

LAPORAN PRAKTIKUM PENGANTAR SISTEM DIGITAL

Dosen: Arif Rifai Dwiyanto, ST., MTI

TABEL KEBENARAN DAN IMPLEMENTASI K-MAP DAN IMPLEMENTASI ENCODER DAN DECODER



Disusun Oleh:

Khoirunnisa

(202410715047)

F3A6

PROGRAM STUDI INFORMATIKA FAKULTAS ILMU KOMPUTER

UNIVERSITAS BHAYANGKARA JARAKARTA RAYA

2025

[illegible]

12	1	1	0	0	0	1	1	0	0	0	1	0	0	1
13	1	1	0	1	0	1	0	0	0	0	0	1	0	1
14	1	1	1	0	0	0	1	0	0	0	0	0	0	0
15	1	1	1	1	0	0	0	0	0	0	0	0	1	1

2) Langkah 2: Menyederhanakan Menggunakan K-map

(a) Susun K-map:

AB\CD	00	01	11	10
00	0	0	0	0
01	1	1	1	0
11	1	1	1	0
10	0	0	0	0

(b) Sederhanakan ekspresi logika menggunakan metode **pairing** atau **grouping**

(i) Grouping

- Grup 1 (m4,m5,m12,m13) → **B C'**
- Grup 2 (m7,m15) → **B C D**

(ii) Gabungkan hasil:

$$F = BC' + BCD$$

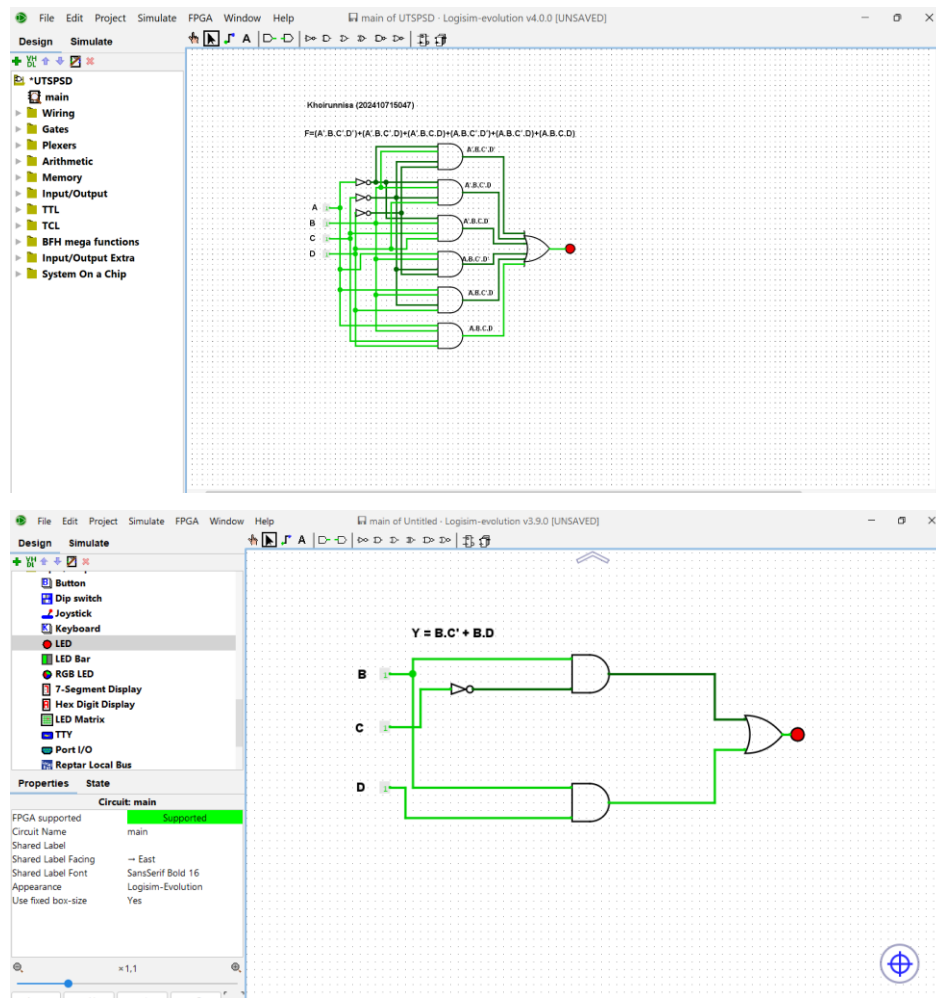
(iii) Sederhanakan aljabar:

