


BUKU PANDUAN

Program Minimum Cost Transportation



Greedy - Brute Force
Kelompok Olaf

☐ MENAMPILKAN HASIL SQUARE DAN L

☒ RANDOM MATRIKS

☐ LOAD MATRIKS

☐ SAVE MATRIKS

Baris Cost
3

Kolom Cost
3

Stock \ Depot	B1	B2	B3	Supply
A1	56	52	83	7
A2	63	61	40	5
A3	23	43	16	18
Demand	10	10	10	30

HITUNG BIAYA TRANSPORTASI

Kelompok Olaf

- **Aditya Sudyana (23222063)**
- **Atikasari (23222049)**
- **Ananda Sadam Firdaus (23222075)**
- **Lukman Eka Arifandhi (23222075)**
- **Naufal Fahmi Zakiuddin (23222070)**

DAFTAR ISI

PENJELASAN APLIKASI	1
TUTORIAL PENGGUNAAN	2
INTERPRETASI TAMPILAN HASIL PERHITUNGAN.....	5

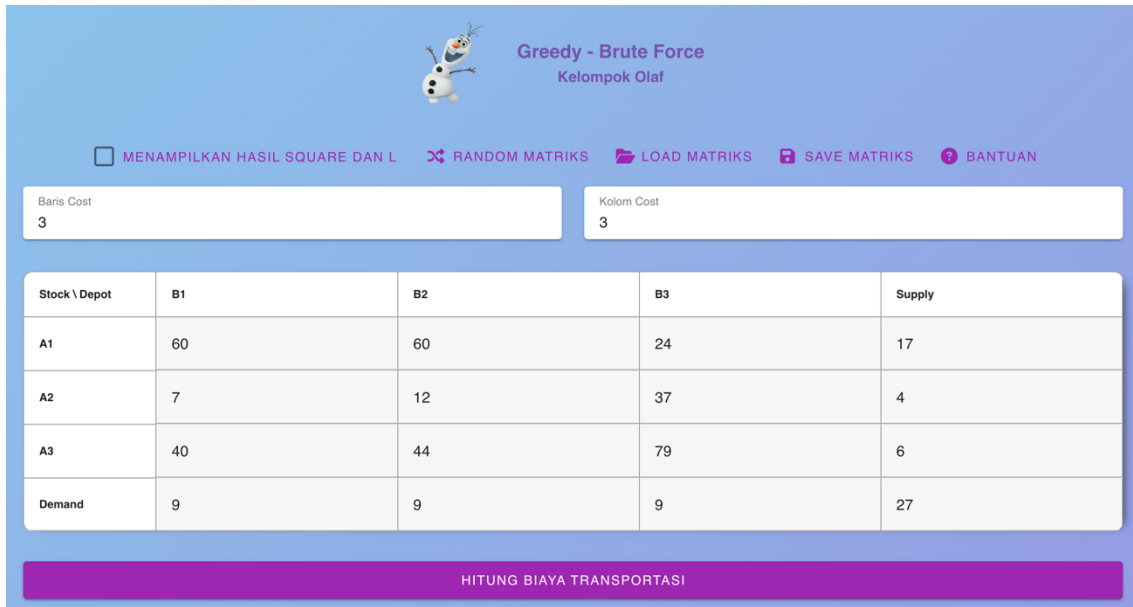
PENJELASAN APLIKASI

Aplikasi yang kami bangun untuk menyelesaikan permasalahan minimum cost transportation adalah berbasis aplikasi web sehingga dapat diakses dan digunakan secara online melalui web browser. Aplikasi web yang kami kembangkan menggunakan Frontend Framework **Vue JS** (<https://vuejs.org/>) dengan bahasa pemrograman **Javascript**. Peran Vue JS adalah sebagai kerangka untuk membangun antarmuka saja sehingga pengguna bisa mengoperasikan permasalahan transportasi ini dengan mudah seperti membangun matriks cost, mengisi tiap-tiap costnya, dan membaca interpretasi dari hasil perhitungan minimum costnya. Sementara itu, bahasa pemrograman inti untuk mengkomputasi minimum costnya 100% menggunakan bahasa pemrograman Javascript. Karena buku panduan ini hanya menjelaskan tentang cara penggunaan aplikasinya saja, terkait hal teknis lainnya sudah kami jabarkan lebih lengkap pada repository yang sudah kami buat pada https://github.com/aditsud/greedy_bruteforce.

