

BAB I

PENDAHULUAN

1.1. Latar Belakang

PT XYZ merupakan salah satu produsen mobil dengan pangsa pasar tinggi di Indonesia, meliputi 8.9% pangsa pasar untuk kategori sedan, 22.4% untuk kategori mobil 4X4 dan 18.80% untuk kategori mobil 4X2 pada tahun 2017 (GAIKINDO, 2018). PT XYZ memiliki 2 pabrik (pabrik K1 dan K2) yang memproduksi mobil. Sejak didirikan pada tahun, PT XYZ telah memproduksi lebih dari 3,3 juta unit kendaraan dan pada tahun 2017 memproduksi 113.435 kendaraan.

K1 menghasilkan 76,08% dari total unit yang diproduksi PT XYZ pada tahun 2017. Produk yang dihasilkan meliputi dua model kendaraan, yaitu model S dan D dengan kapasitas produksi mencapai 150.000 unit per tahun. Setiap model memiliki berbagai *grade*, fitur, warna dan spesifikasi lain yang berbeda. Berdasarkan data produksi periode Januari hingga Juni 2018, rata-rata K1 menghasilkan 100 varian kendaraan dengan 11 jenis warna berbeda setiap bulannya. Volume produksi yang tinggi serta banyaknya variasi produk dapat menimbulkan masalah, yakni timbulnya *waste* persediaan pada *assembly shop*. Sedangkan pada *painting shop*, variasi warna yang tinggi dapat menyebabkan biaya produksi meningkat dan kualitas produk menurun.

PT XYZ menerapkan sistem manufaktur fleksibel yang memungkinkan perusahaan untuk memproses berbagai jenis produk dengan komponen yang berbeda-beda secara kontinu dalam jumlah produksi yang dapat disesuaikan dengan perubahan permintaan (Khausl et al., 2016). Terdapat empat lini produksi utama untuk menghasilkan produk kendaraan, dimulai dengan mencetak rangka dan panel pada *press shop* dan dilanjutkan dengan proses pengelasan pada *welding shop* untuk menyatukan panel-panel menjadi satu rangka mobil. Unit kemudian masuk ke dalam *painting shop* untuk diberikan pelapisan cat dan penambahan *sealer* untuk mencegah kebocoran dan karat pada unit serta meningkatkan estetika. Di tahap akhir, unit dikirim ke *assembly shop* untuk dirakit menjadi sebuah mobil jadi. Di setiap lini produksi, terdapat berbagai potensi permasalahan yang dapat terjadi.

Pada *assembly shop*, dalam memproduksi berbagai tipe produk dalam satu lini perakitan, terdapat dua aspek permasalahan desain yang perlu diperhatikan (Bard et al., 1994): *line balancing* untuk perencanaan jangka panjang dan *product sequencing* untuk jangka pendek. *Line balancing* berkaitan dengan perancangan stasiun kerja, sedangkan *product sequencing* berkaitan dengan penentuan jadwal urutan produk yang akan diproduksi. Dalam perencanaan *product sequencing*, pada umumnya terdapat dua tujuan yang dapat dicapai (Bard et al., 1994): minimasi *flowtime* (waktu tinggal unit dalam sistem) dan minimasi variasi penggunaan material. Setiap tipe kendaraan memerlukan beban kerja serta kebutuhan komponen dan material yang berbeda. Minimasi *flowtime* berupaya untuk menjaga waktu operasi dalam setiap stasiun tetap konstan dan mengurangi kemungkinan terjadinya *line stop* (lini produksi dihentikan sementara) akibat tidak seimbangnya waktu operasi antar produk. Dalam penerapan JIT, untuk dapat memproduksi mobil sesuai dengan tingkat permintaan sekaligus meminimasi persediaan, tujuan yang perlu dicapai adalah menjaga tingkat konsumsi part atau *part consumption rate* (Miltenburg, 1989). *Part consumption rate* adalah tingkat penggunaan material dalam setiap periode. Jika tidak terdapat tingkat *part consumption rate* yang merata, akan timbul *waste* berupa peningkatan persediaan untuk mengantisipasi perubahan permintaan yang signifikan, sehingga dapat menghilangkan keuntungan JIT (Boysen et al, 2009). Apabila variasi tingkat *part consumption rate* rendah, maka jumlah part yang digunakan konstan setiap periodenya. Tingkat *part consumption rate* yang konstan sesuai dengan prinsip *production smoothing/production leveling* (meratakan tingkat produksi) pada JIT.

Production smoothing adalah upaya untuk memproduksi unit dalam jumlah yang sama untuk setiap periode. Terdapat dua jenis *waste* yang dapat timbul jika tidak menerapkan *production smoothing* (Monden, 2012). Yang pertama, ketika pabrik membuat produk dalam jumlah yang bervariasi, maka fasilitas produksi, tenaga kerja, persediaan dan elemen penting lainnya disesuaikan untuk tingkat permintaan tinggi. Hasilnya, saat volume produksi rendah, banyak sumber daya dan tenaga kerja yang terbuang sia-sia jika dibandingkan saat volume produksi tinggi. *Waste* yang kedua terjadi di antara dua proses. Karena proses sebelumnya *supplier* menentukan sumber daya sesuai permintaan tertinggi dari proses selanjutnya perusahaan, maka proses sebelumnya juga akan memiliki sumber daya seperti tenaga kerja dan persediaan yang berlebihan ketika

permintaan rendah. Tujuan meminimasi variasi dalam *part consumption rate* memiliki prioritas yang lebih besar dalam *Toyota Production System* (Monden, 2012). Namun, dalam membentuk jadwal *sequence* produk, sistem produksi diharapkan tidak hanya memperhatikan *assembly shop*, tetapi juga harus memperhatikan lini produksi sebelumnya, yakni *painting shop*. Apabila jadwal urutan produksi hanya didasarkan kepada minimasi penyimpangan *part consumption rate*, maka jumlah *changeover*/pergantian warna pada *painting shop* akan menjadi tinggi, sehingga dapat menyebabkan biaya produksi meningkat dan kualitas produk menurun.

Painting shop memberi warna produk melalui proses pelapisan cat (*coating*). Apabila antara unit satu dengan unit selanjutnya berbeda warna (terjadi *changeover*), maka dilakukan *set up* berupa proses *flushing* sebelum melakukan pengecatan. Proses *flushing* merupakan proses membersihkan *nozzle* dari robot cat untuk menghilangkan sisa warna cat sebelumnya. Proses *flushing* memberikan beberapa dampak negatif, salah satunya adalah warna dari unit sebelumnya dapat mempengaruhi kualitas dari unit selanjutnya, sehingga kualitas *coating* dapat mengalami penurunan setelah pergantian warna (Bolat dan Yano, 1989). Cat yang digunakan berupa *aerosol* atau berupa butiran-butiran cat yang disemprotkan melalui robot. Selain dapat menempel di *nozzle*, partikel cat juga dapat terbang ke seluruh ruangan dan menempel di unit produk yang lain, menyebabkan kontaminasi warna dan berbagai jenis *defect*, seperti *seed*, *dust seed*, *touchmark*, dan lain-lain. Sehingga, semakin sering terjadi pergantian warna, semakin besar resiko kualitas turun. Hal ini senada dengan pernyataan Inman et al. (2003) bahwa fleksibilitas suatu sistem dapat menurunkan kualitas produk. Studi Wang et al. (2010) menunjukkan dalam sistem produksi lot, dengan merubah urutan warna antar lot, kualitas produksi dapat meningkat hingga 4.34%. Selain itu, proses *flushing* memerlukan biaya yang tinggi serta tidak ramah lingkungan karena menghasilkan limbah, salah satunya yaitu *Volatile Organic Chemicals* (VOC). Sumber utama VOC berasal dari *thinner* yang digunakan untuk melakukan proses *flushing* dan dari larutan yang digunakan sebagai bahan cat. Maka, proses *flushing* harus dilakukan seminimal mungkin untuk mengurangi dampak negatif yang dihasilkan. Untuk meminimalkan proses *flushing*, dapat dilakukan dengan dua cara. Yang pertama yakni dengan mengupayakan minimasi *changeover cost* dengan fungsi tujuan yang digunakan McMullen (2000), yang langsung menghitung berapa kali

terjadi *changeover*/pergantian warna. Namun, cara ini menyebabkan pemerataan produksi sulit dicapai karena kurang adaptif jika terjadi gangguan seperti kualitas yang rendah dari proses *Painting Shop* dan *Welding Shop* (Bernier dan Frein, 2004). Cara yang kedua yaitu dengan pemadatan (*densification*), di mana unit dengan warna yang sama berada dalam jarak sedekat mungkin satu sama lain, meskipun tidak berurutan secara langsung. Untuk dapat meminimasi proses *flushing* sehingga menurunkan biaya serta meningkatkan kualitas, maka penjadwalan harus dapat memaksimalkan (*densification*) pemadatan pada *Painting Shop* (Joly dan Frein, 2008). Maka, pada penelitian ini digunakan fungsi tujuan maksimasi *color densification*/pemadatan warna untuk meminimalkan proses *flushing*.

Berdasarkan data produksi periode Januari-Desember 2017, rata-rata produk yang bebas cacat baru 81% dari total produksi, sedangkan target *key performance indicator* (KPI) kualitas yang ditetapkan sebesar 92% produk bebas cacat. 23.44% dari total cacat berupa *seed*, *dust seed*, dan *touch mark* yang salah satunya disebabkan proses *flushing*. Selain meningkatkan biaya produksi, kualitas produk yang rendah pada *painting shop* dapat mengganggu produktivitas *assembly shop* karena produk yang seharusnya diteruskan harus diperbaiki dahulu di *painting shop* dan dapat menyebabkan lini produksi berhenti.

Penjadwalan produksi PT XYZ memegang prinsip *Just in Time* (JIT) dan membuat *sequence* produksi melalui penjadwalan bauran produk (*mixed-model scheduling*). Namun, pembuatan *sequence* kurang memperhatikan dampaknya terhadap proses *flushing* pada *painting shop*. Perusahaan menyadari pentingnya pengaturan jumlah proses *flushing* yang dilakukan, tetapi belum memiliki metode untuk mengoptimalkan penjadwalan *sequence* produksi agar proses *flushing* dapat diminimalkan dan penjadwalan *sequence* yang dihasilkan tetap sesuai prinsip JIT. Akibatnya, rata-rata terjadi 416 pergantian warna dari 543 unit yang diproduksi setiap hari pada periode Januari-Juni 2018. Hal ini menunjukkan bahwa frekuensi proses *flushing* yang dilakukan masih terbilang tinggi. Maka, diperlukan pengembangan model penjadwalan produksi yang ada sehingga dapat mengoptimalkan baik minimasi deviasi *part consumption rate* pada *assembly shop* maupun proses *flushing* pada *painting shop*.

Terdapat perbedaan tujuan antara *assembly shop* dengan *painting shop*. *Assembly shop* menginginkan bauran produk sesuai spesifikasi produk agar dapat meratakan penggunaan sumber daya, sedangkan *painting shop* menghendaki sistem lot warna untuk meminimasi proses *flushing*. Penjadwalan produksi dan pembuatan *sequence* produk perlu mempertimbangkan kedua tujuan tersebut. Dikarenakan terdapat lebih dari satu tujuan yang ingin dicapai, maka permasalahan penjadwalan produksi mobil termasuk dalam *multi-objective optimization problem*. *Multi-objective optimization problem* adalah salah satu jenis permasalahan yang bertujuan untuk mengoptimalkan lebih dari satu tujuan secara simultan (Chiandussi et al., 2012). Dalam permasalahan yang kompleks seperti penjadwalan *sequence* produksi, sulit untuk mendapatkan sekumpulan nilai optimal sebenarnya/*true pareto front* karena terdapat hampir tak hingga kemungkinan solusi yang dapat dihasilkan, sehingga optimasi yang dilakukan bertujuan untuk mendapatkan solusi-solusi yang paling mendekati atau memiliki tingkat konvergensi yang tinggi terhadap nilai optimal sebenarnya (McMullen, 2001). Beberapa tujuan yang perlu dicapai dalam *multi-objective optimization problem* meliputi (Chiandussi et al., 2012): mempertahankan solusi-solusi yang tidak didominasi, membuat algoritma mengarah menuju *pareto front*, menjaga keragaman solusi-solusi yang bersifat *pareto optimal*, dan memberikan pengambil keputusan beberapa alternatif solusi untuk dipilih.

Penjadwalan produksi dengan memperhatikan *assembly shop* dan *painting shop* telah dipelajari sejumlah peneliti. Gagne et al. (2006) membuat penjadwalan *sequence* dengan menggunakan *Ant Colony Optimization* (ACO). Ribeiro et al. (2008) menggunakan metode *Variable Neighborhood Search* (VNS) dan *Iterated Local Search* (ILS). Prandtstetter dan Raidl (2008) menggunakan *Integer Linear Programming* (ILP) dan VNS. Joly dan Frein (2008) membandingkan hasil optimasi penjadwalan *sequence* produksi dari metode *Progressive Construction-Sequence Algorithm* (PCSA), VNS, *Simulated Annealing* (SA), GA, serta ϵ -*Multi-Objective Evolutionary Algorithm* (ϵ -MOEA), dan menunjukkan bahwa metode GA dan ϵ -MOEA menunjukkan hasil yang lebih baik dibandingkan metode optimasi lainnya. Metode ϵ -MOEA lebih baik dibandingkan GA karena dapat memberikan beberapa alternatif solusi dibandingkan GA yang hanya memberikan solusi tunggal. Selain ϵ -MOEA, terdapat metode *Multi-Objective Evolutionary Algorithm* (MOEA) lain yang dapat digunakan, seperti *Non-*

Dominated Sorting Algorithm-II/NSGA-II, Strength Pareto Evolutionary Algorithm/SPEA2, Multi-Objective Evolutionary Algorithm based on Decomposition/MOEA/D, dan Pareto Envelope-Based Selection Algorithm/PESA. ϵ -MOEA mengungguli metode-metode tersebut dalam hal waktu komputasi yang lebih cepat dan tetap mempertahankan keragaman (*diversity*) solusi dan kemampuan dalam mendapatkan solusi yang optimal (Deb, 2005a). Namun, kelemahan ϵ -MOEA adalah beberapa solusi dapat hilang akibat pemilihan nilai ϵ yang tidak tepat (Hernandez-Diaz et al., 2007). Selain itu, dalam ϵ -MOEA perlu mengetahui batas bawah dan batas atas masing-masing fungsi tujuan sebelum melakukan optimasi. Untuk menyempurnakan metode ϵ -MOEA, Yang et al. (2013) mengembangkan metode *Grid-Based Evolutionary Algorithm* (GrEA) dan menunjukkan hasil yang lebih baik dibandingkan ϵ -MOEA, *Preference Ordering Genetic Algorithm/POGA, Fast Hypervolume Based Many-Objective Optimization/HypE, Multiple Single Objective Pareto Sampling/MSOPS*, dan MOEA/D.

Optimasi penjadwalan *sequence* produksi yang memperhatikan kebutuhan *assembly shop* sekaligus kebutuhan *painting shop* diharapkan dapat menghasilkan sistem produksi yang lebih efisien baik dari segi penggunaan sumber daya, kualitas maupun biaya yang dikeluarkan, namun tetap sesuai prinsip JIT. Perusahaan perlu mengembangkan model penjadwalan produksi yang lebih baik dibandingkan yang digunakan saat ini. Penelitian ini bertujuan untuk memenuhi fungsi tujuan minimasi proses *flushing* pada *painting shop* dan minimasi penyimpangan *part consumption rate* pada *assembly shop*. Metode GrEA digunakan untuk mendapatkan *trade-off* optimal antara fungsi tujuan yang ingin dicapai dan mendapatkan beberapa alternatif solusi.

1.2. Perumusan Masalah

PT XYZ adalah produsen mobil. Dalam proses produksi kendaraan mobil, terdapat perbedaan kepentingan antara *assembly shop* dengan *painting shop*. *Assembly shop* menginginkan urutan produksi berupa bauran produk (*mixed-model*) agar dapat meratakan tingkat *part consumption rate*, sedangkan *painting shop* menghendaki urutan *batch* atau lot berdasarkan warna untuk meminimasi proses *flushing* dan meningkatkan kualitas. PT XYZ menyadari pentingnya pengaturan jumlah proses *flushing* yang dilakukan, tetapi belum memiliki metode untuk mengoptimalkan penjadwalan urutan

produksi agar proses *flushing* dapat diminimalkan dan penjadwalan *sequence* yang dihasilkan tetap sesuai prinsip JIT. Akibatnya, frekuensi pergantian warna dan proses *flushing* tinggi, sehingga dapat menurunkan kualitas produk dan meningkatkan biaya produksi. Diperlukan model penjadwalan produksi yang dapat memberikan nilai yang optimal dan lebih baik dibandingkan metode yang dipakai saat ini, baik dari segi minimasi *part consumption rate* bagi *assembly shop* maupun minimasi proses *flushing* melalui maksimasi *color densification* bagi *painting shop*. Metode GrEA digunakan untuk mendapatkan *trade-off* optimal antara fungsi tujuan yang ingin dicapai dan mendapatkan beberapa alternatif solusi jadwal urutan produksi yang dapat dipakai.

1.3. Pertanyaan Penelitian

Berdasarkan perumusan masalah di atas, didapatkan pertanyaan penelitian berikut:

1. Bagaimana jadwal urutan produksi yang dapat meminimasi *part consumption rate* pada *Assembly Shop* dan maksimasi *color densification* pada *Painting Shop* bagi K1 PT XYZ menggunakan GrEA?
2. Bagaimana perbedaan tingkat penyimpangan *part consumption rate* pada *assembly shop* sebelum dan sesudah mempertimbangkan kebutuhan *painting shop*?
3. Bagaimana perbedaan jumlah proses *flushing* pada *painting shop* antara kondisi saat ini dengan hasil optimasi menggunakan GrEA dan pengaruhnya terhadap biaya proses *flushing* dan tingkat kualitas yang dihasilkan?
4. Apakah GrEA menghasilkan solusi yang lebih dominan dibandingkan dengan metode yang dipakai perusahaan?

1.4. Tujuan Penelitian

Tujuan penelitian ini adalah sebagai berikut:

1. Memperoleh jadwal *sequence* produksi yang dapat meminimasi *part consumption rate* pada *Assembly Shop* dan maksimasi *color densification* pada *painting shop* bagi pabrik K1 PT XYZ menggunakan GrEA.
2. Menganalisis perbedaan tingkat deviasi *part consumption rate* pada *assembly shop* sebelum dan sesudah mempertimbangkan kebutuhan *painting shop*.

3. Menganalisis perbedaan jumlah proses *flushing* pada *painting shop* antara kondisi saat ini dengan hasil optimasi menggunakan GrEA dan pengaruhnya terhadap biaya proses *flushing* dan tingkat kualitas proses.
4. Menganalisis hubungan dominansi antara solusi-solusi yang dihasilkan GrEA dengan metode yang dipakai perusahaan.

1.5. Batasan Penelitian

Dalam penelitian ini, batasan penelitian yang digunakan penulis meliputi hal-hal berikut:

1. Penelitian dilakukan pada Januari 2018-Juni 2018.
2. Proses *flushing* yang diamati hanya pada lini *Top Coat Spray* pada *Painting Shop* K1 PT XYZ.
3. Penjadwalan produksi hanya memperhatikan tingkat permintaan masing-masing jenis produk dan tidak memperhatikan waktu siklus proses produksi.
4. Penelitian tidak membahas dampak pembuatan jadwal urutan produksi terhadap hubungan perusahaan dengan *supplier part* secara mendalam.

1.6. Sistematika Penulisan

Sistematika penulisan pada laporan magang ini adalah sebagai berikut:

BAB I PENDAHULUAN

Bab I membahas mengenai latar belakang masalah, perumusan masalah, tujuan penulisan, pembatasan masalah, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab II membahas mengenai landasan teori yang digunakan sebagai acuan dan pedoman dalam melakukan penelitian.

BAB III METODE PENELITIAN

Bab III membahas mengenai kerangka berpikir, alur penelitian dan metode yang digunakan untuk menyelesaikan permasalahan.

BAB IV PENGUMPULAN DAN PENGOLAHAN DATA

Bab IV berisi data-data yang akan digunakan pada pengolahan data sesuai dengan langkah-langkah yang telah ditentukan.

BAB V ANALISIS DAN PEMBAHASAN

Bab V membahas tentang analisis dan pembahasan dari pengolahan data yang telah dilakukan sebelumnya.

BAB VI PENUTUP

Bab VI membahas mengenai simpulan dari hasil analisis dan pembahasan serta saran yang didapatkan berdasarkan kesimpulan yang telah didapat.

BAB II

TINJAUAN PUSTAKA

2.1. Sistem Produksi

Sistem produksi/sistem manufaktur merupakan kumpulan dari sub sistem-sub sistem yang saling berinteraksi dengan tujuan untuk mengubah input berupa bahan, pekerja dan input lainnya menjadi output berupa produk dan digerakkan oleh manajemen (Hartini, 2011). Proses produksi diawali dari sejak konsumen memesan produk sampai menerima produk tersebut. Ketika pihak penjualan (*sales*) menerima pesananan sebuah produk, pesanan langsung menuju ke departemen *manufacturing planning and control* untuk penjadwalan atau dapat langsung mengambil produk dari persediaan dan langsung menuju distribusi untuk dikirim konsumen. Pihak *design engineering* berfungsi mendesain produk yang akan dibuat dan dilanjutkan ke *manufacturing engineering*. Pada proses *purchasing* output dari sistem *Material Requirement Planning & Capacity Requirement Planning* digunakan untuk menentukan komponen yang dibutuhkan. Kemudian dilanjutkan ke proses pembuatan produk dan terakhir didistribusikan ke pelanggan.

2.2. Strategi Respon Terhadap Permintaan Konsumen

Strategi respon merupakan strategi bagaimana cara perusahaan merespon permintaan konsumen yang datang. Terdapat 5 strategi respon terhadap permintaan konsumen (Hartini, 2011):

1. *Design to Order*

Desain dilakukan produsen berdasarkan spesifikasi yang diberikan konsumen. Produsen hanya akan membuat produk apabila perkiraan waktu dan biaya disetujui konsumen. Perusahaan tidak memiliki persediaan barang kecuali contoh produk karena adanya resiko akibat variasi produk yang tinggi.

