

Pengembangan Algoritma *Hybrid* Metaheuristik Untuk Menyelesaikan Permasalahan Penjadwalan Perawatan Pesawat

Asyraf Nur Adianto¹, Hasan Aji Prawira², Nurhadi Siswanto³, Ahmad Rusdiansyah⁴

Departemen Teknik Industri, Institut Teknologi Sepuluh Nopember
Jl. ITS Raya Surabaya, Jawa Timur, Indonesia
siswanto@ie.its.ac.id

INFO ARTIKEL

doi: 10.350587/Matrik
v18i2.589

Jejak Artikel :

Upload artikel
15 November 2019
Revisi oleh reviewer
21 Maret 2020
Publish
30 Maret 2020

Kata Kunci :

Perawatan Pesawat,
Penjadwalan perawatan,
Particle Swarm Optimization,
Crow Search Algorithm,
Greedy Randomized Adaptive
Search Procedures

ABSTRAK

Aircraft Maintenance Problem (AMP) merupakan permasalahan penentuan jadwal kegiatan perawatan pesawat. AMP memiliki dua jenis kegiatan perawatan yang akan diteliti yaitu inspeksi dan *continuous airworthiness maintenance programs* (CAMP). Penelitian ini membandingkan kinerja antara metode *Particle Swarm Optimization* (PSO) dengan metode *Crow Search Algorithm* (CSA). Kedua metode tersebut dihibridisasikan dengan *Greedy Randomized Adaptive Search Procedures* (GRASP) untuk menyelesaikan AMP. Penelitian ini memiliki tujuan yaitu untuk menentukan jumlah periode yang diperlukan untuk perawatan pesawat dan menentukan tugas atau jenis inspeksi dan CAMP yang harus dilakukan dalam setiap periode serta menentukan metode yang ideal untuk menyelesaikan AMP. Permasalahan AMP sendiri merupakan permasalahan kombinatorial yang dapat dikategorikan sebagai permasalahan NP-Hard. Metode metaheuristik digunakan untuk memastikan proses optimasi dapat diselesaikan dengan waktu yang singkat. Percobaan dilakukan menggunakan 16 kondisi dengan empat dataset yang dihasilkan secara acak. Hasil percobaan komputasi menunjukkan bahwa PSO-GRASP menghasilkan solusi yang lebih baik dari CSA-GRASP untuk jumlah inspeksi yang lebih tinggi.

1. Pendahuluan

Jumlah pendapatan dari industri penerbangan berdasarkan pada jumlah penumpang yang mereka layani di setiap operasional pesawat. Oleh karena itu, fokus utama setiap industri penerbangan adalah memaksimalkan jam operasional pesawat untuk memaksimalkan jumlah penumpang yang dilayani. Namun, industri penerbangan tidak bisa memaksakan pesawatnya untuk melakukan penerbangan di setiap waktu [1]. Setiap pesawat yang disewa oleh maskapai memiliki kondisi minimum masing-masing sebelum pesawat tersebut dapat dikembalikan. Kondisi minimum tersebut dapat dicapai dengan melakukan perawatan rutin pada setiap unit pesawat. Waktu dan biaya perawatan yang dibutuhkan satu unit pesawat di setiap perawatan dapat berbeda setiap waktunya. Ketika sebuah pesawat dirawat, pesawat tidak dapat dioperasikan untuk memperoleh *revenue* bagi pemiliknya. Semua maskapai berharap bahwa pesawat yang digunakan memiliki utilisasi tinggi untuk melayani pelanggan dengan meminimalkan jumlah perawatan tanpa melanggar peraturan yang terkait dengan kelaikan pesawat. Penelitian yang terkait dengan masalah-masalah ini disebut *aircraft maintenance / aeronautical maintenance* [1].

Bedasarkan literatur penelitian yang terkait dengan *aeronautical / aircraft maintenance* permasalahan ini disebut dengan *Aircraft Maintenance Routing Problem* (AMRP). Beberapa penelitian telah dilakukan di bidang AMRP, seperti: Gopalan dan Talluri [2], Liang, Chaovalitwongse, Huang, dan Johnson [3], Ezzinbi, Sarhani, El Afia, dan Benadada [4], Al-Thani, Ben Ahmed, dan Haouari [5], Eltoukhy, Chan, Chung, dan Niu [6], dan Safaei dan Jardine [7]. Penelitian-penelitian tersebut hanya mempertimbangkan penjadwalan *inspection check* yaitu A-Check, C-Check, dan D-Check. Dalam praktiknya, ada kegiatan perawatan lain yang perlu dipertimbangkan oleh maskapai, seperti *Continuous Airworthiness Maintenance Program* (CAMP). Perbedaan antara *inspection check* dan CAMP yaitu *inspection*

check melakukan pengecekan visual pada beberapa komponen pesawat dan menentukan keputusan perawatan atau penggantian komponen [8]. Kegiatan perawatan ini dikategorikan sebagai perawatan tidak terjadwal. Sedangkan CAMP melakukan perawatan-perawatan kecil berdasarkan arahan perusahaan produsen pesawat dan harus dilakukan secara teratur. Kegiatan perawatan ini dikategorikan sebagai perawatan terjadwal [9]. Penggabungan aktivitas *inspection check* dan CAMP ini menambah kesulitan dalam menentukan jadwal perawatan.

Setiap Maskapai penerbangan memiliki keinginan untuk memaksimalkan utilitas dari pesawat yang sudah mereka sewa atau beli. Utilitas pesawat dapat ditingkatkan apabila waktu perawatan dan inspeksi pesawat dapat ditekan namun tetap mengikuti standar yang sudah ditentukan.

Banyaknya aktivitas perawatan serta *plan horizon* yang panjang mengakibatkan penjadwalan urutan aktivitas menjadi sangat kompleks. Banyaknya kombinasi untuk urutan aktivitas inspeksi dan perawatan rutin membuat permasalahan ini dapat dikategorikan sebagai *Non Polynomial-Hard Problem* (NP-Hard). Untuk mengatasi masalah ini, metode metaheuristik seperti *Particle Swarm Optimization* (PSO) atau *Crow Search Algorithm* (CSA) digunakan untuk mencari solusi optimal. PSO dan CSA sendiri telah banyak memiliki pengalaman dalam menyelesaikan permasalahan penjadwalan perawatan. Dalam penelitian ini peneliti ingin menguji PSO dan CSA yang dihibridisasi dengan algoritma *Greedy Randomized Adaptive Search Procedures* (GRASP) untuk menemukan solusi permasalahan ini. Algoritma GRASP merupakan algoritma metaheuristik yang didesain untuk membantu algoritma lain untuk mengefisienkan pencarian solusi optimal. Pada penelitian ini akan dibandingkan algoritma PSO-GRASP dan CSA-GRASP. Perbandingan algoritma dilakukan dengan melihat solusi yang dihasilkan dan waktu komputasi yang dibutuhkan untuk menyelesaikan permasalahan.

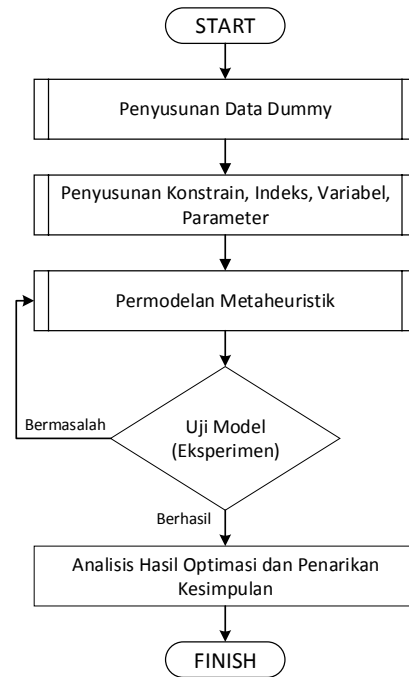
2. Metode Penelitian

Penelitian ini dilakukan dengan beberapa tahapan secara berurutan. Gambar 1 menunjukkan tahapan-tahapan penelitian yang dilakukan.

Data sampel didapatkan dengan cara melakukan diskusi langsung dengan salah satu pakar perawatan pesawat pada Perusahaan Maskapai X. Berdasarkan hasil diskusi tersebut didapatkan sebanyak 14 aktivitas inspeksi dan 294 aktivitas perawatan rutin. Data tersebut nantinya akan dikembangkan lebih lanjut sebagai data eksperimen dalam penelitian ini.

Konstrain dari model konseptual permasalahan dikonversi menjadi notasi matematika ataupun *pseudocode* dengan cara mengubah seluruh data menjadi variabel ataupun parameter, serta beberapa entitas yang terlibat.

