PENYELESAIAN *TRAVELLING SALESMAN PROBLEM* (TSP) ASIMETRIS DENGAN ALGORITMA GENETIK *COMMONALITY*

SKRIPSI

Oleh

Valentia Atiyatna
NIM 051810101044

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2012

PENYELESAIAN
TRAVELLING SALESMAN PROBLEM (TSP) ASIMETRIS DENGAN ALGORITMA GENETIK *COMMONALITY*

SKRIPSI

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan Program Studi Matematika (S1)
dan mencapai gelar Sarjana Sains

Oleh

Valentia Atiyatna
NIM 051810101044

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2012

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Ibunda Martina Laurentia Rajiyah, S.Pd dan Ayahanda Drs. Martinus Asyhudi, terima kasih untuk segala cinta, doa, kesabaran, motivasi, dukungan, kerja keras yang tak pernah henti. Skripsi ini sebagai bukti bahwa penulis mampu mewujudkan harapan kedua orang tua;
2. kakak-kakakku, Georgio Agih Nugroho (alm), Christophora Anung Anindita, S.Pd dan Ignatius Budiyono, S.Pd, untuk cinta, kebersamaan, dukungan, dan motivasi supaya penulis terus belajar dan tidak menyerah. Serta si kecil, Sesilia Atyantika dan Raymunda Kinanti Putri untuk tingkah dan celoteh yang melepaskan semua lelah;
3. keluarga besar Djoyomartono dan Atmowiryono di Yogyakarta untuk seluruh doa dan dukungan kepada penulis;
4. Almamater tercinta, tempat penulis belajar untuk menjadi pribadi yang dapat dibanggakan.

MOTO

"Percayalah kepada TUHAN dengan segenap hatimu, dan janganlah bersandar kepada pengertianmu sendiri. *)

"Ngelmu iku, kalakone kanti laku, lekase kalawan kas, tegese kas njantosani, setya budya pangekering dur hangkara.**)

---

*) Amsal 3:5

**) Pasha, L. 2011. *Butir-Butir Kearifan Jawa*. Yogyakarta : IN AzNa Books.

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

nama : Valentia Atiyatna

NIM : 051810101044

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul "Penyelesaian *Travelling Salesman Problem* (TSP) Asimetris dengan Algoritma Genetik *Commonality*" adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi manapun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa ada tekanan dan paksaan dari pihak mana pun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, 2012

Yang menyatakan,

Valentia Atiyatna

NIM 051810101044

SKRIPSI

PENYELESAIAN *TRAVELLING SALESMAN PROBLEM* (TSP) ASIMETRIS DENGAN ALGORITMA GENETIK *COMMONALITY*

Oleh

Valentia Atiyatna

051810101044

Pembimbing

Dosen Pembimbing Utama : Agustina Pradjaningsih, S.Si., M.Si.

Dosen Pembimbing Anggota : Drs. Moh. Hasan, MSc. Ph.D.

PENGESAHAN

Skripsi yang berjudul "Penyelesaian *Travelling Salesman Problem* (TSP) Asimetris dengan Algoritma Genetik *Commonality*" telah diuji dan disahkan pada:

hari, tanggal :

tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember

Tim Penguji:

Ketua,	Sekretaris,
Agustina Pradjaningsih, SSi., MSi. NIP 197108022000032009	Drs. Moh. Hasan, MSc., PhD. NIP 196404041988021001
Penguji I,	Penguji II,
Drs. Rusli Hidayat, MSc. NIP 196610121993031001	Yuliani Setia Dewi, S.Si, M.Si. NIP 197407162000032001

Mengesahkan

Dekan,

Prof. Drs. Kusno, DEA, Ph.D.

NIP 196101081986021001

RINGKASAN

Penyelesaian *Travelling Salesman Problem* (TSP) dengan Algoritma Genetik *Commonality*; Valentia Atiyatna, 051810101044; 2012: 33 halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Travelling Salesman Problem (TSP) adalah suatu pencarian rute perjalanan *n* kota dengan biaya termurah dengan mengunjungi semua kota hanya satu kali. Biaya dapat diasumsikan sebagai jarak, waktu, bahan bakar, dan sebagainya. Berdasarkan biaya, permasalahan TSP dapat dibagi menjadi dua, yaitu TSP-simetris dan TSP-asimetris. Pada TSP-simetris, biaya dari kota 1 ke kota 2 adalah sama dengan biaya dari kota 2 ke kota 1. Sedangkan pada TSP-asimetris, biaya dari kota 1 ke kota 2 tidak sama dengan biaya dari kota 2 ke kota 1.

Algoritma Genetik *Commonality* merupakan suatu pengembangan dari algoritma Genetika. Jika dalam algoritma genetika baku, proses reproduksi dilakukan melalui operator tukar silang dan mutasi. Setiap kromosom dievaluasi dengan menggunakan fungsi *fitness*. Sedangkan pada algoritma genetik *commonality*, operasi tukar silang didefinisikan kembali dalam dua tahap (Chen & Stephen, 1999) : 1) memelihara *common schema* maksimal dari dua induk, dan 2) melengkapi solusi dengan *construction heuristic*. Karena keturunan yang dihasilkan akan lebih baik dibandingkan dengan kedua induknya, maka evolusi akan terjadi tanpa menyatakan seleksi alam secara eksplisit. Dengan kata lain, evolusi terjadi tanpa melibatkan fungsi *fitness*.

Penelitian dilakukan dengan beberapa langkah. Langkah pertama membangkitkan populasi awal (rute awal) menggunakan *construction heuristic* dengan metode AI (*Arbitrary Insertion*). Langkah kedua melakukan reproduksi solusi baru dengan operasi tukar silang. Langkah ketiga membentuk populasi baru (kumpulan rute).

Berdasarkan penelitian yang telah dilakukan, didapatkan hasil bahwa jarak perjalanan terpendeknya sepanjang 1007 km dengan rute 1-13-24-7-20-23-22-19-16-21-25-4-11-9-15-18-14-2-17-8-10-12-5-3-6 dan diperoleh kesimpulan dapat dihasilkan jarak yang sama dengan rute yang berbeda disebabkan adanya proses penyisipan kota-kota yang tidak terdapat pada *Graf Partial Order* menggunakan metode AI.

PRAKATA

Puji syukur ke hadirat Tuhan Yang Maha Esa yang telah menganugerahkan begitu banyak anugerah, sehingga penulis dapat menyelesaikan skripsi dengan judul "*Penyelesaian Travelling Salesman Problem (TSP) dengan Algoritma Genetik Commonality"*. Skripsi ini disusun untuk memenuhi salah satu syarat untuk menyelesaikan pendidikan strata satu (S1) pada Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak, oleh karena itu penulis ingin menyampaikan ucapan terima kasih kepada:

1. Agustina Pradjaningsih, S.Si., M.Si., selaku Dosen Pembimbing Utama dan Dosen Pembimbing Akademik, dan Drs. Moh. Hasan, MSc. Ph.D., selaku Dosen Pembimbing Anggota yang telah banyak memberikan arahan dan bimbingan sehingga skripsi ini terselesaikan dengan baik;
2. Drs. Rusli Hidayat, MSc. dan Yuliani Setia Dewi, S.Si, M.Si. selaku Dosen Penguji, terima kasih atas kritik dan sarannya;
3. Ibu dan bapak serta keluarga di rumah yang selalu memberikan doa;
4. Mas Didit Yulianto untuk cinta, doa, dukungan, semangat dan motivasi;
5. Faiq, Galih, dan Haris 'Enthunk' untuk seluruh bantuan, waktu, dukungan, dan semangat hingga skripsi ini dapat terselesaikan. Banyak cinta untuk kalian;
6. Rekan-rekan seperjuangan di Matematika, khususnya angkatan 2005, untuk kebersamaan dan semangat dari awal hingga akhir. Apa yang pernah kita miliki akan selalu tersimpan di hati;

Penulis juga menerima segala kritik dan saran dari semua pihak demi kesempurnaan skripsi ini. Akhirnya semoga skripsi ini dapat memberikan manfaat bagi penulis khususnya dan bagi pembaca pada umumnya.

Jember, Desember 2012 Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

Halaman

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Pada kehidupan sehari-hari dapat dijumpai berbagai permasalahan dalam bidang transportasi. Salah satu permasalahan tersebut adalah masalah wiraniaga (*The Travelling Salesman Problem*). *Travelling Salesman Problem* yang selanjutnya disebut TSP dapat dinyatakan sebagai pencarian rute perjalanan *n* kota dengan biaya termurah dengan mengunjungi semua kota hanya satu kali. Biaya dapat diasumsikan sebagai jarak, waktu, bahan bakar, dan sebagainya. Berdasarkan biaya, permasalahan *Travelling Salesman Problem* (TSP) dapat dibagi menjadi dua, yaitu TSP-simetris dan TSP-asimetris. Pada TSP-simetris, biaya dari kota 1 ke kota 2 adalah sama dengan biaya dari kota 2 ke kota 1. Sedangkan pada TSP-asimetris, biaya dari kota 1 ke kota 2 tidak sama dengan biaya dari kota 2 ke kota 1.

Permasalahan TSP dapat diselesaikan dengan berbagai metode. Setiyawan (2005) telah menyelesaikan persoalan TSP simetris dengan membandingkan algoritma Greedy dengan algoritma Semut dengan kesimpulan bahwa algoritma Greedy lebih efisien untuk menyelesaikan persoalan TSP. Ilyas (2007) telah menyelesaikan persoalan TSP untuk graf simetris dengan membandingkan algoritma Greedy dengan algoritma Genetika, disimpulkan bahwa tingkat efisien kedua algoritma tergantung pada banyaknya kota yang digunakan. Dewi (2009) telah menyelesaikan persoalan m-TSP untuk graf asimetrik melalui pemrograman integer dengan metode Branch and Bound, dengan hasil bahwa dengan semakin sedikitnya jumlah minimum kota yang harus dikunjungi tiap sales, maka diperoleh total jarak perjalanan yang semakin pendek pula. Namun masih belum bisa dipastikan bahwa total jarak perjalanan yang paling pendek diperoleh saat menggunakan jumlah minimum kota yang harus dikunjungi tiap sales yang paling sedikit.

Algoritma Genetika bekerja menirukan mekanika genetika dan evolusi alam. Mekanisme yang dimaksud adalah pewarisan sifat-sifat induk beserta variasinya kepada keturunan yang dihasilkan, serta perubahan sekuensial bertambah baik dari waktu ke waktu akibat seleksi alam. Algoritma Genetika sendiri telah banyak diterapkan dalam berbagai permasalahan. Ramadhani (2008) menggunakan algoritma Genetika untuk menyelesaikan masalah optimasi. Ayuningtyas (2008) menggunakan algoritma Genetika untuk masalah penugasan.

Algoritma genetik *commonality* merupakan suatu pengembangan dari algoritma Genetika. Jika dalam algoritma genetika baku, proses reproduksi dilakukan melalui operator tukar silang dan mutasi. Setiap kromosom dievaluasi dengan menggunakan fungsi *fitness*. Kromosom dengan nilai *fitness* yang lebih tingi akan mempunyai kesempatan yang lebih besar untuk dipilih menjadi induk dalam proses reproduksi. Sedangkan pada algoritma genetik *commonality*, operasi tukar silang didefinisikan kembali dalam dua tahap (Chen & Stephen, 1999) : 1) memelihara *common schema* maksimal dari dua induk, dan 2) melengkapi solusi dengan *construction heuristic*. Karena keturunan yang dihasilkan akan lebih baik dibandingkan dengan kedua induknya, maka evolusi akan terjadi tanpa menyatakan seleksi alam secara eksplisit. Dengan kata lain, evolusi terjadi tanpa melibatkan fungsi *fitness*.

1.2 Rumusan Masalah

Dari latar belakang permasalahan yang ditulis adalah bagaimana menyelesaikan masalah TSP asimetris dengan menggunakan algoritma genetik *commonality*?

1.3 Batasan Masalah

Beberapa asumsi yang digunakan pada permasalahan ini adalah sebagai berikut:

a. biaya perjalanan yang digunakan berupa jarak antar kota;
b. setiap dua kota (titik) dihubungkan secara langsung oleh dua garis tak berarah dan tidak memuat *loop*.

1.4 Tujuan

Tujuan yang ingin dicapai dalam penulisan skripsi ini adalah untuk mendapatkan penyelesaian masalah TSP asimetris dengan menggunakan algoritma genetik *commonality.* Untuk membantu perhitungan digunakan software *Borland Delphi*.

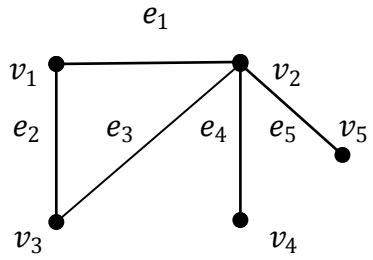
1.5 Manfaat

Manfaat dari penulisan skripsi ini yaitu dapat memberikan alternatif metode dalam menyelesaikan masalah TSP asimetris dengan menggunakan algoritma genetik *commonality* sebagai pengembangan dari algoritma genetika.

BAB 2. TINJAUAN PUSTAKA

2.1 Teori Graf

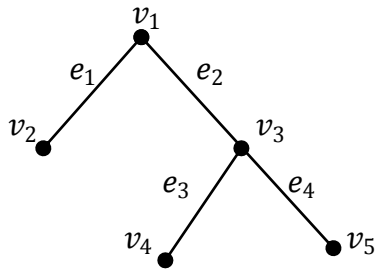
Graf berarah (*directed graph*) didefinisikan secara abstrak sebagai suatu pasangan terurut (*V, E*), dengan *V* suatu himpunan dan *E* suatu relasi biner pada *V*. Graf berarah dapat digambarkan secara geometris sebagai suatu himpunan titik-titik *V* dengan suatu himpunan tanda panah *E* antara pasangan titik-titik (Liu, 1995). Contoh graf berarah dapat dilihat pada gambar 2.1.



Gambar 2.1 Graf berarah (*directed graph*)

Unsur-unsur di dalam *V* dinamakan verteks (*vertex*), sedangkan pasangan terurut di dalam *E* dinamakan rusuk (*edge*) graf berarah tersebut. Sebuah rusuk dikatakan berinsidensi (*incident*) dengan kedua verteks yang dihubungkannya.

Graf tak berarah (*undirected graph*) didefinisikan secara abstrak sebagai suatu pasangan terurut (*V, E*), dengan *V* suatu himpunan dan *E* suatu himpunan yang unsur-unsurnya berupa multihimpunan dengan dua unsur dari *V*. Graf tak berarah dapat direpresentasikan secara geometrik sebagai suatu himpunan titik-titik *V* dengan suatu himpunan garis-garis *E* antara titik-titik tersebut (Liu, 1995). Contoh graf tak berarah dapat dilihat pada gambar 2.2.



Gambar 2.2 Graf tak berarah (*undirected graph*)

2.2 *Travelling Salesman Problem* (TSP) Asimetris

Masalah perjalanan salesman merupakan salah satu persoalan optimasi untuk mendapatkan rute terpendek yang harus dilalui oleh seorang salesman. Salesman tersebut harus mengunjungi sejumlah kota tepat satu kali untuk tiap kota dan kembali ke kota asal dengan biaya perjalanan yang minimum. Biaya tersebut berupa jarak antar kota.

Terdapat dua jenis TSP, yaitu simetris dan asimetris. Pada TSP simetris, biaya dari kota 1 ke kota 2 sama dengan biaya dari kota 2 ke kota 1. Sedang pada TSP asimetris, biaya dari kota 1 ke kota 2 tidak sama dengan biaya dari kota 2 ke kota 1. Untuk TSP asimetris, jumlah jalur yang mungkin adalah permutasi dari jumlah kota yang harus dilalui dibagi dengan jumlah kota yang harus dilalui. Hal ini dapat dipahami karena secara siklus, sebuah jalur dengan urutan 1-2-3 adalah sama dengan jalur 2-3-1 dan jalur 3-1-2. Tetapi jalur dengan urutan 1-2-3 tidaklah sama dengan jalur 3-2-1. Pada TSP asimetris jumlah jalur yang mungkin dapat diperoleh dengan menggunakan rumus permutasi :

$$ {}_nP_k = \frac{n!}{n \times (n-k)!} \tag{2.1} $$

dimana : n = jumlah seluruh kota

k = jumlah kota yang diseleksi

(Suyanto, 2005)

2.3 Algoritma Genetika

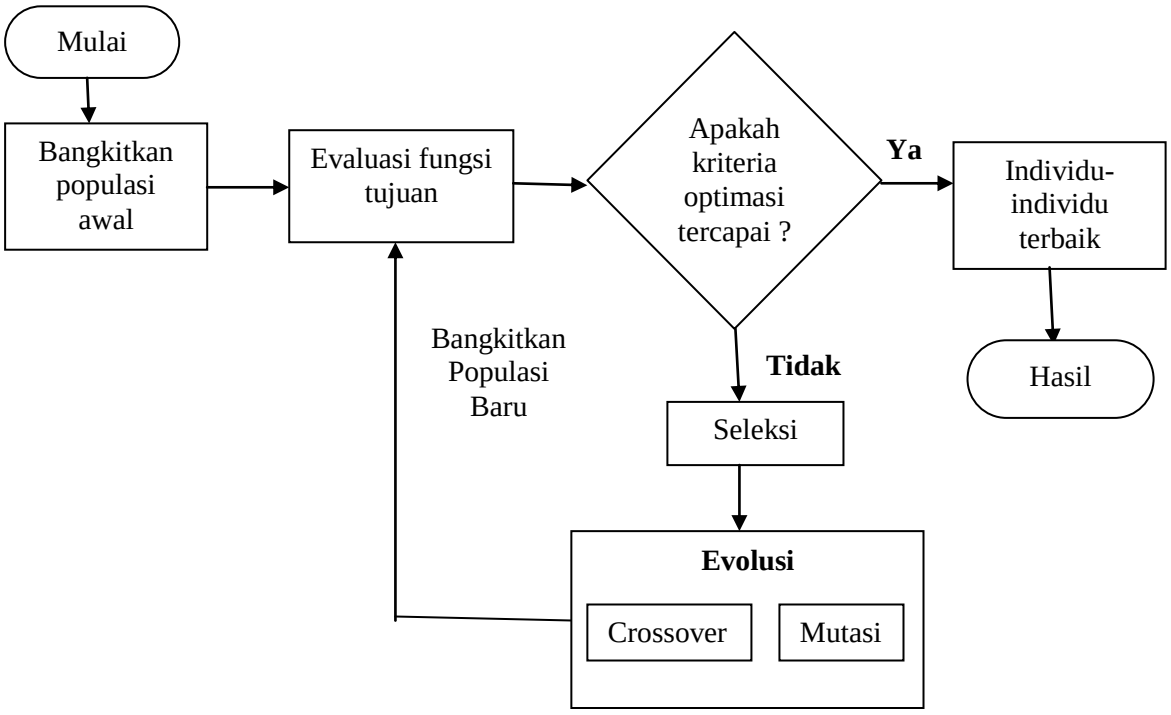
Algoritma genetika adalah algoritma pencarian heuristik yang didasarkan atas mekanisme evolusi biologis. Keberagaman pada evolusi biologis adalah variasi dari kromosom antar individu organisme. Variasi kromosom ini akan mempengaruhi laju reproduksi dan tingkat kemampuan organisme untuk tetap hidup.

Algoritma genetika pertama kali dikembangkan oleh Holland (1975) yang mengatakan bahwa setiap masalah yang berbentuk adaptasi (alami maupun buatan) dapat diformulasikan dalam terminologi genetika. Algoritma genetika adalah simulasi dari proses evolusi Darwin dan operasi genetika atas kromosom.

Pada algoritma genetika, teknik pencarian dilakukan sekaligus atas sejumlah solusi yang mungkin dikenal dengan istilah populasi. Individu yang terdapat dalam satu populasi disebut dengan istilah kromosom. Kromosom ini merupakan suatu solusi yang masih berbentuk simbol. Populasi awal dibangun secara acak, sedangkan populasi berikutnya merupakan hasil evolusi kromosom-kromosom melalui iterasi yang disebut dengan generasi. Pada setiap generasi, kromosom akan melalui proses evaluasi dengan menggunakan alat ukur yang disebut dengan fungsi fitness. Nilai fitness dari suatu kromosom akan menunjukkan kualitas dari kromosom dalam populasi tersebut. Generasi berikutnya dikenal dengan istilah anak (*offspring*) terbentuk dari gabungan dua kromosom generasi sekarang yang bertindak sebagai induk (*parent*) dengan menggunakan operator penyilangan (*crossover*). Selain operator penyilangan, suatu kromosom dapat juga dimodifikasi dengan menggunakan operator mutasi. Populasi generasi yang baru dibentuk dengan cara menyeleksi nilai fitness dari kromosom induk (*parent*) dan nilai fitness dari kromosom anak (*offspring*), serta menolak kromosom-kromosom yang lainnya sehingga ukuran populasi (jumlah kromosom dalam suatu populasi) konstan. Setelah melalui beberapa generasi, maka algoritma ini akan konvergen ke kromosom terbaik.

Struktur umum algoritma genetika dapat dituliskan secara sederhana sebagai berikut :

1. Inisialisasi populasi awal (secara acak atau berdasarkan kriteria tertentu).
2. Evaluasi nilai *fitness* dari setiap kromosom.
3. Seleksi, digunakan untuk mendapatkan kumpulan kromosom baru calon anggota populasi berikutnya.
4. *Crossover*, menghasilkan kromosom baru kombinasi dari induknya.
5. Mutasi, dilakukan dengan mengubah karakter-karakter dari kromosom secara acak.
6. Ulangi langkah 2-5 sampai konvergen ke kromosom terbaik atau sejumlah iterasi telah dilakukan.



Gambar 2.3 Diagram alir algoritma genetika sederhana

2.4 Komponen-komponen Utama Algoritma Genetika

2.4.1 Pengkodean

Pengkodean adalah suatu proses yang sulit dalam algoritma genetika. Hal ini disebabkan karena proses pengkodean untuk setiap permasalahan berbeda-beda karena tidak semua teknik pengkodean cocok untuk setiap permasalahan. Proses pengkodean ini menghasilkan suatu deretan yang kemudian disebut kromosom. Kromosom terdiri dari sekumpulan bit yang dikenal sebagai gen. Ada beberapa macam teknik pengkodean yang dapat dilakukan dalam algoritma genetika (Suyanto, 2005), diantaranya pengkodean biner (*binary encoding*), pengkodean permutasi (*permutation encoding*), pengkodean nilai (*value encoding*), dan pengkodean pohon (*tree encoding*).

Dalam TSP teknik pengkodean yang digunakan adalah pengkodean permutasi (*permutation encoding*), dimana suatu kromosom merepresentasikan suatu permutasi dari nomor urut kota 1, 2, 3, ..., *n*. Contoh pengkodean untuk TSP dapat dilihat pada gambar 2.4.

	1	**2**	**3**	**4**	**5**
1	-	3	2	5	4
2	2	-	3	5	4
3	3	4	-	4	3
4	2	2	3	-	5
5	5	3	4	4	-

Gambar 2.4 Contoh pengkodean untuk TSP

2.4.2 Fungsi Evaluasi

Suatu individu dievaluasi berdasarkan suatu fungsi tertentu sebagai ukuran performansinya. Di dalam evolusi alam, individu yang bernilai fitness tinggi yang

akan bertahan hidup, sedangkan individu yang bernilai *fitness* rendah akan mati (Suyanto, 2005).

Dalam TSP, masalahnya adalah meminimalkan total biaya (masalah minimasi). Oleh karena itu nilai fitness yang bisa digunakan adalah 1 dibagi dengan total biaya. Dalam hal ini yang dimaksud total biaya adalah jumlah jarak kartesian antara satu kota dengan kota lainnya secara melingkar. Jarak antara kota A dan kota B pada diagram kartesian dapat dihitung dengan rumus :

$$\|A - B\| = \sqrt{(X_A - X_B)^2 + (Y_A - Y_B)^2}$$

(X_A, Y_A) menyatakan posisi koordinat kota A dan (X_B, Y_B) adalah posisi koordinat kota B.

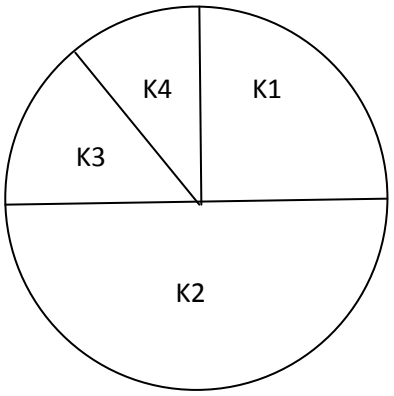
2.4.3 Seleksi

Seleksi akan menentukan individu-individu mana saja yang akan dipilih untuk dilakukan rekombinasi dan bagaimana *offspring* terbentuk dari individu-individu terpilih tersebut. Langkah pertama yang dilakukan dalam seleksi ini adalah pencarian nilai fitness. Masing-masing individu dalam suatu wadah seleksi akan menerima probabilitas reproduksi yang tergantung pada nilai objektif dirinya sendiri terhadap nilai objektif dari semua individu dalam wadah seleksi tersebut.

Suatu metode seleksi yang umum digunakan adalah *roulette-wheel* (roda roulette). Sesuai dengan namanya, metode ini menirukan permainan roulette-wheel dimana masing-masing kromosom menempati potongan lingkaran pada roda roulette secara proporsional sesuai dengan nilai fitnessnya. Kromosom yang memiliki nilai fitness lebih besar menempati potongan lingkaran yang lebih besar dibandingkan dengan kromosom bernilai fitness rendah.

Tabel 2.1 Contoh kromosom dengan nilai fitnessnya

Kromosom	Nilai Fitness
K1	1
K2	2
K3	0,5
K4	0,5
Jumlah	4



Gambar 2.5 Contoh *roulette-wheel* dari tabel 2.1

Metode ini sangat mudah digunakan. Langkah pertama yang dilakukan adalah membuat interval nilai kumulatif (dalam interval [0,1]) dari nilai *fitness* masing-masing kromosom dibagi total nilai *fitness* dari semua kromosom. Sebuah kromosom akan terpilih jika bilangan random yang dibangkitkan berada dalam interval akumulatifnya.

Pada gambar 2.5 , K1 menempati interval nilai kumulatif [0;0,25], K2 berada dalam interval (0,25;0,75], K3 dalam interval (0,75;0,875], dan K4 dalam interval (0,875;1). Misalkan, jika bilangan random yang dibangkitkan adalah 0,6 maka

kromosom K2 terpilih sebagai orang tua. Tetapi jika bilangan random yang dibangkitkan adalah 0,99 maka kromosom K4 yang terpilih (Suyanto, 2005).

2.4.4 Crossover

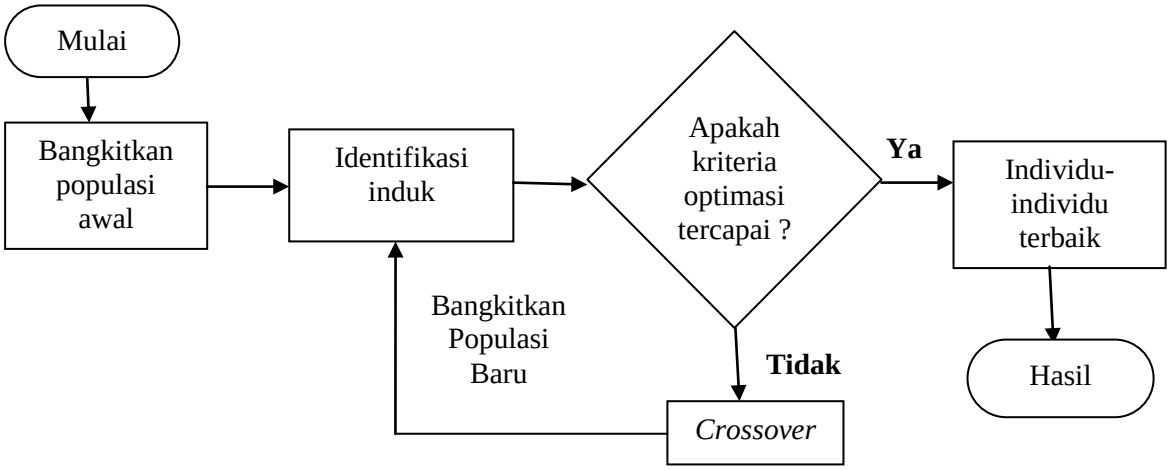
Crossover (penyilangan) dilakukan atas 2 kromosom untuk menghasilkan kromosom anak (*offspring*). Kromosom anak yang terbentuk akan mewarisi sebagian sifat kromosom induknya. Pada *crossover* ada satu parameter yang sangat penting, yaitu peluang *crossover* (p_c). Peluang *crossover* menunjukkan rasio dari anak yang dihasilkan dalam setiap generasi dengan ukuran populasi. Misalkan ukuran populasi (popsize = 100), sedangkan peluang *crossover* (p_c = 0,25), berarti bahwa diharapkan ada 25 kromosom dari 100 kromosom yang ada pada populasi tersebut akan mengalami *crossover*. Pindah silang untuk TSP dapat diimplementasikan dengan skema *order crossover*.

2.4.5 Mutasi

Mutasi adalah proses memodifikasi kromosom anak secara random. Mutasi akan menciptakan individu baru dengan mengubah satu atau lebih gen yang terdapat dalam suatu kromosom. Mutasi berperan untuk menggantikan gen yang hilang dari populasi akibat proses seleksi dan memungkinkan munculnya gen yang tidak ada dalam populasi awal (Kusumadewi & Purnomo, 2005). Pada TSP, operator mutasi biasanya diimplementasikan dengan menukarkan gen termutasi dengan gen lain yang dipilih secara random. Misalnya, kromosom {2, 3, 4, 1, 5} dapat termutasi menjadi kromosom {4, 3, 2, 1, 5}. Dalam hal ini gen 1 dan gen 3 saling ditukarkan. Skema mutasi ini dikenal sebagai *swapping mutation*.

2.5 Algoritma Genetik *Commonality*

Algoritma genetik *commonality* merupakan alternatif lain dalam algoritma genetika, yaitu seleksi berbasis *commonality* (tanpa menggunakan fungsi *fitness*). Pada algoritma genetika baku, evolusi terjadi akibat proses seleksi dan reproduksi. Proses seleksi dilakukan berdasarkan nilai dari suatu fungsi evaluasi yang disebut fungsi *fitness*. Kromosom-kromosom dengan nilai fitness yang tinggi akan mempunyai kesempatan yang besar untuk terpilih menjadi induk dalam proses reproduksi. Proses reproduksi terjadi melalui operasi tukar silang dan operasi mutasi. Pada algoritma genetik *commonality* proses seleksi terjadi secara implisit di dalam operasi tukar silang. Dalam kasus pencarian solusi bagi TSP asimetris, eksplorasi daerah pencarian dilakukan dengan memilih *common schema* dari dua induk. *Common schema* adalah kesamaan pola yang menjelaskan bagian dari *string* yang mempunyai kesamaan posisi tertentu. Eksploitasi yang dilakukan dengan melengkapi *common schema* tersebut akan menjadi sebuah solusi dengan menggunakan *construction heuristic*. *Construction heuristic* tersebut digunakan pada inisialisasi populasi awal. Gambar 2.6 menjelaskan tentang tahapan-tahapan algoritma genetik *commonality*.



Gambar 2.6 Diagram alir algoritma genetik *commonality*

2.5.1 Inisialisasi Populasi Awal

Construction heuristic yang dapat digunakan pada inisialisasi populasi awal adalah metode tetangga terdekat (NN), metode *Arbitrary Insertion* (AI), metode *Convex Hull* / *Arbitrary Insertion* (CH / AI), gabungan metode *Arbitrary Insertion* (AI) dan metode tetangga terdekat (NN), dan gabungan metode *Convex Hull* / *Arbitrary Insertion* (CH / AI) dan metode tetangga terdekat (NN). Dalam inisialisasi populasi awal diambil berbagai simpul (kota) sebagai simpul awal. Diharapkan dengan mengambil berbagai simpul awal akan diperoleh berbagai kombinasi simpul awal yang berbeda. Berbagai kombinasi simpul awal yang berbeda dapat dipandang sebagai representasi berbagai "kelompok kepentingan" dalam kerangka kerja *commonality*.

Langkah-langkah yang dilakukan dalam metode AI adalah :

1. Mengawali hanya dengan sebuah titik *i*
2. Menemukan titik k dengan c_{ik} yang minimal (c_{ik} : biaya atau jarak dari titik *i* ke titik *k*), dan membentuk subtour *i*-*k*-*i*.
3. Memilih secara acak titik *k* yang tidak di dalam subtour untuk masuk ke subtour.
4. Menemukan busur (i, j) dalam subtour dengan $c_{ik} + c_{kj} - c_{ij}$ yang paling minimal. Letakkan *k* diantara *i* dan *j*.
5. Mengulangi langkah 3) hingga diperoleh suatu sikel Hamilton.

Common schemata adalah kumpulan dari beberapa *common schema*. *Common schemata* dari dua solusi yang dibentuk secara *heuristic* adalah solusi parsial yang akan mempunyai proporsi *fit schemata* yang tinggi. Untuk mengukur kemampuan *common schemata* dalam memperbaiki solusi, diasumsikan bahwa probabilitas *construction heuristic* memilih *correct schemata* (*schemata* yang merupakan bagian dari solusi optimal) adalah *p*. Dan probabilitas memilih *incorrect schemata* adalah *1-p*. Jika *heuristic* ini digunakan untuk membangkitkan populasi

awal, maka setiap solusi (induk) diharapkan mempunyai proporsi *correct schemata p*, dan proporsi *incorrect schemata 1-p*. Untuk pemilihan induk secara random, tabel 2.2 memperlihatkan distribusi yang diharapkan dari *correct/ incorrect schemata* dan *common/uncommon schemata* dalam populasi awal.

Tabel 2.2 Distribusi *schemata* yang diharapkan untuk pasangan induk random dalam populasi awal

	p *correct*	$1-p$ *incorrect*
p *correct*	p^2 *common correct*	$p\,(1-p)$ *uncommon*
$1-p$ *incorrect*	$p\,(1-p)$ *uncommon*	$(1-p)^2$ *common incorrect*

Diantara *common schemata*, rasio *correct schemata* terhadap *incorrect schemata* adalah $p^2/(1-p)^2$. Jika *construction heuristic* lebih memilih *correct schemata* daripada *incorrect schemata* $(p > o,5)$, maka $p/(1-p) > 1$ dan

$$\frac{p^2}{(1-p)^2} > \frac{p}{(1-p)} \tag{2.2}$$

Solusi-solusi parsial *common schemata* diharapkan mempunyai rasio *correct schemata* terhadap *incorrect schemata* yang lebih tinggi dibandingkan dengan solusi-solusi dalam populasi awal (induk). Dengan demikian eksploitasi *schemata* dapat dilakukan tanpa menggunakan seleksi berbasis *fitness*.

2.5.2 Kondisi Terminasi

Penghentian evolusi akan dilakukan apabila populasi telah terisi oleh kromosom-kromosom yang identik. Kondisi ini biasa disebut konvergen. Pada

umumnya kondisi ini ditandai dengan telah terlewatinya batasan generasi maksimum. Untuk melihat keadaan konvergensi tersebut, dimisalkan proporsi *correct schemata* dalam populasi awal adalah $\bar{c_0} = 1 - p$. Diasumsikan pula :

1. *Common schemata* yang diperoleh dari dua induk pada populasi awal, dilengkapi dengan *construction heuristic* hingga menjadi sebuah solusi lengkap.
2. Setiap anggota populasi dijodohkan dua kali dengan pasangan random, dan keturunan yang dihasilkan akan mengisi populasi pada generasi berikutnya (algoritma genetik generasional), sehingga *correct schemata* pada generasi j adalah *common correct schemata* pada generasi $j-1$ dan *correct schemata* yang dipilih pada generasi $j-1$.
3. Proporsi *correct schemata* p yang dibangkitkan oleh *construction heuristic* adalah konstan.

Proporsi *correct schemata* c_j yang diharapkan pada generasi j adalah :

$$c_j = c_{j-1}\left(c_{j-1} + 2\,p\,\bar{c}_{j-1}\right) \tag{2.3}$$

Dengan menyatakan $p \in [0,1]$ sebagai $0{,}5\,(1+x)$, $x \in [-1,1]$

$$\frac{c_j}{c_{j-1}} = 1 + x\left(1 - c_{j-1}\right) \tag{2.4}$$

Jika $p > 0{,}5$ maka $x > 0$ dan $c_j > c_{j-1}$. Proporsi *correct schemata* akan bertambah pada setiap generasi jika $p > 0{,}5$.

Lebih jauh lagi, pada saat konvergensi $c* = c_j = c_{j-1}$. Karena itu $c*$ harus memenuhi

$$c*(1-2p) = c*^2\,(1-2p) \tag{2.5}$$

Persamaan (2.4) ini dipenuhi jika $c* = 0$, $c* = 1$, atau $p = 0{,}5$. Untuk $p > 0{,}5$, c_j konvergen ke 1 (semua solusi dalam populasi adalah optimal). Untuk $p < 0{,}5$, c_j konvergen ke 0. Karena itu efektif tidaknya *construction heuristic* akan diperkuat oleh seleksi berbasis *commonality*.

2.5.3 Reproduksi Solusi Baru dengan Operator Tukar Silang

Reproduksi hanya dilakukan melalui operasi tukar silang (*crossover*) tanpa melibatkan operasi mutasi. Dalam operasi ini dilakukan identifikasi *common schema* dari dua induk. Prosedur untuk menentukan *common schema* dari dua induk dideskripsikan sebagai berikut (Chen dan Smith, 1999) :

1. Representasikan kedua induk ke dalam matriks biner;
2. Interseksikan kedua matriks;
3. Jumlahkan kolom-kolom matriks hasil interseksi untuk mendapatkan setiap elemen *predecessor*;
4. Buat graf *partial order*;
    a. Cari elemen yang mempunyai *predecessor* paling sedikit.
    b. Rangkaikan elemen pada *predecessor* dengan *predecessor* terdekat yang paling panjang.
5. Cari lintasan terpanjang dalam graf.

BAB 3. METODOLOGI PENELITIAN

3.1 Data Penelitian

Data yang digunakan dalam penelitian ini berupa sejumlah kota (*n*) dan jarak antar kota dari kota-kota yang akan dikunjungi oleh sales. Data jarak tersebut dibangkitkan sendiri secara acak menggunakan Microsoft Excel. Jumlah kota yang digunakan sebanyak 25 kota dengan jarak antar kota yang dapat dilihat pada lampiran.

