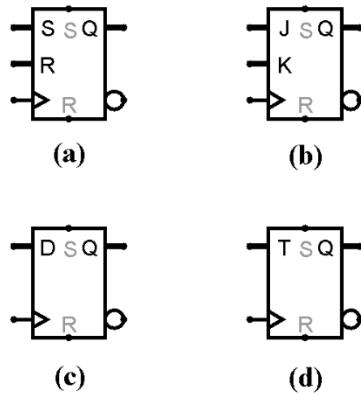
4.	 NAND	 <table border="1"> <thead> <tr> <th colspan="2">Inputs</th> <th>Output</th> </tr> <tr> <th>A</th> <th>B</th> <th>F</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	Inputs		Output	A	B	F	0	0	1	1	0	1	0	1	1	1	1	0	NAND	$F = \overline{A \cdot B}$	Merupakan kebalikan dari gerbang AND. Menghasilkan <i>output</i> bernilai 0 hanya jika seluruh <i>input</i> bernilai 1. Selain kondisi tersebut, <i>output</i> bernilai 1.
Inputs		Output																					
A	B	F																					
0	0	1																					
1	0	1																					
0	1	1																					
1	1	0																					
5.	 NOR	 <table border="1"> <thead> <tr> <th colspan="2">Inputs</th> <th>Output</th> </tr> <tr> <th>A</th> <th>B</th> <th>F</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	Inputs		Output	A	B	F	0	0	1	1	0	0	0	1	0	1	1	0	NOR	$F = \overline{A + B}$	Merupakan kebalikan dari gerbang OR. Menghasilkan <i>output</i> bernilai 1 hanya jika seluruh <i>input</i> bernilai 0. Jika salah satu <i>input</i> bernilai 1, maka <i>output</i> bernilai 0.
Inputs		Output																					
A	B	F																					
0	0	1																					
1	0	0																					
0	1	0																					
1	1	0																					
6.	 EXCLUSIVE OR	 <table border="1"> <thead> <tr> <th colspan="2">Inputs</th> <th>Output</th> </tr> <tr> <th>A</th> <th>B</th> <th>F</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	Inputs		Output	A	B	F	0	0	0	0	1	1	1	0	1	1	1	0	XOR	$F = A \oplus B$	Menghasilkan <i>output</i> bernilai 1 jika <i>input</i> memiliki nilai logika yang berbeda (misalnya 1 dan 0). Jika <i>input</i> memiliki nilai yang sama, maka <i>output</i> bernilai 0.
Inputs		Output																					
A	B	F																					
0	0	0																					
0	1	1																					
1	0	1																					
1	1	0																					

Tabel 2.1 Jenis-jenis Gerbang Logika

2.2 Flip-flop

Flip-flop adalah rangkaian digital logika sekuensial yang mampu menyimpan data sebesar 1-bit, sehingga sering disebut sebagai sel memori dasar. Berbeda dengan gerbang logika yang outputnya hanya bergantung pada input saat ini, *output* flip-flop bergantung pada *input* saat ini dan kondisi *output* sebelumnya (memori).

Ciri utama flip-flop adalah memiliki dua output yang saling berkebalikan, yaitu Q (output data) dan \bar{Q} (komplemen)². Perubahan kondisi output pada flip-flop umumnya disinkronisasi oleh sinyal detak (*clock*). Terdapat beberapa jenis flip-flop yang umum digunakan dalam perancangan sistem digital, antara lain:



Gambar 2.1 Simbol Skematik Flip-flop (a) SR, (b) JK, (c) D, (d) T

1. SR Flip-flop (Set-Reset)

Jenis paling dasar yang memiliki input *Set* (untuk mengaktifkan logika 1) dan *Reset* (untuk mematikan logika 0). Kelemahannya adalah memiliki kondisi terlarang jika input S dan R aktif bersamaan.

2. JK Flip-flop

Pengembangan dari SR Flip-Flop yang mengatasi masalah kondisi terlarang. Jika kedua input J dan K berlogika 1, maka output akan *toggle* (berubah kondisi kebalikan dari sebelumnya). Fitur *toggle* ini menjadikan JK Flip-Flop komponen ideal untuk menyusun rangkaian pencacah (*counter*).

3. D Flip-flop (Data/Delay)

Memiliki satu input data. Output *Q* akan mengikuti input *D* setiap kali sinyal *clock* memicu. Berfungsi untuk menunda data atau sebagai penyimpan buffer.

4. T Flip-flop (Toggle)

Versi sederhana dari JK Flip-Flop di mana input J dan K digabung. Jika input *T* aktif, output akan selalu berpindah kondisi (*toggle*) setiap siklus *clock*.

2.3 Counter

Rangkaian pencacah atau *counter* adalah rangkaian logika sekuensial yang berfungsi untuk menghitung jumlah pulsa detak (*clock*) yang masuk. Rangkaian ini tersusun dari serangkaian *flip-flop* yang dimanipulasi

sedemikian rupa sehingga output-nya dapat merepresentasikan urutan bilangan biner tertentu.

Dalam sistem digital, pencacah diklasifikasikan menjadi beberapa jenis berdasarkan mekanisme pemicuannya:

1. Asynchronous Counter (Ripple Counter)

Output dari *flip-flop* pertama menjadi input *clock* untuk *flip-flop* berikutnya. Kelebihannya adalah rangkaian lebih sederhana, namun memiliki kelemahan berupa *propagation delay* (tunda waktu) yang terakumulasi sehingga kurang cocok untuk frekuensi tinggi .

