

# Penyederhanaan Aljabar Boolean dengan Komputasi DNA

Jesica 13519011

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

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

13519011@std.stei.itb.ac.id

**Abstract**—Dalam penyederhanaan ekspresi boolean, terdapat berbagai macam metode yang mungkin sudah tidak asing untuk didengar. Seperti dengan memanfaatkan hukum-hukum yang berlaku dalam aljabar boolean atau dengan peta karnaugh. Namun selain menggunakan metode yang telah disebutkan sebelumnya, penyederhanaan ekspresi boolean dapat juga dilakukan dengan metode yang berhubungan dengan bidang ilmu biologi ataupun kimia. Melalui makalah ini akan dibahas salah satu metode penyederhaan ekspresi boolean yaitu dengan komputasi DNA.

**Keywords**—aljabar, boolean, biokimia, DNA, komputasi, minterm.

## I. PENDAHULUAN

DNA merupakan salah satu ilmu yang dipelajari di bidang biologi. Secara umum, DNA (*Deoxyribonucleic Acid*) atau asam deoksiribonukleat adalah sejenis asam nukleat yang tergolong biomolekul utama yang menyimpan instruksi genetika setiap organisme dan juga virus. Letak DNA ada di dalam sel, DNA juga menentukan sifat organisme yang diturunkannya. Mulai dari warna mata sampai warna kulit, DNA memprogram keseluruhan tubuh manusia. Komputasi DNA adalah bidang ilmu yang menggunakan molekul biologis sebagai komponen dasar dari perangkat komputasi. Komputasi DNA dapat digunakan untuk menyelesaikan sebuah permasalahan yang tidak bisa diselesaikan oleh komputer klasik.

Komputasi DNA pertama kali ditemukan pada tahun 1994, oleh seorang profesor dari *University of Southern California*, Leonard Adleman. Setelah mengetahui struktur dari DNA, ia terinspirasi untuk menulis makalah yang menunjukkan bahwa DNA dapat digunakan untuk memecahkan persoalan “*Travelling Salesman Problem*”.

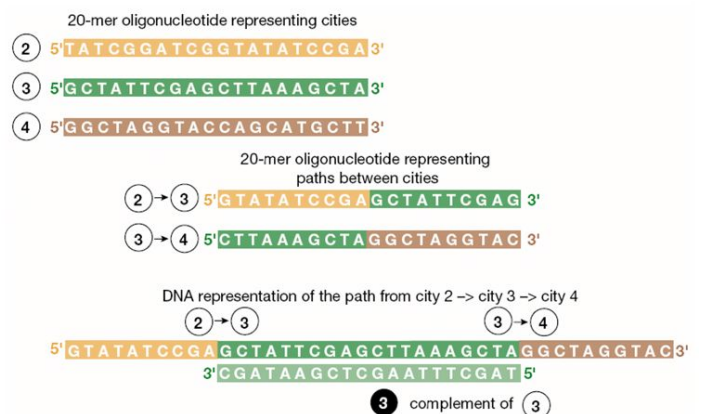
Melalui makalah ini, penulis ingin menunjukkan bahwa terdapat metode alternatif untuk menyederhanakan fungsi boolean, yaitu dengan komputasi DNA. Selain dapat memecahkan persoalan *travelling salesman problem*, komputasi DNA bisa digunakan untuk menyederhanakan fungsi boolean dan mungkin kedepannya juga bisa digunakan untuk menyelesaikan persoalan matematika lain yang lebih kompleks.

## II. LANDASAN TEORI

### A. Komputasi DNA

Seperti yang telah disinggung sebelumnya bahwa metode komputasi DNA dapat digunakan untuk memecahkan persoalan *travelling salesman problem*. *Travelling salesman problem* sendiri adalah sebuah persoalan dimana terdapat sebuah perusahaan yang memiliki seorang *salesman* yang harus mengunjungi sejumlah  $n$  kota dan hanya dapat mengunjungi setiap kota masing-masing satu kali. *Travelling salesman problem* ini adalah bagaimana cara agar *salesman* tersebut memperoleh rute perjalanan terpendek yang melalui setiap kota agar biaya yang dikeluarkan sesedikit mungkin.

Ketika nilai dari  $n$  masih terhitung kecil, misalnya ketika  $n$  bernilai 5, tentu persoalan *travelling salesman problem* ini bukanlah sesuatu yang sulit untuk diselesaikan. Namun bagaimana dengan nilai  $n$  yang besar? Tentu persoalan ini akan sulit diselesaikan secara komputasi dan waktu yang dibutuhkan pun akan membesar secara eksponensial. Dalam kasus ini, Adleman dapat mendemonstrasikan bahwa DNA dapat disusun sedemikian rupa sehingga sebuah tabung reaksi yang penuh dengan blok DNA dapat menyusun dirinya sendiri yang kemudian menghasilkan semua rute yang mungkin untuk memecahkan persoalan *travelling salesman problem*.



