

Aplikasi Algoritma Greedy sebagai Pendekatan untuk Menguraikan Permasalahan Pelangiran Kereta Api Penumpang

Muhammad Dhito Prihardhanto - 13507118

Prodi Teknik Informatika - Sekolah Teknik Elektro dan Informatika (STEI) , Institut Teknologi Bandung
Jalan Ganeca 10 Bandung
e-mail: if17118@students.if.itb.ac.id

ABSTRAK

Dalam transportasi perkeretaapian pengangkutan penumpang di stasiun sebenarnya menyimpan suatu permasalahan. Suatu rangkaian kereta api tiba dan berangkat dari stasiun sesuai dengan jadwal keberangkatan kereta api yang telah ditentukan. Pekerjaan berkaitan dengan kedatangan hingga keberangkatan kereta api (disebut dengan *matching*) dan penjadwalan perpindahan jalur saat kereta api langsir itu disebut dengan *shunting*. Algoritma greedy merupakan salah satu pendekatan yang bisa dilakukan untuk mencari minimum jumlah langsiran yang terjadi itu dan jadwal yang tepat.

Kata kunci: pendekatan algoritma greedy, *shunting track*, *shunting movement*

1. PENDAHULUAN

Pengangkutan penumpang kereta api adalah sebuah proses yang mendukung pelaksanaan jadwal kereta api di stasiun. Kereta api tiba dan berangkat dari stasiun menurut jadwal yang telah ditentukan. Setiap kedatangan dan keberangkatan kereta api di stasiun mungkin terdiri atas banyak dan bahkan mungkin bisa berbeda jenis rangkaian kereta api. Komposisi kereta api tersebut telah dispesifikasikan dalam jadwal perjalanan. Selama jam-jam sibuk hampir seluruh gerbong kereta api yang tersedia telah digunakan untuk mengangkut penumpang, meskipun demikian masih terdapat beberapa gerbong kereta api yang tidak digunakan untuk mengangkut penumpang dalam waktu-waktu tertentu, misal di beberapa negara Eropa, seperti Belanda, pada saat malam hari tidak banyak perjalanan kereta api. Gerbong-gerbong kereta api yang tidak digunakan itu biasanya diparkir di depo stasiun/tempat langsir (*shunt yard*) untuk waktu tertentu. Di Indonesia pun juga sama. Kereta-kereta api biasanya disimpan di suatu depo di stasiun-stasiun besar di mana rangkaian gerbong tersebut mungkin saja digabungkan dengan kereta lainnya.

Kereta api dapat diklasifikasikan menurut tipe dan subtipenya. Sebagai contoh, di Belanda, terdapat rangkaian kereta api bertipe ICM (Inter City Material) yang membawa 3 gerbong. ICM adalah tipenya, dan subtipenya adalah jumlah gerbongnya. Sedangkan di Indonesia tipe kereta api itu seperti gerbong kelas eksekutif, kelas, bisnis, dan ekonomi.

Untuk memarkir suatu rangkaian kereta api, seorang masinis harus melakukan beberapa aksi. Jika kereta hanya maju ke depan, masinis cukup memajukan keretanya saja hingga tempat langsir. Namun, secara umum tidak pernah terjadi seperti itu. Yang sering terjadi adalah kereta harus dimajukan, dimundurkan lagi pindah jalur, dimajukan lagi, lalu bisa diparkir. Peristiwa di mana kereta harus maju dan mundur untuk langsir itu disebut dengan *shunt movement*. Jadi, jika kereta hanya maju saja, hitungannya adalah satu *shunt movement* dan jika kereta harus maju, mundur, maju hitungannya adalah tiga *shunt movement*.

Ketika sebuah rangkaian kereta api yang sedang diparkir tengah dibutuhkan untuk disiapkan mengangkut penumpang karena telah masuk jadwal keberangkatannya, maka kereta api tersebut harus segera dipindahkan dari tempat parkirnya ke jalur di mana kereta api itu akan berangkat. Untuk melakukan hal tersebut sangat dimungkinkan harus terjadi beberapa *shunting movement*.

