

Penjadwalan Pemeliharaan Trek Kereta dengan Metode *Distributed Model Predictive Control*

Muhammad Faris¹, Ircham²

Abstract—This work addresses the development of Distributed Model Predictive Control (DMPC) approaches for the planning of maintenance operations of large-scale railway infrastructure formulated as a Mixed-Integer Linear Programming (MILP) problem. The proposed optimization problem is solved using two different decomposition schemes: Alternating Direction Method of Multipliers (ADMM) and Distributed Robust Safe But Knowledgeable (DRSBK). The original distributed algorithms are modified to handle the non-convex nature of the optimization problem, hence improving the solution quality. The results of large-scale test instances show that DRSBK can outperform the conventional centralized approach and ADMM, by providing the closest-to-optimum solution while requiring the least computation time.

Intisari—Makalah ini mengembangkan metode *Distributed Model Predictive Control* (DMPC) untuk perencanaan operasi pemeliharaan infrastruktur kereta api skala besar yang diformulasikan sebagai masalah *Mixed-Integer Linear Programming* (MILP). Masalah optimasi yang diusulkan kemudian diselesaikan dengan menggunakan dua skema optimasi DMPC yang berbeda: *Alternating Direction Method of Multipliers* (ADMM) dan *Distributed Robust Safe But Knowledgeable* (DRSBK). Selain itu, algoritme standarnya dimodifikasi untuk menangani sifat *non-convex* dari masalah optimasi, sehingga dapat meningkatkan kualitas solusi. Hasil pengujian dalam masalah skala besar menunjukkan bahwa DRSBK dapat mengungguli metode *centralized* yang konvensional dan ADMM, dengan menghasilkan solusi mendekati ke global optimum tetapi membutuhkan waktu komputasi minimum.

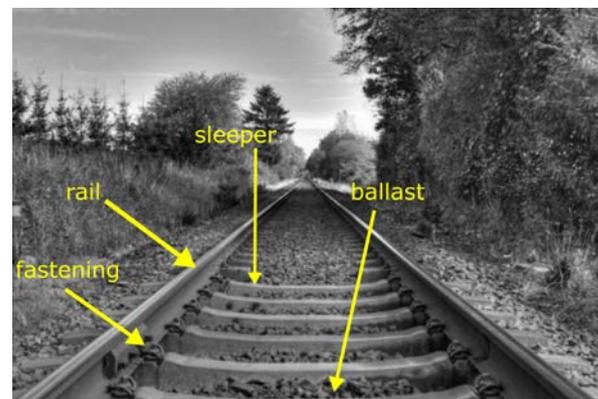
Kata Kunci— Penjadwalan, rekayasa perkeretaapian, optimisasi, MIP, kendali prediktif.

I. PENDAHULUAN

Infrastruktur trek kereta api umumnya tersusun atas berbagai jenis komponen, seperti *sleeper*, *rail*, *ballast*, *fastening*, *weld*, *ballast*, dan sebagainya, seperti yang ditunjukkan dalam Gbr. 1. Semua komponen tersebut saling terkoneksi dan saling mendukung satu sama lain. Di antara semua komponen tersebut, *ballast*, yang merupakan bebatuan kerikil, adalah komponen yang paling vital karena berperan sebagai penopang utama rel dan untuk mengatur perataan di posisi yang ditunjuk [1]. Karena penggunaan jalur kereta api

yang reguler, kualitas *ballast* menurun seiring waktu (degradasi). Untuk mengendalikan level degradasi, *ballast* harus dijaga dan dipelihara agar dapat selalu memenuhi standar teknis dan keselamatan. Dalam hal ini, manajer infrastruktur kereta api bertanggung jawab untuk menentukan tipe pemeliharaan untuk *ballast*, yang meliputi *tamping*, *cleaning*, atau *renewal*; termasuk juga waktu dan lokasi operasi pemeliharaan.

Proses penjadwalan operasi pemeliharaan adalah hal yang kompleks. Hal ini dikarenakan baik biaya operasi maupun unjuk kerja trek harus dapat dioptimalkan secara seimbang. Salah satu cara untuk mendapatkan jadwal yang optimal secara sistematis adalah dengan menggunakan metode optimisasi yang umumnya ditanamkan dalam perangkat sistem pendukung keputusan (*Decision Support System/DSS*).



Gbr. 1 Komponen penyusun trek kereta api (Sumber: IDNTimes).

Dalam literatur, berbagai metode telah diusulkan untuk memproses data kondisi jalur kereta api yang nantinya digunakan untuk menentukan jadwal pemeliharaan yang optimal. Referensi [2] mengembangkan metode perencanaan operasi *tamping* di seluruh jalur kereta api. Tujuannya adalah untuk meminimalkan total biaya operasi *tamping*. Kemudian, ditambahkan opsi operasi *renewal* yang kemudian diaplikasikan dalam *framework Model Predictive Control* (MPC) [3]. Selain itu, pemodelan operasi pemeliharaan yang variatif juga digunakan untuk mengoptimisasi perencanaan pekerjaan pemeliharaan multikomponen pada suatu sistem infrastruktur kereta api yang berskala luas [4].

Pada praktiknya, sistem pendukung keputusan berbasis metode optimisasi harus mempertimbangkan bahwa jalur kereta api sangat panjang dan tersusun atas banyak segmen rel, sehingga cakupannya akan sangat luas. Selain itu, dalam rangka mendapatkan jadwal yang lebih akurat, manajer infrastruktur kereta api membutuhkan keputusan yang berupa peta lokasi segmen trek tertentu dengan ketelitian jarak yang tinggi.

¹Staf pengajar, Departemen Teknik Elektro dan Teknologi Informasi Fakultas Teknik Universitas Gadjah Mada, Jl. Grafika No 2 Kampus UGM Yogyakarta, 55281 (tlp: 0274-552 305; fax: 0274-4321982; e-mail: mfaris@ugm.ac.id)

²Staf pengajar, Jurusan Teknik Sipil Institut Teknologi Nasional Yogyakarta, Jl. Babarsari, Caturtunggal, Yogyakarta 55281 (tlp: 0274-485 390; fax: 0274-487249; e-mail: irc276@yahoo.co.id)

Jika setiap segmen rel memiliki model dinamik yang berbeda dan independen, maka pengambilan keputusan pemeliharaan lintasan kereta api yang optimal akan menghasilkan masalah optimisasi yang berskala besar dengan jumlah variabel proses yang sangat besar. Dalam literatur, metode yang telah ada sebelumnya hampir semua berbasis skema optimisasi terpusat, dengan pemrosesan informasi dan perhitungan semua variabel keputusan dilakukan dalam sebuah sistem yang tersentralisasi (*centralized node*) [2]–[6]. Dari perspektif komputasi, skema *centralized* memiliki kelemahan ketika jumlah variabel proses meningkat secara masif. Hal ini dikarenakan peningkatan jumlah variabel keputusan secara linear dalam masalah optimisasi pemeliharaan seperti itu mengakibatkan peningkatan beban komputasi eksponensial yang diperlukan oleh DSS [7].

