IMPLEMENTASI *TABU SEARCH* DAN ALGORITMA SEMUT DALAM OPTIMASI PERANCANGAN ULANG BERBASIS GRUP TEKNOLOGI *LAYOUT*

(Studi Kasus di CV. Gambang Emas, Sariharjo, Sleman)

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana
Teknik Industri Fakultas Teknologi Industri

Disusun Oleh:

Nama : Anggha Febrian
No. Mhs : 02 522 266

TEKNIK INDUSTRI
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2007

LEMBAR PENGESAHAN

IMPLEMENTASI *TABU SEARCH* DAN ALGORITMA SEMUT DALAM OPTIMASI PERANCANGAN ULANG BERBASIS GRUP TEKNOLOGI *LAYOUT*

(Studi Kasus di CV. Gambang Emas, Sleman)

TUGAS AKHIR

Oleh

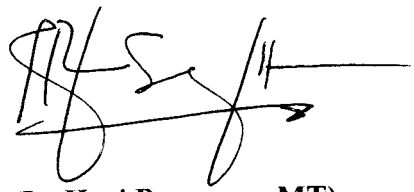
Nama : Anggha Febrian

No. Mhs : 02 522 266

Yogyakarta, 2 Agustus 2007

Menyetujui,

Dosen Pembimbing Tugas Akhir



(Ir. Hari Purnomo, MT)

PERSEMBAHAN

Ku persembahkan karya kecil ini untuk :

Ayah dan Bunda tercinta **(Saby Sabirin & Mahrani)** yang mengajarkan memaknai kehidupan dengan Cinta yang tak pernah berhenti mengalir dan doa yang tiada akhir.

Abang **(Ancha Wardhana, SE)** dan adik-adikku **(Aghid Septian & Astrid Ayang Nabila)** yang telah menjadi kekuatan untuk menjalani segala kemungkinan dalam kehidupan.

Dan untuk ***Inspirasi*** yang tiada pernah berhenti mengalir ....

- Motto -

" Maka nikmat Tuhanmu manakah yang kamu dustakan?"

(*Ar - Rahmaan*)

" Karena sesungguhnya setelah kesulitan ada kemudahan "

(*Al- Insyirah* : 5 - 6)

Karena Usaha adalah raga dan Doa adalah jiwa yang membuat manusia menjadi hidup

KATA PENGANTAR

بسم الله الرحمن الرحيم

Assalamu'alaikum, Wr. Wb

Dengan memanjatkan puji syukur kehadirat Allah SWT yang telah memberikan kekuatan dan petunjuk sehingga penulis dapat menyeleaikan Tugas Akhir dengan judul "Studi Komparasi Implementasi Algoritma *TabuSearch* dan Algoritma Semut Dalam Optimasi Perancangan Ulang Berbasis Grup Teknologi *Layout* (Studi Kasus di CV. Gambang Emas, Keceme, Sariharjo, Sleman)" dengan baik.

Adapun Tugas Akhir ini dilaksanakan sebagai persyaratan untuk menyelesaikan jenjang strata satu (S1) di jurusan Teknik Industri, Fakultas Teknologi Industri, Universitas Islam Indonesia.

Penulis banyak menemui kesulitan dan hambatan dalam menyelesaikan tugas akhir ini. Namun berkat bantuan dan bimbingan dari berbagai pihak akhirnya halangan maupun rintangan ini dapat penulis atasi dengan baik. Untuk itu tidak berlebihan kiranya jika pada kesempatan ini penulis menyampaikan banyak terima kasih kepada :

1. Orang Tua yang selalu memberi semangat,dorongan dan doa.
2. Bapak Ir. Hari Purnomo, MT selaku Dosen Pembimbing tugas akhir yang banyak memberikan masukan dan bimbingan selama tugas akhir ini.
3. Dekan Fakultas Teknologi Industri, Universitas Islam Indonesia.

4. Ketua dan Sekretaris Jurusan Teknik Industri, Universitas Islam Indonesia.
5. Seluruh karyawan CV. Gambang Emas, Sleman yang telah membantu dalam pelaksanaan penelitian di perusahaan.
6. Semua pihak yang telah membantu dalam penyusunan tugas akhir ini yang tidak dapat disebutkan satu persatu

Penulis menyadari bahwa tugas akhir ini masih jauh dari sempurna, walaupun demikian penulis berharap semoga apa yang sudah penulis ketengahkan ini bisa bermanfaat bagi semua pihak, dan semoga seluruh bantuan yang telah disumbangkan kepada penulis dapat diterima Allah SWT sebagai amal sholeh dan dibalasnya dengan pahala besar.

Wassalamu'alaikum, Wr. Wb

Yogyakarta, Agustus 2007

Anggha Febrian

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAKSI

Untuk dapat memenangkan persaingan dalam pasar global maka perusahaan perlu meningkatkan fleksibilitas kebutuhan dan efisiensi dalam produksi. Pembentukan Cellular Manufacturing System *(CMS) dalam desain tata letak pabrik menjadi salah satu keunggulan strategi dalam perkembangan industri yaitu dapat mengurangi jarak dan biaya* material handling. *Dalam penelitian ini dibandingkan dua macam teknik pembentukan sel manufaktur secara heuristik yakni antara Algoritma* Tabu Search dan *Algoritma Semut* (Ant Algorithm). *Kedua Algoritma ini masing-masing memiliki dua tahap penyelesaian, yaitu tahap pertama untuk menentukan urutan mesin yang optimal dan tahap kedua untuk menentukan urutan* part *yang optimal. Pengolahan dengan Algoritma Semut menghasilkan pengelompokan dua sel manufaktur, dengan nilai pengelompokan efisiensi sebesar 51,6 % dan nilai kekuatan pengelompokan* (efficacy) *sebesar 47,6%. Algoritma ini juga menghasilkan* layout *usulan yang dapat mengurangi jarak total* material handling *sebesar 41,74 meter dan pengurangan biaya* material handling *sebesar Rp. 16.903,21 / bulan. Sementara pada hasil pengolahan data Algoritma* Tabu Search *juga menghasilkan pengelompokan dua sel manufaktur, dengan nilai pengelompokan efisiensi sebesar 73,33% dan nilai kekuatan pengelompokan* (efficacy) *sebesar 61,45%. Algoritma* Tabu Search *juga mengurangi jarak total* material handling *sebesar 110,35 meter dan pengurangan biaya* material handling *sebesar Rp. 74.712,00 / bulan. Dengan demikian, pada penelitian ini penggunaan Algoritma* Tabu Search *lebih optimal dibandingkan penggunaan Algoritma Semut dalam pembentukan Sel Manufaktur yang berbasis Grup Teknologi dengan selisih jarak* material handling *sebesar 68,82 meter dan biaya sebesar Rp. 27.809,00 / bulan.*

LEMBAR PENGESAHAN PENGUJI

Implementasi *Tabu Search* dan Algoritma Semut Dalam Optimasi Perancangan Ulang Berbasis Grup Teknologi *Layout*

(Studi Kasus CV. Gambang Emas, Sariharjo, Sleman)

TUGAS AKHIR

Oleh :

Nama : Anggha Febrian
No. Mhs : 02 522 266

Telah Dipertahankan di Depan Sidang Penguji sebagai Salah Satu Syarat untuk Memperoleh Gelar Sarjana Teknik Industri, Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 21 September 2007

Tim Penguji,

Hari Purnomo, Ir, MT.
Ketua

Hartomo, Ir, M.Sc.
Anggota 1

Imam Djati. W, Drs, M.Eng.Sc.
Anggota 2

Mengetahui,
Ketua Jurusan Falkutas Teknologi Industri
Universitas Islam Indonesia

R. Chairul Saleh, Ir, M.Sc, Ph.D

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Saat ini di era globalisasi dalam dunia industri daya saing dan kemampulabaan suatu perusahaan tergantung pada fleksibiltas, produktifitas, dan kualitas produk sesuai dengan keinginan pasar. Untuk dapat memenangkan persaingan baik di pasar dalam negeri maupun luar negeri, perusahaan harus dapat meningkatkan kemampuan bersaingnya. Kemampuan bersaing sebuah perusahaan dapat dilakukan dengan cara menurunkan biaya manufaktur, meningkatkan kualitas hasil produksi, serta menjaga ketepatan waktu pengiriman. Disamping itu perusahaan juga harus memahami keadaan, perannya serta strategi apa yang harus dijalankan untuk menghadapi kondisi di era globalisasi. Keunggulan bersaing suatu perusahaan merupakan hasil dari berbagai kompetensi perusahaan di bidang produksi, pemasaran, keuangan dan jaringan bisnis yang luas.

Dengan adanya faktor-faktor tersebut diatas akan menjadikan persaingan manufaktur yang sekarang ini cenderung meningkat disamping pemicu lain yaitu keinginan konsumen yang meningkat dan berubah-ubah. Dengan perubahan jumlah kuantitas dari produk dan variasi produk maka perlu strategi manufaktur untuk menghadapi tantangan dimasa yang akan datang.

Keunggulan proses manufaktur salah satunya diawali dengan pemilihan desain tata letak pabrik dimana mempertimbangkan kesesuaiannya dengan

karakteristik produk yang akan diproduksi. Agar perusahaan dapat bertahan dalam persaingan, perusahaan membutuhkan fleksibilitas yang tinggi. Fleksibilitas yang dimaksud adalah kemampuan untuk memenuhi variasi sesuai permintaan konsumen dengan jumlah yang diinginkannya.

Untuk dapat memenuhi kriteria efisiensi dan fleksibilitas yang tinggi, lebih tepat kiranya jika perusahaan menerapkan tata letak pabrik bertipe *product family / group technology / cellular manufacturing layout*. Desain *layout* tersebut bertujuan untuk mengurangi waktu *setup*, aktivitas penanganan material, waktu *throughput*, inventori *in-process*, kebutuhan ruangan, waktu *idle* mesin, dan kompleksitas kontrol, yang pada gilirannya akan meningkatkan efisiensi produksi.

Seperti halnya tipe tata letak fasilitas yang lain, tipe tata letak fasilitas berdasarkan kelompok produk *(Group Technology Layout)* ini juga memiliki kelebihan-kelebihan yang harus dimaksimalkan dan kekurangan-kekurangan yang harus diminimalisir agar produksi dapat berjalan dengan optimal. Salah satu keunggulan yang akan dimaksimalkan pada penelitian ini adalah pada kombinasi penyusunan urutan mesinnya untuk semakin memperkecil jarak *material handling* dari produk. Adapun kombinasi penyusunan urutan mesin ini memiliki karakteristik yang mirip dengan permasalahan *Travelling Salesman Problem* dalam penentuan rute terpendek jarak antar kota..

Untuk itu dibutuhkan pengembangan algoritma-algoritma yang bertujuan untuk mengatasi kelemahan metode konvesional yang membutuhkan ruang pencarian yang sangat besar, yaitu sebesar faktorial dari banyaknya kota. Tanpa penerapan algoritma sistem cerdas, pembentukan sel manufaktur dengan metode

konvensional pasti membutuhkan ruang pencarian yang besar sekali.

Beberapa algoritma sistem cerdas sudah dikembangkan untuk pembentukan sel manufaktur, seperti *Tabu Search* (TS), Jaringan Syaraf Tiruan (JST), Algoritma Genetik (AG), Algoritma Semut (AS), *dan Simulated Annealing* (SA). Sedangkan pada penelitian ini akan menggunakan dua buah algoritma yakni Algoritma *Tabu Search* dan Algorima Semut sebagai alat bantu untuk menyelesaikan persoalan sel manufaktur. Dipilihnya kedua algoritma dalam penelitian ini adalah mengingat kedua algoritma tersebut memang dikembangkan untuk optimasi masalah TSP.

Penelitian ini akan membandingkan penelitian terdahulu yaitu Aplikasi Algoritma Semut dalam Pembentukan *Cellular Manufacturing System* (Rahadiyanto Suryoko Dhanasworo, 2006) dan Aplikasi Algoritma *Tabu Search* dalam Pembentukan *Cellular Manufacturing System* (Ade Maulana, 2005) yang keduanya menekankan dan bertujuan untuk Meminimasi Biaya *Material Handling*. Selain itu Penelitian ini juga melanjutkan kedua penelitian di atas dimana penelitian ini akan memperhitungkan jumlah mesin yang optimal pada sistem produksi perusahaan sehingga dapat meningkatkan efisiensi dan efektifitas.

1.2 Perumusan Masalah

1. Seberapa besar selisih jarak dan biaya *material handling* antara *layout* awal dan *layout* sesudah perancangan kembali tata letak fasilitas dengan dengan menggunakan Algoritma Semut?

2. Seberapa besar selisih jarak dan biaya *material handling* antara *layout* awal dan *layout* sesudah perancangan kembali tata letak fasilitas dengan dengan menggunakan *Tabu Search*?
3. Seberapa besar selisih jarak dan biaya *material handling* antara *layout* Algoritma Semut dan *layout* Algoritma *Tabu Search* sesudah perancangan kembali tata letak fasilitas?

1.3 Batasan Masalah

Agar penelitian ini lebih dapat terarah, mudah dipahami serta topik yang dibahas tidak meluas, maka perlu dilakukan pembatasan lingkup penelitian.

1. Kebutuhan luas area dianggap tetap.
2. Biaya penambahan mesin tidak diperhitungkan.
3. Total jarak perpindahan *part* adalah mulai dari mesin yang mengerjakan proses pertama dan berakhir di lokasi mesin yang mengerjakan proses terakhir dari *part* tersebut, tanpa kembali lagi ke lokasi pertama.

1.4 Tujuan Penelitian

Tujuan dilakukannya penelitian ini adalah :

1. Untuk mengetahui seberapa besar selisih jarak dan biaya *material handling* antara *layout* awal dan sesudah perancangan kembali tata letak fasilitas dengan dengan menggunakan Algoritma Semut

2. Untuk mengetahui seberapa besar selisih jarak dan biaya *material handling* antara *layout* awal dan *layout* sesudah perancangan kembali tata letak fasilitas dengan dengan menggunakan Algoritma Semut
3. Untuk mengetahui seberapa besar selisih jarak dan biaya *material handling* antara *layout* awal dan *layout* sesudah perancangan kembali tata letak fasilitas dengan dengan menggunakan Algoritma Semut

1.5 Manfaat Penelitian

Manfaat yang dapat diambil dari hasil penelitian ini adalah:

1. Bagi perusahaan dapat meningkatkan produktivitas dan mengurangi biaya produksi.
2. Penguasaan beberapa aspek keilmuan bagi peneliti seperti manajemen, pengambilan keputusan, produksi serta komputerisasi

1.6 Sistematika Penulisan

Sistematika dari penulisan tugas akhir ini dilanjutkan sebagai berikut :

BAB II. LANDASAN TEORI

Bab ini berisikan landasan teori-teori dasar tentang masalah penelitian, penjelasan mengenai konsep – konsep dasar mengenai permasalahan yang diangkat serta mendukung penelitian yang akan dilakukan

BAB III. METODOLOGI PENELITIAN

Bab ini berisikan penjelasan mengenai obyek penelitian, tempat dan waktu penelitian, teknik pengumpulan data dan kerangka pemecahan masalah.

BAB IV. PENGUMPULAN DAN PENGOLAHAN DATA

Bab ini berisikan data – data yang diperlukan dalam penelitian, pengolahan data tersebut, baik secara langsung maupun tidak dengan bantuan software.

BAB V. PEMBAHASAN

Bab ini membahas hasil penelitian berupa tabel hasil pengolahan data, grafik serta analisa yang menyangkut penjelasan teoritis secara kualitatif, kuantitatif maupun statistik dari hasil penelitian.

BAB VI. PENUTUP

Kesimpulan, memuat pernyataan singkat dan tepat yang dijabarkan dari hasil penelitian dan pembahasan untuk membuktikan atau menjawab permasalahan.

Saran, dibuat berdasarkan pengalaman dan pertimbangan penulis, ditujukan kepada para peneliti (perusahaan) dalam bidang yang sejenis.

DAFTAR PUSTAKA

LAMPIRAN

BAB II

KAJIAN LITERATUR

2.1 Kajian Induktif

Pada bagian ini akan menjelaskan mengenai kajian-kajian literatur yang diperoleh dari penelitian sebelumnya. Dimas Ardiansyah (2002), Farida Dharmayantie (2002), Prestiani Dellyanti (2003), Masrikun (2003), Moch.Dwi Kurniawan (2003), Eko Purnomo (2003), Ida Jamilah (2003), dan R.A.Fitria Miranti (2003) melakukan penelitian tentang penerapan *Algoritma Heuristik* dalam *Cellular Manufacturing System (CMS)* dengan menggunakan metode *Bond Energy Algorithm (BEA), Rank Ordering Clustering I dan II (ROC I dan II)* yang bertujuan untuk mengurangi jarak, waktu perpindahan, biaya *material handling* dan waktu siklus. Dari beberapa metode tersebut diambil yang terbaik berdasarkan *performance measure*.

Dani Hendarto (2002), Sri Purwoningsih (2004), Dian Ayudya Sandjojo (2004), dan Yenni Kusumawati (2005).melakukan penelitian mengenai pembentukan sel manufaktur dengan membandingkan *algoritma heuristik* diatas dengan menggunakan metode yang berbeda, yaitu *Single Linkage Clustering (SLC), Complete Linkage Clustering (CLC), dan Average Linkage Clustering (ALC)*. Metode ini mencari jarak minimun kemudian membentuk *cluster* berdasarkan kemiripan yang dimiliki mesin dan *part*.

Aldita Ekasari (2004) dan Firman Paradisi (2005) melakukan penelitian dengan cara membandingkan beberapa metode diatas untuk dicari yang terbaik, kemudian dilakukan pengembangan sebuah model matematis P-Median. Penelitian ini bertujuan untuk mengurangi jarak *material handling* dan waktu siklus.

Fauzi Leilan (2005), melakukan pembentukan sel manufaktur *part* dan mesin menggunakan aplikasi pembentukan formasi secara Virtual. Penelitian tentang *Cellular Manufacturing System (CMS)* yang menggunakan teknik heuristik antara lain dilakukan oleh Ade Miranda (2002) menggunakan algoritma genetik dalam pembentukan *Cellular Manufacturing System (CMS)* untuk meminimalkan jarak *material handling*. Tujuannya adalah untuk memperoleh tata letak fasilitas yang memiliki tingkat efisiensi yang lebih baik daripada layout yang sudah ada dan jarak serta biaya *material handling* yang lebih rendah.

Ade Maulana (2005) menggunakan *Algoritma Tabu Search* dalam pembentukan sel manufaktur untuk meminimasi biaya *material handling*. Sedangkan Kamal Ardly (2005) menggunakan Algoritma Genetik. Fajar Priyambada (2005) menggunakan Algoritma *Simulated Annealing* untuk tujuan yang sama.

Penelitian yang dilakukan penulis ini akan mengembangkan model perangkat lunak (*Software*) dengan teknik kecerdasan buatan (*Artificial*

Intelligent), yaitu menggunakan Algoritma Semut (*Ant Algorithm*) dan Algoritma *Tabu Search* untuk menyelesaikan masalah pembentukan sel manufaktur dengan tujuan meminimasi jarak dan biaya *material handling*.

2. 2 Kajian Deduktif

2.2.1. Konsep Dasar Tata Letak Pabrik

2.2.1.1. Pengertian Dasar Tata Letak Pabrik

Tata letak pabrik yaitu suatu tata cara pengaturan fasilitas-fasilitas pabrik guna menunjang kelancaran dari segala proses industri (Wignjosubroto.S, 1996). Dapat juga diartikan sebagai susunan dari mesin-mesin dan peralatan produksi di dalam suatu pabrik. Tujuan utama dari tata letak pabrik adalah mengatur area kerja yang tersedia dan segala fasilitas produksi yang ekonomis untuk operasi produksi, aman dan nyaman sehingga akan menaikkan moral kerja dan performansi dari operator.

2.2.1.2. Macam-Macam Tata Letak Pabrik

Beberapa macam *layout* fasilitas (mesin / peralatan), antara lain sebagai berikut (Purnomo, 2004) :

a. *Product layout*

Adalah metode pengaturan dan penempatan semua fasilitas produksi yang diperlukan kedalam suatu departemen tertentu. Layout ini mesin-mesin atau alat bantu disusun menurut urutan proses dari suatu produk. Produk-

produk bergerak secara terus menerus dalam suatu garis perakaitan. Layout ini digunakan apabila volume produksi cukup tinggi dan variasi produk yang tidak banyak dan sesuai untuk produksi yang kontinyu.

b. *Process / functional layout*

Adalah metode pengaturan semua fasilitas produksi yang mempunyai sifat yang sama dikelompokkan ke dalam suatu departemen yang sama. Dengan kata lain material (bahan baku) dipindah menuju departemen-departemen sesuai dengan urutan proses yang dilakukan.

c. *Product family / Group Technology layout*

Tipe tata letak ini biasanya komponen yang tidak sama dikelompokkan ke dalam satu kelompok berdasarkan kesamaan bentuk komponen, mesin atau peralatan yang dipakai. Pengelompokan bukan didasarkan pada kesamaan penggunaan akhir. Mesin-mesin dikelompokkan dalam suatu kelompok dan ditempatkan dalam sebuah "*manufacturing cell*".

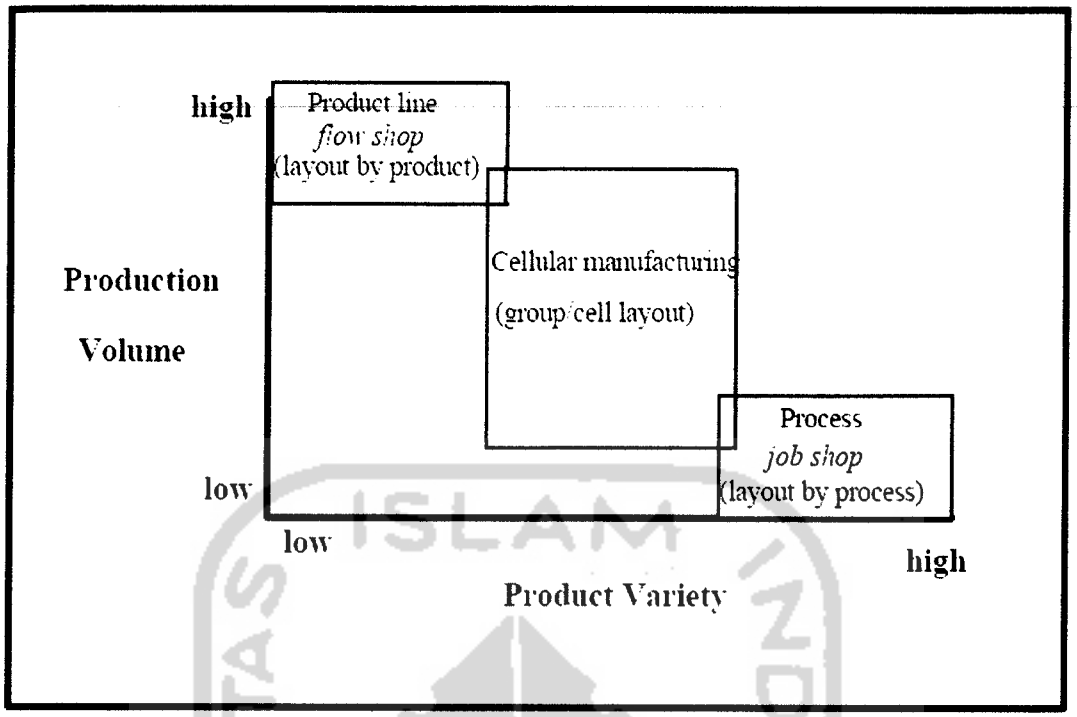
d. *Fixed Position Lay Out*

Pada tata letak fasilitas berdasarkan *product layout* maupun *process layout*, produk bergerak menuju mesin sesuai dengan urutan proses yang dibutuhkan. *Fixed position layout* diartikan sebagai suatu tipe tata letak fasilitas dimana mesin, manusia serta komponen-komponen kecil bergerak menuju lokasi material untuk menghasilkan suatu produk. *Layout* ini biasanya digunakan untuk memproses produk yang relatif besar dan berat sedangkan peralatan yang digunakan ringan dan mudah untuk dipindahkan.

2.2.2 Group Technology

Group Technology (GT) adalah sebuah filosofi dalam dunia manufaktur yang mengidentifikasi dan mengelompokkan *part-part* yang serupa ke dalam kelompok *part* (*part family*) dengan memanfaatkan kesamaan dalam hal rancangan produk dan proses fabrikasi dalam siklus manufaktur. GT bertujuan untuk mengurangi waktu *setup*, aktivitas penanganan material, waktu *throughput*, inventori *in-process*, kebutuhan ruangan, waktu *idle* mesin, dan kompleksitas kontrol, yang pada gilirannya akan meningkatkan efisiensi produksi.

Pendekatan *Group Technology* dalam sistem manufaktur pertama kali diperkenalkan oleh Mitrofanov (1966) dan Burbidge (1975). Beberapa tahun terakhir ini pendekatan tersebut mendapat perhatian yang cukup besar sebagai alternative untuk system produksi dengan jenis *part* yang tingkat variasinya cukup tinggi serta sistem produksi berbasis *batch*.



Gambar 2.1 Hubungan Volume Produksi dengan Variasi Produk (Morad, 2000)

Group Technology merupakan penggabungan antara *flowshop* proses dengan *job shop* proses. Berdasarkan gambar 2.1 diatas *cellular manufacturing* digunakan jika variasi sedang dan volume produksi sedang.

Terdapat beberapa hal yang dapat dianggap sebagai impuls timbulnya pemikiran *Group Technology* :

a. Dengan perkembangan teknologi, variasi produk yang dapat dihasilkan akan semakin beragam dengan ukuran lot yang semakin kecil dan daur hidup produk yang semakin singkat.
b. Dengan sistem manufaktur konvensional (*product layout* dan *process layout*) kondisi seperti ini timbul :
    a) Effisiensi tinggi pada *product layout*.

b) Fleksibilitas tinggi pada *process layout*.

Secara teknis pengaturan mesin dalam *Group Technology* ada tiga kategori yaitu :

1) Group Technology Flow Line Layout.

Tipe *layout* ini digunakan ketika semua komponen pada group mengikuti aturan mesin yang sama. Mekanisme transfer otomatis terkadang digunakan untuk penanganan komponen dalam group.

2) Group Technology Cell Layout

Pada *Group Technology Cell Layout* mengijinkan untuk bergerak / pindah dari satu mesin ke mesin lainnya. Hal ini sangat berbeda dengan *Group Technology Flow Line Layout* dimana komponen-komponen didalam group mengikuti urutan mesin yang sama. Aliran komponen dimungkinkan tidak terarah.

3) Group Technology Center Layout

Tipe *layout* ini didasarkan pada penyusunan mesin. Penyusunan ini dapat meningkatkan perubahan penanganan material dan sesuai pada saat *product mix* sering diubah.

Keuntungan dan kerugian dari *Group Technology* adalah (Purnomo, 2004):

1. Keuntungan

a. Dapat mengurangi pemborosan waktu dalam perpindahan antar kegiatan yang berbeda.

b. Dapat mengurangi waktu setup, ongkos *material handling*, dan area lantai produksi.

c. Apabila ada urutan proses yang terhenti maka dapat dicari alternatif lainnya.

d. Mudah mengidentifikasi "*bottlenecks*" dan cepat merespon perubahan jadwal.

e. Operator makin terlatih, cacat produk dapat dikurangi, dan dapat mengurangi bahan yang terbuang.

2. Kerugian

a. Utilisasi mesin yang rendah.

b. Memungkinkan terjadinya duplikasi mesin.

c. Biaya yang cukup tinggi untuk relokasi mesin.

d. Membutuhkan tingkat kedisiplinan yang tinggi dikarenakan ada kemungkinan part yang diproses berada pada sel yang salah.

2.2.3 *Cellular Manufacturing System*

2.2.3.1 Pengertian Sel Manufaktur

Istilah manufaktur berasal dari dua kata dalam bahasa Latin, yaitu *manus* yang berarti tangan dan *facere* yang berarti membuat. Jadi istilah manufaktur dapat diartikan sebagai suatu proses dimana material diubah dari satu keadaan ke keadaan baru (yang lebih bernilai) melalui suatu kerja , atau dalam pengertian yang lebih luas, proses perubahan material dan informasi menjadi barang-barang

untuk memenuhi kebutuhan umat manusia. Dalam beberapa tahun terakhir telah terjadi pergeseran yang radikal sifatnya dalam perancangan sistem perencanaan dan pengendalian manufaktur dengan melibatkan konsep-konsep yang inovatif sifatnya seperti Sistem Produksi Tepat Waktu (*Just-in-Time Production System*), Teknologi Produksi Teroptimasi (*Optimised Production Technology*), Sistem Produksi Ramping (*Lean Production System*), Manufaktur Kelas Dunia (*World Class Manufacturing*), Pengendalian Mutu Terpadu (*Total Quality Management*), Manufaktur yang terintegrasi oleh komputer (*Computer Integrated Manufacturing*), Sistem Manufaktur yang Fleksibel (*Flexible Manufacturing Systems*), serta *Group Technology* atau *Cellular Manufacturing System (CMS)*.

Sel manufaktur adalah sistem penataan mesin dan komponen yang dikembangkan untuk mengatasi kelemahan sistem penataan yang telah berkembang sebelumnya, yaitu *flowshop layout* dan *job shop layout* (Purnomo, 2004). *Cellular manufacturing* muncul sebagai sebuah strategi yang memungkinkan penyelesaian masalah yang kompleks dan dengan *lead times* yang panjang dalam suatu proses produksi.

2.2.3.2 Konsep Dasar *Cellular Manufacturing System*

Cellular manufacturing System (CMS) merupakan salah satu aplikasi dari GT. Ide yang mendasari CMS adalah pengelompokan mesin ke dalam sel-sel untuk memproduksi *part family*, yaitu sekelompok *part* yang membutuhkan proses-proses manufaktur yang serupa. CMS mendekomposisi suatu sistem produksi ke dalam beberapa sub sistem, yang disebut sel mesin (*machine cell*),

dimana dalam tiap sel dapat diproses satu atau beberapa *part family* secara penuh tanpa melakukan perpindahan antar sel.

Cellular Manufacturing System merupakan strategi manajemen untuk memenangkan persaingan global dengan cara mengurangi biaya produksi, memperbaiki kualitas dan mengurangi lead time pengiriman produk dalam tingkat variasi tinggi dan permintaan rendah.

Proses pembentukan sel mesin memanfaatkan matriks keterhubungan (*incidence matrix*) mesin-komponen, yang dibentuk dari informasi yang terdapat pada *part routing sheet*. Entri ke-(*i,j*) dalam matriks akan bernilai 1 jika komponen ke-*j* diproses oleh mesin ke-*i*; jika tidak, nilainya 0 atau kosong. Matriks ini dimanipulasi sampai diperoleh bentuk diagonal blok yang baik. Contoh *initial matrix* untuk 5 buah komponen dan 5 buah mesin adalah sebagai berikut :

		Komponen				
		1	2	3	4	5
Mesin	1	0	1	1	0	1
	2	1	1	0	1	0
	3	1	0	1	1	1
	4	1	0	0	0	1
	5	0	1	1	1	0

Gambar 2.2 *Incidence Matrix* 5 komponen dan 5 mesin

Pengelompokan *family part* dan mesin-mesin dilakukan dengan cara mengubah-ubah urutan *komponen* dan mesin-mesin dalam *incidence matrix* sehingga terbentuk blok-blok yang berisi seluruhnya atau sebagian besar angka 1, dan *range-range* diluar blok diusahakan seluruhnya atau sebagian besar angka 0.

Angka 0 di dalam blok menyebabkan sel yang terbentuk kurang efektif (terjadi *intracellular*), sedangkan angka 1 diluar blok menyebabkan *part* mengalami *traveling* yang terlalu jauh (terjadi *intercellular*).

Sehingga untuk mendapatkan solusi pembentukan sel manufaktur yang mendekati sempurna (*a near perfect decomposition*) harus dipertimbangkan satu diantara dua tujuan berikut ini :

1. Meminimalkan total `0` yang ada di dalam sel manufaktur (*voids*)
2. Meminimalkan total `1` yang ada di luar sel manufaktur (*exceptional parts*).

Terkait dengan tujuan tersebut maka dipergunakan *performance measures* untuk menghitung performa sel manufaktur yang terbentuk.Ada tiga macam jenis *performance measures* yang biasa digunakan, yaitu (Singh dan Rajawani, 1995) :

1. Grouping Efficiency η

Kegunaan dari jenis pengukuran ini adalah untuk mengetahui keefisienan dari fungsi/kegunaan mesin dalam sebuah sel dan pergerakan inter sel. Adapun formulanya adalah :

$$\eta = w\eta_1 + (1\text{-}w)\eta_2 \quad \text{.....1)}$$

$$\eta_1 = \frac{o-e}{o-e+v} \quad \text{.....2)}$$

$$\eta_2 = \frac{MP-o-v}{MP-o-v+e} \quad \text{.....3)}$$

2. Grouping Efficacy τ

Kegunaan dari jenis pengukuran ini adalah untuk mengetahui kelebihan

sebuah diskriminasi yang rendah dari *grouping efficiency* antara struktur matrik yang bagus dan yang jelek.

Formulanya :

$$\tau = \frac{o - e}{o + v} \quad \text{.....4)}$$

3. *Grouping Measure*

Kegunaannya adalah untuk mengukur keefektifan dari matrik final, jika hasilnya tinggi maka utilitas mesin juga tinggi.

Formulanya :

$$\eta_g = \eta_u - \eta_m \quad \text{.....5)}$$

$$\eta_u = \frac{d}{(d = v)} \quad \text{.....6)}$$

$$\eta_m = 1 - (d / o) \quad \text{..... 7)}$$

Dimana:

M = Jumlah mesin

P = Jumlah part / komponen

o = Jumlah angka 1 dalam matrik keseluruhan

e = Jumlah angka 1 diluar kelompok (*exceptional*)

v = Jumlah angka 0 dalam kelompok (*voids*)

η_1 = Perbandingan angka 1 dalam kelompok terhadap angka 0 dan 1 dalam kelompok

η_2 = Perbandingan angka 0 diluar kelompok terhadap angka 0 dan 1 diluar kelompok

w = Bobot, dalam perhitungan direkomendasikan bernilai 0,5

d = Jumlah angka 1 dalam blok diagonal.

η_{II} = Pengukuran pemrosesan *part* dan sel *part* mesin.

η_{III} = Pengukuran pergerakan *part* antara dua sel.

2.2.3.3 Pembentukan Sel Manufaktur

Hal terpenting dari *Cellular Manufacturing Layout* adalah membuat *layout* pada lantai produksi menjadi lebih efektif. Mesin-mesin dilantai produksi dikelompokkan untuk membuat suatu komponen tertentu. Dalam *Cellular Manufacturing System* kita mengelompokkan *part families* berdasarkan kesamaan proses dan mesin. Prosedur sistematis yang menganalisa informasi dari rute proses pembuatan suatu komponen adalah dengan menggunakan *incident matrix*.

Dalam *incident matrix* terdapat dua nilai yang digunakan, yaitu 0 dan 1. Nilai 1 digunakan bila dalam pembuatan komponen menggunakan mesin, dan nilai 0 digunakan bila dalam pembuatan komponen tidak menggunakan mesin.

Langkah-langkah penentuan *incident matrix* adalah sebagai berikut :

1. Daftar semua komponen yang akan dibuat secara horisontal.
2. Daftar semua mesin yang akan digunakan untuk memproses komponen tersebut secara vertikal.
3. Isikan angka 1 pada koordinat pertemuan antara komponen dengan mesin bila komponen yang bersangkutan diproses di mesin tersebut.

	Machine											
Product	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
1	1			1								
2		1							1			
3			1			1		1				
4	1			1							1	
5			1			1		1				
6	1			1							1	
7			1					1				
8		1							1			
9			1			1		1				
10					1					1		1
11		1							1			
12				1							1	
13	1										1	
14					1		1					1
15					1		1			1		1

Gambar 2. 3 Matrix Awal (Goncalves and Resende, 2002)

	Machine											
Product	M4	M8	M3	M5	M7	M10	M12	M1	M4	M11	M2	M9
3	1	1	1									
5	1	1	1									
7		1	1									
9	1	1	1									
10				1		1	1					
14				1	1		1					
15				1	1	1	1					
1								1	1			
4								1	1	1		
6								1	1	1		
12									1	1		
13								1		1		
2											1	1
8											1	1
11											1	1

Gambar 2. 4 Matrix Akhir (Goncalves and Resende, 2002)

Cells	Machines	Products
1	M3,M6,M8	3,5,7,9
2	M5,M7,M10,M12	10,14,15
3	M1,M4,M11	1,4,6,12,13
4	M2,M9	2,8,11

Gambar 2.5 Hasil Pengelompokkan Produk dan Mesin dari Gambar 2.3 dan 2.4

2.2.3.4 Metode Pembentukan Sel Manufaktur Yang Digunakan

Ada beberapa metode heuristik yang yang didasarkan pada penggunaan incident matrix yang biasa digunakan untuk membentuk sel manufaktur antara lain adalah algoritma *Tabu Search*, Algoritma Genetik, Algoritma *Simulated Annealing*, Jaringan Syaraf Tiruan (*Neural Network*) dan Algoritma Semut. Pada penelitian ini pembentukan sel manufaktur menggunakan dua jenis algoritma yaitu Algoritma Semut dan Algoritma *Tabu Search*.

2.2.4 Perhitungan Jarak Antar Mesin

Terdapat beberapa macam sistem yang dipergunakan untuk melakukan pengukuran jarak suatu lokasi terhadap lokasi lain (Purnomo, 2004), yaitu :

1. *Euclidean*

   Jarak Euclidean merupakan jarak yang diukur lurus antara pusat fasilitas satu dengan pusat fasilitas lainnya, dengan persamaan :

$$d(A,B) = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2} \quad \text{.....8)}$$

Dimana :

$A(x_a, y_a)$ = lokasi mesin A

$B(x_b, y_b)$ = lokasi mesin B

2. *Rectilinear*

Jarak rectilinear juga sering disebut dengan jarak Menhattan, merupakan jarak yang diukur mengikuti jalur tegak lurus. Disebut jalur Menhattan, karena mengingat jalan-jalan dikota Menhattan yang berbentuk garis-garis paralel dan saling tegak lurus antara satu jalan dengan jalan lainnya. Sering digunakan karena mudah perhitungannya, mudah dimengerti dan untuk beberapa masalah lebih sesuai, misalkan untuk menentukan jarak antar kota, jarak antar mesin dimana peralatan pemindahan bahan hanya dapat bergerak secara tegak lurus. Formula untuk menghitung jarak ini adalah :

$$dij = |x_i - x_j| + |y_i - y_j| \quad \text{.....9)}$$

3. *Square Euclidean*

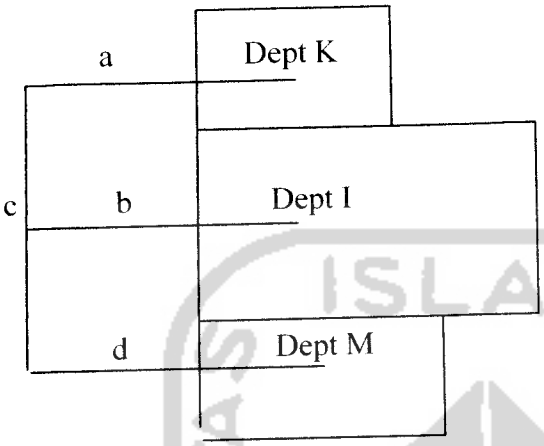
Sebagaimana namanya, *square euclidean* merupakan ukuran jarak dengan mengkuadratkan bobot terbesar suatu jarak antara mesin yang berdekatan. Relatif untuk beberapa persoalan terutama menyangkut persoalan lokasi fasilitas diselesaiakan dengan penerapan *Square Euclidean*. Formulasinya adalah :

$$dij = |(x_i - x_j)^2 + (yi - y_j)^2| \quad \text{.....10)}$$

4. *Aisle*

Jarak *aisle* mengukur jarak sepanjang lintasan yang dilalui alat pengangkut

pemindahan bahan. Misal pada gambar dibawah ini ukuran jarak aisle departemen K dan M adalah a, b dan d

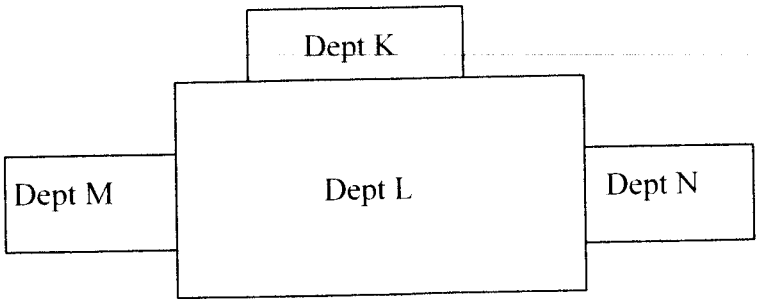


Gambar 2.6 Perhitungan Jarak untuk *Aisle Distance*

5. *Adjacency*

Adjacency merupakan ukuran kedekatan antara fasilitas-fasilitas atau departemen-departemen yang terdapat dalam suatu perusahaan. Biasanya digunakan untuk mengukur kedekatan antar departemen. Kelemahan ukuran jarak *adjacency* adalah tidak dapat memberi perbedaan secara riil jika terdapat dua pasang fasilitas dimana satu dengan yang lainnya tidak berdekatan.

Sebagai contoh gambar dibawah ini, jarak antara departemen K dan N yang tidak saling berdekatan berjarak 40 m, jika jarak departemen M dan N adalah 75 m, maka bukan berarti departemen K dan N mempunyai kedekatan yang lebih tinggi. Dalam hal ini baik d_{kn} maupun d_{mn} dalam adjacency akan sama-sama diberi nilai 0.



Gambar 2.7 *Adjacency Distance*

2.2.5 Biaya *Material Handling*

Biaya material handling adalah merupakan salah satu hal terpenting dalam suatu manajemen perusahaan. Semakin kecil biaya material handling yang dikeluarkan oleh suatu perusahaan berarti tata letak pabrik perusahaan tersebut tergolong baik.