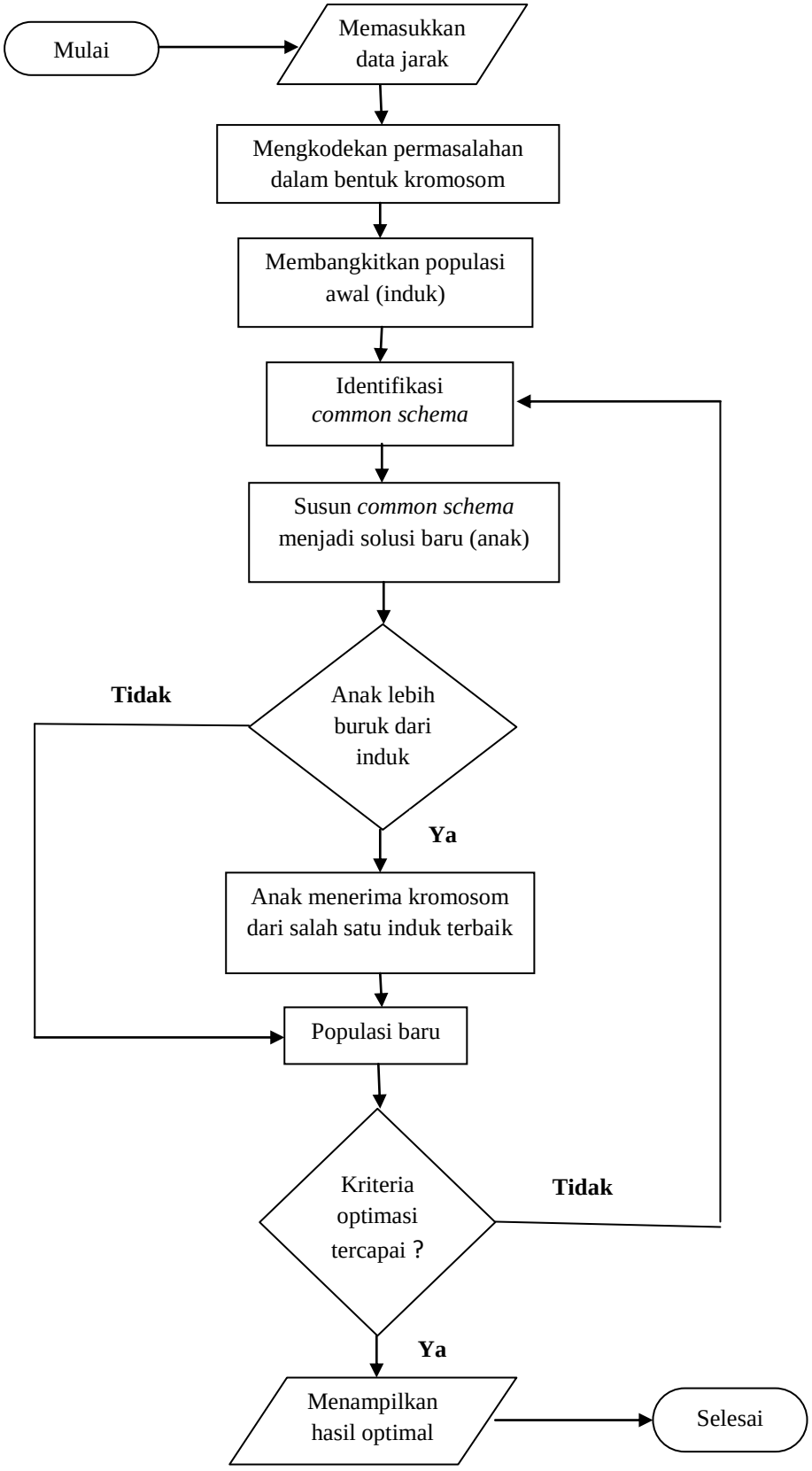
Pada skripsi ini digunakan istilah-istilah dalam ilmu genetika yang merujuk pada penyelesaian TSP itu sendiri. Untuk lebih jelasnya dapat dilihat pada tabel 3.1.

Tabel 3.1 Istilah Ilmu Genetika dalam TSP

Ilmu Genetika	TSP
Gen	Kota
Kromosom	Rute (*tour*)
Induk	Pasangan rute awal
Anak	Rute baru
Populasi	Kumpulan rute yang diperoleh

3.2 Langkah-langkah Penyelesaian

Langkah-langkah penyelesaian TSP asimetris dengan algoritma genetika *commonality* dapat dilihat melalui diagram alir (*flowchart*) berikut :



Gambar 3.1 Diagram alir (*flowchart*) algoritma genetika *commonality*

Diagram alir (*flowchart*) pada gambar 3.1 dapat dijelaskan sebagai berikut:

a. Memasukkan data berupa jarak antar kota
b. Mengkodekan permasalahan TSP asimetris yaitu rute kota yang akan dilewati dalam bentuk kromosom
c. Membangkitkan populasi awal (rute awal) menggunakan *construction heuristic* dengan metode AI
d. Melakukan reproduksi solusi baru dengan operasi tukar silang
    1) Identifikasi *common schema* dari dua induk (pasangan rute awal)
        a) Representasikan kedua induk (pasangan rute awal) ke dalam matriks biner
        b) Interseksikan kedua matriks
        c) Jumlahkan kolom-kolom matriks hasil interseksi untuk mendapatkan setiap elemen *predecessor* (pendahulu)
        d) Buat graf *partial order*
            (1) Cari elemen yang mempunyai *predecessor* paling sedikit
            (2) Rangkailah elemen pada *predecessor* dengan *predecessor* terdekat yang terdekat
        e) Cari lintasan terpanjang dalam graf
    2) Susun *common schema* menjadi solusi baru (anak) yaitu rute baru dengan metode AI
    3) Jika anak (rute baru) lebih buruk dari kedua induknya (rute awal), anak menerima semua kromosom (rute) dari salah satu induknya (rute) yang terbaik
e. Membentuk populasi baru (kumpulan rute)
f. Jika kondisi terminasi belum tercapai, ulangi langkah (d)

BAB 4. HASIL DAN PEMBAHASAN

Pada bab ini dibahas penyelesaian masalah TSP Asimetris menggunakan algoritma genetik *commonality* berdasarkan metode yang terdapat pada bab 3.

4.1 Hasil

4.1.1 Penyelesaian Manual dengan Input 6 Kota

Untuk memberikan gambaran tentang cara kerja algoritma genetik *commonality*, maka akan diberikan contoh penyelesaian secara manual dengan menggunakan sampel data yang kecil berupa 6 buah kota yang diambil dari data pada lampiran A dan disajikan dalam tabel 4.1.

Tabel 4.1 Data jarak 6 buah kota

	1	**2**	**3**	**4**	**5**	**6**
1	-	81	86	64	54	91
2	82	-	96	24	61	41
3	60	89	-	87	93	39
4	85	26	59	-	22	85
5	36	36	40	78	-	34
6	61	72	56	33	41	-

Langkah-langkah penyelesaian TSP Asimetris dengan Algoritma Genetik *Commonality* :

a. Mengkodekan permasalahan TSP Asimetris dalam bentuk kromosom

   Dari permasalahan tersebut, terdapat 6 (enam) kota (*vertex*) yang menjadi gen dalam kromosom (*tour*). Sesuai dengan persamaan (2.1), jika terdapat 6 kota

yang akan dilalui maka terdapat 120 buah jalur yang dapat ditempuh. Sehingga dimungkinkan terdapat 120 buah kromosom dalam satu generasi. Beberapa contoh kromosom tersebut adalah :

Kromosom [1]=[$v_1 v_2 v_3 v_4 v_5 v_6$]

Kromosom [2]=[$v_1 v_2 v_6 v_4\ v_5 v_3$]

Kromosom [3]=[$v_1 v_4 v_2 v_3 v_5\ v_6$]

Kromosom [4]=[$v_1 v_2 v_4 v_3 v_6 v_5$]

Kromosom [5]=[$v_1 v_5 v_2 v_6 v_4 v_3$]

Kromosom [6]=[$v_1 v_4 v_3 v_2\ v_6 v_5$]

b. Membangkitkan populasi awal menggunakan *construction heuristic* dengan metode AI.

Parameter yang dibutuhkan dalam membangkitkan populasi awal adalah *popsize* (ukuran populasi). Dalam algoritma genetika, populasi adalah kumpulan kromosom yang akan bereproduksi. Kromosom yang terbaik akan bertahan, sedangkan yang kurang baik akan hilang. *Popsize* dalam contoh ini adalah enam. Artinya terdapat enam kromosom dalam satu populasi. Populasi awal akan digunakan sebagai induk (pasangan rute awal) dalam algoritma genetik *commonality*.

Untuk membangkitkan populasi awal digunakan metode AI. Langkah-langkah yang dilakukan adalah :

1. Awali hanya dengan sebuah titik *i*
   kita ambil titik 1 (v_1)
2. Temukan titik k dengan c_{ik} yang minimal (c_{ik} : biaya atau jarak dari titik *i* ke titik *k*), dan membentuk subtour *i*-*k*-*i*.
   Dipilih *k* = 5 -> subtour : $v_1 - v_5 - v_1$

3. Memilih secara acak titik *k* yang tidak di dalam subtour untuk masuk ke subtour.

   Dipilih $k = 6$

4. Temukan busur (i, j) dalam subtour dengan $c_{ik} + c_{kj} - c_{ij}$ yang paling minimal. Letakkan *k* diantara *i* dan *j*.

   $c_{16} + c_{65} - c_{15} = 91 + 41 - 54 = 78$

   $c_{15} + c_{56} - c_{16} = 54 + 34 - 91 =$ **-3**

   sehingga subtour menjadi v_1-v_5-v_6

5. ulangi langkah 3

   $k = 3$

   $c_{13} + c_{35} - c_{15} = 86 + 93 - 54 = 125$

   $c_{13} + c_{36} - c_{16} = 86 + 39 - 91 =$ **34**

   sehingga subtour menjadi v_1-v_5-v_3-v_6

6. $k = 4$

   $c_{14} + c_{45} - c_{15} = 64 + 22 - 54 =$ **32**

   $c_{14} + c_{43} - c_{13} = 64 + 59 - 86 = 41$

   $c_{14} + c_{46} - c_{16} = 64 + 85 - 91 = 91$

   sehingga subtour menjadi v_1-v_4-v_5-v_3-v_6

7. $k = 2$

   $c_{12} + c_{24} - c_{14} = 81 + 24 - 64 = 41$

   $c_{12} + c_{25} - c_{15} = 81 + 61 - 54 = 88$

   $c_{12} + c_{23} - c_{13} = 81 + 96 - 86 = 91$

   $c_{12} + c_{26} - c_{16} = 81 + 41 - 91 =$ **31**

   sehingga subtour menjadi v_1-v_4-v_5-v_3-v_2-v_6.

Jadi diperoleh suatu sikel Hamilton, yaitu $v_1 v_4 v_5 v_3 v_2 v_6$.

Karena semua kota telah terlewati maka perhitungan selesai.

Kromosom-kromosom yang diperoleh digunakan sebagai induk (pasangan rute awal).

c. Melakukan reproduksi solusi baru dengan operasi tukar silang

Induk yang kita pilih diambil dari beberapa kromosom. Dua induk (pasangan rute awal) yang kita pilih adalah:

Induk 1 = Kromosom [$v_1 v_4 v_5 v_3 v_2 v_6$] ; jarak = 256

Induk 2 = Kromosom [$v_1 v_4 v_5 v_3 v_2 v_6$] ; jarak = 256

Langkah selanjutnya mengidentifikasi *common schema* dari kedua induk. Tahap pertama dapat dilakukan dengan operator *Maximum Partial Order* (MPO).

Kedua induk tersebut direpresentasikan ke dalam matriks biner dan diinterseksi :

Matriks biner akan mempunyai elemen $(i, j) = 1$ jika *node i* mendahului *node j*. Matriks hasil interseksi dari kedua induk matriks biner induk akan mempunyai elemen $(i, j) = 1$ jika dan hanya jika *node i* mendahului *node j* pada kedua induk. Hasil representasi dua induk dapat dilihat pada gambar 4.1, sedang gambar 4.2 merupakan hasil interseksi dan *predecessor* dari dua induk.

Induk 1

	v_1	v_2	v_3	v_4	v_5	v_6
v_1	0	1	1	1	1	1
v_2	0	0	0	0	0	1
v_3	0	1	0	0	0	1
v_4	0	1	1	0	1	1
v_5	0	1	1	0	0	1
v_6	0	0	0	0	0	0

Induk 2

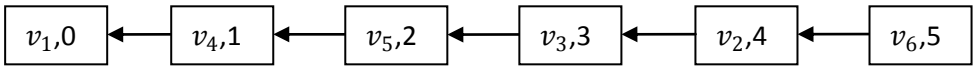
	v_1	v_2	v_3	v_4	v_5	v_6
v_1	0	1	1	1	1	1
v_2	0	0	0	0	0	1
v_3	0	1	0	0	0	1
v_4	0	1	1	0	1	1
v_5	0	1	1	0	0	1
v_6	0	0	0	0	0	0

Gambar 4.1 Representasi matriks biner dua induk

Hasil interseksi : Predecessor : 0 4 3 1 2 5

	v_1	v_2	v_3	v_4	v_5	v_6
v_1	0	1	1	1	1	1
v_2	0	0	0	0	0	1
v_3	0	1	0	0	0	1
v_4	0	1	1	0	1	1
v_5	0	1	1	0	0	1
v_6	0	0	0	0	0	0

Gambar 4.2 Hasil interseksi dan *predecessor* dari dua induk



Dari *predecessor* diperoleh *Maximum Partial Order*: $v_1 v_4 v_5 v_3 v_2 v_6$

Gambar 4.3 *Graf partial order*

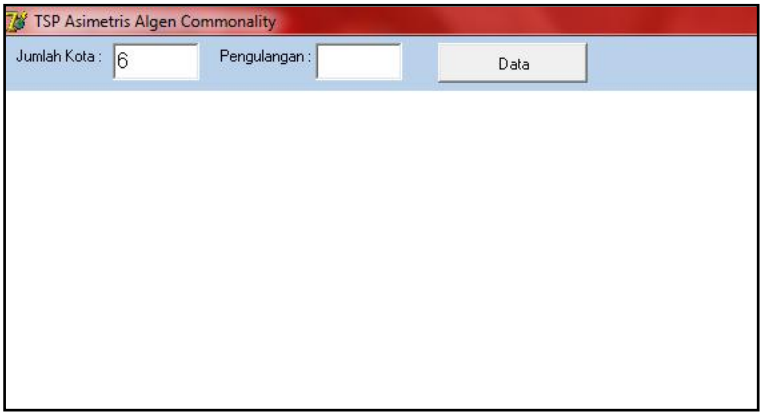
Jika pada *Graf Partial Order* belum membentuk suatu sikel Hamilton, maka elemen-elemen sisa *common schema* disisipkan kembali menggunakan metode AI sehingga menjadi sebuah solusi baru (anak).

Karena pada *Graf Partial Order* sudah terbentuk sikel Hamilton (Gambar 4.3), maka tidak perlu dilakukan proses penyisipan menggunakan metode AI. Pada proses ini dihasilkan kromosom anak $v_1 v_4 v_5 v_3 v_2 v_6$, dengan nilai (jarak) 256 km.

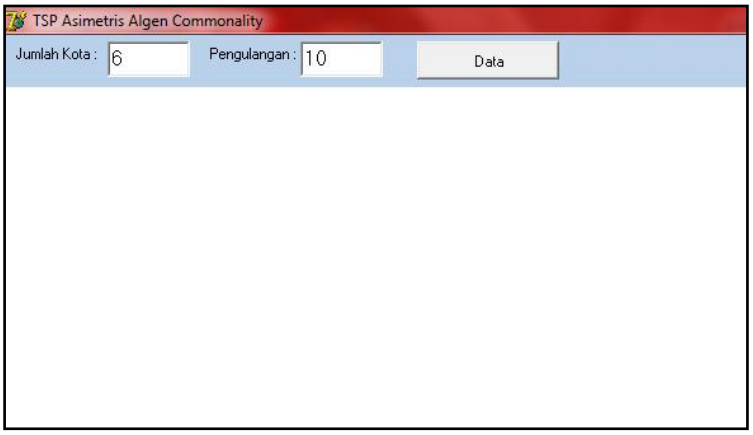
4.1.2 Penyelesaian dengan Menggunakan Program untuk 6 Kota

Untuk mengetahui kesamaan nilai yang dihasilkan, maka data 6 kota diujikan ke dalam program. Langkah-langkah untuk menjalankan programnya adalah:

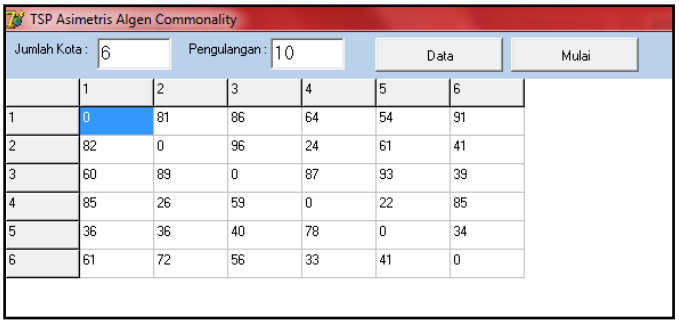
a. Masukkan jumlah kota yang akan dilakukan pada kotak “Jumlah Kota”;



b. Masukkan jumlah pengulangan yang akan diuji pada kotak “Pengulangan”;

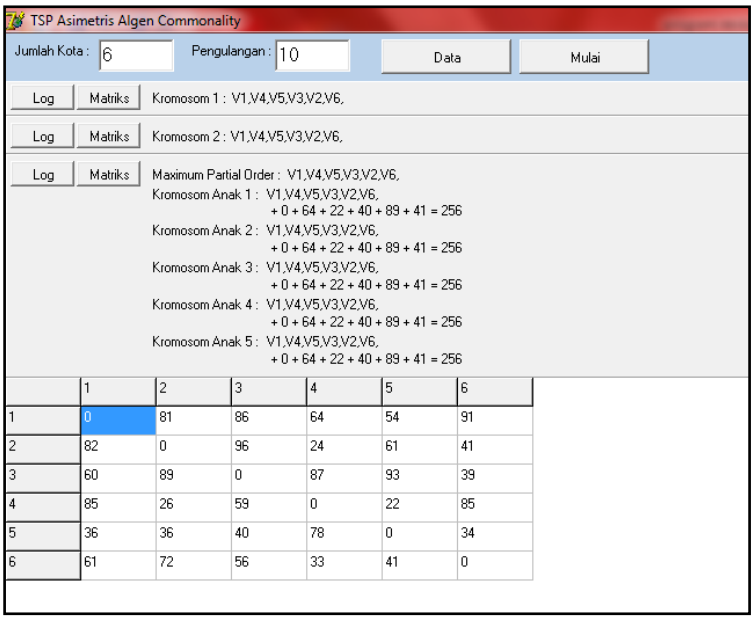


c. Tekan tombol “Data” sehingga akan muncul data *n* kota yang akan diuji;

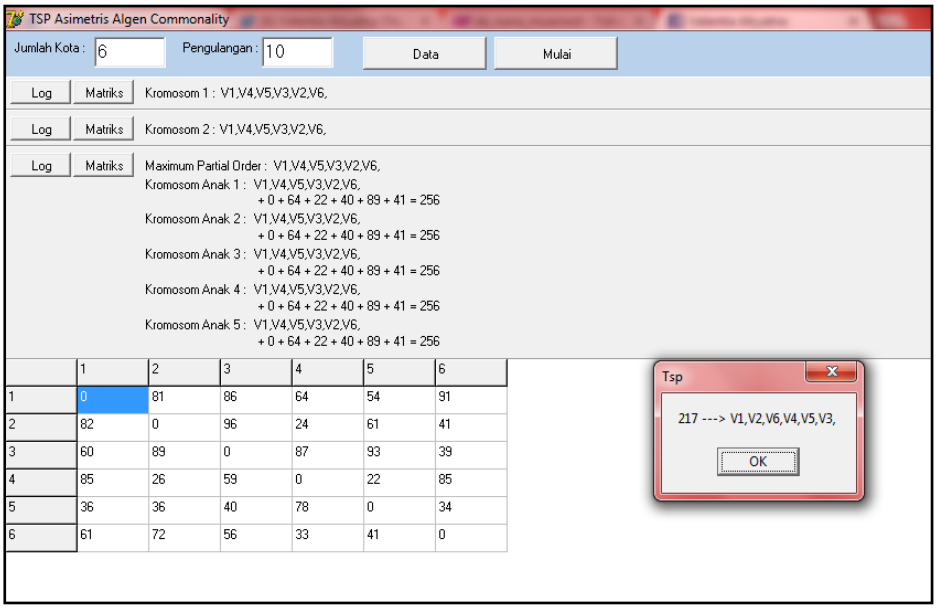


	1	2	3	4	5	6
1	0	81	86	64	54	91
2	82	0	96	24	61	41
3	60	89	0	87	93	39
4	85	26	59	0	22	85
5	36	36	40	78	0	34
6	61	72	56	33	41	0

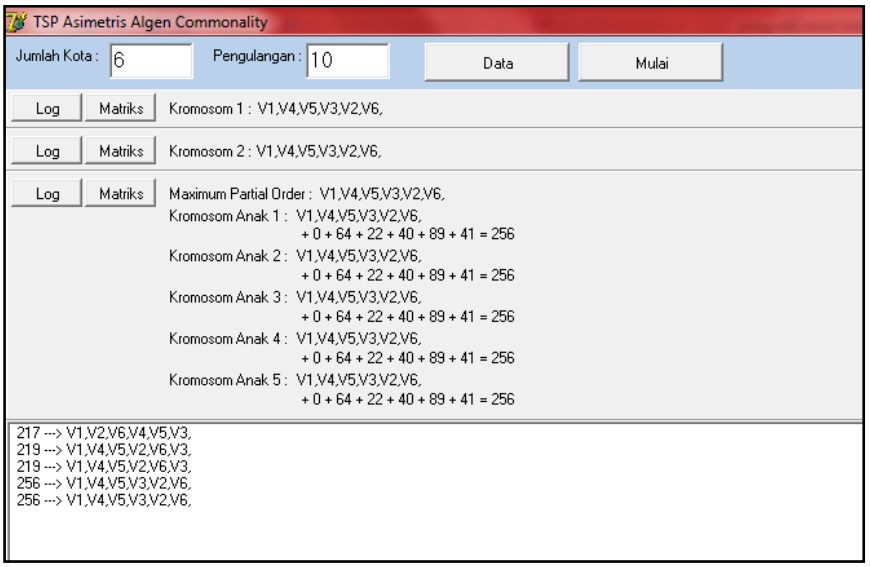
d. Tekan tombol “Mulai” untuk menjalankan program sehingga akan muncul kromosom 1 dan kromosom 2 yang menjadi induk (rute awal), dan 5 kromosom baru yang menjadi solusi (rute akhir) dalam permasalahan ini;



e. Pada akhir proses akan muncul jarak paling minimal yang didapat dari hasil seleksi beberapa kali pengulangan program;



f. Tekan tombol “OK” untuk melihat semua jarak dan rute yang dihasilkan.



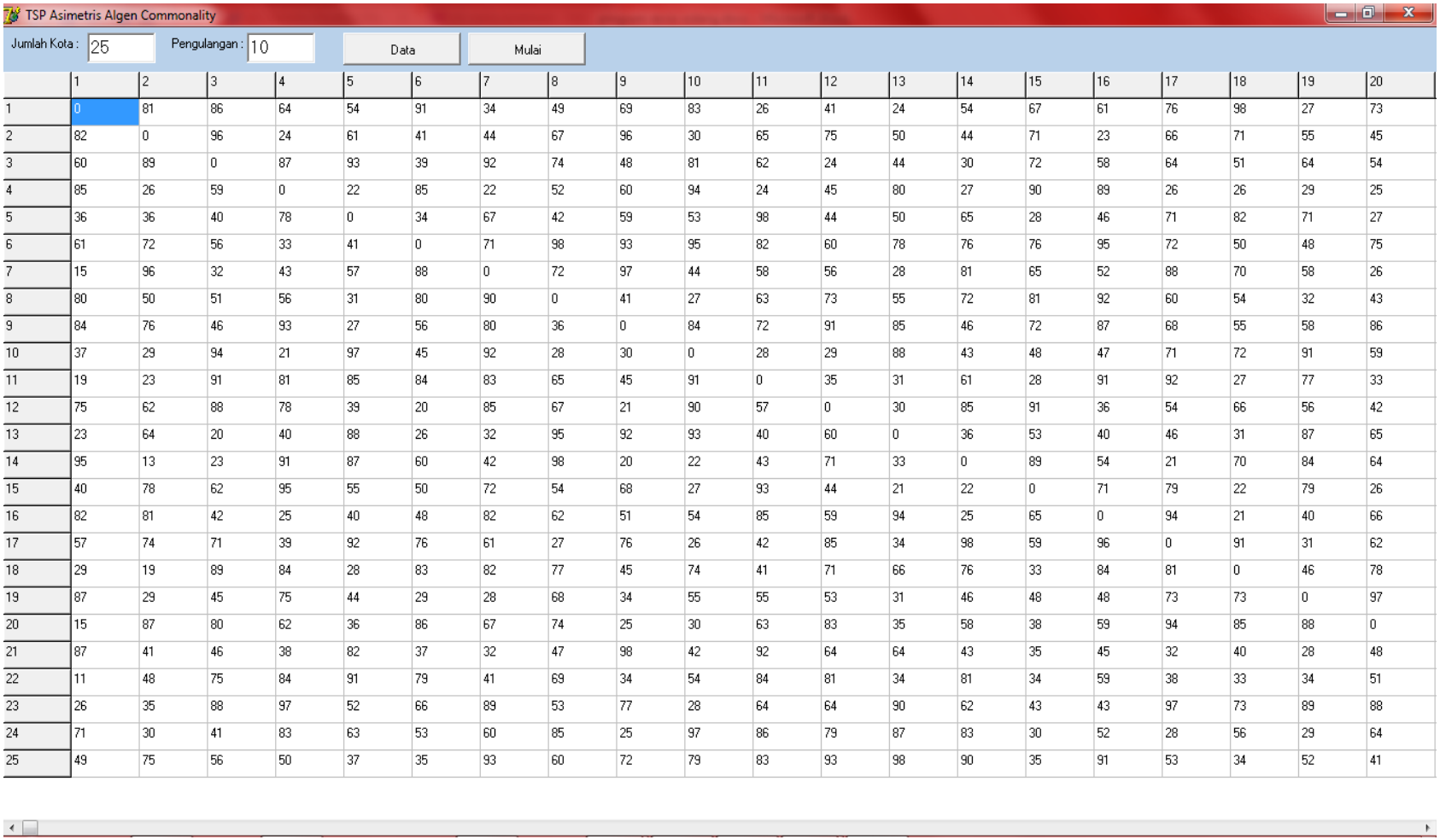
Dari tampilan program dapat dilihat bahwa antara perhitungan manual dan penyelesaikan menggunakan bantuan program dengan software *Borland Delphi* memiliki hasil yang sama.

Dari 10 kali pengulangan yang dilakukan pada program diperoleh jarak minimal yang ditempuh 217 km dengan rute 1-2-6-4-5-3.

4.1.3 Penyelesaian dengan Menggunakan Program untuk 25 Kota

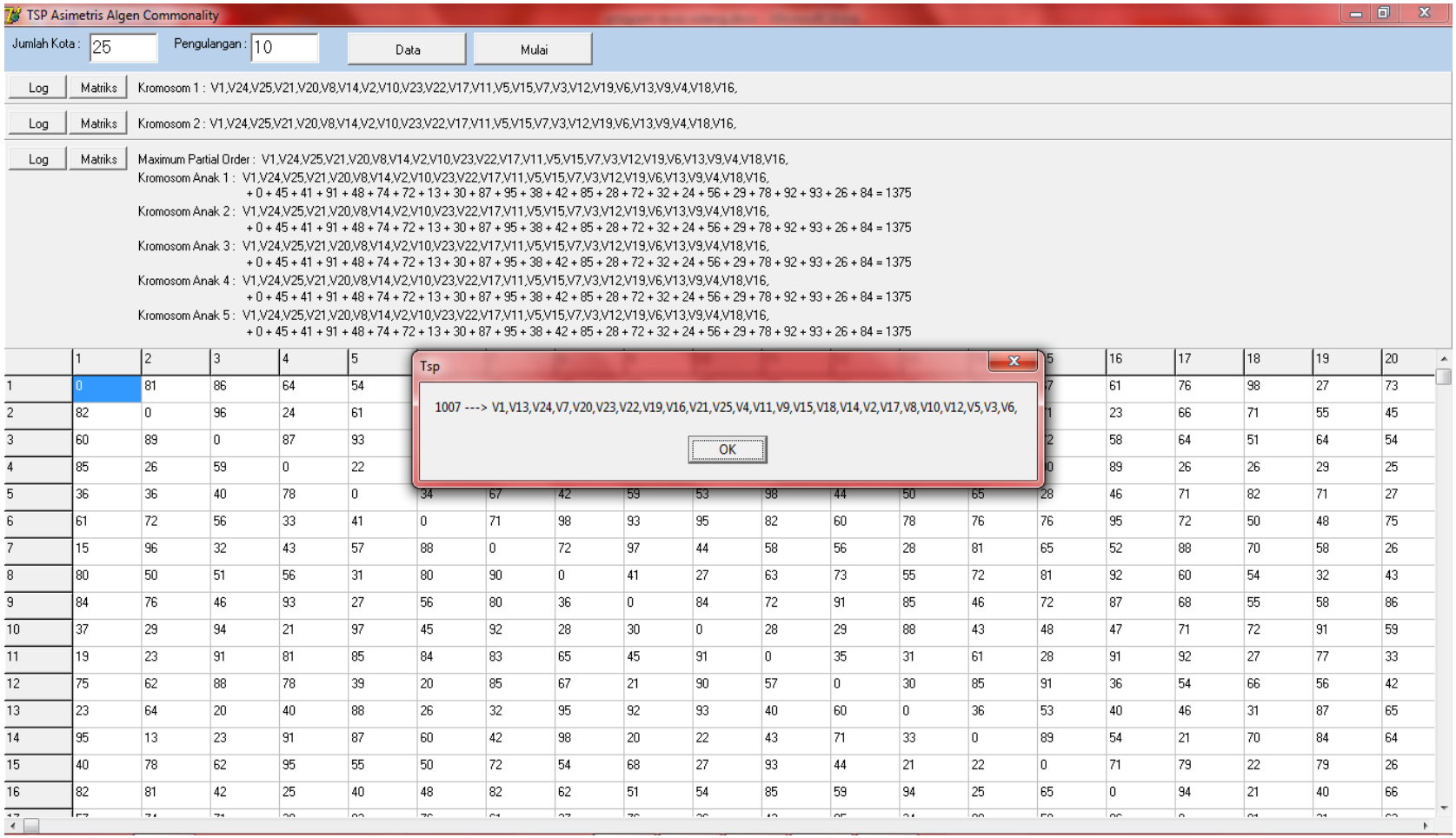
Untuk mempermudah perhitungan dalam memperoleh jarak minimal digunakan bantuan program *Borland Delphi*.

Data yang diinput dalam program tersebut adalah jarak antar kota yang dibangkitkan sendiri menggunakan Microsoft Excel. Output program adalah total jarak yang ditempuh dan rute perjalanannya.

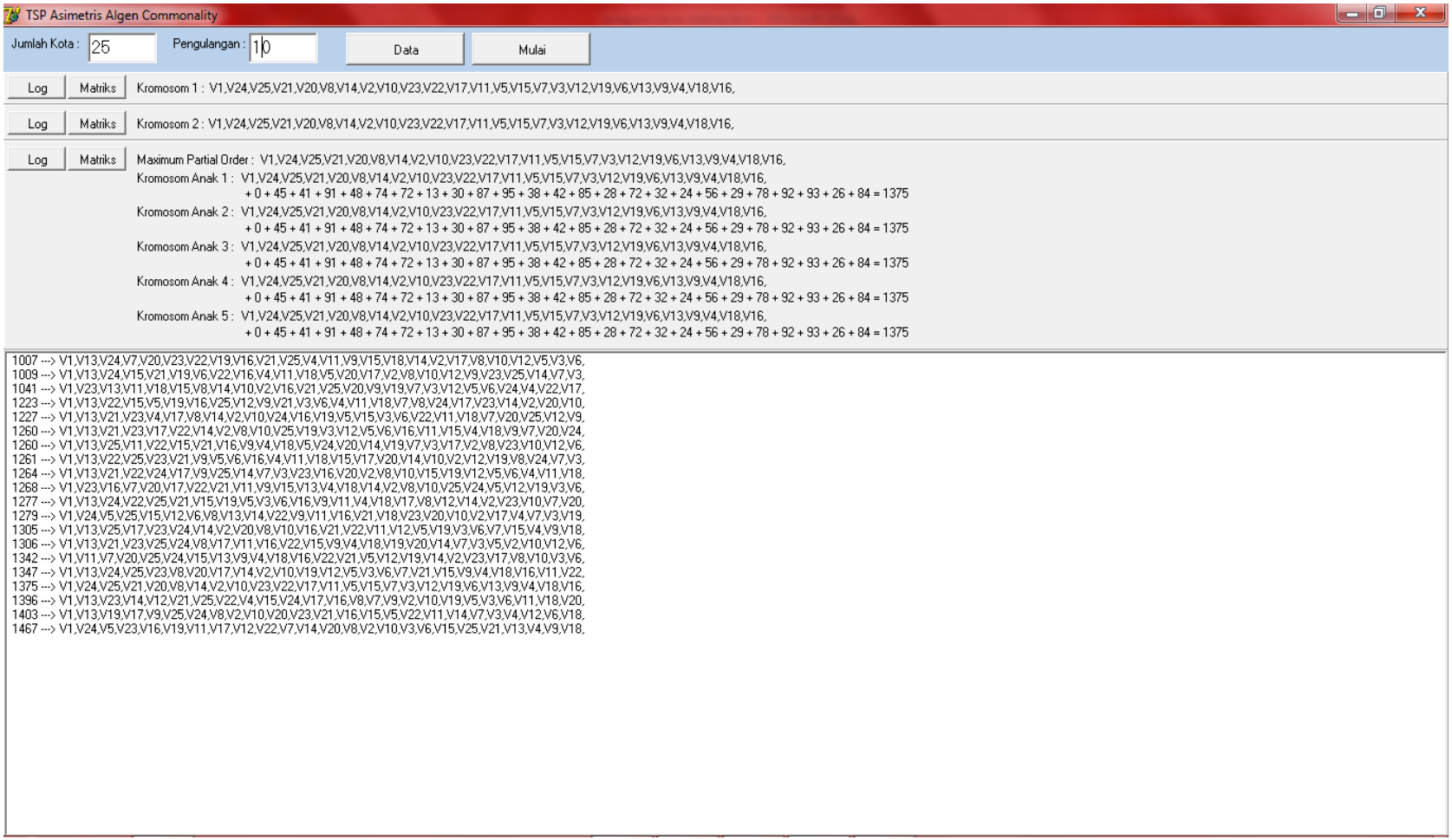


	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	81	86	64	54	91	34	49	69	83	26	41	24	54	67	61	76	98	27	73
2	82	0	96	24	61	41	44	67	96	30	65	75	50	44	71	23	66	71	55	45
3	60	89	0	87	93	39	92	74	48	81	62	24	44	30	72	58	64	51	64	54
4	85	26	59	0	22	85	22	52	60	94	24	45	80	27	90	89	26	26	29	25
5	36	36	40	78	0	34	67	42	59	53	98	44	50	65	28	46	71	82	71	27
6	61	72	56	33	41	0	71	98	93	95	82	60	78	76	76	95	72	50	48	75
7	15	96	32	43	57	88	0	72	97	44	58	56	28	81	65	52	88	70	58	26
8	80	50	51	56	31	80	90	0	41	27	63	73	55	72	81	92	60	54	32	43
9	84	76	46	93	27	56	80	36	0	84	72	91	85	46	72	87	68	55	58	86
10	37	29	94	21	97	45	92	28	30	0	28	29	88	43	48	47	71	72	91	59
11	19	23	91	81	85	84	83	65	45	91	0	35	31	61	28	91	92	27	77	33
12	75	62	88	78	39	20	85	67	21	90	57	0	30	85	91	36	54	66	56	42
13	23	64	20	40	88	26	32	95	92	93	40	60	0	36	53	40	46	31	87	65
14	95	13	23	91	87	60	42	98	20	22	43	71	33	0	89	54	21	70	84	64
15	40	78	62	95	55	50	72	54	68	27	93	44	21	22	0	71	79	22	79	26
16	82	81	42	25	40	48	82	62	51	54	85	59	94	25	65	0	94	21	40	66
17	57	74	71	39	92	76	61	27	76	26	42	85	34	98	59	96	0	91	31	62
18	29	19	89	84	28	83	82	77	45	74	41	71	66	76	33	84	81	0	46	78
19	87	29	45	75	44	29	28	68	34	55	55	53	31	46	48	48	73	73	0	97
20	15	87	80	62	36	86	67	74	25	30	63	83	35	58	38	59	94	85	88	0
21	87	41	46	38	82	37	32	47	98	42	92	64	64	43	35	45	32	40	28	48
22	11	48	75	84	91	79	41	69	34	54	84	81	34	81	34	59	38	33	34	51
23	26	35	88	97	52	66	89	53	77	28	64	64	90	62	43	43	97	73	89	88
24	71	30	41	83	63	53	60	85	25	97	86	79	87	83	30	52	28	56	29	64
25	49	75	56	50	37	35	93	60	72	79	83	93	98	90	35	91	53	34	52	41

Gambar 4.4 Tampilan output data untuk 25 kota



Gambar 4.5 Tampilan output rute dan jarak terpendek untuk 25 kota



Gambar 4.6 Tampilan output semua rute dan jarak untuk 25 kota

Tiap kali pengujian dilakukan dengan menggunakan pengulangan yang berbeda. Dari seluruh pengulangan yang dilakukan diperoleh nilai terkecil yang kemudian menjadi jarak terpendek pada pengujian tersebut.

Pada penelitian ini digunakan 10 kali pengujian yang hasilnya disajikan pada tabel 4.2 berikut ini.

Tabel 4.2 Hasil sepuluh kali pengujian 25 kota

Pengujian ke-	Pengulangan	Jarak (dalam km)	rute
1	10	1007	1-13-24-7-20-23-22-19-16-21-25-4-11-9-15-18-14-2-17-8-10-12-5-3-6
2	20	1007	1-13-24-7-20-23-22-19-16-21-25-4-11-9-15-18-14-2-17-8-10-12-5-3-6
3	30	1007	1-13-24-7-20-23-22-19-16-21-25-4-11-9-15-18-14-2-17-8-10-12-5-3-6
4	40	1007	1-13-24-7-20-23-22-19-16-21-25-4-11-9-15-18-14-2-17-8-10-12-5-3-6
5	50	1007	1-13-24-7-20-23-22-19-16-21-25-4-11-9-15-18-14-2-17-8-10-12-5-3-6
6	60	1007	1-13-24-7-20-23-22-19-16-21-25-4-11-9-15-18-14-2-17-8-10-12-5-3-6
7	70	1007	1-13-24-7-20-23-22-19-16-21-25-4-11-9-15-18-14-2-17-8-10-12-5-3-6
8	80	1007	1-13-24-7-20-23-22-19-16-21-25-4-11-9-15-18-14-2-17-8-10-12-5-3-6
9	90	1007	1-13-24-7-20-23-22-19-16-21-25-4-11-9-15-18-14-2-17-8-10-12-5-3-6
10	100	1007	1-13-24-7-20-23-22-19-16-21-25-4-11-9-15-18-14-2-17-8-10-12-5-3-6

Dari pengujian yang dilakukan diperoleh hasil jarak terpendek yang ditempuh adalah 1113 km dengan rute 1-13-24-7-20-23-22-19-16-21-25-4-11-9-15-18-14-2-17-8-10-12-5-3-6.

4.2 Pembahasan

Pada *output* program terdapat dua hasil yang berbeda. Hal ini dapat terjadi karena pada proses membangkitkan populasi (kumpulan rute) awal digunakan metode AI, dimana setelah menentukan c_{ik} minimal dan membentuk subtour $i - k - i$, langkah selanjutnya adalah memilih secara acak titik k yang tidak berada dalam *subtour* untuk masuk ke dalam *subtour*. Pemilihan acak tersebut sangat berpengaruh terhadap *subtour-subtour* selanjutnya yang akan terbentuk hingga menjadi sebuah *tour*. Pemilihan rute awal juga didasarkan pada pembentukan populasi (kumpulan rute) awal. Setelah populasi awal terbentuk dipilih dua rute (*tour*) yang bernilai minimal. Kemudian mengidentifikasi *common schema* dari kedua induk. Tahap pertama dapat dilakukan dengan operator MPO. Setelah terbentuk *Graph Partial Order, Common Schema* yang telah terbentuk akan disusun menjadi sebuah solusi baru (anak) dengan metode AI.

