

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Permasalahan penjadwalan perawat atau *Nurse Rostering Problem* (NRP) telah menjadi perhatian penting dalam sistem layanan kesehatan di berbagai negara. Kekurangan tenaga dan tingginya tingkat keluar-masuk perawat (*turnover*) memperburuk ketidakseimbangan antara kebutuhan dan ketersediaan sumber daya manusia di sektor kesehatan. Berdasarkan meta-analisis global terhadap 21 studi di 14 negara, dilaporkan bahwa rata-rata tingkat *turnover* perawat mencapai 16% per tahun. Kondisi ini berdampak pada peningkatan beban kerja, penurunan mutu pelayanan, serta meningkatnya risiko insiden klinis (Ren dkk., 2024). Penelitian lain juga menunjukkan bahwa kekurangan tenaga perawat pada unit dengan intensitas tinggi, seperti Instalasi Gawat Darurat (IGD), berhubungan dengan keterlambatan pelayanan, antrean pasien yang panjang, dan menurunnya tingkat kepuasan pasien (Drennan dkk., 2024).

Di Indonesia, masalah ini diperparah oleh rasio perawat terhadap pasien yang belum ideal dan keterbatasan fleksibilitas dalam sistem penjadwalan. Tercatat bahwa beban kerja perawat pada *shift* malam cenderung lebih berat dibandingkan *shift* lainnya, dengan cakupan jumlah pasien lebih besar dan sumber daya lebih terbatas (Sugianto dkk., 2022). Oleh karena itu, perencanaan jadwal perawat yang efektif menjadi salah satu komponen kunci dalam menjaga keberlangsungan pelayanan rumah sakit.

Salah satu aspek paling krusial dalam penjadwalan perawat adalah *shift* malam, yang berimplikasi langsung terhadap kesehatan dan kinerja tenaga perawat. Kubo dkk. (2024) menunjukkan bahwa pola *shift* malam yang tidak proporsional seperti jumlah *shift* malam yang terlalu padat atau tidak disertai waktu istirahat yang cukup dapat menyebabkan kelelahan (*fatigue*), gangguan tidur, penurunan konsentrasi kerja, serta meningkatkan risiko *near miss* atau kesalahan klinis. Disisi lain Klyve dkk. (2023) juga menemukan bahwa paparan *shift* malam secara kumulatif berkontribusi terhadap kelelahan kronis, yang berdampak negatif pada performa

klinis. Dalam konteks operasional rumah sakit, pengaturan struktur *shift* malam menjadi salah satu kendala keras yang sangat memengaruhi kelayakan jadwal.

Selain struktur *shift* malam, pemerataan jumlah perawat per *shift* (*team coverage*) juga berperan penting dalam efektivitas operasional. Ketidakseimbangan jumlah staf antara *shift* pagi, siang, dan malam dapat meningkatkan tekanan kerja pada tim tertentu, memperburuk beban fisik dan mental, serta menurunkan mutu layanan keperawatan (Nydahl dkk., 2024). Pemerataan tim yang tidak optimal sering kali menjadi penyebab ketidaklayakan jadwal (*infeasibility*) dalam optimasi NRP.

Permasalahan ini mendorong berkembangnya berbagai pendekatan optimasi penjadwalan perawat. Sejak awal tahun 2000 hingga sekarang, penelitian NRP mengalami pergeseran dari metode manual menuju algoritma komputasi canggih. Berbagai studi menunjukkan bahwa NRP tergolong masalah NP-hard, yang berarti kompleksitasnya meningkat eksponensial seiring bertambahnya jumlah perawat, *shift*, dan aturan penjadwalan (den Hartog dkk., 2023). Karena itu, banyak penelitian berfokus pada pengembangan metode optimasi yang mampu menghasilkan solusi layak secara cepat.

Salah satu pendekatan yang banyak digunakan adalah metode eksak, seperti *Integer Programming* (IP) dan *Constraint Programming* (CP), yang efektif dalam menangani kendala keras secara presisi (Mystakidis dkk., 2024). Pendekatan ini cocok untuk skala kecil hingga menengah, tetapi memiliki keterbatasan dalam skalabilitas saat jumlah perawat dan aturan meningkat. Untuk mengatasi masalah skala besar, banyak Peneliti beralih ke pendekatan *metaheuristik*, seperti *Tabu Search* yang menawarkan fleksibilitas tinggi dalam menangani dinamika *shift* yang kompleks (Ramli dkk., 2020), serta *Simulated Annealing* (Ceschia dkk., 2023; Zanazzo dkk., 2024) dan *Genetic Algorithm* (Oyeleye & Oladele, 2020) yang dapat mengeksplorasi ruang solusi besar secara efisien.

Dalam beberapa tahun terakhir, pendekatan *hybrid* banyak digunakan sebagai solusi, yaitu penggabungan dua atau lebih algoritma untuk mengatasi keterbatasan metode tunggal. Turhan & Bilgen (2020) menunjukkan efektivitas *Hybrid Simulated Annealing* dalam menangani permasalahan NRP kompleks, sedangkan

Harb dkk. (2024) menunjukkan keunggulan *hybrid* GA–PSO dalam menghasilkan jadwal yang lebih stabil dan adaptif terhadap perubahan operasional. Namun demikian, penelitian yang secara spesifik menggabungkan GA dan SA untuk NRP masih sangat terbatas, sehingga membuka peluang untuk eksplorasi lebih lanjut, terutama untuk mendukung operasional di rumah sakit Indonesia yang memiliki tantangan berbeda dari negara maju.

Evaluasi performa pada penelitian terdahulu umumnya menggunakan metrik kelayakan solusi (*feasibility*), pelanggaran kendala, pemenuhan kebutuhan per *shift*, pemerataan beban kerja (*fairness*), dan waktu komputasi. Sejalan dengan meningkatnya kompleksitas kendala penjadwalan, berbagai penelitian mengusulkan metode eksak, metaheuristik, maupun hibrida untuk menghasilkan jadwal yang layak dan berkualitas. Untuk memperjelas konteks dan posisi penelitian ini, Tabel 2.1 merangkum beberapa penelitian terdahulu beserta karakteristiknya, khususnya terkait pendekatan optimasi yang digunakan, fokus kendala/objektif, dan ruang lingkup evaluasi.

SEKOLAH PASCASARJANA

Tabel 2.1 Ringkasan penelitian terdahulu terkait optimasi NRP

No	Peneliti (Tahun)	Fokus	Metode	Fokus Kendala/Objektif	Temuan
1	Amindoust dkk. (2021)	<i>Nurse scheduling</i> dengan pertimbangan kelelahan	<i>Hybrid Genetic Algorithm</i>	Mempertimbangkan <i>fatigue factor</i> , batasan kerja, preferensi/aturan penjadwalan	Pendekatan hibrida GA efektif untuk menyeimbangkan kualitas jadwal dan faktor kelelahan.
2	Guo & Bard (2022)	Midterm nurse scheduling	<i>Column Generation / Branch-and-Price</i> (berbasis optimisasi eksak)	Pemenuhan kendala keras secara ketat, efisiensi komputasi untuk horizon menengah	<i>Column generation</i> membantu skala lebih besar dengan dekomposisi, namun rancangan model lebih kompleks.
3	den Hartog dkk. (2023)	Kajian kompleksitas NRP	Analisis kompleksitas (teoretis)	Memformalkan aspek NRP dan tingkat kesulitan komputasional	Menegaskan NRP sebagai masalah optimasi kombinatorial yang sangat menantang sehingga perlu <i>strategi</i> penyelesaian yang tepat.
4	Ceschia dkk. (2023)	<i>Real-world nurse rostering</i>	<i>Simulated Annealing</i>	Menjaga <i>feasibility</i> + minimasi pelanggaran/ <i>penalty</i> , perbaikan kualitas roster	SA efektif untuk kasus nyata, namun kualitas dipengaruhi desain <i>neighborhood/penalty</i> dan parameter.
5	Otero-Caicedo dkk. (2023)	<i>Nurse scheduling</i> dengan absenteeism + preferensi	<i>Preventive-Reactive approach</i>	<i>Robustness</i> terhadap ketidakhadiran, minim gangguan jadwal, akomodasi preferensi	Pendekatan preventif-reaktif meningkatkan ketahanan jadwal saat terjadi perubahan/absen.

Tabel 2.1 Ringkasan penelitian terdahulu terkait optimasi NRP (lanjutan)

No	Peneliti (Tahun)	Fokus	Metode	Fokus Kendala/Objektif	Temuan
6	Mystakidis dkk. (2024)	Studi kasus <i>nurse rostering</i> (konteks klinik/RS)	Integer Programming / MIP	Kepatuhan aturan kerja, efisiensi operasional, kepuasan/performance	IP/MIP menunjukkan peningkatan efisiensi & alokasi sumber daya, cocok untuk aturan ketat.
7	Mukhlason dkk. (2024)	<i>Nurse rostering</i>	Reinforcement Learning + SA (reheating) hyper-heuristic	Multi-constraint scheduling, peningkatan kualitas solusi via hyper-heuristic	Kombinasi RL-SA dapat meningkatkan kualitas solusi dibanding baseline tertentu, namun desain reward/heuristic krusial.
8	Alaouchiche dkk. (2024)	<i>Real-world</i> NSP (RS Prancis)	Model matematis (optimisasi) + kriteria <i>workload</i> balancing	Workload balancing , regulasi waktu kerja, preferensi, kesejahteraan perawat	Menawarkan dan membandingkan beberapa kriteria <i>workload</i> balancing; validasi eksperimen besar & umpan balik praktik.
9	Al Hajj dkk. (2025)	<i>Healthcare</i> scheduling/NSP (menggunakan dataset publik)	Machine Learning + Constraint Programming (CSP)	Pembelajaran pola/ <i>constraint</i> dari data historis + penyelesaian eksplisit berbasis CSP	Menggabungkan ML (implicit learning) dan model CSP eksplisit; status preprint (belum peer-review) .

