

Implementasi Model Penjadwalan *Job-Shop* dalam Masalah Penjadwalan Kereta Api Jalur Tunggal dengan Pendekatan *Constraint Programming*

Fajar Yuliawan – NIM: 13503022

*Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
Email: fajar_42nd@yahoo.com*

Abstrak

Masalah penjadwalan kereta api jalur tunggal dapat diselesaikan dengan pendekatan *Constraint Programming*. Hal ini dilakukan dengan memodelkan masalah sebagai sebuah *Constraint Satisfaction Problem (CSP)* kemudian menyelesaikan *CSP* tersebut dengan teknik-teknik pencarian solusi *CSP*. Dalam makalah ini, *CSP* yang digunakan untuk memodelkan masalah penjadwalan kereta api jalur tunggal adalah masalah penjadwalan *Job-Shop*. Selanjutnya, algoritma yang digunakan dalam pencarian solusi adalah *Hill-Climbing*. Dalam masalah penjadwalan kereta api, solusi yang dimaksud adalah jadwal perjalanan kereta api yang memenuhi semua aturan yang diajukan.

Makalah ini membahas aturan-aturan umum dalam penjadwalan kereta api jalur tunggal, pemodelannya sebagai masalah penjadwalan *Job-Shop* dan penyelesaiannya dengan *Hill-Climbing*. Selanjutnya, analisis, perancangan dan implementasi dilakukan untuk menghasilkan sebuah perangkat lunak penjadwalan kereta api jalur tunggal *Kimspoor Scheduler*. Perangkat lunak ini diimplementasikan dengan paradigma pemrograman berorientasi objek menggunakan bahasa pemrograman Java. Sistem operasi yang digunakan dalam pengembangan adalah *Microsoft Windows XP Professional*.

Perangkat lunak *Kimspoor Scheduler* dapat digunakan untuk memasukkan data perjalanan kereta api, menampilkan data yang telah dimasukkan, melakukan penjadwalan dan menampilkan hasil penjadwalan. Pengujian penjadwalan dilakukan dengan menggunakan data perjalanan kereta api di Indonesia yang diperoleh dari PT Kereta Api (Persero). Dalam setiap pengujian yang dilakukan, perangkat lunak selalu dapat menemukan jadwal yang memenuhi semua aturan umum perjalanan kereta api yang diajukan dalam makalah. Keluaran perangkat lunak yang berupa jadwal perjalanan kereta api tersebut ditampilkan dengan menggunakan representasi diagram ruang-waktu.

Kata Kunci: masalah penjadwalan kereta api jalur tunggal, *Constraint Programming*, *Constraint Satisfaction Problem*, masalah penjadwalan *Job-Shop*, *Hill-Climbing*.

1 Pendahuluan

Kereta api merupakan salah satu alat transportasi utama di seluruh dunia. Di Indonesia dan banyak negara lain, kereta api menggunakan jalur tunggal, artinya satu jalur kereta api digunakan untuk dua arah yang berbeda. Karena terdapat banyak kereta api yang menggunakan jalur atau rute yang sama, penjadwalan kereta api menjadi satu hal yang mutlak diperlukan. Namun pembuatan jadwal kereta api bukanlah masalah yang mudah karena terdapat banyak aturan atau batasan (*constraints*) yang harus dipenuhi, misalnya aturan penyusulan (*overtaking rule*), aturan persilangan (*crossing rule*), aturan *headway*, kecepatan maksimal yang diperbolehkan di suatu jalur, dan sebagainya [6].

Salah satu pendekatan yang dapat digunakan untuk menyelesaikan masalah penjadwalan kereta api jalur tunggal adalah pendekatan *Constraint Programming (CP)*. *CP* adalah metode penyelesaian masalah yang terdiri atas dua langkah utama, yaitu *modelling* dan *resolution* [2]. *Modelling* dilakukan dengan cara merepresentasikan masalah yang diberikan sebagai sebuah *Constraint Satisfaction Problem (CSP)*. Selanjutnya, proses pencarian solusi *CSP* dilakukan pada langkah *resolution* dengan menggunakan *constraint solver*.