Secara umum biaya yang termasuk dalam perancangan dan operasi sistem penanganan material adalah sebagai berikut (Purnomo, 2004) :

1. Biaya Investasi

   Yang termasuk dalam biaya ini adalah harga pembelian peralatan, harga komponen alat bantu, dan biaya instalasi.

2. Biaya Operasi

   Biaya perawatan, biaya bahan bakar, dan biaya tenaga kerja.

3. Biaya pembelian muatan, yang digolongkan dalam pembelian pallets dan container.

4. Biaya yang menyangkut masalah pengepakan dan kerusakan material.

Persamaan yang digunakan untuk menghitung ongkos *material handling* (OMH) adalah sebagai berikut :

1. Kapasitas alat pemindahan part =

$$\frac{Ukuran\_alat\_angkut(m^3)}{Ukuran\_unit\_yang\_dipindahkan(m^3)} \quad \text{.....11)}$$

2. Frekuensi Perpindahan = $\frac{Jumlah\_produksi}{Batch\_distribusi}$ .....12)

3. OMH / meter =

$$\frac{Biaya\_operasi\_tiap\_jam(Rp/jam)}{Jarak\_angkut\_tiap\_jam(m/jam)} \quad \text{.....13)}$$

4. Total OMH =

Jarak perpindahan x frekuensi perpindahan x OMH /meter.....14)

2.2.6 Algoritma Semut (*Ant Algorithm*)

2.2.6.1 Konsep Dasar Algoritma Semut

Pada sistem semut alami, semua semut mempunyai kapabilitas untuk menemukan rute terpendek dari sarangnya ke sumber makanan dan kembali lagi kesarangnya. Pada setiap perjalanannya, setiap semut meninggalkan jejak yang dapat digunakan sebagai informasi semut yang lain untuk memilih jejak yang akan dilewati. Apabila pada sebuah siklus ada semut yang berhasil menemukan rute yang pendek, maka semut tersebut akan dapat pulang kesarangnya dalam waktu yang singkat pada rute yang sama. Jejak yang ditinggalkan pada rute ini akan digunakan oleh semut lain, sehingga pada suatu saat semua semut akan

melewati rute yang sama. Konsep ini akan diadopsi untuk membuat sebuah algoritma yang bisa digunakan untuk menyelesaikan kasus kombinatorial. Dalam pencarian kasus kombinatorial, algoritma ini memanfaatkan informasi lokal dari solusi yang telah ditemukan sebelumnya secara random, dan memperbaiki solusi yang telah ditemukan tersebut melalui sebuah mekanisme pengumpulan informasi.

Adapun konsep dasar kerjanya adalah sebagai berikut ; Saat seekor semut yang terisolasi bergerak secara acak, semut ini akan mengikuti jejak yang telah ditinggalkan sebelumnya yang dapat dideteksi dan mempunyai tingkat probabilitas yang tinggi untuk diikuti dan melanjutkan jejak sebelumnya dengan pheromone baru. Tingkah laku kolektif yang muncul disebut dengan tingkah laku *Autocalystic*, dimana semut yang lain dapat mengikuti jejak yang ada, dan jejak yang semakin jelas akan memudahkan bagi semut lain untuk mengikutinya. Proses ini secara khusus terjadi melalui kumpulan umpan balik yang positif, dimana kemungkinan semut untuk memilih pola meingkat siring dengan jumlah semut yang sebelumnya mengikuti pola yang sama (Frieze, *et.al.*, 2000)

Informasi yang digunakan dalam algoritma semut dalam menyelesaikan kasus yang dihadapi adalah intensitas jejak yang telah dilewati oleh semut-semut terdahulu. Intensitas jejak ini sangat dipengaruhi jumah semut yang melewati sebuah rute dan tingkat penguapan jejak semut dari waktu ke waktu. Sehingga pada suatu waktu jejak semut pada rute yang kurang baik akan mendekati 0 dan jejak ini tidak akan terdeteksi oleh semut lain.

Algoritma semut merupakan algoritma yang dikembangkan untuk

optimasi sistem-sistem yang berbasis *Travelling Salesman problem* (TSP). Algoritma ini dapat menemukan rute terpendek dalam ruang pencarian yang lebih sempit dari pada metode konvensional maupun metode analitis. (Purnomo dan Zukhri, 2002).

Pada siklus semut, setiap semut akan berperan sebagai agen yang :

1. Akan berpindah dari kota i ke kota j, pada interval antara t dan (t +1). Kota j dipilih berdasarkan fungsi probabilitas terhadap jarak d (C_i,C_j) dan jumlah jejak yang ada pada edge yang menghubungkan antara C_i dan C_j.
2. Akan mengunjungi kota yang belum pernah dikunjungi sebelumnya. Hal ini diimplementasikan dengan cara menyimpan semua kota yang telah dikunjungi semut hingga semut tersebut menyelesaikan perjalanannya, dan akan mereset kembali setelah semut tersebut menyelesaikan satu siklus.
3. Setelah selesai dalam satu siklus, semut tersebut akan meninggalkan jejak pada setiap edge yang dilaluinya.

Adapun langkah umum Algoritma Semut adalah sebagai berikut :

Langkah 1 : a. Inisialisasi harga parameter-parameter algoritma.
b. Inisialisasi kota pertama setiap semut.

Langkah 2 : Pengisian kota pertama kedalam *tabu list*

Langkah 3 : Penyusunan rute kunjungan setiap semut ke setiap kota.

Langkah 4 : a. Perhitungan panjang rute setiap semut.
b. Pencarian rute terpendek.
c. Perhitungan perubahan *harga intensitas jejak kaki semut antar kota*.

Langkah 5 : a. Perhitungan *harga intensitas jejak kaki semut antar kota* untuk siklus berikutnya.

b. *Reset harga perubahan intensitas jejak semut antar kota.*

Langkah 6 : Pengosongan *tabu list*, dan ulangi langkah 2 bila diperlukan.

2.2.6.2 Mekanisme Algoritma Semut

2.2.6.2.1 Optimasi Pembentukan Sel Manufaktur

Dalam sel manufaktur terdapat sejumlah mesin pemroses dan sejumlah komponen yang harus diatur tata letaknya sedimikian.sehingga pemindahan semua komponen pada mesin pemrosesnya dapat diminimalkan. Pengelompokan komponen dan mesin dimulai dengan mengidentifikasi rute awal setiap komponen yang disusun dalam sebuah matriks insiden (*incidence matrix)*. Dalam matrik insiden ini urutan kolom (arah horizontal) menunjukkan urutan komponen, sedangkan urutan baris bawah (arah vertikal) menunjukkan urutan mesin. Bila sebuah komponen harus diproses oleh sebuah mesin, maka elemen pertemuan antara komponen dan mesin tersebut dalam matrik insiden diisi dengan 1, dan sebaliknya apabila komponen tidak perlu diproses oleh sebuah mesin, maka elemen pertemuan antara komponen dan mesin tersebut dalam matriks insiden diisi dengan 0.

Untuk meminimalkan perbindahan komponen, dalam pembentukan sel manufaktur diusahakan agar elemen-elemen tertentu matriks insiden membentuk blok-blok yang berisi seluruhnya atau sebagian besar angka 1 dan elemen-elemen

sisanya membentuk blok-blok yang seluruhnya atau sebagian besar angka 0.

Konsep TSP dapat diterapkan untuk meminimasi jarak perpindahan komponen. Dalam sistem manufaktur, komponen berpindah dari mesin yang akan mengerjakan proses pertama sampai mesin yang akan mengerjakan proses terakhir dan tidak perlu kembali lagi ke mesin pertama, sehingga konsep jarak dalam TSP perlu dimodifikasi, yaitu sebagai total jarak yang ditempuh komponen dari mesin pemroses pertama sampai mesin pemroses terakhir, apabila jarak antar mesin relatif sama, maka pengertian jarak antar mesin adalahkonversi dari jumlah komponen yang dipindahkan antar mesin. Apabila frekuensi pemindahan komponen dari sebuah mesin ke mesin lainnya besar, maka jarak antara kedua mesin tersebut seharusnya kecil.

Optimasi pembentukan sel manufaktur terdiri dari dua tahap, yaitu :

(a) Tahap 1 : Pengelompokan mesin-mesin ke dalam sel-sel.

Tahap ini bertujuan untuk mengurutkan kembali letak mesin sehingga dengan posisi mesin yang baru didapat jarak *material handling* yang lebih pendek. Hasil optimasi pada tahap ini adalah matriks semi final, yaitu matriks insiden yang mempunyai urutan mesin pemroses optimal, dimana urutan baris belum tentu menunjukkan urutan mesin.

$$\text{Min } Z = \sum_{i=1}^{n-1} d_{ij} \quad \text{.....15)}$$

Dimana $dij = C - fi$

dengan :

dij =frekwensi (jarak) mesin pemroses ke-i ke mesin pemroses ke-j

C = konstanta

fi adalah nilai *material handling part*

fi, jika forward $\sum$ *di*

fi(-1), jika backward

(b) Tahap 2 : Pengelompokan *part-part* ke dalam sel-sel.

Tahap ini bertujuan untuk menempatkan *part* pada posisi yang tepat sehingga didapat nilai minimum total penjumlahan baris dan kolom dari *intermediate matrix* yang telah dibentuk, yaitu *part-machine matrix* dengan susunan mesin yang optimal.

Hasil optimasi pada tahap kedua ini adalah matriks final, yaitu matriks semi final yang mempunyai urutan komponen optimal, dimana urutan kolom belum tentu menunjukkan urutan komponen.

Formulasi yang digunakan dalam membentuk sel manufaktur pada tahap kedua ini adalah :

$$\text{Min } Z = \sum baris + \sum kolom \quad \text{.....16)}$$

dimana :

$$\sum baris = \sum_{m=1}^{b-1} \sum_{p=1}^{k} \left| a_{pm} - a_{p,m+1} \right| \quad \text{.....17)}$$

$$\sum kolom = \sum_{p=1}^{k-1} \sum_{m=1}^{b} \left| a_{pm} - a_{p+1,m} \right| \quad \text{.....18)}$$

dengan :

k = jumlah komponen

b = jumlah mesin

p = indeks komponen, m = indeks mesin

2.2.6.2.2 Sistem Semut

Sistem Semut terbagi menjadi 3 algoritma, yaitu algoritma Siklus-Semut, Algoritma Densitas-Semut dan Algoritma Kuantitas Semut. Dalam penelitian ini Sistem Semut yang digunakan adalah Algoritma Siklus-Semut. Adapun persamaan algoritma siklus-semut adalah sebagai berikut :

$$p_{ij}^{k}(t) = \frac{\left[\tau_{ij}(t)\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}}{\sum_{j \in diijinkan}\left[\tau_{ij}(t)\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}}$$ untuk j $\in$ diijinkan bagi k .....19)

dan

untuk $p_{ij}^{k}(t) = 0$, untuk j lainnya

$\Delta\tau_{ij}^{k} = \frac{Q}{L_k}$, untuk (ij) $\in$ kota asal dan kota tujuan dalam daftar tabu $_k$

$\Delta\tau_{ij}^{k} = 0$ untuk (ij) lainnya. ..... 20)

Dengan :

η = visibilitas (1/jarak kota asal kekota tujuan)

L = panjang rute (jarak kunjungan kesemua kota terakhir ditambah jarak kota terakhir ke kota awal

Q = Tetapan

α = parameter pengendali intensitas jejak semut

β = parameter pengendali visibilitas

2.2.7 Tabu Search

2.2.7.1 Konsep Dasar Tabu Search

Menurut Glover (1986), konsep dasar dari *Tabu Search* adalah merupakan suatu algoritma yang menuntun setiap tahapannya agar dapat menghasilkan kreteria aspirasi yang paling optimum tanpa terjebak ke dalam solusi awal yang ditemukan selama tahapan itu berlangsung. Sehingga maksud dari algoritma ini adalah mencegah terjadinya perulangan ditemukannya solusi yang sama pada suatu iterasi yang akan digunakan lagi pada iterasi selanjutnya (dikatakan tabu).

Untuk menunjang sistematis dari tujuan *Tabu Search* tersebut, maka digunakanlah dua macam *tools*, yaitu *adaptive memory* dan *responsive exploration*. Dengan adanya *adaptive memory* pada Tabu Search ini menuntun suatu prosedur yang mampu dalam melakukan pencarian solusi yang diinginkan dengan lebih ekonomis dan efektif, sedangkan *rensponsive exploration* lebih menekankan pada tahapan tiap proses yang harus dilalui selama proses pencarian itu berlangsung, dimana pada setiap tahapan tersebut dipunyai suatu variable keputusan yang akan menuntun pada tahapan selanjutnya sampai akhir proses pencarian dihentikan.

2.2.7.2 Penggunaan Memori Pada Tabu Search

Memori pada Tabu Search ini mempunyai dua sifat, yaitu *explicit* dan *attributive* (Glover, 1986).

1. *Explicit memory* menyimpan *complete solution* yang umumnya menghabiskan alokasi ruang memori dan waktu, sehingga untuk menghindari hal ini, *complete solution* dikurangi sehingga hanya terdiri dari *elite solution* yang dikunjungi selama pencarian.
2. *Attributive memory* menyimpan informasi tentang atribut dari solusi yang ditemukan yang mungkin dapat berubah dari satu solusi ke solusi lain.

Sedangkan penggunaan dari memori *Tabu Search* ini mengacu pada *adaptive memory* seperti yang telah dijelaskan sebelumnya. Struktur memori dalam *Tabu Search* menggunakan empat prinsip utama yaitu :

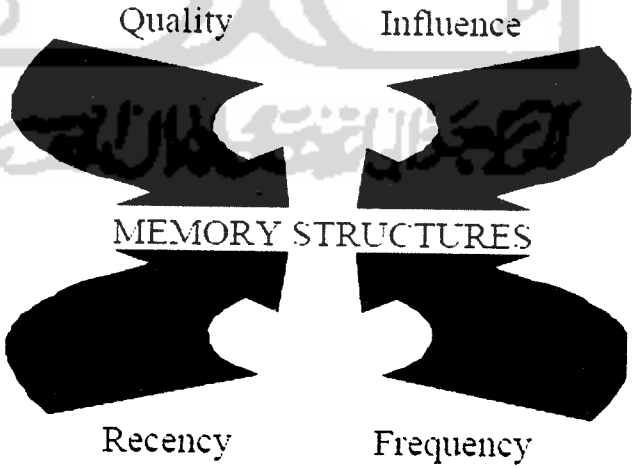
a) *Recency* atau lebih lengkapnya *recency based memory*, memori yang tetap menjaga struktur terbaik dari solusi awal yang ditemukan selama proses pencarian pada setiap iterasinya, sehingga apabila pada suatu iterasi ditemukan struktur / solusi yang lebih baik maka solusi ini akan tetap akan dipertahankan sampai ditemukannya solusi baru yang lebih baik lagi. Agar *recency* dapat bekerja dengan baik, maka didukung oleh *Tabu List*, dan dalam *tabu list* ini terdapat *tabu active*. *Tabu List* merupakan suatu set memori yang memberikan informasi tentang struktur dari solusi awal yang pernah diambil, sehingga *tabu list* akan menghindari digunakannya struktur yang sama untuk menghasilkan suatu solusi jika struktur tersebut

pernah dipakai. Sedangkan *tabu active* adalah tabu list yang berada paling akhir, artinya bahwa solusi itu merupakan solusi dari iterasi yang paling akhir yang sedang dikunjungi. Untuk membatasi range atau jumlah dari *tabu list* sendiri dinamakan dengan *Tabu Tenure*.

b) *Frequency* menyediakan sebuah tipe informasi yang merupakan kumpulan informasi yang telah direkam oleh *recency based memory*. Sehingga *frequency* dan *recency* dapat saling melengkapi untuk membentuk suatu informasi permanen guna mengevaluasi pergerakan / *move* yang terjadi.

c) *Quality* adalah kemampuan untuk membedakan solusi terbaik yang dikunjungi selama pencarian atau iterasi berlangsung.

d) *Influence* mempertimbangkan efek yang terjadi dari pemilihan solusi yang dipilih selama pencarian berlangsung, tidak hanya kualitas saja dipertimbangkan melainkan juga strukturnya.



Gambar 2.8 *Struktur Memori Tabu Search* (Glover and Laguna, 1997)

2.2.7.3 Mekanisme Tabu Search

Sebagai sebuah algoritma, *tabu search* dalam penyelesaiannya harus melewati setiap tahapan tertentu yang telah diatur (Glover, 1986)., adapun tahapan–tahapan tersebut adalah :

1. Membangkitkan solusi awal

   Yang dimaksudkan disini adalah sebelum kita memulai tahapan *tabu search*, kita mempunyai acuan awal sebagai pembanding ketika proses *tabu search* dimulai.

2. Menentukan kreteria aspirasi (*aspiration criteria*)

   Kriteria aspirasi ini fungsinya sebagai fungsi tujuan atau goal yang akan dicapai, contohnya untuk penelitian ini kriteria aspirasinya adalah minimasi jarak material handling.

3. Melakukan *Move*

   Ada beberapa macam *move* yang dapat dipilih selama proses pencarian ini berlangsung :

   1) *Local Search*, yang terdiri dari dua macam yaitu :

      a. Insertion : memilih secara acak satu bagian struktur untuk dipindah ke bagian yang lain.

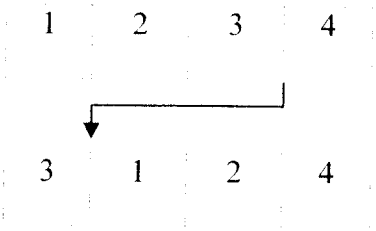
   Contoh : Struktur awal

   1 2 3 4

Gambar 2.9a Ilustrasi *insertion move* (struktur awal)

Jika dengan proses random didapat atribut ke-3, maka struktur dapat

berubah menjadi :



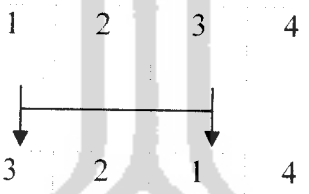
Gambar 2.9b Ilustrasi *insertion move* (struktur akhir)

b. Swap : memilih secara acak dua bagian struktur untuk selanjutnya ditukar posisinya.

Contoh : Struktur awal

Gambar 2.10a Ilustrasi *swap move* (struktur awal)

Jika random menghasilkan 1 dan 3, maka struktur menjadi



Gambar 2.10b Ilustrasi *swap move* (struktur akhir)

2) Neighborhood Search

Untuk pencarian dengan teknik ini setiap kemungkinan atribut dari struktur dapat dipindah – pindah. Permutasi *n-change neighborhood* mengambil n elemen dari matrik solusi (dimana berhubungan dengan item yang sedang diproduksi pada suatu mesin pada satu waktu), dan untuk tiap- tiap

pengubahan item yang sedang diproduksi dengan item lain. Perubahan yang dipakai oleh dua neighborhood dengan melakukan swap elemen matrik atau kombinasi elemen itu dengan menukar elemen lain dalam matrik.

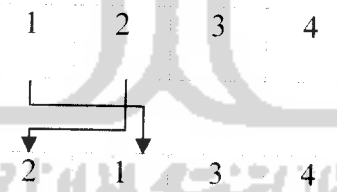
Bila x adalah solusi dari search space X, maka sebuah solusi y $\in$ $N^n{}_{change}$ (x) jika y dan x berbeda paling banyak / tidak lebih dari n elemen : $N^n{}_{change}$ = { y $\in$ X : y dapat diperoleh dari x dengan merubah pada tidak lebih n elemen $x_{i,k}$ dari nilai saat ini ke dalam sebuah elemen himpunan { 0,1,...P}.

Contoh : Struktur awal

Gambar 2.11a Ilustrasi *n-change neighborhood move*(struktur awal)

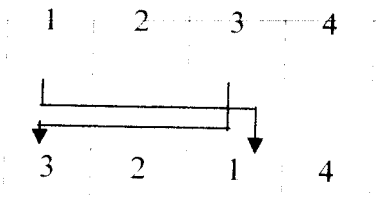
Dipilih 3 change neighborhood, maka struktur atribut diatas dapat berubah menjadi :

i.



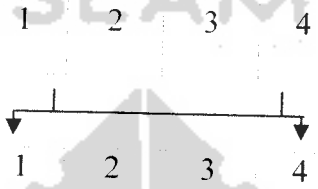
Gambar 2.11b Ilustrasi *n-change neighborhood move*

ii.



Gambar 2.11c Ilustrasi *n-change neighborhood move*

iii.



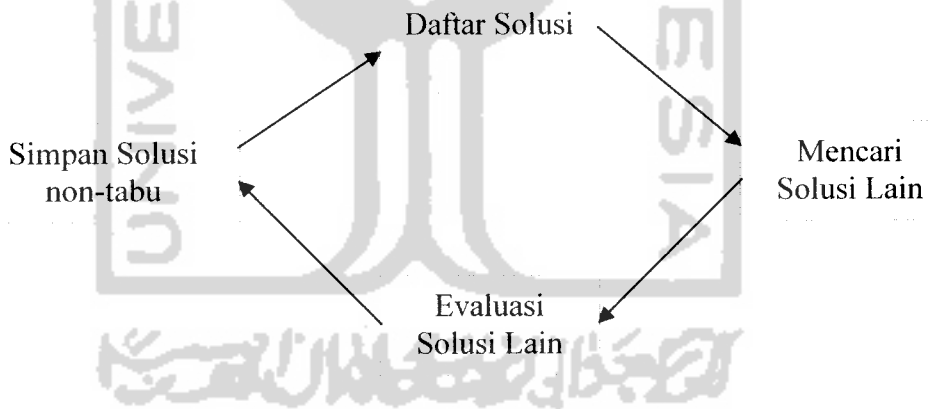
Gambar 2.11d Ilustrasi *n-change neighborhood move* (struktur akhir)

4. Untuk menghindari terulangnya langkah yang diambil, maka dilakukan *tabu test*. *Tabu Test* memanfaatkan *tabu list* yang sudah ada. *Tabu list* berisi atribut solusi – solusi yang telah dikunjungi sebelumnya. Tujuan sebenarnya dari *tabu list* bukan untuk mencegah terulangnya langkah yang telah diambil, tetapi lebih kepada agar tidak mundur. Untuk mencegah perulangan, daftar solusi yang telah dicapai disimpan dalam sebuah tabel. Bagaimanapun juga situasi perulangan jarang terjadi, karena telah dikombinasikan beberapa *neighborhood*, dimana akan memperluas *search space*, membuat kemungkinan mengulang solusi yang telah dikunjungi menjadi hampir tidak mungkin. Solusi pada tabel ini tabu. Pengecualiannya hanya untuk *blockage*

situation. Jika situasi ini terjadi, lalu semua jalur akan menjadi tabu. Untuk menghindari ini dipakai jalur paling awal dalam *tabu list*.

5. *Alternative move* yang lolos tabu test masih harus melewati *aspiration test*, apakah bisa melewati *aspiration threshold* atau tidak, jika tidak maka teruskan iterasi berikutnya.
6. Jika *alternative move* mempunyai *aspiration criteria* yang lebih baik daripada *aspiration threshold* maka dilakukan eksekusi terhadap *alternative move* tersebut dan memperbarui memori yang tidak relevan
7. Jika aturan pemberhentian sudah memenuhi syarat pemberhentian, maka pencarian berhenti.

Secara umum mekanisme Tabu Search dapat digambarkan sebagai berikut :



Gambar 2.12 Mekanisme Umum *Tabu Search*

Untuk *aspiration criteria* atau fungsi tujuan yang dipakai dalam penelitian ini adalah sebagai berikut :

1. Fase 1

$$\text{Min } Z = \sum_{i=1}^{n-1} d_{ij} \quad \text{.....21)}$$

$di = C - fi$

dengan :

di =frekwensi (jarak) mesin pemroses ke-i ke mesin pemroses ke-j

C = konstanta

fi adalah nilai material handling part

fi, jika forward $\sum di$

fi(-1), jika backward

2. Fase 2

$$\text{Min } Z = \sum baris + \sum kolom \quad \text{.....22)}$$

dimana :

$$\sum baris = \sum_{m=1}^{b-1} \sum_{p=1}^{k} \left| a_{pm} - a_{p,m+1} \right| \quad \text{.....23)}$$

$$\sum kolom = \sum_{p=1}^{k-1} \sum_{m=1}^{b} \left| a_{pm} - a_{p+1,m} \right| \quad \text{.....24)}$$

dengan :

k = jumlah komponen

b = jumlah mesin

p = indeks komponen, m = indeks mesin

Tahapan penggunaan memori pada algoritma Tabu Search dapat dijelaskan sebagai berikut :

Pada tahap pertama mekanisme *move* yang dipakai akan menghasilkan *candidate*

list untuk selanjutnya *candidate list* yang dihasilkan ini akan diuji menggunakan *tabu test* untuk mengetahui apakah *candidate* yang sudah ada sudah pernah dieksekusi dan berada pada daftar *tabu list* atau tidak, jika ya maka akan dibuang sedangkan jika tidak maka *candidate* ini akan dieksekusi nilai aspiration criterianya. Jika nilai *aspiration criteria* ini lebih baik dari nilai *aspiration criteria* awal, maka candidate ini akan dipertahankan, jika tidak maka candidate ini akan dibuang. Selanjutnya memori akan mengupdate *candidate* yang baru terpilih untuk dimasukan ke dalam tabu list yang baru. Evaluasi apakah semua *move* yang mungkin dapat dilakukan telah dieksekusi atau belum, jika belum kembali ke langkah awal untuk uji *candidate list*, jika sudah maka cukup. Nilai *aspiration criteria* yang baru hasil dari pencarian candidate yang dipakai diatas akan disimpan dalam memori (Glover, 1986).

BAB III

METODOLOGI PENELITIAN

3.1 Objek Penelitian

Penelitian dilakukan pada CV. Gambang Emas yang berlokasi di desa Keceme, Sariharjo, Sleman. Objek penelitian ini adalah tata letak fasilitas pada perusahaan yang bertujuan untuk mengetahui pengurangan jarak serta biaya pada penanganan bahan yang terjadi dengan diterapkannya konsep *Cellular Manufacturing System.*

3.2 Identifikasi Masalah

Dalam tahap ini dilakukan pengidentifikasian masalah yang dihadapi, yaitu bagaimana membentuk sel manufaktur yang bertujuan untuk meminimasi jarak dan biaya penanganan material.

3.3 Data-data yang Dibutuhkan

Data yang dibutuhkan dalam penelitian ini adalah :

1. Jenis dan dimensi mesin yang digunakan
2. Macam dan jumlah komponen / *part* yang akan diproduksi.
3. *Routing* produksi setiap *part*.

4. Kapasitas alat angkut.
5. Ukuran *batch* distribusi.
6. Jumlah dan upah tenaga kerja.
7. *Layout* awal pabrik.
8. Jenis dan jumlah alat angkut.
9. Harga dan umur ekonomis alat angkut.
10. Hari kerja efektif.

3.4 Metode Pengumpulan Data

Metode yang dilakukan untuk pengumpulan data adalah sebagai berikut :

1. Metode Observasi

   Yaitu melakukan pengamatan dan pencatatan langsung dalam rangka mendapatkan data yang berhubungan terhadap proses produksi yang berkaitan dengan masalah tata letak fasilitas produksi, jenis produksi, dan *part* yang dihasilkan.

2. Studi Kepustakaan

   Studi pustaka dari beberapa referensi ilmiah yang dapat mendukung dalam pengumpulan data dan membahas obyek yang akan diteliti.

3. Literatur Data Perusahaan

   Data-data lain dari literatur yang ada di perusahaan meliputi sejarah perusahaan, data gaji karyawan, proses produksi dan data-data lain.

3.5 Pengolahan Data

3.5.1 Pembentukan Sel Manufaktur

Incidence Matrix merupakan suatu prosedur sistematis yang menganalisa informasi dari rute proses pembuatan suatu *part*. Langkah-langkah penentuan *Incidence Matrix* adalah sebagai berikut :

1. Daftar semua *part* yang akan dibuat secara horisontal.
2. Daftar semua mesin yang akan digunakan untuk memproses *part* tersebut secara vertikal.
3. Isikan angka 1 pada koordinat pertemuan antara *part* dengan mesin bila *part* yang bersangkutan diproses di mesin tersebut.

Sel manufaktur dibentuk berdasarkan *Incidence Matrix*. Pengolahan data dengan menggunakan Algoritma Semut yang bertujuan untuk mengelompokkan komponen/*part* dan mesin dalam bentuk sel-sel.

3.5.2 Langkah Penyelesaian Algoritma Semut

Pada bagian ini akan dijelaskan mengenai algoritma yang digunakan dalam penelitian ini yaitu Algoritma Semut. Ada 2 tahap yang harus ditempuh untuk memecahkan masalah yang diangkat, yaitu : tahap penyusunan mesin dan tahap penyusunan part.

Adapun langkah-langkah Algoritma Semut sebagai berikut :

1. a. Inisialisasi harga parameter-parameter algoritma, seperti; *n (banyak kota)*

berikut x dan y *(koordinat)* atau d *(jarak antar kota)*, Q *(tetapan siklus-semut)*, α *(tetapan pengendali intensitas jejak semut)*, β *(tetapan pengendali visibilitas)*, η *(visibilitas antar kota* $= 1/d_{ij}$*)*, m *(banyak semut)*, ρ *(tetapan penguapan jejak semut)*, NC_{max} *(jumlah siklus maksimum)*, τ_{ij} *(intensitas jejak semut antar kota)*

b. Inisialisasi kota pertama setiap semut.

2. Pengisian kota pertama kedalam *tabu list*
3. Penyusunan rute kunjungan setiap semut ke setiap kota.
4. a. Perhitungan panjang rute setiap semut.

   b. Pencarian rute terpendek.

   c. Perhitungan perubahan *harga intensitas jejak kaki semut antar kota.*
5. a. Perghitungan *harga intensitas jejak kaki semut antar kota* untuk siklus berikutnya.

   b. *Reset harga perubahan intensitas jejak semut antar kota.*
6. Pengosongan *tabu list*, dan ulangi langkah 2 bila diperlukan.

3.5.3 Langkah Penyelesaian dengan Algoritma *Tabu Search*

Pada bagian ini akan dijelaskan mengenai algoritma yang digunakan dalam penelitian ini yaitu algoritma *Tabu Search*. Ada 2 fase yang harus ditempuh untuk memecahkan masalah yang diangkat, yaitu : Fase penyusunan mesin dan fase penyusunan part.

Adapun langkah – langkah algoritma *Tabu Search* sebagai berikut :

Langkah 1 *Initialization*

Tetapkan :

- *Tabulist size*

- Maksimum iterasi.

Langkah 2 *Initial Solution*

Bangkitkan *Initial solution*, kemudian menentukan *Current solution* dan *Best_Global solution*. Kemudian *Current solution* dimasukkan kedalam *Tabulist*.

Langkah 3 *Neighborhood generation*

Bangkitkan semua kemungkinan *solution* , dengan cara menukar posisi mesin atau part.

Langkah 4 *Neighborhood search*

Tentukan *Candidate solution*. Jika *Best_Global solution* lebih baik dari *candidate solution*, maka tetapkan *Best_Global solution* sebagai *candidate solution*.

Langkah 5 *Aspiration*

Jika solusi baru tidak dapat ditentukan, maka pilih solusi terbaik di *Tabulist* untuk dijadikan *candidate solution*.

Langkah 6 *Tabulist update*

Solusi yang lebih baik saat *aspiration* di *update*, kemudian solusi yang lama dibuang.

Langkah 7 *Move*

Solusi yang terbaik dipilih sebagai *candidate solution*.

Langkah 8 *Diversification*

Mencari solusi optimal sampai iterasi yang ditentukan selesai. Bila belum optimal atau iterasi belum habis maka balik ke Langkah 2. Bila sudah optimal masuk ke langkah selanjutnya.

Langkah 9 *Termination*

Bila solusi telah didapat maka hentikan iterasi.

3.5.4 Kontruksi Program Komputer

Setelah algoritma–algoritma kecil yang lebih terperinci didapat kemudian dilakukan sebuah kontruksi program komputer untuk merealisasikan integrasi algoritma yang sudah didapat.

Program komputer dibuat dalam Borland Delphi 7 yang kemudian dijalankan pada komputer dengan kapasitas RAM 384 MB dengan Prosessor Intel pentium IV 1.8 Ghz.

3.5.5 Perhitungan Jarak dan Ongkos *Material Handling*

Untuk perhitungan jarak antar mesin menggunakan *Euclidean Distance*, dengan persamaan ....8) pada bab II sebagai berikut :

$$d(A,B) = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}$$

Euclidean sering digunakan karena lebih mudah dimengerti dan mudah digunakan (Purnomo, 2004).

Dalam penentuan ongkos material handling (OMH), ada beberapa faktor yang harus diperhatikan yaitu :

1. Jumlah, jam kerja dan upah tenaga kerja
2. Jumlah alat angkut
3. Umur Ekonomis dari alat angkut tersebut
4. Biaya Operasional

Dari faktor-faktor tersebut akan dapat diketahui ongkos material handling per satuan jarak, maka total ongkos material handling (Total OMH) pada persamaan ....14) pada bab II, adalah sebagai berikut :

Total OMH = Jarak x frekuensi perpindahan x OMH per meter

3.6 Analisis Perbandingan *Layout* Awal Dengan Hasil Usulan

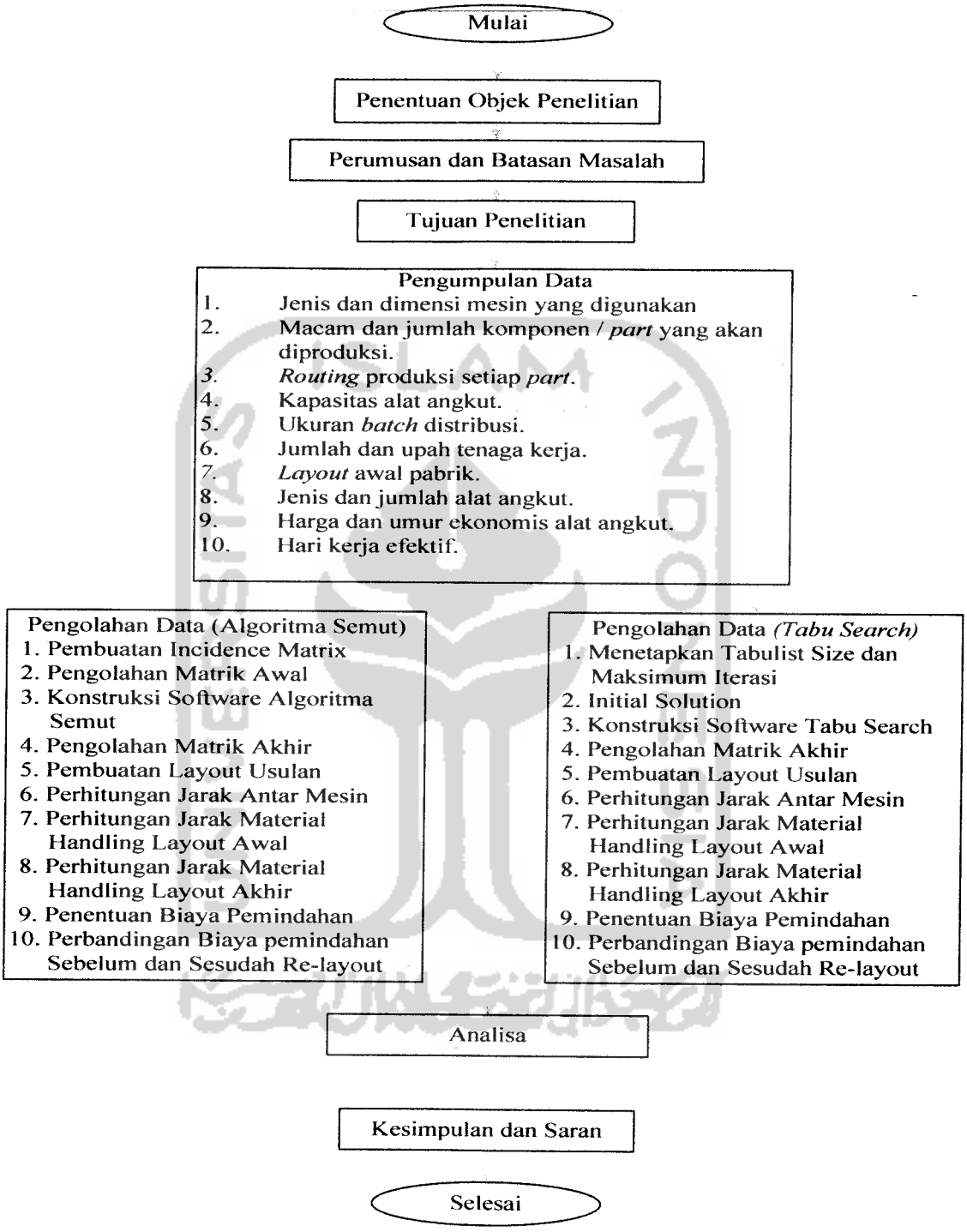
Pembentukan sel manufaktur yang baru tentu saja akan menimbulkan re-layout bagi perusahaan, dan penelitian ini dimaksudkan untuk membandingkan dengan layout awal sehingga dapat menghasilkan poin-poin penting dari perbandingan keduanya.

3.7 Kesimpulan dan Saran

Penarikan kesimpulan terhadap kasus yang diselesaikan pada tahap akhir dalam penelitian ini setelah dilakukan analisa terhadap kasus yang dipecahkan. Penarikan kesimpulan bertujuan untuk menjawab tujuan penelitian yang sudah ditetapkan

Saran-saran juga dikemukakan untuk memberikan masukan mengenai penyelesaian kasus yang dihadapi pada sistem yang diteliti. Selain itu juga diberikan saran-saran perbaikan bagi penelitian berikutnya untuk melakukan pengembangan model dan algoritma dalam penyelesaian kasus yang lebih kompleks akan tetapi mempunyai karakteristik yang sama dengan kasus dalam penelitian ini.

3.8 *Flowchart* Penelitian



Gambar 3.1 *Flowchart* Penelitian

BAB IV

PENGUMPULAN DAN PENGOLAHAN DATA

4.1 Pengumpulan Data

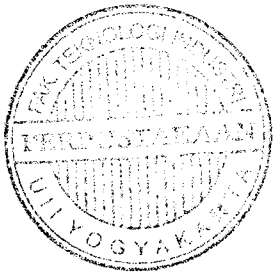
4.1.1 Profil Perusahaan

CV. Gambang Emas merupakan perusahaan yang bergerak dibidang manufaktur. Perusahaan ini didirikan pada tanggal 29 Juli 1999 oleh Bapak Ir. Setiawan Prasetyo. Hasil produksi dari perusahaan ini sebagian besar adalah meja, kursi, dan berbagai jenis lemari.

CV. Gambang Emas merupakan perusahaan furniture yang bertipe *make to order* (MTO) dimana perencanaan produksinya dilakukan setelah ada permintaan. Hasil produksi dari perusahaan ini sebaian besar dipasarkan ke negara-negara Eropa seperti Italia dan Spanyol bahkan hingga ke Amerika.