Tabel 2.1 Ringkasan penelitian terdahulu terkait optimasi NRP (lanjutan)

No	Peneliti (Tahun)	Fokus	Metode	Fokus Kendala/Objektif	Temuan
10	Boðvarsdóttir & Stidsen (2025)	Personnel rostering (multi-objective)	<i>Review multi-objective optimization</i>	Multi-objective trade-off (fairness, cost, coverage, preferensi)	Merangkum metode multi-objektif dan tren riset untuk masalah penjadwalan personel/rostering.
11	Ramli dkk. (2020)	NRP real-world dengan preferensi perawat dan perbaikan kualitas roster	Intelligent <i>Tabu Search</i> (dua fase: semi-random initialization + neighborhood improvement; inovasi neighborhood/representasi)	Menjaga kendala keras tidak dilanggar sejak inisialisasi; objektif mencakup: meminimalkan imbalance coverage, meminimalkan ketidak-sukaan/preferensi negatif, dan pemerataan distribusi <i>shift</i>	<i>Strategi</i> peningkatan neighborhood + inisialisasi semi-random meningkatkan kualitas roster; TS dengan enhancement dinilai efektif & efisien terutama untuk regulasi kerja nyata dan preferensi perawat
12	Abayomi-Alli dkk. (2022)	Pengembangan sistem otomatis NSP untuk meminimalkan pelanggaran <i>soft-constraints</i> dan memenuhi preferensi	Hybrid GA + <i>Tabu Search</i> memory (GATS_NSS)	Optimasi alokasi <i>shift</i> /ward berdasarkan <i>constraint</i> RS; fokus pada pemenuhan kendala keras dan meminimalkan biaya pelanggaran kendala lunak	Pada dataset studi kasus 151 perawat, sistem hybrid menunjukkan performa tinggi: akurasi rata-rata 94%, allocation rate 99%, 0% duplication, 0.5% clash, serta peningkatan waktu komputasi ~94% dibanding manual

Berdasarkan ringkasan pada Tabel 2.1, masih terdapat beberapa kesenjangan. Pertama, sebagian besar penelitian menitikberatkan pada satu pendekatan optimasi tertentu, sehingga perbandingan lintas pendekatan (eksak–metaheuristik–hibrida) pada konfigurasi kendala yang sama belum banyak dilakukan secara konsisten. Kedua, variasi struktur kendala dan fokus objektif (misalnya preferensi, absensi, overtime, fatigue, atau workload balancing) menyebabkan hasil performa antar studi sulit dibandingkan secara langsung. Ketiga, penelitian yang secara eksplisit memusatkan evaluasi pada kombinasi kendala kunci operasional, yaitu struktur/pola shift malam dan pemerataan jumlah perawat per shift (team coverage), dalam kerangka evaluasi yang terukur masih terbatas.

Berdasarkan kesenjangan tersebut, penelitian ini diposisikan untuk melakukan kajian komparatif tiga pendekatan optimasi, yaitu CP-SAT (OR-Tools), Tabu Search, dan Hybrid Genetic Algorithm–Simulated Annealing (HGASA), dengan fokus pada struktur/pola shift malam dan pemerataan team coverage. Evaluasi dilakukan pada beberapa skenario penjadwalan bulanan dengan variasi skala dan batas waktu komputasi menggunakan data studi kasus unit rawat inap SMC RS Telogorejo Semarang serta variasi data dummy, sehingga menghasilkan rekomendasi pemilihan algoritma yang lebih relevan terhadap kebutuhan operasional rumah sakit.

2.2 Dasar Teori

2.2.1 *Nurse Rostering Problem* (NRP)

Penjadwalan perawat atau *Nurse Rostering Problem* (NRP) merupakan masalah penugasan sumber daya manusia, di mana sejumlah perawat harus dijadwalkan untuk mengisi sejumlah *shift* dalam periode waktu tertentu, seperti satu minggu atau satu bulan. Tujuan utama dari NRP adalah memastikan ketersediaan tenaga perawat sesuai kebutuhan pelayanan klinis, sambil tetap mematuhi berbagai aturan dan batasan yang berlaku, baik dari sisi regulasi ketenagakerjaan maupun kebijakan rumah sakit (Ramli dkk., 2020; den Hartog dkk., 2023).

NRP secara umum melibatkan dua jenis kendala utama, yaitu:

1. Kendala keras

Kendala keras merupakan aturan yang harus dipenuhi agar jadwal dianggap layak (*feasible*). Contohnya mencakup jumlah minimal staf per *shift*, batas maksimal jumlah *shift* malam berturut-turut, waktu istirahat antar *shift*, ketentuan regulasi jam kerja, dan ketersediaan tenaga pada hari-hari tertentu. Pelanggaran terhadap kendala keras tidak diperbolehkan karena akan berdampak langsung pada kesinambungan operasional dan keselamatan pasien (den Hartog dkk., 2023; Mystakidis dkk., 2024).

2. Kendala lunak

Kendala lunak merupakan aturan atau preferensi yang sebisa mungkin dipenuhi, tetapi pelanggarannya masih dapat ditoleransi dalam batas tertentu. Contohnya meliputi pemerataan beban kerja (*fairness*), preferensi jadwal individu, atau distribusi *shift* malam yang seimbang. Pelanggaran terhadap kendala lunak biasanya dihitung sebagai *penalty* dalam fungsi objektif (Ramli dkk., 2020; Mystakidis dkk., 2024).

Karakteristik NRP yang melibatkan banyak kendala saling terkait dan tujuan ganda (*multi-objective*) menjadikannya sulit untuk diselesaikan secara manual. Selain itu, penambahan jumlah perawat, *shift*, atau aturan membuat kompleksitasnya meningkat secara eksponensial. Secara matematis, NRP dikategorikan sebagai masalah optimasi kombinatorial NP-*hard*, yang artinya tidak ada algoritma deterministik efisien yang dapat menjamin solusi optimal dalam waktu yang wajar untuk skala besar (den Hartog dkk., 2023; Klyve dkk., 2023; Böðvarsdóttir & Stidsen, 2025).

Dalam praktik modern, penjadwalan perawat juga dikaitkan dengan aspek teknologi melalui penggunaan sistem *electronic rostering* (*e-rostering*) dan *self-rostering*. Sistem ini memungkinkan proses penjadwalan lebih efisien, fleksibel, serta meningkatkan kepuasan dan keseimbangan kerja–hidup tenaga kesehatan. Namun, sistem tersebut tetap memerlukan model optimasi yang kuat untuk memastikan kepatuhan terhadap kendala operasional (O’Connell dkk., 2024).

Dengan demikian, NRP bukan hanya permasalahan matematis, tetapi juga menyangkut aspek operasional dan manajerial rumah sakit. Kombinasi antara pemodelan kendala keras dan lunak, pendekatan optimasi, serta dukungan teknologi *e-rostering* menjadi kunci untuk menghasilkan jadwal yang layak, adil, dan efisien (Ramli dkk., 2020; Mystakidis dkk., 2024; Böðvarsdóttir & Stidsen, 2025; O'Connell dkk., 2024).

2.2.2 Pemodelan Matematis NRP

Untuk menyelesaikan *Nurse Rostering Problem* (NRP), pendekatan matematis digunakan untuk merepresentasikan permasalahan dalam bentuk formulasi optimasi. Pemodelan ini bertujuan menerjemahkan aturan operasional rumah sakit ke dalam bentuk himpunan, variabel keputusan, fungsi objektif, dan kendala, sehingga dapat diselesaikan menggunakan teknik optimasi eksak maupun metaheuristik. Pendekatan ini memberikan struktur formal yang memungkinkan berbagai algoritma diterapkan secara sistematis untuk menghasilkan jadwal yang optimal atau mendekati optimal (*near-optimal*) (Mystakidis dkk., 2024; Böðvarsdóttir & Stidsen, 2025).

1. Himpunan dan Indeks

Dalam formulasi standar NRP, komponen berikut didefinisikan:

- a. $I = \{1, 2, \dots, N\}$: himpunan perawat.
- b. $J = \{1, 2, \dots, S\}$: himpunan *shift* per hari (misalnya pagi, siang, malam).
- c. $K = \{1, 2, \dots, D\}$: himpunan hari dalam periode penjadwalan (misalnya 7 atau 28 hari).

Pendefinisian himpunan dan indeks ini sangat penting karena menentukan struktur kendala dan ruang solusi dalam model optimasi (Sukonpat & Jarumaneeroj, 2023; Ramli dkk., 2020).

