

## PENERAPAN HOMOMORFISME GRUP PADA PENYIMPANAN DATA BERBASIS DNA

### THE APPLICATION OF GROUP HOMOMORPHISM IN THE DNA BASED STORAGE

Oleh: Aprilia Rizqi Nurcahyani<sup>1</sup>, Musthofa<sup>2</sup>

<sup>1</sup>Mahasiswa Program Studi Matematika FMIPA UNY, <sup>2</sup>Program Studi Matematika FMIPA UNY

<sup>1</sup>[aprilia.rizqi8@gmail.com](mailto:aprilia.rizqi8@gmail.com); <sup>2</sup>[musthofa@uny.ac.id](mailto:musthofa@uny.ac.id)

#### Abstrak

Permintaan untuk penyimpanan data meningkat secara eksponensial, tetapi kapasitas media penyimpanan yang ada saat ini diperkirakan tidak dapat mengimbangi. DNA hadir sebagai media penyimpanan yang potensial dengan kepadatan mencapai 2 petabyte/gram dalam ukuran yang sangat kecil dan mempunyai ketahanan hingga ratusan tahun. Tujuan penelitian ini adalah: (1) memperoleh konstruksi kode Kernel menggunakan homomorfisma grup, (2) mengetahui kendala kode Kernel yang dikonstruksi menggunakan homomorfisma grup dari empat kendala yang meliputi kendala *hamming distance*, *reverse*, *reverse complement* dan *GC-content*, (3) menggunakan *de Bruijn Graph* untuk mengetahui susunan genom awal.

Dalam penelitian ini akan dikonstruksi kode DNA berbasis homomorfisma grup untuk mengkodekan informasi. Konstruksi kode DNA dilakukan dengan memetakan data digital yang berbentuk biner ke dalam barisan nukleotida yang tersusun atas empat basa nukleotida {A, C, G, T}. Kode DNA tersebut akan diterjemahkan menggunakan *de Bruijn Graph* untuk mendapatkan susunan genom awal.

Hasil penelitian menunjukkan bahwa homomorfisma grup dapat digunakan untuk mengkonstruksi kode berbasis DNA. Pembentukan kode DNA ini tergantung pada fungsi dan grup yang digunakan. Penggunaan grup yang berbeda dapat menghasilkan kode Kernel yang berbeda dan pemakaian fungsi yang berbeda juga akan berpengaruh pada hasil konstruksi kode DNA nya. Diantara kode Kernel yang digunakan sebagai contoh dalam penelitian ini memenuhi empat kendala.

**Kata kunci:** *kode dna, homomorfisma grup, de bruijn graph*

#### Abstract

*The demand for data storage increases exponentially, but the capacity of the storage media that exists today is estimated to be unable to keep pace.. DNA is present as a potential storage medium with densities reach 2 petabytes/gram in size is very small and has an endurance of up to hundreds of years. The purpose of this research is: (1) obtain the Kernel code to use homomorfisma construction group, (2) know the constraints Kernel code is constructed using homomorfisma a group of four that includes constraints of hamming distance, reverse, reverse complement and GC-content, (3) using the de Bruijn Graph to find out the composition of the early genomes.*

*In this study the DNA code will be constructed based homomorfisma group to encode information. Construction of the DNA code is done by mapping the binary digital data into a sequence of nucleotides that is composed of four bases {A, C, G, T}. The DNA code will be translated using de Bruijn Graph to get an initial genome arrangement.*

*The results showed that homomorfisma groups can be used to construct a DNA-based code. Establishment of DNA code depends on the function and the group used. The use of different groups can produce code for a different Kernel and the wearing of different functions will also impact on the results of its DNA code construction. Between the Kernel code is used as an example in this study meets the four constraints are hamming distance, reverse, reverse complement and GC-content.*