Dari pengujian yang dilakukan untuk 25 kota (data pada lampiran 1) menggunakan bantuan program *Borland Delphi* diperoleh bahwa dari 10 kali pengujian dengan pengulangan yang berbeda diperoleh jarak perjalanan terpendeknya sepanjang 1007 km dengan rute 1-13-24-7-20-23-22-19-16-21-25-4-11-9-15-18-14-2-17-8-10-12-5-3-6. Dari pengujian tersebut juga diperoleh bahwa dapat dihasilkan jarak yang sama dengan rute yang berbeda. Hal ini dapat dilihat pada gambar 4.6. Pada tampilan jarak dan rute terdapat dua jarak yang sama dengan rute yang berbeda. Hal ini disebabkan adanya proses penyisipan kota-kota yang tidak terdapat pada *Graf Partial Order* menggunakan metode AI. Perbedaan rute ini tergantung pada nilai k (kota) mana yang dipilih untuk disisipkan.

Sama seperti algoritma genetika, pada algoritma genetik *commonality* tingkat efisiensi algoritma yang digunakan tergantung pada jumlah kota. Semakin banyak jumlah kota yang digunakan maka semakin banyak pula kemungkinan solusinya.

BAB 5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari penelitian yang dilakukan, dapat disimpulkan beberapa hal berikut :

1. Diperoleh bahwa jarak perjalanan terpendek untuk 25 kota sepanjang 1007 km dengan rute 1-13-24-7-20-23-22-19-16-21-25-4-11-9-15-18-14-2-17-8-10-12-5-3-6.
2. Dapat dihasilkan jarak yang sama dengan rute yang berbeda karena adanya proses penyisipan kota-kota yang tidak terdapat pada *Graf Partial Order* menggunakan metode AI.

5.2 Saran

Dalam skripsi ini penulis menerapkan algoritma genetik *commonality* pada kasus TSP Asimetris dengan inisialisasi awal menggunakan algoritma AI yang kurang efisien jika digunakan untuk kota dengan jumlah besar. Masih ada beberapa algoritma yang dapat digunakan untuk menentukan inisialisasi awal, seperti metode tetangga terdekat (*Nearest Neighbor Method*/NN), metode *Convex Hull/Arbitrary Insertion* (CH/AI), gabungan AI+NN, dan gabungan CH/AI+NN, sehingga masih terdapat kesempatan bagi penulis lain untuk mengadakan penelitian agar dapat diperoleh hasil yang lebih maksimal.

DAFTAR PUSTAKA

Ayuningtyas, D. S. 2008. "Aplikasi Algoritma Genetika pada Masalah Penugasan". Tidak Diterbitkan. Skripsi. Jember: Jurusan Matematika Fakultas MIPA Universitas Jember.

Chartrand, G. & Oellermann, O. R. 1993. *Applied and Algorithmic Graph Theory*. New York : Mc Graw Hill, Inc.

Chen, S. & S. F. Smith. 1999. *Introducing a New Advantage of Crossover: Commonality-Based Selection*. In GECCO 99: *Proceedings of the Genetic and Evolutionary Computation Conference.*

Dewi, E. P. 2009. "Optimasi Rute *Multiple-Travelling Salesman Problem* Melalui Pemrograman *Integer* dengan Metode *Branch and Bound*". Tidak Diterbitkan. Skripsi. Jember: Jurusan Matematika Fakultas MIPA Universitas Jember.

Holland, J. H. 1975. *Adaptation in Natural and Artificial Systems: an introductory analysis with applications to biology, control, and artificial intelligence*. University of Michigan Press.

Ilyas, M. 2007. "Perbandingan Penggunaan Algoritma *Greedy* dan Algoritma Genetika dalam Penyelesaian Masalah Perjalanan Salesman". Tidak Diterbitkan. Skripsi. Jember: Jurusan Matematika Fakultas MIPA Universitas Jember.

Kusumadewi, S. 2003. *Artificial Intelligence* (*Teknik dan Aplikasinya*). Yogyakarta. Graha Ilmu.

------- & Purnomo, H. 2005. *Penyelesaian Masalah Optimasi dengan Teknik-Teknik Heuristik*. Yogyakarta. Graha Ilmu.

Liu, C. L. 1995. *Dasar-Dasar Matematika Diskret* (Edisi Kedua). Jakarta. PT Gramedia Pustaka Utama.

Ramadhani, M. 2008. “Penyelesaian Masalah Optimasi Menggunakan Algoritma Genetika”. Tidak Diterbitkan. Skripsi. Jember: Jurusan Matematika Fakultas MIPA Universitas Jember.

Setiyawan, T. E. 2005. “Perbandingan Penggunaan Algoritma Semut dan Algoritma *Greedy* dalam Menyelesaikan Masalah Perjalanan Salesman”. Tidak Diterbitkan. Skripsi. Jember: Jurusan Matematika Fakultas MIPA Universitas Jember.

Suyanto. 2005. *Algoritma Genetika Dalam Matlab*. Yogyakarta. Andi.

LAMPIRAN

Lampiran 1 Data jarak 25 kota

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	-	81	86	64	54	91	34	49	69	83	26	41	24	54	67	61	76	98	27	73	43	46	61	45	47
2	82	-	96	24	61	41	44	67	96	30	65	75	50	44	71	23	66	71	55	45	71	87	71	74	27
3	60	89	-	87	93	39	92	74	48	81	62	24	44	30	72	58	64	51	64	54	48	57	33	43	59
4	85	26	59	-	22	85	22	52	60	94	24	45	80	27	90	89	26	26	29	25	44	96	32	33	34
5	36	36	40	78	-	34	67	42	59	53	98	44	50	65	28	46	71	82	71	27	36	59	67	79	96
6	61	72	56	33	41	-	71	98	93	95	82	60	78	76	76	95	72	50	48	75	21	28	54	24	30
7	15	96	32	43	57	88	-	72	97	44	58	56	28	81	65	52	88	70	58	26	26	75	90	64	64
8	80	50	51	56	31	80	90	-	41	27	63	73	55	72	81	92	60	54	32	43	94	72	88	74	37
9	84	76	46	93	27	56	80	36	-	84	72	91	85	46	72	87	68	55	58	86	89	57	49	39	84
10	37	29	94	21	97	45	92	28	30	-	28	29	88	43	48	47	71	72	91	59	26	65	87	98	64
11	19	23	91	81	85	84	83	65	45	91	-	35	31	61	28	91	92	27	77	33	25	88	73	84	55
12	75	62	88	78	39	20	85	67	21	90	57	-	30	85	91	36	54	66	56	42	27	38	89	96	89
13	23	64	20	40	88	28	32	95	92	93	40	60	-	36	53	40	46	31	87	65	77	51	47	38	48
14	95	13	23	91	87	60	42	98	20	22	43	71	33	-	89	54	21	70	84	64	34	89	26	42	38
15	40	78	62	95	55	50	72	54	68	27	93	44	21	22	-	71	79	22	79	26	25	48	76	27	80
16	82	81	42	25	40	48	82	62	51	54	85	59	94	25	65	-	94	21	40	66	25	33	85	59	81
17	57	74	71	39	92	76	61	27	76	26	42	85	34	98	59	96	-	91	31	62	34	36	81	76	49
18	49	19	89	84	28	83	82	77	45	74	41	71	66	76	33	84	81	-	46	78	50	70	66	85	48
19	87	29	45	75	44	29	28	68	34	55	55	53	31	46	48	48	73	73	-	97	29	59	85	97	21
20	15	87	80	62	36	86	67	74	25	30	63	83	35	58	38	59	94	85	88	-	44	68	45	59	96
21	87	41	46	38	82	37	32	47	98	42	92	64	64	43	35	45	32	40	28	48	-	53	41	89	43
22	11	48	75	84	91	79	41	69	34	54	84	81	34	81	34	59	38	33	34	51	79	-	32	98	45
23	26	35	88	97	52	66	89	53	77	28	64	64	90	62	43	43	97	73	89	88	88	95	-	73	92
24	71	30	41	83	63	53	60	85	25	97	86	79	87	83	30	52	28	56	29	64	29	21	69	-	41
25	49	75	56	50	37	35	93	60	72	79	83	93	98	90	35	91	53	34	52	41	91	41	75	57	-

Lampiran 2 Kromosom-kromosom yang dihasilkan untuk 6 kota

Kromosom 1 = $[v_1 v_2 v_3 v_4 v_5 v_6]$
Kromosom 2 = $[v_1 v_2 v_3 v_4 v_6 v_5]$
Kromosom 3 = $[v_1 v_2 v_3 v_5 v_4 v_6]$
Kromosom 4 = $[v_1 v_2 v_3 v_5 v_6 v_4]$
Kromosom 5 = $[v_1 v_2 v_3 v_6 v_4 v_5]$
Kromosom 6 = $[v_1 v_2 v_3 v_6 v_5 v_4]$
Kromosom 7 = $[v_1 v_2 v_4 v_3 v_5 v_6]$
Kromosom 8 = $[v_1 v_2 v_4 v_3 v_6 v_5]$
Kromosom 9 = $[v_1 v_2 v_4 v_5 v_3 v_6]$
Kromosom 10 = $[v_1 v_2 v_4 v_5 v_6 v_3]$
Kromosom 11 = $[v_1 v_2 v_4 v_6 v_3 v_5]$
Kromosom 12 = $[v_1 v_2 v_4 v_6 v_5 v_3]$
Kromosom 13 = $[v_1 v_2 v_5 v_3 v_4 v_6]$
Kromosom 14 = $[v_1 v_2 v_5 v_3 v_6 v_4]$
Kromosom 15 = $[v_1 v_2 v_5 v_4 v_3 v_6]$
Kromosom 16 = $[v_1 v_2 v_5 v_4 v_6 v_3]$
Kromosom 17 = $[v_1 v_2 v_5 v_6 v_3 v_4]$
Kromosom 18 = $[v_1 v_2 v_5 v_6 v_4 v_3]$
Kromosom 19 = $[v_1 v_2 v_6 v_3 v_4 v_5]$
Kromosom 20 = $[v_1 v_2 v_6 v_3 v_5 v_4]$
Kromosom 21 = $[v_1 v_2 v_6 v_4 v_3 v_5]$
Kromosom 22 = $[v_1 v_2 v_6 v_4 v_5 v_3]$
Kromosom 23 = $[v_1 v_2 v_6 v_5 v_3 v_4]$
Kromosom 24 = $[v_1 v_2 v_6 v_5 v_4 v_3]$
Kromosom 25 = $[v_1 v_3 v_2 v_4 v_5 v_6]$
Kromosom 26 = $[v_1 v_3 v_2 v_4 v_6 v_5]$
Kromosom 27 = $[v_1 v_3 v_2 v_5 v_4 v_6]$
Kromosom 28 = $[v_1 v_3 v_2 v_5 v_6 v_4]$
Kromosom 29 = $[v_1 v_3 v_2 v_6 v_4 v_5]$
Kromosom 30 = $[v_1 v_3 v_2 v_6 v_5 v_4]$
Kromosom 31 = $[v_1 v_3 v_4 v_2 v_5 v_6]$
Kromosom 32 = $[v_1 v_3 v_4 v_2 v_6 v_5]$
Kromosom 33 = $[v_1 v_3 v_4 v_5 v_2 v_6]$
Kromosom 34 = $[v_1 v_3 v_4 v_5 v_6 v_2]$
Kromosom 35 = $[v_1 v_3 v_4 v_6 v_2 v_5]$
Kromosom 36 = $[v_1 v_3 v_4 v_6 v_5 v_2]$
Kromosom 37 = $[v_1 v_3 v_5 v_2 v_4 v_6]$
Kromosom 38 = $[v_1 v_3 v_5 v_2 v_6 v_4]$
Kromosom 39 = $[v_1 v_3 v_5 v_4 v_2 v_6]$
Kromosom 40 = $[v_1 v_3 v_5 v_4 v_6 v_2]$
Kromosom 41 = $[v_1 v_3 v_5 v_6 v_2 v_4]$
Kromosom 42 = $[v_1 v_3 v_5 v_6 v_4 v_2]$
Kromosom 43 = $[v_1 v_3 v_6 v_2 v_4 v_5]$
Kromosom 44 = $[v_1 v_3 v_6 v_2 v_5 v_4]$
Kromosom 45 = $[v_1 v_3 v_6 v_4 v_2 v_5]$
Kromosom 46 = $[v_1 v_3 v_6 v_4 v_5 v_2]$
Kromosom 47 = $[v_1 v_3 v_6 v_5 v_2 v_4]$
Kromosom 48 = $[v_1 v_3 v_6 v_5 v_4 v_2]$
Kromosom 49 = $[v_1 v_4 v_2 v_3 v_5 v_6]$
Kromosom 50 = $[v_1 v_4 v_2 v_3 v_6 v_5]$
Kromosom 51 = $[v_1 v_4 v_2 v_5 v_3 v_6]$

Kromosom 52 = $[v_1 v_4 v_2 v_5 v_6 v_3]$
Kromosom 53 = $[v_1 v_4 v_2 v_6 v_3 v_5]$
Kromosom 54 = $[v_1 v_4 v_2 v_6 v_5 v_3]$
Kromosom 55 = $[v_1 v_4 v_3 v_2 v_5 v_6]$
Kromosom 56 = $[v_1 v_4 v_3 v_2 v_6 v_5]$
Kromosom 57 = $[v_1 v_4 v_3 v_5 v_2 v_6]$
Kromosom 58 = $[v_1 v_4 v_3 v_5 v_6 v_2]$
Kromosom 59 = $[v_1 v_4 v_3 v_6 v_2 v_5]$
Kromosom 60 = $[v_1 v_4 v_3 v_6 v_5 v_2]$
Kromosom 61 = $[v_1 v_4 v_5 v_2 v_3 v_6]$
Kromosom 62 = $[v_1 v_4 v_5 v_2 v_6 v_3]$
Kromosom 63 = $[v_1 v_4 v_5 v_3 v_2 v_6]$
Kromosom 64 = $[v_1 v_4 v_5 v_3 v_6 v_2]$
Kromosom 65 = $[v_1 v_4 v_5 v_6 v_2 v_3]$
Kromosom 66 = $[v_1 v_4 v_5 v_6 v_3 v_2]$
Kromosom 67 = $[v_1 v_4 v_6 v_2 v_3 v_5]$
Kromosom 68 = $[v_1 v_4 v_6 v_2 v_5 v_3]$
Kromosom 69 = $[v_1 v_4 v_6 v_3 v_2 v_5]$
Kromosom 70 = $[v_1 v_4 v_6 v_3 v_5 v_2]$
Kromosom 71 = $[v_1 v_4 v_6 v_5 v_2 v_3]$
Kromosom 72 = $[v_1 v_4 v_6 v_5 v_3 v_2]$
Kromosom 73 = $[v_1 v_5 v_2 v_3 v_4 v_6]$
Kromosom 74 = $[v_1 v_5 v_2 v_3 v_6 v_4]$
Kromosom 75 = $[v_1 v_5 v_2 v_4 v_3 v_6]$
Kromosom 76 = $[v_1 v_5 v_2 v_4 v_6 v_3]$
Kromosom 77 = $[v_1 v_5 v_2 v_6 v_3 v_4]$
Kromosom 78 = $[v_1 v_5 v_2 v_6 v_4 v_3]$
Kromosom 79 = $[v_1 v_5 v_3 v_2 v_4 v_6]$
Kromosom 80 = $[v_1 v_5 v_3 v_2 v_6 v_4]$
Kromosom 81 = $[v_1 v_5 v_3 v_4 v_2 v_6]$
Kromosom 82 = $[v_1 v_5 v_3 v_4 v_6 v_2]$
Kromosom 83 = $[v_1 v_5 v_3 v_6 v_2 v_4]$
Kromosom 84 = $[v_1 v_5 v_3 v_6 v_4 v_2]$
Kromosom 85 = $[v_1 v_5 v_4 v_2 v_3 v_6]$
Kromosom 86 = $[v_1 v_5 v_4 v_2 v_6 v_3]$
Kromosom 87 = $[v_1 v_5 v_4 v_3 v_2 v_6]$
Kromosom 88 = $[v_1 v_5 v_4 v_3 v_6 v_2]$
Kromosom 89 = $[v_1 v_5 v_4 v_6 v_2 v_3]$
Kromosom 90 = $[v_1 v_5 v_4 v_6 v_3 v_2]$
Kromosom 91 = $[v_1 v_5 v_6 v_2 v_3 v_4]$
Kromosom 92 = $[v_1 v_5 v_6 v_2 v_4 v_3]$
Kromosom 93 = $[v_1 v_5 v_6 v_3 v_2 v_4]$
Kromosom 94 = $[v_1 v_5 v_6 v_3 v_4 v_2]$
Kromosom 95 = $[v_1 v_5 v_6 v_4 v_2 v_3]$
Kromosom 96 = $[v_1 v_5 v_6 v_4 v_3 v_2]$
Kromosom 97 = $[v_1 v_6 v_2 v_3 v_4 v_5]$
Kromosom 98 = $[v_1 v_6 v_2 v_3 v_5 v_4]$
Kromosom 99 = $[v_1 v_6 v_2 v_4 v_3 v_5]$
Kromosom 100 = $[v_1 v_6 v_2 v_4 v_5 v_3]$
Kromosom 101 = $[v_1 v_6 v_2 v_5 v_3 v_4]$
Kromosom 102 = $[v_1 v_6 v_2 v_5 v_4 v_3]$
Kromosom 103 = $[v_1 v_6 v_3 v_2 v_5 v_4]$
Kromosom 104 = $[v_1 v_6 v_3 v_2 v_4 v_5]$
Kromosom 105 = $[v_1 v_6 v_3 v_4 v_2 v_5]$

Kromosom 106 = $[v_1 v_6 v_3 v_4 v_5 v_2]$
Kromosom 107 = $[v_1 v_6 v_3 v_5 v_2 v_4]$
Kromosom 108 = $[v_1 v_6 v_3 v_5 v_4 v_2]$
Kromosom 109 = $[v_1 v_6 v_4 v_2 v_3 v_5]$
Kromosom 110 = $[v_1 v_6 v_4 v_2 v_5 v_3]$
Kromosom 111 = $[v_1 v_6 v_4 v_3 v_2 v_5]$
Kromosom 112 = $[v_1 v_6 v_4 v_3 v_5 v_2]$
Kromosom 113 = $[v_1 v_6 v_4 v_5 v_2 v_3]$
Kromosom 114 = $[v_1 v_6 v_4 v_5 v_3 v_2]$
Kromosom 115 = $[v_1 v_6 v_5 v_2 v_3 v_4]$
Kromosom 116 = $[v_1 v_6 v_5 v_2 v_4 v_3]$
Kromosom 117 = $[v_1 v_6 v_5 v_3 v_2 v_4]$
Kromosom 118 = $[v_1 v_6 v_5 v_3 v_4 v_2]$
Kromosom 119 = $[v_1 v_6 v_5 v_4 v_2 v_3]$
Kromosom 120 = $[v_1 v_6 v_5 v_4 v_3 v_2]$

Lampiran 3 Program untuk Penyelesaian 25 Kota

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, Buttons, Grids;

type
  TForm1 = class(TForm)
    Panel3: TPanel;
    MyHasil1: TComboBox;
    LogInduk1: TMemo;
    PnlInterseksi: TPanel;
    PnlInduk2: TPanel;
    PnlInduk1: TPanel;
    DataKota: TStringGrid;
    MyHasil2: TComboBox;
    LKromosom1: TLabel;
    Kromosom1: TLabel;
    SpeedButton6: TSpeedButton;
    SpeedButton7: TSpeedButton;
    Matriks1: TStringGrid;
    MyIndex: TComboBox;
    Matriks2: TStringGrid;
    LogInduk2: TMemo;
    SpeedButton8: TSpeedButton;
    SpeedButton9: TSpeedButton;
    LKromosom2: TLabel;
    Kromosom2: TLabel;
    Matriks3: TStringGrid;
    SpeedButton10: TSpeedButton;
    Label4: TLabel;
    MPO: TLabel;
    MyIndexSort: TComboBox;
    SpeedButton3: TSpeedButton;
    LogAnak: TMemo;
    MyHasil3: TComboBox;
    Label5: TLabel;
    KromosomAnak1: TLabel;
    Label6: TLabel;
    KromosomAnak2: TLabel;
    HasilAnak1: TLabel;
    HasilAnak2: TLabel;
    Label10: TLabel;
    KromosomAnak3: TLabel;
```

```
    HasilAnak3: TLabel;
    HasilAnak5: TLabel;
    KromosomAnak5: TLabel;
    Label15: TLabel;
    Label16: TLabel;
    KromosomAnak4: TLabel;
    HasilAnak4: TLabel;
    Panel1: TPanel;
    Interseksi: TSpeedButton;
    SpeedButton1: TSpeedButton;
    JmlUlang: TEdit;
    Memo1: TMemo;
    Memo2: TMemo;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label7: TLabel;
    JmlIterasi: TEdit;
    Panel2: TPanel;
    ListBox1: TListBox;
    procedure SpeedButton6Click(Sender: TObject);
    procedure SpeedButton7Click(Sender: TObject);
    //procedure FormCreate(Sender: TObject);
    procedure SpeedButton8Click(Sender: TObject);
    procedure SpeedButton9Click(Sender: TObject);
    procedure InterseksiClick(Sender: TObject);
    procedure SpeedButton10Click(Sender: TObject);
    procedure SpeedButton3Click(Sender: TObject);
    procedure SpeedButton1Click(Sender: TObject);
    procedure SpeedButton2Click(Sender: TObject);
  private
    Procedure AturUlangMyIndex( Nilai : Integer );
    function CariTitik : Integer;
    function CariPosisi( MyData : TComboBox; Cari:String) : Integer;
    function Perhitungan( Proses: String; MyLog: TMemo; MyHasil: TComboBox; MyMatriks: TStringGrid )
: String;
  public
    { Public declarations }
  end;

//const
  //JumlahKota : Integer  = 25 ;
var
  Form1: TForm1;
  JumlahKota: Integer;

implementation

Procedure TForm1.AturUlangMyIndex( Nilai : Integer );
var
  x : integer;
begin
   x := -1;
   repeat
      x := x + 1;
      MyIndex.ItemIndex := x;
   until MyIndex.Text = IntToStr(Nilai);
   MyIndex.DeleteSelected;
end;

function TForm1.CariTitik : Integer;
begin
  randomize;   // bangkitkan nilai Acak

  // random dimulai dari 0 shg jml item di kurangi 1
  // untuk menghindari angka 0 maka ditambah 1
  if MyIndex.Items.Count>1 then
     MyIndex.ItemIndex := Random( MyIndex.Items.Count-1 ) + 1  // ambil nilai Acak
  else
     MyIndex.ItemIndex := 0;
```

```
  CariTitik := StrToInt(MyIndex.Text);
end;

function TForm1.CariPosisi(MyData : TComboBox; Cari:String) : Integer;
var
  x, s : integer;
begin
  s := 0;
  for x:=0 to JumlahKota-1 do
  begin
    MyData.ItemIndex := x;
    if MyData.Text=Cari then
       s := x;
  end;

  CariPosisi := s;
end;

function TForm1.Perhitungan( Proses: String; MyLog: TMemo; MyHasil: TComboBox; MyMatriks:
TStringGrid ) : String;
var
  Data : array[1..30,1..30] of integer;
  Hasil : array[1..30] of integer;
  Nilai : array[1..30] of integer;
  i, k, j, t : integer;
  x, z, p, n : integer;
  c, r : integer;
  s : Boolean;
  h : String;
begin
  MyLog.Clear;
  MyLog.Lines.Add( 'PEMBUATAN '+Proses );
  MyLog.Lines.Add( '------------------' );

  // -------------- buat index -----------------------------
  MyIndex.Items.Clear;
  for x:= 1 to JumlahKota do
     MyIndex.Items.Add( IntToStr(x) );

  //--------------- Data[ Baris, Colom ] -------------------
  for x:=1 to JumlahKota do
    for z:=1 to JumlahKota do
        Data[x,z] := StrToInt( DataKota.Cells[z,x]);

  //---- 1. awali dengan sebuah titik i -----------------------
  i := 1;
  AturUlangMyIndex( i );
  Hasil[1] := i;
  p := 1;   // ---> proses 1

  MyLog.Lines.Add( '' );
  MyLog.Lines.Add( 'i = '+inttostr(i) );

  //---- 2. tentukan titik k dengan cari nilai yang paling kecil ----------
  k := 2;
  for x:=3 to JumlahKota do
     if Data[i,k] > Data[i,x] then k := x;
  AturUlangMyIndex( k );
  Hasil[2] := k;
  inc(p);   // ---> proses 2

  MyLog.Lines.Add( '' );
  MyLog.Lines.Add( 'k = '+inttostr(k) );

  //---- 3. memilih secara acak titik k ----------
  j := CariTitik;
  MyLog.Lines.Add( 'j = '+inttostr(j) );

  AturUlangMyIndex( j );
```

```
  Nilai[1] := Data[i,j] + Data[j,k] - Data[i,k];
  MyLog.Lines.Add( inttostr(Data[i,j])+' + '+inttostr(Data[j,k])+' - '+inttostr(Data[i,k])+' =
'+inttostr(Nilai[1]) );

  Nilai[2] := Data[i,k] + Data[k,j] - Data[i,j];
  MyLog.Lines.Add( inttostr(Data[i,k])+' + '+inttostr(Data[k,j])+' - '+inttostr(Data[i,j])+' =
'+inttostr(Nilai[2]) );

  if Nilai[1]<Nilai[2] then
  begin
     Hasil[2] := j;
     Hasil[3] := k;
  end
  else
     Hasil[3] := j;

  MyLog.Lines.Add( 'Hasil ->V'+inttostr(Hasil[1])+', V'+inttostr(Hasil[2])+', V'+inttostr(Hasil[3])
);

  inc(p);   // ---> proses 3
  MyLog.Lines.Add( '' );

  // lakukan perulangan untuk proses selanjutnya
  for z := 4 to JumlahKota do
  begin
        k := CariTitik;
        MyLog.Lines.Add( 'k = '+inttostr(k) );
        AturUlangMyIndex( k );

        // cari nilai
        for x:=2 to p do
        begin
           j := Hasil[x];
           Nilai[x] := Data[i,k] + Data[k,j] - Data[i,j];

           MyLog.Lines.Add( inttostr(Data[i,k])+' + '+inttostr(Data[k,j])+' -
'+inttostr(Data[i,j])+' = '+inttostr(Nilai[x]) );
        end;

        // Cari index dengan Nilai Paling Kecil
        n := 2;
        for x:=3 to p do
           if Nilai[n]>Nilai[x] then n := x;
        t := Hasil[n];

        h := '';
        for x:=1 to p do
           h := h + 'V'+inttostr( Hasil[x] )+ ', ';
        MyLog.Lines.Add( 'Hasil ->'+h);
        inc(p);   // ---> proses ke X

        MyLog.Lines.Add( '' );

        // Sisipkan nilai n ke Hasil
        s := false;
        for x:=1 to p do
        begin
           if Hasil[x]=t then
              s := true;

           if s then
           begin
              t := Hasil[x];  // tampung sementara
              Hasil[x] := k;  // isi hasil dengan k
              k := t;         // var k isi dengan t
           end;
        end;
  end; // of for z
```

```
  // keluarkan hasilnya
  MyHasil.Items.Clear;
  h := '';
  for x:=1 to JumlahKota do
  begin
     h := h + 'V'+inttostr( Hasil[x] )+ ',';
     MyHasil.Items.Add( inttostr( Hasil[x] ) );
  end;

  // buat matrik
  with MyMatriks do
  begin
      // buat Judul
      for c:=1 to JumlahKota do
          Cells[c,0] := inttostr(c);

      for r:=1 to JumlahKota do
          Cells[0,r] := inttostr(r);

      for r:=1 to JumlahKota do
        for c:=1 to JumlahKota do
            if CariPosisi( MyHasil, IntToStr(c) ) > CariPosisi( MyHasil, IntToStr(r)) then
               Cells[c,r] := '1'
            else
               Cells[c,r] := '0';
  end;

  Perhitungan := h;
end;


procedure TForm1.SpeedButton1Click(Sender: TObject);
var
  c, r : integer;
begin
  Memo2.Clear;
  Listbox1.Clear;
  Interseksi.Visible := true;
  JumlahKota := strtoint(JmlUlang.Text);
  DataKota.ColCount := JumlahKota+1;
  DataKota.RowCount := JumlahKota+1;
  Matriks1.ColCount := JumlahKota+1;
  Matriks1.RowCount := JumlahKota+1;
  Matriks2.ColCount := JumlahKota+1;
  Matriks2.RowCount := JumlahKota+1;
  Matriks3.ColCount := JumlahKota+1;
  Matriks3.RowCount := JumlahKota+1;

  LogInduk1.Align := alClient;
  LogInduk2.Align := alClient;
  LogAnak.Align   := alClient;
  Matriks1.Align  := alClient;
  Matriks2.Align  := alClient;
  Matriks3.Align  := alClient;
  ListBox1.Align := alClient;
  Panel2.Align := alClient;
  Panel2.Visible := False;

  PnlInduk1.Visible := False;
  with DataKota do
  begin
      // atur button/tombol
      if Visible then
         Visible := False
      else
      begin
         Visible := True;
         Align := alClient;
      end;
```

```
Interseksi.Enabled := Visible;

// buat Judul
for c:=1 to JumlahKota do
    Cells[c,0] := inttostr(c);

for r:=1 to JumlahKota do
    Cells[0,r] := inttostr(r);

// ambil data
Cells[1,1] := '0';
Cells[2,1] := '81';
Cells[3,1] := '86';
Cells[4,1] := '64';
Cells[5,1] := '54';
Cells[6,1] := '91';
Cells[7,1] := '34';
Cells[8,1] := '49';
Cells[9,1] := '69';
Cells[10,1] := '83';
Cells[11,1] := '26';
Cells[12,1] := '41';
Cells[13,1] := '24';
Cells[14,1] := '54';
Cells[15,1] := '67';
Cells[16,1] := '61';
Cells[17,1] := '76';
Cells[18,1] := '98';
Cells[19,1] := '27';
Cells[20,1] := '73';
Cells[21,1] := '43';
Cells[22,1] := '46';
Cells[23,1] := '61';
Cells[24,1] := '45';
Cells[25,1] := '47';


Cells[1,2] := '82';
Cells[2,2] := '0';
Cells[3,2] := '96';
Cells[4,2] := '24';
Cells[5,2] := '61';
Cells[6,2] := '41';
Cells[7,2] := '44';
Cells[8,2] := '67';
Cells[9,2] := '96';
Cells[10,2] := '30';
Cells[11,2] := '65';
Cells[12,2] := '75';
Cells[13,2] := '50';
Cells[14,2] := '44';
Cells[15,2] := '71';
Cells[16,2] := '23';
Cells[17,2] := '66';
Cells[18,2] := '71';
Cells[19,2] := '55';
Cells[20,2] := '45';
Cells[21,2] := '71';
Cells[22,2] := '87';
Cells[23,2] := '71';
Cells[24,2] := '74';
Cells[25,2] := '27';

Cells[1,3] := '60';
Cells[2,3] := '89';
Cells[3,3] := '0';
Cells[4,3] := '87';
Cells[5,3] := '93';
Cells[6,3] := '39';
Cells[7,3] := '92';
```

```
Cells[8,3] := '74';
Cells[9,3] := '48';
Cells[10,3] := '81';
Cells[11,3] := '62';
Cells[12,3] := '24';
Cells[13,3] := '44';
Cells[14,3] := '30';
Cells[15,3] := '72';
Cells[16,3] := '58';
Cells[17,3] := '64';
Cells[18,3] := '51';
Cells[19,3] := '64';
Cells[20,3] := '54';
Cells[21,3] := '48';
Cells[22,3] := '57';
Cells[23,3] := '33';
Cells[24,3] := '43';
Cells[25,3] := '59';

Cells[1,4] := '85';
Cells[2,4] := '26';
Cells[3,4] := '59';
Cells[4,4] := '0';
Cells[5,4] := '22';
Cells[6,4] := '85';
Cells[7,4] := '22';
Cells[8,4] := '52';
Cells[9,4] := '60';
Cells[10,4] := '94';
Cells[11,4] := '24';
Cells[12,4] := '45';
Cells[13,4] := '80';
Cells[14,4] := '27';
Cells[15,4] := '90';
Cells[16,4] := '89';
Cells[17,4] := '26';
Cells[18,4] := '26';
Cells[19,4] := '29';
Cells[20,4] := '25';
Cells[21,4] := '44';
Cells[22,4] := '96';
Cells[23,4] := '32';
Cells[24,4] := '33';
Cells[25,4] := '34';

Cells[1,5] := '36';
Cells[2,5] := '36';
Cells[3,5] := '40';
Cells[4,5] := '78';
Cells[5,5] := '0';
Cells[6,5] := '34';
Cells[7,5] := '67';
Cells[8,5] := '42';
Cells[9,5] := '59';
Cells[10,5] := '53';
Cells[11,5] := '98';
Cells[12,5] := '44';
Cells[13,5] := '50';
Cells[14,5] := '65';
Cells[15,5] := '28';
Cells[16,5] := '46';
Cells[17,5] := '71';
Cells[18,5] := '82';
Cells[19,5] := '71';
Cells[20,5] := '27';
Cells[21,5] := '36';
Cells[22,5] := '59';
Cells[23,5] := '67';
Cells[24,5] := '79';
Cells[25,5] := '96';
```

```
Cells[1,6] := '61';
Cells[2,6] := '72';
Cells[3,6] := '56';
Cells[4,6] := '33';
Cells[5,6] := '41';
Cells[6,6] := '0';
Cells[7,6] := '71';
Cells[8,6] := '98';
Cells[9,6] := '93';
Cells[10,6] := '95';
Cells[11,6] := '82';
Cells[12,6] := '60';
Cells[13,6] := '78';
Cells[14,6] := '76';
Cells[15,6] := '76';
Cells[16,6] := '95';
Cells[17,6] := '72';
Cells[18,6] := '50';
Cells[19,6] := '48';
Cells[20,6] := '75';
Cells[21,6] := '21';
Cells[22,6] := '28';
Cells[23,6] := '54';
Cells[24,6] := '24';
Cells[25,6] := '30';

Cells[1,7] := '15';
Cells[2,7] := '96';
Cells[3,7] := '32';
Cells[4,7] := '43';
Cells[5,7] := '57';
Cells[6,7] := '88';
Cells[7,7] := '0';
Cells[8,7] := '72';
Cells[9,7] := '97';
Cells[10,7] := '44';
Cells[11,7] := '58';
Cells[12,7] := '56';
Cells[13,7] := '28';
Cells[14,7] := '81';
Cells[15,7] := '65';
Cells[16,7] := '52';
Cells[17,7] := '88';
Cells[18,7] := '70';
Cells[19,7] := '58';
Cells[20,7] := '26';
Cells[21,7] := '26';
Cells[22,7] := '75';
Cells[23,7] := '90';
Cells[24,7] := '64';
Cells[25,7] := '64';

Cells[1,8] := '80';
Cells[2,8] := '50';
Cells[3,8] := '51';
Cells[4,8] := '56';
Cells[5,8] := '31';
Cells[6,8] := '80';
Cells[7,8] := '90';
Cells[8,8] := '0';
Cells[9,8] := '41';
Cells[10,8] := '27';
Cells[11,8] := '63';
Cells[12,8] := '73';
Cells[13,8] := '55';
Cells[14,8] := '72';
Cells[15,8] := '81';
Cells[16,8] := '92';
Cells[17,8] := '60';
```

```
Cells[18,8] := '54';
Cells[19,8] := '32';
Cells[20,8] := '43';
Cells[21,8] := '94';
Cells[22,8] := '72';
Cells[23,8] := '88';
Cells[24,8] := '74';
Cells[25,8] := '37';

Cells[1,9] := '84';
Cells[2,9] := '76';
Cells[3,9] := '46';
Cells[4,9] := '93';
Cells[5,9] := '27';
Cells[6,9] := '56';
Cells[7,9] := '80';
Cells[8,9] := '36';
Cells[9,9] := '0';
Cells[10,9] := '84';
Cells[11,9] := '72';
Cells[12,9] := '91';
Cells[13,9] := '85';
Cells[14,9] := '46';
Cells[15,9] := '72';
Cells[16,9] := '87';
Cells[17,9] := '68';
Cells[18,9] := '55';
Cells[19,9] := '58';
Cells[20,9] := '86';
Cells[21,9] := '89';
Cells[22,9] := '57';
Cells[23,9] := '49';
Cells[24,9] := '39';
Cells[25,9] := '84';

Cells[1,10] := '37';
Cells[2,10] := '29';
Cells[3,10] := '94';
Cells[4,10] := '21';
Cells[5,10] := '97';
Cells[6,10] := '45';
Cells[7,10] := '92';
Cells[8,10] := '28';
Cells[9,10] := '30';
Cells[10,10] := '0';
Cells[11,10] := '28';
Cells[12,10] := '29';
Cells[13,10] := '88';
Cells[14,10] := '43';
Cells[15,10] := '48';
Cells[16,10] := '47';
Cells[17,10] := '71';
Cells[18,10] := '72';
Cells[19,10] := '91';
Cells[20,10] := '59';
Cells[21,10] := '26';
Cells[22,10] := '65';
Cells[23,10] := '87';
Cells[24,10] := '98';
Cells[25,10] := '64';

Cells[1,11] := '19';
Cells[2,11] := '23';
Cells[3,11] := '91';
Cells[4,11] := '81';
Cells[5,11] := '85';
Cells[6,11] := '84';
Cells[7,11] := '83';
Cells[8,11] := '65';
Cells[9,11] := '45';
```

```
Cells[10,11] := '91';
Cells[11,11] := '0';
Cells[12,11] := '35';
Cells[13,11] := '31';
Cells[14,11] := '61';
Cells[15,11] := '28';
Cells[16,11] := '91';
Cells[17,11] := '92';
Cells[18,11] := '27';
Cells[19,11] := '77';
Cells[20,11] := '33';
Cells[21,11] := '25';
Cells[22,11] := '88';
Cells[23,11] := '73';
Cells[24,11] := '84';
Cells[25,11] := '55';

Cells[1,12] := '75';
Cells[2,12] := '62';
Cells[3,12] := '88';
Cells[4,12] := '78';
Cells[5,12] := '39';
Cells[6,12] := '20';
Cells[7,12] := '85';
Cells[8,12] := '67';
Cells[9,12] := '21';
Cells[10,12] := '90';
Cells[11,12] := '57';
Cells[12,12] := '0';
Cells[13,12] := '30';
Cells[14,12] := '85';
Cells[15,12] := '91';
Cells[16,12] := '36';
Cells[17,12] := '54';
Cells[18,12] := '66';
Cells[19,12] := '56';
Cells[20,12] := '42';
Cells[21,12] := '27';
Cells[22,12] := '38';
Cells[23,12] := '89';
Cells[24,12] := '96';
Cells[25,12] := '89';

Cells[1,13] := '23';
Cells[2,13] := '64';
Cells[3,13] := '20';
Cells[4,13] := '40';
Cells[5,13] := '88';
Cells[6,13] := '26';
Cells[7,13] := '32';
Cells[8,13] := '95';
Cells[9,13] := '92';
Cells[10,13] := '93';
Cells[11,13] := '40';
Cells[12,13] := '60';
Cells[13,13] := '0';
Cells[14,13] := '36';
Cells[15,13] := '53';
Cells[16,13] := '40';
Cells[17,13] := '46';
Cells[18,13] := '31';
Cells[19,13] := '87';
Cells[20,13] := '65';
Cells[21,13] := '77';
Cells[22,13] := '51';
Cells[23,13] := '47';
Cells[24,13] := '38';
Cells[25,13] := '48';

Cells[1,14] := '95';
```

```
Cells[2,14] := '13';
Cells[3,14] := '23';
Cells[4,14] := '91';
Cells[5,14] := '87';
Cells[6,14] := '60';
Cells[7,14] := '42';
Cells[8,14] := '98';
Cells[9,14] := '20';
Cells[10,14] := '22';
Cells[11,14] := '43';
Cells[12,14] := '71';
Cells[13,14] := '33';
Cells[14,14] := '0';
Cells[15,14] := '89';
Cells[16,14] := '54';
Cells[17,14] := '21';
Cells[18,14] := '70';
Cells[19,14] := '84';
Cells[20,14] := '64';
Cells[21,14] := '34';
Cells[22,14] := '89';
Cells[23,14] := '26';
Cells[24,14] := '42';
Cells[25,14] := '38';