Dalam masalah penjadwalan kereta api jalur tunggal, salah satu *CSP* yang dapat digunakan untuk pemodelan adalah masalah penjadwalan *Job-Shop*. Pemodelan dilakukan dengan menganggap perjalanan-perjalanan kereta api sebagai sekumpulan pekerjaan (*jobs*) yang

harus dijadwalkan pada sekumpulan sumber daya (*resources*) berupa petak-petak blok atau segmen-segmen jalur kereta api [3]. Sebuah perjalanan kereta api terdiri atas operasi-operasi yang harus dikerjakan secara berturutan dimana setiap operasi tersebut adalah bagian perjalanan kereta api melewati sebuah petak blok. Urutan operasi-operasi yang harus dikerjakan sama dengan urutan petak blok yang harus dilewati oleh kereta api tersebut. Hal ini ditentukan dari rute yang dilalui oleh kereta api.

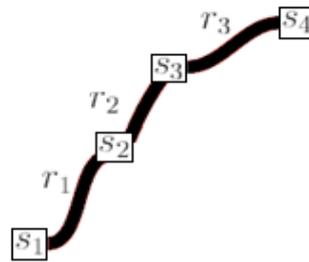
Ada beberapa strategi dan algoritma yang dapat digunakan untuk menyelesaikan masalah penjadwalan *Job-Shop*, antara lain *Backtracking*, *Branch and Bound* dan *Local Search*. Dalam makalah ini, algoritma yang digunakan adalah variasi dari algoritma *Local Search* yang bernama *Hill-Climbing* dengan strategi *variable and value ordering*, dan *constraint propagation*. *Variable and value ordering* digunakan untuk mengurutkan variabel yang akan diberi nilai (urutan operasi yang akan dijadwalkan lebih dulu) dan *constraint propagation* digunakan untuk menghapus nilai-nilai pada domain sebuah variabel jika nilai-nilai tersebut mengarah kepada pelanggaran batasan [4].

2 Deskripsi Masalah Penjadwalan Kereta Api Jalur Tunggal

Jaringan kereta api jalur tunggal terdiri atas sekumpulan stasiun dan petak-petak blok. Petak blok adalah bagian jalan kereta api yang dibatasi oleh dua buah sinyal berturutan. Sinyal-sinyal tersebut digunakan sebagai tanda apakah kereta api boleh menggunakan petak blok yang bersangkutan atau tidak. Hal ini diperlukan untuk mencegah terjadinya tabrakan dengan cara membuat aturan bahwa pada setiap waktu, satu petak blok hanya boleh digunakan oleh tepat satu perjalanan kereta api. Dalam hal ini, konflik terjadi jika satu petak blok digunakan oleh dua kereta api pada waktu yang sama.

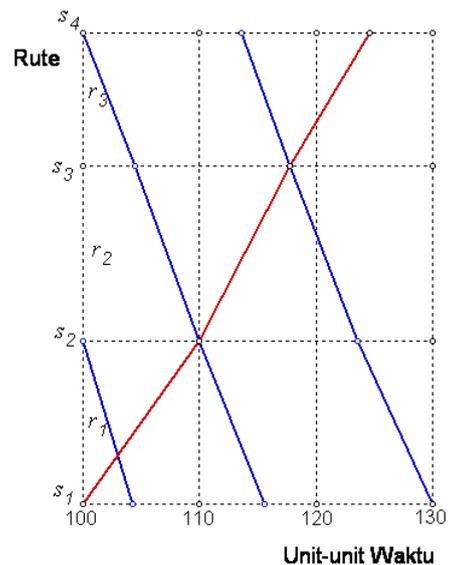
Selanjutnya, rute perjalanan kereta api dipandang sebagai himpunan terurut stasiun-stasiun dan petak-petak blok. Antara satu stasiun dengan stasiun berikutnya bisa terdiri atas satu atau lebih petak-petak blok. Dengan demikian, perjalanan kereta api dapat dipandang sebagai sekumpulan operasi melewati petak-petak blok tersebut atau menunda perjalanan dengan menunggu penggunaan petak-petak blok di stasiun (untuk menghindari penggunaan satu petak blok oleh dua kereta api) [4]. Contoh sebuah rute

perjalanan kereta api jalur tunggal disajikan pada Gambar 1 di bawah ini.



Gambar 1 Contoh Rute Perjalanan Kereta Api

Dalam masalah penjadwalan kereta api, jadwal pada umumnya dinyatakan dalam diagram ruang-waktu (*time-space diagram*) dan dalam makalah ini, satuan waktu terkecil yang digunakan adalah menit. Contoh sebuah diagram ruang-waktu disajikan pada Gambar 2 berikut.



