PENGEMBANGAN MODEL REPRESENTASI KONTINYU UNTUK TATA LETAK MESIN DENGAN MENGGUNAKAN *DIFFERENTIAL EVOLUTION*

SKRIPSI

SISCA PRATIWI
0606077573

UNIVERSITAS INDONESIA
FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK INDUSTRI
DEPOK
JUNI 2010

PENGEMBANGAN MODEL REPRESENTASI KONTINYU UNTUK TATA LETAK MESIN DENGAN MENGGUNAKAN *DIFFERENTIAL EVOLUTION*

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

SISCA PRATIWI
0606077573

UNIVERSITAS INDONESIA
FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK INDUSTRI
DEPOK
JUNI 2010

HALAMAN PERNYATAAN ORISINALITAS

Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun yang dirujuk
telah saya nyatakan dengan benar

Nama	**: Sisca Pratiwi**
NPM	**: 0606077573**
Tanda tangan	**:**
Tanggal	**: Juni 2010**

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh

Nama	: Sisca Pratiwi
NPM	: 0606077573
Program Studi	: Teknik Industri
Judul Skripsi	: Pengembangan model representasi kontinyu untuk Tata Letak Mesin dengan menggunakan *Differential Evolution*

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Industri, Fakultas Teknik, Universitas Indonesia.

DEWAN PENGUJI

Pembimbing : Armand Omar Moeis ST M.Sc ( )

Penguji : Ir. Boy Nurtjahyo,MSc ( )

Penguji : Ir. M. Dachyar,MSc ( )

Penguji : Ir. Djoko Sihono Gabriel,MT. ( )

Ditetapkan di : Depok

Tanggal : 29 Juni 2010

UCAPAN TERIMA KASIH

Puji syukur kepada Tuhan Yesus Kristus atas kasih dan karunia-Nya sehingga dengan kekuatan-Nya skripsi ini dapat selesai dengan baik dan dipersembahkan bagi-Nya. Penulisan skripsi ini dilakukan dalam rangka melengkapi salah satu persyaratan untuk menyelesaikan Program Pendidikan Sarjana Teknik Industri, Fakultas Teknik Universitas Indonesia. Saya menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak, sangatlah sulit bagi saya untuk menyelesaikan skripsi ini. Oleh karena itu, pada kesempatan ini, saya menyampaikan ucapan terima kasih kepada:

1. Bapak Armand Omar Moeis ST., M.Sc selaku dosen pembimbing skripsi dan dosen pembimbing akademis, untuk segala bimbingan, bantuan, arahan, dan dukungan dalam pengerjaan skripsi ini.
2. Bapak Komarudin, ST., M.Eng., yang telah memberikan banyak sekali ide, saran, dan pengarahan, sehingga skripsi ini akhirnya dapat terwujud.
3. Lucia Roly Prihandini dan Feby atas semua bantuan yang telah diberikan, khususnya dalam mempelajari program Java.
4. Orang tua, Papa, Mama, Adik, Oma, serta keluarga besar yang juga turut mendoakan dan mendukung saya.
5. Steven Sulistyo, atas segala bantuan dan kerjasama yang baik dalam menyelesaikan skripsi ini.
6. Teman-teman POFTUI dan GBIK yang tidak bisa disebut satu per satu atas pengertian dan dukungannya dalam pengerjaan skripsi ini.
7. Seluruh teman-teman TI06 yang telah berjuang bersama dan memberikan semangat yang luar biasa.
8. Semua pihak yang terlibat dan telah membantu penulis sehingga skripsi ini dapat terselesaikan dengan baik.

Akhirnya, penulis berharap agar skripsi ini bisa memberikan inspirasi dan manfaat bagi semua pihak yang membacanya dan bagi pengembangan ilmu pengetahuan.

Depok, 22 Juni 2010

Penulis

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sitivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Sisca Pratiwi

NPM : 0606077573

Program Studi : Teknik Industri

Departemen : Teknik Industri

Fakultas : Teknik

Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif** (***Non-exclusive Royalty-Free Right***) atas karya ilmiah saya yang berjudul:

Pengembangan model representasi kontinyu untuk Tata Letak Mesin dengan menggunakan *Differential Evolution*

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalih media/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : Juni 2010

Yang Menyatakan

(Sisca Pratiwi)

ABSTRAK

Nama : Sisca Pratiwi
Program Studi : Teknik Industri
Judul : Pengembangan model representasi kontinyu untuk Tata Letak Mesin dengan menggunakan *Differential Evolution*

Penelitian ini adalah sebuah pengembangan model representasi kontinyu dari permasalahan Tata Letak Mesin. Tata letak mesin yang baik dapat mengurangi biaya pemindahan material dalam sebuah sistem produksi. Di dalam permasalahan tata letak mesin, telah dilakukan banyak metode untuk mendapatkan tata letak yang optimal sehingga meminimalkan biaya pemindahan material, tetapi, belum dilakukan pengembangan dengan representasi kontinyu. Penelitian ini mengembangkan sebuah model menggunakan algoritma Differential Evolution sebagai representasi kontinyu. Lebih jauh lagi, dilakukan perbandingan untuk melihat kelebihan metode ini dari metode lain. Setelah penelitian dilakukan, diperoleh sebuah pengembangan metode DE yang baik untuk problem Tata Letak Mesin berukuran kecil sampai sedang.

Kata kunci:
Optimasi, *Machine Layout Problems*, algoritma *Differential Evolution*

ABSTRACT

Name : Sisca Pratiwi
Study Program : Industrial Engineering
Title : Developing Continue Representation Model for Machine Layout Problems Using Differential Evolution Algorithm

This research is a continue representation model development of Machine Layout Problems. A good machine layout can reduce material handling cost of a production system. In order to find solution for machine layout problems, some researchs has been developed. But, there has not been any continue representation for it. This research developes a model using Differential Evolution algorithm as continue representation. Furthermore, a comparison with previous studies was developed. The result of this research was a DE method development to find solutions for small to middle sized MLP.

Key words:
Optimization, Machine Layout Problems, Differential Evolution Algorithm

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR LAMPIRAN

BAB 1
PENDAHULUAN

1.1 Latar Belakang Masalah

Industri adalah tonggak penting dalam kehidupan perekonomian negara. Industri mencakup barang dan jasa, bahkan termasuk komunikasi. Persaingan di antara pelaku industri juga terus berkembang dan terus meningkat. Pelaku industri berusaha untuk meraih pasar sebanyak-banyaknya. Namun, perusahaan pun terus berusaha mencapai keuntungan sebesar-besarnya dengan melakukan efisiensi biaya dan juga menunjukkan performa yang berkualitas. Dalam sebuah industri, sistem produksi yang baik turut menentukan pencapaian dari industri itu.

Sebuah produk biasanya terdiri dari beberapa bahan baku yang diproses oleh beberapa mesin yang berbeda, lalu dirakit dan dikemas oleh mesin yang berbeda pula. Bahan baku dan barang setengah jadi dipindahkan dari mesin satu ke mesin lain melalui sebuah sistem penanganan material (*Material handling System* / MHS). Tata letak mesin menentukan berapa jauh material tersebut berpindah dan juga biaya yang disebabkan oleh hal tersebut. Dua mesin yang memproses barang secara berurutan seharusnya diletakkan berdekatan untuk meminimalkan biaya. Hal ini mudah jika material diproses dalam alur yang sederhana atau linear. Tetapi, akan sulit untuk meminimalkan biaya tersebut bila alur prosesnya rumit. Dan hal itulah yang biasanya terjadi, karena sebuah barang biasanya memerlukan beberapa kali proses yang berulang.

Perbedaan permintaan di setiap pabrik tentunya akan membedakan aliran material di dalamnya. Pengaturan tata letak mesin ikut menentukan besarnya biaya penanganan material. Permasalahan pengaturan mesin secara efisien ini disebut *Machine Layout Problem* (MLP). Studi menunjukkan bahwa 15% sampai 70% dari total biaya operasional pabrik merupakan biaya penanganan material dan tata letak mesin yang efektif dapat mengurangi biaya ini antara 10% sampai 30% (Tompkins et al., 1996).

Banyak usaha yang dilakukan dalam memecahkan masalah ini menggunakan berbagai metode optimasi. Penelitian yang sudah dilakukan menggunakan metode perhitungan diskrit dalam optimasinya. Menurut tinjauan

pustaka yang sudah dilakukan, belum ada pengembangan menggunakan metode kontinyu. Selama ini, metode kontinyu tidak digunakan karena adanya kemungkinan *overlapping* atau tumpang tindih antara satu mesin dan mesin lain. Karena itu, studi ini menggunakan *Differential Evolution* (DE) yang merupakan perhitungan kontinyu untuk memecahkan permasalahan tata letak mesin ini. Tujuannya adalah untuk melihat apakah metode kontinyu dapat digunakan dengan baik untuk masalah ini dibandingkan dengan metode diskrit yang lain. Model inisial ini nantinya dapat dikembangkan lebih lanjut untuk penelitian yang lebih mendalam.

1.2 Diagram Keterkaitan Masalah

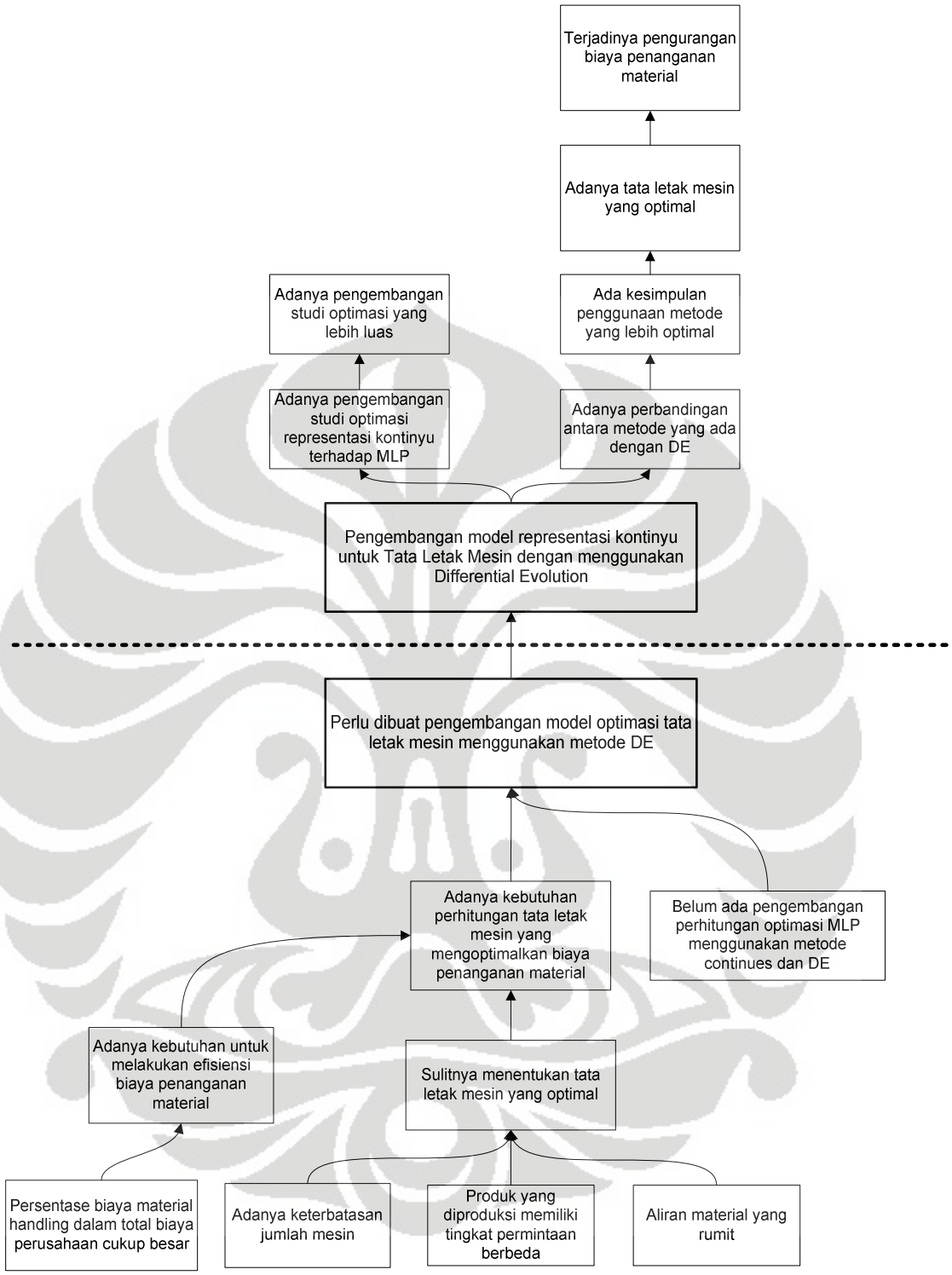
Berdasarkan latar belakang masalah di atas, maka dapat dibuat diagram keterkaitan masalah yang menampilkan permasalahan secara visual dan sistematis. Diagram keterkaitan masalah penelitian ini ditunjukkan oleh Gambar 1.1.

1.3 Rumusan Permasalahan

Permasalahan yang dijadikan fokus penelitian adalah pengembangan model yang dapat menentukan tata letak mesin yang optimal menggunakan metode *Differential Evolution* dan representasi kontinyu. Setelah itu dilihat perbandingannya dengan metode yang sudah dilakukan.

1.4 Tujuan Penelitian

Berdasarkan latar belakang dan diagram keterkaitan masalah yang telah dijelaskan sebelumnya, maka tujuan dari penelitian ini adalah memperoleh hasil dan pengembangan metode DE untuk permasalahan tata letak mesin dan perbandingannya dengan metode lain.



Gambar 1.1 Diagram Keterkaitan Masalah

1.5 Ruang Lingkup Penelitian

Dalam penelitian ini dilakukan pembatasan masalah agar pelaksanaan serta hasil yang akan diperoleh sesuai dengan tujuan pelaksanaannya. Adapun ruang lingkupnya adalah:

1. Penelitian ini menggunakan metode *Differential Evolution* untuk optimasi. DE dipilih karena penelitian ini merupakan representasi perhitungan kontinyu untuk menyelesaikan MLP.
2. Data dan kasus yang digunakan berasal dari penelitian sebelumnya yang telah dipublikasikan.
3. Penelitian ini meminimalkan biaya pemindahan material tanpa memperhitungkan biaya lain seperti biaya pengaturan mesin, dan sebagainya.
4. Bentuk area mesin diasumsikan persegi panjang dengan mengambil batas terluar dari mesin ditambah area kerja.
5. Ukuran mesin adalah tetap.
6. Jarak antar mesin diambil dari titik tengah antar mesin secara *rectilinear*.
7. Total area fasilitas diasumsikan tidak terbatas.
8. Untuk menghindari tumpang tindih antar mesin, dikenakan penalti dalam perhitungan.
9. Pembuatan model dilakukan dengan program komputer berorientasi objek untuk memudahkan pembuatan penelitian khususnya dalam segi *maintenance* karena mudah dilacak apabila terjadi kerusakan.

1.6 Metodologi Penelitian

Metodologi penelitian yang digunakan dalam penelitian ini secara sistematis dijelaskan sebagaimana pada gambar 1.2 dengan uraian sebagai berikut:

1. Perumusan ide-ide topik penelitian dan mengidentifikasi permasalahan

   Pada tahap pertama, dilakukan pencarian tema-tema yang menarik untuk diangkat, baik dari pencarian pada situs-situs internet, jurnal, maupun buku.
2. Studi literatur dasar teori penelitian

Dilakukan studi literatur teori-teori yang menjadi dasar dalam pelaksanaan penelitian. Landasan teori yang terkait dengan penelitian ini adalah *Layout Planning*, *Machine Layout Problems* dan algoritma *Differential Evolution*.

3. Perumusan masalah

   Berdasarkan identifikasi masalah dan studi literatur teori, dapat dirumuskan masalah dalam penelitian ini, yaitu perlu adanya pengembangan optimasi tata letak mesin menggunakan *Differential Evolution* untuk menentukan tata letak mesin terbaik yang meminimalkan biaya pemindahan material.

4. Penentuan topik penelitian

   Setelah masalah teridentifikasi, maka topik dapat ditentukan. Adapun topik dari penelitian ini adalah optimasi permasalahan tata letak mesin menggunakan algoritma *Differential Evolution*.

5. Penentuan tujuan penelitian

   Mengacu pada permasalahan yang ada, maka tujuan penelitian dapat didefinisikan. Adapun, tujuan penelitian ini adalah memperoleh tata letak mesin yang optimal, sehingga dapat mengurangi jarak tempuh dan meminimumkan biaya penanganan material.

6. Pengumpulan data

   Pengumpulan data dilakukan melalui studi literatur yang telah dipublikasikan sebelumnya berupa jumlah aliran material, biaya penanganan material, jumlah mesin, dan ukuran mesin.

7. Pengembangan model.

   Setelah pemantapan studi terhadap dasar teori dan pengumpulan data, maka dapat dibuat model optimasi menggunakan perangkat lunak, yaitu merangkai kode-kode logika pemrograman yang dapat mengaplikasikan algoritma DE pada kasus MLP.

8. Melakukan verifikasi dan validasi model.

   Pada tahap ini, dilakukan verifikasi pada model untuk melihat apakah model yang dibuat ini valid dan dapat dipertanggungjawabkan.

9. Pengujian model.

Pada tahap ini, dilakukan contoh pengolahan data menggunakan model optimasi yang telah disusun, sehingga dapat diperoleh tata letak mesin yang baru.

10. Menganalisis hasil pengolahan data

    Dalam tahap ini dilakukan analisis terhadap hasil pengolahan data untuk dilihat perbaikannya dibandingkan dengan penggunaan metode algoritma lain pada kasus MLP.

11. Menarik kesimpulan

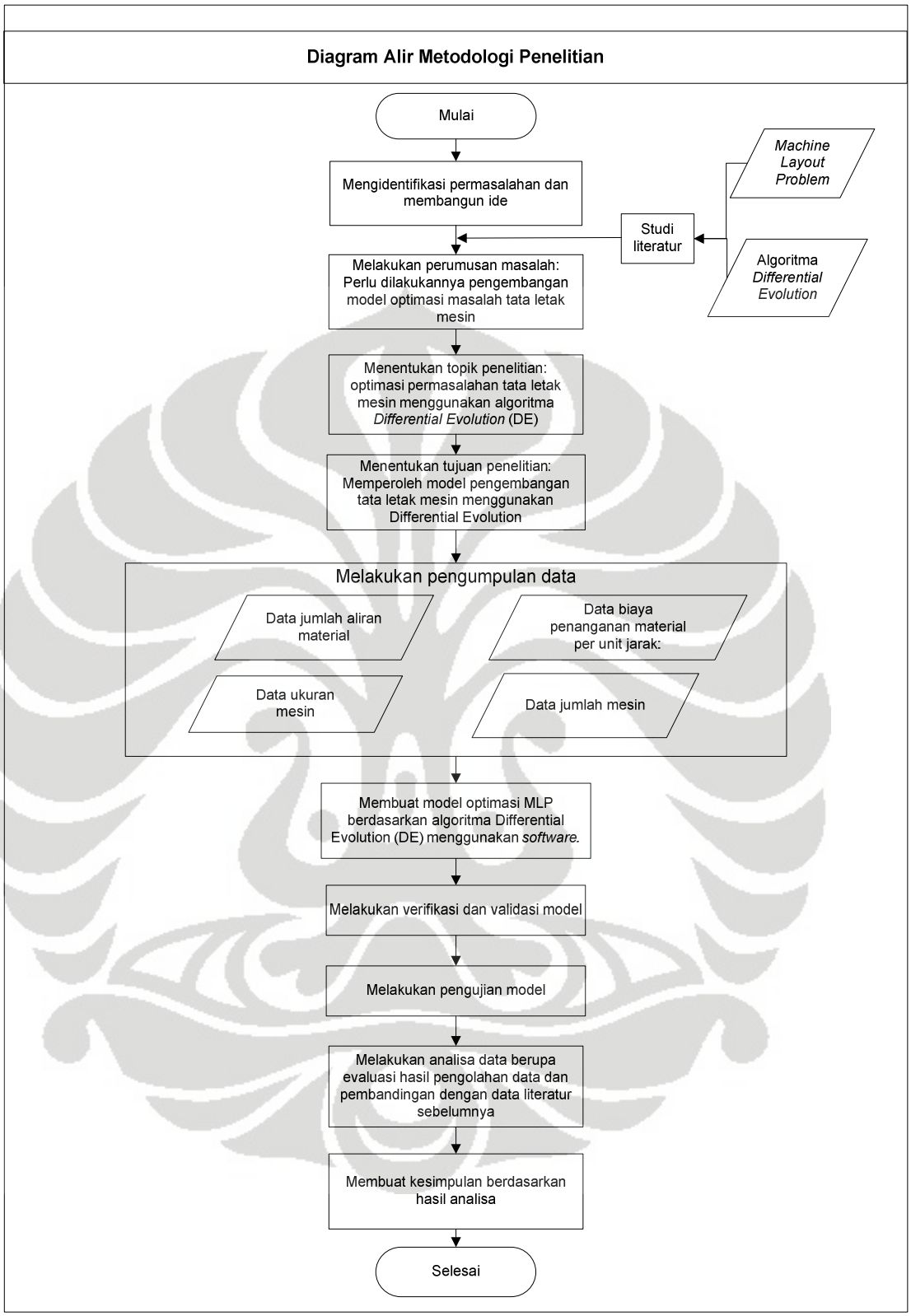
    Dalam tahapan ini akan dihasilkan kesimpulan mengenai keseluruhan penelitian. Kesimpulan dari penelitian ini merupakan ringkasan dari hasil pengolahan data dan analisis yang telah dilakukan sebelumnya.

Untuk menggambarkan secara sistematis, maka dibuat diagram alir metodologi penelitian yang dapat dilihat pada gambar 1.2.

1.7 Sistematika Penulisan

Untuk mempermudah pemahaman alur penelitian ini, maka penulisan penelitian mengenai optimasi masalah tata letak mesin ini disajikan dalam beberapa bab. Bab pertama adalah Pendahuluan. Pada bab Pendahuluan, penulis menjelaskan mengenai latar belakang permasalahan yang menyebabkan dilakukannya penelitian ini. Selain itu, tujuan penelitian dan metodologi penelitian juga dipaparkan dalam bab ini. Penjelasan dalam bab Pendahuluan dilengkapi pula dengan diagram-diagram yang dapat menggambarkan alur permasalahan dan alur penelitian secara sistematis, yaitu diagram keterkaitan masalah dan diagram alir metodologi penelitian.

Pada bab kedua, penulis memaparkan dasar teori yang digunakan dalam mengerjakan penelitian ini. Landasan teori ini diperoleh dari tinjauan pustaka baik dari buku, jurnal, artikel, maupun informasi dari situs-situs di internet. Teori-teori yang dipakai meliputi teori *Machine Layout Problem* (MLP) dan algoritma *Differential Evolution* (DE), yaitu metode yang digunakan untuk melakukan optimasi.



Gambar 1.2. Diagram Alir Metodologi Penelitian

Bab ketiga berisi pengumpulan data yang dibutuhkan penulis dalam melakukan penelitian ini. Data-data tersebut adalah jumlah aliran material, biaya penanganan material, jumlah mesin, dan ukuran mesin.

Bab keempat merupakan bab yang berisi pengolahan data dan analisis. Penulis menjelaskan secara terperinci langkah-langkah yang digunakan dalam pengolahan data sampai diperoleh hasil yang diharapkan. Pengolahan data pada penelitian ini dilakukan dengan menggunakan algoritma *Differential Evolution.* Setelah melakukan pengolahan data, dilakukan analisa hasil pengolahan data.

Setelah pengolahan data dan analisa dilakukan, maka dapat ditarik suatu kesimpulan dari pengerjaan penelitian secara keseluruhan. Kesimpulan ini merupakan jawaban dari tujuan penelitian. Kesimpulan ini ditulis pada bab yang kelima.

BAB 2
LANDASAN TEORI

2.1 Machine *Layout* Problems

2.1.1 Definisi

Machine Layout Problem (MLP) atau permasalahan tata letak mesin merupakan bagian dari *Facility Layout Problems* (FLP) yang merupakan permasalahan menentukan pengaturan fisik dari sebuah sistem produksi. Sistem produksi tidak hanya mengacu pada manufaktur, tetapi juga bisa diaplikasikan pada sistem jasa dan telekomunikasi.

2.1.2 *Facility Layout Problems* / Permasalahan Tata Letak Fasilitas

Facility Layout Problems (FLP) adalah salah satu permasalahan klasik dan tetap aktif sampai sekarang. Banyak penelitian yang telah dilakukan untuk mengoptimasi FLP. Muther (1955) telah mendata beberapa tujuan dari tata letak pabrik. Meskipun tujuan ini ditujukan untuk tata letak pabrik, namun juga relevan untuk FLP. Tujuan itu dapat diringkas sebagai berikut:

- Mengurangi jarak pemindahan material.
- Memiliki aliran material yang seimbang untuk mengurangi bottleneck dalam produksi.
- Menggunakan ruang secara efektif.
- Meningkatkan kepuasan dan keamanan pekerja.
- Memperoleh fleksibilitas sehingga dapat dengan mudah diatur ulang untuk perubahan kondisi.

FLP memainkan peran penting dalam dunia manufaktur karena FLP menggambarkan hubungan fisik dari aktivitas pabrik. FLP merepresentasikan lokasi dari fasilitas di mana material dipindahkan dari satu tempat ke tempat lain. Pemindahan material ini sendiri dapat dikatakan merupakan *non-value added process* yang sedapat mungkin diminimalkan. Karena itu, tata letak fasilitas dapat dikatakan sebuah pendekatan untuk meminimalkan biaya pemindahan material.

FLP dapat dibagi menjadi tiga bagian, *Quadratic Assignment Problem* (QAP), *Unequal Area Facility Layout Problems* (UA-FLP), dan MLP. Perbandingan ketiganya dapat dilihat dalam tabel 2.1

Tabel 2.1 Pembagian FLP

No	Permasalahan	Ukuran Departemen	Kandidat Lokasi	Area Fasilitas
1	QAP	Ukuran sama, dimensi tetap atau diabaikan	Lokasi tetap	Tetap atau diabaikan
2	UA-FLP	Ukuran berbeda, variabel keputusan	Variabel keputusan	Total area departemen
3	MLP	Ukuran berbeda, dimensi tetap	Variabel keputusan	Total area departemen atau dimensi bebas

2.1.3 Metode Penyelesaian MLP

Dalam menentukan tata letak yang terbaik, ada beberapa metode yang dilakukan:

- Pendekatan Konvensional

  Pendekatan konvensional menggunakan grafik dan diagram. Sebuah analisis yang detail dilakukan terhadap rute material, volume yang dipindahkan, jarak, frekuensi, *rate* material yang berpindah, dan biaya perjalanan (Apple, 1977).

- Pendekatan Kuantitatif

  Pendekatan kuantitatif lebih mengutamakan dengan penggunaan lokasi secara optimal dengan perpindahan material. Pendekatan ini mencakup teknik Penelitian Operasional sebagai teknis matematis.