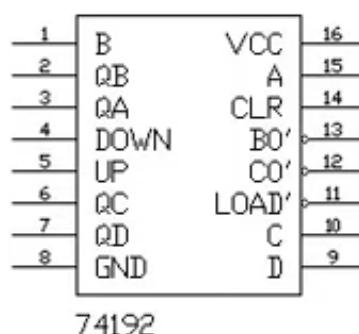
2. Synchronous Counter

Seluruh *flip-flop* dipicu secara serentak oleh satu sumber *clock* yang sama. Hal ini meminimalisir *delay* waktu dan membuat perubahan output terjadi secara simultan. Jenis ini lebih disukai dalam perancangan sistem presisi seperti jam digital .

Selain itu, berdasarkan arah cacahannya, *counter* dibedakan menjadi:

- *Up Counter*: Mencacah dari nilai kecil ke besar (0, 1, 2, ...)
- *Down Counter*: Mencacah dari nilai besar ke kecil (... , 2, 1, 0)

Dalam perancangan jam catur digital, Synchronous Down Counter dengan modulus tertentu (Mod-10 untuk satuan detik, Mod-6 untuk puluhan detik) memegang peranan penting untuk merealisasikan fitur hitung mundur waktu permainan.



Gambar 2.2 IC 74LS192 sebagai *Synchronous Decade Up/Down Counter*

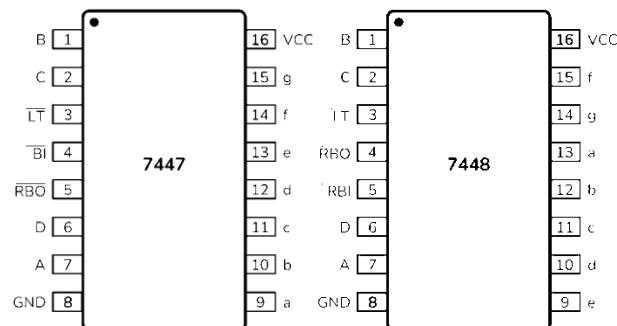
Sumber: Ovaga (2025)

2.4 Decoder

Decoder adalah rangkaian logika kombinasional yang berfungsi untuk menerima input informasi biner dan menerjemahkannya menjadi output pada saluran tertentu yang aktif. Secara umum, *decoder* memiliki n saluran input dan hingga 2^n saluran output.

Namun, dalam aplikasi tampilan digital seperti jam catur, jenis *decoder* yang paling krusial adalah BCD to 7-Segment Decoder. Rangkaian pencacah (*counter*) umumnya menghasilkan output dalam format BCD (*Binary Coded Decimal*) yang terdiri dari 4-bit (misalnya 1001 untuk angka 9). Agar informasi ini dapat dipahami manusia, *decoder* diperlukan untuk mengonversi kode 4-bit tersebut menjadi logika aktivasi untuk 7 segmen LED (a, b, c, d, e, f, g) yang membentuk pola angka desimal.

Contoh komponen standar yang sering digunakan adalah IC 74LS47 (7447) untuk *Common Anode* atau 74LS48 (7448) untuk *Common Cathode*.



Gambar 2.3 Diagram Blok IC sebagai Dekoder BCD ke 7-Segmen

Sumber: Build-Electronic-Circuits (2025), diadaptasi

2.5 Seven Segment Display

Seven Segment Display adalah komponen elektronika yang berfungsi sebagai penampil visual angka desimal (0-9). Komponen ini tersusun dari tujuh buah *Light Emitting Diode* (LED) yang diatur sedemikian rupa membentuk pola angka delapan. Setiap segmen LED dilabeli dengan huruf a hingga g dan seringkali dilengkapi dengan satu LED tambahan (dp atau *dot point*) untuk menandakan titik desimal.

Prinsip kerja komponen ini didasarkan pada polaritas penyambungan kaki LED-nya. Terdapat dua konfigurasi standar, yaitu:

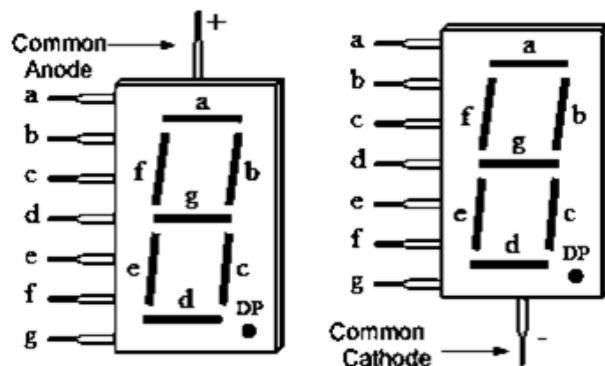
1. *Common Anode* (Anoda Bersama)

Semua kaki anoda (positif) dari LED disatukan dan dihubungkan ke sumber tegangan VCC. Untuk menyalakan segmen tertentu, kaki katoda harus diberi logika LOW (0). Konfigurasi ini kompatibel dengan IC decoder tipe 74LS47.

2. *Common Cathode* (Katoda Bersama)

Semua kaki katoda (negatif) disatukan ke *Ground*. Untuk menyalakan segmen, kaki anoda harus diberi logika HIGH (1). Konfigurasi ini kompatibel dengan IC decoder tipe 74LS48.

Dalam perancangan jam catur ini, pemilihan jenis *seven segment* harus disesuaikan dengan jenis IC *decoder* yang digunakan agar logika penyalaan segmen berjalan sinkron.



Gambar 2.4 Diagram dua macam 7-Segmen
Sumber: IC-Components (2025), diadaptasi

2.6 Sinyal Detak (*Clock Signal*)

Sinyal detak atau *clock* adalah sinyal digital yang berosilasi secara periodik antara logika HIGH (1) dan LOW (0). Dalam sistem digital sekuensial, sinyal ini berfungsi sebagai referensi waktu utama (metronom) untuk menyinkronkan perpindahan *state* pada komponen memori seperti *flip-flop* dan *counter*. Parameter utama sinyal *clock* meliputi:

1. Periode (T)

Waktu yang dibutuhkan untuk menyelesaikan satu siklus gelombang penuh.