Gambar 2 Contoh Diagram Ruang-Waktu

Terdapat beberapa aturan yang harus dipenuhi dalam penjadwalan kereta api jalur tunggal. Aturan yang utama adalah mencegah terjadinya konflik, yaitu penggunaan satu petak blok oleh dua kereta api pada waktu yang sama. Konflik di petak blok dapat terjadi pada dua kereta api yang berjalan pada arah yang berlawanan maupun arah yang sama. Untuk mencegah terjadinya konflik ini, salah satu kereta harus ditunda perjalanannya di stasiun sampai salah satu kereta selesai menggunakan petak blok yang digunakan bersama-sama. Jika arahnya berlawanan, aturan penundaan perjalanan kereta api ini biasa disebut sebagai aturan persilangan dan jika arahnya sama disebut aturan penyusulan [6]. Untuk dua kereta api yang berjalan pada

jalur dan arah yang sama, selain aturan penyusulan (tidak boleh menggunakan petak blok yang sama) juga terdapat aturan *headway*, yaitu selisih waktu minimal antara kedua kereta api tersebut [4][6].

Selain tiga aturan di atas yang menjamin keselamatan perjalanan kereta api, terdapat aturan-aturan lain yang harus diperhatikan dalam penjadwalan kereta api, antara lain:

1. Batas kecepatan maksimal kereta api di petak blok (*speed constraint*) [6].
2. Batas waktu minimal dan maksimal penundaan perjalanan kereta api di stasiun [1].
3. Urutan prioritas dua kereta api yang menggunakan petak blok yang sama [6].
4. Batas waktu minimal antara dua kereta api yang berlawanan arah di stasiun [6].
5. Dua buah kereta bertemu di sebuah stasiun selama suatu selang waktu tertentu [4][1].

3 Masalah Penjadwalan *Job-Shop*

Masalah penjadwalan *Job-Shop* terdiri atas sebuah himpunan pekerjaan $J = \{J_1, J_2, \dots, J_n\}$ yang harus dijadwalkan pada sebuah himpunan sumber daya atau mesin R . Setiap pekerjaan J_i terdiri atas sekumpulan operasi terurut yang harus dilakukan, yaitu $O_i = \{o_{i1}, o_{i2}, \dots, o_{ik}\}$ dengan $k = |O_i|$ adalah banyaknya operasi pada pekerjaan J_i . Kemudian setiap operasi o_{ij} menggunakan sebuah sumber daya yang tunggal $r_{ij} \in R$ selama suatu selang waktu tertentu. Lama waktu penggunaan sumber daya oleh operasi o_{ij} juga diberikan sebagai input ketika pendefinisian masalah penjadwalan *Job-Shop* tersebut, misalkan p_{ij} .

Dalam satu waktu, sebuah sumber daya hanya boleh digunakan oleh satu operasi saja. Dengan demikian, konflik terjadi ketika dua operasi yang berbeda menggunakan sumber daya yang sama dalam waktu yang sama juga [4].

Setiap pekerjaan J_i memiliki waktu pelepasan (*release date*) d_i , yaitu waktu dimulainya pekerjaan tersebut. Selain itu, J_i juga memiliki waktu harapan selesai (*expected completion date*) c_i . Pada makalah ini, waktu harapan selesai didefinisikan sebagai jumlah waktu penggunaan sumber daya oleh setiap operasi pada pekerjaan tersebut. [1] Secara matematis, hal ini ditulis sebagai

$$c_i = d_i + \sum_{j=1}^k p_{ij}, \text{ dengan } k = |O_i|.$$

Selain waktu pelepasan dan waktu harapan selesai, pekerjaan J_i yang sudah dijadwalkan

akan memiliki waktu selesai yang sebenarnya (*actual completion date*), yaitu C_i . Waktu tersebut adalah waktu selesainya operasi terakhir pada pekerjaan J_i . Jika d_{ij} menyatakan waktu pelepasan operasi o_{ij} pada pekerjaan J_i , kita memiliki beberapa hubungan matematis, misalnya

$$d_i = d_{i1}, \text{ dan } C_i = d_{ik} + p_{ik}, \text{ dengan } k = |O_i|.$$

Masalah penjadwalan *Job-Shop* kemudian didefinisikan sebagai pencarian waktu pelepasan setiap operasi o_{ij} pada pekerjaan-pekerjaan yang ada sehingga tidak terjadi konflik, yaitu penggunaan satu sumber daya oleh dua operasi yang sama pada waktu yang sama juga.

