

# **LAPORAN PRAKTIKUM**

**Mata Kuliah**  
**Pengantar Sistem Digital**

## **RANGKAIAN ADDER DAN COMPARATOR**

**Dosen: Arif R. Dwiyanto ST., MTI.**



**Disusun oleh:**

**Fenaudry Luthfiyah Gusti Dzakirah (202410715272)**

**F3A6**

**PROGRAM STUDI INFORMATIKA FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS BHAYANGKARA JAKARTA RAYA**

**2025**

# BAB I

## PENDAHULUAN

### 1.1 Tujuan Praktikum

1. Memahami konsep dasar rangkaian penjumlahan biner (Half Adder dan Full Adder) serta fungsi utamanya dalam sistem logika digital. Mahasiswa mampu membangun rangkaian 8-bit Adder menggunakan Logisim.
2. Mempelajari prinsip kerja rangkaian Comparator untuk membandingkan dua bilangan biner dan menentukan hubungan logis di antara keduanya (lebih besar, lebih kecil, atau sama).
3. Mengetahui cara merancang serta menganalisis Adder dan Comparator berdasarkan tabel kebenaran dan persamaan logika.
4. Mampu membuat, menguji, dan mensimulasikan rangkaian Adder dan Comparator menggunakan perangkat lunak Logisim Evolution untuk memastikan fungsionalitas rangkaian.
5. Melatih kemampuan dalam menghubungkan konsep teori logika digital dengan implementasinya secara praktis dalam sistem komputasi maupun rangkaian elektronik.

### 1.2 Langkah-langkah Kerja

1. Membuat 8-bit Adder

- Tabel kebenaran

Ai	Bi	Cin	Sum	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Tabel kebenaran 8-bit adder direpresentasikan melalui tabel kebenaran full adder, karena 8-bit adder disusun dari delapan buah full adder yang beroperasi secara berantai. Rangkaian Adder ini memiliki total 16 bit, sehingga

menghasilkan kombinasi masukan sebesar  $2^{16}$  atau 65.536 baris pada tabel kebenaran. Setiap full adder memiliki tiga input (A, B, dan C-in) serta dua output (Sum dan C-out).

- Bangun rangkaian 8-bit adder dengan ekspreksi logika berikut:

Rumus umum:

$$S_i = A_i \oplus B_i \oplus C_i$$

$$C_i + 1 = (A_i \cdot B_i) + (C_i \cdot (A_i \oplus B_i))$$

Ekspresi logika 8-bit adder:

$$S_0 = A_0 \oplus B_0 \oplus C_0$$

$$C_1 = (A_0 \cdot B_0) + (C_0 \cdot (A_0 \oplus B_0))$$

$$S_1 = A_1 \oplus B_1 \oplus C_1$$

$$C_2 = (A_1 \cdot B_1) + (C_1 \cdot (A_1 \oplus B_1))$$

$$S_2 = A_2 \oplus B_2 \oplus C_2$$

$$C_3 = (A_2 \cdot B_2) + (C_2 \cdot (A_2 \oplus B_2))$$

$$S_3 = A_3 \oplus B_3 \oplus C_3$$

$$C_4 = (A_3 \cdot B_3) + (C_3 \cdot (A_3 \oplus B_3))$$

$$S_4 = A_4 \oplus B_4 \oplus C_4$$

$$C_5 = (A_4 \cdot B_4) + (C_4 \cdot (A_4 \oplus B_4))$$

$$S_5 = A_5 \oplus B_5 \oplus C_5$$

$$C_6 = (A_5 \cdot B_5) + (C_5 \cdot (A_5 \oplus B_5))$$

$$S_6 = A_6 \oplus B_6 \oplus C_6$$

$$C_7 = (A_6 \cdot B_6) + (C_6 \cdot (A_6 \oplus B_6))$$

$$S_7 = A_7 \oplus B_7 \oplus C_7$$

$$C_8 = (A_7 \cdot B_7) + (C_7 \cdot (A_7 \oplus B_7)) \rightarrow \text{Carry akhir (Cout)}$$

Ekspresi logika pada adder 8-bit menggunakan operasi XOR untuk menghasilkan keluaran Sum dan kombinasi AND-OR untuk menghasilkan Carry. Perhitungan dilakukan berantai dari bit paling rendah hingga bit paling tinggi, di mana hasil Carry sebuah bit menjadi Carry-in untuk bit berikutnya.