Beberapa penelitian sebelumnya di bidang ini juga sudah menggunakan skema *non-centralized*. Telah diusulkan skema optimisasi berbasis *multi-level MPC* untuk pemeliharaan kereta api [8], [9]. Pendekatan yang diusulkan dibagi atas pengoptimal tingkat tinggi, tingkat menengah, dan tingkat rendah, yang memungkinkan pemisahan proses perhitungan pada tingkat yang berbeda. Optimalisasi berbasis hierarki untuk pemeliharaan infrastruktur kereta api juga telah diusulkan [10]. Namun, skema hierarki ini sangat tergantung pada struktur utama dari masalah masing-masing dan dengan demikian menjadi kurang fleksibel untuk menyelesaikan masalah berskala besar.

Dilatarbelakangi oleh masalah tersebut, makalah ini bertujuan untuk mengembangkan DSS berbasis *Distributed Model Predictive Control* (DMPC) untuk mengatasi masalah beban komputasi dari perencanaan operasi pemeliharaan untuk jaringan infrastruktur kereta api berskala besar.

Secara umum, optimisasi perencanaan pemeliharaan rel dapat dirumuskan sebagai masalah *Mixed-Integer Programming* (MIP). Dalam studi literatur, berbagai teknik dekomposisi berbasis Lagrangian yang ditanamkan dalam *framework* DMPC [11] berhasil dikembangkan untuk menyelesaikan masalah optimisasi yang bersifat nonlinear dan MIP, seperti teknik dekomposisi *Parallel/Serial Augmented Lagrangian* dan *Alternating Direction Method of Multipliers* (ADMM) [12], [13]. Lebih lanjut, diaplikasikan teknik relaksasi kontinu untuk variabel biner dan integer yang terdapat pada masalah optimisasi [14]. Solusi yang dihasilkan menghasilkan batas bawah (*bound*) untuk fungsi objektif asli serta memberikan nilai untuk *warmstart* untuk optimisasi MIP. Selain teknik dekomposisi berbasis Lagrangian, *Distributed Robust Safe But Knowledgeable* (DRSBK) [15] juga diterapkan pada masalah MIP dengan kendala kopling yang *non-convex*. Algoritme ini menggunakan pendekatan pengurangan kopling ketika memecahkan satu subproblem. Dengan cara ini, kopling dapat didekomposisi dan dikomunikasikan kepada semua subproblem yang nantinya akan diselesaikan secara terpisah. Pendekatan terdistribusi yang dibahas dapat dilihat sebagai metode heuristik untuk menyelesaikan masalah MIP karena tidak ada jaminan untuk konvergensi menuju global optimum [16].

Kontribusi utama dari makalah ini adalah pengkajian metode DMPC untuk perencanaan operasi pemeliharaan jalur kereta api yang berskala luas. Fokusnya adalah mengurangi beban atau waktu komputasi dan meningkatkan skalabilitas metode optimisasi untuk menangani permasalahan berskala luas tetapi tetap mendapatkan solusi yang mendekati optimal. Untuk itu, teknik dekomposisi masalah berbasis ADMM dan DRSBK diimplementasikan pada makalah ini. Formulasi masalah optimisasi terpusat dikembangkan dari masalah optimisasi pada [3].

Makalah ini disusun sebagai berikut. Masalah optimisasi pemeliharaan pertama kali dijelaskan di Bagian II. Bagian III membahas pengembangan pendekatan optimisasi terdistribusi. Kemudian, dua studi kasus dibahas dalam Bagian IV. Akhirnya, Bagian V memberikan kesimpulan dan saran dari penelitian ini.

II. PERMASALAHAN OPTIMISASI

Formulasi permasalahan dalam makalah ini dikembangkan berdasarkan model optimisasi yang diajukan pada [3]. Selain itu, beberapa kendala ditambahkan ke masalah optimisasi untuk memasukkan masalah-masalah praktis seperti waktu penutupan trek untuk pemeliharaan, batasan anggaran pemeliharaan, dan pencegahan operasi *renewal* yang terlalu dini.

Ketika dipasang pada rel untuk pertama kali, batu-batu *ballast* memiliki sudut-sudut yang tajam sehingga dapat membentuk fondasi yang kuat untuk menopang rel. Karena trek secara teratur dilewati oleh kereta, batu-batu tersebut akan bergeser secara bertahap, dan kondisinya permukaan batunya menjadi tumpul seiring waktu. Dalam industri kereta api, operasi pemeliharaan rutin dilakukan untuk menjaga unjuk kerja lintasan. Operasi yang umum dilakukan untuk *ballast* adalah *tamping* atau *renewal*.

Tamping dilakukan untuk mengatur perataan lintasan sehingga unjuk kerja lintasan ditingkatkan ke tingkat tertentu. Ketika *tamping* tidak lagi efektif, operasi *renewal* dilakukan untuk sepenuhnya menggantikan *ballast* yang lama. Dinamika degradasi *ballast* dan opsi perawatan yang sesuai dapat dimodelkan secara matematis sebagai berikut.