$$F = B(C' + CD) = B(C' + D)$$

$$\text{hasil akhir } \boxed{F = B(C' + D)}$$

3) Langkah 3: Implementasi pada Logisim

- Implementasikan hasil penyederhanaan dalam bentuk rangkaian digital di Logisim.
- Hubungkan input menggunakan **Constant**, dan tambahkan **LED** pada output.
- Simulasikan rangkaian untuk memverifikasi tabel kebenaran sebelum dan sesudah disederhanakan.



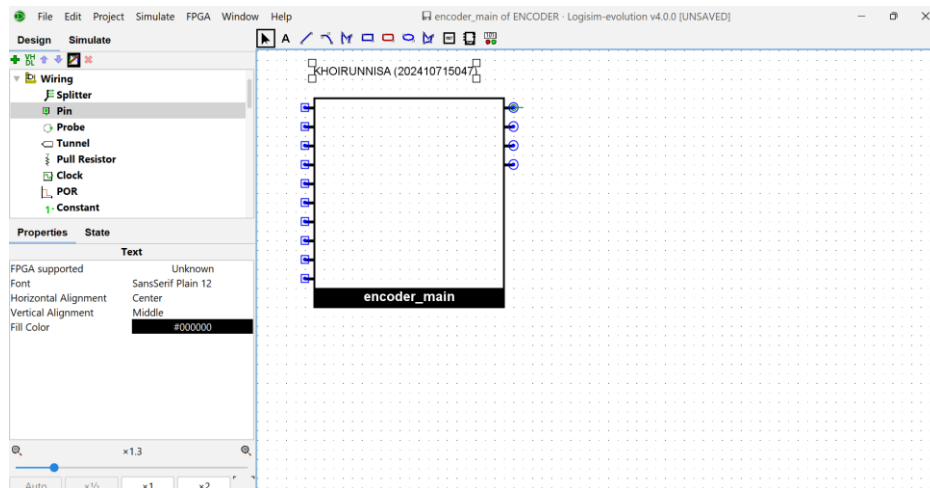
4) **Langkah 4:** perbandingan rangkaian sebelum dan sesudah di sederhanakan

Aspek	Rangkaian Awal	Rangkaian Setelah Disederhanakan
Ekspresi Logika	$(F = (A'B'C'D') + (A'BC'D) + (A'BCD) + (ABC'D') + (ABC'D) + (ABCD))$	$(F = B(C' + D))$
Jumlah Gerbang AND	6 buah	1 buah
Jumlah Gerbang OR	1 buah (menggabungkan semua suku)	1 buah
Jumlah Gerbang NOT	2 buah (untuk A' dan C')	1 buah (untuk C')
Total Komponen	10 gerbang logika	3 gerbang logika
Kebutuhan Koneksi (jalur kabel)	Kompleks dan panjang	Sederhana dan singkat
Output Simulasi	Sama (hasil identik pada semua kombinasi input)	Sama
Efisiensi	Banyak gerbang, boros komponen	Lebih efisien dan mudah dirakit

1.2.2 Encoder dan Decoder

1) Merancang 8-to-3 Encoder

Langkah 1: Membangun 8-to-3 Encoder



Langkah 2: Buat tabel kebenaran untuk Encoder 8-to-3

Bilangan Oktal	A	B	C
	2^2	2^1	2^0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Input								Output		
I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	A	B	C
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0