2. *Make to Order*

Perusahaan menyimpan material standar dan hanya akan memproduksi ketika ada pesanan. Desain produk ditentukan konsumen, sementara perusahaan menetapkan harga dan waktu pengiriman. Produk yang diminta biasanya bersifat tidak standar baik dari segi desain, ukuran, maupun jenis material. Jumlah pesanan sedikit. Mesin yang digunakan bersifat *multipurpose* dan layout produksi berdasarkan fungsi.

3. *Assemble to Order*

Perusahaan menyimpan barang setengah jadi. Setelah pembeli memesan produk, produsen hanya perlu merakit komponen sehingga waktu pelayanan relatif lebih cepat. Produk bersifat standar namun dapat memiliki spesifikasi komponen yang variatif.

4. *Make to Stock*

Produk yang dihasilkan standar, tidak beresiko bagi produsen untuk menyimpan dalam bentuk material, barang setengah jadi maupun produk jadi. *Leadtime* lebih pendek bagi konsumen karena produsen langsung dapat mengirim produk. Biasanya produk bersifat musiman.

5. *Make to Demand*

Make to demand merupakan strategi untuk merespon permintaan yang fleksibel yang dapat mengirim produk dengan kualitas dan waktu pengiriman sesuai dengan yang diinginkan konsumen. Strategi bersifat responsif terhadap pesanan konsumen tetapi dapat mengirim produk dalam waktu yang relatif cepat. Desain, material, komponen, dan produk jadi dapat disimpan dalam gudang.

2.3. Strategi Desain Proses Manufaktur

Strategi desain proses manufaktur merupakan strategi suatu perusahaan dalam menentukan bagaimana produk diproses. Terdapat 5 kategori desain proses manufaktur (Hartini, 2011):

1. *Project*

Dalam strategi *project*, bahan baku, alat, dan tenaga kerja dibawa ke lokasi pengerjaan produk. Produk spesifik dan unik dengan biaya mahal dan tingkat

kesulitan tinggi. Jumlah produk yang dihasilkan relatif kecil. Pekerja mempunyai keahlian khusus.

2. *Job Shop*

Dalam desain *job shop*, layout produksi bersifat *functional layout*, yakni mengelompokkan proses berdasarkan fungsi yang serupa. Produk bertipe sama, tetapi memiliki variasi yang tinggi sehingga memungkinkan adanya proses yang berbeda untuk setiap produknya. Proses produksi tidak selalu bergerak lurus (*forward*) tetapi dapat kembali (*backtracking*).

3. *Small Batch Flow Shop*

Dalam *flow shop*, tempat kerja diatur berdasarkan urutan proses produksi. Seluruh produk membutuhkan kerja yang sama dan mengikuti pola standar yang sama. Dalam *small batch flow shop*, produk tidak dihasilkan secara terus menerus. Desain ini digunakan ketika biaya pembuatan dianggap sama.

4. *Large Batch Flow Shop*

Large batch flow shop digunakan untuk produk-produk non-diskret dalam jumlah yang besar. Produk hanya dihasilkan dalam jumlah sedikit dalam setiap garis produksi, dan setiap bagian membutuhkan pengaturan masing-masing.

5. *Continuous Flow Shop*

Continuous flow shop digunakan untuk produk standar yang tidak memiliki banyak variasi. Mesin yang digunakan bersifat spesifik dan digunakan dalam jangka panjang.

Secara ringkas karakteristik proses manufaktur digambarkan pada tabel 2.1.

Tabel 2.1 Karakteristik Sistem Produksi

	Job Shop	Batch Flow	Small Batch Flow	Large Batch Flow	Continuous
Kelebihan	Kualitas tinggi	Kualitas tinggi	Kualitas tinggi	Biaya rendah	Biaya rendah
Fleksibilitas	Tinggi	Sedang	Sedang	Rendah	Standar
Implikasi	Biaya tinggi	Biaya tinggi	Biaya tinggi	Automasi	Automasi
Permesinan	Umum	Umum	Umum	Khusus	Khusus
Strategi	Make to Order	Assembly to Order	Assembly to Order	Make to Stock	Make to Stock

(Hartini, 2011)

2.4. Perencanaan dan Penjadwalan Produksi *Just in Time*

Just in Time (JIT) memiliki arti untuk memproduksi unit/produk yang diperlukan dalam jumlah yang diperlukan pada waktu yang diperlukan (Monden, 2012). JIT merupakan salah satu pilar Toyota Production System (TPS) dan merupakan metode produksi yang dapat beradaptasi terhadap adanya permasalahan dan perubahan permintaan. JIT tidak hanya mampu mengubah tingkat produksi, tetapi juga mampu mengubah aktivitas produksi dari satu model produk ke model lain dengan cepat dan dengan biaya setup yang minimum (Hartini, 2011).

Tujuan utama TPS adalah meningkatkan profit dengan mengurangi biaya (Monden, 2012). Pengurangan biaya dapat dicapai dengan mengeliminasi *waste*, salah satunya dengan mengurangi persediaan yang berlebihan melalui penjadwalan produksi yang tepat. Upaya untuk menyesuaikan tingkat produksi dengan permintaan disebut *production smoothing*. Konsep dari *production smoothing* adalah menghilangkan sebanyak mungkin variasi jumlah produk dalam lini produksi dan mencapai volume produksi harian yang konstan.

Terdapat dua jenis *waste* yang dapat timbul jika tidak menerapkan *production smoothing* (Monden, 2012). Yang pertama, ketika pabrik membuat produk dalam jumlah yang bervariasi, maka fasilitas produksi, tenaga kerja, persediaan dan elemen penting lainnya disesuaikan untuk tingkat permintaan tinggi. Hasilnya, saat volume produksi rendah, banyak sumber daya dan tenaga kerja yang terbuang sia-sia jika dibandingkan saat volume produksi tinggi. *Waste* yang kedua yaitu menumpuknya persediaan dan tenaga kerja yang berlebihan. *Production smoothing* dapat mengeliminasi *waste* yang timbul dengan menurunkan jadwal produksi bulanan menjadi mingguan atau bahkan harian, sehingga jumlah produk yang dihasilkan seragam untuk setiap unit setiap harinya dan mengurangi persediaan dan penggunaan sumber daya yang berlebihan.

Production Smoothing memiliki dua fase, yakni fase adaptasi bulanan dan adaptasi harian (Monden, 2012). Fase adaptasi bulanan dilakukan dengan membuat rencana produksi bulanan (*Master Production Schedule*) berdasarkan peramalan permintaan dan kapasitas produksi untuk menentukan jadwal produksi harian. Fase kedua, adaptasi harian berfungsi untuk menyesuaikan perubahan permintaan harian dalam satu bulan. Adaptasi harian dilakukan dengan menggunakan sistem kanban dan pembuatan jadwal urutan

(*sequence*) produksi menggunakan *mixed-model scheduling*. *Mixed-model scheduling* adalah metode penjadwalan produksi untuk mencapai produksi harian yang seragam.

Dalam industri automotif, dengan variasi model dan spesifikasi mobil yang beragam, penjadwalan produksi mobil seringkali difokuskan kepada lini perakitan (*assembly*). Secara umum terdapat dua tujuan berbeda yang dapat dicapai dalam penjadwalan produksi pada lini *assembly* (Bard et al., 1994):

1. Beban kerja berlebihan (*work overload*)

Variasi produk yang tinggi menghasilkan variasi waktu proses yang tinggi pula pada stasiun kerja. Jika beberapa model yang memiliki beban kerja tinggi dikerjakan secara berurutan dalam stasiun yang sama, maka beban kerja yang diterima terlalu tinggi dan perlu diatasi, seperti dengan menambah pekerja. *Work overload* dapat dihindari dengan membuat urutan (*sequence*) produk yang menyeimbangkan model dengan beban kerja tinggi dan beban kerja rendah. Optimasi yang dilakukan dapat menggunakan berbagai fungsi tujuan, seperti minimasi *work overload*, minimasi panjang lini produksi, minimasi *throughput time*, minimasi *total idle time*, dan minimasi *line stoppage*.

2. Tujuan Just in Time (JIT)

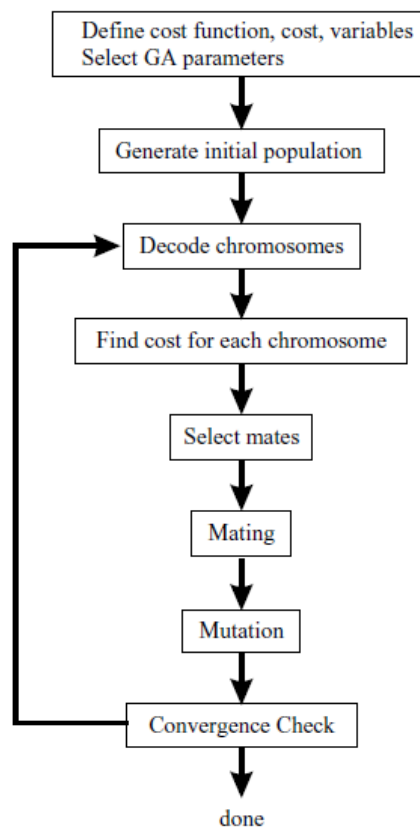
Pendekatan JIT fokus pada deviasi kebutuhan material. Model yang berbeda membutuhkan komponen dan part yang berbeda sehingga *sequence* produk mempengaruhi tingkat permintaan material sering berjalannya waktu. *Sequence* produk yang tidak tepat dapat menyebabkan penumpukan persediaan untuk menghindari kekurangan material pada saat permintaan tinggi. Pendekatan JIT harus dapat menghasilkan distribusi kebutuhan material yang seragam dalam rentang waktu yang ditentukan. Optimasi yang dilakukan menggunakan minimasi *work in process* atau minimasi deviasi *part consumption rate* untuk mendapatkan tingkat penggunaan part dan material yang konstan.

2.5. Genetic Algorithm (GA)

2.5.1. Pengertian dan Konsep Dasar

Genetic Algorithm (GA) adalah salah satu algoritma pencarian yang menggunakan konsep seleksi alam dan gen alami (Goldberg, 1989). GA mengkombinasikan seleksi

strings yang paling sesuai di antara *strings* yang ada dengan pertukaran informasi yang terstruktur namun bersifat acak untuk membentuk algoritma pencarian. GA memungkinkan sebuah populasi yang terdiri dari berbagai individu untuk berevolusi sesuai aturan seleksi tertentu untuk memaksimalkan nilai *fitness* tertentu, seperti meminimasi biaya (Haupt dan Haupt, 2004). Dalam setiap generasi, sekumpulan makhluk buatan (*strings*) diciptakan dengan menggunakan potongan informasi yang terbaik dari *strings* yang sudah ada. Gambar 2.1 menunjukkan diagram alur algoritma GA secara umum.



Gambar 2.1 Diagram Alur GA

(Sumber: Haupt dan Haupt, 2004)

Perbedaan GA dengan metode optimasi lainnya yaitu (Goldberg, 1989):

1. GA bekerja dengan *coding* atau kode dari parameter, bukan dengan parameter itu sendiri.
2. GA mencari dari sebuah populasi/sekumpulan titik, bukan dari sebuah titik tunggal.

3. GA menggunakan informasi hasil (fungsi tujuan), bukan dengan turunan (derivatif) atau pengetahuan tambahan lainnya.
4. GA menggunakan aturan transisi probabilistik, bukan deterministik.

Kelebihan GA dibandingkan metode lainnya meliputi (Haupt dan Haupt, 2004):

1. Dapat mengoptimasi dengan variabel diskrit maupun kontinu
2. Tidak membutuhkan informasi tambahan
3. Dapat menangani jumlah variabel yang besar
4. Dapat keluar dari lokal optima
5. Memberikan sejumlah variabel yang optimum, tidak hanya solusi tunggal
6. Dapat bekerja dengan data bangkitan, data eksperimen, maupun fungsi analitis

2.5.2. Pengkodean Gen dan Kromosom

Dalam GA, satu populasi terdiri dari sekumpulan kromosom. Satu kromosom menunjukkan satu solusi dan terdiri dari sekumpulan gen yang mewakili sifat-sifat yang dimiliki oleh solusi tersebut. Gen merupakan bentuk kode dari variabel yang digunakan dalam permasalahan optimasi. Gen dan kromosom dapat memiliki beberapa bentuk (Haupt dan Haupt, 2004):

1. Biner

Pengkodean biner dilakukan dengan merubah nilai variabel menjadi serangkaian bilangan biner dalam satu kromosom. Pengkodean biner adalah metode pengkodean dasar dan paling umum digunakan dalam GA.

$$chromosome = \left[\underbrace{1110101}_{gen\ 1} \underbrace{1001}_{gen\ 2} \dots \underbrace{10101}_{gen\ N} \right]$$

2. Value Encoding

Value encoding menggunakan nilai aktual dari masing-masing variabel dan diterjemahkan menjadi gen dan dirangkai menjadi kromosom.

$$chromosome = \left[\underbrace{11}_{gen\ 1} \underbrace{32}_{gen\ 2} \underbrace{5}_{gen\ 2} \dots \underbrace{10}_{gen\ N} \right]$$

3. Permutasi

Kromosom dari kode permutasi berisi bilangan integer yang bersifat unik. Pada umumnya pengkodean permutasi digunakan untuk permasalahan penjadwalan atau penentuan rute rantai pasok.

$$chromosome = \left[\underbrace{1}_{gen\ 1} \quad \underbrace{2}_{gen\ 2} \quad \underbrace{5}_{gen\ 3} \quad \underbrace{7}_{gen\ 4} \quad \dots \quad \underbrace{10}_{gen\ N} \right]$$

2.5.3. Seleksi

Dalam GA, tahap seleksi bertujuan untuk menentukan dua individu yang akan dipilih dari populasi untuk disilangkan dan menghasilkan keturunan baru. Terdapat berbagai jenis metode seleksi individu, beberapa di antaranya meliputi (Haupt dan Haupt, 2004):

1. *Pairing from top to bottom*

Metode ini memasangkan kromosom dari individu teratas dalam populasi dengan individu setelahnya hingga terpilih sejumlah individu yang akan digunakan dalam proses persilangan. Dengan kata lain, metode ini mengawinkan solusi pada baris ganjil dengan solusi pada baris genap. Metode ini sangat sederhana meskipun tidak memodelkan seleksi alami secara akurat.

2. *Random pairing*

Pemilihan dua *parent* dilakukan secara acak dengan peluang terpilih yang sama (berdistribusi uniform).

3. *Weighted random pairing*

Dalam *weighted random pairing*, peluang terpilihnya individu menjadi *parent* bergantung pada nilai *cost* yang dimiliki individu tersebut. Kromosom dengan nilai *cost* yang lebih rendah memiliki peluang yang lebih besar untuk terpilih.

4. *Tournament selection*

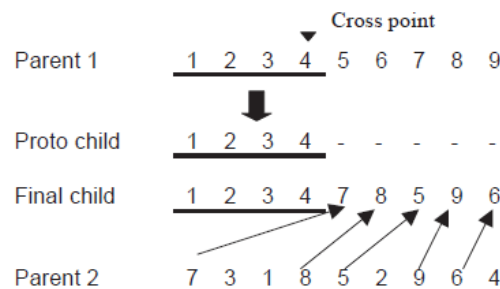
Tournament selection dilakukan dengan memilih sekumpulan kromosom dari populasi dan melakukan seleksi dengan aturan tertentu terhadap kromosom yang terpilih. *Tournament selection* efektif untuk GA dengan populasi yang besar karena tidak perlu melakukan *sorting* terhadap individu di dalam populasi.

2.5.4. Persilangan (*Crossover*)

Persilangan merupakan proses pertukaran informasi antara dua individu untuk menghasilkan keturunan baru yang memiliki sebagian karakteristik dari kedua individu tersebut. Dalam GA, terdapat beberapa jenis metode persilangan untuk pengkodean permutasi (Nearchou, 2004):

1. *One-Point Crossover*

One-point crossover dilakukan dengan menentukan satu titik *crossover* secara acak, kemudian gen keturunan dari titik pertama hingga titik *crossover* akan diisi dengan gen dari *parent* pertama dan sisanya diisi dari *parent* kedua. Gambar 2.2 menunjukkan ilustrasi *one-point crossover*.

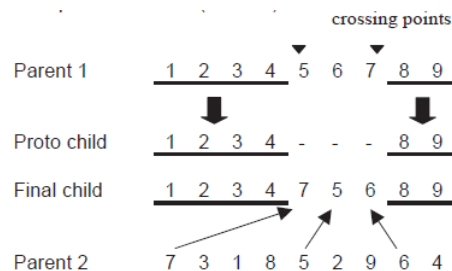


Gambar 2.2 One-Point Crossover

(Sumber: Nearchou, 2004)

2. *Two-Point Crossover model pertama*

Metode *two-point crossover* model pertama diawali dengan memilih dua titik *crossover* secara acak. Gen pada posisi sebelum titik *crossover* pertama dan sesudah titik *crossover* kedua akan diisi oleh gen dari *parent* pertama, sedangkan posisi di antara dua titik *crossover* akan diisi oleh *parent* kedua. Gambar 2.3 menunjukkan ilustrasi *two-point crossover* model pertama.

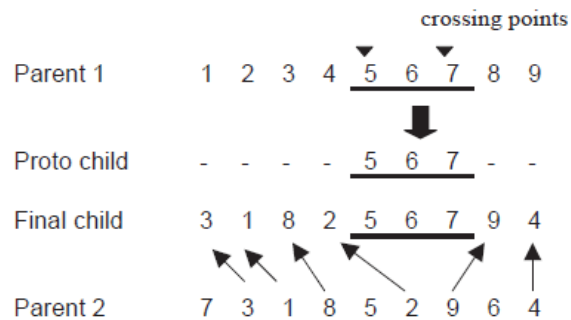


Gambar 2.3 Two-Point Crossover Model Pertama

(Sumber: Nearchou, 2004)

3. *Two-Point Crossover model kedua*

Metode *two-point crossover* model kedua diawali dengan memilih dua titik *crossover* secara acak. Gen pada posisi di antara dua titik *crossover* akan diisi oleh *parent* pertama dan sisanya diisi oleh *parent* kedua. Gambar 2.4 menunjukkan ilustrasi *two-point crossover* model kedua.

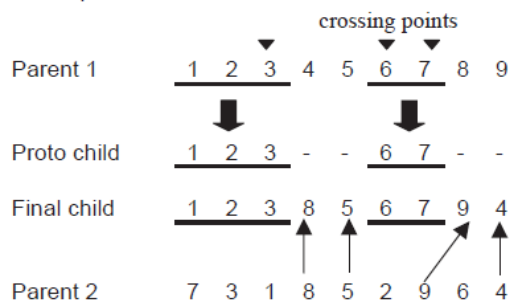


Gambar 2.4 Two-Point Crossover Model Kedua

(Sumber: Nearchou, 2004)

4. *Three-Point Crossover*

Metode *three-point crossover* diawali dengan memilih tiga titik *crossover* secara acak. *Parent* pertama akan mengisi gen pada posisi pertama hingga ke titik *crossover* pertama dan mengisi gen pada posisi di antara dua titik *crossover* berikutnya. Gen pada posisi selain yang diisi oleh *parent* pertama akan diisi dengan gen dari *parent* kedua. Gambar 2.5 menunjukkan ilustrasi *three-point crossover*.

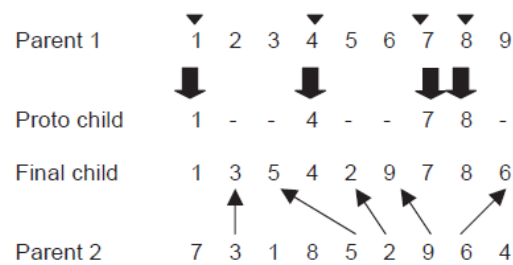


Gambar 2.5 Three-Point Crossover

(Sumber: Nearchou, 2004)

5. *Position-Based Crossover*

Metode *position-based crossover* diawali dengan memilih sejumlah posisi dari *parent* pertama secara acak. Gen yang berada pada posisi tersebut akan mengisi gen untuk keturunan dengan posisi yang sama. Gen pada posisi selain yang telah diisi oleh *parent* pertama akan diisi dengan gen dari *parent* kedua. Gambar 2.6 menunjukkan ilustrasi *position-based crossover*.



Gambar 2.6 Position-Based Crossover

(Sumber: Nearchou, 2004)

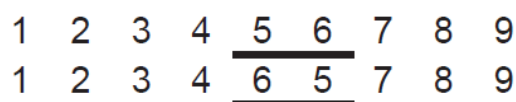
2.5.5. Mutasi

Mutasi merupakan proses perubahan gen. Mutasi diperlukan untuk menjaga keragaman individu dalam populasi dan membantu proses optimasi agar tidak terjebak dalam lokal optima. Mutasi berlangsung dengan merubah posisi satu atau lebih gen ke posisi lain dalam satu kromosom sesuai aturan-aturan tertentu. Untuk menjaga agar mutasi tidak terjadi secara terus-menerus, biasanya ditetapkan peluang terjadinya mutasi.

Beberapa metode mutasi yang dapat digunakan dalam pengkodean permutasi meliputi (Nearchou, 2004):

1. *Adjacent Exchange Mutation*

Adjacent Exchange Mutation dilakukan dengan menukar dua gen yang posisinya bersebelahan satu sama lain. Gambar 2.7 mengilustrasikan *adjacent exchange mutation*.



Gambar 2.7 Adjacent Exchange Mutation

(Sumber: Nearchou, 2004)

2. *Random Exchange Mutation*

Random Exchange Mutation dilakukan dengan menukar dua gen yang posisinya dipilih secara acak. Gambar 2.8 mengilustrasikan *random exchange mutation*.

1	2	<u>3</u>	4	5	6	7	<u>8</u>	9
1	2	<u>8</u>	4	5	6	7	<u>3</u>	9

Gambar 2.8 Random Exchange Mutation

(Sumber: Nearchou, 2004)

3. *Shift Mutation*

Shift Mutation dilakukan dengan memilih satu gen secara acak, kemudian posisinya dipindah ke titik lain dan menggeser gen-gen yang berada setelah gen yang dipilih secara acak. Gambar 2.9 mengilustrasikan *shift mutation*.

