



# VISI

*Dr. Linda Hutauruk, MSc*

Volume 16

Nomor 3

Oktober 2008

Perancangan Coder dan Decoder Sebagai Alat Bantu Deteksi dan Koreksi Kesalahan Penerimaan Data Pada Komunikasi Data

*Sindak Hutauruk dan Liswanto Pasaribu*

Pelestarian Fungsi Hutan dan Lingkungan Hidup  
Dalam Perspektif Hukum Lingkungan

*August P. Silaen*

Analisis Pengaruh Aspek Moneter dan Fiskal  
Terhadap Pertumbuhan Ekonomi Indonesia

*Angandrowa Gulo*

Analisis Pengaruh Penurunan Daya Motor Akibat Penambahan Aksesoris Air Condition Pada Mobil Jenis Toyota Avanza

*Parulian Siagian*

Rancangan Model Kompetensi Pejabat Struktural untuk Diaplikasikan dalam Rekrutmen, Seleksi, Pengembangan dan Perencanaan Suksesi pada Fakultas Psikologi UHN

*Freddy Butarbutar*

Studi Eksperimental Karakteristik Boundary Layer Turbulen di Sekitar Alur Bujur-sangkar Tunggal Pada Pelat Datar

*Antoni R. Malau<sup>1</sup>, Sutardi<sup>2</sup>*

Verba Tindak Tutur (Speech Act Verbs) Bahasa Batak Toba

*Selviana Napitupulu*

Hubungan Antara Religiusitas dengan Penalaran Moral pada Remaja Akhir Suatu Studi Mengenai Religiusitas dan Penalaran Moral serta Hubungannya pada Mahasiswa Kristen Protestan Universitas Padjadjaran Usia Remaja Akhir

*Asina Christina Rosito Pasaribu*

Majalah Ilmiah

Universitas HKBP Nommensen

**PERANCANGAN CODER DAN DECODER  
SEBAGAI ALAT BANTU DETEKSI DAN KOREKSI  
KESALAHAN PENERIMAAN DATA PADA KOMUNIKASI DATA**

*Sindak Hutauruk dan Liswanto Pasaribu*

**ABSTRACT**

*This Research is aim to design Detector and data Correction for 6 bits will by using simplified Karnaugh Map and Quine Mc Cluskey table method. The result be implemented in Programmable Array Logic (PAL). Then this program be simulated by using Electronics Workbench software. The role of the Detector is very important to detect and correct the error in receiving the data.*

**Key Words:** *Detection and Correction Data, Karnaugh Map, Quine Mc Cluskey table, Electronics Workbench(EWB), Programmable Array Logic (PAL).*

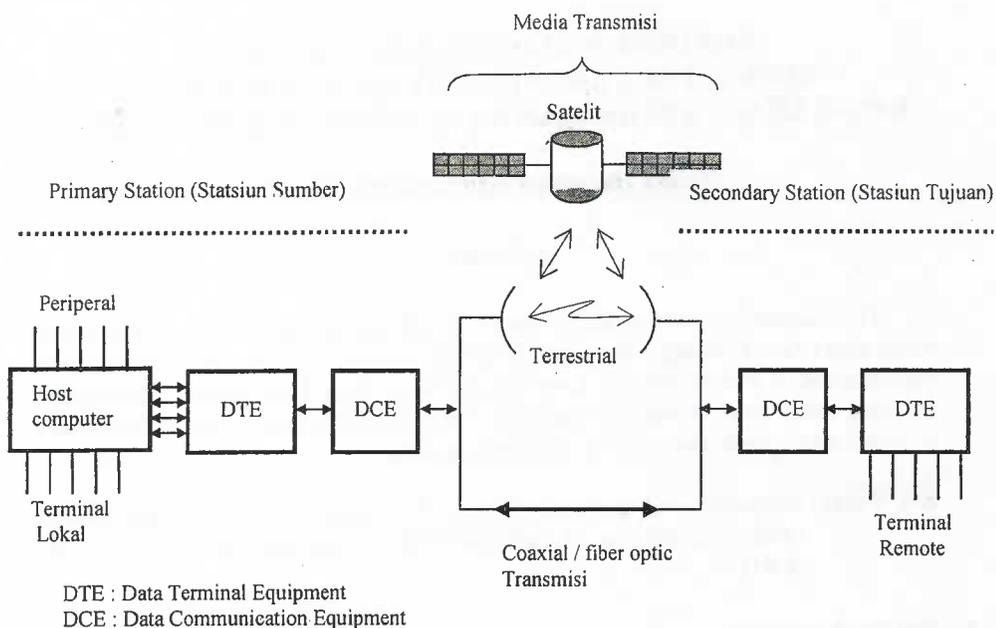
## 1. PENDAHULUAN

Keberhasilan penyampaian informasi dari pengirim (*transmitter*) ke penerima (*receiver*) merupakan salah satu hal yang sangat penting dalam menentukan keandalan sebuah sistem komunikasi. Keandalan sebuah sistem komunikasi data bukan hanya diukur dari kecepatan transfer data atau yang disebut dengan bit rate dalam satuan bps (*bit per second*) tetapi juga keberhasilan sampainya data yang dikirim sampai ke penerima dengan jelas dan benar. Komunikasi data dikatakan berhasil apabila penerima dapat menerima data dengan jelas dan benar serta dapat dimengerti oleh penerima (tujuan).

Dalam penyampaian informasi baik berupa suara (*voice*) maupun data selalu menggunakan media transmisi. Media transmisi yang digunakan dapat berupa kabel (*coaxial, fiber optic*), udara (*terrestrial*), dan satelit seperti yang terlihat pada gambar 1.

Dalam proses komunikasi data, kemungkinan kesalahan data yang diterima oleh penerima dapat terjadi, sehingga sering sekali data tersebut harus dikirim ulang oleh pengirim ke penerima sampai data tersebut diterima dengan benar sesuai dengan data yang dikirim oleh pengirim, yang mengakibatkan waktu penyampaian data secara keseluruhan mengalami keterlambatan. Kesalahan data yang diterima oleh penerima dapat terjadi oleh karena beberapa kemungkinan, diantaranya : 1) Kesalahan sinkronisasi data atau clock pada pengirim maupun penerima, dan 2) Terjadinya interferensi dari kanal-kanal yang berdekatan pada media transmisi.

Agar proses pengiriman data berlangsung dengan cepat, maka pada penerima harus dapat mendeteksi dan mengoreksi data yang salah tersebut sehingga tidak dibutuhkan transfer ulang oleh pengirim terhadap data yang salah diterima oleh penerima



Gbr. 1. Blok Diagram Jaringan Komunikasi Data Sederhana

Data yang dikirim oleh pengirim (*source*) ke penerima (*destination*) akan melalui proses sinkronisasi dan modulasi yang kemudian dikirim melalui media transmisi. Kemungkinan terjadinya kesalahan data yang diterima oleh penerima dapat terjadi karena kesalahan proses sinkronisasi disisi pengirim atau terjadinya interferensi antar kanal pada media transmisi atau sebab-sebab lainnya.

