



Konstruksi Matriks Generator untuk Pembentukan Kode Linear Berbasis *Adjacency Matrix* dari Graf *Wheel*

Jovita Agustin*

Universitas Sebelas Maret, *Penulis Korespondensi: jovitaagustin_89@student.uns.ac.id

Putranto Hadi Utomo

Universitas Sebelas Maret

Diari Indriati

Universitas Sebelas Maret

ABSTRAK

Konstruksi kode linier berbasis matriks ketetanggaan (*adjacency matrix*) dari graf *wheel* bertujuan untuk menghasilkan kode dengan kemampuan deteksi dan koreksi kesalahan yang efisien dan sistematis. Pembentukan dimulai dari representasi graf *wheel* W_n , yang terdiri atas satu *vertex* pusat yang terhubung dengan seluruh *vertex* pada lingkaran serta keterhubungan antar *vertex* pada siklusnya. Struktur ini kemudian direpresentasikan dalam bentuk matriks ketetanggaan $A(W_n)$ yang mencerminkan hubungan antar *vertex* dalam graf. Berdasarkan matriks tersebut disusun matriks generator $G(W_n)$ dengan bentuk umum $G(W_n) = [I_k \mid A(W_n)]$, di mana I_k merupakan matriks identitas berukuran sama dengan jumlah *vertex* pada graf *wheel*. Dalam proses konstruksi dilakukan penghapusan baris identik pada matriks generator agar hanya tersisa baris yang saling bebas linear. Hasil analisis menunjukkan bahwa penghapusan tersebut menyebabkan nilai panjang kode n dan dimensi k menjadi lebih kecil, sedangkan jarak minimum d bergantung pada pola keterhubungan antar *vertex*, di mana perbedaan pola konektivitas dapat memengaruhi besar kecilnya jarak minimum kode. Nilai *rate* kode yang diperoleh menunjukkan efisiensi kode yang baik dengan kemampuan koreksi kesalahan yang tetap terjaga. Secara keseluruhan, konstruksi matriks generator berbasis graf *wheel* menghasilkan kode linier yang sistematis, memiliki struktur yang teratur dan simetris, serta memberikan keseimbangan antara efisiensi dan kemampuan koreksi kesalahan. Pendekatan ini juga membuka peluang pengembangan konstruksi kode berbasis graf lainnya dengan karakteristik yang lebih kompleks.

Kata kunci: Graf *wheel*, kode koreksi kesalahan, kode linier, matriks *adjacency*, matriks generator, parameter kode

PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi yang pesat menuntut tingkat keandalan yang tinggi dalam proses transmisi dan penyimpanan data digital. Dalam proses transmisi, data yang dikirimkan melalui suatu kanal komunikasi rentan mengalami gangguan yang dapat menyebabkan perubahan bit sehingga pesan yang diterima tidak sama dengan pesan yang dikirimkan. Kesalahan tersebut dapat terjadi akibat berbagai faktor seperti gangguan sinyal, interferensi, maupun kegagalan perangkat keras. Oleh karena itu, diperlukan suatu metode yang mampu menjaga keutuhan data dengan mendeteksi dan memperbaiki kesalahan yang terjadi selama proses transmisi maupun penyimpanan. Salah satu metode yang digunakan untuk mengatasi permasalahan tersebut adalah *error correcting code*. *Error correcting code* merupakan metode matematis yang menambahkan redundansi atau bit tambahan pada pesan asli sehingga kesalahan yang terjadi dapat dideteksi dan pada kondisi tertentu diperbaiki (Tomlinson *et al.*, 2017). Dengan mekanisme ini, sistem komunikasi tidak hanya mampu mengidentifikasi adanya kesalahan, tetapi juga berupaya mengembalikan data yang rusak ke bentuk pesan semula. *Error correcting code* banyak digunakan dalam berbagai sistem modern seperti jaringan komputer, komunikasi satelit, sistem telepon seluler, serta media penyimpanan data digital.