1	2	3	4	5	<u>6</u>	7	8	9
1	<u>6</u>	2	3	4	<u>5</u>	7	8	9

Gambar 2.9 Shift Mutation

(Sumber: Nearchou, 2004)

4. *Displacement Mutation*

Displacement Mutation dilakukan dengan memilih sejumlah gen yang berurutan secara acak kemudian menggeser posisinya ke kanan atau ke kiri. Gambar 2.10 mengilustrasikan *displacement mutation*.

1	2	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>	7	8	9
1	2	<u>7</u>	<u>8</u>	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>	9

Gambar 2.10 Displacement Mutation

(Sumber: Nearchou, 2004)

5. *Inversion Mutation*

Inversion Mutation dilakukan dengan dua titik secara acak dan kemudian membalikkan urutan gen di antara dua titik tersebut. Gambar 2.11 mengilustrasikan *inversion mutation*.

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

Gambar 2.11 Inversion Mutation

(Sumber: Nearchou, 2004)

6. *Inversion/Displacement Mutation*

Inversion/Displacement Mutation dilakukan dengan dua titik secara acak dan kemudian membalikkan urutan gen di antara dua titik tersebut. Langkah terakhir yaitu menggeser posisi gen-gen tersebut ke kanan atau ke kiri secara acak. Gambar 2.12 mengilustrasikan *inversion-displacement mutation*.

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

Gambar 2.12 Inversion/Displacement Mutation

(Sumber: Nearchou, 2004)

2.6. Multi-Objective Optimization

2.6.1. Konsep dan Pengertian

Optimasi adalah proses membuat sesuatu menjadi lebih baik. Tujuan optimasi adalah menemukan solusi yang dapat meminimalkan atau memaksimalkan tujuan tunggal dengan menjaga batasan yang dimiliki dari suatu sistem atau proses (Ngatchou et al., 2005). Solusi optimasi menghasilkan nilai tunggal yang menggambarkan kompromi diantara semua tujuan.

Dalam proses optimasi yang memiliki lebih dari satu fungsi tujuan, pada umumnya yang dilakukan adalah menggabungkan beberapa fungsi tujuan menjadi satu fungsi tujuan tunggal, baik dengan memberikan bobot untuk masing-masing fungsi, maupun mengubah fungsi tujuan lain menjadi fungsi pembatas. Secara umum beberapa kekurangan utama dari pendekatan ini meliputi (Ngatchou et al., 2005):

1. Membutuhkan pengetahuan terkait tingkat kepentingan dan prioritas dari masing-masing tujuan.
2. Fungsi agregat hanya menghasilkan satu solusi tunggal.

3. *Trade-off* antara fungsi tujuan satu dengan yang lain tidak dapat dievaluasi.
4. Solusi mungkin tidak dapat dicapai kecuali semesta pencarian berbentuk konveks.

Terdapat beberapa metode klasik yang kerap digunakan untuk melakukan optimasi dengan fungsi tunggal (Ngatchou et al., 2005):

1. Agregasi Berbobot

Metode ini mengubah permasalahan *multi-objective* menjadi *single-objective* dengan persamaan:

$$\text{Minimize } Z = \sum_{j=1}^N w_j f_j(x) \text{ dengan } w_j \geq 0 \text{ dan } \sum_{j=1}^N w_j = 1 \dots\dots\dots(2.1)$$

Kekurangan dari metode ini adalah kesulitan dalam menentukan bobot apabila tidak memiliki informasi yang cukup.

2. Goal Programming

Goal programming berupaya meminimasi deviasi/penyimpangan nilai optimasi dengan target yang sudah ditentukan. Persamaan yang umum digunakan adalah:

$$\text{Minimize } Z = \sum_{j=1}^N w_j |f_j(\bar{x}) - T_j| \dots\dots\dots(2.2)$$

Kekurangan dari metode ini adalah perlu menentukan bobot/prioritas dan target awal yang harus dicapai dalam optimasi.

3. ϵ -Constraint

Metode ini berupaya mencari solusi pareto optimal dari satu fungsi tujuan dengan menjadikan fungsi tujuan lain sebagai *constraint* yang memiliki nilai jangkauan ϵ_i . Penyelesaian dilakukan beberapa kali untuk nilai ϵ_i untuk mendapatkan pareto set. Kekurangan metode ini adalah proses optimasi membutuhkan waktu komputasi yang tinggi serta solusi-solusi yang dihasilkan tidak bersifat global optimal.

Permasalahan yang melibatkan lebih dari satu fungsi tujuan disebut dengan *multi-objective optimization* (MO). Dalam MO, analisis yang seringkali diperhatikan adalah *trade-off* antara fungsi satu tujuan dengan fungsi tujuan lain. Tidak terdapat satu solusi tunggal yang bersifat unik dalam MO, melainkan terdapat sekumpulan solusi optimal yang memiliki *trade-off* yang dapat diterima. MO merupakan salah satu langkah analisis dari *multi-criteria decision making* (MCDM), di mana pengambil keputusan akan

memilih solusi dari sekumpulan solusi yang dianggap optimal. Permasalahan MO yang memiliki lebih dari tiga fungsi tujuan disebut dengan *many-objective optimization problems*.

2.6.2. Pareto Optimality

Multi-objective Optimization (MO) merupakan permasalahan yang berupaya untuk mengoptimalkan sejumlah k fungsi tujuan secara simultan, digambarkan sebagai vektor $F(x)$ (Chiandussi et al., 2012):

$$F(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \\ f_3(x) \\ \vdots \\ f_k(x) \end{bmatrix} \dots\dots\dots (2.3)$$

Berbeda dengan fungsi tujuan tunggal, dalam MO terdapat sekumpulan solusi yang memiliki nilai yang berbeda-beda. Seorang pengambil keputusan/*decision maker* perlu memilih satu atau lebih solusi dengan memilih satu atau lebih vektor.

Secara matematis, permasalahan MO dapat digambarkan sebagai permasalahan minimasi (atau maksimasi) dari sekumpulan fungsi tujuan $F(x) = (f_1(x), f_2(x), \dots, f_k(x))$ yang menjadi subyek dari pembatas pertidaksamaan $g_i(x) \leq 0$, $i = \{1, \dots, m\}$, dan pembatas persamaan $h_j(x) = 0$, $j = \{1, \dots, p\}$. solusi dari MO meminimasi (atau memaksimalkan) komponen dari vektor $F(x)$ di mana x adalah vektor variabel keputusan berdimensi n di mana $x = (x_1, x_2, \dots, x_n)$ dari semesta Ω (Chiandussi et al., 2012).

Untuk membandingkan solusi dalam permasalahan MO, konsep *pareto dominance* dan *pareto optimality* sering digunakan. Solusi termasuk ke dalam himpunan Pareto apabila tidak ada solusi lain yang dapat meminimalkan (atau memaksimalkan) satu tujuan tanpa menaikkan (atau menurunkan) nilai dari tujuan yang lain. Secara matematis konsep *pareto dominance* dapat digambarkan sebagai berikut (Ngatchou et al., 2005):

Terdapat $x, y \in \Omega$, x mendominasi y , dinotasikan $x \prec y$ apabila:

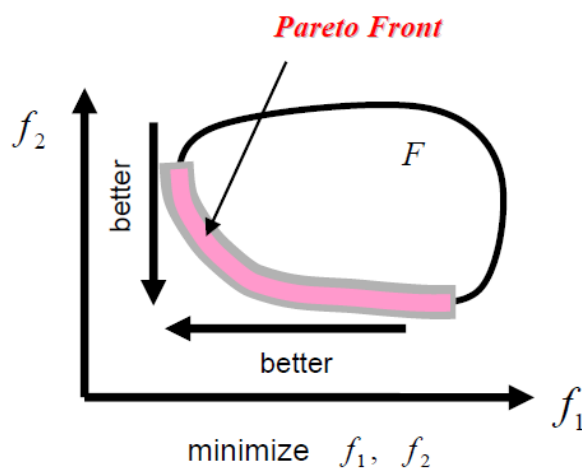
$$\begin{aligned} \forall i \in (1, 2, \dots, M): F_i(x) &\leq F_i(y) \wedge \\ \exists j \in (1, 2, \dots, M): F_j(x) &< F_j(y) \dots\dots\dots (2.4) \end{aligned}$$

Sekumpulan solusi yang optimal secara pareto disebut sebagai himpunan Pareto Optimal. Vektor dari fungsi tujuan setiap solusi digambarkan berada dalam *Front Pareto*.

Metode optimasi dalam permasalahan MO harus dapat mencapai tujuan-tujuan berikut (Zitzler, 2000):

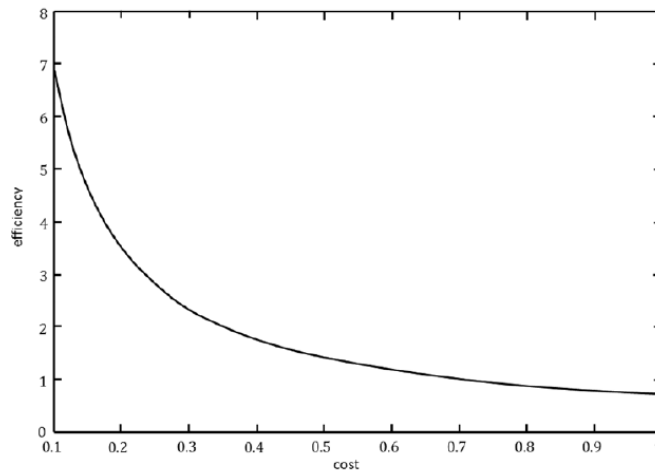
1. *Front Pareto* terbaik yang diketahui harus sedekat mungkin dengan nilai *Front Pareto* yang sesungguhnya. Idealnya, himpunan Pareto terbaik yang diketahui (*best-known Pareto set*) merupakan bagian dari himpunan Pareto optimal. Tujuan ini mendorong optimasi untuk fokus mencari solusi dalam area tertentu dalam *front pareto*.
2. Solusi dalam Pareto terbaik harus tersebar secara merata (*uniformly distributed*) dan beragam sepanjang nilai *Front Pareto* agar dapat memberikan gambaran *trade-off* bagi pengambil keputusan. Tujuan ini mendorong pencarian secara merata ke seluruh *front pareto*.
3. *Front Pareto* terbaik harus menggambarkan keseluruhan spektrum dari *Front Pareto*. Tujuan ini mendorong pencarian pada titik ekstrim dari solusi-solusi yang mungkin.

Gambar 2.13 Menunjukkan ilustrasi *Front Pareto* dari permasalahan minimasi antara f_1 dan f_2 . Sedangkan gambar 2.14 Menunjukkan *Front Pareto* sekaligus *trade-off* antara dua fungsi tujuan, yakni biaya dan efisiensi.



Gambar 2.13 *Front Pareto* dalam Permasalahan Minimasi

(Sumber: Ngatchou et al., 2005)



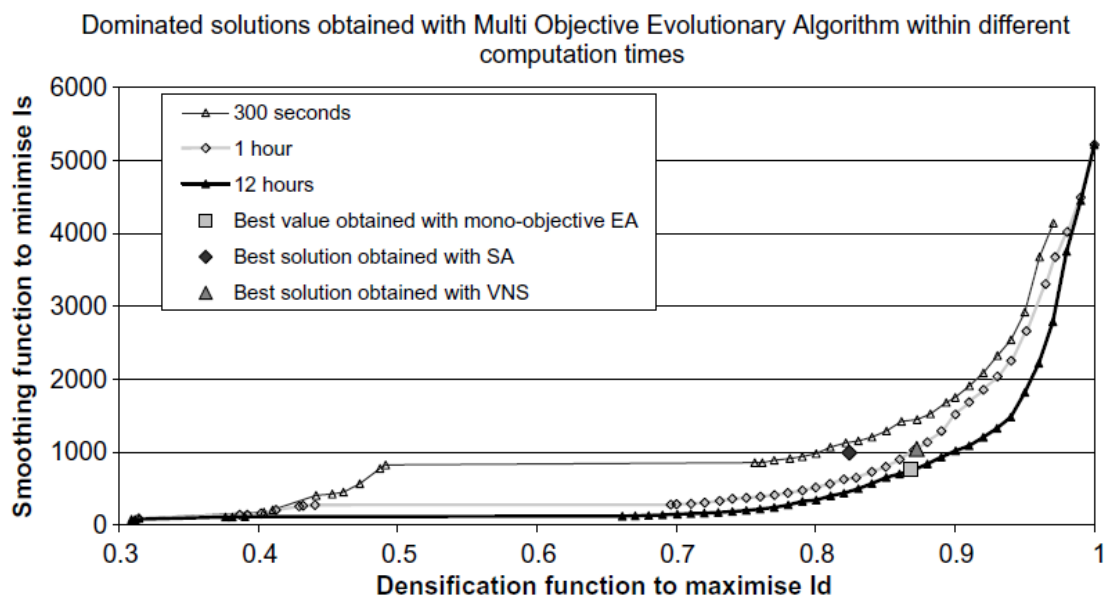
Gambar 2.14 *Front Pareto dan Trade-off Antara Biaya (Cost) dan Efisiensi*

(Sumber: Chiandusi et al., 2012)

2.7. Multi-Objective Evolutionary Algorithm (MOEA)

2.7.1. Konsep dan Pengertian

Multi-Objective Evolutionary Algorithm (MOEA) merupakan modifikasi dari GA untuk menangani permasalahan MO. Kemampuan GA untuk secara simultan mencari solusi di area yang berbeda-beda dalam semesta solusi memungkinkan GA untuk menemukan sekumpulan solusi yang beragam untuk permasalahan yang bersifat kompleks (Konak et al., 2006). Kelebihan MOEA adalah sebagian besar metode MOEA tidak memerlukan pengguna untuk memprioritaskan dan menentukan skala atau bobot untuk masing-masing tujuan. Kelebihan lainnya yaitu sesuai dengan konsep MO, solusi yang dihasilkan tidak bersifat unik, tetapi terdapat sekumpulan solusi yang memiliki *trade-off* masing-masing dan menjadi keputusan dari pengguna untuk memilih solusi yang akan digunakan. Gambar 2.15 menunjukkan hasil optimasi dari beberapa metode meta-heuristik dan hasilnya metode MOEA dengan waktu komputasi 12 jam memberikan output yang paling baik dalam permasalahan penjadwalan *sequence* produksi.



Gambar 2.15 Perbandingan Metode Optimasi Penjadwalan *Sequence* Produksi

(Sumber: Joly dan Frein, 2008)

Beragam jenis metode MOEA telah dikembangkan oleh berbagai peneliti. Beberapa model MOEA yang cukup dikenal yaitu SPEA2 (Zitzler et al., 2001), NSGA-II (Deb et al., 2002), PESA (Corne et al., 2000), ϵ -MOEA (Deb et al., 2005a) dan MOEA/D (Zhang dan Li, 2007). Masing-masing model memiliki karakteristik dan pendekatan masing-masing dalam melakukan optimasi, seperti SPEA2 yang memiliki dua populasi, yakni populasi iterasi yang digunakan dalam proses iterasi dan persilangan, dan populasi arsip yang berisi solusi-solusi optimal, sehingga solusi yang dihasilkan lebih mendekati nilai Pareto aktual namun waktu komputasi yang diperlukan sangat lama. Sedangkan NSGA-II hanya memiliki satu populasi yakni populasi iterasi dan memiliki keakuratan yang lebih rendah dibandingkan SPEA2, tetapi memiliki waktu komputasi yang jauh lebih cepat. Kedua kekurangan metode tersebut diperbaiki dalam ϵ -MOEA dengan menghasilkan solusi-solusi yang akurat dengan waktu komputasi yang cepat. Namun, kelemahan ϵ -MOEA adalah beberapa solusi dapat hilang akibat pemilihan nilai ϵ yang tidak tepat (Hernandez-Diaz et al., 2007). Selain itu, dalam ϵ -MOEA perlu mengetahui batas bawah dan batas atas masing-masing fungsi tujuan sebelum melakukan optimasi.

Metode GrEA, yang memanfaatkan konsep *grid dominance* dan mengembangkan metode NSGA-II dan ϵ -MOEA, terbukti menghasilkan performa yang lebih baik dibandingkan metode lainnya dalam mencapai konvergen dan menjaga keragaman solusi yang dihasilkan untuk permasalahan *many-objective optimization problem*. Tabel 2.2 menunjukkan rata-rata dan simpangan baku indikator dari hasil simulasi metode-metode MOEA untuk beberapa permasalahan berbeda. Metode GrEA memberikan hasil yang lebih baik dan mendominasi sebagian besar permasalahan dibandingkan dengan beberapa metode MOEA yang lain seperti POGA (Di Pierro et al., 2007), HypE (Bader dan Zitzler, 2011), MSOPS (Hughes, 2003).

Tabel 2.2 Perbandingan Metode MOEA

Problem	Metode Dominan
DTLZ1	GrEA
DTLZ2	GrEA
DTLZ3	ϵ -MOEA, MOEA/D
DTLZ4	GrEA
DTLZ5	MSOPS
DTLZ6	MOEA/D
DTLZ7	GrEA
DTLZ5 (I,M)	GrEA
MOTSP	GrEA

(Yang et al., 2013)

Permasalahan DTLZ2, DTLZ4, DTLZ5, dan DTLZ7 digunakan untuk menguji kemampuan algoritma mengatasi permasalahan dengan bentuk dan lokasi yang berbeda. Sementara DTLZ1, DTLZ3 dan DTLZ6 digunakan untuk menguji kemampuan algoritma untuk permasalahan yang lebih rumit (Yang et al., 2013). Penjelasan detail permasalahan DTLZ 1 hingga DTLZ 7 dapat dilihat pada Deb et al. (2005b). Permasalahan DTLZ5(I,M) adalah modifikasi permasalahan DTLZ5 di mana dimensionalitas I dan jumlah tujuan M dapat ditentukan oleh pengguna (Deb dan Saxena, 2006). Permasalahan MOTSP (*Multiobjective Traveling Salesperson Problem*) merupakan permasalahan kombinatorial yang berupaya meminimasi lebih dari satu tujuan dan merupakan pengembangan dari *Traveling Salesman Problem* (Corne dan Knowles, 2007).

2.7.2. Indikator Kualitas MOEA

Terdapat dua tujuan yang ingin dicapai saat mengukur output MOEA (Zhou et al., 2011):

1. Konvergensi terhadap *true pareto front*
2. Distribusi dari solusi yang dihasilkan

Elemen dari vektor yang menggambarkan performansi dari MOEA disebut dengan *quality indicators*. *Quality indicator* digunakan untuk membandingkan antara dua himpunan solusi dari algoritma yang berbeda atau diintegrasikan ke dalam algoritma sebagai indikator untuk mengukur konvergensi solusi selama simulasi. Berdasarkan metode perbandingannya, indikator kualitas dibagi menjadi *unary quality indicators*, kombinasi *unary quality indicators*, dan *binary quality indicator* (Zitzler et al., 2003). *Unary quality indicators* adalah indikator kualitas yang mengukur kualitas hanya dengan satu himpunan solusi saja, sedangkan *binary quality indicator* mengukur kualitas dengan menggunakan sepasang himpunan solusi yang berbeda. Beberapa contoh *unary quality indicators* adalah *enclosing hypercube indicator*, *objective vector indicator*, *hypervolume indicator*, *average best weight combination*, *unary ε -indicator*, dan *error ratio*. Kelemahan dari *unary quality indicator* adalah tidak dapat digunakan untuk mengukur hubungan antara himpunan solusi satu dengan yang lain. Hanya *hypervolume indicator* yang dapat digunakan untuk menentukan apakah himpunan solusi A tidak lebih buruk dibandingkan himpunan solusi B, tetapi tidak dapat diketahui apakah himpunan solusi A lebih baik dibandingkan himpunan solusi B.

Kelemahan *unary quality indicator* tidak dimiliki *binary quality indicator*. *Binary quality indicator* dapat digunakan untuk menentukan berbagai tingkatan perbandingan antara dua himpunan solusi. Tabel 2.3 menunjukkan beberapa *binary quality indicator* beserta kriteria penentuan relasi yang digunakan untuk permasalahan minimasi.

Tabel 2.3 Relasi dan Perbandingan Dua Himpunan Solusi

Indikator	Simbol	Relasi					
		\gg	$>$	\triangleright	\geq	$=$	\parallel
		Dominan mutlak	Dominan	Lebih baik	Dominansi lemah	Sebanding	Tidak dapat dibandingkan
Epsilon	I_ε	$I_\varepsilon(A,B) < 1$	-	$I_\varepsilon(A,B) \leq 1$ $I_\varepsilon(B,A) > 1$	$I_\varepsilon(A,B) \leq 1$	$I_\varepsilon(A,B) = 1$ $I_\varepsilon(B,A) = 1$	$I_\varepsilon(A,B) > 1$ $I_\varepsilon(B,A) > 1$
Additive Epsilon	$I_{\varepsilon+}$	$I_{\varepsilon+}(A,B) < 0$	-	$I_{\varepsilon+}(A,B) \leq 0$ $I_{\varepsilon+}(B,A) > 0$	$I_{\varepsilon+}(A,B) \leq 0$	$I_{\varepsilon+}(A,B) = 0$ $I_{\varepsilon+}(B,A) = 0$	$I_{\varepsilon+}(A,B) > 0$ $I_{\varepsilon+}(B,A) > 0$
Binary Hypervolume	I_{H2}	-	-	$I_H(A,B) > 0$ $I_H(B,A) = 0$	$I_H(A,B) \geq 0$ $I_H(B,A) = 0$	$I_H(A,B) = 0$ $I_H(B,A) = 0$	$I_H(A,B) > 0$ $I_H(B,A) > 0$
Coverage	I_C	-	$I_C(A,B) = 1$ $I_C(B,A) = 0$	$I_C(A,B) = 1$ $I_C(B,A) < 1$	$I_C(A,B) = 1$	$I_C(A,B) = 1$ $I_C(B,A) = 1$	$0 < I_C(A,B) < 1$ $0 < I_C(B,A) < 1$

(Zitzler et al., 2003)

Selain beberapa indikator tersebut, terdapat beberapa indikator lain seperti *utility function R1*, *utility function R2*, *utility function R3* dan *lines of intersection*. Namun, indikator-indikator tersebut tidak dapat digunakan untuk menentukan relasi antara dua himpunan solusi.