Kesalahan data yang diterima oleh penerima dapat dideteksi dan dikoreksi oleh penerima dengan menggunakan alat *Error Detection* dan *Error Correction*. Alat tersebut akan mendeteksi adanya data yang salah atau tidak sama dengan yang dikirimkan oleh pengirim yang kemudian akan melakukan koreksi atas kesalahan tersebut sehingga data yang diterima sama persis dengan yang dikirim.

Pada perancangan ini, kesalahan bit yang diterima oleh penerima dapat dideteksi baik kesalahan itu terjadi pada 1 bit dalam satu word maupun 2 bit dalam 2 word, akan tetapi alat ini hanya bisa mengkoreksi kesalahan yang terjadi 1 bit pada 1 word. Pada perancangan ini jumlah bit yang digunakan dalam satu word adalah 6 bit. Hasil rancangan diimplementasikan dengan mengacu kepada struktur gerbang logik (*logic gate*) PAL (*Programmable Array Logic*), dan kemudian hasil rancangan tersebut disimulasikan dengan perangkat lunak *Electronics Workbench* (EWB).

Penyederhanaan suatu rancangan yang menggunakan komponen-komponen logik dapat dilakukan dengan beberapa metoda, diantaranya dengan menggunakan rumus-rumus Aljabar Boole, Peta Karnaugh (*Karnaugh Map*), atau dengan tabulasi Quine Mc Cluskey.

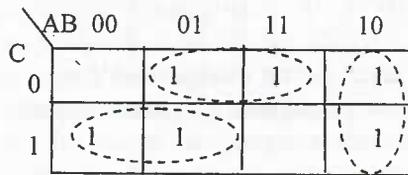
Dalam perancangan ini, penyederhanaan dilakukan dengan metoda peta Karnaugh dan dengan Tabulasi Quine Mc Cluskey. Implementasi dari hasil rancangan dibuat dengan menggunakan struktur logik IC PAL dan disimulasikan dengan perangkat lunak EWB.

Penyederhanaan dengan menggunakan peta karnaugh mudah dilakukan, sebagai contoh : Pada satu ruangan terdapat 3 buah pintu dan satu buah lampu. Apabila satu atau dua buah pintu terbuka maka lampu tersebut akan menyala. Misalkan 3 buah pintu tersebut diwakili oleh variabel A,B,C dan lampu diwakili oleh variabel Z. Misalkan saja alat tersebut dinamakan kendali lampu ruangan.

Tabel 1. Fungsi Alat

INPUT			OUTPUT
A	B	C	Z
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

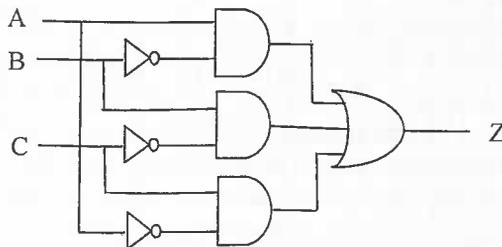
PETA KARNAUGH



Gbr. 2. Penyederhanaan dgn Peta Karnaugh

$$Z = AB' + BC' + A'C$$

Tabel kebenaran (*truth table*) dari alat tersebut seperti pada tabel 1. Pada variabel masukan (input) A,B, dan C berharga '0' bila pintu tertutup (*normally closed*) dan berharga '1' bila pintu terbuka. Keluaran Z berharga '0' bila lampu mati dan berharga '1' bila lampu menyala. Dengan memasukkan semua kondisi bit pada tabel kebenaran sistem ke dalam peta Karnaugh (gambar 2), maka dapat dilakukan penggabungan bit-bit yang berharga '1' sehingga hasil akhir yang diperoleh adalah  $Z = AB' + BC' + A'C$ . Gambar rangkaian yang diperoleh seperti pada gambar 3.



Gambar 3. Rangkaian Kendali Lampu Ruangan

Untuk fungsi-fungsi dengan cacah variable (peubah) yang lebih besar, terlebih untuk sistem dengan keluaran ganda (*MIMO, Multiple Input Multiple Output*) dimana beberapa keluaran harus disederhanakan secara serentak, pemakaian peta Karnaugh menjadi sangat sulit. Disamping itu, bila suatu sel dalam peta Karnaugh mempunyai kemungkinan penggabungan dengan beberapa sel yang

berdekatan, sering kita tak dapat segera menentukan penggabungan mana yang terbaik. Kesulitan-kesulitan ini dapat diatasi dengan metoda tabulasi Quine McCluskey.

Penyederhanaan dengan metoda ini sangat sistematis dan cocok untuk penyederhanaan dengan memakai komputer digital. Tidak ada batasan untuk jumlah variabel (peubah), juga dapat dipakai untuk sistem dengan keluaran ganda. Tetapi fungsi yang akan disederhanakan dengan metoda tabulasi haruslah dalam bentuk jumlah-perkalian standar (*standard sum of products*) atau disebut juga dengan istilah sukumin (*minterm*). Bila fungsi itu masih dalam bentuk perkalian-jumlah (*standard product of sum*) atau disebut dengan istilah sukumaks (*maxterm*), maka terlebih dahulu harus diubah ke bentuk jumlah-perkalian.

Pada metoda Tabulasi Qiune Mc Cluskey, suku "tak peduli" (*don't care*) yang disimbolkan dengan "x" dapat diperlakukan sebagai bit 1 dan dapat pula sebagai bit 0. Dalam penyederhanaan, mula-mula kita menganggap setiap suku don't care itu sebagai bit 1. Terakhir, setelah diketahui suatu suku abaikan itu tidak diperlukan dalam memperoleh fungsi minimum, kita menganggapnya sebagai bit 0 dan mengabaikannya. Dalam metoda ini, selama proses penentuan penyusun utama, kita menganggap semua suku abaikan itu berharga bit 1. Tetapi karena dia tidak harus diliput, suku-suku tersebut tidak kita sertakan dalam tabel pemilihan suku penyusun inti.

**Contoh:** sebuah persamaan fungsi sukumin sebagai berikut :

$$f(v,w,x,y) = \sum m (2,3,7,9,11,13) + \sum d (1,10,15)$$

dengan  $d_i, i = 1, 10, 15$ , adalah suku-suku don't care.

Penentuan penyusun utama dapat ditulis seperti pada Tabel 2. Dapat dilihat pada tabel ini, bahwa semua penyusun dalam kolom-0 dan kolom-1 sudah bergabung di kolom-3 yang menghasilkan 4 penyusun utama.

Tabel 2. Penentuan Penyusun Utama

Kolom - 0			Kolom - 1			Kolom - 2	
Desimal	Biner		Desimal	Biner		Desimal	Biner
1	0001	√	1,3	00-1	√	A* 1, 3, 9, 11	-0-1
2	0010	√	1,9	-001	√	B* 2, 3, 10, 11	-01-
3	0011	√	2,3	001-	√	C* 3, 7, 11, 15	--11
9	1001	√	2,10	-010	√	D* 9, 11, 13, 15	1--1
10	1010	√	3,7	0-11	√		
7	0111	√	3,11	-011	√		
11	1011	√	9,11	10-1	√		
13	1101	√	9,13	1-01	√		
15	1111	√	10,11	101-	√		
			7,15	-111	√		
			11,15	1-11	√		
			13,15	11-1	√		

Pemilihan penyusun minimum dibuat seperti biasa, tetapi suku abaikan (*don't care*) tidak dicantumkan di dalamnya, seperti pada tabel 3. Ini karena suku abaikan tidak harus disertakan/diwakili dalam realisasi fungsinya.