Dalam teori kode, salah satu *error correcting code* yang paling banyak digunakan adalah kode linier biner. Kode linier biner merupakan ruang vektor atas lapangan hingga dengan dua elemen, yaitu $\{0,1\}$ (Dani, 2018). Kode ini memiliki struktur aljabar yang jelas sehingga efisien dalam proses pengkodean (*encoding*) dan penguraian (*decoding*). Pembentukan kode linier umumnya dilakukan melalui matriks generator, yaitu matriks yang baris-barisnya membangkitkan seluruh *codeword* dalam suatu kode linier melalui kombinasi linear. Seiring dengan berkembangnya penelitian di bidang *error correcting code*, berbagai pendekatan matematis mulai dieksplorasi untuk membangun matriks generator, salah satunya melalui teori graf (Albert & Faist, 2024). Pendekatan ini memanfaatkan *adjacency matrix* atau matriks ketetanggaan dari suatu graf, yang merepresentasikan hubungan antar *vertex* menggunakan elemen biner 0 dan 1 (Nasrul *et al.*, 2023). Kode linier biner juga disusun atas elemen biner sehingga terdapat keterkaitan yang kuat antara *adjacency matrix* suatu graf dan struktur matriks generator. Keterkaitan ini memungkinkan *adjacency matrix* digunakan sebagai dasar dalam konstruksi matriks generator kode linier.

Beberapa penelitian sebelumnya menunjukkan bahwa kode linier dapat dibangun dari *adjacency* matriks maupun *incident* matriks berbagai jenis graf. Key dan Rodrigues (2018) membahas konstruksi kode linier yang dibangun dari *adjacency matrix* graf umum serta mengaitkan parameter graf dengan parameter kode yang dihasilkan. Selain itu, Kopparty *et al.* (2024) memperkenalkan konsep *error correcting graph codes* yang menegaskan bahwa struktur graf memiliki peran penting dalam pengembangan kode koreksi kesalahan. Kajian lebih lanjut mengenai pemanfaatan graf dalam pembentukan kode linier dilakukan oleh Shaikh, Jain, dan Reddy (2024) yang mengonstruksi kode linier dari matriks insidensi graf unit $G(\mathbb{Z}_n \oplus \mathbb{Z}_m)$. Penelitian tersebut menganalisis sifat graf unit, seperti keterhubungan dan kebipartitan, serta menentukan parameter kode linier dan kode dual yang dihasilkan, termasuk jarak minimum. Meskipun demikian, secara umum penelitian-penelitian tersebut masih banyak berfokus pada graf-graf tertentu, seperti graf lengkap, graf siklik, dan graf dengan struktur aljabar khusus, sehingga kajian terhadap graf lain dengan struktur berbeda masih relatif terbatas.

Berdasarkan hal tersebut, graf *wheel* dipilih dalam penelitian ini karena memiliki struktur ketetanggaan yang khas dan berbeda dari graf-graf yang umum digunakan dalam konstruksi kode linier. Graf *wheel* dibentuk dari sebuah graf siklik dengan penambahan satu *vertex* pusat yang terhubung ke seluruh *vertex* pada siklus tersebut (Rosen, 2016). Struktur ini menghasilkan *adjacency matrix* dengan pola yang teratur dan simetris, yang berpotensi memengaruhi pembentukan matriks generator serta parameter kode linier yang dihasilkan, seperti panjang kode, dimensi, dan jarak minimum. Dengan memanfaatkan *adjacency matrix* graf *wheel* sebagai dasar konstruksi matriks generator, penelitian ini bertujuan untuk membentuk kode linier biner dan menganalisis karakteristik *error correcting code* yang dihasilkan. Kajian ini diharapkan dapat memberikan kontribusi dalam pengembangan *error correcting code* berbasis graf, khususnya melalui pemanfaatan graf *wheel*, serta memperkaya variasi konstruksi kode linier biner dalam kajian teori kode.

METODE

Metode penelitian yang digunakan dalam penelitian ini adalah studi literatur (*literature study*). Metode ini dipilih karena penelitian bersifat teoretis dan bertujuan untuk mengkaji konsep-konsep matematis serta melakukan konstruksi model berdasarkan teori yang sudah ada tanpa melibatkan data empiris maupun eksperimen lapangan. Seluruh analisis dalam penelitian ini dilakukan secara matematis dan konseptual. Studi literatur dilakukan untuk memperoleh pemahaman mendalam mengenai teori dasar yang mendukung penelitian, serta menelusuri hasil-hasil penelitian terdahulu yang relevan dengan topik konstruksi matriks generator berbasis *adjacency matrix* dari graf *wheel*. Studi literatur atau sering disebut juga