Algoritma yang kami gunakan untuk menyelesaikan permasalahan minimum cost transportation pada aplikasi ini adalah menggunakan kombinasi metode Vogels Approximation dan Bruteforce. Karena aplikasi web berjalan pada engine web browser, kinerja dari komputasi pada program ini akan sangat bergantung terhadap spesifikasi perangkat dan web browser masing-masing pengguna. Aplikasi web yang telah kami bangun dapat diakses pada alamat berikut ini <https://greedy-olaf.herokuapp.com/>.

TUTORIAL PENGGUNAAN

Setelah Anda mengakses tautan yang mengarah ke aplikasi ini, maka akan muncul tampilan seperti gambar berikut ini.



Greedy - Brute Force
Kelompok Olaf

☐ MENAMPILKAN HASIL SQUARE DAN L

Baris Cost
3

Kolom Cost
3

Stock \ Depot	B1	B2	B3	Supply
A1	60	60	24	17
A2	7	12	37	4
A3	40	44	79	6
Demand	9	9	9	27

HITUNG BIAYA TRANSPORTASI

Pada bagian atas halaman, terdapat lima tombol pengaturan yang bisa Anda gunakan untuk membantu memberikan kemudahan dalam penggunaan aplikasi ini. Kelima tombol tersebut secara lengkap dijelaskan pada tabel di bawah ini.

DESKRIPSI

Menampilkan Hasil Square dan L

Jika ini dicentang, maka pada hasil perhitungan minimum costnya nanti akan ditampilkan grafis-grafis yang menunjukkan pola Square dan pola L apabila ditemukan pada hasil perhitungan Vogels Approximation dan/atau Stepping Stone (Brute Force)

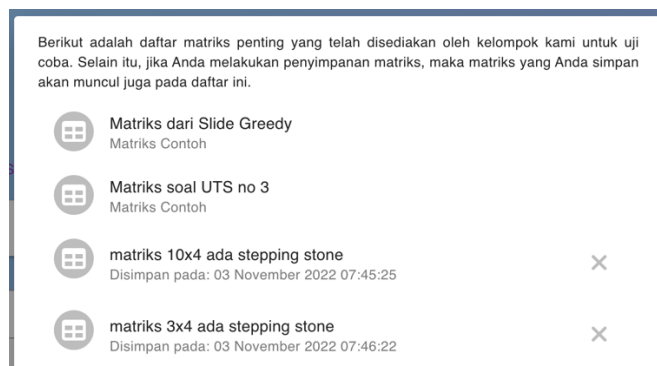
DESKRIPSI

Random Matriks

Jika tombol ini ditekan, maka matriks cost yang berada di bawahnya akan secara acak menghasilkan nilai cost. Begitu juga dengan nilai supply dan demand yang akan secara otomatis menghasilkan nilai secara acak. Dengan tombol ini, Anda tidak perlu mengisi satu per satu cell pada matriks tersebut

Load Matriks

Jika tombol ini ditekan, maka Anda dapat memunculkan matriks-matriks contoh yang telah kami sediakan secara default pada aplikasi ini. Selain itu, Anda juga dapat memunculkan matriks-matriks yang telah Anda simpan secara manual melalui tombol Save Matriks. Berikut adalah contoh tampilan Load Matriks.

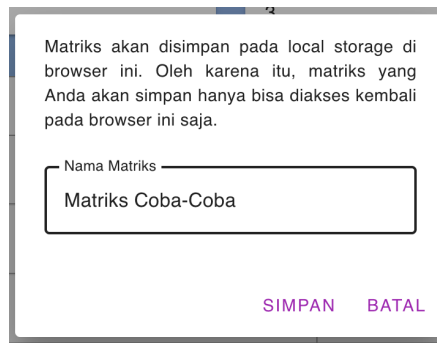


Save Matriks

Jika Anda ingin menyimpan matriks yang sudah Anda ketik, Anda bisa melakukannya dengan menekan tombol Save Matriks. Matriks akan tersimpan pada local storage yang ada di masing-masing web browser sehingga jika Anda ingin memuat matriks yang telah Anda simpan, Anda harus menggunakan web browser yang sama. Saat ingin

DESKRIPSI

menyimpan matriks, Anda hanya diharuskan memberi nama matriks yang ingin Anda simpan seperti pada gambar berikut ini.