**Gambar 2.1** Ilustrasi penggunaan komputasi DNA dalam pencarian rute antar kota, diambil dari [4]

Dalam DNA, pengkodean genetik direpresentasikan dengan empat molekul berbeda yang dikenal dengan adenin (A), timin (T), sitosin (C) dan guanin (G). Ketika keempat “bits” ini

disusun menjadi satu, maka dapat disimpan data dengan jumlah yang sangat besar seperti sebagaimana genom manusia yang disusun dan dikemas menjadi suatu kesatuan yang disebut dengan satu inti sel. Dengan mencampurkan keempat molekul yang telah disebutkan sebelumnya ke dalam sebuah tabung reaksi, molekul-molekul tersebut secara alami menyusun dirinya menjadi untaian DNA. Jika beberapa kombinasi molekul ini dapat merepresentasikan kota dan rute penerbangan, maka setiap untaian DNA dapat merepresentasikan rute penerbangan yang berbeda untuk *salesman*, semuanya itu dihitung sekaligus dalam sintesis untaian DNA yang dapat menyusun dirinya sendiri secara paralel. Kemudian hanya perlu dilakukan eliminasi terhadap rute yang lebih panjang sampai diperoleh rute yang paling pendek.

### B. Aljabar Boolean

Aljabar boolean pertama kali ditemukan pada tahun 1854 oleh seorang ilmuwan berkebangsaan Inggris, George Boole. Penamaan Aljabar boolean sendiri diambil dari nama George Boole. Perbedaan aljabar boolean dengan aljabar pada umumnya adalah aljabar boolean berhubungan dengan variabel biner. Variabel boolean yang paling umum adalah variabel yang bernilai 1 (merepresentasikan *true*) atau bernilai 0 (merepresentasikan *false*). Operator dalam aljabar boolean terdiri dari 2 jenis, yaitu operator biner dan operator uner.

<i>a</i>	<i>b</i>	<i>a · b</i>	<i>a</i>	<i>b</i>	<i>a + b</i>
0	0	0	0	0	0
0	1	0	0	1	1
1	0	0	1	0	1
1	1	1	1	1	1

Tabel 2.1 Contoh pengoperasian dengan operator biner, diambil dari [3]

<i>a</i>	<i>a'</i>
0	1
1	0

Tabel 2.2 Contoh pengoperasian dengan operator uner, diambil dari [3]

1. Hukum identitas: (i) $a + 0 = a$ (ii) $a \cdot 1 = a$	2. Hukum idempoten: (i) $a + a = a$ (ii) $a \cdot a = a$
3. Hukum komplemen: (i) $a + a' = 1$ (ii) $aa' = 0$	4. Hukum dominansi: (i) $a \cdot 0 = 0$ (ii) $a + 1 = 1$
5. Hukum involusi: (i) $(a')' = a$	6. Hukum penyerapan: (i) $a + ab = a$ (ii) $a(a + b) = a$
7. Hukum komutatif: (i) $a + b = b + a$ (ii) $ab = ba$	8. Hukum asosiatif: (i) $a + (b + c) = (a + b) + c$ (ii) $a(b c) = (a b) c$
9. Hukum distributif: (i) $a + (b c) = (a + b)(a + c)$ (ii) $a(b + c) = ab + ac$	10. Hukum De Morgan: (i) $(a + b)' = a' b'$ (ii) $(ab)' = a' + b'$
11. Hukum 0/1 (i) $0' = 1$ (ii) $1' = 0$	

Tabel 2.3 Beberapa hukum yang berlaku dalam aljabar boolean, diambil dari [3]

Dalam aljabar boolean, fungsi boolean dapat diekspresikan dalam bentuk *minterm* ataupun *maxterm*. Adapun perbedaan dari *minterm* dan *maxterm* ini, yaitu pada *minterm* suku (*term*) di dalam ekspresi boolean mengandung literal yang lengkap dalam bentuk hasil kali. Sedangkan pada *maxterm* suku (*term*) di dalam ekspresi boolean mengandung literal yang lengkap dalam bentuk hasil jumlah. Ekspresi boolean yang dinyatakan sebagai penjumlahan dari satu atau lebih *minterm* atau perkalian dari satu atau lebih *maxterm* disebut bentuk kanonik. Ada dua macam bentuk kanonik, yaitu penjumlahan dari hasil kali atau dikenal dengan SOP (*sum-of-product*) dan perkalian dari hasil jumlah atau dikenal dengan POS (*product-of-sum*).

Metode dalam pembentukan *minterm* dan *maxterm* juga berbeda. Untuk *minterm*, setiap peubah yang bernilai 0 dinyatakan dalam bentuk komplemen, sedangkan peubah yang bernilai 1 dinyatakan tanpa komplemen. Sebaliknya, untuk *maxterm*, setiap peubah yang bernilai 0 dinyatakan tanpa komplemen, sedangkan peubah yang bernilai 1 dinyatakan dalam bentuk komplemen.