Cells[1,15] := '40';
Cells[2,15] := '78';
Cells[3,15] := '62';
Cells[4,15] := '95';
Cells[5,15] := '55';
Cells[6,15] := '50';
Cells[7,15] := '72';
Cells[8,15] := '54';
Cells[9,15] := '68';
Cells[10,15] := '27';
Cells[11,15] := '93';
Cells[12,15] := '44';
Cells[13,15] := '21';
Cells[14,15] := '22';
Cells[15,15] := '0';
Cells[16,15] := '71';
Cells[17,15] := '79';
Cells[18,15] := '22';
Cells[19,15] := '79';
Cells[20,15] := '26';
Cells[21,15] := '25';
Cells[22,15] := '48';
Cells[23,15] := '76';
Cells[24,15] := '27';
Cells[25,15] := '80';

Cells[1,16] := '82';
Cells[2,16] := '81';
Cells[3,16] := '42';
Cells[4,16] := '25';
Cells[5,16] := '40';
Cells[6,16] := '48';
Cells[7,16] := '82';
Cells[8,16] := '62';
Cells[9,16] := '51';
Cells[10,16] := '54';
Cells[11,16] := '85';
Cells[12,16] := '59';
Cells[13,16] := '94';
Cells[14,16] := '25';
Cells[15,16] := '65';
Cells[16,16] := '0';
Cells[17,16] := '94';
Cells[18,16] := '21';
Cells[19,16] := '40';
```

```
Cells[20,16] := '66';
Cells[21,16] := '25';
Cells[22,16] := '33';
Cells[23,16] := '85';
Cells[24,16] := '59';
Cells[25,16] := '81';

Cells[1,17] := '57';
Cells[2,17] := '74';
Cells[3,17] := '71';
Cells[4,17] := '39';
Cells[5,17] := '92';
Cells[6,17] := '76';
Cells[7,17] := '61';
Cells[8,17] := '27';
Cells[9,17] := '76';
Cells[10,17] := '26';
Cells[11,17] := '42';
Cells[12,17] := '85';
Cells[13,17] := '34';
Cells[14,17] := '98';
Cells[15,17] := '59';
Cells[16,17] := '96';
Cells[17,17] := '0';
Cells[18,17] := '91';
Cells[19,17] := '31';
Cells[20,17] := '62';
Cells[21,17] := '34';
Cells[22,17] := '36';
Cells[23,17] := '81';
Cells[24,17] := '76';
Cells[25,17] := '49';

Cells[1,18] := '29';
Cells[2,18] := '19';
Cells[3,18] := '89';
Cells[4,18] := '84';
Cells[5,18] := '28';
Cells[6,18] := '83';
Cells[7,18] := '82';
Cells[8,18] := '77';
Cells[9,18] := '45';
Cells[10,18] := '74';
Cells[11,18] := '41';
Cells[12,18] := '71';
Cells[13,18] := '66';
Cells[14,18] := '76';
Cells[15,18] := '33';
Cells[16,18] := '84';
Cells[17,18] := '81';
Cells[18,18] := '0';
Cells[19,18] := '46';
Cells[20,18] := '78';
Cells[21,18] := '50';
Cells[22,18] := '70';
Cells[23,18] := '66';
Cells[24,18] := '85';
Cells[25,18] := '48';

Cells[1,19] := '87';
Cells[2,19] := '29';
Cells[3,19] := '45';
Cells[4,19] := '75';
Cells[5,19] := '44';
Cells[6,19] := '29';
Cells[7,19] := '28';
Cells[8,19] := '68';
Cells[9,19] := '34';
Cells[10,19] := '55';
Cells[11,19] := '55';
```

```
Cells[12,19] := '53';
Cells[13,19] := '31';
Cells[14,19] := '46';
Cells[15,19] := '48';
Cells[16,19] := '48';
Cells[17,19] := '73';
Cells[18,19] := '73';
Cells[19,19] := '0';
Cells[20,19] := '97';
Cells[21,19] := '29';
Cells[22,19] := '59';
Cells[23,19] := '85';
Cells[24,19] := '97';
Cells[25,19] := '21';

Cells[1,20] := '15';
Cells[2,20] := '87';
Cells[3,20] := '80';
Cells[4,20] := '62';
Cells[5,20] := '36';
Cells[6,20] := '86';
Cells[7,20] := '67';
Cells[8,20] := '74';
Cells[9,20] := '25';
Cells[10,20] := '30';
Cells[11,20] := '63';
Cells[12,20] := '83';
Cells[13,20] := '35';
Cells[14,20] := '58';
Cells[15,20] := '38';
Cells[16,20] := '59';
Cells[17,20] := '94';
Cells[18,20] := '85';
Cells[19,20] := '88';
Cells[20,20] := '0';
Cells[21,20] := '44';
Cells[22,20] := '68';
Cells[23,20] := '45';
Cells[24,20] := '59';
Cells[25,20] := '96';


Cells[1,21] := '87';
Cells[2,21] := '41';
Cells[3,21] := '46';
Cells[4,21] := '38';
Cells[5,21] := '82';
Cells[6,21] := '37';
Cells[7,21] := '32';
Cells[8,21] := '47';
Cells[9,21] := '98';
Cells[10,21] := '42';
Cells[11,21] := '92';
Cells[12,21] := '64';
Cells[13,21] := '64';
Cells[14,21] := '43';
Cells[15,21] := '35';
Cells[16,21] := '45';
Cells[17,21] := '32';
Cells[18,21] := '40';
Cells[19,21] := '28';
Cells[20,21] := '48';
Cells[21,21] := '0';
Cells[22,21] := '53';
Cells[23,21] := '41';
Cells[24,21] := '89';
Cells[25,21] := '43';


Cells[1,22] := '11';
```

```
Cells[2,22] := '48';
Cells[3,22] := '75';
Cells[4,22] := '84';
Cells[5,22] := '91';
Cells[6,22] := '79';
Cells[7,22] := '41';
Cells[8,22] := '69';
Cells[9,22] := '34';
Cells[10,22] := '54';
Cells[11,22] := '84';
Cells[12,22] := '81';
Cells[13,22] := '34';
Cells[14,22] := '81';
Cells[15,22] := '34';
Cells[16,22] := '59';
Cells[17,22] := '38';
Cells[18,22] := '33';
Cells[19,22] := '34';
Cells[20,22] := '51';
Cells[21,22] := '79';
Cells[22,22] := '0';
Cells[23,22] := '32';
Cells[24,22] := '98';
Cells[25,22] := '45';


Cells[1,23] := '26';
Cells[2,23] := '35';
Cells[3,23] := '88';
Cells[4,23] := '97';
Cells[5,23] := '52';
Cells[6,23] := '66';
Cells[7,23] := '89';
Cells[8,23] := '53';
Cells[9,23] := '77';
Cells[10,23] := '28';
Cells[11,23] := '64';
Cells[12,23] := '64';
Cells[13,23] := '90';
Cells[14,23] := '62';
Cells[15,23] := '43';
Cells[16,23] := '43';
Cells[17,23] := '97';
Cells[18,23] := '73';
Cells[19,23] := '89';
Cells[20,23] := '88';
Cells[21,23] := '88';
Cells[22,23] := '95';
Cells[23,23] := '0';
Cells[24,23] := '73';
Cells[25,23] := '92';


Cells[1,24] := '71';
Cells[2,24] := '30';
Cells[3,24] := '41';
Cells[4,24] := '83';
Cells[5,24] := '63';
Cells[6,24] := '53';
Cells[7,24] := '60';
Cells[8,24] := '85';
Cells[9,24] := '25';
Cells[10,24] := '97';
Cells[11,24] := '86';
Cells[12,24] := '79';
Cells[13,24] := '87';
Cells[14,24] := '83';
Cells[15,24] := '30';
Cells[16,24] := '52';
Cells[17,24] := '28';
```

```
      Cells[18,24] := '56';
      Cells[19,24] := '29';
      Cells[20,24] := '64';
      Cells[21,24] := '29';
      Cells[22,24] := '21';
      Cells[23,24] := '69';
      Cells[24,24] := '0';
      Cells[25,24] := '41';

      Cells[1,25] := '49';
      Cells[2,25] := '75';
      Cells[3,25] := '56';
      Cells[4,25] := '50';
      Cells[5,25] := '37';
      Cells[6,25] := '35';
      Cells[7,25] := '93';
      Cells[8,25] := '60';
      Cells[9,25] := '72';
      Cells[10,25] := '79';
      Cells[11,25] := '83';
      Cells[12,25] := '93';
      Cells[13,25] := '98';
      Cells[14,25] := '90';
      Cells[15,25] := '35';
      Cells[16,25] := '91';
      Cells[17,25] := '53';
      Cells[18,25] := '34';
      Cells[19,25] := '52';
      Cells[20,25] := '41';
      Cells[21,25] := '91';
      Cells[22,25] := '41';
      Cells[23,25] := '75';
      Cells[24,25] := '57';
      Cells[25,25] := '0';

  end;   // of data kota
end;

procedure TForm1.SpeedButton6Click(Sender: TObject);
begin
  DataKota.Visible  := False;
  LogInduk1.Visible := True;
  LogInduk2.Visible := False;
  LogAnak.Visible   := False;
  Matriks1.Visible  := False;
  Matriks2.Visible  := False;
  Matriks3.Visible  := False;
end;

procedure TForm1.SpeedButton7Click(Sender: TObject);
begin
  DataKota.Visible  := False;
  LogInduk1.Visible := False;
  LogInduk2.Visible := False;
  LogAnak.Visible   := False;
  Matriks1.Visible  := True;
  Matriks2.Visible  := False;
  Matriks3.Visible  := False;
end;

procedure TForm1.SpeedButton8Click(Sender: TObject);
begin
  DataKota.Visible  := False;
  LogInduk1.Visible := False;
  LogInduk2.Visible := True;
  LogAnak.Visible   := False;
  Matriks1.Visible  := False;
  Matriks2.Visible  := False;
  Matriks3.Visible  := False;
end;
```

```
procedure TForm1.SpeedButton9Click(Sender: TObject);
begin
  DataKota.Visible  := False;
  LogInduk1.Visible := False;
  LogInduk2.Visible := False;
  LogAnak.Visible   := False;
  Matriks1.Visible  := False;
  Matriks2.Visible  := True;
  Matriks3.Visible  := False;
end;

procedure TForm1.InterseksiClick(Sender: TObject);
var
  Data : array[1..30,1..30] of integer;
  Hasil : array[1..30] of integer;
  Nilai : array[1..30] of integer;
  total : array[1..5] of integer;
  Total_Kecil_1, Total_Kecil_2 : LongInt;
  i, k, j, u, g : integer;
  r, c, x, xs, z, p, n, t, jp : integer;
  num, itung, kota, balik : integer;
  Jumlah : array[1..30] of Integer;
  h, ha : String;
  s : Boolean;
begin
//ListBox1.Clear;
listbox1.Visible := False;
Panel2.Visible := False;
Memo2.Clear;
//ngitung faktorial
num := 1;
kota := strtoint(JmlUlang.Text);
for itung := 1 to kota do
begin
  num := num*itung;
  label3.Caption := inttostr(num);
end;

for g := 1 to 1000 do
begin
  label2.Caption := inttostr(g);
  Interseksi.Enabled := false;

  // induk 1
  LKromosom1.Caption := 'Kromosom 1 :';
  Kromosom1.Caption := Perhitungan( 'INDUK 1', LogInduk1, MyHasil1, Matriks1);
  PnlInduk1.Visible := True;
  PnlInduk1.Refresh;

  // induk 2
  LKromosom2.Caption := 'Kromosom 2 :';
  Kromosom2.Caption := Perhitungan( 'INDUK 2', LogInduk2, MyHasil2, Matriks2);
  PnlInduk2.Visible := True;
  PnlInduk2.Refresh;

  // buat Judul
  for c:=1 to JumlahKota do
      Matriks3.Cells[c,0] := inttostr(c);

  for r:=1 to JumlahKota do
      Matriks3.Cells[0,r] := inttostr(r);

  for jp:=1 to strtoint(JmlIterasi.Text) do
  begin
    // proses intrseksi
    for r:=1 to JumlahKota do
      for c:=1 to JumlahKota do
      begin
          if Matriks1.Cells[c,r]=Matriks2.Cells[c,r] then
```

```
         Matriks3.Cells[c,r] := Matriks1.Cells[c,r]
      else
         Matriks3.Cells[c,r] := '0';
  end;

LogAnak.Lines.Clear;
LogAnak.Lines.Add('MENCARI MPO');
LogAnak.Lines.Add('---------------------------');
LogAnak.Lines.Add('');

LogAnak.Lines.Add('Jumlah setiap colom hasil interseksi');
// hitung jumlah setiap colom
MyIndexSort.Items.Clear;
for c:=1 to JumlahKota do
begin
  Jumlah[c] := 0;
  for r:=1 to JumlahKota do
     if Matriks3.Cells[c,r]='1' then
        Jumlah[c] := Jumlah[c] + 1;
  MyIndexSort.Items.Add( FormatFloat( '000', Jumlah[c] ) + FormatFloat( '000', c) );
  LogAnak.Lines.Add( IntToStr(Jumlah[c]) +' , '+ IntToStr(c)) ;
end;

LogAnak.Lines.Add('');
LogAnak.Lines.Add('Hasil Pengurutan');
for c:=1 to JumlahKota do
begin
  MyIndexSort.ItemIndex := c-1;
  LogAnak.Lines.Add( IntToStr( StrToInt( Copy( MyIndexSort.Text, 1, 3) ) ) +' , '+
                     IntToStr( StrToInt( Copy( MyIndexSort.Text, 4, 3) ) ) ) ;
end;

LogAnak.Lines.Add('');
LogAnak.Lines.Add('Hasil MPO');
MyIndexSort.ItemIndex := 0;
h := MyIndexSort.Text;
MPO.Caption := '';
for c:=2 to JumlahKota do
begin
    MyIndexSort.ItemIndex := c-1;
    if Copy( h, 1, 3)<>Copy( MyIndexSort.Text, 1, 3) then
       MPO.Caption := MPO.Caption + 'V'+ IntToStr( StrToInt( Copy( h, 4, 3) ) ) + ',';
    h := MyIndexSort.Text;
end;
MPO.Caption := MPO.Caption + 'V'+ IntToStr( StrToInt( Copy( h, 4, 3) ) ) + ',';
LogAnak.Lines.Add( MPO.Caption ) ;

Total_Kecil_1 := 999999999;   // diberi nilai besar sebagai switch
Total_Kecil_2 := 999999999;   // diberi nilai besar sebagai switch
for z := 1 to 5 do
begin
    LogAnak.Lines.Add('');
    LogAnak.Lines.Add('MENCARI KROMOSOM ANAK '+IntToStr(z));
    LogAnak.Lines.Add('---------------------------');
    // -------------- buat index -----------------------------
    MyIndex.Items.Clear;
    for c:= 1 to JumlahKota do
       MyIndex.Items.Add( IntToStr(c) );

    //--------------- Data[ Baris, Colom ] -------------------
    for c:=1 to JumlahKota do
      for r:=1 to JumlahKota do
          Data[c,r] := StrToInt( DataKota.Cells[r,c]);

    // masukkan nilai MPO
    p := 0;
    MyIndexSort.ItemIndex := 0;
    h := MyIndexSort.Text;
    for c:=2 to JumlahKota do
    begin
```

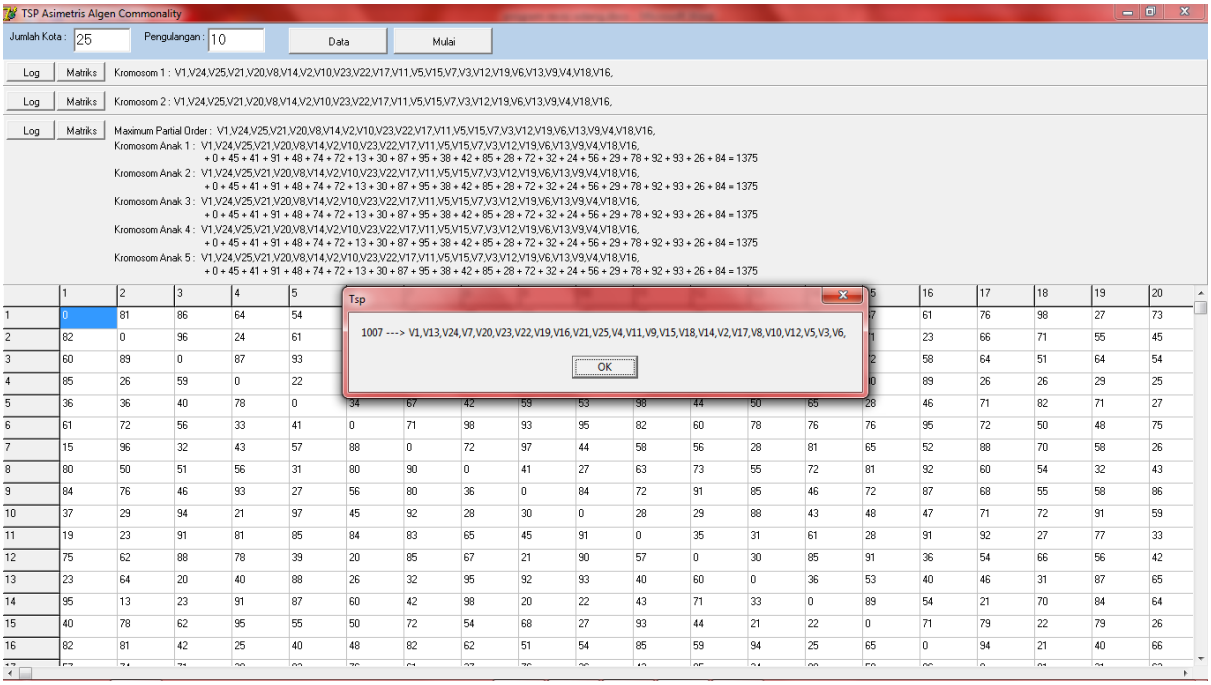
```
            MyIndexSort.ItemIndex := c-1;
            if Copy( h, 1, 3)<>Copy( MyIndexSort.Text, 1, 3) then
            begin
               p := p + 1;
               AturUlangMyIndex( StrToInt( Copy( h, 4, 3) ) );
               Hasil[p] := StrToInt( Copy( h, 4, 3) ) ;
               if p=1 then i := StrToInt( Copy( h, 4, 3) );
               if p=2 then j := StrToInt( Copy( h, 4, 3) );
            end;
            h := MyIndexSort.Text;
         end;
         p := p + 1;
         AturUlangMyIndex( StrToInt( Copy( h, 4, 3) ) );
         Hasil[p] := StrToInt( Copy( h, 4, 3) ) ;

         h := '';
         for x:=1 to p do
            h := h + 'V'+inttostr( Hasil[x] )+ ', ';
         LogAnak.Lines.Add( 'Hasil ->'+h);
         LogAnak.Lines.Add( 'i = '+inttostr(i) );
         LogAnak.Lines.Add( 'j = '+inttostr(j) );

         // lakukan perulangan untuk proses selanjutnya
         for c := p+1 to JumlahKota do
         begin
               k := CariTitik;
               LogAnak.Lines.Add( 'k = '+inttostr(k) );
               AturUlangMyIndex( k );

               // cari nilai
               for x:=2 to p do
               begin
                  j := Hasil[x];
                  Nilai[x] := Data[i,k] + Data[k,j] - Data[i,j];

                  LogAnak.Lines.Add( inttostr(Data[i,k])+' + '+inttostr(Data[k,j])+' -
'+inttostr(Data[i,j])+' = '+inttostr(Nilai[x]) );
               end;

               // Cari index dengan Nilai Paling Kecil
               n := 2;
               for x:=3 to p do
                  if Nilai[n]>Nilai[x] then n := x;
               t := Hasil[n];

               h := '';
               for x:=1 to p do
                  h := h + 'V'+inttostr( Hasil[x] )+ ', ';
               LogAnak.Lines.Add( 'Hasil ->'+h);
               inc(p);   // ---> proses ke X

               LogAnak.Lines.Add( '' );

               // Sisipkan nilai n ke Hasil
               s := false;
               for x:=1 to p do
               begin
                  if Hasil[x]=t then
                     s := true;

                  if s then
                  begin
                     t := Hasil[x];  // tampung sementara
                     Hasil[x] := k;  // isi hasil dengan k
                     k := t;         // var k isi dengan t
                  end;
               end;
         end; // of for z

         // keluarkan hasilnya
```

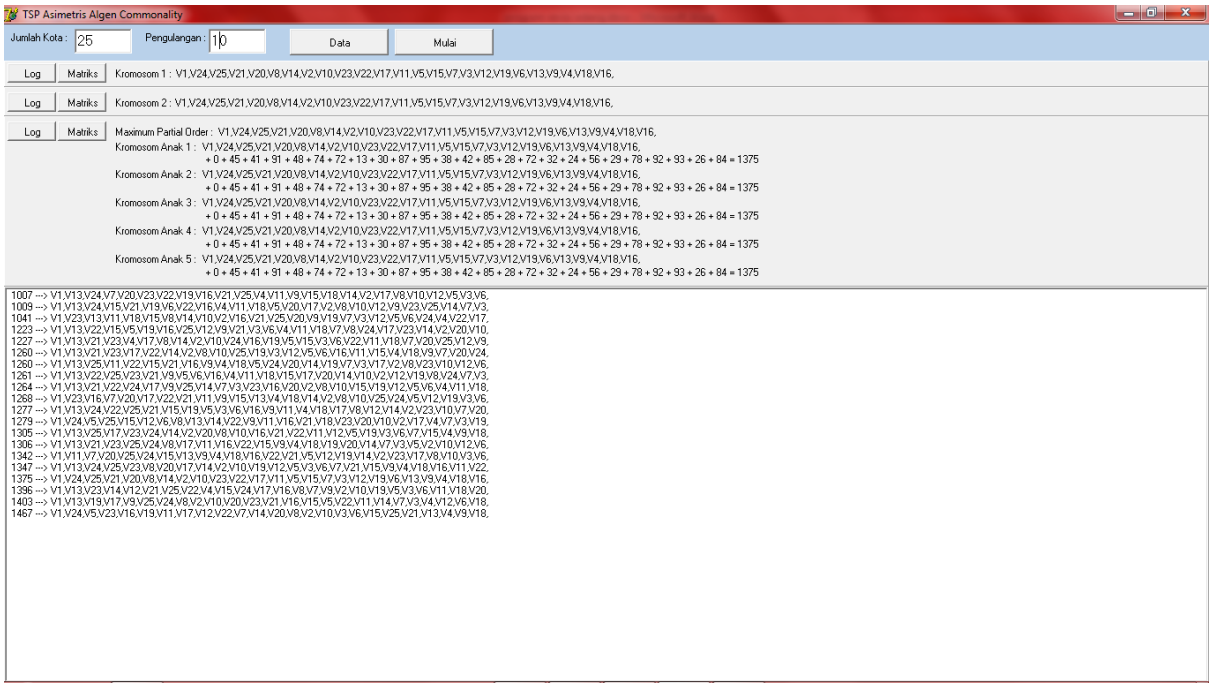
```
        MyHasil3.Items.Clear;
        t := 0;
        h := '';
        ha := '';
        xs := 1;
        for x:=1 to JumlahKota do
        begin
           h := h + 'V'+inttostr( Hasil[x] )+ ',';
           t := t + Data[ xs, Hasil[x]];
           ha := ha +' + '+inttostr( Data[ xs, Hasil[x]] );
           MyHasil3.Items.Add( inttostr( Hasil[x] ) );
           xs := Hasil[x];
        end;
        //Memo1.Lines.Add(inttostr(t));
        if z=1 then KromosomAnak1.Caption := h;
        if z=2 then KromosomAnak2.Caption := h;
        if z=3 then KromosomAnak3.Caption := h;
        if z=4 then KromosomAnak4.Caption := h;
        if z=5 then KromosomAnak5.Caption := h;

        if z=1 then HasilAnak1.Caption := ha+' = '+IntToStr(t);
        if z=2 then HasilAnak2.Caption := ha+' = '+IntToStr(t);
        if z=3 then HasilAnak3.Caption := ha+' = '+IntToStr(t);
        if z=4 then HasilAnak4.Caption := ha+' = '+IntToStr(t);
        if z=5 then HasilAnak5.Caption := ha+' = '+IntToStr(t);


        // cari nilai paling kecil
        if t < Total_Kecil_1 then
        begin
            // ambil hasil kromosom terkecil 2
            Total_Kecil_2 := Total_Kecil_1;
            Kromosom2.Caption := Kromosom1.Caption;
            MyHasil2.Items := MyHasil1.Items;
            for c:=1 to JumlahKota do
              for r:=1 to JumlahKota do
                  Matriks2.Cells[c,r] := Matriks1.Cells[c,r];

            // ambil hasil kromosom terkecil 1
            Total_Kecil_1 := t;
            Kromosom1.Caption := h;
            MyHasil1.Items := MyHasil3.Items;

            // buat matrik
            with Matriks1 do
            begin
                // buat Judul
                for c:=1 to JumlahKota do
                    Cells[c,0] := inttostr(c);

                for r:=1 to JumlahKota do
                    Cells[0,r] := inttostr(r);

                for r:=1 to JumlahKota do
                  for c:=1 to JumlahKota do
                      if CariPosisi( MyHasil3, IntToStr(c) ) > CariPosisi( MyHasil3, IntToStr(r))
then
                         Cells[c,r] := '1'
                      else
                         Cells[c,r] := '0';
            end;
        end
        else if t < Total_Kecil_2 then
        begin
            // ambil hasil kromosom terkecil 1
            Total_Kecil_2 := t;
            Kromosom2.Caption := h;
            MyHasil2.Items := MyHasil3.Items;

            // buat matrik
```

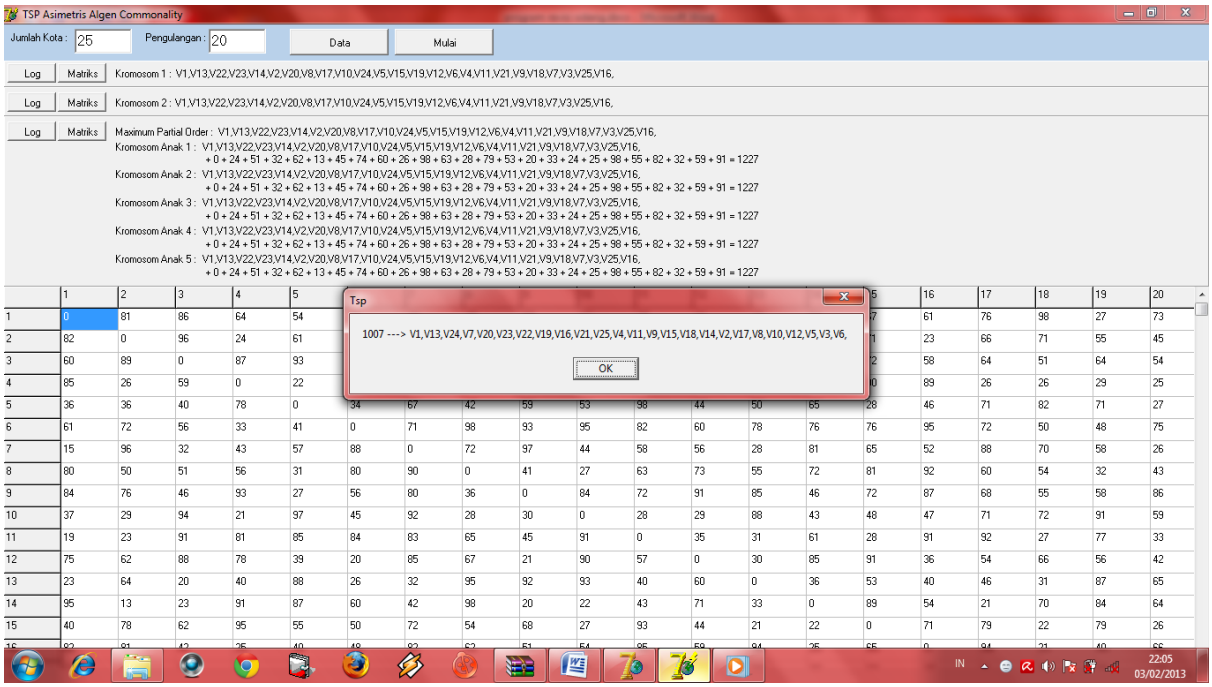
```
            with Matriks2 do
            begin
                // buat Judul
                for c:=1 to JumlahKota do
                    Cells[c,0] := inttostr(c);

                for r:=1 to JumlahKota do
                    Cells[0,r] := inttostr(r);

                for r:=1 to JumlahKota do
                  for c:=1 to JumlahKota do
                      if CariPosisi( MyHasil3, IntToStr(c) ) > CariPosisi( MyHasil3, IntToStr(r))
then
                         Cells[c,r] := '1'
                      else
                         Cells[c,r] := '0';
            end;
        end;
        LKromosom1.Caption := 'Kromosom 1 :'; //+IntToStr(jp)+' :';
        LKromosom2.Caption := 'Kromosom 2 :'; //+IntToStr(jp)+' :';
        PnlInduk1.Refresh;
        PnlInduk2.Refresh;
    end;
    PnlInterseksi.Refresh;
    //Memo2.Visible:=true;

  end; // of jml ulang
  listbox1.Items.Add(' ' + inttostr(t)+ ' ---> ' + h);
  PnlInterseksi.Visible := True;
  Interseksi.Enabled := True;
end;
   Memo2.Clear;
   Memo2.Lines.Add(copy(listbox1.Items.Text,0,700));
   Memo2.Lines.Add(copy(listbox1.Items.Text,1,700));
   Memo2.Lines.Add(copy(listbox1.Items.Text,2,700));
   Memo2.Lines.Add(copy(listbox1.Items.Text,3,700));
   Memo2.Lines.Add(copy(listbox1.Items.Text,4,700));
   ShowMessage(Memo2.Lines.Strings[0]);
   Panel2.Visible := True;
   ListBox1.Visible := True;
   //PnlInterseksi.Visible := False;
   //PnlInduk1.Visible := False;
end;

procedure TForm1.SpeedButton10Click(Sender: TObject);
begin
  DataKota.Visible  := False;
  LogInduk1.Visible := False;
  LogInduk2.Visible := False;
  LogAnak.Visible := False;
  Matriks1.Visible  := False;
  Matriks2.Visible  := False;
  Matriks3.Visible  := True;
end;

procedure TForm1.SpeedButton3Click(Sender: TObject);
begin
  DataKota.Visible  := False;
  LogInduk1.Visible := False;
  LogInduk2.Visible := False;
  LogAnak.Visible   := True;
  Matriks1.Visible  := False;
  Matriks2.Visible  := False;
  Matriks3.Visible  := False;
end;

//procedure TForm1.SpeedButton1Click(Sender: TObject);
//begin
  //DataKota.Visible  := True;
  //LogInduk1.Visible := False;
```

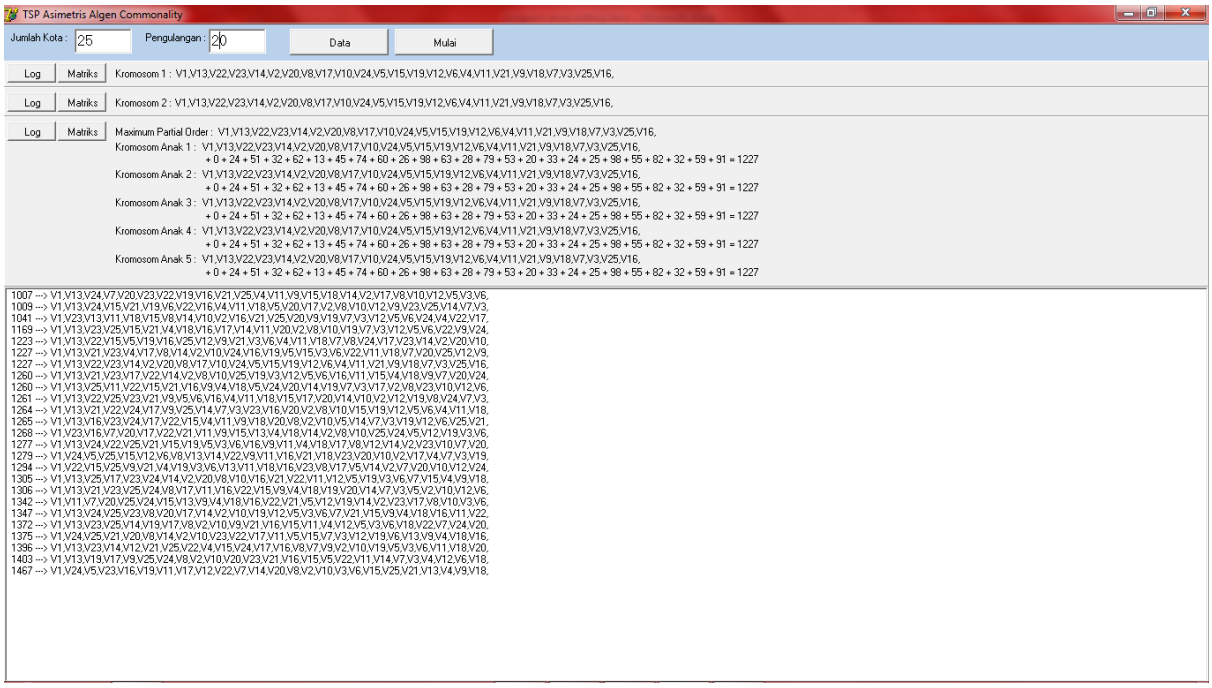
```
  //LogInduk2.Visible := False;
  //LogAnak.Visible   := False;
  //Matriks1.Visible  := False;
  //Matriks2.Visible  := False;
  //Matriks3.Visible  := False;

  //DataKota.Refresh;
//end;

procedure TForm1.SpeedButton2Click(Sender: TObject);
begin
  listbox1.Visible := True;
end;