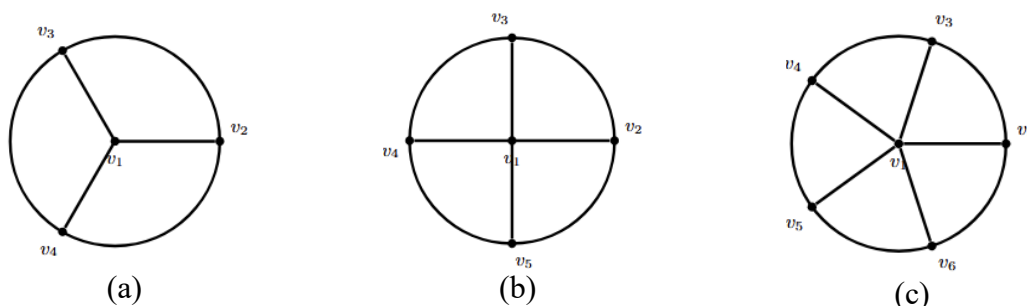
studi kepustakaan, merupakan teknik pengumpulan data dan informasi dengan menelaah berbagai sumber tertulis yang kredibel dan relevan (Febrianto *et al.*, 2024). Sumber-sumber tersebut dapat berupa jurnal ilmiah, buku referensi, laporan penelitian, tesis, disertasi, maupun publikasi daring yang berhubungan dengan objek penelitian. Melalui metode ini, peneliti dapat memperoleh landasan teoritis yang kuat dan memperkuat argumentasi konseptual dalam proses analisis. Dalam penelitian ini, studi literatur dilakukan dengan cara menelusuri dan mengumpulkan referensi dari buku teks teori kode linier, teori graf, serta artikel yang membahas hubungan antara graf dan kode koreksi kesalahan. Hasil penelusuran ini digunakan sebagai dasar dalam menyusun proses konstruksi matriks generator berbasis *adjacency matrix* dari graf *wheel*. Langkah-langkah yang dilakukan dalam penelitian ini diuraikan sebagai berikut.

1. Mempelajari konsep dasar dan definisi yang berkaitan dengan teori kode linier, *error correcting code*, graf *wheel*, dan *adjacency matrix* untuk memperoleh pemahaman terhadap topik penelitian.
2. Membangun graf *wheel* W_n untuk beberapa nilai n , kemudian merepresentasikannya ke dalam bentuk *adjacency matrix* yang memuat hubungan antar *vertex*, selanjutnya menganalisis pola dari matriks *adjacency* graf *wheel*.
3. Mengkonstruksi matriks generator kode linier dengan memanfaatkan *adjacency matrix* yang telah diperoleh. Matriks generator dibentuk dalam bentuk umum $G = [I_k \mid A]$, di mana I_k adalah matriks identitas dan A merupakan *adjacency matrix* dari graf *wheel*.
4. Melakukan perhitungan parameter kode, yaitu panjang kode (n), dimensi (k), jarak minimum (d), dan rate (R). Jarak minimum ditentukan dengan menghitung jarak Hamming, yaitu banyaknya posisi berbeda antara dua *codeword*.

Melalui tahapan tersebut, penelitian ini diharapkan dapat memberikan pemahaman teoretis mengenai hubungan antara struktur graf *wheel* dengan karakteristik kode linier yang dihasilkannya, serta menunjukkan potensi pemanfaatan *adjacency matrix* dalam pembentukan kode koreksi kesalahan.

HASIL DAN PEMBAHASAN

Struktur graf sering dimanfaatkan dalam berbagai bidang matematika karena mampu merepresentasikan hubungan antar objek secara informatif. Melalui graf, hubungan antar *vertex* dan *edge* dapat digambarkan secara visual maupun aljabar, sehingga memudahkan analisis pola keterhubungan suatu sistem. Salah satu bentuk graf yang memiliki struktur khas adalah *graf wheel*. Graf *wheel* yang dinotasikan dengan W_n adalah graf yang dibentuk melalui operasi *join* antara graf siklus C_n dengan graf lengkap K_1 , sehingga dapat dituliskan sebagai $W_n = C_{n-1} + K_1$ (Gallian, 2019). Graf ini terdiri atas satu *vertex* pusat yang terhubung ke seluruh *vertex* pada *cycle* C_n , sehingga membentuk struktur menyerupai roda dengan jari-jari. Ilustrasi graf *wheel* disajikan pada Gambar 1.



