



**PENERAPAN LOGARITMA ZECH DALAM KONSTRUKSI BARISAN DE BRUIJN
TERMODIFIKASI MELALUI METODE PENGGABUNGAN SIKLUS**

***APPLICATION OF ZECH'S LOGARITHM IN THE CONSTRUCTION OF MODIFIED DE
BRUIJN SEQUENCES VIA A CYCLE-JOINING METHOD***

Muhammad Nurcahyo Eko Saputra*, Prodi Matematika FMIPA UNY

Musthofa, Prodi Matematika FMIPA UNY

*e-mail korespondensi: muhammadnurcahyo.2022@student.uny.ac.id

Abstrak

Barisan de Bruijn merupakan barisan siklik biner yang memuat seluruh kemungkinan n -tuple tepat satu kali dalam satu periode. Pada barisan de Bruijn klasik terdapat *run nol* terpanjang sepanjang n yang pada beberapa aplikasi dianggap kurang ideal, sehingga dikembangkan barisan de Bruijn termodifikasi dengan menghilangkan satu nol pada *run* tersebut. Penelitian ini bertujuan untuk menerapkan logaritma Zech dalam konstruksi barisan de Bruijn termodifikasi melalui metode penggabungan siklus. Metode yang digunakan diawali dengan membangkitkan m -sequence menggunakan *Linear Feedback Shift Register* (LFSR) dengan polinomial primitif, kemudian dilakukan t -desimasi untuk memperoleh beberapa barisan periodik. Selanjutnya, Algoritma Berlekamp–Massey digunakan untuk menentukan polinomial minimal *irreducible* non-primitif yang menghasilkan siklus-siklus terpisah. Siklus-siklus tersebut digabung menggunakan metode penggabungan siklus melalui penukaran *successor* pada pasangan konjugat. Dalam proses tersebut, logaritma Zech digunakan untuk menentukan pasangan konjugat dan posisi *state* secara lebih sistematis dengan merepresentasikan setiap *state* dalam bentuk a^i . Hasil penelitian menunjukkan bahwa penggunaan logaritma Zech dapat mempermudah dan meningkatkan efisiensi proses penggabungan siklus, terutama untuk orde besar dengan jumlah siklus yang banyak. Barisan akhir yang diperoleh merupakan barisan de Bruijn termodifikasi dengan periode $2^n - 1$.

Kata kunci: barisan de Bruijn termodifikasi, LFSR, logaritma Zech, metode penggabungan siklus, t -desimasi.

Abstract

A de Bruijn sequence is a cyclic binary sequence that contains every possible n -tuple exactly once within a single period. In the classical de Bruijn sequence, there exists a longest run of zeros of length n , which is considered less desirable in certain applications. Therefore, a modified de Bruijn sequence is developed by removing one zero from the run. This research aims to apply Zech's logarithms in the construction of modified de Bruijn sequences using the cycle joining method. The method begins by generating an m -sequence using a Linear Feedback Shift Register (LFSR) with a primitive polynomial, followed by t -decimation to obtain several periodic sequences. Furthermore, the Berlekamp–Massey Algorithm is applied to determine a non-primitive irreducible minimal polynomial that produces several disjoint cycles. These cycles are then combined using the cycle joining method through successor swapping on conjugate pairs. In this process, Zech's logarithms are utilized to determine conjugate pairs and state positions more systematically by representing each state in the form a^i . The results show that the use of Zech's logarithms simplifies and improves the efficiency of the cycle joining process, especially for higher orders with a large number of cycles. The final result is a modified de Bruijn sequence with period $2^n - 1$.

Keywords: modified de Bruijn sequence, LFSR, Zech's logarithms, cycle joining method, t -decimation.

PENDAHULUAN

Barisan de Bruijn adalah barisan siklik biner yang dibentuk sedemikian sehingga setiap n -tuple muncul tepat satu kali sebagai *state* berurutan dengan panjang n . Barisan ini pertama

kali muncul dari permasalahan kombinatorika yang diajukan oleh A. de Riviere pada akhir abad ke-19 dan kemudian dipopulerkan kembali oleh Nicolaas Govert de Bruijn (1946). Secara umum, suatu barisan de Bruijn memiliki panjang 2^n yang memuat seluruh n -tuple dengan panjang n (Zhu et al., 2021).

Selain sifat kelengkapannya, kualitas suatu barisan de Bruijn juga dapat dianalisis dari kompleksitas linear dan struktur lokal barisan tersebut. Chan et al. (1982) mendefinisikan kompleksitas linear sebagai derajat dari suatu rekurensi linear terpendek atau polinomial minimal yang menghasilkan suatu barisan. Nilai kompleksitas linear yang lebih besar menunjukkan bahwa barisan tersebut sulit untuk diprediksi. Pada barisan de Bruijn klasik sendiri terdapat tepat satu *run* nol sepanjang n yang dalam beberapa aplikasi khususnya di bidang kriptografi dianggap kurang ideal (Musthofa et al., 2022).

Untuk mengatasi masalah tersebut dikembangkan suatu barisan de Bruijn termodifikasi dengan menghilangkan satu angka nol pada *run* nol terpanjang. Akibatnya barisan yang dihasilkan memiliki periode $2^n - 1$ dan tidak lagi memuat *state* nol sepanjang n . Untuk mengkonstruksi suatu barisan de Bruijn termodifikasi, salah satu metode yang bisa dipakai adalah metode penggabungan siklus. Menurut Chang et al. (2021), metode penggabungan siklus digunakan untuk menggabungkan siklus-siklus terpisah yang terbentuk dari hasil pembangkitan LFSR menggunakan polinomial *irreducible* non-primitif. Siklus-siklus ini digabung dengan mencari pasangan konjugat masing-masing siklus, lalu menukar *successor*-nya sehingga semua siklus digabung menjadi satu siklus yang lebih besar.