Pengembangan model metaheuristik dilakukan dengan cara merancang ulang pola pikir yang digunakan dalam mengembangkan formulasi matematika dalam penyelesaian masalah. Model metaheuristik mengharuskan proses optimasi dapat menyelesaikan permasalahan beserta seluruh konstrain yang dimiliki dengan kriteria terminasi tertentu, seperti jumlah iterasi pencarian solusi, rentang nilai yang diperoleh, maupun kesalahan dalam proses perhitungan. Pada tahap ini akan dilakukan pengembangan beberapa modul. Modul pertama yang akan dikembangkan adalah modul inisiasi atau *input* data. Modul ini berfungsi untuk membantu pengguna model untuk dapat dengan mudah memasukkan segala data yang akan dipertimbangkan seperti data *dummy inspection check* dan data *dummy routine maintenance*. Selanjutnya adalah pengembangan modul inisiasi atau input setiap parameter. Parameter yang dipertimbangkan pada model optimasi ini terdiri dari dua bagian, diantaranya yaitu parameter dari kasus dan parameter dari metode metaheuristik. Parameter dari kasus adalah parameter yang diperoleh dengan membatasi ruang lingkup penyelesaian kasus.



Gambar 1. Gambaran Umum Alur Penelitian

Eksperimen dilakukan dengan menguji model perhitungan menggunakan data yang sudah disusun dengan menggunakan setiap grup data set. Eksperimen masing-masing dilakukan sebanyak sepuluh pengulangan / repetisi untuk setiap grup data set dan setiap metode metaheuristik. Hasil perhitungan tersebut kemudian akan dianalisa lebih lanjut pada bagian berikutnya.

Analisa yang dilakukan berfokus pada dua hasil. Hasil pertama ialah untuk membandingkan performansi metode metaheuristik dalam menyelesaikan ukuran data set yang berbeda. Hasil kedua ialah untuk melakukan uji sensitivitas dari beberapa kombinasi parameter metaheuristik.

Respon yang akan diperhatikan untuk melakukan analisa adalah dengan menilai waktu penyelesaian permasalahan serta nilai fungsi tujuan yang diperoleh pada satu kombinasi parameter yang sama untuk setiap metode metaheuristik. Kedua nilai tersebut dicari nilai mean dan standard deviation dari seluruh repetisi yang dilakukan serta untuk setiap grup data set dan metode metaheuristik yang digunakan. Sebuah metode metaheuristik dapat dikatakan memiliki kemampuan yang lebih baik dalam menyelesaikan kasus

dibandingkan dengan metode lain apabila nilai mean yang diperoleh lebih rendah dari metode lainnya. Sebuah metode metaheuristik dikatakan memiliki performa yang konsisten apabila standard deviation dari keseluruhan repetisi yang diberikan memiliki nilai lebih kecil dari metode metaheuristik lainnya.

A. Deskripsi Sistem

1) Fasilitas Perawatan Pesawat di Indonesia

Di Indonesia hanya terdapat satu lokasi yang dapat dijadikan tempat perawatan pesawat secara besar. Lokasi perawatan tersebut terdapat di daerah Bandara Internasional Soekarno Hatta, Cengkareng, Jakarta. Setiap pesawat dari Indonesia yang akan melakukan ataupun membutuhkan perawatan yang dimiliki harus membawa pesawat tersebut pada lokasi tersebut, walaupun wilayah operasional dari pesawat tersebut berada di wilayah terjauh dari Indonesia seperti di Kota Merauke, atau bahkan pesawat yang wilayah operasionalnya berada di luar Indonesia.

Fasilitas perawatan ini terdiri dari empat hanggar dengan kapasitas dan peralatan yang berbeda-beda. Tidak ada satu hanggar pun yang mampu menyelesaikan seluruh kebutuhan perawatan yang diminta oleh pihak maskapai. Apabila fasilitas perawatan pada hanggar tersebut sedang digunakan untuk merawat suatu pesawat, maka pesawat lainnya harus memasuki antrian terlebih dahulu. Sehingga apabila waktu perawatan seharusnya hanya membutuhkan waktu satu hari, namun apabila pesawat tersebut berada pada antrian yang panjang maka ada kemungkinan waktu perawatan yang dibutuhkan lebih dari satu hari.

2) Pembiayaan Perawatan Pesawat

Aktivitas perawatan pesawat merupakan aktivitas yang mengeluarkan dana. Jumlah dana yang dikeluarkan dapat bervariasi bergantung dari perawatan yang dibutuhkan serta cara pembayaran yang disepakati antara maskapai dengan pemilik fasilitas perawatan. Aktivitas yang membutuhkan tenaga kerja dalam jumlah yang banyak, atau yang

mebutuhkan tenaga kerja dengan sertifikasi tertentu akan membuat biaya yang perlu dibayarkan menjadi besar. Selain tenaga kerja yang dibutuhkan, apabila terdapat kerusakan komponen dan komponen tersebut memiliki fungsi yang utama dalam beroperasinya pesawat maka akan membuat biaya menjadi besar. Untuk dapat membayar seluruh biaya yang dibutuhkan maskapai dapat melakukannya dalam dua cara, pertama adalah membayar seluruhnya sesuai dengan kebutuhan perawatan pesawat di saat itu, dan kedua adalah dengan melakukan kontrak kerja sama dalam rentang waktu tertentu dan tidak melihat perawatan apa saja yang dibutuhkan pesawat di setiap waktunya.

3) Sumber Data Optimasi

Data yang dipertimbangkan untuk dapat membuat penjadwalan perawatan dan inspeksi pesawat terdiri dari beberapa data. Data-data tersebut di antaranya yaitu *Inspection Check*, *Continuous Airworthiness Maintenance Program*, dan *Last Done Next Do* (LDND).

Inspeksi pada pesawat merupakan hal yang harus dilakukan oleh setiap pemilik pesawat. Hal ini dikarenakan saat pesawat melakukan kegiatan operasional di udara sering kali menghadapi kondisi yang beragam dan tidak selalu sama, baik suhu, tekanan udara, serta berbagai kondisi geografis lainnya. Kondisi yang tidak pernah sama membuat aktivitas perawatan tidak dapat dilakukan secara berkala pada beberapa komponen yang terdampak tersebut dan hanya dapat dilakukan pesawat saat kondisi visual [8] dari komponen tersebut sudah masuk dalam kondisi 'harus diperbaiki'. Untuk dapat melakukan perawatan sebelum komponen pesawat mengalami kerusakan atau dikenal dengan *preventive maintenance*, maka dibuat jadwal atau interval inspeksi yang harus dilakukan oleh pemilik pesawat tersebut. Rangkaian interval inspeksi dibuat dan disetujui oleh beberapa pihak di antaranya yaitu dari *Federal Aviation Administration* (FAA), produsen pesawat, dan maskapai penerbangan yang menggunakan pesawat terkait. Perusahaan manufaktur pesawat membentuk interval inspeksi dalam bentuk

flight hours (FH), *flight cycle* (FC) maupun berdasarkan perhitungan kalender seperti satu tahun, dua tahun, dan seterusnya. Interval inspeksi yang diperintahkan langsung oleh FAA yang harus dilakukan secara rutin oleh maskapai yaitu setiap 100 *flight hours* (FH) atau setiap 1 tahun operasional pesawat [9]. Namun selain inspeksi rutin tersebut terdapat inspeksi lainnya menyesuaikan tingkat kesulitan inspeksi yang harus dilakukan oleh maskapai. Inspeksi tersebut di rancang menyesuaikan jenis dan kebutuhan dari pesawat yang dimiliki.