end.
```

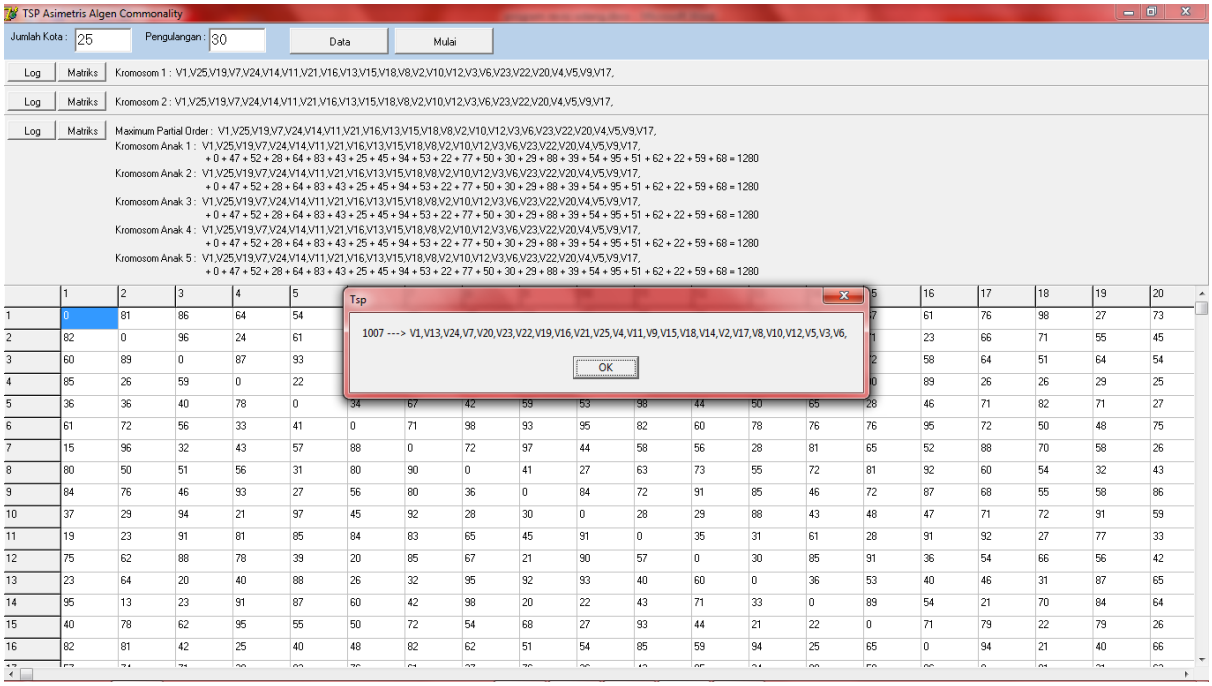
Lampiran 4 Tampilan Output 10 Pengujian dengan 25 Kota

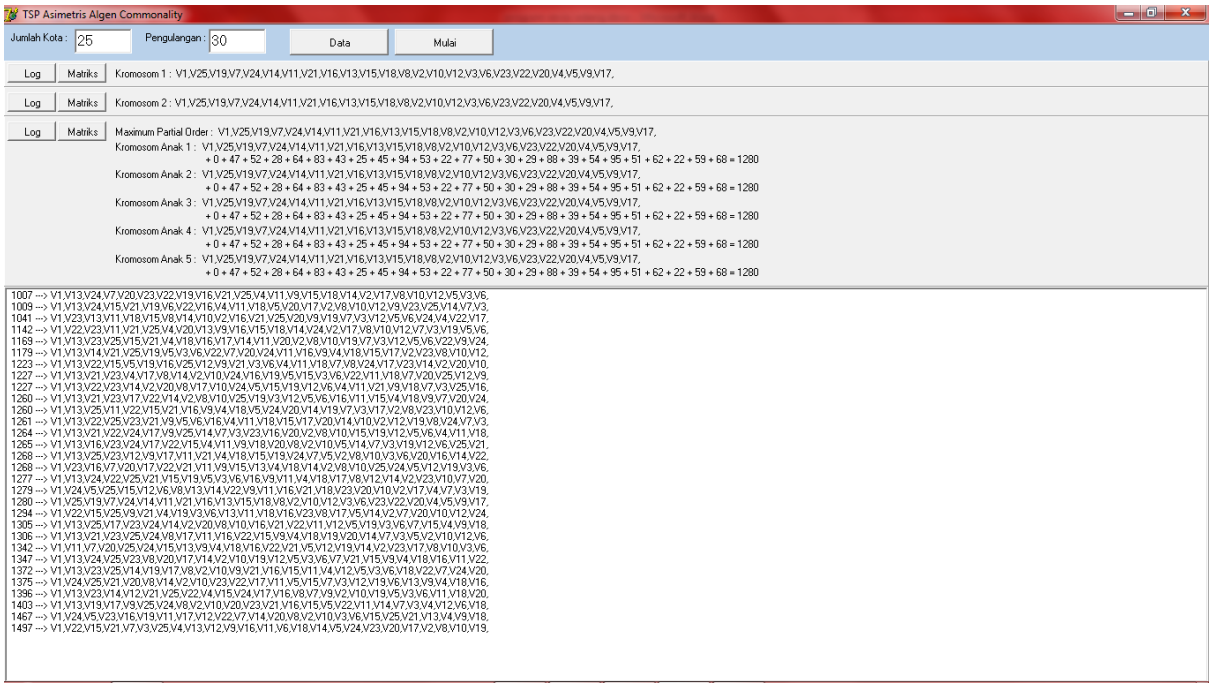


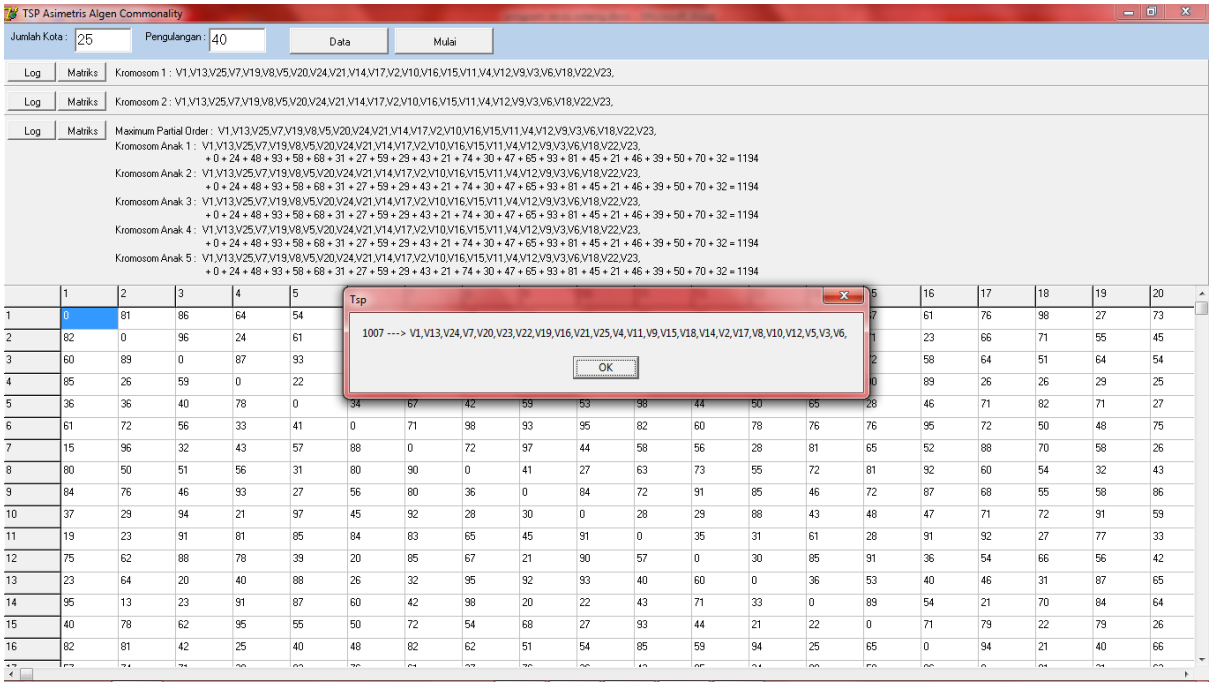

TSP Asimetris Algen Commonality
Jumlah Kota : 25
Pengulangan : 10
Data
Mulai
Log
Matriks
Kromosom 1 : V1,V24,V25,V21,V20,V8,V14,V2,V10,V23,V22,V17,V11,V5,V15,V7,V3,V12,V19,V6,V13,V9,V4,V18,V16,
Kromosom 2 : V1,V24,V25,V21,V20,V8,V14,V2,V10,V23,V22,V17,V11,V5,V15,V7,V3,V12,V19,V6,V13,V9,V4,V18,V16,
Maximum Partial Order : V1,V24,V25,V21,V20,V8,V14,V2,V10,V23,V22,V17,V11,V5,V15,V7,V3,V12,V19,V6,V13,V9,V4,V18,V16,
Kromosom Anak 1 : V1,V24,V25,V21,V20,V8,V14,V2,V10,V23,V22,V17,V11,V5,V15,V7,V3,V12,V19,V6,V13,V9,V4,V18,V16,
+ 0 + 45 + 41 + 91 + 48 + 74 + 72 + 13 + 30 + 87 + 95 + 38 + 42 + 85 + 28 + 72 + 32 + 24 + 56 + 29 + 78 + 92 + 93 + 26 + 84 = 1375
Kromosom Anak 2 : V1,V24,V25,V21,V20,V8,V14,V2,V10,V23,V22,V17,V11,V5,V15,V7,V3,V12,V19,V6,V13,V9,V4,V18,V16,
+ 0 + 45 + 41 + 91 + 48 + 74 + 72 + 13 + 30 + 87 + 95 + 38 + 42 + 85 + 28 + 72 + 32 + 24 + 56 + 29 + 78 + 92 + 93 + 26 + 84 = 1375
Kromosom Anak 3 : V1,V24,V25,V21,V20,V8,V14,V2,V10,V23,V22,V17,V11,V5,V15,V7,V3,V12,V19,V6,V13,V9,V4,V18,V16,
+ 0 + 45 + 41 + 91 + 48 + 74 + 72 + 13 + 30 + 87 + 95 + 38 + 42 + 85 + 28 + 72 + 32 + 24 + 56 + 29 + 78 + 92 + 93 + 26 + 84 = 1375
Kromosom Anak 4 : V1,V24,V25,V21,V20,V8,V14,V2,V10,V23,V22,V17,V11,V5,V15,V7,V3,V12,V19,V6,V13,V9,V4,V18,V16,
+ 0 + 45 + 41 + 91 + 48 + 74 + 72 + 13 + 30 + 87 + 95 + 38 + 42 + 85 + 28 + 72 + 32 + 24 + 56 + 29 + 78 + 92 + 93 + 26 + 84 = 1375
Kromosom Anak 5 : V1,V24,V25,V21,V20,V8,V14,V2,V10,V23,V22,V17,V11,V5,V15,V7,V3,V12,V19,V6,V13,V9,V4,V18,V16,
+ 0 + 45 + 41 + 91 + 48 + 74 + 72 + 13 + 30 + 87 + 95 + 38 + 42 + 85 + 28 + 72 + 32 + 24 + 56 + 29 + 78 + 92 + 93 + 26 + 84 = 1375
1007 ---> V1,V13,V24,V7,V20,V23,V22,V19,V16,V21,V25,V4,V11,V9,V15,V18,V14,V2,V17,V8,V10,V12,V5,V3,V6,
1009 ---> V1,V13,V24,V15,V21,V19,V6,V22,V16,V4,V11,V18,V5,V20,V17,V2,V8,V10,V12,V9,V23,V25,V14,V7,V3,
1041 ---> V1,V23,V13,V11,V18,V15,V8,V14,V10,V2,V16,V21,V25,V20,V9,V19,V7,V3,V12,V5,V6,V24,V4,V22,V17,
1223 ---> V1,V13,V22,V15,V5,V19,V16,V25,V12,V9,V21,V3,V6,V4,V11,V18,V7,V8,V24,V17,V23,V14,V2,V20,V10,
1227 ---> V1,V13,V21,V23,V4,V17,V8,V14,V2,V10,V24,V16,V19,V5,V15,V3,V6,V22,V11,V18,V7,V20,V25,V12,V9,
1260 ---> V1,V13,V21,V23,V17,V22,V14,V2,V8,V10,V25,V19,V3,V12,V5,V6,V16,V11,V15,V4,V18,V9,V7,V20,V24,
1260 ---> V1,V13,V25,V11,V22,V15,V21,V16,V9,V4,V18,V5,V24,V20,V14,V19,V7,V3,V17,V2,V8,V23,V10,V12,V6,
1261 ---> V1,V13,V22,V25,V23,V21,V9,V5,V6,V16,V4,V11,V18,V15,V17,V20,V14,V10,V2,V12,V19,V8,V24,V7,V3,
1264 ---> V1,V13,V21,V22,V24,V17,V9,V25,V14,V7,V3,V23,V16,V20,V2,V8,V10,V15,V19,V12,V5,V6,V4,V11,V18,
1268 ---> V1,V23,V16,V7,V20,V17,V22,V21,V11,V9,V15,V13,V4,V18,V14,V2,V8,V10,V25,V24,V5,V12,V19,V3,V6,
1277 ---> V1,V13,V24,V22,V25,V21,V15,V19,V5,V3,V6,V16,V9,V11,V4,V18,V17,V8,V12,V14,V2,V23,V10,V7,V20,
1279 ---> V1,V24,V5,V25,V15,V12,V6,V8,V13,V14,V22,V9,V11,V16,V21,V18,V23,V20,V10,V2,V17,V4,V7,V3,V19,
1305 ---> V1,V13,V25,V17,V23,V24,V14,V2,V20,V8,V10,V16,V21,V22,V11,V12,V5,V19,V3,V6,V7,V15,V4,V9,V18,
1306 ---> V1,V13,V21,V23,V25,V24,V8,V17,V11,V16,V22,V15,V9,V4,V18,V19,V20,V14,V7,V3,V5,V2,V10,V12,V6,
1342 ---> V1,V11,V7,V20,V25,V24,V15,V13,V9,V4,V18,V16,V22,V21,V5,V12,V19,V14,V2,V23,V17,V8,V10,V3,V6,
1347 ---> V1,V13,V24,V25,V23,V8,V20,V17,V14,V2,V10,V19,V12,V5,V3,V6,V7,V21,V15,V9,V4,V18,V16,V11,V22,
1375 ---> V1,V24,V25,V21,V20,V8,V14,V2,V10,V23,V22,V17,V11,V5,V15,V7,V3,V12,V19,V6,V13,V9,V4,V18,V16,
1396 ---> V1,V13,V23,V14,V12,V21,V25,V22,V4,V15,V24,V17,V16,V8,V7,V9,V2,V10,V19,V5,V3,V6,V11,V18,V20,
1403 ---> V1,V13,V19,V17,V9,V25,V24,V8,V2,V10,V20,V23,V21,V16,V15,V5,V22,V11,V14,V7,V3,V4,V12,V6,V18,
1467 ---> V1,V24,V5,V23,V16,V19,V11,V17,V12,V22,V7,V14,V20,V8,V2,V10,V3,V6,V15,V25,V21,V13,V4,V9,V18,

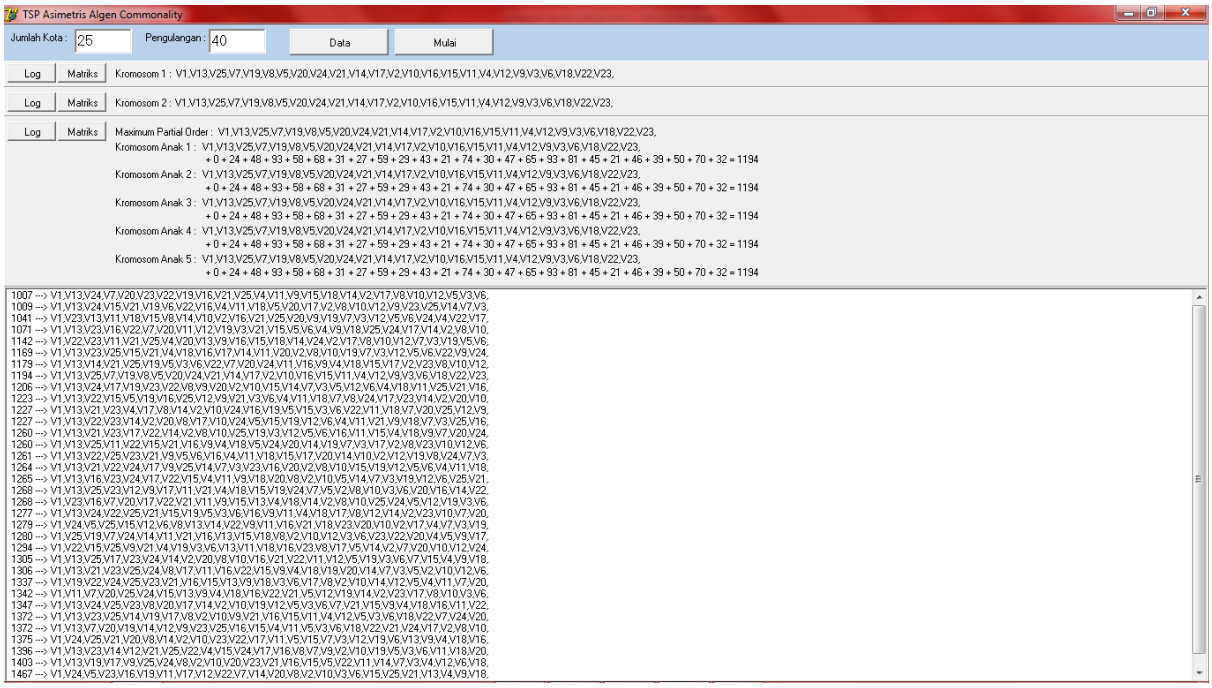

TSP Asimetris Algen Commonality
Jumlah Kota : 25
Pengulangan : 20
Data
Mulai
Log
Matriks
Kromosom 1 : V1,V13,V22,V23,V14,V2,V20,V8,V17,V10,V24,V5,V15,V19,V12,V6,V4,V11,V21,V9,V18,V7,V3,V25,V16,
Kromosom 2 : V1,V13,V22,V23,V14,V2,V20,V8,V17,V10,V24,V5,V15,V19,V12,V6,V4,V11,V21,V9,V18,V7,V3,V25,V16,
Maximum Partial Order : V1,V13,V22,V23,V14,V2,V20,V8,V17,V10,V24,V5,V15,V19,V12,V6,V4,V11,V21,V9,V18,V7,V3,V25,V16,
Kromosom Anak 1 : V1,V13,V22,V23,V14,V2,V20,V8,V17,V10,V24,V5,V15,V19,V12,V6,V4,V11,V21,V9,V18,V7,V3,V25,V16,
+ 0 + 24 + 51 + 32 + 62 + 13 + 45 + 74 + 60 + 26 + 98 + 63 + 28 + 79 + 53 + 20 + 33 + 24 + 25 + 98 + 55 + 82 + 32 + 59 + 91 = 1227
Kromosom Anak 2 : V1,V13,V22,V23,V14,V2,V20,V8,V17,V10,V24,V5,V15,V19,V12,V6,V4,V11,V21,V9,V18,V7,V3,V25,V16,
+ 0 + 24 + 51 + 32 + 62 + 13 + 45 + 74 + 60 + 26 + 98 + 63 + 28 + 79 + 53 + 20 + 33 + 24 + 25 + 98 + 55 + 82 + 32 + 59 + 91 = 1227
Kromosom Anak 3 : V1,V13,V22,V23,V14,V2,V20,V8,V17,V10,V24,V5,V15,V19,V12,V6,V4,V11,V21,V9,V18,V7,V3,V25,V16,
+ 0 + 24 + 51 + 32 + 62 + 13 + 45 + 74 + 60 + 26 + 98 + 63 + 28 + 79 + 53 + 20 + 33 + 24 + 25 + 98 + 55 + 82 + 32 + 59 + 91 = 1227
Kromosom Anak 4 : V1,V13,V22,V23,V14,V2,V20,V8,V17,V10,V24,V5,V15,V19,V12,V6,V4,V11,V21,V9,V18,V7,V3,V25,V16,
+ 0 + 24 + 51 + 32 + 62 + 13 + 45 + 74 + 60 + 26 + 98 + 63 + 28 + 79 + 53 + 20 + 33 + 24 + 25 + 98 + 55 + 82 + 32 + 59 + 91 = 1227
Kromosom Anak 5 : V1,V13,V22,V23,V14,V2,V20,V8,V17,V10,V24,V5,V15,V19,V12,V6,V4,V11,V21,V9,V18,V7,V3,V25,V16,
+ 0 + 24 + 51 + 32 + 62 + 13 + 45 + 74 + 60 + 26 + 98 + 63 + 28 + 79 + 53 + 20 + 33 + 24 + 25 + 98 + 55 + 82 + 32 + 59 + 91 = 1227
Tsp
1007 ---> V1,V13,V24,V7,V20,V23,V22,V19,V16,V21,V25,V4,V11,V9,V15,V18,V14,V2,V17,V8,V10,V12,V5,V3,V6,
OK
22:05
03/02/2013

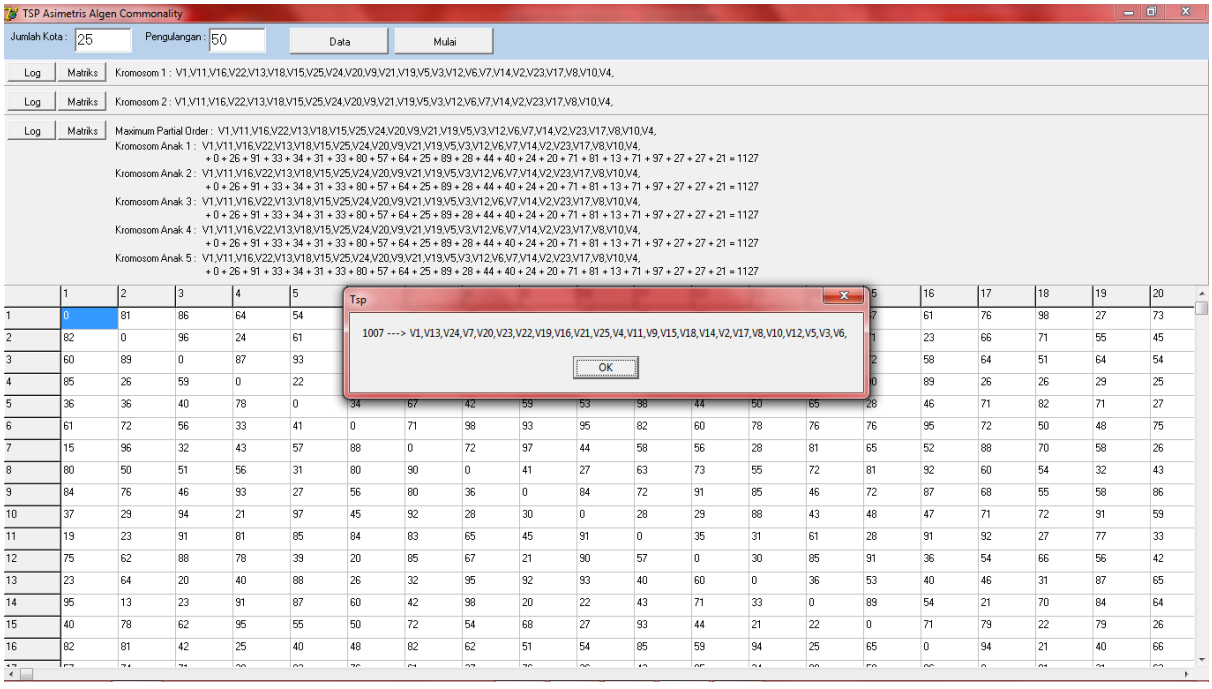

TSP Asimetris Algen Commonality
Jumlah Kota : 25
Pengulangan : 20
Data
Mulai
Log
Matriks
Kromosom 1 : V1,V13,V22,V23,V14,V2,V20,V8,V17,V10,V24,V5,V15,V19,V12,V6,V4,V11,V21,V9,V18,V7,V3,V25,V16,
Kromosom 2 : V1,V13,V22,V23,V14,V2,V20,V8,V17,V10,V24,V5,V15,V19,V12,V6,V4,V11,V21,V9,V18,V7,V3,V25,V16,
Maximum Partial Order : V1,V13,V22,V23,V14,V2,V20,V8,V17,V10,V24,V5,V15,V19,V12,V6,V4,V11,V21,V9,V18,V7,V3,V25,V16,
Kromosom Anak 1 : V1,V13,V22,V23,V14,V2,V20,V8,V17,V10,V24,V5,V15,V19,V12,V6,V4,V11,V21,V9,V18,V7,V3,V25,V16,
+ 0 + 24 + 51 + 32 + 62 + 13 + 45 + 74 + 60 + 26 + 98 + 63 + 28 + 79 + 53 + 20 + 33 + 24 + 25 + 98 + 55 + 82 + 32 + 59 + 91 = 1227
Kromosom Anak 2 : V1,V13,V22,V23,V14,V2,V20,V8,V17,V10,V24,V5,V15,V19,V12,V6,V4,V11,V21,V9,V18,V7,V3,V25,V16,
+ 0 + 24 + 51 + 32 + 62 + 13 + 45 + 74 + 60 + 26 + 98 + 63 + 28 + 79 + 53 + 20 + 33 + 24 + 25 + 98 + 55 + 82 + 32 + 59 + 91 = 1227
Kromosom Anak 3 : V1,V13,V22,V23,V14,V2,V20,V8,V17,V10,V24,V5,V15,V19,V12,V6,V4,V11,V21,V9,V18,V7,V3,V25,V16,
+ 0 + 24 + 51 + 32 + 62 + 13 + 45 + 74 + 60 + 26 + 98 + 63 + 28 + 79 + 53 + 20 + 33 + 24 + 25 + 98 + 55 + 82 + 32 + 59 + 91 = 1227
Kromosom Anak 4 : V1,V13,V22,V23,V14,V2,V20,V8,V17,V10,V24,V5,V15,V19,V12,V6,V4,V11,V21,V9,V18,V7,V3,V25,V16,
+ 0 + 24 + 51 + 32 + 62 + 13 + 45 + 74 + 60 + 26 + 98 + 63 + 28 + 79 + 53 + 20 + 33 + 24 + 25 + 98 + 55 + 82 + 32 + 59 + 91 = 1227
Kromosom Anak 5 : V1,V13,V22,V23,V14,V2,V20,V8,V17,V10,V24,V5,V15,V19,V12,V6,V4,V11,V21,V9,V18,V7,V3,V25,V16,
+ 0 + 24 + 51 + 32 + 62 + 13 + 45 + 74 + 60 + 26 + 98 + 63 + 28 + 79 + 53 + 20 + 33 + 24 + 25 + 98 + 55 + 82 + 32 + 59 + 91 = 1227
1007 ---> V1,V13,V24,V7,V20,V23,V22,V19,V16,V21,V25,V4,V11,V9,V15,V18,V14,V2,V17,V8,V10,V12,V5,V3,V6,
1009 ---> V1,V13,V24,V15,V21,V19,V6,V22,V16,V4,V11,V18,V5,V20,V17,V2,V8,V10,V12,V9,V23,V25,V14,V7,V3,
1041 ---> V1,V23,V13,V11,V18,V15,V8,V14,V10,V2,V16,V21,V25,V20,V9,V19,V7,V3,V12,V5,V6,V24,V4,V22,V17,
1169 ---> V1,V13,V23,V25,V15,V21,V4,V18,V16,V17,V14,V11,V20,V2,V8,V10,V19,V7,V3,V12,V5,V6,V22,V9,V24,
1223 ---> V1,V13,V22,V15,V6,V19,V16,V25,V12,V9,V21,V3,V6,V4,V11,V18,V7,V8,V24,V17,V23,V14,V2,V20,V10,
1227 ---> V1,V13,V21,V23,V4,V17,V8,V14,V2,V10,V24,V18,V19,V5,V15,V3,V6,V22,V11,V18,V7,V20,V25,V12,V9,
1227 ---> V1,V13,V22,V23,V14,V2,V20,V8,V17,V10,V24,V5,V15,V19,V12,V6,V4,V11,V21,V9,V18,V7,V3,V25,V16,
1260 ---> V1,V13,V21,V23,V17,V22,V14,V2,V8,V10,V25,V19,V3,V12,V5,V6,V16,V11,V15,V4,V18,V9,V7,V20,V24,
1260 ---> V1,V13,V25,V11,V22,V15,V21,V16,V9,V4,V18,V5,V24,V20,V14,V19,V7,V3,V17,V2,V8,V23,V10,V12,V6,
1261 ---> V1,V13,V22,V25,V23,V21,V9,V5,V6,V16,V4,V11,V18,V15,V17,V20,V14,V10,V2,V12,V19,V8,V24,V7,V3,
1264 ---> V1,V13,V21,V22,V24,V17,V9,V25,V14,V7,V3,V23,V16,V20,V2,V8,V10,V15,V19,V12,V5,V6,V4,V11,V18,
1265 ---> V1,V13,V16,V23,V24,V17,V22,V15,V4,V11,V9,V18,V20,V8,V2,V10,V5,V14,V7,V3,V19,V12,V6,V25,V21,
1268 ---> V1,V23,V16,V7,V20,V17,V22,V21,V11,V9,V15,V13,V4,V18,V14,V2,V8,V10,V25,V24,V5,V12,V19,V3,V6,
1277 ---> V1,V13,V24,V22,V25,V21,V15,V19,V5,V3,V6,V16,V9,V11,V4,V18,V17,V8,V12,V14,V2,V23,V10,V7,V20,
1279 ---> V1,V24,V5,V25,V15,V12,V6,V8,V13,V14,V22,V9,V11,V16,V21,V18,V23,V20,V10,V2,V17,V4,V7,V3,V19,
1294 ---> V1,V22,V15,V25,V9,V21,V4,V19,V3,V6,V13,V11,V18,V16,V23,V8,V17,V5,V14,V2,V7,V20,V10,V12,V24,
1305 ---> V1,V13,V25,V17,V23,V24,V14,V2,V20,V8,V10,V16,V21,V22,V11,V12,V5,V19,V3,V6,V7,V15,V4,V9,V18,
1306 ---> V1,V13,V21,V23,V25,V24,V8,V17,V11,V16,V22,V15,V9,V4,V18,V19,V20,V14,V7,V3,V6,V2,V10,V12,V6,
1342 ---> V1,V11,V7,V20,V25,V24,V15,V13,V9,V4,V18,V16,V22,V21,V5,V12,V19,V14,V2,V23,V17,V8,V10,V3,V6,
1347 ---> V1,V13,V24,V25,V23,V8,V20,V17,V14,V2,V10,V19,V12,V5,V3,V6,V7,V21,V15,V9,V4,V18,V16,V11,V22,
1372 ---> V1,V13,V23,V25,V14,V19,V17,V8,V2,V10,V9,V21,V16,V15,V11,V4,V12,V5,V3,V6,V18,V22,V7,V24,V20,
1375 ---> V1,V24,V25,V21,V20,V8,V14,V2,V10,V23,V22,V17,V11,V5,V15,V7,V3,V12,V19,V6,V13,V9,V4,V18,V16,
1396 ---> V1,V13,V23,V14,V12,V21,V25,V22,V4,V15,V24,V17,V16,V8,V7,V9,V2,V10,V19,V5,V3,V6,V11,V18,V20,
1403 ---> V1,V13,V19,V17,V9,V25,V24,V8,V2,V10,V20,V23,V21,V16,V15,V5,V22,V11,V14,V7,V3,V4,V12,V6,V18,
1467 ---> V1,V24,V5,V23,V16,V19,V11,V17,V12,V22,V7,V14,V20,V8,V2,V10,V3,V6,V15,V25,V21,V13,V4,V9,V18,

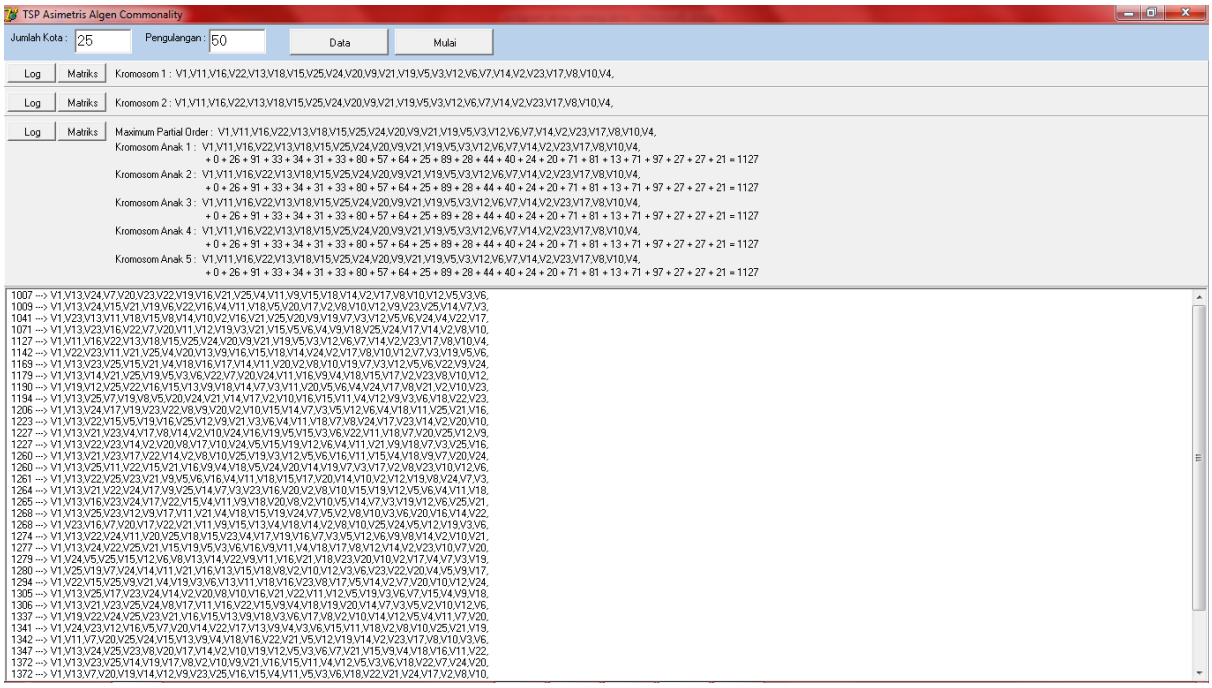

TSP Asimetris Algen Commonality
Jumlah Kota : 25
Pengulangan : 30
Data
Mulai
Log
Matriks
Kromosom 1 : V1,V25,V19,V7,V24,V14,V11,V21,V16,V13,V15,V18,V8,V2,V10,V12,V3,V6,V23,V22,V20,V4,V5,V9,V17,
Kromosom 2 : V1,V25,V19,V7,V24,V14,V11,V21,V16,V13,V15,V18,V8,V2,V10,V12,V3,V6,V23,V22,V20,V4,V5,V9,V17,
Maximum Partial Order : V1,V25,V19,V7,V24,V14,V11,V21,V16,V13,V15,V18,V8,V2,V10,V12,V3,V6,V23,V22,V20,V4,V5,V9,V17,
Kromosom Anak 1 : V1,V25,V19,V7,V24,V14,V11,V21,V16,V13,V15,V18,V8,V2,V10,V12,V3,V6,V23,V22,V20,V4,V5,V9,V17,
+ 0 + 47 + 52 + 28 + 64 + 83 + 43 + 25 + 45 + 94 + 53 + 22 + 77 + 50 + 30 + 29 + 88 + 39 + 54 + 95 + 51 + 62 + 22 + 59 + 68 = 1280
Kromosom Anak 2 : V1,V25,V19,V7,V24,V14,V11,V21,V16,V13,V15,V18,V8,V2,V10,V12,V3,V6,V23,V22,V20,V4,V5,V9,V17,
+ 0 + 47 + 52 + 28 + 64 + 83 + 43 + 25 + 45 + 94 + 53 + 22 + 77 + 50 + 30 + 29 + 88 + 39 + 54 + 95 + 51 + 62 + 22 + 59 + 68 = 1280
Kromosom Anak 3 : V1,V25,V19,V7,V24,V14,V11,V21,V16,V13,V15,V18,V8,V2,V10,V12,V3,V6,V23,V22,V20,V4,V5,V9,V17,
+ 0 + 47 + 52 + 28 + 64 + 83 + 43 + 25 + 45 + 94 + 53 + 22 + 77 + 50 + 30 + 29 + 88 + 39 + 54 + 95 + 51 + 62 + 22 + 59 + 68 = 1280
Kromosom Anak 4 : V1,V25,V19,V7,V24,V14,V11,V21,V16,V13,V15,V18,V8,V2,V10,V12,V3,V6,V23,V22,V20,V4,V5,V9,V17,
+ 0 + 47 + 52 + 28 + 64 + 83 + 43 + 25 + 45 + 94 + 53 + 22 + 77 + 50 + 30 + 29 + 88 + 39 + 54 + 95 + 51 + 62 + 22 + 59 + 68 = 1280
Kromosom Anak 5 : V1,V25,V19,V7,V24,V14,V11,V21,V16,V13,V15,V18,V8,V2,V10,V12,V3,V6,V23,V22,V20,V4,V5,V9,V17,
+ 0 + 47 + 52 + 28 + 64 + 83 + 43 + 25 + 45 + 94 + 53 + 22 + 77 + 50 + 30 + 29 + 88 + 39 + 54 + 95 + 51 + 62 + 22 + 59 + 68 = 1280
Tsp
1007 ---> V1,V13,V24,V7,V20,V23,V22,V19,V16,V21,V25,V4,V11,V9,V15,V18,V14,V2,V17,V8,V10,V12,V5,V3,V6,
OK

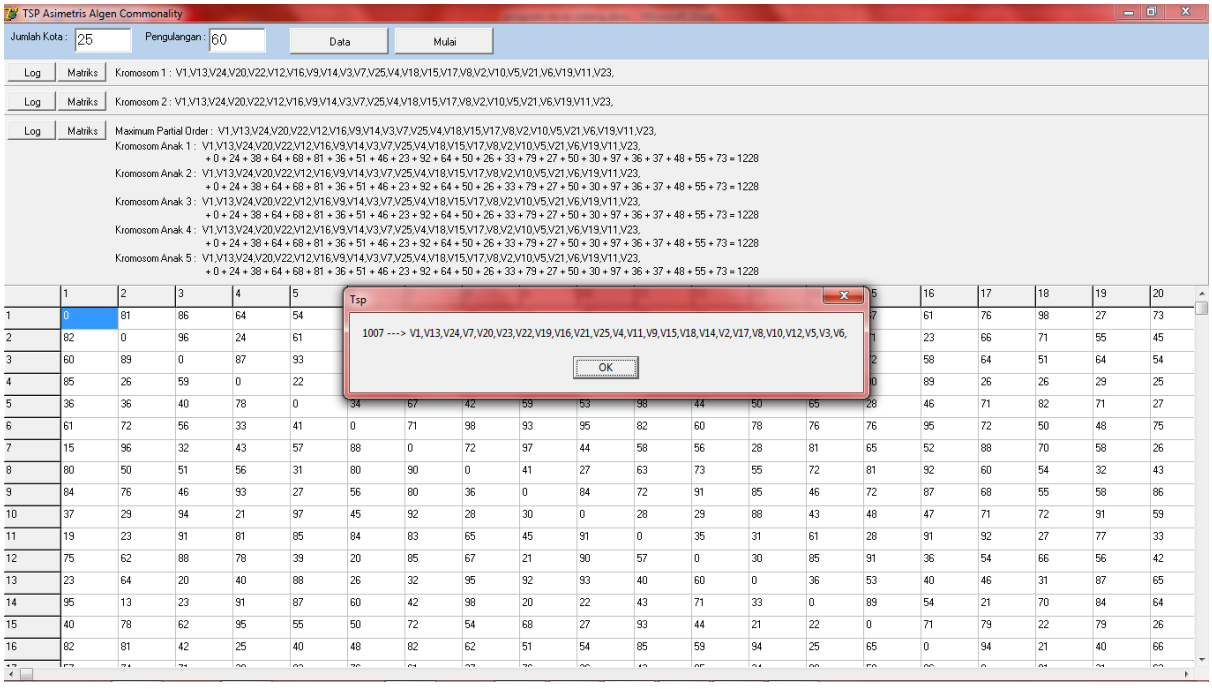

TSP Asimetris Algen Commonality
Jumlah Kota : 25
Pengulangan : 30
Data
Mulai
Log
Matriks
Kromosom 1 : V1,V25,V19,V7,V24,V14,V11,V21,V16,V13,V15,V18,V8,V2,V10,V12,V3,V6,V23,V22,V20,V4,V5,V9,V17,
Log
Matriks
Kromosom 2 : V1,V25,V19,V7,V24,V14,V11,V21,V16,V13,V15,V18,V8,V2,V10,V12,V3,V6,V23,V22,V20,V4,V5,V9,V17,
Log
Matriks
Maximum Partial Order : V1,V25,V19,V7,V24,V14,V11,V21,V16,V13,V15,V18,V8,V2,V10,V12,V3,V6,V23,V22,V20,V4,V5,V9,V17,
Kromosom Anak 1 : V1,V25,V19,V7,V24,V14,V11,V21,V16,V13,V15,V18,V8,V2,V10,V12,V3,V6,V23,V22,V20,V4,V5,V9,V17,
+ 0 + 47 + 52 + 28 + 64 + 83 + 43 + 25 + 45 + 94 + 53 + 22 + 77 + 50 + 30 + 29 + 88 + 39 + 54 + 95 + 51 + 62 + 22 + 59 + 68 = 1280
Kromosom Anak 2 : V1,V25,V19,V7,V24,V14,V11,V21,V16,V13,V15,V18,V8,V2,V10,V12,V3,V6,V23,V22,V20,V4,V5,V9,V17,
+ 0 + 47 + 52 + 28 + 64 + 83 + 43 + 25 + 45 + 94 + 53 + 22 + 77 + 50 + 30 + 29 + 88 + 39 + 54 + 95 + 51 + 62 + 22 + 59 + 68 = 1280
Kromosom Anak 3 : V1,V25,V19,V7,V24,V14,V11,V21,V16,V13,V15,V18,V8,V2,V10,V12,V3,V6,V23,V22,V20,V4,V5,V9,V17,
+ 0 + 47 + 52 + 28 + 64 + 83 + 43 + 25 + 45 + 94 + 53 + 22 + 77 + 50 + 30 + 29 + 88 + 39 + 54 + 95 + 51 + 62 + 22 + 59 + 68 = 1280
Kromosom Anak 4 : V1,V25,V19,V7,V24,V14,V11,V21,V16,V13,V15,V18,V8,V2,V10,V12,V3,V6,V23,V22,V20,V4,V5,V9,V17,
+ 0 + 47 + 52 + 28 + 64 + 83 + 43 + 25 + 45 + 94 + 53 + 22 + 77 + 50 + 30 + 29 + 88 + 39 + 54 + 95 + 51 + 62 + 22 + 59 + 68 = 1280
Kromosom Anak 5 : V1,V25,V19,V7,V24,V14,V11,V21,V16,V13,V15,V18,V8,V2,V10,V12,V3,V6,V23,V22,V20,V4,V5,V9,V17,
+ 0 + 47 + 52 + 28 + 64 + 83 + 43 + 25 + 45 + 94 + 53 + 22 + 77 + 50 + 30 + 29 + 88 + 39 + 54 + 95 + 51 + 62 + 22 + 59 + 68 = 1280
1007 ---> V1,V13,V24,V7,V20,V23,V22,V19,V16,V21,V25,V4,V11,V9,V15,V18,V14,V2,V17,V8,V10,V12,V5,V3,V6,
1009 ---> V1,V13,V24,V15,V21,V19,V6,V22,V16,V4,V11,V18,V5,V20,V17,V2,V8,V10,V12,V9,V23,V25,V14,V7,V3,
1041 ---> V1,V23,V13,V11,V18,V15,V8,V14,V10,V2,V16,V21,V25,V20,V9,V19,V7,V3,V12,V5,V6,V24,V4,V22,V17,
1142 ---> V1,V22,V23,V11,V21,V25,V4,V20,V13,V9,V16,V15,V18,V14,V24,V2,V17,V8,V10,V12,V7,V3,V19,V5,V6,
1169 ---> V1,V13,V23,V25,V15,V21,V4,V18,V16,V17,V14,V11,V20,V2,V8,V10,V19,V7,V3,V12,V5,V6,V22,V9,V24,
1178 ---> V1,V13,V14,V21,V25,V19,V5,V3,V6,V22,V7,V20,V24,V11,V16,V9,V4,V18,V15,V17,V2,V23,V8,V10,V12,
1223 ---> V1,V13,V22,V15,V5,V19,V16,V25,V12,V9,V21,V3,V6,V4,V11,V18,V7,V8,V24,V17,V23,V14,V2,V20,V10,
1227 ---> V1,V13,V21,V23,V4,V17,V8,V14,V2,V10,V24,V16,V19,V5,V15,V3,V6,V22,V11,V18,V7,V20,V25,V12,V9,
1227 ---> V1,V13,V22,V23,V14,V2,V20,V8,V17,V10,V24,V5,V15,V19,V12,V6,V4,V11,V21,V9,V18,V7,V3,V25,V16,
1260 ---> V1,V13,V21,V23,V17,V22,V14,V2,V8,V10,V25,V19,V3,V12,V5,V6,V16,V11,V15,V4,V18,V9,V7,V20,V24,
1260 ---> V1,V13,V25,V11,V22,V15,V21,V16,V9,V4,V18,V5,V24,V20,V14,V19,V7,V3,V17,V2,V8,V23,V10,V12,V6,
1261 ---> V1,V13,V22,V25,V23,V21,V9,V5,V6,V16,V4,V11,V18,V15,V17,V20,V14,V10,V2,V12,V19,V8,V24,V7,V3,
1264 ---> V1,V13,V21,V22,V24,V17,V9,V25,V14,V7,V3,V23,V16,V20,V2,V8,V10,V15,V19,V12,V5,V6,V4,V11,V18,
1265 ---> V1,V13,V16,V23,V24,V17,V22,V15,V4,V11,V9,V18,V20,V8,V2,V10,V5,V14,V7,V3,V19,V12,V6,V25,V21,
1268 ---> V1,V13,V25,V23,V12,V9,V17,V11,V21,V4,V18,V15,V19,V24,V7,V5,V2,V8,V10,V3,V6,V20,V16,V14,V22,
1268 ---> V1,V23,V16,V7,V20,V17,V22,V21,V11,V9,V15,V13,V4,V18,V14,V2,V8,V10,V25,V24,V5,V12,V19,V3,V6,
1277 ---> V1,V13,V24,V22,V25,V21,V15,V19,V5,V3,V6,V16,V9,V11,V4,V18,V17,V8,V12,V14,V2,V23,V10,V7,V20,
1279 ---> V1,V24,V5,V25,V15,V12,V6,V8,V13,V14,V22,V9,V11,V16,V21,V18,V23,V20,V10,V2,V17,V4,V7,V3,V19,
1280 ---> V1,V25,V19,V7,V24,V14,V11,V21,V16,V13,V15,V18,V8,V2,V10,V12,V3,V6,V23,V22,V20,V4,V5,V9,V17,
1294 ---> V1,V22,V15,V25,V9,V21,V4,V19,V3,V6,V13,V11,V18,V16,V23,V8,V17,V5,V14,V2,V7,V20,V10,V12,V24,
1305 ---> V1,V13,V25,V17,V23,V24,V14,V2,V20,V8,V10,V16,V21,V22,V11,V12,V5,V19,V3,V6,V7,V15,V4,V9,V18,
1306 ---> V1,V13,V21,V23,V25,V24,V8,V17,V11,V16,V22,V15,V9,V4,V18,V19,V20,V14,V7,V3,V5,V2,V10,V12,V6,
1342 ---> V1,V11,V7,V20,V25,V24,V15,V13,V9,V4,V18,V16,V22,V21,V5,V12,V19,V14,V2,V23,V17,V8,V10,V3,V6,
1347 ---> V1,V13,V24,V25,V23,V8,V20,V17,V14,V2,V10,V19,V12,V5,V3,V6,V7,V21,V15,V9,V4,V18,V16,V11,V22,
1372 ---> V1,V13,V23,V25,V14,V19,V17,V8,V2,V10,V9,V21,V16,V15,V11,V4,V12,V5,V3,V6,V18,V22,V7,V24,V20,
1375 ---> V1,V24,V25,V21,V20,V8,V14,V2,V10,V23,V22,V17,V11,V5,V15,V7,V3,V12,V19,V6,V13,V9,V4,V18,V16,
1396 ---> V1,V13,V23,V14,V12,V21,V25,V22,V4,V15,V24,V17,V16,V8,V7,V9,V2,V10,V19,V5,V3,V6,V11,V18,V20,
1403 ---> V1,V13,V19,V17,V9,V25,V24,V8,V2,V10,V20,V23,V21,V16,V15,V5,V22,V11,V14,V7,V3,V4,V12,V6,V18,