Namun pencarian pasangan konjugat secara manual kurang efisien terutama untuk orde besar dengan jumlah siklus yang banyak. Maka dari itu dibutuhkan sebuah cara untuk menentukan pasangan konjugat masing-masing siklus tanpa harus mencarinya satu-satu. Salah satu cara yang bisa digunakan adalah dengan mengaplikasikan logaritma Zech. Chang et al. (2021) menjelaskan penerapan logaritma Zech dalam konstruksi suatu barisan de Bruijn dengan panjang 2^n melalui metode penggabungan siklus. Pada penelitian akan dibahas mengenai penerapan logaritma Zech pada metode penggabungan siklus untuk membantu menentukan pasangan konjugat secara lebih sistematis dan efisien dalam konstruksi barisan de Bruijn termodifikasi.

METODE

Bentuk penelitian ini berupa tinjauan pustaka. Mahanum (2021) menjelaskan bahwa tinjauan pustaka merupakan aktivitas untuk meninjau atau mengkaji kembali pustaka-pustaka yang sebelumnya telah dipublikasikan oleh akademisi atau peneliti lain. Pustaka tersebut berkaitan dengan topik yang akan diteliti. Data yang digunakan dalam penelitian ini adalah berbagai pustaka baik yang berupa artikel dari jurnal, buku, ataupun dokumen lainnya yang relevan dengan topik penelitian, yaitu penerapan logaritma Zech dalam konstruksi barisan de Bruijn termodifikasi melalui metode penggabungan siklus. Berikut adalah tahapan-tahapan yang dilakukan dalam penelitian ini:

1. Mengumpulkan berbagai pustaka baik yang berupa artikel dari jurnal, buku, ataupun dokumen lainnya yang relevan dengan topik penelitian, yaitu penerapan logaritma Zech dalam konstruksi barisan de Bruijn termodifikasi melalui metode penggabungan siklus.
2. Berbagai pustaka yang sudah dikumpulkan sebelumnya, baik berupa artikel dari jurnal, buku, maupun dokumen lainnya yang relevan dengan topik penelitian, yaitu penerapan logaritma Zech dalam konstruksi barisan de Bruijn termodifikasi melalui metode penggabungan siklus, akan digunakan untuk mempelajari dan memahami berbagai

konsep yang melandasi topik seperti *Linear Feedback Shift Register* (LFSR), barisan de Bruijn, barisan de Bruijn termodifikasi dan t -desimasi.

3. Mendeskripsikan penerapan metode penggabungan siklus untuk memperoleh barisan de Bruijn termodifikasi.
4. Mendeskripsikan pemanfaatan logaritma Zech pada metode penggabungan siklus.
5. Mengkonstruksi barisan de Bruijn termodifikasi dengan menerapkan metode penggabungan siklus dan logaritma Zech.

Selanjutnya, di bawah ini adalah konsep-konsep yang melandasi topik penelitian seperti *Linear Feedback Shift Register* (LFSR), barisan de Bruijn, barisan de Bruijn termodifikasi dan t -desimasi.

Definisi 1. (Golomb & Gong, 2005) Suatu barisan $a = \{a_i\}, i \geq 0$ dengan $a_i \in GF(2)$ disebut sebagai barisan LFSR jika terdapat koefisien $c_0, c_1, \dots, c_{n-1} \in GF(2)$ yang memenuhi:

$$a_{k+n} = c_0 a_k + c_1 a_{k+1} + \dots + c_{n-1} a_{k+n-1}, k = 0, 1, 2, \dots$$

di mana $(a_0, a_1, \dots, a_{n-1})$ adalah kondisi awal atau disebut *initial state*.

Untuk menyederhanakan penulisan, setiap barisan biner $(a_0, a_1, a_2, \dots)$ selanjutnya ditulis sebagai $a_0 a_1 a_2 \dots$ dan *state* $(a_0, a_1, \dots, a_{n-1})$ ditulis sebagai $a_0 a_1 \dots a_{n-1}$. Berdasarkan definisi di atas dapat dipahami bahwa setiap elemen dalam barisan tersebut memiliki relasi, yaitu setiap elemen baru diperoleh dari kombinasi linear elemen-elemen sebelumnya. Relasi rekurensi linear tersebut dapat dinyatakan secara lebih ringkas melalui polinomial karakteristik.

Definisi 2. (Golomb & Gong, 2005) Polinomial karakteristik dari suatu LFSR adalah polinomial atas $GF(2)$ yang berbentuk:

$$p(x) = c_0 + c_1 x + c_2 x^2 + \dots + c_{n-1} x^{n-1} + x^n$$

di mana $c_i \in GF(2)$ adalah koefisien dari relasi rekurensi LFSR.

Contoh 1. Misalkan suatu LFSR dengan polinomial karakteristik $x^3 + x + 1$ atas $GF(2)$. Berdasarkan definisi dari LFSR, diperoleh relasi rekurensi sebagai berikut:

$$a_{k+3} = a_{k+1} + a_k$$

Dalam $GF(2)$, relasi tersebut dapat ditulis sebagai:

$$a_{k+3} = a_{k+1} \oplus a_k$$

Misalkan dipilih 100 sebagai *initial state*, maka pergeseran bitnya menjadi:

$$100 \rightarrow 001 \rightarrow 010 \rightarrow 101 \rightarrow 011 \rightarrow 111 \rightarrow 110 \rightarrow 100$$

Sehingga didapat suatu barisan biner 1001011.