Epsilon indicator (I_ε) menggunakan aturan dominansi epsilon, yakni dalam permasalahan minimasi, suatu solusi bersifat mendominasi apabila nilainya lebih kecil atau sama dengan nilai solusi kedua dikalikan faktor ε , sesuai persamaan (Zitzler et al., 2003):

$$\forall 1 \leq i \leq n : z_i^1 \leq \varepsilon \cdot z_i^2 \quad \varepsilon > 0 \dots\dots\dots (2.5)$$

$I_\varepsilon(A, B)$ menunjukkan nilai minimal faktor ε agar terdapat paling sedikit satu solusi dari himpunan A yang tidak didominasi oleh seluruh himpunan solusi B.

$$I_\varepsilon(A, B) = \inf_{\varepsilon \in R} \{ \forall z^2 \in B \exists z^1 \in A : z^1 \succ_\varepsilon z^2 \vee z^1 =_\varepsilon z^2 \} \dots\dots\dots (2.6)$$

$$I_\varepsilon(A, B) = \max_{z^2 \in B} \min_{z^1 \in A} \max_{1 \leq i \leq n} \frac{z_i^1}{z_i^2} \dots\dots\dots (2.7)$$

Additive epsilon indicator ($I_{\varepsilon+}$) memiliki makna yang sama dengan $I_\varepsilon(A, B)$, hanya saja aturan dominansi yang dilakukan berdasarkan aturan penjumlahan, sesuai persamaan (Zitzler et al., 2003):

$$\forall 1 \leq i \leq n : z_i^1 \leq \varepsilon + z_i^2 \quad \varepsilon > 0 \dots\dots\dots (2.8)$$

$$I_{\varepsilon+}(A, B) = \max_{z^2 \in B} \min_{z^1 \in A} \max_{1 \leq i \leq n} z_i^1 - z_i^2 \dots\dots\dots (2.9)$$

Binary hypervolume indicator (I_{H2}) merupakan turunan dari *hypervolume indicator* (I_H). *Hypervolume indicator* mengukur volume atau area yang didominasi oleh

solusi-solusi dari himpunan solusi yang dihasilkan suatu algoritma. Volume yang diukur adalah volume antara titik-titik solusi pada *pareto front* dengan titik referensi (*reference point/nadir point*), sesuai persamaan (Guerrero et al., 2010):

$$I_H(A) = \text{volume}(\cup_{\forall a \in A, \forall n \in N} \text{hypercube}(a, n)) \dots\dots\dots (2.10)$$

$I_H(A)$ = Indikator *Hypervolume* dari himpunan solusi A

hypercube = luas dari bidang yang dibentuk antara titik koordinat solusi a dengan titik-titik n

N = himpunan titik referensi/nilai terburuk dari setiap fungsi tujuan

Nilai I_{H2} diukur melalui persamaan (Guerrero et al., 2010):

$$I_{H2}(A, B) = \begin{cases} I_H(A) - I_H(B) & \text{if } \forall z^2 \in B \exists z^1 \in A : z^1 \succ z^2 \\ I_H(A + B) - I_H(B) & \text{lainnya} \end{cases} \dots\dots\dots (2.11)$$

Coverage indicator (I_C) digunakan untuk mengukur berapa proporsi himpunan solusi A yang mendominasi seluruh himpunan solusi B dengan nilai interval [0,1]. Nilai 1 menunjukkan bahwa seluruh himpunan solusi A mendominasi himpunan solusi B, sedangkan nilai 0 menunjukkan himpunan solusi B tidak didominasi oleh himpunan solusi A. Perhitungan nilai I_C dirumuskan dalam persamaan (Zitzler et al., 1999):

$$I_C(A, B) = \frac{|\{z^2 \in B \exists z^1 \in A : z^1 \succ z^2 \vee z^1 = z^2\}|}{|A|} \dots\dots\dots (2.12)$$

Selain beberapa *binary quality indicator* tersebut, terdapat juga indikator *Mutual Domination Rate* (MDR). Indikator MDR memiliki prinsip yang hampir sama dengan I_C , hanya saja dalam MDR yang diperhitungkan hanyalah solusi-solusi himpunan B yang didominasi oleh himpunan A, sesuai persamaan (Guerrero et al., 2010):

$$I_{MDR}(A, B) = \frac{\|\Delta(B, A)\|}{\|A\|} - \frac{\|\Delta(A, B)\|}{\|A\|} \dots\dots\dots (2.13)$$

$\Delta(B, A)$ = solusi himpunan B yang didominasi paling sedikit 1 solusi himpunan A

$\|A\|$ = norma (panjang) vektor himpunan A

Nilai $I_{MDR} = 1$ menunjukkan bahwa himpunan solusi A lebih baik dibandingkan himpunan solusi B. Nilai $I_{MDR} = 0$ menunjukkan bahwa tidak terdapat perbedaan antara himpunan A dan B, dan jika nilai $I_{MDR} = -1$ berarti himpunan solusi A lebih buruk dibandingkan himpunan solusi B.

2.7.3. Kriteria Penghentian Algoritma MOEA

Dalam GA maupun MOEA, di setiap akhir iterasi dilakukan cek konvergensi untuk menentukan apakah algoritma sudah mencapai kondisi yang diinginkan. Algoritma akan berhenti ketika kondisi tertentu, yang disebut *stopping criteria*, sudah terpenuhi. Metode *stopping criteria* yang paling sederhana adalah dengan menggunakan jumlah iterasi (Liu et al., 2018). Algoritma akan berhenti ketika sudah mencapai iterasi tertentu yang sudah ditetapkan. Metode ini sangat bergantung kepada jenis permasalahan, di mana dalam permasalahan tertentu jumlah iterasi yang dibutuhkan bisa lebih besar dibandingkan dengan jumlah yang ditentukan. Jika hal ini terjadi, maka algoritma tidak dapat mencapai titik optimal. Namun, jika jumlah iterasi lebih besar daripada yang dibutuhkan, maka iterasi yang dilakukan tidak memberikan perubahan yang berarti terhadap proses optimasi dan hanya menghabiskan waktu.

Algoritma harus berhenti apabila populasi yang dihasilkan sudah konvergen dan hanya akan kehilangan keragaman apabila iterasi dilanjutkan, atau ketika sudah mencapai solusi optimal dan iterasi selanjutnya tidak akan menghasilkan solusi yang lebih baik. Untuk mengukur konvergensi solusi yang dihasilkan, digunakan *progress indicator/quality indicator* seperti yang sudah dibahas di subbab 2.7.2s. Nilai *quality indicator* maupun pengolahannya akan digunakan sebagai dasar untuk menentukan apakah algoritma harus berhenti sesuai dengan *stopping criteria* yang ditetapkan.

Terdapat sejumlah *stopping criteria* yang dapat diterapkan dalam MOEA. Yang pertama adalah menetapkan ambang batas atau *threshold* (β) dari nilai *quality indicator* (*QI*), dengan persamaan (Liu et al., 2018):

$$Terminate? \begin{cases} Yes & \text{if } QI \leq (or \geq) \beta \\ No & \text{otherwise} \end{cases} \dots\dots\dots (2.14)$$

Penggunaan jumlah iterasi sebagai *stopping criteria* juga termasuk dalam jenis ini. Kekurangan dari metode ini adalah nilai *threshold* sangat bergantung pada jenis permasalahan dan sulit ditentukan tanpa pengetahuan yang memadai.

Metode kedua yang dapat digunakan yakni dengan menggunakan uji statistik untuk menguji hipotesis terkait *quality indicator* yang dihasilkan dari beberapa iterasi. Metode *offline convergence detection* (OFCD) yang dikembangkan Trautmann et al. (2008) menggunakan uji *kolmogorov-smirnov* untuk menarik kesimpulan apakah nilai 3

quality indicator dari dua *running* algoritma yang berbeda berasal dari distribusi yang sama. Algoritma A dijalankan sebanyak α iterasi terlebih dahulu, kemudian algoritma B dijalankan sebanyak β iterasi, dengan nilai $\alpha \ll \beta$. Uji hipotesis dilakukan setiap iterasi selesai pada algoritma B setelah 5 iterasi pertama. Algoritma akan berhenti ketika hipotesis bahwa populasi berasal dari distribusi berbeda ditolak sebanyak tiga kali ($p\text{-value} > 0.05$) untuk seluruh *quality indicator*. Keunggulan dari metode ini adalah dapat mendeteksi kapan algoritma mencapai optimalitas dengan akurat. Sedangkan kelemahannya adalah waktu komputasi yang tinggi karena perlu melakukan pengulangan algoritma yang sama dengan jumlah iterasi yang berbeda.

Trautmann et al.(2009) mengembangkan metode *online convergence detection* (OCD) untuk mengatasi kelemahan yang dimiliki oleh metode OFC. Berbeda dengan OFCD, dalam OCD algoritma hanya dijalankan satu kali. Terdapat 2 uji hipotesis yang dilakukan dengan membandingkan nilai 3 *quality indicator* antara 2 iterasi berturut-turut. Yang pertama adalah uji *Chi-Square* untuk menguji apakah nilai standar deviasi dari *quality indicator* mencapai batas yang ditentukan. Yang kedua adalah melakukan analisis regresi linear tanpa *intercept* dan menguji signifikansi nilai koefisien *slope* menggunakan uji t. Algoritma berhenti ketika salah satu uji hipotesis mengidentifikasi tercapainya konvergensi atau ketika jumlah iterasi mencapai batas yang ditentukan.

Guerrero et al. (2010) menerapkan regresi linear (*ordinary least square*) untuk digunakan sebagai *stopping criteria*, dan disebut dengan *least square stopping criteria* (LSSC). Prinsipnya adalah dengan mengamati kapan *quality indicator* menjadi linear seiring jumlah iterasi yang dilakukan, sehingga tingkat perubahan solusi yang dihasilkan tiap iterasinya menjadi rendah dan mencapai titik di bawah nilai *threshold*. LSSC mengambil nilai *quality indicator* dari sejumlah w_l iterasi iterakhir untuk dimodelkan dalam regresi, sehingga diketahui nilai residual dan nilai koefisien *slope* dari jumlah iterasi terhadap nilai *quality indicator*. Algoritma akan berhenti ketika nilai *slope* dan residual mencapai *threshold* yang ditentukan berdasarkan teorema Chebysev. Kelebihan dari metode ini adalah implementasi yang mudah, sederhana dan tidak membutuhkan waktu komputasi yang tinggi.

2.8. Uji Perbandingan Rataan

Uji perbandingan rataan adalah uji statistik yang digunakan untuk menguji hipotesis bahwa rataan antara dua sampel data adalah sama ($\mu_1 = \mu_2$) (Montgomery dan Runger, 2014). Hipotesis tandingan pada uji ini adalah bahwa terdapat perbedaan rataan yang signifikan antara kedua sampel data. Terdapat beberapa uji yang dapat digunakan untuk menguji perbedaan rataan (Montgomery dan Runger, 2014):

1. Berdistribusi normal, variansi populasi diketahui (Uji Z)

Uji ini dilakukan ketika kedua sampel data berdistribusi normal dan variansi populasinya diketahui. Kedua sampel data bersifat independen. Nilai statistik uji Z dihitung dengan menggunakan:

$$Z = \frac{\bar{X}_1 - \bar{X}_2 - (\mu_1 - \mu_2)}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} \dots\dots\dots (2.15)$$

2. Berdistribusi normal, variansi populasi tidak diketahui (Uji t)

Uji ini dilakukan ketika kedua sampel data berdistribusi normal dan variansi populasinya tidak diketahui. Kedua sampel data bersifat independen. Nilai statistik uji t dihitung dengan menggunakan:

$$t = \frac{\bar{X}_1 - \bar{X}_2 - (\mu_1 - \mu_2)}{S_p \sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} \dots\dots\dots (2.16)$$

$$S_p = \frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{n_1 + n_2 - 2} \dots\dots\dots (2.17)$$

3. Uji perbandingan sampel berpasangan (Uji t berpasangan)

Dalam uji statistik, sampel berpasangan artinya adalah kedua sampel data saling berpasangan. Satu kelompok sampel data diambil dalam kondisi yang sama, sedangkan pasangannya diambil dari sampel yang sama dengan kondisi yang berbeda. Contoh penerapannya adalah menguji perbedaan nilai ujian sebelum dan sesudah menjalani pelatihan, sampelnya sama, tetapi yang pertama diambil sebelum menjalani pelatihan dan pasangannya diambil setelah menjalani pelatihan. Uji ini lebih *powerful* dibandingkan uji dengan sampel independen. Nilai statistik uji t dihitung dengan menggunakan:

$$t = \frac{\bar{D} - (\mu_1 - \mu_2)}{S_D / \sqrt{n}} \dots\dots\dots (2.18)$$

4. Uji nonparametrik sampel berpasangan (Uji Wilcoxon)

Uji nonparametrik adalah uji hipotesis yang bebas asumsi, artinya tidak mensyaratkan sampel harus mengikuti distribusi tertentu. Uji Wilcoxon adalah alternatif bagi uji t dan dapat digunakan baik untuk sampel independen maupun sampel berpasangan. Uji Wilcoxon menghitung jumlah rank pada setiap sampel.

$$W_2 = \frac{(n_1 + n_2)(n_1 + n_2 + 1)}{2} - W_1 \dots\dots\dots (2.19)$$

2.9. Regresi Linear

Analisis regresi adalah studi tentang ketergantungan suatu variabel dependen terhadap satu atau lebih variabel lainnya, yang disebut variabel independen, dengan tujuan mengestimasi dan atau memprediksi nilai rata-ran populasi dari sampel yang ada (Gujarati dan Porter, 2009). Analisis regresi berbeda dengan korelasi. Korelasi mengukur tingkat asosiasi atau hubungan linear antara dua variabel, sedangkan analisis regresi fokus pada upaya untuk mengestimasi atau memprediksi suatu nilai berdasarkan variabel tetap maupun variabel independen. Data yang digunakan dalam analisis regresi dapat berupa data *time series* maupun data *cross-sectional*.

Regresi linear adalah salah satu bentuk analisis regresi. Istilah linear pada regresi linear mengacu pada nilai koefisien variabel yang berpangkat satu (*linear in parameters*), sehingga meskipun dalam model variabel independen berpangkat dua atau lebih, model tetap dapat dianggap sebagai regresi linear.

Analisis regresi linear merupakan salah satu bentuk statistika parametrik, sehingga terdapat beberapa asumsi klasik yang harus dipenuhi (Gujarati dan Porter, 2009):

1. Model regresi bersifat *linear in parameters* sesuai persamaan:

$$Y_t = \hat{\beta}_1 + \hat{\beta}_2 X_1 + \hat{u}_t \dots\dots\dots (2.20)$$

Y_t = variabel dependen

β_1 = koefisien *intercept*

$$\hat{\beta}_1 = \bar{Y} - \hat{\beta}_2 \bar{X} \dots\dots\dots (2.21)$$

β_2 = koefisien *slope*

$$\hat{\beta}_2 = \frac{n \sum X_t Y_t - \sum X_t \sum Y_t}{n \sum X_t^2 - (\sum X_t)^2} \dots\dots\dots (2.22)$$

X_1 = variabel dependen

u_t = residual/error

2. Variabel tetap dan variabel independen memiliki kovariansi 0 dengan galat/error
3. Rataan nilai residual u_t bernilai 0
4. Variansi residual u_t bersifat konstan (homokedastisitas)
5. Tidak terdapat autokorelasi pada residual
6. Jumlah observasi n lebih besar dibanding jumlah parameter yang diestimasi.
7. Harus terdapat variansi dalam data dan tidak terdapat *outlier*.

2.10. Bahasa Pemrograman R

R adalah bahasa pemrograman dan perangkat lunak untuk manipulasi data, perhitungan dan visualisasi grafis. Beberapa fasilitas yang disediakan meliputi (Venables et al., 2016):

- Fasilitas penyimpanan dan penanganan data.
- Sederetan operator untuk perhitungan, terutama perhitungan matriks.
- Koleksi alat bantu (*tools*) analisis data yang lengkap, koheren, dan terintegrasi.
- Fasilitas grafis untuk analisis data.
- Bahasa pemrograman yang simpel dan efektif.

Beberapa kelebihan dari R meliputi (Wickham, 2018):

- Bersifat *open source* sehingga dapat menggunakan dan mendistribusikan algoritma dan hasilnya secara bebas.
- Memiliki komunitas yang beragam dan aktif berkontribusi mengembangkan *tools* dan *packages* untuk membantu memudahkan aplikasi R di berbagai bidang.
- Bahasa pemrograman yang mendukung analisis data.
- Memiliki banyak *tools* untuk membantu permodelan statistik, pembelajaran mesin, visualisasi, impor dan manipulasi data.
- Mudah terhubung dengan bahasa pemrograman lain seperti C, C++, dan Fortran.

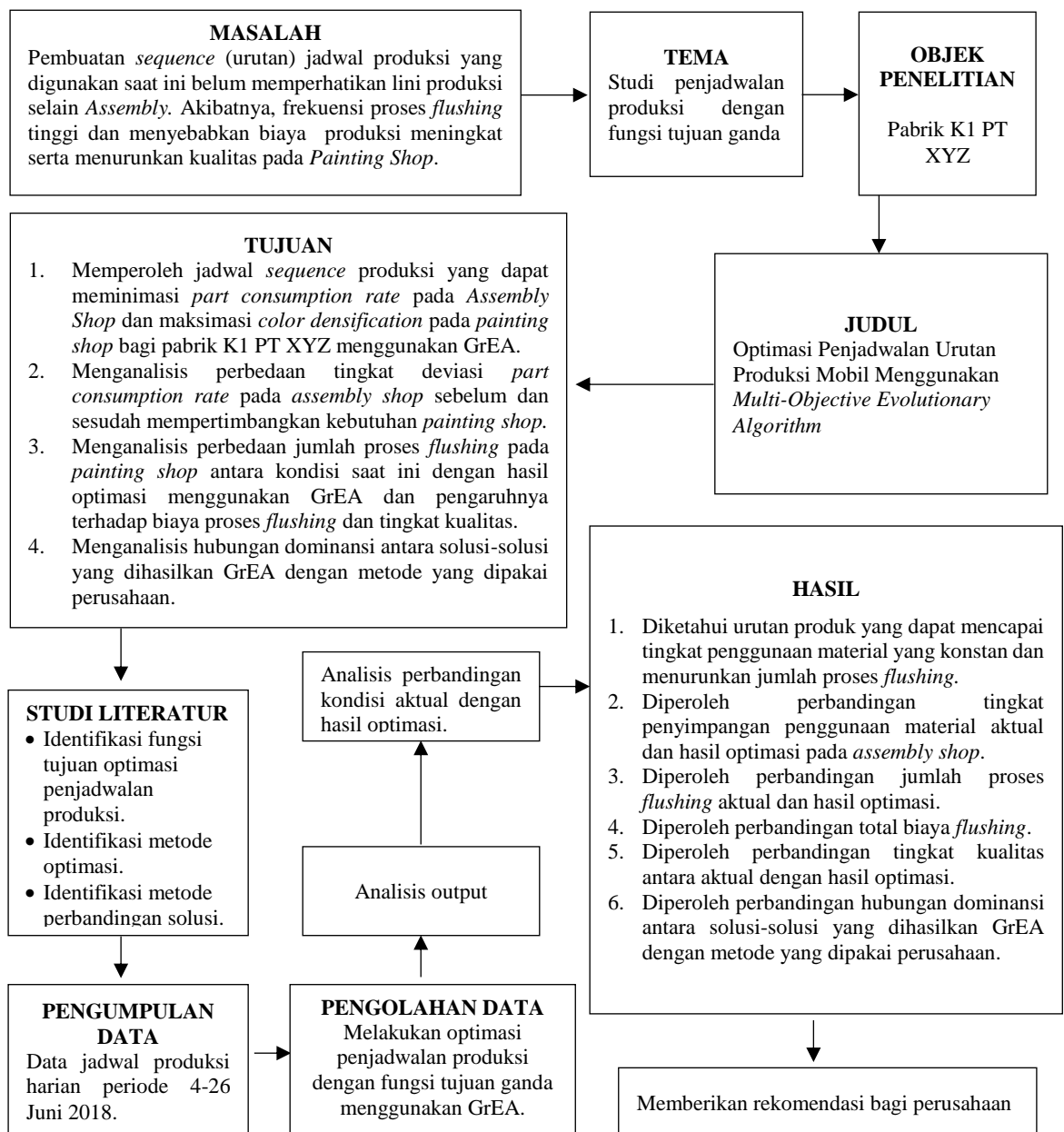
Dalam tampak muka (*interface*) dari RStudio, yang merupakan IDE (*Integrated Development Environment*) untuk bahasa pemrograman R, terdapat panel *console* yang berfungsi menampilkan algoritma yang dijalankan beserta outputnya. Pada panel *output* terdapat beberapa menu, seperti File, Plots, Packages, Help, dan Viewer. Menu File menampilkan file-file yang dapat diakses. Menu Plots menampilkan output grafik. Menu Packages menampilkan *packages* atau paket-paket *tools* yang tersedia. Menu Help berisi tampilan bantuan atau pedoman. Di atas panel *output* terdapat panel untuk menampilkan Environment, yang berisi daftar data, vektor, dan fungsi-fungsi yang ditulis oleh pengguna. Selain itu, terdapat menu History untuk menampilkan riwayat algoritma-algoritma yang telah dijalankan.