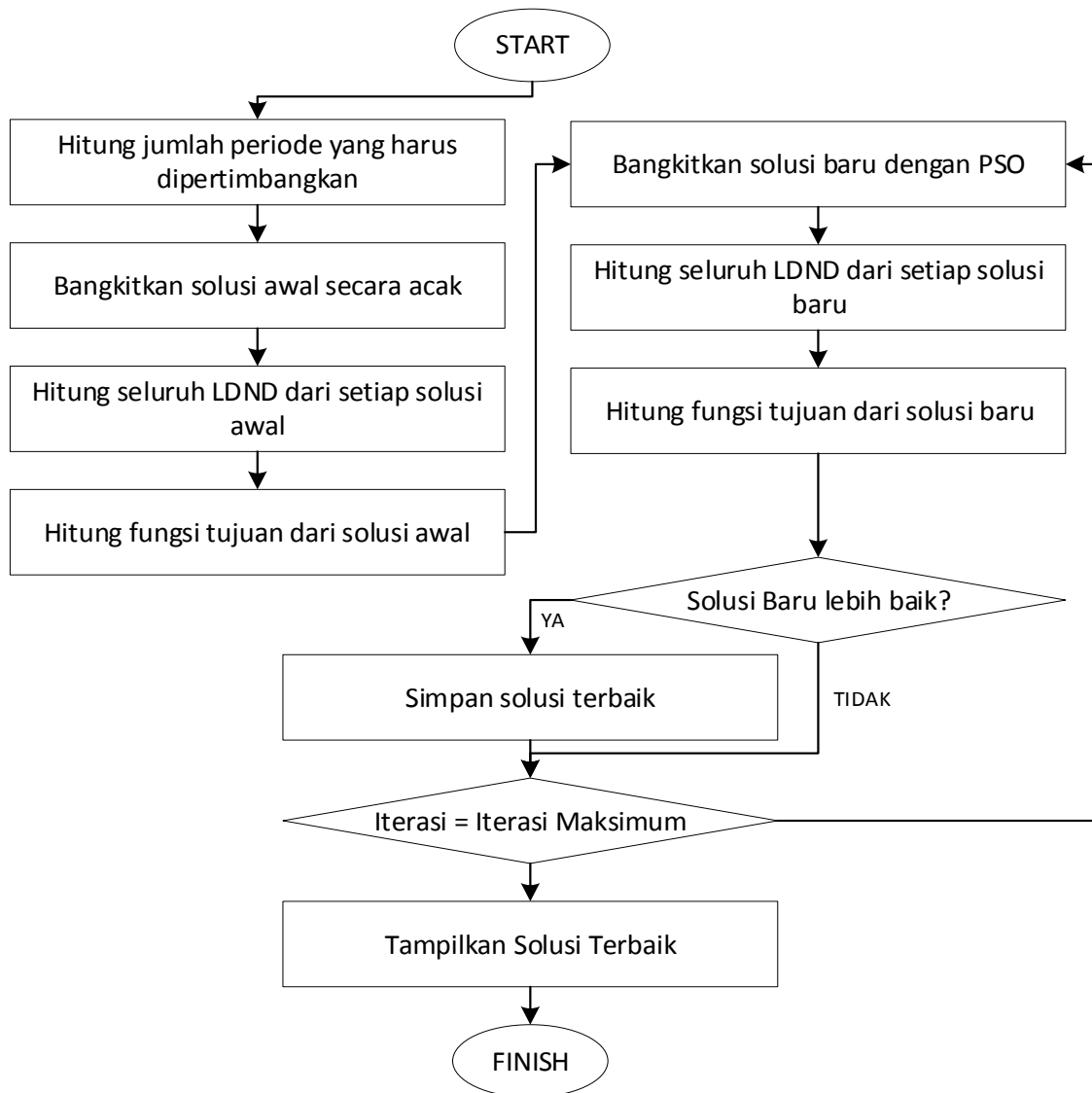
CAMP atau dikenal juga sebagai *Maintenance Program* ataupun *Continuous Airworthiness Program* dibuat oleh produsen dari pesawat yang berkaitan dan dibutuhkan persetujuan oleh *Air Carrier Operations Specifications* (OpSpecs) serta pembuat kebijakan tempat beroperasi pesawat tersebut. CAMP merupakan susunan aktivitas yang dibuat dengan tujuan untuk dapat memastikan keamanan dan kelayakan pesawat dalam melaksanakan operasionalnya di udara. Tipe pesawat yang berbeda membuat pesawat-pesawat tersebut memiliki CAMP yang berbeda.

Data LDND merupakan data yang terus diperbaharui dan terkait erat dengan CAMP maupun *Inspection Check* yang dilakukan pada pesawat. LDND memberikan informasi terkait kapan setiap aktivitas terakhir dilakukan dan kira kira kapan aktivitas inspeksi maupun perawatan rutin tersebut harus dilakukan kembali. Prinsip dasar dari pembaharuan data

ini adalah dengan menambahkan nilai interval dari setiap aktivitas dengan waktu terakhir aktivitas tersebut dilakukan. Terdapat lima komponen data pada LDND aktivitas inspeksi, diantaranya yaitu data interval pengerjaan aktivitas, data next do terdekat di setiap aktivitas (dalam bentuk rentang waktu dari waktu penulisan data LDND tersebut), data threshold setiap aktivitas (tidak boleh ada aktivitas yang boleh berjalan dalam masa ini, dalam bentuk rentang waktu dari waktu diberlakukan aktivitas pada LDND tersebut), durasi pengerjaan setiap aktivitas inspeksi, serta tipe interval dari setiap aktivitas (nilai 1 untuk tipe calendar day, nilai 2 untuk tipe flight hours).

B. Tahapan Optimasi

Penyelesaian kasus penjadwalan inspeksi dan perawatan pesawat dilakukan dengan berfokus pada penggunaan metode metaheuristik. Solusi utama yang dibentuk secara otomatis adalah berbentuk diskrit, dengan berfokus pada pembentukan aktivitas inspeksi yang akan dijadikan acuan. Tujuan utama dari optimasi yang dilakukan adalah mencari jadwal yang mampu memberikan pesawat terbang utilitas yang tinggi. Apabila terdapat pelanggaran yang dilakukan saat jadwal yang dibangkitkan diuji maka terdapat sejumlah penalti. Terdapat beberapa fase atau tahapan yang harus dilalui sebelum hasil optimal dapat diperoleh. Tahapan tersebut dapat dilihat pada Gambar 2.



Gambar 2. Alur Optimasi Penjadwalan Perawatan dan Inspeksi

C. Model Konseptual

Terdapat dua data utama yang digunakan pada proses optimasi ini, di antaranya yaitu data CAMP dan *Inspection Check*, khususnya pada bagian waktu terakhir kali aktivitas perawatan / pengecekan tersebut dilakukan, serta jumlah manhours yang dibutuhkan untuk dapat melaksanakan aktivitas tersebut. *Inspection Check* yang akan dipertimbangkan tidak hanya tipe A dan B seperti yang dilakukan oleh Sriram dan Haghani [10] melainkan seluruh tipe inspeksi, seperti tipe A, tipe B, tipe C, 2 tahunan, dan sebagainya. Tipe inspeksi yang dipertimbangkan mengikuti kebijakan dari setiap maskapai terhadap pesawat yang dimiliki.

Gambar 3 menunjukkan ilustrasi singkat konsep dasar dari keputusan shopvisit yang perlu dilakukan oleh setiap pesawat. *Inspection check* menjadi acuan utama dalam menentukan kapan pesawat perlu melakukan shopvisit. Hal ini dikarenakan *inspection check* memiliki pengaruh lebih dominan dibandingkan dengan CAMP [8]. Dasar tersebut menjadi acuan apakah shopvisit perlu dilakukan atau tidak di waktu tersebut. Saat pada waktu tersebut aktivitas inspeksi perlu dilakukan maka tahapan berikutnya adalah mencari aktivitas perawatan rutin apa saja yang perlu dikerjakan bersamaan dengan inspeksi tersebut. Jika terdapat perawatan rutin yang dapat dikerjakan secara bersamaan maka pada shopvisit yang akan dilakukan baik inspeksi maupun

perawatan rutin yang terpilih akan dilakukan pada waktu yang sama, sedangkan apabila tidak ada perawatan rutin yang perlu dilakukan bersamaan, maka hanya aktivitas inspeksi yang perlu dilakukan pada shopvisit tersebut.

Di setiap shopvisit yang dibuat akan terdapat satu ataupun lebih aktivitas CAMP yang harus dilakukan di waktu yang sama selain dari inspeksi yang harus dilakukan. Pada **Gambar 4** diilustrasikan bahwa terdapat dua shopvisit yang harus dilakukan dalam satu rangkaian aktivitas dari sebuah pesawat terbang. Pada setiap shopvisit (diberi warna hitam) terdapat satu aktivitas inspeksi (diberi warna biru) dan berikutnya diikuti dengan beberapa aktivitas CAMP (diberi warna oranye) dan saat tidak ada agenda shopvisit maka pesawat dapat menjalankan aktivitas operasional (diberi warna hijau) dengan mengikuti batas utilisasi yang sudah ditentukan oleh maskapai.

Terkadang terdapat aktivitas CAMP dengan interval yang cepat sehingga dalam waktu yang berdekatan aktivitas CAMP tersebut perlu dilakukan lebih dari sekali. Pada **Gambar 5** terdapat dua shopvisit yang dilakukan secara terpisah, satu shopvisit untuk A-Check dan lainnya untuk C-Check. Pada kedua shopvisit tersebut, terdapat satu aktivitas CAMP yang sama dan harus dilakukan pada masing-masing shopvisit, yaitu aktivitas CAMP 5 (diberi warna cokelat).