2. Variabel Keputusan

Untuk memformulasikan penjadwalan perawat, digunakan variabel keputusan biner yang merepresentasikan penugasan perawat pada setiap *shift*

dalam satu hari (Ceschia dkk., 2023; Sukonpat & Jarumaneeroj, 2023). Misalkan $I = \{1, \dots, N\}$ himpunan perawat, $J = \{P, S, M, L\}$ himpunan jenis *shift* (Pagi, Siang, Malam, Libur), dan $K = \{1, \dots, D\}$ himpunan hari dalam periode penjadwalan. Variabel keputusan utama didefinisikan sebagai $x_{i,j,k} = 1$ jika perawat i dijadwalkan pada *shift* j dihari k atau $x_{i,j,k} = 0$ jika perawat i tidak dijadwalkan pada *shift* j dihari k dengan $i \in I, j \in J, \text{ dan } k \in K$ yang masing-masing menunjukkan indeks untuk perawat, *shift*, dan hari dalam periode penjadwalan. Sebagai besaran turunan untuk evaluasi, didefinisikan jumlah *shift* malam per perawat:

$$y_i = \sum_{k \in K} x_{i,M,k} \quad (2.1)$$

Dengan:

y_i : jumlah *shift* malam yang diterima perawat ke- i pada satu periode

$x_{i,M,k}$: variabel biner, bernilai 1 jika perawat ke- i dijadwalkan pada *shift* malam M di hari ke- k , dan 0 jika tidak.

i : indeks perawat, $i = 1, \dots, N$

k : indeks hari dalam periode penjadwalan, $k = 1, \dots, D$

K : himpunan hari dalam periode penjadwalan

M : kode *shift* malam (anggota dari himpunan *shift* $J = P, S, M, L$

serta total jam kerja per perawat pada satu periode (dengan h_j durasi kerja *shift* j) didefinisikan sebagai:

$$w_i = \sum_{k \in K} \sum_{j \in J} h_j x_{i,j,k}, \quad (2.2)$$

Dengan:

w_i : total jam kerja perawat ke- i selama satu periode penjadwalan

h_j : durasi (jam kerja) *shift* ke- j

$x_{i,j,k}$: variabel biner, bernilai 1 jika perawat ke- i dijadwalkan pada *shift* ke- j di hari ke- k , dan 0 jika tidak

i : indeks perawat, $i = 1, \dots, N$

j : indeks *shift*, dengan $J = \{P, S, M, L\}$ (Pagi, Siang, Malam, Libur)

k : indeks hari, $k = 1, \dots, D$

J : himpunan jenis *shift*

K : himpunan hari dalam periode penjadwalan

Selain itu digunakan pula variabel *penalty nonnegative* $z_c \geq 0$ untuk masing-masing komponen kendala lunak (misalnya pemerataan *shift* malam, pemerataan tim, atau preferensi libur), yang akan diakumulasikan di fungsi objektif.

Sebagai ilustrasi, variabel *penalty* z_{night} digunakan untuk merepresentasikan ketidakseimbangan distribusi *shift* malam, misalnya ketika terdapat perawat yang menerima jumlah *shift* malam lebih banyak dibanding rekan lainnya. Variabel $z_{coverage}$ menggambarkan ketidakterpenuhinya jumlah perawat pada suatu *shift* sehingga terjadi kekurangan personel, khususnya pada *shift* malam. Sementara itu, variabel z_{prev} menunjukkan *penalty* akibat pelanggaran preferensi individu, misalnya ketika seorang perawat telah mengajukan permintaan hari libur namun tetap dijadwalkan bertugas pada hari tersebut (Ramli dkk., 2020; Guo & Bard, 2022; Zanazzo dkk., 2025).

3. Fungsi Objektif

Dalam penyelesaian *Nurse Rostering Problem* (NRP), fungsi objektif dirumuskan sebagai minimisasi total *penalty* yang timbul akibat pelanggaran kendala lunak yang berkaitan dengan kualitas jadwal. Secara umum, tujuan optimasi adalah menghasilkan jadwal yang layak (*feasible*) terhadap seluruh kendala keras, sekaligus meminimalkan ketidakseimbangan pada aspek-aspek kinerja operasional penjadwalan.

$$\min Z = w_1 \cdot P_{night} + w_2 \cdot P_{coverage} + w_3 \cdot P_{workload} + w_4 \cdot P_{preference} \quad (2.3)$$

Dengan P_{night} merupakan *penalty* pelanggaran struktur *shift* malam (misalnya malam tidak diikuti oleh libur), $P_{coverage}$ merupakan *penalty* ketidakseimbangan jumlah perawat antar *shift* (*team coverage*), $P_{workload}$ *penalty* ketidakmerataan total jam kerja antar perawat, $P_{preference}$ merupakan *penalty* pelanggaran preferensi *shift* atau hari libur staf dan w_1, w_2, w_3, w_4 merupakan bobot prioritas *penalty* yang ditentukan oleh kebijakan rumah sakit.

Fungsi objektif ini digunakan untuk menyeimbangkan beberapa aspek kualitas jadwal, antara lain:

- Pemerataan beban kerja, agar tidak terjadi kelelahan berlebih yang berpotensi menurunkan mutu pelayanan.
- Kesesuaian dengan preferensi staf, untuk meningkatkan kenyamanan kerja dan mengurangi turnover.
- Distribusi *shift* malam yang adil, karena *shift* malam terbukti memiliki dampak signifikan terhadap kelelahan, risiko kesalahan klinis, dan kelelahan kronis (Kubo dkk., 2024; Fukuda & Hayashi, 2024).
- Pemerataan jumlah perawat per *shift* (*team coverage*), untuk menghindari ketidakseimbangan beban antar *shift* yang dapat memengaruhi keselamatan pasien (Nydahl dkk., 2024).

Dengan demikian, fungsi objektif ini tidak hanya meminimalkan pelanggaran kendala, tetapi juga mengoptimalkan kualitas operasional jadwal perawat sesuai kebutuhan rumah sakit.

4. Kendala Keras

Kendala keras merupakan aturan atau syarat yang harus dipenuhi agar jadwal dianggap layak (*feasible*). Pelanggaran terhadap kendala keras tidak dapat ditoleransi karena dapat mengakibatkan gangguan operasional, pelanggaran regulasi ketenagakerjaan, ataupun risiko terhadap keselamatan pasien maupun perawat. Oleh karena itu, kendala keras menjadi dasar pembentukan struktur jadwal dalam permasalahan penjadwalan perawat.

Secara umum, kendala keras dalam *Nurse Rostering Problem* (NRP) meliputi:

a. Satu *Shift* per Hari untuk Setiap Perawat

Setiap perawat hanya dapat dijadwalkan pada satu *shift* dalam satu hari untuk menghindari beban kerja berlebih dan penugasan yang tumpang tindih (den Hartog dkk., 2023; Ramli dkk., 2020).

b. Pemenuhan Kebutuhan Minimal Staf per *Shift*

Setiap *shift* harus diisi oleh jumlah perawat minimal sesuai kebutuhan pelayanan, standar mutu klinis, serta ketentuan keselamatan pasien. Kekurangan jumlah staf pada *shift* tertentu dapat meningkatkan beban kerja dan risiko insiden klinis (Drennan dkk., 2024; Alaouchiche dkk., 2024).

c. Pembatasan Jam Kerja dan Hari Kerja Maksimum

Pembatasan ini mengikuti kebijakan ketenagakerjaan dan ketentuan rumah sakit, misalnya batas jam kerja mingguan, jumlah hari kerja berturut-turut, dan kewajiban pemberian waktu istirahat (Mystakidis dkk., 2024).

d. Aturan Rotasi *Shift* untuk Mengurangi Kelelahan

Rotasi *shift* harus mempertimbangkan waktu pemulihan, terutama setelah *shift* malam. Model biologis dan studi kelelahan menunjukkan bahwa kurangnya waktu istirahat setelah *shift* malam meningkatkan risiko kesalahan dan kelelahan kronis (Klyve dkk., 2023; Kubo dkk., 2024).

e. Ketersediaan Perawat Berdasarkan Jadwal Pribadi atau Tugas Tambahan

Perawat yang sedang cuti, pelatihan, pendidikan, atau penugasan khusus tidak dapat dijadwalkan untuk *shift* klinis pada hari yang bersangkutan.

Kendala keras memainkan peran fundamental dalam memastikan bahwa solusi penjadwalan tidak hanya optimal secara matematis, tetapi juga realistis dan dapat diterapkan dalam konteks operasional rumah sakit. Metode optimasi apa pun, baik berbasis *Constraint Programming*, *Integer Programming*, maupun metaheuristik, harus memastikan bahwa kendala keras ini selalu terpenuhi sebelum mempertimbangkan aspek pengoptimalan lainnya (Turhan & Bilgen, 2020; den Hartog dkk., 2023).

5. Kendala Lunak

Kendala lunak merupakan ketentuan atau preferensi yang diupayakan untuk dipenuhi, namun masih dapat dilanggar dalam batas tertentu. Pelanggaran terhadap kendala ini tidak menyebabkan jadwal menjadi tidak layak, tetapi akan mengurangi kualitas jadwal, misalnya menyebabkan kelelahan perawat atau ketidakpuasan kerja. Oleh karena itu, pelanggaran kendala lunak biasanya diberi nilai *penalty* yang akan diminimalkan melalui fungsi objektif.

Secara umum, kendala lunak dalam Nurse Rostering Problem (NRP) mencakup aspek berikut:

a. Pemerataan Jumlah *Shift* Malam per Perawat

Distribusi *shift* malam perlu dijaga agar tidak terkonsentrasi pada individu tertentu. Beban *shift* malam yang tidak merata terbukti meningkatkan risiko kelelahan, penurunan kinerja, serta potensi kesalahan klinis (Klyve dkk., 2023; Kubo dkk., 2024).

b. Pemenuhan Preferensi Perawat

Perawat dapat memiliki preferensi pribadi terhadap hari libur tertentu atau *shift* kerja yang lebih disukai. Mempertimbangkan preferensi ini dapat meningkatkan kenyamanan kerja dan menurunkan risiko *turnover* (Ramli dkk., 2020; Razali dkk., 2025).

c. Rotasi *Shift* yang Memperhatikan Aspek Kelelahan (*Fatigue*)

Rotasi *shift* idealnya mempertahankan pola kerja dan istirahat yang seimbang. Rotasi yang buruk, seperti penugasan *shift* pagi segera setelah *shift* malam, dapat menyebabkan beban fisiologis yang berlebihan (Fukuda & Hayashi, 2024).

d. Pemerataan Jumlah Perawat pada Setiap *Shift* (*Team Coverage*)

Jumlah perawat antar *shift* (pagi–siang–malam) sebaiknya seimbang sesuai dengan kebutuhan pelayanan agar tidak terjadi ketimpangan beban kerja antar waktu operasional (Nydahl, dkk, 2024; Alaouchiche, dkk, 2024).

- e. Aspek ketersediaan perawat (*availability*), seperti cuti, pelatihan, dan penugasan non-klinis, juga perlu dimodelkan sebagai kendala keras (Sugianto dkk., 2022).