BAB III

METODE PENELITIAN

3.1. Kerangka Berpikir

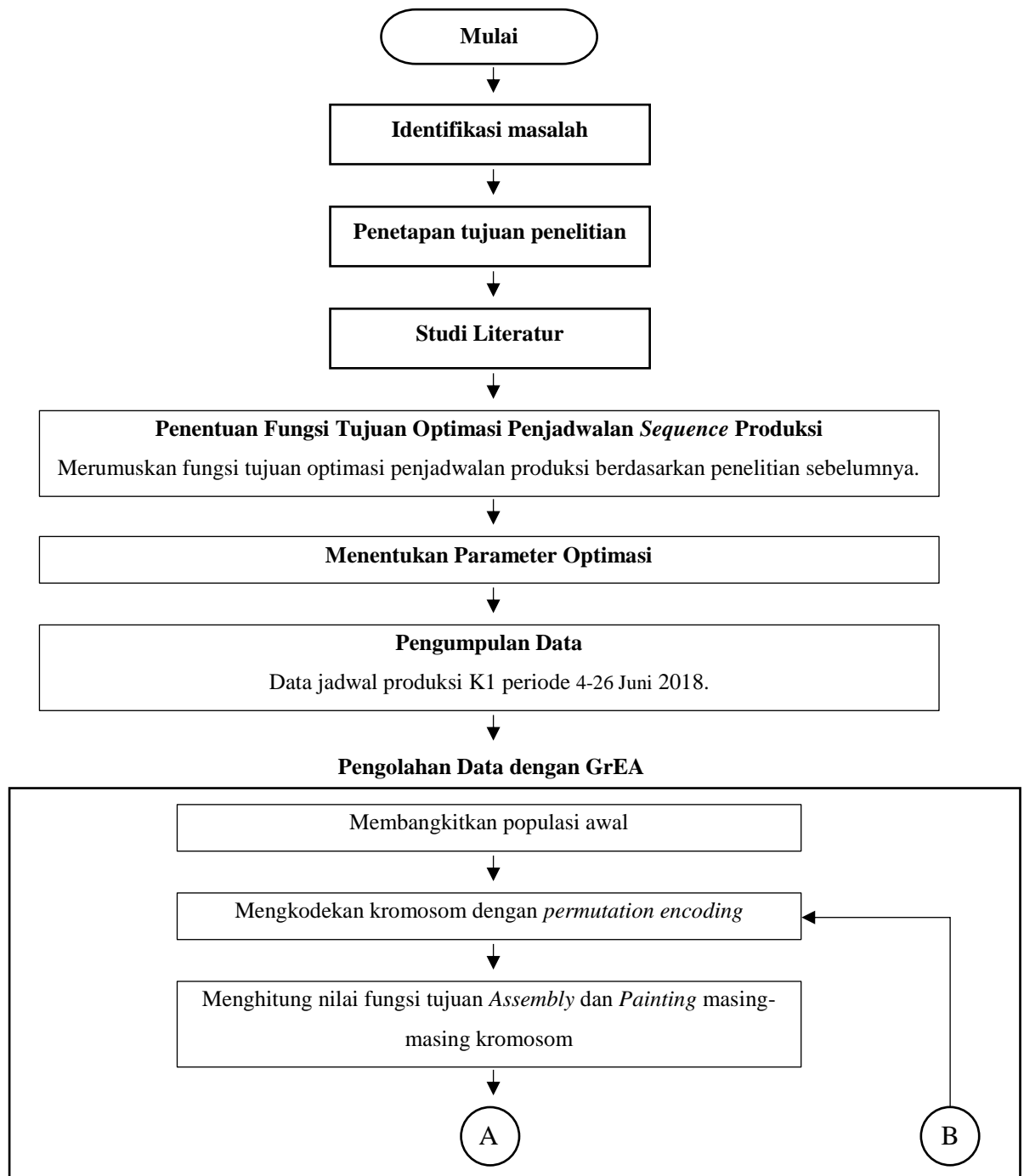
Kerangka pikir penelitian merupakan gambaran alur logika yang sistematis yang dapat memberikan bantuan dalam memahami isi dari penelitian. Kerangka pikir penelitian dapat dilihat pada gambar 3.1.



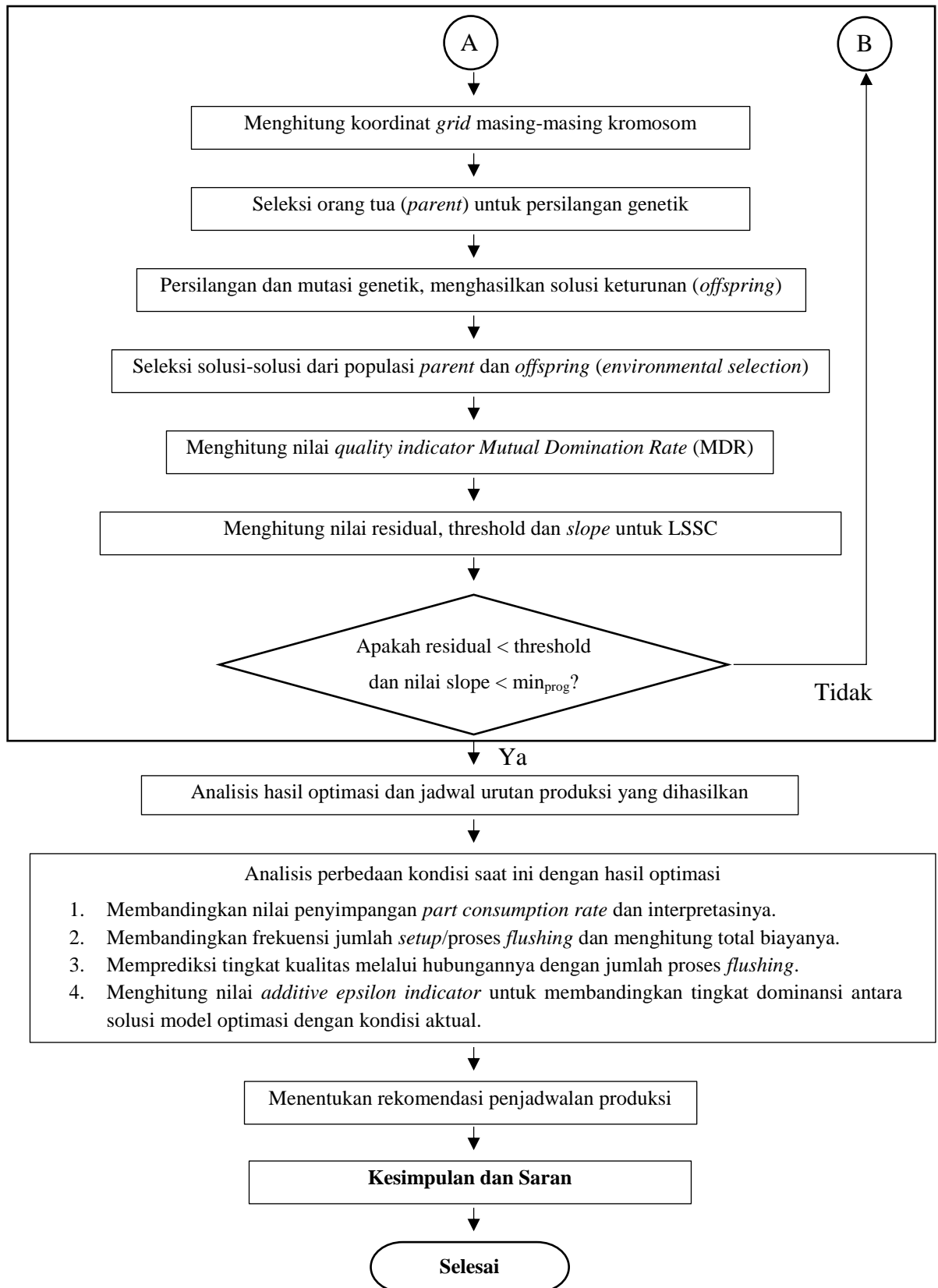
Gambar 3.1 Kerangka Pikir Penelitian Alur Penelitian

3.2. Alur Penelitian

Alur penelitian digunakan untuk mengetahui arah penyelesaian masalah dan menjadi pedoman penulis dalam melakukan penelitian. Alur penelitian ditunjukkan gambar 3.2.



Gambar 3.2 Alur Penelitian



Gambar 3.2 Lanjutan Alur Penelitian

3.3. Tempat dan Waktu Penelitian

Penelitian dilakukan di K1 PT XYZ. Penelitian dilakukan selama 6 bulan mulai dari Januari hingga Juni 2018.

3.4. Desain Penelitian

Desain penelitian yang digunakan mengacu kepada Cooper dan Schindler (2014). Penelitian ini termasuk dalam *formal study*, di mana penelitian bertujuan untuk menjawab pertanyaan penelitian terkait permasalahan penjadwalan produksi di K1 PT XYZ. Penelitian merupakan studi kasus (*case study*), yang berarti bahwa penelitian berupaya untuk menyelesaikan analisis secara mendalam untuk menjawab permasalahan penjadwalan produksi di K1 PT XYZ dan tidak berupaya untuk melakukan generalisasi hasil penelitian.

Dalam penelitian ini, variabel yang dapat diubah adalah urutan (*sequence*) dari produk yang dicapai melalui optimasi penjadwalan produksi dan dilakukan untuk memprediksi dan menemukan nilai optimal dari *part consumption rate* pada *Assembly Shop* dan frekuensi proses *flushing* pada *Painting Shop*. Penelitian dilakukan dengan mengumpulkan data historis perusahaan dalam satu waktu sekaligus (*cross-sectional*). Selain itu, dilakukan wawancara dengan kepala seksi produksi dan *engineering* dan pihak-pihak yang terkait serta mengamati proses produksi agar dapat memahami masalah yang timbul. Penelitian merupakan *ex post facto design* yang artinya peneliti tidak memiliki pengaruh terhadap variabel yang diamati. Berdasarkan lingkungan penelitian, penelitian ini merupakan penelitian *field setting* karena data diambil dalam kondisi aktual di lapangan. Berdasarkan tujuan penelitian, maka penelitian ini termasuk jenis penelitian *causal-predictive* karena berusaha memprediksi nilai deviasi *part consumption rate* dan frekuensi proses *flushing* berdasarkan data jadwal produksi.

3.5. Populasi dan Sampel

Populasi dalam penelitian ini adalah seluruh unit kendaraan yang diproduksi oleh K1 PT XYZ. Pengambilan sampel dilakukan dengan mengumpulkan data historis dari jadwal produksi periode 4-26 Juni 2018 yang didapatkan dari *Central Control Room* (CCR) yang bertanggung jawab menentukan jadwal urutan produksi.

3.6. Teknik Pengumpulan Data

Pengumpulan data dilakukan melalui dua metode:

1. Wawancara

Wawancara dilakukan dengan kepala seksi produksi dan kepala seksi *engineering* dari *painting shop* dan *assembly shop*. Wawancara bertujuan untuk menggali informasi terkait penjadwalan produksi dan kaitannya terhadap proses produksi. Selain itu wawancara dilakukan dengan pihak CCR untuk mengetahui metode penjadwalan *sequence* produksi.

2. Observasi

Observasi langsung dilakukan ke K1 terutama di *painting shop* untuk mengamati proses *flushing* dan kaitannya dengan kualitas produksi. Selain itu observasi dilakukan untuk melihat proses pembuatan jadwal *sequence* produksi.

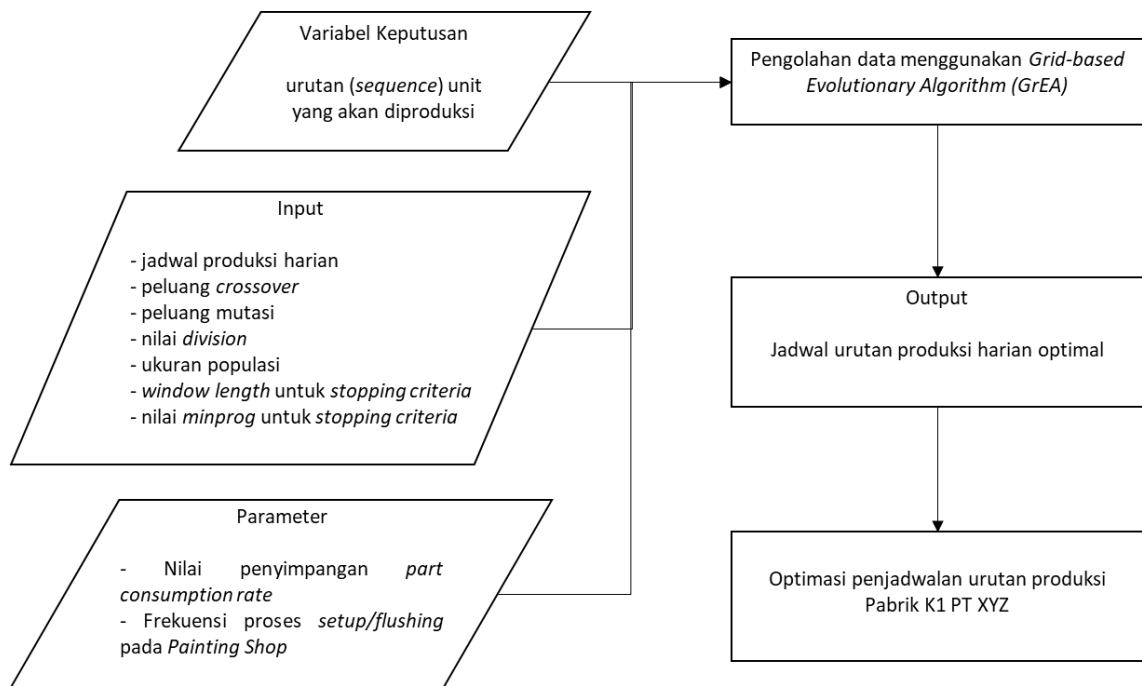
3. Data sekunder

Data sekunder yang digunakan berupa data jadwal produksi K1 selama 10 hari kerja pada periode 4-26 Juni 2018.

3.7. Pengolahan dan Analisis Data

3.7.1. Model Konseptual

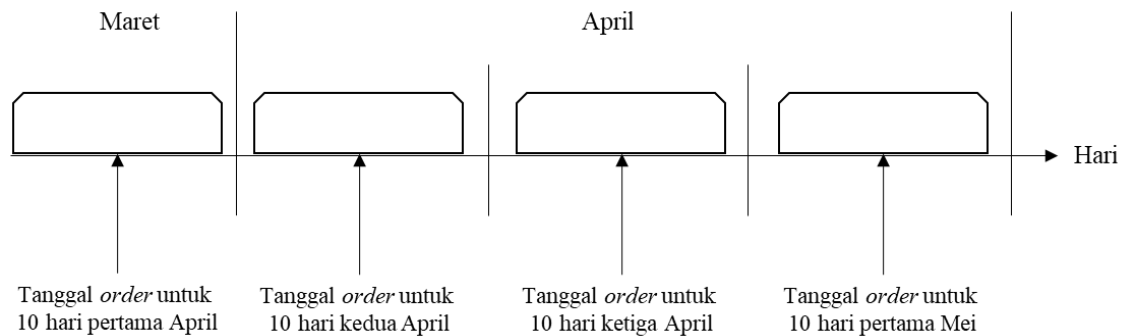
Model konseptual suatu penelitian adalah suatu model yang menunjukkan hubungan logis antara variabel yang telah diidentifikasi dengan tujuan menganalisis masalah penelitian. Gambar 3.3 menunjukkan model konseptual yang digunakan dalam penelitian ini.



Gambar 3.3 Model Konseptual

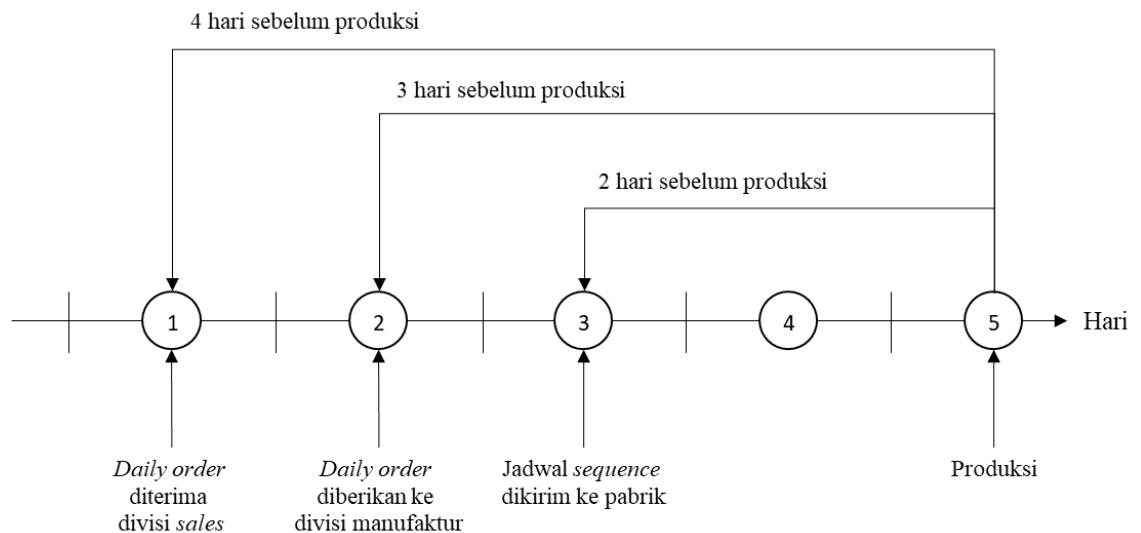
3.7.2. Karakteristik Sistem

Gambar 3.4 menunjukkan interval penerimaan pesanan dari konsumen oleh PT XYZ. Setiap *dealer* atau konsumen mengirim pesanan untuk 10 hari (*10-day order*) berupa rincian spesifikasi model termasuk warna produk yang diminta. Akumulasi jumlah produk selama 3 hari dari *10-day order* tidak boleh melebihi volume produksi yang ditentukan dalam *master production schedule*. *Master production schedule* awal ditentukan berdasarkan ramalan permintaan. *10-day order* dikirim 7 hingga 8 hari sebelum awal periode 10 hari dari pesanan tersebut. Kemudian jadwal jumlah produksi harian direncanakan sesuai *10-day order*. PT XYZ dapat mentolerir perubahan pesanan sebanyak 10% hingga 23% dari rencana produksi awal. Perubahan ini dapat dilakukan maksimal 4 hari produksi sebelum memasuki lini assembly. Selanjutnya divisi penjualan akan mengkategorikan pesanan *dealer* menjadi tipe-tipe yang berbeda baik dari segi model, tipe body, mesin, transmisi, warna, dan lain-lain. Informasi produksi dikirim ke divisi manufaktur 3 hari sebelum memulai produksi. Divisi manufaktur menyiapkan jadwal *sequence* produksi harian untuk *mixed-model assembly line*. Informasi *sequence* harian diberikan 2 hari sebelum produksi dilakukan.



Gambar 3.4 Sistem Pemesanan 10 Hari PT XYZ

Gambar 3.5 menunjukkan alur dari sistem penjadwalan produksi PT XYZ. Pembuatan jadwal *sequence* produksi dilakukan secara harian dan dikirim 2 hari sebelum unit diproduksi. Jenis produk yang akan dibuat disesuaikan dengan perintah dari divisi penjualan. Penelitian ini fokus pada pembuatan jadwal *sequence* produksi harian.



Gambar 3.5 Alur Pemesanan hingga Produksi

3.7.3. Asumsi Model

Asumsi yang digunakan sesuai dengan penelitian Miltenburg (1989), yakni dalam pembuatan *sequence* produksi, setiap tipe produk membutuhkan jumlah part dan material yang sama.

3.7.4. Batasan Model

Berikut merupakan karakteristik model matematis yang digunakan:

1. Produk terdiri dari berbagai tipe sesuai dengan rata-rata 69 *suffix*, 46 spesifikasi dan 8 warna produk.
2. *Time Horizon* untuk perencanaan *sequence* produksi adalah 1 hari.
3. Jumlah dan jenis produk yang akan diproduksi bersifat deterministik.
4. Penjadwalan hanya memperhatikan penggunaan part dan tidak memperhatikan waktu siklus masing-masing varian produk dalam lini *Assembly*.

3.7.5. Formulasi Model

Model matematis yang digunakan berupaya untuk meminimasi deviasi tingkat *part consumption rate* dan meminimasi jumlah proses *flushing* melalui pemadatan (*densification*) *sequence*. Fungsi tujuan yang ingin dicapai dijabarkan sebagai berikut:

1. *Part Consumption Rate* pada Assembly Shop

Part consumption rate merupakan tingkat variasi penggunaan *raw material* dan part dalam proses produksi *Assembly Shop*. Tujuan yang ingin dicapai yaitu meminimasi variasi tersebut sehingga proses produksi berlangsung dengan tingkat penggunaan part yang konstan sesuai dengan prinsip *heijunka* (*leveling production*). Tipe kendaraan adalah jenis model kendaraan yang akan diproduksi, sedangkan *sub-assembly* yang diprioritaskan oleh PT XYZ meliputi *suffix*, *grade*, warna dan *engine type*. Fungsi tujuan I_S diadopsi dari Miltenburg (1989) dan ditunjukkan pada persamaan 3.1.

$$\begin{aligned} \text{Min } I_S = & \sum_{p=1}^N \sum_{k=1}^K \left(X_{k,p} - P \frac{n_k}{N} \right)^2 + \sum_{p=1}^N \sum_{a=1}^A \left(X_{a,p} - XA_p \frac{n_a}{N} \right)^2 + \\ & \sum_{p=1}^N \sum_{b=1}^B \left(X_{b,p} - XB_p \frac{n_b}{N} \right)^2 + \sum_{p=1}^N \sum_{c=1}^C \left(X_{c,p} - XC_p \frac{n_c}{N} \right)^2 + \\ & \sum_{p=1}^N \sum_{d=1}^D \left(X_{d,p} - XD_p \frac{n_d}{N} \right)^2 \dots\dots\dots(3.1) \end{aligned}$$

I_S = part consumption rate

N = banyaknya jumlah kendaraan

K = banyaknya model kendaraan

n_k = banyaknya kendaraan model k k = 1, 2, ..., K

A	= banyaknya jenis <i>suffix</i> mesin kendaraan
n_a	= banyaknya kendaraan dengan <i>suffix</i> a $a = 1, 2, \dots, A$
B	= banyaknya jenis <i>grade</i> kendaraan
n_b	= banyaknya kendaraan dengan <i>suffix</i> b $b = 1, 2, \dots, B$
C	= banyaknya varian warna kendaraan
n_c	= banyaknya kendaraan dengan warna c $c = 1, 2, \dots, C$
D	= banyaknya jenis <i>engine</i> kendaraan
n_d	= banyaknya kendaraan dengan <i>engine</i> d $d = 1, 2, \dots, D$
P	= jumlah kendaraan dalam posisi 1 hingga p
p	= posisi kendaraan dalam <i>sequence</i> $p = 1, 2, \dots, N$
$X_{k,p}$	= banyaknya kendaraan model k antara posisi 1 hingga posisi p
$X_{a,p}$	= jumlah <i>suffix</i> mesin yang diproduksi antara posisi 1 hingga p
$X_{b,p}$	= jumlah <i>grade</i> yang diproduksi antara posisi 1 hingga p
$X_{c,p}$	= jumlah warna yang diproduksi antara posisi 1 hingga p
$X_{d,p}$	= jumlah jenis <i>engine</i> yang diproduksi antara posisi 1 hingga p

Untuk memverifikasi bahwa *sequence* atau urutan produksi yang dihasilkan memiliki penggunaan *part* yang konstan, unit-unit dalam *sequence* dibagi menjadi 16 bagian secara merata, tiap bagian dianggap sebagai 1 jam produksi dengan total 16 jam produksi (2 *shift*). Distribusi penggunaan *part* dapat dilihat dari nilai standar deviasinya (*standard deviation*). Ketika nilai *standard deviation*-nya rendah, dapat disimpulkan bahwa *part* atau *sub-assembly* terdistribusi secara konstan di setiap jam produksi (Monden, 2012). Part yang terdistribusi dengan konstan membuat *supplier* yang mengikuti kanban dari perusahaan memiliki laju produksi yang konstan pula, serta perusahaan dapat meminimalkan/meniadakan *inventory* yang bersifat antisipatif (*safety stock*) apabila terjadi fluktuasi. Pada penelitian ini, unit dibagi menjadi 14 bagian, sesuai waktu kerja efektif produksi perusahaan yakni 7.5 jam pada *shift* pertama dan 6.5 jam pada *shift* kedua. Tabel 3.1 menunjukkan contoh verifikasi *sequence* penggunaan *part front axle* pada lini *assembly*.