Dengan mempertimbangkan prinsip Flexible Maintenance pada proses optimasi, maka tidak menutup kemungkinan bahwa akan terdapat satu jadwal shopvisit yang melakukan lebih dari satu tipe Inspection Check. Apabila terjadi penggabungan waktu inspeksi seperti demikian, maka lamanya waktu shopvisit merupakan hasil penjumlahan seluruh waktu shopvisit yang normal sebelum kedua Inspection Check tersebut digabungkan. Namun walaupun dalam satu shopvisit tersebut terdapat dua Inspection Check, aktivitas CAMP yang dilakukan tidak boleh ada yang dilakukan lebih dari sekali. Hal tersebut dikarenakan jumlah shopvisit yang terjadi hanya sekali, sehingga setiap aktivitas CAMP

yang akan dilakukan pada saat tersebut hanya dapat dilakukan sekali. Pada **Gambar 6** diberikan ilustrasi dimana A-Check dan C-Check harus dilakukan secara bersamaan. Terdapat dua Inspection Check (dengan warna biru) yang harus dilakukan pada satu shopvisit (dengan warna hitam) dan setelah itu diikuti dengan melakukan beberapa aktivitas CAMP (dengan warna oranye) yang berbeda-beda.

Hasil yang akan diperoleh pada proses optimasi terbentuknya data LDND baru untuk beberapa periode mendatang mengikuti konsep yang diilustrasikan pada penjelasan sebelumnya, serta penugasan setiap aktivitas pada setiap periode tersebut. Jumlah periode data tersebut akan dihitung secara otomatis berdasarkan jumlah planning horizon yang diharapkan.

D. Model Matematis

Model matematis dirancang berdasarkan model konseptual yang telah dibuat. Selain itu model matematis pada penelitian ini juga merujuk pada penelitian yang dilakukan oleh Adianto & Siswanto[11] sebelumnya. Berikut merupakan penjelasan model matematis.

Misalkan suatu himpunan tugas inspeksi $\{1, 2, \dots, I\}$ dan suatu himpunan tugas CAMP $\bar{R}\{1, 2, \dots, R\}$ yang harus dijadwalkan pada satu pesawat. Setiap himpunan memiliki data penting, seperti waktu aktivitas berikutnya, waktu interval, waktu ambang, dan durasi. Pada set I dan R dimisalkan memiliki dua unit waktu yang berbeda, seperti hari dan jam penerbangan. Kedua tugas harus dijadwalkan pada himpunan periode $\bar{T}\{1, 2, \dots, T\}$, yang dapat dihitung dengan menggunakan horizon perencanaan h , selanjutnya aktivitas *inspection* (ix_i) dan aktivitas CAMP (ny_r), dan interval aktivitas inspeksi (ix_i) and aktivitas CAMP (iy_r), dihitung berdasarkan persamaan (1).

$$T = \left\lceil \frac{h}{\min_{i \in I} (ix_i), \min_{r \in R} (iy_r)} \right\rceil + 1 \quad (1)$$

Data yang tersedia harus dikonversi dari dua unit waktu yang berbeda menjadi satu unit waktu. Untuk selanjutnya, data interval,

ambang batas, dan durasi tugas inspeksi dikonversikan ke dalam persamaan berikut (2), (4), (6) dan (8), secara berurutan. Untuk selanjutnya, data interval dan data ambang batas tugas CAMP dikonversi oleh persamaan berikut (3), (5) dan (7), masing-masing

$$ncx_{i,1} \begin{cases} nx_i & \text{calendar days} \\ nx_i / u & \text{flight hours} \end{cases} \quad (2)$$

$$ncy_{r,1} \begin{cases} ny_r & \text{calendar days} \\ ny_r / u & \text{flight hours} \end{cases} \quad (3)$$

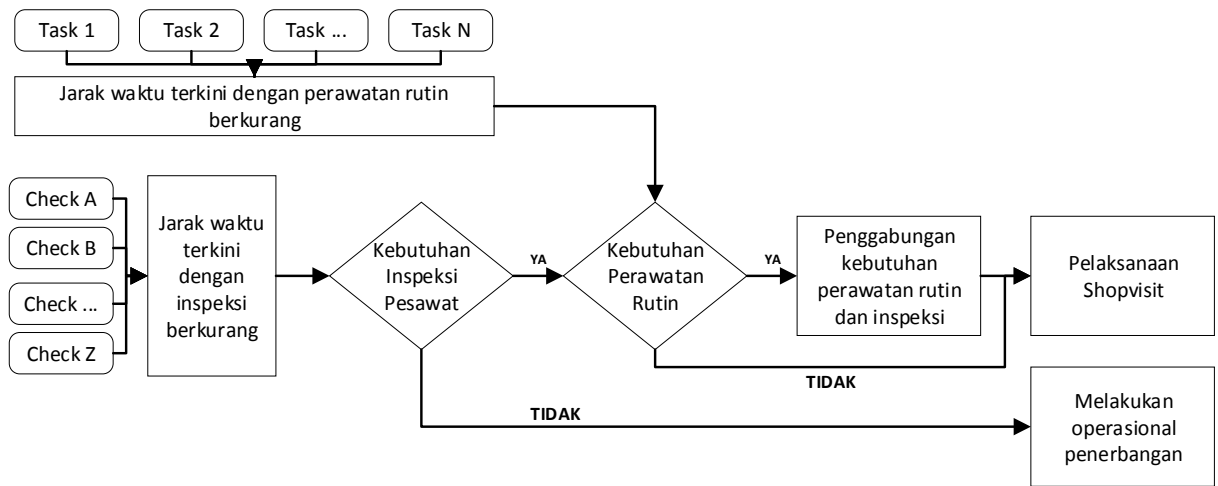
$$tcx_i \begin{cases} tx_i & \text{calendar days} \\ tx_i / u & \text{flight hours} \end{cases} \quad (4)$$

$$tcy_r \begin{cases} ty_r & \text{calendar days} \\ ty_r / u & \text{flight hours} \end{cases} \quad (5)$$

$$icx_i \begin{cases} ix_i & \text{calendar days} \\ ix_i / u & \text{flight hours} \end{cases} \quad (6)$$

$$icy_r \begin{cases} iy_r & \text{calendar days} \\ iy_r / u & \text{flight hours} \end{cases} \quad (7)$$

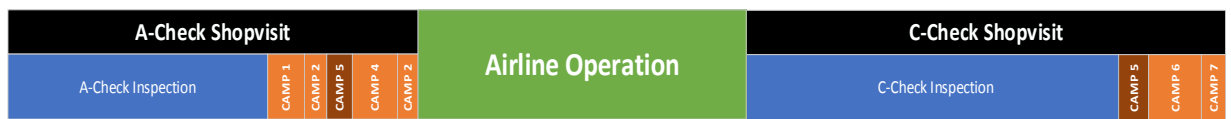
$$dcx_i \begin{cases} dx_i & \text{calendar days} \\ dx_i / u & \text{flight hours} \end{cases} \quad (8)$$



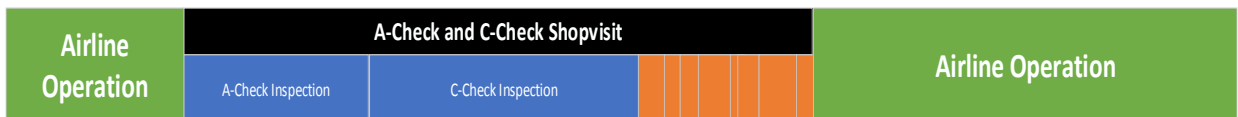
Gambar 3. Model Konseptual Kebutuhan Shopvisit



Gambar 4. Ilustrasi Rangkaian Aktivitas Pada Satu Unit Pesawat Terbang



Gambar 5. Ilustrasi Satu Aktivitas CAMP Yang Dilakukan Berulang Pada Dua Shopvisit Berbeda



Gambar 6. Ilustrasi Satu Shopvisit Terdiri Dari Dua Ataupun Lebih Inspection Check

Indeks dan variabel yang telah dikonversi digunakan untuk menentukan utilitas

maksimum sebuah pesawat dengan mengikuti persamaan berikut.