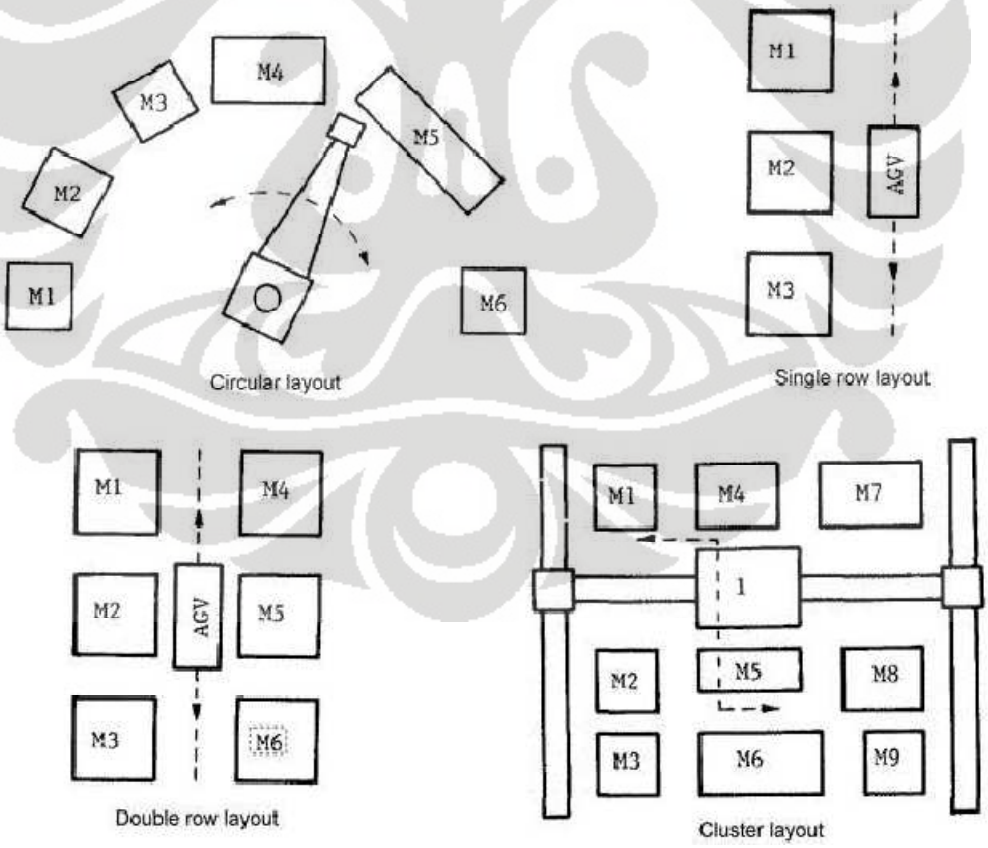
Beberapa pendekatan kuantitatif secara matematis yang digunakan untuk memecahkan permasalahan MLP diantaranya programa linear, programa transportasi, programa integer, dan sebagainya.

Beberapa penelitian sebelumnya menggunakan formulasi QAP atau CRAFT seperti formulasi yang dikembangkan Armour dan Buffa (1963). Di dalam formulasi QAP, biaya penanganan material diminimalkan dengan

mendistribusikan mesin sesuai luas lokasi yang tersedia, tanpa mempedulikan ukuran mesin. Dalam perhitungan CRAFT, fasilitas dijadikan dalam *grid* yang berisi departemen dengan luas area yang tetap tapi bentuknya fleksibel. Karena bentuk dari departemen tidak tetap, maka meskipun sederhana, dapat memberikan solusi yang baik.

Pada tahun 1957, Koopmans dan Beckman mengembangkan QAP model untuk memecahkan permasalahan tata letak *multi-row*. Dalam model ini, tujuannya adalah untuk meminimalkan biaya penanganan material sehingga memaksimalkan keuntungan.

Tahun 1988, Heragu dan Kusiak melakukan penelitian menggunakan metode TAA (Triangle Assignment Algorithm) untuk tata letak berbentuk single-row, double-row, dan cluster. Pola ini adalah pola yang selama ini biasa digunakan dalam tata letak mesin. Berbeda dengan tata letak mesin secara kontinyu, pola ini sudah memiliki baris sendiri dalam meletakkan mesin. Gambarannya dapat dilhat dalam gambar 2.1



Gambar 2.1 Pola Tata Letak Mesin Diskrit

(sumber: Kusiak dan Heragu, 1988)

Souliah (1995) menerapkan *simulated annealing* (SA) untuk memecahkan permasalahan tata letak mesin *cellular* dan mendapatkan solusi dalam waktu yang masuk akal. Algoritma ini mengelompokkan dan mengatur mesin ke dalam sel-sel yang sesuai dengan iterasi.

Gupta dkk (1996) menggunakan Genetic Algorithm (GA) untuk menentukan tata letak *cellular*. Tetapi, dalam formulasinya, dia membatasi pengaturan *cell* kepada *linear single row* dan *linear double row* saja. Tata letak sebenarnya tidak diperhitungkan.

Morad mengembangkan pengunaan GA untuk MLP. Dua algoritma dikembangkan untuk memecahkan permasalahan *single-row* dan *multi-row* dalam *cellular layouts*. Dari studi ini, ditemukan bahwa pendekatan genetik memberikan solusi yang baik dengan waktu perhitungan yang masuk akal. Keuntungan dari algoritma ini adalah tata letak mesin dapat ditentukan menggunakan sedikit data, memberikan keuntungan jika data biaya penataulangan mesin tidak diketahui.

Yang dan Peters (1998) memeriksa apakah permasalahan tata letak mesin dapat sesuai untuk lingkungan *job shop*. Tujuan mereka adalah untuk mengurangi biaya penanganan material dengan membuat tata letak mesin yang optimal seiring berjalannya waktu. Yang dan Peters juga membuat penelitian yang dikaitkan dengan biaya pemasangan mesin dan biaya penataulangan mesin. Fleksibilitas tata letak mesin seperti ini berkaitan dengan penggantian peralatan produksi dan kemampuan adaptasi terhadap peralatan baru. Yang dan Peters memodelkan MLP statis sebagai *Reduced Integer Programming (RIP)* tetapi formulasinya tidak dapat digunakan untuk real-sized problems. Sebagai alternatifnya, penelitian ini menyarankan penggunaan algoritma *Ant Colony Optimization* (ACO) untuk memecahkan masalah ini.

Studi selanjutnya oleh Corry dan Kozan (2004) menggunakan algoritma ACO untuk memecahkan permasalahan tata letak mesin yang bersifat *robust.* Hasil dari penelitian ini adalah algoritma ACO secara konsisten memberikan hasil yang lebih baik daripada metode RIP sebesar 10%-25% terhadap biaya relokasi dan 5%-7% terhadap biaya pemindahan material.

Lebih jauh lagi, Andersen (2006) melakukan perbandingan antara tiga metode untuk permasalahan tata letak mesin ini. Metode yang dibandingkan adalah *reduced integer programming, ant colony optimization*, dan *simulated annealing*. Dari ketiga metode itu, ditemukan bahwa hasil dari menggunakan reduced integer programming tidak sebagus hasil dari SA ataupun ACO. Perbandingan antara SA dan ACO menunjukkan bahwa untuk masalah kecil, ACO lebih baik daripada SA, namun untuk permasalahan yang lebih besar, yang terjadi adalah sebaliknya.

2.2 Algoritma *Differential Evolution*

Ada berbagai teknik yang dapat digunakan untuk melakukan optimasi. Untuk kasus-kasus dengan kendala yang banyak dibutuhkan teknik-teknik optimasi yang lebih modern seperti algoritma optimasi. Algoritma merupakan suatu kumpulan perintah yang digunakan untuk menyelesaikan suatu masalah. Diantara algoritma-algoritma yang ada, algoritma *Differential Evolution* (DE) hadir sebagai algoritma baru yang cukup tangguh dalam menghadapi permasalahan dengan jumlah kendala yang banyak. Prinsip *Differential Evolution* didasarkan pada konsep evolusi biologi, yang terdiri dari proses populasi, proses mutasi, proses pindah silang, dan proses penyeleksian. *Differential Evolution* menggunakan acak sampling sehingga akan menghasilkan penyelesaian berbeda, meskipun model awalnya tidak diubah (Price, 1999). *Differential Evolution* akan menggabungkan elemen-elemen dari solusi-solusi yang telah ada untuk menciptakan solusi baru dengan mewarisi ciri-ciri yang dipunyai oleh tiap induk. Beberapa keunggulan DE adalah strukturnya yang sederhana, mudah diimplementasikan, cepat dalam mencapai tujuan, dan bersifat tangguh. Melalui penggunaan algoritma ini, diharapkan optimasi transportasi dapat dilakukan serta tetap mampu mempertahankan kualitas pelayanan pada pelanggan.

2.2.1 Sejarah

Dalam bidang matematika dan komputasi, algoritma merupakan kumpulan perintah untuk menyelesaikan suatu masalah. Perintah-perintah ini dapat diterjemahkan secara bertahap dari awal hingga akhir. Masalah tersebut dapat

berupa apa saja, dengan catatan untuk setiap masalah, ada kriteria kondisi awal yang harus dipenuhi sebelum menjalankan algoritma. Algoritma sering mempunyai langkah pengulangan (iterasi) atau memerlukan keputusan (logika Boolean dan perbandingan) sampai tugasnya selesai. Algoritma memiliki banyak kegunaan dimana salah satu kegunaanya adalah untuk permasalahan optimasi, sehingga algoritma jenis ini biasa disebut dengan algoritma optimasi. Algoritma banyak digunakan dalam pemecahan masalah optimasi karena pada umumnya di banyak kasus di dunia nyata, permasalahan yang timbul memiliki permasalahan yang sulit atau tidak mungkin dikerjakan dengan menggunakan teknik-teknik optimasi konvensional yang dikerjakan secara manual. Misalnya, dalam kebanyakan kasus berskala besar, permasalahan optimasi yang ada memiliki jumlah variabel yang sangat besar hingga mencapai ratusan, memiliki fungsi-fungsi, baik kendala maupun tujuan, yang bersifat non-linier sehingga memiliki banyak optima lokal, atau fungsi yang non-kontinu.

Algoritma optimasi memiliki jenis yang sangat banyak, namun hingga saat ini belum ada suatu algoritma superior yang dapat menyelesaikan permasalahan. Selama empat dekade, belum ada penelitian yang dapat memberikan solusi algoritma yang terbaik, karena pada prakteknya masih banyak hambatan, seperti fungsi yang *non-differentiable,* non-kontinu, non-linear, multi-dimensi, memiliki banyak *constraint* (kendala), dan bersifat stokastik hingga pada akhir tahun 1995, Storn dan Price menawarkan suatu terobosan baru, yaitu algoritma *Differential Evolution* (DE). DE pertama kali mulai dikembangkan ketika Price mencoba memecahkan permasalahan polynomial Chebyshev yang diajukan oleh Storn. Dalam mencoba memecahkan permasalahan tersebut, Price terinspirasi untuk menggunakan selisih dari vektor dalam mencari suatu solusi penyelesaian, hingga setelah melalui diskusi yang panjang dan simulasi dengan menggunakan program computer yang dilakukan oleh Storn dan Price, dikembangkanlah *Genetic Annealing* yang merupakan cikal bakal dari DE itu sendiri. DE merupakan algoritma yang masuk kedalam kelompok optimasi yang masuk ke dalam sub-kelompok algoritma evolusioner (EA). Sama seperti EA yang lainnya seperti *Genetic Algorithm* (GA), *Evolution Strategy, Learning Classifier System*, dan

lain-lain, DE memiliki konsep yang terinspirasi dari teori evolusi biologi, dimana di dalamnya terdapat reproduksi, mutasi, rekombinasi, dan seleksi.

DE pertama kali dijelaskan oleh Price dan Storn di ICSI *technical report* pada tahun 1995. Satu tahun kemudian, DE sukses didemonstrasikan di *First International Contest on Evolutionary Optimization* yang diadakan bersamaan dengan *International Conference on Evolutionary Computation* yang diadakan oleh IEEE (*Institute of Electrical and Electronics Engineers*) dan berhasil memenangkan tempat ketiga. Terinspirasi dari hasil tersebut, Price dan Storn menulis sebuah artikel untuk jurnal Dr. Dobbs ("*Differential Evolution: A Simple Evolution Strategy for Fast Optimization*") yang diterbitkan pada April 1997 dan selanjutnya mereka menerbitkan artikel lagi untuk *Jounal of Global Optimization* ("*Differential Evolution: A Simple and Effiecient Heuristic for Global Opptimization over Continuous Space*"). Artikel-artikel ini memperkenalkan DE ke komunitas internasional dan mendemonstrasikan keunggulan yang dimiliki oleh DE dibandingkan dengan metode heuristik yang lainnya. Pada tahun 1999, Price membuat suatu ringkasan yang berisi penjelasan mengenai algoritma DE ini dalam bentuk buku yang berjudul "*New Ideas in Optimization*"[1].

DE telah sukses diterapkan di berbagai bidang, baik teknik maupun sains, beberapa contoh diantaranya adalah:

- Desain *filter* digital (Storn, 1996)
- Pengambilan keputusan untuk produksi bahan bakar alkohol (Wang et al., 1998)
- Proses fermentasi untuk *lot size* tertentu (Chiou dan Wang, 1999)
- Perpaduan multi sensor (Joshi dan Sanderson, 1999)
- Optimasi dinamis untuk reaksi polimer yang terus-menerus (Lee et al., 1999)
- Optimasi pertukaran panas (Babu dan Munawar, 2000)
- Perencanaan persediaan (Srikanta dan Rambabu, 2003), dan lain-lain

[1] http://www.icsi.berkeley.edu/~storn/code.html

2.2.2 Konsep Dasar

Seperti yang dijelaskan di atas, DE bekerja dengan meniru teori evolusi biologi. DE memiliki beberapa keunggulan dibandingkan dengan metode optimasi klasik, diantaranya adalah:

- Memiliki populasi yang berisikan calon-calon penyelesaian
- Merupakan metode non-deterministik yang menghasilkan solusi-solusi yang berbeda meskipun model awalnya tidak dirubah, karena bekerja dengan menggunakan *random sampling*.
- Menggunakan elemen-elemen dari solusi-solusi yang telah ada untuk menciptakan solusi baru dengan cirri-ciri yang diwariskan dari elemen-elemen induknya.

Mirip dengan EA lainnya, DE menggunakan vektor-vektor yang merepresentasikan kandidat-kandidat penyelesaian dimana teknik pencariannya dilakukan sekaligus atas sejumlah solusi yang disebut dengan populasi. Populasi awal (generasi ke nol) dibentuk dengan membangkitkan bilangan acak, sedangkan populasi berikutnya merupakan hasil evolusi dari vektor-vektor yang telah melalui tahap reproduksi, mutasi, rekombinasi, dan seleksi. Setiap individu didefinisikan sebagai vektor berdimensi-D dimana vektor-vektor tersebut dilambangkan sebagai $x_{i,g}$ yang merupakan anggota populasi pada generasi ke-g. Populasi dinotasikan sebagai P_x yang terdiri atas vektor-vektor tersebut yang berdimensi Np dimana Np merupakan ukuran populasi. Oleh karena itu, populasi dan vektor yang menjadi calon-calon penyelesaian dapat dilambangkan ke dalam bentuk umum seperti berikut:

$$\boldsymbol{Px,g} \;=\; (xi,g),\; i = 0,1,\dots,Np-1, g = 0,1,\dots,gmax \qquad (2.1)$$
$$\mathbf{x_i,}\boldsymbol{g} \;=\; (xj,i,g), j = 0,1,\dots,D-1$$

Pada setiap generasi, tiap individu calon penyelesaian akan melewati proses evaluasi dimana individu-individu tersebut akan membentuk vektor target dan dihitung fungsi objektifnya (atau seringkali disebut sebagai *fitness function*). Selain itu, individu-individu tersebut akan dilakukan proses mutasi dan pindah

silang (*crossover*) agar dapat membentuk vektor *trial* yang digunakan untuk membentuk populasi anak (populasi pada generasi selanjutnya). Populasi generasi selanjutnya akan dibentuk dengan cara membandingkan fungsi objektif dari vektor induk dan anak (vektor *trial*) di mana individu dengan nilai fungsi objektif yang terbaik akan lolos ke generasi selanjutnya. Proses tersebut akan terus diulang hingga kriteria terminasi terpenuhi.

2.2.3 Tahapan Pengerjaan

Dalam proses pencarian solusi, DE akan melalui tahapan-tahapan berupa inisialisasi, mutasi, pindah silang, seleksi, dan terminasi.

2.2.3.1 Inisialisasi

Tahapan inisialisasi merupakan penetapan parameter kontrol dan populasi awal (generasi ke-0). Tujuan penetapan parameter kontrol adalah untuk menemukan solusi yang dapat diterima melalui sejumlah evaluasi fungsi dan nantinya akan berdampak pada performa DE. DE memiliki parameter kontrol yang tidak banyak, dimana hal ini merupakan salah satu keunggulan DE dibandingkan algoritma optimasi lainnya. Parameter kontrol pada DE diantaranya adalah ukuran populasi, parameter kontrol mutasi, dan parameter kontrol pindah silang.

Ukuran populasi (Np) merupakan jumlah saluran populasi dalam satu generasi (berupa kolom matriks) yang nilainya tetap selama proses pencarian. Namun, jika proses pencarian mengalami hambatan, maka nilai dari Np dapat dinaikkan. Umumnya, nilai dari Np = 10 x D, dimana D merupakan ukuran dimensi (berupa baris matriks). Dimensi merupakan input parameter yang nilainya akan berubah-ubah selama proses pencarian solusi. Populasi awal (berisikan individu sejumlah Np) yang diinisialisasikan, merupakan populasi solusi awal yang dapat diperoleh dari metode heuristic maupun diperoleh dengan pengambilan sampel secara acak.

Parameter kontrol mutasi (F) merupakan parameter kontrol bernilai bilangan asli positif yang berfungsi dalam mengendalikan tingkat evolusi dari populasi.. Nilai efektif dari F umunya berada pada kisaran [0.4,1].Walaupun pada

teorinya nilai dari parameter kontrol mutasi tersebut tidak memiliki batas atas (dapat lebih besar dari 1), sangat jarang nilainya lebih besar dari 1. Selain itu, nilai F yang lebih kecil dari 0.4 juga tidak efektif karena akan membawa vektor mutasi yang mendekati vektor target.

Parameter kontrol pindah silang (Cr) merupakan parameter yang digunakan dalam penentu pewarisan gen yang dimiliki oleh vektor target dan vektor mutasi dalam pembentukan vektor *trial* dengan cara membandingkannya dengan bilangan acak yang dibangkitkan pada proses pindah silang. Nilai dari Cr ini berkisar pada antara [0,1].

Setelah menentukan parameter kontrol, maka dilakukan evaluasi dari populasi awal yang terbentuk. Evaluasi dilakukan dengan cara menghitung nilai dari fungsi objektifnya. Evaluasi ini dilakukan sebagai ukuran dalam menentukan karakteristik dari vektor pada generasi selanjutnya.

2.2.3.2 Mutasi

Setelah melakukan inisialisasi, proses selanjutnya adalah proses mutasi. Mutasi merupakan proses untuk membentuk vektor mutasi ($\mathbf{v_{i,g}}$) yang diperoleh dari mengalikan selisih dari dua vektor pada generasi sekarang yang dipilih secara acak dengan parameter kontrol mutasi (F) lalu dijumlahkan dengan vektor yang ketiga yang juga dipilih secara acak. Oleh karena itu, ukuran populasi minimal adalah empat. Rumus dari proses mutasi ini adalah sebagai berikut:

$$vi, g = xr0, g + F . (xr1, g - xr2, g) \quad (2.2)$$

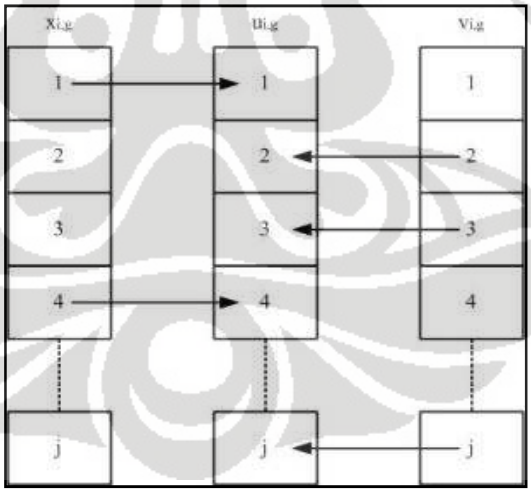
2.2.3.3 Pindah Silang

Untuk melengkapi proses mutasi, DE juga menggunakan proses pindah silang (*crossover*), atau terkadang disebut sebagai rekombinasi diskrit (*discrete recombination*) (Price, Storn, Lampinen 2005). Pindah silang merupakan proses yang bertujuan untuk memperkaya keanekaragaman gen dalam populasi yang akan memasuki generasi yang berikutnya dengan menyilangkan gen yang dimiliki oleh populasi vektor mutan dengan populasi vektor target sehingga membentuk populasi vektor *trial*. Proses pindah silang ini melibatkan parameter kontrol

pindah silang (Cr). Parameter kontrol pindah silang ini merupakan elemen yang menentukan gen-gen mana saja yang diperoleh dari vektor target dan mutan untuk diwariskan kepada vektor *trial*. Penentuan ini dilakukan dengan cara membandingkan nilai Cr tersebut dengan bilangan yang dibangkitkan secara acak. Jika nilai Cr lebih besar dari bilangan acak, maka gen dari vektor mutasi akan lolos untuk memasuki vektor *trial*, sedangkan jika nilai Cr lebih kecil atau sama dengan bilangan acak, maka gen dari vektor target yang akan lolos memasuki vektor trial. Berikut adalah formula umum dari proses pindah silang beserta representasi visualnya:

$$\boldsymbol{u_{i,g}} = u_{j,i,g} = \begin{cases} v_{j,i,g} & if\ (rand_j(0,1) \leq Cr\ or\ j = j_{rand} \\ x_{j,i,g} & if\ (rand_j(0,1) > Cr\ or\ j = j_{rand} \end{cases} \quad (2.3)$$

Setelah diperoleh populasi dari vektor *trial*, maka vektor *trial* itu akan dievaluasi nilai objektifnya sebagaimana evaluasi yang dilakukan terhadap vektor target dimana nilai ini digunakan pada proses selanjutnya, yaitu proses seleksi.



Gambar 2.2 Proses Terjadinya Pindah Silang

2.2.3.4 Seleksi

Tahapan ini merupakan tahapan dimana terjadi pemilihan antara vektor target dan vektor *trial* yang akan lolos untuk masuk ke generasi yang selanjutnya. Penyeleksian dilakukan dengan cara membandingkan nilai yang merupakan hasil

dari evaluasi nilai objektif pada vektor target dan vektor *trial*. Vektor yang akan lolos ke generasi selanjutnya adalah vektor yang memiliki nilai evaluasi yang terbaik seperti yang ditunjukkan oleh bentuk umum di bawah ini:

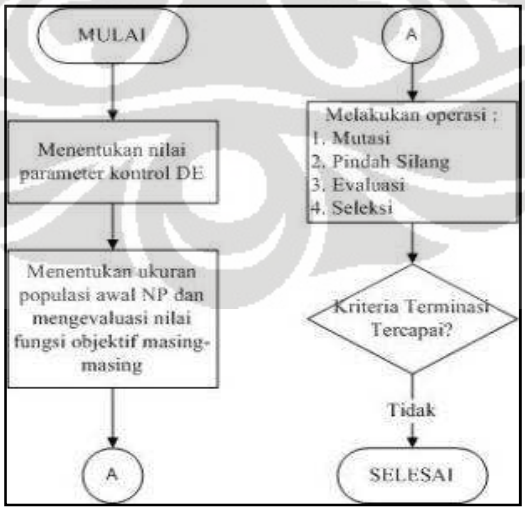
$$x_{i,g+1} = \begin{cases} u_{i,g} \; if \; f(u_{i,g}) \leq f(x_{i,g}) \\ x_{i,g} \; if \; f(u_{i,g}) > f(x_{i,g}) \end{cases} \tag{2.4}$$

2.2.3.5 Terminasi

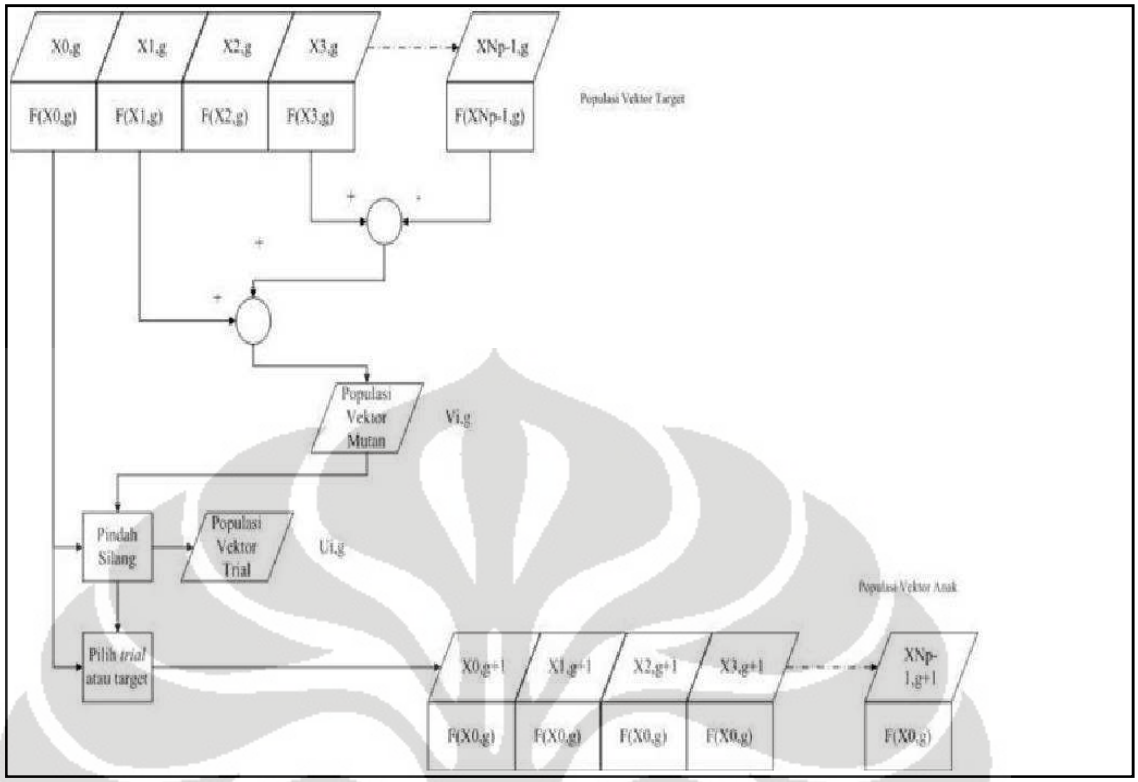
Terminasi merupakan keadaan dimana proses pencarian solusi optimal berhenti. Terminasi terjadi ketika proses pencarian solusi optimal telah mencapai kriteria terminasi. Namun, bila kriteria terminasi belum terpenuhi, maka akan dibentuk lagi generasi baru dengan mengulangi langkah-langkah sebelumnya dari awal. Umumnya kriteria terminasi, seperti disampaikan oleh Price, Storn dan Lampinen (2005) adalah sebagai berikut:

- Jumlah iterasi maksimum
- Waktu komputasi maksimum
- Mencapai keadaan konvergen (nilai dari fungsi objektif yang optimal tidak lagi berubah)

Secara lebih detil, tahapan proses pengerjaan DE ditunjukkan pada gambar 2.6. dan gambar 2.7.



Gambar 2.3 Diagram Alir Tahapan Pengerjaan DE Secara Umum



Gambar 2.4 Diagram Alir Proses Pencarian Solusi DE

2.2.4 Strategi pencarian DE

Dalam perkembangannya, ada beberapa strategi pencarian untuk DE yang juga diperkenalkan oleh Price dan Storn (1996). Perbedaan strategi ini bukan berarti salah satu lebih baik dari yang lain. Setiap permasalahan mempunyai solusi yang cocok dengan strategi tertentu.