1467 ---> V1,V24,V5,V23,V16,V19,V11,V17,V12,V22,V7,V14,V20,V8,V2,V10,V3,V6,V15,V25,V21,V13,V4,V9,V18,
1497 ---> V1,V22,V15,V21,V7,V3,V25,V4,V13,V12,V9,V16,V11,V6,V18,V14,V5,V24,V23,V20,V17,V2,V8,V10,V19,

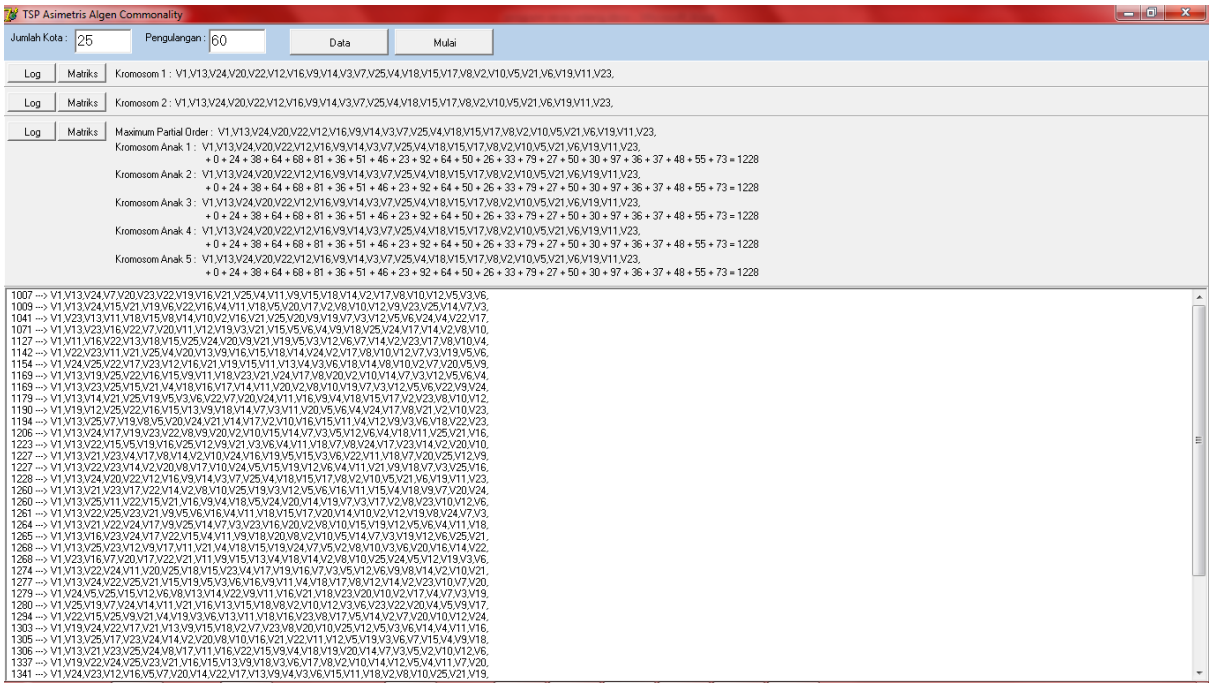

TSP Asimetris Algen Commonality
Jumlah Kota : 25
Pengulangan : 40
Data
Mulai
Log
Matriks
Kromosom 1 : V1,V13,V25,V7,V19,V8,V5,V20,V24,V21,V14,V17,V2,V10,V16,V15,V11,V4,V12,V9,V3,V6,V18,V22,V23,
Log
Matriks
Kromosom 2 : V1,V13,V25,V7,V19,V8,V5,V20,V24,V21,V14,V17,V2,V10,V16,V15,V11,V4,V12,V9,V3,V6,V18,V22,V23,
Log
Matriks
Maximum Partial Order : V1,V13,V25,V7,V19,V8,V5,V20,V24,V21,V14,V17,V2,V10,V16,V15,V11,V4,V12,V9,V3,V6,V18,V22,V23,
Kromosom Anak 1 : V1,V13,V25,V7,V19,V8,V5,V20,V24,V21,V14,V17,V2,V10,V16,V15,V11,V4,V12,V9,V3,V6,V18,V22,V23,
+ 0 + 24 + 48 + 93 + 58 + 68 + 31 + 27 + 59 + 29 + 43 + 21 + 74 + 30 + 47 + 65 + 93 + 81 + 45 + 21 + 46 + 39 + 50 + 70 + 32 = 1194
Kromosom Anak 2 : V1,V13,V25,V7,V19,V8,V5,V20,V24,V21,V14,V17,V2,V10,V16,V15,V11,V4,V12,V9,V3,V6,V18,V22,V23,
+ 0 + 24 + 48 + 93 + 58 + 68 + 31 + 27 + 59 + 29 + 43 + 21 + 74 + 30 + 47 + 65 + 93 + 81 + 45 + 21 + 46 + 39 + 50 + 70 + 32 = 1194
Kromosom Anak 3 : V1,V13,V25,V7,V19,V8,V5,V20,V24,V21,V14,V17,V2,V10,V16,V15,V11,V4,V12,V9,V3,V6,V18,V22,V23,
+ 0 + 24 + 48 + 93 + 58 + 68 + 31 + 27 + 59 + 29 + 43 + 21 + 74 + 30 + 47 + 65 + 93 + 81 + 45 + 21 + 46 + 39 + 50 + 70 + 32 = 1194
Kromosom Anak 4 : V1,V13,V25,V7,V19,V8,V5,V20,V24,V21,V14,V17,V2,V10,V16,V15,V11,V4,V12,V9,V3,V6,V18,V22,V23,
+ 0 + 24 + 48 + 93 + 58 + 68 + 31 + 27 + 59 + 29 + 43 + 21 + 74 + 30 + 47 + 65 + 93 + 81 + 45 + 21 + 46 + 39 + 50 + 70 + 32 = 1194
Kromosom Anak 5 : V1,V13,V25,V7,V19,V8,V5,V20,V24,V21,V14,V17,V2,V10,V16,V15,V11,V4,V12,V9,V3,V6,V18,V22,V23,
+ 0 + 24 + 48 + 93 + 58 + 68 + 31 + 27 + 59 + 29 + 43 + 21 + 74 + 30 + 47 + 65 + 93 + 81 + 45 + 21 + 46 + 39 + 50 + 70 + 32 = 1194
Tsp
1007 ---> V1,V13,V24,V7,V20,V23,V22,V19,V16,V21,V25,V4,V11,V9,V15,V18,V14,V2,V17,V8,V10,V12,V5,V3,V6,
OK

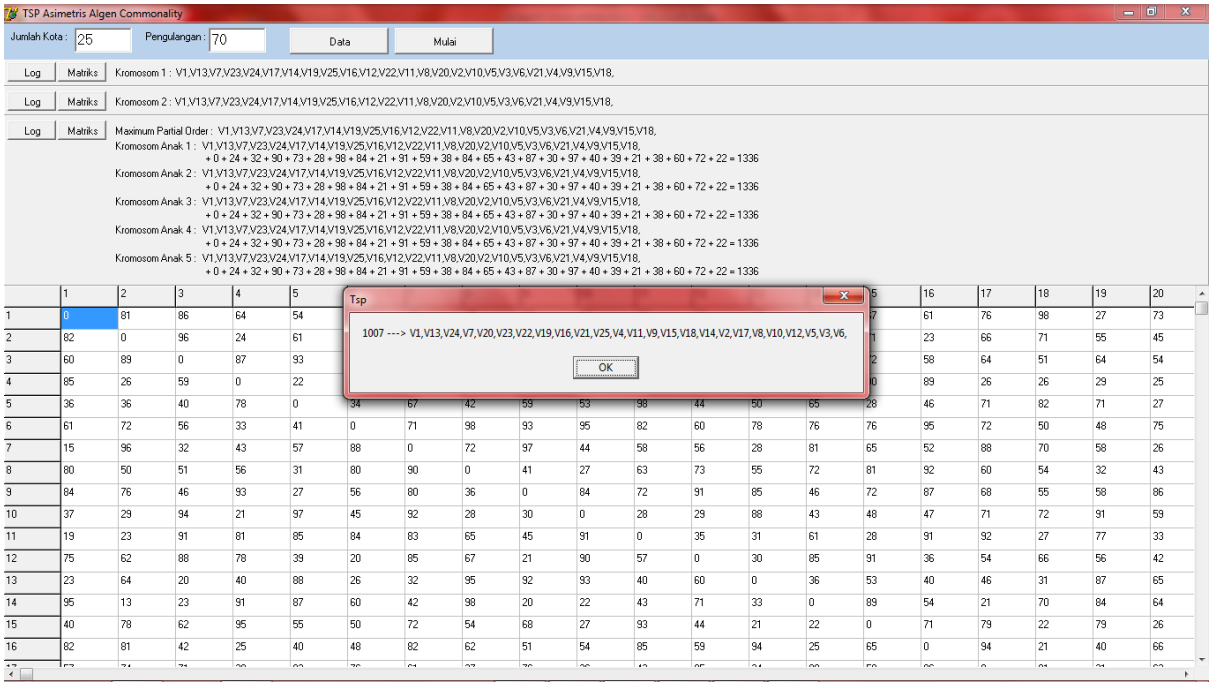

TSP Asimetris Algen Commonality
Jumlah Kota : 25
Pengulangan : 40
Data
Mulai
Log
Matriks
Kromosom 1 : V1,V13,V25,V7,V19,V8,V5,V20,V24,V21,V14,V17,V2,V10,V16,V15,V11,V4,V12,V9,V3,V6,V18,V22,V23,
Log
Matriks
Kromosom 2 : V1,V13,V25,V7,V19,V8,V5,V20,V24,V21,V14,V17,V2,V10,V16,V15,V11,V4,V12,V9,V3,V6,V18,V22,V23,
Log
Matriks
Maximum Partial Order : V1,V13,V25,V7,V19,V8,V5,V20,V24,V21,V14,V17,V2,V10,V16,V15,V11,V4,V12,V9,V3,V6,V18,V22,V23,
Kromosom Anak 1 : V1,V13,V25,V7,V19,V8,V5,V20,V24,V21,V14,V17,V2,V10,V16,V15,V11,V4,V12,V9,V3,V6,V18,V22,V23,
+ 0 + 24 + 48 + 93 + 58 + 68 + 31 + 27 + 59 + 29 + 43 + 21 + 74 + 30 + 47 + 65 + 93 + 81 + 45 + 21 + 46 + 39 + 50 + 70 + 32 = 1194
Kromosom Anak 2 : V1,V13,V25,V7,V19,V8,V5,V20,V24,V21,V14,V17,V2,V10,V16,V15,V11,V4,V12,V9,V3,V6,V18,V22,V23,
+ 0 + 24 + 48 + 93 + 58 + 68 + 31 + 27 + 59 + 29 + 43 + 21 + 74 + 30 + 47 + 65 + 93 + 81 + 45 + 21 + 46 + 39 + 50 + 70 + 32 = 1194
Kromosom Anak 3 : V1,V13,V25,V7,V19,V8,V5,V20,V24,V21,V14,V17,V2,V10,V16,V15,V11,V4,V12,V9,V3,V6,V18,V22,V23,
+ 0 + 24 + 48 + 93 + 58 + 68 + 31 + 27 + 59 + 29 + 43 + 21 + 74 + 30 + 47 + 65 + 93 + 81 + 45 + 21 + 46 + 39 + 50 + 70 + 32 = 1194
Kromosom Anak 4 : V1,V13,V25,V7,V19,V8,V5,V20,V24,V21,V14,V17,V2,V10,V16,V15,V11,V4,V12,V9,V3,V6,V18,V22,V23,
+ 0 + 24 + 48 + 93 + 58 + 68 + 31 + 27 + 59 + 29 + 43 + 21 + 74 + 30 + 47 + 65 + 93 + 81 + 45 + 21 + 46 + 39 + 50 + 70 + 32 = 1194
Kromosom Anak 5 : V1,V13,V25,V7,V19,V8,V5,V20,V24,V21,V14,V17,V2,V10,V16,V15,V11,V4,V12,V9,V3,V6,V18,V22,V23,
+ 0 + 24 + 48 + 93 + 58 + 68 + 31 + 27 + 59 + 29 + 43 + 21 + 74 + 30 + 47 + 65 + 93 + 81 + 45 + 21 + 46 + 39 + 50 + 70 + 32 = 1194
1007 ---> V1,V13,V24,V7,V20,V23,V22,V19,V16,V21,V25,V4,V11,V9,V15,V18,V14,V2,V17,V8,V10,V12,V5,V3,V6,
1009 ---> V1,V13,V24,V15,V21,V19,V6,V22,V16,V4,V11,V18,V5,V20,V17,V2,V8,V10,V12,V9,V23,V25,V14,V7,V3,
1041 ---> V1,V23,V13,V11,V18,V15,V8,V14,V10,V2,V16,V21,V25,V20,V9,V19,V7,V3,V12,V5,V6,V24,V4,V22,V17,
1071 ---> V1,V13,V23,V16,V22,V7,V20,V11,V12,V19,V3,V21,V15,V5,V6,V4,V9,V18,V25,V24,V17,V14,V2,V8,V10,
1142 ---> V1,V22,V23,V11,V21,V25,V4,V20,V13,V9,V16,V15,V18,V14,V24,V2,V17,V8,V10,V12,V7,V3,V19,V5,V6,
1169 ---> V1,V13,V23,V25,V15,V21,V4,V18,V16,V17,V14,V11,V20,V2,V9,V10,V19,V7,V3,V12,V5,V6,V22,V9,V24,
1179 ---> V1,V13,V14,V21,V25,V19,V5,V3,V6,V22,V7,V20,V24,V11,V16,V9,V4,V18,V15,V17,V2,V23,V8,V10,V12,
1194 ---> V1,V13,V25,V7,V19,V8,V5,V20,V24,V21,V14,V17,V2,V10,V16,V15,V11,V4,V12,V9,V3,V6,V18,V22,V23,
1206 ---> V1,V13,V24,V17,V19,V23,V22,V8,V9,V20,V2,V10,V15,V14,V7,V3,V5,V12,V6,V4,V18,V11,V25,V21,V16,
1223 ---> V1,V13,V22,V15,V5,V19,V16,V25,V12,V9,V21,V3,V6,V4,V11,V18,V7,V8,V24,V17,V23,V14,V2,V20,V10,
1227 ---> V1,V13,V21,V23,V4,V17,V8,V14,V2,V10,V24,V16,V19,V5,V15,V3,V6,V22,V11,V18,V7,V20,V25,V12,V9,
1227 ---> V1,V13,V22,V23,V14,V2,V20,V8,V17,V10,V24,V5,V15,V19,V12,V6,V4,V11,V21,V9,V18,V7,V3,V25,V16,
1260 ---> V1,V13,V21,V23,V17,V22,V14,V2,V8,V10,V25,V19,V3,V12,V5,V6,V16,V11,V15,V4,V18,V9,V7,V20,V24,
1260 ---> V1,V13,V25,V11,V22,V15,V21,V16,V9,V4,V18,V5,V24,V20,V14,V19,V7,V3,V17,V2,V8,V23,V10,V12,V6,
1261 ---> V1,V13,V22,V25,V23,V21,V9,V5,V6,V16,V4,V11,V18,V15,V17,V20,V14,V10,V2,V12,V19,V8,V24,V7,V3,
1264 ---> V1,V13,V21,V22,V24,V17,V9,V25,V14,V7,V3,V23,V16,V20,V2,V8,V10,V15,V19,V12,V5,V6,V4,V11,V18,
1265 ---> V1,V13,V16,V23,V24,V17,V22,V15,V4,V11,V9,V18,V20,V8,V2,V10,V5,V14,V7,V3,V19,V12,V6,V25,V21,
1268 ---> V1,V13,V25,V23,V12,V9,V17,V11,V21,V4,V18,V15,V19,V24,V7,V5,V2,V8,V10,V3,V6,V20,V16,V14,V22,
1268 ---> V1,V23,V16,V7,V20,V17,V22,V21,V11,V9,V15,V13,V4,V18,V14,V2,V8,V10,V25,V24,V5,V12,V19,V3,V6,
1277 ---> V1,V13,V24,V22,V25,V21,V15,V19,V5,V3,V6,V16,V9,V11,V4,V18,V17,V8,V12,V14,V2,V23,V10,V7,V20,
1279 ---> V1,V24,V5,V25,V15,V12,V6,V8,V13,V14,V22,V9,V11,V16,V21,V18,V23,V20,V10,V2,V17,V4,V7,V3,V19,
1280 ---> V1,V25,V19,V7,V24,V14,V11,V21,V16,V13,V15,V18,V8,V2,V10,V12,V3,V6,V23,V22,V20,V4,V5,V9,V17,
1294 ---> V1,V22,V15,V25,V9,V21,V4,V19,V3,V6,V13,V11,V18,V16,V23,V8,V17,V5,V14,V2,V7,V20,V10,V12,V24,
1305 ---> V1,V13,V25,V17,V23,V24,V14,V2,V20,V8,V10,V16,V21,V22,V11,V12,V5,V19,V3,V6,V7,V15,V4,V9,V18,
1306 ---> V1,V13,V21,V23,V25,V24,V8,V17,V11,V16,V22,V15,V9,V4,V18,V19,V20,V14,V7,V3,V5,V2,V10,V12,V6,
1337 ---> V1,V19,V22,V24,V25,V23,V21,V16,V15,V13,V9,V18,V3,V6,V17,V8,V2,V10,V14,V12,V5,V4,V11,V7,V20,
1342 ---> V1,V11,V7,V20,V25,V24,V15,V13,V9,V4,V18,V16,V22,V21,V5,V12,V19,V14,V2,V23,V17,V8,V10,V3,V6,
1347 ---> V1,V13,V24,V25,V23,V8,V20,V17,V14,V2,V10,V19,V12,V5,V3,V6,V7,V21,V15,V9,V4,V18,V16,V11,V22,
1372 ---> V1,V13,V23,V25,V14,V19,V17,V8,V2,V10,V9,V21,V16,V15,V11,V4,V12,V5,V3,V6,V18,V22,V7,V24,V20,
1372 ---> V1,V13,V7,V20,V19,V14,V12,V9,V23,V25,V16,V15,V4,V11,V5,V3,V6,V18,V22,V21,V24,V17,V2,V8,V10,
1375 ---> V1,V24,V25,V21,V20,V8,V14,V2,V10,V23,V22,V17,V11,V5,V15,V7,V3,V12,V19,V6,V13,V9,V4,V18,V16,
1396 ---> V1,V13,V23,V14,V12,V21,V25,V22,V4,V15,V24,V17,V16,V8,V7,V9,V2,V10,V19,V5,V3,V6,V11,V18,V20,
1403 ---> V1,V13,V19,V17,V9,V25,V24,V8,V2,V10,V20,V23,V21,V16,V15,V5,V22,V11,V14,V7,V3,V4,V12,V6,V18,
1467 ---> V1,V24,V5,V23,V16,V19,V11,V17,V12,V22,V7,V14,V20,V8,V2,V10,V3,V6,V15,V25,V21,V13,V4,V9,V18,

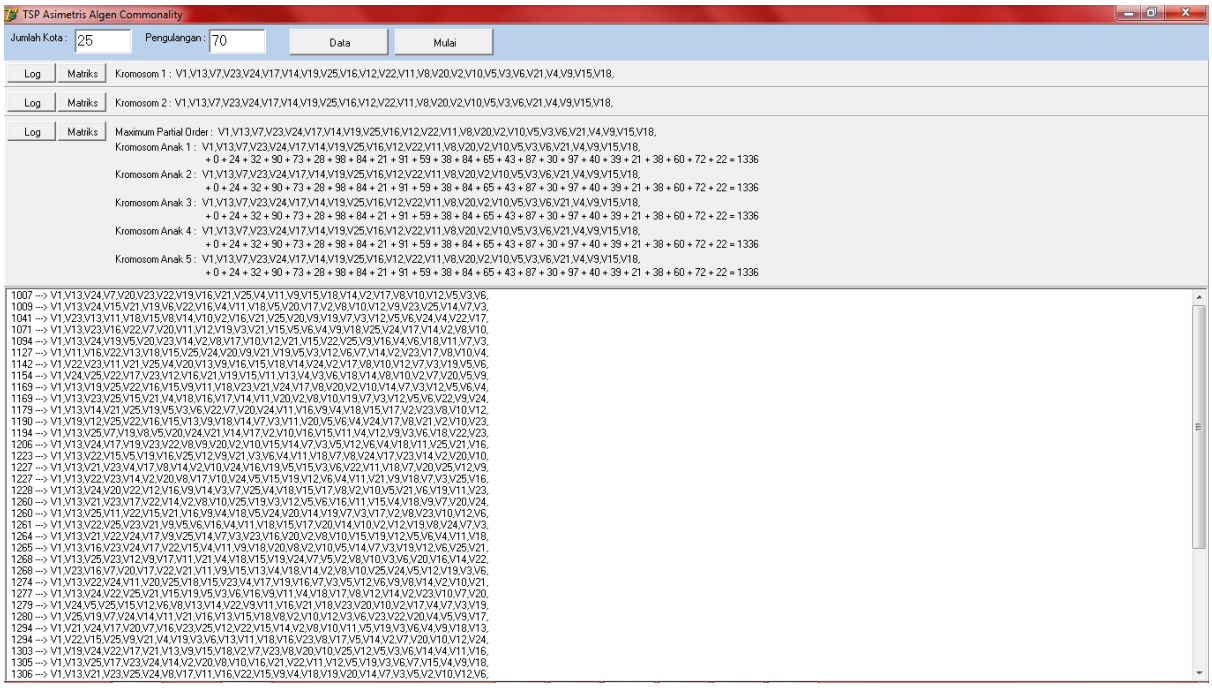

TSP Asimetris Algen Commonality
Jumlah Kota : 25
Pengulangan : 50
Data
Mulai
Log
Matriks
Kromosom 1 : V1,V11,V16,V22,V13,V18,V15,V25,V24,V20,V9,V21,V19,V5,V3,V12,V6,V7,V14,V2,V23,V17,V8,V10,V4,
Log
Matriks
Kromosom 2 : V1,V11,V16,V22,V13,V18,V15,V25,V24,V20,V9,V21,V19,V5,V3,V12,V6,V7,V14,V2,V23,V17,V8,V10,V4,
Log
Matriks
Maximum Partial Order : V1,V11,V16,V22,V13,V18,V15,V25,V24,V20,V9,V21,V19,V5,V3,V12,V6,V7,V14,V2,V23,V17,V8,V10,V4,
Kromosom Anak 1 : V1,V11,V16,V22,V13,V18,V15,V25,V24,V20,V9,V21,V19,V5,V3,V12,V6,V7,V14,V2,V23,V17,V8,V10,V4,
+ 0 + 26 + 91 + 33 + 34 + 31 + 33 + 80 + 57 + 64 + 25 + 89 + 28 + 44 + 40 + 24 + 20 + 71 + 81 + 13 + 71 + 97 + 27 + 27 + 21 = 1127
Kromosom Anak 2 : V1,V11,V16,V22,V13,V18,V15,V25,V24,V20,V9,V21,V19,V5,V3,V12,V6,V7,V14,V2,V23,V17,V8,V10,V4,
+ 0 + 26 + 91 + 33 + 34 + 31 + 33 + 80 + 57 + 64 + 25 + 89 + 28 + 44 + 40 + 24 + 20 + 71 + 81 + 13 + 71 + 97 + 27 + 27 + 21 = 1127
Kromosom Anak 3 : V1,V11,V16,V22,V13,V18,V15,V25,V24,V20,V9,V21,V19,V5,V3,V12,V6,V7,V14,V2,V23,V17,V8,V10,V4,
+ 0 + 26 + 91 + 33 + 34 + 31 + 33 + 80 + 57 + 64 + 25 + 89 + 28 + 44 + 40 + 24 + 20 + 71 + 81 + 13 + 71 + 97 + 27 + 27 + 21 = 1127
Kromosom Anak 4 : V1,V11,V16,V22,V13,V18,V15,V25,V24,V20,V9,V21,V19,V5,V3,V12,V6,V7,V14,V2,V23,V17,V8,V10,V4,
+ 0 + 26 + 91 + 33 + 34 + 31 + 33 + 80 + 57 + 64 + 25 + 89 + 28 + 44 + 40 + 24 + 20 + 71 + 81 + 13 + 71 + 97 + 27 + 27 + 21 = 1127
Kromosom Anak 5 : V1,V11,V16,V22,V13,V18,V15,V25,V24,V20,V9,V21,V19,V5,V3,V12,V6,V7,V14,V2,V23,V17,V8,V10,V4,
+ 0 + 26 + 91 + 33 + 34 + 31 + 33 + 80 + 57 + 64 + 25 + 89 + 28 + 44 + 40 + 24 + 20 + 71 + 81 + 13 + 71 + 97 + 27 + 27 + 21 = 1127
Tsp
1007 ---> V1,V13,V24,V7,V20,V23,V22,V19,V16,V21,V25,V4,V11,V9,V15,V18,V14,V2,V17,V8,V10,V12,V5,V3,V6,
OK

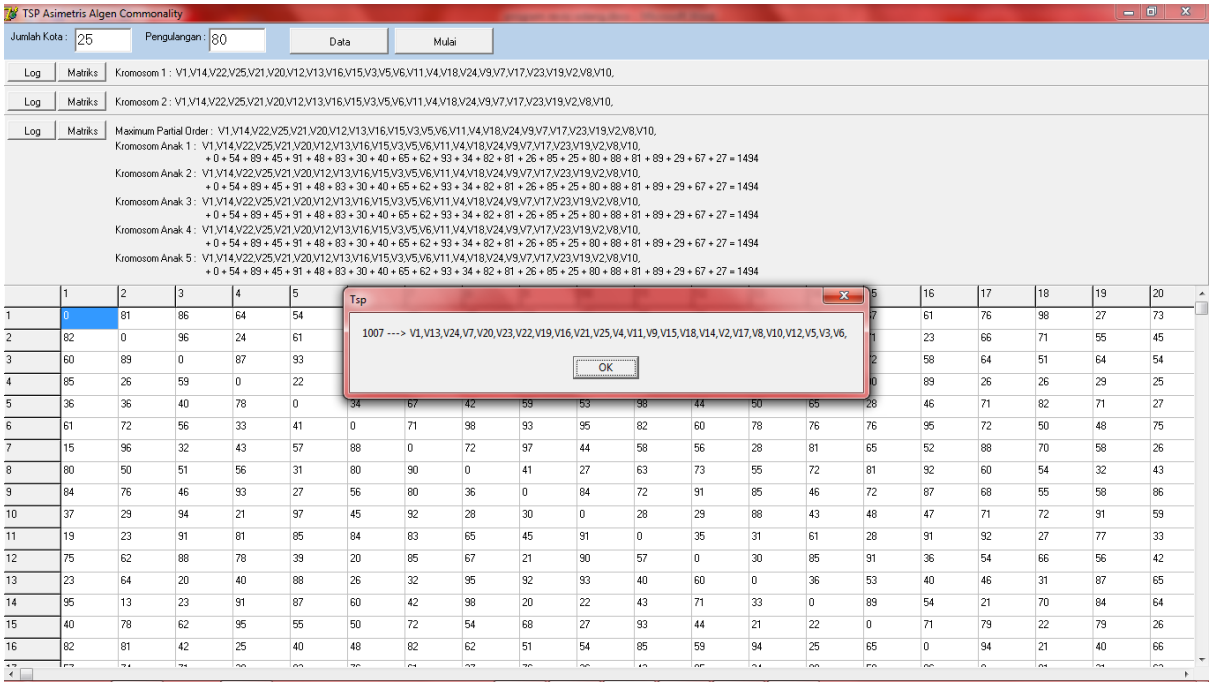

TSP Asimetris Algen Commonality
Jumlah Kota : 25
Pengulangan : 50
Data
Mulai
Log
Matriks
Kromosom 1 : V1,V11,V16,V22,V13,V18,V15,V25,V24,V20,V9,V21,V19,V5,V3,V12,V6,V7,V14,V2,V23,V17,V8,V10,V4,
Kromosom 2 : V1,V11,V16,V22,V13,V18,V15,V25,V24,V20,V9,V21,V19,V5,V3,V12,V6,V7,V14,V2,V23,V17,V8,V10,V4,
Maximum Partial Order : V1,V11,V16,V22,V13,V18,V15,V25,V24,V20,V9,V21,V19,V5,V3,V12,V6,V7,V14,V2,V23,V17,V8,V10,V4,
Kromosom Anak 1 : V1,V11,V16,V22,V13,V18,V15,V25,V24,V20,V9,V21,V19,V5,V3,V12,V6,V7,V14,V2,V23,V17,V8,V10,V4,
+ 0 + 26 + 91 + 33 + 34 + 31 + 33 + 80 + 57 + 64 + 25 + 89 + 28 + 44 + 40 + 24 + 20 + 71 + 81 + 13 + 71 + 97 + 27 + 27 + 21 = 1127
Kromosom Anak 2 : V1,V11,V16,V22,V13,V18,V15,V25,V24,V20,V9,V21,V19,V5,V3,V12,V6,V7,V14,V2,V23,V17,V8,V10,V4,
+ 0 + 26 + 91 + 33 + 34 + 31 + 33 + 80 + 57 + 64 + 25 + 89 + 28 + 44 + 40 + 24 + 20 + 71 + 81 + 13 + 71 + 97 + 27 + 27 + 21 = 1127
Kromosom Anak 3 : V1,V11,V16,V22,V13,V18,V15,V25,V24,V20,V9,V21,V19,V5,V3,V12,V6,V7,V14,V2,V23,V17,V8,V10,V4,
+ 0 + 26 + 91 + 33 + 34 + 31 + 33 + 80 + 57 + 64 + 25 + 89 + 28 + 44 + 40 + 24 + 20 + 71 + 81 + 13 + 71 + 97 + 27 + 27 + 21 = 1127
Kromosom Anak 4 : V1,V11,V16,V22,V13,V18,V15,V25,V24,V20,V9,V21,V19,V5,V3,V12,V6,V7,V14,V2,V23,V17,V8,V10,V4,
+ 0 + 26 + 91 + 33 + 34 + 31 + 33 + 80 + 57 + 64 + 25 + 89 + 28 + 44 + 40 + 24 + 20 + 71 + 81 + 13 + 71 + 97 + 27 + 27 + 21 = 1127
Kromosom Anak 5 : V1,V11,V16,V22,V13,V18,V15,V25,V24,V20,V9,V21,V19,V5,V3,V12,V6,V7,V14,V2,V23,V17,V8,V10,V4,
+ 0 + 26 + 91 + 33 + 34 + 31 + 33 + 80 + 57 + 64 + 25 + 89 + 28 + 44 + 40 + 24 + 20 + 71 + 81 + 13 + 71 + 97 + 27 + 27 + 21 = 1127
1007 ---> V1,V13,V24,V7,V20,V23,V22,V19,V16,V21,V25,V4,V11,V9,V15,V18,V14,V2,V17,V8,V10,V12,V5,V3,V6,
1009 ---> V1,V13,V24,V15,V21,V19,V6,V22,V16,V4,V11,V18,V5,V20,V17,V2,V8,V10,V12,V9,V23,V25,V14,V7,V3,
1041 ---> V1,V23,V13,V11,V18,V15,V8,V14,V10,V2,V16,V21,V25,V20,V9,V19,V7,V3,V12,V5,V6,V24,V4,V22,V17,
1071 ---> V1,V13,V23,V16,V22,V7,V20,V11,V12,V19,V3,V21,V15,V5,V6,V4,V9,V18,V25,V24,V17,V14,V2,V8,V10,
1127 ---> V1,V11,V16,V22,V13,V18,V15,V25,V24,V20,V9,V21,V19,V5,V3,V12,V6,V7,V14,V2,V23,V17,V8,V10,V4,
1142 ---> V1,V22,V23,V11,V21,V25,V4,V20,V13,V9,V16,V15,V18,V14,V24,V2,V17,V8,V10,V12,V7,V3,V19,V5,V6,
1168 ---> V1,V13,V23,V25,V15,V21,V4,V18,V16,V17,V14,V11,V20,V2,V8,V10,V19,V7,V3,V12,V5,V6,V22,V9,V24,
1179 ---> V1,V13,V14,V21,V25,V19,V5,V3,V6,V22,V7,V20,V24,V11,V16,V9,V4,V18,V15,V17,V2,V23,V8,V10,V12,
1190 ---> V1,V19,V12,V25,V22,V16,V15,V13,V9,V18,V14,V7,V3,V11,V20,V5,V6,V4,V24,V17,V8,V21,V2,V10,V23,
1194 ---> V1,V13,V25,V7,V19,V8,V5,V20,V24,V21,V14,V17,V2,V10,V16,V15,V11,V4,V12,V9,V3,V6,V18,V22,V23,
1206 ---> V1,V13,V24,V17,V19,V23,V22,V8,V9,V20,V2,V10,V15,V14,V7,V3,V5,V12,V6,V4,V18,V11,V25,V21,V16,
1223 ---> V1,V13,V22,V15,V5,V19,V16,V25,V12,V9,V21,V3,V6,V4,V11,V18,V7,V8,V24,V17,V23,V14,V2,V20,V10,
1227 ---> V1,V13,V21,V23,V4,V17,V8,V14,V2,V10,V24,V16,V19,V5,V15,V3,V6,V22,V11,V18,V7,V20,V25,V12,V9,
1227 ---> V1,V13,V22,V23,V14,V2,V20,V8,V17,V10,V24,V5,V15,V19,V12,V6,V4,V11,V21,V9,V18,V7,V3,V25,V16,
1260 ---> V1,V13,V21,V23,V17,V22,V14,V2,V8,V10,V25,V19,V3,V12,V5,V6,V16,V11,V15,V4,V18,V9,V7,V20,V24,
1260 ---> V1,V13,V25,V11,V22,V15,V21,V16,V9,V4,V18,V5,V24,V20,V14,V19,V7,V3,V17,V2,V8,V23,V10,V12,V6,
1261 ---> V1,V13,V22,V25,V23,V21,V9,V5,V6,V16,V4,V11,V18,V15,V17,V20,V14,V10,V2,V12,V19,V8,V24,V7,V3,
1264 ---> V1,V13,V21,V22,V24,V17,V9,V25,V14,V7,V3,V23,V16,V20,V2,V8,V10,V15,V19,V12,V5,V6,V4,V11,V18,
1265 ---> V1,V13,V16,V23,V24,V17,V22,V15,V4,V11,V9,V18,V20,V8,V2,V10,V5,V14,V7,V3,V19,V12,V6,V25,V21,
1268 ---> V1,V13,V25,V23,V12,V9,V17,V11,V21,V4,V18,V15,V19,V24,V7,V5,V2,V8,V10,V3,V6,V20,V16,V14,V22,
1268 ---> V1,V23,V16,V7,V20,V17,V22,V21,V11,V9,V15,V13,V4,V18,V14,V2,V8,V10,V25,V24,V5,V12,V19,V3,V6,
1274 ---> V1,V13,V22,V24,V11,V20,V25,V18,V15,V23,V4,V17,V19,V16,V7,V3,V5,V12,V6,V9,V8,V14,V2,V10,V21,
1277 ---> V1,V13,V24,V22,V25,V21,V15,V19,V5,V3,V6,V16,V9,V11,V4,V18,V17,V8,V12,V14,V2,V23,V10,V7,V20,
1279 ---> V1,V24,V5,V25,V15,V12,V6,V8,V13,V14,V22,V9,V11,V16,V21,V18,V23,V20,V10,V2,V17,V4,V7,V3,V19,
1280 ---> V1,V25,V19,V7,V24,V14,V11,V21,V16,V13,V15,V18,V8,V2,V10,V12,V3,V6,V23,V22,V20,V4,V5,V9,V17,
1294 ---> V1,V22,V15,V25,V9,V21,V4,V19,V3,V6,V13,V11,V18,V16,V23,V8,V17,V5,V14,V2,V7,V20,V10,V12,V24,
1305 ---> V1,V13,V25,V17,V23,V24,V14,V2,V20,V8,V10,V16,V21,V22,V11,V12,V5,V19,V3,V6,V7,V15,V4,V9,V18,
1306 ---> V1,V13,V21,V23,V25,V24,V8,V17,V11,V16,V22,V15,V9,V4,V18,V19,V20,V14,V7,V3,V5,V2,V10,V12,V6,
1337 ---> V1,V19,V22,V24,V25,V23,V21,V16,V15,V13,V9,V18,V3,V6,V17,V8,V2,V10,V14,V12,V5,V4,V11,V7,V20,
1341 ---> V1,V24,V23,V12,V16,V5,V7,V20,V14,V22,V17,V13,V9,V4,V3,V6,V15,V11,V18,V2,V8,V10,V25,V21,V19,
1342 ---> V1,V11,V7,V20,V25,V24,V15,V13,V9,V4,V18,V16,V22,V21,V5,V12,V19,V14,V2,V23,V17,V8,V10,V3,V6,
1347 ---> V1,V13,V24,V25,V23,V8,V20,V17,V14,V2,V10,V19,V12,V5,V3,V6,V7,V21,V15,V9,V4,V18,V16,V11,V22,
1372 ---> V1,V13,V23,V25,V14,V19,V17,V8,V2,V10,V9,V21,V16,V15,V11,V4,V12,V5,V3,V6,V18,V22,V7,V24,V20,
1372 ---> V1,V13,V7,V20,V19,V14,V12,V9,V23,V25,V16,V15,V4,V11,V5,V3,V6,V18,V22,V21,V24,V17,V2,V8,V10,

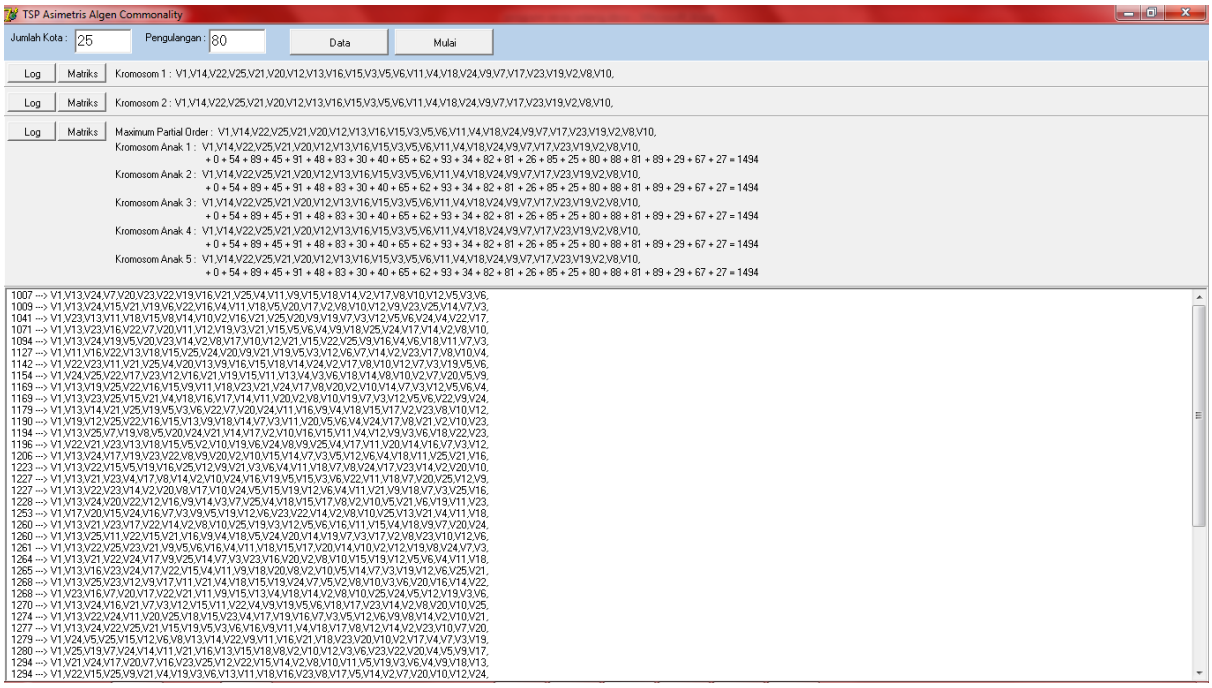

TSP Asimetris Algen Commonality
Jumlah Kota : 25
Pengulangan : 60
Data
Mulai
Log
Matriks
Kromosom 1 : V1,V13,V24,V20,V22,V12,V16,V9,V14,V3,V7,V25,V4,V18,V15,V17,V8,V2,V10,V5,V21,V6,V19,V11,V23,
Kromosom 2 : V1,V13,V24,V20,V22,V12,V16,V9,V14,V3,V7,V25,V4,V18,V15,V17,V8,V2,V10,V5,V21,V6,V19,V11,V23,
Maximum Partial Order : V1,V13,V24,V20,V22,V12,V16,V9,V14,V3,V7,V25,V4,V18,V15,V17,V8,V2,V10,V5,V21,V6,V19,V11,V23,
Kromosom Anak 1 : V1,V13,V24,V20,V22,V12,V16,V9,V14,V3,V7,V25,V4,V18,V15,V17,V8,V2,V10,V5,V21,V6,V19,V11,V23,
+ 0 + 24 + 38 + 64 + 68 + 81 + 36 + 51 + 46 + 23 + 92 + 64 + 50 + 26 + 33 + 79 + 27 + 50 + 30 + 97 + 36 + 37 + 48 + 55 + 73 = 1228
Kromosom Anak 2 : V1,V13,V24,V20,V22,V12,V16,V9,V14,V3,V7,V25,V4,V18,V15,V17,V8,V2,V10,V5,V21,V6,V19,V11,V23,
+ 0 + 24 + 38 + 64 + 68 + 81 + 36 + 51 + 46 + 23 + 92 + 64 + 50 + 26 + 33 + 79 + 27 + 50 + 30 + 97 + 36 + 37 + 48 + 55 + 73 = 1228
Kromosom Anak 3 : V1,V13,V24,V20,V22,V12,V16,V9,V14,V3,V7,V25,V4,V18,V15,V17,V8,V2,V10,V5,V21,V6,V19,V11,V23,
+ 0 + 24 + 38 + 64 + 68 + 81 + 36 + 51 + 46 + 23 + 92 + 64 + 50 + 26 + 33 + 79 + 27 + 50 + 30 + 97 + 36 + 37 + 48 + 55 + 73 = 1228
Kromosom Anak 4 : V1,V13,V24,V20,V22,V12,V16,V9,V14,V3,V7,V25,V4,V18,V15,V17,V8,V2,V10,V5,V21,V6,V19,V11,V23,
+ 0 + 24 + 38 + 64 + 68 + 81 + 36 + 51 + 46 + 23 + 92 + 64 + 50 + 26 + 33 + 79 + 27 + 50 + 30 + 97 + 36 + 37 + 48 + 55 + 73 = 1228
Kromosom Anak 5 : V1,V13,V24,V20,V22,V12,V16,V9,V14,V3,V7,V25,V4,V18,V15,V17,V8,V2,V10,V5,V21,V6,V19,V11,V23,
+ 0 + 24 + 38 + 64 + 68 + 81 + 36 + 51 + 46 + 23 + 92 + 64 + 50 + 26 + 33 + 79 + 27 + 50 + 30 + 97 + 36 + 37 + 48 + 55 + 73 = 1228
Tsp
1007 ---> V1,V13,V24,V7,V20,V23,V22,V19,V16,V21,V25,V4,V11,V9,V15,V18,V14,V2,V17,V8,V10,V12,V5,V3,V6,
OK

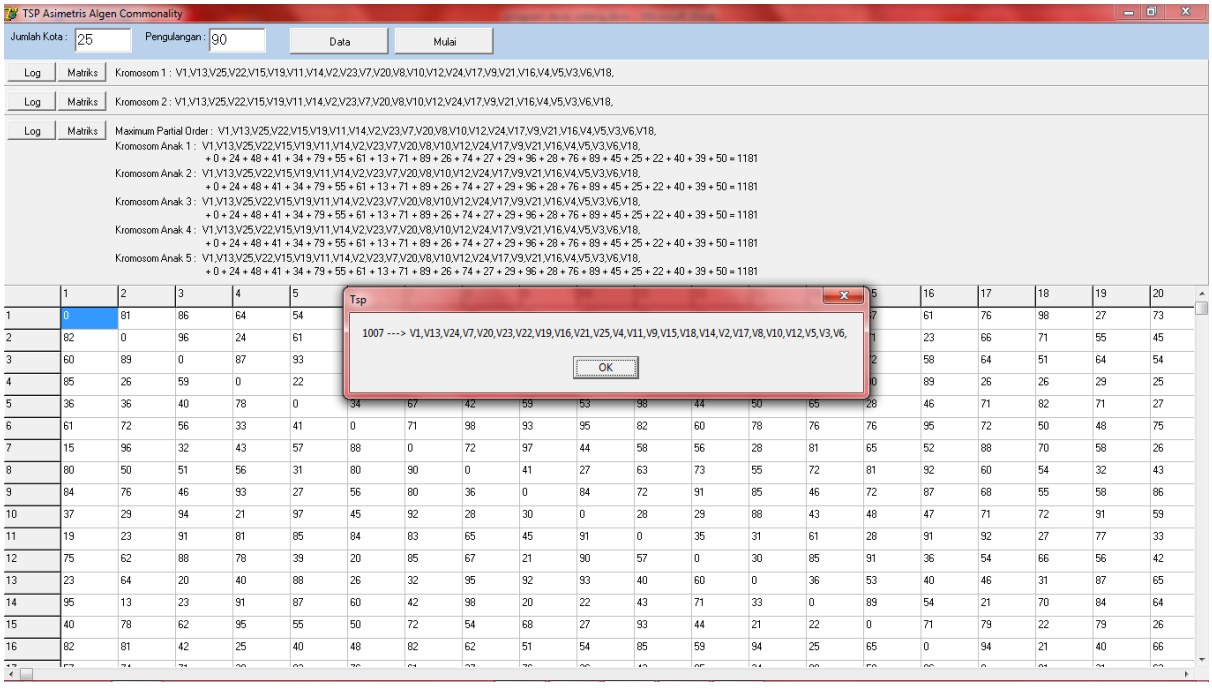

TSP Asimetris Algen Commonality
Jumlah Kota : 25
Pengulangan : 60
Data
Mulai
Log
Matriks
Kromosom 1 : V1,V13,V24,V20,V22,V12,V16,V9,V14,V3,V7,V25,V4,V18,V15,V17,V8,V2,V10,V5,V21,V6,V19,V11,V23,