$$\max Z = \frac{\sum_{t \in \bar{T}} (c_{t+1} - c_t)}{c_{T+1} + m_T} \quad (9)$$

$$\sum_{i \in \bar{I}} x_{i,t} \geq 1, \forall t \in \bar{T} \quad (10)$$

$$\sum_{r \in \bar{R}} y_{r,t} \geq 0, \forall t \in \bar{T} \quad (11)$$

$$o_{i,t+1}^{ins} - o_{i,t}^{ins} \geq 0, \forall i \in \bar{I}, \forall t \in \bar{T} \quad (12)$$

$$o_{r,t+1}^{main} - o_{r,t}^{main} \geq 0, \forall r \in \bar{R}, \forall t \in \bar{T} \quad (13)$$

$$m_t - \sum_{i \in \bar{I}} (x_{i,t} \delta_i^{ins}) = 0, \forall t \in \bar{T} \quad (14)$$

$$i_{i,t}^{ins} - x_{i,t} (c_{t+1} + \phi_i^{ins} + m_t) = 0, \forall i \in \bar{I}, \forall t \in \bar{T} \quad (15)$$

$$i_{i,t}^{ins} - (1 - x_{i,t}) (o_{i,t}^{ins} + m_t) = 0, \forall i \in \bar{I}, \forall t \in \bar{T} \quad (16)$$

$$o_{i,t+1}^{ins} - (i_{i,t}^{ins} + i_{i,t}^{ins}) = 0, \forall i \in \bar{I}, \forall t \in \bar{T} \quad (17)$$

$$i_{r,t}^{main} - y_{r,t} (c_{t+1} + \phi_r^{main} + m_t) = 0, \forall r \in \bar{R}, \forall t \in \bar{T} \quad (18)$$

$$i_{r,t}^{main} - (1 - y_{r,t}) (o_{r,t}^{main} + m_t) = 0, \forall r \in \bar{R}, \forall t \in \bar{T} \quad (19)$$

$$o_{r,t+1}^{main} - (i_{i,t}^{ins} + i_{i,t}^{ins}) = 0, \forall r \in \bar{R}, \forall t \in \bar{T} \quad (20)$$

$$x_{i,t} o_{i,t}^{ins} - c_{t+1} \geq 0, \forall i \in \bar{I}, \forall t \in \bar{T} \quad (21)$$

$$y_{r,t} o_{r,t}^{main} - c_{t+1} \geq 0, \forall r \in \bar{R}, \forall t \in \bar{T} \quad (22)$$

$$o_{i,t+1}^{ins} - \tau_i^{ins} \geq 0, \forall i \in \bar{I}, \forall t \in \bar{T} \quad (23)$$

$$o_{r,t+1}^{main} - \tau_r^{main} \geq 0, \forall r \in \bar{R}, \forall t \in \bar{T} \quad (24)$$

$$c_1 = 0 \quad (25)$$

$$x_{i,t}, y_{r,t} \in \{0,1\} \quad (26)$$

Persamaan (9) merupakan fungsi tujuan dari model. Persamaan (10) dan (11) menjamin hasil keputusan memiliki setidaknya satu tugas yang telah diselesaikan di tiap periode untuk tiap tugas inspeksi dan tugas CAMP. Persamaan (12) dan (13) menjamin data *next do* yang dihasilkan pada periode $t + 1$ selalu lebih besar atau sama dengan data *next do* pada periode t untuk tiap tugas inspeksi dan tugas CAMP. Persamaan (14) menghitung total durasi perawatan yang mencakup seluruh durasi perawatan dari tugas inspeksi yang harus telah dilakukan pada periode t . Persamaan (17) dan (20) menghitung data *next do* pada periode $t + 1$ untuk tiap-tiap tugas inspeksi dan tugas CAMP. Persamaan (17) dan (20) dapat dihitung dengan persamaan (15) dan (18) untuk menghitung perbedaan nilai ketika tugas sedang dilakukan pada periode t , sedang persamaan (16) dan (19) untuk menghitung perbedaan nilai ketika tugas tidak sedang

dilakukan pada periode t . Persamaan (21) dan (22) menjamin data *next do* pada periode t selalu lebih besar atau sama dengan data *next do* pada periode $t + 1$ untuk tiap tugas inspeksi dan tugas CAMP. Persamaan (23) dan (24) menjamin data *next do* pada periode $t + 1$ selalu lebih besar atau sama dengan data ambang batas untuk tiap tugas inspeksi dan tugas CAMP. Persamaan (25) menjamin hari-hari pada periode 1 bernilai nol. Persamaan (26) menjamin variabel keputusan untuk tugas inspeksi dan tugas CAMP beratribut biner.

E. Pengembangan Model Metaheuristik

1) Particle Swarm Optimization (PSO)

PSO adalah metode metaheuristik yang meniru cara kerja kawanan burung saat mencari sumber makanan [12]. Setiap burung akan terbang ke lokasi terbaik berdasarkan semua informasi yang dibagikan oleh burung lain dalam kawanan itu, termasuk dirinya sendiri. Setiap individu harus membagikan nilai fungsi tujuan mereka untuk mencapai fungsi tujuan terbaik pada setiap iterasi dengan menyimpulkan semua informasi menjadi satu yang terbaik. Vektor kecepatan individu perlu diperbarui seperti dalam persamaan (27) dan (28) untuk membuat individu bergerak ke nilai yang ditargetkan.

$$v_{p,(a+1)} = \omega v_{p,a} + b_1 r_1 (pbest_{p,a} - x_{p,a}) \quad (27)$$

$$+ b_2 r_2 (gbest_a - x_{p,a}) \quad (28)$$

$$x_{p,(a+1)} = x_{p,a} + v_{p,(a+1)}$$

Kedua persamaan terdiri dari indeks dan variabel berikut:

- Himpunan individu atau populasi $\bar{P} \{1, 2, \dots, P\}$
- Himpunan iterasi $\bar{A} \{1, 2, \dots, \beta\}$
- $v_{p,a}$: kecepatan pada populasi saat iterasi a
- $x_{p,a}$: solusi populasi saat iterasi a
- $pbest_{p,a}$: solusi terbaik populasi saat iterasi a
- $gbest_a$: solusi terbaik dari semua populasi saat iterasi a

- r_2 : nilai acak dari 0 sampai 1

Beberapa parameter yang diatur secara manual, yaitu:

- b_1 : rasio solusi terbaik dari setiap populasi yang akan memengaruhi variabel kecepatan
- b_2 : rasio solusi terbaik dari semua populasi yang akan memengaruhi variabel kecepatan
- ω : rasio kecepatan iterasi sebelumnya yang akan mempengaruhi variabel kecepatan

Pada persamaan (27), dilakukan perhitungan kecepatan iterasi $a + 1$ dengan menentukan rasio kecepatan ω pada iterasi sebelumnya, rasio selisih solusi yang dihasilkan pada iterasi terakhir dengan solusi terbaik yang dihasilkan dari populasi p , dan rasio selisih antara solusi yang dihasilkan pada iterasi terakhir dengan solusi terbaik yang dihasilkan dari semua populasi. Dengan menggunakan nilai kecepatan yang telah diperbarui, kita dapat menghitung posisi baru setiap individu menggunakan persamaan (28) dengan cara menambahkan solusi yang dihasilkan dari iterasi terakhir dan kecepatan dari hasil persamaan (27). Solusi terbaik adalah g_{best} saat iterasi terakhir.

2) Crow Search Algorithm (CSA)

Crow Search Algorithm (CSA) merupakan sebuah algoritma metaheuristik yang dikembangkan oleh Askarzadeh[13]. CSA merupakan salah satu *population-based technique* yang bekerja berdasarkan kebiasaan dari burung gagak dalam mencari makanan dan menyimpan makanan yang sudah didapatkan ke sarang tersembunyi mereka masing-masing. Penyimpanan secara tersembunyi tersebut dilakukan oleh burung gagak agar makanan mereka tidak tercuri oleh burung gagak lainnya.

Burung gagak memiliki kemampuan inteligensi yang baik. Seekor burung gagak saat akan melakukan pencarian makanan akan memperhatikan gerak gerik dari burung lainnya dan bertindak sesuai dengan kondisi gerak gerik tersebut. Ketika burung gagak

yang diperhatikan lengah maka burung gagak tersebut mencuri makanan dari burung gagak yang diamati khususnya saat sarang burung gagak tersembunyi yang diamati tersebut sudah diketahui dan burung gagak tersebut meninggalkan sarang saat akan berburu. Askarzadeh[13] memaparkan algoritma CSA dalam beberapa tahapan berikut

Pada tahap awal ini rincian terkait kasus yang akan diselesaikan, variabel keputusan dan konstrain perlu didefinisikan dengan jelas. Beberapa parameter CSA yang dapat diatur seperti banyaknya kumpulan burung (N), jumlah maksimal iterasi aktivasi algoritma yang dibutuhkan ($iter_{max}$), panjang jarak perpindahan saat terbang (FL) dan probabilitas kepekaan burung gagak (AP) ditentukan nilainya.