<i>x</i>	<i>y</i>	<i>Minterm</i>		<i>Maxterm</i>	
		Suku	Lambang	Suku	Lambang
0	0	$x'y'$	$m_0$	$x + y$	$M_0$
0	1	$x'y$	$m_1$	$x + y'$	$M_1$
1	0	$xy'$	$m_2$	$x' + y$	$M_2$
1	1	$xy$	$m_3$	$x' + y'$	$M_3$

Tabel 2.4 Cara membentuk *minterm* dan *maxterm* dari tabel kebenaran untuk dua peubah, diambil dari [3]

x	y	z	Minterm		Maxterm	
			Suku	Lambang	Suku	Lambang
0	0	0	$x'y'z'$	$m_0$	$x+y+z$	$M_0$
0	0	1	$x'y'z$	$m_1$	$x+y+z'$	$M_1$
0	1	0	$x'yz'$	$m_2$	$x+y'+z$	$M_2$
0	1	1	$x'yz$	$m_3$	$x+y'+z'$	$M_3$
1	0	0	$xy'z'$	$m_4$	$x'+y+z$	$M_4$
1	0	1	$xy'z$	$m_5$	$x'+y+z'$	$M_5$
1	1	0	$xyz'$	$m_6$	$x'+y'+z$	$M_6$
1	1	1	$xyz$	$m_7$	$x'+y'+z'$	$M_7$

**Tabel 2.5** Cara membentuk *minterm* dan *maxterm* dari tabel kebenaran untuk tiga peubah, diambil dari [3]

Menyederhanakan fungsi boolean artinya mencari bentuk fungsi lain dengan jumlah literal atau operasi yang lebih sedikit namun tetap ekuivalen dengan fungsi sebelumnya. Dengan bentuk yang lebih sederhana ini maka gerbang logika yang digunakan juga menjadi lebih sedikit.

### III. LANGKAH PENYEDERHANAAN DENGAN KOMPUTASI DNA

Langkah pertama, bentuk  $4n$  kelompok oligonukleotida yang dibagi menjadi empat kelompok. Dengan  $n$  adalah jumlah peubah yang terdapat pada sebuah ekspresi boolean. Oligonukleotida dari kelompok pertama merepresentasikan variabel  $x_1, x_2, x_3, \dots, x_n$ . Kemudian oligonukleotida dari kelompok kedua merepresentasikan variabel  $x_1^-, x_2^-, x_3^-, \dots, x_n^-$  ( $x = 1$  jika dan hanya jika  $x^- = 0$ ). Oligonukleotida kelompok ketiga dan keempat merepresentasikan komplemen untaian dari kelompok pertama ( $x_1', x_2', x_3', \dots, x_n'$ ) dan kelompok kedua ( $x_1^-, x_2^-, x_3^-, \dots, x_n^-$ ).

Langkah kedua, Bentuk  $2^n$  kombinasi untaian tunggal molekul DNA untuk  $n$  variabel dari oligonukleotida dari kelompok ketiga dan kelompok keempat.

Langkah ketiga, gabungkan oligonukleotida dari kelompok pertama dan kelompok kedua sesuai dengan persamaan minterm yang ingin disederhanakan.

Langkah keempat, gabungkan  $2^n$  kombinasi oligonukleotida dari hasil langkah kedua dengan gabungan nukleotida dari hasil langkah ketiga. Untuk setiap untaian yang tidak memiliki pasangan akan dibuang.

Langkah kelima, setiap untaian yang berpasangan didenaturasi dengan cara setiap untaian oligonukleotida yang merepresentasikan komplemen dari pasangannya akan tetap berada di tabung reaksi sedangkan sisanya akan dipisahkan.

Langkah keenam, dua minterm dapat digabungkan jika terdapat perbedaan tepat satu literal. Contohnya, 000 yang dapat digabungkan dengan 001, 010, 100, dan seterusnya. Maka untaian komplemen dari minterm yang telah diperoleh dari langkah kelima dapat di-*amplified* atau diperkuat sejumlah  $n$  kali, jika sebuah minterm dapat digabungkan dengan  $n$  minterm.

Langkah ketujuh, bentuk kombinasi untaian dari  $(n-1)$  variabel dan pisahkan untaian tersebut ke  $n$  tabung reaksi yang berbeda. Contohnya, 001 yang memiliki 3 kombinasi seperti 00, 01 dan 10 dipisahkan di 3 tabung reaksi yang berbeda.

Langkah kedelapan, gabungkan dua minterm yang memiliki perbedaan tepat satu literal seperti langkah keenam. Contohnya seperti 001 yang memiliki 3 kombinasi 00, 01 dan 10 dipisahkan ke 3 tabung reaksi yang berbeda. Kemudian di 3

tabung reaksi tersebut gabungkan komplemen dari 001 yaitu  $0'0'1'$ .

Langkah kesembilan, simpan untaian DNA yang merupakan komplemen dari bentuk asli yang terdapat dari langkah sebelumnya.

Langkah kesepuluh, simpan hasil komplemennya dan periksa apakah sudah diperoleh bentuk yang paling sederhana atau belum (masih ada untaian yang sama atau tidak).

Langkah kesebelas, ulangi kembali langkah ketujuh sampai dengan langkah kesepuluh sampai diperoleh bentuk fungsi yang paling sederhana.

Untuk ilustrasi yang lebih jelas dapat dilihat pada contoh penyederhanaan fungsi boolean yang terdapat di bagian berikutnya.