Gambar 1. (a) graf wheel W_4 , (b) graf wheel W_5 , (c) graf wheel W_6

Representasi graf *wheel* pada Gambar 1 dapat diubah ke dalam bentuk aljabar untuk memudahkan analisis struktur dan pola keterhubungan antar *vertex*. Salah satu bentuk representasi yang umum digunakan adalah matriks ketetanggaan (*adjacency matrix*). Menurut Chatrand *et al.* (2016), misalkan G adalah graf berorder n dengan himpunan *vertex* $V(G) = \{v_1, v_2, \dots, v_n\}$. *Adjacency matrix* dari graf G adalah matriks berukuran $n \times n$, dilambangkan dengan $A(G) = [a_{ij}]$ dimana

$$a_{ij} = \begin{cases} 1, & \text{jika } v_i v_j \in E(G), \\ 0, & \text{jika } v_i v_j \notin E(G). \end{cases}$$

Matriks *adjacency* ini berfungsi untuk merepresentasikan hubungan antar *vertex* pada suatu graf sehingga pola keterhubungan dapat dianalisis secara sistematis. Bentuk representasi seperti ini tidak hanya berguna dalam analisis graf, tetapi juga dapat dimanfaatkan dalam pembentukan kode linier biner karena struktur biner pada matriks *adjacency* sesuai dengan sistem bilangan yang digunakan dalam teori kode linier. Dengan demikian, graf *wheel* dapat dijadikan dasar dalam konstruksi kode linier biner melalui pembentukan matriks generator yang bersumber dari matriks *adjacency*-nya. Untuk memberikan gambaran yang lebih jelas mengenai matriks *adjacency* graf *wheel*, berikut matriks *adjacency* dari beberapa graf *wheel*.

1. Matriks *adjacency* graf W_4

$$A(W_4) = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}.$$

2. Matriks *adjacency* graf W_5

$$A(W_5) = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix}.$$

3. Matriks *adjacency* graf W_6

$$A(W_6) = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

Dari *adjacency* matriks tersebut juga dapat diamati bahwa pola keterhubungan antar *vertex* pada setiap graf *wheel* memiliki bentuk yang teratur dan berulang. Dari pola tersebut dapat diperoleh rumus umum untuk menentukan nilai setiap elemen a_{ij} pada matriks *adjacency* berdasarkan posisi *vertex* v_i dan v_j . Rumus umum matriks *adjacency* graf *wheel* dinyatakan sebagai berikut

$$a_{i,j} = \begin{cases} 1, & \text{jika } (i = 1 \text{ dan } j > 1) \text{ atau } (i > 1 \text{ dan } j = 1), \\ 1, & \text{jika } (i, j) \in \{(k, k+1), (k+1, k)\}, \quad k = 2, 3, \dots, n-1, \\ 1, & \text{jika } (i, j) = (2, n) \text{ atau } (n, 2), \\ 0, & \text{untuk lainnya.} \end{cases}$$

Rumus umum yang telah diperoleh menunjukkan pola keterhubungan antar *vertex* pada graf *wheel* yang menjadi dasar dalam penyusunan matriks *adjacency*. Dengan demikian, matriks *adjacency* dapat disusun tanpa perlu menggambar grafnya secara langsung. Rumus tersebut juga memungkinkan pembentukan matriks *adjacency* untuk graf *wheel* dengan berapapun jumlah *vertex*, sehingga struktur matriks untuk graf W_n dapat dibangun secara umum dan teratur. Dari matriks *adjacency* graf *wheel* yang telah diperoleh, selanjutnya dibentuk matriks generator yang digunakan untuk membangun suatu kode linier biner untuk kode koreksi kesalahan. Menurut Pless (1998), matriks generator dari suatu kode C adalah matriks G berukuran $k \times n$, di mana setiap baris pada G merupakan vektor basis dari subruang kode tersebut. Menurut Mallik dan Yildiz (2021), matriks generator memiliki bentuk umum $G = [I_k | A]$ dengan I_k merupakan matriks identitas berukuran $k \times k$ dan A merupakan matriks yang berasal dari struktur tertentu, dalam penelitian ini matriks A adalah matriks *adjacency* dari graf *wheel*. Pada saat membangun matriks generator apabila terdapat dua atau lebih baris pada matriks A yang identik, maka baris-baris tersebut merepresentasikan vektor yang sama dalam ruang vektor biner. Hal dapat menyebabkan jumlah baris efektif pada matriks generator menjadi lebih banyak dari yang diperlukan. Oleh karena itu, setiap baris identik pada matriks A harus dihapus agar matriks generator hanya terdiri dari vektor-vektor basis yang saling bebas linier. Penghapusan baris identik ini penting untuk menjamin bahwa setiap baris dalam matriks generator memberikan informasi yang berbeda terhadap pembentukan kode linier biner. Dengan demikian, matriks generator yang dihasilkan akan memiliki struktur yang efisien. Berikut merupakan bentuk umum matriks generator dari graf *wheel*.