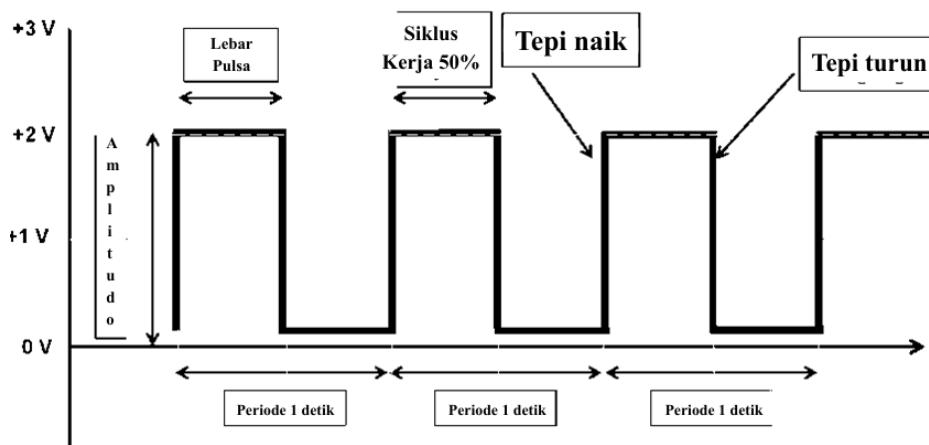
2. Frekuensi (f)

Jumlah siklus yang terjadi dalam satu detik ($f = 1/T$). Untuk penerapan jam digital yang menghitung waktu dalam satuan detik, diperlukan frekuensi *clock* sebesar 1 Hz.

3. Edge Triggering (Pemicuan Tepi)

Perubahan kondisi pada rangkaian biasanya terjadi pada saat transisi sinyal, baik itu tepi naik (*rising edge*, transisi 0 ke 1) atau tepi turun (*falling edge*, transisi 1 ke 0).

Kualitas dan stabilitas sinyal *clock* sangat krusial dalam perancangan jam catur digital untuk memastikan akurasi perhitungan waktu pemain.



Gambar 2.5 Bentuk Gelombang Sinyal Clock Ideal dengan Periode 1 Detik

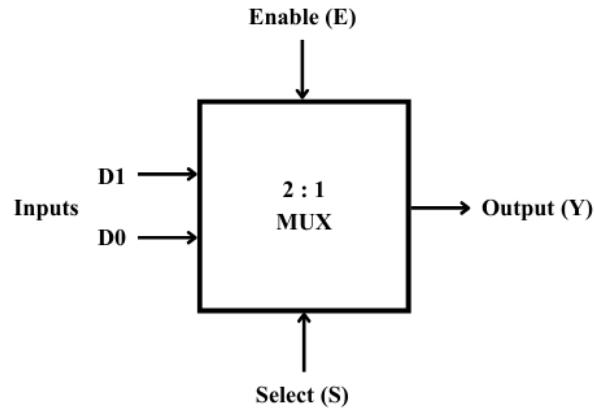
Sumber: Lombardi, M. A. (2021), diadaptasi

2.7 Multiplexer

Multiplexer (MUX) atau selektor data adalah rangkaian logika kombinasional yang berfungsi untuk memilih satu dari beberapa jalur input data dan meneruskannya ke satu jalur output. Pemilihan jalur ditentukan oleh sinyal kendali (*selector line*).

Prinsip kerja *multiplexing* ini sangat krusial dalam sistem jam catur digital, khususnya untuk fungsi Pengatur Giliran (*Turn Switching*). Karena jam catur hanya mengizinkan satu pemain yang waktunya berjalan pada satu waktu, diperlukan logika selektor untuk mengarahkan sinyal detak (*clock*) hanya ke pencacah (*counter*) pemain yang sedang mendapat giliran,

sementara memblokir sinyal ke pemain lawan. Implementasi logika ini dapat dilakukan menggunakan IC Multiplexer khusus atau kombinasi gerbang logika dasar (AND/OR) sebagai sakelar elektronik.

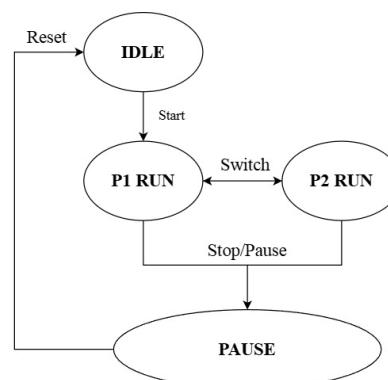


Gambar 2.6 Diagram Blok Logika Multiplexer 2-ke-1

2.8 Finite State Machine

Finite State Machine (FSM) atau mesin keadaan terhingga adalah model komputasi matematis yang digunakan untuk merancang logika kontrol sekusensial. FSM bekerja berdasarkan konsep bahwa sebuah sistem hanya dapat berada dalam satu "keadaan" (*state*) pada satu waktu tertentu. Perpindahan dari satu *state* ke *state* lainnya disebut transisi yang dipicu oleh *input* eksternal atau kondisi tertentu.