4.1.2 Data Umum Tenaga Kerja

CV. Gambang Emas menetapkan jam kerja bagi para karyawannnya dengan jumlah enam hari kerja efektif dalam seminggu yaitu senin – sabtu. Pengaturan jam kerja karyawan yang berlaku pada pada perusahaan ini adalah sebagai berikut :



1. Hari Senin, Selasa, Rabu, Kamis dan Sabtu adalah :

   a. Mulai masuk : Pukul 08.00 – 12.00

   b. Istirahat : Pukul 12.00 – 13.00

   c. Kerja kembali : Pukul 13.00 – 17.00

2. Hari Jumat adalah :

   a. Mulai masuk : Pukul 08.00 – 11.30

   b. Istirahat : Pukul 11.30 – 13.00

   c. Kerja kembali : Pukul 13.00 – 17.30

4.1.3 Peta Proses Operasi

4.1.3.1 Produk *Coffee Table*

Peta Proses Operasi *Coffee Table*

Frame Pjg			Frame Pdk			Frame Tgh			Sunduk Pjg			Sunduk Pdk			Sunduk Tgh			Kloss		
1'	O-1 I-1	Penerimaan	1'	O-8 I-2	Penerimaan	1 2'	O-15 I-3	Penerimaan	1'	O-22 I-4	Penerimaan	1 5'	O-29 I-5	Penerimaan	1'	O-36 I-6	Penerimaan	1'	O-43 I-7	Penerimaan
2 5'	O-2	M Cyrcle	2'	O-9	M Cyrcle	2'	O-16	M Cyrcle	2 2'	O-23	M Cyrcle	2'	O-30	M Cyrcle	2'	O-37	M Cyrcle	2'	O-47	M Cyrcle
3'	O-3	M Radial	3'	O-10	M Radial	3'	O-17	M Radial	3'	O-24	M Radial	3'	O-31	M Radial	3'	O-38	M Radial	3'	O-48	M Radial
3'	O-4	M Thickneser	3'	O-11	M Thickneser	3'	O-18	M Thickneser	3'	O-25	M Planner	3'	O-32	M Planner	3'	O-39	M Planner	3'	O-49	M Bandsaw
3 5'	O-5	M Jointer	3'	O-12	M Jointer	3'	O-19	M Jointer	3'	O-26	M Jointer	3'	O-33	M Jointer	3'	O-40	M Jointer	3'	O-50	M Planner
1'	O-6	M Bor	1'	O-13	M Bor	1'	O-20	M Bor	1'	O-27	M Bor	1 2'	O-34	M Bor	1'	O-41	M Bor	3'	O-51	M Jointer
2 5'	O-7	M Rotter	2'	O-14	M Rotter	2'	O-21	M Rotter	2 1'	O-28	M Rotter	2'	O-35	M Rotter	2'	O-42	M Rotter	1'	O-52	M Bor
																		2'	O-53	M Rotter
1'	I-8	QC	1'	I-9	QC	1'	I-10	QC	1'	I-11	QC	1'	I-12	QC	1'	I-13	QC	1'	I-14	QC
																			▼	
																		40'	O-54 I-15	Rakit dan periksa
																		45'	O-55	Sanding
																		60'	O-56	Finishing
																		20'	O-57 I-16	Packing dan Revisi

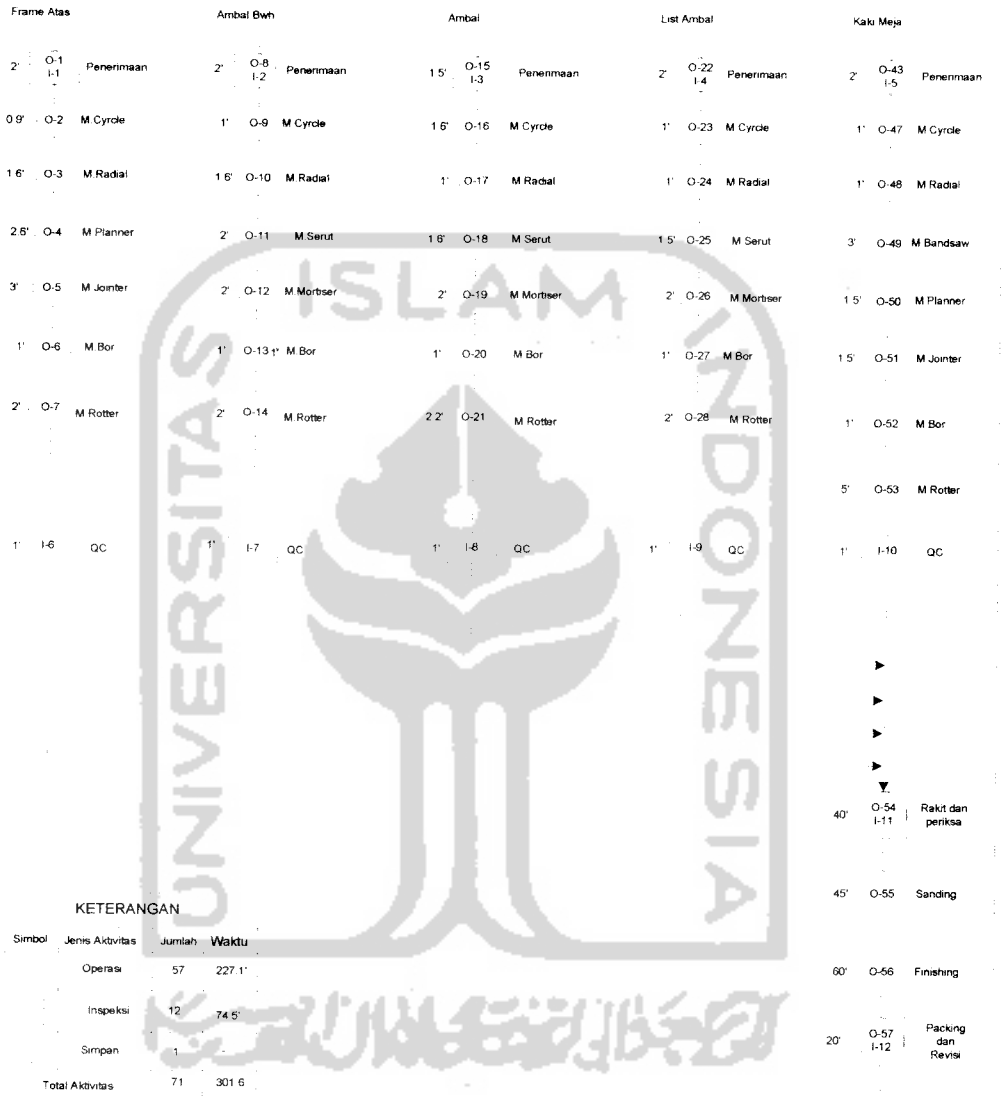
KETERANGAN

Simbol	Jenis Aktivitas	Jumlah	Waktu
	Operasi	57	274 9'
	Inspeksi	16	74 7'
	Simpan	1	-
	Total Aktivitas	74	349 6'

Gambar 4.1 Peta Proses Operasi Produk *Coffee Table*

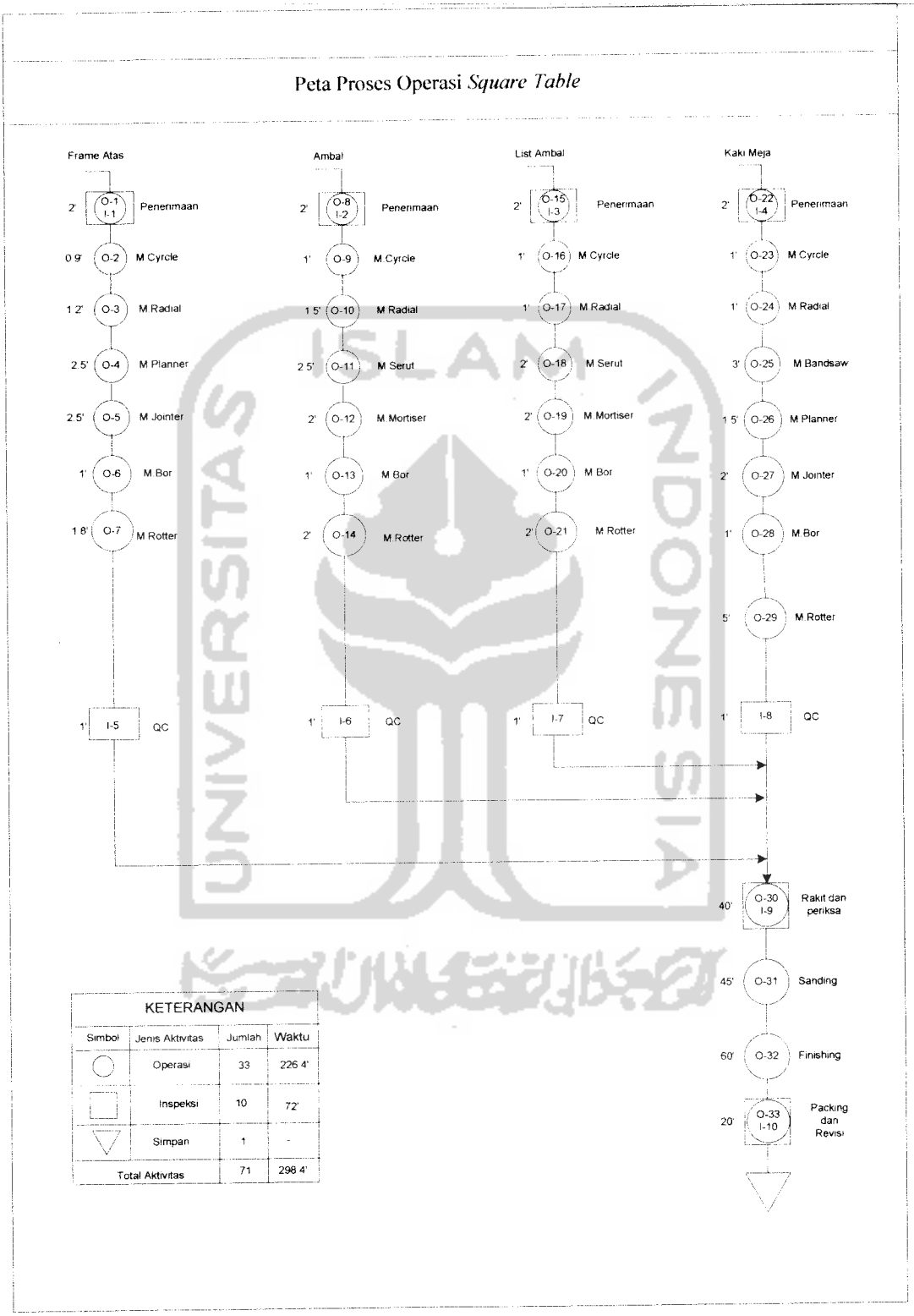
4.1.3.2 Produk *Gothic Table*

Peta Proses Operasi *Gothic Table*



Gambar 4.2 Peta Proses Operasi Produk *Gothic Table*

4.1.3.3 Produk *Square Table*



Gambar 4.3 Peta Proses Operasi Produk *Square Table*

4.1.4 Data Hasil Produksi

Penelitian ini dilakukan di departemen produksi. Produk yang dihasilkan CV. Gambang Emas antara lain meja, almari dan rak. Mesin – mesin produksi yang menunjang dalam kegiatan produksi antara lain, seperti pada tabel 4.1 :

Tabel 4.1 Fungsi Mesin

No.	Nama Mesin	Fungsi
1	Mesin Radial	Untuk memotong kayu sesuai dengan ukuran yang diinginkan
2.	Mesin Cyrcle	Untuk membelah kayu yang telah dipotong menjadi kayu batangan dengan ukuran tertentu
3.	Mesin Planner	Untuk mendapatkan kayu yang benar-benar siku
4.	Mesin Serut	Untuk menghaluskan permukaan kayu
5.	Mesin Bor	Untuk membuat lobang pada pasak, mengikat posisi antar bagian agar tidak mudah goyah
6.	Mesin Thickneser	Untuk menyerut permukaan agar mendapatkan ukuran ketebalan yang diinginkan
7.	Mesin Mortiser	Untuk membuat lobang tempat pen
8.	Mesin Jointer	Untuk menjepit komponen
9.	Mesin Rotter	Untuk membuat alur / profil
10.	Mesin Bandsaw	Untuk memotong

Tabel 4.2 Dimensi Mesin

No.	Nama Mesin	Panjang (cm)	Lebar (cm)
1	Mesin Radial	215	115
2.	Mesin Cyrcle	121	80
3.	Mesin Planner	125	37
4.	Mesin Serut	110	107
5.	Mesin Bor	78	67
6.	Mesin Thickneser	130	90
7.	Mesin Mortiser	132	112
8.	Mesin Jointer	235	240
9.	Mesin Rotter	136.5	99.6
10.	Mesin Bandsaw	77.5	115

Berikut ini posisi atau letak mesin – mesin tersebut dengan titik acuan koordinat kartesius (0,0) :

- Mesin 1 : X =2010 Y = 2230
- Mesin 2 : X = 1320 Y = 2625
- Mesin 3 : X = 1390 Y = 1860
- Mesin 4 : X = 1530 Y = 780
- Mesin 5 : X = 350 Y = 1150
- Mesin 6 : X = 2380 Y = 1800
- Mesin 7 : X = 400 Y = 500
- Mesin 8 : X = 415 Y = 2190
- Mesin 9 : X = 1890 Y = 1360
- Mesin 10 : X = 750 Y = 1500

Didalam penelitian ini hanya akan membahas tiga produk yang menguasai hampir 85 % volume produksi perusahaan, yaitu : *Coffee Table*, *Gothic Table*, dan *Square Table*.

Tabel 4.3 Data Volume Penjualan

No.	Bulan	Volume Penjualan (Unit)		
		Coffee Table	Gothic Table	Square Table
1	Juni 2006	75	62	40
2.	Juli 2006	70	63	45
3.	Agustus 2006	77	65	47
4.	September 2006	82	70	45
5.	Oktober 2006	80	65	50
6.	November 2006	85	73	55
7.	Desember 2006	75	68	48
8.	Januari 2007	78	70	55
9	Februari 2007	85	75	50
10.	Maret 2007	88	69	55
11.	April 2007	95	70	55
12.	Mei 2007	90	65	58

Tabel 4.4 Ukuran *Batch* Distribusi

No	Nama Komponen	*Batch* Distribusi (unit)	Produk
1	Frame Panjang	75	Coffee Table
2.	Frame Pendek	60	Coffee Table
3.	Frame Tengah	65	Coffee Table
4.	Sunduk Panjang	75	Coffee Table
5.	Sunduk Pendek	75	Coffee Table
6.	Sunduk Tengah	60	Coffee Table
7.	Kloss	175	Coffee Table
8.	Frame Atas	75	Gothic Table
		75	Square Table
9.	Ambal Bawah	50	Gothic Table
10.	Ambal	50	Gothic Table
		50	Square Table
11.	List Ambal	75	Gothic Table
		75	Square Table
12.	Kaki Meja	60	Gothic Table
		75	Square Table

Tabel 4.5 Urutan Produksi Tiap-tiap Komponen

No	Nama Komponen	Urutan Produksi
1	Frame Panjang	2-1-6-8-5-9
2.	Frame Pendek	2-1-6-8-5-9
3.	Frame Tengah	2-1-6-8-5-9
4.	Sunduk Panjang	2-1-3-8-5-9
5.	Sunduk Pendek	2-1-3-8-5-9
6.	Sunduk Tengah	2-1-3-8-5-9
7.	Kloss	2-1-10-3-8-5-9
8.	Frame Atas	2-1-3-8-5-9
9.	Ambal Bawah	2-1-4-7-5-9
10.	Ambal	2-1-4-7-5-9
11.	List Ambal	2-1-4-7-5-9
12.	Kaki Meja	2-1-10-3-8-5-9

Keterangan :

1. Mesin Radial
2. Mesin Cyrcle
3. Mesin Planner
4. Mesin Serut
5. Mesin Bor
6. Mesin Thickneser
7. Mesin Mortiser
8. Mesin Jointer
9. Mesin Rotter
10. Mesin Bandsaw

Tabel 4.6 Incidence Matrix Awal

		KOMPONEN											
MESIN		1	2	3	4	5	6	7	8	9	10	11	12
	1	1	1	1	1	1	1	1	1	1	1	1	1
	2	1	1	1	1	1	1	1	1	1	1	1	1
	3	0	0	0	1	1	1	1	1	0	0	0	1
	4	0	0	0	0	0	0	0	0	1	1	1	0
	5	1	1	1	1	1	1	1	1	1	1	1	1
	6	1	1	1	0	0	0	0	0	0	0	0	0
	7	0	0	0	0	0	0	0	0	1	1	1	0
	8	1	1	1	1	1	1	1	1	0	0	0	1
	9	1	1	1	1	1	1	1	1	1	1	1	1
	10	0	0	0	0	0	0	1	0	0	0	0	1

4.1.5 Data Jumlah Mesin

PT. Amelia Surya Cemerlang mempunyai mesin – mesin sebagai berikut :

Tabel 4.7 Data Jumlah Mesin

No.	Nama Mesin	Jumlah
1	Mesin Radial	1
2.	Mesin Cyrcle	1
3.	Mesin Planner	1
4.	Mesin Serut	1
5.	Mesin Bor	1
6.	Mesin Thickneser	1
7.	Mesin Mortiser	1
8.	Mesin Jointer	1
9.	Mesin Rotter	1
10.	Mesin Bandsaw	1

4.1.6 Data Peralatan *Material Handling*

Data peralatan *material handling* yang dimiliki perusahaan adalah sebagai berikut:

Tabel 4.8 Data Peralatan *Material Handling*

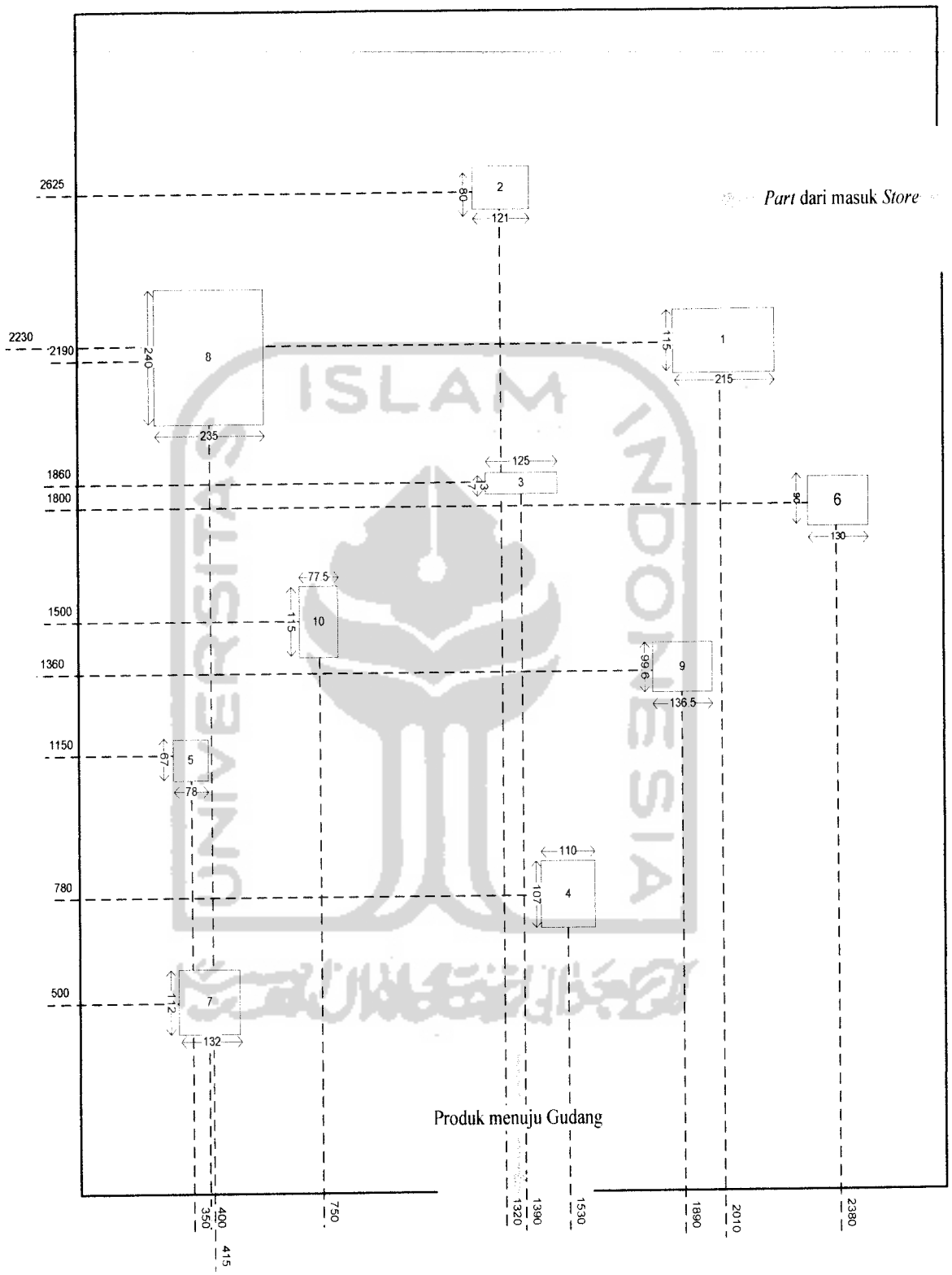
No.	Peralatan & Tenaga Kerja	Jumlah	Harga/unit	Umur Ekonomis	Jumlah
1.	Troli	7	Rp. 350.000	5 tahun	Rp. 2.450.000
2.	Tenaga Kerja	4	Rp. 12.250	-	Rp. 49.000

Ukuran troli = 147 x 100 x 118

4.1.7 Layout Awal Pabrik

Berikut ini layout awal pabrik berupa mesin – mesin dengan titik acuan koordinat kartesius (0,0) dan dengan keterangan sebagai berikut :

1. Mesin Radial
2. Mesin Cyrcle
3. Mesin Planner
4. Mesin Serut
5. Mesin Bor
6. Mesin Thickneser
7. Mesin Mortiser
8. Mesin Jointer
9. Mesin Rotter
10. Mesin Bandsaw



Gambar 4.4 Layout Awal Sebelum Penambahan Mesin

4.2 Pengolahan Data

Data yang telah ada digunakan untuk menganalisiss permasalahan yang ada, maka dilakukan pengolahan data yang akan melalui beberapa tahapan sebagai berikut :

4.2.1 Penentuan Volume Produksi

Untuk Jumlah produksi pada bulan Mei, maka dilakukan Forecasting dengan menggunakan WinQSB. Untuk mengetahui jumlah produk yang diproduksi tiap bulannya kedepan, maka akan dilakukan peramalan untuk tiap tahunnya, selama dua tahun kedepan :

Tabel 4.9 Jumlah Produk yang Diproduksi

Produk	Jumlah Produk rata-rata/bulan Tahun ke (Unit)	
	I	II
Coffee Table	82	103
Gothic Table	61	61
Square Table	63	80

4.2.2 Penentuan Jumlah Mesin Produksi

Penentuan jumlah mesin berguna untuk menentukan apakah jumlah mesin yang dimiliki oleh perusahaan sudah optimal dalam memenuhi permintaan konsumen. Dalam menentukan jumlah mesin, dipengaruhi oleh :

1. Data jumlah produk yang diproduksi per periode
2. Waktu proses untuk setiap mesin dalam memproduksi seluruh produk
3. Jam kerja per bulan : 25 hari/bulan x 8 jam/hari x 60 menit = 12000 menit/bulan
4. Effesiensi mesin 90 %

Jumlah mesin yang optimal dapat memenuhi permintaan yang telah dipesan konsumen ke perusahaan. Untuk mengetahui jumlah mesin yang optimal, maka terlebih dahulu harus menghitung total waktu proses tiap mesin/departemen.

Total Waktu Proses = Waktu proses tiap mesin/departemn x Rata - rata permintaan/hari

$$\text{Jumlah Mesin} = \frac{\text{Total Waktu Proses}}{\text{Total Jam Kerja x Effesiensi Mesin}} \quad \text{.....25)}$$

Untuk menganalisa kapan perusahaan melakukan penambahan mesin, maka akan dianalisa perubahan permintaan per tahunnya, untuk dua tahun kedepan.

Contoh perhitungan untuk Mesin Cyrcle pada tahun pertama

a. Jumlah produksi Coffee Table : 62 Unit/bulan

b. Jumlah produksi End Table : 61 Unit/bulan

c. Jumlah produksi Square Table : 47 Unit/bulan

d. Total Waktu Proses

= Waktu proses tiap mesin/departemen x rata-rata permintaan/bulan

= { (37,4 x 82)+(14,2 x 61)+(12,6 x 63) } = 4726,8

e. Jumlah Mesin = $\frac{4726,8}{12000x0.9}$ = 0,438 ≈ 1 mesin

Dari hasil perhitungan diatas, pada tahun kedua, kapasitas produksi perusahaan masih dapat memenuhi permintaan. Dengan demikian perusahaan tidak perlu melakukan penambahan mesin.

4.2.3 Pengolahan Matrik Awal

Perhitungan pada matik awal bertujuan untuk mengukur nilai effisiensi pengelompokkan dan nilai *efficacy* / kekuatan pengelompokkan. Berikut adalah matrik awal / *Incidence Matrix* :

Tabel 4.10 *Incidence Matrix*Awal

		KOMPONEN											
MESIN		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
	1	1	1	1	1	1	1	1	1	1	1	1	1
	2	1	1	1	1	1	1	1	1	1	1	1	1
	3	0	0	0	1	1	1	1	1	0	0	0	1
	4	0	0	0	0	0	0	0	0	1	1	1	0
	5	1	1	1	1	1	1	1	1	1	1	1	1
	6	1	1	1	0	0	0	0	0	0	0	0	0
	7	0	0	0	0	0	0	0	0	1	1	1	0
	8	1	1	1	1	1	1	1	1	0	0	0	1
	9	1	1	1	1	1	1	1	1	1	1	1	1
	10	0	0	0	0	0	0	1	0	0	0	0	1

Pengukuran Performansi

M = 10 ; P = 12 ; o = 74 ; e = 29 ; v = 34 ; ω = 0.5

a. Effisiensi Pengelompokan

$$\eta_1 = \frac{o-e}{o-e+v} = \frac{74-29}{74-29+34} = 0.5696$$

$$\eta_2 = \frac{MP-o-v}{MP-o-v+e} = \frac{(10x12)-74-34}{(10x12)-74-34+29} = 0.2926$$

$$\eta = \omega\eta_1 + (1-\omega)\eta_2 = 0.5(0.5696) + (1-0.5)(0.4311) = 0.4312$$

b. Kekuatan Pengelompokan

$$\tau = \frac{o-e}{o+v} = \frac{74-29}{74+29} = 0.4167$$

4.2.4 Pengolahan Data Algoritma

4.2.4.1 Pengolahan Data dengan Algoritma Semut

Dalam penelitian ini terdapat dua fase yang harus diselesaikan yaitu tahap 1 untuk penyusunan mesin dan tahap 2 untuk penyusunan *part*. Pengolahan data-data akan melalui tahapan Algoritma Semut.

a. Tahap 1

Pada tahap 1 ini jarak antar kota dalam algoritma Siklus-Semut adalah jarak antar mesin pembentuk sel manufaktur yang sama dengan besaran konversi dari jumlah komponen yang dipindahkan dari sebuah mesin ke mesin yang lain. Perhitungan besaran ini dilakukan berdasarkan frekuensi pemindahan komponen antar mesin sesuai dengan rute masing-masing komponen. Jadi jarak antar kota dalam Algoritma Siklus-Semut merupakan fungsi konversi frekuensi pemindahan komponen antar mesin menjadi jarak antar mesin. Tahap ini bertujuan untuk mengurutkan kembali letak mesin sehingga dengan posisi mesin yang baru didapat jarak material yang lebih pendek. Tahap ini diawali dengan membangun tabel *material handling* antar mesin, dimana tabel ini mengindikasikan total nilai dari perjalanan material yang dilakukan antara 2 mesin. Perhitungan jarak antar mesin dipengaruhi oleh nilai mesin-mesin material handling perpindahan *part* antara dua mesin, dimana nilai frekwensi dihitung berdasarkan formula rumus

yang digunakan yang mengidentifikasikan adanya *forward* atau *backward*, dan *incidence matrix* yang hasilnya tercantum pada tabel 4.10

$di = C - fi$

fi adalah nilai *material handling part*

fi, jika *forward*

fi(-1), jika *backward*

Sebagai contoh akan dihitung jarak mesin 1 ke 2 dan mesin 2 ke 1, dengan menggunakan konstanta (C) = 20, dengan banyaknya perpindahan *part* (*fi*) antara mesin 1 dan mesin 2 adalah sebanyak 2 kali :

- *Mesin 1 ke 2*

$di = 20 - 2 = 18$

- *Mesin 2 ke 1*

$di = 20 - 2(-1) = 22$

Nilai frekwensi (jarak) didapat dari banyaknya perpindahan *part* dari mesin satu ke mesin lainnya, dengan memperhitungkan urutan maju dan urutan mundur dari suatu *part* yang diproses dalam mesin seperti pada contoh diatas. Berikut adalah nilai frekwensi (jarak) dari setiap mesin :

Tabel 4.11 Frekwensi (jarak) Antar Mesin

	1	2	3	4	5	6	7	8	9	10
1	0	8	14	17	8	17	17	11	8	18
2	32	0	14	17	8	17	17	11	8	18
3	26	26	0	20	14	20	20	14	14	18
4	23	23	20	0	17	20	17	20	17	20
5	32	32	26	23	0	17	17	11	8	18
6	23	23	20	20	23	0	20	17	17	20
7	23	23	20	23	23	20	0	11	17	20
8	29	29	26	20	29	23	20	0	11	18
9	32	32	26	23	32	23	23	29	0	18
10	22	22	22	20	22	20	20	22	22	0

Algoritma Semut digunakan untuk mencari solusi terbaik dari penelitian ini. Dimana input data yang diperlukan untuk melakukan pengolahan di algoritma ini yaitu frekwensi (jarak) antar mesin. Adapun parameter yang digunakan dalam penyelesaian kasus ini adalah sebagai berikut :

1. Jumlah semut (m) = 100
2. Jumlah siklus semut (NC_{max}) = 100
3. Tetapan siklus semut = 100
4. Tetapan pengendali intensitas jejak semut (α) = 1
5. Tetapan pengendali visibilitas (β) = 5
6. Tetapan penguapan jejak semut (ρ) = 0.5
7. Intensitas jejak semut (τ) = 0.1

Contoh iterasi pada tahap 1 yaitu urutan pengolahan mesin setelah melalui perhitungan Algoritma Semut didapatkan hasil seperti dibawah ini untuk 2 siklus pertama :

```
Siklus ke: 1
Intens Jejak Semut: 0.1
---------------
Semut     1 -> 2, 9, 3, 8, 7, 10, 6, 1, 5, 4 = 162
Semut     2 -> 1, 9, 5, 8, 10, 7, 4, 6, 2, 3 = 169
Semut     3 -> 8, 10, 2, 9, 7, 6, 3, 5, 1, 4 = 174
Semut     4 -> 7, 9, 10, 8, 3, 6, 5, 4, 2, 1 = 204
Semut     5 -> 6, 10, 2, 5, 9, 7, 3, 8, 4, 1 = 158
Semut     6 -> 7, 6, 9, 10, 2, 5, 8, 4, 3, 1 = 162
Semut     7 -> 4, 7, 5, 10, 9, 3, 8, 6, 2, 1 = 198
Semut     8 -> 2, 6, 8, 9, 10, 7, 3, 5, 4, 1 = 163
Semut     9 -> 4, 5, 9, 1, 2, 3, 8, 7, 10, 6 = 153
Semut    10 -> 3, 8, 6, 9, 10, 5, 7, 4, 1, 2 = 165
Semut    11 -> 8, 9, 10, 2, 5, 6, 3, 7, 1, 4 = 156
Semut    12 -> 9, 4, 6, 8, 10, 5, 7, 2, 3, 1 = 180
Semut    13 -> 6, 8, 9, 3, 5, 7, 2, 4, 1, 10 = 166
Semut    14 -> 6, 9, 10, 8, 3, 7, 4, 5, 1, 2 = 183
Semut    15 -> 8, 9, 3, 5, 4, 7, 10, 1, 2, 6 = 158
Semut    16 -> 4, 1, 9, 10, 8, 3, 5, 7, 6, 2 = 171
Semut    17 -> 5, 8, 9, 3, 10, 7, 6, 1, 2, 4 = 154
Semut    18 -> 2, 5, 9, 10, 4, 6, 8, 3, 7, 1 = 160
Semut    19 -> 5, 9, 10, 1, 2, 6, 7, 3, 4, 8 = 153
Semut    20 -> 2, 7, 9, 10, 1, 3, 8, 4, 6, 5 = 165
Semut    21 -> 5, 9, 6, 7, 2, 8, 4, 3, 10, 1 = 165
Semut    22 -> 5, 9, 7, 8, 6, 4, 2, 10, 3, 1 = 183
Semut    23 -> 8, 9, 4, 7, 5, 6, 2, 3, 10, 1 = 168
Semut    24 -> 1, 5, 9, 6, 8, 4, 7, 3, 10, 2 = 153
Semut    25 -> 9, 8, 4, 5, 3, 10, 7, 6, 1, 2 = 181
Semut    26 -> 8, 9, 6, 2, 5, 10, 3, 7, 1, 4 = 165
Semut    27 -> 6, 10, 4, 5, 9, 1, 8, 3, 7, 2 = 177
Semut    28 -> 1, 2, 9, 10, 8, 4, 5, 6, 3, 7 = 150
Semut    29 -> 8, 9, 10, 1, 5, 7, 6, 3, 4, 2 = 159
Semut    30 -> 8, 9, 10, 6, 3, 5, 7, 4, 1, 2 = 154
Semut    31 -> 6, 1, 9, 10, 7, 5, 8, 4, 3, 2 = 169
Semut    32 -> 9, 3, 8, 4, 10, 5, 6, 1, 2, 7 = 167
Semut    33 -> 6, 9, 10, 8, 4, 3, 5, 7, 1, 2 = 159
Semut    34 -> 9, 10, 5, 8, 6, 2, 7, 3, 1, 4 = 177
Semut    35 -> 2, 9, 10, 5, 8, 7, 1, 4, 3, 6 = 159
Semut    36 -> 6, 9, 10, 3, 5, 8, 4, 1, 2, 7 = 150
Semut    37 -> 7, 2, 9, 10, 3, 8, 4, 5, 6, 1 = 162
Semut    38 -> 7, 6, 8, 9, 4, 5, 3, 10, 1, 2 = 162
Semut    39 -> 9, 10, 7, 6, 5, 8, 4, 1, 2, 3 = 157
Semut    40 -> 7, 9, 10, 4, 1, 5, 8, 6, 3, 2 = 166
Semut    41 -> 7, 1, 2, 9, 4, 5, 8, 10, 6, 3 = 148
```

```
Semut   42 -> 7, 3, 5, 9, 10, 8, 4, 6, 1, 2 = 153
Semut   43 -> 3, 9, 4, 6, 7, 10, 2, 5, 8, 1 = 167
Semut   44 -> 9, 7, 2, 8, 10, 4, 6, 5, 3, 1 = 190
Semut   45 -> 3, 5, 9, 2, 10, 4, 8, 6, 7, 1 = 178
Semut   46 -> 9, 10, 6, 8, 7, 1, 2, 5, 3, 4 = 160
Semut   47 -> 6, 4, 1, 9, 10, 5, 8, 3, 7, 2 = 171
Semut   48 -> 1, 9, 8, 6, 3, 7, 2, 5, 10, 4 = 169
Semut   49 -> 7, 9, 10, 8, 4, 5, 6, 3, 2, 1 = 189
Semut   50 -> 2, 5, 8, 9, 4, 7, 10, 3, 6, 1 = 155
Semut   51 -> 1, 2, 5, 8, 7, 3, 9, 6, 10, 4 = 144
Semut   52 -> 2, 9, 4, 1, 5, 8, 10, 3, 7, 6 = 153
Semut   53 -> 5, 8, 9, 10, 7, 4, 2, 6, 3, 1 = 169
Semut   54 -> 3, 5, 9, 10, 6, 7, 4, 8, 1, 2 = 160
Semut   55 -> 5, 9, 6, 8, 10, 2, 3, 7, 1, 4 = 162
Semut   56 -> 2, 9, 8, 4, 7, 3, 5, 6, 1, 10 = 166
Semut   57 -> 8, 9, 7, 5, 6, 3, 4, 10, 1, 2 = 164
Semut   58 -> 1, 2, 5, 9, 7, 10, 6, 8, 4, 3 = 144
Semut   59 -> 4, 5, 9, 3, 8, 6, 10, 1, 2, 7 = 155
Semut   60 -> 6, 9, 10, 7, 5, 8, 4, 3, 2, 1 = 187
Semut   61 -> 6, 10, 2, 9, 4, 7, 3, 8, 5, 1 = 185
Semut   62 -> 1, 9, 10, 3, 8, 2, 6, 7, 4, 5 = 168
Semut   63 -> 7, 10, 4, 5, 8, 9, 3, 1, 2, 6 = 156
Semut   64 -> 9, 3, 2, 4, 10, 7, 6, 8, 5, 1 = 207
Semut   65 -> 3, 8, 9, 4, 1, 5, 10, 6, 7, 2 = 160
Semut   66 -> 2, 7, 9, 10, 4, 8, 6, 5, 1, 3 = 184
Semut   67 -> 2, 9, 1, 8, 6, 5, 7, 10, 4, 3 = 174
Semut   68 -> 9, 6, 4, 7, 1, 2, 5, 10, 3, 8 = 153
Semut   69 -> 2, 5, 8, 9, 10, 7, 3, 4, 1, 6 = 148
Semut   70 -> 9, 10, 3, 8, 4, 5, 6, 7, 1, 2 = 159
Semut   71 -> 1, 2, 9, 10, 7, 3, 6, 8, 4, 5 = 148
Semut   72 -> 2, 9, 4, 10, 6, 8, 3, 5, 7, 1 = 168
Semut   73 -> 4, 5, 8, 9, 10, 7, 3, 6, 1, 2 = 148
Semut   74 -> 9, 10, 6, 1, 2, 5, 8, 4, 3, 7 = 148
Semut   75 -> 3, 9, 10, 6, 7, 8, 4, 5, 2, 1 = 193
Semut   76 -> 6, 8, 9, 10, 7, 4, 3, 5, 2, 1 = 187
Semut   77 -> 4, 7, 6, 10, 9, 8, 5, 3, 1, 2 = 197
Semut   78 -> 6, 4, 5, 9, 10, 2, 3, 8, 7, 1 = 156
Semut   79 -> 2, 5, 9, 10, 8, 3, 6, 1, 7, 4 = 165
Semut   80 -> 6, 9, 10, 5, 8, 4, 3, 7, 2, 1 = 183
Semut   81 -> 5, 7, 9, 3, 8, 4, 6, 2, 10, 1 = 177
Semut   82 -> 3, 2, 9, 6, 1, 8, 7, 10, 5, 4 = 176
Semut   83 -> 1, 8, 9, 7, 6, 3, 10, 2, 5, 4 = 156
Semut   84 -> 6, 9, 10, 3, 2, 5, 8, 4, 1, 7 = 162
Semut   85 -> 7, 9, 10, 2, 5, 8, 3, 4, 6, 1 = 165
Semut   86 -> 6, 1, 5, 9, 10, 3, 7, 8, 4, 2 = 162
Semut   87 -> 2, 8, 9, 10, 1, 5, 4, 7, 3, 6 = 150
Semut   88 -> 3, 5, 9, 6, 8, 10, 2, 7, 4, 1 = 165
Semut   89 -> 9, 10, 6, 8, 7, 4, 5, 3, 2, 1 = 199
Semut   90 -> 8, 9, 6, 4, 5, 7, 2, 3, 10, 1 = 165
Semut   91 -> 8, 9, 10, 1, 5, 6, 2, 7, 3, 4 = 156
Semut   92 -> 4, 9, 10, 2, 5, 8, 7, 3, 6, 1 = 159
```

```
Semut    93 -> 6, 2, 9, 10, 3, 5, 8, 7, 1, 4 = 156
Semut    94 -> 1, 5, 8, 9, 6, 2, 3, 7, 10, 4 = 150
Semut    95 -> 2, 9, 6, 3, 10, 7, 1, 5, 8, 4 = 151
Semut    96 -> 7, 8, 9, 10, 3, 5, 4, 1, 6, 2 = 171
Semut    97 -> 5, 8, 9, 3, 6, 10, 2, 1, 4, 7 = 176
Semut    98 -> 4, 7, 6, 10, 9, 2, 5, 8, 3, 1 = 182
Semut    99 -> 9, 10, 5, 8, 6, 3, 1, 2, 7, 4 = 168
Semut   100 -> 5, 9, 10, 8, 7, 4, 6, 3, 2, 1 = 189
---------------
Minimal Semut ke 51 -> 1, 2, 5, 8, 7, 3, 9, 6, 10, 4 = 144

Siklus ke: 2
Intens Jejak Semut: 60.5341056942227
---------------
Semut     1 -> 1, 9, 10, 4, 5, 8, 6, 7, 3, 2 = 163
Semut     2 -> 5, 8, 9, 4, 7, 3, 2, 10, 6, 1 = 169
Semut     3 -> 2, 5, 6, 9, 10, 4, 8, 7, 1, 3 = 157
Semut     4 -> 6, 3, 9, 7, 1, 8, 10, 4, 5, 2 = 178
Semut     5 -> 4, 5, 9, 6, 7, 1, 8, 10, 2, 3 = 156
Semut     6 -> 2, 7, 9, 6, 1, 5, 8, 10, 3, 4 = 159
Semut     7 -> 8, 9, 4, 7, 3, 1, 2, 5, 6, 10 = 150
Semut     8 -> 8, 9, 10, 6, 7, 5, 3, 1, 2, 4 = 169
Semut     9 -> 4, 7, 8, 9, 10, 6, 3, 5, 2, 1 = 184
Semut    10 -> 6, 9, 8, 3, 5, 7, 10, 4, 1, 2 = 174
Semut    11 -> 7, 1, 9, 6, 4, 8, 10, 3, 5, 2 = 180
Semut    12 -> 3, 5, 9, 6, 8, 7, 10, 2, 4, 1 = 164
Semut    13 -> 5, 9, 4, 7, 1, 8, 10, 2, 3, 6 = 156
Semut    14 -> 3, 9, 1, 5, 8, 10, 4, 7, 2, 6 = 160
Semut    15 -> 6, 4, 8, 9, 10, 3, 5, 7, 1, 2 = 153
Semut    16 -> 5, 9, 6, 7, 1, 2, 8, 10, 4, 3 = 151
Semut    17 -> 1, 5, 9, 10, 6, 8, 7, 2, 4, 3 = 151
Semut    18 -> 2, 5, 9, 7, 8, 4, 10, 6, 1, 3 = 156
Semut    19 -> 8, 9, 3, 5, 1, 2, 6, 10, 4, 7 = 165
Semut    20 -> 9, 7, 5, 8, 10, 3, 6, 4, 1, 2 = 168
Semut    21 -> 3, 4, 8, 9, 10, 5, 7, 6, 1, 2 = 159
Semut    22 -> 1, 2, 8, 9, 4, 3, 6, 5, 7, 10 = 153
Semut    23 -> 6, 9, 10, 5, 8, 4, 7, 3, 2, 1 = 183
Semut    24 -> 5, 9, 6, 8, 1, 2, 3, 10, 4, 7 = 154
Semut    25 -> 3, 5, 10, 7, 8, 9, 6, 1, 2, 4 = 154
Semut    26 -> 5, 8, 6, 7, 3, 9, 10, 4, 2, 1 = 181
Semut    27 -> 7, 6, 9, 10, 3, 8, 4, 5, 1, 2 = 168
Semut    28 -> 7, 9, 10, 3, 8, 2, 5, 6, 1, 4 = 165
Semut    29 -> 5, 9, 10, 8, 6, 3, 2, 4, 7, 1 = 174
Semut    30 -> 7, 3, 4, 10, 1, 9, 6, 5, 8, 2 = 176
Semut    31 -> 1, 2, 9, 7, 6, 5, 8, 10, 4, 3 = 151
Semut    32 -> 8, 9, 4, 5, 6, 3, 10, 2, 7, 1 = 168
Semut    33 -> 7, 10, 6, 3, 8, 9, 4, 5, 1, 2 = 165
Semut    34 -> 8, 9, 10, 2, 5, 7, 3, 4, 6, 1 = 159
Semut    35 -> 7, 9, 6, 2, 5, 8, 4, 10, 1, 3 = 158
Semut    36 -> 6, 1, 2, 5, 8, 9, 10, 4, 3, 7 = 139
Semut    37 -> 6, 4, 9, 2, 5, 8, 3, 1, 7, 10 = 177
```

```
Semut   38 -> 7, 3, 9, 10, 4, 5, 8, 1, 2, 6 = 154
Semut   39 -> 7, 9, 8, 10, 2, 5, 6, 1, 4, 3 = 171
Semut   40 -> 8, 9, 10, 6, 4, 5, 7, 3, 1, 2 = 157
Semut   41 -> 2, 5, 9, 10, 4, 8, 1, 3, 7, 6 = 157
Semut   42 -> 7, 10, 9, 3, 5, 8, 4, 6, 2, 1 = 188
Semut   43 -> 1, 5, 9, 8, 6, 10, 7, 2, 4, 3 = 168
Semut   44 -> 7, 6, 4, 9, 10, 3, 8, 5, 2, 1 = 204
Semut   45 -> 4, 7, 9, 10, 5, 8, 3, 6, 2, 1 = 186
Semut   46 -> 6, 8, 9, 3, 5, 7, 10, 2, 4, 1 = 167
Semut   47 -> 4, 9, 10, 8, 3, 7, 1, 5, 6, 2 = 174
Semut   48 -> 1, 2, 5, 9, 10, 6, 4, 7, 8, 3 = 145
Semut   49 -> 3, 9, 4, 7, 2, 5, 8, 10, 1, 6 = 153
Semut   50 -> 9, 3, 7, 4, 5, 8, 6, 2, 10, 1 = 183
Semut   51 -> 9, 8, 10, 6, 3, 5, 7, 1, 2, 4 = 166
Semut   52 -> 4, 7, 10, 9, 2, 5, 8, 1, 3, 6 = 173
Semut   53 -> 3, 9, 2, 5, 8, 6, 7, 10, 4, 1 = 171
Semut   54 -> 6, 4, 9, 10, 2, 5, 8, 7, 3, 1 = 162
Semut   55 -> 8, 9, 6, 10, 3, 5, 7, 2, 4, 1 = 170
Semut   56 -> 1, 9, 10, 5, 8, 7, 4, 3, 6, 2 = 165
Semut   57 -> 6, 7, 10, 1, 9, 8, 4, 5, 3, 2 = 188
Semut   58 -> 3, 5, 9, 4, 7, 2, 8, 10, 6, 1 = 157
Semut   59 -> 8, 9, 10, 6, 4, 5, 7, 2, 3, 1 = 166
Semut   60 -> 3, 9, 10, 6, 7, 8, 4, 1, 5, 2 = 175
Semut   61 -> 1, 9, 6, 4, 8, 10, 7, 2, 5, 3 = 166
Semut   62 -> 9, 10, 6, 4, 3, 8, 7, 1, 5, 2 = 175
Semut   63 -> 6, 9, 10, 4, 7, 1, 2, 5, 8, 3 = 148
Semut   64 -> 2, 5, 9, 3, 1, 6, 4, 10, 8, 7 = 167
Semut   65 -> 2, 5, 8, 9, 10, 4, 3, 1, 6, 7 = 151
Semut   66 -> 3, 9, 10, 7, 6, 8, 4, 5, 2, 1 = 190
Semut   67 -> 2, 5, 9, 10, 4, 3, 8, 1, 6, 7 = 154
Semut   68 -> 5, 9, 10, 4, 7, 1, 2, 8, 6, 3 = 148
Semut   69 -> 2, 5, 9, 6, 1, 8, 10, 4, 3, 7 = 151
Semut   70 -> 4, 5, 9, 10, 3, 8, 7, 6, 1, 2 = 150
Semut   71 -> 9, 10, 4, 7, 8, 2, 5, 6, 3, 1 = 175
Semut   72 -> 7, 9, 6, 1, 5, 8, 2, 3, 10, 4 = 163
Semut   73 -> 8, 9, 10, 2, 5, 6, 1, 3, 7, 4 = 156
Semut   74 -> 7, 10, 9, 4, 2, 8, 6, 1, 5, 3 = 179
Semut   75 -> 3, 5, 9, 10, 8, 7, 1, 2, 6, 4 = 150
Semut   76 -> 6, 10, 3, 8, 9, 2, 5, 7, 4, 1 = 170
Semut   77 -> 7, 9, 5, 8, 10, 4, 1, 2, 6, 3 = 166
Semut   78 -> 2, 5, 9, 10, 7, 6, 4, 1, 3, 8 = 145
Semut   79 -> 8, 9, 4, 10, 1, 5, 6, 7, 2, 3 = 158
Semut   80 -> 5, 8, 4, 3, 9, 10, 7, 2, 6, 1 = 166
Semut   81 -> 2, 5, 9, 7, 6, 8, 4, 3, 1, 10 = 160
Semut   82 -> 4, 6, 2, 9, 10, 7, 1, 5, 8, 3 = 157
Semut   83 -> 7, 4, 6, 2, 5, 9, 10, 1, 8, 3 = 159
Semut   84 -> 8, 9, 6, 4, 5, 10, 3, 2, 1, 7 = 186
Semut   85 -> 1, 9, 10, 7, 4, 5, 8, 2, 3, 6 = 160
Semut   86 -> 2, 5, 8, 9, 10, 1, 4, 3, 7, 6 = 147
Semut   87 -> 2, 5, 9, 10, 6, 8, 7, 4, 1, 3 = 151
Semut   88 -> 4, 9, 1, 3, 5, 8, 7, 6, 10, 2 = 170
```

```
Semut    89 -> 3, 5, 9, 10, 2, 8, 4, 1, 6, 7 = 153
Semut    90 -> 1, 5, 9, 7, 3, 8, 10, 4, 6, 2 = 154
Semut    91 -> 6, 10, 5, 9, 4, 2, 8, 7, 3, 1 = 173
Semut    92 -> 1, 2, 5, 8, 9, 10, 7, 4, 3, 6 = 139
Semut    93 -> 4, 7, 9, 6, 2, 5, 8, 10, 3, 1 = 165
Semut    94 -> 2, 5, 9, 10, 4, 7, 8, 3, 1, 6 = 160
Semut    95 -> 5, 9, 7, 6, 10, 3, 8, 4, 1, 2 = 158
Semut    96 -> 7, 9, 6, 8, 5, 10, 4, 2, 3, 1 = 187
Semut    97 -> 5, 9, 4, 7, 2, 8, 10, 6, 3, 1 = 166
Semut    98 -> 4, 10, 8, 9, 5, 7, 2, 3, 6, 1 = 182
Semut    99 -> 4, 10, 7, 3, 8, 9, 2, 5, 6, 1 = 165
Semut   100 -> 6, 3, 5, 9, 10, 7, 8, 2, 4, 1 = 169
---------------
Minimal Semut ke 36 -> 6, 1, 2, 5, 8, 9, 10, 4, 3, 7 = 139
```

Setelah memasukan data yang diperlukan pada pengolahan Algoritma Semut yaitu urutan mesin dan frekwensi (jarak) antar mesin, maka diperoleh hasil dari pengolahan urutan mesin yang terpendek.