$$x_{1,i}(k+1) = a_{1,i}x_{1,i}(k) + f_{1,i}(x_i(k), u_i(k)) \quad (1)$$

$$x_{2,i}(k+1) = a_{2,i}x_{2,i}(k) + f_{2,i}(x_i(k), u_i(k))$$

$$x_i(k) = [x_{1,i}(k) \ x_{2,i}(k)]^T$$

dengan variabel status $x_{1,i}(k)$ dan $x_{2,i}(k)$ masing-masing adalah *state* tingkat degradasi trek/unjuk kerja trek secara keseluruhan dan memori degradasi pada *timestep* k untuk segmen trek i . Memori adalah variabel yang digunakan untuk mempertimbangkan fakta bahwa hasil operasi *tamping* tidak dapat membuat unjuk kerja lintasan kembali seperti ketika *ballast* pertama kali dipasang. Selain itu, $a_{1,i}$ adalah laju degradasi unjuk kerja dan $a_{2,i}$ adalah laju degradasi pada memori. Proses degradasi sangat ditentukan oleh parameter ini dan nilainya dapat berbeda untuk setiap segmen trek. Hal ini dikarenakan jalur kereta api adalah sistem parameter

terdistribusi yang berubah seiring waktu dan ruang. Fungsi $f_{1,i}$ dan $f_{2,i}$ adalah fungsi nonkontinu yang didefinisikan sebagai

$$f_{1,i}(x_i(k), u_i(k)) = \begin{cases} 0, & \text{if } u_i(k) = 1 \\ -a_{1,i}x_{1,i}(k) + a_{2,i}x_{2,i}(k) + \alpha, & \text{if } u_i(k) = 2 \\ -a_{1,i}x_{1,i}(k) + h_{\min}, & \text{if } u_i(k) = 3 \end{cases}$$

dan

$$f_{2,i}(x_i(k), u_i(k)) = \begin{cases} 0, & \text{if } u_i(k) = 1 \\ \alpha, & \text{if } u_i(k) = 2 \\ -a_{2,i}x_{2,i}(k) + h_{\min}, & \text{if } u_i(k) = 3 \end{cases}$$

dengan α dan h_{\min} adalah *offset tamping* dan batas minimum degradasi trek. Sistem ini memiliki tiga jenis masukan atau keputusan yang diambil pada setiap *timestep*, yaitu tidak ada operasi pemeliharaan $u_i(k) = 1$; *tamping* $u_i(k) = 2$; atau *renewal* $u_i(k) = 3$. Perhatikan bahwa model yang diusulkan di sini memiliki sedikit perbedaan dari yang ada di [3]. Pertama, variabel memori didefinisikan untuk memiliki laju degradasi $a_{2,i}$. Dengan ini, variabel memori diasumsikan berevolusi secara eksponensial dari waktu ke waktu.

Karena penggunaan masukan yang bersifat integer, sistem ini dapat dikategorikan sebagai sistem nonlinear. Salah satu cara agar variabel *integer* dalam (1) dapat diimplementasikan dalam komputer adalah dengan menggunakan model *Mixed Logical Dynamical* (MLD). Tiga opsi masukan pemeliharaan dapat diwakili oleh dua variabel biner $\delta_{1,i}(k)$ dan $\delta_{2,i}(k)$ untuk masing-masing operasi *renewal* dan *tamping* di bagian trek i . Tabel konversi diberikan dalam Tabel I.

TABEL I
KONVERSI VARIABEL MASUKAN INTEGER KE BINER

$u_i(k)$	$\delta_{1,i}(k)$	$\delta_{2,i}(k)$
1	0	0
2	0	1
3	1	0

Opsi untuk melakukan operasi *tamping* dan *renewal* pada waktu k yang sama untuk suatu segmen i dihilangkan dengan menggunakan kendala berikut.

$$\delta_{1,i}(k) + \delta_{2,i}(k) \leq 1. \tag{2}$$

Karena terdapat variabel-variabel biner, maka model dinamis menjadi nonlinear, yang dirumuskan sebagai berikut.

$$\begin{aligned} x_{1,i}(k+1) &= a_{1,i}x_{1,i}(k) + \delta_{1,i}(k)(-a_{1,i}x_{1,i}(k) + h_{\min}) + \\ &\quad \delta_{2,i}(k)(-a_{1,i}x_{1,i}(k) + a_{2,i}x_{2,i}(k) + \alpha) \\ x_{2,i}(k+1) &= a_{2,i}x_{2,i}(k) + \delta_{2,i}(k)\alpha + \\ &\quad \delta_{1,i}(k)(-a_{2,i}x_{2,i}(k) + h_{\min}) \end{aligned}$$

Untuk memfasilitasi pemrosesan variabel biner, beberapa variabel bantu ditambahkan pada model dinamik, sehingga menghasilkan persamaan linear sebagai berikut.

$$\begin{aligned} \begin{bmatrix} x_{1,i}(k+1) \\ x_{2,i}(k+1) \end{bmatrix} &= \begin{bmatrix} a_{1,i} & 0 \\ 0 & a_{2,i} \end{bmatrix} \begin{bmatrix} x_{1,i}(k) \\ x_{2,i}(k) \end{bmatrix} + \begin{bmatrix} h_{\min} & \alpha \\ h_{\min} & \alpha \end{bmatrix} \\ &\quad + \begin{bmatrix} -a_{1,i} & -a_{1,i} & a_{2,i} & 0 \\ 0 & 0 & 0 & a_{2,i} \end{bmatrix} \begin{bmatrix} z_{1,i}(k) \\ z_{2,i}(k) \\ z_{3,i}(k) \\ z_{4,i}(k) \end{bmatrix} \end{aligned} \tag{3}$$

dengan variabel pendukung dideskripsikan sebagai berikut.

$$\begin{aligned} z_{1,i}(k) &= \delta_{1,i}(k)x_{1,i}(k) & z_{2,i}(k) &= \delta_{2,i}(k)x_{1,i}(k) \\ z_{3,i}(k) &= \delta_{2,i}(k)x_{2,i}(k) & z_{4,i}(k) &= \delta_{1,i}(k)x_{2,i}(k) \end{aligned} \tag{4}$$

dan vektor pendukung dapat didefinisikan sebagai $z_i(k) = [z_{1,i}(k) \ z_{2,i}(k) \ z_{3,i}(k) \ z_{4,i}(k)]^T$. Untuk dapat diimplementasikan dalam program, (4) diformulasikan menggunakan empat kendala linear berikut.

$$\begin{aligned} z_{p,i}(k) &\leq h_{\max}\delta_{l,i}(k) \\ z_{p,i}(k) &\geq h_{\min}\delta_{l,i}(k) \\ z_{p,i}(k) &\leq x_{j,i}(k) - h_{\min}(1 - \delta_{l,i}(k)) \\ z_{p,i}(k) &\geq x_{j,i}(k) - h_{\max}(1 - \delta_{l,i}(k)) \end{aligned} \tag{5}$$

untuk $p \in \{1,2,3,4\}$ dan $l \in \{1,2\}$ dan h_{\max} adalah batas tingkatan degradasi maksimum pada trek. Dengan ini, akan terdapat enam belas persamaan untuk setiap segmen trek.

