PRISMA 8 (2025): 377-384
PRISMA, Prosiding Seminar Nasional Matematika
https://journal.unnes.ac.id/sju/index.php/prisma/
ISSN 2613-9189

Konstruksi Kode DNA dan Implementasinya Menggunakan Python

Muhammad Arsyad[a,*], Putranto Hadi Utomo[a]

[a] *Program Studi Matematika, Universitas Sebelas Maret, Surakarta 57126, Indonesia*

* *Alamat Surel: muharsyad2201@gmail.com*

Abstrak

Penelitian ini berfokus pada konstruksi kode DNA di atas lapangan hingga $\mathbb{F}_2$ dengan memperhatikan berbagai batasan kombinatorial, seperti jarak Hamming, batasan terbalik, batasan komplemen terbalik, dan konten GC tetap. Kode DNA dikonstruksikan menggunakan pendekatan teori koding dengan menerapkan konsep kode *reversible self-dual*. Penelitin ini juga mencakup studi literatur tentang teori koding, kode linier, kode *reversibel self-dual*, dan kode DNA untuk memahami dasar-dasar serta batasan yang terkait. Matriks generator yang digunakan untuk membangung kode DNA mencakup variasi seperti $(I_n \mid I_n), (I_n \mid R_n)$, $(I_n \mid I_n + J_n)$, dan $(I_n \mid R_n + J_n)$. Implementasi algoritma konstruksi kode DNA dilakukan menggunakan bahasa pemrograman Python untuk memvalidasi dan menguji kode yang dihasilkan. Hasil penelitian menunjukan bahwa metode yang dikembangkan dapat menghasilkan kode DNA yang memenuhi semua batasan kombinatorial DNA yang ditetapkan. Kode DNA yang dihasilkan memiliki parameter panjang $2n$ dan dimensi n untuk n bilangan genap, dan jarak minimum $d = 2$ untuk matriks generator $(I_n \mid I_n)$ dan $(I_n \mid R_n)$, serta jarak minimum $d = 4$ untuk matrik generator $(I_n \mid I_n + J_n)$ dan $(I_n \mid R_n + J_n)$.

Kata kunci:

Kode DNA, lapangan hingga $\mathbb{F}_2$, kode *reversible self-dual*, implementasi Python.

1. Pendahuluan

1.1. Latar belakang masalah

DNA, sebagai molekul pembawa informasi genetik, memainkan peran fundamental dalam pengembangan, fungsi, dan reproduksi makhluk hidup. Setiap molekul DNA terdiri dari dua untai komplementer yang tersusun dari urutan empat basa nukleotida: Adenin ($\mathcal{A}$), Sitosin ($\mathcal{C}$), Guanin ($\mathcal{G}$), dan Timin ($\mathcal{T}$). Dalam konteks teori pengkodean, setiap untai DNA dapat direpresentasikan sebagai kata kode yang terdiri dari alfabet $\{\mathcal{A}, \mathcal{C}, \mathcal{G}, \mathcal{T}\}$, membentuk himpunan kode DNA dengan panjang tetap n.

Kode DNA yang memenuhi batasan kombinatorial DNA, seperti batasan jarak Hamming, batasan terbalik, batasan komplemen terbalik, dan batasan konten GC, menjadi krusial dalam konteks pengkodean informasi genetik (Gaborit & King, 2005). Dalam hal ini, penting untuk meminimalkan hibridasi yang tidak diinginkan antara untai DNA yang berbeda, sekaligus memastikan bahwa semua kata kode memiliki karakteristik termodinamika yang serupa (Limbachiya *et al*., 2016)

Pendekatan aljabar terhadap konstruksi kode DNA umumnya dilakukan dengan menggunakan lapangan hingga atau *ring*. Sebagai contoh, dalam penelitian (Butar-Butar & Bukit, 2022; Gaborit & King, 2005; Kim *et al*., 2021; Varbanov *et al*., 2014), kode DNA dikonstruksikan atas lapangan hingga $\mathbb{F}_4$, yang elemen-elemennya $\{0, 1, \omega, \omega^2\}$ memenuhi hubungan $\omega^2 + \omega + 1 = 0$, dan dalam penelitian (Feng *et al*., 2015) kode DNA dikonstruksikan atas *ring* $\mathbb{Z}_4$, yang elemen-elemennya $\{0, 1, 2, 3\}$. Namun implementasi kode semacara ini dalam sistem komputer mengalami kesulitan karena struktur $\mathbb{F}_4$ dan $\mathbb{Z}_4$ lebih kompleks dibandingkan dengan sistem biner yang digunakan dalam komputasi digital. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan konstruksi kode DNA atas lapangan hingga $\mathbb{F}_2$, yang lebih mudah