Dari pengolahan data Algoritma Semut diatas didapat urutan mesin dengan jarak terpendek adalah :

4, 9, 5, 7, 3, 6, 2, 8, 10, 1

Intermediate Matrix adalah martiks yang urutan mesinnya telah optimal, yang akan gunakan dalam mencari urutan komponen terbaik, seperti pada tabel 4.12 dibawah ini :

Tabel 4.12 *Intermediate Matrix* Algoritma Semut

		KOMPONEN											
MESIN		1	2	3	4	5	6	7	8	9	10	11	12
	4	0	0	0	0	0	0	0	0	1	1	1	0
	9	1	1	1	1	1	1	1	1	1	1	1	1
	5	1	1	1	1	1	1	1	1	1	1	1	1
	7	0	0	0	0	0	0	0	0	1	1	1	0
	3	0	0	0	1	1	1	1	1	0	0	0	1
	6	1	1	1	0	0	0	0	0	0	0	0	0
	2	1	1	1	1	1	1	1	1	1	1	1	1
	8	1	1	1	1	1	1	1	1	0	0	0	1
	10	0	0	0	0	0	0	1	0	0	0	0	1
	1	0	0	0	0	0	0	0	0	1	1	1	0

b. Tahap 2

Pada tahap ini bertujuan untuk menempatkan *part* pada posisi yang tepat. Kriteria aspirasi pada tahap 2 ini berbeda dengan tahap 1, dimana kriteria aspirasi berupa nilai minimum total penjumlahan baris dan kolom dari *intermediate matrix* yang telah ada. Penjumlahan baris mengidentifikasikan perbedaan pada mesin dan penjumlahan kolom mengidentifikasikan perbedaan pada *part*. Input yang dibutuhkan adalah *intermediate matrix* yaitu *part-machine matrix* dengan susunan mesin yang optimal.

Sebagai contoh terlebih dahulu akan dihitung nilai kriteria aspirasi untuk solusi awal melalui urutan *part* yang diperoleh dari data perusahaan, yaitu sebagai berikut :

Solusi awal :

1 2 3 4 5 6 7 8 9 10 11 12

- Penjumlahan baris

$$\sum \text{ baris} = \min \sum_{i=1}^{m-1} \sum_{j=1}^{n-1} \left| Pn_j m_i - Pn_j (m+1)_i \right|$$

Misal untuk antara dua mesin 1 dan 2, didapat :

M2	1	1	1	1	1	1	1	1	1	1	1	1
M1	1	1	1	1	1	1	1	1	1	1	1	1
	0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

$$\sum = \mathbf{0}$$

Dengan cara yang sama digunakan untuk mesin yang lain.

- Penjumlahan kolom

$$\sum \text{ kolom} = \min \sum_{i=1}^{m-1} \sum_{j=1}^{n-1} \left| Pn_j m_i - P(n+1)_j m_i \right|$$

Part	**1**	**2**	
	1	1	**0**
	1	1	**0**
	1	1	**0**
	0	0	**0**
	0	0	**0**
	1	1	**0**
	1	1	**0**
	1	1	**0**
	0	0	**0**
		$\sum$	**= 0**

Dengan cara yang sama digunakan untuk *part* yang lain. Sehingga setelah semua perhitungan penjumlahan baris dan kolom dari *intermediate matrix* didapat nilai penjumlahan baris dan kolom sebesar 78. Nilai tersebut digunakan sebagai nilai untuk iterasi pertama.

Setelah memasukan data yang diperlukan pada pengolahan Algoritma Semut. yaitu urutan mesin yang optimal dan solusi awal *part*, maka diperoleh hasil dari pengolahan penjumlahan baris dan kolom terkecil.

Contoh iterasi pada tahap 2 setelah melalui Algoritma Semut didapatkan hasil seperti dibawah ini untuk 2 siklus pertama :

```
Siklus ke: 1
Intens Jejak Semut: 0.1
---------------
Semut     1 -> 6, 10, 12, 3, 7, 9, 1, 4, 8, 11, 2, 5 = 36
Semut     2 -> 8, 11, 2, 5, 7, 10, 1, 4, 9, 12, 3, 6 = 36
Semut     3 -> 8, 11, 2, 5, 9, 12, 1, 6, 7, 10, 3, 4 = 36
Semut     4 -> 10, 1, 4, 7, 11, 2, 5, 8, 12, 3, 6, 9 = 28
```

```
Semut     5 -> 11, 2, 5, 8, 10, 1, 6, 9, 12, 3, 4, 7 = 31
Semut     6 -> 12, 3, 6, 9, 11, 2, 5, 8, 10, 1, 4, 7 = 26
Semut     7 -> 1, 5, 8, 11, 2, 6, 9, 10, 3, 4, 12, 7 = 23
Semut     8 -> 2, 5, 8, 11, 1, 6, 9, 12, 3, 7, 10, 4 = 36
Semut     9 -> 3, 6, 9, 12, 2, 7, 10, 11, 1, 5, 8, 4 = 28
Semut    10 -> 4, 7, 10, 1, 5, 8, 11, 2, 6, 9, 12, 3 = 34
Semut    11 -> 5, 8, 11, 1, 4, 9, 12, 2, 6, 7, 10, 3 = 34
Semut    12 -> 6, 9, 12, 2, 5, 8, 11, 3, 4, 10, 7, 1 = 36
Semut    13 -> 7, 10, 1, 4, 8, 11, 2, 5, 9, 12, 3, 6 = 35
Semut    14 -> 8, 12, 2, 5, 9, 11, 3, 6, 7, 10, 1, 4 = 28
Semut    15 -> 10, 1, 4, 7, 11, 3, 8, 12, 5, 6, 9, 2 = 28
Semut    16 -> 7, 10, 1, 4, 8, 12, 3, 9, 11, 2, 5, 6 = 25
Semut    17 -> 1, 5, 8, 11, 2, 6, 9, 12, 3, 7, 10, 4 = 36
Semut    18 -> 3, 6, 9, 12, 2, 7, 10, 11, 1, 5, 8, 4 = 28
Semut    19 -> 4, 8, 10, 1, 5, 7, 11, 2, 6, 9, 12, 3 = 34
Semut    20 -> 5, 9, 12, 2, 6, 8, 11, 1, 4, 7, 10, 3 = 34
Semut    21 -> 6, 10, 12, 3, 5, 8, 11, 2, 4, 9, 7, 1 = 36
Semut    22 -> 7, 10, 1, 4, 8, 11, 2, 5, 9, 12, 3, 6 = 35
Semut    23 -> 8, 12, 2, 5, 9, 11, 1, 6, 7, 10, 3, 4 = 28
Semut    24 -> 9, 12, 3, 6, 8, 11, 2, 5, 7, 10, 1, 4 = 32
Semut    25 -> 11, 2, 5, 8, 10, 1, 6, 7, 12, 3, 4, 9 = 26
Semut    26 -> 12, 3, 6, 9, 11, 2, 5, 8, 10, 1, 4, 7 = 26
Semut    27 -> 1, 5, 8, 10, 2, 4, 9, 11, 3, 6, 12, 7 = 23
Semut    28 -> 2, 5, 8, 11, 1, 6, 9, 12, 3, 7, 10, 4 = 36
Semut    29 -> 3, 7, 10, 12, 2, 6, 9, 11, 1, 5, 8, 4 = 28
Semut    30 -> 4, 7, 10, 1, 5, 8, 11, 2, 6, 9, 12, 3 = 34
Semut    31 -> 5, 9, 11, 1, 4, 8, 12, 2, 6, 7, 10, 3 = 26
Semut    32 -> 6, 9, 12, 2, 5, 8, 11, 1, 4, 10, 7, 3 = 36
Semut    33 -> 7, 10, 1, 4, 8, 11, 2, 5, 9, 12, 3, 6 = 35
Semut    34 -> 8, 11, 2, 5, 9, 12, 1, 4, 7, 10, 3, 6 = 36
Semut    35 -> 9, 1, 4, 7, 11, 2, 5, 8, 12, 3, 6, 10 = 28
Semut    36 -> 11, 2, 5, 8, 10, 1, 4, 7, 12, 3, 6, 9 = 26
Semut    37 -> 12, 3, 6, 9, 11, 2, 5, 8, 10, 1, 4, 7 = 26
Semut    38 -> 1, 5, 8, 11, 2, 6, 9, 10, 3, 4, 12, 7 = 23
Semut    39 -> 2, 6, 9, 11, 1, 5, 8, 12, 3, 7, 10, 4 = 28
Semut    40 -> 2, 6, 9, 12, 3, 5, 8, 11, 1, 7, 10, 4 = 36
Semut    41 -> 3, 7, 10, 12, 2, 6, 9, 11, 1, 5, 8, 4 = 28
Semut    42 -> 5, 8, 11, 1, 4, 9, 12, 2, 6, 7, 10, 3 = 34
Semut    43 -> 6, 9, 12, 2, 5, 8, 11, 3, 4, 10, 7, 1 = 36
Semut    44 -> 7, 10, 1, 4, 8, 11, 2, 5, 9, 12, 3, 6 = 35
Semut    45 -> 8, 11, 2, 5, 9, 12, 1, 4, 7, 10, 3, 6 = 36
Semut    46 -> 9, 12, 3, 6, 8, 11, 2, 5, 7, 10, 1, 4 = 32
Semut    47 -> 9, 12, 3, 6, 10, 1, 4, 7, 11, 2, 5, 8 = 32
Semut    48 -> 11, 2, 5, 8, 10, 1, 6, 9, 12, 3, 4, 7 = 31
Semut    49 -> 12, 3, 6, 9, 11, 2, 5, 8, 10, 1, 4, 7 = 26
Semut    50 -> 1, 5, 8, 11, 2, 6, 9, 12, 3, 4, 10, 7 = 35
Semut    51 -> 2, 6, 9, 12, 3, 5, 8, 11, 1, 7, 10, 4 = 36
Semut    52 -> 3, 7, 10, 1, 5, 8, 11, 2, 6, 9, 12, 4 = 34
Semut    53 -> 5, 8, 11, 1, 4, 9, 12, 2, 6, 7, 10, 3 = 34
Semut    54 -> 6, 9, 12, 2, 5, 10, 11, 3, 4, 8, 7, 1 = 28
Semut    55 -> 7, 11, 1, 4, 8, 10, 2, 5, 9, 12, 3, 6 = 35
```

```
Semut    56 -> 8, 11, 2, 5, 9, 12, 1, 6, 7, 10, 3, 4 = 36
Semut    57 -> 9, 1, 4, 7, 11, 2, 5, 8, 10, 12, 3, 6 = 32
Semut    58 -> 10, 1, 4, 7, 11, 2, 5, 8, 12, 3, 6, 9 = 28
Semut    59 -> 11, 2, 6, 8, 12, 3, 5, 9, 10, 1, 4, 7 = 23
Semut    60 -> 1, 4, 8, 10, 12, 3, 6, 9, 11, 2, 5, 7 = 27
Semut    61 -> 1, 5, 8, 11, 2, 6, 9, 12, 3, 7, 10, 4 = 36
Semut    62 -> 3, 6, 9, 12, 2, 5, 8, 11, 1, 7, 10, 4 = 36
Semut    63 -> 4, 7, 10, 1, 5, 8, 11, 2, 6, 9, 12, 3 = 34
Semut    64 -> 5, 8, 11, 2, 4, 9, 12, 1, 6, 7, 10, 3 = 34
Semut    65 -> 6, 9, 12, 2, 5, 8, 11, 3, 4, 10, 7, 1 = 36
Semut    66 -> 7, 10, 1, 4, 8, 11, 2, 5, 9, 12, 3, 6 = 35
Semut    67 -> 8, 11, 2, 5, 9, 10, 1, 4, 7, 12, 3, 6 = 26
Semut    68 -> 9, 12, 3, 6, 8, 11, 2, 5, 7, 10, 1, 4 = 32
Semut    69 -> 10, 1, 4, 7, 11, 2, 5, 8, 12, 3, 6, 9 = 28
Semut    70 -> 11, 2, 5, 8, 12, 1, 6, 9, 10, 3, 4, 7 = 23
Semut    71 -> 12, 3, 6, 9, 11, 2, 5, 8, 10, 1, 4, 7 = 26
Semut    72 -> 1, 5, 8, 11, 2, 6, 9, 12, 3, 4, 10, 7 = 35
Semut    73 -> 2, 6, 9, 12, 3, 5, 8, 11, 1, 7, 10, 4 = 36
Semut    74 -> 3, 7, 10, 1, 5, 8, 11, 12, 2, 6, 9, 4 = 36
Semut    75 -> 4, 7, 10, 1, 5, 8, 11, 2, 6, 9, 12, 3 = 34
Semut    76 -> 5, 9, 12, 2, 6, 8, 11, 1, 4, 7, 10, 3 = 34
Semut    77 -> 7, 10, 1, 4, 8, 11, 2, 5, 6, 12, 9, 3 = 31
Semut    78 -> 7, 11, 1, 5, 8, 10, 2, 4, 9, 12, 3, 6 = 35
Semut    79 -> 9, 12, 2, 5, 8, 11, 3, 6, 7, 10, 1, 4 = 32
Semut    80 -> 10, 1, 4, 7, 11, 2, 5, 8, 12, 3, 6, 9 = 28
Semut    81 -> 11, 2, 5, 8, 10, 1, 6, 7, 12, 3, 4, 9 = 26
Semut    82 -> 12, 3, 6, 9, 11, 2, 5, 8, 10, 1, 4, 7 = 26
Semut    83 -> 1, 4, 8, 10, 2, 5, 9, 11, 3, 6, 12, 7 = 23
Semut    84 -> 2, 6, 9, 11, 3, 5, 8, 12, 1, 7, 10, 4 = 28
Semut    85 -> 3, 7, 10, 12, 2, 6, 9, 11, 1, 5, 8, 4 = 28
Semut    86 -> 5, 9, 1, 6, 11, 2, 7, 10, 12, 3, 4, 8 = 36
Semut    87 -> 2, 6, 10, 3, 7, 9, 12, 1, 5, 8, 11, 4 = 36
Semut    88 -> 6, 9, 12, 2, 5, 8, 11, 1, 4, 10, 7, 3 = 36
Semut    89 -> 7, 10, 1, 4, 8, 11, 2, 5, 9, 12, 3, 6 = 35
Semut    90 -> 8, 11, 2, 5, 7, 12, 1, 4, 9, 10, 3, 6 = 26
Semut    91 -> 9, 12, 3, 6, 8, 11, 2, 5, 7, 10, 1, 4 = 32
Semut    92 -> 10, 1, 4, 7, 11, 2, 5, 8, 12, 3, 6, 9 = 28
Semut    93 -> 11, 2, 6, 8, 12, 3, 5, 9, 10, 1, 4, 7 = 23
Semut    94 -> 1, 4, 7, 10, 12, 3, 6, 9, 11, 2, 5, 8 = 28
Semut    95 -> 1, 5, 8, 11, 2, 6, 9, 12, 3, 7, 10, 4 = 36
Semut    96 -> 3, 6, 9, 12, 2, 5, 8, 11, 1, 7, 10, 4 = 36
Semut    97 -> 4, 7, 10, 1, 5, 8, 11, 2, 6, 9, 12, 3 = 34
Semut    98 -> 5, 8, 11, 2, 6, 9, 12, 1, 4, 7, 10, 3 = 34
Semut    99 -> 6, 10, 12, 3, 5, 9, 11, 2, 4, 8, 7, 1 = 28
Semut   100 -> 7, 10, 1, 4, 8, 11, 2, 5, 9, 12, 3, 6 = 35
---------------
Minimal Semut ke 7 -> 1, 5, 8, 11, 2, 6, 9, 10, 3, 4, 12, 7
= 23
```

```
Siklus ke: 2
Intens Jejak Semut: 325.801381139478
---------------
Semut     1 -> 1, 5, 9, 11, 2, 6, 8, 12, 3, 7, 10, 4 = 28
Semut     2 -> 5, 8, 11, 2, 6, 9, 12, 1, 4, 7, 10, 3 = 34
Semut     3 -> 8, 11, 1, 4, 7, 10, 2, 5, 9, 12, 3, 6 = 36
Semut     4 -> 10, 1, 5, 8, 11, 2, 4, 7, 12, 3, 6, 9 = 26
Semut     5 -> 1, 5, 8, 10, 2, 4, 9, 11, 3, 6, 12, 7 = 23
Semut     6 -> 3, 7, 10, 12, 2, 6, 9, 11, 1, 5, 8, 4 = 28
Semut     7 -> 6, 9, 12, 2, 5, 8, 11, 1, 4, 10, 7, 3 = 36
Semut     8 -> 10, 1, 5, 7, 11, 2, 4, 8, 12, 3, 6, 9 = 28
Semut     9 -> 1, 4, 8, 10, 2, 5, 9, 11, 3, 6, 7, 12 = 23
Semut    10 -> 3, 6, 10, 12, 2, 5, 9, 11, 1, 7, 8, 4 = 28
Semut    11 -> 5, 9, 12, 2, 6, 8, 11, 1, 4, 7, 10, 3 = 34
Semut    12 -> 8, 11, 2, 5, 7, 12, 1, 4, 9, 10, 3, 6 = 26
Semut    13 -> 11, 1, 5, 8, 10, 2, 6, 7, 12, 3, 4, 9 = 26
Semut    14 -> 1, 5, 8, 11, 2, 6, 9, 12, 3, 4, 10, 7 = 35
Semut    15 -> 4, 8, 11, 1, 5, 7, 10, 2, 6, 9, 12, 3 = 34
Semut    16 -> 7, 10, 1, 4, 8, 11, 2, 5, 9, 12, 3, 6 = 35
Semut    17 -> 9, 12, 3, 6, 8, 11, 2, 5, 7, 10, 1, 4 = 32
Semut    18 -> 12, 3, 6, 9, 11, 2, 5, 8, 10, 1, 4, 7 = 26
Semut    19 -> 2, 6, 9, 12, 3, 5, 10, 11, 1, 7, 8, 4 = 28
Semut    20 -> 5, 8, 11, 1, 4, 9, 12, 2, 6, 7, 10, 3 = 34
Semut    21 -> 7, 10, 1, 4, 8, 11, 2, 5, 9, 12, 3, 6 = 35
Semut    22 -> 10, 1, 4, 7, 11, 2, 5, 8, 12, 3, 6, 9 = 28
Semut    23 -> 12, 3, 7, 9, 11, 2, 5, 8, 10, 1, 4, 6 = 27
Semut    24 -> 3, 6, 9, 12, 2, 5, 10, 11, 1, 7, 8, 4 = 28
Semut    25 -> 5, 9, 12, 2, 6, 8, 11, 1, 4, 7, 10, 3 = 34
Semut    26 -> 7, 10, 1, 4, 8, 11, 2, 5, 9, 12, 3, 6 = 35
Semut    27 -> 10, 1, 4, 7, 11, 2, 5, 8, 9, 12, 3, 6 = 32
Semut    28 -> 12, 3, 6, 9, 11, 2, 5, 8, 10, 1, 4, 7 = 26
Semut    29 -> 3, 7, 10, 12, 2, 6, 9, 11, 1, 5, 8, 4 = 28
Semut    30 -> 6, 9, 12, 2, 5, 8, 11, 1, 4, 7, 10, 3 = 34
Semut    31 -> 8, 11, 2, 5, 7, 12, 1, 4, 9, 10, 3, 6 = 26
Semut    32 -> 11, 1, 5, 8, 10, 2, 6, 7, 12, 3, 4, 9 = 26
Semut    33 -> 1, 5, 8, 11, 2, 6, 9, 10, 3, 4, 12, 7 = 23
Semut    34 -> 3, 7, 10, 12, 2, 6, 9, 11, 1, 5, 8, 4 = 28
Semut    35 -> 6, 9, 12, 2, 5, 8, 11, 1, 4, 10, 7, 3 = 36
Semut    36 -> 9, 12, 2, 5, 8, 11, 3, 6, 7, 10, 1, 4 = 32
Semut    37 -> 11, 2, 6, 8, 12, 3, 5, 9, 10, 1, 4, 7 = 23
Semut    38 -> 1, 5, 8, 11, 2, 6, 9, 10, 3, 4, 12, 7 = 23
Semut    39 -> 3, 6, 9, 12, 2, 7, 10, 11, 1, 5, 8, 4 = 28
Semut    40 -> 5, 8, 11, 2, 6, 9, 12, 1, 4, 7, 10, 3 = 34
Semut    41 -> 8, 11, 2, 5, 7, 12, 1, 4, 9, 10, 3, 6 = 26
Semut    42 -> 11, 2, 5, 8, 10, 1, 4, 7, 12, 3, 6, 9 = 26
Semut    43 -> 1, 4, 8, 10, 12, 3, 6, 9, 11, 2, 5, 7 = 27
Semut    44 -> 3, 7, 10, 12, 2, 6, 9, 11, 1, 5, 8, 4 = 28
Semut    45 -> 6, 9, 12, 2, 5, 8, 11, 3, 4, 10, 7, 1 = 36
Semut    46 -> 9, 12, 2, 5, 8, 11, 1, 6, 7, 10, 3, 4 = 32
Semut    47 -> 11, 2, 5, 8, 12, 1, 6, 9, 10, 3, 4, 7 = 23
Semut    48 -> 2, 5, 9, 11, 1, 6, 8, 12, 3, 7, 10, 4 = 28
```

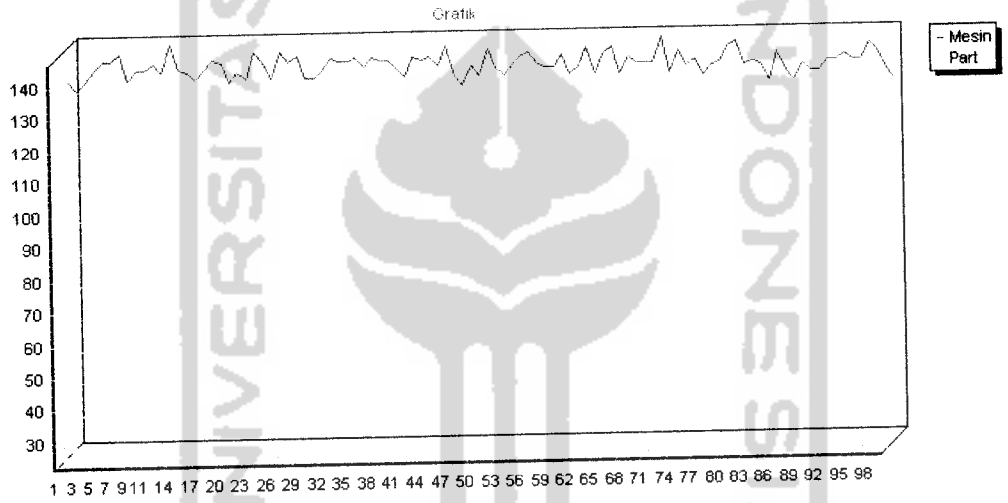
```
Semut   49 -> 5, 8, 11, 1, 4, 7, 10, 2, 6, 9, 12, 3 = 34
Semut   50 -> 7, 10, 1, 4, 8, 11, 2, 5, 9, 12, 6, 3 = 33
Semut   51 -> 9, 12, 3, 7, 11, 2, 6, 10, 4, 5, 8, 1 = 32
Semut   52 -> 9, 1, 5, 10, 2, 4, 8, 11, 3, 7, 12, 6 = 28
Semut   53 -> 6, 9, 12, 2, 7, 11, 3, 8, 10, 1, 4, 5 = 36
Semut   54 -> 1, 5, 8, 11, 2, 6, 9, 12, 3, 7, 10, 4 = 36
Semut   55 -> 4, 8, 11, 1, 5, 7, 10, 2, 6, 9, 12, 3 = 34
Semut   56 -> 6, 10, 12, 3, 5, 9, 11, 2, 4, 8, 7, 1 = 28
Semut   57 -> 9, 12, 3, 6, 8, 11, 2, 5, 7, 10, 1, 4 = 32
Semut   58 -> 12, 3, 6, 9, 11, 2, 5, 8, 10, 1, 4, 7 = 26
Semut   59 -> 2, 6, 9, 12, 3, 5, 8, 11, 1, 7, 10, 4 = 36
Semut   60 -> 5, 8, 11, 2, 6, 9, 12, 1, 4, 7, 10, 3 = 34
Semut   61 -> 8, 11, 2, 5, 7, 12, 1, 4, 9, 10, 3, 6 = 26
Semut   62 -> 11, 2, 5, 8, 10, 1, 6, 9, 12, 3, 4, 7 = 31
Semut   63 -> 1, 5, 8, 11, 2, 6, 9, 12, 3, 7, 10, 4 = 36
Semut   64 -> 4, 8, 11, 1, 5, 7, 10, 2, 6, 9, 12, 3 = 34
Semut   65 -> 7, 10, 1, 4, 8, 11, 2, 5, 9, 12, 3, 6 = 35
Semut   66 -> 10, 1, 4, 7, 11, 2, 5, 8, 12, 3, 6, 9 = 28
Semut   67 -> 1, 4, 8, 10, 2, 5, 9, 11, 3, 6, 12, 7 = 23
Semut   68 -> 3, 7, 10, 12, 2, 6, 9, 11, 1, 5, 8, 4 = 28
Semut   69 -> 6, 9, 12, 2, 5, 8, 11, 1, 4, 10, 7, 3 = 36
Semut   70 -> 9, 12, 2, 5, 8, 11, 3, 6, 7, 10, 1, 4 = 32
Semut   71 -> 11, 2, 6, 8, 12, 3, 5, 9, 10, 1, 4, 7 = 23
Semut   72 -> 2, 6, 9, 11, 3, 5, 8, 12, 1, 7, 10, 4 = 28
Semut   73 -> 5, 8, 11, 1, 4, 9, 12, 2, 6, 7, 10, 3 = 34
Semut   74 -> 7, 11, 1, 4, 8, 10, 2, 5, 9, 12, 3, 6 = 35
Semut   75 -> 10, 1, 5, 7, 11, 2, 4, 8, 12, 3, 6, 9 = 28
Semut   76 -> 1, 5, 8, 10, 2, 4, 9, 11, 3, 6, 12, 7 = 23
Semut   77 -> 3, 7, 10, 12, 2, 6, 9, 11, 1, 5, 8, 4 = 28
Semut   78 -> 6, 9, 12, 2, 5, 10, 11, 3, 4, 8, 7, 1 = 28
Semut   79 -> 9, 12, 2, 6, 8, 11, 3, 5, 7, 10, 1, 4 = 32
Semut   80 -> 12, 2, 6, 9, 11, 3, 5, 8, 10, 1, 4, 7 = 26
Semut   81 -> 2, 6, 9, 11, 1, 5, 8, 12, 3, 7, 10, 4 = 28
Semut   82 -> 4, 8, 11, 1, 5, 9, 12, 2, 6, 7, 10, 3 = 34
Semut   83 -> 7, 10, 1, 4, 8, 11, 2, 5, 9, 12, 3, 6 = 35
Semut   84 -> 10, 1, 4, 7, 11, 2, 5, 8, 12, 3, 6, 9 = 28
Semut   85 -> 12, 3, 6, 9, 11, 2, 5, 8, 10, 1, 4, 7 = 26
Semut   86 -> 3, 6, 9, 12, 2, 7, 10, 11, 1, 5, 8, 4 = 28
Semut   87 -> 6, 9, 12, 2, 5, 8, 11, 1, 4, 7, 10, 3 = 34
Semut   88 -> 8, 11, 2, 5, 9, 10, 1, 4, 7, 12, 3, 6 = 26
Semut   89 -> 10, 1, 4, 7, 11, 2, 5, 8, 9, 12, 3, 6 = 32
Semut   90 -> 12, 3, 6, 9, 11, 2, 5, 8, 10, 1, 4, 7 = 26
Semut   91 -> 3, 6, 10, 12, 2, 7, 9, 11, 1, 5, 8, 4 = 28
Semut   92 -> 6, 9, 12, 2, 5, 8, 11, 1, 4, 10, 7, 3 = 36
Semut   93 -> 8, 12, 2, 5, 9, 11, 1, 4, 7, 10, 3, 6 = 28
Semut   94 -> 11, 2, 5, 8, 10, 1, 6, 9, 12, 3, 4, 7 = 31
Semut   95 -> 1, 5, 8, 11, 2, 6, 9, 12, 3, 4, 10, 7 = 35
Semut   96 -> 4, 7, 10, 1, 5, 8, 11, 2, 6, 9, 12, 3 = 34
Semut   97 -> 6, 10, 1, 4, 8, 11, 2, 5, 9, 12, 3, 7 = 35
Semut   98 -> 9, 12, 3, 6, 8, 11, 2, 5, 7, 10, 1, 4 = 32
Semut   99 -> 12, 3, 6, 9, 11, 2, 5, 8, 10, 1, 4, 7 = 26
```

```
Semut  100 -> 2, 6, 9, 12, 3, 5, 10, 11, 1, 7, 8, 4 = 28
---------------
Minimal Semut ke 5 -> 1, 5, 8, 10, 2, 4, 9, 11, 3, 6, 12, 7
= 23
```

Dari pengolahan data Algoritma Semut diatas didapat urutan terbaik dari *part* adalah :

11, 2, 3, 6, 8, 12, 1, 5, 9, 10,4, 7

Berikut adalah grafik panjang rute setiap semut untuk pencarian mesin dan *part*



Gambar 4.5 Grafik Panjang Rute Setiap Semut

4.2.4.2 Pengolahan Data dengan Algoritma *Tabu Search*

Dalam penelitian ini terdapat dua fase yang harus diselesaikan yaitu fase 1 untuk penyusunan mesin dan fase 2 untuk penyusunan part. Pengolahan data-data akan melalui tahapan Algoritma Tabu Search.

a. Fase 1

Fase 1 ini bertujuan untuk mengurutkan kembali letak mesin sehingga dengan posisi mesin yang baru didapat jarak material yang lebih pendek. Fase ini diawali dengan membangun table material handling antar mesin, dimana table ini mengindikasikan total nilai dari perjalanan material yang dilakukan antara 2 mesin. Perhitungan jarak antar mesin dipengaruhi oleh nilai mesin-mesin material handling perpindahan part antara dua mesin, dimana nilai frekwensi dihitung berdasarkan formula rumus yang digunakan yang mengidentifikasikan adanya *forward* atau *backward*, dan *incidence matrix* yang hasilnya tercantum pada tabel 4.10

$di = C - fi$

fi adalah nilai material handling part

fi, jika forward

$fi(-1)$, jika backward

Sebagai contoh akan dihitung jarak mesin 1 ke 2 dan mesin 2 ke 1, dengan menggunakan konstanta (C) = 30, dengan banyaknya perpindahan part (fi) antara mesin 1 dan mesin 2 adalah sebanyak 8 kali :

- *Mesin 1 ke 2*

$di = 30 - 8 = 22$

- *Mesin 2 ke 1*

$di = 30 - 8(-1) = 38$

Nilai frekwensi (jarak) didapat dari banyaknya perpindahan part dari mesin satu ke mesin lainnya, dengan memperhitungkan urutan maju dan urutan mundur dari suatu part yang diproses dalam mesin seperti pada contoh diatas. Berikut adalah nilai frekwensi (jarak) dari setiap mesin :

Tabel 4.13 Frekwensi (jarak) Antar Mesin

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	0	8	14	17	8	17	17	11	8	18
2	32	0	14	17	8	17	17	11	8	18
3	26	26	0	20	14	20	20	14	14	18
4	23	23	20	0	17	20	17	20	17	20
5	32	32	26	23	0	17	17	11	8	18
6	23	23	20	20	23	0	20	17	17	20
7	23	23	20	23	23	20	0	11	17	20
8	29	29	26	20	29	23	20	0	11	18
9	32	32	26	23	32	23	23	29	0	18
10	22	22	22	20	22	20	20	22	22	0

Algoritma Tabu Search digunakan untuk mencari solusi terbaik dari tugas akhir ini. Dimana input data yang diperlukan untuk melakukan pengolahan di algoritma ini yaitu frekwensi (jarak) antar mesin, dengan jumlah iterasi yang digunakan yaitu 50 iterasi, *move* yang digunakan yaitu *neighborhood search*.

Setelah memasukan data yang diperlukan pada pengolahan *Algoritma Tabu Search* yaitu urutan mesin dan frekwensi (jarak) antar mesin, maka diperoleh hasil dari pengolahan urutan mesin yang terpendek.