Terdapat beberapa kriteria optimasi yang dapat digunakan untuk mendapatkan jadwal dengan kualitas sebaik mungkin. Dalam makalah ini, kriteria optimasi yang digunakan adalah total keterlambatan minimum. Sebuah pekerjaan dikatakan terlambat jika waktu selesai sebenarnya pekerjaan tersebut lebih besar daripada waktu harapan selesai. Dalam hal ini, keterlambatan pekerjaan didefinisikan sebagai

$$T = \max \{C_i - c_i, 0\}.$$

Dengan demikian, total keterlambatan semua pekerjaan adalah

$$TT = \sum_{i=1}^n T_i, \text{ dengan } n = |J|.$$

4 Pemodelan Masalah

Bagian utama pemodelan masalah penjadwalan kereta api jalur tunggal dengan model masalah penjadwalan *Job-Shop* telah dijelaskan sebelumnya, yaitu peran perjalanan kereta api dan petak-petak blok sebagai pekerjaan dan sumber daya pada masalah penjadwalan *Job-Shop*.

Sekarang misalkan diberikan n buah perjalanan kereta api J_1, J_2, \dots, J_n yang harus dijadwalkan pada m buah rute. Setiap rute terdiri atas petak-petak blok dan stasiun yang berturutan. Jika sebuah perjalanan J_i melewati sebuah rute yang terdiri atas k buah petak blok, maka pekerjaan yang merepresentasikan perjalanan J_i tersebut terdiri atas k buah operasi $o_{i1}, o_{i2}, \dots, o_{ik}$. Setiap operasi yang dilakukan dalam perjalanan J_i tersebut menggunakan tepat satu sumber daya berupa satu petak blok yang ada pada rute yang dilalui, yaitu operasi o_{ij} menggunakan petak blok ke- j pada rute yang dilalui oleh J_i . Dengan demikian, urutan operasi pada pekerjaan J_i ditentukan oleh urutan petak-petak blok pada rute yang dilalui oleh J_i .

Dalam masalah penjadwalan *Job-Shop* secara umum, waktu penggunaan sumber daya oleh suatu operasi diberikan sebagai input. Dalam masalah penjadwalan kereta api, waktu ini sama dengan lama waktu perjalanan kereta api melewati sebuah petak blok. Hal ini dapat ditentukan dari kecepatan kereta api dan jarak petak blok dengan rumus

$$\text{waktu} = \text{jarak} / \text{kecepatan}.$$

5 Penyelesaian Masalah

Algoritma yang digunakan dalam penyelesaian masalah penjadwalan ini adalah *Hill-Climbing*. Algoritma ini terdiri atas dua bagian utama, yaitu pencarian sebuah solusi awal dan bagian kedua adalah pencarian solusi lain secara progresif untuk mengurangi total keterlambatan dari solusi yang telah ditemukan sebelumnya. *Pseudo-code* dari algoritma ini adalah sebagai berikut.

```
function Find-Good-Solution ()
  solusi ← Find-First-Solution ()
  repeat
    himpunan tetangga ← Nb (solusi)
    kriteria penghentian ← True
    for each tetangga in himpunan tetangga do
      if TT (tetangga) ≤ TT (solusi)
        then solusi ← tetangga
        kriteria penghentian ← False
      endif
    endfor
  until kriteria penghentian = True
  return solusi
```

Terdapat tiga fungsi utama pada skema di atas. Fungsi yang pertama adalah *Find-First-Solution* (). Fungsi ini digunakan untuk mencari satu solusi awal. Pencarian ini dilakukan secara cepat, walaupun total keterlambatan yang diperoleh masih besar. Fungsi yang kedua adalah fungsi ketetanggaan *Nb* yang mengembalikan himpunan tetangga dari solusi yang sudah ditemukan. Fungsi yang terakhir adalah fungsi objektif *TT* untuk menghitung total keterlambatan dari sebuah solusi.