- Menyusun rangkaian menggunakan delapan gerbang XOR dan delapan gerbang AND pada Logisim.
- Hubungkan input A dan B menggunakan Constant.
- Tambahkan LED pada output Sum (S) dan Carry (C).

## 2. Comparator

- Tabel kebenaran

A	B	Output A > B	Output A = B	Output A < B
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

Tabel kebenaran untuk comparator 8-bit tidak bisa dibuat secara praktis karena tingkat kompleksitasnya sangat tinggi. Comparator ini memiliki 16 bit masukan secara total, yaitu 8 bit untuk input A dan 8 bit untuk input B. Hal ini menghasilkan  $2^{16}$  atau 65.536 kemungkinan kombinasi input yang harus ditampilkan dalam tabel kebenaran. Pada comparator 8-bit, ada tiga output yang mungkin, yaitu A > B, A = B, dan A < B. Ketiga output ini mewakili satu-satunya kondisi yang dapat terjadi ketika dua angka biner dibandingkan. Jika nilai biner A lebih besar dari B, output A > B akan bernilai 1, sedangkan output lainnya akan bernilai 0. Jika A dan B sama, output A = B bernilai 1. Sedangkan saat nilai A lebih kecil daripada B, output A < B akan bernilai 1.

Menentukan kondisi perbandingan bit menggunakan tiga sinyal logika

- $(A = B): E_i = (A_i \oplus B_i)'$
- $(A > B): G_i = A_i \cdot B_i'$
- $(A < B): L_i = A_i' \cdot B_i$

- Menyusun ekspresi logika

- $A = B$ , akan bernilai benar jika semua bit bernilai sama

$$(A = B) = E_7 \cdot E_6 \cdot E_5 \cdot E_4 \cdot E_3 \cdot E_2 \cdot E_1 \cdot E_0$$

- $A > B$ , akan bernilai benar jika pada suatu bit ke-i,  $A > B$  dan semua bit yang lebih signifikan bernilai sama

$$A > B = G_7 + (E_7 G_6) + (E_7 E_6 G_5) + (E_7 E_6 E_5 G_4) + \dots + (E_7 E_6 E_5 E_4 \\ E_3 E_2 E_1 G_0)$$

- $A < B$ , akan bernilai benar jika pada suatu bit ke-i,  $A < B$  dan semua bit yang lebih signifikan bernilai sama

$$A < B = L_7 + (E_7 L_6) + (E_7 E_6 L_5) + (E_7 E_6 E_5 L_4) + \dots + (E_7 E_6 E_5 E_4 E_3 \\ E_2 E_1 L_0)$$

- Menggabungkan semua sinyal kesetaraan

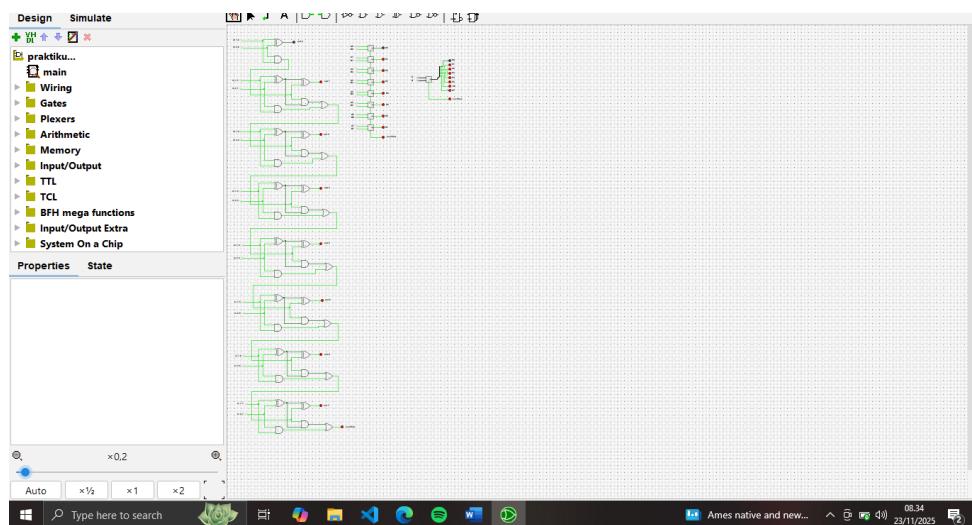
Semua output kesetaraan (E7-E0) dikombinasikan secara berantai menggunakan gerbang AND