Sebagai konsekuensinya, agar tetap sesuai dengan jadwal perjalanan kereta api yang telah ditentukan, maka harus ada jadwal langsiran (*shunt schedule*) di setiap stasiun. Jadwal langsiran tersebut terdiri atas beberapa aksi yang mengindikasikan kereta api mana yang akan dilangsirkan dan dari mana ke mana. Dengan demikian jumlah *shunt movement* yang pasti akan dapat ditentukan. Penjadwalan langsiran akan lebih mudah jika jika semua kedatangan dan keberangkatan kereta api dapat dilaksanakan sesuai dengan harapan. Akan tetapi tidak semuanya akan berjalan dengan harapan. Jika dalam jadwal langsiran terdapat banyak *shunt movement*, jadwal akan menyebabkan beban kerja (*workload*) yang tinggi bagi masinis serta rawan terjadi kekacauan. Hal tersebut akan turut menyebabkan keterlambatan dalam jadwal kereta api, yang harus dihindari. Oleh karena itu tujuan dari pembuatan jadwal langsiran (*shunt schedule*) dengan jumlah *shunt movement* minimal.

2. METODE

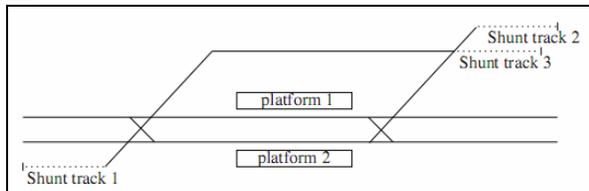
2.1 Deskripsi Permasalahan

Masukan (*input*) untuk permasalahan langsiran (*shunting problem*) di sebuah stasiun yang diberikan terdiri atas jadwal kereta api di stasiun tersebut dan *layout* stasiun (letak jalur-jalur kerta api dan tempat langsiran). Jadwal kereta api akan menggambarkan bagaimana kedatangan dan keberangkatan yang terjadi padastasiun. Setiap *event* akan dikarakterisasi oleh waktu, komposisi kereta, arahnya, jalur kereta yang dibutuhkan, dan apakah kereta dalam status tiba atau berangkat.

Tempat langsiran (*shunt yard*) terdiri atas sejumlah jalur untuk menyimpan sejumlah rangkaian kereta api. *shunt yard* dan jalur utama memiliki kapasitas terbatas untuk menyimpan sejumlah rangkaian kereta api. Ada sebuah jaringan rel yang nebhubungkan jalur *shun yard* dengan jalur utama.

Solusi yang dicari adalah suatu daftar *shunt movement* yang terjadi di antara *event-event* sehingga *event-event* tersebut dapat terjadi. Caranya adalah mencari solusi dengan jumlah minimum dari *shunt movement*. Asumsi yang digunakan dalam pendekatan yang akan dilakukan adalah *shunt movement* memakan waktu nol. Sehingga dimungkinkan dapat terjadi *shunt movement* dalam jumlah tak terhingga yang dpaat dilakukan di antara dua *event*.

Berikut ini adalah ilustrasi *shunting problem*. Misalkan diberikan sebuah stasiun dengan layout seperti di bawah ini:



Gambar 1. Layout contoh stasiun

Dalam contoh tersebut misalkan ada empat tipe kereta api, yaitu *A*, *B*, *C*, dan *D*. Setiap rangkaian terdiri atas beberapa gerbong yang terdiri atas tipe-tipe tersebut. Jadi ketika disebutkan kereta *AB* maka yang dimaksud adalah rangkaian kereta api yang terdiri atas tipe *A* dan tipe *B*, di mana posisi gerbong tipe *A* adalah di sambungan sebelah kiri gerbong tipe *B*, tidak peduli ke mana arah kereta akan berjalan. Dengan demikian kereta *AB* dan *BA* adalah berbeda komposisi. Diasumsikan bahwa kapasitas dari setiap jalur rel adalah 3 gerbong kereta. Misalkan menurut jadwal kereta api, keberangkatan dan kedatangan kereta api adalah terurut sebagai berikut:

- e1 Kereta *AB* tiba dari kiri di platform 1.
- e2 Kereta *AA* tiba dari kanan di platform 2.
- e3 Kereta *CCC* tiba dari kanan di platform 1.
- e4 Kereta *CC* berangkat dari platform 1 ke kiri.
- e5 Kereta *AA* berangkat dari platform 2 ke kanan.