Pelanggaran terhadap kendala lunak tidak menghilangkan kelayakan jadwal, tetapi akan diberikan *penalty* dalam fungsi objektif, sehingga semakin besar pelanggaran maka akan mengakibatkan semakin besar *penalty* dan menjadikan kualitas jadwal semakin rendah. Dengan demikian, kendala lunak berperan penting dalam:

- a. meningkatkan kualitas kesejahteraan perawat,
 - b. mendukung keselamatan pasien, dan
 - c. menjaga stabilitas operasional unit perawatan.
- (Ramli, dkk, 2020; Sukonpat & Jarumaneeroj, 2022)

6. Karakteristik Formulasi

Formulasi matematis dalam *Nurse Rostering Problem* (NRP) pada dasarnya merupakan permasalahan optimasi kombinatorial yang melibatkan banyak variabel keputusan biner dan kendala yang saling bergantung. Permasalahan ini bersifat *multi-objective*, karena harus memperhitungkan berbagai aspek kualitas jadwal (misalnya beban kerja, pemerataan *shift* malam, dan preferensi staf) secara bersamaan. Selain itu, NRP dikategorikan sebagai NP-hard, yang berarti bahwa kompleksitas komputasi meningkat secara eksponensial seiring bertambahnya jumlah perawat, *shift*, dan hari penjadwalan (den Hartog dkk., 2023).

Pendekatan metode eksak, seperti *Mixed Integer Programming* (MIP), *Integer Linear Programming* (ILP), dan *Constraint Programming* (CP-SAT), efektif dalam menghasilkan solusi optimal terutama pada ukuran masalah kecil hingga menengah, karena metode ini mampu menjamin pemenuhan kendala secara ketat dan memberikan solusi yang dapat dibuktikan optimal (Mystakidis dkk., 2024). Namun, untuk ukuran masalah yang lebih besar, metode eksak sering mengalami peningkatan waktu komputasi yang signifikan, sehingga menjadi kurang efisien.

Untuk permasalahan dengan skala besar dan struktur kendala yang kompleks, pendekatan metaheuristik seperti *Tabu Search*, *Simulated Annealing*, atau algoritma *hybrid* (misalnya *Hybrid Genetic Algorithm–Simulated Annealing*) menjadi lebih efektif. Pendekatan metaheuristik tidak menjamin solusi optimal secara matematis, namun mampu menghasilkan solusi yang baik dalam waktu komputasi yang lebih singkat, dengan kemampuan eksplorasi ruang solusi yang luas dan fleksibilitas dalam menangani berbagai variasi aturan penjadwalan (Ramli dkk., 2020; Zanazzo dkk., 2025).

Selain metode optimasi, formulasi penjadwalan modern juga mempertimbangkan aspek teknologi operasional, seperti integrasi dengan sistem *electronic rostering* (*e-rostering*). Sistem ini memungkinkan penjadwalan dilakukan secara digital, disertai fitur pengaturan preferensi staf, pemantauan beban kerja, dan penyesuaian jadwal secara dinamis sesuai kebutuhan unit keperawatan. Integrasi preferensi staf dalam formulasi matematis diketahui dapat meningkatkan kepuasan kerja dan mengurangi risiko *turnover*, serta mendukung kesinambungan layanan keperawatan (O'Connell dkk., 2024).

2.2.3 **Constraint Programming & CP-SAT (Google OR-Tools)**

Constraint Programming (CP) merupakan paradigma pemodelan dan penyelesaian masalah optimasi kombinatorial yang berfokus pada pernyataan kendala (*constraints*), bukan pada prosedur penyelesaian. Dalam CP, solusi didefinisikan sebagai kumpulan nilai yang memenuhi semua kendala yang telah dirumuskan (Perron dkk., 2023). CP sangat sesuai digunakan pada *scheduling problem*, termasuk *Nurse Rostering Problem* (NRP), karena dapat secara fleksibel mengakomodasi kendala keras dan lunak, seperti jumlah minimal staf, batas *shift* malam, serta preferensi perawat.

Salah satu keunggulan CP dibandingkan metode optimasi konvensional adalah kemampuannya melakukan *constraint propagation* yaitu proses otomatis yang mempersempit ruang solusi tanpa eksplorasi eksplisit terhadap semua

kemungkinan kombinasi. Hal ini membuat CP efisien untuk kasus skala kecil hingga menengah, terutama ketika aturan klinis cukup ketat (Mystakidis dkk., 2024; O'Connell dkk., 2024).

Salah satu implementasi CP modern yang sangat populer adalah CP-SAT *Solver* dari Google OR-Tools. CP-SAT merupakan kombinasi *Constraint Programming* dan SAT *solver* sehingga mampu menangani model dengan banyak variabel biner secara efisien. CP-SAT menggunakan teknik *branch-and-bound*, *linear relaxation*, serta *learning-based propagation* untuk mempercepat pencarian solusi optimal atau mendekati optimal (Perron & Feydy, 2020).

Dalam CP-SAT, variabel keputusan umumnya berupa variabel biner:

$$x_{i,j,k} \in \{0,1\}, \forall i \in I, j \in J, k \in K \quad (2.4)$$

Dengan:

$x_{i,j,k}$: variabel biner, bernilai 1 jika perawat ke- i dijadwalkan pada *shift* ke- j di hari ke- k , dan 0 jika tidak

i : indeks perawat, $i = 1, \dots, |I|$

j : indeks *shift* (Pagi, Siang, Malam, Libur), $j = 1, \dots, |J|$

k : indeks hari dalam periode penjadwalan, $k = 1, \dots, |K|$

I : himpunan perawat

J : himpunan jenis *shift*

K : himpunan hari dalam periode penjadwalan

Beberapa kendala pada CP-SAT dapat diformulasikan sebagai berikut:

1. Kebutuhan minimal staf per *shift* (*coverage constraint*):

$$\sum_{i \in I} x_{i,j,k} \geq \text{minStaff}_{j,k}, \forall j \in J, k \in K \quad (2.5)$$

Dengan:

$x_{i,j,k}$: variabel biner penugasan perawat ke- i pada *shift* ke- j di hari ke- k

$\text{minStaff}_{j,k}$: kebutuhan minimal jumlah perawat yang harus bertugas pada

shift ke- j di hari ke- k

i : indeks perawat, $i \in I$

j : indeks *shift*, $j \in J$

k : indeks hari, $k \in K$

I : himpunan perawat

J : himpunan jenis *shift*

K : himpunan hari dalam periode penjadwalan

2. Satu *shift* per hari untuk setiap perawat:

$$\sum_{j \in J} x_{i,j,k} \leq 1, \forall i \in I, k \in K \quad (2.6)$$

Dengan:

$x_{i,j,k}$: variabel biner penugasan perawat ke- i pada *shift* ke- j di hari ke- k

i : indeks perawat, $i \in I$

j : indeks *shift*, $j \in J$

k : indeks hari, $k \in K$

J : himpunan jenis *shift*

I : himpunan perawat

K : himpunan hari dalam periode penjadwalan

konstanta 1: batas maksimum *shift* yang boleh diterima satu perawat dalam satu hari

3. Kendala rotasi *shift* malam menggunakan *reified constraint* / kendala bersyarat yang secara konseptual dituliskan seperti:

$$\text{Jika } x_{i,M,k} = 1 \text{ maka } x_{i,L,k+1} = 1 \quad (2.7)$$

dan direpresentasikan pada *CP-SAT (OR-Tools)* menggunakan fungsi *AddBoolOr*($[Not(x_{i,M,k}), x_{i,L,k+1}]$) untuk menambahkan kendala logika OR atas daftar variable biner, dengan:

$x_{i,M,k}$: variabel biner, 1 jika perawat ke- i dijadwalkan pada *shift* malam di hari ke- k

$x_{i,L,k+1}$: variabel biner, 1 jika perawat ke- i dijadwalkan libur di hari ke- $k + 1$

M : kode *shift* malam (anggota himpunan *shift* J)

L : kode hari libur (anggota himpunan *shift* J)

$\text{Not}(\cdot)$: negasi boolean terhadap variabel biner

4. Fungsi objektif untuk meminimalkan *penalty* pelanggaran pada kendala lunak:

$$\min Z = \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} c_{i,j,k} x_{i,j,k} \quad (2.8)$$

Dengan:

Z : nilai fungsi objektif yang akan diminimalkan (*penalty* jadwal)

$c_{i,j,k}$: koefisien *penalty* penugasan perawat ke- i pada *shift* ke- j di hari ke- k

$x_{i,j,k}$: variabel biner penugasan perawat ke- i pada *shift* ke- j di hari ke- k

i : indeks perawat, $i \in I$

j : indeks *shift*, $j \in J$

k : indeks hari, $k \in K$

I : himpunan perawat

J : himpunan jenis *shift*

K : himpunan hari dalam periode penjadwalan

Keunggulan utama CP-SAT dalam konteks NRP:

- Mampu memformulasikan kendala kompleks dalam *bentuk pseudo-Boolean*.
- Mendukung *reified constraints*, yaitu kendala bersyarat yang sangat relevan untuk aturan rotasi *shift*.
- Memungkinkan integrasi fungsi objektif multi-kriteria melalui *weighted sum* atau *lexicographic optimization*.

- d. Skalabilitas cukup baik untuk penjadwalan menengah (20–100 perawat) dengan rentang penjadwalan mingguan atau bulanan.

Keterbatasan CP-SAT dalam konteks NRP:

- a. Sensitif terhadap skala masalah: waktu komputasi dapat meningkat signifikan untuk jumlah perawat besar (>150).
- b. Membutuhkan formulasi kendala yang ketat untuk mencapai kinerja optimal.
- c. Tidak selalu fleksibel terhadap perubahan mendadak seperti *real-time schedule adjustment* (Zanazzo dkk., 2024).