To cite this article:
Arsyad, M, & Utomo, P. H. (2025). Konstruksi Kode DNA Dan Implementasinya Menggunakan Python. *PRISMA, Prosiding Seminar Nasional Matematika 8*, 377-384

diimplementasikan secara komputasional. Dengan mennggunakan pendekatan aljabar, kode DNA yang dihasilkan akan memenuhi keempat batasan kombinatorial DNA. Konstruksi kode DNA ini akan diimplementasikan menggunakan bahasa pemrograman Python, memungkinkan penerapan yang lebih efisien dalam sistem komputasi modern

1.2. Rumusan masalah

Penelitian ini berfokus pada masalah bagaimana mengkonstruksikan kode DNA atas lapangan hingga $\mathbb{F}_2$, dengan memenuhi batasan kombinatorial DNA dan mengimplementasikan konstruksi kode DNA secara komputasional menggunakan bahasa pemrograman Python.

1.3. Tujuan

Penelitian ini bertujuan untuk mengembangkan dan mengimplementasikan konstruksi kode DNA atas lapangan hingga $\mathbb{F}_2$ yang memenuhi batasan kombinatorial. Dengan pendekatan ini, diharapkan dapat mempermudah proses komputasi dan integrasi dalam sistem komputer modern.

1.4. Manfaat

Manfaat yang diharapkan dari penelitian ini adalah tersedianya sebuah algoritma konstruksi kode DNA atas lapangan hingga $\mathbb{F}_2$ dan tersedia program dalam bahasa pemrograman Python untuk memfasilitasi konstruksi kode DNA yang diperlukan untuk penelitian lebih lanjut.

2. Metode

Metode penelitian ini mengintegrasikan studi literatur dan pengembangan algoritma untuk konstruksi kode DNA, serta implementasi menggunakan Python. Studi literatur dimulai dengan kajian mengenai teori koding (Huffman & Pless, 2003) meliputi kode linier, jarak Hamming, lapangan hingga, dan lain-lain. Serta mengkaji mengenai kode DNA (Gaborit & King, 2005, Limbachiya *et al*., 2016), kode *reversible self-dual* (Kim *et al*., 2020).

Selanjutnya, penelitian ini menyelidiki sifat-sifat kode *reversible self-dual* dan bagaimana sifat-sifat tersebut memengaruhi konstrusi kode DNA atas lapangan $\mathbb{F}_2$. Dalam implementasinya, algoritma konstruksi kode DNA mengacu pada pemilihan dan penggunaan matriks generator yang sesuai untuk menghasilkan kode yang memenuhi batasan kombinatorial yang ditentukan.

Proses implementasi ini melibatkan pengembangan fungsi-fungsi Python untuk menguji dan memastikan bahwa kode yang dihasilkan memenuhi semua batasan yang telah ditetapkan. Fungsi-fungsi ini mencakup konstruksi kode dari matriks generator, pembentukan subkode *self-reverse*, dan *self-reverse complementary*, serta pengecekan bobot GC dari kode yang dihasilkan. Implementasi Python memungkinkan analisis yang efisien dan validasi hasil konstruksi kode DNA.

3. Hasil dan Pembahasan

Ruang vektor $\mathbb{F}_q^n$ adalah himpunan semua n-tupel dari elemen-elemen di $\mathbb{F}_q$. Kode C atas $\mathbb{F}_q$ dengan panjang n dan jumlah elemen M, dilambangkan $C - [n, M]$, adalah subhimpunan dari $\mathbb{F}_q^n$ dengan ukuran M. Jika C adalah subruang berdimensi k, maka C disebut kode linier berdimensi k dan dinotasikan $C - [n, k]$.

Matriks generator untuk kode $C - [n, k]$ adalah matriks G berukuran $k \times n$ yang baris-barisnya membentuk basis dari C. Kode dual $C^\perp$ dari kode C adalah himpunan vektor $x \in \mathbb{F}_q^n$ yang ortogonal terhadap semua vektor di C, yaitu $C^\perp = \{x \in \mathbb{F}_q^n | x \cdot c = 0, \forall c \in C\}$. Kode $C^\perp$ memiliki dimensi $n - k$ dan panjang n. Kode C disebut *self-orthogonal* jika $C \subseteq C^\perp$ dan *self-dual* jika $C = C^\perp$.

