

PENDEKATAN BARISAN KOMBINATORIAL UNTUK PEMECAHAN MASALAH CAR SEQUENCING

Ahmad Sabri

Universitas Gunadarma
Jl. Margonda Raya No. 100, Depok, Jawa Barat, 16424
sabri@staff.gunadarma.ac.id

ABSTRAK

Masalah pengurutan mobil (*car sequencing problem / CSP*) adalah tentang bagaimana menentukan urutan mobil pada jalur perakitan, di mana pada mobil-mobil ini akan dipasang opsi-opsi tertentu pada stasiun kerja yang sesuai sepanjang jalur perakitan. Setiap stasiun kerja hanya diperuntukkan untuk pemasangan satu opsi, dan memiliki keterbatasan dalam kapasitas pemasangan. Karena keterbatasan stasiun kerja dan kebutuhan untuk memenuhi demand, pengurutan mobil yang tidak tepat akan mengakibatkan gagalnya pemasangan opsi-opsi yang telah ditetapkan, dan tidak terpenuhinya demand. Penelitian terdahulu memberikan berbagai metode untuk menemukan solusi masalah ini, antara lain metode SAT, heuristik, metaheuristik, teori permainan, dan algoritma evolusioner adaptif. Namun belum terdapat penelitian yang melakukan pendekatan dari sisi kombinatorika unta. Penelitian ini mengajukan pendekatan kombinatorial dengan mengkodekan masalah CSP dalam bentuk barisan integer, di mana setiap integer berkorespondensi satu-satu dengan vektor yang mendefinisikan opsi-opsi pada mobil pada jalur perakitan. Pengkodean ini menghasilkan barisan integer terestriksi, di mana restriksi mengacu pada keterbatasan kapasitas stasiun kerja. Hasil dari penerapan metode ini adalah ruang solusi untuk semua kemungkinan urutan mobil berdasarkan demand yang berbeda-beda. Pendekatan ini membangun suatu keterhubungan antara CSP dengan kombinatorika unta, yang diharapkan dapat menjadi topik penelitian baru yang dapat dikembangkan lebih luas.

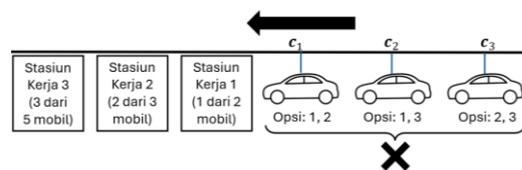
Kata Kunci: *car sequencing problem, kombinatorika unta, barisan kombinatorial*

PENDAHULUAN

Masalah pengurutan mobil (*car sequencing problem / CSP*) dijelaskan sebagai berikut: Sejumlah mobil akan diproduksi dengan sistem jalur perakitan linier. Spesifikasi setiap mobil disesuaikan dengan serangkaian opsi/kelengkapan tertentu seperti kaca gelap, perangkat GPS, lis samping, dsb. Jalur perakitan bergerak secara linier dan simultan, dengan rute melalui beberapa stasiun kerja untuk pemasangan opsi-opsi ini. Satu stasiun kerja hanya diperuntukkan untuk pemasangan satu opsi. Mobil-mobil dalam rangkaian tidak dapat diposisikan secara acak karena setiap stasiun kerja memiliki kapasitas terbatas dan memerlukan waktu untuk menyiapkan opsi-opsi tersebut saat jalur perakitan bergerak. Kendala kapasitas ini dinyatakan sebagai $\frac{p_i}{q_i}$, yang menunjukkan bahwa stasiun kerja i dapat memasang opsi

i pada sebanyak p_i mobil dari setiap q_i mobil yang masuk.

Gambar 1 menunjukkan urutan mobil yang tidak tepat dalam jalur perakitan. Pada perpindahan pertama, mobil c_1, c_2, c_3 berpindah maju satu posisi secara simultan. Pada tahap ini, mobil c_1 masuk ke stasiun kerja 1.