Contoh implementasi CP-SAT pada NRP antara lain digunakan untuk mengatur:

- 1. Maksimal *shift* malam berturut-turut.
- 2. Istirahat wajib setelah *shift* malam.
- 3. Pemerataan distribusi *shift* dan hari libur.
- 4. Keterisian minimal per *shift* (*team coverage*).

2.2.4 Optimasi *Mixed-Integer (Pseudo Boolean)* dalam CP-SAT

Mixed-Integer Programming (MIP) adalah pendekatan matematis yang digunakan untuk menyelesaikan permasalahan optimasi yang melibatkan variabel biner dan bilangan bulat (integer). Dalam konteks CP-SAT (Google OR-Tools), MIP dimanfaatkan untuk memformulasikan kendala penjadwalan secara lebih ketat menggunakan ekspresi linear dan *pseudo-Boolean*. Pendekatan ini memungkinkan representasi fleksibel dari aturan-aturan kompleks dalam *Nurse Rostering Problem* (NRP) sambil mempertahankan efisiensi komputasi (Perron & Feydy, 2020; Mystakidis dkk., 2024).

MIP sangat efektif untuk mengontrol *hard-constraint* seperti:

- 1. Batas maksimal *shift* malam berturut-turut,
- 2. Jumlah minimal staf per *shift*,
- 3. Waktu istirahat antar *shift*,
- 4. Batasan jam kerja mingguan,
- 5. dan keterisian tim minimal per *shift* (*team coverage*).

Pendekatan ini sering dipilih sebagai baseline *solver* dalam penelitian NRP karena hasilnya deterministik dan dapat mencapai solusi optimal jika waktu pencarian mencukupi (Zanazzo, dkk., 2024; Böðvarsdóttir & Stidsen, 2025). Dalam CP-SAT, MIP direpresentasikan menggunakan ekspresi *pseudo-Boolean*, yaitu kendala linear dengan variabel biner:

$$\sum_{i \in I} a_i x_i \leq b \quad (2.9)$$

Dengan:

x_i : variabel biner yang merepresentasikan keputusan pada elemen ke- i

a_i : koefisien bilangan bulat (integer) yang menunjukkan kontribusi variabel x_i terhadap sisi kiri kendala

i : indeks elemen/variabel, $i \in I$

I : himpunan indeks variabel biner yang terlibat dalam kendala

b : konstanta bilangan bulat (integer) yang menjadi batas maksimum nilai $\sum_{i \in I} a_i x_i$

2.2.5 *Tabu Search*

Tabu Search (TS) merupakan salah satu algoritma metaheuristik yang banyak digunakan untuk menyelesaikan permasalahan penjadwalan perawat yang tergolong dalam permasalahan kombinatorial berskala besar dan bersifat NP-hard. Prinsip utama TS adalah melakukan pencarian solusi lokal secara iteratif sambil menghindari *local optimum* melalui pemanfaatan struktur memori adaptif seperti *tabu list* dan *aspiration criteria*. Struktur memori ini menyimpan langkah atau solusi yang telah dikunjungi sebelumnya, sehingga pencarian dapat diarahkan menuju ruang solusi yang lebih luas dan mengurangi kemungkinan terjebak pada solusi lokal yang *sub-optimal* (Napalit & Ballera, 2023).

Dalam konteks NRP, TS diaplikasikan untuk menentukan penugasan *shift* perawat dengan mempertimbangkan *hard-constraints*, seperti jumlah maksimum hari kerja, urutan *shift* (misalnya *shift* malam tidak boleh langsung diikuti oleh *shift*

pagi), serta pemenuhan jumlah minimal perawat dalam setiap *shift*. Selain itu, TS juga memperhatikan *soft-constraints* seperti preferensi individu terhadap *shift* dan permintaan hari libur, yang berperan penting dalam meningkatkan kepuasan kerja tenaga keperawatan (Ramli dkk., 2020). Proses pencarian solusi dilakukan melalui eksplorasi *neighborhood*, seperti pertukaran atau pemindahan *shift* antar perawat untuk memperbaiki kualitas jadwal. Mekanisme *tabu tenure* kemudian mencegah langkah-langkah pencarian yang berulang, sehingga proses optimasi menjadi lebih efektif.

Secara matematis, tujuan algoritma *Tabu Search* pada permasalahan NRP dapat dinyatakan dalam fungsi objektif berikut:

$$\min f(S) = \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^d C_{ijk} x_{ijk} \quad (2.10)$$

Dengan:

$f(S)$: nilai fungsi objektif (total *penalty*) dari solusi jadwal S

x_{ijk} : variabel biner, bernilai 1 jika perawat ke- i dijadwalkan pada *shift* ke- j di hari ke- k , dan 0 jika tidak

C_{ijk} : nilai *penalty*/biaya penugasan perawat ke- i pada *shift* ke- j di hari ke- k

i : indeks perawat, $i = 1, \dots, n$

j : indeks *shift*, $j = 1, \dots, m$

k : indeks hari, $k = 1, \dots, d$

n : jumlah perawat

m : jumlah jenis *shift*

d : jumlah hari dalam periode penjadwalan

S : satu solusi jadwal (kombinasi lengkap penugasan *shift* untuk semua perawat dan hari)

Setiap iterasi TS menghasilkan solusi tetangga $S' \in N(S)$, yang dinilai berdasarkan fungsi *penalty*:

$$f(S') = f(S) + \Delta(S, S') \quad 2.11$$

Dengan:

$f(S')$: nilai fungsi objektif (total *penalty*) dari solusi tetangga S'

$f(S)$: nilai fungsi objektif (total *penalty*) dari solusi saat ini S

$\Delta_{S,S'}$: perubahan nilai fungsi objektif (delta *penalty*) akibat perpindahan (move) dari solusi S ke solusi tetangga S'

S : solusi jadwal saat ini

S' : salah satu solusi tetangga dari S yang dihasilkan melalui operasi move

Tahapan kerja algoritma *Tabu Search* dapat dijelaskan sebagai berikut:

1. Inisialisasi solusi awal (*initial solution*)

Proses dimulai dengan membangkitkan satu solusi awal yang layak, yaitu jadwal perawat yang memenuhi seluruh *hard-constraints*. Solusi ini bisa diperoleh secara acak atau menggunakan heuristik sederhana (misalnya penugasan berurutan berdasarkan kebutuhan *shift*).

2. Evaluasi solusi dan pembentukan *neighborhood*

Setiap solusi awal kemudian dievaluasi menggunakan fungsi objektif $f(S)$ yang menghitung total *penalty* dari pelanggaran *soft-constraints*. Dari solusi tersebut, dibangkitkan sekumpulan solusi tetangga (*neighborhood*) melalui operasi *move*, seperti menukar *shift* antar perawat atau menggeser hari kerja tertentu.

3. Pemilihan solusi terbaik dan pembaruan *tabu list*

Solusi terbaik dari himpunan tetangga yang tidak termasuk dalam *tabu list* dipilih sebagai kandidat baru. Langkah atau *move* yang baru saja dilakukan kemudian dimasukkan ke dalam *tabu list* untuk mencegah pengulangan solusi yang sama dalam beberapa iterasi berikutnya (*tabu tenure*).

4. Kriteria aspirasi (*aspiration criteria*)

Jika suatu solusi tabu menghasilkan nilai objektif yang lebih baik dibandingkan solusi terbaik sejauh ini ($f(S)' < f(S^*)$), maka larangan tabu tersebut dapat diabaikan dan solusi itu diterima.

5. Pembaruan solusi dan iterasi

Solusi terbaik sementara (S^*) diperbarui setiap kali ditemukan hasil yang lebih optimal. Proses ini diulang sampai mencapai jumlah iterasi maksimum atau ketika perbaikan tidak lagi signifikan.

Dengan mekanisme ini, *Tabu Search* mampu melakukan eksplorasi yang luas terhadap ruang solusi tanpa terjebak dalam *local optimum*. Penerapan struktur memori adaptif memungkinkan algoritma ini menyeimbangkan antara *intensification* (pendalaman pencarian di sekitar solusi baik) dan *diversification* (penjelajahan solusi baru) secara efisien (Napalit & Ballera, 2023; Abayomi-Alli dkk., 2022).

Hasil penelitian menunjukkan bahwa TS mampu menghasilkan solusi penjadwalan yang efisien dan fleksibel. Ramli dkk. (2020) mengembangkan pendekatan TS dengan peningkatan strategi *neighborhood generation* dan *intensification* sehingga mampu menghasilkan jadwal perawat yang lebih sesuai dengan regulasi kerja dan preferensi perawat. Abayomi-Alli dkk. (2022) menggabungkan TS dengan *Genetic Algorithm* dalam sistem GATS_NSS (*Genetic Algorithm + Tabu Search – Nurse Scheduling System*) dan berhasil meningkatkan akurasi penjadwalan serta mempercepat waktu komputasi secara signifikan. Selanjutnya, Napalit dan Ballera (2023) menerapkan TS secara *hybrid* dengan *Firefly Algorithm* untuk meningkatkan nilai *fitness* dan menurunkan waktu eksekusi dalam permasalahan penjadwalan rumah sakit.

Dengan kemampuannya dalam mengeksplorasi *local search* dan menghindari jebakan *local optimum*, *Tabu Search* menjadi metode yang sangat relevan untuk penelitian komparasi algoritma dalam menyelesaikan *Nurse Rostering Problem*, khususnya yang melibatkan kendala *shift* malam dan *team/skill coverage*.

2.2.6 Genetic Algorithm (GA)

Genetic Algorithm (GA) merupakan salah satu algoritma evolusioner yang banyak digunakan untuk menyelesaikan *Nurse Rostering Problem* (NRP) karena kemampuannya melakukan pencarian solusi optimal dalam ruang solusi yang kompleks. GA terinspirasi dari prinsip evolusi biologis Darwin, di mana proses seleksi, *crossover*, dan mutasi digunakan untuk menghasilkan populasi solusi baru yang semakin mendekati optimum global. Pada dasarnya, setiap solusi (jadwal perawat) direpresentasikan sebagai kromosom, sedangkan kumpulan solusi disebut populasi. Dalam setiap iterasi (*generation*), solusi terbaik dipertahankan sementara solusi lain dimodifikasi melalui operator genetik untuk meningkatkan kualitas penjadwalan (Amindoust dkk., 2021).

