

## BAB II TINJAUAN PUSTAKA

### 2.1 Aljabar Boolean

Barnett (2011) menyatakan bahwa Aljabar Boolean dipublikasikan oleh George Boole dalam *An Investigation of the Laws of Thought* pada tahun 1854. Dalam karya ini, Boole mengembangkan sistem dari simbol-simbol ( $x$  dan  $+$ ) yang mempresentasikan operasi-operasi pada himpunan yang secara simbolis dipresentasikan oleh huruf-huruf. Rosen (2012) dalam buku *Discrete Mathematics and Its Applications*, Aljabar Boolean menyediakan aturan dan operasi untuk bekerja dengan himpunan  $\{0, 1\}$ . Operasi dalam Aljabar Boolean yang sebagian besar digunakan yaitu komplemen, penjumlahan, dan perkalian Boolean. Komplemen dari elemen, dilambangkan dengan tanda bar ( $\bar{\phantom{x}}$ ), didefinisikan dengan  $\bar{0} = 1$  dan  $\bar{1} = 0$ . Penjumlahan Boolean, dilambangkan dengan tanda tambah ( $+$ ) atau OR, mempunyai nilai-nilai berikut:

$$1 + 1 = 1, \quad 1 + 0 = 1, \quad 0 + 1 = 1, \quad 0 + 0 = 0$$

Perkalian Boolean, dilambangkan dengan tanda titik di tengah ( $\cdot$ ) atau AND, mempunyai nilai-nilai berikut:

$$1 \cdot 1 = 1, \quad 1 \cdot 0 = 0, \quad 0 \cdot 1 = 0, \quad 0 \cdot 0 = 0$$

Identitas dalam Aljabar Boolean sangat berguna untuk menyederhanakan rangkaian digital, (Rosen, 2012). Identitas dalam Aljabar Boolean dapat dilihat pada Tabel 2.1.

**Tabel 2.1 Identitas Boolean**

No	Identitas	Nama
1	$\overline{\overline{x}}$	Hukum Komplemen Ganda
2	$x + x = x$ $x \cdot x = x$	Hukum Idempoten
3	$x + 0 = x$ $x \cdot 1 = x$	Hukum Identitas
4	$x + 1 = 1$ $x \cdot 0 = 0$	Hukum Dominansi
5	$x + y = y + x$ $xy = yx$	Hukum Komulatif
6	$x + (y + z) = (x + y) + z$ $x(yz) = (xy)z$	Hukum Asosiatif
7	$x + yz = (x + y)(x + z)$ $x(y + z) = xy + xz$	Hukum Distributif
8	$\overline{(xy)} = \overline{x} + \overline{y}$ $\overline{(x + y)} = \overline{x} \overline{y}$	Hukum De Morgan's
9	$x + xy = x$ $x(x + y) = x$	Hukum Absorpsi
10	$x + \overline{x} = 1$ $x \overline{x} = 0$	Hukum Komplemen

## 2.2 Minterms

*Minterms* dari peubah Boolean  $x_1, x_2, \dots, x_n$  adalah hasil perkalian Boolean  $y_1 y_2 \dots y_n$ , di mana  $y_i = x_i$  atau  $y_i = \overline{x_i}$ , (Rosen, 2012). *Minterms*  $y_1 y_2 \dots y_n$  bernilai 1, jika dan hanya jika setiap  $y_i$  bernilai 1, dan ini terjadi, jika dan hanya jika  $x_i = 1$  ketika  $y_i = x_i$  dan  $x_i = 0$  ketika  $y_i = \overline{x_i}$ .

### 2.3 *Sum-of-Products*

*Sum-of-Products* adalah penjumlahan dari *minterms* atau bentuk normal disjungtif dari Fungsi Boolean, (Rosen, 2012). Contoh *Sum-of-Products* adalah  $F(x, y, z) = \overline{xy}z + x\overline{y}z + \overline{x}yz$ .