Tabel 3. Pemilihan Penyusun Minimum

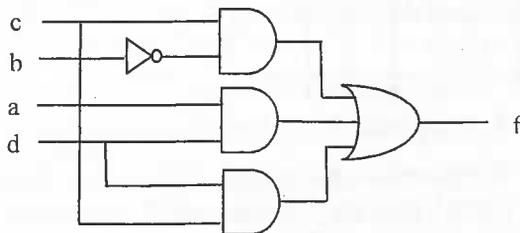
Penyusun Utama		Sukumin					
		2	3	7	9	11	13
A	1,3,9,11		X		X	X	
* B	2,3,10,11	X	X			X	X
* C	3,7,11,15		X	X		X	
* D	9,11,13,15				X	X	X
		√	√	√	√	√	√

Dari tabel 3. diperoleh hasil penyederhanaan sebagai berikut:

$$f(v,w,x,y) = B + C + D$$

$$= b'c + cd + ad$$

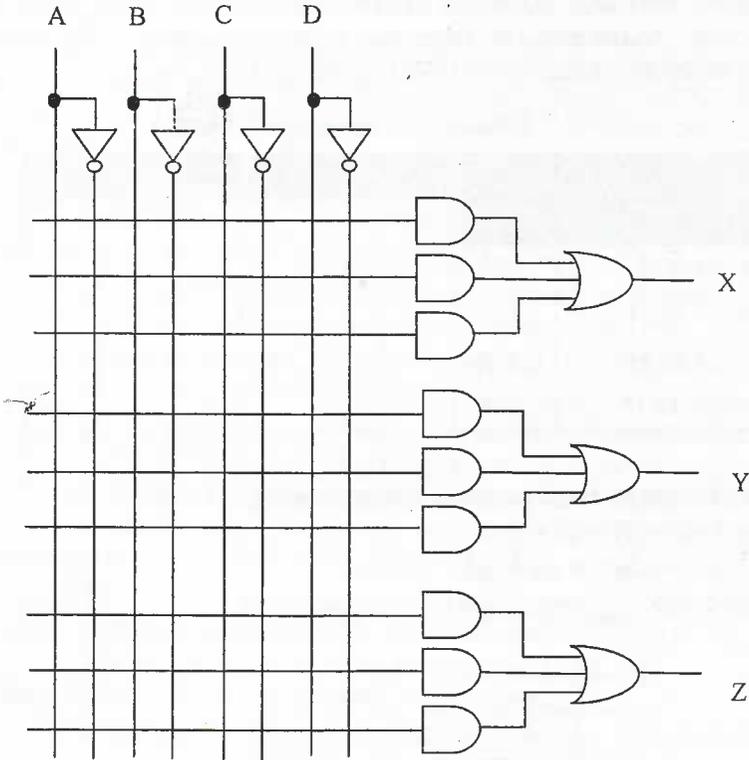
dan hasil rangkaian logiknya seperti pada gambar 4.



Gambar 4. Realisasi Rangkaian Logik

Programmable Array Logic adalah sebuah komponen chip (IC) yang berisi gerbang-gerbang logik yang dapat diprogram sesuai dengan rancangan yang dibuat. Rangkaian logik dari PAL terdiri dari gerbang OR yang inputnya dihubungkan dengan beberapa gerbang AND seperti pada gambar 5.

Kelebihan dari PAL dibandingkan dengan komponen programmable lainnya adalah bahwa outputnya dapat *difeedback* (diumpanbalikkan) ke inputnya sehingga dapat lebih menyederhanakan rangkaian yang relatif kompleks. Dengan merealisasikan hasil rancangan pada PAL, maka ukuran hasil rancangan dapat diminimalkan, juga kecepatan dapat dimaksimalkan karena time delay yang terjadi pada setiap komponen IC dapat diperkecil.



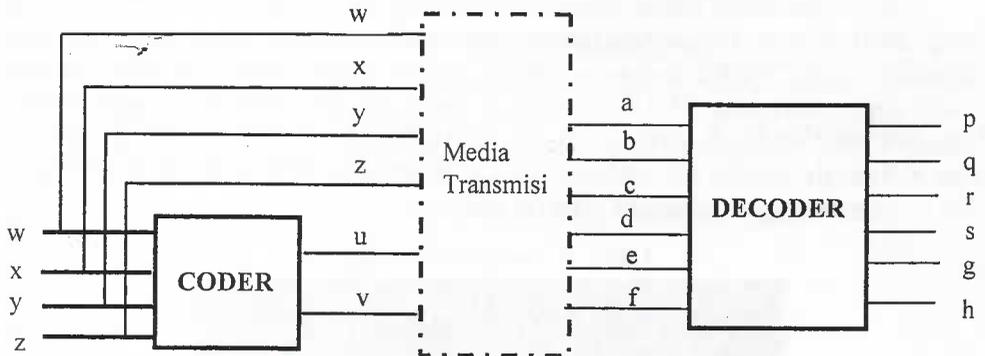
Gbr. 5. Rangkaian PAL (*Programmable Array Logic*)

Electronics Workbench merupakan perangkat lunak (*software*) dari interactive Image Tecnology Ltd. Software EWB dapat diperoleh dengan men-download EWB dari situs [www.electronicsworkbench.com](http://www.electronicsworkbench.com). EWB (*Electronics Workbench*) adalah salah satu jenis software elektronika yang digunakan untuk melakukan simulasi terhadap cara kerja dari suatu rangkaian listrik maupun digital. Perlunya simulasi adalah untuk menguji apakah rangkaian tersebut dapat berjalan dengan baik dan sesuai dengan pendekatan teori yang digunakan, tanpa harus membuat rangkaian itu secara nyata. Simulasi yang dilakukan dengan menggunakan EWB adalah simulasi yang menghasilkan keluaran yang ideal, artinya keluaran tidak terpengaruh oleh faktor-faktor ketidak idealan seperti gangguan (*noise*) dan *time delay*.

Penelitian ini dilakukan dengan tujuan : 1) mencari solusi agar kesalahan data yang diterima oleh penerima tidak perlu di retransmisi oleh pengirim sehingga waktu pengiriman data dapat berlangsung dengan cepat dan benar, 2) melakukan implementasi hasil rancangan dalam sebuah IC PAL dengan menggunakan simulasi perangkat lunak EWB dan 3) mengaplikasikan teori-teori penyederhanaan rangkaian logik yang diimplementasikan pada sebuah alat deteksi dan koreksi kesalahan bit data yang diterima dalam komunikasi data.

## 2. METODOLOGI PENELITIAN

Alat yang dirancang terdiri dari dua bagian utama, yaitu Coder pada sisi pengirim dan Decoder pada sisi penerima. Coder akan mengkodekan bit-bit data yang akan dikirim dan hasil pengkodean ini akan ditambahkan ke dalam susunan bit data yang selanjutnya ditransmisikan melalui media transmisi. Pada sisi penerima, bit-bit data yang telah ditambahkan kode dimasukkan ke dalam Decoder untuk mengetahui ada tidaknya kesalahan bit yang diterima oleh penerima yaitu dengan cara memeriksa dan membandingkan susunan bit yang diterima dengan yang dikirim oleh pengirim. Gambar 6. adalah gambar blok diagram alat yang dirancang untuk data 6 bit.