Dalam konteks penjadwalan perawat, GA efektif dalam menangani *hard-constraints*, seperti batasan jumlah maksimal hari kerja perawat, aturan urutan *shift*, serta pemenuhan jumlah perawat minimum per *shift*. Selain itu, GA juga mampu mengakomodasi kendala lunak, seperti preferensi perawat terhadap *shift* dan hari libur, yang dapat meningkatkan kepuasan kerja dan mengurangi *turnover* staf (Razali dkk., 2025). Penerapan GA memungkinkan proses eksplorasi ruang solusi yang lebih luas dibandingkan metode deterministik konvensional, sehingga dapat menghasilkan jadwal yang lebih fleksibel dan sesuai kebutuhan operasional rumah sakit.

Secara matematis, tujuan dari GA dalam *Nurse Rostering Problem* adalah meminimalkan pelanggaran terhadap *hard-constraints* sekaligus memaksimalkan kepuasan terhadap *soft-constraints*. Fungsi objektif yang umum digunakan dapat dirumuskan sebagai:

$$\min f(S) = \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^d (\alpha \cdot H_{ijk} + \beta \cdot S)_{ijk} \quad (2.12)$$

Dengan:

$f(S)$: nilai fungsi objektif (total *penalty*) dari solusi jadwal S

H_{ijk} : komponen *penalty*/pelanggaran **hard-constraint** untuk perawat ke- i , *shift* ke- j , hari ke- k

(misalnya pelanggaran batas maksimal hari kerja, pola urutan *shift*, dsb.)

S_{ijk} : komponen *penalty*/pelanggaran **soft-constraint** untuk perawat ke- i , *shift* ke- j , hari ke- k

(misalnya ketidaksesuaian dengan preferensi *shift* atau permintaan libur)

α : bobot kepentingan untuk komponen *hard-constraint* H_{ijk}

β : bobot kepentingan untuk komponen *soft-constraint* S_{ijk}

i : indeks perawat, $i = 1, \dots, n$

j : indeks *shift*, $j = 1, \dots, m$

k : indeks hari, $k = 1, \dots, d$

n : jumlah perawat dalam jadwal

m : jumlah jenis *shift*

d : jumlah hari dalam periode penjadwalan

S : satu solusi jadwal (kromosom) yang sedang dievaluasi

Pada optimasi berbasis *Genetic Algorithm*, proses seleksi umumnya membutuhkan ukuran kualitas yang semakin besar semakin baik (*maximize fitness*). Sementara itu, fungsi objektif $f(x)$ pada permasalahan penjadwalan biasanya berupa total penalti yang semakin kecil semakin baik (*minimize penalty*). Karena $f(x) \geq 0$, maka fungsi *fitness* dapat dibentuk melalui transformasi pada persamaan berikut:

$$Fitness(x) = \frac{1}{1 + f(x)} \quad (2.13)$$

Transformasi pada persamaan (2.13) menghasilkan fungsi yang monoton menurun terhadap *penalty*: jika *penalty* turun maka *fitness* naik. Konstanta 1 pada penyebut menjaga agar saat $f(x) = 0$ tidak terjadi pembagian nol dan sekaligus membuat *fitness* berada pada rentang (0,1). Sebagai ilustrasi, jika $f(x) =$

0 maka $Fitness(x) = 1$ (solusi tanpa *penalty*), sedangkan jika $f(x) = 9$ maka $Fitness(x) = \frac{1}{10} = 0.1$, sehingga solusi dengan *penalty* lebih kecil akan memiliki peluang seleksi lebih tinggi.

Dalam penelitian Razali dkk. (2025), bobot preferensi *shift* dihitung berdasarkan kombinasi antara ketersediaan ($A_{q,r,s,t}$) dan tingkat preferensi perawat ($P_{q,r,s,t}$) sehingga fungsi objektif menjadi:

$$\max Z = \sum_{q=1}^Q \sum_{r=1}^R \sum_{s=1}^S \sum_{t=1}^T W_{q,r,s,t} X_{q,r,s,t} \quad (2.14)$$

Dengan:

Z: nilai total bobot preferensi jadwal yang ingin dimaksimalkan

$W_{q,r,s,t}$: bobot preferensi penugasan untuk kombinasi indeks (q, r, s, t) , dipengaruhi oleh faktor α (reward bila preferensi dipenuhi) dan β (*penalty* bila tidak terpenuhi)

$X_{q,r,s,t}$: variabel biner yang bernilai 1 jika penugasan pada kombinasi (q, r, s, t) dipilih dalam jadwal, dan 0 jika tidak

q : indeks perawat, $q = 1, \dots, Q$

r : indeks jenis *shift*, $r = 1, \dots, R$

s : indeks hari atau periode waktu, $s = 1, \dots, S$

t : indeks dimensi tambahan (misalnya minggu, unit, atau tipe preferensi), $t = 1, \dots, T$

Q : jumlah total perawat

R : jumlah total jenis *shift*

S : jumlah total hari/periode yang dipertimbangkan

T : jumlah total kategori/dimensi preferensi yang dimodelkan

Tahapan utama algoritma GA dijelaskan sebagai berikut:

1. Inisialisasi populasi

Kumpulan jadwal awal dibangkitkan secara acak atau dengan heuristik sederhana yang memenuhi *hard-constraints*.

2. Evaluasi *fitness*

Setiap jadwal dihitung nilai *fitness*-nya berdasarkan fungsi *penalty* dan preferensi.

3. Seleksi (*selection*)

Jadwal dengan *fitness* tertinggi dipilih sebagai induk untuk proses reproduksi berikutnya, misalnya menggunakan metode *roulette wheel* atau *tournament selection*.

4. *Crossover* (reproduksi)

Dua kromosom induk dikombinasikan untuk menghasilkan kromosom baru (*offspring*) melalui pertukaran sebagian segmen *shift* antar perawat.

5. Mutasi

Beberapa gen (*shift* perawat) diubah secara acak untuk memperluas ruang pencarian dan mencegah *premature convergence*.

6. Evaluasi dan pembaruan populasi

Populasi baru menggantikan yang lama bila menghasilkan nilai *fitness* yang lebih baik.

Tahapan algoritma GA tersebut dilakukan berulang hingga mencapai generasi maksimum atau ketika tidak ada peningkatan signifikan terhadap nilai *fitness*. Pendekatan evolusioner ini memungkinkan GA menjelajahi ruang solusi yang luas dan menemukan kombinasi penjadwalan yang efisien dan adil (Harb dkk., 2024).

Penelitian Amindoust dkk. (2021) mengusulkan model *hybrid* GA yang mempertimbangkan faktor kelelahan perawat dalam penjadwalan. Model ini secara signifikan mengurangi beban kerja berlebih terutama pada *shift* malam, sehingga meningkatkan kualitas layanan dan mengurangi risiko kelelahan. Razali dkk. (2025) menggunakan GA untuk mengoptimasi preferensi *shift* kerja perawat dan menunjukkan peningkatan bobot preferensi jadwal dibandingkan penjadwalan manual. Sementara itu, Harb dkk. (2024) menggabungkan GA dengan pendekatan

cerdas berbasis *IoT (internet of things)* dan klasifikasi pasien, yang memungkinkan penjadwalan perawat berdasarkan tingkat keparahan pasien dan beban kerja, sehingga memberikan solusi penjadwalan yang lebih adaptif dan responsif.

Dengan fleksibilitasnya dalam menggabungkan berbagai batasan dan preferensi, GA menjadi salah satu pendekatan metaheuristik yang sangat relevan untuk penjadwalan perawat dengan kendala *shift* malam dan *team coverage*, terutama dalam lingkungan rumah sakit modern yang dinamis.

2.2.7 *Simulated Annealing (SA)*

Simulated Annealing (SA) merupakan algoritma metaheuristik berbasis proses fisika *annealing* yang digunakan untuk menemukan solusi optimal atau mendekati optimal pada permasalahan optimasi kombinatorial yang kompleks. Prinsip SA terinspirasi dari proses pendinginan logam yang memungkinkan sistem mencapai keadaan energi minimum melalui penurunan suhu secara bertahap. Dalam konteks optimasi, SA menggunakan probabilitas penerimaan solusi yang lebih buruk di awal proses pencarian untuk menghindari terjebak dalam *local optimum* dan secara perlahan mengurangi probabilitas tersebut seiring dengan penurunan “temperatur” (Ceschia dkk., 2023).

Pada permasalahan *Nurse Rostering Problem (NRP)*, SA diaplikasikan untuk menentukan penugasan *shift* perawat dengan memperhatikan berbagai kendala keras seperti jumlah perawat minimum per *shift*, batasan jam kerja, dan aturan urutan *shift* (termasuk *shift* malam dan waktu istirahat minimum), serta *soft-constraints* seperti preferensi perawat dan keadilan distribusi beban kerja. Pendekatan ini memungkinkan penjadwalan perawat yang fleksibel namun tetap memenuhi ketentuan regulasi dan operasional rumah sakit (Mukhlason dkk., 2024).

