

Logika Tiga-Nilai dan Penerapannya

Muhammad Naufal Fakhrihal 13518115

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

mn.fakhrihal@students.itb.ac.id

13518115@std.stei.itb.ac.id

Abstrak—Logika tiga-nilai mendefinisikan tiga nilai kebenaran, yaitu dua nilai kebenaran seperti dalam logika boolean *benar*, *salah*, serta nilai tambahan yaitu *tidak diketahui*. Operasi dasar pada logika boolean dapat digeneralisasi ke logika tiga-nilai dengan beberapa penambahan. Logika tiga-nilai diterapkan dalam pembuatan komputer terner yang menawarkan keuntungan dibandingkan komputer biner terutama berupa kerapatan data yang lebih tinggi. Selain itu, logika tiga-nilai dapat digunakan dalam memproses data yang nilainya tidak pasti, seperti membandingkan data *null* dalam *Structured Query Language*.

Abstract—Three-valued logic defines three truth value, which consist of *true*, *false*, and additional value *unknown*. Basic operations in Boolean logic can be generalized to its three-valued logic counterparts, with several additional properties. Three valued logic has its uses in ternary computers, which offer several advantages, including higher data density compared to its binary counterparts. It is also used to process data with indeterminate values, such as in *Structured Query Language null comparison*.

Kata Kunci—Basis Data, Komputer, Logika, Sistem Bilangan Terner

Keywords—Computer, Database, Logic, Ternary Numeral System

I. PENDAHULUAN

Aljabar Boolean mengenal dua buah nilai yang saling berlawanan. Menyesuaikan dengan konteks, kedua nilai ini dapat memiliki sebutan yang berbeda (nol - satu, salah - benar, on - off, high -low, dan sebagainya), tetapi mereka memiliki makna yang sama: satu nilai menyatakan suatu kondisi, sedangkan nilai yang satunya lagi menyatakan kondisi yang sebaliknya. Dalam elektronika, misalnya, HIGH menyatakan adanya arus listrik yang mengalir melalui suatu beban, dan LOW menyatakan tidak adanya arus listrik yang melewati beban.

Aljabar Boolean sering diterapkan dalam logika, penentuan keputusan, dan juga dalam komputer (komputer digital hanya mengenal kedua nilai ini, dan semua hal yang dapat dilakukan komputer berasal dari manipulasi kedua nilai ini).

Dalam banyak bidang, Aljabar dan Logika Boolean dengan dua nilai tersebut sudah cukup. Tetapi, hal ini kurang mencerminkan cara berpikir manusia dan cara kerja alam yang

banyak terdapat ketidakpastian. Sebuah nilai baru yang menyatakan ketidakpastian ini dapat didefinisikan bersama dengan dua nilai yang terdefinisi di logika Boolean sehingga membentuk logika tiga-nilai (*Three-valued Logic*)

Makalah ini akan menjelaskan apa itu logika tiga-nilai dan membandingkannya dengan logika dua-nilai (aljabar boolean), serta menjelaskan aplikasi dari logika tiga-nilai.

II. PENJELASAN SINGKAT LOGIKA BOOLEAN

Logika Boolean atau aljabar Boolean biasanya berhubungan dengan nilai-nilai kebenaran. Nilai yang terdefinisi dalam logika Boolean adalah benar (*true*, 1) dan salah (*false*, 0).

George Boole merumuskan aturan-aturan dalam Aljabar Boolean dalam bukunya *The Laws of Thought*. Dalam aljabar Boolean minimal terdefinisi tiga buah operasi:

1. Sum (OR)

Sum merupakan operasi biner, yaitu menerima dua buah operan, dan menghasilkan 1 atau nilai yang ekuivalen dengan 1 jika dan hanya jika paling sedikit satu operannya bernilai 1 atau ekuivalen dengan 1. Operator sum dapat direpresentasikan dengan beberapa simbol. Secara umum, operator sum direpresentasikan dengan tanda (+). Dalam logika proposisi, operasi ini dinamakan disjungsi dan direpresentasikan dengan simbol (\vee). Dalam aljabar himpunan, operasi ini diberi nama gabungan (*union*) dan diberi tanda (\cup). Perilaku dari sebuah operasi dapat direpresentasikan dalam tabel kebenaran. Dalam tabel kebenaran, dilakukan pendaftaran semua kombinasi nilai operan yang mungkin serta hasil operasinya. Tabel kebenaran untuk operasi sum adalah sebagai berikut.