Contoh iterasi urutan pengolahan mesin setelah melalui *Algoritma Tabu Seacrh* didapatkan hasil seperti dibawah ini untuk iterasi pertama :

```
Pencarian Urutan Mesin:
-------------------------------------------------------------
---------------
Iterasi ke: 1
Global Optimum:
1, 2, 3, 4, 5, 6, 7, 8, 9, 10: 145
Local Optimum:
1, 2, 3, 4, 5, 6, 7, 8, 9, 10: 145
-------------------------------------------------------------
---------------
Swap[1,2] = 2, 1, 3, 4, 5, 6, 7, 8, 9, 10: 169
Swap[1,3] = 3, 2, 1, 4, 5, 6, 7, 8, 9, 10: 178
Swap[1,4] = 4, 2, 3, 1, 5, 6, 7, 8, 9, 10: 157
Swap[1,5] = 5, 2, 3, 4, 1, 6, 7, 8, 9, 10: 175
Swap[1,6] = 6, 2, 3, 4, 5, 1, 7, 8, 9, 10: 172
Swap[1,7] = 7, 2, 3, 4, 5, 6, 1, 8, 9, 10: 154
Swap[1,8] = 8, 2, 3, 4, 5, 6, 7, 1, 9, 10: 166
Swap[1,9] = 9, 2, 3, 4, 5, 6, 7, 8, 1, 10: 187
Swap[1,10] = 10, 2, 3, 4, 5, 6, 7, 8, 9, 1: 173
Swap[2,3] = 1, 3, 2, 4, 5, 6, 7, 8, 9, 10: 160
Swap[2,4] = 1, 4, 3, 2, 5, 6, 7, 8, 9, 10: 157
Swap[2,5] = 1, 5, 3, 4, 2, 6, 7, 8, 9, 10: 163
Swap[2,6] = 1, 6, 3, 4, 5, 2, 7, 8, 9, 10: 172
Swap[2,7] = 1, 7, 3, 4, 5, 6, 2, 8, 9, 10: 154
Swap[2,8] = 1, 8, 3, 4, 5, 6, 7, 2, 9, 10: 160
Swap[2,9] = 1, 9, 3, 4, 5, 6, 7, 8, 2, 10: 175
Swap[2,10] = 1, 10, 3, 4, 5, 6, 7, 8, 9, 2: 177
Swap[3,4] = 1, 2, 4, 3, 5, 6, 7, 8, 9, 10: 145
Swap[3,5] = 1, 2, 5, 4, 3, 6, 7, 8, 9, 10: 148
Swap[3,6] = 1, 2, 6, 4, 5, 3, 7, 8, 9, 10: 157
Swap[3,7] = 1, 2, 7, 4, 5, 6, 3, 8, 9, 10: 145
Swap[3,8] = 1, 2, 8, 4, 5, 6, 7, 3, 9, 10: 145
Swap[3,9] = 1, 2, 9, 4, 5, 6, 7, 8, 3, 10: 157
Swap[3,10] = 1, 2, 10, 4, 5, 6, 7, 8, 9, 3: 157
Swap[4,5] = 1, 2, 3, 5, 4, 6, 7, 8, 9, 10: 148
Swap[4,6] = 1, 2, 3, 6, 5, 4, 7, 8, 9, 10: 154
Swap[4,7] = 1, 2, 3, 7, 5, 6, 4, 8, 9, 10: 151
Swap[4,8] = 1, 2, 3, 8, 5, 6, 7, 4, 9, 10: 160
Swap[4,9] = 1, 2, 3, 9, 5, 6, 7, 8, 4, 10: 165
Swap[4,10] = 1, 2, 3, 10, 5, 6, 7, 8, 9, 4: 153
Swap[5,6] = 1, 2, 3, 4, 6, 5, 7, 8, 9, 10: 151
Swap[5,7] = 1, 2, 3, 4, 7, 6, 5, 8, 9, 10: 142
Swap[5,8] = 1, 2, 3, 4, 8, 6, 7, 5, 9, 10: 154
Swap[5,9] = 1, 2, 3, 4, 9, 6, 7, 8, 5, 10: 169
Swap[5,10] = 1, 2, 3, 4, 10, 6, 7, 8, 9, 5: 165
Swap[6,7] = 1, 2, 3, 4, 5, 7, 6, 8, 9, 10: 142
Swap[6,8] = 1, 2, 3, 4, 5, 8, 7, 6, 9, 10: 145
Swap[6,9] = 1, 2, 3, 4, 5, 9, 7, 8, 6, 10: 153
Swap[6,10] = 1, 2, 3, 4, 5, 10, 7, 8, 9, 6: 151
Swap[7,8] = 1, 2, 3, 4, 5, 6, 8, 7, 9, 10: 148
Swap[7,9] = 1, 2, 3, 4, 5, 6, 9, 8, 7, 10: 162
```

```
Swap[7,10] = 1, 2, 3, 4, 5, 6, 10, 8, 9, 7: 152
Swap[8,9] = 1, 2, 3, 4, 5, 6, 7, 9, 8, 10: 160
Swap[8,10] = 1, 2, 3, 4, 5, 6, 7, 10, 9, 8: 167
Swap[9,10] = 1, 2, 3, 4, 5, 6, 7, 8, 10, 9: 156
-------------------------------------------------------------
---------------
Iterasi ke: 2
Global Optimum:
1, 2, 3, 4, 7, 6, 5, 8, 9, 10: 142
Local Optimum:
1, 2, 3, 4, 7, 6, 5, 8, 9, 10: 142
-------------------------------------------------------------
---------------
Swap[1,2] = 2, 1, 3, 4, 7, 6, 5, 8, 9, 10: 166
Swap[1,3] = 3, 2, 1, 4, 7, 6, 5, 8, 9, 10: 175
Swap[1,4] = 4, 2, 3, 1, 7, 6, 5, 8, 9, 10: 163
Swap[1,5] = 7, 2, 3, 4, 1, 6, 5, 8, 9, 10: 160
Swap[1,6] = 6, 2, 3, 4, 7, 1, 5, 8, 9, 10: 145
Swap[1,7] = 5, 2, 3, 4, 7, 6, 1, 8, 9, 10: 166
Swap[1,8] = 8, 2, 3, 4, 7, 6, 5, 1, 9, 10: 181
Swap[1,9] = 9, 2, 3, 4, 7, 6, 5, 8, 1, 10: 184
Swap[1,10] = 10, 2, 3, 4, 7, 6, 5, 8, 9, 1: 170
Swap[2,3] = 1, 3, 2, 4, 7, 6, 5, 8, 9, 10: 157
Swap[2,4] = 1, 4, 3, 2, 7, 6, 5, 8, 9, 10: 163
Swap[2,5] = 1, 7, 3, 4, 2, 6, 5, 8, 9, 10: 160
Swap[2,6] = 1, 6, 3, 4, 7, 2, 5, 8, 9, 10: 145
Swap[2,7] = 1, 5, 3, 4, 7, 6, 2, 8, 9, 10: 154
Swap[2,8] = 1, 8, 3, 4, 7, 6, 5, 2, 9, 10: 175
Swap[2,9] = 1, 9, 3, 4, 7, 6, 5, 8, 2, 10: 172
Swap[2,10] = 1, 10, 3, 4, 7, 6, 5, 8, 9, 2: 174
Swap[3,4] = 1, 2, 4, 3, 7, 6, 5, 8, 9, 10: 148
Swap[3,5] = 1, 2, 7, 4, 3, 6, 5, 8, 9, 10: 151
Swap[3,6] = 1, 2, 6, 4, 7, 3, 5, 8, 9, 10: 136
Swap[3,7] = 1, 2, 5, 4, 7, 6, 3, 8, 9, 10: 139
Swap[3,8] = 1, 2, 8, 4, 7, 6, 5, 3, 9, 10: 157
Swap[3,9] = 1, 2, 9, 4, 7, 6, 5, 8, 3, 10: 154
Swap[3,10] = 1, 2, 10, 4, 7, 6, 5, 8, 9, 3: 154
Swap[4,5] = 1, 2, 3, 7, 4, 6, 5, 8, 9, 10: 148
Swap[4,6] = 1, 2, 3, 6, 7, 4, 5, 8, 9, 10: 142
Swap[4,7] = 1, 2, 3, 5, 7, 6, 4, 8, 9, 10: 142
Swap[4,8] = 1, 2, 3, 8, 7, 6, 5, 4, 9, 10: 157
Swap[4,9] = 1, 2, 3, 9, 7, 6, 5, 8, 4, 10: 153
Swap[4,10] = 1, 2, 3, 10, 7, 6, 5, 8, 9, 4: 148
Swap[5,6] = 1, 2, 3, 4, 6, 7, 5, 8, 9, 10: 145
Swap[5,7] = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10: 145 --> Sudah di
tabulist
Swap[5,8] = 1, 2, 3, 4, 8, 6, 5, 7, 9, 10: 160
Swap[5,9] = 1, 2, 3, 4, 9, 6, 5, 8, 7, 10: 156
Swap[5,10] = 1, 2, 3, 4, 10, 6, 5, 8, 9, 7: 150
Swap[6,7] = 1, 2, 3, 4, 7, 5, 6, 8, 9, 10: 145
Swap[6,8] = 1, 2, 3, 4, 7, 8, 5, 6, 9, 10: 160
```

```
Swap[6,9] = 1, 2, 3, 4, 7, 9, 5, 8, 6, 10: 162
Swap[6,10] = 1, 2, 3, 4, 7, 10, 5, 8, 9, 6: 146
Swap[7,8] = 1, 2, 3, 4, 7, 6, 8, 5, 9, 10: 151
Swap[7,9] = 1, 2, 3, 4, 7, 6, 9, 8, 5, 10: 172
Swap[7,10] = 1, 2, 3, 4, 7, 6, 10, 8, 9, 5: 164
Swap[8,9] = 1, 2, 3, 4, 7, 6, 5, 9, 8, 10: 157
Swap[8,10] = 1, 2, 3, 4, 7, 6, 5, 10, 9, 8: 171
Swap[9,10] = 1, 2, 3, 4, 7, 6, 5, 8, 10, 9: 153
------------------------------------------------------------
--------------
```

Dari pengolahan data Algoritma *Tabu Search* diatas didapat urutan mesin dengan jarak terpendek adalah :

1 2 5 8 9 10 4 7 3 6

Intermediate Matrix adalah martiks yang urutan mesinnya telah optimal, yang akan gunakan dalam mencari urutan komponen terbaik, seperti pada tabel 4.14 dibawah ini :

Table 4.14 *Intermediate Matrix*Algoritma *Tabu Search*

		KOMPONEN											
MESIN		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
	1	1	1	1	1	1	1	1	1	1	1	1	1
	2	1	1	1	1	1	1	1	1	1	1	1	1
	5	1	1	1	1	1	1	1	1	1	1	1	1
	8	1	1	1	1	1	1	1	1	0	0	0	1
	9	1	1	1	1	1	1	1	1	1	1	1	1
	10	0	0	0	0	0	0	1	0	0	0	0	1
	4	0	0	0	0	0	0	0	0	1	1	1	0
	7	0	0	0	0	0	0	0	0	1	1	1	0
	3	0	0	0	1	1	1	1	1	0	0	0	1
	6	1	1	1	0	0	0	0	0	0	0	0	0

b. Fase 2

Pada fase ini bertujuan untuk menempatkan *part* pada posisi yang tepat. Kriteria aspirasi pada fase 2 ini berbeda dengan fase 1, dimana kriteria aspirasi berupa nilai minimum total penjumlahan baris dan kolom dari *intermediate matrix* yang

telah ada. Penjumlahan baris mengidentifikasikan perbedaan pada mesin dan penjumlahan kolom mengidentifikasikan perbedaan pada *part*. Input yang dibutuhkan adalah *intermediate matrix* yaitu *part-machine matrix* dengan susunan mesin yang optimal.

Sebagai contoh terlebih dahulu akan dihitung nilai kriteria aspirasi untuk solusi awal melalui urutan *part* yang diperoleh dari data perusahaan, yaitu sebagai berikut :

Solusi awal :

1 2 3 4 5 6 7 8 9 10 11 12

- Penjumlahan baris

$$\sum baris = \sum_{m=1}^{b-1} \sum_{p=1}^{k} \left| a_{pm} - a_{p\,m+1} \right|$$

Misal untuk antara dua mesin 1 dan 2, didapat :

M2	1	1	1	1	1	1	1	1	1	1	1	1
M1	1	1	1	1	1	1	1	1	1	1	1	1
	0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

$$\sum = \mathbf{0}$$

Dengan cara yang sama digunakan untuk mesin yang lain.

- Penjumlahan kolom

$$\sum \text{ kolom} = \min \sum_{i=1}^{m-1} \sum_{j=1}^{n-1} \left| Pn_j m_i - P(n+1)_j m_i \right|$$

Part	1	2	
	1	1	0
	1	1	0
	1	1	0
	0	0	0
	0	0	0
	1	1	0
	1	1	0
	1	1	0
	0	0	0
	$\sum$		= 0

Dengan cara yang sama digunakan untuk part yang lain. Sehingga setelah semua perhitungan penjumlahan baris dan kolom dari *intermediate matrix* didapat nilai penjumlahan baris dan kolom sebesar 78. Nilai tersebut digunakan sebagai nilai untuk iterasi pertama.

Algoritma Tabu Search digunakan untuk mencari solusi terbaik dari tugas akhir ini. Jumlah iterasi yang digunakan yaitu 100 iterasi, *move* yang digunakan yaitu *neighborhood search.*

Setelah memasukan data yang diperlukan pada pengolahan *Algoritma Tabu Search* yaitu urutan mesin yang optimal dan solusi awal part, maka diperoleh hasil dari pengolahan penjumlahan baris dan kolom terkecil.

Contoh iterasi pada fase 2 setelah melalui *Algoritma Tabu Seacrh* didapatkan hasil seperti dibawah ini untuk iterasi pertama dan kedua :

```
------------------------------------------------------------
Pencarian Urutan Part:
------------------------------------------------------------
---------------
Iterasi ke: 1
Global Optimum:
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12: 77
Local Optimum:
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12: 77
```

```
Swap[1,2] = 2, 1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12: 77
Swap[1,3] = 3, 2, 1, 4, 5, 6, 7, 8, 9, 10, 11, 12: 77
Swap[1,4] = 4, 2, 3, 1, 5, 6, 7, 8, 9, 10, 11, 12: 79
Swap[1,5] = 5, 2, 3, 4, 1, 6, 7, 8, 9, 10, 11, 12: 83
Swap[1,6] = 6, 2, 3, 4, 5, 1, 7, 8, 9, 10, 11, 12: 83
Swap[1,7] = 7, 2, 3, 4, 5, 6, 1, 8, 9, 10, 11, 12: 82
Swap[1,8] = 8, 2, 3, 4, 5, 6, 7, 1, 9, 10, 11, 12: 81
Swap[1,9] = 9, 2, 3, 4, 5, 6, 7, 8, 1, 10, 11, 12: 83
Swap[1,10] = 10, 2, 3, 4, 5, 6, 7, 8, 9, 1, 11, 12: 89
Swap[1,11] = 11, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1, 12: 83
Swap[1,12] = 12, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 1: 79
Swap[2,3] = 1, 3, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12: 77
Swap[2,4] = 1, 4, 3, 2, 5, 6, 7, 8, 9, 10, 11, 12: 81
Swap[2,5] = 1, 5, 3, 4, 2, 6, 7, 8, 9, 10, 11, 12: 85
Swap[2,6] = 1, 6, 3, 4, 5, 2, 7, 8, 9, 10, 11, 12: 85
Swap[2,7] = 1, 7, 3, 4, 5, 6, 2, 8, 9, 10, 11, 12: 85
Swap[2,8] = 1, 8, 3, 4, 5, 6, 7, 2, 9, 10, 11, 12: 83
Swap[2,9] = 1, 9, 3, 4, 5, 6, 7, 8, 2, 10, 11, 12: 87
Swap[2,10] = 1, 10, 3, 4, 5, 6, 7, 8, 9, 2, 11, 12: 93
Swap[2,11] = 1, 11, 3, 4, 5, 6, 7, 8, 9, 10, 2, 12: 87
Swap[2,12] = 1, 12, 3, 4, 5, 6, 7, 8, 9, 10, 11, 2: 82
Swap[3,4] = 1, 2, 4, 3, 5, 6, 7, 8, 9, 10, 11, 12: 81
Swap[3,5] = 1, 2, 5, 4, 3, 6, 7, 8, 9, 10, 11, 12: 81
Swap[3,6] = 1, 2, 6, 4, 5, 3, 7, 8, 9, 10, 11, 12: 81
Swap[3,7] = 1, 2, 7, 4, 5, 6, 3, 8, 9, 10, 11, 12: 81
Swap[3,8] = 1, 2, 8, 4, 5, 6, 7, 3, 9, 10, 11, 12: 79
Swap[3,9] = 1, 2, 9, 4, 5, 6, 7, 8, 3, 10, 11, 12: 85
Swap[3,10] = 1, 2, 10, 4, 5, 6, 7, 8, 9, 3, 11, 12: 91
Swap[3,11] = 1, 2, 11, 4, 5, 6, 7, 8, 9, 10, 3, 12: 85
Swap[3,12] = 1, 2, 12, 4, 5, 6, 7, 8, 9, 10, 11, 3: 78
Swap[4,5] = 1, 2, 3, 5, 4, 6, 7, 8, 9, 10, 11, 12: 77
Swap[4,6] = 1, 2, 3, 6, 5, 4, 7, 8, 9, 10, 11, 12: 77
Swap[4,7] = 1, 2, 3, 7, 5, 6, 4, 8, 9, 10, 11, 12: 77
Swap[4,8] = 1, 2, 3, 8, 5, 6, 7, 4, 9, 10, 11, 12: 77
Swap[4,9] = 1, 2, 3, 9, 5, 6, 7, 8, 4, 10, 11, 12: 83
Swap[4,10] = 1, 2, 3, 10, 5, 6, 7, 8, 9, 4, 11, 12: 91
Swap[4,11] = 1, 2, 3, 11, 5, 6, 7, 8, 9, 10, 4, 12: 83
Swap[4,12] = 1, 2, 3, 12, 5, 6, 7, 8, 9, 10, 11, 4: 78
Swap[5,6] = 1, 2, 3, 4, 6, 5, 7, 8, 9, 10, 11, 12: 77
Swap[5,7] = 1, 2, 3, 4, 7, 6, 5, 8, 9, 10, 11, 12: 77
Swap[5,8] = 1, 2, 3, 4, 8, 6, 7, 5, 9, 10, 11, 12: 77
Swap[5,9] = 1, 2, 3, 4, 9, 6, 7, 8, 5, 10, 11, 12: 85
Swap[5,10] = 1, 2, 3, 4, 10, 6, 7, 8, 9, 5, 11, 12: 93
Swap[5,11] = 1, 2, 3, 4, 11, 6, 7, 8, 9, 10, 5, 12: 85
Swap[5,12] = 1, 2, 3, 4, 12, 6, 7, 8, 9, 10, 11, 5: 78
Swap[6,7] = 1, 2, 3, 4, 5, 7, 6, 8, 9, 10, 11, 12: 77
Swap[6,8] = 1, 2, 3, 4, 5, 8, 7, 6, 9, 10, 11, 12: 77
Swap[6,9] = 1, 2, 3, 4, 5, 9, 7, 8, 6, 10, 11, 12: 85
Swap[6,10] = 1, 2, 3, 4, 5, 10, 7, 8, 9, 6, 11, 12: 93
```

```
Swap[6,11] = 1, 2, 3, 4, 5, 11, 7, 8, 9, 10, 6, 12: 85
Swap[6,12] = 1, 2, 3, 4, 5, 12, 7, 8, 9, 10, 11, 6: 76
Swap[7,8] = 1, 2, 3, 4, 5, 6, 8, 7, 9, 10, 11, 12: 77
Swap[7,9] = 1, 2, 3, 4, 5, 6, 9, 8, 7, 10, 11, 12: 85
Swap[7,10] = 1, 2, 3, 4, 5, 6, 10, 8, 9, 7, 11, 12: 93
Swap[7,11] = 1, 2, 3, 4, 5, 6, 11, 8, 9, 10, 7, 12: 83
Swap[7,12] = 1, 2, 3, 4, 5, 6, 12, 8, 9, 10, 11, 7: 77
Swap[8,9] = 1, 2, 3, 4, 5, 6, 7, 9, 8, 10, 11, 12: 85
Swap[8,10] = 1, 2, 3, 4, 5, 6, 7, 10, 9, 8, 11, 12: 85
Swap[8,11] = 1, 2, 3, 4, 5, 6, 7, 11, 9, 10, 8, 12: 77
Swap[8,12] = 1, 2, 3, 4, 5, 6, 7, 12, 9, 10, 11, 8: 76
Swap[9,10] = 1, 2, 3, 4, 5, 6, 7, 8, 10, 9, 11, 12: 77
Swap[9,11] = 1, 2, 3, 4, 5, 6, 7, 8, 11, 10, 9, 12: 77
Swap[9,12] = 1, 2, 3, 4, 5, 6, 7, 8, 12, 10, 11, 9: 74
Swap[10,11] = 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 10, 12: 77
Swap[10,12] = 1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 11, 10: 82
Swap[11,12] = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 11: 82
-------------------------------------------------------
---------------
Iterasi ke: 2
Global Optimum:
1, 2, 3, 4, 5, 6, 7, 8, 12, 10, 11, 9: 74
Local Optimum:
1, 2, 3, 4, 5, 6, 7, 8, 12, 10, 11, 9: 74
-------------------------------------------------------
---------------
Swap[1,2] = 2, 1, 3, 4, 5, 6, 7, 8, 12, 10, 11, 9: 74
Swap[1,3] = 3, 2, 1, 4, 5, 6, 7, 8, 12, 10, 11, 9: 74
Swap[1,4] = 4, 2, 3, 1, 5, 6, 7, 8, 12, 10, 11, 9: 76
Swap[1,5] = 5, 2, 3, 4, 1, 6, 7, 8, 12, 10, 11, 9: 80
Swap[1,6] = 6, 2, 3, 4, 5, 1, 7, 8, 12, 10, 11, 9: 80
Swap[1,7] = 7, 2, 3, 4, 5, 6, 1, 8, 12, 10, 11, 9: 79
Swap[1,8] = 8, 2, 3, 4, 5, 6, 7, 1, 12, 10, 11, 9: 80
Swap[1,9] = 12, 2, 3, 4, 5, 6, 7, 8, 1, 10, 11, 9: 77
Swap[1,10] = 10, 2, 3, 4, 5, 6, 7, 8, 12, 1, 11, 9: 80
Swap[1,11] = 11, 2, 3, 4, 5, 6, 7, 8, 12, 10, 1, 9: 86
Swap[1,12] = 9, 2, 3, 4, 5, 6, 7, 8, 12, 10, 11, 1: 82
Swap[2,3] = 1, 3, 2, 4, 5, 6, 7, 8, 12, 10, 11, 9: 74
Swap[2,4] = 1, 4, 3, 2, 5, 6, 7, 8, 12, 10, 11, 9: 78
Swap[2,5] = 1, 5, 3, 4, 2, 6, 7, 8, 12, 10, 11, 9: 82
Swap[2,6] = 1, 6, 3, 4, 5, 2, 7, 8, 12, 10, 11, 9: 82
Swap[2,7] = 1, 7, 3, 4, 5, 6, 2, 8, 12, 10, 11, 9: 82
Swap[2,8] = 1, 8, 3, 4, 5, 6, 7, 2, 12, 10, 11, 9: 82
Swap[2,9] = 1, 12, 3, 4, 5, 6, 7, 8, 2, 10, 11, 9: 80
Swap[2,10] = 1, 10, 3, 4, 5, 6, 7, 8, 12, 2, 11, 9: 84
Swap[2,11] = 1, 11, 3, 4, 5, 6, 7, 8, 12, 10, 2, 9: 90
Swap[2,12] = 1, 9, 3, 4, 5, 6, 7, 8, 12, 10, 11, 2: 86
Swap[3,4] = 1, 2, 4, 3, 5, 6, 7, 8, 12, 10, 11, 9: 78
Swap[3,5] = 1, 2, 5, 4, 3, 6, 7, 8, 12, 10, 11, 9: 78
Swap[3,6] = 1, 2, 6, 4, 5, 3, 7, 8, 12, 10, 11, 9: 78
Swap[3,7] = 1, 2, 7, 4, 5, 6, 3, 8, 12, 10, 11, 9: 78
```

```
Swap[3,8] = 1, 2, 8, 4, 5, 6, 7, 3, 12, 10, 11, 9: 78
Swap[3,9] = 1, 2, 12, 4, 5, 6, 7, 8, 3, 10, 11, 9: 76
Swap[3,10] = 1, 2, 10, 4, 5, 6, 7, 8, 12, 3, 11, 9: 82
Swap[3,11] = 1, 2, 11, 4, 5, 6, 7, 8, 12, 10, 3, 9: 88
Swap[3,12] = 1, 2, 9, 4, 5, 6, 7, 8, 12, 10, 11, 3: 84
Swap[4,5] = 1, 2, 3, 5, 4, 6, 7, 8, 12, 10, 11, 9: 74
Swap[4,6] = 1, 2, 3, 6, 5, 4, 7, 8, 12, 10, 11, 9: 74
Swap[4,7] = 1, 2, 3, 7, 5, 6, 4, 8, 12, 10, 11, 9: 74
Swap[4,8] = 1, 2, 3, 8, 5, 6, 7, 4, 12, 10, 11, 9: 74
Swap[4,9] = 1, 2, 3, 12, 5, 6, 7, 8, 4, 10, 11, 9: 74
Swap[4,10] = 1, 2, 3, 10, 5, 6, 7, 8, 12, 4, 11, 9: 80
Swap[4,11] = 1, 2, 3, 11, 5, 6, 7, 8, 12, 10, 4, 9: 88
Swap[4,12] = 1, 2, 3, 9, 5, 6, 7, 8, 12, 10, 11, 4: 84
Swap[5,6] = 1, 2, 3, 4, 6, 5, 7, 8, 12, 10, 11, 9: 74
Swap[5,7] = 1, 2, 3, 4, 7, 6, 5, 8, 12, 10, 11, 9: 74
Swap[5,8] = 1, 2, 3, 4, 8, 6, 7, 5, 12, 10, 11, 9: 74
Swap[5,9] = 1, 2, 3, 4, 12, 6, 7, 8, 5, 10, 11, 9: 74
Swap[5,10] = 1, 2, 3, 4, 10, 6, 7, 8, 12, 5, 11, 9: 82
Swap[5,11] = 1, 2, 3, 4, 11, 6, 7, 8, 12, 10, 5, 9: 90
Swap[5,12] = 1, 2, 3, 4, 9, 6, 7, 8, 12, 10, 11, 5: 86
Swap[6,7] = 1, 2, 3, 4, 5, 7, 6, 8, 12, 10, 11, 9: 74
Swap[6,8] = 1, 2, 3, 4, 5, 8, 7, 6, 12, 10, 11, 9: 74
Swap[6,9] = 1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Swap[6,10] = 1, 2, 3, 4, 5, 10, 7, 8, 12, 6, 11, 9: 82
Swap[6,11] = 1, 2, 3, 4, 5, 11, 7, 8, 12, 10, 6, 9: 90
Swap[6,12] = 1, 2, 3, 4, 5, 9, 7, 8, 12, 10, 11, 6: 86
Swap[7,8] = 1, 2, 3, 4, 5, 6, 8, 7, 12, 10, 11, 9: 72
Swap[7,9] = 1, 2, 3, 4, 5, 6, 12, 8, 7, 10, 11, 9: 74
Swap[7,10] = 1, 2, 3, 4, 5, 6, 10, 8, 12, 7, 11, 9: 80
Swap[7,11] = 1, 2, 3, 4, 5, 6, 11, 8, 12, 10, 7, 9: 90
Swap[7,12] = 1, 2, 3, 4, 5, 6, 9, 8, 12, 10, 11, 7: 85
Swap[8,9] = 1, 2, 3, 4, 5, 6, 7, 12, 8, 10, 11, 9: 72
Swap[8,10] = 1, 2, 3, 4, 5, 6, 7, 10, 12, 8, 11, 9: 82
Swap[8,11] = 1, 2, 3, 4, 5, 6, 7, 11, 12, 10, 8, 9: 90
Swap[8,12] = 1, 2, 3, 4, 5, 6, 7, 9, 12, 10, 11, 8: 86
Swap[9,10] = 1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 11, 9: 82
Swap[9,11] = 1, 2, 3, 4, 5, 6, 7, 8, 11, 10, 12, 9: 82
Swap[9,12] = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12: 77 -->
Sudah di tabulist
Swap[10,11] = 1, 2, 3, 4, 5, 6, 7, 8, 12, 11, 10, 9: 74
Swap[10,12] = 1, 2, 3, 4, 5, 6, 7, 8, 12, 9, 11, 10: 74
Swap[11,12] = 1, 2, 3, 4, 5, 6, 7, 8, 12, 10, 9, 11: 74
```

Dari pengolahan data *Algoritma Tabu Seacrh* diatas didapat urutan terbaik dari part adalah :

1 2 3 4 5 12 7 8 6 10 11 9

4.2.5 Pengolahan Matrik Akhir

Pengolahan matrik akhir adalah untuk mengetahui sjauh mana peningkatan yang terjadi setelah didapat urutan mesin dan komponen yang optimal. Berikut adalah matrik akhir dimana urutan mesin dan komponen telah optimal

4.2.5.1 Pengolahan Matrik Akhir Algoritma Semut

Tabel 4.15 Matrik Akhir Algoritma Semut

		KOMPONEN												
		11	2	3	6	8	12	1	5	9	10	4	7	
MESIN	4	1	0	0	0	0	0	0	0	1	1	0	0	
	9	1	1	1	1	1	1	1	1	1	1	1	1	
	5	1	1	1	1	1	1	1	1	1	1	1	1	
	7	1	0	0	0	0	0	0	0	1	1	0	0	
	3	0	0	0	1	1	1	0	1	0	0	1	1	
	6	0	1	1	0	0	0	1	0	0	0	0	0	
	2	1	1	1	1	1	1	1	1	1	1	1	1	
	8	0	1	1	1	1	1	1	1	0	0	1	1	
	10	0	0	0	0	0	1	0	0	0	0	0	1	
	1	1	1	1	1	1	1	1	1	1	1	1	1	

Pengukuran Performansi

M = 10 ; P = 12 ; o = 74 ; e = 25 ; v = 29 ; ω = 0.5

a. Effisiensi Pengelompokkan

$$\eta_1 = \frac{o-e}{o-e+v} = \frac{74-25}{74-25+29} = 0.628$$

$$\eta_2 = \frac{MP-o-v}{MP-o-v+e} = \frac{(10x12)-74-29}{(10x12)-74-29+25} = 0.405$$

$$\eta = \omega\eta_1 + (1-\omega)\eta_2 = (0.5\text{x}0.628) + (1\text{-}0.5)\text{x}0.405 = 0.516$$

b. Kekuatan Pengelompokkan

$$\tau = \frac{o-e}{o+v} = \frac{74-25}{74+29} = 0.476$$

4.2.5.2 Pengolahan Matrik Akhir Algoritma *Tabu Search*

Tabel 4.16 Matrik Akhir *Tabu Search*

		KOMPONEN											
MESIN		1	2	3	4	5	12	7	8	6	10	11	9
	1	1	1	1	1	1	1	1	1	1	1	1	1
	2	1	1	1	1	1	1	1	1	1	1	1	1
	5	1	1	1	1	1	1	1	1	1	1	1	1
	8	1	1	1	1	1	1	1	1	1	0	0	0
	9	1	1	1	1	1	1	1	1	1	1	1	1
	10	0	0	0	0	0	1	1	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	1	1	1
	7	0	0	0	0	0	0	0	0	0	1	1	1
	3	0	0	0	1	1	1	1	1	1	0	0	0
	6	1	1	1	0	0	0	0	0	0	0	0	0

Pengukuran Performansi

M = 10 ; P = 12 ; o = 74 ; e = 23 ; v = 9 ; ω = 0.5

a. Effisiensi Pengelompokkan

$$\eta_1 = \frac{o-e}{o-e+v} = \frac{74-23}{74-23+9} = 0.85$$

$$\eta_2 = \frac{MP-o-v}{MP-o-v+e} = \frac{(10x12)-74-9}{(10x12)-74-9+23} = 0.62$$

$$\eta = \omega\eta_1 + (1-\omega)\eta_2 = (0.5x0.58) + (1\text{-}0.5)x0.67 = 0.7333$$

b. Kekuatan Pengelompokkan

$$\tau = \frac{o-e}{o+v} = \frac{74-23}{74+9} = 0.6145$$

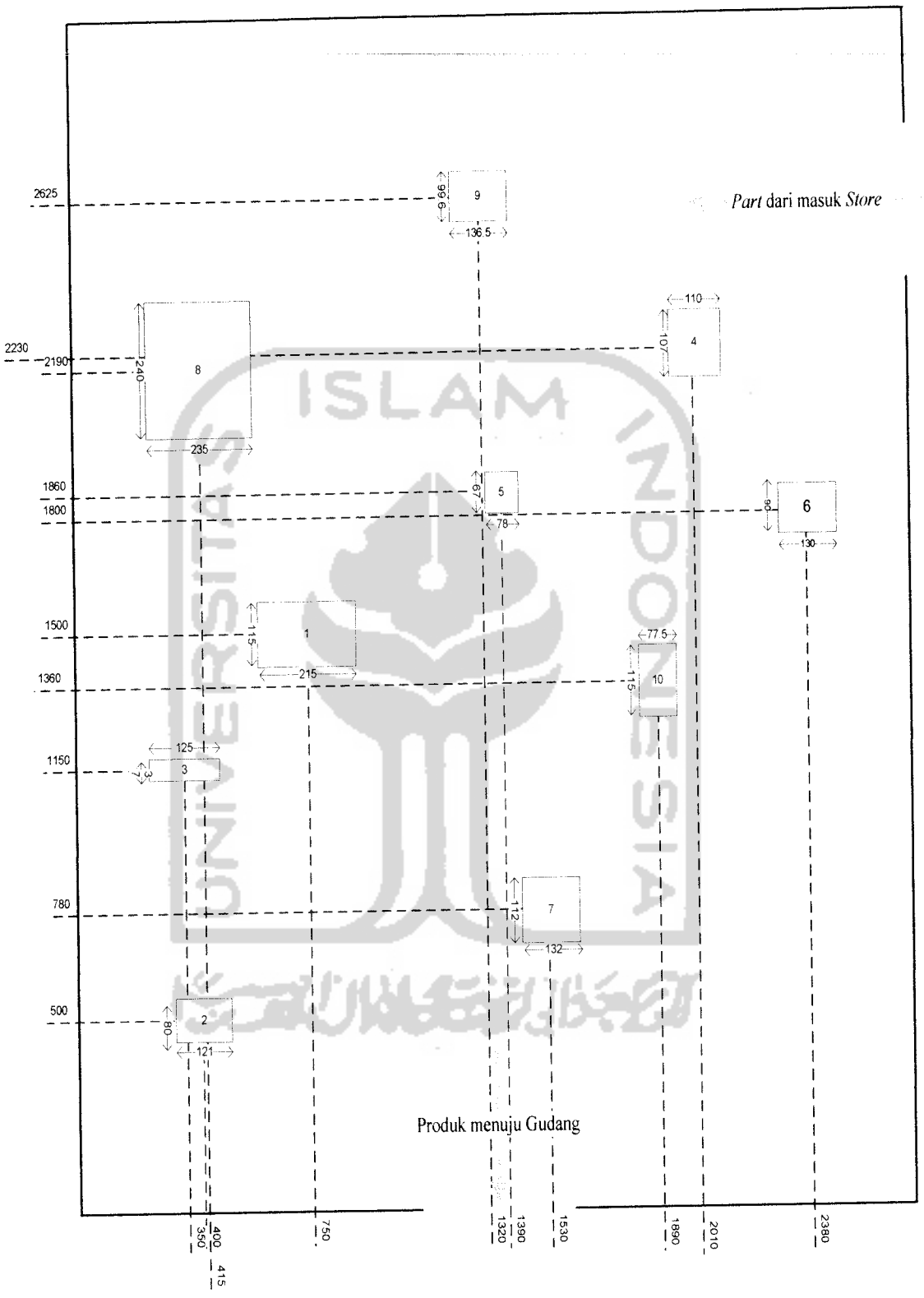
4.2.6 Pembuatan *Layout* Usulan

4.2.6.1 Pembuatan *Layout* Usulan dengan Algoritma Semut

Dari hasil pembentukan sel manufaktur, didapatkan data penggantian posisi mesin:

1. Posisi mesin 1 diganti mesin 4.
2. Posisi mesin 2 diganti mesin 9.
3. Posisi mesin 3 diganti mesin 5.
4. Posisi mesin 4 diganti mesin 7..
5. Posisi mesin 5 diganti mesin 3.
6. Posisi mesin 6 tetap pada mesin 6
7. Posisi mesin 7 diganti mesin 2.
8. Posisi mesin 8 tatap pada mesin 8
9. Posisi mesin 9 diganti mesin 10.
10. Posisi mesin 10 diganti mesin 1.

Berdasarkan data penggantian posisi mesin diatas, maka dibentuklah sebuah *layout* usulan pabrik sbb:



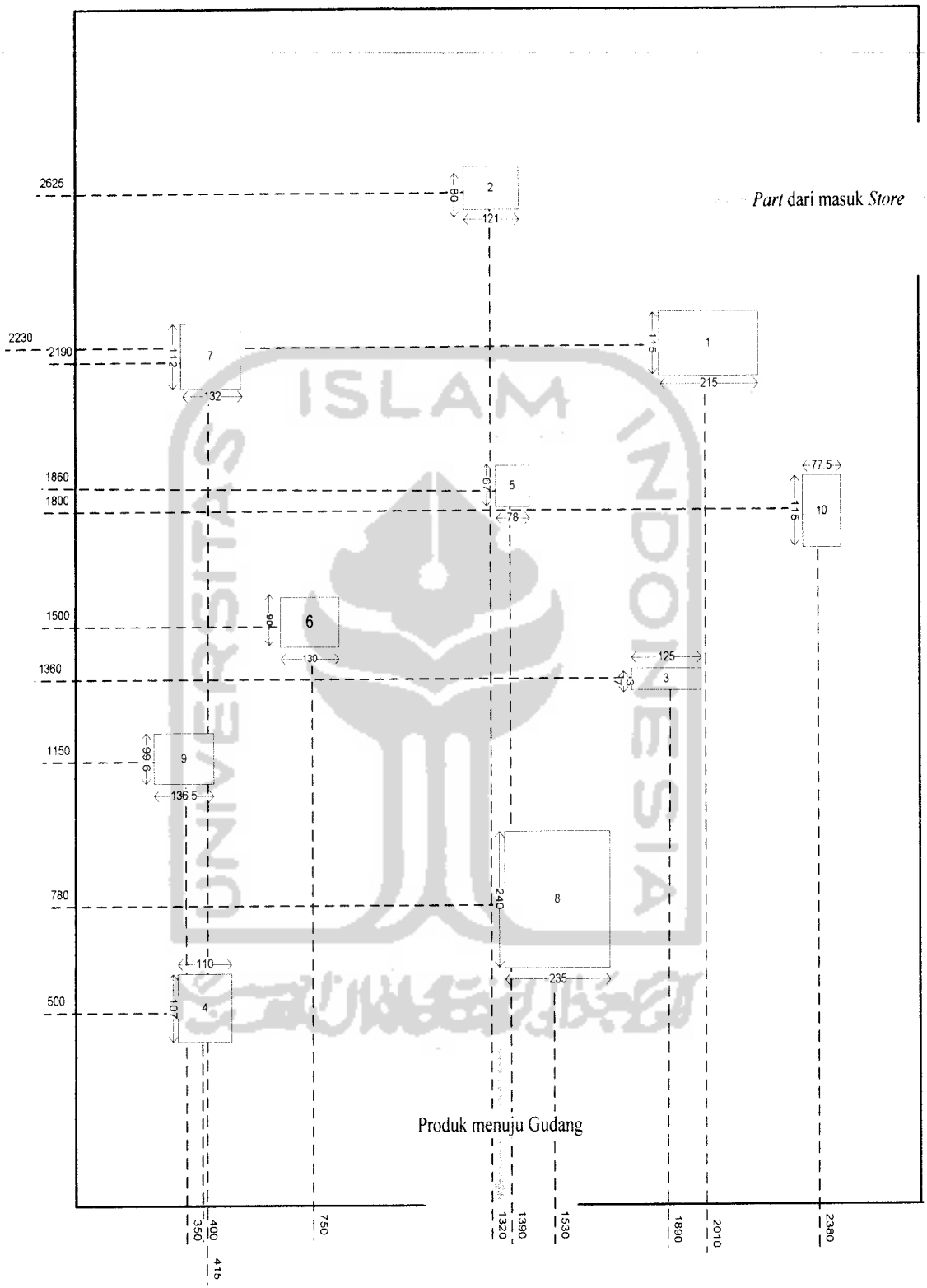
Gambar 4.6 *Layout* Usulan Pabrik dengan Algoritma Semut

4.2.6.2 Pembuatan *Layout* Usulan dengan *Tabu Search*

Dari hasil pembentukan sel manufaktur, didapatkan data penggantian posisi mesin:

1. Posisi mesin 1 tetap pada mesin 1.
2. Posisi mesin 2 tetap pada mesin 2.
3. Posisi mesin 3 diganti mesin 5.
4. Posisi mesin 4 diganti mesin 8..
5. Posisi mesin 5 diganti mesin 9.
6. Posisi mesin 6 diganti mesin 10
7. Posisi mesin 7 diganti mesin 4.
8. Posisi mesin 8 diganti mesin 7
9. Posisi mesin 9 diganti mesin 3.
10. Posisi mesin 10 diganti mesin 6.

Berdasarkan data penggantian posisi mesin diatas, maka dibentuklah sebuah *layout* usulan pabrik sbb:



Gambar 4.7 *Layout* Usulan Pabrik dengan Algoritma *Tabu Search*

4.2.7 Perhitungan Jarak Antar Mesin Berdasarkan Koordinat Mesin

Untuk menghitung jarak antar mesin digunakan suatu formulasi yaitu jarak *Euclidean* sesuai dengan persamaan ....8) pada bab II . Adapun contoh perhitungan jarak antar mesin untuk mesin 1 dan mesin 2 adalah sebagai berikut :

$$d(1,2) = \sqrt{(2010-1320)^2 + (2230-2625)^2}$$

$$= 795.06 \approx 7.95\text{m}$$

Hasil perhitungan untuk jarak antar mesin yang lain dapat dilihat pada tabel dibawah ini :

Tabel 4.17 Matrik Awal Jarak Antar Mesin

	1	2	3	4	5	6	7	8	9	10
1	-	7.95	7.22	15.27	19.80	5.67	23.63	15.96	8.78	14.56
2		-	7.68	18.57	17.65	13.43	23.16	10.04	13.87	12.61
3			-	10.89	12.59	9.92	16.82	10.29	7.07	7.34
4				-	12.37	13.28	11.64	17.98	6.83	10.62
5					-	21.32	6.52	10.42	15.54	5.32
6						-	23.69	20.03	6.59	16.57
7							-	16.90	17.20	10.59
8								-	16.92	7.67
9									-	11.49
10										-

4.2.8 Perhitungan Jarak *Material Handling* pada *Layout* Awal

Total jarak *material handling* tiap *part* bisa diketahui dari jarak *material handling* dan frekwensi perpindahannya. Jarak *material handling* yang harus ditempuh oleh suatu komponen dalam proses produksinya dapat dihitung dengan memperhatikan jarak antar mesin dan urutan proses produksinya. Pada tabel 4.14 dapat dilihat perhitungan jarak *material handling.*

Perhitungan jarak *material handling* dilakukan dengan menjumlahkan jarak antar mesin yang dilalui oleh komponen yang bersangkutan. Contoh perhitungan *material handling* untuk komponen frame panjang adalah sebagai berikut :

Routing frame panjang = 2-1-6-8-5-9

Jarak antar mesin (meter) = 7.95+5.67+20.03+10.42+15.54

Total Jarak (meter) = 59.62

Hasil perhitungan untuk jarak *material handling* untuk komponen lain dapat dilihat pada table dibawah ini :

Tabel 4.18 Hasil Perhitungan Jarak *Material Handling Layout* Awal

No	Nama Komponen	Jarak (Meter)
1	*frame panjang*	59.62
2	*frame pendek*	59.62
3	*frame tengah*	59.62
4	*sunduk panjang*	51.43
5	*sunduk pendek*	51.43
6	*sunduk tengah*	51.43
7	*kloss*	66.11
8	*frame atas*	51.43
9	*ambal bawah*	56.93
10	*ambal*	56.93
11	*list ambal*	56.93
12	*kaki meja*	66.11

(Perhitungan selengkapnya terlampir)

Berdasarkan jarak antar mesin dan frekuensi aliran bahan maka dapat ditentukan jarak total yang ditempuh selama kegiatan produksi .