Jarak Hamming $d_H(x, y)$ antara dua vektor $x, y \in \mathbb{F}_q^n$ adalah jumlah koordinat yang berbeda antara x dan y. Jarak minimum suatu kode C adalah jarak Hamming terkecil antara kata kode yang berbeda dalam C, yang juga disebut bobot minimum. Bobot (*weight*) vektor $x \in \mathbb{F}_q^n$, dilambangkan $wt(x)$, adalah jumlah koordinat bukan nol dalam x. Bobot-GC ($wt_{GC}(x)$) adalah jumlah koordinat G dan C dalam x.

Kode DNA D dengan panjang n adalah himpunan kata kode $(x_1, x_2, \cdots, x_n)$ di mana setiap x_i merupakan elemen dari $\{\mathcal{A}, \mathcal{C}, \mathcal{G}, \mathcal{T}\}$, yang mewakili empat basa nukleotida dalam DNA. Notasi$(\widehat{\ })$ digunakan untuk menunjukan kompelemn Watson-Crick: $\hat{\mathcal{A}} = \mathcal{T}, \hat{\mathcal{T}} = \mathcal{A}, \hat{\mathcal{C}} = \mathcal{G}, \hat{\mathcal{G}} = \mathcal{C}$.

Pada penelitian ini, kode DNA dikonstruksikan atas lapangan hingga $\mathbb{F}_2$, dengan setiap basa nukleotida dipetakan ke $\mathbb{F}_2^2$ sebagai berikut: $\mathcal{A} \to 00, \mathcal{C} \to 01, \mathcal{G} \to 10, \mathcal{T} \to 11$. Dengan demikian, kode DNA D atas $\mathbb{F}_2$ memiliki panjang $2n$, di mana n bilangan bulat.

Untuk kata kode $x = (x_1, x_2, \cdots, x_{2n})$ dalam D, kebalikan (*reverse*) dari x dilambangkan x^R yaitu $x^R = (y_1, y_2, \cdots, y_{2n})$ di mana

$$y_i = \begin{cases} x_{2n-i} & \text{untuk } i = 1,3,\cdots,2n-1 \\ x_{2n-i+2} & \text{untuk } i = 2,4,\cdots,2n. \end{cases} \tag{1}$$

Komplemen dari x dilambangkan sebagai $x^C = (\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_{2n})$, dan komplemen terbaliknya (*reverse complement*) adalah $x^{RC} = (x^R)^C = (x^C)^R$. Enumerator bobot lengkap kode DNA D atas $\mathbb{F}_2$ adalah

$$CWE_D(a,b,c,d) = \sum_{x \in D} a^{n_{00}(x)} b^{n_{11}(x)} c^{n_{01}(x)} d^{n_{10}(x)}. \tag{2}$$

Sedangkan enumerator bobot-GC adalah

$$GCW_D(a,b) = CWE_D(a,a,b,b) = \sum_{x \in D} a^{n_{00}(x)+n_{11}(x)} b^{n_{01}(x)+n_{10}(x)}. \tag{3}$$

Di mana $n_s(x)$ adalah jumlah kemunculan s pada kata kode x

Dalam konsntruksi kode DNA D, terdapat empat batasan kombinatorial:

- Batasan jarak Hamming: $d_H(x,y) \geq d, \forall x, y \in D$ dengan $x \neq y$.
- Batasan terbalik: $d_H(x^R, y \geq\ d, \forall x, y \in D$ termasuk $x = y$.
- Batasan komplemen terbalik: $d_H(x^{RC}, y) \geq d, \forall x, y \in D$ termasuk $x = y$.
- Batasan konten GC tetap: Setiap kata kode $x \in D$ harus memiliki bobot GC yang sama.