Matriks akan disimpan pada local storage di browser ini. Oleh karena itu, matriks yang Anda akan simpan hanya bisa diakses kembali pada browser ini saja.

Nama Matriks

Matriks Coba-Coba

SIMPAN BATAL

Bantuan

Tombol ini berfungsi untuk menampilkan buku pedoman seperti yang sedang Anda lihat saat ini

Selain kelima tombol pengaturan yang sudah kami jelaskan sebelumnya, Anda diharuskan menentukan matriks yang berada di bawah kelima tombol pengaturan tersebut. Matriks harus ditentukan dengan sedemikian rupa sesuai dengan permasalahan biaya transportasi yang ingin Anda pecahkan. Anda juga bisa mengatur jumlah baris dan jumlah kolom dari matriks cost yang ingin Anda gunakan melalui isian Baris Cost dan isian Kolom Cost yang telah disediakan. Selanjutnya, Anda harus menekan tombol **HITUNG BIAYA TRANSPORTASI** untuk menghitung jumlah minimum costnya.

INTERPRETASI TAMPILAN HASIL PERHITUNGAN

Hasil perhitungan minimum cost tertera pada bagian paling atas halaman result seperti pada gambar berikut:

OPTIMUM COST: 761

Matriks Vogels Approximation Cost: 761

Stock \ Depot	B1	B2	B3	B4	Supply
A1	7	87	27 (1)	12 (4)	5
A2	58 (7)	97	95	73	7
A3	99	32 (7)	14 (4)	92	11
Demand	7	7	5	4	23

1. square pants. Different Cost Value: 42
(A1,B3)=1 → (A3,B3)=4 → (A3,B2)=7 → (A1,B2)=0

2. square pants. Different Cost Value: 93
(A1,B4)=4 → (A1,B3)=1 → (A3,B3)=4 → (A3,B4)=0

Sementara (satu atau beberapa) tabel yang akan muncul di bawah nilai **optimum cost** adalah riwayat tabel yang menjelaskan asal muasal mengapa muncul angka optimum cost seperti demikian. Adapun beberapa kondisi matriks yang mungkin akan muncul pada hasil perhitungan pada aplikasi ini yaitu sebagai berikut:

(1) Matriks hasil perhitungan vogels approximation tanpa ditemukan pola Square dan pola L.

Matriks Vogels Approximation Cost: 140

Stock \ Depot	B1	B2	B3	B4	Supply
A1	2	3	5 (10)	6	10
A2	3	2 (10)	4	5	10
A3	3 (10)	5	7	4 (10)	20
Demand	10	10	10	10	40

(2) Matriks hasil perhitungan vogels approximation dengan ditemukan pola Square dan/atau pola L tetapi tanpa matriks hasil perhitungan stepping stone dikarenakan pola Square dan pola L yang ditemukan tidak menghasilkan perbedaan nilai cost yang di bawah nol.

Matriks Vogels Approximation Cost: 761

Stock \ Depot	B1	B2	B3	B4	Supply
A1	7	87	27 (1)	12 (4)	5
A2	58 (7)	97	95	73	7
A3	99	32 (7)	14 (4)	92	11
Demand	7	7	5	4	23

1. square pants. Different Cost Value: 42
(A1,B3)=1 → (A3,B3)=4 → (A3,B2)=7 → (A1,B2)=0
2. square pants. Different Cost Value: 93
(A1,B4)=4 → (A1,B3)=1 → (A3,B3)=4 → (A3,B4)=0

atau jika menggunakan pengaturan tampilan pola Square dan/atau pola L menjadi:

Matriks Vogels Approximation Cost: 761

Stock \ Depot	B1	B2	B3	B4	Supply
A1	7	87	27 (1)	12 (4)	5
A2	58 (7)	97	95	73	7
A3	99	32 (7)	14 (4)	92	11
Demand	7	7	5	4	23