Di dalam perancangan jam catur digital, FSM bertindak sebagai pengendali utama (*main controller*) yang mengatur alur permainan. Sistem ini dapat dimodelkan menggunakan Diagram State (*State Diagram*) dengan keadaan-keadaan utama sebagai berikut:



Gambar 2.7 Diagram *state* FSM pada Perancangan Logika Kontrol Jam Catur

1. *IDLE/RESET*: Kondisi awal saat sistem baru dinyalakan atau di-reset.
Waktu belum berjalan.
2. *RUN P1*: Giliran Pemain 1 (Putih). Waktu P1 berkurang, Waktu P2 diam.
3. *RUN P2*: Giliran Pemain 2 (Hitam). Waktu P2 berkurang, Waktu P1 diam.
4. *PAUSE*: Permainan dihentikan sementara. Kedua waktu diam.
Penerapan FSM memastikan tidak terjadi konflik logika, seperti waktu kedua pemain berjalan bersamaan atau jam berjalan saat kondisi *pause*.

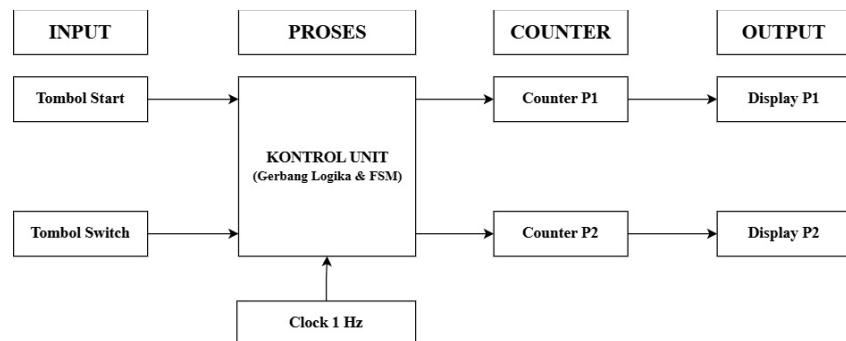
BAB III

PEMBAHASAN

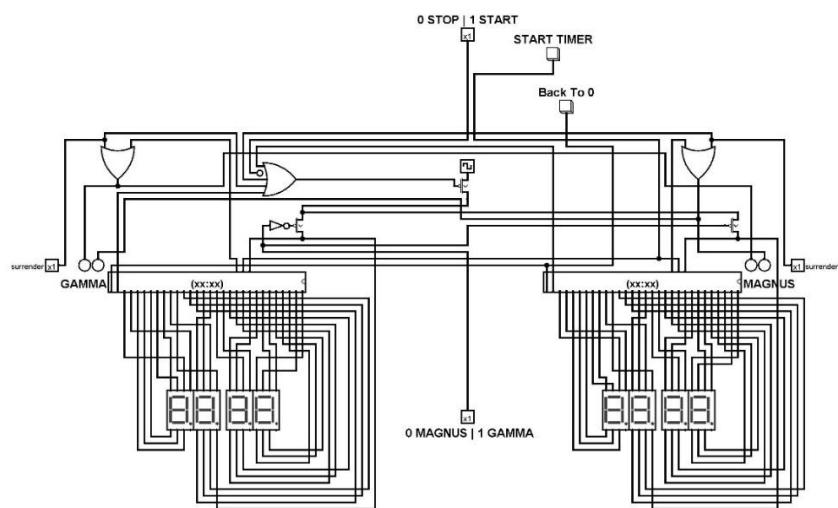
3.1 Perancangan Sistem

Rangkaian dasar jam catur digital ini diadaptasi dan dikembangkan dari desain referensi terbuka oleh Aladeenb pada platform GitHub [1] yang kemudian dimodifikasi oleh penulis untuk memenuhi spesifikasi tugas akhir ini. Perancangan jam catur digital ini dilakukan dengan pendekatan modular, membagi sistem menjadi beberapa blok fungsional utama yang saling terintegrasi. Hal ini bertujuan untuk mempermudah proses *debugging* dan analisis logika. Berdasarkan analisis kebutuhan, sistem dirancang terdiri dari empat bagian utama: blok input, blok kontrol pusat (*Control Unit*), blok pencacah waktu (*Counter*), dan blok antarmuka tampilan (*Display Interface*).

Arsitektur sistem secara umum dapat dilihat pada diagram blok berikut:



Gambar 3.1 Diagram Blok Arsitektur Sistem Jam Catur Digital



Gambar 3.2 Rangkaian Utama Jam Catur Digital

Berdasarkan Gambar 3.1, prinsip kerja sistem adalah sebagai berikut:

1. Blok Input

Terdiri dari antarmuka tombol (*push button*) yaitu *Tombol Start* untuk memulai permainan dan *Tombol Switch* untuk memindahkan giliran antar pemain. Sinyal dari blok ini menjadi pemicu utama perubahan kondisi (*state*) pada sistem.

2. Blok Proses (Kontrol Unit)

Merupakan otak dari sistem yang berisi rangkaian gerbang logika dan implementasi *Finite State Machine* (FSM). Blok ini menerima sinyal detak (*Clock 1 Hz*) dan mendistribusikannya ke pencacah pemain yang sedang aktif sesuai logika giliran.

3. Blok Counter

Terdiri dari dua unit pencacah mundur (*down counter*) terpisah, yaitu *Counter P1* (Putih) dan *Counter P2* (Hitam), yang berfungsi menyimpan dan memproses sisa waktu permainan.