Dua kode C dan C' dikatakan permutasi ekuivalen jika C' dapata diperoleh dari C melalui permutasi koordinat tertentu. Dengan kata lain, jika elemen-elemen kode dalam C diubah mengikuti permutasi tertentu, kode yang dihasilkan akan menjadi C'. Permutasi σ dalam grup simetris S_{2n} adalah automorfisma dari kode C jika kode tersebut tetap tidak berubah setelah permutasi diterapkan, yaitu $C = \sigma C$. Semua automorfisma dari C membentuk grup yang dikenal sebagai grup autormorfisma, dinotasikan sebagai $Aut(C)$. Untuk kode DNA atas $\mathbb{F}_2$ dengan panjang $2n$, permutasi yang digunakan didefinisikan sebagai:

$$\sigma = \begin{cases} (i, 2n-i) & \text{untuk } i \text{ ganjil} \\ (i, 2n-i+2) & \text{untuk } i \text{ genap.} \end{cases} \tag{4}$$

Kode disebut *reversible* jika setiap kata kode tetap berada dalam himpunan kode setelah diterapkan permutasi tertentu. Dengan kata lain, jika permutasi ini diterapkan pada elemen-elemen dalam setiap kata kode, kode yang dihasilkan tetap merupakan anggota dari himpunan kode. Kode *self-dual* yang *reversible* atau *reversible self-dual* adalah kode di mana setiap kata kode memiliki *dot product* nol dengan setiap kata kode lainnya, termasuk dirinya sendiri. Kode *self-dual* C adalah *reversible self-dual* jika permutasi σ tersebut termasuk dalam grup automorfisma $Aut(C)$ (Kim *et al*., 2020). Dalam penelitian ini, istilah *reversible* merujuk pada permutasi 4 yang telah didefinisikan di atas.

Untuk kode DNA D atas $\mathbb{F}_2$ yang dihasilkan dari matriks A, matriks *reverse*-nya dilambangkan dengan A^R. Jika $A = [a_1, a_2, a_3, a_4, \cdots, a_{2n-3}, a_{2n-2}, a_{2n-1}, a_{2n}]$ dengan a_i sebagai vektor kolom dari A, maka $A^R = [a_{2n-1}, a_{2n}, a_{2n-3}, a_{2n-2}, \cdots, a_3, a_4, a_1, a_2]$. Matriks identitas I_n berukuran $n \times n$, dan I_n^R adalah matriks identitas yang dibalik sesuai dengan permutasi 4 untuk kode DNA D atas $\mathbb{F}_2$. Jika $I_{2n} = [i_1, i_2, i_3, i_4, \cdots, i_{2n-3}, i_{2n-2}, i_{2n-1}, i_{2n}]$, maka $I_{2n}^R = [i_{2n-1}, i_{2n}, i_{2n-3}, i_{2n-2}, \cdots, i_3, i_4, i_1, i_2]$.

Proposisi 1. Misalkan C adalah kode dengan panjang $2n$ atas $\mathbb{F}_2$. Kode C dikatakan *self-dual* jika matriks generator bentuk standar untuk C adalah

$$G = (I_n \quad | \quad A) \tag{5}$$

dimana I_n adalah matriks identitas berukuran $n \times n$ dan A adalah matriks ortogonal berukuran $n \times n$.

Bukti. Misalkan C adalah kode dengan panjang $2n$ atasa $\mathbb{F}_2$ dengan matriks generator bentuk standar $G = (I_n \quad | \quad A)$, di mana I_n matriks identitas $n \times n$ dan A matriks berukuran $n \times n$. Kode C dikatakan *self-dual* jika $C = C^{\perp}$, yaitu setiap kata kode $x \in C$ ortogonal terhadap setiap kata kode lainnya dalam C. Ini terjadi jika:

$$GG^T = (I_n \quad | \quad A)\binom{I_n}{A^T} = (I_n + AA^T) = 0.$$

Agar $I_n + AA^T = 0$ terpenuhi, A haruslah matriks ortogonal, yaitu $AA^T = I_n$. Oleh karena itu, C adalah *self-dual* jika matriks A ortogonal.

■

Proposisi 2. Misalkan C adalah kode *reversible self-dual* atas $\mathbb{F}_2$ dengan panjang $2n$ dan n genap. Maka, terdapat matriks generator G untuk kode C dalam bentuk-bentuk berikut:

- $G_1 = (I_n \quad | \quad I_n)$
- $G_2 = (I_n \quad | \quad R_n)$
- $G_3 = (I_n \quad | \quad I_n + J_n)$
- $G_4 = (I_n \quad | \quad R_n + J_n)$

Dimana I_n adalah matriks identitas berukuran $n \times n$, R_n adalah matriks anti diagonal berukuran $n \times n$, dan J_n adalah matriks berukuran $n \times n$ dengan semua elemen 1.

