OPTIMASI PENYUSUNAN PEGAS DENGAN METODE SISTEM PERBEDAAN BATASAN DAN ALGORITMA JALUR TERPENDEK

Johan Varian Alfa1), **Rully Soelaiman**2), **Chastine Fatichah**3)

Teknik Informatika, Fakultas Teknologi Informasi , ITS - Surabaya1), 2), 3)
Institut Teknologi Sepuluh Nopember
Email: jowhunt@gmail.com1), rully130270@gmail.com2), chastine.fatichah@gmail.com3)

ABSTRAK

Pada permasalahan nyata, khususnya dunia fisika, penyusunan pegas dengan batasan-batasan tertentu yang optimal merupakan salah satu permasalahan optimasi yang muncul, dimana batasan yang diberikan adalah besaran-besaran yang membentuk gaya pegas. Pada penelitian ini, diusulkan sebuah desain algoritma optimasi penyusunan pegas, yang dimulai dengan memodelkan permasalahan ke dalam graf, kemudian menggunakan metode sistem perbedaan batasan dan juga algoritma jalur terpendek untuk menghasilkan susunan pegas yang optimal. Sistem perbedaan batasan digunakan untuk memodelkan permasalahan ke dalam bentuk pertidaksamaan. Kemudian dicari penyelesaiannya dengan menggunakan konsep graf yang disebut graf batasan. Penyelesaian akhir yang digunakan agar mendapatkan solusi yang optimal adalah algoritma jalur terpendek. Algortima jalur terpendek yang digunakan adalah algoritma Perbaikan Dijkstra. Hasilnya mampu menghasilkan susunan pegas yang optimal dan benar. Dan setelah diuji coba, algoritma Perbaikan Dijkstra yang digunakan mampu lebih efisien dari segi performa waktu eksekusi dibandingkan algoritma Bellman-Ford. Penghematan waktu yang didapat dengan menggunakan algoritma Perbaikan Dijkstra rata-rata mencapai 83,55%.

Kata Kunci: Graf, Jalur Terpendek, Optimasi, Sistem Perbedaan Batasan

ABSTRACT

In the real world problems, especially on physics, optimal arrangement goods, with certain constraint, is one of favorite problem in the world. One of these problems is the optimal arrangement spring, which there is some constraint that be formed by physics quantities that made elastic force. In this research, design of arrangement spring optimization algorithm is proposed, which is began with modeling the problem to graph, then use the system of difference constraint method and shortest path algorithm to produce optimal arrangement spring. System of difference constraint is used to model problems into inequality. Then, we found solution using graph concept was called constraint graph. Finally, shortest path algorithm is used to obtain the optimal solution. Improved Dijkstra is selected as shortest path algorithm that is used in this algorithm. The result can produce the right and optimal arrangement of spring. Improved Dijkstra is more efficient than Bellman-Ford algorithm in running time. This algorithm can save an average running time of up to 83,55%.

Key Words: *Graph, Optimization, Shortest path, System of difference constraint.*

1. Pendahuluan

Permasalahan jalur terpendek merupakan permasalahan yang umum dan terkenal dalam teori graf. Dalam teorinya, jalur terpendek merupakan algoritma untuk menentukan bobot terkecil dari sebuah jalur dalam graf berarah [1]. Jalur merupakan sebuah rangkaian *verteks* dalam sebuah graf yang terhubung melalui *edge* antara sebuah *verteks* dengan *verteks* berikutnya pada rangkaian *verteks* tersebut. Bobot dari sebuah jalur merupakan hasil penjumlahan dari bobot semua *edge* yang menghubungkan setiap *verteks* dari *verteks* asal hingga *verteks* tujuan dalam sebuah graf berarah dan berbobot.

Berbagai macam algoritma permasalahan jalur terpendek telah dibuat. Secara umum permasalahan jalur terpendek dibagi menjadi 2, jalur terpendek sumber tunggal (*Single-Source Shortest Paths*) dan jalur terpendek semua pasangan (*All-Pairs Shortest Paths*). Pada jalur terpendek sumber tunggal, beberapa algoritma yang digunakan antara lain Dijkstra [2] dan juga Bellman-Ford [3] dan beberapa perkembangan dari kedua algoritma tersebut [4][5][6]. Sedangkan untuk jalur terpendek semua pasangan, algoritma yang digunakan adalah Floyd-Warshall dan Johnson.

Permasalahan optimasi penyusunan pegas merupakan permasalahan nyata yang sering muncul saat ini. Permasalahan ini banyak muncul terutama dalam percobaan-percobaan fisika.