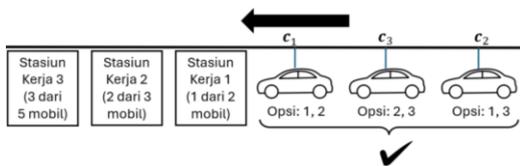


Gambar 1. Urutan mobil yang tidak tepat ($c_1c_2c_3$) berdasarkan opsi dan restriksi pada stasiun kerja.

Stasiun kerja 1 kemudian beroperasi untuk memasang opsi 1 pada mobil tersebut. Pada perpindahan kedua, mobil c_1 masuk ke stasiun kerja 2, dan mobil c_2 masuk ke

stasiun kerja 1. Oleh karena mobil c_2 juga mendapatkan opsi 1, maka seharusnya stasiun kerja 1 memasang opsi 1 pada mobil c_2 . Namun hal ini tidak dimungkinkan karena stasiun kerja 1 sebelumnya sudah melayani mobil c_1 , dan kapasitas stasiun kerja ini hanya 1 mobil dari setiap 2 mobil yang masuk. Dalam hal ini, opsi 1 tidak dapat dipasang pada mobil kedua. Susunan mobil demikian menyebabkan jalur produksi memberikan output tidak sebagaimana yang diharapkan.

Salah satu susunan mobil yang tepat ditunjukkan pada Gambar 2. Pada perpindahan pertama, mobil c_1 masuk ke stasiun kerja 1, untuk kemudian dilakukan pemasangan opsi 1. Pada perpindahan kedua, mobil c_1 masuk ke stasiun kerja 2 dan mobil c_3 masuk ke stasiun kerja 1, dan stasiun ini *idle* karena mobil c_3 tidak memerlukan opsi 1. Pergerakan ini tidak melanggar keterbatasan kapasitas stasiun kerja 1. Dapat diperiksa bahwa pergerakan-pergerakan selanjutnya berlangsung tanpa melanggar batasan stasiun kerja, sampai jalur produksi menuntaskan mobil terakhir (c_3).



Gambar 2. Urutan mobil yang tepat ($c_1c_3c_2$) berdasarkan opsi dan restriksi pada stasiun kerja.

Beberapa pembahasan tentang metode untuk menemukan solusi CSP antara lain model SAT dan hibrid [1], pemrograman heuristik [2], metaheuristik konstruktif [3], dan heuristik dengan *block batches* [4]. Solusi CSP dengan pendekatan teori permainan diberikan oleh [5], dan dengan pendekatan algoritma evolusioner diferensial adaptif diberikan oleh [6].

Sejauh penelusuran yang dilakukan, belum terdapat penelitian yang melakukan pendekatan dari sisi kombinatorika untai, di mana masalah CSP ini dapat dipandang sebagai masalah pembentukan barisan

kombinatorial terestriksi. Pendekatan kombinatorial yang diusulkan dalam penelitian ini adalah dengan mengkodekan semua kemungkinan urutan mobil dengan barisan integer, di mana sebuah integer mewakili sebuah tipe mobil. Setiap integer berkorespondensi satu-satu dengan sebuah vektor yang mendefinisikan tipe mobil yang diwakili integer tersebut. Pencarian ruang solusi dilakukan dengan membentuk *generating tree* untuk barisan paralel, yang belum pernah dibahas dalam penelitian sebelumnya.

Tujuan dari penelitian ini adalah membangun keterhubungan antara masalah CSP dengan kajian kombinatorika pada untai (*combinatorics on words*). Keterhubungan ini memungkinkan penggunaan algoritma kombinatorika untai untuk menemukan solusi masalah CSP. Beberapa penelitian terkait barisan kombinatorial terestriksi yang dapat digunakan untuk mengembangkan metode ini terdapat pada penelitian [7-10].

Pendekatan kombinatorial untuk masalah CSP yang digagas pada penelitian ini membuka sebuah topik baru yang potensial untuk dikembangkan dalam kajian kombinatorika pada untai, yaitu konsep barisan paralel beserta algoritma pembangkitnya.

METODE PENELITIAN

Untuk mendeskripsikan CSP diperlukan data sebagaimana diberikan oleh Tabel 1.