Bukti. Misalkan kode C adalah kode *reversible self-dual* dengan matriks generator bentuk standar $G = (I_n \quad | \quad A)$, maka matriks generator untuk kode C^R adalah $G^R = (A^R \quad | \quad I_n^R)$ juga menghasilkan kode C. C juga kode *self-dual* maka A harus ortogonal. Dan matriks ortogonal juga non-singular, yang berarti $A^{-1} = A^T$. Karena A non-singular, maka A invertible. Dengan demikian, G^R dapat diklikan dengan $(A^R)^{-1}$ untuk mendapatkan matriks G dalam bentuk standar

- Untuk $G_1^R = (I_n^R \quad | \quad I_n^R)$

  $$(I_n^R)^{-1}G = (I_n^R)^{-1}(I_n^R \quad | \quad I_n^R) = ((I_n^R)^{-1}I_n^R \quad | \quad (I_n^R)^{-1}I_n^R) = (I_n \quad | \quad I_n).$$

- Untuk $G_2^R = (R_n^R \quad | \quad I_n^R)$

  $$(R_n^R)^{-1}G = (R_n^R)^{-1}(R_n^R \quad | \quad I_n^R) = ((R_n^R)^{-1}R_n^R \quad | \quad (R_n^R)^{-1}I_n^R) = (I_n \quad | \quad R_n).$$

- Untuk $G_3^R = ((I_n + J_n)^R \quad | \quad I_n^R)$

  $$((I_n + J_n)^R)^{-1}G = (I_n^R + J_n^R)^{-1}((I_n + J_n)^R \quad | \quad I_n^R)$$
  $$((I_n + J_n)^R)^{-1}G = ((I_n^R + J_n^R)^{-1}(I_n + J_n)^R \quad | \quad (I_n^R + J_n^R)^{-1}I_n^R)$$
  $$((I_n + J_n)^R)^{-1}G = (I_n \quad | \quad I_n + J_n).$$

- Untuk $G_4^R = ((R_n + J_n)^R \quad | \quad I_n^R)$

$$((R_n + J_n)^R)^{-1}G = (R_n^R + J_n^R)^{-1}((R_n + J_n)^R \quad | \quad I_n^R)$$
$$((R_n + J_n)^R)^{-1}G = ((R_n^R + J_n^R)^{-1}(R_n + J_n)^R \quad | \quad (R_n^R + J_n^R)^{-1}I_n^R)$$
$$((R_n + J_n)^R)^{-1}G = (I_n \quad | \quad R_n + J_n).$$

■

Dalam kode *reversible self-dual*, sebuah kata kode x disebut *self-reverse* jika $x = x^R$ dan *self-reverse complementary* jika $x = x^{RC}$. Artinya, x bisa sama dengan x^R atau x^{RC}, tetapi tidak keduanya secara bersamaan. Kode C_{SR} terdiri dari semua kata kode yang *self-reverse*, sementara C_{SRC} terdiri dari semua kata kode yang *self-reverse complementary*. Secara formal x dikatakan *self-reverse* jika dan hanya jika $x + x^R = 0$ dan kata kode x dikatakan *self-reverse complementary* jika dan hanya jika $x + x^{RC} = 0$.

Algoritma untuk konstruksi kode DNA dari kode *reversible self-dual* sebagai berikut.

Algoritma 1. Konstruksi Kode DNA dengan batasan jarak Hamming, batasan terbalik, batasan komplemen terbalik, dan batasan konten GC tetap.

Input: Matriks generator $(I_n \quad | \quad M)$, dengan M dari $\{I_n, R_n, I_n + J_n, R_n + J_n\}$ dan bobot GC u
Output: Kode DNA D_{GC}
Procedure ConstructDNACode(generator_matrix, weight_gc)
    Hasilkan kode linier C dari matriks generator G
    Tentukan subkode *self-reverse* C_{SR} dari C
    Tentukan subkode *self-reverse complementary* C_{SRC} dari C
    $C_{baru} \leftarrow C - C_{SR} - C_{SRC}$
    Inisialisai kode DNA D
    For setiap kata kode $x_i \in C_{baru}$ do
        If $x_i^R \notin D$ and $x_i^{RC} \notin D$ then
            $D \leftarrow D \cup x_i$
        End if
    End for
    Inisialisasi D_{GC}
    For setiap kata kode $x_i \in D$ do
        If $wt_{GC}(x_i) == u$ then
            $D_{GC} \leftarrow D_{GC} \cup x_i$
        End if
    End for
    Return D_{GC}
End Procedure

Proposisi 3. Kode DNA atas $\mathbb{F}_2$ yang dibangun dari matriks generator $G = (I_n \quad I_n)$ mempunyai jarak minimum $d = 2$ untuk setiap n.