Tabel I. Tabel Kebenaran untuk Operasi Sum

a	b	$a+b$
0	0	0
0	1	1
1	0	1
1	1	1

2. Product (AND)

Product merupakan operasi biner yang menerima dua buah operan dan menghasilkan 1 atau nilai yang ekuivalen dengan 1 jika dan hanya jika kedua operan bernilai 1 atau ekuivalen dengan 1. Dalam aljabar Boolean, operator ini diberi simbol titik atau dot (\cdot) atau dapat dinyatakan dengan mengkonkatenasi kedua operan (ab ekuivalen dengan $a.b$ atau $a \cdot b$). Dalam logika proporsional, operasi ini dikenal sebagai operasi konjungsi (\wedge), dan dalam aljabar himpunan, dikenal sebagai operasi irisan (*union* \cap) Tabel kebenaran untuk operasi ini adalah sebagai berikut.

Tabel II. Tabel Kebenaran untuk Operasi Product.

a	b	$a.b$
0	0	0
0	1	0
1	0	0
1	1	1

3. Complement (NOT)

Complement merupakan operasi uner yang menerima satu buah operan dan menghasilkan nilai yang berbalikan dengan nilai operan semula. Sehingga komplemen dari nilai 1 atau yang ekuivalen dengan 1 bernilai 0 atau yang ekuivalen dengan 0. Operasi ini dapat dinyatakan dengan memberikan tanda petik ($'$) di depan operan. Dalam logika proporsional, operasi ini juga disebut dengan invers dan dinyatakan dengan tanda (\sim), dan dalam aljabar himpunan, operasi ini dinyatakan dengan memberi garis di atas operan. Tabel kebenaran untuk operasi ini adalah sebagai berikut:

Tabel III. Tabel Kebenaran untuk Operasi Complement

a	$'a$
0	1
1	0

Dengan ketiga operasi tersebut, dalam aljabar boolean berlaku aksioma-aksioma berikut

1. Identitas
 $a + 0 = a$
 $a.1 = a$
2. Komutasi
 $a + b = b + a$
 $a.b = b.a$
3. Distribusi
 $a.(b + c) = a.b + a.c$
 $a + (b.c) = (a + b).(a + c)$
4. Komplemen
 $a + 'a = 1$
 $a.a = 0$

Aksioma-aksioma ini dapat diperluas menjadi

hukum-hukum antara lain:

1. Identitas
 $a + 0 = a$
 $a.1 = a$
2. Null / Dominasi
 $a + 1 = 1$
 $a.0 = 0$
3. Komplemen
 $a + 'a = 1$
 $a.'a = 0$
4. Idempoten
 $a + a = a$
 $a.a = a$
5. Involusi
 $''(a) = a$
6. Absorpsi
 $a + (a.b) = a$
 $a.(a + b) = a$
7. Komutasi
 $a + b = b + a$
 $a.b = b.a$
8. Asosiasi
 $a + (b + c) = (a + b) + c$
 $a.(b.c) = (a.b).c$
9. Distribusi
 $a.(b + c) = a.b + a.c$
 $a + (b.c) = (a + b).(a + c)$
10. De Morgan
 $''(a + b) = 'a.'b$
 $''(a.b) = 'a + 'b$
11. 0 / 1
 $'0 = 1$
 $'1 = 0$

Selain dari ketiga operasi dasar tersebut, dalam banyak aplikasi (seperti elektronika) dikenal juga operasi-operasi yang dapat dinyatakan sebagai hasil dari gabungan ketiga operasi di atas, antara lain:

1. NOR

Operasi ini adalah kebalikan dari operasi OR. Operasi ini menghasilkan 0 jika dan hanya jika paling sedikit satu dari operannya bernilai 1. Operasi ini dapat dinyatakan dalam operasi OR dan NOT sebagai $'(a + b)$. Tabel kebenaran untuk operasi ini adalah sebagai berikut:

Tabel IV. Tabel Kebenaran untuk Operasi NOR

a	b	$a \text{ NOR } b$
0	0	1
0	1	0
1	0	0
1	1	0

2. NAND

Operasi ini adalah kebalikan dari operasi OR. Operasi ini menghasilkan 0 jika dan hanya jika semua operannya bernilai 1. Operasi ini dapat dinyatakan dalam operasi AND dan NOT sebagai $\overline{(a.b)}$. Tabel kebenaran untuk operasi ini adalah sebagai berikut:

Tabel V. Tabel Kebenaran untuk Operasi NAND

a	b	$a \text{ NAND } b$
0	0	1
0	1	1
1	0	1
1	1	0

3. XOR (Exclusive-OR)

Operasi ini menerima dua buah operan dan menghasilkan 1 jika dan hanya jika salah satu operannya bernilai 1, tetapi tidak keduanya. Operasi ini dapat dinyatakan dalam operasi NOT, OR, dan AND sebagai $\overline{a.b} + a.\overline{b}$. Dalam elektronika, implementasi operasi XOR dalam bentuk gerbang logika umum digunakan dalam merancang rangkaian *adder* bilangan biner. Tabel kebenaran dari operasi ini adalah sebagai berikut:

Tabel VI. Tabel Kebenaran untuk Operasi XOR

a	b	$a \text{ XOR } b$
0	0	0
0	1	1
1	0	1
1	1	0

4. XNOR

Operasi ini merupakan kebalikan dari operasi XOR. Operasi ini menerima dua buah operan dan menghasilkan 1 jika dan hanya kedua operan bernilai sama-sama 0 atau 1. Operasi ini dapat dinyatakan dalam operasi NOT, OR, dan AND sebagai $a.b + \overline{a}.\overline{b}$. Tabel kebenaran dari operasi ini adalah sebagai berikut:

Tabel VII. Tabel Kebenaran untuk Operasi XNOR

a	b	$a \text{ XNOR } b$
0	0	1
0	1	0
1	0	0
1	1	1

III. LOGIKA TIGA NILAI

A. Sistem Bilangan Terner

Sistem bilangan terner (*ternary numeral system*) merupakan sistem bilangan dengan basis tiga. dalam sistem bilangan ini, angka satuan yang terdefinisi adalah 0, 1, dan 2.

Sebagai contoh, bilangan 74 dapat dinyatakan dalam sistem bilangan terner sebagai 2202_3 .

$$2202_3 = 2 \times 3^3 + 2 \times 3^2 + 0 \times 3^1 + 2 \times 3^0 = 74$$

B. Sistem Bilangan Terner Seimbang

Sistem bilangan terner seimbang (*balanced ternary numeral system*) merupakan varian dari sistem bilangan terner. Yang membedakannya adalah pada sistem bilangan terner seimbang, angka-angka pembentuknya adalah -1, 0 dan 1. untuk mempermudah penulisan, angka -1 dapat direpresentasikan dengan simbol lain, seperti tanda minus (-) sebagai angka -1 dan tanda plus (+) sebagai angka 1.

Kelebihan dari sistem bilangan ini adalah tidak memerlukan tanda negatif, karena nilai negatif sudah direpresentasikan dalam angka -1. Untuk menjadikan sebuah bilangan dalam representasi terner seimbang negatif, cukup dengan membalikkan angka pada tiap digitnya (1 menjadi -1, -1 menjadi 1). Sebagai contoh, bilangan 74 dapat direpresentasikan dalam sistem bilangan terner seimbang sebagai $+0 \text{ -- } -_{bal3}$.

$$+0 \text{ -- } -_{bal3} = 1 \times 3^4 + 0 \times 3^3 - 1 \times 3^2 + 1 \times 3^1 - 1 \times 3^0 = 74$$

Sementara representasi bilangan -74 didapat dengan mengubah simbol + menjadi - dan sebaliknya.

$$-0 \text{ ++ } -_{bal3} = -1 \times 3^4 + 0 \times 3^3 + 1 \times 3^2 - 1 \times 3^1 + 1 \times 3^0 = -74$$

Konversi dari sistem bilangan terner ke sistem bilangan terner seimbang dilakukan dengan mengubah digit bernilai 2 menjadi - dan menambahkan *carry* sebesar 1 ke digit di sebelah kiri digit tersebut.