Total Jarak Perpindahan = Frekuensi x Jarak Perpindahan

Contoh : Untuk komponen Frame Panjang pada Produk Coffee Table

Total Jarak Perpindahan = 3 x 59,62 = 178.86 meter

Dibawah ini adalah data frekuensi, dan total jarak perpindahan untuk masing – masing komponen per-produk seperti pada tabel 4.19, 4.20, dan 4.21

Tabel 4.19 Total Jarak Produk *Coffee Table*

No	Nama Komponen	Total Kebutuhan (Unit)	Batch	Frek.	Jarak (meter)	Total Jarak (meter)
1	*frame panjang*	164	75	3	59.62	178.86
2	*frame pendek*	164	60	3	59.62	178.86
3	*frame tengah*	82	65	2	59.62	119.24
4	*sunduk panjang*	164	75	3	51.43	154.28
5	*sunduk pendek*	164	75	3	51.43	154.28
6	*sunduk tengah*	82	60	2	51.43	102.85
7	*kloss*	656	175	4	66.11	264.45
TOTAL						1152.82

Tabel 4.20 Total Jarak Produk *Gothic Table*

No	Nama Komponen	Total Kebutuhan (Unit)	Batch	Frek.	Jarak (meter)	Total Jarak (meter)
1	*frame atas*	244	75	4	51.43	205.71
2	*ambal bawah*	61	50	2	56.93	113.86
3	*ambal*	61	50	2	56.93	113.86
4	*list ambal*	244	75	4	56.93	227.71
5	*kaki meja*	244	60	5	66.11	330.56
TOTAL						991.69

Tabel 4.21 Total Jarak Produk *Square Table*

No	Nama Komponen	Total Kebutuhan (Unit)	Batch	Frek.	Jarak (meter)	Total Jarak (meter)
1	*frame atas*	252	75	4	51.43	205.71
2	*ambal*	63	50	2	56.93	113.86
3	*list ambal*	252	75	4	56.93	227.71
4	*kaki meja*	252	60	5	66.11	330.56
TOTAL						877.83

Berikut pada tabel 4.22 adalah total jarak *material handling* pada *layout* awal untuk semua produk :

Tabel 4.22 Total Jarak *Material Handling* pada *Layout* Awal

No	Produk	Jarak
1	Coffee Table	1152.82
2	End Table	991.69
3	Square Table	877.83
TOTAL		**3022.34**

4.2.9 Perhitungan Jarak *Material Handling Layout* Usulan

4.2.9.1 Perhitungan Jarak *Material Handling* Algoritma Semut

Berdasarkan hasil pengoalahan dari Algoritma Semut telah didapatkan urutan mesin terbaik. Setelah terbentuk urutan mesin terbaik, urutan mesin pada *layout* awal diganti dengan urutan mesin usulan. karena posisinya tetap dan hanya urutan mesinnya yang berubah maka matrik jarak *material handling* antara setiap posisi mesin produksi pada layout usulan sama dengan matrik awal.

Dengan demikian bisa diketahui ada atau tidaknya nilai pengurangan jarak material handling tanpa harus mengaplikasikannya terlebih dahulu. Setelah didapatkan urutan mesin dan komponen yang optimal, berikut ini adalah jarak antar mesin yang optimal untuk layout usulan :

Tabel 4.23 Matrik Jarak Antar Mesin pada Algoritma Semut (Meter)

Mesin	4	9	5	7	3	6	2	8	10	1
4	-	7.95	7.22	15.27	19.80	5.67	23.63	15.96	8.78	14.56
9		-	7.68	18.57	17.65	13.43	23.16	10.04	13.87	12.61
5			-	10.89	12.59	9.92	16.82	10.29	7.07	7.34
7				-	12.37	13.28	11.64	17.98	6.83	10.62
3					-	21.32	6.52	10.42	15.54	5.32
6						-	23.69	20.03	6.59	16.57
2							-	16.90	17.20	10.59
8								-	16.92	7.67
10									-	11.49
1										-

Sehingga akan didapat perubahan jarak yang dilalui komponen pada setiap perpindahan mesin Perhitungan jarak *material handling* dilakukan dengan menjumlahkan jarak antar mesin yang dilalui oleh komponen yang bersangkutan. Contoh perhitungan *material handling* untuk komponen frame panjang adalah sebagai berikut :

Routing frame panjang = 2-1-6-8-5-9

Jarak antar mesin (meter) = 10.59+16.67+20.03+10.29+7.68

Total Jarak (meter) = 65.18

Hasil perhitungan untuk jarak *material handling* untuk komponen lain dapat dilihat pada table dibawah ini :

Tabel 4.24 Hasil Perhitungan Jarak *Material Handling* Algoritma Semut

No	Nama Komponen	Jarak (Meter)
1	*frame panjang*	65.18
2	*frame pendek*	65.18
3	*frame tengah*	65.18
4	*sunduk panjang*	44.31
5	*sunduk pendek*	44.31
6	*sunduk tengah*	44.31
7	*kloss*	66.02
8	*frame atas*	44.31
9	*ambal bawah*	59.00
10	*ambal*	59.00
11	*list ambal*	59.00
12	*kaki meja*	66.02

(Perhitungan selengkapnya terlampir)

Dibawah ini adalah jarak *material handling* setiap komponen untuk *layout* usulan, sepeti pada tabel 4.25, 4.26, dan 4.27

Tabel 4.25 Total Jarak *Produk Coffee* Table Algoritma Semut

No	Nama Komponen	Total Kebutuhan (Unit)	Batch	Frek.	Jarak (meter)	Total Jarak (meter)
1	*frame panjang*	164	75	3	65.18	195.53
2	*frame pendek*	164	60	3	65.18	195.53
3	*frame tengah*	82	65	2	65.18	130.35
4	*sunduk panjang*	164	75	3	44.31	132.92
5	*sunduk pendek*	164	75	3	44.31	132.92
6	*sunduk tengah*	82	60	2	44.31	88.61
7	*kloss*	656	175	4	66.02	264.07
TOTAL						1139.94

Tabel 4.26 Total Jarak Produk *Gothic Table* Algoritma Semut

No	Nama Komponen	Total Kebutuhan (Unit)	Batch	Frek.	Jarak (meter)	Total Jarak (meter)
1	*frame atas*	244	75	4	44.31	177.22
2	*ambal bawah*	61	50	2	59.00	118.01
3	*ambal*	61	50	2	59.00	118.01
4	*list ambal*	244	75	4	59.00	236.01
5	*kaki meja*	244	60	5	66.02	330.09
TOTAL						979.34

Tabel 4.27 Total Jarak Produk *Square Table* Algoritma Semut

No	Nama Komponen	Total Kebutuhan (Unit)	Batch	Frek.	Jarak (meter)	Total Jarak (meter)
1	*frame atas*	252	75	4	44.31	177.22
2	*ambal*	63	50	2	59.00	118.01
3	*list ambal*	252	75	4	59.00	236.01
4	*kaki meja*	252	60	5	66.02	330.09
TOTAL						861.33

Selisih total jarak antara *layout* awal dan *layout* usulan Algoritma Semut seperti pada tabel 4.28

Tabel 4.28 Selisih Total Jarak *Layout* Awal dengan *Layout* Algoritma Semut

Produk	Jarak (Meter)		
	Awal	Usulan	Selisih
Coffee Table	1152.82	1139.94	-12.88
End Table	991.69	979.34	-12.35
Square Table	877.83	861.33	-16.50
	3022.34	**2980.60**	**-41.74**

4.2.9.2 Perhitungan Jarak *Material Handling Tabu Search*

Berdasarkan hasil pengolahan dari *Tabu Search* telah didapatkan urutan mesin terbaik. Setelah terbentuk urutan mesin terbaik, urutan mesin pada *layout* awal diganti dengan urutan mesin usulan. karena posisinya tetap dan hanya urutan mesinnya yang berubah maka matrik jarak *material handling* antara setiap posisi mesin produksi pada layout usulan sama dengan matrik awal.

Dengan demikian bisa diketahui ada atau tidaknya nilai pengurangan jarak material handling tanpa harus mengaplikasikannya terlebih dahulu. Setelah didapatkan urutan mesin dan komponen yang optimal, berikut ini adalah jarak antar mesin yang optimal untuk layout usulan :

Tabel 4.29 Matrik Jarak Antar Mesin pada *Tabu Search* (Meter)

Mesin	1	2	5	8	9	10	4	7	3	6
1	-	7.95	7.22	15.27	19.80	5.67	23.63	15.96	8.78	14.56
2		-	7.68	18.57	17.65	13.43	23.16	10.04	13.87	12.61
5			-	10.89	12.59	9.92	16.82	10.29	7.07	7.34
8				-	12.37	13.28	11.64	17.98	6.83	10.62
9					-	21.32	6.52	10.42	15.54	5.32
10						-	23.69	20.03	6.59	16.57
4							-	16.90	17.20	10.59
7								-	16.92	7.67
3									-	11.49
6										-

Sehingga akan didapat perubahan jarak yang dilalui komponen pada setiap perpindahan mesin. Contoh perhitungan *material handling* untuk komponen frame panjang adalah sebagai berikut :

Routing frame panjang = 2-1-6-8-5-9

Jarak antar mesin (meter) = 7.95+14.56+10.62+10.89+12.59

Total Jarak (meter) = 56.61

Berikut adalah tabel perhitungan jarak material handling layout usulan :

Tabel 4.30 Hasil Perhitungan Jarak *Material Handling Tabu Search*

No	Nama Komponen	Jarak (Meter)
1	*frame panjang*	56.61
2	*frame pendek*	56.61
3	*frame tengah*	56.61
4	*sunduk panjang*	47.04
5	*sunduk pendek*	47.04
6	*sunduk tengah*	47.04
7	*kloss*	50.52
8	*frame atas*	47.04
9	*ambal bawah*	71.37
10	*ambal*	71.37
11	*list ambal*	71.37
12	*kaki meja*	50.52

Dibawah ini adalah jarak *material handling* setiap komponen untuk *layout* usulan, sepeti pada tabel 4.31, 4.32, dan 4.33

Tabel 4.31 Total Jarak Produk *Coffee Table* Algoritma *Tabu Search*

No	Nama Komponen	Total Kebutuhan (Unit)	Batch	Frek.	Jarak (meter)	Total Jarak (meter)
1	*frame panjang*	164	75	3	56.61	169.83
2	*frame pendek*	164	60	3	56.61	169.83
3	*frame tengah*	82	65	2	56.61	113.22
4	*sunduk panjang*	164	75	3	47.04	141.13
5	*sunduk pendek*	164	75	3	47.04	141.13
6	*sunduk tengah*	82	60	2	47.04	94.08
7	*kloss*	656	175	4	50.52	202.07
TOTAL						1031.29

Tabel 4.32 Total Jarak Produk *Gothic Table* Algoritma *Tabu Search*

No	Nama Komponen	Total Kebutuhan (Unit)	Batch	Frek.	Jarak (meter)	Total Jarak (meter)
1	*frame atas*	244	75	4	47.04	188.17
2	*ambal bawah*	61	50	2	71.37	142.74
3	*ambal*	61	50	2	71.37	142.74
4	*list ambal*	244	75	4	71.37	285.48
5	*kaki meja*	244	60	5	50.52	252.59
TOTAL						1011.72

Tabel 4.33 Total Jarak Produk *Square Table* Algoritma *Tabu Search*

No	Nama Komponen	Total Kebutuhan (Unit)	Batch	Frek.	Jarak (meter)	Total Jarak (meter)
1	*frame atas*	252	75	4	47.04	188.17
2	*ambal*	63	50	2	71.37	142.74
3	*list ambal*	252	75	4	71.37	285.48
4	*kaki meja*	252	60	5	50.52	252.59
TOTAL						868.98

Selisih total jarak antara *layout* awal dan *layout* usulan *Tabu Search* seperti pada tabel 4.34

Tabel 4.34 Selisih Total Jarak *Layout* Awal dengan *Layout Tabu Search*

Produk	Jarak (Meter)		
	Awal	Usulan	Selisih
Coffee Table	1152.82	1031.29	-121.52
End Table	991.69	1011.72	20.03
Square Table	877.83	868.98	-8.86
	3022.34	**2911.99**	**-110.35**

4.2.10 Penentuan Biaya Pemindahan

Biaya pemindahan diperoleh berdasarkan besarnya jarak yang digunakan untuk memindahkan dari mesin satu ke mesin yang lainnya. Selain itu, biaya pemindahan juga ditentukan berdasarkan frekuensi pengangkutan pada tiap – tiap mesin. Semakin besar frekuensinya maka semakin besar pula biaya yang dikeluarkan.

Komponen-komponen yang terlibat dalam pemindahan *part* adalah sebagai berikut :

1. Jumlah tenaga kerja 4 orang.
2. Upah tenaga kerja Rp. 12.250,00 / orang / hari
3. Alat angkut berjumlah 7 buah, seharga Rp.350.00,00 / unit
4. Umur ekonomis 5 tahun
5. Hari kerja 25 hari / bulan

Besarnya biaya pemindahan *part* per-meter adalah sebagai berikut :

1. Penentuan biaya penyusutan dengan menggunakan garis lurus

   Biaya penyusutan per-bulan = $\frac{7x350.000}{5x12}$ = Rp 40.833,33

2. Biaya Operator = 4 x Rp.12.250

   = Rp. 49.000 / hari

   = Rp. 1.225.000 / bulan

3. Total biaya operasional per-bulan

   Biaya operasional = Biaya penyusutan + biaya operator

   = Rp. 40.833,33 + 1.225.000

   = Rp. 1.265.833,33

4. Biaya pemindahan per-meter

Biaya pemindahan bahan = Biaya total / jarak

= Rp 1.265.833,33 / 3022,34 meter

= Rp. 418,43 / meter

4.2.11 Perbandingan Biaya Sesudah dan Sebelum *Re-Layout*

4.2.11.1 Perbandingan Biaya Sebelum dan Sesudah *Re-Layout* dengan Algoritma Semut

Biaya pemindahan per-meter = Total biaya operasional variabel / jarak

= Rp. 1.225.000 / 3022,34 meter

= Rp. 405,32 / meter

1. Untuk biaya pemindahan *part* sebelum re-*layout* adalah :

   Biaya pemindahan = Rp. 405,32 / meter x 3022,34meter

   = Rp 1.225.000,00

2. Untuk biaya pemindahan *part* sesudah re-layout adalah :

   Biaya pemindahan = Rp 405,32 / meter x 2980,60 meter

   = Rp. 1.208.096,79

3. Selisih biaya pemindahan sebelum dan sesudah *re-layout* :

   Selisih biaya = Rp 1.225.000,00 – Rp. 1.208.096,79

   = Rp. 16.903,21

4.2.11.2 Perbandingan Biaya Sebelum dan Sesudah *Re-Layout* dengan *Tabu Search*

Biaya pemindahan per-meter = Total biaya operasional variabel / jarak

= Rp. 1.225.000 / 3022,34 meter

= Rp. 405,32 / meter

1. Untuk biaya pemindahan *part* sebelum re-*layout* adalah :

   Biaya pemindahan = Rp. 405,32 / meter x 3022,34meter

   = Rp 1.225.000,00

2. Untuk biaya pemindahan *part* sesudah re-layout adalah :

   Biaya pemindahan = Rp 405,32 / meter x 2911,99 meter

   = Rp. 1.180.287,79

3. Selisih biaya pemindahan sebelum dan sesudah *re-layout* :

   Selisih biaya = Rp 1.225.000,00 – Rp. 1.180.287,79

   = Rp. 74.712,21

4.2.12 Perbandingan Aplikasi Algoritma Semut dan Algoritma *Tabu Search* Dalam Perancangan Ulang *Layout*

1. Perbandingan jarak *material handling* antara Algoritma Semut dan *Tabu Search* seperti table 4.35 berikut :

Table 4.35 Perbandingan Jarak *Material Handling*

Produk	Jarak (Meter)		
	Alg Semut	Alg. *Tabu Search*	Selisih
Coffee Table	1139.94	1031.29	108.65
End Table	979.34	1011.72	-32.38
Square Table	861.33	868.98	-7.65
	2980.60	**2911.99**	**68.62**

2. Perbandingan biaya *material handling* :

Total biaya *material handling* Algoritma Semut = Rp. 1.208.096,79

Total biaya *material handling* Algoritma *Tabu Search* = Rp. 1.180.287,79

Maka didapat selisih sebesar = Rp. 27.809,00

BAB V

PEMBAHASAN

Permasalahan yang diangkat dalam penelitian ini yaitu meminimasi jarak dan biaya pemindahan bahan, dalam hal ini mencari alternatif solusi terbaik dari kondisi *layout* awal dan *layout* usulan berdasarkan hasil yang didapatkan dari pengolahan data menggunakan Algoritma Semut dan Algoritma *Tabu Search*..

Kedua Algoritma ini dipilih karena sama – sama berbasis pada teknik heuristic yang sangat cocok untuk penyelesaian masalah optimasi. Dengan demikian akan diperoleh suatu perbandingan yang nyata dalam masalah pembentukan Cellular Manufacturing System (CMS) dengan tujuan menekan biaya *material handling*.

5.1 Analisis Implementasi Algoritma Semut

5.1.1 Analisis Tata Letak Usulan dengan Algoritma Semut

Dari hasil pengolahan Algoritma Semut didapat pergantian posisi mesin sebagai berikut :

1. Posisi mesin 1 diganti mesin 4.
2. Posisi mesin 2 diganti mesin 9.
3. Posisi mesin 3 diganti mesin 5.
4. Posisi mesin 4 diganti mesin 7..
5. Posisi mesin 5 diganti mesin 3.
6. Posisi mesin 6 tetap pada mesin 6

7. Posisi mesin 7 diganti mesin 2.
8. Posisi mesin 8 tatap pada mesin 8
9. Posisi mesin 9 diganti mesin 10.
10. Posisi mesin 10 diganti mesin 1.

Re-layout dengan penggantian posisi mesin, diasumsikan tidak akan mengganggu jalannya proses produksi. Penggantian dilakukan hanya unutuk mengurangi jarak pemindahan bahan sehingga biaya *material handling* dapat ditekan.

Hasil perubahan susunan mesin pada Algoritma Semut ini merupakan hasi perhitungan yang memiliki fungsi tujuan sesuai dengan persamaan pada BAB II yakni persamaan ....15) untuk optimasi mesin dan persamaan .....16) untuk optimasi *part*. Dari fungsi tujuan tersebut algoritma ini akan perhitungan untuk menemukan rute terpendek.

Algoritma ini menjalankan perhitungan dengan konsep dasarnya yaitu masing-masing semut akan ditempatkan pada setiap mesin ataupun *part* yang akan mencari rute terpendeknya berdasarkan probabilitas sesuai dengan persamaan .....19) pada sistem semut dengan menggunakan parameter-parameter yang telah ditetapkan. Dengan demikian setiap semut akan meninggalkan jejak semut yang akan diikuti oleh semut selanjutnya pada setiap jarak antar kota *(edge)*, yang akan dihitung dan ditentukan jarak terpendeknya dari masing-masing iterasi. Hasil dari masing masing iterasi akan dimasukkan ke dalam *Tabu List* untuk dipilih rute yang terpendek dari seluruh iterasi.

Sama halnya seperti diatas, penggunaan ukuran *batch* distribusi disini sangat penting karena berpengaruh terhadap frekuensi pemindahan yang nantinya akan menentukan besarnya total jarak *material handling*.

Berdasarkan perhitungan pada Bab IV didapat jarak *material handling layout* akhir seperti pada tabel dibawah ini :

Tabel 5.1 Jarak *Material Handling Layout* Usulan (Algoritma Semut)

No	Nama Komponen	Produk	Total Jarak (meter)
1	Frame Panjang	*Coffee Table*	195.53
2	Frame Pendek	*Coffee Table*	195.53
3	Frame Tengah	*Coffee Table*	130.35
4	Sunduk Panjang	*Coffee Table*	132.92
5	Sunduk Pendek	*Coffee Table*	132.92
6	Sunduk Tengah	*Coffee Table*	88.61
7	Kloss	*Coffee Table*	264.07
8	Frame Atas	*GothicTable*	177.22
9	Ambal Bawah	*GothicTable*	118.01
10	Ambal	*GothicTable*	118.01
11	List Ambal	*GothicTable*	236.01
12	Kaki Meja	*GothicTable*	330.09
13	Frame Atas	*Square Table*	177.22
14	Ambal	*Square Table*	118.01
15	List Ambal	*Square Table*	236.01
16	Kaki Meja	*Square Table*	330.09

Berdasarkan tabel diatas, total jarak *material handling* pada *layou*t akhir adalah 2980,60 meter.

Dengan demikian diperoleh biaya pemindahan per meter yaitu sebesar Rp 405,32 /meter dan jarak total *material handling* yaitu 2980,60 meter, maka total biaya pemindahan tiap bulan yang harus dikeluarkan CV. Gambang Emas adalah sebesar Rp 1.208.096,79 / bulan.

5.2 Analisis Implementasi Algoritma *Tabu Search*

5.2.1 Analisis Tata Letak Usulan dengan Algoritma *Tabu Search*

Dari hasil pengolahan Algoritma *Tabu Search* didapat pergantian posisi mesin sebagai berikut :

1. Posisi mesin 1 tetap pada mesin 1.
2. Posisi mesin 2 tetap pada mesin 2.
3. Posisi mesin 3 diganti mesin 5.
4. Posisi mesin 4 diganti mesin 8..
5. Posisi mesin 5 diganti mesin 9.
6. Posisi mesin 6 diganti mesin 10
7. Posisi mesin 7 diganti mesin 4.
8. Posisi mesin 8 diganti mesin 7
9. Posisi mesin 9 diganti mesin 3.
10. Posisi mesin 10 diganti mesin 6.

Re-layout dengan penggantian posisi mesin, diasumsikan tidak akan mengganggu jalannya proses produksi. Penggantian dilakukan hanya unutuk mengurangi jarak pemindahan bahan sehingga biaya *material handling* dapat ditekan.

Hasil perubahan susunan mesin pada Algoritma *tabu Search* ini merupakan hasi perhitungan yang memiliki fungsi tujuan sesuai dengan persamaan pada BAB II yakni persamaan ....15) untuk optimasi mesin dan persamaan .....16) untuk optimasi *part*. Dari fungsi tujuan tersebut algoritma ini akan perhitungan untuk menemukan rute terpendek.

Keunikan dari Algoritma *Tabu Search* karena memiliki Sistematis untuk mencapai tujuan dengan menggunakan dua macam *tools* yakni *Adapative Memory* dan *Responsive Memory*. *Adaptive Memory* menuntun suatu prosedur yang mampu mencari solusi yang diinginkan dengan efektif dengan prinsip : *recency* yang menjaga struktur terbaik dari solusi awal yang ditemukan; *frequency* yang merekam informasi guna mengevaluasi pergerakan/*move* yang terjadi; *quality* yang memiliki kemampuan untuk membedakan solusi terbaik dari tiap iterasi; dan *influence* untuk mempertimbangkan efek dan kualitas dari pemilihan iterasi selama pencarian berlangsung. Sedangkan *responsive exploration* lebih menekankan pada tahapan tiap proses yang harus dilalui selama proses pencarian itu berlangsung, dimana pada setiap tahapan tersebut dipunyai suatu variable keputusan yang akan menuntun pada tahapan selanjutnya sampai akhir proses pencarian dihentikan. Dengan demikian akan dihasilkan kombinasi mesin dengan rute *material handling* terpendek.

5.2.2 Hasil Pembentukan Sel Manufaktur untuk Algoritma *Tabu Search*

Pengelompokkan part dan mesin terbaik menghasilkan dua sel manufaktur yai'u :

Sel 1 : Part Family : 1 2 3 4 5 12 7 8 6

Mesin : 1 2 5 8 9

Sel 2 : Part Family : 10 11 9

Mesin : 10 4 7 3 6

Sehingga pembentukan sel manufaktur yang terbentuk adalah dua sel manufaktur. Pembentukan sel manufaktur yang dilakukan pada penelitian ini dilakukan

berdasarkan urutan produksi setiap komponen dan hubungan antara komponen dengan mesin yang ditunjukan melalui matrik komponen-mesin.

Pada *layout* usulan dengan Algoritma T*abu Search* mempunyai nilai effisiensi dan nilai *efficacy* pengelompokan berdasarkan matriks akhir yaitu :

1. Effisiensi pengelompokkan

   $\eta = 0.7333 = 73.33\ \%$

2. *Efficacy* pengelompokkan

   $\tau = 0.6145 = 61.45\ \%$

Sama halnya seperti diatas, penggunaan ukuran *batch* distribusi disini sangat penting karena berpengaruh terhadap frekuensi pemindahan yang nantinya akan menentukan besarnya total jarak *material handling*.

Pembentukan sel manufaktur, nilai effisiensi dan nilai *efficacy* diatas diperoleh dari pengolahan matriks akhir yang dihasilkan oleh Algoritma Semut. Yaitu dengan melakukan pengukuran performansi dari beberapa alternatif sel manufaktur yang ada, dengan menggabungkan mesin-mesin yang berdekatan dan memilki kerja ke dalam satu sel manufaktur. Dari hasil perhitungan ternyata diperoleh nilai effisiensi dan *efficacy* terbesar adalah dengan mengelompokkan mesin-mesin seperti di atas.

Berdasarkan perhitungan pada Bab IV didapat jarak *material handling layout* akhir seperti pada tabel dibawah ini :

Tabel 5.2 Jarak *Material Handling Layout* Usulan (Algoritma *Tabu Search)*

No	Nama Komponen	Produk	Total Jarak (meter)
1	Frame Panjang	*Coffee Table*	169.83
2	Frame Pendek	*Coffee Table*	169.83
3	Frame Tengah	*Coffee Table*	113.22
4	Sunduk Panjang	*Coffee Table*	141.13
5	Sunduk Pendek	*Coffee Table*	141.13
6	Sunduk Tengah	*Coffee Table*	94.08
7	Kloss	*Coffee Table*	202.07
8	Frame Atas	*GothicTable*	188.17
9	Ambal Bawah	*GothicTable*	142.74
10	Ambal	*GothicTable*	142.74
11	List Ambal	*GothicTable*	285.48
12	Kaki Meja	*GothicTable*	252.59
13	Frame Atas	*Square Table*	188.17
14	Ambal	*Square Table*	142.74
15	List Ambal	*Square Table*	285.48
16	Kaki Meja	*Square Table*	252.59

Berdasarkan tabel diatas, total jarak *material handling* pada *layout* akhir adalah 2911,99 meter.

Berdasarkan biaya pemindahan per meter yaitu sebesar Rp 405,32 /meter dan jarak total *material handling* yaitu 2911,99 meter, maka total biaya pemindahan tiap bulan yang harus dikeluarkan CV. Gambang Emas adalah sebesar Rp 1.180.287,79 / bulan.

5.3 Analisis Perbandingan Aplikasi Algoritma Semut dan Algoritma *Tabu Search* Dalam Perancangan Ulang *Layout*

Berdasarkan perhitungan dan pengolahan data dari kedua algoritma tersebut maka pada kasus ini diperoleh hasil bahwa Algoritma *Tabu Search* lebih baik dengan selisih sebagai berikut :

1. Untuk selisih jarak total *material handling* sebesar 68.62 meter.
2. Untuk selisih biaya total *material handling* sebesar Rp. 27.809,00

Dengan demikian, dalam kasus ini minimasi terbaik didapat dengan menggunakan Algoritma *Tabu Search.*

5.4 Analisis Pengaruh Aplikasi Algoritma Semut dan Algoritma *Tabu Search* Terhadap Fasilitas Lain.

Pengolahan data dengan Algoritma Semut posisi mesin 2 yang merupakan mesin awal proses terletak mendekati pintu keluar produk jadi menuju gudang, sedangkan mesin 9 yang merupakan mesin akhir proses justru terletak mendekati pintu masuk *part* dari *store* bahan baku. Hal ini justru merugikan karena akan menambah jarak *material handling* secara keseluruhan, baik pada saat proses produksi akan dimulai dan setelah produk jadi akan dipindahkan ke gudang.

Dari hasil pengolahan data dengan Algoritma *Tabu Search* posisi mesin 2 tetap pada posisinya di dekat pintu masuk *part* dari *store*, dan mesin 9 yang merupakan mesin akhir justru semakin mendekati pintu keluar produk jadi menuju gudang. Dengan demikian, total jarak *material handling* secara keseluruhan dari produk justru akan semakin kecil.

BAB VI

PENUTUP

6.1 KESIMPULAN

1. Berdasarkan hasil perhitungan, selisih jarak dan biaya *material handling* antara *layout* awal dan *layout* sesudah perancangan kembali tata letak fasilitas dengan dengan menggunakan Algoritma Semut adalah sebesar 41,74 meter dan Rp. 16.903,21 / bulan. .
2. Berdasarkan hasil perhitungan, selisih jarak dan biaya *material handling* antara *layout* awal dan *layout* sesudah perancangan kembali tata letak fasilitas dengan dengan menggunakan Algoritma *Tabu Search* adalah sebesar 110,35 meter dan Rp. 74.712,21 / bulan..
3. Berdasarkan perhitungan, selisih jarak dan biaya *material handling* antara *layout* Algoritma Semut dan *layout* Algoritma *Tabu Search* sesudah perancangan kembali tata letak fasilitas adalah sebesar 68.62 meter dan Rp. 27.809,00. Sedangkan, untuk tingkat efisiensi dan *efficacy* dari Algoritma Semut dan Algoritma *Tabu Search* adalah :

    a. Algoritma Semut

        - Effisiensi pengelompokkan

          $\eta = 0.516 = 51.6\ \%$

        - *Efficacy* pengelompokkan

          $\tau = 0.476 = 47.6\ \%$

b. Algoritma *Tabu Search*

- Effisiensi pengelompokkan

  $\eta = 0.7333 = 73.33\ \%$

- *Efficacy* pengelompokkan

  $\tau = 0.6145 = 61.45\ \%$

6.2 SARAN

Saran yang dapat penulis berikan pada perusahaan, berdasarkan *layout* usulan yang dihasilkan adalah sebagai berikut :

1. Pihak CV. Gambang Emas perlu melakukan Peramalan *(Forecasting)* untuk memperkirakan permintaan dari pasar agar produksi dapat berjalan dengan lebih baik.
2. Penyempurnaan tata letak fasilitas produksi pada perusahaan CV. Gambang Emas perlu dilakukan untuk meningkatkan efesiensi. Dengan perubahan tata letak tersebut akan diperoleh hasil yang lebih optimal.
3. Sebaiknya perusahaan dalam hal ini CV. Gambang Emas mempertimbangkan usulan perbaikan tata letak fasilitas produksi ini, karena adanya pengurangan biaya total perpindahan material yang cukup besar. Berkurangnya total biaya perpindahan material ini berarti akan memperbesar keuntungan yang diperoleh perusahaan.
4. Untuk penelitian lebih lanjut, hendaknya peneliti menganalisa biaya jika adanya penambahan mesin.

DAFTAR PUSTAKA

Glover, F., Laguna, M. 1997. *Tabu Search*. Boston: Kluwer Academic Publishers.

Aguilar, J. 2001. A General Ant Colony Model To Solve Combinatorial Optimization Problems. Vol. 2 No. 1 pp 7-18

Dorigo, M., Coloni, A. 1996. The Ant System : Optimization By A Colony Of Cooperating Agents. *IEEE Transaction on System, Man and Cybernetics-Komponen B.* vol 26 no..1 pp1-13.

Dorigo, M., Coloni, A. 1994. Distributed Optimization by Ant Colonies. *Proceedings of ECAL91.* pp 134-142

Dorigo,M., Gambardella, LM. 1997. Ant Colony System : A Cooperative Learning Approach To The Traveling Salesman Problem, *IEEE Transactions on Evolutionary Computation.* Vol. 1 no. 1

Maulana, A. 2005. Penerapan Cellular Manucakturing System untuk Meminimalkan Biaya Material Handling dengan Menggunakan Algoritma Tabu Search (Tugas Akhir). Yogyakarta : FTI. UII

Purnomo, H., Ridwan, Andi M. 2002. Pembentukan Sel Manufaktur Berbasis TSP yang Dimodifikasi Menggunakan Algoritma Genetik. *TEKNOIN.* Vol.6, No.3, hal 213-223,2001

Purnomo, H., Zukhri, Z. 2002. Studi Komparasi Penerapan Cellular Manufacturing Antara Algoritma Heuristic dan Algoritma Semut. *Prosiding Seminar Nasional TIMP.*

Purnomo, H., Zukhri, Z. 2002. Optimasi Pembentukan Sel Manufaktur Berbasis TSP Dengan Algoritma Semut. *Prosiding Seminar Nasional Pengukuran Kinerja dan Perencanaan Strategi. Jurusan Teknik Manajemen Industri.* pp195-206

Purnomo, H. 2004. *Perencanaan & Perancangan Fasilitas*. Yogyakarta : Penerbit Graha Ilmu.

Suryoko, R. 2006. Pembentukan Sel Manufaktur Dengan Menggunakan Algoritma Semut (Tugas Akhir). Yogyakarta : FTI. UII

Singh, N., Rajawani, D. 1995, *Cellular Manufacturing System, Design, Planning and Control*. London: Chapman & Hall

Kusumadewi, S., Purnomo, H. 2005, *Penyelesaian Masalah Optimasi dengan Teknik-teknik Heuristik*. Yogyakarta : Graha Ilmu

Wignjosoebroto, S., 1996, *Tata Letak Pabrik dan Pemindahan Bahan*. Jakarta : Penerbit Guna Widya

Wahyuningsih, T. 2006. Penerapan Cellular Manufacturing System Untuk Meminimasi Biaya Material Handling (Tugas Akhir). Yogyakarta : FTI. UII

LAMPIRAN

5.1.2 Hasil Pembentukan Sel Manufaktur untuk Algoritma Semut

Pengelompokkan part dan mesin terbaik menghasilkan dua sel manufaktur yaitu :

Sel 1 : Part Family : 11 2 3 6 8 12 1 5 9

Mesin : 4 9 5 7 3 6 2 8

Sel 2 : Part Family : 10 4 7

Mesin : 10 1

Dengan demikian dikatakan bahwa bentuk umum layout awal tidak berubah, hanya saja posisi setiap mesin produksi pada layout awal diganti dengan mesin produksi layout usulan yang pengaturannya berdasarkan hasil pembentukan sel manufaktur.

Pada *layout* usulan dengan Algoritma Semut mempunyai nilai effisiensi dan nilai *efficacy* pengelompokan berdasarkan matriks akhir yaitu :

1. Effisiensi pengelompokkan

   $\eta = 0.516 = 51.6\ \%$

2. *Efficacy* pengelompokkan

   $\tau = 0.476 = 47.6\ \%$

Pembentukan sel manufaktur, nilai effisiensi dan nilai *efficacy* diatas diperoleh dari pengolahan matriks akhir yang dihasilkan oleh Algoritma Semut. Yaitu dengan melakukan pengukuran performansi dari beberapa alternatif sel manufaktur yang ada, dengan menggabungkan mesin-mesin yang berdekatan dan memilki kerja ke dalam satu sel manufaktur. Dari hasil perhitungan ternyata diperoleh nilai effisiensi dan *efficacy* terbesar adalah dengan mengelompokkan mesin-mesin seperti di atas.

HASIL PERAMALAN

Produk : Coffee Table

Hasil Peramalan Tahun Pertama

8/2007 nth	Actual Data	Forecast by LR	Forecast Error	CFE	MAD	MSE	MAPE (%)	Tracking Signal	R-square
1	55	52.70514	2.294865	2.294865	2.294865	5.266404	4.172481	1	1
2	50	54.3345	-4.3345	-2.03964	3.314684	12.02716	6.420744	-0.61533	0.272599
3	57	55.96387	1.036125	-1.00351	2.555164	8.375958	4.886417	-0.39274	0.21713
4	62	57.59324	4.406757	3.403244	3.018063	11.13685	5.441731	1.127625	0.21851
5	60	59.22261	0.777386	4.18063	2.569927	9.030343	4.612514	1.62675	0.346129
6	65	60.85198	4.148018	8.328648	2.832942	10.39296	4.907355	2.939929	0.406214
7	55	62.48135	-7.48135	0.847294	3.497001	16.90406	6.149513	0.242292	0.491573
8	58	64.11073	-6.11073	-5.26343	3.823717	19.45867	6.697791	-1.37652	0.758855
9	65	65.7401	-0.7401	-6.00353	3.481092	17.35746	6.080104	-1.72461	0.8238
10	68	67.36946	0.630539	-5.37299	3.196037	15.66147	5.56482	-1.68114	0.796811
11	75	68.99883	6.001167	0.628178	3.451049	17.5117	5.786341	0.182025	0.587771
12	70	70.6282	-0.6282	-2.67E-05	3.215812	16.08528	5.378932	-8.30E-06	0.662939
13		72.25758							
14		73.88694							
15		75.51631							
16		77.14568							
17		78.77505							
18		80.40443							
19		82.03379							
20		83.66316							
21		85.29253							
22		86.92191							
23		88.55127							
24		90.18064							
		-2.67E-05							
)		3.215812							
=		16.08528							
PE		5.378932							
Signal		-8.30E-06							
quare		0.6629389							

		a=51.0758							
		b=1.6294							

Hasil Peramalan Tahun Kedua

2007 1	Actual Data	Forecast by LR	Forecast Error	CFE	MAD	MSE	MAPE (%)	Tracking Signal	R-sqaure
1	55	52.58334	2.416656	2.416656	2.416656	5.840229	4.393921	1	1
2	50	54.24638	-4.24638	-1.82973	3.33152	11.936	6.443344	-0.54922	0.244544
3	57	55.90943	1.090572	-0.73915	2.584538	8.353784	4.933324	-0.28599	0.219751
4	62	57.57247	4.427528	3.688374	3.045285	11.16609	5.485287	1.211175	0.232832
5	60	59.23551	0.764488	4.452862	2.589126	9.049761	4.643059	1.719832	0.364317
6	65	60.89856	4.101444	8.554306	2.841179	10.34511	4.920868	3.01083	0.424243
7	55	62.5616	-7.5616	0.992706	3.515525	17.03549	6.181939	0.282378	0.512325
8	58	64.22464	-6.22464	-5.23193	3.854164	19.74932	6.750714	-1.35748	0.789317
9	65	65.88769	-0.88769	-6.11962	3.524556	17.64251	6.152376	-1.73628	0.858147
10	68	67.55073	0.449272	-5.67035	3.217027	15.89844	5.603208	-1.76261	0.830831
11	75	69.21377	5.786232	0.115883	3.450591	17.49681	5.795187	3.36E-02	0.612244
12	70	70.87682	-0.87682	-0.76093	3.23611	16.10281	5.416637	-0.23514	0.690707
13	73	72.53986	0.460144	-0.30079	3.022574	14.88042	5.04846	-9.95E-02	0.728218
14	74	74.2029	-0.2029	-0.50369	2.821169	13.82047	4.70744	-0.17854	0.773741
15	76	75.86594	0.134056	-0.36963	2.642028	12.9003	4.405371	-0.1399	0.804664
16	78	77.52898	0.471016	0.101387	2.50634	12.1079	4.167777	4.05E-02	0.824184
17	79	79.19202	-0.19202	-9.06E-02	2.370203	11.39784	3.936911	-3.82E-02	0.852425
18	81	80.85507	0.144928	5.43E-02	2.246577	10.7658	3.728134	2.42E-02	0.870342
19	83	82.51811	0.481888	0.536179	2.153699	10.2114	3.562474	0.248957	0.880399
20	84	84.18115	-0.18115	0.355026	2.055071	9.702468	3.395133	0.172756	0.898575
21	86	85.8442	0.1558	0.510826	1.96463	9.241602	3.242087	0.260011	0.909387
22	87	87.50724	-0.50724	3.59E-03	1.898385	8.833224	3.121221	1.89E-03	0.926629
23	89	89.17028	-0.17028	-0.16669	1.82325	8.450432	2.993834	-9.14E-02	0.93715
24	91	90.83333	0.166672	-2.29E-05	1.754226	8.099488	2.876723	-1.30E-05	0.942402
25		92.49637							
26		94.15941							
27		95.82246							
28		97.4855							
29		99.14854							
30		100.8116							
31		102.4746							

32		104.1377							
33		105.8007							
34		107.4638							
35		109.1268							
36		110.7898							
		-2.29E-05							
		1.754226							
		8.099488							
		2.876723							
gnal		-1.30E-05							
are		0.942402							
		a=50.9203							
		b=1.6630							

Hasil Peramalan Tahun Ketiga

007	Actual Data	Forecast by LR	Forecast Error	CFE	MAD	MSE	MAPE (%)	Tracking Signal	R-square
1	55	52.42194	2.578064	2.578064	2.578064	6.646414	4.687389	1	1
2	50	54.10418	-4.104176	-1.52611	3.34112	11.74534	6.44787	-0.45677	0.206358
3	57	55.78641	1.213589	-0.31252	2.631943	8.321157	5.008281	-0.11874	0.218939
4	62	57.46865	4.531349	4.218826	3.106794	11.37415	5.583368	1.357935	0.251341
5	60	59.15089	0.84911	5.067936	2.655257	9.243516	4.749731	1.908642	0.385208
6	65	60.83313	4.16687	9.234806	2.907193	10.59673	5.026537	3.176537	0.446236
7	55	62.51537	-7.515369	1.719437	3.565504	17.15159	6.260505	0.482242	0.526059
8	58	64.19761	-6.197609	-4.47817	3.894517	19.80894	6.813633	-1.14987	0.801081
9	65	65.87984	-0.879845	-5.35802	3.559553	17.69396	6.206964	-1.50525	0.872685
10	68	67.56209	0.437912	-4.92011	3.247389	15.94374	5.650666	-1.5151	0.847001
11	75	69.24432	5.755676	0.835571	3.475415	17.50593	5.834627	0.240423	0.626585
12	70	70.92656	-0.926559	-9.10E-02	3.263011	16.11864	5.458712	-2.79E-02	0.706659
13	73	72.6088	0.391197	0.300209	3.042102	14.89052	5.080033	9.87E-02	0.745126
14	74	74.29104	-0.291039	9.17E-03	2.845598	13.83296	4.745266	3.22E-03	0.791684
15	76	75.97328	2.67E-02	3.59E-02	2.657672	12.91081	4.431259	1.35E-02	0.823337
16	78	77.65552	0.344482	0.380371	2.513098	12.1113	4.181908	0.151356	0.843328
17	79	79.33775	-0.337753	4.26E-02	2.385136	11.40558	3.961063	1.79E-02	0.872217
18	81	81.02	-2.00E-02	2.26E-02	2.25374	10.77196	3.742375	1.00E-02	0.89055
19	83	82.70223	0.297768	0.320389	2.150794	10.20968	3.56429	0.148963	0.900835