Gambar 6. Blok Diagram Rancangan Alat

### 2.1. Tahap-tahap Perancangan

Tahap-tahap perancangan yang dilakukan adalah sebagai berikut :

1. Perancangan Coder, dilakukan dengan tahapan berikut,
  - a. Menetapkan ketentuan syarat pembentukan bit kode sebagai keluaran Coder.
  - b. Melakukan perancangan Coder dengan menggunakan metoda penyederhanaan peta Karnaugh.
  - c. Merealisasikan hasil rancangan ke dalam struktur gerbang PAL
  - d. Melakukan simulasi dengan menggunakan perangkat lunak EWB.
2. Perancangan Decoder, dilakukan dengan tahapan berikut :
  - a. Membuat tabulasi keluaran Decoder dengan terlebih dahulu menetapkan ketentuan bit kode keluaran sebagai indikator ada tidaknya kesalahan bit yang diterima.
  - b. Melakukan perancangan Coder dengan menggunakan metoda penyederhanaan peta Karnaugh dan metoda Tabulasi Quine Mc Cluskey.
  - c. Merealisasikan hasil rancangan ke dalam struktur gerbang PAL
  - d. Melakukan simulasi dengan menggunakan perangkat lunak EWB.

#### 2.1.1. Perancangan Coder

Pada umumnya jumlah bit informasi yang digunakan tergantung dari jenis kode yang digunakan, misalnya ASCII Code menggunakan 8 bit word yang sudah termasuk bit paritas, atau kode EBCDIC yang menggunakan 8 bit tanpa bit paritas.

Pada perancangan ini, jumlah bit informasi yang digunakan adalah 6 bit word (w,x,y,z,u,v) yang terdiri dari 4 bit informasi (w,x,y,z) dan 2 bit untuk kode (u,v), hal ini dilakukan untuk tidak terlalu panjang dalam perancangannya, akan tetapi sudah cukup untuk jelas dan dapat diterapkan untuk informasi yang jumlah bitnya lebih dari 6 bit. Pada perancangan ini, 4 bit informasi digunakan akan ditambah dengan 2 bit kode sehingga membentuk 6 bit word. Bit ke 5 dibentuk oleh bit 1,3 dan 4 dengan ketentuan ke 4 bit ini (bit 1,3,4,5) harus mempunyai bit '1' ganjil. Bit 6 dibentuk oleh bit 2,3, dan 4 dengan ketentuan ke 4 bit ini (bit 2,3,4,6) juga harus mempunyai bit '1' ganjil, sehingga 6 bit word ini akan mempunyai kemungkinan enam belas susunan word seperti pada tabel 4.

Ada enam belas kemungkinan susunan 4 bit informasi ditambah 2 bit kode yang dikirimkan ke tujuan/penerima melalui media transmisi dimana dua bit kode dihasilkan oleh CODER dengan ketentuan seperti yang disebut pada butir 1 di atas, maka dapat dirancang CODER tersebut. Perancangan CODER tersebut dengan menggunakan gerbang-gerbang logika melalui penyederhanaan dengan metoda peta Karnaugh. Empat bit informasi (w,x,y,z) sebagai input Coder dan bit kelima dan ke enam sebagai keluaran Coder ( u dan v ).

Tabel 4. Susunan Bit Word

Bit Word					
Bit Informasi				Bit Kode	
1	2	3	4	5	6
w	x	y	z	u	v
0	0	0	0	1	1
0	0	0	1	0	0
0	0	1	0	0	0
0	0	1	1	1	1
0	1	0	0	1	0
0	1	0	1	0	1
0	1	1	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	0	1	1	0
1	0	1	0	1	0
1	0	1	1	0	1
1	1	0	0	0	0
1	1	0	1	1	1
1	1	1	0	1	1
1	1	1	1	0	0

Fungsi keluaran Coder untuk u pada tabel 4. disederhanakan dengan menggunakan metoda peta Karnaugh yang hasilnya seperti pada tabel 5.

Tabel 5. Fungsi keluaran Coder untuk u

wx yz	00	01	11	10
00	1 1			
01			1 1	
11	1 1			
10			1 1	

Dari tabel 5. diperoleh persamaan keluaran untuk u, yaitu :

$$u = w'y'z' + w'yz + wy'z + wyz' \dots\dots\dots 1.$$

Fungsi keluaran Coder untuk v pada tabel 6. disederhanakan dengan menggunakan metoda peta Karnaugh yang hasilnya seperti pada tabel 6.

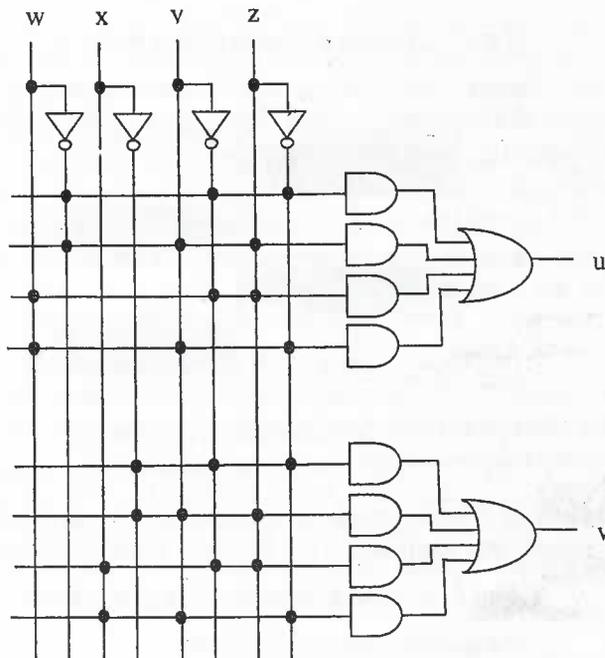
Tabel 6. Fungsi keluaran Coder untuk v

wx yz	00	01	11	10
00	1			1
01		1 1		
11	1			1
10		1 1		

Dari tabel 6. diperoleh persamaan keluaran untuk v, yaitu :

$$v = x'y'z' + x'yz + xy'z + xyz' \dots\dots\dots 2.$$

Keluaran u (persamaan 1.) dan keluaran v (persamaan 2.) dari Coder direalisasikan dengan menggunakan PAL seperti pada gambar 7.



Gambar 7. Realisasi rangkaian Coder pada PAL

**2.1.2. Perancangan Decoder**

Pada sisi penerima bit word yang diterima harus dideteksi kemungkinan kesalahan bit yang diterima dan dikoreksi apabila memang terdapat kesalahan bit, sehingga data yang diterima benar-benar sesuai dengan aslinya ( $p = w, q = x, r = y, s = z$ ). Deteksi dan koreksi kesalahan bit ini dilakukan oleh sebuah Decoder. Perancangan Decoder ini dengan menggunakan metoda penyederhanaan peta Karnaugh dan tabulasi Quine Mc Cluskey yang diimplementasikan menggunakan PAL.

