

# PENGGUNAAN TABEL KEBENARAN DALAM MERANCANG DESAIN DIGITAL

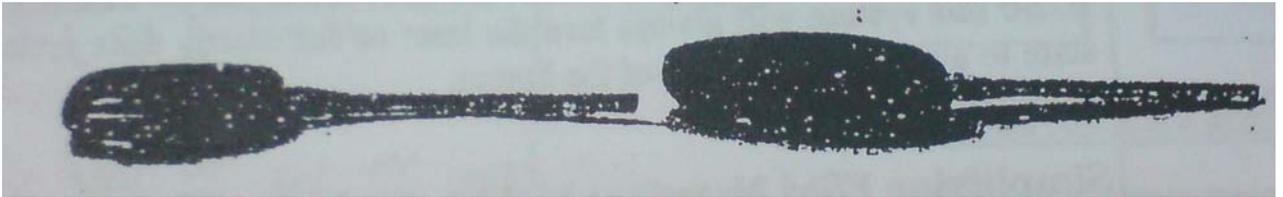
**Tommy – NIM : 13507109**  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung  
Jl. Ganesha no. 10 Bandung  
Email : tommy\_gunardi@hotmail.com

## ABSTRAK

Tabel Kebenaran merupakan salah satu bagian dari pokok bahasan logika. Pemanfaatan Tabel Kebenaran dalam kehidupan sehari – hari sangat beragam macamnya. Salah satu contoh pemakaian tabel kebenaran dalam kehidupan sehari – hari adalah penggunaannya dalam merancang desain digital. Desain digital yang dimaksud disini contohnya adalah kunci pengaman mobil. Di sini, kita memiliki banyak state berbeda – beda. Fungsi tabel kebenaran di sini adalah membentuk persamaan – persamaan yang nantinya akan dipakai untuk merangkai desain digital yang ingin kita rangkai. Dalam makalah ini, akan diperlihatkan pemakaian tabel kebenaran dalam merancang desain digital.

**Kata Kunci :** Tabel Kebenaran, desain digital,

## 1. Pendahuluan



Gambar 1 : Kunci jaman dulu dengan kunci sekarang ini

Pernahkah anda memperhatikan bahwa kunci mobil – mobil sekarang ini memiliki kepala plastik yang lebih tebal daripada kunci – kunci terdahulu? Alasannya, percaya atau tidak, adalah ada chip komputer di dalam kepala plastik kunci, yang berguna sebagai kunci pengaman mobil.

ke chip kunci mobil. Chip kunci mobil membalas sinyal radio itu dengan mengirimkan respon *identifier* (ID) via sinyal radio. Chip pada kunci yang mengirim balik sinyal radio tersebut disebut *transponder*. Jika *basestation* tidak menerima respon atau respon dari *transponder* memiliki ID yang berbeda dari ID yang sudah terprogram di komputer mobil, komputer akan mati dan mobil tidak akan menyala.



Gambar 2 : Kunci mobil sekarang

Ketika pengendara menstarter mobil, komputer mobil ( di bawah cover dan berkomunikasi dengan menggunakan *basestation* ) mengeluarkan sinyal radio



Gambar 3 : Kunci mobil sekarang

Kunci mobil adalah salah satu contoh perancangan

kontroller desain digital. Proses perancangan kontroller desain ini akan mencakup 5 tahap yakni :

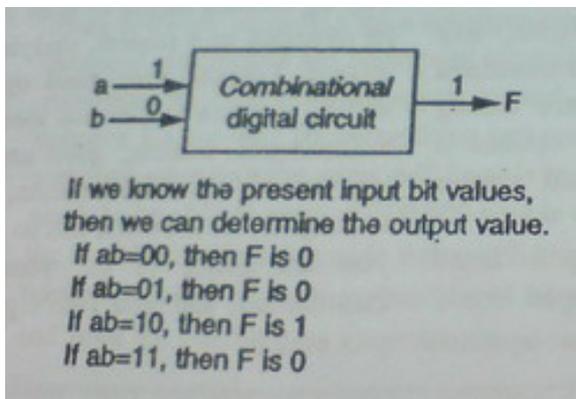
1. Capture the FSM
2. Create the achitecture
3. Encode the states
4. Create the state table
5. Implement the combinational logic

Untuk lebih memahami step – step di atas akan lebih baik jika kita lebih mengenal lebih jauh tentang perancangan ini.