Berikut ini adalah 5 strategi yang dikembangkan oleh Price dan Storn:

- DE/Rand/1

  Individu induk dipilih secara acak dari populasi, dan vektor mutasi diambil berdasarkan jarak satu individu acak.

- DE/Rand/2

  Individu induk dipilih secara acak dari populasi, dan vektor mutasi diambil berdasarkan jarak dua individu acak.

- DE/Best/1

  Individu induk merupakan yang terbaik sesuai fungsi tujuan dari populasi, dan vektor mutasi diambil berdasarkan jarak satu individu acak

- DE/Best/2

  Individu induk merupakan yang terbaik sesuai fungsi tujuan dari populasi, dan vektor mutasi diambil berdasarkan jarak dua individu acak

- DE/rand-to-best/1

  Merupakan kombinasi dari formulasi *random* dan *best*

Perbedaan dari kelima strategi ini terletak pada formula pencarian vektor anak. Perbandingan formula ini dapat dilihat pada Tabel 2.2

Tabel 2.2 Pembagian FLP

Strategi	Formulasi
DE/Rand/1	$\omega = x_1 + F.(x_2 - x_3)$
DE/Rand/2	$\omega = x_5 + F.(x_1 + x_2 - x_3 - x_4)$
DE/Best/1	$\omega = x_{best} + F.(x_2 - x_3)$
DE/Best/2	$\omega = x_{best} + F.(x_1 + x_2 - x_3 - x_4)$
DE/rand-to-best/1	$\omega = x_{ind} + \lambda.(x_{best} - x_1) + F.(x_2 - x_3)$

Lebih jauh lagi, banyak pengembangan dari DE yang terus dilakukan oleh peneliti, sepeti yang dikemukakan oleh Feoktistov dalam bukunya *Differential Evolution*: *In Search of Solutions*, tetapi, hal itu tidak dibicarakan lebih lanjut dalam penelitian ini.

2.2.5 Permasalahan Kombinatorial dengan DE

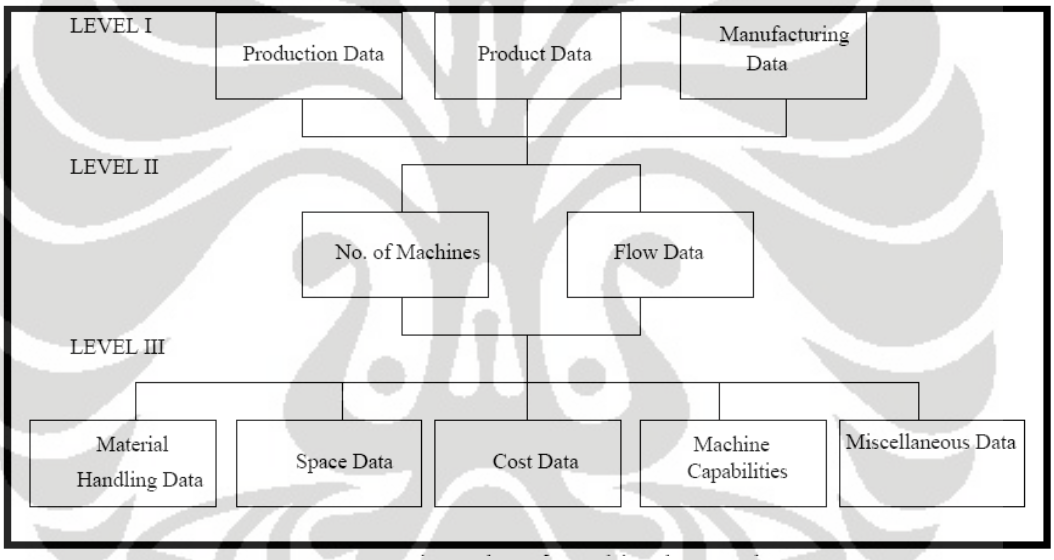
Dalam banyak kasus, seringkali permasalahan yang ditemui berupa permasalahan kombinatorial yang memiliki fungsi-fungsi yang berupa integer. Beberapa contoh permasalahan yang berupa masalah kombinatorial diantaranya adalah masalah penjadwalan, tata letak, dan *routing*. Pada masalah-masalah tersebut, variabel-variabel yang ada berupa integer dan bersifat stokastik, sehingga kemungkinan dari solusi-solusi yang ditawarkan ada banyak, sehingga kemungkinan untuk mendapatkan nilai optimum lokal pun menjadi sangat besar. Untuk kasus-kasus yang berskala kecil, umumnya permasalahan tersebut dapat dipecahkan dengan *Integer Linear Programming* (ILP), namun untuk kasus yang

lebih besar, solusi yang ditawarkan dapat menjadi amat banyak. Misalnya, dalam kasus *routing*, ada banyak sekali kemungkinan dalam menentukan urutan jalur terpendek dari satu tempat ke tempat lainnya jika jumlah tempat yang harus dilewati ada puluhan atau ratusan. Kasus seperti ini sulit untuk dipecahkan dengan menggunakan teknik optimasi konvensional seperti ILP karena banyaknya kemungkinan kombinasi penyelesaian yang tersedia dapat menyebabkan hasil optimum yang diperoleh dapat berupa nilai optimum lokal, bukan optimum global.

Oleh karena itu, untuk menyelesaikan permasalahan kombinatorial yang rumit diperlukan pendekatan algoritmik seperti DE, dimana pendekatan algoritmik dapat membawa solusi penyelesaian menjauhi nilai-nilai optima lokal. Pemecahan permasalahan optimasi kombinatorial dengan menggunakan DE merupakan solusi yang sangat tepat karena, seperti yang sudah dijelaskan sebelumnya, DE memiliki struktur yang sederhana dan parameter kontrol yang relative sedikit jika dibandingkan dengan algoritma evolusi lainnya. Selain itu, DE juga menawarkan berbagai solusi berupa populasi solusi sehingga dari solusi akhir yang ditawarkan dapat dipilih solusi mana yang paling optimum.

BAB 3
PENGUMPULAN DATA

Hasan dan Albin (1994) memberikan studi lebih lanjut mengenai tipe data yang dibutuhkan dalam MLP. Data tata letak mesin ditunjukkan dalam hierarki berdasarkan seberapa detail tata letaknya dirancang. Ketika tata letak yang dibutuhkan hanyalah untul mencari susunan mesin, data yang merepresentasilan jumlah mesin dan hubungan aliran materialnya sudah cukup. Jika tata letak yang dibutuhkan lebih detail, maka lebih banyak data yang dibutuhkan. Dalam menemukan data untuk level III tentu lebih sulit didapat dari perusahaan. Hierarki ini dapat dilihat di Gambar 3.1.



Gambar 3.1 Hierarki Kebutuhan Data MLP

Sesuai dengan pembatasan masalah dari penelitian ini, yaitu untuk meminimalkan biaya pemindahan material, maka data utama yang dibutuhkan untuk MLP ini adalah:

- Aliran material
- Area atau dimensi mesin

Data aliran material yang digunakan dalam perhitungan merupakan one-side matriks. Hal ini berarti bahwa data aliran material dari satu mesin ke mesin selanjutnya sama untuk sebaliknya dan tidak perlu dihitung ulang. Aliran material dari mesin satu ke mesin dua sama dengan mesin dua ke mesin satu. Dengan

demikian, dapat juga dikatakan bahwa data aliran material menunjukkan bobot kepentingan satu mesin dengan mesin lainnya, seperti yang digunakan dalam beberapa publikasi.

Ukuran mesin yang digunakan di sini diasumsikan sudah termasuk area kerja, gang, dan lainnya. Memang dalam perhitungan ini dilakukan simpifikasi terhadap hal-hal tersebut di atas yang perhitungannya dilakukan oleh bidang studi lain seperti tata letak pabrik, dan sejenisnya. Selain itu, dalam perhitungan ini diasumsikan juga bentuk mesin sebagai segi empat.

Untuk membandingkan apakah metode *Differential Evolution* yang digunakan lebih baik dari penelitian terdahulu, digunakan beberapa problem yang telah diselesaikan pada jurnal dan publikasi terdahulu. Daftar set data yang digunakan dapat dilihat pada Tabel 3.1. Data lengkap yang digunakan dapat dilihat pada Lampiran 1.

Tabel 3.1 Daftar Set Data

No	Set Data	Jumlah mesin	Referensi
1	KH1	4	Heragu and Kusiak. (1988)
2	KH2	4	Heragu and Kusiak. (1988)
3	KH3	4	Heragu and Kusiak. (1988)
4	KH4	4	Heragu and Kusiak. (1988)
5	KH5	4	Heragu and Kusiak. (1988)
6	KH6	4	Heragu and Kusiak. (1988)
7	KH7	4	Heragu and Kusiak. (1988)
8	KH8	4	Heragu and Kusiak. (1988)
9	KH9	4	Heragu and Kusiak. (1988)
10	NUG5	5	Nugent et al (1968)
11	NUG6	6	Nugent et al (1968)
12	NUG7	7	Nugent et al (1968)
13	NUG8	8	Nugent et al (1968)
14	NUG12	12	Nugent et al (1968)
15	NUG15	15	Nugent et al (1968)

16	NUG20	20	Nugent et al (1968)
17	OPT8	8	VIP-PLANOPT *Engineering Optimization Software*
18	OPT12	12	VIP-PLANOPT *Engineering Optimization Software*
19	OPT20	20	VIP-PLANOPT *Engineering Optimization Software*
20	OPT28	28	VIP-PLANOPT *Engineering Optimization Software*
21	OPT50	50	VIP-PLANOPT *Engineering Optimization Software*
22	OPT75	75	VIP-PLANOPT *Engineering Optimization Software*
23	DUN62	62	Dunker, et al. (2003)

Set data yang digunakan dipilih dari beberapa problem yang telah banyak digunakan. Tabel 3.2 dapat digunakan untuk membandingkan karakteristik dari set data yang digunakan dalam penelitian sebelumnya. Dapat dilihat bahwa setiap peneliti yang berbeda telah mengaplikasikan berbagai karakteristik yang berbeda pula dalam penggunaan set data untuk penelitiannya.

Tabel 3.2 Karakteristik Data

No	**Literatur**	**Jumlah Problem Diuji**	**Ukuran Problem Minimum**	**Ukuran Problem Maksimum**
1	Heragu and Kusiak. (1988)	17	4	30
2	Nugent et al (1968)	8	5	30
3	VIP-PLANOPT *Engineering Optimization Software*	12	3	125
4	Dunker, et al. (2003)	1	62	62

Dalam proses pengumpulan data, ada keterbatasan karena ada beberapa penelitian yang tidak mempublikasikan data yang digunakan, sehingga perhitungan tidak dapat dilakukan. Karena hal itu, maka penelitian ini hanya menggunakan beberapa set data yang telah dipublikasikan.

Problem data tersebut telah digunakan oleh beberapa publikasi untuk melakukan penelitian dan optimasi. Daftar publikasi yang telah melakukan pengolahan data terhadap set data tersebut dapat dilihat pada Tabel 3.3.

Tabel 3.3 Daftar Publikasi yang Melakukan Pengolahan Data terhadap Set Data

No	Problem	Metode				
		TAA for Single Row	***TAA for Double Row***	**TAA *for Cluster* MLP**	***Coevolu-tionary Algorithm***	**HOT**
1	KH1	Heragu and Kusiak. (1988)				
2	KH2	Heragu and Kusiak. (1988)				
3	KH3	Heragu and Kusiak. (1988)				
4	KH4	Heragu and Kusiak. (1988)				
5	KH5	Heragu and Kusiak. (1988)				
6	KH6	Heragu and Kusiak. (1988)				
7	KH7	Heragu and Kusiak. (1988)				
8	KH8	Heragu and Kusiak. (1988)				
9	KH9	Heragu and Kusiak. (1988)				
10	NUG5	Heragu and Kusiak. (1988)	Heragu and Kusiak. (1988)			
11	NUG6	Heragu and Kusiak. (1988)	Heragu and Kusiak. (1988)			
12	NUG7	Heragu and Kusiak. (1988)	Heragu and Kusiak. (1988)			
13	NUG8	Heragu and Kusiak. (1988)	Heragu and Kusiak. (1988)			
14	NUG12	Heragu and Kusiak. (1988)	Heragu and Kusiak. (1988)	Heragu and Kusiak. (1988)		
15	NUG15	Heragu and Kusiak. (1988)	Heragu and Kusiak. (1988)	Heragu and Kusiak. (1988)		
16	NUG20	Heragu and Kusiak. (1988)	Heragu and Kusiak. (1988)	Heragu and Kusiak. (1988)		
17	OPT50					Mir and Imam (2001)
18	DUN62				Dunker, et al. (2003)	

Hasil dari publikasi di atas adalah hasil yang akan digunakan sebagai perbandingan pada pengolahan data. Selain dari data tersebut di atas, ada juga penggunaan *software* VIP-PLANOPT, di mana data berlabel OPT diambil dari benchmark problem *software* tersebut.

Publikasi-publikasi tersebut telah melakukan pengolahan data dan menghasilkan solusi yang bisa dilihat pada Tabel 3.4.

Tabel 3.4 Solusi Publikasi Terdahulu

No	Problem	Metode					
		TAA Single Row	***TAA Double Row***	**TAA *Cluster***	***Coevoluti-onary Algorithm***	**PLANOPT**	**HOT**
1	KH1	225					
2	KH2	440					
3	KH3	510					
4	KH4	465					
5	KH5	19.68					
6	KH6	359					
7	KH7	318					
8	KH8	60					
9	KH9	244					
10	NUG5	1.165	1.14				
11	NUG6	2.085	2.01				
12	NUG7	5.42	3.98				
13	NUG8	7.995	4.95				
14	NUG12	31.525	17.91	15.77			
15	NUG15	62.624	34.98	29.09			
16	NUG20	178.149	91.47	70.86			
17	OPT8					213.5	
18	OPT12					5384.43	
19	OPT20					1157	
20	OPT28					6447.25	
21	OPT50					78224.68	80794.24
22	OPT75					34396.38	
23	DUN62				4221911		

BAB 4
PENGOLAHAN DATA DAN ANALISA

4.1 Pengolahan Data

Berdasarkan kumpulan data yang ada, akan dilakukan pembuatan model untuk menghitung semua data set yang ada. Sesuai tujuan penelitian, yaitu untuk meminimalkan biaya pemindahan material, maka perlu dilakukan pencarian posisi terbaik dan perhitungan jarak antar mesin yang akan menjadi variabel dalam perhitungan ini. Perhitungan jarak ini yang akan menjadi dasar dalam pehitungan biaya pemindahan material.

4.1.1 Penyusunan Algoritma

Penyelesaian masalah MLP ini menggunakan algoritma *Differential Evolution* (DE). Untuk melakukan pencarian solusi optimal, digunakan bahasa pemrograman Java.

Pembuatan program optimasi MLP menggunakan algoritma DE dengan Java ini didasarkan pada fungsi objektif meminimumkan biaya pemindahan material. Maka, *output* yang diharapkan dari program ini berupa posisi koordinat mesin yang meminimalkan biaya pemindahan material.

Berikut ini adalah langkah-langkah pembuatan program berdasarkan algoritma DE.

1. Penetapan Parameter Kontrol

Ada tiga parameter kontrol yang ditetapkan nilainya, yaitu ukuran populasi, parameter kontrol mutasi, dan parameter kontrol pindah silang. Ukuran populasi merupakan parameter untuk menentukan jumlah solusi awal. Ukuran populasi ditentukan berdasarkan jumlah dimensi yang telah ditentukan. Jumlah dimensi ditentukan berdasarkan jumlah variabel yang ada yang digunakan untuk menentukan nilai optimal.

Pada permasalahan ini, variabel yang menentukan atau jumlah dimensi adalah jumlah mesin yang akan ditentukan koordinatnya. Untuk setiap mesin, dibutuhkan dua angka, yaitu untuk koordinat x dan y. Oleh sebab itu, ukuran

dimensi merupakan jumlah mesin dikali dua. Misalnya, untuk set data problem dengan jumlah mesin 8, maka dimensinya adalah 16.

Ukuran populasi sendiri umumnya ditentukan sebesar 10 x dimensi untuk mendapat hasil yang optimal. Karena itu, untuk kasus dengan jumlah mesin 8, populasinya adalah 160. Namun, aturan untuk ukuran populasi ini tidak mutlak. Jika proses pencarian mengalami kesulitan, maka jumlah ukuran populasi dapat dinaikkan.

Nilai parameter mutasi (F) sebaiknya tidak terlalu kecil untuk menghindari populasi konvergen sebelum mencapai mininum. Sebaliknya, F juga tidak boleh terlalu besar karena jumlah evaluasi fungsi akan bertambah seiring dengan meningkatnya F. Nilai parameter *crossover* (Cr) juga tidak boleh terlalu kecil karena akan mengurangi keragaman dan dapat menyebabkan pencarian berhenti. Nilai Cr juga tidak boleh terlalu besar untuk menghindari gangguan terlalu besar dan kecepatan keragaman berkurang.

Nilai F dan Cr terbaik ditetapkan sebesar 0.4 dan 0.5. Hal ini dapat dibuktikan dengan percobaan menggunakan sampel nilai F dan Cr. Percobaan dilakukan dengan untuk nilai yaitu antar 0.4-0.9 dan nilai Cr yaitu 0.3 – 0.9. Percobaan dilakukan untuk set data problem OPT08. Pemilihan ini dilakukan berdasarkan problem yang memiliki ukuran tidak terlalu kecil, sehingga hasilnya tidak terlalu bervariasi untuk setiap parameter. Dan juga tidak terlalu besar yang membutuhkan waktu terlalu lama untuk percobaan. Hasil percobaan menggunakan parameter F dan Cr yang berbeda dapat dilihat pada tabel di bawah ini.

Dilihat dari tabel, parameter F=0.4 dan Cr=0.5 memberikan hasil yang paling baik menurut fungsi objektif, yaitu minimum biaya pemindahan material, sehingga hasil yang paling kecil adalah yang paling baik. Oleh karena itu, parameter F sebesar 0.4 dan Cr sebesar 0.5 disimpulkan memberikan hasil yang paling baik dan akan dipakai dalam pengolahan data.

Tabel 4.1. Hasil Percobaan Nilai F dan Cr

F	Cr	Hasil	F	Cr	Hasil
0.4	0.3	213.01	0.7	0.3	258.7
	0.4	231		0.4	247.2
	0.5	193		0.5	329.8
	0.6	210.5		0.6	205.3
	0.7	214		0.7	193
	0.8	252.5		0.8	211.5
	0.9	456.4		0.9	219
0.5	0.3	204.3	0.8	0.3	365.5
	0.4	221		0.4	356.1
	0.5	220		0.5	365.6
	0.6	210		0.6	341.6
	0.7	205.5		0.7	349.9
	0.8	222		0.8	198
	0.9	232		0.9	223.5
0.6	0.3	245.32	0.9	0.3	374.9
	0.4	220.5		0.4	429.2
	0.5	193.5		0.5	454.12
	0.6	213.5		0.6	440.65
	0.7	227.5		0.7	365.4
	0.8	217		0.8	212
	0.9	202		0.9	212

Selain 3 parameter kontrol diatas, pada tahap ini juga ditentukan kriteria terminasi program. Kriteria terminasi dapat ditentukan berdasarkan jumlah iterasi maksimum, waktu proses ataupun ketika hasil optimum sudah tidak berubah (unimproved). Pada kasus ini, kriteria yang digunakan adalah jumlah iterasi maksimum. Program akan mencapai terminasi apabila telah melakukan 40000 iterasi atau telah mencapai 5000 iterasi yang tidak ada peningkatan. Terminasi ditentukan berdasarkan yang paling lebih dahulu dicapai dari kedua kriteria tersebut. Pertimbangan jumlah iterasi didasarkan pada kemungkinan untuk mendapat solusi yang paling baik.

2. Menentukan Populasi Awal

Langkah selanjutnya adalah membentuk populasi awal. Populasi awal merupakan sebuah matriks yang berisi sejumlah individu awal. Untuk kasus MLP, setiap individu awal akan merepresentasikan letak titik koordinat mesin. Populasi awal ditetapkan secara acak.

Pada proses ini bilangan acak yang digunakan memiliki *range* masing-masing. *Range* ini direpresentasikan dalam program Java sehingga dapat diubah secara dinamis. *Range* 100 menunjukkan bilangan acak yang diinisialisasi memiliki batas bawah -100 dan batas atas 100. Batas ini merupakan batas koordinat awal mesin, sehingga letak mesin tidak terlalu jauh dan tetap dalam batasan untuk menemukan hasil yang *feasible.*

3. Menentukan Fungsi objektif

Fungsi objektif adalah meminimumkan biaya pemindahan material sesuai dengan konstanta *material flow* dan variabel jarak mesin, yaitu:

$$\min \sum_{i=1}^{n} \quad \sum_{j=1}^{n} \quad c_{ij} f_{ij}\, d_{ij} + p_{ij} \tag{4.1}$$

dengan,

n = jumlah mesin

c_{ij} = biaya pemindahan material per unit jarak dari mesin i ke mesin j.

f_{ij} = jumlah aliran material dari mesin i ke mesin j

d_{ij} = jarak antara mesin i ke mesin j.

p_{ij} = penalti antara mesin i dan j.

f_{ij} atau jumlah aliran material dari mesin i ke mesin j didapat dari data problem yang telah ada. Susunan dari aliran material ini bervariasi dalam setiap set data untuk menguji kemampuan program menyelesaikan masalah yang bervariasi.

Dalam perhitungan ini, c_{ij} atau biaya pemindahan material per unit jarak dari mesin i ke j diasumsikan sebanding dengan jarak antar mesin atau sebesar 1$/unit/jarak.

Di dalam perhitungan jarak, ada beberapa formulasi yang dapat digunakan, antara lain secara euclidean atau *rectilinear*. Untuk penelitian ini, jarak antar mesin dihitung secara *rectilinear* dengan rumus sebagai berikut:

$$d_{ij} = |x_i - x_j| + |y_i - y_j| \quad (4.2)$$

d_{ij} = jarak antara mesin i ke mesin j.

x_i = posisi koordinat x mesin i

y_i = posisi koordinat y mesin i

Sebagai kendala, ada penalti yang dihitung apabila terjadi tumpang tindih antar mesin. Penalti dapat dituliskan dalam rumus sebagai berikut.

$$p_{ij} = \frac{(intx_{ij}\ x\ inty_{ij})}{L}\ x\ M \quad (4.3)$$

p_{ij} = penalti antara mesin i dan j

$intx_{ij}$ = irisan panjang antara mesin i dan j

$inty_{ij}$ = irisan lebar antara mesin i dan j

L = total luas semua mesin

M = angka yang sangat besar.

Irisan akan terjadi apabila ada tumpang tindih diantara mesin. Untuk melihat apakah ada tumpang tindih atau tidak, dilakukan pengecekan dengan formula berikut:

jika,

$$|x_i - x_j| < \frac{(l_i + l_j)}{2} \quad (4.4)$$

dan

$$|y_i - y_j| < \frac{(w_i + w_j)}{2} \quad (4.5)$$

maka,

$$intx_{ij} = \frac{(l_i + l_j)}{2} - |x_i - x_j| \quad (4.6)$$

$$inty_{ij} = \frac{(w_i + w_j)}{2} - |y_i - y_j| \quad (4.7)$$

x_i = posisi koordinat x mesin i

y_i = posisi koordinat y mesin i

l_i = panjang mesin i

w_i = lebar mesin i

Syarat terjadinya irisan adalah bahwa kedua syarat (4.4) dan (4.5) harus terjadi. Jika salah satu tidak terjadi, maka tidak terjadi irisan dan $intx_{ij}$ dan $inty_{ij}$ bernilai 0.

4. Evaluasi Fungsi Objektif

Setiap individu awal dievaluasi dengan menggunakan fungsi objektif di atas atau penentuan solusi awal. Solusi awal berperan sebagai vektor target atau vektor orang tua (Vektor A). Solusi awal ditentukan dengan menggunakan pendekatan SPV (*Smallest Position Value*), yaitu mencari nilai fungsi objektif yang paling kecil dari populasi yang ada. Pendekatan ini dilakukan supaya hasil optimal yang didapat dari algoritma *Differential Evolution* lebih terjamin untuk mendekati solusi optimal global.

5. Proses Mutasi

Proses mutasi bertujuan untuk membuat individu baru yang disebut individu mutan. Secara teori, individu ini merupakan individu awal yang mengalami perubahan nilai pada dimensinya. Oleh karena itu, untuk membuatnya, harus diambil 3 individu acak dari populasi awal. Berikut ini adalah rumus untuk membentuk individu mutan:

$$Individu\ mutan = a + (b-c) * F \quad (4.8)$$

F adalah parameter mutasi yang telah ditentukan pada langkah 1.

6. Pindah Silang

Proses pindah silang bertujuan untuk membentuk individu baru yaitu individu *trial*. Parameter atau nilai dimensi individu *trial* ini sebagian berasal dari parameter individu target dan sebagian lagi berasal dari individu mutan, dengan mempertimbangkan operator pindah silang (Cr) dan bilangan acak.

Jika bilangan acak r (antara 0 sampai 1) yang dihasilkan lebih kecil atau sama nilainya dengan Cr maka yang berpeluang menjadi nilai dimensi ke-i individu *trial* adalah nilai dimensi ke-i individu mutan. Bila hal sebaliknya yang terjadi, maka nilai dimensi ke-i individu *trial* adalah nilai dimensi ke-i individu awal.

7. Evaluasi

Setiap individu pada populasi *trial* akan dievaluasi seperti yang dilakukan pada individu pada populasi awal. Individu dengan nilai jarak paling kecil selanjutnya masuk ke proses seleksi.

8. Proses Penyeleksian

Proses ini merupakan tahap untuk menentukan individu yang layak masuk ke anggota generasi berikutnya, yaitu dengan cara membandingkan nilai fungsi objektif individu target dengan individu *trial*. Untuk kasus ini, individu dengan jarak terkecil yang akan terpilih.