Tabel 1. CSP

Opsi	Tipe Mobil				Kapasitas stasiun kerja
	1	2	...	k	
Opsi 1	c_{11}	c_{21}	...	c_{k1}	p_1/q_1
Opsi 2	c_{12}	c_{22}	...	c_{k2}	p_2/q_2
⋮	⋮	⋮	⋮	⋮	
Opsi m	c_{1m}	c_{2m}	...	c_{km}	p_m/q_m
Demand Mobil	d_1	d_2	...	d_k	n

Terdapat k tipe mobil dengan m opsi yang tersedia. Setiap jenis mobil

didefinisikan oleh sebuah vektor opsi. Untuk mobil tipe i , vektor opsinya diberikan oleh:

$$c_i = \begin{bmatrix} c_{i1} \\ c_{i2} \\ \vdots \\ c_{im} \end{bmatrix}$$

di mana:

$$c_{ij} = \begin{cases} 1, & \text{jika tipe mobil } i \text{ mendapat opsi } j, \\ 0, & \text{jika tidak demikian.} \end{cases}$$

Melalui pendekatan barisan kombinatorial, sebuah solusi untuk CSP dengan n mobil, k tipe mobil, dan m opsi (stasiun kerja) diperoleh dengan membangun m barisan biner terestriksi dengan panjang n secara simultan, di mana:

- barisan ke i berkorespondensi dengan stasiun kerja ke i dari m stasiun kerja yang tersedia,
- barisan ke i memiliki restriksi yang didefinisikan berdasarkan kapasitas stasiun kerja.

Ruang solusi merupakan himpunan semua solusi yang mungkin. Untuk mendapatkan ruang solusi, didefinisikan berdasarkan entitas berikut:

- n : total demand mobil.
- d_i : demand mobil tipe i untuk $i = 1, 2, \dots, k$, di mana $\sum_{i=1}^k d_i = n$.
- $C = \{c_1, c_2, \dots, c_k\}$, di mana c_i adalah vektor biner yang mengkodekan opsi dari mobil tipe i .
- $c \in C^n$.
- $c(i)$: barisan entri pada baris ke i dari c .

Misalkan $C = \{c_1, c_2, c_3\}$ di mana

$$c_1 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, c_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, c_3 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}.$$

Diberikan sebuah $c \in C^4$. Jika $c = c_2 c_1 c_3 c_1$, yang dalam bentuk barisan vektor dinyatakan sebagai.

$$c = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix},$$

Maka :

$$c(1) = 0101, c(2) = 0010, c(3) = 1101.$$

- p_i/q_i adalah kapasitas stasiun kerja i , untuk $i = 1, 2, \dots, m$. Notasi ini bermakna stasiun kerja i dapat mengerjakan sebanyak p_i mobil untuk setiap q_i mobil yang masuk.
- Barisan CarSeq (*car sequence*) $s = s_1 s_2 \dots s_n$ merupakan barisan dari subscript pada c . Misalkan $c = c_2 c_1 c_3 c_1$, maka $s = 2131$.
- Barisan $d_1 d_2 \dots d_k$ merupakan vektor Parikh dari s (vektor Parikh dari $s = s_1 s_2 \dots s_n \in \{0, 1, \dots, k\}^n$ adalah barisan $d_1 d_2 \dots d_k$ di mana p_i adalah banyaknya kemunculan simbol i pada s .)

Ruang solusi untuk CSP diberikan oleh: subset dari C^n , di mana $c(i)$ adalah barisan biner yang menghindari faktor dengan panjang maksimum q_i dan memiliki p_i simbol 1 yang berurutan, di mana vektor Parikh dari s adalah $d_1 d_2 \dots d_k$.

Faktor panjang r dari sebuah barisan $s = s_1 s_2 \dots s_n$ adalah sebuah sub barisan panjang yang terdiri dari r entri berurutan dari s . Sebagai contoh: barisan 42645531 memiliki faktor-faktor antara lain 264, 4553, 42645, 64553, dsb.