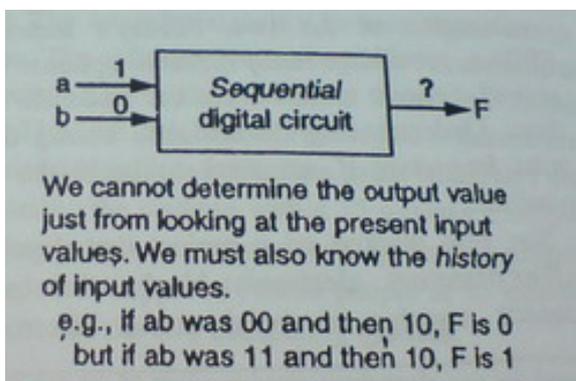
## 2. Pengertian Dasar

### 2.1 Rangkaian Kombinasional dan Sekuensial

Rangkaian kombinasional adalah rangkaian yang tidak memperdulikan data masukan sebelumnya dan hanya memperhitungkan data masukan sekarang ini. Rangkaian sekuensial adalah rangkaian yang peduli dan bergantung dengan data masukan sebelumnya. Dalam penerapannya, kedua rangkaian digunakan sesuai dengan kebutuhan.



Gambar 4 : Sirkuit Kombinasional



Gambar 5 : Sirkuit Sekuensial

### 2.2 Representations of Boolean Function

Fungsi Boolean memiliki 3 macam representasi yakni *Equations*, *Circuit*, dan *Tabel Kebenaran*. Masing – masing fungsi memiliki kelebihan dan kekurangannya masing – masing. Tapi bagaimanapun juga, semuanya tetap merupakan satu kesatuan yang sama.

#### 2.2.1 Equations

*Equations* adalah persamaan matematika yang menghubungkan satu ekspresi dengan ekspresi lainnya.

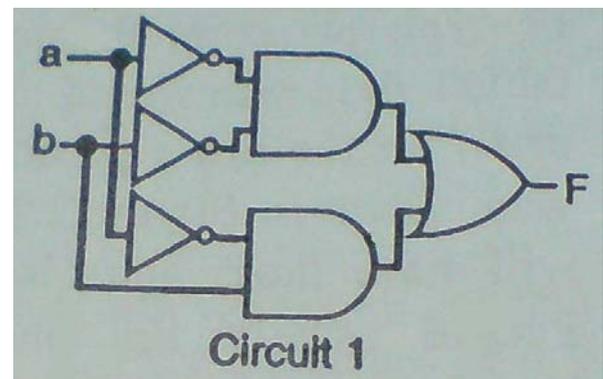
Contoh:

$$F(a,b) = a'b' + a'b$$

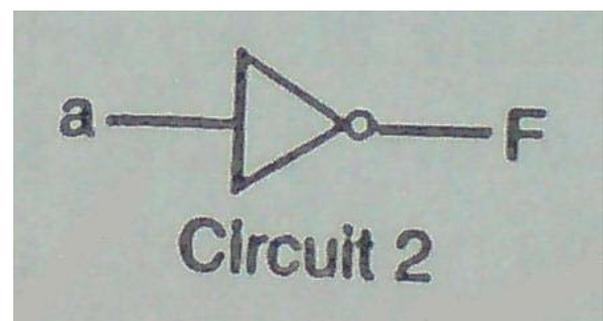
$F(a,b)$  akan bernilai 1 atau 0 tergantung kepada input  $a$  dan  $b$ . Kelebihan dari *Equations* sebagai representasi Fungsi Boolean dibandingkan representasi lainnya adalah kita dapat dengan mudah memanipulasi sebuah equation dengan operasi aljabar sehingga kita bisa mempermudah persamaan, membuktikan dua buah equations menyatakan fungsi yang sama, dan lain – lain.

#### 2.2.2 Circuits

Cara kedua merepresentasikan fungsi Boolean adalah menggunakan sirkuit yang berisi logic gates. Sirkuit adalah hubungan antar komponen.