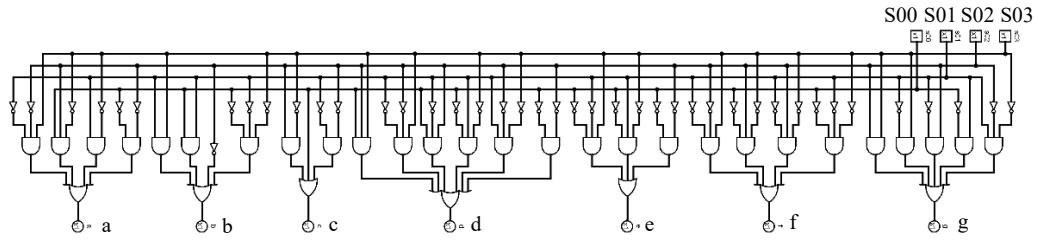
4. Blok Output

Merupakan antarmuka visual berupa *Display Seven Segment* yang menampilkan data waktu dari masing-masing *counter* (Display P1 dan Display P2) agar dapat dibaca oleh pemain.

3.2 Perancangan Decoder

Pada sistem ini, modul *decoder* tidak menggunakan IC standar (seperti 7447), melainkan dirancang secara manual menggunakan prinsip logika kombinasional dasar. Tujuannya adalah untuk menerjemahkan sinyal biner 4-bit dari pencacah (*counter*) menjadi sinyal aktif yang sesuai untuk setiap segmen LED (a, b, c, d, e, f, g).

Realisasi rangkaian *decoder* dibangun berdasarkan Tabel Kebenaran *BCD to 7-Segment* (lihat di lampiran) yang diimplementasikan menggunakan gerbang logika dasar AND, OR, dan NOT. Struktur internal dari rangkaian *decoder* tersebut diperlihatkan pada Gambar 3.3.



Gambar 3.3 Rangkaian Logika Kombinasional Internal Decoder BCD ke 7-Segment (Gate Level)

Berdasarkan Gambar 3.3, dapat dilihat bahwa setiap jalur output segmen (a-g) merupakan hasil dari persamaan Boolean yang dibentuk oleh kombinasi gerbang logika. Sinyal input (s00-s03) diproses secara paralel untuk menghasilkan pola nyala angka desimal pada layar secara *real-time*.

3.3 Perancangan Counter

Rangkaian pencacah (*counter*) pada sistem ini berfungsi sebagai penyimpan data waktu dan melakukan operasi perhitungan mundur (*countdown*) setiap detik. Jenis pencacah yang digunakan adalah *Synchronous Down Counter* yang disusun secara bertingkat (*cascaded*).

Sistem waktu pada jam catur (Menit:Detik) memerlukan konfigurasi modulus yang berbeda untuk setiap digitnya agar sesuai dengan format waktu standar:

1. Counter Satuan Detik

Dikonfigurasi sebagai Modulus-10 (Decade Counter). Pencacah ini menghitung dari 9 hingga 0. Saat mencapai 0, ia akan mengirimkan sinyal pemicu (*borrow*) ke pencacah puluhan detik.

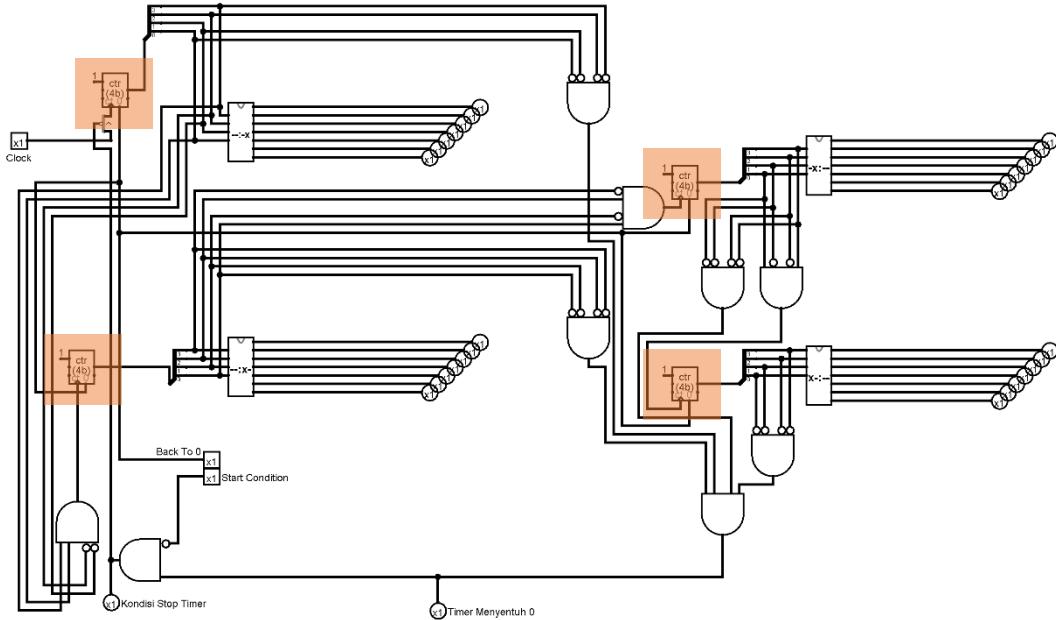
2. Counter Puluhan Detik

Dikonfigurasi sebagai Modulus-6. Pencacah ini menghitung dari 5 hingga 0, merepresentasikan siklus 60 detik dalam satu menit.

3. Counter Satuan & Puluhan Menit

Dikonfigurasi serupa untuk menghitung sisa menit permainan.

Implementasi rangkaian ini di Logisim menggunakan komponen *Counter 4-bit* yang saling terhubung (*di-cascade*), seperti diperlihatkan pada Gambar 3.4.



Gambar 3.4 Skema Rangkaian *Cascaded Counter* (Pencacah Bertingkat) untuk Menit dan Detik pada Sub-sirkuit (MM:SS).