20	84	84.38448	-0.384476	-6.41E-02	2.062478	9.706591	3.408961	-3.11E-02	0.919436
21	86	86.06671	-6.67E-02	-0.1308	1.967441	9.244585	3.250323	-6.65E-02	0.930496
22	87	87.74895	-0.748947	-0.87975	1.912055	8.849873	3.141711	-0.4601	0.948157
23	89	89.43119	-0.431191	-1.31094	1.84767	8.47318	3.02618	-0.70951	0.958934
24	91	91.11343	-0.113426	-1.42436	1.77541	8.120667	2.905282	-0.80227	0.964309
25	93	92.79567	0.20433	-1.22003	1.712566	7.79751	2.797859	-0.7124	0.965567
26	95	94.47791	0.522095	-0.69794	1.666779	7.50809	2.711387	-0.41873	0.963776
27	96	96.16014	-0.160141	-0.85808	1.610978	7.230962	2.617143	-0.53264	0.968708
28	98	97.84238	0.157616	-0.70046	1.559072	6.973601	2.529418	-0.44928	0.970277
29	100	99.52462	0.47538	-0.22508	1.521703	6.740924	2.458589	-0.14791	0.969258
30	101	101.2069	-0.206863	-0.43195	1.477875	6.517653	2.383464	-0.29227	0.97359
31	103	102.8891	0.110901	-0.32104	1.433779	6.307803	2.310051	-0.22392	0.975146
32	105	104.5713	0.428665	0.10762	1.402369	6.116426	2.25062	7.67E-02	0.974507
33	106	106.2536	-0.253578	-0.14596	1.367558	5.933028	2.189668	-0.10673	0.978257
34	108	107.9358	0.064186	-8.18E-02	1.329223	5.758648	2.127014	-6.15E-02	0.979688
35	110	109.6181	0.381943	0.300171	1.302158	5.598284	2.076163	0.230518	0.979245
36	111	111.3003	-0.300293	-1.22E-04	1.274328	5.445281	2.026007	-9.58E-05	0.982482
37		112.9825							
38		114.6648							
39		116.347							
40		118.0293							
41		119.7115							
42		121.3937							
43		123.076							
44		124.7582							
45		126.4404							
46		128.1227							
47		129.8049							
48		131.4872							
		-1.22E-04							
		1.274328							
		5.445281							
		2.026007							
ıal		-9.58E-05							
e		0.9824815							
		a=50.7397							
		b=1.6822							

Hasil Peramalan (unit)

Data	Tahun 1	Tahun 2	Tahun 3
55	73	93	113
50	74	95	115
57	76	96	117
62	78	98	119
60	79	100	120
65	81	101	122
55	83	103	124
58	84	105	125
65	86	106	127
68	87	108	129
75	89	110	130
70	91	111	132

Rata - rata (unit)

61.66667	81.75	102.1667	122.75
62	82	103	123

Produk : Gothic Table

Hasil Peramalan Tahun Pertama

2007	Actual Data	Forecast by DES	Forecast Error	CFE	MAD	MSE	MAPE (%)	Tracking Signal	R-square
1	55								
2	56	55	1	1	1	1	1.785714	1	1
3	58	55.49	2.509998	3.509998	1.754999	3.650046	3.056649	2	1
4	63	56.764	6.236	9.745998	3.248666	15.39593	5.337237	3	1
5	58	59.9343	-1.9343	7.811703	2.920074	12.48232	4.836676	2.675173	1
6	66	59.27182	6.72818	14.53988	3.681695	19.03954	5.908183	3.949236	0.900615
7	61	62.509	-1.509	13.03088	3.319579	16.24579	5.335781	3.925463	1
8	63	62.06094	0.93906	13.96994	2.979505	14.05094	4.786465	4.68868	1
9	68	62.48075	5.519249	19.48919	3.296973	16.10234	5.202725	5.911239	0.945706
10	62	65.22296	-3.22296	16.26623	3.288749	15.46735	5.202236	4.946023	1
11	63	63.89051	-0.89051	15.37572	3.048925	13.99992	4.823363	5.042997	1
12	58	63.33424	-5.33424	10.04148	3.256681	15.31394	5.220963	3.083348	0.969188
13		60.6704							
14		60.6704							
15		60.6704							
16		60.6704							
17		60.6704							
18		60.6704							
19		60.6704							
20		60.6704							
21		60.6704							
22		60.6704							
23		60.6704							
24		60.6704							
		10.04148							
		3.256681							
		15.31394							
		5.220963							
ıal		3.083348							
re		0.9691877							
		Alpha=0.7							
		F(0)=55							
		F'(0)=55							

Hasil Peramalan Tahun Kedua

3/2007 ıth	Actual Data	Forecast by DES	Forecast Error	CFE	MAD	MSE	MAPE (%)	Tracking Signal	R-square
1	55								
2	56	55	1	1	1	1	1.785714	1	1
3	58	55.4761	2.523903	3.523903	1.761951	3.685043	3.068635	2	1
4	63	56.72348	6.27652	9.800423	3.266808	15.58826	5.366667	3	1
5	58	59.8316	-1.8316	7.968822	2.908006	12.52989	4.814483	2.740305	1
6	66	59.25827	6.741734	14.71056	3.674751	19.1141	5.894536	4.003143	0.907908
7	61	62.4129	-1.4129	13.29765	3.297776	16.26114	5.298152	4.032309	1
8	63	62.04338	0.956619	14.25427	2.963326	14.06885	4.758194	4.810228	1
9	68	62.46331	5.536686	19.79096	3.284996	16.1421	5.181193	6.024653	0.95213
10	62	65.13968	-3.13968	16.65128	3.268849	15.44382	5.168171	5.093927	1
11	63	63.90208	-0.90208	15.7492	3.032172	13.98081	4.794541	5.194034	1
12	58	63.35366	-5.35366	10.39554	3.243217	15.31544	5.197806	3.205317	0.970572
13	61	60.75209	0.247913	10.64345	2.993608	14.04428	4.798523	3.555392	0.966819
14	61	60.62011	0.379894	11.02335	2.792553	12.97505	4.477312	3.947408	0.965065
15	61	60.78829	0.211712	11.23506	2.608207	12.05146	4.182295	4.307578	0.962114
16	61	60.90525	9.48E-02	11.32981	2.440644	11.24863	3.913831	4.64214	0.958777
17	61	60.9616	3.84E-02	11.36821	2.290504	10.54568	3.673151	4.963194	0.955558
18	61	60.9853	1.47E-02	11.38292	2.156633	9.925362	3.458501	5.278095	0.952613
19	61	60.99458	0.005425	11.38834	2.037122	9.373954	3.266856	5.590408	0.94996
20	61	60.99805	1.95E-03	11.39029	1.930007	8.880588	3.095084	5.901682	0.947576
21	61	60.99931	6.87E-04	11.39098	1.833541	8.436559	2.940387	6.212555	0.945428
22	61	60.99976	2.40E-04	11.39122	1.746241	8.034818	2.800387	6.523278	0.943484
23	61	60.99992	8.39E-05	11.3913	1.66687	7.669599	2.673103	6.833945	0.941718
24	61	60.99997	3.05E-05	11.39133	1.594399	7.336138	2.556883	7.144592	0.940107
25		60.99999							
26		60.99999							
27		60.99999							
28		60.99999							
29		60.99999							
30		60.99999							
31		60.99999							
32		60.99999							
33		60.99999							
34		60.99999							
35		60.99999							
36		60.99999							
		11.39133							

D		1.594399
E		7.336138
PE		2.556883
Signal		7.144592
quare		0.9401072
		Alpha=0.69
		F(0)=55
		F'(0)=55

Hasil Peramalan Tahun Ketiga

3/2007 ıth	Actual Data	Forecast by DES	Forecast Error	CFE	MAD	MSE	MAPE (%)	Tracking Signal	R-square
1	55								
2	56	55	1	1	1	1	1.785714	1	
3	58	55.4761	2.523903	3.523903	1.761951	3.685043	3.068635	2	
4	63	56.72348	6.27652	9.800423	3.266808	15.58826	5.366667	3	
5	58	59.8316	-1.8316	7.968822	2.908006	12.52989	4.814483	2.740305	
6	66	59.25827	6.741734	14.71056	3.674751	19.1141	5.894536	4.003143	0.907907
7	61	62.4129	-1.4129	13.29765	3.297776	16.26114	5.298152	4.032309	
8	63	62.04338	0.956619	14.25427	2.963326	14.06885	4.758194	4.810228	
9	68	62.46331	5.536686	19.79096	3.284996	16.1421	5.181193	6.024653	0.9521
10	62	65.13968	-3.13968	16.65128	3.268849	15.44382	5.168171	5.093927	
11	63	63.90208	-0.90208	15.7492	3.032172	13.98081	4.794541	5.194034	
12	58	63.35366	-5.35366	10.39554	3.243217	15.31544	5.197806	3.205317	0.970571
13	61	60.75209	0.247913	10.64345	2.993608	14.04428	4.798523	3.555392	0.966818
14	61	60.62011	0.379894	11.02335	2.792553	12.97505	4.477312	3.947408	0.96506
15	61	60.78829	0.211712	11.23506	2.608207	12.05146	4.182295	4.307578	0.96211
16	61	60.90525	9.48E-02	11.32981	2.440644	11.24863	3.913831	4.64214	0.95877
17	61	60.9616	3.84E-02	11.36821	2.290504	10.54568	3.673151	4.963194	0.95555
18	61	60.9853	1.47E-02	11.38292	2.156633	9.925362	3.458501	5.278095	0.95261
19	61	60.99458	0.005425	11.38834	2.037122	9.373954	3.266856	5.590408	0.94996
20	61	60.99805	1.95E-03	11.39029	1.930007	8.880588	3.095084	5.901682	0.94757
21	61	60.99931	6.87E-04	11.39098	1.833541	8.436559	2.940387	6.212555	0.94542
22	61	60.99976	2.40E-04	11.39122	1.746241	8.034818	2.800387	6.523278	0.94348
23	61	60.99992	8.39E-05	11.3913	1.66687	7.669599	2.673103	6.833945	0.9417
24	61	60.99997	3.05E-05	11.39133	1.594399	7.336138	2.556883	7.144592	0.94010
25	61	60.99999	7.63E-06	11.39134	1.527966	7.030466	2.450347	7.45523	0.93863
26	61	61	3.81E-06	11.39134	1.466848	6.749247	2.352333	7.765866	0.93727

27	61	61	0	11.39134	1.41043	6.48966	2.261859	8.076501	0.93602
28	61	61	0	11.39134	1.358192	6.249302	2.178086	8.387136	0.9348
29	61	61	0	11.39134	1.309685	6.026113	2.100297	8.69777	0.93378
30	61	61	0	11.39134	1.264524	5.818316	2.027874	9.008405	0.93278
31	61	61	0	11.39134	1.222373	5.624372	1.960278	9.31904	0.93185
32	61	61	0	11.39134	1.182942	5.442941	1.897043	9.629675	0.9309
33	61	61	0	11.39134	1.145975	5.272849	1.83776	9.94031	0.93016
34	61	61	0	11.39134	1.111248	5.113066	1.782071	10.25094	0.92939
35	61	61	0	11.39134	1.078564	4.962681	1.729657	10.56158	0.92867
36	61	61	0	11.39134	1.047748	4.82089	1.680238	10.87221	0.9279
37		61							
38		61							
39		61							
40		61							
41		61							
42		61							
43		61							
44		61							
45		61							
46		61							
47		61							
48		61							
E		11.39134							
D		1.047748							
E		4.82089							
PE		1.680238							
Signal		10.87221							
quare		0.927989							
		Alpha=0.69							
		F(0)=55							
		F'(0)=55							

Hasil Peramalan (unit)

Data	Tahun 1	Tahun 2	Tahun 3
55	61	61	61
56	61	61	61
58	61	61	61
63	61	61	61
58	61	61	61
66	61	61	61
61	61	61	61
63	61	61	61
68	61	61	61
62	61	61	61
63	61	61	61
58	61	61	61

Rata-rata (unit)

60.91667	61	61	61
61	61	61	61

Produk : Square Table

Hasil Peramalan Tahun Pertama

8/2007 nth	Actual Data	Forecast by LR	Forecast Error	CFE	MAD	MSE	MAPE (%)	Tracking Signal	R-square
1	36	39	-3	-3	2.999996	8.999977	8.333323	-1	1
2	41	40.31818	0.68182	-2.31818	1.840908	4.732428	4.998149	-1.25926	0.284462
3	43	41.63636	1.36364	-0.95454	1.681819	3.77479	4.389184	-0.56756	0.145343
4	41	42.95454	-1.95454	-2.90908	1.75	3.786153	4.483684	-1.66233	0.403877
5	46	44.27272	1.727276	-1.18181	1.745455	3.625619	4.337936	-0.67708	0.331868
6	51	45.59091	5.409092	4.227287	2.356061	7.897728	5.382624	1.794218	0.256818
7	44	46.90909	-2.90909	1.318199	2.435065	7.978452	5.558187	0.54134	0.373699
8	51	48.22727	2.772728	4.090927	2.477273	7.942148	5.543004	1.651383	0.406065
9	46	49.54545	-3.54545	0.545475	2.59596	8.456379	5.783504	0.210125	0.55473
10	51	50.86364	0.136364	0.681839	2.35	7.612601	5.231892	0.290144	0.628942
11	51	52.18182	-1.18182	-0.49998	2.243801	7.047518	4.966928	-0.22283	0.733177
12	54	53.5	0.5	2.29E-05	2.098485	6.481058	4.630177	1.09E-05	0.761617
13		54.81818							
14		56.13636							
15		57.45454							
16		58.77273							
17		60.09091							
18		61.40909							
19		62.72727							
20		64.04546							
21		65.36363							
22		66.68182							
23		68							
24		69.31818							
		2.29E-05							
		2.098485							
		6.481058							
E		4.630177							
Signal		1.09E-05							
quare		0.7616165							
		a=37.6818							
		b=1.3182							

Hasil Peramalan Tahun Kedua

8/2007 nth	Actual Data	Forecast by LR	Forecast Error	CFE	MAD	MSE	MAPE (%)	Tracking Signal	R-squa
1	36	38.89667	-2.89667	-2.89667	2.896671	8.390704	8.046309	-1	
2	41	40.24624	0.753761	-2.14291	1.825216	4.47943	4.942376	-1.17406	0.2565
3	43	41.5958	1.404198	-0.73871	1.684877	3.643544	4.383442	-0.43844	0.1470
4	41	42.94537	-1.94537	-2.68408	1.749999	3.67877	4.473781	-1.53376	0.4077
5	46	44.29493	1.705067	-0.97901	1.741013	3.524467	4.320358	-0.56232	0.3459
6	51	45.6445	5.355503	4.376492	2.343428	7.717291	5.350462	1.86756	0.2697
7	44	46.99406	-2.99406	1.382431	2.436375	7.895449	5.558208	0.567413	0.3918
8	51	48.34363	2.656372	4.038803	2.463875	7.790557	5.514504	1.639208	0.4247
9	46	49.69319	-3.69319	0.345612	2.600466	8.440457	5.793857	0.132904	0.5813
10	51	51.04276	-4.28E-02	0.302856	2.344695	7.596595	5.222854	0.129167	0.6590
11	51	52.39232	-1.39232	-1.08947	2.258115	7.082228	4.996235	-0.48247	0.7688
12	54	53.74189	0.258114	-0.83135	2.091449	6.497594	4.619715	-0.3975	0.7984
13	55	55.09145	-9.14E-02	-0.9228	1.937602	5.998423	4.277142	-0.47626	0.8352
14	57	56.44102	0.558983	-0.36382	1.83913	5.592282	4.04168	-0.19782	0.8435
15	58	57.79058	0.209419	-0.1544	1.730482	5.222387	3.796306	-8.92E-02	0.8624
16	59	59.14014	-0.14014	-0.29454	1.631086	4.897215	3.573883	-0.18058	0.8871
17	61	60.48971	0.510288	0.215744	1.565157	4.624461	3.412862	0.137842	0.8915
18	62	61.83928	0.160725	0.376469	1.487133	4.368982	3.23766	0.253151	0.9038
19	63	63.18884	-0.18884	0.187626	1.418802	4.140912	3.083034	0.132243	0.9212
20	65	64.53841	0.461594	0.64922	1.370941	3.94452	2.964389	0.473557	0.9233
21	66	65.88797	0.11203	0.76125	1.310993	3.757283	2.831311	0.580666	0.9317
22	67	67.23753	-0.23753	0.523716	1.2622	3.589062	2.71873	0.414923	0.9443
23	68	68.5871	-0.5871	-0.06338	1.232847	3.448002	2.638062	-5.14E-02	0.9597
24	70	69.93667	0.063332	-4.96E-05	1.184117	3.304502	2.531913	-4.19E-05	0.9635
25		71.28623							
26		72.6358							
27		73.98536							
28		75.33492							
29		76.68449							
30		78.03406							
31		79.38362							
32		80.73318							
33		82.08275							
34		83.43231							
35		84.78188							
36		86.13145							

		-4.96E-05							
)		1.184117							
		3.304502							
E		2.531913							
Signal		-4.19E-05							
uare		0.9635166							
		a=37.5471							
		b=1.3496							

Hasil Peramalan Tahun Ketiga

007	Actual Data	Forecast by LR	Forecast Error	CFE	MAD	MSE	MAPE (%)	Tracking Signal	R-squ
1	36	38.71471	-2.71471	-2.71471	2.714706	7.369631	7.540851	-1	
2	41	40.08498	0.915024	-1.79968	1.814865	4.10345	4.886308	-0.99163	0.20
3	43	41.45525	1.54475	-0.25493	1.724827	3.531051	4.455019	-0.1478	0.145
4	41	42.82552	-1.82552	-2.08045	1.75	3.481419	4.454386	-1.18883	0.391
5	46	44.19579	1.804211	-0.27624	1.760842	3.43617	4.347949	-0.15688	0.353
6	51	45.56606	5.433941	5.1577	2.373025	7.784761	5.399088	2.17347	0.286
7	44	46.93633	-2.93633	2.221367	2.453498	7.904374	5.581145	0.905388	0.407
8	51	48.3066	2.693398	4.914764	2.483485	7.823126	5.543648	1.978979	0.442
9	46	49.67687	-3.67687	1.237892	2.616084	8.456043	5.815821	0.473185	0.600
10	51	51.04714	-4.71E-02	1.19075	2.35919	7.610662	5.243482	0.504729	0.680
11	51	52.41741	-1.41741	-0.22666	2.273573	7.101425	5.01946	-9.97E-02	0.792
12	54	53.78769	0.212315	-1.43E-02	2.101802	6.513395	4.633936	-6.83E-03	0.822
13	55	55.15796	-0.15796	-0.1723	1.952275	6.014284	4.299571	-8.83E-02	0.860
14	57	56.52822	0.471775	0.299473	1.846525	5.60059	4.051579	0.162182	0.86
15	58	57.89849	0.101505	0.400978	1.730191	5.227904	3.793141	0.231754	0.889
16	59	59.26877	-0.26877	0.13221	1.638852	4.905675	3.584541	0.080672	0.914
17	61	60.63904	0.360962	0.493172	1.563682	4.62477	3.408494	0.315391	0.919
18	62	62.00931	-9.31E-03	0.483864	1.477328	4.367843	3.219967	0.327526	0.931
19	63	63.37958	-0.37958	0.104286	1.419551	4.14554	3.082206	0.073464	0.949
20	65	64.74985	0.250153	0.354439	1.361081	3.941392	2.947338	0.26041	0.951
21	66	66.12012	-0.12012	0.234322	1.301988	3.754393	2.815655	0.179972	0.960
22	67	67.49039	-0.49039	-0.25607	1.265097	3.59467	2.72094	-0.20241	0.973
23	68	68.86066	-0.86066	-1.11673	1.247513	3.470586	2.657668	-0.89517	0.989
24	70	70.23093	-0.23093	-1.34766	1.205155	3.328201	2.560678	-1.11825	0.993
25	72	71.6012	0.398796	-0.94887	1.172901	3.201434	2.480406	-0.80899	0.987
26	73	72.97147	2.85E-02	-0.92034	1.128887	3.078333	2.386508	-0.81526	0.988

27	74	74.34174	-0.34174	-1.26209	1.099733	2.968647	2.315223	-1.14763	0.9931
28	76	75.71201	0.287987	-0.9741	1.070742	2.865586	2.24607	-0.90974	0.9902
29	77	77.08228	-8.23E-02	-1.05638	1.036657	2.767006	2.172304	-1.01903	0.9917
30	79	78.45255	0.547447	-0.50893	1.02035	2.684762	2.122993	-0.49878	0.987(
31	80	79.82282	0.177177	-0.33176	0.993151	2.599169	2.061654	-0.33404	0.9865
32	81	81.1931	-0.1931	-0.52486	0.96815	2.519111	2.004677	-0.54212	0.9892
33	83	82.56337	0.43663	-8.82E-02	0.952043	2.448551	1.959871	-9.27E-02	0.986(
34	84	83.93364	6.64E-02	-2.19E-02	0.925994	2.376664	1.904551	-2.36E-02	0.9873
35	85	85.30391	-0.30391	-0.32578	0.90822	2.311399	1.860351	-0.3587	0.9904
36	87	86.67418	0.325821	4.58E-05	0.892042	2.250142	1.819077	5.13E-05	0.9890
37		88.04445							
38		89.41472							
39		90.78499							
40		92.15526							
41		93.52553							
42		94.89581							
43		96.26608							
44		97.63634							
45		99.00661							
46		100.3769							
47		101.7472							
48		103.1174							
		4.58E-05							
		0.8920419							
		2.250142							
		1.819077							
ıal		5.13E-05							
e		0.9890177							
		a=37.3444							
		b=1.3703							

PERI

ktu proses

roses = ∑ w

pen

: {(37.4 x 8

: {(54 x 82)

: {(39 x 82)

: {(15 x 82)

: {(24 x 82) -

: {(55 x 82) +

: {(18.4 x 82)

: {(37 x 82) +

: {(0 x 82) + ('

{(0 x 82) + (1

esin

$$\frac{4726,8}{12000x0.9} =$$

Hasil Peramalan (unit)

Data	Tahun 1	Tahun 2	Tahun 3
36	55	72	89
41	57	73	90
43	58	74	91
41	59	76	93
46	61	77	94
51	62	79	95
44	63	80	97
51	65	81	98
46	66	83	100
51	67	84	101
51	68	85	102
54	70	87	104

Rata - rata (unit)

46.25	62.58333	79.25	96.16667
47	63	80	97

2. Mesin Radial : $\frac{6378,1}{12000x0.9} = 0,59 \approx 1$

3. Mesin Planner : $\frac{5206,4}{12000x0.9} = 0,48 \approx 1$

4. Mesin Thickneser : $\frac{1230}{12000x0.9} = 0.114 \approx 1$

5. Mesin Bandsaw : $\frac{3456}{12000x0.9} = 0.32 \approx 1$

6. Mesin Jointer : $\frac{6742}{12000x0.9} = 0,62 \approx 1$

7. Mesin Bor : $\frac{3181,8}{12000x0.9} = 0.295 \approx 1$

8. Mesin Rotter : $\frac{7686}{12000x0.9} = 0,73 \approx 1$

9. Mesin Serut : $\frac{1241}{12000x0.9} = 0,115 \approx 1$

10. Mesin Mortiser : $\frac{1362}{12000x0.9} = 0,126 \approx 1$

PERHITUNGAN JARAK *MATERIAL HANDLING*

1. Perhitungan Jarak *Material Handling Layout* Awal

No	Nama Komponen	Routing	Material Handling	Jarak (Meter)
1	*frame panjang*	2-1-6-8-5-9	7.95+5.67+20.03+10.42+15.54	59.62
2	*frame pendek*	2-1-6-8-5-9	7.95+5.67+20.03+10.42+15.54	59.62
3	*frame tengah*	2-1-6-8-5-9	7.95+5.67+20.03+10.42+15.54	59.62
4	*sunduk panjang*	2-1-3-8-5-9	7.95+7.22+10.29+10.42+15.54	51.43
5	*sunduk pendek*	2-1-3-8-5-9	7.95+7.22+10.29+10.42+15.54	51.43
6	*sunduk tengah*	2-1-3-8-5-9	7.95+7.22+10.29+10.42+15.54	51.43
7	*kloss*	2-1-10-3-8-5-9	7.95+14.56+7.34+10.29+10.42+15.54	66.11
8	*frame atas*	2-1-3-8-5-9	7.95+7.22+10.29+10.42+15.54	51.43
9	*ambal bawah*	2-1-4-7-5-9	7.95+15.27+11.64+6.52+15.54	56.93
10	*ambal*	2-1-4-7-5-9	7.95+15.27+11.64+6.52+15.54	56.93
11	*list ambal*	2-1-4-7-5-9	7.95+15.27+11.64+6.52+15.54	56.93
12	*kaki meja*	2-1-10-3-8-5-9	7.95+14.56+7.34+10.29+10.42+15.54	66.11

2. Perhitungan Jarak *Material Handling Layout* Algoritma Semut

No	Nama Komponen	Routing	Material Handling	Jarak (Meter)
1	*frame panjang*	2-1-6-8-5-9	10.59+16.67+20.03+10.29+7.68	65.18
2	*frame pendek*	2-1-6-8-5-9	10.59+16.67+20.03+10.29+7.68	65.18
3	*frame tengah*	2-1-6-8-5-9	10.59+16.67+20.03+10.29+7.68	65.18
4	*sunduk panjang*	2-1-3-8-5-9	10.59+5.32+10.42+10.29+7.68	44.31
5	*sunduk pendek*	2-1-3-8-5-9	10.59+5.32+10.42+10.29+7.68	44.31
6	*sunduk tengah*	2-1-3-8-5-9	10.59+5.32+10.42+10.29+7.68	44.31
7	*kloss*	2-1-10-3-8-5-9	10.59+11.49+15.54+10.42+10.29+7.68	66.02
8	*frame atas*	2-1-3-8-5-9	10.59+5.32+10.42+10.29+7.68	44.31
9	*ambal bawah*	2-1-4-7-5-9	10.59+14.59+15.27+10.89+7.68	59.00
10	*ambal*	2-1-4-7-5-9	10.59+14.59+15.27+10.89+7.68	59.00
11	*list ambal*	2-1-4-7-5-9	10.59+14.59+15.27+10.89+7.68	59.00
12	*kaki meja*	2-1-10-3-8-5-9	10.59+11.49+15.54+10.42+10.29+7.68	66.02

ALGORITMA SEMUT

1. Pencarian Urutan Mesin

```
Siklus ke: 1
---------------
Minimal Semut ke 51 -> 1, 2, 5, 8, 7, 3, 9, 6, 10, 4 = 144

Siklus ke: 2
---------------
Minimal Semut ke 36 -> 6, 1, 2, 5, 8, 9, 10, 4, 3, 7 = 139

Siklus ke: 3
---------------
Minimal Semut ke 84 -> 1, 2, 5, 9, 10, 7, 6, 4, 3, 8 = 136

Siklus ke: 4
---------------
Minimal Semut ke 28 -> 1, 2, 9, 6, 3, 5, 8, 10, 4, 7 = 139

Siklus ke: 5
---------------
Minimal Semut ke 51 -> 3, 9, 10, 1, 2, 5, 8, 4, 7, 6 = 138

Siklus ke: 6
---------------
Minimal Semut ke 8 -> 1, 2, 5, 8, 9, 10, 4, 7, 6, 3 = 133

Siklus ke: 7
---------------
Minimal Semut ke 72 -> 1, 2, 5, 9, 10, 8, 4, 6, 3, 7 = 144

Siklus ke: 8
---------------
Minimal Semut ke 83 -> 1, 2, 5, 9, 10, 7, 3, 8, 4, 6 = 136

Siklus ke: 9
---------------
Minimal Semut ke 93 -> 8, 9, 10, 1, 2, 5, 6, 3, 4, 7 = 141

Siklus ke: 10
---------------
Minimal Semut ke 84 -> 1, 2, 5, 9, 10, 4, 7, 3, 8, 6 = 136

Siklus ke: 11
---------------
Minimal Semut ke 38 -> 5, 8, 9, 10, 4, 7, 1, 2, 3, 6 = 142
```

```
Siklus ke: 12
---------------
Minimal Semut ke 51 -> 3, 8, 9, 4, 7, 10, 1, 2, 5, 6 = 140

Siklus ke: 13
---------------
Minimal Semut ke 46 -> 3, 5, 9, 10, 6, 1, 2, 8, 4, 7 = 139

Siklus ke: 14
---------------
Minimal Semut ke 9 -> 1, 2, 5, 8, 9, 10, 6, 4, 7, 3 = 133

Siklus ke: 15
---------------
Minimal Semut ke 38 -> 2, 5, 9, 4, 7, 6, 8, 10, 1, 3 = 147

Siklus ke: 16
---------------
Minimal Semut ke 53 -> 3, 9, 10, 4, 7, 1, 2, 5, 8, 6 = 142

Siklus ke: 17
---------------
Minimal Semut ke 42 -> 6, 8, 7, 1, 2, 5, 9, 10, 4, 3 = 142

Siklus ke: 18
---------------
Minimal Semut ke 27 -> 1, 2, 5, 9, 4, 3, 8, 7, 10, 6 = 141

Siklus ke: 19
---------------
Minimal Semut ke 86 -> 1, 2, 5, 9, 10, 4, 3, 8, 7, 6 = 136

Siklus ke: 20
---------------
Minimal Semut ke 60 -> 1, 2, 5, 9, 6, 8, 10, 4, 7, 3 = 139

Siklus ke: 21
---------------
Minimal Semut ke 1 -> 1, 2, 5, 9, 6, 10, 7, 3, 8, 4 = 141

Siklus ke: 22
---------------
Minimal Semut ke 91 -> 3, 5, 9, 10, 1, 2, 8, 4, 7, 6 = 138

Siklus ke: 23
---------------
Minimal Semut ke 31 -> 6, 1, 2, 5, 9, 10, 7, 3, 8, 4 = 139
```

```
Siklus ke: 24
---------------
Minimal Semut ke 100 -> 4, 7, 6, 3, 5, 9, 10, 1, 2, 8 = 138

Siklus ke: 25
---------------
Minimal Semut ke 28 -> 1, 2, 5, 9, 10, 6, 8, 4, 3, 7 = 139

Siklus ke: 26
---------------
Minimal Semut ke 41 -> 1, 2, 5, 9, 6, 3, 8, 7, 4, 10 = 144

Siklus ke: 27
---------------
Minimal Semut ke 16 -> 2, 5, 9, 10, 6, 4, 7, 1, 3, 8 = 142

Siklus ke: 28
---------------
Minimal Semut ke 59 -> 1, 2, 9, 10, 6, 3, 5, 8, 4, 7 = 136

Siklus ke: 29
---------------
Minimal Semut ke 57 -> 1, 2, 5, 8, 9, 10, 4, 7, 3, 6 = 133

Siklus ke: 30
---------------
Minimal Semut ke 93 -> 1, 2, 5, 9, 10, 6, 3, 8, 4, 7 = 133

Siklus ke: 31
---------------
Minimal Semut ke 96 -> 3, 5, 9, 10, 1, 2, 8, 7, 6, 4 = 141

Siklus ke: 32
---------------
Minimal Semut ke 90 -> 1, 2, 5, 9, 10, 6, 4, 7, 3, 8 = 133

Siklus ke: 33
---------------
Minimal Semut ke 81 -> 3, 8, 9, 10, 6, 1, 2, 5, 4, 7 = 142

Siklus ke: 34
---------------
Minimal Semut ke 17 -> 1, 2, 5, 9, 10, 8, 4, 7, 3, 6 = 141

Siklus ke: 35
---------------
Minimal Semut ke 74 -> 1, 2, 5, 8, 9, 10, 3, 4, 6, 7 = 138
```

```
Siklus ke: 36
---------------
Minimal Semut ke 27 -> 6, 4, 7, 3, 5, 8, 9, 10, 1, 2 = 141

Siklus ke: 37
---------------
Minimal Semut ke 85 -> 1, 5, 8, 9, 10, 4, 7, 2, 3, 6 = 142

Siklus ke: 38
---------------
Minimal Semut ke 44 -> 3, 8, 9, 6, 4, 1, 2, 5, 10, 7 = 145

Siklus ke: 39
---------------
Minimal Semut ke 10 -> 1, 2, 5, 9, 10, 7, 6, 8, 4, 3 = 139

Siklus ke: 40
---------------
Minimal Semut ke 36 -> 1, 2, 5, 8, 9, 10, 6, 3, 4, 7 = 133

Siklus ke: 41
---------------
Minimal Semut ke 71 -> 1, 2, 5, 9, 10, 3, 8, 6, 4, 7 = 138

Siklus ke: 42
---------------
Minimal Semut ke 34 -> 1, 2, 5, 9, 10, 3, 8, 4, 7, 6 = 135

Siklus ke: 43
---------------
Minimal Semut ke 71 -> 1, 2, 5, 9, 10, 4, 7, 6, 3, 8 = 133

Siklus ke: 44
---------------
Minimal Semut ke 88 -> 1, 2, 5, 9, 10, 7, 8, 4, 3, 6 = 142

Siklus ke: 45
---------------
Minimal Semut ke 17 -> 3, 8, 9, 10, 7, 1, 2, 5, 6, 4 = 139

Siklus ke: 46
---------------
Minimal Semut ke 36 -> 1, 2, 5, 9, 4, 7, 8, 6, 3, 10 = 145

Siklus ke: 47
---------------
Minimal Semut ke 25 -> 3, 5, 9, 4, 7, 1, 2, 8, 10, 6 = 142
```

3. Perhitungan Jarak *Material Handling Layout* Algoritma *Tabu Search*

No	Nama Komponen	Routing	Material Handling	Jarak (Meter)
1	*frame panjang*	2-1-6-8-5-9	7.95+14.56+10.62+10.89+12.59	56.61
2	*frame pendek*	2-1-6-8-5-9	7.95+14.56+10.62+10.89+12.59	56.61
3	*frame tengah*	2-1-6-8-5-9	7.95+14.56+10.62+10.89+12.59	56.61
4	*sunduk panjang*	2-1-3-8-5-9	7.95+8.78+6.83+10.89+12.59	47.04
5	*sunduk pendek*	2-1-3-8-5-9	7.95+8.78+6.83+10.89+12.59	47.04
6	*sunduk tengah*	2-1-3-8-5-9	7.95+8.78+6.83+10.89+12.59	47.04
7	*kloss*	2-1-10-3-8-5-9	7.95+5.67+6.59+6.83+10.89+12.59	50.52
8	*frame atas*	2-1-3-8-5-9	7.95+8.78+6.83+10.89+12.59	47.04
9	*ambal bawah*	2-1-4-7-5-9	7.95+23.63+16.90+10.29+12.59	71.37
10	*ambal*	2-1-4-7-5-9	7.95+23.63+16.90+10.29+12.59	71.37
11	*list ambal*	2-1-4-7-5-9	7.95+23.63+16.90+10.29+12.59	71.37
12	*kaki meja*	2-1-10-3-8-5-9	7.95+5.67+6.59+6.83+10.89+12.59	50.52

Siklus ke: 48

Minimal Semut ke 13 -> 1, 2, 5, 9, 10, 8, 4, 3, 7, 6 = 144

Siklus ke: 49

Minimal Semut ke 98 -> 3, 5, 9, 10, 1, 2, 8, 4, 7, 6 = 138

Siklus ke: 50

Minimal Semut ke 81 -> 1, 2, 5, 8, 9, 6, 7, 10, 4, 3 = 141

Siklus ke: 51

Minimal Semut ke 71 -> 3, 8, 9, 6, 1, 2, 5, 10, 4, 7 = 142

Siklus ke: 52

Minimal Semut ke 24 -> 1, 2, 5, 9, 10, 4, 7, 3, 8, 6 = 136

Siklus ke: 53

Minimal Semut ke 73 -> 3, 8, 9, 10, 1, 2, 5, 7, 6, 4 = 138

Siklus ke: 54

Minimal Semut ke 99 -> 5, 9, 10, 1, 2, 8, 4, 7, 3, 6 = 144

Siklus ke: 55

Minimal Semut ke 14 -> 1, 2, 5, 9, 7, 6, 8, 10, 4, 3 = 142

Siklus ke: 56

Minimal Semut ke 52 -> 1, 2, 5, 7, 9, 10, 4, 3, 8, 6 = 145

Siklus ke: 57

Minimal Semut ke 93 -> 6, 4, 1, 2, 5, 8, 9, 10, 7, 3 = 139

Siklus ke: 58

Minimal Semut ke 9 -> 1, 2, 5, 9, 6, 8, 10, 3, 4, 7 = 141

Siklus ke: 59

Minimal Semut ke 35 -> 1, 2, 5, 9, 10, 6, 4, 7, 3, 8 = 133

Siklus ke: 60

Minimal Semut ke 75 -> 3, 5, 9, 10, 1, 2, 8, 4, 7, 6 = 138

Siklus ke: 61

Minimal Semut ke 84 -> 1, 2, 9, 10, 3, 5, 8, 4, 7, 6 = 138

Siklus ke: 62

Minimal Semut ke 34 -> 1, 2, 5, 9, 4, 7, 3, 8, 10, 6 = 136

Siklus ke: 63

Minimal Semut ke 79 -> 3, 4, 7, 1, 2, 5, 9, 10, 6, 8 = 139

Siklus ke: 64

Minimal Semut ke 100 -> 1, 2, 5, 9, 6, 3, 10, 4, 7, 8 = 142

Siklus ke: 65

Minimal Semut ke 95 -> 6, 10, 1, 2, 5, 9, 3, 8, 4, 7 = 143

Siklus ke: 66

Minimal Semut ke 6 -> 7, 3, 9, 10, 1, 2, 5, 8, 4, 6 = 141

Siklus ke: 67

Minimal Semut ke 6 -> 7, 3, 5, 9, 10, 1, 2, 8, 4, 6 = 141

Siklus ke: 68

Minimal Semut ke 46 -> 2, 5, 9, 10, 1, 3, 8, 4, 7, 6 = 141

Siklus ke: 69

Minimal Semut ke 91 -> 6, 8, 9, 10, 1, 2, 5, 7, 3, 4 = 141

Siklus ke: 70

Minimal Semut ke 60 -> 1, 2, 5, 9, 4, 6, 8, 10, 7, 3 = 142

Siklus ke: 71

Minimal Semut ke 28 -> 6, 1, 2, 5, 9, 10, 3, 8, 4, 7 = 138

Siklus ke: 72

Minimal Semut ke 88 -> 1, 2, 5, 8, 9, 10, 7, 6, 4, 3 = 136

Siklus ke: 73

Minimal Semut ke 59 -> 3, 5, 9, 10, 1, 2, 8, 4, 7, 6 = 138

Siklus ke: 74

Minimal Semut ke 26 -> 1, 2, 5, 8, 9, 10, 3, 7, 6, 4 = 138

Siklus ke: 75

Minimal Semut ke 18 -> 1, 2, 5, 8, 9, 10, 6, 3, 7, 4 = 139

Siklus ke: 76

Minimal Semut ke 26 -> 4, 6, 8, 9, 10, 1, 2, 5, 7, 3 = 141

Siklus ke: 77

Minimal Semut ke 55 -> 1, 2, 8, 9, 10, 7, 3, 5, 6, 4 = 139

Siklus ke: 78

Minimal Semut ke 52 -> 1, 2, 5, 9, 10, 4, 3, 6, 7, 8 = 142

Siklus ke: 79

Minimal Semut ke 96 -> 1, 2, 9, 7, 3, 5, 8, 10, 4, 6 = 142

Siklus ke: 80

Minimal Semut ke 4 -> 1, 2, 5, 8, 9, 10, 4, 7, 6, 3 = 133

Siklus ke: 81

Minimal Semut ke 90 -> 8, 9, 10, 1, 2, 5, 6, 3, 4, 7 = 141

Siklus ke: 82

Minimal Semut ke 93 -> 1, 2, 9, 10, 7, 3, 5, 8, 4, 6 = 139

Siklus ke: 83

Minimal Semut ke 79 -> 3, 8, 9, 10, 1, 2, 5, 7, 6, 4 = 138

Siklus ke: 84

Minimal Semut ke 52 -> 3, 5, 9, 10, 7, 1, 2, 8, 4, 6 = 142

Siklus ke: 85

Minimal Semut ke 83 -> 1, 2, 5, 9, 10, 4, 6, 3, 8, 7 = 136

Siklus ke: 86

Minimal Semut ke 91 -> 1, 2, 5, 8, 9, 10, 4, 7, 3, 6 = 133

Siklus ke: 87

Minimal Semut ke 84 -> 2, 8, 9, 10, 1, 5, 6, 3, 4, 7 = 144

Siklus ke: 88

Minimal Semut ke 94 -> 1, 2, 5, 8, 9, 4, 10, 6, 7, 3 = 141

Siklus ke: 89

Minimal Semut ke 88 -> 6, 4, 7, 3, 5, 8, 9, 10, 1, 2 = 141

Siklus ke: 90

Minimal Semut ke 62 -> 1, 2, 3, 5, 9, 4, 6, 8, 10, 7 = 142

Siklus ke: 91

Minimal Semut ke 26 -> 6, 3, 8, 9, 10, 1, 2, 5, 4, 7 = 141

Siklus ke: 92

Minimal Semut ke 16 -> 7, 9, 10, 1, 2, 3, 5, 8, 4, 6 = 144

Siklus ke: 93

Minimal Semut ke 74 -> 7, 6, 4, 3, 5, 9, 10, 1, 2, 8 = 141

Siklus ke: 94

Minimal Semut ke 19 -> 1, 2, 5, 9, 10, 4, 7, 3, 8, 6 = 136

Siklus ke: 95

Minimal Semut ke 39 -> 3, 5, 9, 6, 8, 10, 1, 2, 4, 7 = 144

```
Siklus ke: 96
---------------
Minimal Semut ke 93 -> 1, 2, 5, 8, 9, 10, 4, 7, 6, 3 = 133

Siklus ke: 97
---------------
Minimal Semut ke 27 -> 4, 7, 6, 3, 9, 10, 1, 2, 5, 8 = 138

Siklus ke: 98
---------------
Minimal Semut ke 16 -> 1, 2, 5, 9, 10, 6, 7, 3, 8, 4 = 136

Siklus ke: 99
---------------
Minimal Semut ke 49 -> 7, 3, 5, 8, 9, 10, 4, 6, 1, 2 = 145

Siklus ke: 100
---------------
Minimal Semut ke 53 -> 4, 5, 9, 10, 1, 2, 3, 8, 7, 6 = 141

****************
Hasil Akhir Mesin:
Siklus ke 6 -> 4, 9, 5, 7, 3, 6, 2, 8, 10, 1 = 133
```