**Keywords:** *dna code, group homomorphism, de bruijn graph*

## PENDAHULUAN

Teknologi informasi berkembang sangat pesat berbanding lurus dengan sistem penyimpanannya. Hampir seluruh pengarsipan data ditransformasikan ke dalam data digital. Hal ini berakibat pada ledakan data yang sangat besar. *International Data Corporation* (IDC) menyebutkan bahwa semua data digital di seluruh dunia diperkirakan tumbuh lebih dari 16 zettabyte pada tahun 2017. Kondisi yang cukup mengkhawatirkan, permintaan untuk menyimpan data meningkat secara eksponensial akan tetapi tidak diimbangi dengan kapasitas media penyimpanannya. Meskipun media penyimpanan yang ada saat ini telah mengalami perbaikan kepadatan sebesar 185 terabyte ( $185 \times 10^{12}$ ) (www.sony.net), untuk menyimpan data dengan ukuran exabyte ( $10^{18}$ ) bahkan zettabyte ( $10^{21}$ ) data masih akan memerlukan ratusan bahkan jutaan unit media penyimpanan. Disamping itu ketahanan dari media penyimpanan data juga menjadi hal penting yang seharusnya dipikirkan. *Disk* yang berputar dalam jangka waktu tiga sampai lima tahun, atau tape dalam jangka waktu sepuluh sampai tiga puluh tahun akan memerlukan refresh untuk menggosok data yang rusak, menggantikan unit apabila terjadi kerusakan unit dan untuk memperbaiki teknologi (Bornholt, J., et al., 2016:1).

Disinilah *Deoxyribonucleic acid* (DNA) muncul sebagai media penyimpanan dengan kemampuan menyimpan data yang potensial. DNA dapat menyimpan data dengan kepadatan mencapai 2 petabyte/gram (Goldman, N., et al., 2013:2) dan memiliki ketahanan hingga 500 tahun dalam lingkungan keras (Allentoft, M. E., 2012:6). DNA bersifat *non-volatile* dan mempunyai kapasitas penyimpanan yang sangat tinggi (Yazdi, S. M., 2015:1). Menjadikan DNA sebagai media penyimpanan data memiliki tantangan yang sangat besar. Samuel Greengard (2017:17) menjelaskan bahwa tantangan ada pada teknis utama yaitu tidak semua sekuensi DNA diciptakan sama dan efisiensi, serta keakuratan dari proses pengkodean dapat bervariasi

berdasarkan pola *sequencing*. Kemudian biaya yang dikeluarkan untuk satu kali percobaan di laboratorium saat proses *sequencing* DNA sangatlah besar sehingga saat ini belum memungkinkan untuk diproduksi secara komersial.

DNA yang digunakan sebagai media penyimpanan informasi adalah kumpulan *oglinukleotida* (untai DNA pendek) yang berjumlah besar dan tidak menghasilkan *cross hybridization* yang selanjutnya dikenal sebagai kode DNA. dalam mengkonstruksi kode DNA beberapa kendala diberikan yaitu kendala *hamming distance* (jarak hamming), kendala *reserve*, kendala *reserve complement* diperlukan untuk menghindari adanya *cross-hybridization* (hibridisasi silang) dan kendala *GC-content* digunakan untuk memastikan suhu yang diperoleh mencair serupa (Selvakumar, R., 2014:228).

Mengacu pada permasalahan tersebut penulis bermaksud untuk menggunakan homomorfisma grup untuk membangun kode DNA yang akan memenuhi batasan-batasan yang disebutkan. Homomorfisma grup digunakan untuk memperoleh kode Kernel yang nantinya dapat digunakan untuk mengkodekan data digital ke dalam DNA. penelitian ini juga akan membahas mengenai proses baca oleh mesin yang disebut sebagai *genom assembler*. Kerja dari *genom assembler* adalah mengembalikan genom (rangkaiian string) semula setelah dipotong-potong menjadi beberapa fragmen. Sejauh ini *de Bruijn Graph* adalah metode yang paling dipercaya untuk menyusun kembali fragmen-fragmen kecil menjadi genom awal.

## TINJAUAN PUSTAKA

Berikut ini akan diberikan definisi sebagai dasar teori dalam penelitian ini antara lain Grup, Homomorfisma, *Deoxyribonucleic acid* (DNA), dan *de Bruijn Graph*.

## 1. Grup

Istilah grup telah digunakan oleh Galois sekitar tahun 1830 untuk menggambarkan himpunan dari fungsi satu-satu pada himpunan berhingga yang dapat dikelompokkan dalam himpunan tertutup dengan operasi komposisi (Gallian, J. A., 2013:42). Berikut akan diberikan definisi grup

**Definisi 2.1** (Gallian, J. A., 2013:43) Misalkan  $G$  adalah himpunan dengan operasi biner  $*$ . Himpunan  $G$  disebut grup dengan operasi biner jika memenuhi:

- Sifat asosiatif yaitu  $a * (b * c) = (a * b) * c$  untuk setiap  $a, b, c \in G$ .
- Terdapat  $e$  elemen identitas, yaitu terdapat  $e \in G$  sehingga  $a * e = e * a = a$  untuk semua  $a \in G$ .
- Untuk setiap  $a \in G$  mempunyai invers, yaitu terdapat  $b \in G$  sehingga  $a * b = b * a = e$ .