9. Terminasi

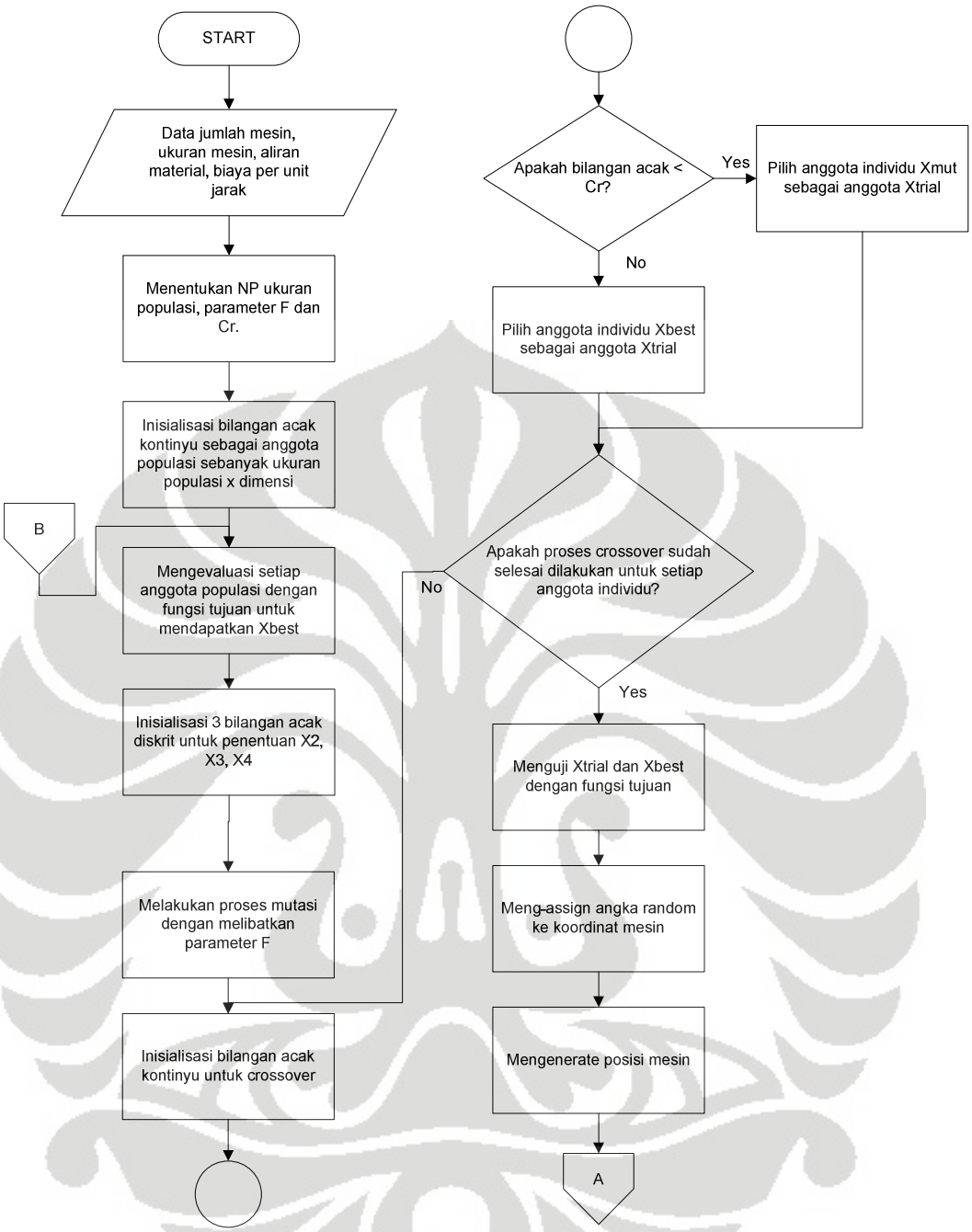
Tahap ke-2 hingga 8 merupakan proses untuk 1 kali iterasi. Proses ini akan berulang terus sampai kriteria terminasi tercapai. Program selanjutnya akan memilih individu dengan jarak terkecil dari iterasi yang telah dilakukan.

4.1.2 Pembuatan Model

4.1.2.1 Pembuatan Model dengan Java

Java merupakan salah satu produk dari Sun Systems untuk membuat program. Java sangat luas dipakai, baik untuk program web ataupun mobile. Keunggulan dari Java adalah Java merupakan program *open source*, sehingga program yang sudah ada dapat dipergunakan, dimodifikasi, dan dikembangkan secara luas. Baik pengolahan data ataupun input data pada penelitian ini dilakukan di dalam pengkodean model Java.

Selain karena merupakan *open source*, Java juga dipilih karena merupakan pemrograman berdasarkan objek sehingga memudahkan dalam modifikasi program. Selaini itu, Java juga saat ini sudah digunakan secara luas untuk berbagai aplikasi sehingga memudahkan dalam penggunaannya.



Gambar 4.1. Diagram Alir Optimasi MLP Menggunakan DE

Gambar 4.1. (sambungan)

Untuk perhitungan dalam penelitian ini, digunakan program dasar DE dengan *source code* Java yang dikembangkan oleh pengembang DE sendiri, yaitu Rainer Storn[2] dengan penambahan modifikasi yang cukup banyak. Seperti yang dikemukakan dalam manual penggunaan program ini, Storn bukan hanya memperbolehkan, namun sangat mendorong untuk program ini digunakan dan dimodifikasi.

Dengan menggunakan program ini, hasil yang didapat merupakan hasil solusi peletakan koordinat mesin yang menghasilkan biaya pemindahan material terkecil. Program ini juga memungkinkan penggunaan parameter yang dinamis dan dapat diubah untuk mendapatkan hasil yang optimal.

Model ini secara garis besar bisa dibagi dalam 3 bagian utama:

- *Interface*

  Interface adalah tampilan luar dari program, yaitu bagian yang bersentuhan langsung dengan pengguna. Bagian interface ini memiliki panel-panel untuk memasukkan input data berupa parameter yang ingin digunakan dalam perhitungan, pemilihan problem dan straetegi, penjelasan mengenai status model dan status solusi.

- *Differential Evolution Optimizer*

  Bagian ini adalah bagian utama dari model ini, yaitu perhitungan formulasi *Differential Evolution*. Di bagian ini, langkah-langkah algoritma DE dipresentasikan. Terdapat pula perhitungan DE dengan beberapa strategi.

- Problem/ fungsi tujuan

  Ini adalah bagian yang dikembangkan sendiri, yaitu pemasukan fungsi tujuan untuk MLP. Selain itu, juga dilakukan input data untuk masing-masing data set.

[2] Storn, Rainer. DeApp – An Application in Java for the Usage of *Differential Evolution*. International Computer Science Institute.

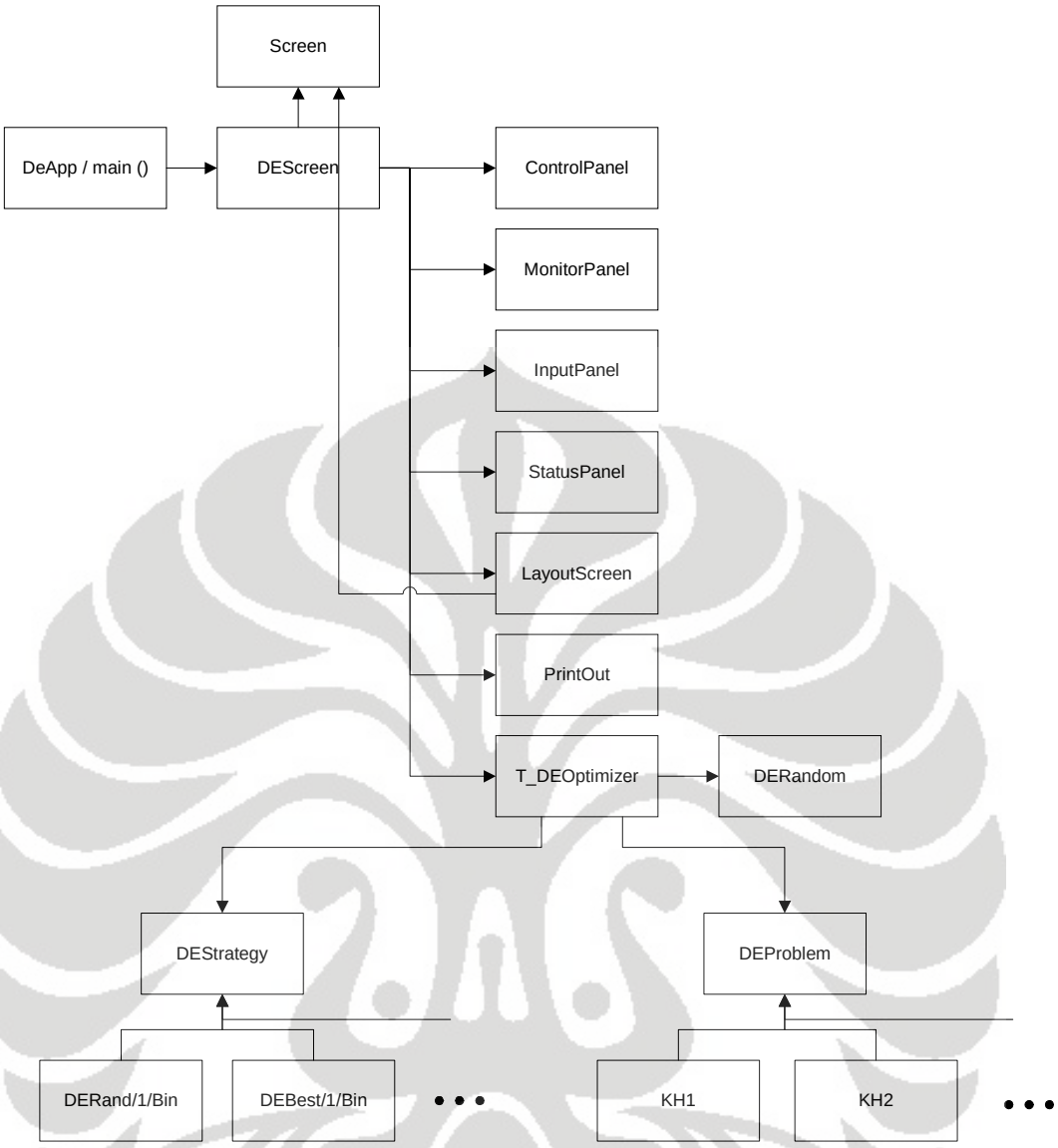
4.1.2.2 Struktur Kelas dalam Model Menggunakan Java

Java memiliki kelas-kelas untuk memudahkan dalam pengembangan model berdasarkan objek. Berikut adalah daftar kelas yang digunakan dan penjelasan singkat mengenai kelas tersebut.

Tabel 4.2 Struktur Kelas Model Menggunakan Java

Package	Kelas	Keterangan
Main	DEApp	Kelas utama untuk menjalankan program dalam Java
DE	T_DEOptimizer	Merupakan kelas di mana dilakukan optimasi DE sesuai dengan strategi yang dipilih
	DERandom	Kelas untuk menghasilkan angka random
	DEStrategy	Kelas untuk melakukan pemilihan strategi DE yang digunakan
	DEBest1Bin	Strategi-strategi DE yang merupakan anakan dari kelas DEStrategy yang mengandung formula yang berbeda untuk setiap strategi
	DERan1Bin	
	…	
Problem	DEProblem	Kelas untuk memilih problem / data set yang digunakan
	KH1	Berisi data dan fungsi tujuan dari data set yang ada.
	KH2	
	…	
Panel	ControlPanel	Kelas untuk melakukan pemilihan problem dan strategi dari user
	InpuPanel	Kelas untuk memasukkan parameter dari user
	MonitorPanel	Menampilkan hasil dari perhitungan
	StatusPanel	Menampilkan status running program
Screen	DEScreen	Kelas untuk menampung semua data grafik yang ditampilkan
	*Layout*Screen	Kelas untuk membuat tampilan *layout* akhir
	PrintOut	Untuk mencetak hasil solusi koordinat ke individual file.
	Screen	Kelas untuk membangun tampilan grafik.

Kelas satu dan yang lain terhubung oleh aliran informasi yang dialirkan dari satu kelas ke kelas lain. Gambaran struktur kelas dalam program ini adalah sebagai berkut:

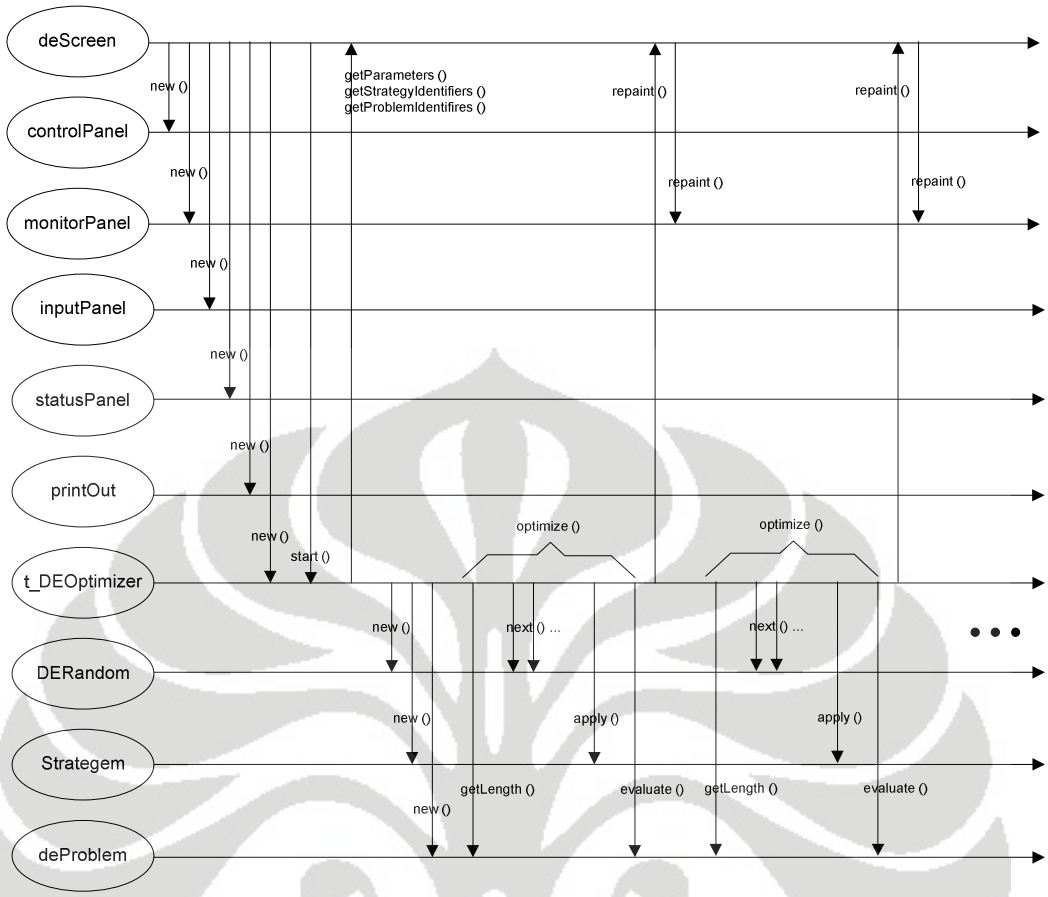


Gambar 4.2 Hubungan Antar Kelas dalam Model Menggunakan Java

Tanda panah di atas menunjukkan hubungan antara kelas, di mana informasi dari kelas tersebut dialirkan ke kelas lain sesuai tanda panah.

4.1.2.3 Penyusunan Algoritma dalam Model Menggunakan Java

Dalam penyusunan algoritma sesuai dengan tujuan penelitian, dibutuhkan metode-metode yang sesuai. Secara garis besar, algoritma dalam penyusunan program ini dapat digambarkan dalam grafik berikut.



Gambar 4.3 Algoritma dalam Model Menggunakan Java

Tanda panah ke kanan menunjukkan aliran waktu, yang seiring berjalannya waktu, akan dilakukan eksekusi metode sesuai dengan tanda panah vertikal. Tanda panah vertikal menunjukkan aliran cara kerja program yang akan terus dilaksanakan sampai kriteria terminasi terpenuhi.

Untuk mempermudah penggunaan dan menambah utilitas program, maka pada program ini ditambahkan pengaturan-pengaturan, sehingga bisa menampilkan data permintaan secara lebih sistematis dan terintegrasi dengan program optimasi DE tersebut. Beberapa penambahan fungsi dari program ini adalah:

- Pemilihan strategi

  Fungsi ini digunakan untuk memilih beberapa strategi DE yang ada. Tapi, untuk perhitungan pada penelitian ini, yang akan digunakan hanyalah Best1Bin. Fungsi ini dikembangkan oleh sumber kode DE, yaitu Storn. Fungsi ini dapat digunakan untuk

melihat perbandingan beberapa strategi DE untuk mencari solusi dari MLP.

- Fungsi *Refresh*

  Fungsi ini adalah fungsi grafis yang menentukan seberapa sering tampilan diupdate sesuai dengan iterasi. Misalnya, untuk angka *refresh* 1, maka untuk setiap iterasi, akan ditampilkan hasil minimumnya. Tapi untuk angka *refresh* 10, maka hasil baru akan diupdate setiap kelipatan 10 iterasi. Fungsi ini berguna untuk mempercepat waktu running program. Semakin besar angka *refresh*, maka waktu perhitungan akan semakin cepat, karena program tidak perlu melakukan perintah menggambar tampilan setiap saat.

- *Draw*

  Fungsi ini digunakan untuk menampilkan hasil *layout* akhir.

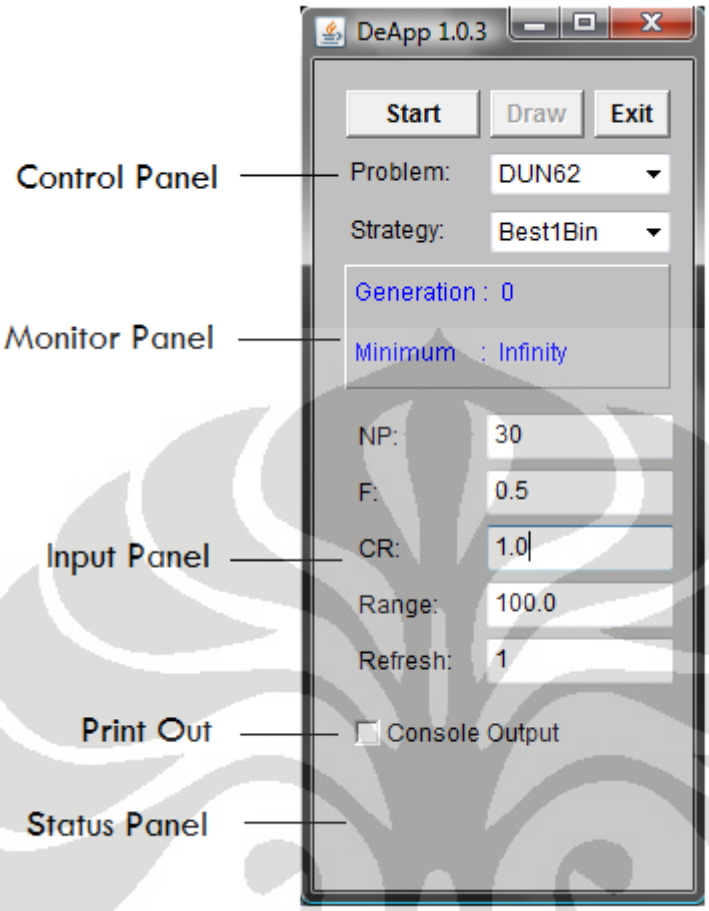
- *Console output*

  Fungsi ini digunakan untuk mengeluarkan hasil solusi koordinat optimum. Hasil solusi ini akan dikeluarkan ke dalam satu *file output*.

Manfaat pengaturan tampilan program ini sangat besar, terutama pada fungsi *draw*, yang memudahkan pengguna melihat tampilan hasil dari perhitungan.

4.1.2.4 Tampilan Program

Adapun tampilan program ketika pertama kali dieksekusi adalah seperti pada gambar 4.4.



Gambar 4.4 Tampilan Model Menggunakan Java

4.1.3 Verifikasi dan Validasi Program

Sebelum program digunakan untuk menganalisa permasalahan MLP ini, terlebih dahulu harus dilakukan verifikasi dan validasi program. Tahap verifikasi merupakan tahap melihat kesesuian antara model program yang didapat dengan konseptual model yang kita buat/inginkan. Parameter model program dikatakan telah terverifikasi apabila telah berjalan sesuai konseptual model**.** Beberapa proses verifikasi yang dilakukan dalam proses pembuatan program ini antara lain adalah sebagai berikut:

- Memastikan kebenaran logika pemikiran, yaitu kesesuaian penulisan *source code* dengan konsep algoritma *Differential Evolution* dan dapat dijalankan dengan baik
- Memastikan bahwa program sudah memasukkan semua batasan masalah yang ada

- Memastikan fungsi tujuan
- Memastikan ada perubahan *output* jika parameter diubah (pengujian sensitivitas).

Setelah proses verifikasi, selanjutnya adalah proses validasi. Validasi dilakukan unuk dapat memastikan bahwa program menghasilkan *output* yang benar sesuai fungsinya. Artinya, hasil perhitungan yang dihasilkan oleh program bernilai sama dengan perhitungan manual.

Bagian yang divalidasi dari program adalah bagian fungsi tujuan. Untuk bagian optimasi DE sudah tervalidasi oleh pengembang, Storn. Untuk fungsi tujuan, perlu dilakukan validasi untuk mengecek kevalidan perhitungan.

Validasi dilakukan dengan menghitung satu set data, yaitu KH1 dengan 4 mesin. Adapun parameter yang digunakan dalam perhitungan ini sama dengan paramater yang digunakan pada pengolahan data yang terdapat dalam tabel 4.3:

Tabel 4.3 Parameter untuk Validasi Program

Iterasi Maksimum	40000
NP	100
F	0.4
Cr	0.5

Adapun data yang digunakan dalam problem ini adalah set Data KH1 seperti ada di tabel 4.4:

Tabel 4.4 Ukuran Mesin untuk Validasi Program

Mesin	Panjang	Lebar
1	2	2
2	4	4
3	6	6
4	2	2

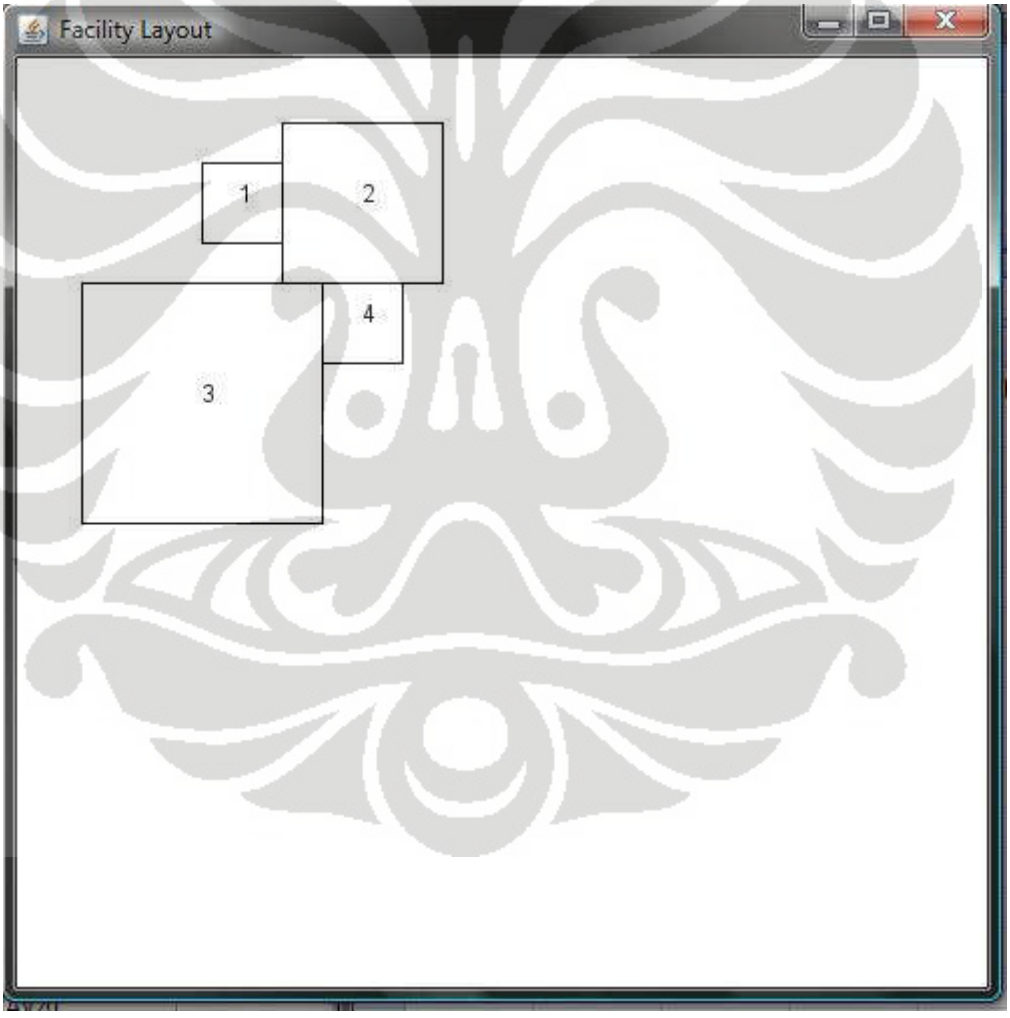
Tabel 4.5 Aliran Material untuk Validasi Program

dari/ke	1	2	3	4
1	0	3	6	6
2	3	0	9	3
3	6	9	0	6
4	6	3	6	0

Hasil run program menunjukkan bahwa solusi optimal menghasilkan biaya *material handling* sebesar 168 dengan konfigurasi koordinat mesin sebagai berikut:

Tabel 4.6 Solusi Koordinat Mesin untuk Validasi Program

Mesin	Koordinat X	Koordinat Y
1	-9.728833758	-55.60367002
2	-6.728833758	-55.60367002
3	-10.72883376	-50.60367002
4	-6.728833758	-52.60367002



Gambar 4.5 Tampilan *Layout* Model Java untuk Validasi Program

Selanjutnya, perlu dilakukan perhitungan manual untuk membuktikan bahwa hasil biaya *material handling* yang diperoleh dari konfigurasi ini benar. Langkah-langkah yang dilakukan adalah sebagai berikut:

1. Menghitung Jarak Antar Mesin

   Untuk menghitung biaya pemindahan material, dibutuhkan data jarak antar mesin yang didapat dari posisi mesin. Jarak antar mesin dihitung secara *rectilinear* menurut persamaan (4.2). Berikut ini adalah matriks perhitungan jarak antar mesin.

Tabel 4.7 Jarak Antar Mesin untuk Validasi Program secara Manual

dari/ke	1	2	3	4
1	0	3	6	6
2	3	0	9	3
3	6	9	0	6
4	6	3	6	0

2. Perhitungan biaya pemindahan material

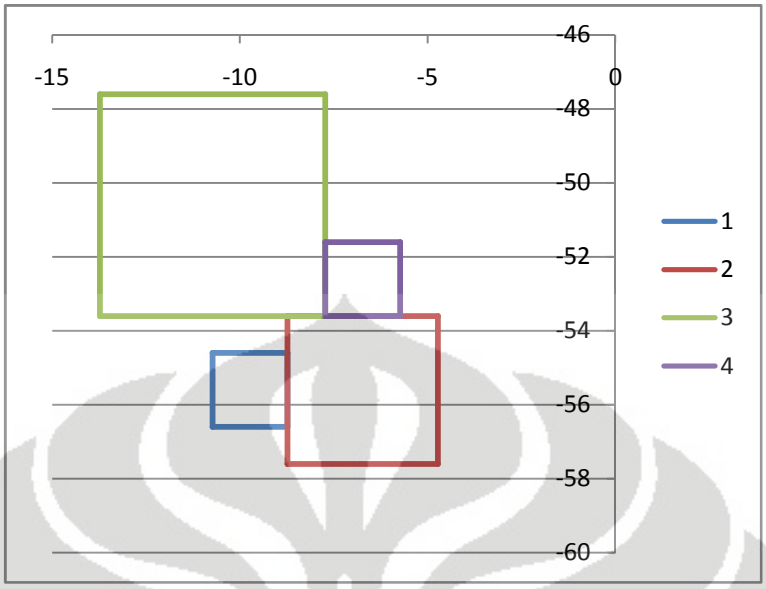
   Selanjutnya dilakukan perkalian antara jarak dengan biaya pemindahan material yang didapat dari data. Hasil dari perkalian antara kedua matriks di atas akan menghasilkan total biaya pemindahan material sebagai berikut:

Tabel 4.8 Total Biaya Pemindahan Material untuk Validasi Program secara Manual

	1	2	3	4
1	0	30	30	0
2	0	0	0	60
3	0	0	0	48
4	0	0	0	0
Total biaya pemindahan material				168

Dari perhitungan di atas, dapat dilihat bahwa hasil perhitungan manual telah sama dengan hasil run program. Lebih jauh lagi, untuk memastikan kesamaan gambar solusi koordinat mesin yang dihasilkan dari program, dilakukan pembuatan grafik secara manual dengan program Microsoft Excel.