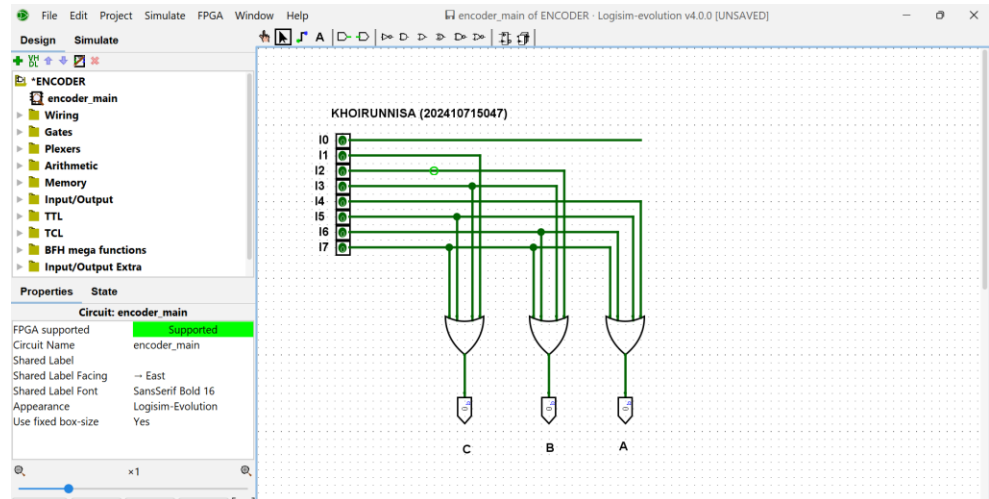
0	0	0	0	0	0	0	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---