1. square pants. Different Cost Value: 42
(A1,B3)=1 → (A3,B3)=4 → (A3,B2)=7 → (A1,B2)=0
2. square pants. Different Cost Value: 93
(A1,B4)=4 → (A1,B3)=1 → (A3,B3)=4 → (A3,B4)=0

(3) Matriks hasil perhitungan vogels approximation dengan pola Square dan/atau pola L beserta (satu atau beberapa) hasil perhitungan matriks hasil perhitungan stepping stone (dengan atau tanpa) ditemukan pola Square dan/atau pola L-nya.

Matriks Vogels Approximation Cost: 99

Stock \ Depot	B1	B2	B3	B4	Supply
A1	2 (6)	3	5 (2)	1	8
A2	7	3	5	1 (10)	10
A3	4	1 (8)	7 (7)	2 (5)	20
Demand	6	8	9	15	38

1. square pants. Different Cost Value: -1
(A2,B4)=10 → (A3,B4)=5 → (A3,B3)=7 → (A2,B3)=0
2. square pants. Different Cost Value: 0
(A1,B1)=6 → (A1,B3)=2 → (A3,B3)=7 → (A3,B1)=0
3. square pants. Different Cost Value: 1
(A1,B3)=2 → (A3,B3)=7 → (A3,B4)=5 → (A1,B4)=0
4. square pants. Different Cost Value: 3
(A2,B4)=10 → (A3,B4)=5 → (A3,B2)=8 → (A2,B2)=0
5. square pants. Different Cost Value: 4
(A1,B3)=2 → (A3,B3)=7 → (A3,B2)=8 → (A1,B2)=0

Stepping Stone Step ke-1 Cost: 92

Stock \ Depot	B1	B2	B3	B4	Supply
A1	2 (6)	3	5 (2)	1	8
A2	7	3	5 (7)	1 (3)	10
A3	4	1 (8)	7	2 (12)	20
Demand	6	8	9	15	38

1. square pants. Different cost value: 0
(A1,B3)=2 → (A2,B3)=7 → (A2,B4)=3 → (A1,B4)=0
2. L pants. Different cost value: 1
(A1,B1)=6 → (A1,B3)=2 → (A2,B3)=7 → (A2,B4)=3 → (A3,B4)=12 → (A3,B1)=0
3. square pants. Different cost value: 1
(A2,B3)=7 → (A2,B4)=3 → (A3,B4)=12 → (A3,B3)=0
4. L pants. Different cost value: 3
(A1,B3)=2 → (A2,B3)=7 → (A2,B4)=3 → (A3,B4)=12 → (A3,B2)=8 → (A1,B2)=0
5. square pants. Different cost value: 3
(A2,B4)=3 → (A3,B4)=12 → (A3,B2)=8 → (A2,B2)=0
6. square pants. Different cost value: 5
(A1,B1)=6 → (A1,B3)=2 → (A2,B3)=7 → (A2,B1)=0

atau jika menggunakan pengaturan tampilan pola Square dan/atau pola L menjadi:

Matriks Vogels Approximation Cost: 99

Stock \ Depot	B1	B2	B3	B4	Supply
A1	2 (6)	3	5 (2)	1	8
A2	7	3	5	1 (10)	10
A3	4	1 (8)	7 (7)	2 (5)	20
Demand	6	8	9	15	38

1. square pants. Different Cost Value: -1
(A2,B4)=10 → (A3,B4)=5 → (A3,B3)=7 → (A2,B3)=0
2. square pants. Different Cost Value: 0
(A1,B1)=6 → (A1,B3)=2 → (A3,B3)=7 → (A3,B1)=0
3. square pants. Different Cost Value: 1
(A1,B3)=2 → (A3,B3)=7 → (A3,B4)=5 → (A1,B4)=0
4. square pants. Different Cost Value: 3
(A2,B4)=10 → (A3,B4)=5 → (A3,B2)=8 → (A2,B2)=0