Hasil dari grafik itu adalah sebagai berikut.



Gambar 4.5 Tampilan *Layout* Model Manual untuk Validasi Program

Hasil diatas hanya berbeda posisi rotasi saja dengan hasil dari program Java, karena untuk program grafis Java, titik (0,0) dihitung dari kiri atas, sedangkan untuk excel, seperti grafik pada umumnya, titik (0,0) dihitung dari kiri bawah. Tetapi, hal tersebut tidak berpengaruh kepada solusi secara keseluruhan. Dengan demikian program telah tervalidasi.

4.2 Pengolahan Data dan Hasil

Setelah program siap, dilakukan pengolahan data untuk 23 problem satu per satu. Pengolahan data dilakukan dengan perangkat keras komputer 1.6 GHz, 1G RAM.

Pengaturan parameter untuk mengevaluasi algoritma yang digunakan pada pengolahan data adalah sebagai berikut:

Tabel 4.9 Parameter untuk Pengolahan Data

Maksimum Iterasi	40000
F (parameter mutasi)	0.4
Cr (parameter *crossover*)	0.5

Pengolahan data dilakukan masing-masing 5 kali untuk setiap problem kecuali untuk problem OPT28, OPT50, OPT75, dan NUG62 yang masing-masing dilakukan sebanyak 3 kali. Replikasi perlu dilakukan karena DE menggunakan angka random yang bisa menghasilkan solusi yang berbeda untuk setiap percobaan. Replikasi memastikan bahwa yang hasil yang diperoleh adalah solusi dari optimum global, bukan optimum lokal. Dari replikasi itu, dipilih hasil yang paling minimum untuk setiap problem.

Tabel 4.10 Hasil Replikasi Pengolahan Data

No	Problem	Replikasi					Minimum
		1	2	3	4	5	
1	KH1	168	168	168	168	172	168
2	KH2	335	335	335	335	335	335
3	KH3	360	400	400	360	360	360
4	KH4	370	335	335	340	340	335
5	KH5	16.2	15.2	16.2	15.2	14.6	14.6
6	KH6	267	271	269	267	271	267
7	KH7	250	248	248	248	248	248
8	KH8	42	44	44	44	42	42
9	KH9	192	192	192	192	192	192
10	NUG5	0.85	0.77	0.82	0.77	0.77	0.77
11	NUG6	1.525	1.355	1.49	1.355	1.44	1.355
12	NUG7	2.79	2.77	2.65	2.64	3.24	2.64
13	NUG8	4.92	5.06	4.87	4.367	4.97	4.367
14	NUG12	15.76	17.84	13.28	13.09	15.44	13.09
15	NUG15	30.32	21.09	21.97	25.39	22.5	21.09
16	NUG20	58.39	54.68	51.33	60.33	60.55	51.33
17	OPT8	222.5	198	193.5	213	192.5	192.5
18	OPT12	2924.346	3476.13	2918.98	3022.6	2968.72	2918.98
19	OPT20	1457.96	1325.88	1500.33	1487.2	1449.9	1325.88
20	OPT28	1801.634	2065.47	1752.206			1752.206
21	OPT50	155650.1	144192.3	134939			134939
22	OPT75	64204.16	56955.59	55560.25			55560.25
23	DUN62	5195260	6837126	5283572.5			5195260

Bisa dilihat bahwa semakin besar dimensi problem, semakin banyak pula kombinasi susunan mesin yang terjadi dan menghasilkan nilai optimum lokal yang berbeda-beda pula.

Setelah program dijalankan, didapat hasil dari pengolahan data, yaitu biaya pemindahan material yang minimum. Selain itu, juga bisa dilihat hasil konfigurasi koordinat mesin yang menghasilkan biaya pemindahan material yang minimum. Hasil titik koordinat dan gambar susunan *layout* akhirnya dapat dilihat pada bagian Lampiran. Adapun hasil dari pengolahan data berupa biaya pemindahan material yang paling kecil setelah replikasi dapat dilihat pada Tabel 4.11.

Tabel 4.11 Hasil Pengolahan Data Menggunakan DE

No	Problem	NP	Waktu (s)	Iterasi	Minimum
1	KH1	100	6	3510	168
2	KH2	100	6	2820	335
3	KH3	100	6	3340	360
4	KH4	100	6	3102	335
5	KH5	100	8	4594	14.6
6	KH6	100	8	3484	267
7	KH7	100	10	5020	248
8	KH8	100	8	4150	42
9	KH9	100	7	4140	192
10	NUG5	100	9	4760	0.77
11	NUG6	100	15	5420	1.355
12	NUG7	100	12	4260	2.64
13	NUG8	100	13	5070	4.367
14	NUG12	100	17	4309	13.09
15	NUG15	150	43	5530	21.09
16	NUG20	200	63	7340	51.33
17	OPT8	100	62	4336	192.5
18	OPT12	150	24	5310	2918.98
19	OPT20	300	190	10660	1325.88
20	OPT28	280	850	15170	1752.206
21	OPT50	300	2160	10990	134939
22	OPT75	400	14400	72720	55560.25
23	DUN62	400	14400	87310	5195260

Untuk menguji kualitas algoritma DE yang disarankan untuk memecahkan kasus MLP, dilakukan perbandingan hasil yang diperoleh dengan penelitian yang telah dilakukan dengan data set yang sama.

Set data KH1-KH9 sebelumnya diselesaikan oleh Kusiak dan Heragu menggunakan metode *single-row*. Selain itu, Kusiak dan Heragu pun

menyelesaikan set data NUG5-NUG8 dengan *single row* dan *double row*. Berikut adalah tabel hasil pengolahan datanya.

Tabel 4.12 Hasil Pengolahan Data Single-Row, Double-Row, dan DE

No	Data Set	TAA Single Row	TAA Double Row	DE
1	KH1	225		168
2	KH2	440		335
3	KH3	510		360
4	KH4	465		335
5	KH5	19.68		14.6
6	KH6	359		267
7	KH7	318		248
8	KH8	60		42
9	KH9	244		192
10	NUG5	1.165	1.14	0.77
11	NUG6	2.085	2.01	1.355
12	NUG7	5.42	3.98	2.64
13	NUG8	7.995	4.95	4.367

Untuk set data yang lain, dilakukan perbandingan hasil dengan cara membandingkan selisih optimum *cost* dengan optimum *cost* yang terbaik dari penelitian-penelitian sebelumnya. Hasil perbandingan dengan angka negatif menunjukkan bahwa hasil percobaan dengan DE lebih buruk dari penelitian sebelumnya.

4.3 Analisa Pengolahan Data

4.3.1 Analisa Hasil

- KH1-KH9

  Problem ini diambil dari Kusiak dan Heragu yang menyelesaikan dengan metode TAA (*Triangle Assignment Algorithm*) *single-row*. Metode ini sebenarnya memiliki konsep penyelesaian yang berbeda dengan optimasi representasi kontinyu. Data ini dipakai untuk membuktikan bahwa model representasi kontinyu dengan DE dapat digunakan untuk menyelesaikan problem kecil.

Tabel 4.13 Tabel Perbandingan DE dengan Metode Lain

No	Problem	Metode						Best known	DE	%
		TAA Single Row	TAA Double Row	TAA Cluster	Coevolu-tionary Algorithm	PLANOPT	HOT			
1	NUG12	31.525	17.91	15.77				15.77	13.09	16.99%
2	NUG15	62.624	34.98	29.09				29.09	21.09	27.50%
3	NUG20	178.149	91.47	70.86				70.86	51.33	27.56%
4	OPT8					213.5		213.5	192.5	9.84%
5	OPT12					5384.43		5384.43	2918.98	45.79%
6	OPT20					1157		1157	1325.88	-14.6%
7	OPT28					6447.25		6447.25	1752.206	72.82%
8	OPT50					78224.68	80794.24	78224.7	134939.0	-72.5%
9	OPT75					34396.38		34396.7	55560.25	-61.5%
10	DUN62				4221911			4221911	5195260	-23.1%

Dari hasil pengolahan data, terlihat bahwa hasil yang diperoleh DE sangat baik untuk menyelesaikan problem ini. Waktu yang digunakan untuk perhitungan pun sangat cepat. Sehingga dapat disimpulkan bahwa DE dapat digunakan untuk problem set ini.

- NUG5-NUG8

  Data untuk perhitungan problem ini diambil dari Nugent, 1968. Data ini juga digunakan oleh Kusiak dan Heragu untuk diuji menggunakan TAA *single-row* dan *double-row*. Menurut jurnal tersebut, data lebih baik didapatkan oleh pengaturan *double-row*. Sama seperti *single-row*, *double-row* juga merupakan metode tradisional yang tidak bisa diperbandingkan dengan algoritma DE. Percobaan dilakukan untuk melihat apakah DE bisa diaplikasian untuk menyelesaikan problem dengan jumlah mesin ukuran sedang, yaitu 5-8 mesin. Dari hasil pengolahan data, terlihat bahwa DE memberikan solusi yang optimal untuk masalah dengan jumlah mesin 5-8 mesin.

- NUG12-NUG20

  Data untuk problem ini diambil dari jurnal yang sama seperti problem 10-13. Kusiak dan Heragu selanjutnya melakukan percobaan dengan TAA *single-row, double-row,* dan *cluster*. Hasil yang terbaik diperoleh dengan

perhitungan TAA untuk cluster. Metode cluster sendiri merupakan metode yang baik untuk menyelesaikan permasalahan tata letak mesin dan tidak terbatas pada satu atau dua baris, tapi sudah merupakan *multi-row*.

Hasil dari percobaan menunjukkan bahwa penggunaan metode DE lebih baik sebesar 16.99%-27.56%. Angka ini cukup baik dan hasil *layout* yang ditampilkan oleh program pun sudah baik. Dapat disimpulkan bahwa DE disarankan untuk menyelesaikan permasalahan dengan jumlah mesin ini.

- OPT8-OPT28

  Untuk problem ini, digunakan data dari *software* VIP-PLANOPT. *Software* ini adalah piranti lunak yang dikembangkan untuk menyelesaikan masalah tata letak fasilitas. *Software* ini sudah banyak digunakan dalam jurnal-jurnal sebagai referensi maupun pemecah solusi. Di dalam *software* ini ada beberapa data problem yang digunakan sebagai benchmark. Problem benchmark inilah yang digunakan sebagai data set OPT8-75.

  Untuk OPT8, dihasilkan solusi yang lebih baik oleh DE sebesar 9.84% daripada solusi yang dihasilkan oleh *software* VIP-PLANOPT. Untuk OPT12, dihasilkan solusi yang lebih baik sebesar 45.79%. Kedua hasil ini sangat baik, terlebih untuk OPT12 yang menghasilkan solusi yang sangat baik.

  Namun, untuk OPT20, perhitungan menggunakan DE ternyata memberikan hasil yang lebih buruk dari VIP-PLANOPT. Perbandingan menunjukkan perbedaan sebesar 14.60% lebih buruk. Hal ini memperlihatkan bahwa mulai ada kesulitan dalam pencarian solusi untuk ukuran mesin yang semakin besar.

  Untuk OPT28, ternyata perhitungan dengan DE memberikan hasil yang jauh lebih baik, yaitu sebesar 72.82%. Untuk hasil ini, sudah dilakukan pengecekan ulang, baik terhadap *software* VIP-PLANOPT maupun program DE sendiri. Namun, hasil yang diberikan memang benar bahwa DE lebih baik dari VIP-PLANOPT.

Jika kita melihat *layout* yang dihasilkan pada lampiran, terlihat bahwa sebenarnya solusi yang didapat masih bisa dioptimalkan lebih jauh lagi. Hal ini memperlihatkan bahwa DE mulai mengalami kesulitan dalam pencarian solusi untuk masalah dengan dimensi yang besar.

- OPT50-OPT75

  Kedua problem ini adalah problem yang berukuran besar. Kedua problem ini juga diambil dari benchmark problem *software* PLANOPT. Seperti yang terlihat dari hasil pengolahan data, hasil yang diperoleh lebih buruk dari pemecahan masalah menggunakan VIP-PLANOPT. Khusus untuk OPT50, telah diuji juga oleh Mir dan Imam (2001) menggunakan metode HOT yang hasilnya tidak lebih baik daripada PLANOPT.

  Pengolahan data memberikan hasil 72.5% dan 61.53% lebih buruk bagi kedua problem. Waktu pencarian pun cukup lama, yaitu mencapai 4 jam. Hal ini dikarenakan perhitungan DE ini belum bisa menyelesaikan problem berdimensi besar. DE menggunakan vektor secara keseluruhan untuk mencapai optimum, tidak satu per satu individu. Semakin banyak individu, akan semakin sulit pencariannya. Untuk problem ini, disarankan tambahan algoritma sebagai pemecah masalah.

- DUN62

  Problem ini memiliki 62 mesin dan diambil dari Dunker, et al yang meneyelesaikannya dengan *Coevolutionary Algorithm* (CA). Sama seperti problem 21 dan 22, pencarian mengalami masalah dan tidak memberikan hasil yang baik. Hasil yang didapat adalah 23.05% lebih buruk dari hasil CA.

  Hal ini menunjukkan perhitungan ini belum dapat digunakan untuk menyelesaikan problem dengan dimensi besar.

4.3.2 Analisa Waktu *Running Program*

Waktu yang dibutuhkan untuk pengolahan data menggunakan Java untuk menyelesaikan problem ini bervariasi. Waktu yang dibutuhkan bertambah seiring bertambahnya jumlah mesin yang diolah dalam problem.

Untuk problem KH, waktu yang dibutuhkan relatif sama, yaitu sekitar 6-10 detik. Untuk problem berukuran sedang, yaitu NUG5-NUG12, waktu yang dibutuhkan sekitar 10-20 detik. Untuk problem dengan ukuran mesin 12-20, waktu yang dibutuhkan adalah sekitar satu menit. Untuk menyelesaikan problem berukuran besar, ternyata membutuhkan waktu yang cukup lama, yaitu 12 menit sampai 4 jam, itupun belum memberikan hasil yang optimal.

Dari hasil itu, kita bisa menyimpulkan bahwa untuk problem kecil sampai sedang, waktu yang dibutuhkan untuk pengolahan data relatif cepat.

4.3.3 Analisa Metode

Dari hasil pengolahan data, ada beberapa hal yang bisa dipelajari. Hasil pengolahan data ini akan menggambarkan kemampuan algoritma *Differential Evolution* untuk menyelesaikan permasalahan tata letak, khususnya MLP atau tata letak mesin.

1. Perhitungan menggunakan *Differential Evolution* sangat baik untuk problem dengan dimensi kecil sampai sedang, namun tidak menunjukkan hasil yang baik untuk problem dengan dimensi besar.

   Seperti terlihat dalam pengolahan data, untuk problem berdimensi kecil sampai sedang, hasil yang ditampilkan sangat baik, bahkan jika dibandingkan dengan penelitian sebelumnya. Waktu yang digunakan untuk pencarian solusi juga sangat cepat. Kedua hal tersebut memberikan alasan bahwa DE sangat direkomendasikan untuk menyelesaikan MLP.

   Dalam penelitian ini, *Differential Evolution* yang digunakan tidak menggunakan tambahan algoritma seperti *local search* dan formulasi tambahan lain. Hal ini menyebabkan adanya kesulitan pencarian dalam problem berdimensi besar.

   Differential Evolution melakukan pencarian berdasarkan sifat-sifat induknya berdasarkan perbedaan vektor pada populasi. Sifat-sifat dalam vektor itu diturunkan bersamaan untuk setiap anggota dalam individu. Oleh karena itu, vektor anak yang dihasilkan pun bersifat global untuk keseluruhan anggota individu, bukan untuk masing-masing individu. Hal

ini menyebabkan pencarian untuk proble berdimensi besar akan mengalami kesulitan.

Jumlah ukuran populasi sangat berpengaruh dalam hal ini. Semakin banyak jumlah ukuran populasi, akan semankin banyak kombinasi vektor yang bisa dihasilkan dan kemungkinan untuk mendapatkan hasil yang lebih optimal pun semakin besar. Namun, jika ukuran populasi dinaikkan, waktu pencarian pun akan jauh lebih lama, karena itulah hal tersebut tidak diaplikasikan dalam penelitian ini.

2. Pemakaian metode penalti

   Perbedaan metode representasi kontinyu dengan metode diskrit adalah memungkinkan terjadinya tumpang tindih antar mesin. Untuk mencegah hal itu, maka dikenakan aturan penalti seperti persamaan (4.3). Dari hasil pengolahan data, persamaan ini sangat efektif untuk mencegah terjadinya tumpang tindih. Namun, semakin besar angka hasil, tentunya angka M untuk penalti juga harus makin besar. Dari pengolahan data, untuk problem 1-21 tidak terjadi tumpang tindih. Tumpang tindih terjadi pada problem 22 dan 23 dalam jumlah kecil.

3. Dibutuhkan pencarian dengan metode *local search*.

   Cara yang efektif untuk mengatasi masalah di atas adalah dengan melakukan kombinasi DE dengan metode lain, seperti *local search* dan tambahan formulasi. Yang dimaksud dengan *local search* di sini adalah pencarian posisi terbaik untuk masing-masing anggota individu. Ada beberapa cara yang dapat dilakukan, seperti dengan melakukan penggeseran sejauh beberapa unit jarak, dan mencari posisi terbaik, dan sebagainya. Dalam penelitian ini, *local search* seperti ini belum diaplikasikan. Hal ini sangat disarankan dalam pengembangan studi ini selanjutnya.

4. Dibutuhkan formulasi tambahan untuk melakukan pergeseran mesin

   Selain dari pergeseran *local search*, ada formulasi pergeseran yang bisa diterapkan, seperti melakukan pergeseran mesin. Hal ini belum diterapkan dalam penelitian kali ini. Pergeseran mesin sendiri memiliki beberapa cara yang sudah dilakukan dalam beberapa publikasi seperti

melakukan pergeseran mendekati centroid sesuai gradien, melakukan pergeseran horizontal, dan sebagainya. Jika hal ini diterapkan, tentu akan memberikan hasil yang lebih baik pada perhitungan menggunakan DE.

5. Ada beberapa kendala tambahan yang bisa dimasukkan ke dalam perhitungan

   Sesuai dengan pembatasan masalah, penelitian ini hanya melakukan perhitungan terhadap biaya pemindahan material. Pada kenyataannya, tentu saja masih banyak hal yang mempengaruhi susunan mesin yang harus dipertimbangkan seperti aliran produksi, resiko keamanan, kebisingan, dan sebagainya. Karena itu, perlu dilakukan modifikasi. Perhitungan dari DE ini adalah sebagai dasar dalam menentukan tata letak mesin. Selanjutnya, hal-hal tersebut di atas dapat dimasukkan dalam perhitungan sebagai kendala.

BAB 5
KESIMPULAN

Berdasarkan hasil pengembangan model representasi kontinyu untuk Tata Letak Mesin dengan menggunakan *Differential Evolution* dan bantuan bahasa pemrograman Java, diperoleh kesimpulan sebagai berikut.

- Pengembangan model representasi kontinyu untuk Tata Letak Mesin dengan menggunakan *Differential Evolution* dan bantuan bahasa pemrograman Java telah menghasilkan satu model yang menyelesaikan 23 masalah tata letak mesin dengan data yang telah dipublikasikan sebelumnya.
- Hasil dari pengolahan data menunjukkan bahwa problem dengan ukuran mesin kecil sampai sedang (<28 mesin) bisa diselesaikan dengan baik. Sebaliknya, program belum bisa menyelesaikan problem dengan ukuran besar.
- Hasil perbandingan dengan studi sebelumnya untuk jumlah mesin 12-28 mesin menunjukkan improvement dengan perbedaan sebesar 9.84% sampai 72.82%. Tetapi, untuk mesin ukuran 20-75 mesin, hasil pengolahan data menunjukkan hasil yang lebih buruk sebesar 14.6% sampai 72.5%.

Penelitian ini merupakan model dasar representasi kontinyu. Untuk penelitian ke depan, sangat disarankan pengembangan yang lebih mendalam.

DAFTAR REFERENSI

Anderses, R. (2006). Solution methods to the machine *layout* problem. Thesis of Kongens Lyngby IMM Technical University of Denmark.

Apple, J.M. (1977). *Plant Layout and Material handling*. New York: John Wiley and Sons.

Armour, G.C., & E.S. Buffa. (1963). A heuristic algorithm and simulation approach to relative location of facilities. *Management Science*, 9, 294–309.

Corry, P. & E. Kozan. (2004). Ant colony optimisations for machine *layout* problems. *Computational Optimization and Applications*, 28, 287-310.

Dunker, et al. (2003). A Coevolutionary Algorithm for A *Facility Layout Problems*. International Journals of Production Research, 41, 3479 – 3500.

Engineering Optimization *Software*. (1996). *PLANOPT User's Manual Ver 1.50*. Florida: Deltona.

Feoktistov, Vitaliy. (2006). *Differential Evolution*: In Search of Solutions. Springer.

Gupta, et. al. (1996). A genetic algorithm-based approach to cell composition and *layout* design problems. *Int. Journal of Production Research*, 34, 447-482.

Hassan, MMD, & Albin, M. (1994). Managing data requirement for the machine *layout* problem. *Int. Journal of Computer Applications in Technology*, 7, 28-37.

Heragu & Kusiak. (1988). Machine *Layout* Problem in Flexible Manufacturing Systems. Operations Research, Operations Research in Manufacturing, 36, No. 2, 258-276.

Mir & Imam. (2001). A Hybrid optimization approach for *layout* design of un-equal-area facilities. *Computers & Industrial Engineering*, 39, 46-63.

Nugent, C. E., T. E. Vollmann, & J. Ruml. (1968). An Experimental Comparison of Techniques for the Assignment of Facilities to Locations. *Operation Research*, 16, 150-173.

Price, K.V. (1999). An Introduction to *Differential Evolution*. In *New Ideas in Optimization* (pp. 79-108). UK: Mc Graw-Hill.

Price, Kenneth V, Rainer M. Storn, & Jouni A. Lampinen. (2005). *Differential Evolution: A Practical Approach to Global Optimization*. Germany: Springer.

Souliah, A. (1995). Simulated annealing for manufacturing systems *layout* design. *European Journal of Operational Research*, 82, 592–614.

Storn, Rainer. (1996). On the usage of *Differential Evolution* for function optimization. In *Biennal Conference of the North American Fuzzy Information Processing Society (NAFIPS 1996*), Berkeley, CA, New York, IEEE, pages 519-523.

Tompkins , J.A. et al. (1996). *Facilities Planning* (2nd ed.). New York: Wiley.

Ursen, Rasmus K.(2005). *Differential Evolution* Made Easy. *Technical Report no 1*.

Yang, T. & B.A. Peters. (1998). Flexible machine *layout* design for dynamic and uncertain production environments. *European Journal of Operational Research*, 108, 49–64.

Set Data KH1-KH9

Ukuran Mesin

Nomor Mesin	Panjang	Lebar
1	2	2
2	4	4
3	6	6
4	2	2

Aliran Material

KH1

	1	2	3	4
1	0	10	5	0
2	0	0	0	20
3	0	0	0	8
4	0	0	0	0

KH2

	1	2	3	4
1	0	10	15	15
2	0	0	0	5
3	0	0	0	40
4	0	0	0	0

KH3

	1	2	3	4
1	0	40	40	10
2	0	0	0	0
3	0	0	0	10
4	0	0	0	0

KH4

	1	2	3	4
1	0	10	15	20
2	0	0	10	15
3	0	0	0	10
4	0	0	0	0

KH5

	1	2	3	4
1	0	0.3	0	0.3
2	0	0	1	0.5
3	0	0	0	1
4	0	0	0	0

KH6

	1	2	3	4
1	0	1	2	39
2	0	0	2	0
3	0	0	0	40
4	0	0	0	0

KH7

	1	2	3	4
1	0	3	8	2
2	0	0	2	2
3	0	0	0	40
4	0	0	0	0

KH8

	1	2	3	4
1	0	2	2	2
2	0	0	2	2
3	0	0	0	0
4	0	0	0	0

KH9

	1	2	3	4
1	0	4	2	4
2	0	0	0	0
3	0	0	0	40
4	0	0	0	0

Set Data NUG5-NUG20

Ukuran Mesin

Nomor Mesin	Panjang	Lebar
1	0.04	0.04
2	0.02	0.02
3	0.05	0.05
4	0.03	0.03
5	0.01	0.01
6	0.05	0.05
7	0.08	0.08
8	0.01	0.01
9	0.05	0.05
10	0.04	0.04
11	0.05	0.05
12	0.01	0.01
13	0.03	0.03
14	0.02	0.02
15	0.03	0.03
16	0.01	0.01
17	0.02	0.02
18	0.03	0.03
19	0.04	0.04
20	0.09	0.09

Aliran Material

NUG5

	1	2	3	4	5
1	0	0	0	0	0
2	5	0	0	0	0
3	2	3	0	0	0
4	4	0	0	0	0
5	1	2	0	5	0

NUG6

	1	2	3	4	5	6
1	0	0	0	0	0	0
2	5	0	0	0	0	0
3	2	3	0	0	0	0
4	4	0	0	0	0	0
5	1	2	0	5	0	0
6	0	2	0	2	10	0

NUG7

	1	2	3	4	5	6	7
1	0	0	0	0	0	0	0
2	5	0	0	0	0	0	0
3	2	3	0	0	0	0	0
4	4	0	0	0	0	0	0
5	1	2	0	5	0	0	0
6	0	2	0	2	10	0	0
7	0	2	5	2	0	5	0

NUG8

	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0
2	5	0	0	0	0	0	0	0
3	2	3	0	0	0	0	0	0
4	4	0	0	0	0	0	0	0
5	1	2	0	5	0	0	0	0
6	0	2	0	2	10	0	0	0
7	0	2	5	2	0	5	0	0
8	6	0	5	10	0	1	10	0

NUG12

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

NUG15

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
4	5	3	10	0	0	0	0	0	0	0	0	0	0	0	0
5	1	2	2	1	0	0	0	0	0	0	0	0	0	0	0
6	0	2	0	1	3	0	0	0	0	0	0	0	0	0	0
7	1	2	2	5	5	2	0	0	0	0	0	0	0	0	0
8	2	3	5	0	5	2	6	0	0	0	0	0	0	0	0
9	2	2	4	0	5	1	0	5	0	0	0	0	0	0	0
10	2	0	5	2	1	5	1	2	0	0	0	0	0	0	0
11	2	2	2	1	0	0	5	10	10	0	0	0	0	0	0
12	0	0	2	0	3	0	5	0	5	4	5	0	0	0	0
13	4	10	5	2	0	2	5	5	10	0	0	3	0	0	0
14	0	5	5	5	5	5	1	0	0	0	5	3	10	0	0
15	0	0	5	0	5	10	0	0	2	5	0	0	2	4	0