$$A = I_4 + I_5 + I_6 + I_7$$

$$B = I_2 + I_3 + I_6 = I_7$$

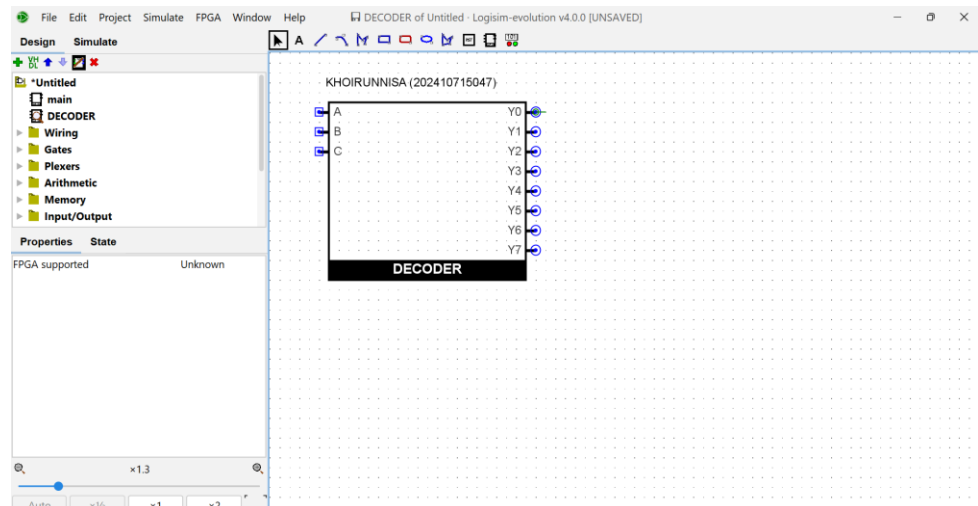
$$C = I_1 + I_3 + I_5 + I_7$$

Langkah 3: Implementasikan rangkaian menggunakan Logism



2) Merancang 3-to-8 Decoder

Langkah 1: Membangun 3-to-8 Decoder

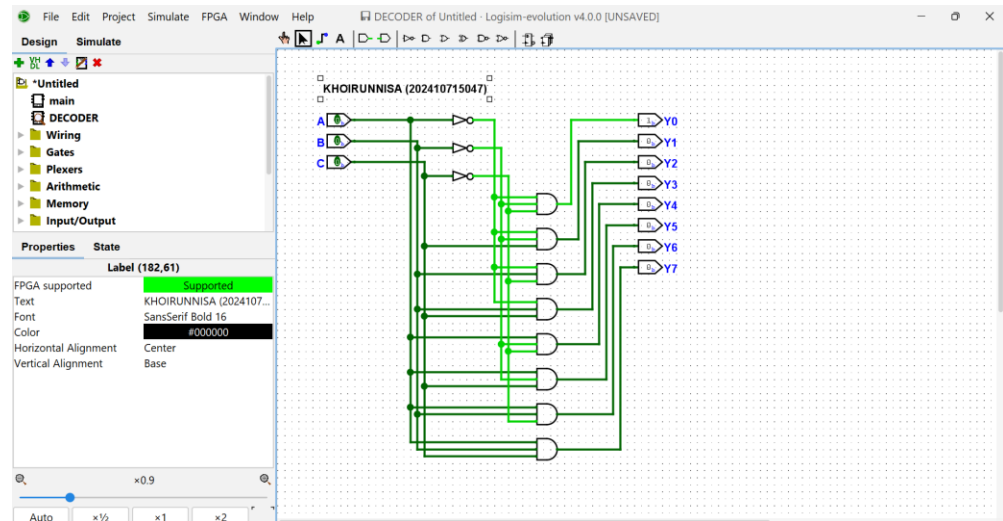


Langkah 2: Buat tabel kebenaran untuk Decoder 3-to-8

Input			Output							
A	B	C	Y ₀	Y ₁	Y ₂	Y ₃	Y ₄	Y ₅	Y ₆	Y ₇
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0

1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

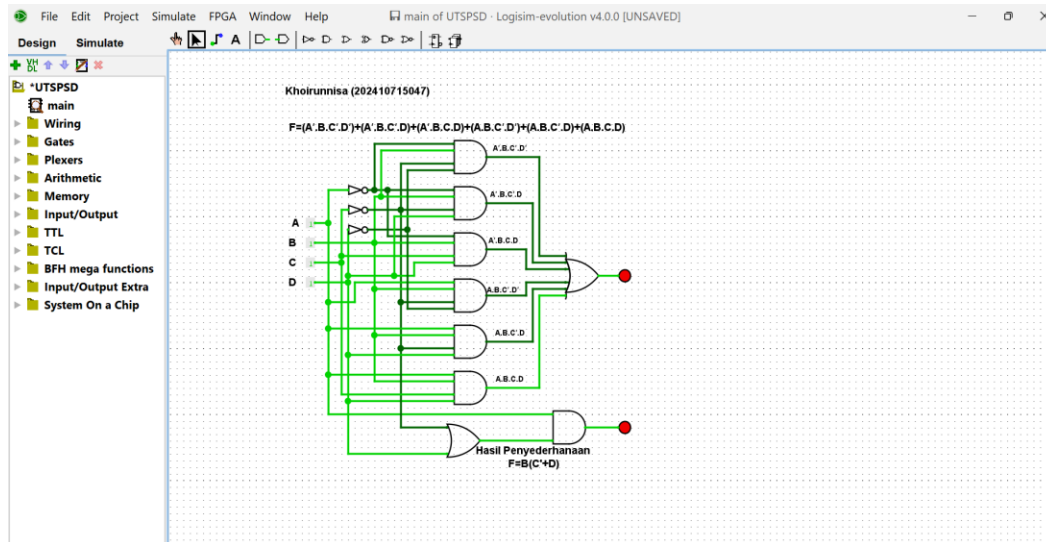
Langkah 3: Implementasikan rangkaian menggunakan Logisim