Kemudian jika suatu grup memenuhi  $a * b = b * a$  untuk semua  $a, b \in G$  maka disebut grup abelian. Setiap grup mempunyai sub himpunan dan terdapat sub himpunan yang memenuhi operasi grup yang disebut dengan subgrup.

**Definisi 2.2** (Gallian, J. A., 2013: 61) jika sub himpunan  $H$  dari grup  $G$  merupakan grup dengan operasi yang sama pada  $G$ , maka  $H$  disebut subgrup dari  $G$ .

## 2. Homomorfisma

Istilah homomorfisma berasal dari kata Yunani *homo*, "seperti", dan *morphe*, "bentuk." Homomorfisma merupakan generalisasi alami dari isomorfisma. Konsep kelompok homomorfisma diperkenalkan oleh Camille Yordania tahun 1870, dalam bukunya yang berpengaruh *des Traite substitutions* (Gallian, J. A., 2013:194).

**Definisi 2.4** (Herstein, I. N., 1996:67) Misalkan  $G, G'$  adalah dua grup, pemetaan  $\varphi: G \rightarrow G'$

disebut homomorfisma jika  $\varphi(ab) = \varphi(a)\varphi(b)$  untuk semua  $a, b \in G$ .

Dalam definisi ini produk di sisi kiri pada  $\varphi(ab)$  adalah dari  $G$ , sedangkan produk  $\varphi(a)\varphi(b)$  adalah dari  $G'$ .

**Definisi 2.5** (Fraleigh, J. B., 1989:123) Misalkan  $\varphi: G \rightarrow G'$  adalah homomorfisma, subgrup  $\varphi^{-1}(\{e'\})$ , berisi himpunan semua elemen  $G$  yang dipetakan oleh  $\varphi$  ke elemen identitas  $e' \in G'$  disebut **kernel** dari  $\varphi$ , dinotasikan dengan  $\text{Ker}(\varphi)$ .

## 3. Deoxyribonucleic acid (DNA)

Istilah DNA berasal dari dua kata yaitu deoxyribosa yang berarti "Gula Pentosa" dan kata nukleic yang berarti "Nukleat". Suatu molekul polimer nukleotida (polinukleotida) yang berisis informasi genetik yang terdapat di dalam sel (Syukriani, Y., 2012:2) merupakan definisi dari *Deoxyribonucleic acid* (DNA). DNA dalam sel-sel manusia berfungsi sebagai wadah atau tempat penyimpanan informasi genetik dari makhluk hidup.

### a. Struktur Gen DNA

Struktur DNA merupakan dua untai nukleotida yang disebut dengan *Double Helix*. DNA adalah kombinasi dari empat basa yaitu adenin (A), guanin (G), Sitosin (C), dan timin (T). Adenin berpasangan dengan timin dan sitosin berpasangan dengan guanin untuk membentuk pasangan-pasangan basa.

### b. Komputasi DNA

Komputasi DNA pertama dicetuskan oleh seorang pakar komputasi asal Amerika bernama Leonard Adleman tahun 1994. Adleman (1994) memberikan solusi atas pemecahan masalah dari jembatan Hamilton dengan menciptakan komputer DNA yang dapat menyelesaikan masalah optimasi. Ini membuktikan bahwa DNA memiliki potensi untuk melakukan perhitungan berkali-kali lebih cepat daripada komputer biasa. Hal tersebut kita sebut sebagai komputasi DNA.

langkah awal dari komputasi DNA adalah membangun himpunan untai DNA yang dapat mengkodekan informasi (Selvakumar, 2014). Kualitas dari serangkaian untai DNA tergantung pada kemungkinan untuk hibridisasi yaitu, proses bergabungnya dua untai DNA tunggal untuk membentuk sebuah molekul untai ganda (Aboluion, N. A., 2011:9).

Gaborit&King (2005:99-100) menjelaskan bahwa untuk membangun kode DNA dipenuhi empat kendala berbeda, yaitu:

- i. Kendala jarak Hamming (*hamming distance*),
- ii. Kendala *reverse*,
- iii. Kendala *reverse-complement*,
- iv. dan kendala *GC-content* yang tetap

Kendala pada nomor satu sampai dengan tiga ditujukan untuk membuat satu set untai DNA agar terhindar dari hibridisasi silang (*cross hybridization*). Sedangkan kendala *GC-konten* digunakan untuk memastikan bahwa suhu mencair serupa.