Panjang periode barisan yang dibangkitkan oleh LFSR ditentukan oleh orde dari polinomial karakteristik yang digunakan. Dalam $GF(2)$, operasi penjumlahan ekuivalen dengan operasi XOR, sehingga relasi rekurensi linear pada LFSR dapat dinyatakan sebagai operasi XOR dari elemen-elemen sebelumnya. Barisan yang dibentuk dari LFSR juga dapat dipandang

sebagai kumpulan *state* berurutan dengan panjang tertentu. Misalkan diberikan suatu barisan biner $a_0a_1a_2 \dots$. Dengan memotongnya sepanjang n secara berurutan akan diperoleh *state-state*, yaitu: $a_0a_1 \dots a_{n-1}, a_1a_2 \dots a_n, \dots$ yang disebut sebagai *n-tuple*.

Contoh 2. Misalkan suatu LFSR dengan polinomial primitif $x^4 + x + 1$ atas $GF(2)$. Diambil seluruh *n-tuple* tak nol untuk dijadikan *initial state*, maka didapatkan barisan sebagai berikut:

Tabel 1. Tabel barisan hasil LFSR

<i>Initial State</i>	Hasil Barisan
1000	100010011010111
0001	000100110101111
0010	001001101011110
0100	010011010111100
1001	100110101111000
0011	001101011110001
0110	011010111100010
1101	110101111000100
1010	101011110001001
0101	010111100010011
1011	101111000100110
0111	011110001001101
1111	111100010011010
1110	111000100110101
1100	110001001101011

Berdasarkan contoh di atas dapat diketahui bahwa barisan yang dihasilkan sebenarnya merupakan pergeseran siklik satu sama lain sehingga dapat dipandang sebagai satu siklus yang sama. Secara umum, pada LFSR dengan polinomial primitif berderajat n , seluruh *state* tak nol akan menghasilkan satu siklus dengan periode maksimum $2^n - 1$. Barisan ini disebut sebagai *maximum-length sequence* atau *m-sequence*. Hal ini berbeda dengan LFSR yang menggunakan polinomial *irreducible* non-primitif di mana seluruh *state* yang dihasilkan akan terbagi ke dalam siklus-siklus terpisah.

Definisi 3. (Rosenfeld, 2002) Suatu barisan siklik adalah barisan $a_0a_1 \dots a_r$ yang dapat dibaca secara melingkar. Artinya, a_0 mengikuti a_r dan $a_1 \dots a_ra_0, \dots, a_ra_0 \dots a_{r-1}$ semuanya dianggap sebagai siklus yang sama dengan $a_0a_1 \dots a_r$. Diberikan bilangan asli $n \geq 1$ dan $k \geq 2$, suatu siklus dengan panjang k^n huruf disebut sebagai siklus lengkap $[1,2]$ atau barisan de

Bruijn jika setiap state $a_i a_{i+1} \dots a_{i+q-1}$ ($1 \leq i \leq k^n$) memuat semua kemungkinan k^n dengan urutan terurut $b_1 b_2 \dots b_q$ atas alphabet A dengan $|A| = k$.

Definisi 4. (Musthofa et al., 2022) Barisan de Bruijn termodifikasi biner adalah barisan biner tak hingga dan periodik yang diperoleh dengan menghapus satu nol dari rangkaian nol terpanjang dalam barisan de Bruijn biner. Barisan ini memiliki periode $2^n - 1$.

Contoh 3. Misalkan suatu barisan de Bruijn 00010111. Dengan menghapus satu nol pada rangkaian nol terpanjangnya didapat barisan baru, yaitu 0010111 yang memiliki periode $2^3 - 1 = 7$.

Konstruksi suatu barisan de Bruijn termodifikasi dapat dilakukan dengan memanfaatkan struktur barisan yang dihasilkan oleh LFSR, misalnya dengan menggunakan m -sequence yang memiliki periode maksimum $2^n - 1$. Untuk LFSR dengan polinomial *irreducible* non-primitif sendiri, siklus-siklus terpisah dapat digabung menjadi suatu barisan de Bruijn termodifikasi menggunakan metode penggabungan siklus.

Cara lain yang juga masih memanfaatkan barisan hasil LFSR, yaitu dengan melakukan t -desimasi pada suatu m -sequence. Pada penelitian ini t -desimasi dilakukan untuk mempartisi m -sequence menjadi sebanyak t barisan. Barisan tersebut lalu dimasukkan ke dalam Algoritma Berlekamp-Massey untuk mendapatkan polinomial minimalnya. Apabila didapat $f(x)$ *irreducible* non-primitif, maka siklus-siklus terpisah dapat dibangkitkan yang kemudian digabung menggunakan metode penggabungan siklus.

HASIL DAN PEMBAHASAN

Metode penggabungan siklus merupakan salah satu metode yang dapat digunakan untuk membangun barisan de Bruijn termodifikasi. Metode ini berakar dari teori *feedback shift register* yang selanjutnya dikembangkan melalui berbagai konstruksi barisan de Bruijn berbasis struktur siklus oleh Fredricksen (1982). Ide utamanya adalah dengan menggabung siklus-siklus terpisah yang dihasilkan dari LFSR dengan polinomial *irreducible* non-primitif. Siklus tersebut digabung untuk memperoleh satu siklus yang lebih panjang. Proses penggabungan dilakukan dengan mencari pasangan konjugat antar siklus kemudian menukar *successor* dari *state* yang dipilih sebagai pasangan konjugatnya.

Definisi 5. (Li & Lin, 2018) Misalkan $S = s_0 s_1 \dots s_{n-1}$ adalah suatu n -tuple biner derajat n . Jika $S = 00 \dots 0$, maka S disebut sebagai *state nol*. Untuk setiap $S \neq 00 \dots 0$, $\bar{S} = \bar{s}_0 \bar{s}_1 \dots \bar{s}_{n-1}$ didefinisikan sebagai konjugat dari S , yaitu *state* yang didapatkan dari hasil mengkomplemenkan bit pertama dari S . Dua siklus C_1 dan C_2 dikatakan *adjacent* jika terdapat pasangan konjugat $(S, \bar{S})$ sehingga S berada pada C_1 dan $\bar{S}$ berada pada C_2 . Kedua siklus tersebut dapat digabung menjadi satu siklus dengan menukar *successor* dari pasangan konjugat $(S, \bar{S})$.