Kesalahan mungkin terjadi selama transmisi data, sehingga data yang diterima berbeda dengan yang dikirim. Perancangan alat ini akan mendeteksi dan mengoreksi kesalahan satu bit yang terjadi pada 6-bit data yang diterima. Kesalahan yang terjadi adalah perubahan pada bit 0 menjadi 1 atau bit 1 menjadi 0. (deteksi dan koreksi tidak dapat dilakukan dengan benar bila terjadi lebih dari 1 bit kesalahan). Contoh : Jika 000011 ditransmisikan, maka diperoleh : 000011 (tidak ada kesalahan), 100011 (kesalahan terjadi pada bit pertama), 010011 (kesalahan terjadi pada bit ke dua), 001011, 000111, 000001, dan 000010 .

Masukan Decoder adalah yang diterima dari media transmisi yaitu a, b, c, d, e, dan f, dan keluarannya adalah p, q, r, s, g, dan h. Bit g dan h merupakan indikator ada tidaknya kesalahan bit yang diterima. Ada tiga kemungkinan keluaran yang diperoleh, yaitu :

a. Tidak terjadi kesalahan bit yang diterima dengan yang dikirim, artinya  $wxyz = pqrs$  maka  $g = '1'$  dan  $h = '1'$ .

Contoh :  $abcdef = wxyzuv = 000011$  maka  $pqrsg h = 000011$

b. Terjadi kesalahan satu bit dalam dua word data, maka  $g = '0'$  dan  $h = '1'$ .

Contoh : data word yang diterima pada masukan Decoder adalah abcdef = 000001, dan dari 16 (enam belas) kemungkinan word yang ada (tabel 4.), terdapat 2 (dua) data word yang berbeda 1 (satu) bit kesalahan yaitu wxyzuv = 000011 dan 100001. Dalam hal ini terdapat satu bit yang berbeda pada 2 (dua) data informasi (wxyz) dalam data word tersebut. Bit yang berbeda ini dituliskan sebagai don't care sehingga data yang diterima sebagai keluaran Decoder adalah pqrs = X000, dan untuk kondisi ini g = '0' dan h = '1'.

c. Terjadi kesalahan 2(dua) bit dalam dua word data, maka g = '1' dan h = '0'.

Contoh : data word yang diterima pada masukan Decoder adalah abcdef = 000000, dan dari 16 kemungkinan word yang ada (tabel 3.1.), terdapat 2 (dua) data word yang berbeda satu bit kesalahan pada pengirim yaitu wxyzuv = 000100 dan 001000. Dalam hal ini terdapat dua bit yang berbeda pada 2 (dua) data informasi (wxyz) dalam data word tersebut. Bit yang berbeda ini dituliskan sebagai don't care sehingga data yang diterima sebagai keluaran Decoder adalah pqrs = 00XX, dan untuk kondisi ini g = '1' dan h = '0'.

Pada perancangan Decoder ini, tidak terdapat keluaran yang mempunyai 1 bit kesalahan hanya dalam 1 word saja. Untuk selanjutnya kemungkinan seluruh keluaran *Error Decoder* dapat dilihat pada tabel 7.

Tabel 7. Error Decoder

Nu	a	b	c	d	e	f	p	q	r	s	g	h
1	0	0	0	0	0	0	0	0	x	x	1	0
2	0	0	0	0	0	1	x	0	0	0	0	1
3	0	0	0	0	1	0	0	x	0	0	0	1
4	0	0	0	0	1	1	0	0	0	0	1	1
5	0	0	0	1	0	0	0	0	0	1	1	1
6	0	0	0	1	0	1	0	x	0	1	0	1
7	0	0	0	1	1	0	x	0	0	1	0	1
8	0	0	0	1	1	1	0	0	x	x	1	0
9	0	0	1	0	0	0	0	0	1	0	1	1
10	0	0	1	0	0	1	0	x	1	0	0	1
11	0	0	1	0	1	0	x	0	1	0	0	1
12	0	0	1	0	1	1	0	0	x	x	1	0
13	0	0	1	1	0	0	0	0	x	x	1	0
14	0	0	1	1	0	1	x	0	1	1	0	1
15	0	0	1	1	1	0	0	x	1	1	0	1
16	0	0	1	1	1	1	0	0	1	1	1	1
17	0	1	0	0	0	0	x	1	0	0	0	1
18	0	1	0	0	0	1	0	1	x	x	1	0
19	0	1	0	0	1	0	0	1	0	0	1	1
20	0	1	0	0	1	1	0	x	0	0	0	1
21	0	1	0	1	0	0	0	x	0	1	0	1
22	0	1	0	1	0	1	0	1	0	1	1	1
23	0	1	0	1	1	0	0	1	x	x	1	0
24	0	1	0	1	1	1	x	1	0	1	0	1

Tabel 7. Error Decoder (lanjutan)

Nu	a	b	c	d	e	f	p	q	r	s	g	h
25	0	1	1	0	0	0	0	x	1	0	0	1
26	0	1	1	0	0	1	0	1	1	0	1	1
27	0	1	1	0	1	0	0	1	x	x	1	0
28	0	1	1	0	1	1	x	1	1	0	0	1
29	0	1	1	1	0	0	x	1	1	1	0	1
30	0	1	1	1	0	1	0	1	x	x	1	0
31	0	1	1	1	1	0	0	1	1	1	1	1
32	0	1	1	1	1	1	0	x	1	1	0	1
33	1	0	0	0	0	0	1	x	0	0	0	1
34	1	0	0	0	0	1	1	0	0	0	1	1
35	1	0	0	0	1	0	1	0	x	x	1	0
36	1	0	0	0	1	1	x	0	0	0	0	1
37	1	0	0	1	0	0	x	0	0	1	0	1
38	1	0	0	1	0	1	1	0	x	x	1	0
39	1	0	0	1	1	0	1	0	0	1	1	1
40	1	0	0	1	1	1	1	x	0	1	0	1
41	1	0	1	0	0	0	x	0	1	0	0	1
42	1	0	1	0	0	1	1	0	x	x	1	0
43	1	0	1	0	1	0	1	0	1	0	1	1
44	1	0	1	0	1	1	1	x	1	0	0	1
45	1	0	1	1	0	0	1	x	1	1	0	1
46	1	0	1	1	0	1	1	0	1	1	1	1
47	1	0	1	1	1	0	1	0	x	x	1	0
48	1	0	1	1	1	1	x	0	1	1	0	1
49	1	1	0	0	0	0	1	1	0	0	1	1
50	1	1	0	0	0	1	1	x	0	0	0	1
51	1	1	0	0	1	0	x	1	0	0	0	1
52	1	1	0	0	1	1	1	1	x	x	1	0
53	1	1	0	1	0	0	1	1	x	x	1	0
54	1	1	0	1	0	1	x	1	0	1	0	1
55	1	1	0	1	1	0	1	x	0	1	0	1
56	1	1	0	1	1	1	1	1	0	1	1	1
57	1	1	1	0	0	0	1	1	x	x	1	0
58	1	1	1	0	0	1	x	1	1	0	0	1
59	1	1	1	0	1	0	1	x	1	0	0	1
60	1	1	1	0	1	1	1	1	1	0	1	1
61	1	1	1	1	0	0	1	1	1	1	1	1
62	1	1	1	1	0	1	1	x	1	1	0	1
63	1	1	1	1	1	0	x	1	1	1	0	1
64	1	1	1	1	1	1	1	1	1	x	1	0

Untuk penyederhanaan tabel 7., dilakukan dengan menggunakan peta karnaugh dan tabulasi Quine Mc Cluskey.