#### 4. De Bruijn Graph

Mengadopsi dari teori graf Euler, seorang matematikawan asal Belanda, Nicholas de Bruijn, menciptakan metode dalam membaca rangkaian string DNA yang panjang. Untai DNA tersusun dari jutaan nukleotida yang panjang sedangkan teknologi *sequencing* yang ada saat ini tidak memiliki kemampuan untuk membaca seluruh urutan nukleotida dalam satu kali baca sehingga mengharuskan memotong dalam fragmen kecil. Saat ini kemampuan membaca DNA mencapai 100 nukleotida dalam satu kali baca (Compeau, P., dkk., 2011:988). Ide-ide Euler disesuaikan untuk mencari urutan siklik dari pesan yang diambil dari alfabet tertentu yang mungkin dari setiap kata dengan panjang  $k$  ( $k$ -mer) muncul sebagai string karakter siklik berturut-turut dengan urutan tepat sekali. Selanjutnya lebih dikenal dengan istilah *de Bruijn Graph*.

#### PEMBAHASAN

Berikut ini akan dibahas mengenai hasil dari konstruksi kode DNA dan teknik *de Bruijn Graph* dalam membaca kode.

### 1. Konstruksi Kode Kernel

Kode Kernel didefinisikan sebagai kernel dari homomorfisma grup. Kernel dari homomorfisma grup dipilih karena memenuhi kendala yang diberikan. Berikut akan dibahas mengenai contoh dari kode Kernel. Misalkan  $G_1, G_2, \dots, G_n$  adalah grup dan  $G = G_1 \times G_2 \times \dots \times G_n$ . Grup  $S$  adalah abelian dan  $\mu: G \rightarrow S$  didefinisikan sebagai

$$\mu(g_1, g_2, \dots, g_n) = \mu_1(g_1)\mu_2(g_2) \dots \mu_n(g_n)$$

dimana  $\mu_i$  adalah homomorfisma dari  $G_i$  ke  $S$ , untuk  $i = 1, 2, \dots, n$ .

Kemudian  $\mu$  merupakan sebuah homomorfisma yang disebutkan pada teorema berikut:

**Teorema 3.1** (Selvakumar, R., 2003:21) *Pemetaan  $\mu$  adalah sebuah homomorfisma dari  $G$  ke  $S$ .*

Bukti:

Misalkan  $g = (g_1, g_2, \dots, g_n)$  dan  $h = (h_1, h_2, \dots, h_n)$ , untuk setiap  $g, h \in G$ .

$$\begin{aligned} \mu: G \times G \times \dots \times G &\rightarrow S \\ \mu(g, h) &= \mu((g_1 \times g_2 \times \dots \times g_n)(h_1 \times h_2 \times \dots \times h_n)) \\ &= \mu(g_1 h_1, g_2 h_2, \dots, g_n h_n) \\ &= \mu_1(g_1 h_1)\mu_2(g_2 h_2) \dots \mu_n(g_n h_n) \\ &= \mu_1(g_1)\mu_1(h_1)\mu_2(g_2)\mu_2(h_2) \dots \mu_n(g_n)\mu_n(h_n) \\ &= (\mu_1(g_1)\mu_2(g_2) \dots \mu_n(g_n))(\mu_1(h_1)\mu_2(h_2) \dots \mu_n(h_n)) \\ &= \mu(g) \mu(h) \end{aligned}$$

Terbukti bahwa  $\mu$  adalah homomorfisma dari  $G$  ke  $S$ .

Judson, T. W. (2011:167) menjelaskan beberapa sifat dari homomorfisma grup sebagai berikut:

Misalkan  $\varphi: G_1 \rightarrow G_2$  sebuah homomorfisma dari grup

1. Jika  $e$  adalah identitas dari  $G_1$ , maka  $\varphi(e)$  adalah identitas dari  $G_2$ ;
2. Untuk semua elemen  $g \in G_1$ ,  $\varphi(g^{-1}) = [\varphi(g)]^{-1}$ ;
3. Jika  $H_1$  adalah subgrup dari  $G_1$ , maka  $\varphi(H_1)$  adalah subgrup dari  $G_2$ ;
4. Jika  $H_2$  adalah subgrup dari  $G_2$ , maka  $\varphi^{-1}(H_2) = \{g \in G: \varphi(g) \in H_2\}$  adalah

subgrup dari  $G_1$ . Selanjutnya, jika  $H_2$  normal di  $G_2$ , maka  $\varphi^{-1}(H_2)$  normal di  $G_1$ .