1. Matriks generator W_4

$$G(W_4) = \left[\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{array} \right].$$

2. Matriks generator W_5

$$G(W_5) = \left[\begin{array}{ccc|cccc} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \end{array} \right].$$

3. Matriks generator w_6

$$G(W_6) = \left[\begin{array}{cccccc|cccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \end{array} \right].$$

Dengan demikian, struktur dari graf *wheel* dapat diubah menjadi representasi aljabar dalam bentuk kode linier yang memiliki kemampuan tertentu dalam mendeteksi maupun mengoreksi kesalahan pada proses transmisi data. Representasi ini diperoleh melalui pembentukan matriks generator $G(W_n)$ yang dibangun berdasarkan matriks *adjacency* dari graf *wheel* W_n . Matriks generator tersebut berfungsi sebagai dasar untuk membentuk seluruh himpunan kata kode (*codewords*) dalam suatu sistem kode linier. Setelah matriks generator $G(W_n)$ terbentuk, tahap berikutnya adalah menentukan parameter-parameter utama dari kode linier yang dihasilkan, yang meliputi panjang kode (n), dimensi kode (k), jarak minimum (d), dan Rate (R).

Menurut Fish *et al.* (2021), panjang kode yang dinotasikan dengan n adalah banyaknya elemen yang membentuk suatu kata kode. Semakin besar nilai n semakin panjang pula kata

kode yang terbentuk. Dimensi kode yang dinotasikan dengan k adalah banyaknya elemen informasi yang dapat dipilih bebas untuk membentuk setiap kata kode dalam suatu kode linier (Hao *et al.*, 2020). Nilai k sama dengan jumlah baris pada matriks generator, karena setiap baris merepresentasikan satu vektor basis dalam ruang kode. Semakin besar nilai k , semakin banyak informasi yang dapat dikodekan tanpa mengubah struktur dasar kode linier. Jarak minimum yang dinotasikan dengan d merupakan jarak terkecil antara kata kode yang berbeda (Tsunoda *et al.*, 2016). Jarak minimum juga bisa disebut sebagai jarak Hamming minimum. Nilai d menentukan seberapa besar kemampuan kode linier dalam mendeteksi dan mengoreksi kesalahan (Haymaker dan McMillon, (2023)). Untuk menentukan jarak minimum pada kode linier dari matriks generator, tentukan seluruh kemungkinan *codeword* yang dihasilkan dari setiap pesan biner p . Selanjutnya, bobot Hamming dari masing-masing *codeword* dihitung. Jarak minimum kode diperoleh dari nilai bobot Hamming terkecil selain nol. Berikut beberapa *codeword* dari matriks generator $G(W_4)$ beserta bobot Hammingnya ditunjukkan pada Tabel 2.

Tabel 2. Daftar *codeword* matriks $G(W_4)$

Pesan (p)	<i>Codeword</i> (c)	Bobot hamming
0000	00000000	0
0001	00011110	4
0010	00101101	4
0011	00110011	4
0100	01001011	4
0101	01010101	4
0110	01100110	4
0111	01111000	4
1000	10000111	4
1001	10011001	4
1010	10101010	4
1011	10110100	4
1100	11001100	4
1101	11010010	4
1110	11100001	4
1111	11111111	8

Berdasarkan Tabel 2, bobot Hamming paling kecil selain nol adalah 4, maka jarak minimum untuk matriks generator $G(W_4)$ adalah 4. Selain dengan jarak minimum, salah satu cara lain untuk menilai kualitas suatu kode adalah dengan melihat laju kode atau *rate*-nya. Laju kode menggambarkan perbandingan antara banyaknya simbol informasi yang dikirim dengan panjang keseluruhan kata kode, yang secara matematis dinyatakan dengan $R = \frac{k}{n}$, $0 < R \leq 1$, dengan k jumlah simbol informasi dan n panjang kata kode (Lin and Li, 2022). Nilai R menunjukkan keseimbangan antara efisiensi dan ketahanan kode. Jika nilai R semakin besar maka semakin sedikit redundansi yang digunakan sehingga kode menjadi lebih efisien tetapi kemampuan untuk mendeteksi dan mengoreksi kesalahan akan menurun. Sebaliknya, jika nilai R semakin kecil, maka redundansi pada kode menjadi lebih banyak sehingga beban efisiensi meningkat, tetapi kemampuan dalam mendeteksi dan memperbaiki kesalahan menjadi lebih kuat. Berdasarkan hasil konstruksi dari beberapa graf *wheel*, diperoleh parameter n , k , d , dan R seperti ditunjukkan pada Tabel 2.