Langkah-langkah yang dilakukan dalam pencarian solusi pertama ini adalah sebagai berikut:

1. Jadwalkan operasi-operasi tanpa melanggar urutan pada satu perjalanan.
2. Buat senarai yang menyimpan konflik karena penggunaan petak blok yang sama.
3. Pilih konflik yang terjadi paling awal dan urutan penggunaan petak blok.
4. Ubah waktu pelepasan operasi yang ditunda karena konflik dan juga waktu

pelepasan operasi-operasi setelahnya pada satu perjalanan yang sama.

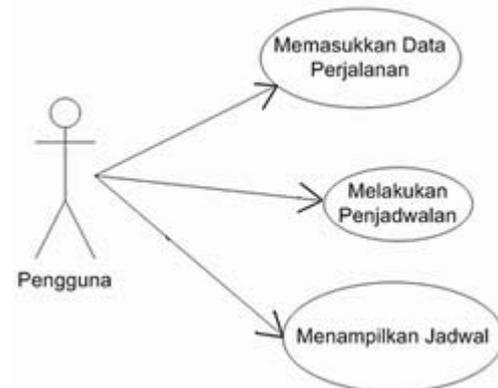
5. Ulangi langkah 2, 3 dan 4 di atas sampai senarai pada langkah dua kosong, yaitu tidak ada lagi konflik yang terjadi.

6 Analisis, Perancangan dan Implementasi Perangkat Lunak

Pada tahap analisis dilakukan identifikasi kebutuhan-kebutuhan utama perangkat lunak, diagram *use-case* dan diagram kelas tahap analisis. Kebutuhan utama perangkat lunak ini adalah:

1. Memasukkan data perjalanan kereta api.
2. Melihat data perjalanan yang telah dimasukkan.
3. Mengubah atau menghapus data masukan.
4. Melakukan penjadwalan.
5. Menampilkan jadwal dengan representasi diagram ruang-waktu.
6. Menyimpan diagram ruang-waktu sebagai file gambar.

Diagram *use-case* dari perangkat lunak ini disajikan pada Gambar 3 berikut.



Gambar 3 Diagram Use-Case

Selanjutnya dilakukan identifikasi kelas-kelas tahap analisis dan hubungan antara kelas-kelas yang telah dibuat tersebut. Hubungan ini kemudian direpresentasikan dalam diagram kelas tahap analisis.

Pada tahap perancangan, kelas-kelas yang dibuat dibagi menjadi tiga paket utama, yaitu paket antarmuka, paket penjadwalan dan paket basis data. Paket antarmuka berisi kelas-kelas untuk menangani antarmuka masukan dan keluaran, paket penjadwalan berisi kelas-kelas yang melakukan proses penjadwalan dan paket basis data berisi kelas-kelas yang menangani penyimpanan dan pengambilan data dari dan ke basis data.

Tahap implementasi dilakukan dengan menggunakan perangkat keras dengan spesifikasi:

1. Prosesor AMD Athlon 64 Processor 3500+ dengan *core speed* 2475,2 MHz.
2. Memori dengan tipe DDR2, ukuran 768MB dan frekuensi 275 MHz.

Sistem operasi yang digunakan adalah Microsoft Windows XP Professional Service Pack 3 Build (2600). Perangkat lunak dan kakas pendukung lain yang digunakan adalah:

1. NetBeans IDE 6.1 (Build 200804211638) untuk pembuatan kode sumber perangkat lunak.
2. *Compiler* Java versi 1.6.0_03-b05 untuk melakukan kompilasi kode sumber dan pemaketan menjadi sebuah file jar yang dapat dieksekusi.
3. XAMPP versi 1.5.5 yang sudah mengandung Apache 2.2.3, MySQL 5.0.27, PHP 5.2.0 dan phpMyAdmin 2.9.1.1. XAMPP dan semua isinya digunakan untuk pengaturan basis data yang digunakan oleh perangkat lunak ini.
4. Pustaka *mysql-connector-5.0.5* untuk melakukan koneksi dengan basis data MySQL dan pustaka *GapekaChart*, *JCommon* dan *JfreeChart* untuk pembuatan grafik diagram ruang-waktu.

Secara umum, kelas-kelas yang dibuat pada tahap implementasi ini sama dengan kelas-kelas yang dibuat pada tahap perancangan. Namun ada beberapa perubahan kecil yang terjadi, yaitu penambahan kelas yang menangani log penjadwalan.