Sifat-sifat tersebut berguna untuk mengkonstruksi kode Kernel yang akan digunakan dalam mengkonstruksi kode DNA. berikut adalah definisi dari kode DNA.

**Definisi 3.2** (Aboulion, N. A., 2011:18) *Kode DNA dengan panjang n adalah himpunan codewords  $\{x|x = (x_1, x_2, \dots, x_n)\}$  dengan  $x_i \in \{A, C, G, T\}$  (merepresentasikan empat nukleotida dalam DNA).*

**Contoh 3.1** Diberikan  $K = \{00, 01, 10, 11\}$  dimana  $A \rightarrow 00, C \rightarrow 01, G \rightarrow 10, T \rightarrow 11$  dan  $(S, +)$  grup. Kemudian definisikan  $\mu: K \times K \times K \rightarrow S$  sebagai  $\mu(xyz) = \mu_1(x)\mu_2(y)\mu_3(z)$  dimana  $S = Z_2$  dan  $\mu_i: K \rightarrow S$  didefinisikan sebagai  $\mu_i(00) = 0, \mu_i(11) = \mu_i(01) = \mu_i(10) = 1$ .

Tabel 1. Kode Kernel atas  $Z_2$

AAA	AGG	CCA	AAT	CGT	ATA	CAG	AGC
CGA	TCG	GCT	GGA	TCC	CCT	ACC	GGT
TGG	ATT	TTA	TAA	ACG	GCA	TGC	GAG
CTC	TAT	GTG	CAC	GAC	CTG	GTC	TTT

Kode Kernel atas  $Z_2$  menghasilkan 32 kode.

**Contoh 3.2** Didefinisikan  $\mu: M \times M \times M \rightarrow S$  sebagai  $\mu(xyz) = \mu_1(x)\mu_2(y)\mu_3(z)$  dimana  $S = Z_3$  dan  $M = \{A, C, G, T\}$ .  $A \rightarrow 00, C \rightarrow 01; 10; 22, G \rightarrow 02; 20; 11, T \rightarrow 21; 12$  dan  $\mu_i: M \rightarrow S$  didefinisikan sebagai  $\mu_i(00) = \mu_i(12) = \mu_i(21) = 0, \mu_i(10) = \mu_i(01) = \mu_i(22) = 1, \mu_i(02) = \mu_i(20) = \mu_i(11) = 2$ .

Tabel 2. Kode Kernel atas  $Z_3$

ACG	CGA	TCG	GCT
AGC	GCA	TGC	CGT
GAC	CAG	CTG	GTC
AAA	TTT	GGG	CCC

Kode Kernel atas  $Z_3$  menghasilkan 16 kode.

**Contoh 3.3** Didefinisikan  $\mu: M \times M \times M \times M \rightarrow S$  sebagai  $\mu(wxyz) = \mu_1(w)\mu_2(x)\mu_3(y)\mu_4(z)$  dimana  $S = Z_3$  dan  $M = \{A, C, G, T\}$ .  $A \rightarrow 00, C \rightarrow 01; 10; 22, G \rightarrow 01; 20; 11, T \rightarrow 21; 12$  dan  $\mu_i: M \rightarrow S$  didefinisikan sebagai

$$\mu_i(00) = \mu_i(12) = \mu_i(21) = 0, \mu_i(10) = \mu_i(01) = \mu_i(22) = 1, \mu_i(02) = \mu_i(20) = \mu_i(11) = 2.$$

Tabel 3. Kode Kernel atas  $Z_3$  dengan 4 Nukleotida

AAAA	ACGT	AGTC	ACTG	ATGC	ATCG
TCAG	TGAC	TACG	TAGC	AAGC	AACG
AGCA	AGAC	GACT	GATC	GTAC	GTCA
CTGA	CTAG	CATG	CAGT	CGTA	CGAT
TCTG	TCGA	TGCT	TGTA	GGGA	AGGG
CCCA	CACC	CCAC	ACCC	GGGT	TGGG
CCCT	CTCC	CCTC	TCCC	GCGC	GGCC
CGGC	CCGG	CGAA	GCAA	CAGA	CAAG
GCTT	GTCT	CGTT	GTTC	CTGT	CTTG
TCGA	TGCA	TTGC	TTCG	GCCG	CGCG
ACAG	ACGA	GAGG	GGAG	GACA	GAAC
GCAT	GCTA	GTGG	GGTG	AGCT	TTTT

Kode Kernel atas  $Z_3$  dengan panjang nukleotida 4 menghasilkan 72 kode.