- Menentukan prioritas perbandingan. Jika terdapat bit ke-i (dimulai dari MSB) dengan  $G_i = 1$ , maka hasil akhir adalah  $A > B$ , dan rangkaian berhenti mengevaluasi bit yang lebih rendah. Jika terdapat bit ke-i dengan  $L_i = 1$ , maka hasil akhir adalah  $A < B$ .
- Menggabungkan logika hasil (G, L, E)
  - Sinyal  $A > B$  diperoleh dari kondisi  $G_7 + (E_7 \cdot G_6) + (E_7 \cdot E_6 \cdot G_5) + \dots + (E_7 \cdot E_6 \dots E_1 \cdot G_0)$
  - Sinyal  $A < B$  diperoleh dari kondisi sebaliknya
  - Sinyal  $A = B$  bernilai 1 jika seluruh  $E_i = 1$
- Meneruskan sinyal ke output akhir
  - EQ (Equal /  $A == B$ )
  - GT (Greater /  $A > B$ )
  - LT (Less /  $A < B$ )

### 1.3 Hasil Simulasi

#### 1. 8-bit Adder

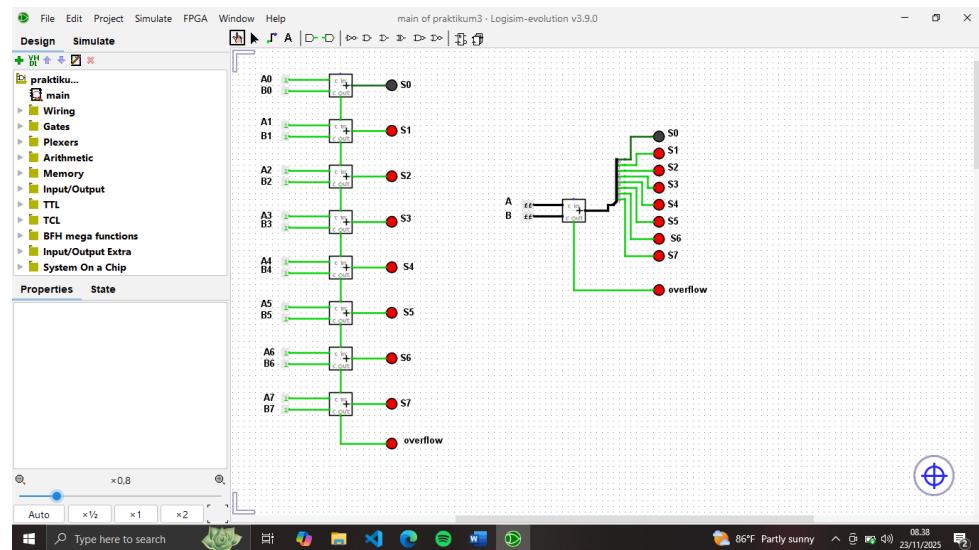
- Gerbang Logika



Rangkaian penjumlahan biner 8-bit dibuat menggunakan gerbang logika dasar atau full-adder. Proses penjumlahan dimulai dari bit terkecil ( $A_0, B_0$ ), di mana tiap bit melakukan operasi penjumlahan yang menghasilkan nilai sum dan carry-out. Sinyal sum diperoleh melalui dua tahap gerbang XOR, pertama untuk  $A \oplus B$ , kemudian hasilnya di-XOR-kan lagi dengan carry-in

(Cin) untuk menghasilkan Sum ( $A_0 \oplus B_0 \oplus Cin$ ). Sedangkan carry-out (Cout) dibentuk dari kombinasi tiga gerbang AND yang hasilnya di-OR-kan ( $Cout = (A_i \cdot B_i) + (C_i \cdot (A_i \oplus B_i))$ ). Carry-out dari setiap full adder diteruskan menjadi carry-in untuk full adder berikutnya, membentuk susunan ripple carry adder 8-bit. Hasil penjumlahan (sum) ditampilkan melalui delapan LED, dan carry terakhir dari bit ke-7 berfungsi sebagai sinyal indikator overflow.