3.4 Perancangan Kontrol

Unit kontrol merupakan bagian paling krusial yang berfungsi sebagai pengatur alur permainan. Bagian ini menerjemahkan input dari tombol pengguna menjadi sinyal kendali yang mengaktifkan atau mematikan pencacah (*counter*) pemain tertentu. Logika kontrol ini dirancang menggunakan kombinasi gerbang logika (AND, OR, NOT) dan elemen memori (*flip-flop* atau *latch*) untuk membentuk sebuah *Finite State Machine* (FSM). Terdapat dua mekanisme kontrol utama dalam sistem ini:

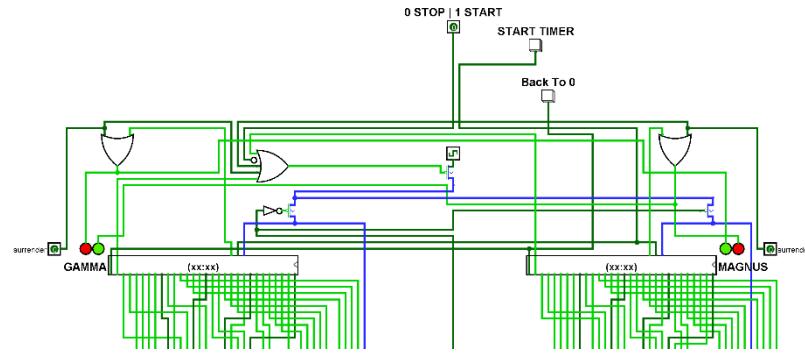
1. *Gating Clock* (Gerbang Detak)

Sinyal clock utama 1 Hz tidak langsung dihubungkan ke kedua counter. Melainkan, ia melewati gerbang AND yang berfungsi sebagai sakelar. Jika giliran pemain 1 aktif (logika 1), maka gerbang AND pemain 1 terbuka dan *clock* masuk. Sebaliknya, jika giliran pemain 2, maka gerbang pemain 1 tertutup.

2. *State Switching* (Perpindahan Keadaan)

Menggunakan rangkaian *flip-flop* atau *latch* sederhana yang kondisinya dibalik (*toggled*) oleh tombol *Switch*. *Output* dari rangkaian ini (*Q* dan *Q'*) digunakan sebagai sinyal *Enable* untuk memilih pemain mana yang aktif.

Implementasi logika kontrol ini pada rangkaian utama (*Main Circuit*) ditunjukkan pada Gambar 3.5.



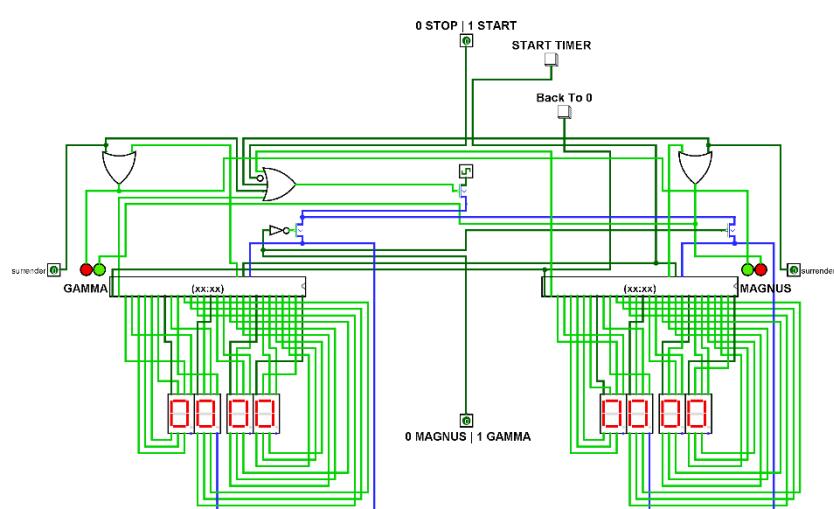
Gambar 3.5 Rangkaian Logika Kontrol Utama (Main Control Logic)

3.5 Hasil Pengujian dan Simulasi

Pengujian sistem dilakukan menggunakan simulator Logisim untuk memverifikasi fungsionalitas logika kontrol dan akurasi pencacahan waktu. Skenario pengujian mencakup kondisi awal (*reset*), perpindahan giliran (*switching*), dan fitur jeda (*pause*). Berikut adalah hasil simulasi dari beberapa kondisi utama:

1. Kondisi Awal (Reset/Idle)

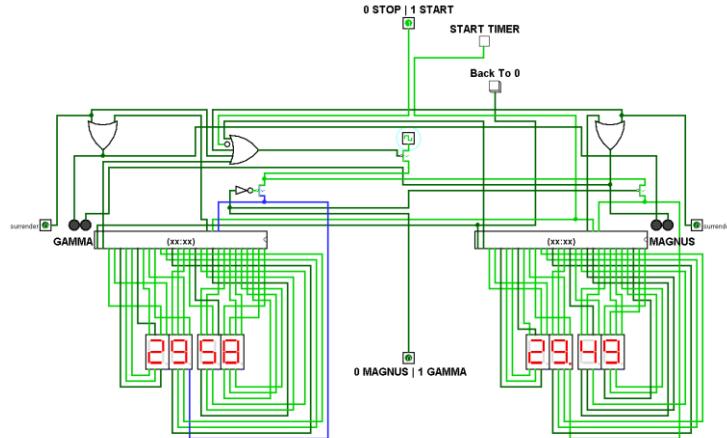
Pada kondisi ini, sistem baru dinyalakan atau tombol *Reset* ditekan. Kedua pencacah waktu menunjukkan nilai awal (misalnya :00 atau sesuai konfigurasi) dan belum ada aktivitas pencacahan.