Bukti. Matriks generator $G = (I_n \quad | \quad I_n)$, dimana I_n adalah matriks identitas berukuran $n \times n$. Kode yang dihasilkan oleh matriks generator G adalah himpunan semua katakode $c = uG$ dimana $u \in \mathbb{F}_2^n$. Untuk setiap vektor $u \in \mathbb{F}_2^n$, kata kode $c = uG$ memiliki bentuk

$$c = uG = u(I_n \quad | \quad I_n) = (uI_n \quad | \quad I_n) = (u \quad | \quad u).$$

Artinya, kata kode c adalah penggabungan dua kali dari vektor u. Jarak minimum d dari kode dapat ditentukan dengan menemukan bobot minimum dari kata kode bukan nol dalam C. Jika dipilih u sebagai vektor satuan, yaitu $u = e_i$ dengan elemen ke-i bernilai 1 dan yang lainnya 0, maka kata kode c yang dihasilkan adalah

$$c = (0, \cdots, 0, 1, 0, \cdots, 0 \quad | \quad 0, \cdots, 0, 1, 0, \cdots, 0).$$

Dimana 1 hanya muncul pada posisi ke-i di kedua bagian u. Bobot dari kata kode ini adalah 2, karena ada dua posisi yang bernilai 1. Oleh karena itu, bobot minimum dari kata kode bukan nol dalam C adalah 2. Dengan demikian jarak minimum d untuk kode yang dihasilkan oleh matriks generator $G = (I_n \quad | \quad I_n)$ adalah $d = 2$.

■

Proposisi 4. Kode DNA atas $\mathbb{F}_2$ yang dibangun dari matriks generator $G = (I_n \quad | \quad R_n)$ mempunyai jarak minimum $d = 2$ untuk setiap n.

Bukti. Matriks generator $G = (I_n \quad | \quad R_n)$, dengan I_n sebagai matriks identitas berukurang $n \times n$, dan R_n sebagai matriks anti-diagonal berukuran $n \times n$. Kode yang dihasilkan oleh matriks generator G adalah himpunan kata kode $c = uG$ dimana $u \in \mathbb{F}_2^n$. Untuk seitap vektor $u \in \mathbb{F}_2^n$, kata kode c memiliki bentuk:

$$c = uG = u(I_n \quad | \quad R_n) = (u \quad | \quad uR_n).$$

Artinya, kata kode c adalah penggabungan dari vektor u dan hasil perkalian uR_n. Karena R_n adalah matriks anti-diagonal, setiap baris ke-i dari R_n hanya memiliki satu elemen 1, yang berada di kolom $n + 1 - i$, sedangkan elemen lainnya adalah 0. Oleh karena itu, jika $u = e_i$, dimaan e_i adalah vektor satuan yang hanya memiliki elemen ke-i bernilai 1, maka

$$uR_n = e_{n+1-i}.$$

Ini berarti kata kode c yang dihasilkan oleh vektor $u = e_i$ adalah

$$c = (e_i \quad | \quad e_{n+1-i}).$$

Memiliki bobot 2, karena ada dua posisi bernilai 1. Oleh karena itu, bobot minimum dari kata kode bukan nol dalam C adalah 2. Dengan demikian jarak minimum d untuk kode yang dihasilkan oleh matriks generator $G = (I_n \quad | \quad R_n)$ adalah $d = 2$.

■

Proposisi 5. Kode DNA atas $\mathbb{F}_2$ yang dibangun dari matriks generator $G = (I_n \quad | \quad I_n + J_n)$ mempunyai jarak minimum $d = 4$ untuk setiap n.