BAB II

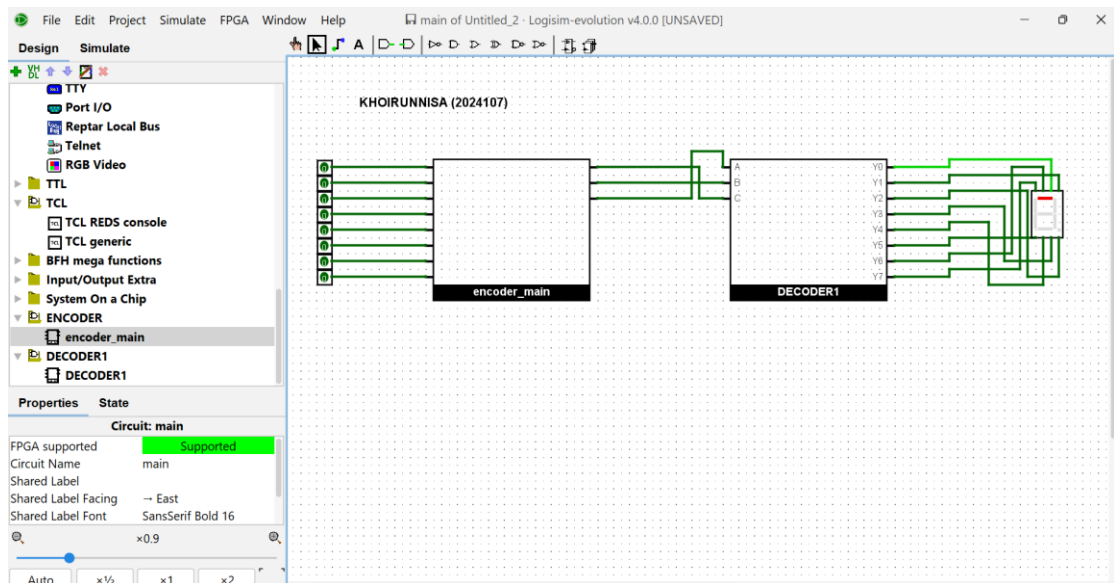
HASIL SIMULASI

2.1 Hasil Simulasi K-Map



Dalam simulasi ini, setiap kombinasi logika A, B, C, dan D dipilah ke dalam beberapa kelompok minterm yang diuji lewat gerbang AND. Setiap kelompok cuma aktif kalau seluruh syarat variabelnya klop 100%. Output dari semua minterm itu kemudian dikumpulkan lewat satu gerbang OR, jadi lampu indikator bakal nyala begitu ada satu saja kelompok yang memenuhi aturan logikanya. Hasil uji menunjukkan rangkaian berjalan persis seperti yang seharusnya. Kelompok minterm yang dipakai dalam rangkaian $(A'.B.C'.D') + (A'.B.C'.D) + (A'.B.C.D) + (A.B.C'.D') + (A.B.C'.D) + (A.B.C.D)$. Bagian penyederhanaan $P(B,C,D)$ juga bereaksi sesuai pola logikanya dan konsisten dengan desain rangkaian.

2.2 Hasil simulasi Encoder dan Decoder



Dalam simulasi ini, encoder langsung mengubah setiap input yang diaktifkan menjadi kode biner empat bit. Kode tersebut dikirim ke decoder, yang kemudian menyalakan output yang sesuai sambil mematikan output lain. Alurnya berjalan mulus—saat satu input aktif, hanya satu LED yang menyala dan yang lainnya langsung nonaktif. Ini nunjukkin kalau hubungan antara encoder dan decoder sudah pas, dan keseluruhan rangkaian bekerja persis seperti yang diharapkan.

BAB III

KESIMPULAN

Berdasarkan hasil praktikum, dapat dinyatakan bahwa penggunaan Karnaugh Map (K-Map) sangat membantu dalam melakukan penyederhanaan fungsi logika yang berasal dari tabel kebenaran. Dengan metode pengelompokan pada K-Map, rangkaian logika yang awalnya rumit dapat dipadatkan menjadi bentuk yang lebih sederhana, sehingga komponen yang diperlukan menjadi lebih sedikit namun tetap menghasilkan output yang sama. Percobaan membuktikan bahwa penyederhanaan ini meningkatkan efisiensi rangkaian tanpa mengubah perilaku logikanya.

Pada bagian encoder, rangkaian bekerja dengan mengonversi satu input aktif menjadi kode biner sehingga jalur data lebih ringkas dan mudah diolah. Sebaliknya, decoder mengubah kode biner tersebut kembali menjadi satu keluaran spesifik. Hasil simulasi menunjukkan bahwa kedua rangkaian menjalankan fungsinya dengan tepat: encoder menghasilkan kombinasi biner yang benar, dan decoder menyalakan output sesuai kode yang diterimanya.

Secara keseluruhan, praktikum ini memperlihatkan bahwa pemahaman mengenai K-Map, encoder, dan decoder sangat penting dalam desain sistem digital. Ketiga konsep tersebut berkontribusi besar dalam membuat rangkaian menjadi lebih efisien, terstruktur, dan mampu bekerja sesuai kebutuhan.