Selanjutnya, $V_i(k)$ berisi variabel biner dan pendukung sebagai berikut.

$$V_i(k) = [\delta_{1,i}(k) \ \delta_{2,i}(k) \ z_i(k)]^T.$$

Kemudian model *state space* dapat dirumuskan dalam bentuk berikut.

$$x_i(k+1) = A_i x_i(k) + B_i V_i(k) \tag{6}$$

dengan A_i dan B_i adalah matriks paramater yang dapat dilihat lebih lanjut pada [3].

Kondisi awal dan laju degradasi untuk setiap segmen trek ditentukan menurut data dari studi kasus. Hal ini bertujuan untuk menggambarkan distribusi dinamika yang tersebar sepanjang jalur trek kereta. Laju degradasi ini diasumsikan konstan (*time-invariant*). Kendala terkait meliputi kondisi awal.

$$x_{1,i}(0) = x_{i,0}^1, \quad x_{2,i}(0) = x_{i,0}^2 \tag{7}$$

dengan $x_{i,0}^1$ dan $x_{i,0}^2$ adalah nilai awal untuk variabel tingkat degradasi dan memori *offset*, yang umumnya diambil dari hasil pengukuran di lapangan. Nilai kedua variabel hanya sama saat *ballast* pertama kali dipasang atau ketika dilakukan pemeliharaan pada segmen trek tersebut. Selanjutnya, kendala berikut ini digunakan untuk memastikan bahwa unjuk kerja trek selalu dalam kisaran yang dapat diterima.

$$h_{\min} \leq x_{1,i}(k) \leq h_{\max}, \quad h_{\min} \leq x_{2,i}(k) \leq h_{\max}. \tag{8}$$

Operasi pembaruan hanya dapat diizinkan setelah memori *offset* dianggap tinggi, yang dimodelkan dalam kendala berikut.

$$x_{2,i}(k) - h_r \geq (r_i - 1)h_{\max} \quad (9)$$

$$r_i - \delta_1(k) \geq 0 \quad (10)$$

dengan r_i dan h_r adalah indikator untuk *switching* di bagian track i dan ambang pencegahan untuk memungkinkan operasi pembaruan.

Menurut pengaruhnya terhadap segmen trek, kendala-kendala yang didefinisikan sebelum ini dapat dikategorikan sebagai kendala individu. Kendala individu hanya memengaruhi masing-masing individu segmen trek. Selain kendala yang efeknya bersifat individual, terdapat juga kendala kopling, yang secara langsung memengaruhi sebagian atau semua segmen trek. Kendala kopling yang pertama adalah maksimum dana yang tersedia untuk melaksanakan operasi pemeliharaan dalam rentang waktu tertentu. Kendala ini penting untuk dipertimbangkan karena biaya operasi pemeliharaan infrastruktur sangat mahal dan dana yang dianggarkan sangat terbatas. Kendala ini dibuat terpisah untuk masing-masing operasi dan dapat ditulis sebagai berikut.

$$\sum_{k=1}^T \sum_{i=1}^N \delta_{2,i}(k) \leq g_t \quad (11)$$

$$\sum_{k=1}^T \sum_{i=1}^N \delta_{1,i}(k) \leq g_r \quad (12)$$

dengan g_t dan g_r adalah jumlah maksimum operasi *tamping* dan *renewal* yang diizinkan untuk dilaksanakan dalam horizon prediksi T .

Selanjutnya, pelaksanaan operasi harus dialokasikan pada rentang waktu tertentu, agar tidak mengganggu lalu lintas kereta pada jam sibuk. Pada praktiknya, waktu maksimal operasi pemeliharaan ini kurang dari tujuh jam dan hanya boleh dilaksanakan pada malam hari di akhir pekan [1]. Waktu maksimum penutupan jalur kereta selama dilangsungkan operasi pemeliharaan rutin ini dilambangkan sebagai t_{\max} . Batasan waktu ini berlaku untuk masing-masing operasi *tamping* dan *renewal*. Kendala ini dapat ditulis sebagai berikut [5], [17].

$$\sum_{i \in P(k)} t_{t1} \delta_{2,i}(k) + t_{t2} (1 - \delta_{2,i}(k)) + t_{tb} I_{tb,i} + t_{ta} I_{ta,i} \leq t_{\max} \quad (13)$$

$$\sum_{i \in P(k)} t_{r1} \delta_{1,i}(k) + t_{r2} (1 - \delta_{1,i}(k)) + t_{rb} I_{rb,i} + t_{ra} I_{ra,i} \leq t_{\max} \quad (14)$$

dengan t_{t1} dan t_{t2} adalah waktu saat melangsungkan operasi pemeliharaan dan waktu tempuh bagi mesin *tamping* untuk mencapai segmen trek i dari posisi tertentu. Set $P(k)$ berisi segmen-segmen trek yang akan mendapat tindakan pemeliharaan dari mesin *tamping* pada *timestep* k . Kendala yang serupa juga berlaku untuk variabel t_{r1} dan t_{r2} untuk operasi *renewal*. Diasumsikan juga bahwa dalam setiap operasi, mesin bergerak searah dari titik awal trek menuju titik akhir di ujung trek. Pada praktiknya, mesin pemeliharaan ini bekerja dalam mode operasi dan perjalanan. Untuk mengganti mode, dibutuhkan waktu transisi (sebelum dan sesudah) [17],

yang dilambangkan dengan t_{tb}, t_{ta} untuk *tamping* dan t_{rb}, t_{ra} untuk *renewal*. Namun, hal ini hanya berlaku di awal dan akhir operasi, serta jalur yang tidak berurutan. Untuk itu, ditambahkan dua variabel indikator untuk setiap segmen dan setiap operasi, yakni $I_{tb,i}, I_{ta,i}, I_{rb,i}, I_{ra,i}$. Karena kendala (13) dan (14) memengaruhi jadwal pemeliharaan di semua atau beberapa segmen trek, kedua kendala tersebut termasuk sebagai kopling.