NUG20

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	5	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	10	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	5	5	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	2	1	5	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	10	5	2	5	6	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	3	1	4	2	5	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	1	2	4	1	2	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
10	5	4	5	0	5	6	0	1	2	0	0	0	0	0	0	0	0	0	0	0
11	5	2	0	10	2	0	5	10	0	5	0	0	0	0	0	0	0	0	0	0
12	5	5	0	2	0	0	10	10	3	5	5	0	0	0	0	0	0	0	0	0
13	0	0	0	2	5	10	2	2	5	0	2	2	0	0	0	0	0	0	0	0
14	0	10	5	0	1	0	2	0	5	5	5	10	2	0	0	0	0	0	0	0
15	5	10	1	2	1	2	5	10	0	1	1	5	2	5	0	0	0	0	0	0
16	4	3	0	1	1	0	1	2	5	0	10	0	1	5	3	0	0	0	0	0
17	4	0	0	5	5	1	2	5	0	0	0	1	0	1	0	0	0	0	0	0
18	0	5	5	2	2	0	1	2	0	5	2	1	0	5	5	0	5	0	0	0
19	0	10	0	5	5	1	0	2	0	5	2	2	0	5	10	2	2	1	0	0
20	1	5	0	5	1	5	10	10	2	2	5	5	5	0	10	0	0	1	6	0

Set Data OPT8 - OPT75

OPT8

Ukuran Mesin

Nomor Mesin	Panjang	Lebar
1	2	3
2	4	5
3	2	2
4	3	3
5	2	4
6	4	4
7	4	4
8	3	4

Aliran Material

	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0
3	2	4	0	0	0	0	0	0
4	0	3	2	0	0	0	0	0
5	0	6	0	5	0	0	0	0
6	0	0	3	2	0	0	0	0
7	2	0	1	0	0	4	0	0
8	0	2	0	2	4	0	1	0

OPT12

Ukuran Mesin

Nomor Mesin	Panjang	Lebar
1	5	10
2	1	2
3	5	5
4	6	8
5	20	5
6	15	10
7	20	20
8	1	4
9	5	50
10	10	10
11	10	6
12	20	14

Aliran Material

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

OPT20

Ukuran Mesin

Nomor Mesin	Panjang	Lebar
1	1	2
2	2	2
3	1	1
4	2	3
5	3	3
6	2	2
7	2	1
8	2	3
9	3	2
10	3	3
11	2	3
12	1	2
13	3	2
14	3	3
15	3	3
16	2	2
17	3	2
18	2	2
19	2	2
20	2	1

Aliran Material

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	3	0	0	4	2	0	0	4	0	0	5	3	0	5	0	0	1	0	0
2	0	0	1	0	1	2	5	0	3	0	0	0	2	0	3	0	3	1	2	3
3	0	0	0	4	0	0	3	0	0	0	1	0	0	0	0	0	5	0	2	3
4	0	0	0	0	4	0	0	1	5	3	0	2	0	0	4	5	0	1	0	0
5	0	0	0	0	0	0	0	0	1	4	1	5	0	0	3	2	0	5	0	4
6	0	0	0	0	0	0	3	0	0	5	0	0	3	0	0	0	2	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	4	0	2	0	3	2	0	1
8	0	0	0	0	0	0	0	0	0	0	2	0	0	5	0	4	0	1	0	0
9	0	0	0	0	0	0	0	0	0	3	0	5	0	0	0	2	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	5	0	1	2	4	0	3	4	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	5	5	4	0	4	3	1
12	0	0	0	0	0	0	0	0	0	0	0	0	5	0	2	0	0	1	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	2	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	5	1	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	4	3	3
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	5	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	5
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	1
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OPT28

Ukuran Mesin

Nomor Mesin	Panjang	Lebar
1	3.5	2.5
2	2.5	4
3	2	3
4	2	3
5	2	3
6	4	4
7	4	3
8	2	3
9	3	2
10	3	3
11	2	3
12	4	2
13	3	4
14	3	3
15	3	3
16	3	4
17	4	2
18	4	4
19	4	3
20	4	3
21	3	3
22	2	2
23	3	4
24	4	3
25	3	3
26	2	3
27	3	2
28	4	4

Aliran Material

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	3	1	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	3	3	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	3	3	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	1	0	1	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	2	0	0	2	2	0	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	1	2	0	0	0	4	4	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	2	0	1	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	2	3	0	0	1	0	2	0	0	2	3	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	2	2	0	0	4	2	4	2	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	4	4	0	0	2	3	0	0	0	0	0	2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
18	1	1	0	1	3	0	0	1	0	3	3	1	0	2	4	3	0	0	0	0	0	0	0	0	0	0	0	0
19	0	2	2	0	0	0	0	0	0	0	2	0	0	1	2	0	1	4	0	0	0	0	0	0	0	0	0	0
20	0	3	2	0	0	0	1	0	0	0	1	0	0	0	3	0	2	1	3	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
23	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
24	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0
25	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	4	0	0	0	1	0	0	0	0
26	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	1	0	2	0	0	0
27	0	0	0	0	0	1	0	0	0	0	0	2	0	0	0	0	3	0	0	0	2	1	0	2	3	1	0	0
28	4	0	3	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	3	0	1	0	0	0

OPT50

Ukuran Mesin

Nomor Mesin	Panjang	Lebar	Nomor Mesin	Panjang	Lebar
1	1	4	26	2	2
2	1	3	27	2	4
3	4	4	28	1	2
4	5	2	29	5	3
5	2	6	30	6	3
6	1	4	31	3	6
7	5	6	32	3	1
8	4	1	33	3	5
9	2	5	34	5	5
10	5	2	35	5	4
11	4	1	36	6	5
12	5	3	37	5	2
13	5	6	38	3	5
14	3	5	39	5	2
15	5	2	40	2	4
16	5	3	41	5	5
17	1	3	42	3	1
18	1	4	43	4	1
19	2	2	44	4	4
20	5	4	45	2	4
21	1	4	46	3	4
22	2	5	47	2	3
23	5	4	48	5	5
24	4	6	49	1	6
25	4	4	50	3	3

Aliran Material

(lanjutan)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	2	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	4	7	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	6	5	7	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	9	1	3	7	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	4	9	3	6	8	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	8	7	5	5	4	5	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	4	5	9	5	2	7	6	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	6	5	10	4	5	5	4	3	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	6	7	4	3	3	4	4	5	2	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	6	3	3	5	7	3	5	4	5	4	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	5	8	4	7	7	6	3	6	10	8	7	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	4	4	3	7	9	5	5	8	2	7	6	7	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	4	5	10	4	4	3	2	8	9	3	6	4	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	7	5	4	3	5	9	8	5	5	2	4	2	6	9	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	1	9	8	5	4	4	6	9	7	5	5	4	6	8	9	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	7	3	3	4	3	5	3	7	1	5	2	5	9	7	5	4	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	5	5	3	6	2	7	7	3	3	4	6	5	5	10	5	2	5	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	7	3	6	1	2	4	5	9	5	7	4	2	4	8	4	9	5	9	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	5	10	7	9	6	5	5	7	9	8	5	6	6	6	9	1	6	4	5	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	5	2	4	6	9	6	8	5	4	1	9	3	3	5	5	5	8	7	9	2	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	5	7	7	5	6	8	6	7	10	4	7	4	1	6	4	3	7	9	6	9	2	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	6	5	4	6	10	6	9	7	9	8	5	10	8	9	5	6	9	7	4	7	2	10	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	7	3	7	8	3	1	5	7	2	3	4	4	6	9	6	9	4	7	7	6	5	5	8	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	5	4	3	6	3	2	6	5	6	7	5	6	5	3	3	4	6	5	5	5	3	5	5	5	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	4	8	7	6	8	6	6	5	5	6	7	6	6	8	4	8	4	7	6	1	8	3	8	4	7	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	2	5	3	6	4	7	9	8	5	9	8	4	7	6	4	2	4	6	3	1	3	4	4	5	3	2	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	8	2	7	5	5	6	7	4	5	7	7	1	9	5	6	3	5	8	1	3	4	5	5	2	6	3	2	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	6	7	3	5	6	6	8	7	6	8	4	1	6	2	3	3	8	7	8	8	3	1	3	1	5	4	1	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	9	4	5	7	6	4	9	5	9	5	3	5	5	8	6	10	5	6	6	3	6	3	6	7	8	6	6	5	3	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	6	4	5	5	9	6	6	3	9	7	5	2	3	5	5	7	2	6	6	7	7	1	5	4	9	4	5	3	7	6	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	6	6	5	7	5	4	6	7	8	10	8	5	5	7	8	7	7	7	6	5	6	4	5	5	1	6	5	6	3	6	6	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	4	4	7	7	5	4	6	6	6	3	4	5	4	5	6	5	2	10	7	7	7	6	6	9	5	6	6	4	6	4	7	9	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	8	4	8	10	3	6	2	5	5	4	5	4	3	4	6	5	3	9	7	5	6	3	2	10	8	5	7	5	4	6	5	8	2	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	5	7	5	9	7	6	5	4	8	7	3	8	5	4	7	6	3	5	4	3	5	4	5	5	7	4	4	5	4	1	6	2	3	5	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
37	6	5	4	6	5	8	5	5	6	7	5	6	6	5	4	9	5	6	4	8	4	2	3	5	5	6	7	6	4	5	5	4	4	4	4	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38	6	4	8	8	7	7	4	5	3	3	4	5	3	6	3	8	5	7	8	6	4	7	3	3	7	3	10	7	7	8	7	7	6	4	8	8	6	0	0	0	0	0	0	0	0	0	0	0	0	0
39	7	6	6	5	4	6	3	5	4	4	8	8	9	5	6	7	6	9	9	8	6	7	7	6	5	8	6	4	7	6	4	5	6	4	10	8	6	5	0	0	0	0	0	0	0	0	0	0	0	0
40	7	4	8	4	1	4	5	3	3	8	10	6	6	7	2	7	7	3	8	6	6	3	5	6	5	7	6	4	5	5	2	5	6	6	7	7	8	6	2	0	0	0	0	0	0	0	0	0	0	0
41	2	6	5	7	4	4	3	4	7	7	9	5	4	4	2	2	5	6	4	5	4	5	8	5	5	5	5	8	7	7	6	8	7	6	8	3	6	3	7	5	0	0	0	0	0	0	0	0	0	0
42	5	8	7	2	4	5	3	8	5	4	6	8	5	3	4	6	4	10	5	6	7	4	3	3	1	8	7	5	5	8	10	6	7	5	7	7	1	4	4	6	2	0	0	0	0	0	0	0	0	0
43	4	4	8	8	7	7	4	6	7	7	3	7	5	5	8	3	8	6	8	7	6	7	4	4	3	5	3	4	7	6	7	6	6	6	3	8	2	5	7	2	1	4	0	0	0	0	0	0	0	0
44	6	8	5	5	3	2	5	5	3	6	9	3	7	4	4	5	7	7	3	9	9	6	8	7	10	6	6	3	4	3	6	10	8	3	6	3	4	5	5	10	7	2	4	0	0	0	0	0	0	0
45	5	8	9	6	5	8	4	6	3	4	5	6	6	4	3	5	7	5	5	6	6	4	6	10	3	9	5	6	6	5	7	7	5	8	3	6	2	6	6	6	6	5	3	10	0	0	0	0	0	0
46	3	5	7	5	4	1	3	5	5	4	7	5	3	8	8	6	5	4	4	4	7	8	5	7	10	6	8	6	6	5	7	10	4	4	6	6	6	6	2	5	4	4	3	3	4	0	0	0	0	0
47	6	2	5	7	5	2	9	7	1	5	6	7	7	5	6	5	4	7	9	5	6	7	8	4	3	6	8	8	2	7	4	6	6	3	7	6	5	4	6	3	5	2	7	5	6	6	0	0	0	0
48	5	6	5	6	5	4	4	6	2	6	6	4	3	3	8	5	6	4	3	3	2	9	6	2	8	1	4	5	4	5	9	6	5	5	4	6	6	7	6	7	6	7	4	5	5	8	8	0	0	0
49	5	6	3	5	6	4	8	5	5	4	5	3	7	2	7	8	4	6	8	2	3	9	6	8	9	8	6	6	5	6	6	7	6	5	4	3	5	7	8	6	5	5	6	5	6	5	5	8	0	0
50	8	4	8	9	5	6	3	6	5	5	9	5	7	5	7	5	6	5	5	6	3	5	7	3	5	6	6	9	6	7	2	9	9	6	6	3	10	4	4	6	7	7	4	9	3	6	6	8	6	0

OPT75

Ukuran Mesin

Nomor Mesin	Panjang	Lebar	Nomor Mesin	Panjang	Lebar
1	1.4	3.7	39	2.5	4.2
2	3.6	3.8	40	3.2	2.7
3	5.2	5.8	41	5.8	3.8
4	1.8	4.3	42	5	3.3
5	5.9	4.1	43	4.7	1.3
6	2.8	4.3	44	5.8	2.7
7	5.7	3	45	3.2	5.9
8	4.1	3.1	46	4.8	1.6
9	1	5.8	47	5.7	5.1
10	1.9	1.4	48	3.3	3.8
11	3.1	3.6	49	3.3	5.3
12	1.6	1.6	50	6	1.6
13	6.4	1.7	51	5.2	5.7
14	5.4	5.3	52	3.6	2.3
15	5.9	4.2	53	3.1	2.8
16	1.7	1.7	54	5.2	3.8
17	2.4	4.9	55	3.8	6.4
18	5.6	3.8	56	3.2	3.4
19	3.6	4.3	57	3	3
20	2.8	8.8	58	1.4	1.8
21	3.4	3.9	59	1.3	1.1
22	3.6	5.8	60	3.4	4.7
23	1.6	3.8	61	1.9	5.5
24	1	3.3	62	2	5.9
25	2.6	2.7	63	1.3	3.4
26	3.9	2.2	64	4.9	4.8
27	2.7	4.2	65	5.8	3.6
28	3	3.8	66	2.9	3.4
29	1.6	5.6	67	3.6	3.9
30	1.9	2.7	68	4.8	3.2
31	4.6	3.4	69	1.2	1.6
32	5	5.9	70	5.1	1.4
33	3.5	5.3	71	2.9	7.1
34	4.4	5.1	72	4.7	4.6
35	3	3.7	73	3.1	4.1
36	2.8	1.8	74	3	4.2
37	5.8	3.9	75	1.4	3.6
38	1.1	4.8			

Aliran Material

Aliran Material (bagian 1)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	1	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	1	0	0	2	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	2	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	1	0	0	3	0	4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	5	0	0	0	2	3	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	1	1	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	1	0	4	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	1	0	4	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	3	0	0	0	0	2	0	0	0	2	1	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	4	1	1	1	0	1	0	1	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	1	0	5	0	0	1	0	5	0	4	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	2	1	0	5	0	0	0	0	0	0	0	0	0	0	4	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	5	0	0	0	3	0	0	0	0	1	0	4	0	0	0	0	5	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	2	3	1	0	0	2	1	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	3	0	0	0	0	3	0	0	0	0	0	4	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	1	0	0	1	0	0	0	0	4	2	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	3	3	0	4	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	1	0	0	1	5	0	0	4	0	0	0	0	0	5	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	3	0	0	0	5	0	0	0	0	0	3	0	0	0	0	3	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	3	5	0	0	4	2	5	3	1	1	0	0	0	4	1	0	0	1	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	3	0	2	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	5	1	0	0	0	0	0	0	0	0	0	0	0	0	0
28	1	1	0	0	0	0	1	4	0	1	0	5	1	0	0	0	0	0	3	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
29	5	2	0	5	5	0	0	0	1	3	0	0	0	3	0	0	0	3	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	1	2	0	0	4	0	0	0	0	0	0	1	2	0	0	0	0	0	4	5	3	0	0	2	0	0	0	0	0	0	0	0	0	0	0
31	0	4	0	0	3	0	1	0	0	0	0	0	0	0	0	0	0	0	3	0	0	1	0	1	0	0	0	2	0	0	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0	0	3	0	4	0	2	0	1	4	4	0	0	0	2	3	0	2	0	5	0	2	0	0	0	0	0	0	0	0	0
33	0	2	4	0	0	3	0	0	4	1	0	0	3	4	0	1	1	0	1	4	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
34	2	0	0	4	0	0	5	0	1	0	0	3	0	0	0	0	0	5	0	0	0	0	0	0	4	1	0	5	0	0	0	0	0	0	0	0	0	0
35	0	0	0	1	5	0	3	0	0	0	1	2	5	0	0	0	1	0	4	0	1	0	0	0	0	4	0	5	0	1	0	0	3	0	0	0	0	0
36	4	0	0	0	5	0	0	0	0	0	5	1	0	0	0	4	0	0	0	0	1	4	0	0	1	0	0	0	2	5	0	0	1	2	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0	1	0	1	0	5	0	0	0	0	0	0	4	4	3	0	1	0	0	0	0	1	3	0	3	3	0	0	0
38	1	0	0	0	0	3	0	4	0	2	1	0	4	0	0	2	0	5	0	0	2	0	1	1	0	1	0	1	0	0	0	0	1	0	0	5	0	0

Aliran Material (bagian 2)

	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Aliran Material (bagian 3)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38
39	0	0	5	4	0	2	0	0	1	1	0	0	0	2	5	0	0	0	2	0	0	3	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0
40	0	0	0	2	0	0	0	0	3	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	3	0	0	0	0	0	0	0	0
41	0	3	0	0	0	2	0	3	0	0	2	3	1	0	0	0	0	3	4	0	0	3	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
42	1	2	0	3	0	2	0	0	0	0	0	1	0	5	0	0	0	0	3	3	0	0	3	0	0	0	0	2	0	1	0	0	0	2	4	0	0	0
43	0	0	0	0	1	0	4	0	0	1	0	0	1	0	0	0	0	0	0	5	0	0	0	0	0	0	2	1	2	0	0	5	2	5	0	0	0	5
44	5	1	1	0	0	0	0	3	3	4	0	0	0	0	0	0	0	0	0	0	5	4	0	1	0	5	1	0	0	0	0	5	0	0	3	0	0	1
45	0	0	0	0	0	1	0	0	0	0	0	0	0	0	3	0	1	0	5	0	0	5	0	4	0	0	0	0	0	2	0	0	0	0	0	2	0	0
46	0	0	0	3	2	5	0	0	1	0	5	0	0	0	3	2	1	0	0	4	0	1	0	0	0	4	0	0	2	5	2	0	0	0	5	0	0	2
47	0	1	0	0	0	5	0	0	1	0	0	5	0	3	0	0	1	3	0	4	0	1	0	2	0	0	1	0	0	0	0	0	5	0	0	0	0	1
48	0	0	0	0	0	0	0	0	0	0	4	1	5	1	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	3	1	2	0	0	0	0	0	0
49	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	3	0	0	0	0	0	0	4	0	0	0	5	0	0
50	0	0	3	5	0	1	0	0	0	0	0	1	0	0	2	0	0	0	5	4	0	0	0	0	0	0	0	4	0	0	0	0	1	5	1	0	0	1
51	2	3	0	2	0	0	1	1	1	0	0	0	1	2	4	0	1	0	5	0	0	0	0	0	5	0	4	0	0	0	0	0	0	2	0	0	0	0
52	0	0	2	2	0	1	0	3	0	0	0	5	0	0	0	0	5	0	0	0	0	0	0	1	0	0	0	0	5	0	1	0	4	1	4	0	0	0
53	0	0	1	0	5	0	0	0	0	2	0	0	0	0	0	4	1	0	5	2	0	0	0	0	1	0	0	0	3	0	1	2	0	0	0	0	0	0
54	0	0	0	0	0	0	0	5	3	0	0	0	1	0	0	2	3	3	0	0	2	0	0	0	0	0	1	0	0	0	0	1	0	2	0	0	0	0
55	2	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	2	0	0	2	0	0	1	2	0	0	0	0	0	0	0	3	0	4	0	2
56	0	0	0	0	0	0	2	0	3	1	0	0	0	0	0	0	0	0	0	0	4	1	0	0	3	0	0	0	0	3	0	0	0	0	0	0	0	0
57	0	0	0	0	0	0	0	2	0	5	0	4	0	0	0	0	0	4	0	5	0	2	0	0	1	3	0	0	0	0	1	0	4	0	0	0	5	0
58	1	4	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	4	3	0	0	3	0	0	0	4	0	0	0	0	0	0	0	0
59	0	4	0	1	0	0	0	0	1	0	0	0	4	1	0	0	0	0	0	0	0	1	1	3	5	0	0	0	0	0	0	1	0	0	0	5	0	0
60	1	0	0	4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	1	0	5	0	0	0	0	1
61	0	4	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	3	0	0	0	1	0	0
62	0	0	0	0	2	4	0	0	0	3	2	4	0	0	0	0	0	3	0	5	1	3	0	0	0	0	0	0	0	0	4	0	4	0	0	0	0	0
63	0	0	0	1	0	0	0	0	0	0	0	5	0	0	3	5	1	0	0	0	0	0	1	0	0	2	5	0	5	0	0	2	0	0	0	0	0	1
64	4	0	3	0	2	4	0	0	0	0	5	0	4	5	0	0	3	0	0	0	1	0	0	0	3	5	0	0	0	0	0	1	0	1	0	0	1	0
65	2	0	0	1	0	0	1	0	0	0	0	0	2	2	0	3	1	0	4	1	0	0	3	0	0	0	5	0	1	4	0	0	0	0	0	0	0	0
66	0	5	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	5	4	2	0	1	1	0	0	0	3	5	0	3	0	1	0	0
67	1	0	0	0	0	0	0	0	4	5	0	0	2	0	0	0	1	0	4	0	1	0	0	0	1	0	1	0	0	5	0	0	0	0	0	0	0	0
68	0	0	0	0	5	0	0	0	2	3	0	3	0	0	0	1	0	1	0	5	0	0	0	4	1	0	1	0	0	5	1	0	0	0	0	0	0	0
69	1	2	0	1	3	1	0	1	0	0	0	3	0	0	2	0	0	0	3	1	0	0	0	4	0	0	0	1	0	1	0	0	0	1	3	0	0	0
70	5	5	1	0	0	0	0	3	0	0	0	0	1	0	0	0	2	0	0	0	0	2	0	4	0	0	0	0	0	0	0	1	0	1	0	0	0	0
71	0	5	0	0	0	0	1	0	4	0	0	0	1	0	0	5	1	0	0	0	0	2	0	0	4	0	0	0	5	2	2	0	5	0	0	2	0	4
72	0	5	0	0	0	0	0	0	3	0	2	0	0	0	0	3	3	0	0	0	0	2	0	0	0	0	0	0	1	0	3	0	1	0	0	0	0	5
73	0	0	0	0	0	0	3	3	5	0	0	2	0	0	5	0	2	2	0	0	4	0	0	1	0	0	5	4	2	0	0	0	0	1	1	1	0	0
74	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	2	0	0	0	0	0	0	0	0	0	3	0	1	0	0	0	1	0	3	5	0
75	2	0	5	0	0	0	1	0	0	1	0	0	0	0	0	0	0	2	0	3	4	0	0	0	0	0	4	0	0	0	0	5	2	0	0	0	1	4

Aliran Material (bagian 4)

	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75
39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
42	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
43	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
44	4	0	1	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
45	0	0	5	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
46	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
47	0	0	2	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
48	0	1	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
49	1	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
50	0	0	0	4	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
51	5	0	0	4	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
52	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
53	0	0	1	0	0	0	0	0	5	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
54	0	0	5	0	5	2	0	2	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
55	0	0	1	0	0	3	0	0	0	0	0	4	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
56	0	0	0	0	0	0	4	0	3	0	0	0	2	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
57	1	0	0	0	5	0	0	5	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
58	0	0	0	3	1	0	3	0	3	1	0	0	1	0	0	3	4	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
59	0	0	0	1	0	0	1	5	0	0	0	4	0	0	0	4	0	1	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
60	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
61	0	0	5	2	5	0	0	0	4	0	5	0	0	3	0	0	0	0	2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
62	0	0	0	0	0	1	0	5	0	0	0	3	0	0	4	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
63	0	0	0	5	0	0	5	0	0	1	0	0	0	0	0	0	4	5	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
64	0	0	4	0	2	0	1	4	0	0	1	0	0	1	0	0	0	0	0	5	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
65	0	0	0	0	0	5	0	0	0	5	0	0	0	0	0	0	0	5	5	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
66	0	0	0	0	0	0	0	0	2	0	0	0	5	0	0	1	0	0	3	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0
67	0	0	2	0	0	0	0	4	0	0	0	0	0	0	5	1	0	3	0	0	0	0	0	1	2	1	0	0	0	0	0	0	0	0	0	0	0
68	1	0	0	0	0	2	3	0	1	0	0	1	0	0	0	0	0	2	0	4	0	0	0	0	4	1	1	0	0	0	0	0	0	0	0	0	0
69	0	0	0	3	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	3	2	0	0	0	0	0	0	0
70	0	0	5	0	0	0	0	0	5	5	0	0	0	3	1	0	2	0	4	0	0	0	0	0	0	3	0	2	0	0	0	0	0	0	0	0	0
71	0	0	0	0	0	0	4	0	0	2	0	0	0	4	0	0	0	0	0	0	0	3	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0
72	5	5	0	0	0	4	1	2	0	4	0	0	0	3	0	0	4	0	2	5	0	0	0	0	0	0	0	0	0	1	0	4	3	0	0	0	0
73	0	0	0	1	4	0	1	0	3	0	0	2	0	5	1	2	0	2	0	0	0	1	0	4	0	5	0	0	0	4	0	2	0	0	0	0	0
74	5	0	4	0	1	5	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	4	1	0	5	0	0	0	0	5	0	0
75	1	0	3	0	0	0	0	1	1	0	0	0	0	0	0	0	0	4	0	1	0	3	1	0	0	0	0	0	1	0	4	0	0	0	1	0	0

Set Data DUN62

Ukuran Mesin

Nomor Mesin	Panjang	Lebar	Nomor Mesin	Panjang	Lebar
1	15	14	32	17	8
2	13	10	33	21	19
3	16	14	34	21	10
4	20	13	35	15	10
5	16	13	36	12	9
6	21	14	37	21	17
7	19	17	38	16	9
8	19	14	39	18	14
9	21	21	40	15	9
10	20	17	41	17	12
11	13	11	42	17	12
12	14	12	43	11	9
13	19	18	44	20	8
14	21	17	45	21	17
15	21	20	46	20	13
16	21	7	47	19	10
17	20	19	48	20	14
18	16	9	49	18	10
19	17	11	50	13	8
20	16	15	51	18	11
21	14	10	52	16	10
22	19	16	53	20	10
23	20	15	54	19	19
24	18	9	55	21	11
25	18	14	56	20	7
26	14	14	57	11	7
27	16	11	58	17	11
28	12	12	59	21	11
29	17	13	60	13	7
30	13	10	61	21	10
31	14	13	62	17	16