Tabel 3.1 Verifikasi *Part Consumption Rate*

Tipe Part	Jam																rataaan	std dev.
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16		
A1	9	7	7	9	8	7	8	8	8	8	7	8	9	7	7	8	7.813	0.750
A2	6	5	7	6	5	6	7	5	7	6	5	7	6	6	5	6	5.938	0.772
A3	5	6	5	5	6	6	4	6	4	6	6	5	4	6	5	6	5.313	0.793
A4	3	3	3	2	3	3	3	3	3	2	3	3	3	2	3	3	2.813	0.403
A5	2	2	2	2	3	2	2	2	2	3	2	1	3	2	2	2	2.125	0.500
A6	1	1	1	1	1	2	1	1	2	0	2	1	1	1	1	1	1.125	0.500

(Monden, 2012)

2. Minimasi Proses *Flushing* pada *Painting Shop*

Proses *flushing* adalah proses pembersihan *nozzle spray* dan dilakukan untuk menghilangkan sisa-sisa cat dari warna sebelumnya yang masih menempel di *nozzle*. Proses *flushing* memerlukan biaya yang tinggi serta tidak ramah lingkungan karena menghasilkan limbah. Untuk dapat meminimasi proses *flushing*, dapat dilakukan dengan dua cara. Yang pertama yakni dengan mengupayakan minimasi *changeover cost* dengan fungsi tujuan seperti yang digunakan McMullen (2000). Namun, cara ini menyebabkan pemerataan produksi sulit dicapai karena kurang adaptif jika terjadi gangguan seperti kualitas yang rendah dari proses *Painting Shop* dan *Welding Shop* (Bernier dan Frein, 2004). Cara yang kedua yaitu dengan pemadatan (*densification*), di mana unit dengan warna yang sama berada dalam jarak sedekat mungkin satu sama lain, meskipun tidak berurutan secara langsung. Untuk dapat meminimasi proses *flushing*, maka penjadwalan harus dapat memaksimalkan (*densification*) pemadatan pada *Painting Shop*. Fungsi pemadatan *Painting Shop* diadopsi dari Joly dan Frein (2008) dan ditunjukkan pada persamaan 3.2.

$$I_{Di,j} = \sum_{\substack{j'=1 \\ j' \neq j}}^{n_c} \frac{1}{|P_{c,j'} - P_{c,j}|} \dots\dots\dots (3.2)$$

$I_{Dc,j}$ = Fungsi kedekatan (*proximity*) antara kendaraan dengan tipe warna sama

C = banyaknya varian warna kendaraan

n_c = banyaknya kendaraan dengan warna c $c = 1, 2, \dots, C$

B_c = bobot warna c $c = 1, 2, \dots, C$

$P_{c,j}$ = posisi kendaraan ke- j dengan warna c

$$p = 1, 2, \dots, N; \quad j = 1, 2, \dots, n_c; \quad c = 1, 2, \dots, C$$

Diperoleh $I_{Dc,j \max}$

$$I_{Dc,j \max} = \sum_{\substack{j'=1 \\ j' \neq j}}^{n_c} \frac{1}{|j'-j|} \dots \dots \dots (3.3)$$

Fungsi pemadatan pada *Painting Shop*:

$$\text{Max } I_D = \frac{\sum_{c=1}^C B_c n_c \sum_{j=1}^{n_c} \frac{I_{Dc,j}}{I_{Dc,j \max}}}{\sum_{c=1}^C B_c n_c} \dots \dots \dots (3.4)$$

3.7.6. Algoritma Pencarian yang Digunakan

3.7.6.1. Algoritma Dasar

Fungsi tujuan pada persamaan (3.1) dan (3.4) dioptimasi menggunakan *Grid-Based Evolutionary Algorithm* (GrEA). Algoritma ditunjukkan pada tabel 3.2.

Tabel 3.2 Algoritma GrEA

Require: P (population), N (population size) 1: $P \leftarrow \text{Initialize}(P)$ 2: while <i>termination criterion not fulfilled</i> do 3: $\text{Grid-setting}(P)$ 4: $\text{Fitness_assignment}(P)$ 5: $P' \leftarrow \text{Mating_selection}(P)$ 6: $P'' \leftarrow \text{Variation}(P')$ 7: $P \leftarrow \text{Environmental_selection}(P \cup P'')$ 8: end while 9: return P

(Yang et al., 2013)

Algoritma diawali dengan membangkitkan populasi awal $P(t)$, $t=0$, secara acak sejumlah N , dengan t menunjukkan iterasi ke- t yang dilakukan. Kriteria penghentian yang digunakan adalah jumlah iterasi t . Selama kriteria penghentian algoritma belum terpenuhi, maka proses iterasi akan terus berlangsung. Dalam proses iterasi, nilai solusi untuk setiap tujuan dihitung terlebih dahulu, kemudian diubah menjadi titik-titik koordinat dalam proses *Grid_setting*. Pada *Fitness_assignment*, dihitung nilai *fitness* yang akan digunakan sebagai kriteria pemilihan solusi, yang meliputi: *Grid Ranking* (GR), *Grid Crowding Distance* (GCD), dan *Grid Coordinat Point Distance* (GCPD). Kriteria GR dan GCPD digunakan untuk menjamin kesesuaian solusi dengan nilai optimal aktual

(*convergence*), sedangkan GCD digunakan untuk menjamin keragaman solusi dalam populasi (*diversity*). Setelah itu, dilakukan pemilihan *parent* dan melakukan persilangan. Hasil persilangan berupa keturunan (*offspring*) akan dibandingkan dengan solusi dalam populasi melalui seleksi lingkungan (*environmental selection*). Jika *offspring* memiliki nilai yang lebih baik, maka akan menggantikan anggota populasi yang lebih jelek. Iterasi diulang kembali mulai dari *Gridd_setting* hingga mencapai jumlah iterasi yang ditetapkan.

3.7.6.2. Konsep Dominansi

Konsep dominansi digunakan dalam *multi-objective optimization problem* untuk menentukan solusi yang bersifat *pareto-optimal*, yakni solusi yang memiliki nilai lebih unggul dibanding solusi lainnya. Dalam GrEA, digunakan dua konsep dominansi, yakni *pareto-dominance* dan *grid-dominance*. Dalam permasalahan minimasi berlaku:

$$\begin{aligned} &\text{Minimize } F(x) = (F_1(x), F_2(x), \dots, F_M(x)) \\ &\text{Subject to } x \in \Omega \dots\dots\dots (3.5) \end{aligned}$$

x menunjukkan vektor solusi dari semesta solusi Ω , dan $F_i (i = 1, 2, \dots, M)$ menunjukkan tujuan ke- i yang ingin diminimasi. Terdapat $x, y \in \Omega$, x mendominasi secara pareto y , dinotasikan $x \prec y$ apabila:

$$\begin{aligned} &\forall i \in (1, 2, \dots, M): F_i(x) \leq F_i(y) \wedge \\ &\exists j \in (1, 2, \dots, M): F_j(x) < F_j(y) \dots\dots\dots (3.6) \end{aligned}$$

sedangkan konsep *grid-dominance* mirip dengan *pareto-dominance*, hanya saja yang dibandingkan adalah nilai koordinat $G_i(x)$ ($i = 1, 2, \dots, M$) dan $G_i(y)$. Terdapat $x, y \in \Omega$, x mendominasi secara *grid* terhadap y , dinotasikan $x \prec_{grid} y$:

$$\begin{aligned} &\forall i \in (1, 2, \dots, M): G_i(x) \leq G_i(y) \wedge \\ &\exists j \in (1, 2, \dots, M): G_j(x) < G_j(y) \dots\dots\dots (3.7) \end{aligned}$$

3.7.6.3. Menghitung Nilai Kecocokan (*Fitness Value*)

Fitness value digunakan untuk memilih solusi yang optimal. Dalam GrEA, *fitness value* yang digunakan yakni *Grid Ranking* (GR), *Grid Crowding Distance* (GCD), dan *Grid Coordinate Point Distance* (GCPD). Untuk menghitung nilai-nilai tersebut, nilai

solusi dari setiap tujuan diubah menjadi titik-titik koordinat. Langkah pertama yaitu dimulai dengan mencari batas bawah dan batas atas dari setiap fungsi tujuan melalui persamaan:

$$lb_k = \min_k(P) - \frac{\max_k(P) - \min_k(P)}{2 \text{ div}} \dots\dots\dots (3.8)$$

$$ub_k = \max_k(P) + \frac{\max_k(P) - \min_k(P)}{2 \text{ div}} \dots\dots\dots (3.9)$$

ub_k = batas atas fungsi tujuan k , $k = 1, 2, \dots, M$

lb_k = batas bawah fungsi tujuan k , $k = 1, 2, \dots, M$

\min_k = nilai minimal solusi k dalam populasi P

\max_k = nilai maksimal solusi k dalam populasi P

div = nilai divisi atau maksimal nilai koordinat yang dihasilkan

Setelah mengetahui batas bawah dan batas atas setiap tujuan, maka dilanjutkan dengan menghitung nilai *distance* atau jarak antar koordinat untuk setiap fungsi tujuan dan mencari koordinat setiap solusi melalui persamaan:

$$d_k = \frac{(ub_k - lb_k)}{\text{div}} \dots\dots\dots (3.10)$$

$$G_k(x) = \left\lfloor \frac{F_k(x) - lb_k}{d_k} \right\rfloor \dots\dots\dots (3.11)$$

d_k = jarak antar koordinat dalam fungsi tujuan k , $k = 1, 2, \dots, M$

$G_k(x)$ = koordinat untuk fungsi tujuan k dari solusi x

$F_k(x)$ = nilai solusi x untuk fungsi tujuan k

Langkah selanjutnya adalah menghitung nilai *Grid Difference* (GD), yakni total nilai absolut antara selisih koordinat solusi x dengan solusi lainnya untuk setiap fungsi tujuan. Setelah itu, dihitung nilai GR, GCD, dan GCPD.

$$GD(x, y) = \sum_{k=1}^M |G_k(x) - G_k(y)| \dots\dots\dots (3.12)$$

$$GR(x) = \sum_{k=1}^M G_k(x) \dots\dots\dots (3.13)$$

$$GCD(x) = \sum_{y \in N(x)} (M - GD(x, y)) \dots\dots\dots (3.14)$$

$$GCPD(x) = \sqrt{\sum_{k=1}^M \left(\frac{F_k(x) - (lb_k(x) + G_k(x) \cdot d_k)}{d_k} \right)^2} \dots\dots\dots (3.15)$$

3.7.6.4. Seleksi Orang Tua (*Parent*)

Tabel 3.3 menjelaskan algoritma yang digunakan dalam pemilihan *parent* yang akan digunakan dalam tahap persilangan genetik (*crossover*).

Tabel 3.3 Seleksi *Parent*

<p>Require: individuals p, q randomly chosen from the population</p> <pre> 1: if $p < q$ or $p <_{grid} q$ then 2: return p 3: else if $q < p$ or $q <_{grid} p$ then 4: return q 5: else if $GCD(p) < GCD(q)$ then 6: return p 7: else if $GCD(q) < GCD(p)$ then 8: return q 9: else if $random(0,1) < 0.5$ then 10: return p 11: else 12: return q 13: end if </pre>

(Yang et al., 2013)

Proses seleksi *parent* dilakukan untuk mendapatkan calon orang tua terbaik yang dapat menghasilkan keturunan (*offspring*) dengan sifat unggul. Metode seleksi yang digunakan adalah *binary tournament selection*, di mana dua calon *parent* dipilih secara acak dari populasi kemudian diseleksi menggunakan *pareto dominance* atau *grid dominance*. Apabila keduanya bersifat tidak saling mendominasi (*non-dominated*), maka seleksi dilanjutkan dengan memilih *parent* yang memiliki nilai GCD yang paling rendah. Jika keduanya memiliki nilai GCD yang sama, maka *parent* dipilih dari salah satu solusi tersebut secara acak. Seleksi *parent* dilakukan sebanyak dua kali untuk mendapatkan sepasang *parent* yang bersifat *non-dominated* berdasarkan hasil masing-masing seleksi.

3.7.6.5. Persilangan dan Mutasi

Metode persilangan dan mutasi yang digunakan mengacu pada penelitian Nearchou (2004), yang menunjukkan bahwa kombinasi antara dua jenis *crossover*, yakni *one-point crossover* dan *three-point crossover* dengan metode mutasi *shift mutation* memiliki performa yang lebih baik dibandingkan kombinasi lainnya untuk permasalahan penjadwalan *flowshop*. *Crossover* yang digunakan dipilih secara acak pada setiap iterasi. *One-point crossover* dilakukan dengan menentukan satu titik *crossover* secara acak,

kemudian gen keturunan dari titik pertama hingga titik *crossover* akan diisi dengan gen dari *parent* pertama dan sisanya diisi dari *parent* kedua. Sedangkan *three-point crossover* diawali dengan memilih tiga titik *crossover* secara acak. *Parent* pertama akan mengisi gen pada posisi pertama hingga ke titik *crossover* pertama dan mengisi gen pada posisi di antara dua titik *crossover* berikutnya. Gen pada posisi selain yang diisi oleh *parent* pertama akan diisi dengan gen dari *parent* kedua.

3.7.6.6. Seleksi Alam (*Environmental Selection*)

Tabel 3.4 menjelaskan algoritma yang digunakan dalam pemilihan solusi yang unggul dan akan digunakan sebagai populasi untuk iterasi selanjutnya. Algoritma diawali dengan membuat populasi arsip Q yang masih kosong. Kemudian, dengan menggunakan pendekatan *fast nondominated sorting*, setiap kromosom dalam populasi P dipisahkan menjadi beberapa *front*. Satu *front* hanya berisi kromosom-kromosom yang saling tidak mendominasi (*non-dominated*) dan kromosom lainnya akan masuk *front* berikutnya.

Tabel 3.4 Seleksi Alam (*Environmental Selection*)

```

Require:  $N$  (archive size)
1: Generate an empty set  $Q$  for archive
2:  $(F_1, F_2, \dots, F_i, \dots) \leftarrow \text{Pareto\_nondominated\_sort}(P)$ 
3: Find the critical front  $F_i$  where  $(0 \leq N - |F_1 \cup F_2 \cup \dots \cup F_{i-1}| < F_i)$ 
4:  $Q \leftarrow F_1 \cup F_2 \cup \dots \cup F_{i-1}$ 
5: if  $|Q| = N$  then
6:   return  $Q$ 
7: end if
8: Grid_setting ( $F_i$ )
9: Initialization ( $F_i$ )
10: while  $|Q| < N$  do
11:    $q \leftarrow \text{Findout\_best}(F_i)$ 
12:    $Q \leftarrow Q \cup \{q\}$ 
13:    $F_i \leftarrow F_i \setminus \{1\}$ 
14:   GCD_calculation ( $F_i, q$ )
15:   GR_adjustment ( $F_i, q$ )
16: end while
17: return  $Q$ 

```

(Yang et al., 2013)

Setelah melakukan pensortiran, dicari *critical front* (F_i), yaitu *front* di mana akumulasi jumlah anggota *front* dari F_1 hingga F_i kurang dari atau sama dengan jumlah populasi (N). Populasi Q diisi dengan anggota F_1 hingga F_{i-1} . Apabila jumlah anggota Q sama dengan N , maka Q menjadi populasi P untuk iterasi selanjutnya. Apabila jumlah anggota Q kurang dari N , maka jumlah yang belum terpenuhi akan diambil dari F_i , diawali

dengan menghitung ulang nilai $G_k(x)$ hanya untuk populasi dalam F_i . Langkah berikutnya adalah menghitung nilai GR dan GCPD dan nilai GCD diisi dengan nilai 0 sesuai algoritma pada tabel 3.5.

Tabel 3.5 Algoritma Initialization

```

1: for all  $p \in P$  do
2:    $GR\_assignment(p)$ 
3:    $GCPD\_assignment(p)$ 
4:    $GCD(p) \leftarrow 0$ 
5: end for

```

(Yang et al., 2013)

Kemudian, selama jumlah Q kurang dari N , dicari solusi terbaik (q) sesuai tabel 3.6, yang diawali dengan memilih solusi pertama dalam F_i sebagai solusi terbaik. Langkah berikutnya yaitu membandingkan nilai q dengan solusi kedua hingga solusi terakhir dalam F_i secara bertahap. Apabila nilai GR dari q lebih kecil dari solusi lain, maka q tidak diubah. Namun apabila nilai GR solusi lain lebih kecil, maka q akan diganti dengan solusi tersebut. Apabila nilai GR sama, berikutnya yang dibandingkan adalah nilai GCD dengan aturan yang sama dengan membandingkan GR. Apabila nilai GCD sama, maka yang dibandingkan adalah nilai GCPD. Jika nilai GCPD sama, maka q dipilih secara acak dari dua solusi yang dibandingkan.

Tabel 3.6 Algoritma Findout best (P)

```

Require:  $q$  (best solution in  $P$ ),  $p_i$  (the  $i$ th solution in  $P$ )
1:  $q \leftarrow p_1$ 
2: for  $i = 2$  to  $|P|$  do
3:   if  $GR(p_i) < GR(q)$  then
4:      $q \leftarrow p_i$ 
5:   else if  $GR(p_i) = GR(q)$  then
6:     if  $GCD(p_i) < GCD(q)$  then
7:        $q \leftarrow p_i$ 
8:     else if  $GCD(p_i) = GCD(q)$  then
9:       if  $GCPD(p_i) < GCPD(q)$  then
10:         $q \leftarrow p_i$ 
11:       end if
12:     end if
13:   end if
14: end for
15: return  $q$ 

```

(Yang et al., 2013)

Setelah ditemukan solusi terbaik q , maka solusi q akan dimasukkan ke dalam populasi Q dan dikeluarkan dari F_i . Langkah selanjutnya adalah menghitung nilai GCD dari setiap solusi yang tersisa sesuai tabel 3.7. Nilai GCD dihitung dengan mencari nilai

GD dari setiap solusi dalam F_i dengan solusi yang ada di dalam populasi Q . Kemudian, dilakukan pengaturan nilai GR dari setiap solusi F_i sesuai dengan tabel 3.8. Untuk setiap solusi yang memiliki nilai GD 0 terhadap solusi q nilainya akan diberi penalti sebesar $M+2$. Untuk setiap solusi yang didominasi oleh q diberi penalti M . Untuk solusi yang saling tidak mendominasi dengan q diberi penalti paling sedikit $M-GD$.

Tabel 3.7 Algoritma Perhitungan GCD

```
Require:  $P$  (candidate set),  $q$  (picked individual),  $N(q) := \{p \in P \mid GD(p,q) < M\}$ 
1: for all  $p \in N(q)$  do
2:    $GCD(p) \leftarrow GCD(p) + (M - GD(p,q))$ 
5: end for
```

(Yang et al., 2013)

Tabel 3.8 Algoritma Pengaturan Nilai GR (*GR Adjustment*)

```
Require:  $P$  (candidate set),  $q$  (picked individual),  $M$  (number of objectives),  $PD(p)$  (maximum
punishment degree of  $p$ ),  $E(q) := \{p \in P \mid GD(p,q) = 0\}$ ,  $N(q) := \{p \in P \mid GD(p,q) < M\}$ ,  $G(q) := \{p \in P \mid q \prec_{grid} p\}$ ,  $NG(q) := \{p \in P \mid q \not\prec_{grid} p\}$ 
1: for all  $p \in E(q)$  do
2:    $GR(p) \leftarrow GR(p) + M + 2$ 
3: end for
4: for all  $p \in G(q)$  do
5:    $GR(p) \leftarrow GR(p) + M$ 
6: end for
7: for all  $p \in NG(q) \wedge p \notin E(q)$  do
8:    $PD(p) \leftarrow 0$ 
9: end for
10: for all  $p \in N(q) \cap p \in NG(q) \wedge p \notin E(q)$  do
11:   if  $PD(p) < M - GD(p,q)$  then
12:      $PD(p) \leftarrow M - GD(p,q)$ 
13:     for all  $r \in G(p) \wedge r \notin G(q) \cup \wedge E(q)$  do
14:       if  $PD(r) < PD(p)$  then
15:          $PD(r) \leftarrow PD(p)$ 
16:       end if
17:     end for
18:   end if
19: end for
20: for all  $p \in NG(q) \wedge p \notin E(q)$  do
21:    $GR(p) \leftarrow GR(p) + PD(p)$ 
22: end for
```

(Yang et al., 2013)

Quality indicator mengukur kualitas antara estimasi solusi generasi tertentu dengan solusi optimal. Pengukuran indikator melibatkan 3 faktor (Guerrero et al., 2010):

1. Jarak antar elemen dalam *pareto front* saat ini dengan *pareto front* optimal (menentukan seberapa dekat solusi dengan global optima).
2. Jarak antara elemen *pareto front* optimal dengan elemen yang paling terdekat dalam solusi saat ini (menentukan seberapa baik solusi saat ini melingkupi *pareto front* optimal).
3. Distribusi elemen dalam solusi saat ini dan *pareto front* optimal.