Gambar 3.6 Tampilan Sistem pada Kondisi Awal (Ready/Reset)

2. Kondisi Giliran Pemain 1 (White Turn)

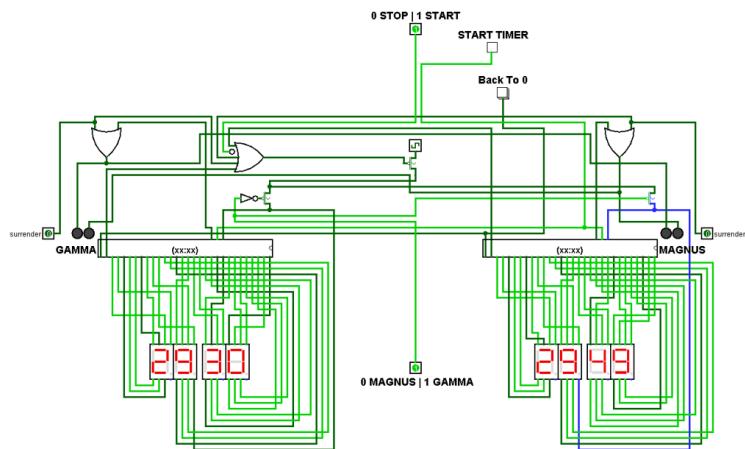
Saat tombol *Switch* ditekan (atau logika *selector* bernilai 0), logika kontrol mengalihkan jalur *clock* ke pencacah Pemain 1 (Magnus). Indikator LED di sisi Pemain 1 menyala dan waktu mulai menghitung mundur, sementara waktu Pemain 2 (Gamma) dibekukan (tetap diam).



Gambar 3.7 Simulasi Giliran Pemain 1 (Magnus)

3. Kondisi Giliran Pemain 2 (Black Turn)

Saat tombol *Switch* ditekan (atau logika *selector* bernilai 0), logika kontrol mengalihkan jalur *clock* ke pencacah Pemain 2 (Gamma). Indikator LED di sisi Pemain 2 menyala dan waktu mulai menghitung mundur, sementara waktu Pemain 1 (Magnus) dibekukan (tetap diam).



Gambar 3.8 Simulasi Giliran Pemain 2 (Gamma)

Berdasarkan hasil simulasi di atas, dapat disimpulkan bahwa rangkaian jam catur digital telah bekerja sesuai dengan prinsip kerja yang dirancang.

BAB IV

PENUTUP

4.1 Kesimpulan

Berdasarkan perancangan dan simulasi yang dilakukan, dapat disimpulkan bahwa:

1. Logika Kontrol

Control Unit berbasis FSM berhasil mengatur seluruh alur permainan (*start, switch, pause*) tanpa konflik logika.

2. Akurasi Waktu

Rangkaian *Cascaded Down Counter* berfungsi akurat dalam melakukan perhitungan mundur sesuai format waktu standar.

3. Visualisasi

Decoder manual yang dirancang sukses menerjemahkan sinyal biner ke tampilan *Seven Segment* dengan jelas.

4. Fungsionalitas

Simulasi Logisim membuktikan seluruh blok sistem bekerja terintegrasi dan memenuhi spesifikasi jam catur digital.

4.2 Saran

Demi pengembangan sistem yang lebih optimal di masa mendatang, penulis menyampaikan beberapa saran sebagai berikut:

1. Realisasi Perangkat Keras

Mengimplementasikan rancangan simulasi ke dalam *hardware* nyata menggunakan IC TTL atau FPGA.

2. Mode Waktu Variatif

Mengembangkan logika sistem untuk mendukung mode *Fischer Increment* atau penambahan waktu pasca-langkah.

3. Fitur Audio

Menambahkan modul *buzzer* sebagai indikator suara saat tombol ditekan atau waktu habis.

DAFTAR PUSTAKA

- Aladeenb. (2020). *Digital Chess Timer: Simulation of a chess timer using Logisim [Source code]*. GitHub. <https://github.com/Aladeenb/digital-chess-timer>
- Analog Devices. (2022). *Fundamental of Multiplexers: Basic Principles and Applications*. Analog Devices Technical Articles.
- Dahl, Ø. N. (2023). *7400 Series Guide: 74HC48/74LS48 (BCD to 7-Segment)*. Build Electronic Circuits. <https://www.build-electronic-circuits.com/7400-series-integrated-circuits/74hc48-74ls48/>
- Floyd, T. L. (2015). *Digital Fundamentals* (11th ed.). Pearson Education.
- Harris, D. M., & Harris, S. L. (2012). *Digital Design and Computer Architecture* (2nd ed.). Morgan Kaufmann.
- IC Components. (2024). *Comprehensive Guide to Understanding 7-Segment LED Displays*. <https://www.ic-components.com/blog/comprehensive-guide-to-understanding-7-segment-led-displays.jsp>
- Lombardi, M. A. (2021). *An Evaluation of Dependencies of Critical Infrastructure Timing Systems on the Global Positioning System (GPS)* (NIST Technical Note 2189). National Institute of Standards and Technology. <https://doi.org/10.6028/NIST.TN.2189>
- Maini, A. K. (2007). *Digital Electronics: Principles, Devices and Applications*. John Wiley & Sons.
- Ovaga Technologies. (2023, October 27). *IC 74192 UP/Down Counter Datasheet PDF, Circuit, Pin Diagram*. <https://www.ovaga.com/blog/package/ic-74192-up-down-counter-datasheet-pdf-circuit-pin-diagram>
- Texas Instruments. (1988). *SN74LS47 BCD-to-Seven-Segment Decoders/Drivers (Rev. A)* [Data sheet]. Texas Instruments Incorporated.
- Texas Instruments. (1988). *SN74LS192, SN74LS193 Synchronous 4-Bit Up/Down Counters (Rev. A)* [Data sheet]. Texas Instruments Incorporated.
- Tocci, R. J., Widmer, N. S., & Moss, G. L. (2017). *Digital Systems: Principles and Applications* (12th ed.). Pearson Education.
- Wakerly, J. F. (2018). *Digital Design: Principles and Practices* (5th ed.). Pearson Education.