### IV. CONTOH PENYEDERHANAAN FUNGSI BOOLEAN

Sebagai contoh, diberikan ekspresi boolean dengan tiga peubah sebagai berikut.

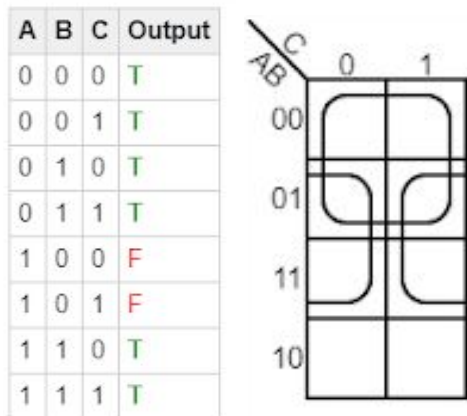
$$Y = \bar{A} \bar{B} \bar{C} + \bar{A} B \bar{C} + \bar{A} B C + A \bar{B} \bar{C} + A B \bar{C} + A B C$$

Ekspresi boolean tersebut akan disederhanakan dengan 3 metode berbeda yaitu, dengan hukum-hukum aljabar boolean, peta karnaugh dan komputasi DNA.

#### A. Dengan Hukum-Hukum yang Berlaku dalam Aljabar Boolean

$$\begin{aligned}
 Y &= \bar{A} \bar{B} \bar{C} + \bar{A} B \bar{C} + \bar{A} B C + A \bar{B} \bar{C} + A B \bar{C} + A B C \\
 &= \bar{A} \bar{B} \bar{C} + \bar{A} B \bar{C} + \bar{A} B C + A \bar{B} \bar{C} + A B \bar{C} + A B C \\
 &\text{(Hukum komutatif)} \\
 &= \bar{A} \bar{B} + \bar{A} B \bar{C} + A B \bar{C} + A \bar{B} \bar{C} + A B C \\
 &\text{(Hukum redundansi)} \\
 &= \bar{A} \bar{B} + B \bar{C} A + B \bar{C} \bar{A} + \bar{A} B C + A B C \\
 &\text{(Hukum komutatif)} \\
 &= \bar{A} \bar{B} + B \bar{C} + \bar{A} B C + A B C \\
 &\text{(Hukum redundansi)} \\
 &= \bar{A} \bar{B} + B \bar{C} + B \bar{C} A + B \bar{C} \bar{A} \\
 &\text{(Hukum komutatif)} \\
 &= \bar{A} \bar{B} + B \bar{C} + B \bar{C} \\
 &\text{(Hukum redundansi)} \\
 &= \bar{A} \bar{B} + B(\bar{C} + C) \\
 &\text{(Hukum distributif)} \\
 &= \bar{A} \bar{B} + B 1 \\
 &\text{(Hukum komplemen)} \\
 &= \bar{A} \bar{B} + B \\
 &\text{(Hukum identitas)} \\
 &= \bar{A} B \\
 &\text{(Hukum redundansi)}
 \end{aligned}$$

B. Dengan Peta Karnaugh



Gambar 4.1 Peta karnaugh dari persoalan di atas.

C. Dengan Komputasi DNA

Langkah pertama, bentuk 4n kelompok oligonukleotida dengan n bernilai 3 sehingga diperoleh 12 kelompok oligonukleotida. Dengan oligonukleotida dari tiap kelompok merepresentasikan variabel sebagai berikut.

- Kelompok pertama :  $A, B, C$
- Kelompok kedua :  $A', B', C'$
- Kelompok ketiga :  $A, B, C'$
- Kelompok keempat :  $A', B', C$

Langkah kedua, buat  $2^3$  kombinasi dari kelompok oligonukleotida ketiga dan keempat.

$$\begin{matrix} A' B' C' & A' B' C & A' B' C' & A' B' C' \\ A' B' C' & A' B' C & A' B' C' & A' B' C' \end{matrix}$$

Langkah ketiga, gabungkan oligonukleotida dari kelompok pertama dan kelompok kedua sesuai dengan persamaan minterm yang ingin disederhanakan atau bisa dilakukan dengan menuliskan kembali persamaan tersebut sebagai berikut.

$$Y = ABC + \bar{A}BC + A\bar{B}C + \bar{A}\bar{B}C + ABC + \bar{A}BC$$

$$\begin{matrix} ABC & \bar{A}BC & A\bar{B}C \\ \bar{A}BC & \bar{A}BC & \bar{A}BC \end{matrix}$$

Langkah keempat, gabungkan hasil dari langkah kedua dengan hasil dari langkah ketiga

$ABC$ $\bar{A}'B'C'$ $T_0$	$\bar{A}BC$ $\bar{A}'B'C'$ $T_1$	$A\bar{B}C$ $\bar{A}'B'C'$ $T_2$	$\bar{A}BC$ $\bar{A}'B'C'$ $T_3$
$A'B'C'$	$A'B'C'$	$ABC$ $A'B'C'$	$ABC$ $A'B'C'$

$T_4$	$T_5$	$T_6$	$T_7$
-------	-------	-------	-------

Tabel 4.1 Hasil dari penggabungan oligonukleotida dari kelompok ketiga dan kelompok keempat, diambil dari [5].