Tabel 8.a. Peta Karnaugh untuk output p Decoder

abc def	000	001	011	010	110	111	101	100
000				x	1	1	x	1
001	x				1	x	1	1
011			x		1	1	1	x
010		x			x	1	1	1
110	x				1	x	1	1
111				x	1	1	x	1
101		x			x	1	1	1
100			x		1	1	1	x

p = a ..... 3

Tabel 8.b. Peta Karnaugh untuk output q Decoder

abc def	000	001	011	010	110	111	101	100
000			x	1	1	1		x
001		x	1	1	x	1		
011			1	x	1	1	X	
010	x		1	1	1	x		
110		x	1	1	x	1		
111			x	1	1	1		x
101	x		1	1	1	x		
100			1	x	1	1	X	

q = b ..... 4

Tabel 8.c. Peta Karnaugh untuk output r Decoder

abc def	000	001	011	010	110	111	101	100
000	x	1	1			x	1	
001		1	1	x		1	x	
011		x	1		x	1	1	
010		1	x			1	1	x
110		1	1	x		1	x	
111	x	1	1			x	1	
101		1	x			1	1	x
100		x	1		x	1	1	

r = c ..... 5

Tabel 8.d. Peta Karnaugh untuk output s Decoder

abc def		000	001	011	010	110	111	101	100
		000	x					x	
s	001				x			x	
	011		x			x			
	010			x					x
	110	1	1	1	x	1	1	x	1
	111	x	1	1	1	1	x	1	1
	101	1	1	x	1	1	1	1	x
	100	1	x	1	1	x	1	1	1

s = d ..... 6

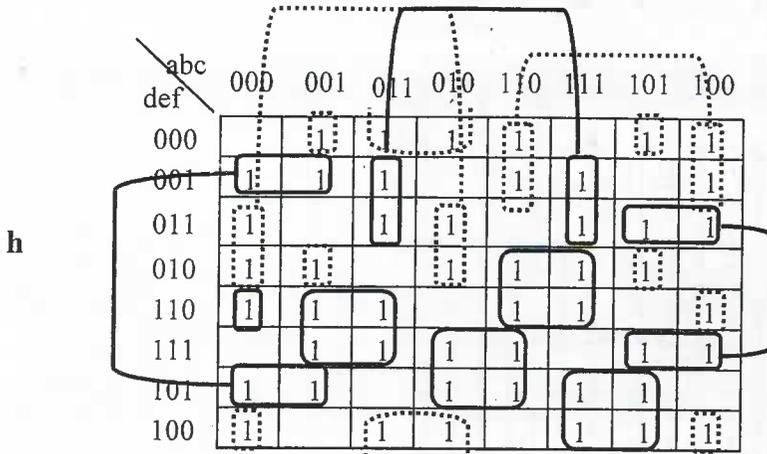
Tabel 8.e. Peta Karnaugh untuk output g Decoder

abc def		000	001	011	010	110	111	101	100
		000	1	1			1	1	
g	001			1	1			1	1
	011	1	1			1	1		
	010			1	1			1	1
	110			1	1			1	1
	111	1	1			1	1		
	101			1	1			1	1
	100	1	1			1	1		

$$\begin{aligned}
 g &= a'b'e'f' + a'b'ef + a'be'f + a'bef + ab'e'f + ab'ef + abe'f + abef \\
 &= a'e'(b'f' + bf) + a'e(b'f + bf') + ae'(b'f + bf') + ae(b'f' + bf) \\
 &= (a'e' + ae)(b'f' + bf) + (a'e + ae')(b'f + bf') \\
 &= km + ln \dots\dots\dots 7
 \end{aligned}$$

dengan  $k = (a'e' + ae)$ ,  $l = (a'e + ae')$   
 $m = (b'f' + bf)$ ,  $n = (b'f + bf')$

Tabel 8.f. Peta Karnaugh untuk output h Decoder



$$\begin{aligned}
 h &= a'b'e'f + a'c'd'e + b'c'df + b'cd'f + a'be'f + ac'd'e + a'cde + bc'df + bcd'f + ab'ef + acde' + abef \\
 &= a'e'(b'f+bf') + a'e(c'd' + cd) + b'f(c'd + cd') + ae'(c'd' + cd) + bf(c'd + cd') + ae(b'f + bf') \\
 &= (a'e' + ae)(b'f+bf') + (a'e + ae')(c'd' + cd) + (b'f + bf)(c'd + cd') \\
 &= \mathbf{kn + lw + mx} \dots\dots\dots 8
 \end{aligned}$$

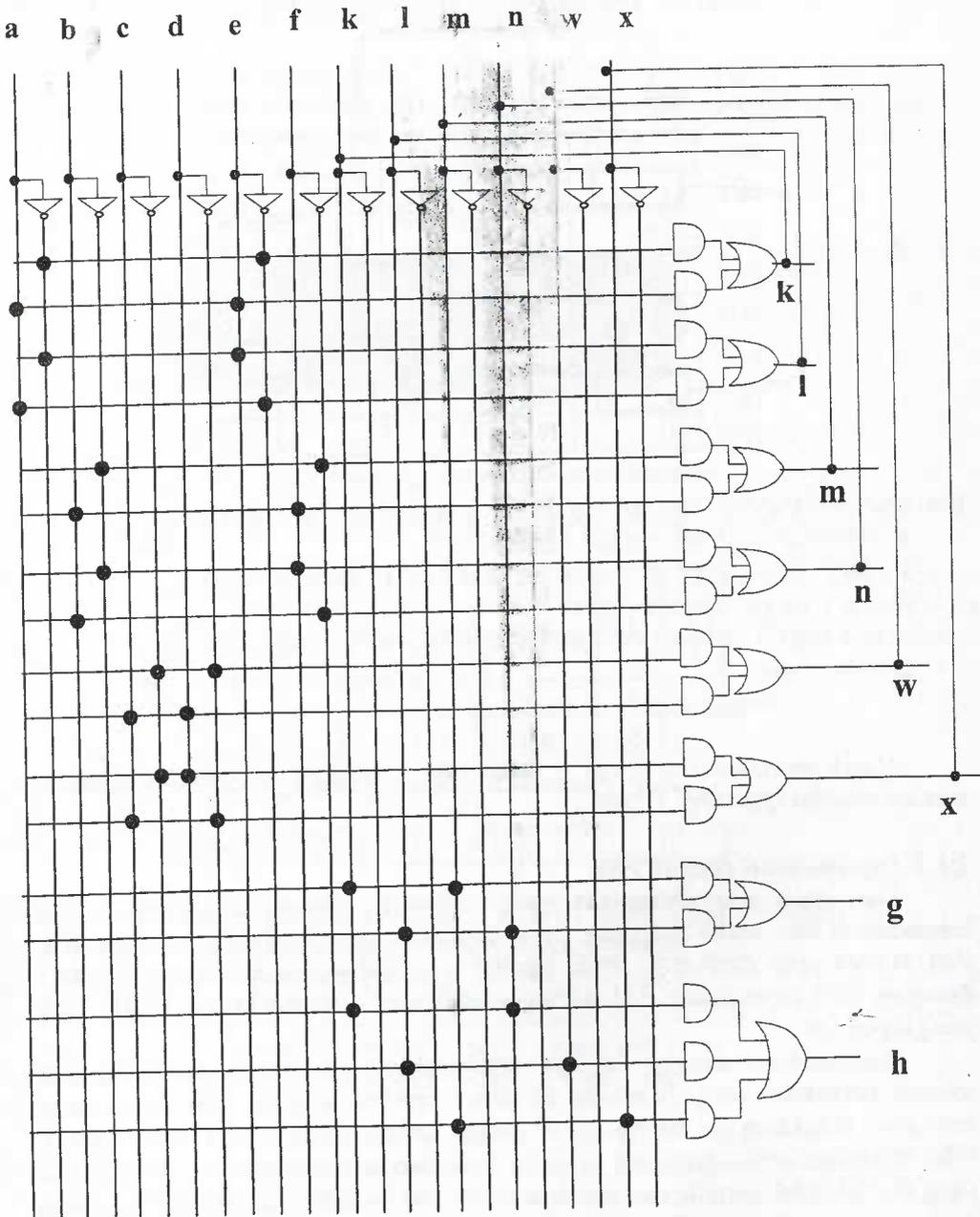
dengan  $w = (c'd' + cd)$   
 $x = (c'd + cd')$