$$2202_3 = 221 - = 22 \text{ ++ } = 3 \text{ -- } = +0 \text{ -- } -_{bal3}$$

C. Logika Tiga-Nilai

Dalam logika tiga-nilai dikenal tiga buah nilai yang terdefinisi, dengan satu buah nilai menjadi nilai tengah antara kedua nilai lainnya. Jika kedua nilai lainnya adalah benar (*true*) dan salah (*false*), nilai ketiga adalah tidak diketahui (*unknown*), *null*, *neutral*, dan dapat menyatakan keadaan yang tidak benar dan tidak salah ataupun keadaan yang benar sekaligus salah. Ketiga nilai tersebut dapat direpresentasikan dalam sistem bilangan terner dan sistem bilangan terner seimbang. Untuk selanjutnya penulis menggunakan representasi terner seimbang dalam merepresentasikan nilai kebenaran.

Tabel VIII. Representasi Nilai Kebenaran Logika Tiga Nilai dalam Sistem Bilangan Terner Seimbang

Nilai	Representasi Terner	Representasi Terner Seimbang
Benar / <i>True</i>	0	+
Tidak Diketahui / <i>Unknown</i>	1	0
Salah / <i>False</i>	2	-

Banyak sifat dan aksioma dalam logika Boolean berlaku juga dalam logika tiga-nilai, dengan perbedaan pada operasi-operasi yang ada disebabkan adanya tambahan

kombinasi nilai operan yang mungkin. Dalam logika boolean, terdapat $2^2 = 4$ jenis operasi uner yang mungkin dan $2^{2^2} = 16$ jenis operator biner yang mungkin. Dibandingkan dengan logika boolean, jumlah operasi unik dalam logika tiga-nilai lebih banyak, dengan $3^3 = 27$ jenis operasi uner dan $3^{3^2} = 19683$ operasi biner.

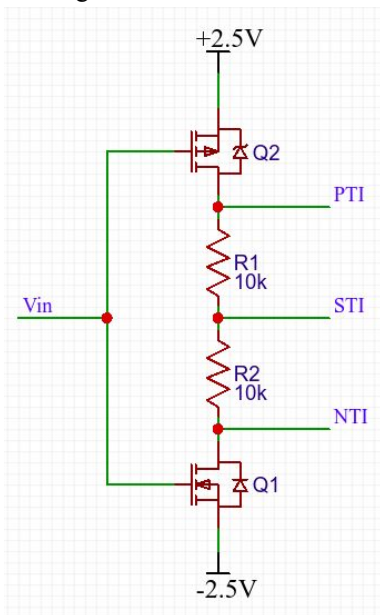
1. NOT / NEG / Inverter

Operasi NOT atau negasi dalam logika tiga-nilai berfungsi sama dengan operasi NOT dalam logika boolean, yaitu membalikkan nilai operannya. Negasi dar benar adalah salah, negasi dari salah adalah benar, dan negasi dari tidak diketahui adalah dirinya sendiri. Tabel kebenaran dari operasi ini adalah sebagai berikut:

Tabel IX. Tabel Kebenaran Operasi Ternary NOT

a	$-a$
-	+
0	0
+	-

Operasi ini dapat diimplementasikan dalam rangkaian elektronika dengan menggunakan transistor NMOS dan PMOS sebagai berikut.



Gambar 1. Rangkaian Ternary Inverter Menggunakan MOSFET dan Resistor

Rangkaian di atas mengeluarkan tiga buah sinyal keluaran. Keluaran tengah merupakan hasil dari operasi NOT, dan operasi ini juga dapat dinamai sebagai *Standard Ternary Inverter* (STI). Dua keluaran lainnya merupakan variasi dari operasi STI, dengan perbedaan hasil operasi jika masukan memiliki nilai *tidak diketahui*. Operasi ini masing diberi nama *Negative Ternary Inverter* (NTI) dan *Positive Ternary Inverter* (PTI), dan masing masing menghasilkan nilai *salah* dan *benar* jika

masukannya bernilai *tidak diketahui*. Tabel kebenaran untuk kedua operasi tersebut adalah sebagai berikut.