Sebagai contoh, misalkan diberikan suatu LFSR dengan polinomial karakteristik $x^4 + x^3 + x^2 + x + 1$. Register bit menghasilkan siklus-siklus terpisah: $C_0 = 10001, C_1 = 01001, C_2 = 11101$. Dengan mencari pasangan konjugat dan menukar *successor* dari pasangan tersebut didapatkan siklus baru sepanjang $2^4 - 1 = 15$, yaitu 111000101001101.

Berdasarkan contoh di atas, proses penggabungan siklus dilakukan dengan menukar *successor* dari pasangan konjugat sehingga siklus-siklus yang awalnya terpisah dapat tergabung menjadi satu siklus periodik yang lebih panjang. Chang, Ezerman, Ling, et al. (2019) menjelaskan bahwa pada konstruksi barisan de Bruijn menggunakan metode penggabungan

siklus, seluruh siklus termasuk siklus yang memuat *state* nol ikut digabung sehingga diperoleh barisan de Bruijn dengan panjang 2^n .

Pada penelitian ini, penggabungan hanya dilakukan pada siklus-siklus yang memuat *state* tak nol sehingga *state* nol tidak muncul pada barisan akhir yang diperoleh. Dengan demikian, barisan yang dihasilkan merupakan barisan de Bruijn termodifikasi. Metode ini memungkinkan konstruksi barisan de Bruijn termodifikasi melalui struktur siklus dari LFSR tanpa perlu membangkitkan seluruh barisan secara langsung. Akan tetapi, pencarian pasangan konjugat secara manual menjadi kurang efisien untuk orde yang besar karena setiap *state* harus diperiksa terhadap siklus lainnya sehingga akan digunakan logaritma Zech sebagai alat bantu untuk menentukan pasangan konjugat secara lebih sistematis dan efisien.

Definisi 6. (Chang et al., 2021) Diberikan suatu medan hingga $GF(2^n)$ dengan elemen primitif a . Untuk setiap $i \in \{0, 1, \dots, 2^n - 2\}$, logaritma Zech relatif terhadap a didefinisikan sebagai bilangan bulat $\tau(i)$ yang memenuhi:

$$1 + a^i = a^{\tau(i)}$$

Nilai $\tau(i)$ disebut sebagai logaritma Zech dari i .

Berdasarkan definisi tersebut, diketahui bahwa logaritma Zech mengubah penjumlahan $1 + a^i$ menjadi bentuk eksponen $a^{\tau(i)}$. Artinya operasi penjumlahan pada medan hingga dapat didefinisikan kembali ke dalam bentuk eksponen. Hal ini tentunya sangat membantu ketika *state* LFSR direpresentasikan sebagai a^k . Pasangan-pasangan konjugat yang diperoleh dari logaritma Zech ini dapat didaftarkan dalam satu tabel.

Contoh 4. Misalkan suatu polinomial primitif $x^4 + x + 1$ dengan akar α . Karena $a \in GF(2^4)$ dan a adalah akar primitif dari $p(x) = x^4 + x + 1$ maka setiap elemen $GF(2^4)$ dapat direpresentasikan sebagai kelas polinomial berderajat kurang dari 4 berikut:

$$GF(2^4) = \{0, 1, \alpha, \alpha^2, 1 + \alpha, 1 + \alpha^2, \alpha + \alpha^2, 1 + \alpha + \alpha^2\}$$

Untuk mengubah masing-masing representasi menjadi bentuk a^i dapat dilakukan pengalihan dengan α . Ingat karena polinomial primitif yang dipakai adalah $x^4 + x + 1$ dan α adalah akarnya, maka berlaku: $\alpha^4 = \alpha + 1$. Dengan definisi dari logaritma Zech, yaitu $1 + a^i = a^{\tau(i)}$ dapat ditambahkan angka 1 pada masing-masing a^i . Sehingga didapatkan tabel logaritma Zech sebagai berikut:

Tabel 2. Tabel logaritma Zech

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$\tau(i)$	4	8	14	1	10	13	9	2	7	5	12	11	6	3

Logaritma Zech juga dapat digunakan untuk menentukan posisi pasangan konjugat. Telah dijelaskan sebelumnya bahwa t -desimasi membagi m -sequence sebanyak t siklus terpisah. Nilai t inilah yang akan digunakan pada definisi $a^t = \beta$ untuk menentukan posisi suatu pasangan konjugat.

Contoh 5. Misalkan 10001, 01111, 00101 adalah hasil 3-desimasi m -sequence yang terbentuk dari polinomial primitif $x^4 + x + 1$.

Definisikan $a^3 = \beta$ untuk mengubah setiap a^i ke dalam $a^k \cdot \beta^j$ dengan $k = 0, \dots, t-1$ adalah nomor kelas dan $j = 0, \dots, \frac{2^n-1}{t}$ adalah posisi pada kelas. Setiap kelas nantinya akan berkorespondensi dengan siklus-siklus yang dibentuk dari $f(x)$, sehingga dapat dikatakan bahwa k adalah nomor siklus dan j adalah posisi siklus dari suatu *state*. Sehingga didapat:

Tabel 3. Tabel nomor siklus dan posisi suatu *state*

$a^0 = a^0 \cdot \beta^0$	$a^5 = a^2 \cdot \beta^1$	$a^{10} = a^1 \cdot \beta^3$
$a^1 = a^1 \cdot \beta^0$	$a^6 = a^0 \cdot \beta^2$	$a^{11} = a^2 \cdot \beta^3$
$a^2 = a^2 \cdot \beta^0$	$a^7 = a^1 \cdot \beta^2$	$a^{12} = a^0 \cdot \beta^4$
$a^3 = a^0 \cdot \beta^1$	$a^8 = a^2 \cdot \beta^2$	$a^{13} = a^1 \cdot \beta^4$
$a^4 = a^1 \cdot \beta^1$	$a^9 = a^0 \cdot \beta^3$	$a^{14} = a^2 \cdot \beta^4$