## 2. Kendala pada Kode DNA

Beberapa kendala diberikan untuk mengkonstruksi kode DNA yaitu kendala *reverse*, *reverse complement*, *hamming distance* dan *GC-content*. Menurut Aboluion, N. A. (2011:10) empat kendala tersebut dapat didefinisikan sebagai berikut:

- i. Misalkan  $H(x, y)$  menunjukkan jarak Hamming antara dua kata (yaitu jumlah posisi dimana mereka berbeda-beda). Kendala Jarak Hamming adalah bahwa  $H(x, y) \geq d$  untuk semua  $x, y \in C$  dengan  $x \neq y$ , untuk beberapa dirumuskan jarak minimum  $d$ .
- ii. Kendala *reverse* adalah bahwa  $H(x^R, y) \geq d$  untuk semua  $x, y \in C$ , dimana  $x^R$  adalah kebalikan dari codeword  $x$ .
- iii. Kendala *reverse-complement* adalah  $H(x^{RC}, y) \geq d$  untuk semua  $x, y \in C$ , dimana  $x^{RC}$  adalah *reverse-complement*  $x$  diperoleh dengan mengambil  $x$  dan melakukan simbol persimpangan yaitu  $A \leftrightarrow T, C \leftrightarrow G$ .
- iv. Kendala *GC-konten* adalah bahwa setiap codeword  $x \in C$  memiliki *GC-konten* yang sama. *GC-konten* dari sebuah kata DNA didefinisikan menjadi jumlah posisi C dan G yang tetap dalam satu codon.

Perhatikan kembali Contoh 3.1 menunjukkan bahwa kode kernel memenuhi kendala:

a. Kendala *Reverse*

Tabel 4. Kendala *Reverse* Kode Kernel Contoh 3.1

Kode	Reverse	Kode	Reverse
AAA	AAA	AAT	TAA
CGA	AGC	CAC	CAC
TGG	GGT	CGT	TGC
CTC	CTC	TCC	CCT
AGG	GGA	ACG	GCA
TCG	GCT	GAC	CAG
ATT	TTA	ATA	ATA
TAT	TAT	CTG	GTC
CCA	ACC	GAG	GAG
GTG	GTG	TTT	TTT

Terbukti bahwa kode Kernel Contoh 3.1 memenuhi kendala *reverse*.

b. Kendala *Reverse Complement*

Tabel 5. Kendala *Reverse Complement* Kode Kernel Contoh 3.1

Kode	Reverse Complement	Kode	Reverse Complement
AAA	TTT	GTG	CAC
CGA	TCG	GGA	TCC
TGG	CCA	CGT	ACG
CTC	GAG	GAC	GTC
ATT	AAT	CCT	AGG
TAT	ATA	GCA	TGC
GCT	AGC	CTG	CAG
TTA	TAA	ACC	GGT

Berdasarkan tabel diatas kode Kernel contoh 3.1 memenuhi kendala *reverse complement*.

c. Kendala *Hamming Distance* (Jarak Hamming)

- d(AAA, AGG) = 2
- d(AAA, CCA) = 2
- d(AAA, AAT) = 1
- d(AAA, CGT) = 3
- d(AAA, ATA) = 1
- d(AAA, CAG) = 2
- d(AAA, AGC) = 2
- d(AGG, CCA) = 3
- d(AGG, AAT) = 2
- d(AGG, CGT) = 2
- d(CGA, TCG) = 3
- d(CGA, GCT) = 3
- d(CGA, GGA) = 1
- d(CGA, TCC) = 3
- d(CGA, CCT) = 2
- d(CGA, ACC) = 3
- d(CGA, GGT) = 2
- d(TCG, GCT) = 3
- d(TCG, GGA) = 3
- d(TCG, TCC) = 1

Berdasarkan hasil perhitungan diatas maka kode kernel mempunyai jarak hamming  $d(C) = 1$ .

d. Kendala *GC-content*

Pengamatan pada kode Kernel coontoh 3.1 menghasilkan 32 kode dengan masing-masing kode memenuhi jumlah G genap dan C genap atau jumlah G ganjil dan C ganjil.

Jadi kode Kernel pada Contoh 3.1 memenuhi empat kendala.

Menggunakan langkah yang sama kode Kernel Contoh 3.2 dan Contoh 3.3 memenuhi tiga kendala yaitu kendala *reserve*, *reserve complement* dan kendala *hamming distance*.