Untuk permasalahan seperti ini, dapat pula diselesaikan dengan memanfaatkan konsep sistem perbedaan batasan (*sistem of difference constraint*) yang kemudian diinterpretasikan dalam bentuk jalur terpendek [1]. Dengan solusi dari sistem perbedaan batasan, akan didapatkan batasan nilai yang optimal dari permasalahan yang ada yang disebut sebagai solusi yang paling baik (*feasible solution*). Untuk mendapatkan solusi dari sistem perbedaan batasan dapat memanfaatkan bobot dari algoritma jalur terpendek dengan merepresentasikan persamaan yang didapat dari sistem perbedaan batasan dalam bentuk graf. Diangkat dari permasalahan nyata tersebut, dilakukan sebuah studi untuk melakukan optimasi penyusunan pegas dengan memanfaatkan konsep sistem perbedaan batasan dan algoritma jalur terpendek untuk mendapatkan solusi dari sistem perbedaan batasan tersebut. Kemudian, dilakukan pula implementasi algoritma yang telah dibangun dan mencoba mengujinya dengan permasalahan yang sejenis pada sistem penilaian *daring* (*online judge system*) [7].

2. Tinjauan Pustaka

Bagian ini menjelaskan penjelasan umum tentang metode-metode yang digunakan pada optimasi penyusunan pegas berdasarkan referensi pustaka. Pertama akan dibahas algoritma perbaikan Dijkstra yang merupakan salah satu algoritma permasalahan jalur terpendek dijelaskan dan yang terakhir adalah penjelasan tentang sistem perbedaan batasan sebagai metode untuk memodelkan permasalahan ke dalam bentuk pertidaksamaan.

A. Perbaikan Dijkstra dalam Algoritma Pelabelan

Pada algoritma pelabelan, algoritma Dijkstra dapat digunakan dalam berbagai penyelesaian permasalahan nyata seperti : *multi-point routing*, ilmu survei dan pemetaan, pencarian jalur terpendek transportasi dan arus logistik, sistem cerdas transportasi, ilmu jaringan komputer, dan aplikasi nyata lainnya. Banyak penelitian telah dilakukan dalam penerapan algoritma pelabelan Dijkstra (*Dijkstra's label algorithm*) dalam berbagai kasus nyata tersebut. Wang Shu-Xi [2], memberikan sebuah perbaikan algoritma Dijkstra dalam algoritma pelabelan.

Algoritma Dijikstra sebenarnya telah cukup efisien dalam prosesnya, namun ada beberapa kekurangan yang perlu diperbaiki. Setidaknya ada 3 perbaikan yang dilakukan antara lain :

1) Memperbaiki akhir dari mekanisme algoritma Dijkstra untuk kasus-kasus yang menyebabkan perulangan tak hingga (*infinite loop*).

2) Perbaikan agar algoritma tersebut mampu mendapatkan *verteks-verteks* yang bertetangga, secara spesifik *verteks-verteks* sebelumnya, pada jalur terpendek yang ditemukan.
3) Perbaikan proses penetapan "p-label" secara bersamaan pada lebih dari satu *verteks*.

Perbaikan algoritma Djikstra ini lebih kepada proses pelabelan, belum kepada efisiensi algoritma. Justru dengan perbaikan ini cenderung membuat efisiensi algoritma menjadi rendah karena menghasilkan beberapa proses baru yang harus dilakukan.

B. Sistem Perbedaan Batasan

Pada permasalahan program linier, ada beberapa permasalahan yang dapat diselesaikan dengan memanfaatkan algoritma jalur terpendek dari sumber tunggal [1]. Tujuan permasalahan pada umumnya adalah mengoptimalkan sebuah fungsi linier yang bergantung pada sebuah himpunan pertidaksamaan linier. Secara umum dalam permasalahan program linier, diberikan sebuah matriks A berordo $m \times n$, sebuah vektor b berukuran m dan sebuah c berukuran n. Tujuannya adalah untuk menemukan vektor x dengan elemen sejumlah n yang mengoptimalkan (bisa meminimalkan ataupun memaksimalkan) fungsi sasaran $\sum_{i=1}^{n} c_i x_i$ yang bergantung pada batasan m yang diberikan oleh pertidaksamaan $Ax \le b$. Beberapa permasalahan program linier tidak terlalu memperhatikan fungsi sasaran, namun lebih mencari solusi yang layak / baik (*feasible solution*) dimana setiap vektor x dapat memenuhi $Ax \le b$ atau menentukan bahwa tidak ada solusi yang layak dari permasalahan yang ada. Caranya adalah dengan menggunakan konsep sistem perbedaan batasan.

Dalam sebuah sistem perbedaan batasan, setiap baris pada program linier matriks A berordo $m \times n$, elemen-elemennya terdiri dari sebuah nilai 1, sebuah nilai -1 dan sisanya bernilai 0. Batasan yang diberikan oleh $Ax \le b$ adalah sekumpulan perbedaan batasan sejumlah m termasuk n yang tidak diketaui nilainya, dimana masing-masing batasan tersebut merupakan sebuah pertidaksamaan linier yang berbentuk seperti Persamaan (1).