Kromosom 2 : V1,V13,V24,V20,V22,V12,V16,V9,V14,V3,V7,V25,V4,V18,V15,V17,V8,V2,V10,V5,V21,V6,V19,V11,V23,
Maximum Partial Order : V1,V13,V24,V20,V22,V12,V16,V9,V14,V3,V7,V25,V4,V18,V15,V17,V8,V2,V10,V5,V21,V6,V19,V11,V23,
Kromosom Anak 1 : V1,V13,V24,V20,V22,V12,V16,V9,V14,V3,V7,V25,V4,V18,V15,V17,V8,V2,V10,V5,V21,V6,V19,V11,V23,
+ 0 + 24 + 38 + 64 + 68 + 81 + 36 + 51 + 46 + 23 + 92 + 64 + 50 + 26 + 33 + 79 + 27 + 50 + 30 + 97 + 36 + 37 + 48 + 55 + 73 = 1228
Kromosom Anak 2 : V1,V13,V24,V20,V22,V12,V16,V9,V14,V3,V7,V25,V4,V18,V15,V17,V8,V2,V10,V5,V21,V6,V19,V11,V23,
+ 0 + 24 + 38 + 64 + 68 + 81 + 36 + 51 + 46 + 23 + 92 + 64 + 50 + 26 + 33 + 79 + 27 + 50 + 30 + 97 + 36 + 37 + 48 + 55 + 73 = 1228
Kromosom Anak 3 : V1,V13,V24,V20,V22,V12,V16,V9,V14,V3,V7,V25,V4,V18,V15,V17,V8,V2,V10,V5,V21,V6,V19,V11,V23,
+ 0 + 24 + 38 + 64 + 68 + 81 + 36 + 51 + 46 + 23 + 92 + 64 + 50 + 26 + 33 + 79 + 27 + 50 + 30 + 97 + 36 + 37 + 48 + 55 + 73 = 1228
Kromosom Anak 4 : V1,V13,V24,V20,V22,V12,V16,V9,V14,V3,V7,V25,V4,V18,V15,V17,V8,V2,V10,V5,V21,V6,V19,V11,V23,
+ 0 + 24 + 38 + 64 + 68 + 81 + 36 + 51 + 46 + 23 + 92 + 64 + 50 + 26 + 33 + 79 + 27 + 50 + 30 + 97 + 36 + 37 + 48 + 55 + 73 = 1228
Kromosom Anak 5 : V1,V13,V24,V20,V22,V12,V16,V9,V14,V3,V7,V25,V4,V18,V15,V17,V8,V2,V10,V5,V21,V6,V19,V11,V23,
+ 0 + 24 + 38 + 64 + 68 + 81 + 36 + 51 + 46 + 23 + 92 + 64 + 50 + 26 + 33 + 79 + 27 + 50 + 30 + 97 + 36 + 37 + 48 + 55 + 73 = 1228
1007 ---> V1,V13,V24,V7,V20,V23,V22,V19,V16,V21,V25,V4,V11,V9,V15,V18,V14,V2,V17,V8,V10,V12,V5,V3,V6,
1009 ---> V1,V13,V24,V15,V21,V19,V6,V22,V16,V4,V11,V18,V5,V20,V17,V2,V8,V10,V12,V9,V23,V25,V14,V7,V3,
1041 ---> V1,V23,V13,V11,V18,V15,V8,V14,V10,V2,V16,V21,V25,V20,V9,V19,V7,V3,V12,V5,V6,V24,V4,V22,V17,
1071 ---> V1,V13,V23,V16,V22,V7,V20,V11,V12,V19,V3,V21,V15,V5,V6,V4,V9,V18,V25,V24,V17,V14,V2,V8,V10,
1127 ---> V1,V11,V16,V22,V13,V18,V15,V25,V24,V20,V9,V21,V19,V5,V3,V12,V6,V7,V14,V2,V23,V17,V8,V10,V4,
1142 ---> V1,V22,V23,V11,V21,V25,V4,V20,V13,V16,V15,V18,V14,V24,V2,V17,V8,V10,V12,V7,V3,V19,V5,V6,
1154 ---> V1,V24,V25,V22,V17,V23,V12,V16,V21,V19,V15,V11,V13,V4,V3,V6,V18,V14,V8,V10,V2,V7,V20,V5,V9,
1169 ---> V1,V13,V19,V25,V22,V16,V15,V9,V11,V18,V23,V21,V24,V17,V8,V20,V2,V10,V14,V7,V3,V12,V5,V6,V4,
1169 ---> V1,V13,V23,V25,V15,V21,V4,V18,V16,V17,V14,V11,V20,V2,V8,V10,V19,V7,V3,V12,V5,V6,V22,V9,V24,
1179 ---> V1,V13,V14,V21,V25,V19,V5,V3,V6,V22,V7,V20,V24,V11,V16,V9,V4,V18,V15,V17,V2,V23,V8,V10,V12,
1190 ---> V1,V19,V12,V25,V22,V16,V15,V13,V9,V18,V14,V7,V3,V11,V20,V5,V6,V4,V24,V17,V8,V21,V2,V10,V23,
1194 ---> V1,V13,V25,V7,V19,V8,V5,V20,V24,V21,V14,V17,V2,V10,V16,V15,V11,V4,V12,V9,V3,V6,V18,V22,V23,
1206 ---> V1,V13,V24,V17,V19,V23,V22,V8,V9,V20,V2,V10,V15,V14,V7,V3,V5,V12,V6,V4,V18,V11,V25,V21,V16,
1223 ---> V1,V13,V22,V15,V5,V19,V16,V25,V12,V9,V21,V3,V6,V4,V11,V18,V7,V8,V24,V17,V23,V14,V2,V20,V10,
1227 ---> V1,V13,V21,V23,V4,V17,V8,V14,V2,V10,V24,V16,V19,V5,V15,V3,V6,V22,V11,V18,V7,V20,V25,V12,V9,
1227 ---> V1,V13,V22,V23,V14,V2,V20,V8,V17,V10,V24,V5,V15,V19,V12,V6,V4,V11,V21,V9,V18,V7,V3,V25,V16,
1228 ---> V1,V13,V24,V20,V22,V12,V16,V9,V14,V3,V7,V25,V4,V18,V15,V17,V8,V2,V10,V5,V21,V6,V19,V11,V23,
1260 ---> V1,V13,V21,V23,V17,V22,V14,V2,V8,V10,V25,V19,V3,V12,V5,V6,V16,V11,V15,V4,V18,V9,V7,V20,V24,
1260 ---> V1,V13,V25,V11,V22,V15,V21,V16,V9,V4,V18,V5,V24,V20,V14,V19,V7,V3,V17,V2,V8,V23,V10,V12,V6,
1261 ---> V1,V13,V22,V25,V23,V21,V9,V5,V6,V16,V4,V11,V18,V15,V17,V20,V14,V10,V2,V12,V19,V8,V24,V7,V3,
1264 ---> V1,V13,V21,V22,V24,V17,V9,V25,V14,V7,V3,V23,V16,V20,V2,V8,V10,V15,V19,V12,V5,V6,V4,V11,V18,
1265 ---> V1,V13,V16,V23,V24,V17,V22,V15,V4,V11,V9,V18,V20,V8,V2,V10,V5,V14,V7,V3,V19,V12,V6,V25,V21,
1268 ---> V1,V13,V25,V23,V12,V9,V17,V11,V21,V4,V18,V15,V19,V24,V7,V5,V2,V8,V10,V3,V6,V20,V16,V14,V22,
1268 ---> V1,V23,V16,V7,V20,V17,V22,V21,V11,V9,V15,V13,V4,V18,V14,V2,V8,V10,V25,V24,V5,V12,V19,V3,V6,
1274 ---> V1,V13,V22,V24,V11,V20,V25,V18,V15,V23,V4,V17,V19,V16,V7,V3,V5,V12,V6,V9,V8,V14,V2,V10,V21,
1277 ---> V1,V13,V24,V22,V25,V21,V15,V19,V5,V3,V6,V16,V9,V11,V4,V18,V17,V8,V12,V14,V2,V23,V10,V7,V20,
1279 ---> V1,V24,V5,V25,V15,V12,V6,V8,V13,V14,V22,V9,V11,V16,V21,V18,V23,V20,V10,V2,V17,V4,V7,V3,V19,
1280 ---> V1,V25,V19,V7,V24,V14,V11,V21,V16,V13,V15,V18,V8,V2,V10,V12,V3,V6,V23,V22,V20,V4,V5,V9,V17,
1294 ---> V1,V22,V15,V25,V9,V21,V4,V19,V3,V6,V13,V11,V18,V16,V23,V8,V17,V5,V14,V2,V7,V20,V10,V12,V24,
1303 ---> V1,V19,V24,V22,V17,V21,V13,V9,V15,V18,V2,V7,V23,V8,V20,V10,V25,V12,V5,V3,V6,V14,V4,V11,V16,
1305 ---> V1,V13,V25,V17,V23,V24,V14,V2,V20,V8,V10,V16,V21,V22,V11,V12,V5,V19,V3,V6,V7,V15,V4,V9,V18,
1306 ---> V1,V13,V21,V23,V25,V24,V8,V17,V11,V16,V22,V15,V9,V4,V18,V19,V20,V14,V7,V3,V5,V2,V10,V12,V6,
1337 ---> V1,V19,V22,V24,V25,V23,V21,V16,V15,V13,V9,V18,V3,V6,V17,V8,V2,V10,V14,V12,V5,V4,V11,V7,V20,
1341 ---> V1,V24,V23,V12,V16,V5,V7,V20,V14,V22,V17,V13,V9,V4,V3,V6,V15,V11,V18,V2,V8,V10,V25,V21,V19,

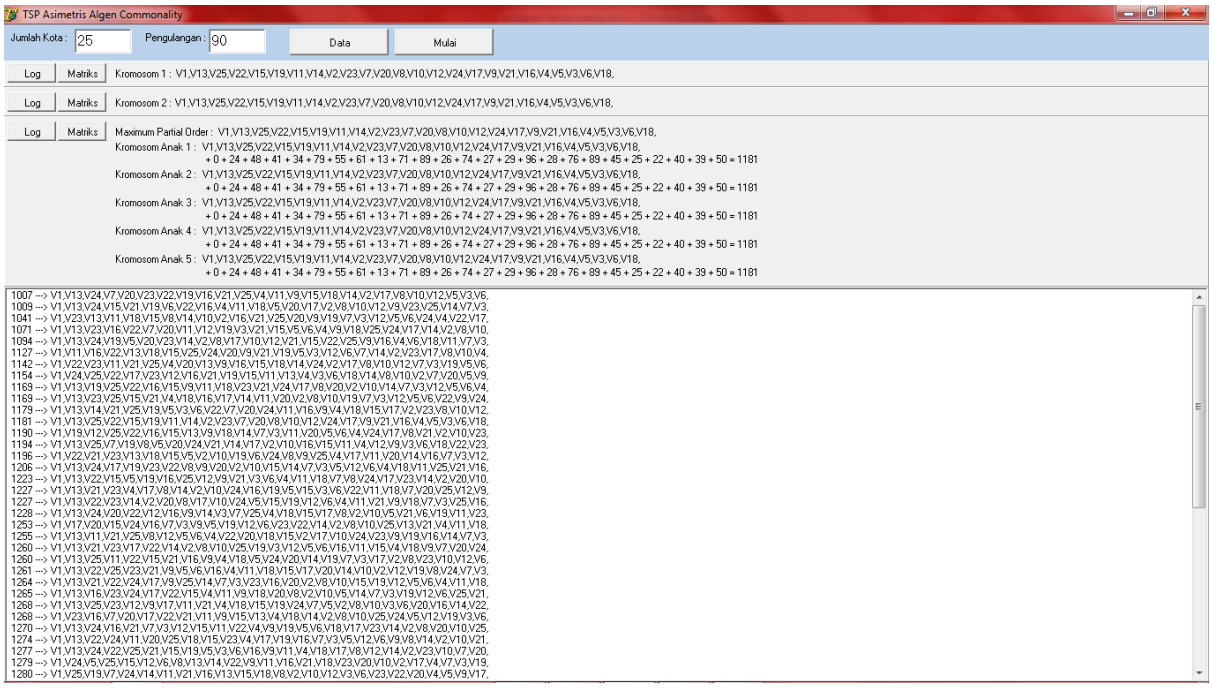

TSP Asimetris Algen Commonality
Jumlah Kota : 25
Pengulangan : 70
Data
Mulai
Log
Matriks
Kromosom 1 : V1,V13,V7,V23,V24,V17,V14,V19,V25,V16,V12,V22,V11,V8,V20,V2,V10,V5,V3,V6,V21,V4,V9,V15,V18,
Kromosom 2 : V1,V13,V7,V23,V24,V17,V14,V19,V25,V16,V12,V22,V11,V8,V20,V2,V10,V5,V3,V6,V21,V4,V9,V15,V18,
Maximum Partial Order : V1,V13,V7,V23,V24,V17,V14,V19,V25,V16,V12,V22,V11,V8,V20,V2,V10,V5,V3,V6,V21,V4,V9,V15,V18,
Kromosom Anak 1 : V1,V13,V7,V23,V24,V17,V14,V19,V25,V16,V12,V22,V11,V8,V20,V2,V10,V5,V3,V6,V21,V4,V9,V15,V18,
+ 0 + 24 + 32 + 90 + 73 + 28 + 98 + 84 + 21 + 91 + 59 + 38 + 84 + 65 + 43 + 87 + 30 + 97 + 40 + 39 + 21 + 38 + 60 + 72 + 22 = 1336
Kromosom Anak 2 : V1,V13,V7,V23,V24,V17,V14,V19,V25,V16,V12,V22,V11,V8,V20,V2,V10,V5,V3,V6,V21,V4,V9,V15,V18,
+ 0 + 24 + 32 + 90 + 73 + 28 + 98 + 84 + 21 + 91 + 59 + 38 + 84 + 65 + 43 + 87 + 30 + 97 + 40 + 39 + 21 + 38 + 60 + 72 + 22 = 1336
Kromosom Anak 3 : V1,V13,V7,V23,V24,V17,V14,V19,V25,V16,V12,V22,V11,V8,V20,V2,V10,V5,V3,V6,V21,V4,V9,V15,V18,
+ 0 + 24 + 32 + 90 + 73 + 28 + 98 + 84 + 21 + 91 + 59 + 38 + 84 + 65 + 43 + 87 + 30 + 97 + 40 + 39 + 21 + 38 + 60 + 72 + 22 = 1336
Kromosom Anak 4 : V1,V13,V7,V23,V24,V17,V14,V19,V25,V16,V12,V22,V11,V8,V20,V2,V10,V5,V3,V6,V21,V4,V9,V15,V18,
+ 0 + 24 + 32 + 90 + 73 + 28 + 98 + 84 + 21 + 91 + 59 + 38 + 84 + 65 + 43 + 87 + 30 + 97 + 40 + 39 + 21 + 38 + 60 + 72 + 22 = 1336
Kromosom Anak 5 : V1,V13,V7,V23,V24,V17,V14,V19,V25,V16,V12,V22,V11,V8,V20,V2,V10,V5,V3,V6,V21,V4,V9,V15,V18,
+ 0 + 24 + 32 + 90 + 73 + 28 + 98 + 84 + 21 + 91 + 59 + 38 + 84 + 65 + 43 + 87 + 30 + 97 + 40 + 39 + 21 + 38 + 60 + 72 + 22 = 1336
Tsp
1007 ---> V1,V13,V24,V7,V20,V23,V22,V19,V16,V21,V25,V4,V11,V9,V15,V18,V14,V2,V17,V8,V10,V12,V5,V3,V6,
OK

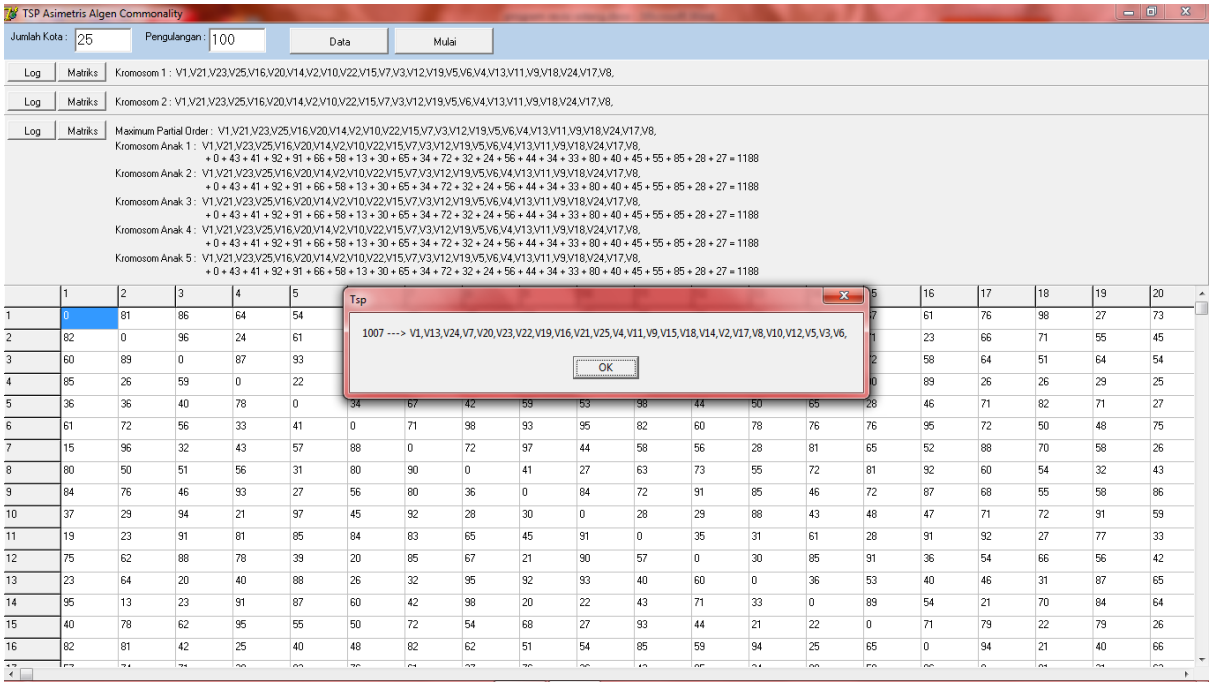

TSP Asimetris Algen Commonality
Jumlah Kota : 25
Pengulangan : 70
Data
Mulai
Log
Matriks
Kromosom 1 : V1,V13,V7,V23,V24,V17,V14,V19,V25,V16,V12,V22,V11,V8,V20,V2,V10,V5,V3,V6,V21,V4,V9,V15,V18,
Kromosom 2 : V1,V13,V7,V23,V24,V17,V14,V19,V25,V16,V12,V22,V11,V8,V20,V2,V10,V5,V3,V6,V21,V4,V9,V15,V18,
Maximum Partial Order : V1,V13,V7,V23,V24,V17,V14,V19,V25,V16,V12,V22,V11,V8,V20,V2,V10,V5,V3,V6,V21,V4,V9,V15,V18,
Kromosom Anak 1 : V1,V13,V7,V23,V24,V17,V14,V19,V25,V16,V12,V22,V11,V8,V20,V2,V10,V5,V3,V6,V21,V4,V9,V15,V18,
+ 0 + 24 + 32 + 90 + 73 + 28 + 98 + 84 + 21 + 91 + 59 + 38 + 84 + 65 + 43 + 87 + 30 + 97 + 40 + 39 + 21 + 38 + 60 + 72 + 22 = 1336
Kromosom Anak 2 : V1,V13,V7,V23,V24,V17,V14,V19,V25,V16,V12,V22,V11,V8,V20,V2,V10,V5,V3,V6,V21,V4,V9,V15,V18,
+ 0 + 24 + 32 + 90 + 73 + 28 + 98 + 84 + 21 + 91 + 59 + 38 + 84 + 65 + 43 + 87 + 30 + 97 + 40 + 39 + 21 + 38 + 60 + 72 + 22 = 1336
Kromosom Anak 3 : V1,V13,V7,V23,V24,V17,V14,V19,V25,V16,V12,V22,V11,V8,V20,V2,V10,V5,V3,V6,V21,V4,V9,V15,V18,
+ 0 + 24 + 32 + 90 + 73 + 28 + 98 + 84 + 21 + 91 + 59 + 38 + 84 + 65 + 43 + 87 + 30 + 97 + 40 + 39 + 21 + 38 + 60 + 72 + 22 = 1336
Kromosom Anak 4 : V1,V13,V7,V23,V24,V17,V14,V19,V25,V16,V12,V22,V11,V8,V20,V2,V10,V5,V3,V6,V21,V4,V9,V15,V18,
+ 0 + 24 + 32 + 90 + 73 + 28 + 98 + 84 + 21 + 91 + 59 + 38 + 84 + 65 + 43 + 87 + 30 + 97 + 40 + 39 + 21 + 38 + 60 + 72 + 22 = 1336
Kromosom Anak 5 : V1,V13,V7,V23,V24,V17,V14,V19,V25,V16,V12,V22,V11,V8,V20,V2,V10,V5,V3,V6,V21,V4,V9,V15,V18,
+ 0 + 24 + 32 + 90 + 73 + 28 + 98 + 84 + 21 + 91 + 59 + 38 + 84 + 65 + 43 + 87 + 30 + 97 + 40 + 39 + 21 + 38 + 60 + 72 + 22 = 1336
1007 ---> V1,V13,V24,V7,V20,V23,V22,V19,V16,V21,V25,V4,V11,V9,V15,V18,V14,V2,V17,V8,V10,V12,V5,V3,V6,
1009 ---> V1,V13,V24,V15,V21,V19,V6,V22,V16,V4,V11,V18,V5,V20,V17,V2,V8,V10,V12,V9,V23,V25,V14,V7,V3,
1041 ---> V1,V23,V13,V11,V18,V15,V8,V14,V10,V2,V16,V21,V25,V20,V9,V19,V7,V3,V12,V5,V6,V24,V4,V22,V17,
1071 ---> V1,V13,V23,V16,V22,V7,V20,V11,V12,V19,V3,V21,V15,V5,V6,V4,V9,V18,V25,V24,V17,V14,V2,V8,V10,
1094 ---> V1,V13,V24,V19,V5,V20,V23,V14,V2,V8,V17,V10,V12,V21,V15,V22,V25,V9,V16,V4,V6,V18,V11,V7,V3,
1127 ---> V1,V11,V16,V22,V13,V18,V15,V25,V24,V20,V9,V21,V19,V5,V3,V12,V6,V7,V14,V2,V23,V17,V8,V10,V4,
1142 ---> V1,V22,V23,V11,V21,V25,V4,V20,V13,V9,V16,V15,V18,V14,V24,V2,V17,V8,V10,V12,V7,V3,V19,V5,V6,
1154 ---> V1,V24,V25,V22,V17,V23,V12,V16,V21,V19,V15,V11,V13,V4,V3,V6,V18,V14,V8,V10,V2,V7,V20,V5,V9,
1169 ---> V1,V13,V19,V25,V22,V16,V15,V9,V11,V18,V23,V21,V24,V17,V8,V20,V2,V10,V14,V7,V3,V12,V5,V6,V4,
1169 ---> V1,V13,V23,V25,V15,V21,V4,V18,V16,V17,V14,V11,V20,V2,V8,V10,V19,V7,V3,V12,V5,V6,V22,V9,V24,
1179 ---> V1,V13,V14,V21,V25,V19,V5,V3,V6,V22,V7,V20,V24,V11,V16,V9,V4,V18,V15,V17,V2,V23,V8,V10,V12,
1190 ---> V1,V19,V12,V25,V22,V16,V15,V13,V9,V18,V14,V7,V3,V11,V20,V5,V6,V4,V24,V17,V8,V21,V2,V10,V23,
1194 ---> V1,V13,V25,V7,V19,V8,V5,V20,V24,V21,V14,V17,V2,V10,V16,V15,V11,V4,V12,V9,V3,V6,V18,V22,V23,
1206 ---> V1,V13,V24,V17,V19,V23,V22,V8,V9,V20,V2,V10,V15,V14,V7,V3,V5,V12,V6,V4,V18,V11,V25,V21,V16,
1223 ---> V1,V13,V22,V15,V5,V19,V16,V25,V12,V9,V21,V3,V6,V4,V11,V18,V7,V8,V24,V17,V23,V14,V2,V20,V10,
1227 ---> V1,V13,V21,V23,V4,V17,V8,V14,V2,V10,V24,V16,V19,V5,V15,V3,V6,V22,V11,V18,V7,V20,V25,V12,V9,
1227 ---> V1,V13,V22,V23,V14,V2,V20,V8,V17,V10,V24,V5,V15,V19,V12,V6,V4,V11,V21,V9,V18,V7,V3,V25,V16,
1228 ---> V1,V13,V24,V20,V22,V12,V16,V9,V14,V3,V7,V25,V4,V18,V15,V17,V8,V2,V10,V5,V21,V6,V19,V11,V23,
1260 ---> V1,V13,V21,V23,V17,V22,V14,V2,V8,V10,V25,V19,V3,V12,V5,V6,V16,V11,V15,V4,V18,V9,V7,V20,V24,
1260 ---> V1,V13,V25,V11,V22,V15,V21,V16,V9,V4,V18,V5,V24,V20,V14,V19,V7,V3,V17,V2,V8,V23,V10,V12,V6,
1261 ---> V1,V13,V22,V25,V23,V21,V9,V5,V6,V16,V4,V11,V18,V15,V17,V20,V14,V10,V2,V12,V19,V8,V24,V7,V3,
1264 ---> V1,V13,V21,V22,V24,V17,V9,V25,V14,V7,V3,V23,V16,V20,V2,V8,V10,V15,V19,V12,V5,V6,V4,V11,V18,
1265 ---> V1,V13,V16,V23,V24,V17,V22,V15,V4,V11,V9,V18,V20,V8,V2,V10,V5,V14,V7,V3,V19,V12,V6,V25,V21,
1268 ---> V1,V13,V25,V23,V12,V9,V17,V11,V21,V4,V18,V15,V19,V24,V7,V5,V2,V8,V10,V3,V6,V20,V16,V14,V22,
1268 ---> V1,V23,V16,V7,V20,V17,V22,V21,V11,V9,V15,V13,V4,V18,V14,V2,V8,V10,V25,V24,V5,V12,V19,V3,V6,
1274 ---> V1,V13,V22,V24,V11,V20,V25,V18,V15,V23,V4,V17,V19,V16,V7,V3,V5,V12,V6,V9,V8,V14,V2,V10,V21,
1277 ---> V1,V13,V24,V22,V25,V21,V15,V19,V5,V3,V6,V16,V9,V11,V4,V18,V17,V8,V12,V14,V2,V23,V10,V7,V20,
1279 ---> V1,V24,V5,V25,V15,V12,V6,V8,V13,V14,V22,V9,V11,V16,V21,V18,V23,V20,V10,V2,V17,V4,V7,V3,V19,
1280 ---> V1,V25,V19,V7,V24,V14,V11,V21,V16,V13,V15,V18,V8,V2,V10,V12,V3,V6,V23,V22,V20,V4,V5,V9,V17,
1294 ---> V1,V21,V24,V17,V20,V7,V16,V23,V25,V12,V22,V15,V14,V2,V8,V10,V11,V5,V19,V3,V6,V4,V9,V18,V13,
1294 ---> V1,V22,V15,V25,V9,V21,V4,V19,V3,V6,V13,V11,V18,V16,V23,V8,V17,V5,V14,V2,V7,V20,V10,V12,V24,
1303 ---> V1,V19,V24,V22,V17,V21,V13,V9,V15,V18,V2,V7,V23,V8,V20,V10,V25,V12,V5,V3,V6,V14,V4,V11,V16,
1305 ---> V1,V13,V25,V17,V23,V24,V14,V2,V20,V8,V10,V16,V21,V22,V11,V12,V5,V19,V3,V6,V7,V15,V4,V9,V18,
1306 ---> V1,V13,V21,V23,V25,V24,V8,V17,V11,V16,V22,V15,V9,V4,V18,V19,V20,V14,V7,V3,V5,V2,V10,V12,V6,

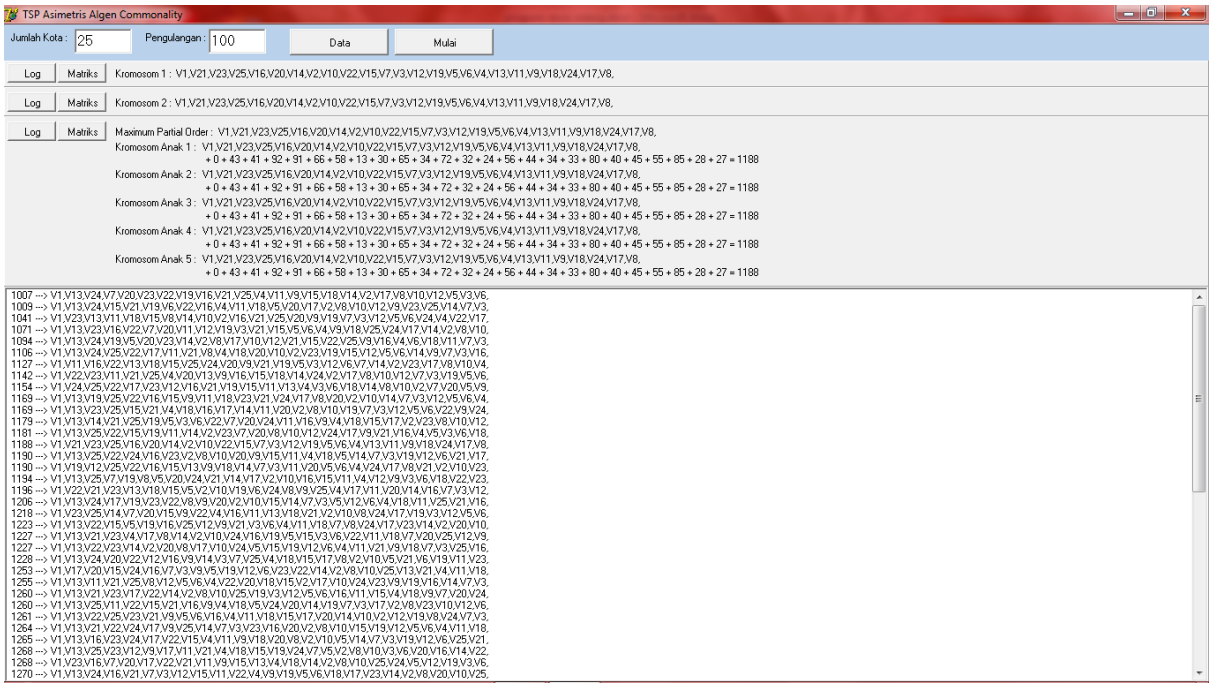

TSP Asimetris Algen Commonality
Jumlah Kota : 25
Pengulangan : 80
Data
Mulai
Log
Matriks
Kromosom 1 : V1,V14,V22,V25,V21,V20,V12,V13,V16,V15,V3,V5,V6,V11,V4,V18,V24,V9,V7,V17,V23,V19,V2,V8,V10,
Kromosom 2 : V1,V14,V22,V25,V21,V20,V12,V13,V16,V15,V3,V5,V6,V11,V4,V18,V24,V9,V7,V17,V23,V19,V2,V8,V10,
Maximum Partial Order : V1,V14,V22,V25,V21,V20,V12,V13,V16,V15,V3,V5,V6,V11,V4,V18,V24,V9,V7,V17,V23,V19,V2,V8,V10,
Kromosom Anak 1 : V1,V14,V22,V25,V21,V20,V12,V13,V16,V15,V3,V5,V6,V11,V4,V18,V24,V9,V7,V17,V23,V19,V2,V8,V10,
+ 0 + 54 + 89 + 45 + 91 + 48 + 83 + 30 + 40 + 65 + 62 + 93 + 34 + 82 + 81 + 26 + 85 + 25 + 80 + 88 + 81 + 89 + 29 + 67 + 27 = 1494
Kromosom Anak 2 : V1,V14,V22,V25,V21,V20,V12,V13,V16,V15,V3,V5,V6,V11,V4,V18,V24,V9,V7,V17,V23,V19,V2,V8,V10,
+ 0 + 54 + 89 + 45 + 91 + 48 + 83 + 30 + 40 + 65 + 62 + 93 + 34 + 82 + 81 + 26 + 85 + 25 + 80 + 88 + 81 + 89 + 29 + 67 + 27 = 1494
Kromosom Anak 3 : V1,V14,V22,V25,V21,V20,V12,V13,V16,V15,V3,V5,V6,V11,V4,V18,V24,V9,V7,V17,V23,V19,V2,V8,V10,
+ 0 + 54 + 89 + 45 + 91 + 48 + 83 + 30 + 40 + 65 + 62 + 93 + 34 + 82 + 81 + 26 + 85 + 25 + 80 + 88 + 81 + 89 + 29 + 67 + 27 = 1494
Kromosom Anak 4 : V1,V14,V22,V25,V21,V20,V12,V13,V16,V15,V3,V5,V6,V11,V4,V18,V24,V9,V7,V17,V23,V19,V2,V8,V10,
+ 0 + 54 + 89 + 45 + 91 + 48 + 83 + 30 + 40 + 65 + 62 + 93 + 34 + 82 + 81 + 26 + 85 + 25 + 80 + 88 + 81 + 89 + 29 + 67 + 27 = 1494
Kromosom Anak 5 : V1,V14,V22,V25,V21,V20,V12,V13,V16,V15,V3,V5,V6,V11,V4,V18,V24,V9,V7,V17,V23,V19,V2,V8,V10,
+ 0 + 54 + 89 + 45 + 91 + 48 + 83 + 30 + 40 + 65 + 62 + 93 + 34 + 82 + 81 + 26 + 85 + 25 + 80 + 88 + 81 + 89 + 29 + 67 + 27 = 1494
Tsp
1007 ---> V1,V13,V24,V7,V20,V23,V22,V19,V16,V21,V25,V4,V11,V9,V15,V18,V14,V2,V17,V8,V10,V12,V5,V3,V6,
OK


TSP Asimetris Algen Commonality
Jumlah Kota : 25
Pengulangan : 80
Data
Mulai
Log
Matriks
Kromosom 1 : V1,V14,V22,V25,V21,V20,V12,V13,V16,V15,V3,V5,V6,V11,V4,V18,V24,V9,V7,V17,V23,V19,V2,V8,V10,
Kromosom 2 : V1,V14,V22,V25,V21,V20,V12,V13,V16,V15,V3,V5,V6,V11,V4,V18,V24,V9,V7,V17,V23,V19,V2,V8,V10,
Maximum Partial Order : V1,V14,V22,V25,V21,V20,V12,V13,V16,V15,V3,V5,V6,V11,V4,V18,V24,V9,V7,V17,V23,V19,V2,V8,V10,
Kromosom Anak 1 : V1,V14,V22,V25,V21,V20,V12,V13,V16,V15,V3,V5,V6,V11,V4,V18,V24,V9,V7,V17,V23,V19,V2,V8,V10,
+ 0 + 54 + 89 + 45 + 91 + 48 + 83 + 30 + 40 + 65 + 62 + 93 + 34 + 82 + 81 + 26 + 85 + 25 + 80 + 88 + 81 + 89 + 29 + 67 + 27 = 1494
Kromosom Anak 2 : V1,V14,V22,V25,V21,V20,V12,V13,V16,V15,V3,V5,V6,V11,V4,V18,V24,V9,V7,V17,V23,V19,V2,V8,V10,
+ 0 + 54 + 89 + 45 + 91 + 48 + 83 + 30 + 40 + 65 + 62 + 93 + 34 + 82 + 81 + 26 + 85 + 25 + 80 + 88 + 81 + 89 + 29 + 67 + 27 = 1494
Kromosom Anak 3 : V1,V14,V22,V25,V21,V20,V12,V13,V16,V15,V3,V5,V6,V11,V4,V18,V24,V9,V7,V17,V23,V19,V2,V8,V10,
+ 0 + 54 + 89 + 45 + 91 + 48 + 83 + 30 + 40 + 65 + 62 + 93 + 34 + 82 + 81 + 26 + 85 + 25 + 80 + 88 + 81 + 89 + 29 + 67 + 27 = 1494
Kromosom Anak 4 : V1,V14,V22,V25,V21,V20,V12,V13,V16,V15,V3,V5,V6,V11,V4,V18,V24,V9,V7,V17,V23,V19,V2,V8,V10,
+ 0 + 54 + 89 + 45 + 91 + 48 + 83 + 30 + 40 + 65 + 62 + 93 + 34 + 82 + 81 + 26 + 85 + 25 + 80 + 88 + 81 + 89 + 29 + 67 + 27 = 1494
Kromosom Anak 5 : V1,V14,V22,V25,V21,V20,V12,V13,V16,V15,V3,V5,V6,V11,V4,V18,V24,V9,V7,V17,V23,V19,V2,V8,V10,
+ 0 + 54 + 89 + 45 + 91 + 48 + 83 + 30 + 40 + 65 + 62 + 93 + 34 + 82 + 81 + 26 + 85 + 25 + 80 + 88 + 81 + 89 + 29 + 67 + 27 = 1494
1007 ---> V1,V13,V24,V7,V20,V23,V22,V19,V16,V21,V25,V4,V11,V9,V15,V18,V14,V2,V17,V8,V10,V12,V5,V3,V6,
1009 ---> V1,V13,V24,V15,V21,V19,V6,V22,V16,V4,V11,V18,V5,V20,V17,V2,V8,V10,V12,V9,V23,V25,V14,V7,V3,
1041 ---> V1,V23,V13,V11,V18,V15,V8,V14,V10,V2,V16,V21,V25,V20,V9,V19,V7,V3,V12,V5,V6,V24,V4,V22,V17,
1071 ---> V1,V13,V23,V16,V22,V7,V20,V11,V12,V19,V3,V21,V15,V5,V6,V4,V9,V18,V25,V24,V17,V14,V2,V8,V10,
1094 ---> V1,V13,V24,V19,V5,V20,V23,V14,V2,V8,V17,V10,V12,V21,V15,V22,V25,V9,V16,V4,V6,V18,V11,V7,V3,
1127 ---> V1,V11,V16,V22,V13,V18,V15,V25,V24,V20,V9,V21,V19,V5,V3,V12,V6,V7,V14,V2,V23,V17,V8,V10,V4,
1142 ---> V1,V22,V23,V11,V21,V25,V4,V20,V13,V9,V16,V15,V18,V14,V24,V2,V17,V8,V10,V12,V7,V3,V19,V5,V6,
1154 ---> V1,V24,V25,V22,V17,V23,V12,V16,V21,V19,V15,V11,V13,V4,V3,V6,V18,V14,V8,V10,V2,V7,V20,V5,V9,
1169 ---> V1,V13,V19,V25,V22,V16,V15,V9,V11,V18,V23,V21,V24,V17,V8,V20,V2,V10,V14,V7,V3,V12,V5,V6,V4,
1169 ---> V1,V13,V23,V25,V15,V21,V4,V18,V16,V17,V14,V11,V20,V2,V8,V10,V19,V7,V3,V12,V5,V6,V22,V9,V24,