Dengan begitu setiap a^i telah dibagi ke dalam 3 kelas eksponen sesuai dengan 3-desimasi.

$$E_0 = \{a^0, a^3, a^6, a^9, a^{12}\}$$

$$E_1 = \{a^1, a^4, a^7, a^{10}, a^{13}\}$$

$$E_2 = \{a^2, a^5, a^8, a^{11}, a^{14}\}$$

Perhatikan barisan 10001, 01111, dan 00101. Karena setiap bit ini adalah representasi dari suatu n -tuple, maka dapat ditulis sebagai:

$$C_0 = 1000 \rightarrow 0001 \rightarrow 0011 \rightarrow 0110 \rightarrow 1100$$

$$C_1 = 0111 \rightarrow 1111 \rightarrow 1110 \rightarrow 1101 \rightarrow 1011$$

$$C_2 = 0010 \rightarrow 0101 \rightarrow 1010 \rightarrow 0100 \rightarrow 1001$$

Dipilih C_0 sebagai siklus pertama. Dikarenakan setiap n -tuple pada m -sequence direpresentasikan ke dalam a^i , maka siklus C_0 tersebut berkorespondensi dengan:

$$E_0 = \{a^0, a^3, a^6, a^9, a^{12}\}$$

Misalkan dicari pasangan konjugat dari 0001, karena 0001 berkorespondensi dengan a^3 maka konjugat dari 0001 juga harus berkorespondensi dengan konjugat dari a^3 . Berdasarkan Contoh 4, a^3 memiliki pasangan konjugat a^1 yang berada pada siklus ke-1 posisi ke-0 ($a^1 = a^1 \cdot \beta^0$). Sehingga konjugat dari 0001, yaitu 1001 harus berada pada posisi $a^1 = a^1 \cdot \beta^0$.

Dengan mengetahui posisi dari suatu konjugat, metode penggabungan siklus menjadi lebih efisien terutama untuk n besar. Selanjutnya akan dijelaskan mengenai langkah-langkah penerapan logaritma Zech pada metode penggabungan siklus dari awal hingga didapat suatu barisan de Bruijn termodifikasi. Untuk $n \geq 10$ diperlukan bantuan pemrograman dengan kode seperti berikut:

Algoritma 1. Konstruksi Barisan de Bruijn Termodifikasi Menggunakan Logaritma Zech dan Metode Penggabungan Siklus	
<i>Input:</i>	
n → derajat	
$p(x)$ → polinomial primitif $p(x)$ derajat n	
t → parameter t-desimasi (membagi habis $2^n - 1$)	
<i>Output:</i>	
s → Barisan de Bruijn termodifikasi	
1:	Bangkitkan m-sequence S menggunakan LFSR dengan polinomial primitif $p(x)$
2:	Lakukan t -desimasi terhadap S untuk memperoleh barisan S_t
3:	Ambil $2n$ bit pertama dari S_t , kemudian terapkan algoritma Berlekamp-Massey untuk memperoleh polinomial minimal $f(x)$
4:	Gunakan $f(x)$ sebagai polinomial karakteristik LFSR untuk membangkitkan beberapa siklus terpisah: $C = \{C_0, C_0, \dots, C_{t-1}\}$
5:	Bangun tabel logaritma Zech terhadap elemen primitif α dari $GF(2^n)$
6:	Definisikan: $\beta = \alpha^t$
7:	Partisi $\alpha^i \in GF(2^n)$ ke dalam kelas-kelas eksponen: $E_k = \{\alpha^k \beta^j \mid j = 0, \dots, (2^n - 1)/t - 1\}$ Untuk $k = 0, \dots, t - 1$
6:	Definisikan pemetaan: $\phi = \alpha^i \rightarrow (k, j)$ dengan: k = nomor kelas/siklus j = posisi α^i dalam kelas tersebut
7:	Pilih satu siklus sebagai siklus utama
8:	Selama masih terdapat siklus lain yang belum digabungkan, lakukan: Pilih suatu <i>state</i> x pada siklus utama Representasikan x sebagai α^i

	Gunakan tabel Zech untuk menentukan pasangan konjugat dari x Tentukan siklus yang memuat x' menggunakan pemetaan φ Gabungkan kedua utama dengan dengan menukar <i>successor</i> dari x dan x' Perbarui siklus utama sebagai hasil penggabungan Akhiri perulangan
9:	Setelah semua siklus tergabung menjadi satu siklus panjang, ambil bit pertama dari setiap <i>state</i> secara berurutan
10:	Hasilkan barisan s sebagai barisan de Bruijn termodifikasi
11:	Hasil akhir berupa s

Kode di atas dapat langsung dijalankan hingga didapat suatu barisan de Bruijn termodifikasi. Untuk $n < 10$, perhitungan secara manual masih dapat dilakukan seperti contoh berikut.