Aliran Material

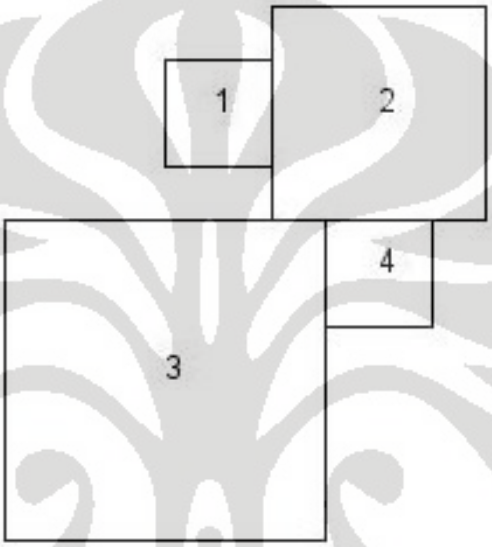
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62
1	0	51	93	60	42	0	13	6	45	0	63	96	0	14	0	32	0	43	0	52	45	0	94	55	30	0	59	72	29	0	0	0	32	51	19	0	23	30	40	0	85	89	48	90	97	0	0	0	0	0	41	75	52	84	69	9	0	59	68	24	95	59
2	0	0	44	0	88	22	32	16	25	37	12	46	23	0	99	10	86	0	18	23	16	0	74	93	8	7	57	69	30	41	21	37	51	25	64	0	25	40	41	84	0	70	77	63	0	0	47	0	6	18	0	0	28	97	0	74	52	0	68	0	18	72
3	0	0	0	33	35	6	8	45	13	14	71	94	17	92	54	56	67	62	63	29	93	21	55	85	84	73	37	24	9	85	7	52	0	54	63	47	36	26	77	23	93	25	6	22	21	0	8	45	28	16	99	32	71	51	49	72	68	0	31	27	54	68
4	0	0	0	0	92	90	11	56	11	0	18	18	98	0	0	83	0	56	0	81	23	71	84	0	27	0	5	0	61	91	0	65	12	0	0	28	52	0	82	0	84	53	0	0	94	27	5	0	12	43	84	0	74	51	91	50	0	76	33	0	10	0
5	0	0	0	0	0	83	13	31	59	61	53	54	18	0	94	59	0	46	56	19	65	49	0	51	0	55	0	23	65	42	0	30	76	48	0	29	84	83	16	70	76	0	70	73	0	72	96	17	75	0	77	14	0	75	55	50	0	0	88	55	12	0
6	0	0	0	0	0	0	0	42	0	72	20	63	84	0	0	35	0	0	0	76	43	7	6	0	0	0	32	29	87	0	0	0	47	0	0	19	43	0	21	0	0	0	49	0	86	0	0	0	0	0	0	0	0	0	0	21	5	27	29	0	11	11
7	0	0	0	0	0	0	0	0	83	60	41	62	94	11	0	62	0	0	0	42	83	71	0	0	26	0	26	14	91	0	0	0	65	0	0	0	9	16	16	0	0	0	20	64	0	12	47	91	92	85	0	45	59	13	0	35	83	0	17	0	33	8
8	0	0	0	0	0	0	0	0	18	65	8	7	37	44	52	97	86	42	38	36	25	27	24	82	45	82	7	36	12	17	26	85	79	79	50	55	82	17	18	33	60	54	16	30	76	96	41	48	0	22	67	69	61	32	60	11	69	97	19	88	31	11
9	0	0	0	0	0	0	0	0	0	0	61	91	0	98	17	62	90	54	42	50	29	22	49	0	52	61	0	69	67	39	82	48	71	48	10	10	68	14	55	23	0	0	0	0	31	57	80	0	0	80	65	84	34	52	64	91	9	0	72	37	22	10
10	0	0	0	0	0	0	0	0	0	0	42	75	77	88	0	23	75	11	28	17	8	52	76	0	48	0	32	99	36	75	9	54	0	23	0	84	20	76	27	26	97	43	8	24	26	0	22	0	85	0	0	54	96	64	95	51	69	44	0	0	62	65
11	0	0	0	0	0	0	0	0	0	0	0	23	99	52	21	96	12	0	0	55	58	40	31	0	0	25	52	0	22	65	40	64	0	50	0	0	48	84	53	0	18	38	0	22	42	92	22	54	30	88	62	93	7	45	0	18	65	0	47	0	6	57
12	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	24	0	0	0	65	90	84	11	32	75	65	28	83	94	55	69	0	30	0	34	0	78	85	0	15	0	0	10	35	80	0	23	6	0	40	0	73	0	30	74	68	0	99	70	0	24	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	82	0	0	62	96	9	98	76	0	61	0	0	9	95	0	61	0	0	97	68	91	9	19	0	61	34	71	0	0	9	0	0	70	0	0	0	94	29	0	0	79	0	8	45	0	90	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17	38	0	37	0	82	52	0	0	0	6	0	70	43	96	0	0	0	0	0	0	0	57	0	0	0	0	0	65	72	0	0	28	0	47	40	0	0	0	41	24	57	0	0	67	0	79	35
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	29	0	39	91	0	0	32	0	0	98	0	56	0	74	0	15	0	0	0	6	65	0	0	0	79	0	24	0	92	59	58	38	0	0	83	0	29	74	13	0	60	54	0	25	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	87	81	17	56	92	17	72	42	82	87	53	72	87	36	83	54	36	77	31	21	29	68	88	74	33	36	58	84	94	91	68	12	38	82	17	85	10	85	24	71	44	33	51	37	41	36
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26	46	0	0	0	0	0	44	0	27	0	0	0	25	0	0	0	62	56	0	0	0	0	0	0	80	0	0	0	78	0	0	51	0	0	0	79	0	0	98	0	74	37
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	90	97	70	59	90	98	0	10	0	0	52	95	0	0	94	0	0	57	67	0	0	66	69	0	9	0	39	0	0	0	41	0	0	37	0	0	0	92	0	0	94	25	95	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	27	13	0	0	0	0	0	16	0	29	78	0	78	71	0	0	85	75	17	0	0	0	0	23	0	0	0	0	0	0	0	0	0	0	80	0	87	44	0	0	0	11	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	91	71	14	36	95	96	51	97	32	14	12	91	16	27	45	86	17	93	41	42	21	73	22	12	7	11	88	81	95	13	66	39	70	68	89	62	0	15	22	66	18	16
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	98	76	98	25	15	41	18	11	60	60	10	52	48	77	75	31	44	7	36	33	47	42	24	84	57	58	24	36	93	95	9	99	0	74	41	21	67	63	50	91	16
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	66	86	0	77	37	19	14	0	0	45	51	0	34	85	41	80	51	18	53	33	0	39	90	6	0	68	89	80	71	0	0	27	72	61	34	73	85	82	53	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	79	49	31	27	79	72	69	59	35	62	72	43	76	27	93	0	0	62	24	89	0	77	0	0	13	44	25	0	8	0	54	58	0	32	0	9	0	95	63
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	77	0	67	82	46	0	0	0	0	0	0	0	75	0	0	0	0	0	0	27	45	0	0	0	0	0	0	64	0	0	73	32	0	0	85	0	39	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	53	35	0	61	0	0	90	34	0	0	0	32	78	68	78	73	62	70	94	0	0	0	0	83	29	37	25	0	53	45	40	57	0	24	0	56	40
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17	0	20	0	0	56	43	0	0	0	20	0	0	0	0	0	29	78	85	0	0	0	0	82	0	0	0	0	0	41	16	0	13	0	54	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	35	11	15	30	44	10	45	0	42	68	19	61	83	0	52	71	20	0	91	88	0	67	0	54	95	0	84	79	55	75	93	58	55	18	48
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	37	0	0	60	0	0	24	0	68	0	0	0	0	79	30	0	72	0	0	0	50	0	0	98	0	33	0	87	67	0	83	0	87	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	29	0	90	71	59	0	88	9	10	75	56	97	98	70	22	93	90	28	66	56	19	75	69	79	40	41	6	90	71	54	24	10
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	74	49	91	0	0	42	56	0	0	0	54	19	0	21	0	0	11	40	0	73	0	89	0	0	27	31	7	0	6	74	8	0
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	71	0	0	0	0	0	11	0	0	0	0	55	0	0	0	28	0	0	0	76	0	0	87	0	40	0
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	63	0	0	90	0	0	31	26	19	0	22	0	0	80	0	0	78	53	0	67	23	70	83	51	43	0
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	65	33	50	41	85	93	35	0	0	0	0	65	42	97	10	0	40	92	59	22	99	22	0	0	0	59	0	25	46
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	46	8	0	0	0	0	0	47	62	42	0	0	0	0	60	0	42	0	0	85	58	0	46	72	0	24	0
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	54	0	0	0	0	0	0	0	72	0	59	0	0	0	0	51	69	0	0	96	0	0	14	88	23	54
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	82	0	0	13	0	0	0	0	0	0	23	0	0	0	74	0	0	70	0	54	0
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	68	0	84	26	34	58	51	19	80	42	62	35	38	58	79	54	0	79	86	54	98	31	33	78	60
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	71	0	46	0	0	17	37	0	63	0	0	58	0	39	56	35	65	11	89	91
39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	82	0	43	0	98	94	0	30	0	0	0	94	0	0	32	0	41	0
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	24	0	0	0	0	0	0	42	0	67	0	0	49	0	36	5
41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17	32	7	84	0	74	0	5	0	13	0	36	22	0	51	0
42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	69	48	0	0	88	0	15	78	22	0	0	0	86	0	0	17	0	24	69
43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	95	0	0	35	56	0	89	90	29	64	47	7	0	0	0	74	0	27	95
44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	46	85	86	7	99	0	0	59	94	0	0	43	0	0	77	0	36	0
45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	62	0	73	5	0	0	0	0	88	83	55	21	86	50	13	0
46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26	0	0	0	0	39	0	98	0
47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0	72	0	91	0	0	0	88	46	0	79	0	62	7
48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	41	0	61	0	74	51	0	10	0	44	72	0	83	36
49	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	53	0	65	0	26	11	0	9	0	59	0
50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19	0	0	18	9	0	21	33	53	55	14
51	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	48	0	0	11	0	8	0
52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	76	0	8	0	8	20	76
53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	65	0	0	0	0	70	0	60	0
54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	83	99	0	87	11
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	55	0	89	0	79	10
56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	28	47	73	91	68
57	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	84	0	0	44
58	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	81	0	87	0
59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	31	80	25
60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	76	0
61	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	82
62	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Keterangan:

Untuk hasil solusi dari Java,

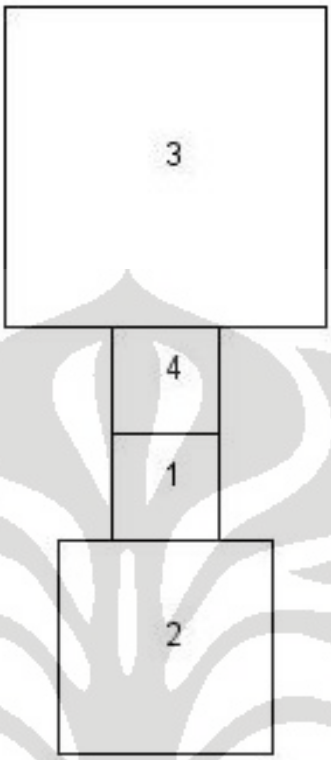
- Best [n-1] menunjukkan posisi koordinat X dari mesin ke n, dan
- Best [d+n-1] menunjukkan posisi koordinat Y dari mesin ke n, dengan d = jumlah total mesin.

KH1



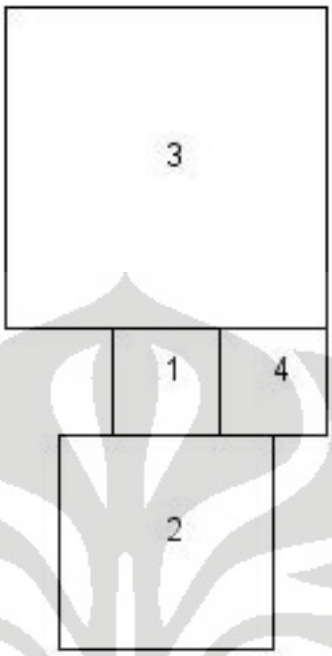
best[0]=	-9.72883	best[4]=	-55.6037
best[1]=	-6.72883	best[5]=	-55.6037
best[2]=	-10.7288	best[6]=	-50.6037
best[3]=	-6.72883	best[7]=	-52.6037

KH2



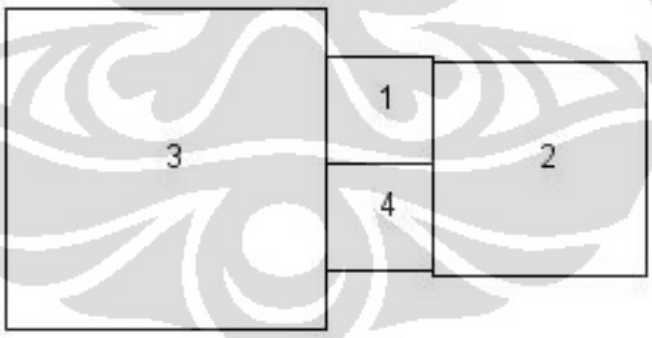
best[0]=	-17.6079	best[4]=	-21.5671
best[1]=	-17.6079	best[5]=	-24.5671
best[2]=	-23.6079	best[6]=	-21.5671
best[3]=	-19.6079	best[7]=	-21.5671

KH3



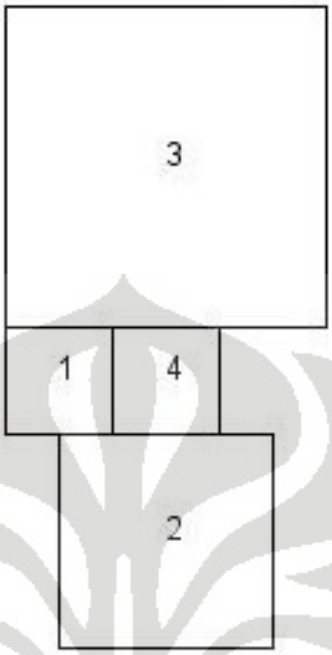
best[0]=	-29.7075	best[4]=	67.61304
best[1]=	-29.7002	best[5]=	70.61354
best[2]=	-29.7073	best[6]=	63.6117
best[3]=	-31.7083	best[7]=	67.61243

KH4



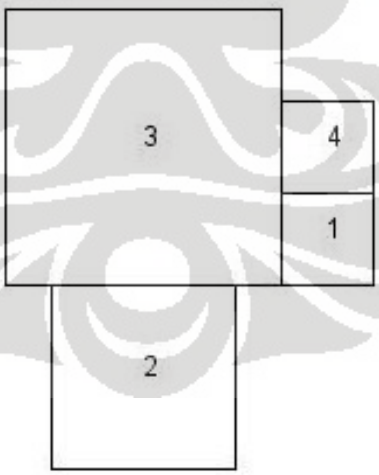
best[0]=	127.2956	best[4]=	10.10489
best[1]=	130.2956	best[5]=	11.2522
best[2]=	123.2956	best[6]=	11.2522
best[3]=	127.2956	best[7]=	12.10489

KH5



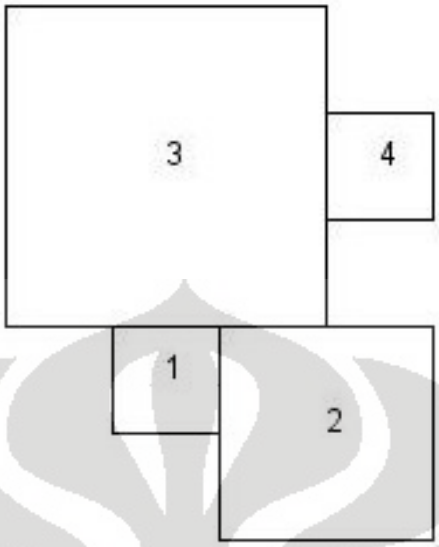
best[0]=	90.47184	best[4]=	-27.844
best[1]=	92.47184	best[5]=	-24.844
best[2]=	92.47184	best[6]=	-31.844
best[3]=	92.47184	best[7]=	-27.844

KH6



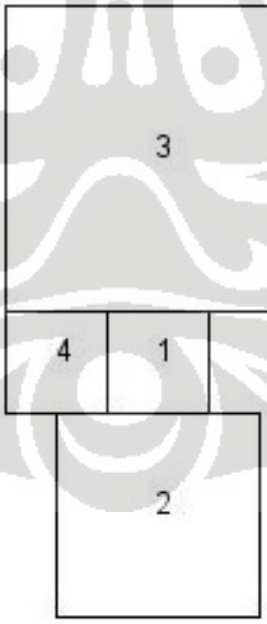
best[0]=	43.03219	best[4]=	29.04618
best[1]=	39.03219	best[5]=	32.04618
best[2]=	39.03219	best[6]=	27.04618
best[3]=	43.03219	best[7]=	27.04618

KH7



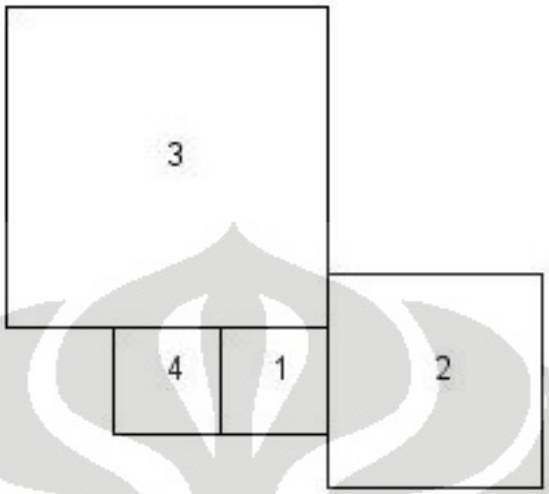
best[0]=	27.64898	best[4]=	-66.9279
best[1]=	30.64898	best[5]=	-65.9279
best[2]=	27.64898	best[6]=	-70.9279
best[3]=	31.64898	best[7]=	-70.9279

KH8



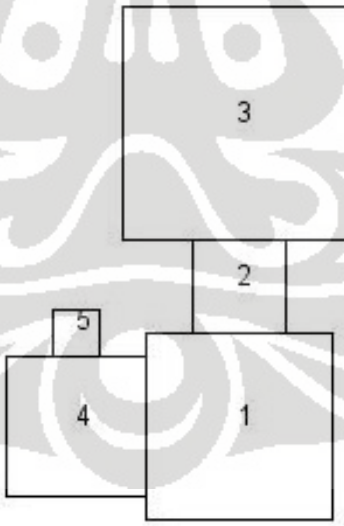
best[0]=	11.29246	best[4]=	-44.9207
best[1]=	11.29246	best[5]=	-41.9207
best[2]=	11.29246	best[6]=	-48.9207
best[3]=	9.292462	best[7]=	-44.9207

KH9



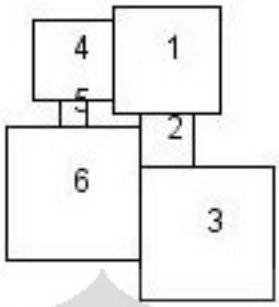
best[0]=	22.22798	best[4]=	-41.6167
best[1]=	25.22798	best[5]=	-41.6167
best[2]=	20.22798	best[6]=	-45.6167
best[3]=	20.22798	best[7]=	-41.6167

NUG5



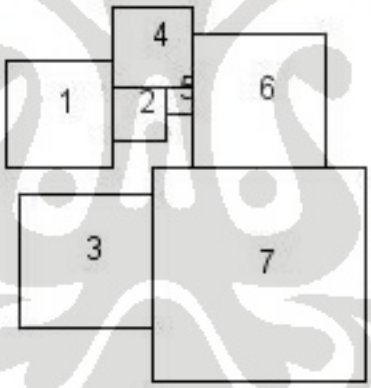
best[0]=	-17.7395	best[5]=	-28.921
best[1]=	-17.7395	best[6]=	-28.951
best[2]=	-17.7395	best[7]=	-28.986
best[3]=	-17.7745	best[8]=	-28.921
best[4]=	-17.7745	best[9]=	-28.941

NUG6



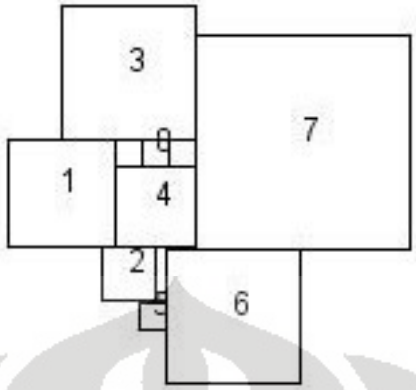
best[0]=	-33.6629	best[6]=	-45.7913
best[1]=	-33.6629	best[7]=	-45.7613
best[2]=	-33.6479	best[8]=	-45.7263
best[3]=	-33.6979	best[9]=	-45.7913
best[4]=	-33.6979	best[10]=	-45.7713
best[5]=	-33.6979	best[11]=	-45.7413

NUG7



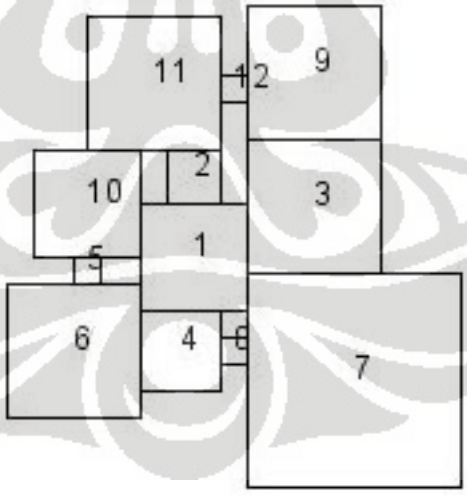
best[0]=	-14.2554	best[7]=	29.81158
best[1]=	-14.2254	best[8]=	29.81158
best[2]=	-14.2454	best[9]=	29.86696
best[3]=	-14.2204	best[10]=	29.78658
best[4]=	-14.2104	best[11]=	29.80658
best[5]=	-14.1804	best[12]=	29.80658
best[6]=	-14.1804	best[13]=	29.87158

NUG8



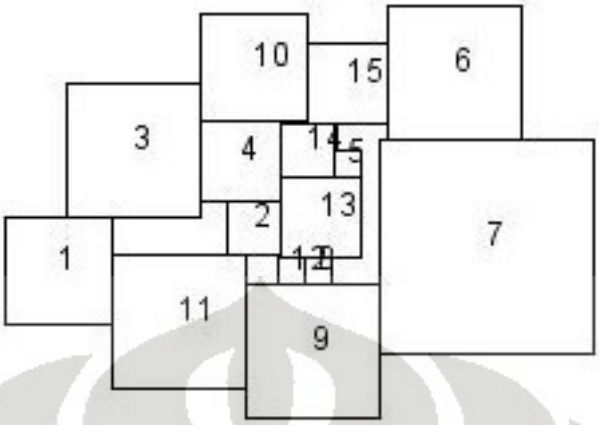
best[0]=	-31.9575	best[8]=	-15.6945
best[1]=	-31.9325	best[9]=	-15.6645
best[2]=	-31.9325	best[10]=	-15.7395
best[3]=	-31.9225	best[11]=	-15.6895
best[4]=	-31.9225	best[12]=	-15.6476
best[5]=	-31.8925	best[13]=	-15.6476
best[6]=	-31.8675	best[14]=	-15.7126
best[7]=	-31.9225	best[15]=	-15.7095

NUG12



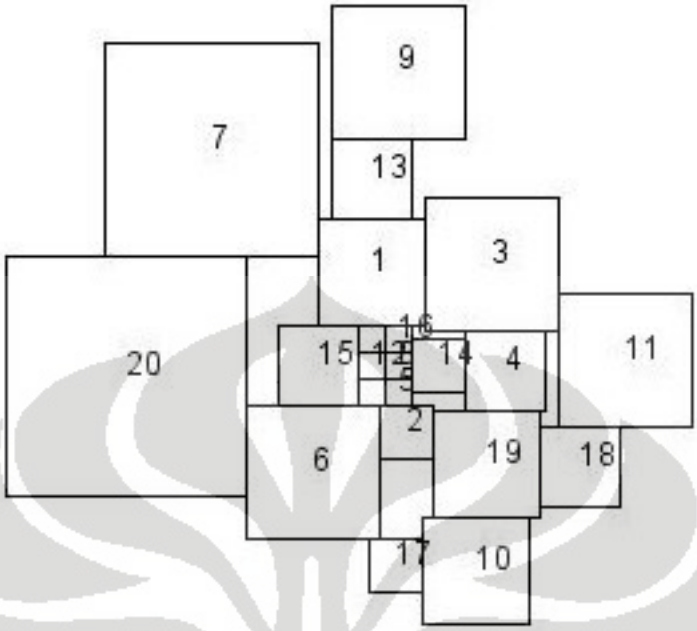
best[0]	7.370071	best[6]	7.430071	best[12]	-12.7028	best[18]	-12.6574
best[1]	7.370071	best[7]	7.385071	best[13]	-12.7328	best[19]	-12.6678
best[2]	7.415071	best[8]	7.415079	best[14]	-12.7224	best[20]	-12.7724
best[3]	7.365071	best[9]	7.330071	best[15]	-12.6678	best[21]	-12.7228
best[4]	7.330071	best[10]	7.355079	best[16]	-12.6978	best[22]	-12.7678
best[5]	7.325071	best[11]	7.385079	best[17]	-12.6678	best[23]	-12.7657

NUG15



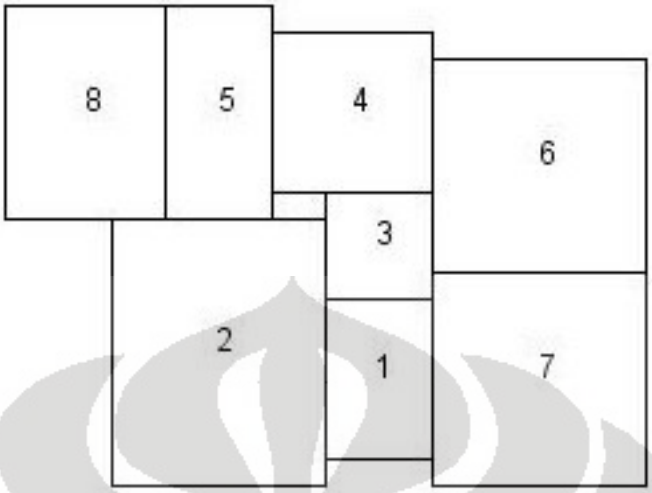
best[0]	9.938491	best[8]	10.03349	best[15]	15.1824	best[23]	15.2124
best[1]	10.01184	best[9]	10.01184	best[16]	15.1659	best[24]	15.1059
best[2]	9.966844	best[10]	9.983491	best[17]	15.1374	best[25]	15.2009
best[3]	10.00684	best[11]	10.02638	best[18]	15.1409	best[26]	15.1824
best[4]	10.04684	best[12]	10.03684	best[19]	15.1424	best[27]	15.1624
best[5]	10.08684	best[13]	10.03184	best[20]	15.10832	best[28]	15.1374
best[6]	10.09849	best[14]	10.04684	best[21]	15.17332	best[29]	15.1124
best[7]	10.03638			best[22]	15.1824		