Tabel X. Tabel Kebenaran Operasi NTI

a	$NTI(a)$
-	+
0	-
+	-

Tabel XI. Tabel Kebenaran Operasi PTI

a	$PTI(a)$
-	+
0	+
+	-

2. OR

Sama dengan operasi OR pada logika Boolean, operasi OR dalam logika tiga-nilai menghasilkan nilai *benar* jika paling sedikit salah satu operannya bernilai *benar*, menghasilkan nilai *salah* jika semua operannya bernilai *salah*, dan menghasilkan nilai *tidak diketahui* untuk kasus lainnya. Tabel kebenaran untuk operasi ini adalah sebagai berikut:

Tabel XII. Tabel Kebenaran Operasi Ternary OR

a	b	$a \vee b$
-	-	-
-	0	0
-	+	+
0	-	0
0	0	0
0	+	+
+	-	+
+	0	+
+	+	+

Dapat dilihat bahwa dalam representasi bilangan, operasi OR menghasilkan nilai maksimum dari kedua nilai operan, sehingga operasi ini dapat juga disebut sebagai operasi MAX.

3. AND

Sama dengan operasi AND pada logika Boolean, operasi AND dalam logika tiga-nilai menghasilkan nilai *benar* jika paling sedikit kedua operannya bernilai *benar*, menghasilkan nilai *salah* jika ada operannya yang bernilai *salah*, dan menghasilkan nilai *tidak diketahui* untuk kasus lainnya. Tabel kebenaran untuk operasi ini adalah sebagai berikut:

Tabel XIII. Tabel Kebenaran Operasi Ternary AND

a	b	$a \wedge b$
-----	-----	--------------

-	-	-
-	0	-
-	+	-
0	-	-
0	0	0
0	+	0
+	-	-
+	0	0
+	+	+

Dapat dilihat bahwa dalam representasi bilangan, operasi AND menghasilkan nilai minimum dari kedua nilai operan, sehingga operasi ini dapat juga disebut sebagai operasi MIN.

Dalam logika boolean, semua jenis operasi dapat didefinisikan menggunakan tiga operasi dasar, yaitu NOT, OR, dan AND. Hal ini tidak berlaku dalam logika tiga-nilai. Sebagai contoh, *Ternary Adder* tidak dapat dirangkai hanya dengan menggunakan operasi STI, NTI, PTI, OR, dan AND. Operasi dasar tambahan yang biasanya didefinisikan selain kelima operasi di atas untuk menyelesaikan masalah ini adalah:

1. *Increment* dan *Decrement*

Kedua operasi ini menerima satu operan dan mengeluarkan hasil *increment* atau *decrement* dari nilai operan dalam representasi bilangan dalam modulo 3, sehingga nilai terbesar (*benar* / 1) jika dilakukan *increment* menghasilkan nilai terkecil (*salah* / -1), begitu juga sebaliknya. Operasi ini juga dapat disebut sebagai operasi *rotate up* dan *rotate down* atau *cycle* dan *inverse cycle*. Tabel kebenaran untuk kedua operasi ini adalah sebagai berikut.

Tabel XIV. Tabel Kebenaran Operasi *Increment* dan *Decrement*

<i>a</i>	<i>Increment(a)</i>	<i>Decrement(a)</i>
-	0	+
0	+	-
+	-	0

2. *Decoder*

Operasi ini menerima satu operan dan menghasilkan *benar* jika nilai operan sama dengan nilai yang dikehendaki, dan menghasilkan *salah* jika tidak. Terdapat tiga variasi operasi ini, masing-masing membandingkan nilai operan dengan tiga nilai yang telah terdefinisi dalam logika tiga-nilai (*benar*, *salah*, *tidak diketahui*). Tabel kebenaran untuk ketiga operasi ini adalah sebagai berikut.

Tabel XV. Tabel Kebenaran Operasi *Decoder*

<i>a</i>	<i>Is True(a)</i>	<i>Is Unknown(a)</i>	<i>Is False(a)</i>
-	-	-	+
0	-	+	-
+	+	-	-

Dapat dilihat bahwa operasi *Is False* ekuivalen dengan operasi NTI, dan operasi *Is True* merupakan negasi dari operasi PTI. Operasi *Is Unknown* dapat dinyatakan sebagai $Is\ Unknown(a) = - (Is\ True(a) \vee Is\ False(a))$.