Variabel keadaan optimal \bar{X} dan variabel keputusan \bar{V} untuk semua bagian trek di cakrawala prediksi dapat diperoleh dengan meminimalkan fungsi tujuan berikut.

$$J(\bar{X}, \bar{V}) = \sum_{i=1}^N \sum_{k=1}^T Q x_i(k) + \lambda R V_i(k) = \sum_{i=1}^N J_{\text{ind}}(\bar{X}_i, \bar{V}_i) \quad (15)$$

dengan Q dan R adalah matriks dengan hanya entri nonnegatif, sementara \bar{X}_i dan \bar{V}_i adalah variabel keadaan dan masukan atas prediksi cakrawala untuk bagian trek i . Variabel *state* $x_i(k)$ pada (15) dapat diganti dengan menggunakan teknik yang sama seperti pada [3], sehingga hanya $V_i(k)$ sebagai satu-satunya variabel keputusan. Selanjutnya, masalah optimisasi dapat ditulis dalam bentuk berikut.

$$\underset{\bar{V}}{\text{minimize}} J(\bar{V}) = \sum_{i=1}^N J_{\text{ind}}(\bar{V}_i) \quad (16)$$

$$\text{subject to} \quad E \bar{V} \leq g_{\text{ind}}$$

$$\sum_{i=1}^N F_i \bar{V}_i \leq g_{\text{coup}} \quad (17)$$

dengan E dan g_{ind} adalah matriks parameter dan vektor yang terkait dengan semua kendala individu, sementara F_i dan g_{coup} adalah matriks parameter dan vektor yang terkait dengan kendala kopling. Pemisahan secara eksplisit ini berguna untuk implementasi sistem kendali prediktif terdistribusi. Selain itu, tidak ada variabel kopling antara subproblem dalam fungsi objektif. Perlu dicatat bahwa optimisasi masalah optimisasi ini dapat dikategorikan sebagai *NP-hard problems* dan *Mixed-Integer Linear Programming* (MILP) karena penggunaan dua jenis variabel, yakni kontinu dan diskret di dalam persamaan.

III. METODE DISTRIBUTED MODEL PREDICTIVE CONTROL (DMPC)

Pada makalah ini, dua metode dekomposisi diimplementasikan menggunakan *framework* DMPC, yaitu ADMM dan DRSBK. Selain itu, modifikasi pada kedua metode tersebut untuk menyelesaikan masalah optimisasi yang diajukan juga dibahas. Berikut penjelasan lengkap mengenai cara kerja kedua metode tersebut.

A. Alternating Direction Method of Multipliers (ADMM)

Tidak seperti metode dekomposisi Lagrangian lainnya, ADMM menggunakan teknik bolak-balik (*alternating*). Teknik ini dapat memisahkan bagian kuadrat pada fungsi objektif untuk diselesaikan secara individual dengan membuat

konstan variabel keputusan dari subproblem lainnya [18]. Dengan cara ini, semua subproblem diselesaikan secara berurutan dan menggunakan nilai yang telah diperbarui dari subproblem yang diselesaikan sebelumnya. Bentuk *unscaled* ADMM dipilih untuk kemudahan implementasi [19].

Untuk dapat mengimplementasikan ADMM, bentuk masalah optimisasi terpusat pada (16) dan (17) harus ditransformasikan menjadi bentuk *augmented Lagrangian* [20]. Persyaratan lain adalah bahwa setiap kendala kopling harus dikonversi ke dalam persamaan *equality* [21]. Dengan demikian, vektor \bar{S} didefinisikan sebagai variabel *slack* untuk kendala kopling, baik untuk operasi *tamping* maupun *renewal*, di atas cakrawala prediksi. Lagrangian yang diperbesar dapat ditulis sebagai berikut.

$$\begin{aligned} L(\bar{V}, \bar{S}, \gamma) &= \sum_{i=1}^N J_{\text{ind}}(\bar{V}_i) + \sum_{k=1}^T s(k) \\ &+ \gamma \left(\sum_{i=1}^N F_i \bar{V}_i + \sum_{k=1}^T F_s s(k) - g_{\text{coup}} \right) + \\ &\frac{\rho}{2} \left\| \sum_{i=1}^N F_i \bar{V}_i + \sum_{k=1}^T F_s s(k) - g_{\text{coup}} \right\|_2^2 \end{aligned} \quad (18)$$

$$\text{subject to } E\bar{V} \leq g_{\text{ind}} \quad (19)$$

dengan F_i dan F_s adalah matriks parameter dari kendala kopling untuk masukan dan variabel *slack*. Selain itu, $s(k)$ adalah vektor *slack* untuk setiap langkah waktu k , sehingga $\bar{S} = [s^T(1) \dots s^T(T)]^T$. Selanjutnya, persamaan Lagrangian untuk *dual problem* dapat ditulis sebagai berikut.

$$q(\gamma) = \inf_{\bar{V}, \bar{S}} (L(\bar{V}, \bar{S}, \gamma)). \quad (20)$$

Kemudian, variabel ganda dapat diperbarui dengan menyelesaikan masalah maksimalisasi fungsi di atas [22]. Metode subgradien dalam praktiknya digunakan untuk melakukan pembaruan variabel *dual*.

Setelah semua *subproblem* diselesaikan, hasilnya akan dikumpulkan oleh koordinator untuk memperbarui nilai dari variabel *dual*. Kehadiran koordinator juga berarti bahwa nilai dari satu *dual* digunakan untuk menentukan harga umum (*common price*) untuk semua *subproblem*.

B. Modifikasi untuk Metode ADMM

Untuk menyelesaikan masalah optimisasi yang dirumuskan pada makalah ini, beberapa modifikasi diperlukan pada metode standar ADMM. Sifat kendala kopling yang tidak mulus (*non-smooth*) (13) dan (14) disebabkan oleh semua variabel keputusan dalam persamaan kendala bersifat diskret. Metode berbasis gradien rekursif akan bergantian melompat antara titik-titik diskret ini. Akibatnya, iterasi pembaruan ganda subgradien mungkin tidak dapat konvergen atau bahkan tidak dapat menuju wilayah yang layak. Salah satu cara untuk mengatasi masalah ini adalah dengan menerapkan teknik relaksasi kontinu untuk variabel keputusan biner, sedemikian rupa sehingga model MILP yang kompleks menjadi model *Linear Programming* (LP) yang lebih sederhana dan mudah untuk diselesaikan [10]. Selain itu, nilai fungsi objektif yang

diperoleh relaksasi LP tersebut dapat digunakan sebagai batas untuk optimisasi MILP pada langkah berikutnya:

$$J_{\text{ind}}(\bar{V}_i^{\text{LP}}) \leq J_{\text{ind}}(\bar{V}_i).$$

Karena konvergensi residu variabel *primal* tidak dapat dijamin, maka algoritme mengambil solusi terbaik yang ditemukan selama seluruh prosedur iterasi. Selanjutnya, untuk mengakhiri iterasi, dua persyaratan harus terpenuhi. Pertama, vektor residual dari kopling diperiksa dalam setiap iterasi j untuk menilai kondisi kelayakan telah tercapai atau belum [12]. Vektor residual pada iterasi j , $r^{(j)}$, didefinisikan sebagai berikut.