$ABC$ $\bar{A}'B'C'$ $T_0$	$\bar{A}BC$ $\bar{A}'B'C'$ $T_1$	$A\bar{B}C$ $\bar{A}'B'C'$ $T_2$	$\bar{A}BC$ $\bar{A}'B'C'$ $T_3$
		$ABC$ $A'B'C'$ $T_6$	$ABC$ $A'B'C'$ $T_7$

Tabel 4.2 Untaian oligonukleotida yang tidak berpasangan dihapus dari tabel, diambil dari [5].

Langkah kelima, untaian yang berpasangan didenaturasi dan oligonukleotida yang bukan merupakan komplementer dipisahkan dari tabung reaksi.

$\bar{A}'B'C'$ $T_0$	$\bar{A}'B'C'$ $T_1$	$\bar{A}'B'C'$ $T_2$	$\bar{A}'B'C'$ $T_3$
		$A'B'C'$ $T_6$	$A'B'C'$ $T_7$

Tabel 4.2 Untaian oligonukleotida yang bukan komplementer dihapus dari tabel, diambil dari [5].

Langkah keenam, dua minterm yang berbeda tepat 1 literal digabungkan seperti  $T_0 - T_1, T_0 - T_2, T_1 - T_3, T_2 - T_3, T_2 - T_6, T_3 - T_7$ , dan  $T_6 - T_7$ . Maka setiap untaian komplemen yang diperoleh dari langkah lima dapat di-amplified sebanyak 2 kali.

$$\begin{aligned} &AMPLIFY (T_0, T_0', T_0'', T_{00}, T_{00}', T_{00}'', T_{000}, T_{000}', T_{000}'') \\ &AMPLIFY (T_1, T_1', T_1'', T_{10}, T_{10}', T_{10}'', T_{100}, T_{100}', T_{100}'') \\ &AMPLIFY (T_2, T_2', T_2'', T_{20}, T_{20}', T_{20}'', T_{200}, T_{200}', T_{200}'') \\ &AMPLIFY (T_3, T_3', T_3'', T_{30}, T_{30}', T_{30}'', T_{300}, T_{300}', T_{300}'') \\ &AMPLIFY (T_6, T_6', T_6'', T_{60}, T_{60}', T_{60}'', T_{600}, T_{600}', T_{600}'') \\ &AMPLIFY (T_7, T_7', T_7'', T_{70}, T_{70}', T_{70}'', T_{700}, T_{700}', T_{700}'') \end{aligned}$$

Oligonukleotida	Oligonukleotida	Hasil penggabungan	
$\bar{A}'B'C'$ $T_0$	$\bar{A}'B'C'$ $T_1$	$\bar{A}'B'C'$ $T_0'$	$\bar{A}'B'C'$ $T_1'$
$\bar{A}'B'C'$ $T_0''$	$\bar{A}'B'C'$ $T_2$	$\bar{A}'B'C'$ $T_0''$	$\bar{A}'B'C'$ $T_2''$
$\bar{A}'B'C'$ $T_1'$	$\bar{A}'B'C'$ $T_3$	$\bar{A}'B'C'$ $T_1''$	$\bar{A}'B'C'$ $T_3'$

$\bar{A}'B'C'$ $T_2'$	$\bar{A}'B'C'$ $T_3'$	$\bar{A}'B'C'$ $T_2''$	$\bar{A}'B'C'$ $T_3''$
$\bar{A}'B'C'$ $T_2''$	$A'B'C'$ $T_6$	$\bar{A}'B'C'$ $T_2''$	$A'B'C'$ $T_6'$
$\bar{A}'B'C'$ $T_3''$	$A'B'C'$ $T_7$	$\bar{A}'B'C'$ $T_3''$	$A'B'C'$ $T_7'$

**Tabel 4.3** Hasil kombinasi dari 2 minterm yang berbeda tepat 1 literal, diambil dari [5].

Langkah ketujuh, bentuk untaian baru dari 2 variabel dan pisahkan dalam 3 tabung reaksi yang berbeda. Kemudian *amplify* tabung reaksi tersebut sesuai dengan minterm di dalamnya.

$T_0, T_0'$ $T_0''$	$T_1, T_1'$ $T_1''$	$T_2, T_2'$ $T_2''$	$T_3, T_3'$ $T_3''$	$T_6, T_6'$ $T_6''$	$T_7, T_7'$ $T_7''$
$AC$	$AC$	$AC$	$AC$	$AC$	$AC$
$BC$	$BC$	$BC$	$BC$	$BC$	$BC$
$AB$	$AB$	$AB$	$AB$	$AB$	$AB$
$T_8$	$T_9$	$T_{10}$	$T_{11}$	$T_{12}$	$T_{13}$