7 Pengujian Perangkat Lunak

Perangkat lunak ini diuji dengan menggunakan data perjalanan kereta api yang diperoleh dari PT Kereta Api (Persero). Pengujian dilakukan dengan tujuan:

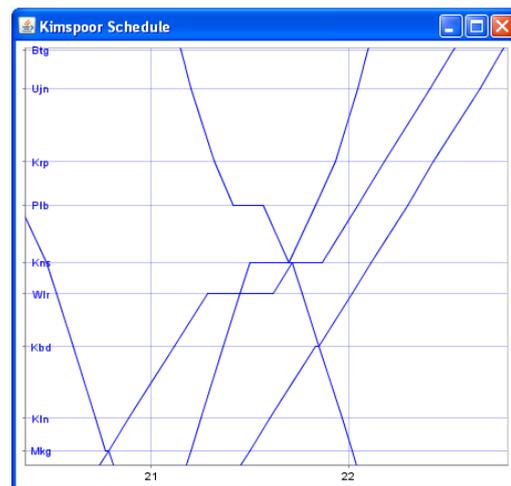
1. Mengetahui apakah perangkat lunak yang diimplementasikan telah sesuai dengan kebutuhan-kebutuhan utama perangkat lunak yang dispesifikasikan pada tahap analisis, yaitu kebutuhan fungsional dan kebutuhan non-fungsional.
2. Mengetahui apakah keluaran perangkat lunak yang berupa jadwal perjalanan kereta api telah sesuai dengan aturan-aturan yang ada.
3. Mengetahui kualitas jadwal yang dihasilkan (total keterlambatan, keterlambatan rata-rata, keterlambatan maksimum, keterlambatan minimum dan banyaknya perjalanan yang terlambat)

Oleh karena itu, pengujian dibagi menjadi dua bagian, yaitu pengujian fungsional dan pengujian penjadwalan. Pengujian fungsional dilaksanakan berdasarkan kebutuhan-

kebutuhan fungsional yang telah dibuat pada tahap analisis dan pengujian penjadwalan dilaksanakan untuk menguji jadwal yang dihasilkan oleh perangkat lunak.

Kasus uji yang digunakan dalam pengujian dapat dibagi menjadi dua. Yang pertama adalah kasus uji dimana basis data yang digunakan masih kosong. Kasus uji ini digunakan untuk menguji kebutuhan-kebutuhan fungsional perangkat lunak, seperti memasukkan data perjalanan kereta api, mengubah atau menghapus data dan melihat data yang telah dimasukkan. Kasus uji yang kedua adalah kasus uji dimana basis data sudah terisi dengan data perjalanan kereta api di Indonesia. Kasus uji ini digunakan untuk pengujian hasil penjadwalan.

Pengujian fungsional yang telah dilaksanakan menunjukkan bahwa perangkat lunak sudah memenuhi semua kebutuhan fungsional yang telah dispesifikasikan pada tahap analisis. Dan pengujian penjadwalan menunjukkan bahwa perangkat lunak selalu dapat memperoleh jadwal yang memenuhi semua aturan umum yang diajukan. Setelah perangkat lunak yang diajukan. Setelah perangkat lunak memperoleh jadwal pertama, perangkat lunak ini juga mampu memperoleh jadwal tetangga lain yang lebih baik (memiliki total keterlambatan yang lebih kecil). Gambar menyajikan contoh hasil keluaran perangkat lunak berupa diagram ruang-waktu yang memenuhi semua aturan yang diajukan.



Gambar 4 Contoh Keluaran Perangkat Lunak

Selanjutnya, Tabel 1 berikut ini menunjukkan hasil pengujian penjadwalan dengan 100 perjalanan dan 3 rute.

Tabel 1 Hasil Penjadwalan dengan 100 Perjalanan dan 3 Rute

| Jadwal ke- | Keterlambatan Total | Keterlambatan Rata-rata |
|------------|---------------------|-------------------------|
| 1 | 10 jam 14 menit | 6,14 menit |
| 2 | 10 jam 10 menit | 6,10 menit |
| 3 | 10 jam 9 menit | 6,09 menit |
| 4 | 10 jam 1 menit | 6,01 menit |
| 5 | 9 jam 52 menit | 5,92 menit |
| 6 | 9 jam 43 menit | 5,83 menit |
| 7 | 9 jam 41 menit | 5,81 menit |
| 8 | 9 jam 36 menit | 5,76 menit |
| 9 | 9 jam 19 menit | 5,59 menit |

Tabel 1 di atas menunjukkan beberapa jadwal yang dihasilkan dalam sekali penjadwalan. Jadwal yang diperoleh pertama kali memiliki total keterlambatan 10 jam 14 menit. Selanjutnya, perangkat lunak mencari jadwal lain yang lebih baik secara progresif. Jadwal terbaik yang ditemukan adalah jadwal ke-9 yang memiliki total keterlambatan 9 jam 19 menit.