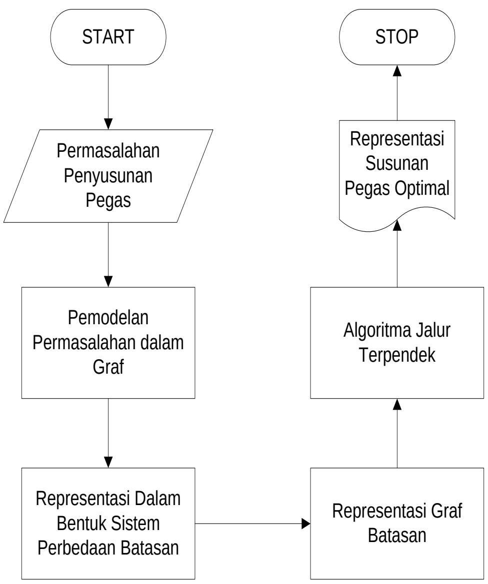
$$x_j - x_i \le b_k \tag{1}$$

di mana $1 \le i, j \le n$ dan $1 \le k \le m$

Konsep sistem perbedaan batasan banyak terjadi pada berbagai aplikasi nyata yang berbeda-beda. Dalam penyelesaiannya, konsep sistem perbedaan batasan dapat memanfaatkan algoritma jalur terpendek dengan merepresentasikan program linier yang ada ke dalam bentuk graf. Hal ini membawa keuntungan tersendiri karena permasalahan-permasalahan ini dapat diselesaikan dengan waktu eksekusi yang relatif lebih cepat daripada proses pemrograman linier pada umumnya.

3. Metode

Pada bagian ini, diberikan penjelasan gambaran metode secara umum beserta langkah-langkah algoritma secara detail yang disusun pada proses optimasi penyusunan pegas. Langkah-langkah algoritma tersebut disusun mulai dari pemodelan permasalahan pegas dalam graf, representasi model ke dalam sistem perbedaan batasan, representasi ke dalam graf batasan hingga algoritma jalur terpendek yang digunakan untuk menemukan hasil yang optimal dari susunan pegas. Secara umum, gambaran desain algoritma pada penelitian ini dapat dilihat pada Gambar 1.



Gambar 1. Diagram alur metode

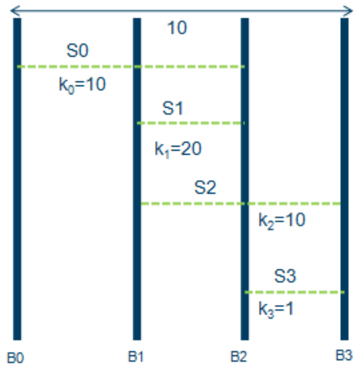
A. Pemodelan Permasalahan dalam Graf

Diberikan susunan atau konfigurasi pegas-pegas yang berisi gulungan pegas-pegas namun tidak memiliki medan magnet. Susunan ini dibentuk dengan sejumlah N bar (dari 0 hingga N-1) dan sejumlah M pegas (dari 0 hingga M-1) dimana setiap pegas menghubungkan dua buah bar yang berbeda.

Setiap pegas memiliki nilai konstanta sendiri-sendiri. Pegas ini memiliki panjang yang sangat kecil sehingga panjang pegas sama dengan pergeseran (jarak antara 2 bar) pada pegas tersebut. Selain itu, massa dan lebar pada bar diabaikan, sehingga jika ada dua buah pegas yang terkait pada satu buah bar yang sama sebenarnya dua pegas tersebut saling terkait. Dan susunan suatu pegas dari bar awal hingga akhir tersusun secara seri.

Penataan susunan pegas diposisikan secara horizontal dimana jarak antara bar ke 0 dan bar ke (N-1) ditetapkan sejauh *d*. Kemudian bar yang tersisa antara kedua bar tersebut dapat diatur sedemikian rupa dengan jarak tertentu. Bar-bar antara kedua bar tersebut dapat diatur sedemikian rupa baik jarak maupun posisinya tanpa harus memikirkan urutannya, namun bar ke-0 dan ke-(N-1) tetap sebagai bar pertama dan terakhir.

Kemudian diberikan salah satu contoh permasalahan susunan pegas. Jika jumlah bar dinotasikan N, jumlah pegas dinotasikan M, jarak maksimal susunan pegas dinotasikan d, dan konstanta pegas dinotasikan k. Diberikan data masukan N = 4, M = 4, dan d = 10 dengan masing-masing pegas memiliki konfigurasi masukan. Untuk pegas pertama terletak antara bar ke-0 dan bar ke-2 dengan nilai k_0 = 10, kemudian pegas kedua terletak antara bar ke-1 dan bar ke-2 dengan nilai k_1 = 20, kemudian pegas ketiga terletak antara bar ke-1 dan bar ke-3 dengan nilai k_2 = 10, dan pegas keempat terletak antara bar ke-2 dan bar ke-3 dengan nilai k_3 = 1. Maka gambar susunan awal pegas sesuai data masukan tersebut ditunjukkan pada Gambar 2.