$$r^{(j)} = g_{\text{coup}} - \left(\sum_{i=1}^N F_i \bar{V}_i^{(j)} + \sum_{k=1}^T F_s s(k)^{(j)} \right).$$

Kondisi kelayakan ditunjukkan oleh nilai-nilai nonpositif pada seluruh baris vektor atau $r^{(j)} < 0$. Hal ini untuk menjamin kendala kopling dapat selalu visibel. Selanjutnya, jika perbedaan antara fungsi tujuan nilai iterasi saat ini dan fungsi objektif terbaik yang dihasilkan di bawah nilai ambang batas, maka iterasi dapat diakhiri.

C. Distributed Robust Safe but Knowledgeable (DRSBK)

DRSBK diterapkan pada optimisasi terdistribusi koordinasi multikendaraan atau multiagen [20]. Algoritme ini awalnya ditujukan untuk masalah MILP. Konsepnya adalah sebagai berikut. Algoritme ini mengurangi alokasi sumber daya dalam kendala kopling untuk setiap subproblem, sehingga kendala kopling sama sekali tidak dimasukkan ke dalam fungsi objektif berbentuk Lagrangian. Hal ini dapat diilustrasikan dengan persamaan berikut.

$$\text{minimize}_{\bar{V}_i} J_{\text{ind}}(\bar{V}_i) \quad (21)$$

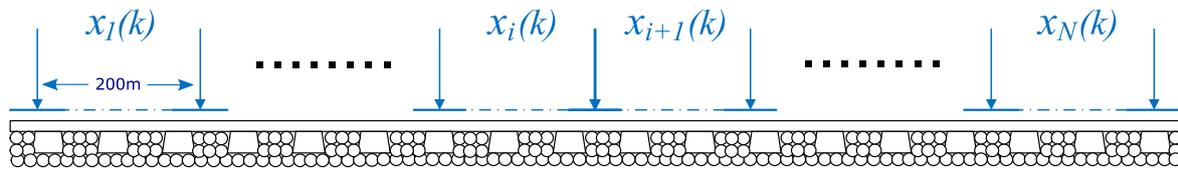
$$\text{subject to (19)}$$

$$F_i \bar{V}_i \leq g_{\text{coup}} - \sum_{j=1, j \neq i}^{N-1} F_j \bar{V}_j \quad (22)$$

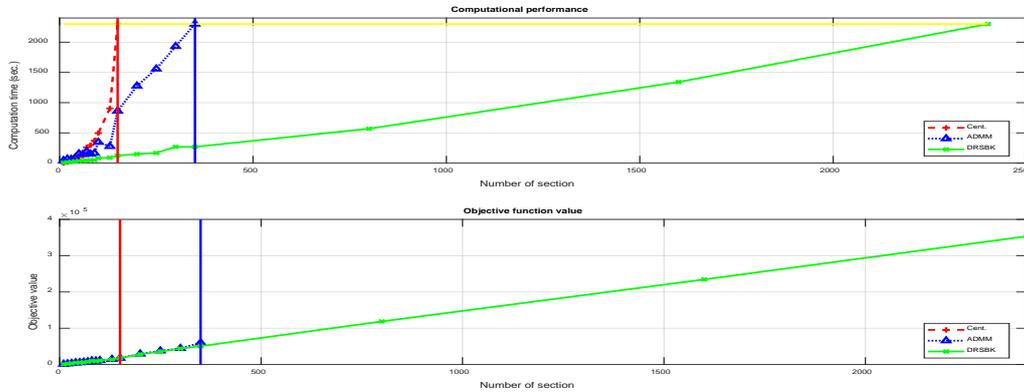
dengan \bar{V}_j berisi variabel keputusan untuk bagian trek i di atas cakrawala prediksi. Set kedua kendala di (22) adalah kopling dengan sumber daya yang berkurang, yaitu sisa waktu penutupan pemeliharaan yang tersedia selama cakrawala prediksi. Proses reduksi dapat dilakukan dengan memperbaiki keputusan dari subproblem lain. Dengan cara ini, kendala kopling dapat dipisahkan, sehingga masalah dapat diselesaikan secara individual dan berurutan. Salah satu keuntungan dari menetapkan kopling ke dalam individu kendala adalah bahwa solusi yang layak jauh lebih mudah diperoleh. Selain itu, tugas koordinator di DRSBK hanya memeriksa kelayakan solusi yang dihasilkan.

D. Modifikasi untuk metode DRSBK

Solusi dari algoritme standar DRSBK yang noniteratif memungkinkan solusinya terjebak dalam titik optimal lokal atau bahkan solusinya tidak visibel. Oleh karena itu, algoritme ini dimodifikasi sedemikian rupa sehingga urutan subproblem



Gbr. 2 Ilustrasi pembagian segmen trek.



Gbr. 3 Perbandingan waktu komputasi dan nilai objektif dari semua metode.

TABEL II
PERBANDINGAN NILAI RATA-RATA PARAMETER DARI SEMUA METODE

Parameter/algortme	Centralized	ADMM	DRSBK
Waktu komputasi	965,22571	247,26976	71,32943
Nilai fungsi objektif	9.677,17	10.042,61	9.677,06
Total degradasi trek	7.744,2	8.219,61	7.744,09
Jumlah <i>tamping</i>	124,3	92,3	124,3
Jumlah <i>renewal</i>	2,3	3	2,3
Nilai objektif ternormalisasi	-	-6,490%	-0,001%

yang akan diproses di setiap iterasi dihasilkan secara acak. Jika keluaran dari *solver* menunjukkan hasil iterasi yang tidak layak, urutan di iterasi selanjutnya akan dihasilkan lagi secara acak. Metode pemeriksaan kelayakan solusi DRSBK berbeda dari ADMM, yakni dengan menjumlahkan semua nilai dari variabel indikator kelayakan dari masing-masing subproblem. Solusinya dianggap layak jika jumlah total nilai indikator sama dengan jumlah segmen trek. Selain itu, kriteria pemberhentian iterasinya serupa dengan algoritme ADMM di atas.