D. Penerapan Logika Tiga-Nilai

1. Komputer Terner (*Ternary Computer*)

Menurut hukum Moore, jumlah transistor meningkat dua kali lipat tiap dua tahun. Sejalan dengan bertambahnya jumlah transistor, perkembangan teknologi juga membuat ukuran transistor semakin kecil sehingga perangkat elektronik menjadi lebih padu (*compact*). Suatu saat, ukuran transistor menjadi sangat kecil hingga mencapai batas ukuran minimum agar transistor bisa bekerja. Setelah mencapai batas ini, ukuran transistor tidak bisa diperkecil lagi. Salah satu solusi untuk meningkatkan kerapatan data adalah dengan mengganti sistem digital komputer dari yang menggunakan logika dua-nilai menjadi logika tiga-nilai. Komputer yang menggunakan logika tiga-nilai dalam sistem digitalnya disebut sebagai komputer terner (*ternary computer*). Alih-alih bit, ukuran data terkecil dalam komputer terner adalah trit (*ternary digit*). Satu trit memiliki tiga buah nilai logika, ukuran data yang dikandung oleh satu trit adalah $\log_2 3 = 1.58$ kali dari satu bit pada komputer biner.

Salah satu komputer terner yang pernah dibuat adalah Setun yang dikembangkan di Moscow State University mulai tahun 1956 dan selesai pada tahun 1958. Komputer ini menggunakan sistem bilangan terner seimbang dalam implementasinya.



Gambar 2. Komputer Terner Setun (Сетунь)

Sumber: <http://ternary.3neko.ru>

Pada saat komputer Setun dikembangkan, teknologi digital belum berkembang seperti sekarang. Kebanyakan komputer pada masa itu masih

menggunakan tabung vakum dan teknologi transistor masih sulit diakses. Sementara itu, tabung vakum sulit diandalkan sehingga komputer ini didesain menggunakan inti ferit dan dioda superkonduktor yang bekerja sebagai transformator arus terkontrol yang memungkinkan komputer ini menggunakan logika tiga-nilai. Dibandingkan dengan komputer biner pada masa itu, performa komputer Setun lebih cepat, lebih bisa diandalkan, dan lebih murah. Pada tahun 1970, dikembangkan komputer Setun 70 yang memperkenalkan konsep tryte (ekuivalen dengan byte pada komputer biner) yang setara dengan 6 trit (~9.5 bit).

Pada saat ini, teknologi transistor sudah sangat berkembang. Akan tetapi, konstruksi rangkaian untuk logika tiga-nilai menggunakan transistor lebih kompleks daripada rangkaian untuk logika boolean, sehingga kurang diminati. Meski demikian, terdapat beberapa penelitian yang meneliti desain rangkaian untuk mengimplementasi logika tiga-nilai dalam rangkaian elektronika, terutama desain transistor yang cocok untuk rangkaian ini.

Jenis-jenis transistor yang dapat digunakan untuk merangkai rangkaian logika tiga-nilai antara lain *metal-oxide-semiconductor field-effect transistor* (MOSFET), *carbon nanotube field-effect transistor* (CNTFET), dan *molybdenum disulfide field-effect transistor* (MoS₂ FET) dan *black phosphorus field-effect transistor* (BP FET). Gerbang logika NOT, AND, dan OR dalam logika tiga-nilai dapat dibuat menggunakan MOSFET dengan konfigurasi seperti rangkaian gerbang logika boolean, dengan tambahan resistor sebagai pembagi tegangan. Salah satu masalah yang timbul dari desain ini adalah resistor merupakan komponen pasif yang mendisipasi energi, sehingga desain dengan MOSFET kurang efisien dalam segi energi. Desain menggunakan CNTFET menghilangkan komponen resistor ini, dan menawarkan kelebihan antara lain resistansi *carbon nanotube* yang sangat kecil sehingga lebih efisien dan mampu bekerja dalam frekuensi tinggi. Kemudian, teknologi semikonduktor dua dimensi menawarkan alternatif baru dalam merancang rangkaian logika tiga-nilai, salah satunya dengan menggunakan BP FET bersamaan dengan MoS₂ FET. Desain ini mengurangi jumlah transistor yang dibutuhkan (Desain menggunakan CNTFET membutuhkan enam buah transistor untuk membangun sebuah gerbang logika NOT).