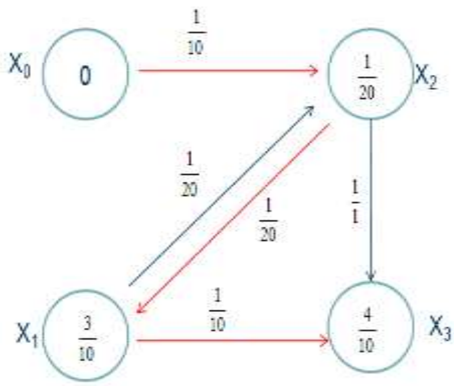


Gambar 2. Contoh susunan awal pegas

Untuk mendapatkan susunan yang optimal, dapat dilakukan dengan menyusun ulang pegas. Namun akan banyak kemungkinan susunan ulang yang harus dicoba.

Untuk mendapatkan susunan pegas secara lebih efisien dibutuhkan sebuah representasi atau pemodelan dimana dari model tersebut dapat diberikan representasi susunan pegas yang optimal melalui sebuah algoritma. Pemodelan permasalahan ke dalam bentuk graf dilakukan dengan representasi bar sebagai *verteks* dan pegas sebagai *edge*. Graf yang dibangun adalah graf berarah dan berbobot, dimana bobot *edge* didapatkan dari nilai $\frac{1}{k}$ dari setiap pegas.

Untuk mendapatkan representasi susunan pegas optimal, berupa gaya pegas maksimal yang optimal, didapatkan dengan algoritma jalur terpendek. Gambar 3 merupakan model graf yang didapatkan dari contoh permasalahan susunan pegas. Proses pemodelan permasalahan dalam bentuk graf dijelaskan lebih detail di bagian berikutnya.



Gambar 3. Model graf dari susunan awal

Gaya pegas yang optimal, yang merepresentasikan susunan pegas yang optimal, didapatkan dengan rumus seperti Persamaan (2).

$$|F| = k \cdot x \quad (2)$$

dimana :

$|F|$ adalah nilai mutlak dari gaya pegas

k adalah konstanta pegas

x adalah panjang pegas

Pada permasalahan susunan pegas ini, didapatkan bahwa pada susunan pegas yang optimal nilai x dan k selalu berbanding terbalik. Nilai x untuk setiap pegas tidak didapatkan dari data masukan, namun batas maksimal jarak antara bar pertama hingga bar terakhir, yang dinotasikan d, ditentukan oleh data masukan. Dengan kata lain, nilai d merupakan jumlah nilai x untuk susunan yang optimum, sehingga dirumuskan pada Persamaan (3).

$$d = x_1 + x_2 + ... + x_n = sum(x_n) \quad (3)$$

Untuk mendapatkan nilai masing-masing panjang pegas (x), bisa didapatkan dengan memanfaatkan perbandingan nilai konstanta pegas beserta nilai jumlah panjang pegas (x) yang optimum dengan Persamaan (4).

$$x_n = \frac{\frac{1}{k_n}}{\frac{1}{k_1} + \frac{1}{k_2} + ... + \frac{1}{k_n}} \times d \quad (4)$$

Nilai gaya pegas untuk setiap pegas didapatkan dari perkalian nilai konstanta pegas dan panjang pegas. Jika disubstitusikan panjang pegas seperti rumus sebelumnya, maka akan didapatkan persamaan gaya pegas untuk setiap pegas adalah seperti Persamaan (5).

$$F = \frac{d}{sum(\frac{1}{k})} \quad (5)$$

Untuk mendapatkan gaya maksimum, maka dibutuhkan nilai $sum(\frac{1}{k})$ terkecil. Maka dengan memanfaatkan algoritma jalur terpendek, akan didapatkan gaya maksimum yang optimal.

B. Representasi Model dalam Sistem Perbedaan Batasan

Tahap ini merupakan tahap untuk merepresentasikan soal serta model graf yang didapat atau permasalahan ke dalam sebuah model sistem pertidaksamaan dengan menggunakan algoritma sistem perbedaan batasan. Sistem perbedaan batasan $Ax \le b$, dimana matriks nilai x, yang berordo $i \times 1$, merupakan representasi bar dan matriks A berordo $m \times n$ merupakan representasi pegas yang menhubungkan dua buah nilai x_i dan x_j dengan memiliki batasan b sejumlah k sehingga menjadi sebuah pertidaksamaan $x_j - x_i \le b_k$, dimana $1 \le i, j \le n$ dan $1 \le k \le m$.