### 2.4 Metode *Quine-McCluskey*

Seda (2008) menyatakan bahwa Metode *Quine-McCluskey* adalah algoritma yang tepat berdasarkan aplikasi sistematis dari hukum distributif, hukum komplemen dan hukum idempoten. Tetapi Seda juga mengatakan bahwa metode tersebut memberikan prosedur yang unik dari komputasi, sehingga secara sederhana dapat diimplementasikan, tetapi, bahkan untuk contoh yang sederhana tidak menjamin solusi yang optimal. Kemudian, menurut Tomaszewski, *et al* (2003), Metode *Quine-McCluskey* adalah teknik yang sangat sederhana dan sistematis untuk menyederhanakan Fungsi Boolean. Metode tersebut mempunyai terutama dua keunggulan dibandingkan dengan Peta *Karnaugh*. Pertama, secara sistematis untuk menghasilkan fungsi sederhana yang kurang bergantung pada pola visual. Kedua, skema yang layak untuk menangani banyak peubah. Rosen (2012) mengatakan bahwa Metode *Quine-McCluskey* adalah prosedur untuk menyederhanakan Fungsi Boolean dalam bentuk *Sum-of-Products* yang dapat menjadi mekanisme untuk digunakan ketika ada lebih dari empat peubah dan tidak bergantung pada inspeksi visual untuk mengidentifikasi istilah-istilah untuk kelompok. Metode tersebut terdiri dari dua bagian. Bagian pertama menemukan istilah-

istilah yang menjadi kandidat untuk dimasukkan dalam hasil sederhana sebagai penjumlahan dari hasil perkalian Boole. Bagian kedua menentukan istilah tersebut untuk benar-benar digunakan.

Metode *Quine-McCluskey* menggunakan urutan langkah-langkah untuk menyederhanakan Fungsi Boolean dalam bentuk *Sum-of-Product*, (Rosen, 2012). Langkah-langkah tersebut adalah:

- a. Nyatakan setiap *minterms* dalam  $n$  peubah menjadi *bit string* yang panjangnya  $n$  dengan '1' di posisi ke- $i$ , jika  $x_i$  terjadi dan '0' dalam posisi tersebut, jika  $\overline{x_i}$  terjadi.
- b. Kelompokkan *bit string* sesuai dengan banyak '1' yang dipunyainya.
- c. Tentukan semua hasil dalam  $n - 1$  peubah yang dapat dibentuk dengan mengkombinasikan *minterms*. *Minterms* yang dapat dikombinasikan ditunjukkan oleh *bit string* yang berbeda tepat satu posisi. Representasikan hasil tersebut dalam  $n - 1$  peubah dengan '1' di posisi ke- $i$ , jika  $x_i$  terjadi pada hasil, '0' di posisi tersebut, jika  $\overline{x_i}$  terjadi, dan '-' dalam posisi tersebut, jika terjadi berbeda tepat satu posisi.
- d. Tentukan semua hasil dalam  $n - 2$  peubah yang dapat dibentuk dengan mengkombinasikan *minterms* dalam  $n - 1$  peubah sebelumnya. Hasil dalam  $n - 1$  peubah yang dapat dikombinasikan ditunjukkan oleh *bit string* yang berbeda tepat satu posisi, serta mempunyai '-' di posisi yang sama.
- e. Lanjutkan mengkombinasikan hasil Fungsi Boolean menjadi hasil dalam peubah yang lebih sedikit selama mungkin.

- f. Cari semua hasil Fungsi Boolean yang muncul yang tidak digunakan untuk membentuk hasil dari penyederhanaan Fungsi Boolean.

## 2.5 *Bubble Sort*

Urut menurut Sjukani (2009) dalam buku Struktur Data (Algoritma dan Struktur Data 2) dengan C, C++ adalah data yang berada dalam tempat penyimpanan, dengan urutan tertentu baik urut menaik atau urut menurun. Pengurutan adalah proses urut. Kemudian, Munir (2007) dalam buku Algoritma dan Pemrograman mengatakan bahwa pengurutan merupakan proses mengatur sekumpulan objek menurut urutan atau susunan tertentu. Pengurutan dikatakan stabil jika dua atau lebih data yang sama tetap pada urutan yang sama sesudah pengurutan.

Terdapat beberapa metode pengurutan, salah satunya adalah *Bubble Sort*. *Bubble Sort* melakukan prinsip pertukaran elemen dalam pengurutan sebanyak  $n - 1$  langkah dengan  $n$  adalah panjang data, (Munir, 2007).

## 2.6 Klasifikasi

Klasifikasi adalah penyusunan bersistem dalam kelompok atau golongan menurut kaidah atau standar yang sudah ditetapkan, (Anonim, 2008).