- e6 Kereta *DC* tiba dari kiri di platform 1.
- e7 Kereta *CDC* berangkat dari platform 1 ke kanan.
- e8 Kereta *BA* tiba dari kanan di platform 2.
- e9 Kereta *BB* berangkat dari platform 2 ke kanan.
- e10 Kereta *AA* berangkat dari platform 1 ke kiri.

Sedangkan berikut ini adalah jadwal langsiran (*shunt schedule*) yang valid untuk jadwal kereta api di atas:

Antara e2 dan e3 → Kereta *AB* langsir dari platform 1 ke shunt track 2

Antara e5 dan e6 → Kereta *C* langsir dari platform 1 ke shunt track 1

Antara e6 dan e7 → Kereta *C* langsir dari shunt track 1 ke platform 1

Antara e8 dan e9 → Kereta *A* langsir dari platform 2 ke shunt track 2, Kereta *AA* langsir dari shunt track 2 ke platform 1, dan Kereta *B* langsir dari shunt track 2 ke platform 2

Contoh solusi di atas terdiri atas 6 *shunt movement*. Dalam contoh tersebut keputusan apakah kereta akan langsir ke kiri atau ke kanan memegang peran yang sangat penting.

2.1 Pendekatan dengan Algoritma Greedy

Ide dasar dari pendekatan ini adalah mengamati dan mendaftar *event-event* yang terjadi dan secara iteratif menentukan aksi apa yang akan dilakukan. Keputusan dalam setiap iterasi berdasarkan pada situasi yang dihasilkan dari keputusan sebelumnya dan *event* sekarang (*current event*). Dengan cara ini pendekatan yang dilakukan adalah mencoba untuk memperluas cakupan situasi yang diberikan sebaik mungkin. Oleh karena itu pendekatan ini dikategorikan ke dalam greedy.

Dalam hal praktis tampak bahwa pihak perencana (baca: pihak perkeretaapian) lebih menyukai situasi di mana rangkaian kereta api yang akan diberangkatkan sudah tersedia di suatu jalur rel di stasiun tersebut dan dalam komposisi yang sesuai. Hal seperti ini juga biasa terjadi di Indonesia untuk perjalanan kereta api jarak jauh.

Filosofi itu disebut dengan istilah *be ready for departure*. Filosofi itu yang akan digunakan sebagai *guideline* dalam membangun pendekatan greedy setelah ini.

Sebagai awalan, asumsi yang digunakan selama cakrawala perencanaan adalah kereta api yang datang berkorespondensi satu-satu dengan kereta api yang berangkat. Asumsi lainnya adalah stasiun dalam keadaan kosong pada keadaan awal dan akhir dari cakrawala perencanaan. Dengan demikian hal ini dapat dibenarkan dengan mengambil cakrawala perencanaan untuk membentuk satu jam padat berikutnya, karena selama jam padat seluruh materi dibutuhkan dalam jadwal kereta api.

Jadi heuristik yang disajikan dapat diadopsi ketika asumsi ini dijatuhkan.

Algoritma greedy ini terjadi atas dua langkah utama:

- 1) Mulai dengan stasiun kosong (jalur rel bebas dari kereta api).
- 2) Periksa daftar *event* dalam waktu, dan untuk setiap *event DO*:
 - a) **IF** *event* adalah keberangkatan, **THEN** pindahkan seluruh kereta ke jalur langsung (*shunt track*).
 - b) **ELSE** *event* adalah kedatangan, **THEN** pasangankan rangkaian kereta ke rangkaian kereta yang sudah ditempatkan pada *shunt track*.

Tabel 1 Daftar event perjalanan kereta api

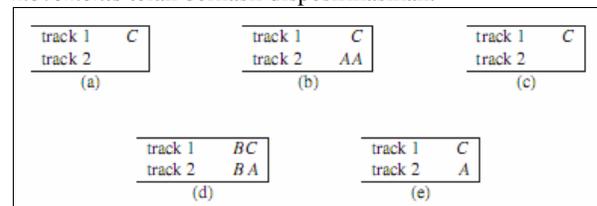
Event	e1	e2	e3	e4	e5	e6	e7	e8
Kedatangan	C	A	BB			AA		
Keberangkatan				B	BA		AA	C
Platform/Jalur	2	1	2	2	1	2	1	2