Nilai-nilai untuk sistem perbedaan batasan $Ax \le b$ didapatkan berdasarkan data masukan untuk setiap proses penyusunan pegas dimana batasan b merupakan nilai $\frac{1}{k}$. Maka akan dihasilkan sebuah sistem pertidaksamaan matriks dengan bar -bar penghubungnya (yang nilainya diambil dari contoh sebelumnya) dari sistem perbedaan batasan tersebut seperti Persamaan (6).

$$\begin{pmatrix} -1 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & -1 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \le \begin{pmatrix} 1/k_1 \\ 1/k_2 \\ 1/k_2 \\ 1/k_3 \\ 1/k_4 \end{pmatrix} \quad (6)$$

Dari pertidaksamaan matriks tersebut akan dihasilkan beberapa pertidaksamaan (7).

$$\begin{aligned} x_2 - x_1 &\le b_1 \\ x_2 - x_3 &\le b_2 \\ x_3 - x_2 &\le b_3 \\ x_4 - x_3 &\le b_4 \\ x_4 - x_2 &\le b_5 \end{aligned} \quad (7)$$

Jadi nilai x_1 hingga x_4 merupakan representasi bar , dan matriks A merupakan penghubung dua buah nilai x ke dalam bentuk pertidaksamaan, sehingga nilainya hanya -1, 0, dan 1 yang merepresentasikan pegas yang menghubungkan dua bar .

Nilai batasan b_k merupakan sebuah variabel yang didapatkan dari variabel-

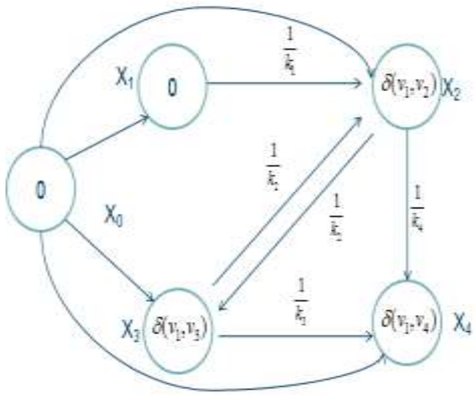
variabel dari data masukan. Setelah didapatkan semua model-model persamaan tersebut dari permasalahan susunan pegas yang ada, maka langkah berikutnya adalah menyelesaikannya secara optimal. Nilai yang dicari adalah batasan pada sistem pertidaksamaan dari sistem perbedaan batasan. Proses penyelesaian adalah mencari nilai pegas (x) dari sistem pertidaksamaan yang ada. Namun, jika dicari secara langsung dengan program linier akan memakan waktu yang lama. Oleh karena itu dicari solusinya dengan menggunakan konsep graf terutama dengan algoritma jalur terpendek.

C. Representasi Graf Batasan

Pada sistem perbedaan batasan $Ax \le b$, sebuah matriks program linier A berordo $m \times n$ dapat direpresentasikan dalam bentuk graf. Sebuah graf dengan *verteks* berjumlah n dan dengan *edge* sejumlah m. Setiap *verteks* v_i untuk $i = 1,2,\ldots,n$ berhubungan dengan variabel x_i sejumlah n dan setiap *edge* yang berarah dalam graf berhubungan dengan batasan b sejumlah m termasuk dua buah variabel x_i dan x_j pada pertidaksamaan dalam setiap batasan. Dengan demikian sistem perbedaan batasan $Ax \le b$ akan dapat direpresentasikan dalam sebuah graf batasan yang berarah dan berbobot $G = (V, E)$.

Graf $G = (V, E)$ yang dibangun adalah graf yang berarah dan berbobot, maka batasan b direpresentasasikan dalam bobot *edge* yang berarah. Graf batasan didapatkan juga dari model graf yang dibangun dari soal sebelumnya.

Sebuah *verteks* v_0 ditambahkan dimana *verteks* v_0 terhubung dengan seluruh *verteks* v_i. Maka himpunan *verteks* V terdiri dari seluruh *verteks* v_i sebagai representasi variabel x_i pada sistem perbedaan batasan dengan tambahan *verteks* v_0. Sedangkan himpunan *edge* E berisi seluruh perbedaan batasan yang ada yang melibatkan dua buah variabel v_i dalam sebuah pertidaksamaan $x_j - x_i \le b_k$ dan sebuah *edge* $e = (v_0, v_i)$ untuk setiap variabel x_i. Dengan demikian, dapat pula direpresentasikan bahwa setiap bobot *edge* (v_i, v_j) merupakan nilai batasan pada setiap pertidaksamaan, atau secara umum dapat ditulis $w(v_i, v_j) = b_k$. Kemudian untuk setiap *edge* $e = (v_0, v_i)$ diberi bobot 0. Kemudian dari hasil pertidaksamaan dari sistem perbedaan batasan akan digambarkan graf batasan sebagai representasinya yang digambarkan pada Gambar 4.