Sejumlah N burung gagak diletakkan secara acak dalam ruang pencarian d -dimensi. Setiap burung gagak mewakili sebagian dari *feasible solution* dari problem yang dimasukkan dan d merupakan jumlah variabel keputusan yang dimiliki. Beberapa keterangan tersebut dapat terbentuk matriks untuk seluruh burung gagak seperti pada persamaan (29). *Memory* dari setiap gagak diinisiasikan. Tahap penyimpanan *memory* ini masih dalam iterasi awal dan setiap burung gagak masih belum memiliki pengalaman dalam menghadapi kondisi sekitarnya sehingga diasumsikan ingatan mereka dalam menyimpan makanannya masih sama dengan posisi mereka saat ini. Berdasarkan keterangan tersebut dapat terbentuk matriks untuk seluruh burung gagak seperti persamaan (30).

$$crows = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_d^1 \\ x_1^2 & x_2^2 & \dots & x_d^2 \\ \dots & \dots & \dots & \dots \\ x_1^N & x_2^N & \dots & x_d^N \end{bmatrix} \quad (29)$$

$$memory = \begin{bmatrix} m_1^1 & m_2^1 & \dots & m_d^1 \\ m_1^2 & m_2^2 & \dots & m_d^2 \\ \dots & \dots & \dots & \dots \\ m_1^N & m_2^N & \dots & m_d^N \end{bmatrix} \quad (30)$$

Setiap posisi dari burung gagak tersebut diukur kualitasnya dengan cara mengubah posisi burung gagak dalam satu variabel keputusan menjadi sebuah urutan kualitatif. Urutan tersebut dimasukkan ke dalam fungsi tujuan permasalahan utama.

Seluruh burung gagak membangkitkan target posisi baru dalam ruang pencarian yang ada dengan catatan satu burung gagak akan mencoba mengikuti burung gagak lain yang dipilih secara acak untuk mengetahui dimana burung gagak lain tersebut menyimpan makanannya berdasarkan *memory* burung gagak lainnya. Notasi matematis dari keterangan tersebut dapat dilihat pada persamaan (31)

$$x_i^{i,(iter+1)} = \begin{cases} x_i^{i,iter} + r_i \times FL_i^{i,iter} \times (m_j^{i,iter} - x_i^{i,iter}) & r_i \geq AP_i^{i,iter} \\ \text{random position,} & \text{otherwise} \end{cases} \quad (31)$$

Feasibility solution dari posisi gagak yang dibangkitkan perlu dicek kembali. Apabila posisi terbaru burung gagak tersebut *feasible* dalam memecahkan masalah yang ada maka burung gagak akan berpindah ke posisi tersebut. Selain itu burung gagak akan tetap pada posisinya saat ini. Setelah itu mencari nilai fungsi tujuan untuk setiap variabel keputusan pada posisi burung gagak terbaru. Setiap burung gagak memperbaharui *memory* mereka masing masing dengan persamaan (32). Apabila nilai fungsi tujuan yang dihasilkan pada posisi terbaru lebih baik maka *memory* akan diperbaharui sesuai posisi terbaru tersebut. Apabila tidak, maka burung gagak akan menyimpan *memory* yang merupakan posisi sebelumnya.

$$m_i^{i,(iter+1)} = \begin{cases} x_i^{i,(iter+1)} & f(x_i^{i,(iter+1)}) \text{ better than } f(m_i^{i,iter}) \\ m_i^{i,(iter+1)} & \text{otherwise} \end{cases}$$

Selama algoritma CSA ini dijalankan, tahapan akan terus diulangi hingga jumlah iterasi

maksimal yaitu $iter_{max}$ terpenuhi. Posisi terbaik yang didapatkan kemudian disimpan sebagai solusi teroptimisasi dari permasalahan yang dimasukkan.

3) Greedy Randomized Adaptive Search Procedures (GRASP)

GRASP merupakan metode pencarian solusi optimal yang dikembangkan oleh Feo dan Resende[14]. Metode ini merupakan metode yang adaptif, sehingga dapat digunakan di berbagai kasus optimasi yang ada. Kemampuan adaptasi dari metode ini karena solusi yang dibangkitkan akan terus terbaharui di setiap iterasi perhitungan mengikuti solusi representatif yang dibangkitkan oleh metode GRASP ini. Metode GRASP terdiri dari dua tahapan yang dilakukan secara iterative hingga solusi terbaik ditemukan. Dua tahapan tersebut terdiri dari tahap konstruksi dan tahap *local search*.

Tahap konstruksi dilakukan dengan cara membangkitkan solusi satu per satu. Solusi yang dibangkitkan nantinya disusun ulang mengikuti perhitungan fungsi tujuan dari setiap kasus optimasi yang dihadapi. Proses penyusunan ulang tersebut disebut sebagai tahap *local search*. Pada tahap konstruksi, solusi dibangkitkan hanya sebagian kecil saja sebagai perwakilan dari seluruh solusi yang diharapkan. Pada saat tahap konstruksi selesai dilakukan, maka dilakukan tahap *local search* dengan membangkitkan solusi secara utuh (sesuai dengan yang dibutuhkan oleh kasus yang dihadapi) dengan mempertimbangkan solusi representatif yang sudah dibangkitkan di tahap konstruksi.

Penelitian ini memiliki variabel keputusan dengan bentuk matriks dua dimensi: $x_{(i,t)}$ dan $y_{r,t}$ yang masing-masing mewakili tugas inspeksi dan CAMP, pada setiap periode. Masing-masing metode metaheuristik harus menghasilkan variabel-variabel tersebut, seperti yang diilustrasikan pada Tabel 1. Menggunakan formulir ini, metode metaheuristik kadang-kadang dapat menghasilkan solusi yang melanggar beberapa kendala masalah. Untuk menghindari

menghasilkan solusi yang tidak valid, GRASP diimplementasikan pada kedua metode metaheuristik dan menciptakan bentuk variabel keputusan baru yang diisi dengan ID tugas inspeksi, seperti diilustrasikan dalam Tabel 2.

Tabel 1. Form Perubahan Variabel Keputusan

	Per 1	Per 2	...	Per T
$In - 1$	1	0	...	1
$In - 2$	0	1	...	1
...
$In - l$	0	0	...	0
$Ca - 1$	0	0	...	1
$Ca - 2$	1	1	...	1
...
$Ca - R$	1	0	...	1

Tabel 2. Form Perubahan Variabel Keputusan dengan GRASP

	Per 1	Per 2	...	Per T
$In - ref$	3	4	...	2

4) *Hybrid PSO GRASP & Hybrid CSA GRASP*

Metode PSO dan CSA harus menghasilkan data dengan mengisi form seperti yang dijelaskan di bagian sebelumnya dengan nilai desimal dari 0 hingga 1, seperti yang diilustrasikan dalam Tabel 3. Nilai desimal tersebut dikonversi menjadi ID tugas inspeksi dengan menemukan perkalian terdekat nilai desimal dengan I . Untuk contoh, jika $I = 6$, bentuk yang dihasilkan pada Tabel 3 akan ditransformasikan seperti diilustrasikan pada Tabel 2.

Tabel 3. Form Variabel Keputusan dengan GRASP

	Per 1	Per 2	...	Per T
$In-ref$	0.439	0.72	...	0.272

$$ncx_{ref,t} - x_{i,t} \cdot ncx_{i,t} \geq 0 \tag{33}$$

$$ncx_{ref,t} - y_{r,t} \cdot ncy_{r,t} \geq 0 \tag{34}$$

Untuk mengkonversi variabel keputusan yang dihasilkan dengan GRASP, seperti pada Tabel 2, ke dalam bentuk variabel keputusan asli seperti pada Tabel 3, digunakan metode pencarian lokal yang memenuhi persamaan (33) dan (34). Kedua persamaan memastikan tugas inspeksi dan CAMP yang dipilih untuk

dilakukan selanjutnya harus lebih rendah atau sama dengan nilai tugas yang dimaksud pada inspeksi berikutnya.

5) *Model yang diajukan*

Model yang diajukan memiliki tahapan-tahapan penyelesaian. Pertama, dibuat tugas *inspection* menggunakan *random number generator* dan mengonversikannya menggunakan metode GRASP untuk iterasi pertama, dan menggunakan PSO-GRASP atau CSA-GRASP untuk iterasi selanjutnya.