**Tabel 4.4** Hasil kombinasi dari 2 variabel, diambil dari [5].

SEPARATE ( $T_8, T_{80}, T_{800}$ )  
 AMPLIFY ( $T_8, T_8', T_8'', T_{80}, T_{80}', T_{80}'', T_{800}, T_{800}', T_{800}''$ )  
 $[T_8: AB, T_{80}: BC, T_{800}: AC]$   
 SEPARATE ( $T_9, T_{90}, T_{900}$ )  
 AMPLIFY ( $T_9, T_9', T_9'', T_{90}, T_{90}', T_{90}'', T_{900}, T_{900}', T_{900}''$ )  
 $[T_9: BC, T_{90}: AB, T_{900}: AC]$   
 SEPARATE ( $T_{10}, T_{100}, T_{1000}$ )  
 AMPLIFY ( $T_{10}, T_{10}', T_{10}'', T_{100}, T_{100}', T_{100}'', T_{1000}, T_{1000}', T_{1000}''$ )  
 SEPARATE ( $T_{11}, T_{110}, T_{1100}$ )  
 AMPLIFY ( $T_{11}, T_{11}', T_{11}'', T_{110}, T_{110}', T_{110}'', T_{1100}, T_{1100}', T_{1100}''$ )  
 SEPARATE ( $T_{12}, T_{120}, T_{1200}$ )  
 AMPLIFY ( $T_{12}, T_{12}', T_{12}'', T_{120}, T_{120}', T_{120}'', T_{1200}, T_{1200}', T_{1200}''$ )  
 SEPARATE ( $T_{13}, T_{130}, T_{1300}$ )  
 AMPLIFY ( $T_{13}, T_{13}', T_{13}'', T_{130}, T_{130}', T_{130}'', T_{1300}, T_{1300}', T_{1300}''$ )

Langkah kedelapan, gabungkan kembali kedua minterm yang berbeda tepat 1 literal.

**Bandingkan  $T_0-T_1$**

MERGE $T_0-T_8$ $\bar{A}'B'C'$ $AB$	MERGE $T_{00}-T_{80}$ $\bar{A}'B'C'$ $BC$	MERGE $T_{000}-T_{800}$ $\bar{A}'B'C'$ $AC$
---	---	---

**Tabel 4.5** Hasil penggabungan  $T_0-T_8, T_{00}-T_{80}, T_{000}-T_{800}$  untuk tabung reaksi  $T_0$ , diambil dari [5].

MERGE $T_1-T_9$ $\bar{A}'B'C'$	MERGE $T_{10}-T_{90}$ $\bar{A}'B'C'$	MERGE $T_{100}-T_{900}$ $\bar{A}'B'C'$
-----------------------------------	---	---

$BC$	$AB$	$AC$
------	------	------

**Tabel 4.5** Hasil penggabungan  $T_1-T_9, T_{10}-T_{90}, T_{100}-T_{900}$  untuk tabung reaksi  $T_1$ , diambil dari [5].

Kemudian hasil yang diperoleh dari perbandingan  $T_0$  dengan  $T_1$  adalah  $AB$  karena  $AB$  terdapat di kedua tabung atau tabel di atas. Lakukan hal yang sama pada oligonukleotida sisanya dan diperoleh hasil seperti di bawah ini.

**Bandingkan  $T_0'-T_2$**

MERGE [ $T_0'-T_8', T_{00}'-T_{80}', T_{000}'-T_{800}'$ ](Untuk tabung  $T_0'$ )  
 MERGE [ $T_2-T_{10}, T_{20}-T_{100}, T_{200}-T_{1000}$ ](Untuk tabung  $T_2$ )  
 Hasil yang diperoleh adalah  $AC$ .

**Bandingkan  $T_1'-T_3$**

MERGE [ $T_1'-T_9', T_{10}'-T_{90}', T_{100}'-T_{900}'$ ](Untuk tabung  $T_1'$ )  
 MERGE [ $T_3-T_{11}, T_{30}-T_{110}, T_{300}-T_{1100}$ ](Untuk tabung  $T_3$ )  
 Hasil yang diperoleh adalah  $AC$ .

**Bandingkan  $T_2'-T_3'$**

MERGE [ $T_2'-T_{10}', T_{20}'-T_{100}', T_{200}'-T_{1000}'$ ](Untuk tabung  $T_2'$ )  
 MERGE [ $T_3-T_{11}, T_{30}-T_{110}, T_{300}-T_{1100}$ ](Untuk tabung  $T_3$ )  
 Hasil yang diperoleh adalah  $AB$ .

**Bandingkan  $T_2''-T_6'$**

MERGE [ $T_2''-T_{10}'', T_{20}''-T_{100}'', T_{200}''-T_{1000}''$ ](Untuk tabung  $T_2''$ )  
 MERGE [ $T_6'-T_{12}', T_{60}'-T_{120}', T_{600}'-T_{1200}'$ ](Untuk tabung  $T_6'$ )  
 Hasil yang diperoleh adalah  $BC$ .

**Bandingkan  $T_3''-T_7$**

MERGE [ $T_3''-T_{11}'', T_{30}''-T_{110}'', T_{300}''-T_{1100}''$ ](Untuk tabung  $T_3''$ )  
 MERGE [ $T_7-T_{13}, T_{70}-T_{130}, T_{700}-T_{1300}$ ](Untuk tabung  $T_7$ )  
 Hasil yang diperoleh adalah  $BC$ .