Gambar 4. Contoh graf batasan

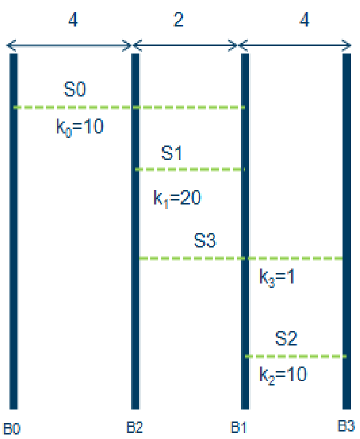
Dari graf batasan yang telah dibangun tersebut, maka dicari nilai setiap *verteks* v_i sebagai representasi nilai x_i yang dicari dengan menggunakan algoritma jalur terpendek. Dengan demikian, akan didapatkan nilai yang paling optimal untuk setiap nilai x_i atau sebagai solusi yang layak dari sistem perbedaan batasan yang telah dibangun. Nilai x_i yang optimum akan menghasilkan gaya pegas yang paling optimal sehingga juga akan menghasilkan susunan pegas yang optimal.

D. Algoritma Jalur Terpendek

Tahap ini merupakan tahap untuk penyelesaian dalam proses pengoptimal penyusunan pegas, yaitu dengan mencari nilai optimal dari nilai x_i dengan mencari jalur terpendek dari graf batasan yang telah dibangun. Jalur yang terbentuk adalah dari bar pertama yang direpresentasikan dengan *verteks* pertama.

Jalur tersebut melewati setiap pegas yang direpresentasikan dengan *edge* untuk menuju bar-bar yang terhubung atau *verteks-verteks* yang terhubung hingga bar terakhir atau *verteks* terakhir. Setiap bar atau *verteks* haruslah terlewati namun tidak semua pegas harus diambil sebagai sebuah jalur. Cukup jalur tertentu yang akan menjadi jalur terpendek menuju bar tujuan.
Algoritma permasalahan jalur terpendek yang diambil adalah jalur terpendek sumber tunggal. Ada dua algoritma jalur terpendek sumber tunggal, yaitu Dijkstra dan juga Bellman-Ford. Secara umum penyelesaian jalur terpendek pada graf batasan adalah menggunakan Bellman-Ford. Hal ini disebabkan karena Bellman-Ford dapat mengatasi permasalahan bobot *edge* yang bernilai negatif, sedangkan Dijkstra tidak, dan nilai batasan yang direpresentasikan dalam bentuk bobot *edge* dapat bernilai negatif.

Pada proses penyusunan pegas, nilai bobot *edge* didapatkan berdasarkan nilai konstanta pegas yang tidak mungkin negatif. Oleh karena itu, pada permasalahan optimasi penyusunan pegas, algoritma Dijkstra dapat digunakan. Dan pada penelitian ini, algoritma Perbaikan Dijkstra [2] digunakan untuk mendapatkan nilai gaya pegas maksimum sebagai representasi susunan pegas yang optimal.
Dengan demikian, dari data masukan, representasi graf akhir hasil dari sistem perbedaan batasan, didapatkan susunan pegas yang optimal seperti pada Gambar 5.



Gambar 5. Contoh Susunan Pegas Optimal

4. Hasil dan Pembahasan

Rangkaian uji coba dibagi menjadi dua bagian, yaitu uji coba kebenaran dan uji coba performa. Uji coba kebenaran dilakukan untuk membuktikan kebenaran hasil implementasi, kemudian uji coba performa dilakukan untuk menguji waktu dan pemakaian memori yang dibutuhkan program untuk melakukan beberapa data uji. Terakhir, dilakukan uji coba perbandingan dengan algoritma-algoritma jalur terpendek yang berbeda.
Dari hasil uji coba akan didapatkan susunan pegas yang optimal dengan memanfaatkan desain algoritma yang telah dibangun tersebut.

A. Uji Coba Kebenaran

Pada uji coba ini, dilakukan pengujian program dengan data masukan berupa graf dengan jumlah *verteks* dan *edge* yang sedikit. Uji coba dilakukan dengan memberi masukan ke program dengan data masukan sesuai pada Tabel 1 serta ditunjukkan pula hasilnya juga. Selanjutnya, dilakukan pemeriksaan secara manual untuk menguji kebenaran hasil keluaran dengan bantuan representasi graf pada Gambar 6 yang didapatkan dari model pertidaksamaan oleh metode sistem perbedaan batasan.

Data dari Tabel 1 untuk baris pertama berisi nilai M=5; N=5 dan d=90. Kemudian untuk sejumlah data masukan jumlah pegas, dimasukkan data setiap pegas yang berisi 2 buah bar yang dihubungkan oleh kedua ujung pegas, serta nilai konstanta pegas. Kemudian data masukan dimodelkan ke dalam sistem perbedaan batasan seperti Persamaan (8).

$$\begin{pmatrix} -1 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} \le \begin{pmatrix} 1/20 \\ 1/25 \\ 1/25 \\ 1/50 \\ 1/50 \\ 1/5 \\ 1/25 \end{pmatrix} \quad (8)$$

Sehingga menghasilkan pertidaksamaan (9).