HASIL DAN PEMBAHASAN

Penerapan pendekatan kombinatorial untuk masalah CSP diberikan dalam ilustrasi berikut, yang diambil dari contoh 2.1 dalam [2]:

Tabel 2. Target produksi

Ops	Tipe Mobil			Kapasitas stasiun kerja
	1	2	3	
Ops 1	1	0	0	1/2
Ops 2	0	1	0	2/3
Ops 3	1	1	0	3/5
Ops 4	0	0	1	3/6
Demand Mobil	d_1	d_2	d_3	$n = 5$

Berdasarkan Tabel 2 dan metodologi yang diusulkan, didefinisikan entitas berikut ini:

- $n = d_1 + d_2 + d_3 = 2 + 1 + 2 = 5$.
- $d_1 = 2, d_2 = 1, d_3 = 2$ (banyak permintaan mobil tipe 1, 2, 3).
- $C = \{c_1, c_2, c_3\}$, di mana c_i adalah vektor opsi untuk mobil tipe i . Berdasarkan kolom 1, 2, 3 pada Tabel 2, vektor-vektor tersebut adalah:

$$c_1 = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}, c_2 = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}, c_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

- Kapasitas stasiun kerja 1, 2, 3, dan 4 masing-masing diberikan oleh: $p_1/q_1 = 1/2$, $p_2/q_2 = 2/3$, $p_3/q_3 = 3/5$, $p_4/q_4 = 3/6$.

Ruang solusi untuk CSP di atas diberikan oleh dua kriteria berikut:

1. Semua barisan vektor $c \in C^5$, di mana $c(i)$ adalah barisan biner yang menghindari faktor dengan panjang maksimum q_i dan memiliki p_i simbol 1 yang berurutan. Lebih spesifik, restriksi untuk setiap barisan $c(i)$ adalah:

- $c(1)$: menghindari faktor dengan panjang maksimum 2 dan memiliki 1 simbol 1 yang berurutan,
- $c(2)$: menghindari faktor dengan panjang maksimum 3 dan memiliki 2 simbol 1 yang berurutan,
- $c(3)$: menghindari faktor dengan panjang maksimum 5 dan memiliki 3 simbol 1 yang berurutan,
- $c(4)$: menghindari faktor dengan panjang maksimum 6 dan memiliki 3 simbol 1 yang berurutan.

2. Vektor Parikh dari c adalah $d_1 d_2 d_3$.

Tabel 3, 4, dan 5 memberikan skema ekspansi barisan untuk pembentukan ruang solusi masalah CSP ini. Tabel 3 menyajikan solusi yang diawali oleh c_1 , Tabel 4 menyajikan solusi yang diawali oleh c_2 , dan Tabel 5 menyajikan solusi yang diawali oleh c_3 .

Oleh karena total demand adalah 5,

maka ekspansi barisan dilakukan sampai dengan level 5, yang direpresentasikan sebagai kolom L1, L2, ..., L5. Sel yang dicoret menandakan ekspansi barisan yang melanggar restriksi. Sebagai contoh pada Tabel 3, barisan $c_1 c_1$ (L1= c_1 , L2= c_2) merupakan ekspansi yang melanggar restriksi, yang ditandai dengan pencoretan sel c_1 paling atas pada kolom L2. Barisan ini berkorespondensi dengan barisan vektor

$$\begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}.$$

Barisan ini memberikan $c(1) = 11$, yang merupakan barisan dengan dua simbol 1 yang berurutan. Dalam konteks CSP, barisan ini mengkodekan 2 mobil masuk secara berurutan ke stasiun kerja 1. Hal ini melanggar batasan 1/2 untuk stasiun kerja tersebut, yaitu hanya dapat melayani 1 mobil dari 2 mobil yang masuk secara berurutan.