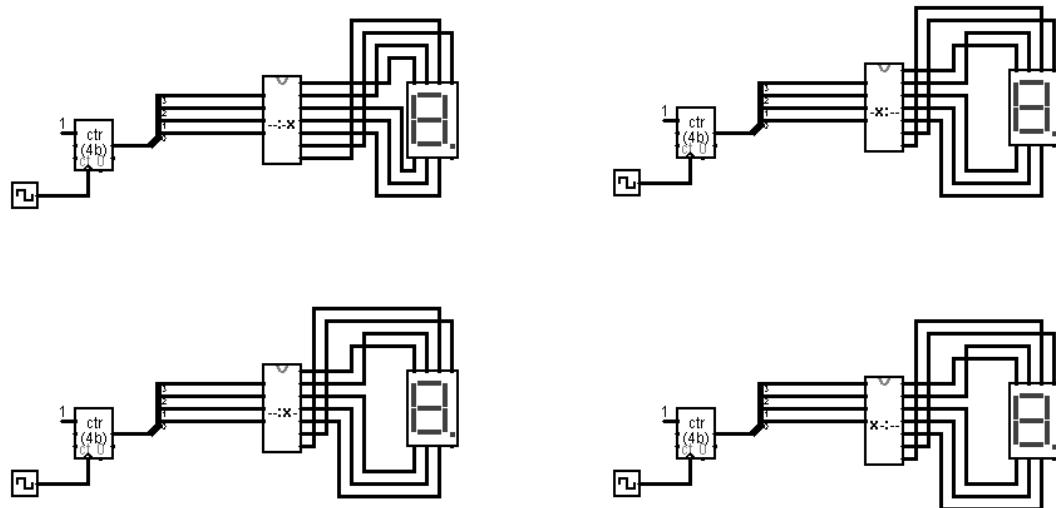
LAMPIRAN

Lampiran 1. Tabel Kebenaran Decoder BCD ke 7-Segment

s03	s02	s01	s00	g	f	e	d	c	b	a
0	0	0	0	0	1	1	1	1	1	1
0	0	0	1	0	0	0	0	1	1	0
0	0	1	0	1	0	1	1	0	1	1
0	0	1	1	1	0	0	1	1	1	1
0	1	0	0	1	1	0	0	1	1	0
0	1	0	1	1	1	0	1	1	0	1
0	1	1	0	1	1	1	1	1	0	1
0	1	1	1	0	0	0	0	1	1	1
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	0	1	1	1	1
1	0	1	0	1	0	1	1	0	1	1
1	0	1	1	0	0	0	0	1	1	0
1	1	0	0	1	0	1	0	0	0	0
1	1	0	1	1	0	0	1	1	0	0
1	1	1	0	1	0	0	0	0	1	0
1	1	1	1	1	0	0	1	1	1	0

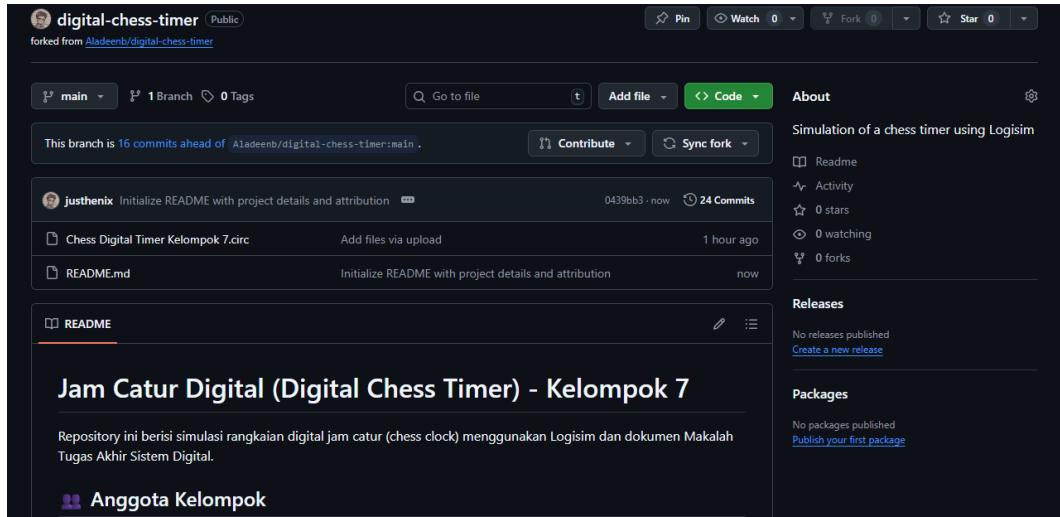
Tabel L1.1 Tabel Kebenaran Decoder BCD ke 7-Segment

Lampiran 2. Rangkaian Sub-Sirkuit (Logisim)



Gambar L2.1 Rangkaian Internal Sub-sirkuit Pencacahan dan Decoder

Lampiran 3. Repotori Github Proyek



Gambar L3.1 Tampilan Repotori GitHub Proyek Kelompok 7.

Tautan Repotori: <https://github.com/justhenix/digital-chess-timer>

Catatan Atribusi:

Proyek ini merupakan pengembangan (fork) dari repotori "digital-chess-timer" oleh Aladeenb. Kredit logika dasar diberikan kepada pengembang asli dengan modifikasi fitur dan antarmuka oleh penulis untuk kebutuhan Tugas Akhir.