Indikator yang digunakan dalam penelitian ini adalah *Mutual Domination Rate* (MDR) sesuai persamaan 2.13. Indikator ini memiliki keunggulan yakni lebih sederhana dan membutuhkan waktu komputasi yang lebih rendah dibandingkan indikator lainnya. Indikator ini digunakan untuk membandingkan antara solusi yang didominasi antara generasi saat ini dengan generasi sebelumnya (Guerrero et al., 2010).

Nilai $I_{MDR} = 1$ menunjukkan bahwa populasi pada iterasi ke- t lebih baik dibandingkan iterasi sebelumnya. Nilai $I_{MDR} = 0$ menunjukkan bahwa tidak terdapat perubahan yang signifikan antar iterasi, dan jika nilai $I_{MDR} = -1$ berarti iterasi t memiliki solusi yang lebih rendah dibandingkan iterasi sebelumnya.

Proses pencarian solusi akan terus berlangsung selama kriteria penghentian (*stopping criterion*) belum terpenuhi. *Stopping criterion* dapat menetapkan maksimal iterasi/generasi yang dilakukan atau dapat menggunakan algoritma lain. Dalam penelitian ini, *stopping criterion* yang digunakan adalah *Least Square Stopping Criterion* (LSSC). LSSC menggunakan konsep regresi linear sederhana untuk mengidentifikasi kapan nilai *quality indicator* menjadi stagnan.

LSSC menggunakan konsep *ordinary least square* atau regresi linear sederhana untuk memprediksi nilai indikator dengan persamaan (Guerrero et al., 2010):

$$y = a + bx \dots\dots\dots (3.17)$$

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sum xi^2 & \sum xi \\ \sum xi & w_l \end{bmatrix} \begin{bmatrix} \sum xi * yi \\ \sum yi \end{bmatrix} \dots\dots\dots (3.18)$$

Variabel x menunjukkan nomer generasi/iterasi, sedangkan variabel y menunjukkan nilai *quality indicator* yang digunakan. Notasi w_l menunjukkan *window length* atau berapa jumlah generasi yang akan diamati. Setelah mendapatkan nilai variabel a dan b , langkah selanjutnya adalah menghitung nilai residual:

$$res = \frac{\sum_i (y_i - (a + b x_i))^2}{w_l} \sim \frac{x^2}{w_l} \dots\dots\dots (3.19)$$

Nilai residual berdistribusi *Chi-Square* dengan derajat kebebasan w_l-n . didapatkan nilai rata-rata dan variansi distribusi sebagai berikut (Guerrero, 2010):

$$\mu = 1 - \frac{2}{w_l} \dots\dots\dots (3.20)$$

$$\sigma^2 = \frac{2}{w_l} - \frac{4}{(w_l)^2} \dots\dots\dots (3.21)$$

Digunakan nilai *threshold* sesuai teorema Chebysev untuk menentukan apakah 99% nilai residual berasal dari distribusi uniform melalui persamaan (Guerrero, 2010):

$$thres = \mu + 3\sigma \dots\dots\dots (3.22)$$

Ketika nilai residual berada di bawah *threshold*, hal ini menunjukkan jika tren nilai indikator mulai bersifat *uniform* atau stagnan seiring bertambahnya iterasi. Selain menggunakan nilai residual, nilai *slope* (variabel b) digunakan untuk menunjukkan bahwa pengaruh perubahan tiap iterasi sudah tidak memadai. Berdasarkan rumusan tersebut, maka didapatkan *stopping criteria* yang digunakan dalam model optimasi, sesuai dengan persamaan (Guerrero, 2010):

$$stop = residual < threshold \wedge b < min_{prog} \dots\dots\dots (3.23)$$

Kelebihan dari LSSC dibandingkan *stopping criteria* lainnya adalah implementasi yang mudah, hanya membutuhkan dua parameter (*window length* dan min_{prog}), dan bersifat *robust* (konfigurasi yang sama dapat digunakan untuk berbagai jenis permasalahan yang berbeda).

3.7.7. Parameter Heuristik

Simulasi dilakukan menggunakan komputer dengan prosesor Intel Core i5-4210U 2,7 GHz dan RAM 4.00 GB. Ukuran populasi sebesar 50 kromosom di setiap generasi. Nilai *division* yang digunakan adalah 9 sesuai dengan rekomendasi Yang et al. (2013) untuk permasalahan optimasi yang belum diketahui. Peluang *crossover* yang digunakan sebesar 1.0. Peluang mutasi yang digunakan bersifat adaptif, artinya nilai peluang berubah setiap pergantian generasi. Penggunaan peluang mutasi yang adaptif bertujuan untuk mendapatkan populasi yang bervariasi pada iterasi-iterasi awal dan populasi yang lebih stabil pada iterasi selanjutnya. Sesuai penelitian Nearchou (2004), peluang mutasi pada generasi pertama sebesar 0.8 dan untuk generasi berikutnya dikalikan 0.99, sesuai persamaan:

$$\text{peluang mutasi } (t + 1) = \text{peluang mutasi } (t) \times 0.99 \dots\dots\dots (3.24)$$

Parameter *stopping criteria* yang digunakan yakni nilai *window length* (w_l) sebesar 60 generasi dan nilai $\text{min}_{\text{prog}} = 0.00002$ sesuai studi Guerrero et al. (2010).

3.7.8. Analisis Hasil Pengolahan Data

Pada tahap analisis, hasil yang didapatkan dari pengolahan data akan dijabarkan secara mendalam. Tahap analisis akan menjawab tujuan penelitian yang akan dicapai, meliputi urutan (*sequence*) produk yang dihasilkan berdasarkan optimasi dua fungsi tujuan, perbedaan tingkat deviasi *part consumption rate* pada *Assembly Shop* sebelum dan sesudah mempertimbangkan kebutuhan *Painting Shop* serta perbedaan jumlah proses *flushing* pada *Painting Shop* sebelum dan sesudah dilakukan optimasi penjadwalan. Statistika deskriptif akan digunakan untuk membandingkan kondisi sebelum dan sesudah dilakukan optimasi.

Tingkat *part consumption rate* menunjukkan distribusi penggunaan *part* dalam rentang waktu tiap jam sesuai tabel 3.1. Perbandingan antara *part consumption rate* kondisi aktual dengan hasil optimasi dilakukan dengan melakukan uji perbandingan rata-rata dengan uji *t* berpasangan atau uji jumlah rank *wilcoxon* (Montgomery dan Runger, 2014). Kedua uji tersebut digunakan untuk membandingkan 2 nilai rata-rata, dengan hipotesis awal adalah kedua sampel (sebelum dan sesudah dilakukan optimasi) tidak memiliki perbedaan rata-rata yang signifikan. Uji *t* berpasangan (*paired t-test*) menggunakan asumsi bahwa selisih antara kedua data berdistribusi normal, sehingga perlu diuji normalitas terlebih dahulu menggunakan uji Shapiro-Wilk. Apabila selisih tidak berdistribusi normal, maka hasil uji *t* dapat menjadi bias, sehingga perlu diganti dengan menggunakan uji rank Wilcoxon yang tidak membutuhkan asumsi normalitas. Apabila antara kondisi aktual dengan hasil optimasi memiliki hasil uji $p\text{-value} > 0.05$, maka dapat disimpulkan bahwa tingkat persebaran penggunaan *part* sama atau tidak berbeda secara signifikan. Sedangkan apabila terdapat perbedaan persebaran penggunaan *part* yang signifikan, maka perlu dilihat jenis *sub-assembly* mana yang memiliki persebaran penggunaan *part* yang lebih sedikit. *Sub-assembly* mesin dan model memiliki jumlah varian kurang dari 5, sehingga tidak cukup untuk melakukan uji perbandingan

rataan. Maka, analisis hanya dilakukan dengan membandingkan langsung nilai dari masing-masing part dalam spesifikasi tersebut.

Jumlah proses *flushing* dihitung dengan melihat berapa banyak terjadi perubahan warna dari unit satu ke unit selanjutnya. Total biaya proses *flushing* dihitung dengan mengalikan antara jumlah proses *flushing* dengan biaya proses *flushing* untuk satu kali proses. Tingkat kualitas hasil optimasi diprediksi dengan menggunakan regresi linear sederhana, dengan variabel jumlah proses *flushing* sebagai variabel independen dan tingkat kualitas sebagai variabel dependen. Regresi linear digunakan karena sederhana dan mudah diinterpretasi nilai hubungan antar variabelnya. Performa model prediksi diukur menggunakan nilai R^2 , nilai *root mean squared error* (RMSE), dan nilai *mean absolute error* (MAE). Nilai kualitas antara aktual dengan hasil optimasi kemudian dibandingkan.

Quality Indicator untuk mengetahui apakah solusi hasil optimasi lebih baik dibandingkan kondisi saat ini. Salah satu indikator yang sering digunakan adalah *hypervolume*, yang berfungsi mengukur nilai kedekatan (konvergensi) terhadap titik optimal sebenarnya/*true pareto front* sekaligus sebagai ukuran persebaran solusi yang dihasilkan (Florea et al., 20016). *Hypervolume* mengukur volume atau luas antara titik-titik solusi pada *pareto front* dengan titik referensi (*reference point/nadir point*). Namun, *hypervolume* dan *unary quality indicator* lainnya tidak dapat secara jelas mengidentifikasi perbandingan antara dua solusi yang berbeda. *Binary quality indicator* dapat mengidentifikasi seberapa kuat tingkat dominansi suatu solusi, apakah bersifat dominan mutlak, dominan, dominan lemah, setara atau bahkan tidak dapat dibandingkan (Zitzler et al., 2003). Terdapat beberapa jenis *binary quality indicator* yang dapat digunakan, di antaranya *binary hypervolume*, *epsilon indicator*, *additive epsilon indicator*, dan *coverage indicator*.

Binary hypervolume mengukur nilai *hypervolume* dari area yang didominasi oleh solusi pertama namun tidak didominasi oleh area kedua. Dikarenakan nilai *true pareto front* pada permasalahan ini tidak diketahui, nilai *reference point* yang digunakan adalah nilai terburuk dari masing-masing fungsi tujuan, atau dengan kata lain nilai yang didominasi oleh semua solusi lainnya (Wright et al., 2014). Kelemahan dari indikator ini adalah memerlukan informasi tambahan berupa nilai *reference point* sebagai titik acuan.

Indikator *coverage* mengukur berapa proporsi solusi B yang didominasi oleh solusi A. Kelemahan dari indikator ini adalah tidak dapat diketahui seberapa baik dominansi oleh suatu solusi serta nilainya sulit diinterpretasi apabila dua set solusi tidak dapat dibandingkan (Zitzler et al., 2003). Dalam permasalahan minimasi, *epsilon indicator* mengukur berapa nilai ϵ minimal yang dapat membuat solusi B yang dikalikan dengan faktor ϵ tidak mendominasi solusi A berdasarkan aturan perkalian. Sedangkan *additive epsilon indicator* mengukur berapa nilai ϵ minimal yang dapat membuat solusi B yang ditambah dengan faktor ϵ tidak mendominasi solusi A. Baik *epsilon indicator* maupun *additive indicator* dapat mengatasi kelemahan indikator *coverage* dengan memberikan informasi suatu solusi mendominasi solusi lain sebesar faktor ϵ serta membutuhkan waktu komputasi yang singkat. Kelemahan dari *epsilon indicator* adalah kriteria harus disesuaikan apabila permasalahan berupa maksimasi. *Additive epsilon indicator* lebih fleksibel dalam tanpa perlu merubah kriteria dominansi. Maka, dalam penelitian ini digunakan *additive epsilon indicator* untuk menentukan hubungan dominansi antara solusi optimasi dengan kondisi saat ini.

Dalam simulasi model, performa sistem dapat bervariasi. Oleh karena itu, simulasi perlu diulang beberapa kali untuk mendapatkan hasil yang dapat diandalkan. Pengulangan algoritma minimal dilakukan sebanyak 10 kali agar mendapatkan hasil yang dapat dipercaya secara statistik (Daellenbach dan McNikle, 2005). Algoritma akan dijalankan untuk penjadwalan 10 hari yang berbeda berdasarkan data jadwal produksi periode 4-26 Juni 2018, dan dibandingkan dengan jadwal aktual yang telah digunakan oleh perusahaan untuk mendapatkan hasil yang meyakinkan berkaitan dengan dominansi hasil optimasi terhadap kondisi aktual perusahaan saat ini.

3.7.9. Rekomendasi Penjadwalan Produksi

Langkah berikutnya adalah memberikan rekomendasi jadwal produksi K1 PT XYZ. Rekomendasi dengan menunjukkan sekumpulan solusi optimal yang dihasilkan beserta nilai deviasi *part consumption rate* pada *assembly shop* dan nilai fungsi pemadatan *paint shop* dari setiap solusi yang dihasilkan.

3.7.10. Kesimpulan dan Saran

Berisi kesimpulan dari tujuan penelitian yang telah dibuat sebelumnya. Pada tahap ini juga akan dijabarkan saran-saran yang berkaitan dengan penelitian, sehingga hasil penelitian dapat menjadi pertimbangan pihak-pihak terkait sesuai dengan kepentingan masing-masing.

BAB IV

PENGUMPULAN DAN PENGOLAHAN DATA

4.1. Pengumpulan Data

Data didapatkan dari departemen *central control room* (CCR) yang bertugas melakukan pembuatan jadwal urutan produksi pada pabrik K1. Data berupa jadwal produksi yang berisi informasi spesifikasi produk dan jumlah yang diproduksi. Spesifikasi produk terdiri atas beberapa jenis *sub-assembly*. Gambar 4.1 menunjukkan contoh uraian spesifikasi produk.

Tabel 4.1 Uraian *Sub-Assembly* Produk

Spesifikasi	1A	GUN155	R	S	D	T	M	H	D	218
Kategori	Suffix	Kode Dasar Model	Jenis Setir	Nama Model	Lokasi produksi	Jenis Transmisi	<i>Grade</i>	Jenis Mesin	Tujuan	Warna

Sub-assembly yang diperhatikan dalam pembuatan jadwal urutan produksi adalah *suffix*, nama model, *grade*, jenis mesin, dan warna. *Suffix* merupakan ID khusus pada setiap varian mobil dan untuk mobil dengan tipe *suffix* berbeda dapat memiliki fitur tambahan atau aksesoris yang berbeda, meskipun spesifikasi lainnya sama persis. Nama model merujuk pada model dasar mobil yang dibuat dan PT XYZ memiliki 2 model, yakni model S dan model M. *Grade* menunjukkan tingkat kualitas mobil dan terbagi menjadi 4 *grade* pada setiap model, sehingga total terdapat 8 jenis *grade*. Semakin tinggi *grade*, semakin lengkap fitur yang dimiliki mobil serta semakin tinggi kualitas material yang digunakan. Jenis mesin dikategorikan berdasarkan bahan bakar yang digunakan dan daya yang dikeluarkan mesin. Warna mobil merujuk pada warna eksterior atau warna luar mobil. Saat ini rata-rata terdapat 7 hingga 9 warna yang diproduksi setiap hari. Tabel 4.2 menunjukkan jumlah dan banyaknya varian unit yang diproduksi.

Tabel 4.2 Data Produksi

Tanggal	Jumlah Unit	Suffix	Model	Grade	Jenis Mesin	Warna
4-Jun-19	537	70	2	8	4	9
5-Jun-19	555	74	2	8	4	8
6-Jun-19	567	69	2	8	4	8
7-Jun-19	567	69	2	8	4	8
8-Jun-19	584	70	2	8	4	7
20-Jun-19	455	64	2	8	4	7
21-Jun-19	588	76	2	8	4	8
22-Jun-19	622	70	2	8	4	7
25-Jun-19	522	67	2	8	4	8
26-Jun-19	697	65	2	8	4	7

4.2. Pengolahan Data

4.2.1. Verifikasi Model

Verifikasi model dilakukan untuk memastikan bahwa model yang dibuat sudah menggunakan data yang tepat dan logis secara matematis (Daellenbach dan MckNikle, 2005). Verifikasi dilakukan dengan memeriksa memeriksa hasil dari setiap persamaan yang ditulis di *software* RStudio. Setelah semua persamaan logis, maka model dijalankan untuk melihat apakah masih terdapat kesalahan/*error* yang menyebabkan algoritma berhenti atau memberikan hasil yang tidak diinginkan.

4.2.2. Implementasi Model

Model digunakan untuk mendapatkan jadwal urutan produksi yang dapat meminimasi penyimpangan *part consumption rate* sekaligus meminimasi jumlah proses *flushing* dengan memaksimalkan nilai densifikasi pada *paint shop*, sesuai persamaan 3.1 dan persamaan 3.4. Data yang digunakan adalah data produksi yang bersifat deterministik. Solusi-solusi yang bersifat *pareto optimal* dicari dengan menggunakan algoritma *Grid-based Evolutionary Algorithm* (GrEA) dengan nilai parameter yang sudah disebutkan pada sub bab 3.7.7. Algoritma ditulis menggunakan bahasa pemrograman R dan dijalankan melalui *software* RStudio.

4.2.3. Output Model

4.2.3.1. Jadwal Urutan Produksi

Jadwal urutan produksi berisi *sequence* atau urutan unit-unit yang akan diproduksi dilengkapi dengan spesifikasi produk dalam jangka waktu satu hari. Data jadwal urutan produksi aktual dan alternatif jadwal urutan produksi hasil pengolahan dengan algoritma GrEA dapat dilihat pada lampiran.

4.2.3.2. Rekapitulasi Fungsi Tujuan Optimal

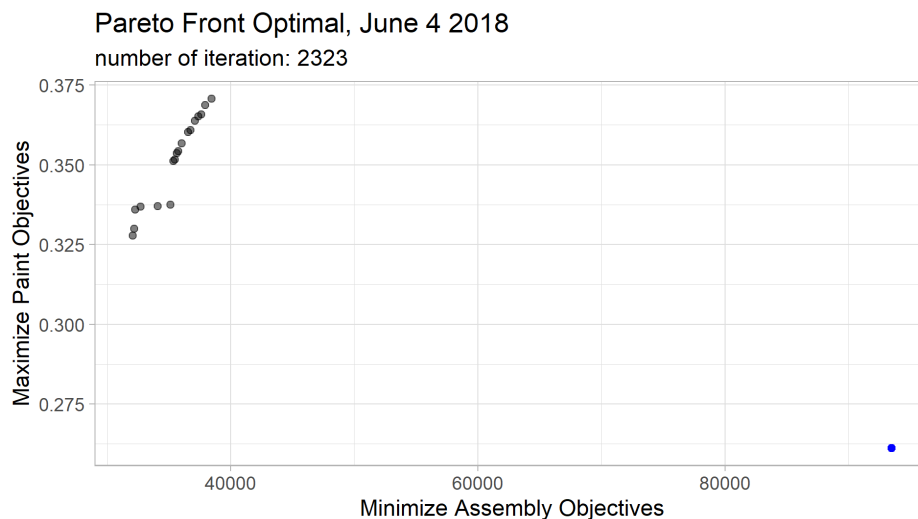
Output model GrEA adalah solusi-solusi dengan fungsi tujuan yang bersifat *pareto optimal*, yakni antara solusi alternatif satu dengan yang lain sama-sama baik dan tidak saling mendominasi. Tabel 4.3 menunjukkan rekapitulasi nilai fungsi tujuan minimasi penyimpangan *part consumption rate* dan tabel 4.4 menunjukkan maksimasi nilai densifikasi (*color densification*) pada *paint shop* untuk meminimasi proses setup yang didapatkan dari *running* algoritma GrEA. Nilai *mean* menunjukkan rata-rata dari nilai solusi alternatif yang dihasilkan, sedangkan *std dev.* menunjukkan nilai standar deviasinya.