- Aritmatika



Pada rangkaian ini terdapat dua masukan utama (A dan B), masing-masing terdiri dari 8 bit. Kedua input tersebut terhubung langsung ke modul adder, sehingga proses penjumlahan bilangan biner dilakukan secara otomatis oleh blok aritmatika. Adder akan melakukan penjumlahan bit per bit dari posisi terendah hingga tertinggi dengan mempertimbangkan carry-in di dalam blok. Hasil dari proses penjumlahan ini ditampilkan melalui 8 keluaran S0 - S7 yang menunjukkan nilai sum dari bit 0 - 7. Sedangkan carry terakhir dari hasil penjumlahan bit tertinggi ditampilkan melalui LED overflow sebagai indikator terjadinya melebihi batas hasil penjumlahan.

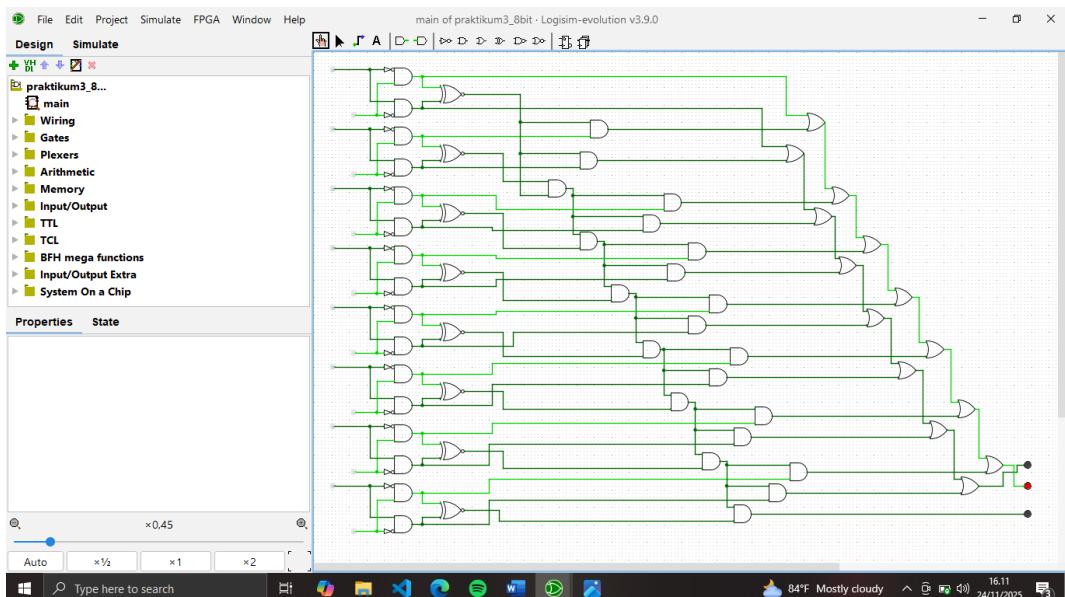
- Tabel pengujian

A	B	C-in	Sum	C-out
00000000	00000000	0	00000000	0
00001101	00000111	0	00010100	0
10101010	01010101	0	11111111	0
11111111	11111111	0	11111110	1