Gambar 6 : Sirkuit 1



Gambar 7 : Sirkuit 2

Kedua gambar diatas sama sama merepresentasikan

fungsi F. Namun sirkuit yang bawah menggunakan lebih sedikit gate, padahal fungsi yang direpresentasikan pada sirkuit ini sama dengan sirkuit atasnya. Kelebihan dari Circuits sebagai representasi Fungsi Boolean dibandingkan representasi lainnya adalah sirkuit merepresentasikan implementasi fisik dari fungsi Boolean, dan tujuan kita adalah mengimplementasikan sirkuit digital secara fisik. Hal ini dapat dilakukan sirkuit.

### 2.2.3 Tabel Kebenaran

Menurut diktat struktur diskrit, tabel kebenaran adalah tabel untuk menentukan nilai kebenaran proposisi majemuk. Tabel Kebenaran menampilkan hubungan antar anilai kebenaran untuk konjungsi, disjungsi dan ingkaran. Pada Makalah ini penulis ingin menunjukkan penggunaan tabel kebenaran terhadap pemecahan masalah lain. Masalah yang akan kita bahas adalah masalah secure car key. Kita dapat merepresentasikan fungsi Boolean dengan menggunakan Tabel Kebenaran. Bagian kiri tabel kebenaran berisi input, dan bagian kanan tabel berisi output. Masing – masing direpresentasikan sebuah baris tiap kombinasi.

Contoh:

Tabel 1. Contoh tabel kebenaran

Input			Output
A	B	C	Z
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Tabel Kebenaran di atas memiliki tiga buah input. Input di sini berupa 3 bit binary number. Fungsi dengan 3 variabel akan membutuhkan 8 kombinasi input di bagian kiri. Begitupun dengan input 2 Bit atau lainnya. Untuk input 2 bit kita akan membutuhkan 4 kombinasi input, yakni 00, 01, 10, dan 11. Sehingga dapat ditentukan rumus untuk mencari panjang baris untuk input n bit adalah  $2^n$ . Tidak seperti sirkuit dan persamaan, fungsi boolean hanya memiliki 1 macam tabel kebenaran. Keuntungan menggunakan tabel kebenaran adalah sebuah fungsi hanya memiliki satu buah tabel kebenaran. Maka kita dapat mengkonvert berbagai macam fungsi boolean ke dalam tabel kebenaran untuk menentukan jika representasi berbeda

merepresentasikan suatu fungsi yang sama. Jika dua atau lebih representasi merepresentasikan suatu fungsi yang sama, maka tabel kebenaran mereka pasti identik.

Tabel kebenaran juga cukup intuitif karena tabel kebenaran dengan jelas menampilkan semua kemungkinan input.

### 2.3 Converting among Boolean Function Representation

Kita dapat merangkai desain rangkaian kombinasional dengan mengidentifikasi fungsi boolean yang tepat untuk menyelesaikan permasalahan, dan kemudian merancang representasi sirkuit yang sesuai dengan fungsi itu. Fungsi boolean itu dapat berupa *Equations* dan Tabel Kebenaran. Selanjutnya, kita harus mengubah fungsi yang ada menjadi sirkuit. Selain itu, kita juga bisa mengubah fungsi yang satu menjadi fungsi lainnya. Karena ada tiga representasi, kita akan memiliki 6 macam operasi tergantung dari apa yang ingin kita ubah.

### 2.4 Membuat rangkaian

Apa itu FSM? FSM adalah singkatan dari Finite State Machine atau Mesin state terbatas. FSM memiliki beberapa bagian : yang paing penting adalah bagaimana cara merepresentasikan setiap state yang mungkin.

Contoh intuitifnya adalah hamster peliharaan. Setelah dipelihara, kita dapat megetahui bahwa hamster pada dasarnya memiliki 4 macam state. Tidur, Makan, Berlari di roda, dan berusaha kabur.

Sekarang kita akan merancang rangkaian kunci pengaman mobil yang sebenarnya. Maka kita harus mengikuti step per step pembuatan rangkaian. Anggap ID dari kunci adalah 4 Bit binary dan nilainya 1011 ( Pada rangkaian yang sebenarnya, ID dari kunci mobil nilainya 32 bit atau lebih).