Untuk mendapatkan harga h dapat juga dilakukan dengan menggunakan metoda tabulasi Quine Mc Cluskey

**2.1.3. Implementasi dengan PAL**

Dari hasil penyederhanaan yang diperoleh dengan menggunakan peta Karnaugh di atas, maka rangkaian dapat diimplementasikan dengan menggunakan PAL seperti pada gambar 8. Pada gambar 8 terlihat bahwa k, l, m, n, w, dan x diumpan balik ke masukan PAL sehingga lebih menyederhanakan rangkaian logik yang diperoleh.

Implementasi dengan PAL ini hanya dilakukan dengan merealisasikan seluruh rangkaian yang dirancang ke dalam struktur PAL di atas kertas yang kemudian rangkaian ini disimulasikan dengan menggunakan EWB. Implementasi tidak dilakukan secara langsung ke dalam PAL karena ketiadaan PAL Programmer yang dimiliki oleh peneliti dan lembaga UHN. Hal ini tidak mengurangi keandalan hasil rancangan karena dipastikan akan berjalan dengan baik pada PAL karena hasil simulasi berjalan dengan baik.



Gambar 8. Implementasi PAL dari Decoder enam bit data

**2.1.4. Simulasi dengan Perangkat Lunak EWB**

Hasil akhir rangkaian Decoder yang diperoleh dengan implementasi PAL di atas disimulasikan dengan menggunakan alat bantu perangkat lunak EWB dengan hasil seperti pada tabel 9.