Pada ekspansi 4 barisan berikutnya yang disajikan pada Tabel 3, pelanggaran restriksi juga terjadi pada

$$c = c_1 c_2 c_1 c_1 = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix},$$

$c(1) = 1011$ melanggar restriksi 1/2 untuk stasiun kerja 1, karena terdapat 2 simbol 1 berurutan.

$$c = c_1 c_2 c_1 c_2 = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix},$$

$c(3) = 1111$ melanggar restriksi 3/5 untuk stasiun kerja 3, karena terdapat 4 simbol 1 berurutan (seharusnya maksimum 3).

$$c = c_1 c_2 c_1 c_3 c_1 = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$c(3) = 11101$ melanggar restriksi 3/5 untuk stasiun kerja 3, karena terdapat 4

simbol 1 (seharusnya maksimum 3) dari faktor panjang 5.

$$c = c_1 c_2 c_1 c_3 c_2 = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$c(3) = 11101$ melanggar restriksi 3/5 untuk stasiun kerja 3, karena terdapat 4 simbol 1 (seharusnya maksimum 3) dari faktor panjang 5.

Ekspansi barisan yang valid diberikan oleh barisan ke-enam pada Tabel 3, yaitu

$$c = c_1 c_2 c_1 c_3 c_3 = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Pada barisan ini, keseluruhan $c(1), c(2), c(3), c(4)$ tidak melanggar restriksi yang diterapkan.

Dengan hanya mengambil *subscript* dari c , barisan ini menginduksi CarSeq (car sequence) $s = 12133$. Oleh karena barisan ini sukses dieskansi sampai level 5, maka barisan ini merepresentasikan urutan valid dalam jalur produksi berdasar restriksi yang berlaku. Barisan 12133 memberikan vektor Parikh 221, yang bermakna bahwa susunan mobil $c_1 c_2 c_1 c_3 c_3$ dalam jalur produksi memberikan output 2 mobil tipe 1, 2 mobil tipe 2, dan 1 mobil tipe 3. Selanjutnya adalah memeriksa apakah output yang dihasilkan sesuai dengan demand yang diinginkan.

Tabel 3. Pembentukan ruang solusi untuk barisan diawali c_1 .

L1	L2	L3	L4	L5	Car Seq	Parikh
c_1	c_2	c_1	e_1			
			e_2			
			c_3			
		c_2	e_1			
			e_2			
			c_3			12133

c_3	c_1	c_2	e_1					
			e_2					
			c_3			12233	122	
		c_3	e_1					
			e_2					
			c_3			12313	212	
		c_3	c_2	e_1				
				e_2				
				c_3			12323	122
	c_3		c_1			12331	212	
			c_2			12332	122	
			c_3			12333	113	
	c_3	c_1	c_2	e_1				
				e_2				
				c_3			13123	212
			c_3	c_1			13131	302
				c_2			13132	212
				c_3			13133	203
		c_3	c_2	c_1	e_1			
					e_2			
					c_3			13212
			c_3	c_2	e_1			
					e_2			
					c_3			13223
c_3			c_2	c_1			13231	212
				c_2			13232	122
				c_3			13233	113
c_3	c_2	c_1	e_1					
			e_2					
			c_3			13312	212	
	c_3	c_2	c_1			13313	203	
			c_2			13321	212	
			c_3			13322	122	
	c_3	c_2	c_1			13323	113	
			c_2			13331	203	
			c_3			13332	113	
c_3	c_2	e_3						

Tabel 4. Ilustrasi pembentukan ruang solusi.

L1	L2	L3	L4	L5	Seq	Parikh		
c_2	c_1	c_2	e_1					
			e_2					
			c_3			c_1	21231	221
		c_3	c_2			c_2	21232	131
			c_3			c_3	21233	122
			c_3	c_2	e_1			
	e_2							

	c_2	c_3	c_3	21323	122	
			c_1	21331	212	
			c_2	21332	122	
			c_3	21333	113	
		c_1	e_1			
			e_2			
			c_1	22131	221	
			c_2	22132	131	
			c_3	22133	122	
		c_3	c_2	e_1		
				e_2		
				c_3	22313	122
	c_1		e_1			
			e_2			
			c_3	22323	032	
	c_3		c_1	22331	122	
			c_2	22332	032	
			c_3	22333	023	
	c_1		c_3	e_1		
				e_2		
				c_3	23123	122
				c_1	23131	212
				c_2	23132	122
				c_3	23133	113
			c_2	e_1		
		e_2				
		c_3		23213	122	
		e_1				
		e_2				
		c_3		23223	032	
		c_3	c_1	23231	122	
			c_2	23232	032	
			c_3	23233	023	
c_1			e_1			
			c_2	23312	122	
			c_3	23313	113	
	c_1	23321	122			
	c_2	23322	032			
	c_3	23323	023			
c_3	c_1	23331	113			
	c_2	23332	023			
	e_3					

Tabel 5. Ilustrasi pembentukan ruang solusi.