Contoh 6. Akan diberikan contoh penggunaan Algoritma 1 pada polinomial derajat 6. Misalkan dipilih polinomial primitif $p(x) = x^6 + x + 1$ dengan elemen primitif α . *m-sequence* yang dihasilkan dari $p(x)$ dilakukan 3-desimasi. Dengan algoritma Berlekamp-Massey didapat $f(x)$ yang sesuai, yaitu $x^6 + x^4 + x^2 + x + 1$. Bentuk siklus-siklus terpisah menggunakan $f(x)$.

$$\begin{aligned}
C_0 &= 100000 \rightarrow 000001 \rightarrow 000010 \rightarrow 000101 \rightarrow 001010 \rightarrow 010100 \\
&\rightarrow 101001 \rightarrow 010010 \rightarrow 100100 \rightarrow 001001 \rightarrow 010011 \rightarrow 100110 \\
&\rightarrow 001100 \rightarrow 011001 \rightarrow 110010 \rightarrow 100101 \rightarrow 001011 \rightarrow 010110 \\
&\rightarrow 101100 \rightarrow 011000 \rightarrow 110000 \\
C_1 &= 111001 \rightarrow 110011 \rightarrow 100111 \rightarrow 001110 \rightarrow 011100 \rightarrow 111000 \\
&\rightarrow 110001 \rightarrow 100010 \rightarrow 000100 \rightarrow 001000 \rightarrow 010001 \rightarrow 100011 \\
&\rightarrow 000110 \rightarrow 001101 \rightarrow 011011 \rightarrow 110111 \rightarrow 101111 \rightarrow 011111 \\
&\rightarrow 111111 \rightarrow 111110 \rightarrow 111100 \\
C_2 &= 110100 \rightarrow 101000 \rightarrow 010000 \rightarrow 100001 \rightarrow 000011 \rightarrow 000111 \\
&\rightarrow 001111 \rightarrow 011110 \rightarrow 111101 \rightarrow 111011 \rightarrow 110110 \rightarrow 101101 \\
&\rightarrow 011010 \rightarrow 110101 \rightarrow 101010 \rightarrow 010101 \rightarrow 101011 \rightarrow 010111 \\
&\rightarrow 101110 \rightarrow 011101 \rightarrow 111010
\end{aligned}$$

Bangun tabel logaritma Zech dari $p(x)$.

Tabel 4. Tabel logaritma Zech

i	$\tau(i)$	i	$\tau(i)$
1	6	19	56
2	12	20	59
3	32	21	42
4	24	22	50
5	62	28	41
7	26	29	60
8	48	30	46
9	45	31	34
10	61	36	54
11	25	37	44
13	35	38	49
14	52	39	43
15	23	40	55
16	33	51	53
17	47	57	58
18	27		

Definisikan $\alpha^3 = \beta$. Untuk setiap α^i ubah dalam bentuk $\alpha^k \cdot \beta^j$ dengan $k = 0, \dots, t-1$ dan $j = 0, \dots, \frac{2^n-1}{t} - 1$ untuk menentukan lokasi suatu a^i yang nantinya akan digunakan untuk menentukan posisi suatu *state*.

Tabel 5. Tabel nomor siklus dan posisi suatu *state*

$a^0 = a^0 \cdot \beta^0$	$a^{21} = a^0 \cdot \beta^7$	$a^{42} = a^0 \cdot \beta^{14}$
$a^1 = a^1 \cdot \beta^0$	$a^{22} = a^1 \cdot \beta^7$	$a^{43} = a^1 \cdot \beta^{14}$
$a^2 = a^2 \cdot \beta^0$	$a^{23} = a^2 \cdot \beta^7$	$a^{44} = a^2 \cdot \beta^{14}$
$a^3 = a^0 \cdot \beta^1$	$a^{24} = a^0 \cdot \beta^8$	$a^{45} = a^0 \cdot \beta^{15}$
$a^4 = a^1 \cdot \beta^1$	$a^{25} = a^1 \cdot \beta^8$	$a^{46} = a^1 \cdot \beta^{15}$
$a^5 = a^2 \cdot \beta^1$	$a^{26} = a^2 \cdot \beta^8$	$a^{47} = a^2 \cdot \beta^{15}$

$a^6 = a^0 \cdot \beta^2$	$a^{27} = a^0 \cdot \beta^9$	$a^{48} = a^0 \cdot \beta^{16}$
$a^7 = a^1 \cdot \beta^2$	$a^{28} = a^1 \cdot \beta^9$	$a^{49} = a^1 \cdot \beta^{16}$
$a^8 = a^2 \cdot \beta^2$	$a^{29} = a^2 \cdot \beta^9$	$a^{50} = a^2 \cdot \beta^{16}$
$a^9 = a^0 \cdot \beta^3$	$a^{30} = a^0 \cdot \beta^{10}$	$a^{51} = a^0 \cdot \beta^{17}$
$a^{10} = a^1 \cdot \beta^3$	$a^{31} = a^1 \cdot \beta^{10}$	$a^{52} = a^1 \cdot \beta^{17}$
$a^{11} = a^2 \cdot \beta^3$	$a^{32} = a^2 \cdot \beta^{10}$	$a^{53} = a^2 \cdot \beta^{17}$
$a^{12} = a^0 \cdot \beta^4$	$a^{33} = a^0 \cdot \beta^{11}$	$a^{54} = a^0 \cdot \beta^{18}$
$a^{13} = a^1 \cdot \beta^4$	$a^{34} = a^1 \cdot \beta^{11}$	$a^{55} = a^1 \cdot \beta^{18}$
$a^{14} = a^2 \cdot \beta^4$	$a^{35} = a^2 \cdot \beta^{11}$	$a^{56} = a^2 \cdot \beta^{18}$
$a^{15} = a^0 \cdot \beta^5$	$a^{36} = a^0 \cdot \beta^{12}$	$a^{57} = a^0 \cdot \beta^{19}$
$a^{16} = a^1 \cdot \beta^5$	$a^{37} = a^1 \cdot \beta^{12}$	$a^{58} = a^1 \cdot \beta^{19}$
$a^{17} = a^2 \cdot \beta^5$	$a^{38} = a^2 \cdot \beta^{12}$	$a^{59} = a^2 \cdot \beta^{19}$
$a^{18} = a^0 \cdot \beta^6$	$a^{39} = a^0 \cdot \beta^{13}$	$a^{60} = a^0 \cdot \beta^{20}$
$a^{19} = a^1 \cdot \beta^6$	$a^{40} = a^1 \cdot \beta^{13}$	$a^{61} = a^1 \cdot \beta^{20}$
$a^{20} = a^2 \cdot \beta^6$	$a^{41} = a^2 \cdot \beta^{13}$	$a^{62} = a^2 \cdot \beta^{20}$