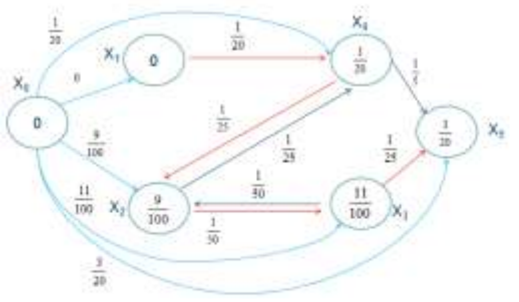
$$x_4 - x_1 \le \frac{1}{20} \qquad x_4 - x_3 \le \frac{1}{50}$$
$$x_4 - x_2 \le \frac{1}{25} \qquad x_5 - x_3 \le \frac{1}{1} \qquad (9)$$
$$x_5 - x_4 \le \frac{1}{25}$$

Tabel 1. Contoh data masukan dan keluaran

Data Masukan	N	M	d
	5	5	90
untuk M=5	Bar awal	Bar Akhir	K
	0	3	20
	1	2	50
	1	3	25
	2	4	25
	3	4	5
Data Keluaran	Fmaks (optimum) = 600		

Kemudian, dari sistem pertidaksamaan tersebut, dibentuk graf batasan seperti pada Gambar 6. Dari graf batasan tersebut dicari nilai optimal dari setiap *verteks* yang merupakan representasi posisi bar dengan memanfaat algoritma jalur terpendek, dimana jalur dari x_1 hingga x_5 yang ditandai dengan warna merah merupakan jalur terpendek yang didapatkan. Sehingga panjang dari setiap pegas yang optimum didapatkan dengan menggunakan rumus dari Persamaan (4).

Gaya pegas maksimum dari susunan pegas yang dihasilkan merupakan nilai terkecil dibandingkan dengan semua kemungkinan yang lain , dimana gaya pegas yang didapatkan dengan rumus pada Persamaan (5) dengan $sum(\frac{1}{k})$ merupkan jalur terpendek dari graf batasan tersebut, sehingga didapatkan gaya pegas maksimal yang optimal.

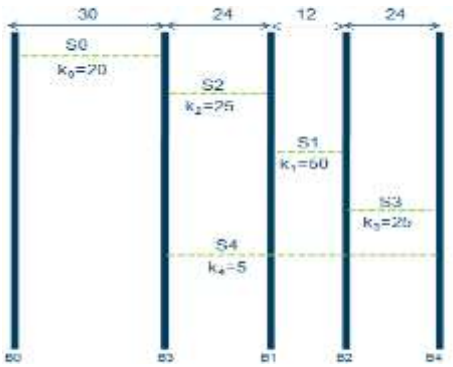


Gambar 6. Representasi graf batasan uji coba dan jalur terpendeknya

Persamaan (10) merupakan hasil akhir perhitungan dari nilai gaya pegas yang didapatkan dan hasilnya sesuai dengan data keluaran dari program.

$$F = \frac{d}{sum(\frac{1}{k})} = \frac{90}{\frac{3}{20}} = 90 \times \frac{20}{3} = 600 \qquad (10)$$

Hasil representasi susunan pegas yang optimal pada uji coba kebenaran digambarkan pada Gambar 7.



Gambar 7. Representasi susunan pegas optimal

B. Uji Coba Performa pada SPOJ

Pada dasarnya, uji coba pada situs SPOJ tidak sekedar uji coba performa, namun sekaligus menjadi uji coba kebenaran karena dengan mendapatkan umpan balik *accepted*, maka hasil implementasi telah dinyatakan benar untuk setiap kasus data uji yang ada pada soal. Uji coba dilakukan dengan mengguggah kode sumber hasil implementasi algoritma pada tesis ini pada salah satu soal di SPOJ yang berjudul "*Spring Loaded*" [7].

Data masukan didapatkan dari server SPOJ pada soal tersebut, sehingga tidak dapat diketahui keseluruhan data masukan yang digunakan.

Pengujian dilakukan beberapa kali untuk mendapat rata-rata waktu eksekusi program pada soal "*Spring Loaded*". Pada Gambar 8 ditunjukkan grafik hasil beberapa kali uji coba pada situs SPOJ. Performa ditunjukkan dalam bentuk waktu eksekusi program. Didapatkan rata-rata waktu eksekusi program pada situs SPOJ sebesar 0,05 detik dengan standar deviasi sebesar 0,01 detik.

Gambar 8. Hasil uji coba pada situs SPOJ

C. Uji Coba Perbandingan Algoritma Jalur Terpendek yang Berbeda

Pada uji coba ini dilakukan perbandingan kecepatan eksekusi program antara 2 buah algoritma jalur terpendek sumber tunggal, yaitu algoritma Perbaikan Dijkstra dan juga algoritma Bellman-Ford.