Bukti. Matriks generator $G = (I_n \quad | \quad I_n + J_n)$, dimana I_n adalah matriks identitas berukuran $n \times n$, dan J_n adalah matriks berukuran $n \times n$ yang semua elemennya 1. Kode yang dihasilkan oleh matriks generator G adalah himpunan semua kata kode $c = uG$ dimana $u \in \mathbb{F}_2^n$. Untuk setiap vektor $u \in \mathbb{F}_2^n$, kata kode c memiliki bentuk

$$c = uG = u(I_n \quad | \quad I_n + J_n) = (u \quad | \quad u + uJ_n) = (u \quad | \quad u + \mathbf{1}_n).$$

Dimana $\mathbf{1}_n$ adalah vektor yang semua elemennya 1. Misalkan $u = e_i$, dimana e_i adalah vektor satuan dengan elemen ke-i adalah 1 dan lainnya adalah 0. Maka kata kode c akan memiliki bentuk:

$$c = (e_i \quad | \quad e_i + \mathbf{1}_n).$$

Kata kode ini memiliki bobot n, yang diperoleh dari e_1 yang memiliki satu elemen 1, dan dari $e_i + 1_n$ yang memiliki $n - 1$ elemen 1. Namun, ini bukan bobot minimum. Misalkan diambil dua vektor satuan e_i dan e_j dengan $i \neq j$. c_i adalah kata kode yang dihasilkan dari e_iG, dan c_j adalah kata kode yang dihasilkan dari e_jG. Maka penjumlahan dari c_i dan c_j adalah

$$c_i + c_j = (1,0,\cdots,0 \quad | \quad 0,1,\cdots,1) + (0,1,0,\cdots,0 \quad | \quad 1,0,1\cdots,1)$$

$$c_i + c_j = (1,1,0,\cdots,0 \quad | \quad 1,1,0,\cdots,0).$$

Kata kode $c_i + c_j$ memiliki bobot 4, Oleh karena itu, bobot minimum dari kata kode bukan nol dalam C adalah 4. Dengan demikian jarak minimum untuk kode yang dihasilkan oleh matriks generator $G = (I_n \quad | \quad I_n + J_n)$ adalah $d = 4$.

■

Proposisi 6. Kode DNA atas $\mathbb{F}_2$ yang dibangun dari matriks generator $G = (I_n \quad | \quad R_n + J_n)$ mempunyai jarak minimum $d = 4$ untuk setiap n.

Bukti. Matriks generator $G = (I_n \quad | \quad R_n + J_n)$, dimana R_n adalah matriks anti-diagonal berukuran $n \times n$, dan J_n adalah matriks berukuran $n \times n$ yang semua elemenya 1. Kode yang dihasilkan oleh matriks generator G adalah himpunan kata kode $c = uG$ dimana $u \in \mathbb{F}_2^n$. Untuk setiap vektor $u \in \mathbb{F}_2^n$, kata kode c memiliki bentuk:

$$c = uG = u(I_n \quad | \quad R_n + J_n) = (uI_n \quad | \quad uR_n + uJ_n) = (u \quad | \quad uR_n + \mathbf{1}_n).$$

Dimana $\mathbf{1}_n$ adalah vektor yang semua elemennya 1. Karena R_n adalah matriks anti-diagonal, setiap baris ke-i dari R_n hanya memiliki satu elemen 1, yang berada di kolom $n + 1 - i$, sedangkan elemen lainnya 0. Misalkan $u = e_i$, dimana e_i adalah vektor satuan dengan elemen ke-i adalah 1 dan lainnya adlaah 0. Maka kata kode c akan memilik bentuk:

$$c = uG = (e_i \quad | \quad e_{n+1-i} + \mathbf{1}_n).$$

Kata kode ini memiliki bobot n, yang diperoleh dari e_i yang memiliki satu elemen 1, dan dari $e_{n+1-i} + 1_n$ yang memiliki $n - 1$ elemen 1. Namun, ini bukan bobot minimum. Misalkan diambil dua vektor satuan e_i dan e_j dengan $i \neq j$. c_i adlah kata kode yang dihasilkan dari e_iG, dan c_j adalah kata kode yang dihasilkan dari e_jG. Maka penjumlahan dari c_i dan c_j adalah

$$c_i + c_j = (1,0,\cdots,0 \quad | \quad 1,\cdots,1,0) + (0,1,0,\cdots,0 \quad | \quad 1,\cdots,1,0,1)$$

$$c_i + c_j = (1,1,0,\cdots,0 \quad | \quad 0,\cdots,0,1,1).$$

Kata kode $c_i + c_j$ memiliki bobot 4, Oleh karena itu, bobot minimum dari kata kode bukan nol dalam C adalah 4. Dengan demikian jarak minimum untuk kode yang dihasilkan oleh matriks generator $G = (I_n \quad | \quad R_n + J_n)$ adalah $d = 4$.