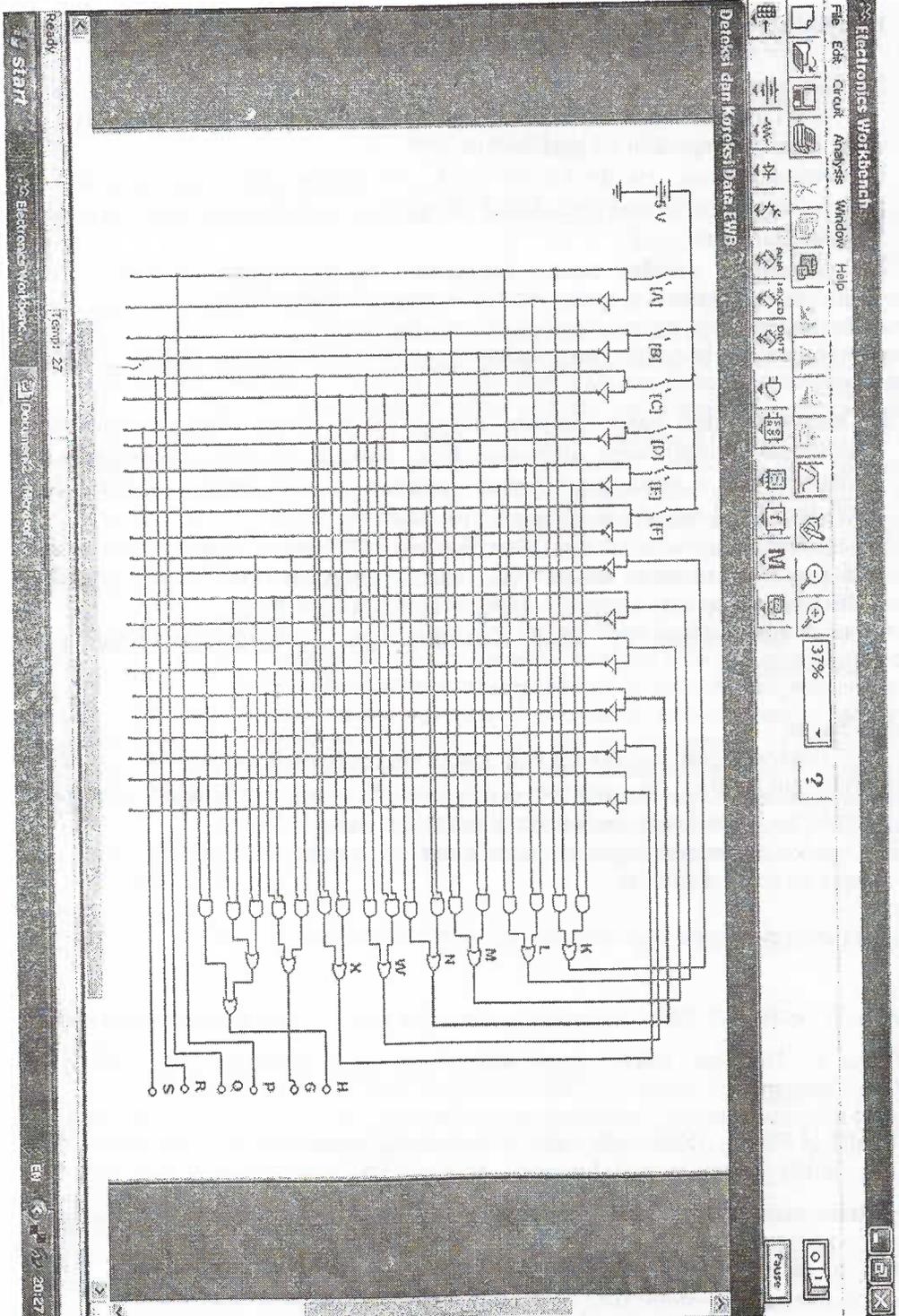
Tabel 9. Hasil Simulasi.Decoder

Nu	a	b	c	d	e	f	p	q	r	s	g	h
1	0	0	0	0	0	0	0	0	0	0	1	0
2	0	0	0	0	0	1	x/0	0	0	0	0	1
3	0	0	0	0	1	0	0	x/0	0	0	0	1
4	0	0	0	0	1	1	0	0	0	0	1	1
5	0	0	0	1	0	0	0	0	0	1	1	1
6	0	0	0	1	0	1	0	x/0	0	1	0	1
7	0	0	0	1	1	0	x	0	0	1	0	1
8	0	0	0	1	1	1	0	0	x/0	x/1	1	0
9	0	0	1	0	0	0	0	0	1	0	1	1
10	0	0	1	0	0	1	0	x/0	1	0	0	1
11	0	0	1	0	1	0	x/0	0	1	0	0	1
12	0	0	1	0	1	1	0	0	x/1	x	1	0
13	0	0	1	1	0	0	0	0	x/1	x/1	1	0
14	0	0	1	1	0	1	x/0	0	1	1	0	1
15	0	0	1	1	1	0	0	x/0	1	1	0	1
16	0	0	1	1	1	1	0	0	1	1	1	1
17	0	1	0	0	0	0	x/0	1	0	0	0	1
18	0	1	0	0	0	1	0	1	x/0	x/0	1	0
19	0	1	0	0	1	0	0	1	0	0	1	1
20	0	1	0	0	1	1	0	x/1	0	0	0	1
21	0	1	0	1	0	0	0	x/1	0	1	0	1
22	0	1	0	1	0	1	0	1	0	1	1	1
23	0	1	0	1	1	0	0	1	x/0	x/1	1	0
24	0	1	0	1	1	1	x/0	1	0	1	0	1
25	0	1	1	0	0	0	0	x/1	1	0	0	1
26	0	1	1	0	0	1	0	1	1	0	1	1
27	0	1	1	0	1	0	0	1	x/1	x/0	1	0
28	0	1	1	0	1	1	x/0	1	1	0	0	1
29	0	1	1	1	0	0	x/0	1	1	1	0	1
30	0	1	1	1	0	1	0	1	x/1	x/1	1	0
31	0	1	1	1	1	0	0	1	1	1	1	1
32	0	1	1	1	1	1	0	x/1	1	1	0	1
33	1	0	0	0	0	0	1	x/0	0	0	0	1
34	1	0	0	0	0	1	1	0	0	0	1	1

Tabel 9. Hasil Simulasi Decoder (lanjutan)

Nu	a	b	c	d	e	f	p	q	r	s	g	h
35	1	0	0	0	1	0	1	0	x/0	x/0	1	0
36	1	0	0	0	1	1	x/1	0	0	0	0	1
37	1	0	0	1	0	0	x/1	0	0	1	0	1
38	1	0	0	1	0	1	1	0	x/0	x/1	1	0
39	1	0	0	1	1	0	1	0	0	1	1	1
40	1	0	0	1	1	1	1	x/0	0	1	0	1
41	1	0	1	0	0	0	x/1	0	1	0	0	1
42	1	0	1	0	0	1	1	0	x/1	x/0	1	0
43	1	0	1	0	1	0	1	0	1	0	1	1
44	1	0	1	0	1	1	1	x/0	1	0	0	1
45	1	0	1	1	0	0	1	x/0	1	1	0	1
46	1	0	1	1	0	1	1	0	1	1	1	1
47	1	0	1	1	1	0	1	0	x/1	x/1	1	0
48	1	0	1	1	1	1	x/1	0	1	1	0	1
49	1	1	0	0	0	0	1	1	0	0	1	1
50	1	1	0	0	0	1	1	x/1	0	0	0	1
51	1	1	0	0	1	0	x/1	1	0	0	0	1
52	1	1	0	0	1	1	1	1	x/0	x/0	1	0
53	1	1	0	1	0	0	1	1	x/0	x/1	1	0
54	1	1	0	1	0	1	x/1	1	0	1	0	1
55	1	1	0	1	1	0	1	x/1	0	1	0	1
56	1	1	0	1	1	1	1	1	0	1	1	1
57	1	1	1	0	0	0	1	1	x/1	x/0	1	0
58	1	1	1	0	0	1	x/1	1	1	0	0	1
59	1	1	1	0	1	0	1	x/1	1	0	0	1
60	1	1	1	0	1	1	1	1	1	0	1	1
61	1	1	1	1	0	0	1	1	1	1	1	1
62	1	1	1	1	0	1	1	x/1	1	1	0	1
63	1	1	1	1	1	0	x/1	1	1	1	0	1
64	1	1	1	1	1	1	1	1	x/1	x/1	1	0

Dari semua kemungkinan kombinasi abcdef yang dilakukan, tidak terdapat kesalahan sehingga hasil keluaran dari simulasi (tabel 9.) sama dengan perencanaan awal (tabel 7.) dengan catatan bahwa keadaan don't care dapat dianggap bit 1 atau bit 0. Gambar rangkaian Decoder yang disimulasikan dapat dilihat pada gambar 9.



Gambar 9. Gambar Decoder yang Disimulasikan dengan EWB

### 3. KESIMPULAN DAN SARAN

#### 3.1. Kesimpulan

Dari hasil simulasi deteksi dan koreksi kesalahan pada komunikasi data ini, maka dapat disimpulkan sebagai berikut :

1. Perancangan simulasi deteksi dan koreksi kesalahan pada komunikasi data ini dapat memberi kemudahan dalam mengetahui apakah data yang dikirimkan benar atau salah.
2. Perancangan simulasi deteksi dan koreksi kesalahan pada komunikasi data ini, menggunakan program EWB, sehingga mudah dimengerti dan dapat dijalankan tanpa harus merancang secara langsung.
3. Pendeteksian kesalahan tidak dapat diketahui jika kesalahan lebih dari satu bit data.
4. Dengan simulasi yang dilakukan dengan EWB, bahwa semua kemungkinan kombinasi abcdef yang dilakukan, tidak terdapat kesalahan sehingga hasil keluaran dari simulasi sama dengan perencanaan awal, dengan asumsi bahwa keadaan don't care dapat dianggap bit 1 atau bit 0.
5. Simulasi yang berhasil dilakukan dengan EWB akan dapat dengan mudah diimplementasikan ke dalam PAL, karena rangkaian yang dibuat dipetakan dengan menggunakan gerbang-gerbang yang ada pada PAL.
6. Hasil perancangan ini dapat digunakan sebagai modul percobaan pada Laboratorium.

#### 3.2. Saran

Dari hasil perancangan ini, disarankan hal-hal berikut :

1. Dibutuhkan PAL Programmer untuk merealisasikannya kedalam sebuah IC sehingga benar-benar dapat digunakan secara nyata.
2. Digunakan penggabungan dengan metoda lain agar koreksi kesalahan lebih dari satu bit dapat dilakukan.

### 4. DAFTAR PUSTAKA

- Alan B. Marcovitz, 2002, *Introduction To Logic Design*, Mc Graw Hill, Singapore
- Roger L. Tokheim, 1985, *Digital Electronics*, Second Edition, Mc Graw Hill, Singapore
- Wayne Tomasi, 1994, *Advanced Electronic Communications Systems*, Third Edition, Prentice Hall International, Inc., USA
- William Stalling, 2000, *Data & Computer Communication*, 6<sup>th</sup> Edition, Prentice Hall Inc., New Jersey