3. **Konstruksi *De Bruijn Graph***

Menggunakan metode *de Bruijn Graph*, memudahkan dalam membaca dan mengembalikan genom awal pada kode DNA. Berdasarkan kode Kernel pada Contoh 3.1, jika diberikan fungsi sebagai berikut:

- AAA→00000 CCA→01100 CGT→01111
- CAG→10010 CGA→00001 GCT→11000
- TCC→11110 ACC→00101 TGG→00010
- TTA→10001 ACG→11101 TGC→01011
- CTC→00100 GTG→00111 GAC→11011
- GTC→10110 AGG→01000 AAT→01110
- ATA→10111 AGC→01101 TCG→10000
- GGA→11100 CCT→01010 GGT→11010
- ATT→00011 TAA→11001 GCA→10100
- GAG→10101 TAT→00110 CAC→10011
- CTG→01001 TTT→11111

Menggunakan kata MATEMATIKA, akan dikodekan dalam kode DNA dengan langkah sebagai berikut:

a. Mengubah kata dalam bentuk biner

MATEMATIKA→100110110000011010100100  
010110011011000001101010010010011001011  
1000001

b. Mengubah kode biner ke dalam kode DNA menggunakan kode kernel dengan terlebih dulu membuat kode blok yang terdiri dari 5 angka biner.

10011|01100|00011|01010|01000|10110|01101|10000|01101|01001|00100|11001|01110|00001

maka diperoleh kode DNA sebagai berikut:

Tabel 6. Kode Blok dari Kode Kernel Contoh 3.1

Kode Blok	Kode Kernel	Kode Blok	Kode Kernel
10011	CAC	10000	TCG
01100	CCA	01100	CCA
00011	ATT	10011	CAC
01010	CCT	00100	CTC
01000	AGG	11001	TAA
10110	GTC	01110	AAT
01101	AGC	00001	CGA

Sehingga diperoleh kode DNA dari kata MATEMATIKA yaitu CACCCAATTCCTAGGGT CAGCTCGCCACACCTCTAAAATCGA. Kode tersebut selanjutnya disebut genom awal.

c. Konstruksi *de Bruijn Graph*

Diasumsikan pembacaan 4 nukleotida tanpa terjadi overlapp yaitu adanya perulangan dalam membaca urutan nukleotida. Hasil baca adalah sebagai berikut:

CACC, ACCC, CCCA, CCAA, CAAT, AATT, ATTC, TTCC, TCCT, CCTA, CTAG, TAGG, AGGG, GGGT, GGTC, GTCA, TCAG, CAGC, AGCT, GCTC, CTCG, TCGC, CGCC, GCCA, CCAC, CACA, ACAC, CACC, CCTC, CTCT, TCTA, CTAA, TAAA, AAAA, AAAT, AATC, ATCG, TCGA.

d. Menentukan *k*-mer dari Simpul

Semua kemungkinan *k*-mer ditunjukkan pada Tabel 7.

Tabel 7. Daftar Semua Kemungkinan *k*-mer Menggunakan Kode Kernel Contoh 3.1

Edge	Node	<i>k</i> -mer
1	CAC ACC	CACC
2	ACC CCC	ACCC
3	CCC CCA	CCCA
4	CCA CAA	CCAA
5	CAA	CAAT
⋮	⋮	⋮
42	ACA CAC	ACAC

Maka diperoleh kemungkinan banyaknya *k*-mer adalah 42.

e. Konstruksi Genom Awal

Semua kemungkinan *k*-mer disusun sedemikian hingga menjadi susunan seperti berikut:

```

C A C C
  A C C C
    C C C A
      C C A A
        C A A T
          A A T T
            A T T C
              ⋮ ⋮ ⋮
                G A C A
                  A C A C
    
```

Dengan mengambil urutan pertama nukleotida dari masing-masing *k*-mer akan diperoleh susunan genom awal sebagai berikut CACCCAATTCCTAGGGTCAGCTCGCCA CACCTCTAAAATCGA.

## SIMPULAN DAN SARAN

### 1. Simpulan

Kesimpulan yang dapat diberikan adalah sebagai berikut:

- a. Tiga contoh kode Kernel yang dapat dikonstruksi menggunakan homomorfisma grup yaitu
  - i. Kode kernel atas  $Z_2$  menghasilkan 32 kode dengan panjang nukleotida 3.