Kelebihan dari komputer terner adalah seperti yang telah disebutkan, mampu memuat data lebih banyak dalam ukuran ruang yang sama dibanding dengan komputer biner. Hal ini berarti dalam mengirim data, misalnya, untuk ukuran data tertentu, komputer terner dapat mengirim data melalui *bus* dengan lebar (jumlah

jalur dalam *bus*) lebih sedikit dibandingkan dengan komputer biner. Jika komputer terner tersebut menggunakan sistem bilangan terner seimbang, tidak memerlukan sebuah konvensi untuk data *signed integer* (seperti *Two's complement* pada komputer biner) karena sudah direpresentasikan oleh nilai -1 pada digit paling signifikan.

2. Basis Data SQL

Structured Query Language mengenal nilai kebenaran *unknown* di samping nilai *true* dan *false*. Hal ini merupakan konsekuensi dari adanya tipe data *null*. *null* menyatakan tidak adanya data atau data yang belum terdefinisi. Dalam membandingkan nilai, *null* bisa diartikan apa saja. Oleh karena itu, jika sebuah data dibandingkan dengan *null*, tidak bisa dipastikan apakah nilai *null* sama dengan nilai data yang dibandingkan atau tidak, sehingga hasil perbandingan bernilai *unknown*.

IV. KESIMPULAN

Logika tiga-nilai merupakan perluasan dari logika boolean dan memiliki banyak kemiripan dengan logika boolean. Logika tiga-nilai memiliki aplikasi dalam beberapa bidang dan dapat menjadi alternatif baru dalam menyelesaikan masalah yang sebelumnya diselesaikan menggunakan logika boolean.

Dibangunnya komputer terner menunjukkan mungkinnya penerapan logika tiga-nilai sebagai dasar dari sistem digital komputer beserta keunggulannya. Penerapan logika tiga-nilai dalam komputer terner dapat menjadi solusi untuk mengatasi limitasi dari ukuran transistor serta kompleksitas sirkuit pada komputer biner.

V. UCAPAN TERIMA KASIH

Penulis menyampaikan rasa syukur kepada Tuhan Yang Maha Esa serta berterima kasih kepada dosen pengampu mata kuliah IF2120 Matematika Diskrit serta semua pihak yang telah mendukung dan membantu penulis dalam menyelesaikan makalah ini.

REFERENSI

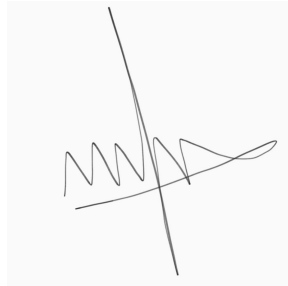
- [1] Abdurrohman, Maman. (2017). *Organisasi dan Arsitektur Komputer edisi Revisi Keempat*. Bandung: Penerbit Informatika.
- [2] Shelburne, Brian J. (2006). *Balanced (Signed) Ternary Notation*.
- [3] <http://homepage.divms.uiowa.edu/~jones/ternary/logic.shtml>. Diakses 5 Desember 2019.
- [4] <https://medium.com/@rxseger/exploring-ternary-logic-building-a-simple-ternary-inverter-using-complementary-mosfets-c997f2f87c4d>. Diakses 5 Desember 2019.
- [5] <http://www.computer-museum.ru/english/setun.htm>. Diakses 5 Desember 2019.
- [6] K Sasi Saketh, P. Reena Monica. (2013). *Ternary logic implementation and its application using CNTFET*.
- [7] Mingqiang Huang, Xingli Huang, Guangchao Zhao, Philippe Coquet, Bengkang Tay. (2019). *Design and Implementation of Ternary Logic Integrated Circuits by Using Novel Two-Dimensional Materials*.

[8] <https://modern-sql.com/concept/three-valued-logic>. Diakses 5 Desember 2019.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 6 Desember 2019

A square box containing a handwritten signature in black ink. The signature is stylized and appears to be 'Muhammad Naufal Fakhrizal'.

Muhammad Naufal Fakhrizal 13518115