Untuk iterasi hingga jumlah iterasi maksimum tercapai, dilakukan pembaruan waktu maksimum dari setiap periode perawatan dalam setiap solusi yang dihasilkan, kemudian menyusun semua tuga *inspection* dan CAMP yang dipesan. Menggunakan tugas yang dipesan, dilakukan perhitungan durasi perawatan yang dibutuhkan. Menggunakan tugas yang dipesan dan durasi perawatan di setiap periode, dilakukan perhitungan kapan tugas inspeksi dan CAMP selesai dilakukan dan kemudian memperbarui nilai utilitas pesawat. Menggunakan tugas yang dipesan, dilakukan validasi solusi yang dihasilkan setiap periode. Setiap solusi yang valid akan ditandai dan akan digunakan untuk menghitung nilai fungsi objektif dari solusi yang dihasilkan. Tahapan-tahapan dari metode Hybrid PSO-GRASP atau CSA-GRASP digambarkan pada Gambar 7.

Nilai fungsi tujuan dihitung berdasarkan nilai utilitas dari solusi yang valid dan nilai penalti dari solusi yang tidak valid di setiap periode. Nilai penalti setiap periode memiliki angka yang berbeda berdasarkan periodenya. Peningkatan nilai periode berarti penurunan nilai penalti sehingga nilai penalti tertinggi akan berada di periode pertama. Nilai penalti dihitung berdasarkan persamaan (35). Ketika periode T sama dengan 30 dan solusi yang dihasilkan memiliki solusi tidak valid di periode $t = \{2, 5, 10\}$ maka $penalty_2$ sama dengan 2900, $penalty_5$ sama dengan 2600, $penalty_{10}$ sama dengan 2100. Dari nilai-nilai itu kita dapat menyimpulkan bahwa solusi yang

dihasilkan memiliki total penalti sama dengan 7600.

```

Generate referred inspection task.
For ( from p ← 1 to p ← P̄ )
  For ( from t ← 1 to t ← T̄ )
    Update max period time.
    For ( from i ← 1 to i ← Ī )
      Construct task order of all inspection
      task.
    End
  For ( from r ← 1 to r ← R̄ )
    Construct task order of all CAMP
    task.
  End
  Update maintenance duration.
  Update next do value of the next period.
  Update utilization of the aircraft.
  Validate generated solution.
End
Compute objective function value for each
solution population.
End
Save maximum objective value of each
generated population

```

Gambar 7. Pseudocode of PSO-GRASP Model

$$penalty_t - 100(\bar{T} \cdot t + 1) = 0 \quad (35)$$

$$ofvalue_p - \frac{\sum_{t \in \bar{T}} vld_{p,t}(c_{t+1} - c_t)}{c_{T+1} + m_T} + penalty_t(1 - vld_{p,t}) = 0 \quad (36)$$

Nilai fungsi tujuan dihitung menggunakan nilai utilitas dan nilai penalti mengikuti persamaan (36). Ketika solusi yang dihasilkan dalam populasi p dihitung nilai fungsi objektifnya, nilai fungsi objektif saat ini akan ditambahkan dengan penjumlahan nilai utilitas dalam setiap periode yang ditandai sebagai solusi yang valid dibagi dengan waktu *next* dopada periode terakhir \bar{T} ditambahkan oleh durasi perawatan periode terakhir \bar{T} seperti yang dijelaskan dalam persamaan (9). Ketika solusi yang dihasilkan dalam populasi p dan periode t ditandai sebagai solusi yang tidak valid, nilai saat ini dari fungsi tujuan akan

dikurangi dengan nilai penalti, seperti yang dijelaskan dalam persamaan (35).

3. Hasil dan Pembahasan

A. Desain Eksperimen

Penelitian ini mengevaluasi kinerja kedua metode PSO-GRASP dan CSA-GRASP dengan empat set data masalah acak. Data set mempunyai skala yang berbeda, dari skala kecil hingga skala besar, dengan lima dan 20 tugas pemeriksaan, dikombinasikan dengan 500 dan 1000 tugas CAMP. Kombinasi setiap dataset dijelaskan pada Tabel 4. Seperti Grup G1 memiliki kombinasi lima tugas inspeksi dan 500 tugas CAMP. Eksperimen model optimasi dilakukan pada sebuah perangkat komputer pribadi. Spesifikasi dari komputer pribadi itu ialah komputer notebook dengan processor Intel Core i7 Gen 7th (7700 HQ) dan RAM sebesar 16GB. Komputer tersebut dijalankan dengan menggunakan sistem operasi Ubuntu versi 18.04.2 LTS. Model optimasi disusun dalam bahasa R dan dijalankan pada r-base versi 3.6.0 (Planting of a Tree) dengan bantuan GUI (Graphical User Interface) dari RStudio Desktop versi 1.2.1335.

Tabel 4. Kombinasi Dataset

Group	Inspection Task	CAMP Task
G1	5	500
G2	5	1000
G3	20	500
G4	20	1000

B. Analisis Hasil Komputasi

Tabel 5 merupakan hasil pengolahan data beberapa grup dataset yang dioleh menggunakan dua metode metaheuristik, yaitu PSO-GRASP dan CSA-GRASP. Data yang ditampilkan pada tabel tersebut merupakan data hasil optimasi dengan nilai fungsi tujuan maksimum yang dapat diperoleh dari setiap grup beserta dari setiap plan horizon yang digunakan. Terdapat tiga nilai parameter plan horizon yang digunakan pada eksperimen ini

diantaranya yaitu 730 hari, 1460 hari dan 2190 hari. Hasil menunjukkan bahwa 75% percobaan menunjukkan bahwa PSO-GRASP mampu memberikan nilai fungsi tujuan yang lebih baik jika dibandingkan dengan CSA-GRASP pada grup data dan plan horizon serupa, diantaranya pada Test12, Test13, Test14, Test15, Test16, Test17, Test19, Test20 dan Test21 dan 25% lainnya mampu dikuasai oleh CSA-GRASP. Waktu pengolahan data yang paling singkat dari seluruh percobaan 50% mampu diperoleh dengan menggunakan CSA-GRASP, diantaranya pada Test12, Test13, Test14, Test16, Test17 dan Test20. Dari hasil eksperimen yang sudah dilakukan terlihat bahwa semakin banyak jumlah data pada dataset yang dianalisa, dengan menggunakan parameter dan batasan yang sama membuat nilai fungsi tujuan yang dihasilkan pada kedua metode cenderung menurun serta waktu optimasi yang cenderung meningkat. Hasil berikutnya melihat apakah terdapat parameter lain yang dapat mempengaruhi nilai fungsi tujuan yang dihasilkan.

Hasil yang ditunjukkan Tabel 6 bahwa perbedaan parameter iterasi dan populasi dapat mempengaruhi nilai fungsi tujuan yang dapat diperoleh serta waktu komputasi yang dibutuhkan pada dataset G3. Nilai yang dihasilkan pada Test23 dijadikan acuan dasar untuk membandingkan nilai fungsi tuuan dan waktu pengolahan data pada parameter iterasi dan populasi awal hingga dinaikan menjadi dua kali lipat nilai parameter dasar tersebut. Pada PSO-GRASP, meningkatkan dua kali lipat pada kedua parameter iterasi dan populasi mampu memberikan perbedaan yang signifikan secara positif terhadap nilai fungsi tujuan, yaitu sebesar 15% sedangkan pada CSA-GRASP, meningkatkan dua kali lipat hanya pada parameter iterasi yang mampu

memberikan perbedaan yang signifikan secara positif terhadap nilai fungsi tujuan, yaitu sebesar 8.45%.

Pada beberapa eksperimen yang sudah dilakukan dan dijelaskan pada bagian sebelumnya terlihat bahwa kedua metode mampu menyelesaikan permasalahan *aircraft maintenance* ini dengan baik. Metode CSA dan PSO memiliki perbedaan dimana CSA tidak mempertimbangkan keberadaan solusi terbaik dari seluruh populasi dikarenakan sifat dari burung gagak yang sukar untuk bekerja sama, dan adanya fungsi diversifikasi perhatian apabila seekor burung gagak merasa sudah diawasi oleh burung lainnya yang membuat pergerakannya cenderung acak, maka fokus pencarian solusi dapat lebih luas walaupun tetap secara perlahan menuju satu solusi terbaik. Prosedur yang dilakukan pada metode CSA ini sangat membantu apabila kombinasi solusi yang dibangkitkan memiliki banyak kemungkinan seperti permasalahan ini. Karena fokus pencarian solusi lebih luas, hal ini menyebabkan metode CSA cenderung sulit untuk konvergen sehingga dalam penelitian ini didapatkan hasil bahwa metode PSO-GRASP dapat menemukan solusi yang lebih baik dari CSA-GRASP pada data set yang besar. Metode GRASP sendiri sangat membantu metode PSO ataupun CSA dalam menemukan solusi optimal. Hal ini disebabkan karena GRASP menggunakan local search method untuk memastikan nilai solusi yang dibangkitkan memenuhi seluruh persyaratan untuk dinyatakan valid terlebih dahulu melalui proses decoding solusi yang sudah dibangkitkan oleh PSO ataupun CSA didalamnya. GRASP juga menuntut PSO dan CSA untuk membangkitkan solusi sesuai dengan encoded form yang sudah ditentukan sebelum bisa melalui proses decoding.