Tabel 2 berikut ini menunjukkan hasil beberapa penjadwalan. Dalam tabel tersebut, pencarian jadwal yang lebih baik secara progresif tidak ditampilkan. Yang ditampilkan hanyalah jadwal awal dan jadwal akhir yang ditemukan saja.

Tabel 2 Hasil Beberapa Penjadwalan

| Banyak Perjalanan / Banyak Rute | Keterlambatan Total Jadwal Awal | Keterlambatan Total Jadwal Akhir |
|---------------------------------|---------------------------------|----------------------------------|
| 26 / 1 | 30 menit | 30 menit |
| 72 / 4 | 5 jam 4 menit | 4 jam 59 menit |
| 51 / 51 | 2 jam 30 menit | 2 jam 30 menit |
| 70 / 1 | 8 jam 33 menit | 7 jam 37 menit |
| 100 / 3 | 10 jam 14 menit | 9 jam 19 menit |

8 Kesimpulan

Kesimpulan yang dapat diambil implementasi model penjadwalan *Job-Shop* dalam masalah penjadwalan kereta api jalur tunggal ini adalah:

1. Masalah penjadwalan kereta api jalur tunggal dapat dimodelkan sebagai kasus khusus dari masalah penjadwalan *Job-Shop*. Penyelesaiannya kemudian dapat

dilakukan dengan pendekatan *Constraint Programming*.

2. Tidak semua aturan-aturan perjalanan kereta api jalur tunggal dapat dimodelkan dengan menggunakan model tersebut, tapi secara umum, semua aturan-aturan umum perjalanan kereta api jalur tunggal, yaitu aturan persilangan, aturan penyusulan, aturan *headway*, aturan kecepatan maksimal petak blok dan aturan waktu minimum penundaan perjalanan di stasiun dapat dimodelkan dengan model tersebut.
3. Algoritma *Hill-Climbing* yang digunakan untuk penyelesaian masalah selalu dapat mencari jadwal perjalanan kereta api yang memenuhi semua aturan umum perjalanan kereta api yang diajukan.
4. Walaupun jadwal perjalanan kereta api selalu dapat ditemukan, mekanisme pencarian jadwal seperti di atas memiliki beberapa kelemahan dalam keoptimalan jadwal akhir yang ditemukan (dengan kriteria optimasi total keterlambatan minimum), yaitu jadwal akhir yang ditemukan hanya merupakan jadwal terbaik diantara suatu himpunan jadwal saja dan masih terdapat kemungkinan adanya sebuah perjalanan yang memiliki keterlambatan sangat besar.
5. Perangkat lunak yang dihasilkan dapat digunakan untuk memasukkan data perjalanan kereta api, menampilkan data yang telah dimasukkan, melakukan penjadwalan dan menampilkan hasil penjadwalan dalam diagram ruang-waktu.

9 Daftar Pustaka

- [1] Mladenovc, Snezana & Cangalovic, Mirjana, 2007, *Heuristic Approach to Train Scheduling*, University of Belgrade.
- [2] Monfroy, Eric, 2001, *Constraint Programming: Introduction*, Universite de Nantes.
- [3] Oliveira, Elias & Smith, Barbara M., 2000, *A Job-Shop Scheduling Model for the Single-Track Railway Scheduling Problem*.
- [4] Oliveira, Elias, 2001, *Solving Single-Track Railway Scheduling Problem Using Constraint Programming*, PhD thesis, School of Computing: University of Leeds.
- [5] Smith, Barbara M., 2001, *Lecture Notes in Constraint Satisfaction and Constraint Programming*, University of Leeds.
- [6] Supriadi, Uned, 2001, *Perencanaan Perjalanan Kereta Api dan Pelaksanaannya*, Kantor Pusat PT Kereta Api (Persero).