Langkah paling penting dalam algoritma tersebut adalah langkah 2a dan 2b. Dalam langkah itu keputusan utama diambil. Dalam langkah 2a ditentukan di manashunt track kereta ditempatkan untuk persiapan keberangkatan. Dalam poin ini, tidak dipedulikan bagaimana rangkaian kereta itu datang ke *shunt track* tersebut, tetapi hanya untuk menentukan bahwa rangkaian kereta ditempatkan di situ adalah untuk persiapan keberangkatan. Bagaimana rangkaian itu datang pada posisinya pada *shunt track* akan ditentukan pada iterasi berikutnya. Sedangkan dalam langkah 2b dilakukan langkah menyambungkan rangkaian kereta yang baru saja tiba ke rangkaian kereta yang sudah ditempatkan untuk keberangkatan dari jalur tertentu dalam iterasi sebelumnya.

Agar lebih memahami ide dasar dari pendekatan greedy ini akan diberikan sebuah contoh dimana suatu stasiun terdiri atas 2 jalur (*platform*) dan 2 *shunt track*. Daftar event diberikan pada tabel 1 di atas.

Iterasi diawali dari daftar paling akhir yaitu *event e8*. Karena *event* ini merupakan keberangkatan maka diputuskan untuk menempatkan kereta ke *shunt track* 1. Situasi dua *shunt track* setelah keputusan ini diilustrasikan pada gambar 2a. *Event* berikutnya adalah *e7* yang juga keberangkatan. Oleh karena itu, kereta AA ditempatkan pada *shunt track* 2 (gambar 2b). Selanjutnya untuk melangsirkan kereta AA yang datang pada *event e6* memiliki pilihan yang bagus yaitu dipasangkan pada *shunt track* 2. Karena penggabungan serupa itu, *shunt track* 2 tersebut menjadi kosong (gambar 2c). Berikutnya, kereta pada *event e5* dapat ditempatkan pada *shunt track* 2 dan kereta pada *event e4* ditempatkan pada *shunt track* 1 sehingga menghasilkan situasi seperti pada gambar 2d. Sekarang *event e3* kereta BB tidak dapat dipasangkan seluruhnya ke *shunt track*, tetapi solusinya adalah kereta

tersebut harus dipisahkan (*split*) terlebih dahulu dan ditempatkan masing-masing ke tiap *shunt track*. Karena di masing-masing itu telah terdapat kereta BC dan BA maka gerbong B yang masuk ke masing-masing *shunt track* tersebut akan saling menghilangkan sehingga menghasilkan situasi seperti gambar 2e. Yang perlu dicatat bahwa aksi tersebut artinya sama dengan terjadi 2 *shunt movement* terpisah. Terakhir, dua *event e2* dan *e1* dijalankan dengan memasangkannya sesuai dengan korespondensi gerbong dengan tipe yang sama yang telah ada pada masing-masing *shunt track*. Dari ilustrasi di atas telah ditunjukkan bagaimana algoritma tersebut telah menentukan setiap kereta yang tiba dan kereta yang datang merupakan sebagai suatu pasangan dan melalui *shunt track* pekerjaan itu dilakukan. Dalam cara ini *shunt movements* telah berhasil dispesifikasikan.



Gambar 2. Situasi event dari tabel 1

3. KESIMPULAN

Dari ulasan di atas, telah ditunjukkan bahwasannya filosofi algoritma greedy yang berprinsip "take what you can get now!" dapat membantu menguraikan permasalahan yang ada di dunia perkeretaapian, khususnya dalam masalah penjadwalan *shunting movement* (langsiran) kereta api untuk mendapatkan penjadwalan yang tepat dan ideal. Namun metode yang digunakan di sini masih merupakan yang sederhana. Metode yang dibutuhkan akan lebih kompleks lagi ketika permasalahan semakin rumit. Ketika itu akan ada pendekatan lain yang digunakan, seperti dengan menggunakan algoritma program dinamis.

REFERENSI

- [1] Marjan van den Akker Hilbrandt Baarsma Johann Hurink Maciej Modelski Jacob Jan Paulus Igrid Reijnen Dan Roozmond Jan Schreuderdkk., "Shunting Passenger Trains: Getting Ready For Departure", Universiteit Utrecht, 35, 2008, halaman 1-9.
- [2] Rinaldi Munir, *Strategi Algoritmik*, Departemen Teknik Informatika ITB, 2005.