- ii. Kode kernel atas  $Z_3$  menghasilkan 16 kode dengan panjang nukleotida 3.
- iii. Kode kernel atas  $Z_3$  menghasilkan 72 kode dengan panjang nukleotida 4.
- b. Kode Kernel pada contoh pertama memenuhi empat kendala yaitu *kendala hamming distance*, *reserve*, *reserve complement* dan *GC-content*. Sedangkan kode Kernel contoh kedua dan ketiga memenuhi tiga kendala yaitu *kendala hamming distance*, *reserve*, dan *reserve complement*.
- c. Diperoleh genom sebagai berikut:
  - i. Menggunakan kode Kernel Contoh 3.1 diketahui genom awal dari *de Bruijn Graph* adalah  
ACCCAATTCCTAGGGTCAGCTCGC  
CACACCTCTAAAATCGA.
  - ii. Menggunakan kode Kernel Contoh 3.2 diketahui genom awal dari *de Bruijn Graph* adalah  
GGGCGTAAAGCATGCGACTCGGGG  
CGTAAAGCATGCGACGGGGGGCTG  
AAAGAC.
  - iii. Tanpa Menggunakan kode Kernel diketahui genom awal dari *de Bruijn Graph* adalah  
GCGTAACGGGCACCGCGTAACGGG  
CAGCGCCTAAC.

## 2. Saran

Diharapkan untuk peneliti selanjutnya dapat mengembangkan penelitian ini dengan menentukan metode yang sesuai agar dapat digunakan untuk mengoreksi terjadinya eror pada kode DNA dan menentukan metode untuk menangani kasus overlapp.

## DAFTAR PUSTAKA

Aboluion, N. A. (2011). The construction of DNA codes using a computer algebra system. Disertasi Doktor, tidak diterbitkan, University of Glamorgan, Walues, UK.

Adleman, L.M. (1994). Molecular computation of solutions to combinatorial problems. *Science*, 266, 1021-1024.

Allentoft, M. E., Collins, M., Harker, D., et al. (2012). The Half-life of DNA in Bone:

Measuring Decay Kinetics in 158 Dated Fossils. *Biological Sciences*, 279(1748):6.

Bornholt, J., Lopez, L., Carmean, D. M., et al. (2016). A DNA-Based Archival Storage System. *ACM 978-1-4503-4091*, 1-2.

Compeau, P., Pevzner, P. A., Tesler, G. (2011). How to Apply de Bruijn Graphs to Genome Assembly. 29(11), 987-988.

Fraleigh, J. B. (1994). *A First Course in Abstract Algebra*. USA: Addison Wesley.

Gaborit, P.&King, O.D. (2005). Linear constructions for DNA codes. *Theoretical Computer Science*. 334, 99-100.

Gallian, J. A. (2013). *Contemporary Abstract Algebra*. Boston: Cengage Learning.

Goldman, N., Bertone, P., Chen, S., et al (2013). Towards practical, higj-capacity, low maintance information storage in syntesized DNA. *Nature*, 494, 2.

Greengard, S. (2017). Cracking the Code on DNA Storage. *Communications of the ACM*, 60(7), 16-18.

Herstein, I. N. (1995). *Abstract Algebra*. USA: Viacom Company.

IDC. Where in the world is storage. [http://www.idc.com/downloads/where\\_is\\_storage\\_infographic\\_24338.pdf](http://www.idc.com/downloads/where_is_storage_infographic_24338.pdf), 2013. (diakses pada tanggal 25 Maret 2018 pukul 10.49)

Judson, T. W. (2011). *Abstract Algebra Theory and Application*. Texas: Stephen F. Austin States University.

Selvakumar, R. (2014). Unconventional Construction of DNA codes: Grup Homomorphism. *Journal of Discrete Mathematical Sciences and Cryptography*, 17(3), 227-237.

Selvakumar, R.&Balasubramanie, P. (2003). Construction of Kernel Cods and Its Trellis. *ACCST research Journal*. I. 94-96.

Sony. Sony develops magnetic tape technology with the world's highest recording density. <http://www.sony.net/SonyInfo/News/Press/>

201404/14-044E/2014. (diakses diakses  
pada tanggal 2 April 2018 pukul 14.13)

Sukirman, H. (2000). *Pengantar Aljabar Abstrak*.  
Yogyakarta: UNY Press.

Syukriani, Y. F. (2012). *DNA Forensik*.  
Bandung: Fakultas Kedokteran Universitas  
Padjadjaran.

Yazdi, S. M., Kiah, H. M., Garcia, E. R., et al.  
(2015). DNA-Based Storage: Trends and  
Methods, 1