Dengan begitu setiap a^i telah dibagi ke dalam 3 kelas eksponen sesuai dengan 3-desimasi.

$$E_0 = \{a^{3i} \mid 0 \leq i \leq 20\} = \{a^0, a^3, a^6, \dots, a^{60}\}$$

$$E_1 = \{a^{3i+1} \mid 0 \leq i \leq 20\} = \{a^1, a^4, a^7, \dots, a^{61}\}$$

$$E_2 = \{a^{3i+2} \mid 0 \leq i \leq 20\} = \{a^2, a^5, a^8, \dots, a^{62}\}$$

Pilih siklus pertama pada $f(x)$ yang memiliki *state* pertama sama dengan pada *m-sequence*.

$$\begin{aligned} C_0 &= 100000 \rightarrow 000001 \rightarrow 000010 \rightarrow 000101 \rightarrow 001010 \\ &\rightarrow 010100 \rightarrow 101001 \rightarrow 010010 \rightarrow 100100 \rightarrow 001001 \\ &\rightarrow 010011 \rightarrow 100110 \rightarrow 001100 \rightarrow 011001 \rightarrow 110010 \\ &\rightarrow 100101 \rightarrow 001011 \rightarrow 010110 \rightarrow 101100 \rightarrow 011000 \rightarrow 110000 \end{aligned}$$

$$E_0 = \{a^{3i} \mid 0 \leq i \leq 20\} = \{a^0, a^3, a^6, \dots, a^{60}\}$$

Misalkan dipilih 000010 yang bersesuaian dengan a^6 . Maka dapat dilihat dari tabel logaritma Zech, karena a^6 memiliki konjugat a^1 maka konjugat dari 000010, yaitu 100010 harus bersesuaian dengan a^1 . Sehingga didapat urutan siklus kedua $f(x)$ sebagai berikut:

$$\begin{aligned}
C_1 &= 100010 \rightarrow 000100 \rightarrow 001000 \rightarrow 010001 \rightarrow 100011 \\
&\rightarrow 000110 \rightarrow 001101 \rightarrow 011011 \rightarrow 110111 \rightarrow 101111 \\
&\rightarrow 011111 \rightarrow 111111 \rightarrow 111110 \rightarrow 11110 \rightarrow 0111001 \\
&\rightarrow 110011 \rightarrow 100111 \rightarrow 001110 \rightarrow 011100 \rightarrow 111000 \\
&\rightarrow 110001
\end{aligned}$$

yang bersesuaian dengan:

$$E_1 = \{a^{3i+1} \mid 0 \leq i \leq 20\} = \{a^1, a^4, a^7, \dots, a^{61}\}$$

Gabung siklus pertama dan kedua menggunakan pasangan konjugat yang sudah dipilih. Lakukan langkah yang sama sampai semua siklus tergabung menjadi satu siklus dengan panjang $2^n - 1$. Ambil bit pertama pada masing-masing *state* untuk mendapatkan barisan de Bruijn termodifikasi dengan panjang $2^n - 1 = 63$:

$$S = 10000010001101111100111000101011101000011110110101001001100101$$

Berdasarkan contoh di atas, polinomial primitif dipilih pertama kali untuk membangun *m-sequence* yang selanjutnya dipartisi menggunakan *t*-desimasi. Pemilihan *n* sebagai derajat polinomial primitif perlu diperhatikan. Jika *m-sequence* yang dihasilkan memiliki panjang $2^n - 1 = p$ dengan *p* adalah bilangan prima, maka *t* yang bisa dipilih hanya $\{1, 2^n - 1\}$. Pemilihan *t* = 1 akan menghasilkan *m-sequence* itu sendiri. Sementara pemilihan *t* = $2^n - 1$ sama saja membagi barisan menjadi sebanyak $2^n - 1$ barisan dengan masing-masing hanya terdapat satu *state*. Kondisi tersebut menyebabkan algoritma Berlekamp-Massey menghasilkan polinomial minimal berderajat sangat rendah sehingga tidak memunculkan siklus-siklus terpisah yang diperlukan dalam proses penggabungan siklus. Sehingga diperlukan nilai *n* yang dapat menghasilkan *m-sequence* panjang $2^n - 1 = p$ dengan *p* merupakan bilangan komposit.

Untuk *n* yang menghasilkan *m-sequence* panjang $2^n - 1 = p$ dengan *p* merupakan bilangan komposit, terdapat nilai *t* lain yang tidak dapat digunakan selain $t = \{1, 2^n - 1\}$. Perhatikan contoh berikut.

Contoh 7. Misalkan 100010011010111 yang dihasilkan dari polinomial primitif $p(x) = x^4 + x + 1$. Dipilih nilai *t* = 5, maka didapat barisan baru:

$$101,000,011,011, \& 101$$

Dengan Berlekamp-Massey maka didapat polinomial minimalnya yaitu $x^2 + x + 1$.

Berdasarkan contoh di atas, polinomial minimal barisan hasil desimasi memiliki derajat yang lebih rendah dibandingkan dengan polinomial primitif awal. Polinomial $x^2 + x + 1$ merupakan polinomial primitif derajat 2 sehingga tidak akan membentuk siklus-siklus terpisah pada LFSR melainkan suatu *m-sequence*. Oleh karena itu, pemilihan nilai *t* menjadi penting agar dapat diperoleh suatu barisan yang memiliki representasi polinomial *irreducible* non-primitif sehingga dapat menghasilkan siklus-siklus terpisah LFSR.