Stepping Stone Step ke-1 Cost: 92

Stock \ Depot	B1	B2	B3	B4	Supply
A1	2 (6)	3	5 (2)	1	8
A2	7	3	5 (7)	1 (3)	10
A3	4	1 (8)	7	2 (12)	20
Demand	6	8	9	15	38

1. square pants. Different cost value: 0
(A1,B3)=2 → (A2,B3)=7 → (A2,B4)=3 → (A1,B4)=0
2. L pants. Different cost value: 1
(A1,B1)=6 → (A1,B3)=2 → (A2,B3)=7 → (A2,B4)=3 → (A3,B4)=12 → (A3,B1)=0
3. square pants. Different cost value: 1
(A2,B3)=7 → (A2,B4)=3 → (A3,B4)=12 → (A3,B3)=0
4. L pants. Different cost value: 3
(A1,B3)=2 → (A2,B3)=7 → (A2,B4)=3 → (A3,B4)=12 → (A3,B2)=8 → (A1,B2)=0
5. square pants. Different cost value: 3
(A2,B4)=3 → (A3,B4)=12 → (A3,B2)=8 → (A2,B2)=0
6. square pants. Different cost value: 5
(A1,B1)=6 → (A1,B3)=2 → (A2,B3)=7 → (A2,B1)=0

Setelah mengetahui secara umum terkait interpretasi hasil result yang akan muncul setelah proses komputasi tersebut, berikut adalah penjelasan yang lebih detail terkait informasi pola-pola square dan/atau L yang mungkin akan ditemukan di setiap bawah tiap-tiap tabel hasil perhitungan, seperti contoh berikut ini:

1. square pants. Different Cost Value: -1
 $(A2,B4)=10 \rightarrow (A3,B4)=5 \rightarrow (A3,B3)=7 \rightarrow (A2,B3)=0$
2. square pants. Different Cost Value: 0
 $(A1,B1)=6 \rightarrow (A1,B3)=2 \rightarrow (A3,B3)=7 \rightarrow (A3,B1)=0$
3. square pants. Different Cost Value: 1
 $(A1,B3)=2 \rightarrow (A3,B3)=7 \rightarrow (A3,B4)=5 \rightarrow (A1,B4)=0$
4. square pants. Different Cost Value: 3
 $(A2,B4)=10 \rightarrow (A3,B4)=5 \rightarrow (A3,B2)=8 \rightarrow (A2,B2)=0$

Dari contoh di atas, ditemukan 4 pola Square pada suatu perhitungan Stepping Stone. Pada Pola nomor satu, terdapat deskripsi **square pants** yang bermakna pola pada nomor satu termasuk pola Square. Jika deskripsi yang tertulis adalah **L pants**, maka pola tersebut bermakna pola L.

Selanjutnya terdapat deskripsi **Different Cost Value**. Ini merupakan nilai perubahan cost yang diterapkan pada suatu pola saat kita menambahkan satu poin pada titik yang isian vogelsnya 0. Sehingga jika kita menghitung different cost value pada pola nomor satu, menghasilkan nilai $5+2-1-7 = -1$. Sehingga dengan hasil **-1** ini menandakan bahwa jika vogelsnya kita sesuaikan dengan menerapkan pola square pada pola nomor 1 ini, kita mungkin akan mendapatkan nilai optimum cost secara keseluruhan yang lebih optimal lagi. Ilustrasi pola nomor satu yang ditemukan dapat dilihat pada gambar berikut:

Matriks Vogels Approximation Cost: 99

Stock \ Depot	B1	B2	B3	B4	Supply
A1	2 (6)	3	5 (2)	1	8
A2	7	3	5	1 (10)	10
A3	4	1 (8)	7 (7)	2 (5)	20
Demand	6	8	9	15	38

Selain keterangan terkait jenis pola yang ditemukan dan berapa nilai different cost valuenya, selain itu keterangan lainnya yang bisa didapatkan adalah tentang lokasi titik-titik itu berada. Jika melihat pada pola nomor satu, maka titik-titik tersebut berada di $(A2, B4)=10$ lalu bergeser ke $(A3, B4)=5$ lalu bergeser ke $(A3, B3)=7$ dan berakhir di titik $(A2, B3)=0$.

Sekian