Berdasarkan Tabel 2, setiap graf *wheel* menghasilkan parameter kode linear yaitu panjang kode, dimensi, dan jarak minimum yang berbeda. Perbedaan nilai tersebut berkaitan dengan variasi struktur graf dan karakteristik matriks adjacency yang terbentuk. Pada graf W_4 diperoleh nilai $n = 8$, $k = 4$, dan $d = 4$. Nilai tersebut menunjukkan bahwa setengah dari panjang kode merupakan bit informasi, sedangkan sisanya merupakan bit redundan

yang terbentuk dari kombinasi linear antar baris pada matriks *adjacency*. Struktur graf yang relatif sederhana menyebabkan tidak banyak baris identik pada matriks, sehingga nilai k tetap tinggi dengan laju kode $R = 0.5$. Graf W_5 menghasilkan parameter $n = 8$, $k = 3$, dan $d = 3$.

Tabel 2. Parameter graf wheel

Graf	n	k	d	R
W_4	8	4	4	0.5
W_5	8	3	3	0.375
W_6	12	6	4	0.5

Penurunan nilai k dibandingkan graf sebelumnya disebabkan oleh adanya beberapa baris pada matriks *adjacency* yang identik atau saling bergantung secara linear. Baris yang sama tersebut tidak menambah informasi baru terhadap ruang vektor kode, sehingga perlu dihapus pada proses pembentukan matriks generator. Penghapusan ini mengurangi jumlah baris yang bebas linear dan berdampak pada penurunan nilai k serta laju kode yang menjadi $R = 0.375$. Selanjutnya, graf W_6 menghasilkan nilai $n = 12$, $k = 6$, dan $d = 4$. Meskipun ukuran matriks *adjacency* bertambah, proses eliminasi baris identik tetap menjaga keseimbangan antara bit informasi dan bit redundan. Dengan demikian, diperoleh laju kode $R = 0.5$ yang menunjukkan efisiensi yang serupa dengan kode hasil graf W_4 . Secara keseluruhan parameter (n, k, d) pada ketiga graf menunjukkan bahwa struktur graf *wheel* berpengaruh terhadap hubungan linear antar baris pada matriks *adjacency* yang digunakan dalam pembentukan kode linear. Semakin banyak baris yang identik atau bergantung secara linear, semakin kecil nilai k yang dihasilkan. Dengan kata lain, keunikan pola keterhubungan pada graf *wheel* menentukan keseimbangan antara bit informasi dan bit redundan pada kode yang dibentuk, sehingga memengaruhi efisiensi dan karakteristik dari kode linear yang dihasilkan.

SIMPULAN

Berdasarkan hasil konstruksi dan analisis yang dilakukan, dapat disimpulkan bahwa graf *wheel* dapat dijadikan dasar pembentukan kode linier melalui matriks generator berbentuk $[I \mid A]$, dengan I sebagai matriks identitas dan A sebagai matriks *adjacency* dari graf tersebut. Struktur graf *wheel* yang simetris dan teratur mempermudah proses pembentukan matriks generator serta menghasilkan pola keterhubungan yang sistematis. Kode linier yang diperoleh memiliki parameter $[n, k, d]$ yang dipengaruhi oleh jumlah *vertex* dan pola keterhubungan graf. Nilai panjang kode n ditentukan oleh banyaknya kolom pada matriks generator, sedangkan dimensi kode k bergantung pada jumlah baris yang bebas linear. Dalam proses konstruksi dilakukan penghapusan baris identik agar matriks generator hanya terdiri atas vektor-vektor basis yang saling bebas linear. Penghapusan tersebut menyebabkan nilai n dan k menjadi lebih kecil dibandingkan sebelum dilakukan penyederhanaan baris.