■

Program Python yang digunakan dalam penelitian ini untuk implementasi konstruksi kode DNA dapat diakses secara publik melalui repositori GitHub https://github.com/muharsyad/pydnacode. Di repositori tersebut, dapat ditemukan kode lengkap beserta dokumentasi yang menjelaskan cara penggunaan dan algoritma yang diterapkan. Dengan akses ke repositori ini, pengguna dapat melihat, mengunduh, dan memodifikasi kode sesuai kebutuhan mereka, serta berkontribusi pada pengembangan lebih lanjut dari proyek ini. Harapanya, program ini dapat berguna bagi peneliti dan praktisi di bidang bioteknologi dan teori koding yang tertarik dalam aplikasi kode DNA.

4. Simpulan

Penelitian ini menunjukkan bahwa konstruksi kode DNA atas lapangan hingga $\mathbb{F}_2$ dapat dilakukan dengan mempertimbangkan berbagai batasan kombinatorial, seperti jarak Hamming, batasan terbalik, batasan kompelemen terbali, dan batasan konten GC tetap. Studi literatur mengenai teori koding, kode linier, dan kode DNA memberikan landasan teori yang kuat dalam memahami dan mengimplementasikan konsep kode *reversible self-dual* dalam konstruksi kode DNA.

Hasil penelitian mengonfirmasi bahwa kode DNA yang dibangun menggunakan matriks generator $G = (I_n \quad M)$, dimana M dipilih dari $\{I_n, R_n, I_n + J_n, R_n + J_n\}$, mampu memenuhi semua batasan kombinatorial yang ditetapkan. Kode yang dihasilkan memiliki panjang $2n$, dimensi n untuk n bilangan genap, serta jarak minimum $d = 2$ dan $d = 4$, tergantung pada matriks generator yang digunakan.

Implementasi praktis menggunakan bahasa pemrograman Python telah dilakukan dalam menguji dan memvalidasi kode yang dihasilkan. Proses ini menunjukkan bahwa pendekatan yang menggabungkan teori koding dengan aplikasi praktis melalui pemrograman dapat mencapai hasil yang valid dan sesuai.

Penelitian ini membuka peluang untuk eksplorasi lebih lanjut, seperti pengembangan metode konstruksi yang dapat menghasilkan kode DNA dengan jarak minimum yang lebih besar, serta aplikasi kode DNA dalam konsteks teori koding.

Daftar Pustaka

Butar-Butar, J. L., Bukit, M. B. (2022). Metode Reversible Self-Dual Untuk Konstruksi Kode DNA Atas Lapangan Hingga $GF(4)$. *Jambura J. Math, 4*(2), 188-199.

Feng, B., Bai, S. S., Chen, B. Y., Zhou, X. N., (2015). The Constructions of DNA Codes from Linear Self-Dual Codes over $\mathbb{Z}_4$. *International Conference on Computer Information Systems and Industrial Applications*.

Gaborit, P., King, O. D. (2005). Linear Construction for DNA Codes. *Theoretical Computer Science, 334*, 99-113.

Huffman, C. W., Pless, V. (2003). Fundamentals of Error-Correcting Codes. *Cambridge University Press.*, New York.

Bernardo, A. B. I. (2002). Language And Mathematical Problem Solving Among Bilinguals. *The Journal of Psychology, 136*(3), 283-297.

Kim, H. J., Choi, W. H., Lee, Y. (2020). Construction of Reversible Self-Dual Codes. *Finite Fields and their Applications, 67*.

Kim, H. J., Choi, W. H., Lee, Y. (2021). Designing DNA Codes from Reversible Self-Dual Codes over $GF(4)$. *Discrete Mathematics, 344*.

Limbachiya, D., Rao, B., Gupta, M. K. (2016). The Art of DNA Strings: Sixteen Years of DNA Coding Theory. *CoRR*.

Varbanov, Z., Todorov, T., Hristova, M., (2014). A Method for Constructing DNA Codes from Additive Self-Dual Codes Over $GF(4)$. *ROMAI J, 10*(2), 203-211.