**Bandingkan  $T_6''-T_7'$**

MERGE [ $T_6''-T_{12}'', T_{60}''-T_{120}'', T_{600}''-T_{1200}''$ ](Untuk tabung  $T_6''$ )  
 MERGE [ $T_7-T_{13}, T_{70}-T_{130}, T_{700}-T_{1300}$ ](Untuk tabung  $T_7$ )  
 Hasil yang diperoleh adalah  $AB$ .

Langkah kesembilan, simpan untaian DNA yang merupakan komplemen dari bentuk asli yang terdapat dari langkah sebelumnya.

Bentuk asli :

$$AB + AC + AC + AB + BC + BC + AB$$

Komplemen :

$$\bar{A}'B' + \bar{A}'C' + \bar{A}'C' + \bar{A}'B' + B'C' + B'C' + A'B'$$

Langkah kesepuluh, simpan hasil komplemennya dan periksa apakah sudah diperoleh bentuk yang paling sederhana atau belum (masih ada untaian yang sama atau tidak). Dalam kasus ini hasil yang diperoleh bukan merupakan bentuk yang paling sederhana sehingga diulangi kembali langkah sebelumnya.

$\bar{A}'C'$ $T_0$	$\bar{A}'C'$ $T_1$	$B'C'$ $T_2$



$\bar{A} \bar{B}'$ $T_0'$	$\bar{A} B'$ $T_1'$	$A \bar{B}'$ $T_2'$
	$B' C'$ $T_1''$	

**Tabel 4.6** Bentuk komplemen yang diperoleh dari langkah kesembilan, diambil dari [5].

Langkah kesebelas, mirip seperti langkah ketujuh, bentuk untaian baru dari 1 variabel dan pisahkan dalam tabung reaksi yang berbeda.

$T_0$	$T_0'$	$T_1$	$T_1'$	$T_2$	$T_2'$
$\bar{A}$	$B$	$C$	$B$	$C$	$A$
$C$	$\bar{A}$	$\bar{A}$	$\bar{A}$	$B$	$B$
$T_3$	$T_4$	$T_5$	$T_6$	$T_7$	$T_8$

**Tabel 4.7** Kombinasi dari 1 variabel, diambil dari [5].

Langkah kedua belas, gabungkan 2 minterm yang berbeda tepat 1 bit. Dari contoh diatas dapat kita bandingkan,  $T_0-T_1$ ,  $T_0-T_1'$ ,  $T_0-T_1''$ ,  $T_0'-T_1$ ,  $T_0'-T_1'$ ,  $T_1-T_2$ ,  $T_1''-T_2$ ,  $T_1-T_2$ . Kemudian *amplify* untaian komplemen minterm yang diperoleh dari langkah kesembilan.

*AMPLIFY* ( $T_0, T_{01}, T_{02}, T_{03}, T_{04}, T_{05}, T_{06}, T_{07}, T_{08}$ )  
*AMPLIFY* ( $T_0', T_{01}', T_{02}', T_{03}', T_{04}', T_{05}', T_{06}', T_{07}', T_{08}'$ )  
*AMPLIFY* ( $T_1, T_{11}, T_{12}, T_{13}, T_{14}, T_{15}, T_{16}, T_{17}, T_{18}$ )  
*AMPLIFY* ( $T_1', T_{11}', T_{12}', T_{13}', T_{14}', T_{15}', T_{16}', T_{17}', T_{18}'$ )  
*AMPLIFY* ( $T_1'', T_{11}'', T_{12}'', T_{13}'', T_{14}'', T_{15}'', T_{16}'', T_{17}'', T_{18}''$ )  
*AMPLIFY* ( $T_2, T_{21}, T_{22}, T_{23}, T_{24}, T_{25}, T_{26}, T_{27}, T_{28}$ )  
*AMPLIFY* ( $T_2', T_{21}', T_{22}', T_{23}', T_{24}', T_{25}', T_{26}', T_{27}', T_{28}'$ )

Langkah ketiga belas, bentuk untaian gabungan dari 1 variabel dan pisahkan di 2 tabung reaksi yang berbeda dan *amplify* sesuai dengan minterm nya.

*SEPARATE* ( $T_3, T_{30}$ )  
*AMPLIFY* ( $T_3, T_3', T_3'', T_{30}, T_{30}', T_{30}''$ )  
*SEPARATE* ( $T_4, T_{40}$ )  
*AMPLIFY* ( $T_4, T_4', T_4'', T_{40}, T_{40}', T_{40}''$ )  
*SEPARATE* ( $T_5, T_{50}$ )  
*AMPLIFY* ( $T_5, T_5', T_5'', T_{50}, T_{50}', T_{50}''$ )  
*SEPARATE* ( $T_6, T_{60}$ )  
*AMPLIFY* ( $T_6, T_6', T_6'', T_{60}, T_{60}', T_{60}''$ )  
*SEPARATE* ( $T_7, T_{70}$ )  
*AMPLIFY* ( $T_7, T_7', T_7'', T_{70}, T_{70}', T_{70}''$ )  
*SEPARATE* ( $T_8, T_{80}$ )  
*AMPLIFY* ( $T_8, T_8', T_8'', T_{80}, T_{80}', T_{80}''$ )  
*SEPARATE* ( $T_9, T_{90}$ )  
*AMPLIFY* ( $T_9, T_9', T_9'', T_{90}, T_{90}', T_{90}''$ )

Sekarang 2 minterm yang berbeda tepat 1 literal telah

tergabung. Lanjutkan dengan kembali melakukan perbandingan seperti yang dilakukan di langkah kedelapan.