NUG20



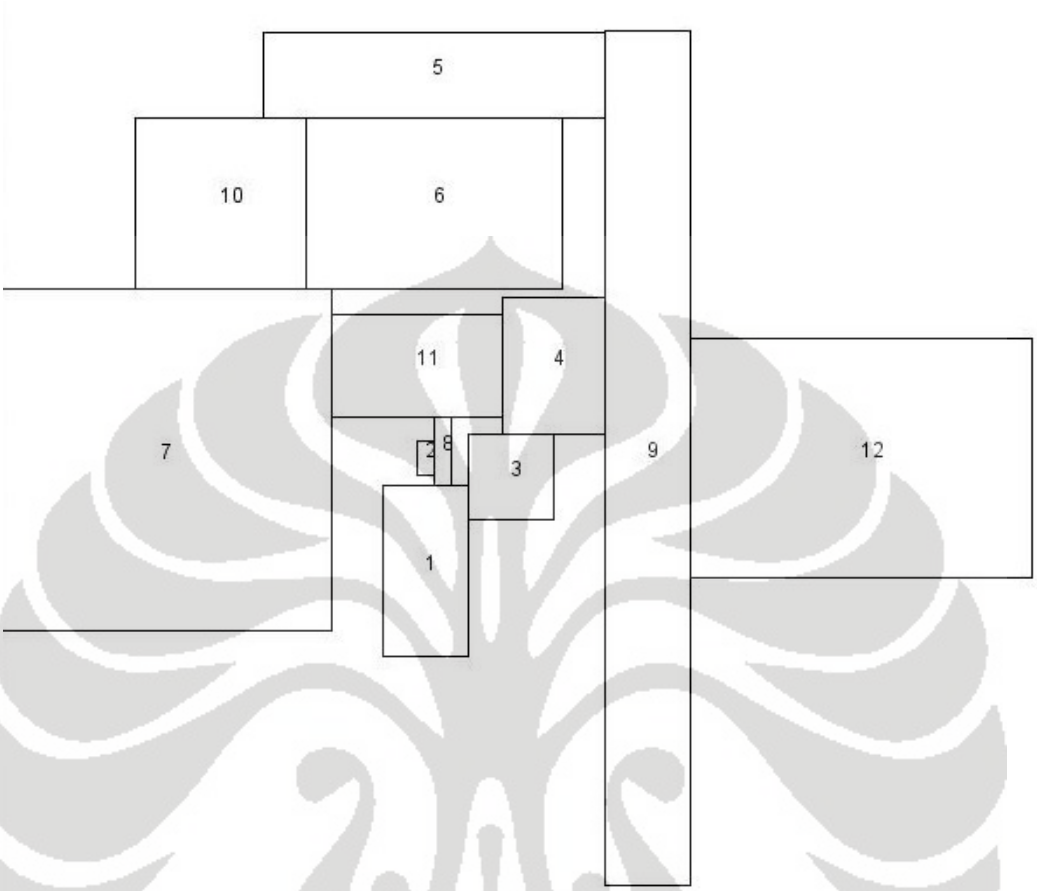
best[0]	0.406276	best[10]	0.501276	best[20]	-21.2004	best[30]	-21.1665
best[1]	0.418949	best[11]	0.406057	best[21]	-21.1397	best[31]	-21.1654
best[2]	0.451276	best[12]	0.406057	best[22]	-21.2028	best[32]	-21.2354
best[3]	0.456057	best[13]	0.431057	best[23]	-21.1628	best[33]	-21.1654
best[4]	0.416057	best[14]	0.386057	best[24]	-21.1547	best[34]	-21.1654
best[5]	0.383949	best[15]	0.416057	best[25]	-21.1254	best[35]	-21.1747
best[6]	0.346276	best[16]	0.415854	best[26]	-21.2457	best[36]	-21.0904
best[7]	0.416057	best[17]	0.483949	best[27]	-21.1647	best[37]	-21.1265
best[8]	0.416057	best[18]	0.448949	best[28]	-21.2754	best[38]	-21.1278
best[9]	0.445854	best[19]	0.313949	best[29]	-21.0878	best[39]	-21.1607

OPT8



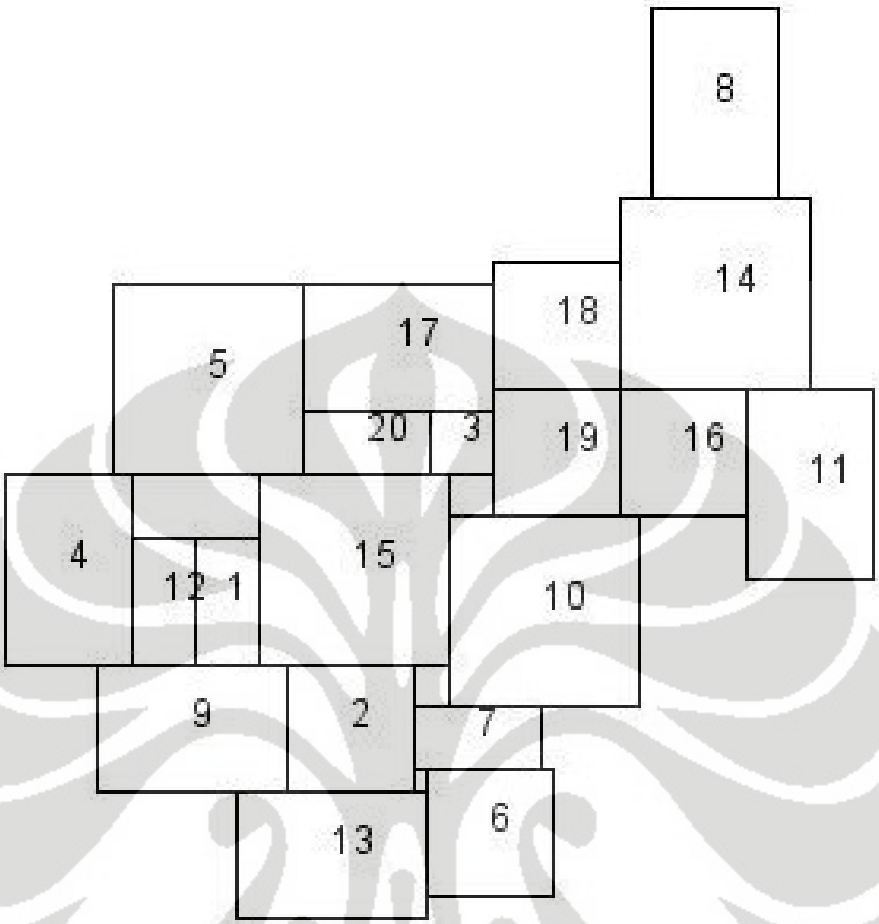
best[0]	1.979479	best[8]	-49.0978
best[1]	-1.02052	best[9]	-49.5978
best[2]	1.979479	best[10]	-51.5978
best[3]	1.479479	best[11]	-54.0978
best[4]	-1.02052	best[12]	-54.0978
best[5]	4.979479	best[13]	-53.0978
best[6]	4.979479	best[14]	-49.0978
best[7]	-3.52052	best[15]	-54.0978

OPT12



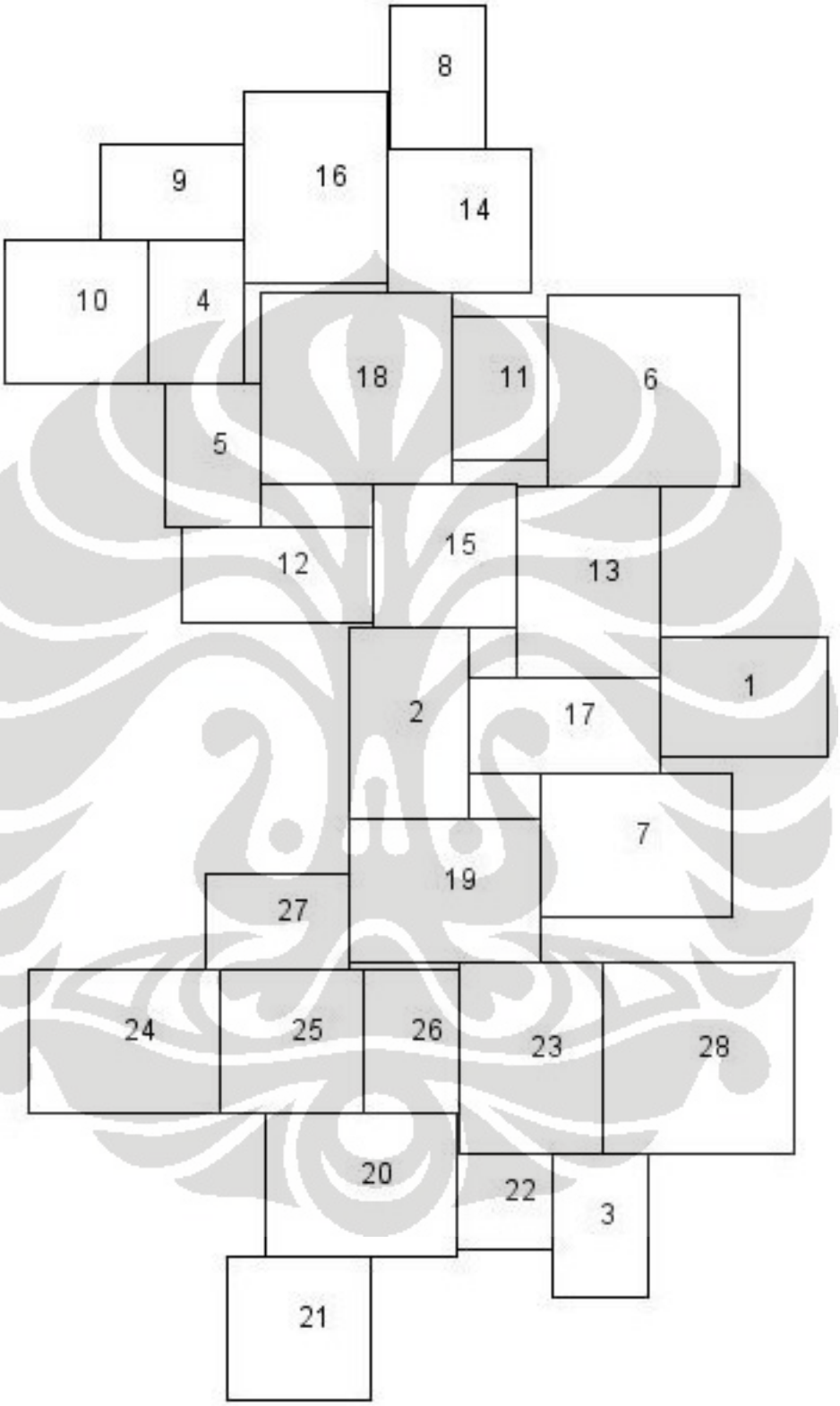
best[0]	17.18078	best[6]	1.616738	best[12]	2.795578	best[18]	-3.7345
best[1]	17.18078	best[7]	18.18078	best[13]	-3.76083	best[19]	-4.20442
best[2]	22.18078	best[8]	30.11674	best[14]	-2.7045	best[20]	-3.76083
best[3]	24.61674	best[9]	5.11687	best[15]	-9.2045	best[21]	-18.7345
best[4]	17.61674	best[10]	16.61674	best[16]	-26.2345	best[22]	-9.20442
best[5]	17.61687	best[11]	42.61674	best[17]	-18.7345	best[23]	-3.76083

OPT20



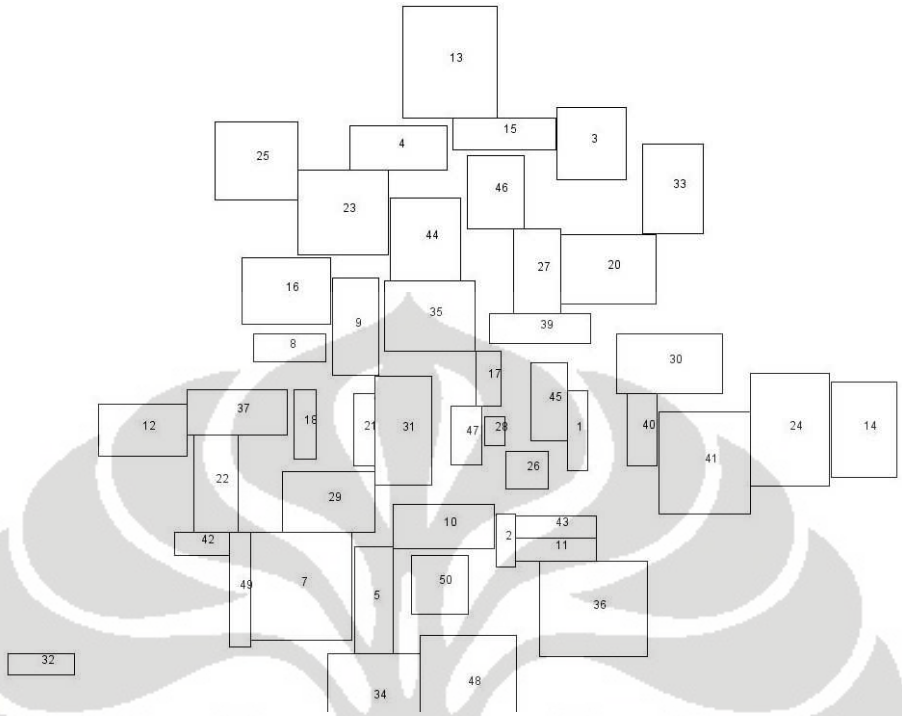
best[0]	-26.7968	best[10]	-17.5833	best[20]	5.601382	best[30]	3.747296
best[1]	-24.8397	best[11]	-27.7968	best[21]	7.601382	best[31]	5.601382
best[2]	-23.0833	best[12]	-25.1378	best[22]	3.101382	best[32]	9.601384
best[3]	-29.2968	best[13]	-19.0899	best[23]	5.101382	best[33]	0.747296
best[4]	-27.0899	best[14]	-24.7968	best[24]	2.101382	best[34]	5.101382
best[5]	-22.6378	best[15]	-19.5833	best[25]	9.247297	best[35]	3.247296
best[6]	-22.8397	best[16]	-24.0899	best[26]	7.747297	best[36]	1.601382
best[7]	-19.0899	best[17]	-21.5899	best[27]	-2.2527	best[37]	1.247297
best[8]	-27.3397	best[18]	-21.5833	best[28]	7.601382	best[38]	3.247297
best[9]	-21.7968	best[19]	-24.5833	best[29]	5.747297	best[39]	3.101382

OPT28



best[0]	29.67216	best[14]	23.4137	best[28]	-17.6845	best[42]	-20.6656
best[1]	22.67215	best[15]	20.68167	best[29]	-17.1656	best[43]	-28.3267
best[2]	26.65706	best[16]	25.92216	best[30]	-6.66557	best[44]	-17.077
best[3]	18.18167	best[17]	21.56017	best[31]	-25.7538	best[45]	-24.1656
best[4]	18.56017	best[18]	23.4137	best[32]	-22.7538	best[46]	-13.6656
best[5]	27.56021	best[19]	21.65705	best[33]	-24.077	best[47]	-7.48578
best[6]	27.4137	best[20]	20.36508	best[34]	-14.577	best[48]	-4.48577
best[7]	23.23678	best[21]	24.65705	best[35]	-30.6656	best[49]	-7.16557
best[8]	17.68167	best[22]	25.21502	best[36]	-28.2538	best[50]	-10.1656
best[9]	15.68167	best[23]	16.71502	best[37]	-25.7538	best[51]	-10.4858
best[10]	24.56017	best[24]	20.21502	best[38]	-24.1656	best[52]	-10.4858
best[11]	19.9137	best[25]	22.71502	best[39]	-20.2538	best[53]	-10.4858
best[12]	26.42202	best[26]	19.9137	best[40]	-20.077	best[54]	-12.9858
best[13]	23.68167	best[27]	28.71502	best[41]	-27.6656	best[55]	-10.1656

OPT50



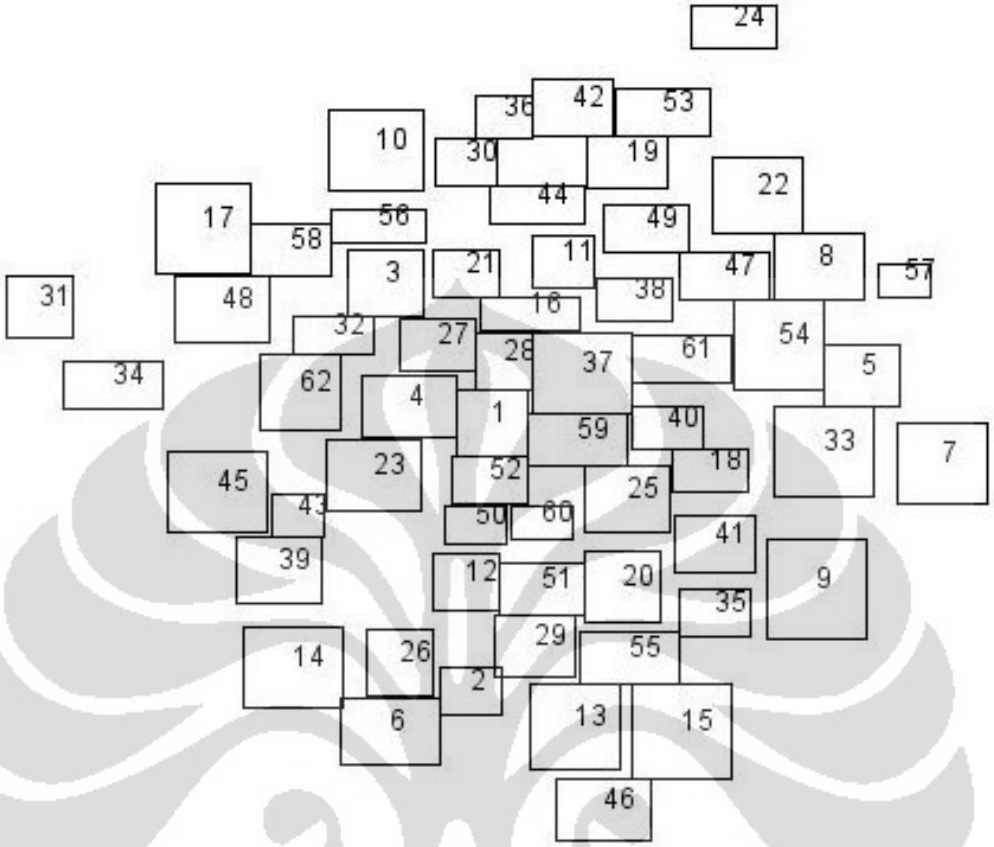
best[0]	0.66926	best[25]	-1.95625	best[50]	2.560891	best[75]	4.608164
best[1]	-3.04673	best[26]	-1.41685	best[51]	8.241991	best[76]	-5.71826
best[2]	1.411692	best[27]	-3.60874	best[52]	-12.3543	best[77]	2.565713
best[3]	-8.60996	best[28]	-12.2462	best[53]	-12.1296	best[78]	6.24928
best[4]	-9.89712	best[29]	5.459328	best[54]	11.35666	best[79]	-0.91156
best[5]	-6.97144	best[30]	-8.38016	best[55]	23.01019	best[80]	2.566429
best[6]	-13.664	best[31]	-27.1694	best[56]	10.62317	best[81]	14.68201
best[7]	-14.2646	best[32]	5.646699	best[57]	-1.74002	best[82]	-9.99404
best[8]	-10.8561	best[33]	-9.89367	best[58]	-2.82426	best[83]	16.475
best[9]	-6.26862	best[34]	-6.99209	best[59]	7.538464	best[84]	-3.38423
best[10]	-0.4346	best[35]	1.506679	best[60]	8.738152	best[85]	11.81165
best[11]	-21.8972	best[36]	-17.0027	best[61]	2.544911	best[86]	1.616685
best[12]	-5.94161	best[37]	-4.11429	best[62]	-16.5922	best[87]	23.41785
best[13]	15.55674	best[38]	-1.25896	best[63]	2.522206	best[88]	-2.73008
best[14]	-3.12446	best[39]	4.042015	best[64]	-12.8577	best[89]	2.505442
best[15]	-14.4573	best[40]	7.288802	best[65]	-4.68453	best[90]	4.22167
best[16]	-3.96434	best[41]	-18.828	best[66]	-0.15477	best[91]	8.445796
best[17]	-13.4721	best[42]	-0.42539	best[67]	2.233925	best[92]	7.543524
best[18]	-5.42017	best[43]	-7.19783	best[68]	19.64381	best[93]	-7.35555
best[19]	2.274649	best[44]	-0.79129	best[69]	-5.8323	best[94]	1.055694
best[20]	-10.4157	best[45]	-3.57388	best[70]	2.525182	best[95]	-9.81982
best[21]	-18.1061	best[46]	-5.09004	best[71]	5.30389	best[96]	2.796233
best[22]	-11.5031	best[47]	-4.99467	best[72]	-8.76915	best[97]	15.7946
best[23]	11.71237	best[48]	-16.848	best[73]	2.509613	best[98]	10.80341
best[24]	-15.9966	best[49]	-6.47129	best[74]	-11.4514	best[99]	10.54963

OPT75


31
59
74
45
68
6
61
41
5
38
40
7
71
54
35
13
9
64
51
22
58
72
2
26
37
16
63
46
42
50
48
67
33
52
43
15
36
12
69
11
57
8
34
44
25
65
10
29
24
30
18
28
1
66
23
3
53
75
27
20
17
62
21
32
19
4
49
73
60
14
70
47
56
39
55

best[0]	-36.3554	best[38]	-7.14226	best[75]	4.745559	best[113]	19.3285
best[1]	-0.61956	best[39]	-8.00833	best[76]	-7.8682	best[114]	-14.7816
best[2]	-1.30287	best[40]	1.291223	best[77]	5.013743	best[115]	-18.9688
best[3]	0.364872	best[41]	3.717383	best[78]	10.06712	best[116]	-5.31845
best[4]	-17.4396	best[42]	-6.76088	best[79]	-16.912	best[117]	-2.84466
best[5]	-12.1108	best[43]	-39.3128	best[80]	-19.0653	best[118]	1.24669
best[6]	-1.95301	best[44]	-3.32205	best[81]	-14.2665	best[119]	-22.4698
best[7]	-4.92516	best[45]	-12.0909	best[82]	-0.20055	best[120]	-5.34331
best[8]	-12.8451	best[46]	-12.1114	best[83]	-11.1936	best[121]	15.72671
best[9]	-22.8818	best[47]	-31.2286	best[84]	1.030087	best[122]	-2.62703
best[10]	-12.7976	best[48]	-36.917	best[85]	-0.19037	best[123]	10.03709
best[11]	-15.2229	best[49]	-2.35406	best[86]	-2.72237	best[124]	-4.4541
best[12]	-17.4795	best[50]	-19.768	best[87]	-13.9392	best[125]	-10.202
best[13]	-2.71128	best[51]	-10.8727	best[88]	14.83362	best[126]	-3.18935
best[14]	14.60586	best[52]	-40.11	best[89]	-2.39506	best[127]	6.324381
best[15]	-18.4942	best[53]	7.072753	best[90]	-5.51625	best[128]	-13.4316
best[16]	-7.91937	best[54]	-10.5443	best[91]	7.146602	best[129]	21.45171
best[17]	7.182336	best[55]	-16.7032	best[92]	3.700708	best[130]	19.40746
best[18]	-2.35413	best[56]	-8.64799	best[93]	10.04915	best[131]	-0.53734
best[19]	-11.5627	best[57]	-5.7706	best[94]	6.160008	best[132]	-9.25563
best[20]	-20.0558	best[58]	-9.66281	best[95]	8.759344	best[133]	-26.4207
best[21]	-8.53129	best[59]	-20.3344	best[96]	-9.20119	best[134]	13.08788
best[22]	-14.1287	best[60]	-6.76358	best[97]	3.484125	best[135]	-19.2448
best[23]	-9.66972	best[61]	-22.7891	best[98]	2.68791	best[136]	6.561561
best[24]	-35.1047	best[62]	-15.2229	best[99]	1.030086	best[137]	-5.66586
best[25]	-12.2789	best[63]	-24.825	best[100]	-7.20182	best[138]	-9.86154
best[26]	-16.3515	best[64]	-30.886	best[101]	5.195469	best[139]	1.820314
best[27]	11.44986	best[65]	-34.2265	best[102]	3.876589	best[140]	5.35454
best[28]	-18.506	best[66]	-27.7806	best[103]	2.086657	best[141]	-2.79421
best[29]	-6.17563	best[67]	-40.3662	best[104]	2.696115	best[142]	-23.0737
best[30]	-8.42011	best[68]	3.031223	best[105]	-28.6617	best[143]	-2.09483
best[31]	-15.7723	best[69]	-37.6513	best[106]	10.20756	best[144]	13.36809
best[32]	-22.23	best[70]	2.916826	best[107]	-2.29691	best[145]	-13.3371
best[33]	0.244223	best[71]	6.684326	best[108]	-0.65567	best[146]	-9.24961
best[34]	11.2177	best[72]	-7.68037	best[109]	-13.3851	best[147]	12.02355
best[35]	-17.4396	best[73]	-6.86094	best[110]	-2.71128	best[148]	-24.1259
best[36]	16.5867	best[74]	-21.1346	best[111]	-6.4781	best[149]	4.983504
best[37]	-10.5446			best[112]	-14.5571		

DUN62


24
42
53
36
10
30
19
22
44
17
56
58
49
11
8
57
3
21
47
31
48
38
16
32
27
54
28
37
61
34
5
62
4
1
59
40
33
7
23
52
18
45
25
43
50
60
41
39
12
51
20
9
35
14
26
29
55
2
6
13
15
46

best[0]	-10.11	best[31]	-43.6463	best[62]	-3.96106	best[93]	-22.4267
best[1]	-14.3184	best[32]	59.98038	best[63]	52.40197	best[94]	2.026895
best[2]	-32.4672	best[33]	-89.8383	best[64]	-33.4366	best[95]	-11.9248
best[3]	-27.658	best[34]	36.93816	best[65]	-7.24837	best[96]	35.82503
best[4]	67.87629	best[35]	-7.6211	best[66]	-13.9731	best[97]	-68.2522
best[5]	-31.3185	best[36]	9.147538	best[67]	61.2804	best[98]	-14.6692
best[6]	84.79514	best[37]	20.00771	best[68]	4.716313	best[99]	-29.791
best[7]	58.96172	best[38]	-55.1753	best[69]	-37.0674	best[100]	27.00804
best[8]	58.407	best[39]	27.14754	best[70]	31.28107	best[101]	-2.85408
best[9]	-34.6333	best[40]	36.82919	best[71]	-61.2329	best[102]	21.38823
best[10]	4.789096	best[41]	7.058163	best[72]	-38.1337	best[103]	-70.5283
best[11]	-15.3125	best[42]	-51.0745	best[73]	29.65358	best[104]	15.50804
best[12]	7.681717	best[43]	-0.54799	best[74]	60.10142	best[105]	-49.7651
best[13]	-52.0652	best[44]	-67.8557	best[75]	47.3705	best[106]	10.47443
best[14]	29.96005	best[45]	13.57503	best[76]	60.99204	best[107]	77.49719
best[15]	-2.20169	best[46]	39.03623	best[77]	-26.8646	best[108]	-35.0763
best[16]	-70.8613	best[47]	-67.2034	best[78]	-44.7722	best[109]	-28.0675
best[17]	35.73235	best[48]	22.42027	best[79]	6.172985	best[110]	-45.0763
best[18]	18.26957	best[49]	-13.4167	best[80]	-59.0182	best[111]	17.33217
best[19]	17.69395	best[50]	0.693269	best[81]	30.45385	best[112]	30.9317
best[20]	-15.331	best[51]	-10.5967	best[82]	-35.3646	best[113]	8.27809
best[21]	45.77483	best[52]	25.80299	best[83]	-52.0674	best[114]	-69.5344
best[22]	-35.0653	best[53]	50.37626	best[84]	7.06033	best[115]	-20.5674
best[23]	40.82182	best[54]	18.86878	best[85]	-87.7064	best[116]	45.47681
best[24]	18.34632	best[55]	-33.7937	best[86]	11.90698	best[117]	-45.5229
best[25]	-29.6165	best[56]	77.06079	best[87]	46.7226	best[118]	-33.7498
best[26]	-21.5138	best[57]	-52.3183	best[88]	-20.5264	best[119]	-40.5675
best[27]	-7.38341	best[58]	7.90334	best[89]	-17.1671	best[120]	-0.60416
best[28]	-1.00556	best[59]	0.222431	best[90]	42.9317	best[121]	16.85351
best[29]	-15.5851	best[60]	30.17407	best[91]	-58.7427	best[122]	-17.4049
best[30]	-105.786	best[61]	-50.7084	best[92]	-28.5383	best[123]	-10.255