Setelah ditemukan nilai *n* dan *t* yang sesuai, siklus-siklus tersebut akan digabung menggunakan metode penggabungan siklus dengan bantuan logaritma Zech. Setiap *state* pada *m-sequence* direpresentasikan dahulu ke dalam bentuk a^i dan dipartisi sehingga sesuai dengan

siklus yang dihasilkan oleh polinomial $f(x)$. Penggunaan logaritma Zech mempermudah dalam proses penggabungan siklus karena setiap pasangan konjugat sudah terdaftar terlebih dahulu ke dalam satu tabel yang juga mencakup posisi dan nomor siklus. Dengan hanya melihat tabel logaritma Zech, pencarian pasangan dan posisi konjugat tidak perlu dilakukan untuk seluruh *state*. Untuk n yang sangat besar dengan siklus yang sangat banyak, pendekatan ini dapat secara signifikan mempersingkat waktu yang dibutuhkan dalam proses penggabungan siklus.

Setelah pasangan konjugat diperoleh, proses penggabungan siklus dilakukan dengan menukar successor pada pasangan konjugat sehingga seluruh siklus tak nol bergabung menjadi satu siklus yang lebih panjang. Dikarenakan seluruh n -tuple tak nol tetap muncul tepat satu kali dalam satu periode barisan, hasil akhir dari proses ini merupakan suatu barisan de Bruijn termodifikasi dengan panjang periode $2^n - 1$.

SIMPULAN

Berdasarkan hasil dan pembahasan, maka dapat ditarik simpulan bahwa metode penggabungan siklus dapat digunakan untuk memperoleh barisan de Bruijn termodifikasi dengan cara menggabungkan siklus-siklus terpisah hasil pembangkitan LFSR dengan polinomial *irreducible* non-primitif. Proses penggabungan dilakukan dengan menukar *successor* pada pasangan konjugat. Logaritma Zech dapat dimanfaatkan untuk mempermudah dalam menentukan pasangan konjugat antar siklus. Dengan merepresentasikan setiap *state* dalam bentuk a^i , hubungan antar *state* dapat dicatat dalam tabel logaritma Zech sehingga pencarian pasangan konjugat tidak perlu dilakukan secara langsung terhadap seluruh *state*. Pendekatan ini menjadi lebih efisien terutama untuk orde n yang lebih besar dengan jumlah siklus lebih banyak. Konstruksi barisan de Bruijn termodifikasi dilakukan dengan membangun *m-sequence* menggunakan polinomial primitif, kemudian melakukan *t*-desimasi untuk memperoleh beberapa barisan periodik. Barisan-barisan tersebut dianalisis menggunakan algoritma Berlekamp–Massey untuk memperoleh polinomial minimal yang membentuk siklus-siklus terpisah. Selanjutnya, siklus-siklus tersebut digabung menggunakan metode penggabungan siklus dengan memanfaatkan logaritma Zech hingga diperoleh satu siklus panjang yang memuat seluruh n -tuple tak nol tepat satu kali.

UCAPAN TERIMA KASIH

Terimakasih kepada seluruh Dosen Program Studi Matematika yang telah memberikan ilmu hingga terselesaikannya artikel ini.

DAFTAR PUSTAKA

- Chan, A. H., Games, R. A., & Key, E. L. (1982). On the complexities of de Bruijn sequences. *Journal Of Combinatorial Theory, Series A*, 33, 233–246. [https://doi.org/10.1016/0097-3165\(82\)90038-3](https://doi.org/10.1016/0097-3165(82)90038-3)
- Chang, Z., Ezerman, M. F., Fahreza, A. A., Ling, S., Szmidt, J., & Wang, H. (2021). Binary de Bruijn sequences via Zech's logarithms. *SN Computer Science*, 2(4), 314. <https://doi.org/10.1007/s42979-021-00683-9>
- Chang, Z., Ezerman, M. F., Ling, S., & Wang, H. (2019). On binary de Bruijn sequences from LFSRs with arbitrary characteristic polynomials. *Designs, Codes and Cryptography*, 87(5), 1137–1160. <https://doi.org/10.1007/s10623-018-0509-y>

- de Bruijn, N. G. (1946). A combinatorial problem. *Proceedings of the Section of Sciences of the Koninklijke Nederlandse Akademie van Wetenschappen Te Amsterdam*, 49(7), 758–764.
- Fredricksen, H. (1982). A survey of full length nonlinear shift register cycle algorithms. *SIAM Review*, 24(2), 195–221. <https://doi.org/10.1137/1024041>
- Golomb, S. W., & Gong, G. (2005). *Signal Design for Good Correlation for Wireless Communication, Cryptography, and Radar*. Cambridge University Press. <https://doi.org/10.1017/CBO9780511546907>
- Li, M., & Lin, D. (2018). De Bruijn sequences, adjacency graphs, and cyclotomy. *IEEE Transactions on Information Theory*, 64(4), 2941–2952. <https://doi.org/10.1109/TIT.2017.2787742>
- Mahanum. (2021). Tinjauan kepustakaan. *ALACRITY: Journal Of Education*. <https://doi.org/10.52121/alacrity.v1i2.20>
- Musthofa, Wijayanti, I. E., Palupi, D. J. E., & Ezerman, M. F. (2022). A new approach to determine the minimal polynomials of binary modified de Bruijn sequences. *Mathematics*, 10(15). <https://doi.org/10.3390/math10152577>
- Rosenfeld, V. R. (2002). Enumerating de Bruijn sequences. *MATCH Communications in Mathematical and in Computer Chemistry*, 71–83.
- Zhu, Y., Chang, Z., Ezerman, M. F., & Wang, Q. (2021). An efficiently generated family of binary de Bruijn sequences. *Discrete Mathematics*, 344(6), 112368. <https://doi.org/10.1016/j.disc.2021.112368>