IV. STUDI KASUS

Pada bagian ini, metode konvensional *centralized MPC* dan dua metode DMPC (ADMM dan DRSBK) disimulasikan dalam dua jenis kasus. Parameter keluarannya kemudian dibandingkan dan dianalisis. Semua simulasi dilakukan pada komputer serba guna dengan prosesor Intel Core-i5 dan RAM 8GB. Semua masalah LP dan MILP diselesaikan oleh *solver* Gurobi 7.5, yang dipanggil menggunakan MATLAB R2017a. Selain itu, asumsi dan pengaturan umum berikut dipertimbangkan.

- Langkah waktu untuk intervensi pemeliharaan adalah satu bulan. *Prediction* dan *control horizon* ditetapkan menjadi 12 bulan. Sedangkan *implementation horizon* ditetapkan menjadi enam bulan.

- Model bersifat deterministik, yakni tidak ada proses stokastik atau *disturbance* dari aktivitas operasi kereta api pada umumnya, seperti operasi pemeliharaan reaktif. Selain itu, bobot *trade-off* pada (6) dipilih menjadi $\lambda = 10$.
- Peralatan pemeliharaan dan personel diasumsikan selalu tersedia sepanjang waktu. Studi kasus melibatkan jalur kereta api tunggal, yang terdiri atas sejumlah bagian jalur. Setiap panjang bagian trek adalah 200 m. Tata letak digambarkan pada Gbr. 2.
- Parameter (kondisi awal dan laju degradasi) untuk setiap bagian lintasan dapat berbeda, sesuai dengan skenario yang sesuai. Kondisi awal variabel level degradasi dan memori diasumsikan berbeda. Satu skenario biasanya menyajikan parameter yang berbeda untuk setiap segmen trek, yang dihasilkan secara acak sesuai distribusi Gaussian. Dengan cara ini, korelasi spasial antar segmen trek mempertimbangkan kondisi bahwa sebagian segmen trek memiliki tingkat lebih cepat daripada segmen yang lain [23]. Selain itu, laju degradasi diasumsikan konstan dalam rentang waktu simulasi.

A. Eksperimen I: Pengujian pada Kasus Jumlah Segmen Trek Berskala Besar

Dalam eksperimen ini, setiap pendekatan diuji pada kasus penjadwalan pemeliharaan dengan target sebuah jalur kereta

yang dibagi menjadi 140 segmen. Jumlah variabel tersebut dianggap besar dan memberatkan komputasi metode *centralized*. Perbandingan hasil simulasi untuk setiap metode disajikan pada Tabel II. Nilai pada tabel tersebut menyajikan nilai rata-rata dari sepuluh skenario yang berbeda.

Dari hasil eksperimen terlihat bahwa solusi DRSBK memiliki waktu yang paling cepat di antara metode lainnya. Selisih waktunya pun sangat jauh dibanding metode *centralized*. ADMM juga mampu memperkecil kecepatan komputasi di bawah metode *centralized*. Hal ini sesuai dengan teori, yakni masalah optimisasi berskala besar yang dibagi-bagi menjadi lebih kecil akan lebih mudah diselesaikan.

Yang menarik di sini adalah walaupun DRSBK dapat menyelesaikan permasalahan optimisasi dengan lebih cepat, kualitas solusi yang dihasilkan hampir sama dengan metode *centralized*, dengan rata-rata nilai objektif ternormalisasinya kurang dari 0,01%. Selain itu, secara umum solusi DRSBK juga lebih baik daripada solusi ADMM. Dalam sistem MLD dengan masukan diskret yang jumlahnya sangat terbatas, DRSBK unggul karena tidak perlu mengubah fungsi objektif sama sekali, sehingga mampu memproses masalah dengan lebih cepat.

B. Eksperimen II: Pengujian dengan Jumlah Segmen Naik Secara Bertahap

Selanjutnya, eksperimen kedua menyajikan perbandingan hasil simulasi dari ketiga metode dengan skema jumlah segmen trek yang dinaikkan secara bertahap. Bila waktu komputasi dari suatu metode sudah mendekati batas waktu metode *centralized* (dalam percobaan ini adalah sekitar 2.300 detik), maka eksperimen metode tersebut akan dihentikan. Terlihat dalam Gbr. 3, pada masalah berskala kecil, metode *centralized* dapat menghasilkan solusi lebih cepat dibanding metode DMPC. Namun, pada $N = 20$, DRSBK mampu membalik kondisi, menjadi lebih cepat. Hal ini disusul pula oleh ADMM pada $N = 70$. Selain itu, semakin besar jumlah segmen treknya, maka jarak waktunya pun akan semakin lebar. Hal ini dikarenakan permasalahan ini bersifat *NP-hard*.

Dalam menyelesaikan permasalahan optimisasi penjadwalan pemeliharaan trek kereta berskala rendah ke besar, DRSBK jauh lebih baik daripada ADMM. ADMM hanya mampu menangani sekitar 350 segmen, sedangkan DRSBK mampu mencapai 2.400 segmen dalam rentang waktu yang sama.

V. KESIMPULAN

Dalam makalah ini, dua teknik dekomposisi dalam *framework* DMPC telah diujikan untuk optimisasi penjadwalan operasi pemeliharaan trek kereta api berskala besar. Metode pertama, yakni ADMM, bekerja berdasarkan teori dualitas Lagrangian. Untuk mengarahkan iterasi menuju daerah solusi yang visibel, metode ini dimodifikasi menggunakan pemrosesan dua langkah (LP dan MILP). Di samping itu, metode DRSBK diimplementasikan. Untuk menghindari solusi yang nonvisibel dan untuk mendapatkan solusi yang lebih optimal dengan tetap mempertahankan kesederhanaan, algoritme diperluas menjadi berulang dan urutan subproblem yang diproses dihasilkan secara acak.

Dalam studi kasus, ditunjukkan bahwa kedua metode DMPC dapat memecahkan masalah optimisasi lebih cepat daripada pendekatan *centralized* dengan jumlah segmen trek di atas 130. DRSBK adalah metode tercepat, yang juga mampu menghasilkan solusi yang hampir mendekati solusi *centralized*. Selain itu, ADMM juga bekerja lebih cepat dari *centralized* untuk menyelesaikan permasalahan optimisasi di atas, tetapi solusinya masih cukup bagus.