10011001	11001100	1	01100110	1
----------	----------	---	----------	---

Pada pengujian pertama, input A dan B masing-masing bernilai 00000000 dengan C-out bernilai 0, menandakan bahwa tidak terjadi penambahan nilai maupun kelebihan hasil. Pengujian kedua menggunakan nilai acak 00001101 dan 00000111 dengan C-in 0 menghasilkan sum 00010100 dan C-out bernilai 0, yang sesuai dengan hasil penjumlahan biner tanpa overflow. Pengujian ketiga memakai pola komplemen 10101010 dan 01010101 menghasilkan sum 11111111 dengan C-out 0, membuktikan bahwa rangkaian mampu menangani semua kombinasi bit dan tetap memberikan keluaran yang benar. Pada pengujian keempat, kedua input berada pada nilai maksimum 11111111, dan hasil penjumlahan menunjukkan 11111110 dengan C-out 1, yang menandakan total penjumlahan melebihi kapasitas 8 bit sehingga terjadi overflow. Pengujian kelima menggunakan input 10011001 dan 11001100 dengan C-in 1 menghasilkan sum 01100110 dengan C-out 1, yang menunjukkan rangkaian dapat memproses input carry-in dari sumber eksternal secara akurat.

## 2. Comparator



Dari hasil praktikum 8-bit comparator yang berfungsi untuk membandingkan dua bilangan biner 8-bit (A dan B), proses perbandingan dilakukan secara bertahap mulai dari bit yang paling signifikan (A7 dan B7) hingga bit paling rendah (A0 dan B0). Setiap pasangan bit Ai dan Bi diproses untuk menentukan apakah Ai lebih besar, lebih kecil, atau sama dengan Bi. Untuk mengetahui kesamaan bit, digunakan

logika XNOR. Sinyal kesamaan ini kemudian digunakan sebagai informasi penyebar ke tahap berikutnya.

Hasil perbandingan akan ditentukan berdasarkan tiga kondisi yang saling eksklusif, yaitu  $A > B$ ,  $A = B$ , dan  $A < B$ . Kondisi  $A > B$  terjadi jika ada bit  $A_i$  pada posisi tertentu yang lebih tinggi dari  $B_i$  dan semua bit yang lebih signifikan sama. Sedangkan kondisi  $A = B$  berarti seluruh bit  $A_i$  dan  $B_i$  sama, dan kondisi  $A < B$  terjadi jika ada bit  $A_i$  yang lebih rendah dari  $B_i$  dengan semua bit lebih signifikan sama. Berdasarkan hasil ini, output untuk kondisi  $A > B$ ,  $A < B$ , dan  $A = B$  diberikan nilai logik sesuai untuk menunjukkan hubungan antara kedua bilangan.

#### 1.4 Kesimpulan

Berdasarkan hasil praktikum, dapat disimpulkan bahwa rangkaian Adder dan Comparator memiliki peran penting dalam sistem logika digital dan pengolahan data biner. Pada rangkaian 8-bit Adder, proses penjumlahan dilakukan secara berantai menggunakan konsep full adder, di mana setiap bit menghasilkan keluaran *sum* dan *carry* yang diteruskan ke bit berikutnya. Simulasi pada Logisim menunjukkan bahwa rangkaian dapat bekerja dengan benar untuk berbagai kombinasi input, termasuk kondisi tanpa *carry*, penjumlahan dengan *carry-in*, hingga terjadinya *overflow*. Hal ini membuktikan bahwa rancangan logika yang dibangun sudah sesuai dengan teori.

Pada rangkaian 8-bit Comparator, perbandingan dua bilangan biner dilakukan mulai dari bit paling signifikan hingga bit paling rendah menggunakan sinyal kesetaraan, lebih besar, dan lebih kecil. Hasil simulasi menunjukkan bahwa sistem mampu menentukan kondisi  $A > B$ ,  $A = B$ , atau  $A < B$  secara eksklusif untuk setiap kombinasi input. Mekanisme penyebaran sinyal kesetaraan pada setiap bit berhasil memastikan perbandingan berjalan akurat.

Secara keseluruhan, praktikum ini membantu memperkuat pemahaman hubungan antara tabel kebenaran, ekspresi logika, dan implementasi praktis rangkaian digital. Melalui simulasi Logisim, mahasiswa dapat mengimplementasikan teori secara nyata dan mengamati perilaku rangkaian Adder serta Comparator secara langsung, sehingga menambah wawasan dalam perancangan sistem elektronik maupun komputasi digital.