#### Bandingkan $T_0-T_1$

*MERGE* [ $T_0-T_3, T_{01}-T_{03}$ ]

*MERGE* [ $T_1-T_5, T_{11}-T_{50}$ ]

Hasil yang diperoleh adalah  $\bar{A}$ .

#### Bandingkan $T_0-T_1'$

*MERGE* [ $T_{02}-T_3', T_{03}-T_{30}'$ ]

*MERGE* [ $T_1'-T_6, T_{11}'-T_{60}$ ]

Hasil yang diperoleh adalah  $\bar{A}$ .

#### Bandingkan $T_0'-T_1$

*MERGE* [ $T_{02}'-T_4, T_{03}'-T_{40}$ ]

*MERGE* [ $T_{12}-T_5', T_{13}-T_{50}'$ ]

Hasil yang diperoleh adalah  $\bar{A}$ .

#### Bandingkan $T_0'-T_1'$

*MERGE* [ $T_{02}'-T_4', T_{03}'-T_{40}'$ ]

*MERGE* [ $T_{12}'-T_6', T_{13}'-T_{60}'$ ]

Hasil yang diperoleh adalah  $\bar{A}$ .

#### Bandingkan $T_1''-T_2$

*MERGE* [ $T_1''-T_7, T_{11}''-T_{70}$ ]

*MERGE* [ $T_2-T_8, T_{21}-T_{80}$ ]

Hasil yang diperoleh adalah  $B$ .

#### Bandingkan $T_0-T_1''$

*MERGE* [ $T_{04}-T_3'', T_{05}-T_{30}''$ ]

*MERGE* [ $T_{12}''-T_7', T_{13}''-T_{70}'$ ]

Hasil yang diperoleh adalah  $C$ .

#### Bandingkan $T_2'-T_1'$

*MERGE* [ $T_{14}'-T_6', T_{15}'-T_{60}'$ ]

*MERGE* [ $T_2'-T_9, T_{21}'-T_{90}$ ]

Hasil yang diperoleh adalah  $B$ .

#### Bandingkan $T_1-T_2$

*MERGE* [ $T_{15}-T_5'', T_{15}-T_{50}''$ ]

*MERGE* [ $T_{22}-T_8, T_{23}-T_{80}$ ]

Hasil yang diperoleh adalah  $C$ .

Hasil :  $A + \bar{A} + \bar{A} + \bar{A} + B + B + C + C$

Hasil yang diperoleh masih bisa disederhanakan sehingga diulangi kembali langkah kesepuluh sampai langkah ini hingga diperoleh bentuk yang paling sederhana. Bentuk yang paling sederhana yang diperoleh dalam contoh kasus ini adalah  $\bar{A} + B$ .

## V. KESIMPULAN

Proses penyederhanaan ekspresi boolean dengan metode komputasi DNA memang bukan tergolong proses yang singkat. Namun melalui makalah ini, diharapkan dapat menambah pengetahuan pembaca menjadi mengenai metode alternatif untuk menyederhanakan ekspresi boolean. Dalam pengaplikasiannya, metode ini juga memerlukan bantuan manusia untuk mengoperasikan prosedur biokimia yang terlibat di dalamnya. Namun dari metode ini pula terlihat bahwa DNA tidak hanya berada dalam lingkup bidang biologi melainkan dapat bermanfaat di bidang matematika ataupun informatika.

## VI. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Tuhan Yang Maha Esa dan juga kepada Bapak dan Ibu dosen pengampu mata kuliah IF2120 karena sudah membimbing penulis dan membekali penulis dengan berbagai pengetahuan baru di semester ini. Penulis juga berterima kasih karena melalui tugas makalah ini, penulis berkesempatan untuk mengeksplorasi hal-hal baru khususnya tentang komputasi DNA ini.

## REFERENCES

- [1] Loeffler, J. "What is DNA Computing, How Does it Work, and Why it's Such a Big Deal"  
<https://interestingengineering.com/what-is-dna-computing-how-does-it-work-and-why-its-such-a-big-deal> diakses pada 6 Desember 2020 pukul 14.47
- [2] Welianto, A. "DNA: Struktur dan Fungsinya"  
<https://www.kompas.com/skola/read/2020/02/26/180000969/dna--struktur-dan-fungsinya?page=all> diakses pada 6 Desember 2020 pukul 14.53
- [3] R. Munir, Matematika Diskrit. Bandung: Departemen Teknik Informatika Institut Teknologi Bandung, 2003.
- [4] C. Bogard, E. Rouchka, B. Arazi, "DNA Media Storage", Progress in Natural Science 18, 2008.
- [5] S. Paul, G. Sahoo, "Simplification of Boolean Algebra through DNA Computing", International Journal of Computer Applications 17, 2010.

## 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.

Jakarta, 11 Desember 2020



Jesica - 13519011