2. Pencarian Urutan Part

```
Siklus ke: 1
---------------
Minimal Semut ke 7 -> 1, 5, 8, 11, 2, 6, 9, 10, 3, 4, 12, 7 = 23

Siklus ke: 2
---------------
Minimal Semut ke 5 -> 1, 5, 8, 10, 2, 4, 9, 11, 3, 6, 12, 7 = 23

Siklus ke: 3
---------------
Minimal Semut ke 19 -> 11, 2, 6, 8, 12, 3, 5, 9, 10, 1, 4, 7 = 23

Siklus ke: 4
---------------
Minimal Semut ke 8 -> 11, 2, 3, 6, 8, 12, 1, 5, 9, 10, 4, 7 = 21

Siklus ke: 5
---------------
Minimal Semut ke 1 -> 11, 2, 5, 8, 12, 3, 6, 9, 10, 1, 4, 7 = 23
```

```
Siklus ke: 6
---------------
Minimal Semut ke 21 -> 11, 2, 5, 8, 12, 3, 6, 9, 10, 1, 4, 7 = 23

Siklus ke: 7
---------------
Minimal Semut ke 3 -> 11, 2, 6, 8, 12, 3, 5, 9, 10, 1, 4, 7 = 23
Siklus ke: 8
---------------
Minimal Semut ke 17 -> 1, 4, 8, 10, 2, 5, 9, 11, 3, 6, 12, 7 = 23

Siklus ke: 9
---------------
Minimal Semut ke 22 -> 11, 2, 5, 8, 12, 3, 6, 9, 10, 1, 4, 7 = 23

Siklus ke: 10
---------------
Minimal Semut ke 2 -> 11, 2, 6, 8, 12, 3, 5, 9, 10, 1, 4, 7 = 23

Siklus ke: 11
---------------
Minimal Semut ke 11 -> 11, 2, 6, 8, 12, 1, 5, 9, 10, 3, 4, 7 = 23

Siklus ke: 12
---------------
Minimal Semut ke 18 -> 11, 2, 6, 8, 12, 3, 5, 9, 10, 1, 4, 7 = 23

Siklus ke: 13
---------------
Minimal Semut ke 10 -> 1, 5, 8, 11, 2, 6, 9, 10, 3, 4, 12, 7 = 23

Siklus ke: 14
---------------
Minimal Semut ke 11 -> 1, 5, 8, 11, 2, 4, 9, 10, 3, 6, 12, 7 = 23

Siklus ke: 15
---------------
Minimal Semut ke 14 -> 11, 2, 6, 8, 12, 3, 5, 9, 10, 1, 4, 7 = 23

Siklus ke: 16
---------------
Minimal Semut ke 29 -> 11, 2, 6, 8, 12, 1, 5, 9, 10, 3, 4, 7 = 23

Siklus ke: 17
---------------
Minimal Semut ke 9 -> 11, 2, 5, 8, 12, 1, 6, 9, 10, 3, 4, 7 = 23
```

```
Siklus ke: 18
---------------
Minimal Semut ke 2 -> 11, 2, 5, 8, 12, 3, 6, 9, 10, 1, 4, 7 = 23

Siklus ke: 19
---------------
Minimal Semut ke 7 -> 1, 5, 8, 10, 2, 4, 9, 11, 3, 6, 12, 7 = 23

Siklus ke: 20
---------------
Minimal Semut ke 5 -> 1, 4, 8, 10, 2, 5, 9, 11, 3, 6, 7, 12 = 23

Siklus ke: 21
---------------
Minimal Semut ke 12 -> 11, 2, 5, 8, 12, 1, 6, 9, 10, 3, 4, 7 = 23

Siklus ke: 22
---------------
Minimal Semut ke 15 -> 11, 2, 6, 8, 12, 3, 5, 9, 10, 1, 4, 7 = 23

Siklus ke: 23
---------------
Minimal Semut ke 15 -> 1, 5, 8, 11, 2, 6, 9, 10, 3, 4, 12, 7 = 23

Siklus ke: 24
---------------
Minimal Semut ke 1 -> 11, 2, 6, 8, 12, 3, 5, 9, 10, 1, 4, 7 = 23

Siklus ke: 25
---------------
Minimal Semut ke 26 -> 11, 2, 6, 8, 12, 1, 5, 9, 10, 3, 4, 7 = 23

Siklus ke: 26
---------------
Minimal Semut ke 24 -> 11, 2, 5, 8, 12, 1, 6, 9, 10, 3, 4, 7 = 23

Siklus ke: 27
---------------
Minimal Semut ke 10 -> 1, 4, 8, 10, 2, 5, 9, 11, 3, 6, 12, 7 = 23

Siklus ke: 28
---------------
Minimal Semut ke 15 -> 11, 2, 6, 8, 12, 3, 5, 9, 10, 1, 4, 7 = 23

Siklus ke: 29
---------------
Minimal Semut ke 3 -> 11, 2, 5, 8, 12, 3, 6, 9, 10, 1, 4, 7 = 23
```

```
Siklus ke: 30
---------------
Minimal Semut ke 1 -> 11, 2, 6, 8, 12, 3, 5, 9, 10, 1, 4, 7 = 23

Siklus ke: 31
---------------
Minimal Semut ke 10 -> 1, 5, 8, 11, 2, 6, 9, 10, 3, 4, 12, 7 = 23

Siklus ke: 32
---------------
Minimal Semut ke 9 -> 11, 2, 5, 8, 12, 1, 6, 9, 10, 3, 4, 7 = 23

Siklus ke: 33
---------------
Minimal Semut ke 23 -> 1, 5, 8, 11, 2, 4, 9, 10, 3, 6, 12, 7 = 23

Siklus ke: 34
---------------
Minimal Semut ke 22 -> 11, 2, 5, 8, 12, 3, 6, 9, 10, 1, 4, 7 = 23

Siklus ke: 35
---------------
Minimal Semut ke 2 -> 11, 2, 5, 8, 12, 1, 6, 9, 10, 3, 4, 7 = 23

Siklus ke: 36
---------------
Minimal Semut ke 51 -> 5, 8, 12, 4, 9, 11, 2, 6, 7, 10, 1, 3 = 22

Siklus ke: 37
---------------
Minimal Semut ke 17 -> 1, 4, 8, 10, 2, 5, 9, 11, 3, 6, 12, 7 = 23

Siklus ke: 38
---------------
Minimal Semut ke 10 -> 11, 2, 6, 8, 12, 1, 5, 9, 10, 3, 4, 7 = 23

Siklus ke: 39
---------------
Minimal Semut ke 12 -> 11, 2, 6, 8, 12, 3, 5, 9, 10, 1, 4, 7 = 23

Siklus ke: 40
---------------
Minimal Semut ke 37 -> 1, 4, 8, 10, 2, 5, 9, 11, 3, 6, 12, 7 = 23

Siklus ke: 41
---------------
Minimal Semut ke 31 -> 11, 2, 6, 8, 12, 3, 5, 9, 10, 1, 4, 7 = 23
```

```
Siklus ke: 42
---------------
Minimal Semut ke 9 -> 11, 2, 5, 8, 12, 3, 6, 9, 10, 1, 4, 7 = 23

Siklus ke: 43
---------------
Minimal Semut ke 7 -> 1, 4, 8, 10, 2, 5, 9, 11, 3, 6, 12, 7 = 23

Siklus ke: 44
---------------
Minimal Semut ke 12 -> 1, 5, 8, 11, 2, 6, 9, 10, 3, 4, 12, 7 = 23

Siklus ke: 45
---------------
Minimal Semut ke 2 -> 11, 2, 6, 8, 12, 1, 5, 9, 10, 3, 4, 7 = 23

Siklus ke: 46
---------------
Minimal Semut ke 4 -> 11, 2, 6, 8, 12, 3, 5, 9, 10, 1, 4, 7 = 23

Siklus ke: 47
---------------
Minimal Semut ke 4 -> 11, 2, 5, 8, 12, 3, 6, 9, 10, 1, 4, 7 = 23

Siklus ke: 48
---------------
Minimal Semut ke 10 -> 11, 2, 5, 8, 12, 1, 6, 9, 10, 3, 4, 7 = 23

Siklus ke: 49
---------------
Minimal Semut ke 41 -> 12, 4, 7, 10, 2, 5, 9, 11, 3, 6, 8, 1 = 25

Siklus ke: 50
---------------
Minimal Semut ke 46 -> 9, 1, 4, 8, 12, 3, 6, 10, 11, 2, 7, 5 = 24

Siklus ke: 51
---------------
Minimal Semut ke 5 -> 8, 12, 3, 6, 10, 1, 5, 9, 11, 2, 4, 7 = 27

Siklus ke: 52
---------------
Minimal Semut ke 3 -> 12, 4, 7, 10, 2, 5, 9, 11, 3, 6, 8, 1 = 25

Siklus ke: 53
---------------
Minimal Semut ke 15 -> 1, 5, 9, 11, 3, 7, 10, 2, 6, 8, 12, 4 = 26
```

```
Siklus ke: 54
---------------
Minimal Semut ke 80 -> 1, 5, 9, 11, 3, 7, 10, 2, 6, 8, 12, 4 = 26

Siklus ke: 55
---------------
Minimal Semut ke 95 -> 1, 5, 9, 11, 3, 7, 10, 2, 6, 8, 12, 4 = 26

Siklus ke: 56
---------------
Minimal Semut ke 100 -> 1, 5, 9, 11, 3, 7, 10, 2, 6, 8, 12, 4 = 26

Siklus ke: 57
---------------
Minimal Semut ke 51 -> 12, 4, 7, 10, 1, 5, 9, 11, 3, 6, 8, 2 = 25

Siklus ke: 58
---------------
Minimal Semut ke 25 -> 1, 5, 9, 11, 3, 7, 10, 2, 6, 8, 12, 4 = 26

Siklus ke: 59
---------------
Minimal Semut ke 44 -> 12, 4, 7, 10, 1, 5, 9, 11, 2, 6, 8, 3 = 25

Siklus ke: 60
---------------
Minimal Semut ke 53 -> 12, 4, 7, 10, 1, 5, 9, 11, 2, 6, 8, 3 = 25

Siklus ke: 61
---------------
Minimal Semut ke 56 -> 3, 6, 10, 1, 5, 9, 11, 2, 7, 12, 8, 4 = 24

Siklus ke: 62
---------------
Minimal Semut ke 24 -> 12, 4, 7, 10, 1, 5, 9, 11, 3, 6, 8, 2 = 25

Siklus ke: 63
---------------
Minimal Semut ke 1 -> 7, 12, 3, 6, 10, 1, 5, 9, 11, 2, 4, 8 = 25

Siklus ke: 64
---------------
Minimal Semut ke 76 -> 12, 4, 7, 10, 1, 5, 9, 11, 3, 6, 8, 2 = 25

Siklus ke: 65
---------------
Minimal Semut ke 19 -> 12, 3, 7, 10, 1, 5, 9, 11, 4, 6, 8, 2 = 27
```

```
Siklus ke: 66
---------------
Minimal Semut ke 2 -> 1, 5, 8, 11, 2, 6, 9, 10, 3, 4, 12, 7 = 23

Siklus ke: 67
---------------
Minimal Semut ke 10 -> 11, 2, 5, 8, 12, 1, 6, 9, 10, 3, 4, 7 = 23

Siklus ke: 68
---------------
Minimal Semut ke 11 -> 1, 4, 8, 10, 2, 5, 9, 11, 3, 6, 12, 7 = 23

Siklus ke: 69
---------------
Minimal Semut ke 14 -> 11, 2, 5, 8, 12, 1, 6, 9, 10, 3, 4, 7 = 23

Siklus ke: 70
---------------
Minimal Semut ke 18 -> 11, 2, 5, 8, 12, 1, 6, 9, 10, 3, 4, 7 = 23

Siklus ke: 71
---------------
Minimal Semut ke 13 -> 11, 2, 6, 8, 12, 3, 5, 9, 10, 1, 4, 7 = 23

Siklus ke: 72
---------------
Minimal Semut ke 33 -> 1, 4, 8, 10, 2, 5, 9, 11, 3, 6, 12, 7 = 23

Siklus ke: 73
---------------
Minimal Semut ke 23 -> 1, 4, 8, 10, 2, 5, 9, 11, 3, 6, 12, 7 = 23

Siklus ke: 74
---------------
Minimal Semut ke 41 -> 11, 2, 6, 8, 12, 1, 5, 9, 10, 3, 4, 7 = 23

Siklus ke: 75
---------------
Minimal Semut ke 8 -> 11, 2, 5, 8, 12, 3, 6, 9, 10, 1, 4, 7 = 23

Siklus ke: 76
---------------
Minimal Semut ke 31 -> 8, 12, 4, 7, 10, 1, 3, 9, 11, 2, 5, 6 = 22

Siklus ke: 77
---------------
Minimal Semut ke 1 -> 11, 2, 5, 8, 12, 1, 6, 9, 10, 3, 4, 7 = 23
```

s ke: 90
---
al Semut ke

s ke: 91
---
al Semut ke

s ke: 92
---
al Semut ke

s ke: 93
---
al Semut ke

s ke: 94
---
al Semut ke

s ke: 95
---
al Semut ke

s ke: 96
---
al Semut ke

s ke: 97
---
al Semut ke

s ke: 98
---
al Semut ke

s ke: 99
---
al Semut ke

s ke: 100
---
al Semut k
****
Akhir Par
s ke 4 ->

Siklus ke: 78

Minimal Semut ke 34 -> 11, 2, 6, 8, 12, 3, 5, 9, 10, 1, 4, 7 = 23

Siklus ke: 79

Minimal Semut ke 7 -> 1, 4, 8, 10, 2, 5, 9, 11, 3, 6, 12, 7 = 23

Siklus ke: 80

Minimal Semut ke 2 -> 11, 2, 5, 8, 12, 3, 6, 9, 10, 1, 4, 7 = 23

Siklus ke: 81

Minimal Semut ke 1 -> 11, 2, 6, 8, 12, 1, 5, 9, 10, 3, 4, 7 = 23

Siklus ke: 82

Minimal Semut ke 13 -> 11, 2, 5, 8, 12, 3, 6, 9, 10, 1, 4, 7 = 23

Siklus ke: 83

Minimal Semut ke 26 -> 1, 4, 8, 10, 2, 5, 9, 11, 3, 6, 12, 7 = 23

Siklus ke: 84

Minimal Semut ke 78 -> 9, 1, 5, 8, 12, 3, 6, 10, 11, 2, 7, 4 = 24

Siklus ke: 85

Minimal Semut ke 12 -> 12, 3, 7, 10, 1, 5, 9, 11, 4, 6, 8, 2 = 27

Siklus ke: 86

Minimal Semut ke 92 -> 12, 4, 7, 10, 1, 5, 9, 11, 3, 6, 8, 2 = 25

Siklus ke: 87

Minimal Semut ke 18 -> 12, 3, 7, 10, 1, 5, 9, 11, 2, 6, 8, 4 = 27

Siklus ke: 88

Minimal Semut ke 9 -> 1, 4, 8, 10, 2, 5, 9, 11, 3, 6, 12, 7 = 23

Siklus ke: 89

Minimal Semut ke 29 -> 1, 4, 8, 10, 2, 5, 9, 11, 3, 6, 12, 7 = 23

```
Siklus ke: 90
---------------
Minimal Semut ke 5 -> 11, 2, 5, 8, 12, 1, 6, 9, 10, 3, 4, 7 = 23

Siklus ke: 91
---------------
Minimal Semut ke 9 -> 11, 2, 6, 8, 12, 3, 5, 9, 10, 1, 4, 7 = 23

Siklus ke: 92
---------------
Minimal Semut ke 7 -> 11, 2, 6, 8, 12, 3, 5, 9, 10, 1, 4, 7 = 23

Siklus ke: 93
---------------
Minimal Semut ke 12 -> 1, 5, 8, 11, 2, 6, 9, 10, 3, 4, 12, 7 = 23

Siklus ke: 94
---------------
Minimal Semut ke 13 -> 1, 4, 8, 10, 2, 5, 9, 11, 3, 6, 12, 7 = 23

Siklus ke: 95
---------------
Minimal Semut ke 5 -> 1, 5, 8, 11, 2, 6, 9, 10, 3, 4, 12, 7 = 23

Siklus ke: 96
---------------
Minimal Semut ke 21 -> 11, 2, 5, 8, 12, 1, 6, 9, 10, 3, 4, 7 = 23

Siklus ke: 97
---------------
Minimal Semut ke 11 -> 11, 2, 6, 8, 12, 3, 5, 9, 10, 1, 4, 7 = 23

Siklus ke: 98
---------------
Minimal Semut ke 1 -> 11, 3, 6, 8, 12, 2, 5, 9, 10, 1, 4, 7 = 23

Siklus ke: 99
---------------
Minimal Semut ke 12 -> 11, 2, 6, 8, 12, 3, 5, 9, 10, 1, 4, 7 = 23

Siklus ke: 100
---------------
Minimal Semut ke 13 -> 1, 5, 8, 11, 2, 6, 9, 10, 3, 4, 12, 7 = 23
****************
Hasil Akhir Part:
Siklus ke 4 -> 11, 2, 3, 6, 8, 12, 1, 5, 9, 10, 4, 7 = 21
```

```
==========================================
 HASIL AKHIR:
 Mesin: 4, 9, 5, 7, 3, 6, 2, 8, 10, 1 = 133
 Part: 11, 2, 3, 6, 8, 12, 1, 5, 9, 10, 4, 7 = 21
==========================================
```

ALGORITMA TABU SEARCH

1. Pencarian Urutan Mesin

```
Iterasi ke: 1
Global Optimum:
1, 2, 3, 4, 5, 6, 7, 8, 9, 10: 145
Local Optimum:
1, 2, 3, 4, 5, 6, 7, 8, 9, 10: 145

Iterasi ke: 2
Global Optimum:
1, 2, 3, 4, 7, 6, 5, 8, 9, 10: 142
Local Optimum:
1, 2, 3, 4, 7, 6, 5, 8, 9, 10: 142

Iterasi ke: 3
Global Optimum:
1, 2, 6, 4, 7, 3, 5, 8, 9, 10: 136
Local Optimum:
1, 2, 6, 4, 7, 3, 5, 8, 9, 10: 136

Iterasi ke: 4
Global Optimum:
1, 2, 6, 4, 7, 3, 5, 8, 9, 10: 136
Local Optimum:
1, 2, 4, 6, 7, 3, 5, 8, 9, 10: 139

Iterasi ke: 5
Global Optimum:
1, 2, 6, 4, 7, 3, 5, 8, 9, 10: 136
Local Optimum:
1, 2, 4, 7, 6, 3, 5, 8, 9, 10: 136
```

```
Iterasi ke: 6
Global Optimum:
1, 2, 6, 4, 7, 3, 5, 8, 9, 10: 136
Local Optimum:
1, 2, 3, 7, 6, 4, 5, 8, 9, 10: 139

Iterasi ke: 7
Global Optimum:
1, 2, 6, 4, 7, 3, 5, 8, 9, 10: 136
Local Optimum:
1, 2, 5, 7, 6, 4, 3, 8, 9, 10: 136

Iterasi ke: 8
Global Optimum:
1, 2, 6, 4, 7, 3, 5, 8, 9, 10: 136
Local Optimum:
1, 2, 8, 7, 6, 4, 3, 5, 9, 10: 139

Iterasi ke: 9
Global Optimum:
1, 2, 6, 4, 7, 3, 5, 8, 9, 10: 136
Local Optimum:
1, 2, 8, 4, 6, 7, 3, 5, 9, 10: 139

Iterasi ke: 10
Global Optimum:
1, 2, 6, 4, 7, 3, 5, 8, 9, 10: 136
Local Optimum:
1, 2, 8, 4, 7, 6, 3, 5, 9, 10: 136

Iterasi ke: 11
Global Optimum:
1, 2, 6, 4, 7, 3, 5, 8, 9, 10: 136
Local Optimum:
1, 2, 9, 4, 7, 6, 3, 5, 8, 10: 139

Iterasi ke: 12
Global Optimum:
1, 2, 6, 4, 7, 3, 5, 8, 9, 10: 136
Local Optimum:
1, 2, 10, 4, 7, 6, 3, 5, 8, 9: 139

Iterasi ke: 13
Global Optimum:
1, 2, 6, 4, 7, 3, 5, 8, 9, 10: 136
Local Optimum:
1, 2, 6, 4, 7, 10, 3, 5, 8, 9: 140
```

Iterasi ke: 14
Global Optimum:
1, 2, 6, 4, 7, 3, 5, 8, 9, 10: 136
Local Optimum:
1, 2, 6, 4, 10, 7, 3, 5, 8, 9: 141

Iterasi ke: 15
Global Optimum:
1, 2, 6, 4, 7, 3, 5, 8, 9, 10: 136
Local Optimum:
1, 2, 6, 10, 4, 7, 3, 5, 8, 9: 138

Iterasi ke: 16
Global Optimum:
1, 2, 6, 4, 7, 3, 5, 8, 9, 10: 136
Local Optimum:
1, 2, 10, 6, 4, 7, 3, 5, 8, 9: 139

Iterasi ke: 17
Global Optimum:
1, 2, 6, 4, 7, 3, 5, 8, 9, 10: 136
Local Optimum:
1, 2, 9, 6, 4, 7, 3, 5, 8, 10: 139

Iterasi ke: 18
Global Optimum:
1, 2, 6, 4, 7, 3, 5, 8, 9, 10: 136
Local Optimum:
1, 2, 9, 6, 10, 7, 3, 5, 8, 4: 144

Iterasi ke: 19
Global Optimum:
1, 2, 6, 4, 7, 3, 5, 8, 9, 10: 136
Local Optimum:
1, 2, 9, 10, 6, 7, 3, 5, 8, 4: 139

Iterasi ke: 20
Global Optimum:
1, 2, 6, 4, 7, 3, 5, 8, 9, 10: 136
Local Optimum:
1, 2, 9, 10, 6, 4, 3, 5, 8, 7: 139

Iterasi ke: 21
Global Optimum:
1, 2, 6, 4, 7, 3, 5, 8, 9, 10: 136
Local Optimum:
1, 2, 9, 10, 4, 6, 3, 5, 8, 7: 139

Iterasi ke: 22
Global Optimum:
1, 2, 6, 4, 7, 3, 5, 8, 9, 10: 136
Local Optimum:
1, 2, 8, 10, 4, 6, 3, 5, 9, 7: 142

Iterasi ke: 23
Global Optimum:
1, 2, 6, 4, 7, 3, 5, 8, 9, 10: 136
Local Optimum:
1, 2, 8, 10, 4, 7, 3, 5, 9, 6: 139

Iterasi ke: 24
Global Optimum:
1, 2, 6, 4, 7, 3, 5, 8, 9, 10: 136
Local Optimum:
1, 2, 5, 10, 4, 7, 3, 8, 9, 6: 139

Iterasi ke: 25
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133

Iterasi ke: 26
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 8, 4, 7, 3, 6, 9, 10: 139

Iterasi ke: 27
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 8, 4, 7, 6, 3, 9, 10: 136

Iterasi ke: 28
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 9, 4, 7, 6, 3, 8, 10: 136

Iterasi ke: 29
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 10, 4, 7, 6, 3, 8, 9: 136

Iterasi ke: 30
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 8, 10, 4, 7, 6, 3, 5, 9: 136

Iterasi ke: 31
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 9, 10, 4, 7, 6, 3, 5, 8: 136

Iterasi ke: 32
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 9, 10, 4, 6, 7, 3, 5, 8: 139

Iterasi ke: 33
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 9, 10, 6, 4, 7, 3, 5, 8: 136

Iterasi ke: 34
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 8, 10, 6, 4, 7, 3, 5, 9: 136

Iterasi ke: 35
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 10, 6, 4, 7, 3, 8, 9: 136

Iterasi ke: 36
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 6, 10, 4, 7, 3, 8, 9: 135

Iterasi ke: 37
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 9, 10, 4, 7, 3, 8, 6: 136

Iterasi ke: 38
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 8, 9, 10, 4, 7, 3, 5, 6: 136

Iterasi ke: 39
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 8, 9, 10, 4, 6, 3, 5, 7: 139

Iterasi ke: 40
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 8, 9, 10, 6, 4, 3, 5, 7: 139

Iterasi ke: 41
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 9, 10, 6, 4, 3, 8, 7: 136

Iterasi ke: 42
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 7, 10, 6, 4, 3, 8, 9: 138

Iterasi ke: 43
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 7, 6, 10, 4, 3, 8, 9: 138

Iterasi ke: 44
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 6, 7, 10, 4, 3, 8, 9: 138

Iterasi ke: 45
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 6, 3, 10, 4, 7, 8, 9: 139

Iterasi ke: 46
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 6, 8, 10, 4, 7, 3, 9: 139

Iterasi ke: 47
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 6, 9, 10, 4, 7, 3, 8: 139

Iterasi ke: 48
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 8, 9, 10, 4, 7, 3, 6: 133

Iterasi ke: 49
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 8, 9, 10, 4, 7, 6, 3: 133

Iterasi ke: 50
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 8, 9, 10, 4, 6, 7, 3: 136

Iterasi ke: 51
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 8, 9, 10, 6, 4, 7, 3: 133

Iterasi ke: 52
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 8, 9, 10, 3, 4, 7, 6: 135

Iterasi ke: 53
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 8, 9, 6, 3, 4, 7, 10: 138

Iterasi ke: 54
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 8, 10, 6, 3, 4, 7, 9: 139

Iterasi ke: 55
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 9, 10, 6, 3, 4, 7, 8: 139

Iterasi ke: 56
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 9, 10, 6, 8, 4, 7, 3: 136

Iterasi ke: 57
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 9, 10, 3, 8, 4, 7, 6: 135

Iterasi ke: 58
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 9, 6, 3, 8, 4, 7, 10: 138

Iterasi ke: 59
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 9, 7, 3, 8, 4, 6, 10: 141

Iterasi ke: 60
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 9, 7, 3, 8, 10, 6, 4: 139

Iterasi ke: 61
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 9, 10, 3, 8, 7, 6, 4: 138

Iterasi ke: 62
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 9, 10, 3, 4, 7, 6, 8: 138

Iterasi ke: 63
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 9, 10, 6, 4, 7, 3, 8: 133

Iterasi ke: 64
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 8, 10, 6, 4, 7, 3, 9: 136

Iterasi ke: 65
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 8, 10, 4, 6, 7, 3, 9: 139

Iterasi ke: 66
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 8, 10, 4, 7, 6, 3, 9: 136

Iterasi ke: 67
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 9, 10, 4, 7, 6, 3, 8: 133

Iterasi ke: 68
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 8, 9, 10, 4, 7, 6, 3, 5: 139

Iterasi ke: 69
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 8, 9, 10, 4, 6, 7, 3, 5: 142

Iterasi ke: 70
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 8, 9, 10, 7, 6, 4, 3, 5: 142

Iterasi ke: 71
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 9, 10, 7, 6, 4, 3, 8: 136

Iterasi ke: 72
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 9, 4, 7, 6, 10, 3, 8: 140

Iterasi ke: 73
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 9, 4, 7, 10, 6, 3, 8: 138

Iterasi ke: 74
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 8, 4, 7, 10, 6, 3, 9: 138

Iterasi ke: 75
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 8, 9, 7, 10, 6, 3, 4: 141

Iterasi ke: 76
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 8, 9, 10, 7, 6, 3, 4: 136

Iterasi ke: 77
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 8, 9, 10, 7, 3, 6, 4: 136

Iterasi ke: 78
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 8, 9, 10, 7, 3, 4, 6: 136

Iterasi ke: 79
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 8, 9, 10, 6, 3, 4, 7: 133

Iterasi ke: 80
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 8, 9, 10, 3, 6, 4, 7: 135

Iterasi ke: 81
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 8, 9, 6, 3, 10, 4, 7: 136

Iterasi ke: 82
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 8, 10, 6, 3, 9, 4, 7: 139

Iterasi ke: 83
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 9, 10, 6, 3, 8, 4, 7: 133

Iterasi ke: 84
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 9, 10, 7, 3, 8, 4, 6: 136

Iterasi ke: 85
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 9, 10, 4, 3, 8, 7, 6: 136

Iterasi
Global (
1, 2, 5,
Local Op
1, 2, 3,

Iterasi
Global O
1, 2, 5,
Local Op
1, 2, 4,

Iterasi
Global Op
1, 2, 5,
Local Opt
1, 3, 4,

Iterasi k
Global Op
1, 2, 5,
Local Opt
7, 3, 4,

Iterasi k
Global Opt
1, 2, 5, (
Local Opti
4, 3, 7, 1

Iterasi ke
Global Opt
1, 2, 5, 6
Local Optii
3, 4, 7, 1

Iterasi ke
Global Opt
1, 2, 5, 6,
Local Optir
6, 4, 7, 1,

Hasil akhi
1, 2, 5, 6,

Iterasi ke: 86
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 8, 9, 10, 4, 3, 5, 7, 6: 139

Iterasi ke: 87
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 8, 9, 10, 4, 3, 5, 6, 7: 139

Iterasi ke: 88
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 8, 9, 10, 7, 3, 5, 6, 4: 139

Iterasi ke: 89
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 8, 9, 4, 7, 3, 5, 6, 10: 141

Iterasi ke: 90
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 8, 9, 4, 6, 3, 5, 7, 10: 144

Iterasi ke: 91
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 9, 4, 6, 3, 8, 7, 10: 141

Iterasi ke: 92
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 9, 4, 3, 6, 8, 7, 10: 144

Iterasi ke: 93
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 5, 7, 4, 3, 6, 8, 9, 10: 142

Iterasi ke: 94
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 3, 7, 4, 5, 6, 8, 9, 10: 145

Iterasi ke: 95
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 2, 4, 7, 3, 5, 6, 8, 9, 10: 139

Iterasi ke: 96
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
1, 3, 4, 7, 2, 5, 6, 8, 9, 10: 145

Iterasi ke: 97
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
7, 3, 4, 1, 2, 5, 6, 8, 9, 10: 142

Iterasi ke: 98
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
4, 3, 7, 1, 2, 5, 6, 8, 9, 10: 142

Iterasi ke: 99
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
3, 4, 7, 1, 2, 5, 6, 8, 9, 10: 139

Iterasi ke: 100
Global Optimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Local Optimum:
6, 4, 7, 1, 2, 5, 3, 8, 9, 10: 145

Hasil akhir urutan mesin dengan jarak minimum:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133

---

2. Pencarian Urutan Part

```
Iterasi ke: 1
Global Optimum:
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12: 77
Local Optimum:
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12: 77

Iterasi ke: 2
Global Optimum:
1, 2, 3, 4, 5, 6, 7, 8, 12, 10, 11, 9: 74
Local Optimum:
1, 2, 3, 4, 5, 6, 7, 8, 12, 10, 11, 9: 74

Iterasi ke: 3
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72

Iterasi ke: 4
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
2, 1, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72

Iterasi ke: 5
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
3, 1, 2, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72

Iterasi ke: 6
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
3, 1, 2, 5, 4, 12, 7, 8, 6, 10, 11, 9: 72

Iterasi ke: 7
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
1, 3, 2, 5, 4, 12, 7, 8, 6, 10, 11, 9: 72

Iterasi ke: 8
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
2, 3, 1, 5, 4, 12, 7, 8, 6, 10, 11, 9: 72
```

Iterasi ke: 9
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
2, 3, 1, 8, 4, 12, 7, 5, 6, 10, 11, 9: 72

Iterasi ke: 10
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
3, 2, 1, 8, 4, 12, 7, 5, 6, 10, 11, 9: 72

Iterasi ke: 11
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
1, 2, 3, 8, 4, 12, 7, 5, 6, 10, 11, 9: 72

Iterasi ke: 12
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
1, 2, 3, 6, 4, 12, 7, 5, 8, 10, 11, 9: 72

Iterasi ke: 13
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
2, 1, 3, 6, 4, 12, 7, 5, 8, 10, 11, 9: 72

Iterasi ke: 14
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
3, 1, 2, 6, 4, 12, 7, 5, 8, 10, 11, 9: 72

Iterasi ke: 15
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
3, 1, 2, 4, 6, 12, 7, 5, 8, 10, 11, 9: 72

Iterasi ke: 16
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
1, 3, 2, 4, 6, 12, 7, 5, 8, 10, 11, 9: 72

Iterasi ke: 17
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
2, 3, 1, 4, 6, 12, 7, 5, 8, 10, 11, 9: 72

Iterasi ke: 18
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
2, 3, 1, 5, 6, 12, 7, 4, 8, 10, 11, 9: 72

Iterasi ke: 19
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
3, 2, 1, 5, 6, 12, 7, 4, 8, 10, 11, 9: 72

Iterasi ke: 20
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
1, 2, 3, 5, 6, 12, 7, 4, 8, 10, 11, 9: 72

Iterasi ke: 21
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
1, 2, 3, 8, 6, 12, 7, 4, 5, 10, 11, 9: 72

Iterasi ke: 22
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
2, 1, 3, 8, 6, 12, 7, 4, 5, 10, 11, 9: 72

Iterasi ke: 23
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
3, 1, 2, 8, 6, 12, 7, 4, 5, 10, 11, 9: 72

Iterasi ke: 24
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
3, 1, 2, 6, 8, 12, 7, 4, 5, 10, 11, 9: 72

Iterasi ke: 25
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
1, 3, 2, 6, 8, 12, 7, 4, 5, 10, 11, 9: 72

Iterasi ke: 26
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
2, 3, 1, 6, 8, 12, 7, 4, 5, 10, 11, 9: 72

Iterasi ke: 27
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
2, 3, 1, 4, 8, 12, 7, 6, 5, 10, 11, 9: 72

Iterasi ke: 28
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
3, 2, 1, 4, 8, 12, 7, 6, 5, 10, 11, 9: 72

Iterasi ke: 29
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
1, 2, 3, 4, 8, 12, 7, 6, 5, 10, 11, 9: 72

Iterasi ke: 30
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
1, 2, 3, 5, 8, 12, 7, 6, 4, 10, 11, 9: 72

Iterasi ke: 31
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
2, 1, 3, 5, 8, 12, 7, 6, 4, 10, 11, 9: 72

Iterasi ke: 32
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
3, 1, 2, 5, 8, 12, 7, 6, 4, 10, 11, 9: 72

Iterasi ke: 33
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
3, 1, 2, 8, 5, 12, 7, 6, 4, 10, 11, 9: 72

Iterasi ke: 34
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
1, 3, 2, 8, 5, 12, 7, 6, 4, 10, 11, 9: 72

Iterasi ke: 35
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
2, 3, 1, 8, 5, 12, 7, 6, 4, 10, 11, 9: 72

Iterasi ke: 36
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
2, 3, 1, 6, 5, 12, 7, 8, 4, 10, 11, 9: 72

Iterasi ke: 37
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
3, 2, 1, 6, 5, 12, 7, 8, 4, 10, 11, 9: 72

Iterasi ke: 38
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
3, 2, 1, 6, 7, 12, 5, 8, 4, 10, 11, 9: 72

Iterasi ke: 39
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
1, 2, 3, 6, 7, 12, 5, 8, 4, 10, 11, 9: 72

Iterasi ke: 40
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
2, 1, 3, 6, 7, 12, 5, 8, 4, 10, 11, 9: 72

Iterasi ke: 41
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
2, 1, 3, 12, 7, 6, 5, 8, 4, 10, 11, 9: 72

Iterasi ke: 42
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
3, 1, 2, 12, 7, 6, 5, 8, 4, 10, 11, 9: 72

Iterasi ke: 43
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
1, 3, 2, 12, 7, 6, 5, 8, 4, 10, 11, 9: 72

Iterasi ke: 44
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
1, 3, 2, 7, 12, 6, 5, 8, 4, 10, 11, 9: 72

Iterasi ke: 45
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
2, 3, 1, 7, 12, 6, 5, 8, 4, 10, 11, 9: 72

Iterasi ke: 46
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
3, 2, 1, 7, 12, 6, 5, 8, 4, 10, 11, 9: 72

Iterasi ke: 47
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
3, 2, 1, 7, 12, 5, 6, 8, 4, 10, 11, 9: 72

Iterasi ke: 48
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
1, 2, 3, 7, 12, 5, 6, 8, 4, 10, 11, 9: 72

Iterasi ke: 49
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
2, 1, 3, 7, 12, 5, 6, 8, 4, 10, 11, 9: 72

Iterasi ke: 50
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
2, 1, 3, 5, 12, 7, 6, 8, 4, 10, 11, 9: 72

Iterasi ke: 51
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
3, 1, 2, 5, 12, 7, 6, 8, 4, 10, 11, 9: 72

Iterasi ke: 52
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
1, 3, 2, 5, 12, 7, 6, 8, 4, 10, 11, 9: 72

Iterasi ke: 53
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
1, 3, 2, 8, 12, 7, 6, 5, 4, 10, 11, 9: 72

Iterasi ke: 54
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
2, 3, 1, 8, 12, 7, 6, 5, 4, 10, 11, 9: 72

Iterasi ke: 55
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
3, 2, 1, 8, 12, 7, 6, 5, 4, 10, 11, 9: 72

Iterasi ke: 56
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
3, 2, 1, 6, 12, 7, 8, 5, 4, 10, 11, 9: 72

Iterasi ke: 57
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
1, 2, 3, 6, 12, 7, 8, 5, 4, 10, 11, 9: 72

Iterasi ke: 58
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
2, 1, 3, 6, 12, 7, 8, 5, 4, 10, 11, 9: 72

Iterasi ke: 59
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
2, 1, 3, 7, 12, 6, 8, 5, 4, 10, 11, 9: 72

Iterasi ke: 60
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
3, 1, 2, 7, 12, 6, 8, 5, 4, 10, 11, 9: 72

Iterasi ke: 61
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
3, 1, 2, 7, 12, 8, 6, 5, 4, 10, 11, 9: 72

Iterasi ke: 62
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
3, 1, 2, 12, 7, 8, 6, 5, 4, 10, 11, 9: 72

Iterasi ke: 63
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
1, 3, 2, 12, 7, 8, 6, 5, 4, 10, 11, 9: 72

Iterasi ke: 64
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
2, 3, 1, 12, 7, 8, 6, 5, 4, 10, 11, 9: 72

Iterasi ke: 65
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
2, 3, 1, 12, 7, 6, 8, 5, 4, 10, 11, 9: 72

Iterasi ke: 66
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
3, 2, 1, 12, 7, 6, 8, 5, 4, 10, 11, 9: 72

Iterasi ke: 67
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
1, 2, 3, 12, 7, 6, 8, 5, 4, 10, 11, 9: 72

Iterasi ke: 68
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
1, 2, 3, 12, 7, 5, 8, 6, 4, 10, 11, 9: 72

Iterasi ke: 69
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
2, 1, 3, 12, 7, 5, 8, 6, 4, 10, 11, 9: 72

Iterasi ke: 70
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
3, 1, 2, 12, 7, 5, 8, 6, 4, 10, 11, 9: 72

Iterasi ke: 71
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
3, 1, 2, 12, 7, 4, 8, 6, 5, 10, 11, 9: 72

Iterasi ke: 72
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
1, 3, 2, 12, 7, 4, 8, 6, 5, 10, 11, 9: 72

Iterasi ke: 73
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
2, 3, 1, 12, 7, 4, 8, 6, 5, 10, 11, 9: 72

Iterasi ke: 74
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
2, 3, 1, 7, 12, 4, 8, 6, 5, 10, 11, 9: 72

Iterasi ke: 75
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
3, 2, 1, 7, 12, 4, 8, 6, 5, 10, 11, 9: 72

Iterasi ke: 76
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
1, 2, 3, 7, 12, 4, 8, 6, 5, 10, 11, 9: 72

Iterasi ke: 77
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
1, 2, 3, 4, 12, 7, 8, 6, 5, 10, 11, 9: 72

Iterasi ke: 78
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
2, 1, 3, 4, 12, 7, 8, 6, 5, 10, 11, 9: 72

Iterasi ke: 79
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
3, 1, 2, 4, 12, 7, 8, 6, 5, 10, 11, 9: 72

Iterasi ke: 80
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
3, 1, 2, 8, 12, 7, 4, 6, 5, 10, 11, 9: 72

Iterasi ke: 81
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
1, 3, 2, 8, 12, 7, 4, 6, 5, 10, 11, 9: 72

Iterasi ke: 82
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
2, 3, 1, 8, 12, 7, 4, 6, 5, 10, 11, 9: 72

Iterasi ke: 83
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
2, 3, 1, 6, 12, 7, 4, 8, 5, 10, 11, 9: 72

Iterasi ke: 84
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
3, 2, 1, 6, 12, 7, 4, 8, 5, 10, 11, 9: 72

Iterasi ke: 85
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
1, 2, 3, 6, 12, 7, 4, 8, 5, 10, 11, 9: 72

Iterasi ke: 86
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
1, 2, 3, 7, 12, 6, 4, 8, 5, 10, 11, 9: 72

Iterasi ke: 87
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
2, 1, 3, 7, 12, 6, 4, 8, 5, 10, 11, 9: 72

Iterasi ke: 88
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
3, 1, 2, 7, 12, 6, 4, 8, 5, 10, 11, 9: 72

Iterasi ke: 89
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
3, 1, 2, 7, 12, 4, 6, 8, 5, 10, 11, 9: 72

Iterasi ke: 90
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
1, 3, 2, 7, 12, 4, 6, 8, 5, 10, 11, 9: 72

Iterasi ke: 91
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
1, 3, 2, 7, 12, 8, 6, 4, 5, 10, 11, 9: 72

Iterasi ke: 92
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
2, 3, 1, 7, 12, 8, 6, 4, 5, 10, 11, 9: 72

Iterasi ke: 93
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
3, 2, 1, 7, 12, 8, 6, 4, 5, 10, 11, 9: 72

Iterasi ke: 94
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
3, 2, 1, 12, 7, 8, 6, 4, 5, 10, 11, 9: 72

Iterasi ke: 95
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
1, 2, 3, 12, 7, 8, 6, 4, 5, 10, 11, 9: 72

Iterasi ke: 96
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
2, 1, 3, 12, 7, 8, 6, 4, 5, 10, 11, 9: 72

Iterasi ke: 97
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
2, 1, 3, 12, 7, 6, 8, 4, 5, 10, 11, 9: 72

Iterasi ke: 98
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
2, 1, 3, 6, 7, 12, 8, 4, 5, 10, 11, 9: 72

Iterasi ke: 99
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
1, 2, 3, 6, 7, 12, 8, 4, 5, 10, 11, 9: 72

Iterasi ke: 100
Global Optimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
Local Optimum:
3, 2, 1, 6, 7, 12, 8, 4, 5, 10, 11, 9: 72
Hasil akhir urutan part dengan jarak minimum:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72

Jadi hasil akhirnya adalah:

Urutan Mesin:
1, 2, 5, 6, 4, 7, 3, 8, 9, 10: 133
Urutan Part:
1, 2, 3, 4, 5, 12, 7, 8, 6, 10, 11, 9: 72