Tabel 5. Hasil Uji Grup *Data Set* Dengan 10 Replikasi

ID	Plan Horizon	Group	PSO-GRASP			CSA-GRASP		
			Last Maint. Finish Time	Obj Value	CPU Time	Last Maint. Finish Time	Obj Value	CPU Time
Test11	730	G1	782	78.14	110.45	782	78.14	119.42

Tabel 5. Hasil Uji Grup *Data Set* Dengan 10 Replikasi

ID	Plan Horizon	Group	PSO-GRASP			CSA-GRASP		
			Last Maint. Finish Time	Obj Value	CPU Time	Last Maint. Finish Time	Obj Value	CPU Time
Test12		G2	781	78.11	181.19	783	77.91	180.11
Test13		G3	1100	49	258.64	1219	47.58	252.79
Test14		G4	923	48.43	390.59	1226	47.88	387.84
Test15		G1	1776	77.03	197.65	1482	77	204.25
Test16	1460	G2	1673	77.35	337.11	1481	76.98	332.76
Test17		G3	2361	46.3	508.42	2479	42.48	499.95
Test18		G4	2362	43.4	744.06	2518	43.61	758
Test19	2190	G1	2819	76.84	289.77	2827	76.62	313.21
Test20		G2	2669	76.89	496.17	2831	76.52	484.7
Test21		G3	4171	41.99	710.32	3910	41.87	724.14
Test22		G4	4419	41.46	1058.52	3709	41.98	1076.52

Tabel 6. Hasil Uji Grup *Data Set* 3 Dengan Perbedaan Parameter

ID	Iteration / Population Limit	PSO-GRASP				CSA-GRASP			
		Last Maint. Finish Time	Obj Value	% Gap	CPU Time	Last Maint. Finish Time	Obj Value	% Gap	CPU Time
Test23	1000/100	1051	47.48	-	240.68	918	46.85	-	240.75
Test24	1000/200	861	53.08	11.79	475.16	967	49.95	6.62	467.84
Test25	2000/100	1146	49.92	5.14	481.61	811	50.81	8.45	479.33
Test26	2000/200	914	54.6	15.00	930.88	928	48.6	3.74	815.95

4. Kesimpulan dan Saran

Permasalahan penjadwalan aktivitas inspeksi dan perawatan rutin pada pesawat merupakan permasalahan yang rutin dihadapi oleh pihak maskapai penerbangan. Tingkat penelitian terkait penggabungan penjadwalan kedua aktivitas tersebut saat ini masih sangat sedikit. Jumlah penelitian yang sedikit membuat pihak maskapai penerbangan tidak paham bagaimana penjadwalan perlu dilakukan agar dapat meningkatkan keuntungan mereka, khususnya dari operasional pesawat yang sudah dimiliki. Model optimasi yang sudah dikembangkan pada penelitian ini terbukti mampu menyelesaikan permasalahan tersebut dan dapat digunakan langsung oleh maskapai penerbangan yang membutuhkan penjadwalan aktivitas perawatan rutin dan inspeksi dengan mempertimbangkan waktu operasional yang akan diberikan. Model optimasi yang sudah dikembangkan terbukti mampu membangkitkan solusi aktivitas perawatan

dengan memaksimalkan utilitas dari pesawat yang melakukan perawatan tersebut.

Metode CSA-GRASP terbukti dapat menyelesaikan permasalahan penjadwalan inspeksi dan perawatan rutin pesawat namun dengan hasil yang tidak begitu baik jika dibandingkan dengan metode PSO-GRASP, khususnya pada dataset dengan skala yang besar. Hal ini dapat dilihat bahwa pada dataset G4 algoritma PSO-GRASP memberikan nilai objektif yang lebih kecil dari pada CSA-GRASP dengan waktu komputasi yang relatif sama. CSA-GRASP mampu memberikan performa yang lebih baik dibandingkan dengan PSO-GRASP dataset yang digunakan merupakan data dengan skala yang kecil serta dengan parameter *planhorizon* yang kecil. Hal ini dapat dilihat bahwa pada *plan horizon* 730 algoritma CSA-GRASP memberikan nilai objektif yang lebih kecil atau sama dengan algoritma PSO-GRASP pada semua dataset.

Bagi penelitian terkait berikutnya, peneliti dapat mencoba mengembangkan ruang lingkup

lain dari kasus penjadwalan inspeksi dan perawatan pesawat dengan mengurangi asumsi yang digunakan pada pengembangan model optimasi ini. Selain itu peneliti dapat mencoba menggunakan model optimasi dengan bantuan metode metaheuristik lain untuk menemukan metode metaheuristik yang sesuai untuk permasalahan perawatan dan inspeksi pada pesawat (*problem-specific*).

5. Persantunan

Penulis mengucapkan terima kasih yang sebesar-besarnya atas pendanaan penelitian ini dari Kementrian Riset Teknologi dan Pendidikan Tinggi melalui Skema Hibah Penelitian Tesis Magister dengan No. 781/PKS/ITS/2019.

6. Daftar Pustaka

- [1] F. Gargiulo, D. Pascari, and S. Venticinque, "A Multi-agent and Dynamic Programming Algorithm for Aeronautical Maintenance Planning," *Int. Conf. P2P, Parallel, Grid, Cloud Internet Comput.*, no. 6681263, pp. 410–415, 2013.
- [2] R. Gopalan and K. T. Talluri, "The Aircraft Maintenance Routing Problem," *Oper. Res.*, vol. 46, no. 2, pp. 260–271, 1998.
- [3] Z. Liang, W. A. Chaovalitwongse, H. C. Huang, and E. L. Johnson, "On a New Rotation Tour Network Model for Aircraft Maintenance Routing Problem," *Transp. Sci.*, vol. 45, no. 1, pp. 109–120, 2011.
- [4] O. Ezzinbi, M. Sarhani, A. El Afia, and Y. Benadada, "A metaheuristic approach for solving the airline maintenance routing with aircraft on ground problem," *Proc. 2nd IEEE Int. Conf. Logist. Oper. Manag. GOL 2014*, pp. 48–52, 2014.
- [5] N. A. Al-Thani, M. Ben Ahmed, and M. Haouari, "A model and optimization-based heuristic for the operational aircraft maintenance routing problem," *Transp. Res. Part C Emerg. Technol.*, vol. 72, pp. 29–44, 2016.
- [6] A. E. E. Eltoukhy, F. T. S. Chan, S. H. Chung, and B. Niu, "A model with a solution algorithm for the operational aircraft maintenance routing problem," *Comput. Ind. Eng.*, vol. 120, pp. 346–359, 2018.
- [7] N. Safaei and A. K. S. Jardine, "Aircraft routing with generalized maintenance constraints," *Omega (United Kingdom)*, vol. 80, pp. 111–122, 2018.
- [8] G. Nickles, H. Him, S. Koenig, A. Gramopadhye, and B. Melloy, "A Descriptive Model of Aircraft Inspection Activities," pp. 1–6, 1999.
- [9] Federal Aviation Administration, "FAR Part 91 Sec. 91.409 effective as of 08/01/2004." .
- [10] C. Sriram and A. Haghani, "An optimization model for aircraft maintenance scheduling and re-assignment," *Transp. Res. Part A Policy Pract.*, vol. 37, pp. 29–48, 2003.
- [11] A. N. Adianto and N. Siswanto, "Hybridized Particle Swarm Optimization for Aircraft Inspection Check and Continuous Airworthiness Maintenance Program," in *International Conference on Industrial Technology (ICONIT) - Presented*, 2019.
- [12] B. Santosa and T. J. Ai, *Pengantar Metaheuristik - Implementasi dengan Matlab*, 1st ed. Surabaya: ITS Tekno Sains, 2017.
- [13] A. Askarzadeh, "A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm," *Comput. Struct.*, vol. 169, pp. 1–12, Jun. 2016.
- [14] T. A. Feo and M. G. C. Resende, "Greedy Randomized Adaptive Search Procedures," *J. Glob. Optim.*, pp. 109–133, 1995.