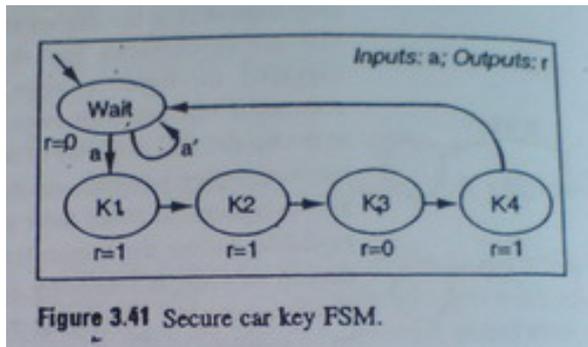
#### 2.4.1 Capture the FSM

Pada step ini kita akan membuat FSM yang menjabarkan sifat controller. Asumsikan controller memiliki input yang bernilai 1 ketika komputer mobil meminta ID dari kunci. Maka controller awalnya menunggu ( dalam kondisi wait sampai masukan input berupa 1). Kunci kemudian mengirimkan ID secara berurutan dimulai dari bit paling kanan.

Dengan output r :

- 1 untuk cycle pertama
- 1 untuk cycle kedua
- 0 untuk cycle ketiga

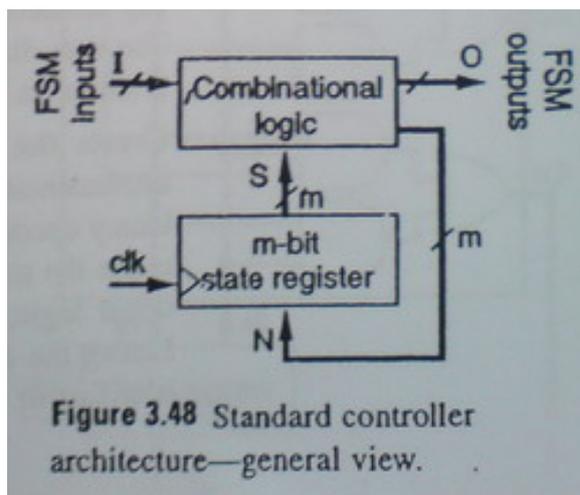
1 untuk cycle keempat



Gambar 8 : FSM

### 2.4.2 Create the Architecture

Membuat arsitektur standar dengan menggunakan sebuah state register dengan ukuran yang tepat, dan kombinasional logic dengan input menjadi state register bit, dan input FSM input. Lalu output dan next state menjadi output. Kita memiliki standard asitektur secara umum pada Gambar :



Gambar 9 : arsitektur kontroler umum

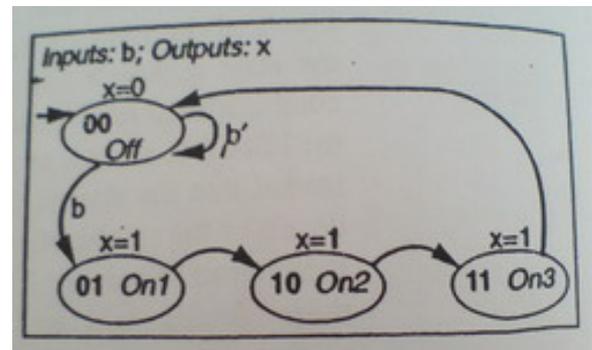
State register di sini memiliki 2 -Bit untuk merepresentasikan masing - masing dari 4 state. Kombinasional logicnya mempunyai input eksternal b, dan input s1 dan s0 yang datang dari state register. Dan memiliki output x dan output n1 dan n0 masuk ke state register.

### 2.4.3 Encode the states

Meng assign angka biner yang unik ke masing - masing state. Setiap angka biner merepresentasikan state yang dikenal dengan encoding. Semua encoding akan terus berjalan sampai memiliki encoding yang unik. Kita dapat meng-encode state - statenya sebagai berikut :

Off = 1                      On1 = 01  
On2 = 10                     On3 = 11

State diagram untuk state yang sudah di encode dapat dilihat pada gambar di bawah ini:



Gambar 10 : State Diagram

### 2.4.4 Create the state table

Membuat tabel kebenaran untuk kombinasional logic sedemikian rupa sehingga logic akan menghasilkan output FSM yang benar dan next statenya.

Pertama - tama kita mengisi semua kemungkinan input pada bagian kiri tabel. Lalu dengan membaca input state diagram, kita akan mengetahui next statenya.