Jarak minimum d dipengaruhi oleh pola keterhubungan antar *vertex*, di mana semakin kompleks struktur graf maka nilai d cenderung meningkat sehingga kemampuan koreksi kesalahan menjadi lebih baik. Selain itu, *rate* kode menunjukkan efisiensi kode yang dihasilkan. Secara keseluruhan perbedaan struktur graf *wheel* serta proses penghapusan baris identik berpengaruh langsung terhadap panjang kode, dimensi, dan jarak minimum. Oleh karena itu, pendekatan berbasis graf *wheel* memberikan konstruksi kode linier yang sistematis dan efisien untuk penerapan dalam sistem pengkodean serta koreksi kesalahan. Pendekatan konstruksi matriks generator berbasis graf pada penelitian ini dapat diperluas dengan menggunakan graf yang lebih kompleks, seperti graf helm (*helm graph*), graf roda gigi (*gear graph*), atau graf tak beraturan (*irregular graph*), untuk memperoleh kode linier

dengan nilai jarak minimum d yang lebih besar dan kemampuan koreksi kesalahan yang lebih optimal.

DAFTAR PUSTAKA

- Albert, V. V., & Faist, P. (2024). Graph adjacency code. *The Error Correction Zoo*. <https://errorcorrectionzoo.org/c/graph>
- Bierbrauer, J. (2004). *Introduction to coding theory*. Chapman & Hall/CRC.
- Chartrand, G., Lesniak, L., & Zhang, P. (2016). *Graphs and digraphs* (6th ed.). CRC Press.
- Dani, A. A. I. (2018). Sifat kode linier biner berdasarkan bobot kode. *Pena Teknik: Jurnal Ilmiah Ilmu-Ilmu Teknik*, 13, 1–24.
- Febrianto, A., Siroj, R. A., & Hartatiana. (2024). Studi literatur: Landasan dalam memilih metode penelitian yang tepat. *Journal of Educational Research and Development*, 1(2), 259–263.
- Fish, W., Key, J. D., & Mwambene, E. (2021). Special LCD codes from products of graphs. *Applicable Algebra in Engineering, Communication and Computing*, 1–27.
- Gallian, J. A. (2019). A dynamic survey of graph labeling (22nd ed.). *The Electronic Journal of Combinatorics*.
- Ganagi, A. B., & Ramane, H. S. (2016). Hamming distance between the strings generated by adjacency matrix of a graph and their sum. *Algebra and Discrete Mathematics*, 82–93.
- Hao, J., Huang, H., Livshyts, G., & Tikhomirov, K. (2020). Distribution of the minimum distance of random linear codes. In *2020 IEEE International Symposium on Information Theory (ISIT)* (pp. 114–119). IEEE.
- Haymaker, K., & McMillon, E. (2023). Parity-check codes from disjoint matrices. *arXiv*. <https://arxiv.org/abs/2311.17262>
- Key, J. D., & Rodrigues, B. G. (2018). LCD codes from adjacency matrices of graphs. *Applicable Algebra in Engineering, Communication and Computing*, 29(3–4), 351–362.
- Kopparty, S., Potukuchi, A., & Sha, H. (2024). *Error correcting graph codes*. University of Toronto & York University.
- Lin, S., & Li, J. (2022). *Fundamentals of classical and modern error-correcting codes*. Cambridge University Press.
- Mallik, S., & Yildiz, B. (2021). Isodual and self-dual codes from graphs. *arXiv*.
- Nasrul, N., Rahmawati, D., & Pratama, R. (2023). Representasi adjacency matrix dalam analisis struktur graf biner. *Jurnal Matematika dan Aplikasi*, 21(2), 95–104.
- Pless, V. (1998). *Introduction to the theory of error-correcting codes* (3rd ed.). John Wiley & Sons.
- Rosen, K. H. (2016). *Discrete mathematics and its applications* (8th ed.). McGraw-Hill Education.
- Shaikh, W. M., Jain, R. S., & Reddy, B. S. (2024). Construction of linear codes from the unit graph $G(\mathbb{Z}_n \oplus \mathbb{Z}_m)$. *arXiv*.
- Tomlinson, M., Tjhai, C. J., Ambroze, M. A., Ahmed, M., & Jibril, M. (2017). *Error correction coding and decoding: Bounds, codes, decoders, analysis and applications*. Springer.
- Tsunoda, Y., Fujiwara, Y., Ando, H., & Vandendriessche, P. (2016). Bounds on separating redundancy of linear codes and rates of X-codes. *IEEE Transactions on Information Theory*.