1179 ---> V1,V13,V14,V21,V25,V19,V5,V3,V6,V22,V7,V20,V24,V11,V16,V9,V4,V18,V15,V17,V2,V23,V8,V10,V12,
1190 ---> V1,V19,V12,V25,V22,V16,V15,V13,V9,V18,V14,V7,V3,V11,V20,V5,V6,V4,V24,V17,V8,V21,V2,V10,V23,
1194 ---> V1,V13,V25,V7,V19,V8,V5,V20,V24,V21,V14,V17,V2,V10,V16,V15,V11,V4,V12,V9,V3,V6,V18,V22,V23,
1196 ---> V1,V22,V21,V23,V13,V18,V15,V5,V2,V10,V19,V6,V24,V8,V9,V25,V4,V17,V11,V20,V14,V16,V7,V3,V12,
1206 ---> V1,V13,V24,V17,V19,V23,V22,V8,V9,V20,V2,V10,V15,V14,V7,V3,V5,V12,V6,V4,V18,V11,V25,V21,V16,
1223 ---> V1,V13,V22,V15,V6,V19,V16,V25,V12,V9,V21,V3,V6,V4,V11,V18,V7,V8,V24,V17,V23,V14,V2,V20,V10,
1227 ---> V1,V13,V21,V23,V4,V17,V8,V14,V2,V10,V24,V16,V19,V5,V15,V3,V6,V22,V11,V18,V7,V20,V25,V12,V9,
1227 ---> V1,V13,V22,V23,V14,V2,V20,V8,V17,V10,V24,V5,V15,V19,V12,V6,V4,V11,V21,V9,V18,V7,V3,V25,V16,
1228 ---> V1,V13,V24,V20,V22,V12,V16,V9,V14,V3,V7,V25,V4,V18,V15,V17,V8,V2,V10,V5,V21,V6,V19,V11,V23,
1253 ---> V1,V17,V20,V15,V24,V16,V7,V3,V9,V5,V19,V12,V6,V23,V22,V14,V2,V8,V10,V25,V13,V21,V4,V11,V18,
1260 ---> V1,V13,V21,V23,V17,V22,V14,V2,V8,V10,V25,V19,V3,V12,V5,V6,V16,V11,V15,V4,V18,V9,V7,V20,V24,
1260 ---> V1,V13,V25,V11,V22,V15,V21,V16,V9,V4,V18,V5,V24,V20,V14,V19,V7,V3,V17,V2,V8,V23,V10,V12,V6,
1261 ---> V1,V13,V22,V25,V23,V21,V9,V5,V6,V16,V4,V11,V18,V15,V17,V20,V14,V10,V2,V12,V19,V8,V24,V7,V3,
1264 ---> V1,V13,V21,V22,V24,V17,V9,V25,V14,V7,V3,V23,V16,V20,V2,V8,V10,V15,V19,V12,V5,V6,V4,V11,V18,
1265 ---> V1,V13,V16,V23,V24,V17,V22,V15,V4,V11,V9,V18,V20,V8,V2,V10,V5,V14,V7,V3,V19,V12,V6,V25,V21,
1268 ---> V1,V13,V25,V23,V12,V9,V17,V11,V21,V4,V18,V15,V19,V24,V7,V5,V2,V8,V10,V3,V6,V20,V16,V14,V22,
1268 ---> V1,V23,V16,V7,V20,V17,V22,V21,V11,V9,V15,V13,V4,V18,V14,V2,V8,V10,V25,V24,V5,V12,V19,V3,V6,
1270 ---> V1,V13,V24,V16,V21,V7,V3,V12,V15,V11,V22,V4,V9,V19,V5,V6,V18,V17,V23,V14,V2,V8,V20,V10,V25,
1274 ---> V1,V13,V22,V24,V11,V20,V25,V18,V15,V23,V4,V17,V19,V16,V7,V3,V5,V12,V6,V9,V8,V14,V2,V10,V21,
1277 ---> V1,V13,V24,V22,V25,V21,V15,V19,V5,V3,V6,V16,V9,V11,V4,V18,V17,V8,V12,V14,V2,V23,V10,V7,V20,
1279 ---> V1,V24,V5,V25,V15,V12,V6,V8,V13,V14,V22,V9,V11,V16,V21,V18,V23,V20,V10,V2,V17,V4,V7,V3,V19,
1280 ---> V1,V25,V19,V7,V24,V14,V11,V21,V16,V13,V15,V18,V8,V2,V10,V12,V3,V6,V23,V22,V20,V4,V5,V9,V17,
1294 ---> V1,V21,V24,V17,V20,V7,V16,V23,V25,V12,V22,V15,V14,V2,V8,V10,V11,V5,V19,V3,V6,V4,V9,V18,V13,
1294 ---> V1,V22,V15,V25,V9,V21,V4,V19,V3,V6,V13,V11,V18,V16,V23,V8,V17,V5,V14,V2,V7,V20,V10,V12,V24,


TSP Asimetris Algen Commonality
Jumlah Kota : 25
Pengulangan : 90
Data
Mulai
Log
Matriks
Kromosom 1 : V1,V13,V25,V22,V15,V19,V11,V14,V2,V23,V7,V20,V8,V10,V12,V24,V17,V9,V21,V16,V4,V5,V3,V6,V18,
Kromosom 2 : V1,V13,V25,V22,V15,V19,V11,V14,V2,V23,V7,V20,V8,V10,V12,V24,V17,V9,V21,V16,V4,V5,V3,V6,V18,
Maximum Partial Order : V1,V13,V25,V22,V15,V19,V11,V14,V2,V23,V7,V20,V8,V10,V12,V24,V17,V9,V21,V16,V4,V5,V3,V6,V18,
Kromosom Anak 1 : V1,V13,V25,V22,V15,V19,V11,V14,V2,V23,V7,V20,V8,V10,V12,V24,V17,V9,V21,V16,V4,V5,V3,V6,V18,
+ 0 + 24 + 48 + 41 + 34 + 79 + 55 + 61 + 13 + 71 + 89 + 26 + 74 + 27 + 29 + 96 + 28 + 76 + 89 + 45 + 25 + 22 + 40 + 39 + 50 = 1181
Kromosom Anak 2 : V1,V13,V25,V22,V15,V19,V11,V14,V2,V23,V7,V20,V8,V10,V12,V24,V17,V9,V21,V16,V4,V5,V3,V6,V18,
+ 0 + 24 + 48 + 41 + 34 + 79 + 55 + 61 + 13 + 71 + 89 + 26 + 74 + 27 + 29 + 96 + 28 + 76 + 89 + 45 + 25 + 22 + 40 + 39 + 50 = 1181
Kromosom Anak 3 : V1,V13,V25,V22,V15,V19,V11,V14,V2,V23,V7,V20,V8,V10,V12,V24,V17,V9,V21,V16,V4,V5,V3,V6,V18,
+ 0 + 24 + 48 + 41 + 34 + 79 + 55 + 61 + 13 + 71 + 89 + 26 + 74 + 27 + 29 + 96 + 28 + 76 + 89 + 45 + 25 + 22 + 40 + 39 + 50 = 1181
Kromosom Anak 4 : V1,V13,V25,V22,V15,V19,V11,V14,V2,V23,V7,V20,V8,V10,V12,V24,V17,V9,V21,V16,V4,V5,V3,V6,V18,
+ 0 + 24 + 48 + 41 + 34 + 79 + 55 + 61 + 13 + 71 + 89 + 26 + 74 + 27 + 29 + 96 + 28 + 76 + 89 + 45 + 25 + 22 + 40 + 39 + 50 = 1181
Kromosom Anak 5 : V1,V13,V25,V22,V15,V19,V11,V14,V2,V23,V7,V20,V8,V10,V12,V24,V17,V9,V21,V16,V4,V5,V3,V6,V18,
+ 0 + 24 + 48 + 41 + 34 + 79 + 55 + 61 + 13 + 71 + 89 + 26 + 74 + 27 + 29 + 96 + 28 + 76 + 89 + 45 + 25 + 22 + 40 + 39 + 50 = 1181
Tsp
1007 ---> V1,V13,V24,V7,V20,V23,V22,V19,V16,V21,V25,V4,V11,V9,V15,V18,V14,V2,V17,V8,V10,V12,V5,V3,V6,
OK


TSP Asimetris Algen Commonality
Jumlah Kota : 25
Pengulangan : 90
Data
Mulai
Log
Matriks
Kromosom 1 : V1,V13,V25,V22,V15,V19,V11,V14,V2,V23,V7,V20,V8,V10,V12,V24,V17,V9,V21,V16,V4,V5,V3,V6,V18,
Kromosom 2 : V1,V13,V25,V22,V15,V19,V11,V14,V2,V23,V7,V20,V8,V10,V12,V24,V17,V9,V21,V16,V4,V5,V3,V6,V18,
Maximum Partial Order : V1,V13,V25,V22,V15,V19,V11,V14,V2,V23,V7,V20,V8,V10,V12,V24,V17,V9,V21,V16,V4,V5,V3,V6,V18,
Kromosom Anak 1 : V1,V13,V25,V22,V15,V19,V11,V14,V2,V23,V7,V20,V8,V10,V12,V24,V17,V9,V21,V16,V4,V5,V3,V6,V18,
+ 0 + 24 + 48 + 41 + 34 + 79 + 55 + 61 + 13 + 71 + 89 + 26 + 74 + 27 + 29 + 96 + 28 + 76 + 89 + 45 + 25 + 22 + 40 + 39 + 50 = 1181
Kromosom Anak 2 : V1,V13,V25,V22,V15,V19,V11,V14,V2,V23,V7,V20,V8,V10,V12,V24,V17,V9,V21,V16,V4,V5,V3,V6,V18,
+ 0 + 24 + 48 + 41 + 34 + 79 + 55 + 61 + 13 + 71 + 89 + 26 + 74 + 27 + 29 + 96 + 28 + 76 + 89 + 45 + 25 + 22 + 40 + 39 + 50 = 1181
Kromosom Anak 3 : V1,V13,V25,V22,V15,V19,V11,V14,V2,V23,V7,V20,V8,V10,V12,V24,V17,V9,V21,V16,V4,V5,V3,V6,V18,
+ 0 + 24 + 48 + 41 + 34 + 79 + 55 + 61 + 13 + 71 + 89 + 26 + 74 + 27 + 29 + 96 + 28 + 76 + 89 + 45 + 25 + 22 + 40 + 39 + 50 = 1181
Kromosom Anak 4 : V1,V13,V25,V22,V15,V19,V11,V14,V2,V23,V7,V20,V8,V10,V12,V24,V17,V9,V21,V16,V4,V5,V3,V6,V18,
+ 0 + 24 + 48 + 41 + 34 + 79 + 55 + 61 + 13 + 71 + 89 + 26 + 74 + 27 + 29 + 96 + 28 + 76 + 89 + 45 + 25 + 22 + 40 + 39 + 50 = 1181
Kromosom Anak 5 : V1,V13,V25,V22,V15,V19,V11,V14,V2,V23,V7,V20,V8,V10,V12,V24,V17,V9,V21,V16,V4,V5,V3,V6,V18,
+ 0 + 24 + 48 + 41 + 34 + 79 + 55 + 61 + 13 + 71 + 89 + 26 + 74 + 27 + 29 + 96 + 28 + 76 + 89 + 45 + 25 + 22 + 40 + 39 + 50 = 1181
1007 ---> V1,V13,V24,V7,V20,V23,V22,V19,V16,V21,V25,V4,V11,V9,V15,V18,V14,V2,V17,V8,V10,V12,V5,V3,V6,
1009 ---> V1,V13,V24,V15,V21,V19,V6,V22,V16,V4,V11,V18,V5,V20,V17,V2,V8,V10,V12,V9,V23,V25,V14,V7,V3,
1041 ---> V1,V23,V13,V11,V18,V15,V8,V14,V10,V2,V16,V21,V25,V20,V9,V19,V7,V3,V12,V5,V6,V24,V4,V22,V17,
1071 ---> V1,V13,V23,V16,V22,V7,V20,V11,V12,V19,V3,V21,V15,V5,V6,V4,V9,V18,V25,V24,V17,V14,V2,V8,V10,
1094 ---> V1,V13,V24,V19,V5,V20,V23,V14,V2,V8,V17,V10,V12,V21,V15,V22,V25,V9,V16,V4,V6,V18,V11,V7,V3,
1127 ---> V1,V11,V16,V22,V13,V18,V15,V25,V24,V20,V9,V21,V19,V5,V3,V12,V6,V7,V14,V2,V23,V17,V8,V10,V4,
1142 ---> V1,V22,V23,V11,V21,V25,V4,V20,V13,V9,V16,V15,V18,V14,V24,V2,V17,V8,V10,V12,V7,V3,V19,V5,V6,
1154 ---> V1,V24,V25,V22,V17,V23,V12,V16,V21,V19,V15,V11,V13,V4,V3,V6,V18,V14,V8,V10,V2,V7,V20,V5,V9,
1169 ---> V1,V13,V19,V25,V22,V16,V15,V9,V11,V18,V23,V21,V24,V17,V8,V20,V2,V10,V14,V7,V3,V12,V5,V6,V4,
1169 ---> V1,V13,V23,V25,V15,V21,V4,V18,V16,V17,V14,V11,V20,V2,V8,V10,V19,V7,V3,V12,V5,V6,V22,V9,V24,
1179 ---> V1,V13,V14,V21,V25,V19,V5,V3,V6,V22,V7,V20,V24,V11,V16,V9,V4,V18,V15,V17,V2,V23,V8,V10,V12,
1181 ---> V1,V13,V25,V22,V15,V19,V11,V14,V2,V23,V7,V20,V8,V10,V12,V24,V17,V9,V21,V16,V4,V6,V3,V6,V18,
1190 ---> V1,V19,V12,V25,V22,V16,V15,V13,V9,V18,V14,V7,V3,V11,V20,V5,V6,V4,V24,V17,V8,V21,V2,V10,V23,
1194 ---> V1,V13,V25,V7,V19,V8,V5,V20,V24,V21,V14,V17,V2,V10,V16,V15,V11,V4,V12,V9,V3,V6,V18,V22,V23,
1196 ---> V1,V2,V21,V23,V13,V18,V15,V5,V2,V10,V19,V6,V24,V8,V9,V25,V4,V17,V11,V20,V14,V16,V7,V3,V12,
1206 ---> V1,V13,V24,V17,V19,V23,V22,V8,V9,V20,V2,V10,V15,V14,V7,V3,V5,V12,V6,V4,V18,V11,V25,V21,V16,
1223 ---> V1,V13,V22,V15,V5,V19,V16,V25,V12,V9,V21,V3,V6,V4,V11,V18,V7,V8,V24,V17,V23,V14,V2,V20,V10,
1227 ---> V1,V13,V21,V23,V4,V17,V8,V14,V2,V10,V24,V16,V19,V5,V15,V3,V6,V22,V11,V18,V7,V20,V25,V12,V9,
1227 ---> V1,V13,V22,V23,V14,V2,V20,V8,V17,V10,V24,V5,V15,V19,V12,V6,V4,V11,V21,V9,V18,V7,V3,V25,V16,
1228 ---> V1,V13,V24,V20,V22,V12,V16,V9,V14,V3,V7,V25,V4,V18,V15,V17,V8,V2,V10,V5,V21,V6,V19,V11,V23,
1253 ---> V1,V17,V20,V15,V24,V16,V7,V3,V9,V5,V19,V12,V6,V23,V22,V14,V2,V8,V10,V25,V13,V21,V4,V11,V18,
1255 ---> V1,V13,V11,V21,V25,V8,V12,V5,V6,V4,V22,V20,V18,V15,V2,V17,V10,V24,V23,V9,V19,V16,V14,V7,V3,
1260 ---> V1,V13,V21,V23,V17,V22,V14,V2,V8,V10,V25,V19,V3,V12,V5,V6,V16,V11,V15,V4,V18,V9,V7,V20,V24,
1260 ---> V1,V13,V25,V11,V22,V15,V21,V16,V9,V4,V18,V5,V24,V20,V14,V19,V7,V3,V17,V2,V8,V23,V10,V12,V6,
1261 ---> V1,V13,V22,V25,V23,V21,V9,V5,V6,V16,V4,V11,V18,V15,V17,V20,V14,V10,V2,V12,V19,V8,V24,V7,V3,
1264 ---> V1,V13,V21,V22,V24,V17,V9,V25,V14,V7,V3,V23,V16,V20,V2,V8,V10,V15,V19,V12,V5,V6,V4,V11,V18,
1265 ---> V1,V13,V16,V23,V24,V17,V22,V15,V4,V11,V9,V18,V20,V8,V2,V10,V5,V14,V7,V3,V19,V12,V6,V25,V21,
1268 ---> V1,V13,V25,V23,V12,V9,V17,V11,V21,V4,V18,V15,V19,V24,V7,V5,V2,V8,V10,V3,V6,V20,V16,V14,V22,
1268 ---> V1,V23,V16,V7,V20,V17,V22,V21,V11,V9,V15,V13,V4,V18,V14,V2,V8,V10,V25,V24,V5,V12,V19,V3,V6,
1270 ---> V1,V13,V24,V16,V21,V7,V3,V12,V15,V11,V22,V4,V9,V19,V5,V6,V18,V17,V23,V14,V2,V8,V20,V10,V25,
1274 ---> V1,V13,V22,V24,V11,V20,V25,V18,V15,V23,V4,V17,V19,V16,V7,V3,V5,V12,V6,V9,V8,V14,V2,V10,V21,
1277 ---> V1,V13,V24,V22,V25,V21,V15,V19,V5,V3,V6,V16,V9,V11,V4,V18,V17,V8,V12,V14,V2,V23,V10,V7,V20,
1279 ---> V1,V24,V5,V25,V15,V12,V6,V8,V13,V14,V22,V9,V11,V16,V21,V18,V23,V20,V10,V2,V17,V4,V7,V3,V19,
1280 ---> V1,V25,V19,V7,V24,V14,V11,V21,V16,V13,V15,V18,V8,V2,V10,V12,V3,V6,V23,V22,V20,V4,V5,V9,V17,


TSP Asimetris Algen Commonality
Jumlah Kota : 25
Pengulangan : 100
Data
Mulai
Log
Matriks
Kromosom 1 : V1,V21,V23,V25,V16,V20,V14,V2,V10,V22,V15,V7,V3,V12,V19,V5,V6,V4,V13,V11,V9,V18,V24,V17,V8,
Kromosom 2 : V1,V21,V23,V25,V16,V20,V14,V2,V10,V22,V15,V7,V3,V12,V19,V5,V6,V4,V13,V11,V9,V18,V24,V17,V8,
Maximum Partial Order : V1,V21,V23,V25,V16,V20,V14,V2,V10,V22,V15,V7,V3,V12,V19,V5,V6,V4,V13,V11,V9,V18,V24,V17,V8,
Kromosom Anak 1 : V1,V21,V23,V25,V16,V20,V14,V2,V10,V22,V15,V7,V3,V12,V19,V5,V6,V4,V13,V11,V9,V18,V24,V17,V8,
+ 0 + 43 + 41 + 92 + 91 + 66 + 58 + 13 + 30 + 65 + 34 + 72 + 32 + 24 + 56 + 44 + 34 + 33 + 80 + 40 + 45 + 55 + 85 + 28 + 27 = 1188
Kromosom Anak 2 : V1,V21,V23,V25,V16,V20,V14,V2,V10,V22,V15,V7,V3,V12,V19,V5,V6,V4,V13,V11,V9,V18,V24,V17,V8,
+ 0 + 43 + 41 + 92 + 91 + 66 + 58 + 13 + 30 + 65 + 34 + 72 + 32 + 24 + 56 + 44 + 34 + 33 + 80 + 40 + 45 + 55 + 85 + 28 + 27 = 1188
Kromosom Anak 3 : V1,V21,V23,V25,V16,V20,V14,V2,V10,V22,V15,V7,V3,V12,V19,V5,V6,V4,V13,V11,V9,V18,V24,V17,V8,
+ 0 + 43 + 41 + 92 + 91 + 66 + 58 + 13 + 30 + 65 + 34 + 72 + 32 + 24 + 56 + 44 + 34 + 33 + 80 + 40 + 45 + 55 + 85 + 28 + 27 = 1188
Kromosom Anak 4 : V1,V21,V23,V25,V16,V20,V14,V2,V10,V22,V15,V7,V3,V12,V19,V5,V6,V4,V13,V11,V9,V18,V24,V17,V8,
+ 0 + 43 + 41 + 92 + 91 + 66 + 58 + 13 + 30 + 65 + 34 + 72 + 32 + 24 + 56 + 44 + 34 + 33 + 80 + 40 + 45 + 55 + 85 + 28 + 27 = 1188
Kromosom Anak 5 : V1,V21,V23,V25,V16,V20,V14,V2,V10,V22,V15,V7,V3,V12,V19,V5,V6,V4,V13,V11,V9,V18,V24,V17,V8,
+ 0 + 43 + 41 + 92 + 91 + 66 + 58 + 13 + 30 + 65 + 34 + 72 + 32 + 24 + 56 + 44 + 34 + 33 + 80 + 40 + 45 + 55 + 85 + 28 + 27 = 1188
Tsp
1007 ---> V1,V13,V24,V7,V20,V23,V22,V19,V16,V21,V25,V4,V11,V9,V15,V18,V14,V2,V17,V8,V10,V12,V5,V3,V6,
OK


TSP Asimetris Algen Commonality
Jumlah Kota : 25
Pengulangan : 100
Data
Mulai
Log
Matriks
Kromosom 1 : V1,V21,V23,V25,V16,V20,V14,V2,V10,V22,V15,V7,V3,V12,V19,V5,V6,V4,V13,V11,V9,V18,V24,V17,V8,
Log
Matriks
Kromosom 2 : V1,V21,V23,V25,V16,V20,V14,V2,V10,V22,V15,V7,V3,V12,V19,V5,V6,V4,V13,V11,V9,V18,V24,V17,V8,
Log
Matriks
Maximum Partial Order : V1,V21,V23,V25,V16,V20,V14,V2,V10,V22,V15,V7,V3,V12,V19,V5,V6,V4,V13,V11,V9,V18,V24,V17,V8,
Kromosom Anak 1 : V1,V21,V23,V25,V16,V20,V14,V2,V10,V22,V15,V7,V3,V12,V19,V5,V6,V4,V13,V11,V9,V18,V24,V17,V8,
+ 0 + 43 + 41 + 92 + 91 + 66 + 58 + 13 + 30 + 65 + 34 + 72 + 32 + 24 + 56 + 44 + 34 + 33 + 80 + 40 + 45 + 55 + 85 + 28 + 27 = 1188
Kromosom Anak 2 : V1,V21,V23,V25,V16,V20,V14,V2,V10,V22,V15,V7,V3,V12,V19,V5,V6,V4,V13,V11,V9,V18,V24,V17,V8,
+ 0 + 43 + 41 + 92 + 91 + 66 + 58 + 13 + 30 + 65 + 34 + 72 + 32 + 24 + 56 + 44 + 34 + 33 + 80 + 40 + 45 + 55 + 85 + 28 + 27 = 1188
Kromosom Anak 3 : V1,V21,V23,V25,V16,V20,V14,V2,V10,V22,V15,V7,V3,V12,V19,V5,V6,V4,V13,V11,V9,V18,V24,V17,V8,
+ 0 + 43 + 41 + 92 + 91 + 66 + 58 + 13 + 30 + 65 + 34 + 72 + 32 + 24 + 56 + 44 + 34 + 33 + 80 + 40 + 45 + 55 + 85 + 28 + 27 = 1188
Kromosom Anak 4 : V1,V21,V23,V25,V16,V20,V14,V2,V10,V22,V15,V7,V3,V12,V19,V5,V6,V4,V13,V11,V9,V18,V24,V17,V8,
+ 0 + 43 + 41 + 92 + 91 + 66 + 58 + 13 + 30 + 65 + 34 + 72 + 32 + 24 + 56 + 44 + 34 + 33 + 80 + 40 + 45 + 55 + 85 + 28 + 27 = 1188
Kromosom Anak 5 : V1,V21,V23,V25,V16,V20,V14,V2,V10,V22,V15,V7,V3,V12,V19,V5,V6,V4,V13,V11,V9,V18,V24,V17,V8,
+ 0 + 43 + 41 + 92 + 91 + 66 + 58 + 13 + 30 + 65 + 34 + 72 + 32 + 24 + 56 + 44 + 34 + 33 + 80 + 40 + 45 + 55 + 85 + 28 + 27 = 1188
1007 ---> V1,V13,V24,V7,V20,V23,V22,V19,V16,V21,V25,V4,V11,V9,V15,V18,V14,V2,V17,V8,V10,V12,V5,V3,V6,
1009 ---> V1,V13,V24,V15,V21,V19,V6,V22,V16,V4,V11,V18,V5,V20,V17,V2,V8,V10,V12,V9,V23,V25,V14,V7,V3,
1041 ---> V1,V23,V13,V11,V18,V15,V8,V14,V10,V2,V16,V21,V25,V20,V9,V19,V7,V3,V12,V5,V6,V24,V4,V22,V17,
1071 ---> V1,V13,V23,V16,V22,V7,V20,V11,V12,V19,V3,V21,V15,V5,V6,V4,V9,V18,V25,V24,V17,V14,V2,V8,V10,
1094 ---> V1,V13,V24,V19,V5,V20,V23,V14,V2,V8,V17,V10,V12,V21,V15,V22,V25,V9,V16,V4,V6,V18,V11,V7,V3,
1106 ---> V1,V13,V24,V25,V22,V17,V11,V21,V8,V4,V19,V20,V10,V2,V23,V19,V15,V12,V5,V6,V14,V9,V7,V3,V16,
1127 ---> V1,V11,V16,V22,V13,V18,V15,V25,V24,V20,V9,V21,V19,V5,V3,V12,V6,V7,V14,V2,V23,V17,V8,V10,V4,
1142 ---> V1,V22,V23,V11,V21,V25,V4,V20,V13,V9,V16,V15,V18,V14,V24,V2,V17,V8,V10,V12,V7,V3,V19,V5,V6,
1154 ---> V1,V24,V25,V22,V17,V23,V12,V16,V21,V19,V15,V11,V13,V4,V3,V6,V18,V14,V8,V10,V2,V7,V20,V5,V9,
1169 ---> V1,V13,V19,V25,V22,V16,V15,V9,V11,V18,V23,V21,V24,V17,V8,V20,V2,V10,V14,V7,V3,V12,V5,V6,V4,
1169 ---> V1,V13,V23,V25,V15,V21,V4,V18,V16,V17,V14,V11,V20,V2,V8,V10,V19,V7,V3,V12,V5,V6,V22,V9,V24,
1179 ---> V1,V13,V14,V21,V25,V19,V5,V3,V6,V22,V7,V20,V24,V11,V16,V9,V4,V18,V15,V17,V2,V23,V8,V10,V12,
1181 ---> V1,V13,V25,V22,V15,V19,V11,V14,V2,V23,V7,V20,V8,V10,V12,V24,V17,V9,V21,V16,V4,V5,V3,V6,V18,
1188 ---> V1,V21,V23,V25,V16,V20,V14,V2,V10,V22,V15,V7,V3,V12,V19,V5,V6,V4,V13,V11,V9,V18,V24,V17,V8,
1190 ---> V1,V13,V25,V22,V24,V16,V23,V2,V8,V10,V20,V9,V15,V11,V4,V18,V5,V14,V7,V3,V19,V12,V6,V21,V17,
1190 ---> V1,V19,V12,V25,V22,V16,V15,V13,V9,V18,V14,V7,V3,V11,V20,V5,V6,V4,V24,V17,V8,V21,V2,V10,V23,
1194 ---> V1,V13,V25,V7,V19,V8,V5,V20,V24,V21,V14,V17,V2,V10,V16,V15,V11,V4,V12,V9,V3,V6,V18,V22,V23,
1196 ---> V1,V22,V21,V23,V13,V18,V15,V5,V2,V10,V19,V6,V24,V8,V9,V25,V4,V17,V11,V20,V14,V16,V7,V3,V12,
1206 ---> V1,V13,V24,V17,V19,V23,V22,V8,V9,V20,V2,V10,V15,V14,V7,V3,V5,V12,V6,V4,V18,V11,V25,V21,V16,
1218 ---> V1,V23,V25,V14,V7,V20,V15,V9,V22,V4,V16,V11,V13,V18,V21,V2,V10,V8,V24,V17,V19,V3,V12,V5,V6,
1223 ---> V1,V13,V22,V15,V5,V19,V16,V25,V12,V9,V21,V3,V6,V4,V11,V18,V7,V8,V24,V17,V23,V14,V2,V20,V10,
1227 ---> V1,V13,V21,V23,V4,V17,V8,V14,V2,V10,V24,V16,V19,V5,V15,V3,V6,V22,V11,V18,V7,V20,V25,V12,V9,
1227 ---> V1,V13,V22,V23,V14,V2,V20,V8,V17,V10,V24,V5,V15,V19,V12,V6,V4,V11,V21,V9,V18,V7,V3,V25,V16,
1228 ---> V1,V13,V24,V20,V22,V12,V16,V9,V14,V3,V7,V25,V4,V18,V15,V17,V8,V2,V10,V5,V21,V6,V19,V11,V23,
1253 ---> V1,V17,V20,V15,V24,V16,V7,V3,V9,V5,V19,V12,V6,V23,V22,V14,V2,V8,V10,V25,V13,V21,V4,V11,V18,
1255 ---> V1,V13,V11,V21,V25,V8,V12,V5,V6,V4,V22,V20,V18,V15,V2,V17,V10,V24,V23,V9,V19,V16,V14,V7,V3,
1260 ---> V1,V13,V21,V23,V17,V22,V14,V2,V8,V10,V25,V19,V3,V12,V5,V6,V16,V11,V15,V4,V18,V9,V7,V20,V24,
1260 ---> V1,V13,V25,V11,V22,V15,V21,V16,V9,V4,V18,V5,V24,V20,V14,V19,V7,V3,V17,V2,V6,V23,V10,V12,V6,
1261 ---> V1,V13,V22,V25,V23,V21,V9,V5,V6,V16,V4,V11,V18,V15,V17,V20,V14,V10,V2,V12,V19,V8,V24,V7,V3,
1264 ---> V1,V13,V21,V22,V24,V17,V9,V25,V14,V7,V3,V23,V16,V20,V2,V8,V10,V15,V19,V12,V5,V6,V4,V11,V18,
1265 ---> V1,V13,V16,V23,V24,V17,V22,V15,V4,V11,V9,V18,V20,V8,V2,V10,V5,V14,V7,V3,V19,V12,V6,V25,V21,
1268 ---> V1,V13,V25,V23,V12,V9,V17,V11,V21,V4,V18,V15,V19,V24,V7,V5,V2,V8,V10,V3,V6,V20,V16,V14,V22,
1268 ---> V1,V23,V16,V7,V20,V17,V22,V21,V11,V9,V15,V13,V4,V18,V14,V2,V8,V10,V25,V24,V5,V12,V19,V3,V6,
1270 ---> V1,V13,V24,V16,V21,V7,V3,V12,V15,V11,V22,V4,V9,V19,V5,V6,V18,V17,V23,V14,V2,V8,V20,V10,V25,