Secara matematis, prinsip dasar *Simulated Annealing* didasarkan pada fungsi energi $E(S)$ dari suatu solusi S , yang dianalogikan dengan tingkat energi sistem fisik. Algoritma bertujuan mencari keadaan dengan energi minimum (solusi optimal) melalui mekanisme penerimaan solusi baru S' dari tetangga $N(S)$ berdasarkan probabilitas:

$$P(\text{accept}) = 1 \text{ jika } E(S') < E(S); e^{-\frac{E(S')-E(S)}{T}} \text{ jika } E(S') \geq E(S) \quad (2.15)$$

Dengan:

$P(\text{accept})$: probabilitas penerimaan solusi baru S'

$E(S)$: nilai **energi** (total *penalty*) dari solusi saat ini S

$E(S')$: nilai **energi** dari solusi tetangga S'

T : **temperature parameter** pada iterasi saat ini, yang akan menurun secara bertahap sesuai cooling schedule

$e^{(\cdot)}$: fungsi eksponensial, basis bilangan $e \approx 2,718$

S : solusi jadwal saat ini

S' : solusi tetangga yang dihasilkan dari S melalui suatu move (misalnya tukar *shift*)

Dari persamaan diatas pada tahap awal, nilai T tinggi memungkinkan penerimaan solusi yang lebih buruk agar pencarian tidak terjebak di *local optimum*. Seiring waktu, nilai T menurun (*cooling schedule*) sehingga hanya solusi yang lebih baik yang akan diterima (Ceschia dkk., 2023).

Dalam *Nurse Rostering Problem*, fungsi energi dapat direpresentasikan sebagai total *penalty* terhadap pelanggaran batasan penjadwalan:

$$E(S) = \sum_{h=1}^H \lambda_h \cdot v_h(S) \quad (2.16)$$

Dengan:

$E(S)$: nilai energi/total *penalty* dari solusi jadwal S

$v_h(S)$: jumlah pelanggaran terhadap batasan ke- h pada solusi S

λ_h : bobot *penalty* untuk batasan ke- h , mencerminkan tingkat kepentingan/keparahan pelanggaran batasan tersebut

h : indeks batasan penjadwalan, $h = 1, \dots, H$

H : jumlah total jenis batasan yang dimasukkan dalam fungsi energi

S: satu solusi jadwal yang sedang dievaluasi

Tahapan kerja algoritma SA dapat dijelaskan sebagai berikut:

1. Inisialisasi solusi awal (*initialization*)
Membangkitkan jadwal awal yang memenuhi *hard-constraints*.
2. *Energy evaluation*
Menghitung nilai $E(S)$ sebagai dasar perbandingan.
3. *Generate neighbor solution*
Melakukan *move* seperti pertukaran *shift* antar perawat.
4. Penerimaan solusi baru (*acceptance criterion*)
Jika $E(S') < E(S)$, maka solusi diterima; jika lebih buruk, diterima dengan probabilitas $e^{-\Delta E/T}$.
5. *Cooling schedule*
menurunkan nilai T Secara bertahap menurut *schedule* tertentu, misalnya $T_{k+1} = \alpha T_k$, dengan $0 < \alpha < 1$
6. *Reheating* (opsional)
Jika konvergensi terlalu cepat, nilai suhu dinaikkan kembali untuk memperluas eksplorasi solusi (Mukhlason dkk., 2024). Mekanisme *reheating* ini terbukti meningkatkan kemampuan eksplorasi algoritma dan menghindari *stagnation*, sementara pendekatan *multi-neighborhood* seperti yang diterapkan oleh Zanazzo dkk. (2025) menambah fleksibilitas pencarian dalam ruang solusi kompleks.

Ceschia dkk. (2023) mengembangkan model SA dengan *local search* berbasis dua *neighborhood* untuk menyelesaikan permasalahan penjadwalan nyata di berbagai rumah sakit Italia, termasuk pengaturan *shift* malam dan preferensi hari libur, dan terbukti memberikan hasil solusi berkualitas tinggi dalam waktu komputasi yang efisien. Mukhlason dkk. (2024) mengusulkan pendekatan *Simulated Annealing with Reheating* berbasis reinforcement learning yang mampu meningkatkan kualitas solusi hingga 82% dibandingkan metode pembandingan konvensional. Selain itu, Zanazzo dkk. (2025) menerapkan *multi-neighborhood* SA

pada permasalahan integrasi penugasan perawat–pasien, yang melibatkan kendala *team coverage* dan beban kerja *shift* malam, dan menunjukkan performa yang unggul pada skala besar.

Dengan kemampuannya dalam menghindari jebakan solusi lokal dan menyeimbangkan eksplorasi serta eksploitasi, SA menjadi metode metaheuristik yang sangat relevan untuk penelitian komparatif penyelesaian NRP, khususnya yang melibatkan kendala *shift* malam dan kebutuhan pemerataan beban (*team/skill coverage*).

2.2.8 Hybrid GA-SA

Hybrid Genetic Algorithm – Simulated Annealing (HGA–SA) merupakan pendekatan metaheuristik gabungan yang mengombinasikan kekuatan eksplorasi dari *Genetic Algorithm* (GA) dan kemampuan eksploitasi serta penghindaran local optimum dari *Simulated Annealing* (SA). GA melakukan pencarian global melalui mekanisme seleksi, *crossover*, dan mutasi, sedangkan SA digunakan untuk melakukan *local search* dengan *probabilistic acceptance criterion* yang memungkinkan penerimaan solusi kurang baik pada tahap awal pencarian untuk mencegah konvergensi prematur. Pendekatan *hybrid* ini bertujuan menyeimbangkan eksplorasi dan eksploitasi dalam ruang solusi (Amindoust dkk., 2021).

HGA–SA pada NRP, dapat mengoptimalkan penugasan *shift* dengan tetap memenuhi *hard-constraints* seperti jumlah minimal perawat per *shift*, batas urutan *shift* (misalnya larangan *shift* pagi setelah *shift* malam), batas maksimum jam kerja, dan pengaturan hari libur. *Soft-constraints* seperti preferensi individu perawat juga dapat dimasukkan sebagai *penalty* dalam fungsi objektif. HGA–SA dirancang agar proses pencarian solusi tidak hanya berfokus pada eksplorasi global, tetapi juga memperbaiki solusi lokal secara adaptif. Proses GA menghasilkan populasi solusi awal melalui mekanisme seleksi dan reproduksi, sedangkan SA berperan melakukan *local refinement* terhadap individu terbaik dengan mempertimbangkan probabilitas penerimaan solusi yang lebih buruk untuk mencegah konvergensi prematur (Turhan & Bilgen, 2020).

Secara matematis, fungsi objektif HGA-SA dapat dinyatakan sebagai kombinasi dua komponen optimasi:

$$\min F(S) = \alpha \cdot f_{GA}(S) + (1 - \alpha) \cdot f_{SA}(S) \quad (2.17)$$

Dengan:

$F(S)$: nilai fungsi objektif gabungan HGA-SA untuk solusi jadwal S

$f_{GA}(S)$: nilai fungsi evaluasi berbasis GA (total *penalty* hard + kendala lunak untuk solusi S)

$f_{SA}(S)$: nilai fungsi evaluasi berbasis SA (komponen energi/*penalty* yang dikendalikan oleh skema temperatur)

α : parameter pengontrol keseimbangan kontribusi GA dan SA, dengan $\alpha \in [0,1]$

$1 - \alpha$: bobot pelengkap untuk komponen SA

S : satu solusi jadwal (kromosom) yang sedang dievaluasi

Dalam implementasinya, SA dijalankan setelah setiap siklus GA (generasi) selesai, untuk memperbaiki individu terbaik (*best chromosome*) menggunakan *neighborhood search* lokal. Mekanisme penerimaan solusi mengikuti aturan probabilitas:

$$P(\text{accept}) = e^{\frac{E(S') - E(S)}{T}} \quad (2.18)$$

Dengan:

$P(\text{accept})$: probabilitas penerimaan solusi baru S' dalam tahap SA

$E(S)$: nilai energi/total *penalty* dari solusi saat ini S

$E(S')$: nilai energi/total *penalty* dari solusi tetangga S'

T : temperatur pada iterasi saat ini, yang menurun secara bertahap sesuai cooling schedule

$e^{(\cdot)}$: fungsi eksponensial

S : solusi jadwal sebelum move (solusi saat ini)

S' : solusi tetangga hasil move dari S

Tahapan kerja algoritma HGA-SA secara umum meliputi:

1. Inisialisasi populasi dengan jadwal awal yang memenuhi *hard-constraints*.
2. Evaluasi *fitness* tiap individu menggunakan fungsi *penalty* $f_{GA}(S)$
3. Seleksi, *crossover*, dan mutasi untuk membentuk populasi baru.
4. Penerapan SA lokal pada individu terbaik, di mana solusi diperbaiki menggunakan *neighborhood moves* dengan probabilitas penerimaan $P(accept)$.
5. Pembaruan suhu (*cooling*) secara adaptif hingga suhu minimum tercapai.
6. Iterasi berulang hingga jumlah generasi maksimum atau konvergensi.

Amindoust dkk. (2021) menunjukkan bahwa penggunaan *Hybrid GA* dengan struktur kromosom adaptif dapat meningkatkan kualitas solusi dan mempercepat konvergensi pada penjadwalan perawat yang melibatkan *shift* malam. Turhan & Bilgen (2020) menggunakan SA untuk memperbaiki solusi awal berbasis *fix-and-optimize*, sehingga meminimalkan stagnasi pencarian. Selain itu, pendekatan *hybrid* juga sejalan dengan konsep penggabungan eksplorasi GA dan eksploitasi SA pada optimasi kombinatorial berskala kompleks (Mahmoudinazlou & Kwon, 2024).

Dengan demikian, HGA-SA menjadi pendekatan yang relevan untuk penelitian NRP yang melibatkan kendala *shift* malam dan *team coverage*, karena mampu menghasilkan solusi yang berkualitas tinggi dengan efisiensi komputasi yang lebih baik dibandingkan penggunaan GA atau SA secara Tunggal

2.2.9 Pengujian Algoritma

Pengujian algoritma optimasi pada permasalahan *Nurse Rostering Problem* (NRP) umumnya dilakukan melalui pendekatan eksperimen komputasi yaitu metode ilmiah yang digunakan secara luas dalam penelitian optimasi dan kecerdasan komputasional. Eksperimen komputasi dirancang untuk mengevaluasi

performa algoritma berdasarkan metrik terukur, variasi skenario, serta prosedur evaluasi yang dapat direplikasi (Angriman dkk., 2019; Hoos & Stützle, 2005).