Untuk penelitian lebih lanjut, metode dapat diuji menggunakan data riil hasil pengukuran di lapangan. Memasukkan perilaku degradasi stokastik ke dalam model juga digunakan untuk mempertimbangkan ketidakpastian dalam operasi pemeliharaan rel kereta api. Selain itu, metode ini dapat dikembangkan menjadi DMPC hierarki untuk memfasilitasi waktu yang berbeda skala dalam perencanaan pemeliharaan atau kondisi kesempatan terbatas (*chance constrained*).

REFERENSI

- [1] C. Esvelde, *Modern Railway Track*. Zaltbommel, The Netherlands: MRT-Productions, 2001.
- [2] M. Wen, R. Li, dan K. B. Salling, "Optimization of Preventive Condition-Based Tamping for Railway Tracks," *European Journal of Operational Research*, Vol. 252, No. 2, hal.455–465, 2016.
- [3] Z. Su, A. Núñez, A. Jamshidi, S. Baldi, Z. Li, R. Dollevoet, dan B. De Schutter, "Model Predictive Control for Maintenance Operations Planning of Railway Infrastructures," *Proceedings of 6th International Conference Computational Logistics (ICCL)*, 2015, hal. 673–688.
- [4] R. Li dan R. Roberti, "Optimal Scheduling of Railway Track Possessions in Large-Scale Projects with Multiple Construction Works," *Journal of Construction Engineering and Management*, Vol. 143, No. 6, hal. 1-11, 2017.
- [5] S.M. Famurewa, T. Xin, M. Rantatalo, dan U. Kumar, "Optimisation of Maintenance Track Possession Time: A Tamping Case Study," *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, Vol. 229, No. 1, hal. 12–22, 2015.
- [6] E. Gustavsson, "Scheduling Tamping Operations on Railway Tracks Using Mixed Integer Linear Programming," *EURO Journal on Transportation and Logistics*, Vol. 4, No. 1, hal. 97–112, 2015.
- [7] F. Guo, C. Wen, J. Mao, G. Li, dan Y. D. Song, "A Distributed Hierarchical Algorithm for Multi-Cluster Constrained Optimization," *Automatica*, Vol. 77, hal. 230–238, 2017.
- [8] Z. Su, A. Núñez, S. Baldi, dan B. De Schutter, "Model Predictive Control for Rail Condition-Based Maintenance: A Multilevel Approach," *Proceedings of the 19th IEEE Intelligent Transportation Systems Conference (ITSC 2016)*, 2016, hal. 354–359.
- [9] Z. Su, A. Jamshidi, A. Núñez, S. Baldi, dan B. De Schutter, "Multi-level Condition-based Maintenance Planning for Railway Infrastructures – A Scenario-based Chance-constrained Approach," *Transportation Research Part C: Emerging Technologies*, Vol. 84, hal. 92–123, 2017.
- [10] K. Verbert, B. De Schutter, dan R. Babuška, "Timely Condition-based Maintenance Planning for Multi-component Systems," *Reliability Engineering & System Safety*, Vol. 159, hal. 310–321, 2017.
- [11] J.M. Maestre dan R.R. Negenborn, *Distributed Model Predictive Control Made Easy*, Dordrecht, The Netherlands: Springer Publishing Company, Inc., 2013.
- [12] M.J. Feizollahi, M. Costley, S. Ahmed, dan S. Grijalva, "Large-scale Decentralized Unit Commitment," *International Journal of Electrical Power and Energy Systems*, Vol. 73, hal. 97–106, 2015.
- [13] C. He, L. Wu, T. Liu, dan M. Shahidehpour, "Robust Co-optimization Scheduling of Electricity and Natural Gas Systems via ADMM," *IEEE Transactions on Sustainable Energy*, Vol. 8, No. 2, hal. 658–670, 2017.
- [14] S. Sebastio, G. Gnecco, dan A. Bemporad, "Optimal Distributed Task Scheduling in Volunteer Clouds," *Computers & Operation Research*,

- Vol. 81, No. C, hal. 231–246, May 2017.
- [15] Y. Kuwata dan J.P. How, “Cooperative Distributed Robust Trajectory Optimization Using Receding Horizon MILP,” *IEEE Transactions on Control Systems Technology*, Vol. 19, No. 2, hal. 423–431, 2011.
- [16] S. Boyd, N. Parikh, E. Chu, B. Peleato, dan J. Eckstein, “Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers,” *Foundations and Trends in Machine Learning*, Vol. 3, No. 1, hal. 1–122, 2011.
- [17] R. Li, K.B. Salling, A. Zoeteman, dan A.R.M. Wolfert, “Preventive Condition-based Tamping for Railway Tracks: A Decision Support Model,” *Proceedings of the Fifth International Symposium on Life-Cycle Civil Engineering (IALCCE 2016)*, 2016, hal. 355.
- [18] L. Li, R.R. Negenborn, dan B. De Schutter, “Distributed Model Predictive Control for Cooperative Synchronodal Freight Transport,” *Transportation Research Part E: Logistics and Transportation Review*, Vol. 105, hal. 240–260, 2017.
- [19] S. Boyd, N. Parikh, B.P.E Chu, dan J. Eckstein, “Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers,” *Foundations and Trends in Machine Learning*, Vol. 3, No. 1, hal. 1–122, 2011.
- [20] B. Yang dan M. Johansson, “Distributed Optimization and Games: A Tutorial Overview,” in *Networked Control Systems*, A. Bemporad, M. Heemels, and M. Johansson, Eds. London, UK: Springer London, 2010, hal. 109–148.
- [21] S. Sebastio, G. Gnecco, dan A. Bemporad, “Optimal Distributed Task Scheduling in Volunteer Clouds,” *Comput. Oper. Res.*, Vol. 81, No. C, hal. 231–246, May 2017.
- [22] R.R. Negenborn, B. De Schutter, dan J. Hellendoorn, “Multi-agent Model Predictive Control for Transportation Networks: Serial Versus Parallel Schemes,” *Engineering Applications of Artificial Intelligence*, Vol. 21, No. 3, hal. 353–366, Apr. 2008.
- [23] I.A. khoy, P.-O. Larsson-Kräik, A. Nissen, dan U. Kumar, “Cost-Effective Track Geometry Maintenance Limits,” *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, Vol. 230, No. 2, hal. 611–622, 2016.