	Inputs			Outputs		
	s1	s0	b	x	n1	n0
Off	0	0	0	0	0	0
	0	0	1	0	0	1
On1	0	1	0	1	1	0
	0	1	1	1	1	0
On2	1	0	0	1	1	1
	1	0	1	1	1	1
On3	1	1	0	1	0	0
	1	1	1	1	0	0

Gambar 11 : Tabel Kebenaran

### 2.4.5 Implement the combinational logic

Mengimplementasikan kombinasional logic dengan semua metode.

Kita mengambil bagian - bagian dimana outout 1. Untuk x = 1 kita mendapatkan fungsi :

$$x = s1 + s0...(1)$$

Untuk  $n1 = 1$ , kita mendapatkan fungsi :

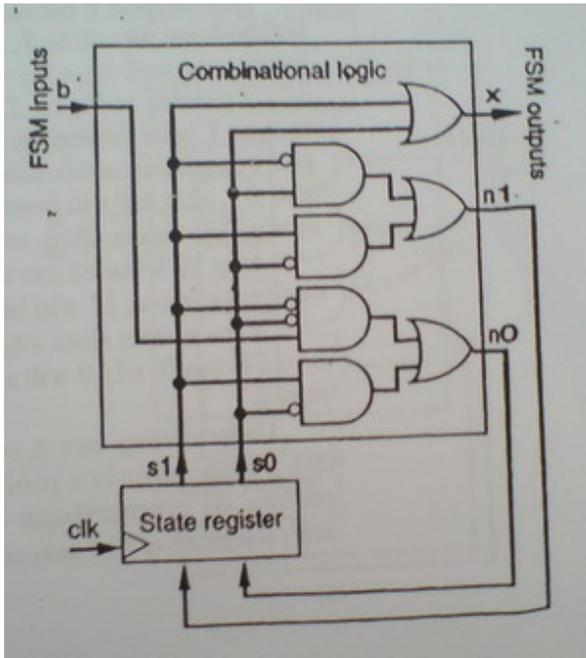
$$ni = s1'.s0.b' + s1'.s0.b + s1.s0'.b' + s1.s0'.b$$

$$ni = s1's0 + s1s0...(2)$$

Untuk  $n0 = 1$

$$ni = s1'.s0'.b + s1.s0'...(3)$$

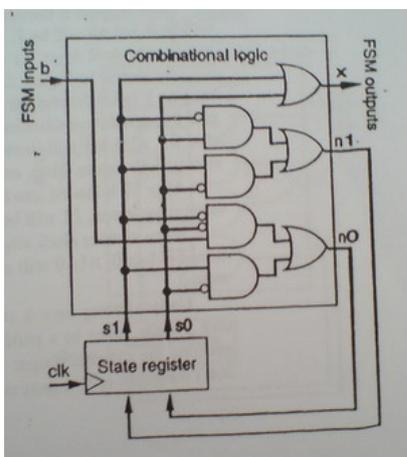
Dengan ketiga persamaan itu dapat dibuat rangkaian :



Gambar 12 : rangkaian hasil implementasi FSM

### 3. Hasil dan pembahasan

Sirkuit yang diperoleh adalah :



Gambar 13 : Hasil proses dalam bentuk sirkuit

Jadi melalui proses di atas kita mendapatkan hasil rangkaian fisik dalam bentuk sirkuit.

### 4. Kesimpulan

Tabel Kebenaran adalah salah satu prinsip logika yang dapat kita gunakan dalam membantu menyelesaikan masalah. Kita dapat membuat rangkaian dengan metode *five step controller desing process* yang isinya:

1. Capture the FSM
2. Create the achitecture
3. Encode the states
4. Create the state table
5. Implement the combinational logic

Di sini tabel kebenaran berperan penting dalam membuat rangkaian. Tanpa tabel kebenaran, alhasil tahapan tidak dapat dilakukan. Selain itu Tabel kebenaran dapat berbentuk banyak sekali tergantung permasalahan yang kita hadapi.

### DAFTAR REFERENSI

- [1] Vahid, Frank.(2007). "Digital Design", Wiley, chapter 2-3: USA.
- [2] Munir, Rinaldi.(2008). Diktat Kuliah IF2091 Struktur Diskrit, Departemen Teknik Informatika, Institut Teknologi Bandung.
- [3] [www.ecufocus.com/cd-products.asp?type1=29](http://www.ecufocus.com/cd-products.asp?type1=29)