L1	L2	L3	L4	L5	Seq	Parikh	
	c_1	c_2	e_1				
			c_1	e_1			
				e_2			
				c_3	31213	212	
			c_2	e_1			
				e_2			
		c_3		31223	122		
		c_3	c_1	31231	212		
			c_2	31232	122		
			c_3	31233	113		
		c_3	c_1	c_1	e_1		
					c_2	31312	212
					c_3	31313	203
				c_2	c_1	31321	212
					c_2	31322	122
					c_3	31323	113
			c_3	c_1	31331	203	
				c_2	31332	113	
	e_3						
	c_2		c_1	c_2	e_1		
					e_2		
					c_3	32123	122
		c_3		c_1	32131	212	
				c_2	32132	122	
				c_3	32133	113	
		c_2	c_1	e_1			
				e_2			
				c_3	32213	122	
			c_3	c_1	32231	122	
				c_2	32232	032	
				c_3	32233	023	
		c_3	c_1	c_1	e_1		
					c_2	32312	122
c_3					32313	113	
c_2	e_1						
	e_2						
	c_3		32323	023			
c_3	c_1		32331	113			
	c_2		32332	023			
	e_3						
c_1	c_1	e_1					
		c_1	33121	212			
		c_2	33122	122			

			c_3	c_3	33123	113	
				c_1	33131	203	
				c_2	33132	113	
			c_2	c_1	e_3		
					e_1		
					c_2	33212	122
				c_2	c_3	33213	113
					c_1	33221	122
					e_2		
				c_3	c_3	33223	023
					c_1	33231	113
					c_2	33232	023
			c_3	c_1	e_3		
					e_1		
					c_2	33312	113
c_2	c_1	33321		113			
	c_2	33322		023			
	e_3						

Berdasarkan Tabel 3, 4, dan 5, diperoleh ruang solusi yang dikategorikan berdasarkan demand, sebagaimana ditampilkan pada Tabel 6.

Perhatikan bahwa ruang solusi untuk contoh ini tidak memuat barisan demand 230, 311, dan 320. Hal ini dikarenakan tidak ada barisan dengan vektor Parikh 230, 311, atau 320 yang dapat dibentuk tanpa melanggar restriksi yang berlaku. Kolom **CarSeq** merupakan semua kemungkinan urutan mobil, yang dikategorikan berdasarkan demand. Kolom **Card** merupakan *cardinality* (banyak anggota himpunan) dari solusi untuk masing-masing demand. Secara keseluruhan, *cardinality* untuk ruang solusi adalah 92.

Tabel 6. Ruang solusi.

Demand	CarSeq				Card
023	22333	23233	23323	23332	10
	32233	32323	32332	33223	
	33232	33322			
032	22323	22332	23223	23232	6
	23322	32232			
113	12333	13233	13323	13332	20
	21333	23133	23313	23331	
	31233	31323	31332	32133	
	32313	32331	33123	33132	

122	33213	33231	33312	33321	28
	12233	12323	12332	13223	
	13232	21233	21323	21332	
	22133	22313	22331	23123	
	23132	23213	23231	23312	
	23321	31223	31232	31322	
	32123	32132	32213	32231	
	32312	33122	33212	33221	
131	21232	22132			2
203	13133	13313	13331	31313	6
	31331	33131			
212	12133	12313	12331	13123	16
	13132	13213	13231	13312	
	21331	23131	31213	31231	
	31312	31321	32131	33121	
221	13212	21231	22131		3
302	13131				1
TOTAL					92

Misalkan demand untuk mobil tipe 1, tipe 2, dan tipe 3 berturut-turut adalah 1, 3, dan 1. Dengan mengacu pada ruang solusi pada Tabel 6 untuk Demand=131, maka terdapat dua kemungkinan pengurutan mobil dalam rangkaian perakitan, yaitu $c_2c_1c_2c_3c_2$ dan $c_2c_2c_1c_3c_2$.