Dalam penelitian optimasi kombinatorial seperti NRP, eksperimen komputasi dipandang sebagai metode standar karena solusi tidak dapat ditentukan secara analitik dan harus dibandingkan melalui pengujian empiris (Ceschia dkk., 2023; Ramli dkk., 2020).

Sejalan dengan perkembangan penelitian optimasi di bidang *Nurse Rostering Problem*, literatur menyajikan beberapa aspek utama yang perlu diperhatikan dalam pengujian algoritma. Secara umum, kajian-kajian terdahulu mengelompokkan pembahasan pengujian ke dalam beberapa komponen, meliputi perancangan eksperimen komputasi, pemilihan metrik evaluasi kinerja, analisis sensitivitas dan *robustness*, validasi solusi yang dihasilkan, serta analisis statistik untuk menilai signifikansi perbedaan kinerja antar algoritma. Kelima aspek tersebut dapat dirangkum sebagai berikut.

1. Eksperimen Komputasi

Angriman dkk. (2019) dan Hoos & Stützle (2005) menekankan bahwa eksperimen algoritma perlu dirancang secara sistematis dengan memperhatikan beberapa prinsip umum, antara lain:

- a. *Reproducibility*: eksperimen harus dapat diulang pada konfigurasi yang identik sehingga peneliti lain dapat memverifikasi hasil.
- b. *Controlled variation*: perubahan parameter (misalnya ukuran *instance* atau nilai *hyperparameter*) harus terukur dan terdokumentasi secara jelas.
- c. *Fairness*: algoritma yang dibandingkan dijalankan pada himpunan *instance* dan kondisi input yang sama untuk menjamin perbandingan yang adil.
- d. *Quantitative measurement*: keluaran algoritma harus dievaluasi dengan metrik numerik yang terdefinisi dengan baik.

Selain itu, pada algoritma stokastik (misalnya *Tabu Search* dan algoritma hibrida), pengulangan eksperimen pada *instance* yang sama merupakan proses baku untuk mengurangi bias akibat sifat acak dan memperoleh estimasi yang lebih stabil terhadap performa rata-rata algoritma (Hoos & Stützle, 2005).

2. Evaluasi Kinerja Algoritma

Evaluasi algoritma optimasi dilakukan menggunakan serangkaian metrik yang relevan dengan struktur masalah. Dalam konteks NRP, literatur menyebutkan beberapa kategori metrik standar (Ceschia dkk., 2023; den Hartog dkk., 2023), antara lain:

a. *Feasibility Rate*

Mengukur kemampuan algoritma menghasilkan jadwal yang memenuhi seluruh kendala keras. Tingkat kelayakan ini merupakan indikator dasar untuk menilai apakah algoritma mampu menghasilkan solusi yang dapat diterapkan.

b. *Constraint Violation / Penalty*

Banyak studi NRP menggunakan jumlah pelanggaran atau nilai *penalty* untuk menilai kualitas solusi terhadap kendala lunak, misalnya preferensi hari libur dan pemerataan *shift* malam (Zanazzo dkk., 2025; Guo & Bard, 2022). Semakin besar pelanggaran, semakin tinggi nilai *penalty* sehingga kualitas jadwal dinilai semakin rendah.

c. *Workload Fairness / Variance*

Metrik pemerataan, seperti varians atau koefisien variasi jam kerja, digunakan untuk menilai distribusi beban kerja antar perawat, sehingga tidak terjadi ketimpangan yang ekstrem dalam jumlah jam kerja atau jumlah *shift* malam (Alaouchiche dkk., 2024).

d. *Fulfillment Rate / Coverage Satisfaction*

Metrik ini menilai tingkat pemenuhan kebutuhan staf per *shift*, yang menjadi indikator penting dalam menjamin kecukupan tenaga pada setiap *shift* pelayanan (Turhan & Bilgen, 2020).

e. *Computation Time*

Waktu komputasi merupakan metrik utama dalam menilai efisiensi algoritma, khususnya untuk *solver* seperti CP-SAT. Metrik ini digunakan untuk mengukur berapa lama algoritma membutuhkan waktu hingga

menemukan solusi *feasible* atau hingga mencapai batas waktu yang ditentukan (Perron dkk., 2023).

f. Nilai fungsi objektif komposit

Pada banyak formulasi NRP, kualitas solusi dirangkum dalam sebuah fungsi objektif yang menggabungkan berbagai komponen *penalty*. Nilai fungsi objektif yang lebih kecil (atau akurasi komposit yang lebih tinggi) menunjukkan jadwal yang lebih mendekati konfigurasi ideal menurut aturan dan preferensi yang dimodelkan.

3. Uji Sensitivitas dan *Robustness*

Uji sensitivitas merupakan prosedur penting dalam penelitian optimasi untuk menilai bagaimana perubahan parameter memengaruhi kualitas solusi algoritma. Hoos & Stützle (2005) serta Angriman dkk. (2019) menjelaskan bahwa analisis sensitivitas dan *robustness* digunakan untuk:

- a. menilai stabilitas algoritma terhadap variasi kondisi masalah,
- b. mengevaluasi ketahanan performa terhadap perubahan konfigurasi eksperimen, dan
- c. mengidentifikasi rentang pengaturan yang memberikan keseimbangan terbaik antara kualitas solusi dan biaya komputasi.

Dalam konteks *Nurse Rostering Problem*, studi-studi terdahulu melakukan analisis sensitivitas dengan berbagai cara, antara lain dengan:

- a. mengubah ukuran *instance* atau skala skenario, misalnya dengan memperbesar jumlah perawat atau panjang horizon penjadwalan untuk melihat bagaimana algoritma berperilaku pada masalah yang semakin kompleks (Ceschia dkk., 2023; Ramli dkk., 2020),
- b. memvariasikan batas waktu komputasi (*time limit*) kemudian mengamati perbaikan kualitas solusi serta gejala berkurangnya tambahan peningkatan kualitas seiring penambahan waktu (Hoos & Stützle, 2005; Angriman dkk., 2019), dan

- c. menyesuaikan parameter algoritmik seperti ukuran populasi, probabilitas mutasi, atau parameter struktur tabu, terutama pada metaheuristik (Harb dkk., 2024; Turhan & Bilgen, 2020).

Melalui variasi-variasi tersebut, penelitian NRP tidak hanya menilai performa algoritma pada satu konfigurasi tetap, tetapi juga menilai *robustness* algoritma ketika skala masalah membesar dan ketika batas waktu komputasi diubah. Pendekatan ini sejalan dengan praktik umum dalam eksperimen komputasi untuk optimasi kombinatorial, di mana kinerja algoritma dievaluasi pada beberapa tingkat kesulitan *instance* dan beberapa level sumber daya komputasi yang tersedia (Hoos & Stützle, 2005; Angriman dkk., 2019).

4. Validasi Solusi

Validasi solusi bertujuan memastikan bahwa jadwal yang dihasilkan tidak hanya baik secara numerik, tetapi juga layak diterapkan secara operasional. Secara umum, validasi dapat dibedakan menjadi dua jenis (Amindoust dkk., 2021):

a. Validasi Teknis

Validasi teknis dilakukan dengan memverifikasi bahwa solusi:

- 1) tidak melanggar kendala keras,
- 2) mematuhi aturan struktur *shift* (misalnya pola *shift* malam dan hari libur),
- 3) memenuhi kebutuhan minimal per *shift*, dan
- 4) konsisten serta bebas dari konflik penugasan.

Dalam banyak penelitian, proses ini dilakukan melalui *built-in constraint checker* yang secara sistematis memeriksa setiap solusi yang dihasilkan algoritma (Perron dkk., 2023; Guo & Bard, 2022).

b. Validasi Praktis

Di samping validasi teknis, penelitian NRP modern merekomendasikan keterlibatan pemangku kepentingan klinis seperti kepala ruangan, supervisor keperawatan, atau manajer unit untuk menilai kelayakan jadwal dari sisi implementasi nyata. Validasi ini mencakup aspek kenyamanan

kerja, kesinambungan asuhan, kebijakan internal, dan konteks organisasi (O'Connell dkk., 2024).

5. Analisis Statistik Hasil Eksperimen

Analisis statistik digunakan untuk menilai apakah perbedaan kinerja antar algoritma bersifat signifikan secara statistik atau sekadar akibat variasi acak.

Studi komparatif algoritma umumnya menggunakan:

- a. analisis deskriptif, seperti nilai rata-rata dan standar deviasi kualitas solusi
- b. visualisasi, seperti *boxplot* dan kurva *trade-off* antara kualitas dan waktu komputasi (Ceschia dkk., 2023), serta
- c. uji statistik inferensial (misalnya ANOVA atau uji non-parametrik) yang digunakan untuk memastikan apakah perbedaan kinerja beberapa algoritma pada banyak *instance* benar-benar signifikan atau hanya terjadi karena kebetulan/variasi acak (Demšar, 2006; Angriman dkk., 2019).
- d. uji lanjut (*post-hoc*) seperti *Tukey Honestly Significant Difference* (*Tukey HSD*) untuk mengidentifikasi pasangan algoritma mana yang berbeda signifikan satu sama lain setelah ANOVA menunjukkan adanya perbedaan rata-rata kinerja yang signifikan. Uji *post-hoc* ini penting untuk mengendalikan risiko kesimpulan yang keliru, yaitu ketika analisis menyatakan ada perbedaan kinerja antar algoritma padahal sebenarnya tidak ada, terutama saat dilakukan banyak perbandingan berpasangan (Demšar, 2006).