Kedua algoritma ini pada dasarnya mampu menyelesaikan permasalahan penyusunan pegas dan keduanya telah diujicobakan pada situs SPOJ dan sama-sama memberikan umpan balik *accpeted*. Namun terdapat perbedaan yang cukup besar pada waktu eksekusi program.

Penghematan waktu yang didapatkan dengan menggunakan algoritma Dijkstra dibanding algoritma Bellman-Ford rata-rata hingga mencapai 83,55%. Perhitungan penghematan waktu dirumuskan dalam Persamaan (11).

$$H = \frac{W_B - W_D}{W_B} \times 100\% \qquad (11)$$

Dimana:
H adalah penghematan waktu eksekusi
W_B adalah waktu eksekusi dengan algoritma Bellman-Ford
W_D adalah waktu eksekusi dengan algoritma Dijkstra.

Pada Gambar 9 ditunjukkan grafik hasil perbandingan uji coba pada algoritma dalam tesis ini yang mengacu pada Tabel 2. Sumbu-x menunjukkan tanggal percobaan saat mengunggah program ke situs SPOJ dan sumbu-y menunjukkan nilai waktu dalam detik menggunakan skala 2 desimal, dengan "Bellman-Ford" merupakan hasil algoritma Bellman-Ford yang digunakan sebagai algoritma jalur terpendek dalam penyelesaianya dan "Perbaikan Dijkstra" merupakan hasil algoritma Perbaikan Dijkstra yang digunakan sebagai algoritma jalur terpendek dalam penyelesaianya.

Gambar 9. Grafik perbandingan algoritma

Tabel 2. Tabel Uji perbandingan algoritma

Tanggal Percobaan	**Waktu (detik)** Perbaikan Dijkstra	**Eksekusi** Bellman-Ford	**Penghe-matan Waktu (%)**
01/12/2013	0.05	0.32	84.38
01/12/2013	0.04	0.31	87.10
25/12/2013	0.07	0.31	77.42
26/12/2013	0.07	0.31	77.42
26/12/2013	0.04	0.32	87.50
27/12/2013	0.05	0.31	83.87
01/01/2014	0.04	0.31	87.10
02/01/2014	0.06	0.31	80.65
02/01/2014	0.04	0.32	87.50
03/01/2014	0.05	0.31	83.87
03/01/2014	0.06	0.31	80.65
05/01/2014	0.05	0.31	83.87
06/01/2014	0.05	0.33	84.85
Rata-rata			83.55

5. Kesimpulan

Dalam penelitian ini dibangun desain algoritma optimasi penyusunan pegas dengan menggunakan konsep graf dalam pemodelannya, kemudian menggunakan metode sistem perbedaan batasan, serta algoritma jalur terpendek dalam melakukan optimasi penyusunannya.

Berdasarkan hasil uji coba dan analisis yang dilakukan, maka diambil kesimpulan. Pertama, permasalahan Penyusunan Pegas dapat dimodelkan dalam bentuk graf dan dapat diselesaikan dengan sistem perbedaan batasan dan algoritma jalur terpendek. Dan telah teruji kebenarannya setelah algoritma tersebut diimplementasikan. Kedua, Algoritma yang diusulkan lebih efisien dari segi waktu eksekusi program dengan menggunakan algoritma Perbaikan Dijkstra sebagai algoritma jalur terpendek. Penghematan waktu eksekusi rata-rata mencapai 83,55% dibandingkan dengan menggunakan algoritma Bellman-Ford.

DAFTAR PUSTAKA

[1] Cormen, Thomas H., Leiserson, Charles E., Rivest, Ronald L. dan Stein, Clifford. [2001]. *Introduction to Algorithms*, Second Edition. MIT Press.

[2] Wang Shu-Xi [2012], "The Improved Dijkstra's Shortest Path Algorithm and Its Application", *ELSEVIER: Procedia Enginering* 29, pp 1186-1190

[3] A.V. Goldberg, T. Radzik [1993], "A Heuristic Improvement Of The Bellman–Ford Algorithm", *AMLETS: Appl. Math. Lett.* 6, pp 3–6.

[4] D. Cantone, S. Faro [2004], "Two-Levels-Greedy: A Generalization Of Dijkstra's Shortest Path Algorithm", *Electron. Notes Discrete Math.* 17, pp 81–86.

[5] T. Takaoka [2004], "A Faster Algorithm For The All-Pairs Shortest Path Problem And Its Application, In: K.Y. Chwa, J.I. Munro (Eds.)", *COCOON, In: Lecture Notes in Computer Science*, vol. 3106, Springer, pp. 278–289.

[6] J. Fakcharoenphol [2006], "S. Rao, Planar Graphs, Negative Weight *Edges*, Shortest Paths, And Near Linear Time", *J. Comput. System Sci.* 72, pp 868–889.

[7] Anonim, [2013]. Sphere Online Judge. *SPRING LOADED*. http://www.spoj.com/problems/SPRING/, diakses tanggal 22 September 2013