Barisan solusi bersifat refleksif, yang berarti jika $s_1s_2 \dots s_n$ merupakan solusi, maka $s_ns_{n-1} \dots s_1$ juga merupakan solusi.

PENUTUP

Dengan pendekatan barisan kombinatorial, diperoleh bahwa masalah CSP pada dasarnya merupakan masalah barisan kombinatorial terestriksi. Metode yang dibangun pada penelitian ini memperlihatkan bagaimana barisan kombinatorial mengkodekan masalah CSP, dan membangun seluruh barisan yang mungkin berdasarkan restriksi yang diberikan, untuk mengkodekan ruang solusi.

Pendekatan kombinatorial ke depannya memungkinkan pengembangan algoritma efisien yang dapat mengkodekan seluruh elemen ruang solusi. Selain itu dapat pula dikembangkan kajian tentang barisan kombinatorial di mana entri-entri-nya adalah vektor.

DAFTAR PUSTAKA

- [1] C. Artigues, E. Hebrard, V. Mayer-Eichberger, M. Siala, and T. Walsh, "SAT and Hybrid Models of the Car Sequencing Problem," 2014, pp. 268–283. doi: 10.1007/978-3-319-07046-9_19.
- [2] M. Siala, E. Hebrard, and M.-J. Huguet, "A study of constraint programming heuristics for the car-sequencing problem," *Eng Appl Artif Intell*, vol. 38, pp. 34–44, Feb. 2015, doi: 10.1016/j.engappai.2014.10.009.
- [3] I. Moya, M. Chica, and J. Bautista, "Constructive metaheuristics for solving the Car Sequencing Problem under uncertain partial demand," *Comput Ind Eng*, vol. 137, p. 106048, Nov. 2019, doi: 10.1016/j.cie.2019.106048.
- [4] Y. Yu, X. Lu, T. Zhao, M. Cheng, L. Liu, and W. Wei, "Heuristic approaches for the car sequencing problems with block batches," *EURASIP J Wirel Commun Netw*, vol. 2022, no. 1, p. 26, Mar. 2022, doi: 10.1186/s13638-022-02113-7.
- [5] S. Bysko, J. Krystek, and S. Bysko, "Two approaches to car sequencing in the paint shop," *J Phys Conf Ser*, vol. 1780, no. 1, p. 012028, Feb. 2021, doi: 10.1088/1742-6596/1780/1/012028.
- [6] P. Chutima and T. Kirdphoksap, "Solving Many-Objective Car Sequencing Problems on Two-Sided Assembly Lines Using an Adaptive Differential Evolutionary Algorithm," *Engineering Journal*, vol. 23, no. 4, pp. 121–156, Aug. 2019, doi: 10.4186/ej.2019.23.4.121.
- [7] A. Bernini, S. Bilotta, R. Pinzani, A. Sabri, and V. Vajnovszki, "Gray code orders for q -ary words avoiding a given factor," *Acta Inform*, vol. 52, no. 7–8, pp. 573–592, Nov. 2015, doi: 10.1007/s00236-015-0225-2.
- [8] A. Sabri and V. Vajnovszki, "More Restricted Growth Functions: Gray Codes and Exhaustive Generation," *Graphs Comb*, vol. 33, no. 3, pp. 573–582, May 2017, doi: 10.1007/s00373-017-1774-7.
- [9] A. Sabri and V. Vajnovszki, "On the exhaustive generation of generalized ballot sequences in lexicographic and Gray code order," *Pure Mathematics and Applications*, vol. 28, no. 1, pp. 109–119, Oct. 2019, doi: 10.1515/puma-2015-0035.
- [10] A. Sabri and V. Vajnovszki, "Exhaustive generation for ballot sequences in lexicographic and Gray code order," in *Proceedings of GASCom 2018: Random and Exhaustive Generation of Combinatorial Structures*, Athens, Greece, 2018, pp. 195–201.