Tabel 4.3 Rekapitulasi Minimasi Penyimpangan *Part Consumption Rate*

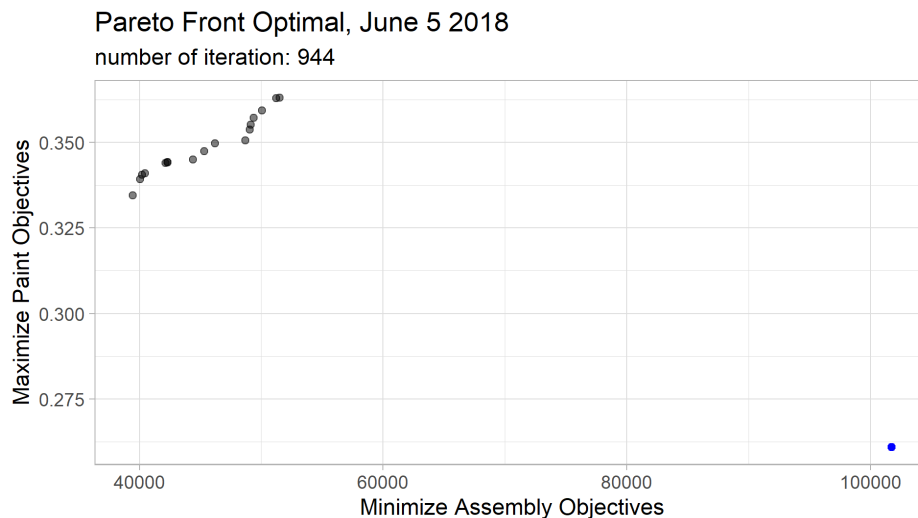
Tanggal	Hasil Aktual	Hasil Optimasi					Iterasi	Waktu (jam)
		Jumlah Alternatif	Min	Max	Mean	Std dev.		
4-Jun-18	93493.020	18	32068.97	38466.16	35479.854	1994.633	2323	106.716
5-Jun-18	101749.900	17	39447.20	51521.86	45411.416	4161.035	944	38.122
6-Jun-18	101469.200	17	40995.67	50307.99	45012.259	3356.483	1937	78.223
7-Jun-18	149250.700	18	36149.29	38202.60	37131.572	689.261	1937	84.679
8-Jun-18	153231.600	21	38283.70	53878.68	45929.949	5361.124	699	28.228
20-Jun-18	74150.350	16	24922.65	29746.72	27360.315	1779.870	2399	96.880
21-Jun-18	162519.900	14	49229.44	65725.53	57844.243	6729.944	935	37.758
22-Jun-18	114757.700	14	49796.94	55370.15	52087.267	1803.199	3579	164.415
25-Jun-18	148341.200	14	34798.82	39351.80	37129.394	1248.984	2416	97.566
26-Jun-18	79153.260	14	51127.41	58894.08	54639.500	2769.338	1300	74.858
Rataan	120513.756				43802.577		1793.88	77.858

Tabel 4.4 Rekapitulasi Maksimasi Denfisikasi Warna

Tanggal	Hasil Aktual	Hasil Optimasi				Iterasi	Waktu (jam)
		Jumlah Alternatif	Min	Max	Mean	Std dev.	
4-Jun-18	0.261	18	0.327	0.370	0.352	0.014	2323
5-Jun-18	0.261	17	0.334	0.363	0.349	0.008	944
6-Jun-18	0.266	17	0.365	0.388	0.375	0.007	1937
7-Jun-18	0.287	18	0.358	0.370	0.366	0.004	1937
8-Jun-18	0.287	21	0.303	0.356	0.327	0.016	699
20-Jun-18	0.276	16	0.320	0.366	0.343	0.015	2399
21-Jun-18	0.276	14	0.334	0.359	0.345	0.008	935
22-Jun-18	0.293	14	0.418	0.430	0.425	0.004	3579
25-Jun-18	0.282	14	0.341	0.385	0.372	0.013	2416
26-Jun-18	0.304	14	0.353	0.373	0.363	0.007	1300
Rataan	0.281				0.363		1793.88
							77.858

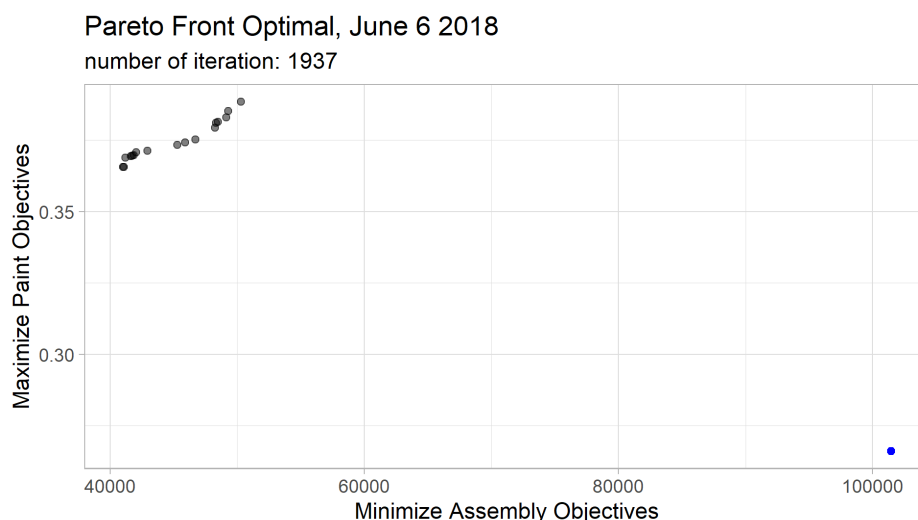
**Gambar 4.1 Perbandingan Fungsi Tujuan 4 Juni 2018**

Gambar 4.1 menunjukkan diagram tebar (*scatter plot*) nilai fungsi tujuan dari solusi alternatif GrEA (ditunjukkan dengan warna hitam) dan solusi aktual (ditunjukkan dengan warna biru) untuk jadwal produksi tanggal 4 Juni 2018. Fungsi tujuan dari solusi aktual untuk minimasi penyimpangan *part consumption rate* dan maksimasi densifikasi warna berturut-turut adalah 93493.02 dan 0.261. Optimasi menggunakan GrEA berhenti setelah 2323 iterasi dengan waktu komputasi 106.72 jam dan dihasilkan 18 alternatif solusi berbeda. Untuk fungsi tujuan minimasi penyimpangan *part consumption rate*, nilai yang dihasilkan GrEA memiliki rata-rata 35479.85, sedangkan fungsi tujuan densifikasi warna memiliki rata-rata 0.352.



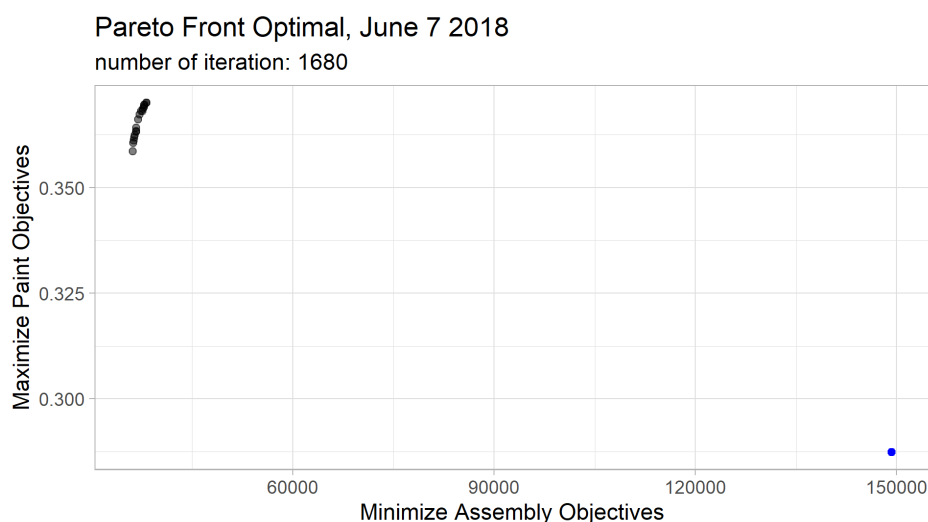
Gambar 4.2 Perbandingan Fungsi Tujuan 5 Juni 2018

Gambar 4.2 menunjukkan diagram tebar (*scatter plot*) nilai fungsi tujuan dari solusi alternatif GrEA (ditunjukkan dengan warna hitam) dan solusi aktual (ditunjukkan dengan warna biru) untuk jadwal produksi tanggal 5 Juni 2018. Fungsi tujuan dari solusi aktual untuk minimasi penyimpangan *part consumption rate* dan maksimasi densifikasi warna berturut-turut adalah 101749.9 dan 0.261. Optimasi menggunakan GrEA berhenti setelah 944 iterasi dengan waktu komputasi 38.12 jam dan dihasilkan 17 alternatif solusi berbeda. Untuk fungsi tujuan minimasi penyimpangan *part consumption rate*, nilai yang dihasilkan GrEA memiliki rata-rata 45411.42, sedangkan fungsi tujuan densifikasi warna memiliki rata-rata 0.349.



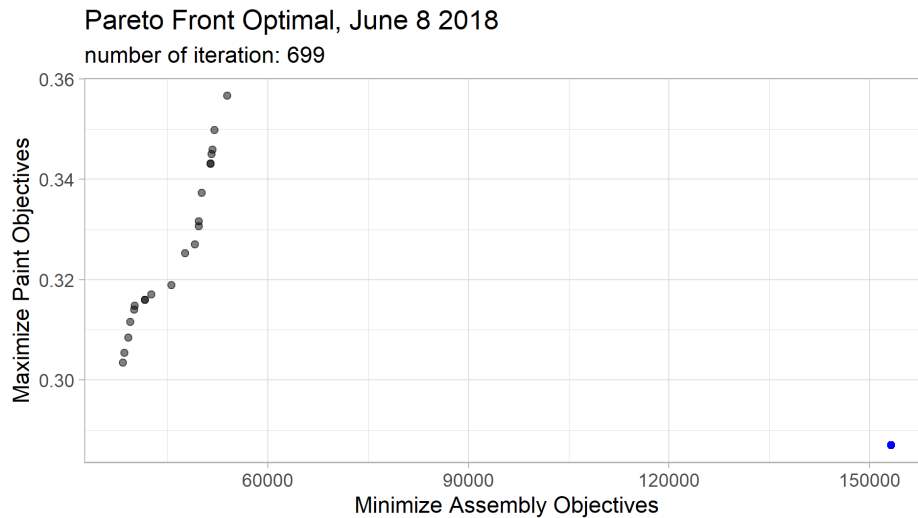
Gambar 4.3 Perbandingan Fungsi Tujuan 6 Juni 2018

Gambar 4.3 menunjukkan diagram tebar (*scatter plot*) nilai fungsi tujuan dari solusi alternatif GrEA (ditunjukkan dengan warna hitam) dan solusi aktual (ditunjukkan dengan warna biru) untuk jadwal produksi tanggal 6 Juni 2018. Fungsi tujuan dari solusi aktual untuk minimasi penyimpangan *part consumption rate* dan maksimasi densifikasi warna berturut-turut adalah 101469.2 dan 0.266. Optimasi menggunakan GrEA berhenti setelah 1937 iterasi dengan waktu komputasi 78.22 jam dan dihasilkan 17 alternatif solusi berbeda. Untuk fungsi tujuan minimasi penyimpangan *part consumption rate*, nilai yang dihasilkan GrEA memiliki rata-rata 45012.26, sedangkan fungsi tujuan densifikasi warna memiliki rata-rata 0.375.



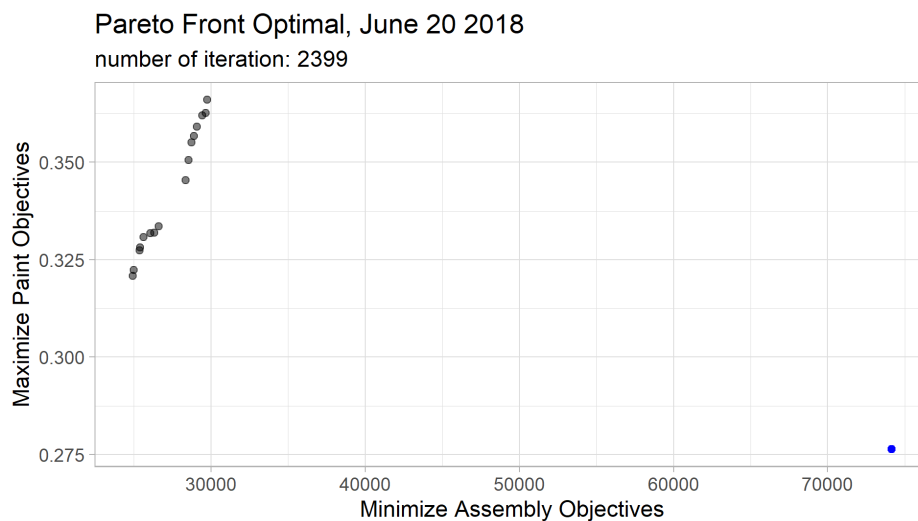
Gambar 4.4 Perbandingan Fungsi Tujuan 7 Juni 2018

Gambar 4.4 menunjukkan diagram tebar (*scatter plot*) nilai fungsi tujuan dari solusi alternatif GrEA (ditunjukkan dengan warna hitam) dan solusi aktual (ditunjukkan dengan warna biru) untuk jadwal produksi tanggal 7 Juni 2018. Fungsi tujuan dari solusi aktual untuk minimasi penyimpangan *part consumption rate* dan maksimasi densifikasi warna berturut-turut adalah 149250.7 dan 0.287. Optimasi menggunakan GrEA berhenti setelah 1937 iterasi dengan waktu komputasi 84.68 jam dan dihasilkan 18 alternatif solusi berbeda. Untuk fungsi tujuan minimasi penyimpangan *part consumption rate*, nilai yang dihasilkan GrEA memiliki rata-rata 37131.57, sedangkan fungsi tujuan densifikasi warna memiliki rata-rata 0.366.



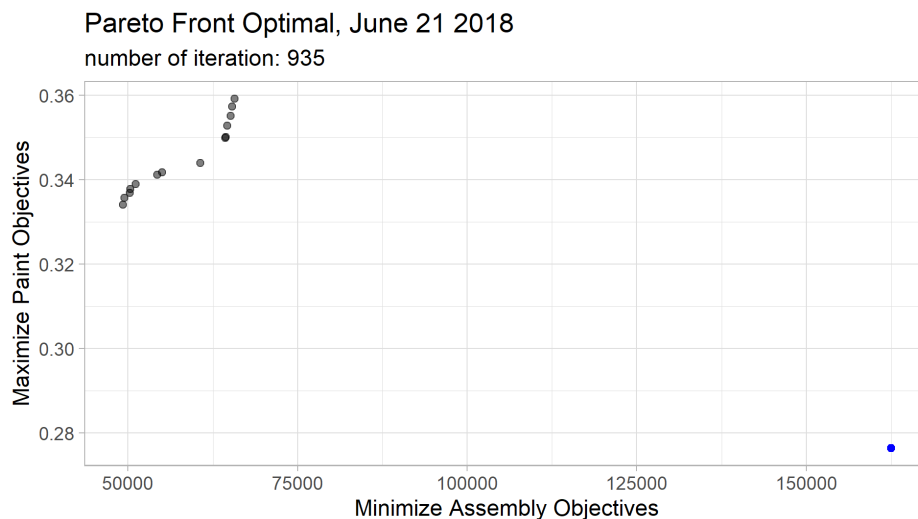
Gambar 4.5 Perbandingan Fungsi Tujuan 8 Juni 2018

Gambar 4.5 menunjukkan diagram tebar (*scatter plot*) nilai fungsi tujuan dari solusi alternatif GrEA (ditunjukkan dengan warna hitam) dan solusi aktual (ditunjukkan dengan warna biru) untuk jadwal produksi tanggal 8 Juni 2018. Fungsi tujuan dari solusi aktual untuk minimasi penyimpangan part consumption rate dan maksimasi densifikasi warna berturut-turut adalah 153231.6 dan 0.287. Optimasi menggunakan GrEA berhenti setelah 699 iterasi dengan waktu komputasi 28.23 jam dan dihasilkan 21 alternatif solusi berbeda. Untuk fungsi tujuan minimasi penyimpangan *part consumption rate*, nilai yang dihasilkan GrEA memiliki rata-rata 45929.49, sedangkan fungsi tujuan densifikasi warna memiliki rata-rata 0.327.



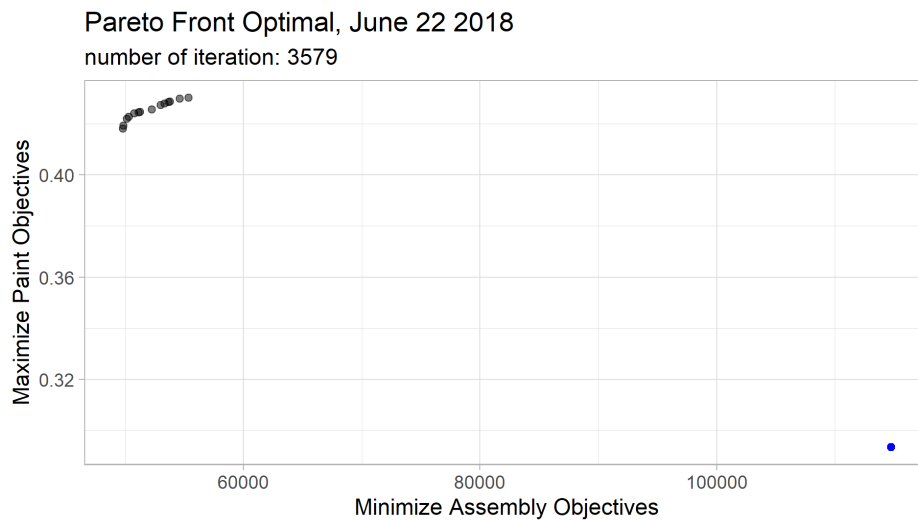
Gambar 4.6 Perbandingan Fungsi Tujuan 20 Juni 2018

Gambar 4.6 menunjukkan diagram tebar (*scatter plot*) nilai fungsi tujuan dari solusi alternatif GrEA (ditunjukkan dengan warna hitam) dan solusi aktual (ditunjukkan dengan warna biru) untuk jadwal produksi tanggal 20 Juni 2018. Fungsi tujuan dari solusi aktual untuk minimasi penyimpangan part consumption rate dan maksimasi densifikasi warna berturut-turut adalah 74150.35 dan 0.276. Optimasi menggunakan GrEA berhenti setelah 2399 iterasi dengan waktu komputasi 96.88 jam dan dihasilkan 16 alternatif solusi berbeda. Untuk fungsi tujuan minimasi penyimpangan *part consumption rate*, nilai yang dihasilkan GrEA memiliki rata-rata 27360.32, sedangkan fungsi tujuan densifikasi warna memiliki rata-rata 0.343.



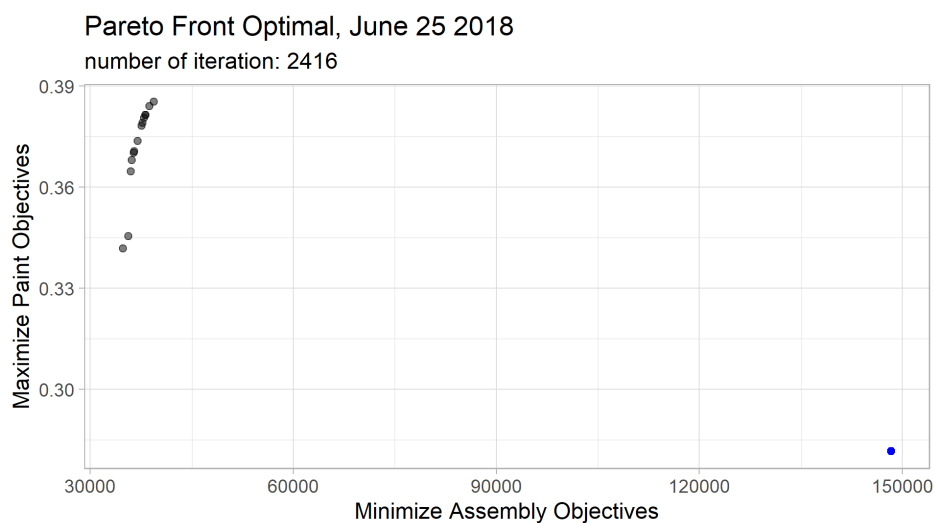
Gambar 4.7 Perbandingan Fungsi Tujuan 21 Juni 2018

Gambar 4.7 menunjukkan diagram tebar (*scatter plot*) nilai fungsi tujuan dari solusi alternatif GrEA (ditunjukkan dengan warna hitam) dan solusi aktual (ditunjukkan dengan warna biru) untuk jadwal produksi tanggal 21 Juni 2018. Fungsi tujuan dari solusi aktual untuk minimasi penyimpangan part consumption rate dan maksimasi densifikasi warna berturut-turut adalah 162519.9 dan 0.261. Optimasi menggunakan GrEA berhenti setelah 935 iterasi dengan waktu komputasi 37.76 jam dan dihasilkan 14 alternatif solusi berbeda. Untuk fungsi tujuan minimasi penyimpangan *part consumption rate*, nilai yang dihasilkan GrEA memiliki rata-rata 57844.24, sedangkan fungsi tujuan densifikasi warna memiliki rata-rata 0.345.



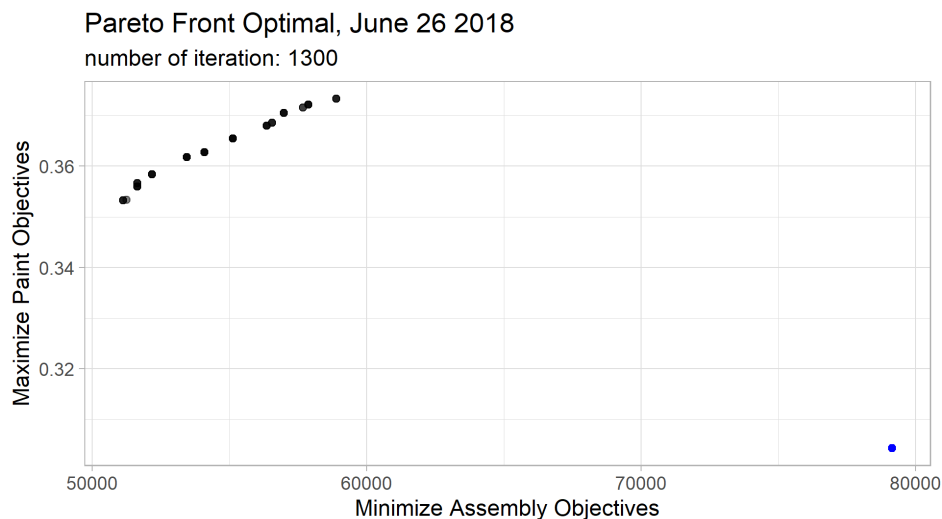
Gambar 4.8 Perbandingan Fungsi Tujuan 22 Juni 2018

Gambar 4.8 menunjukkan *scatter plot* nilai fungsi tujuan dari solusi alternatif GrEA (ditunjukkan dengan warna hitam) dan solusi aktual (ditunjukkan dengan warna biru) untuk jadwal produksi tanggal 22 Juni 2018. Fungsi tujuan dari solusi aktual untuk minimasi penyimpangan part consumption rate dan maksimasi densifikasi warna berturut-turut adalah 114757.7 dan 0.293. Optimasi menggunakan GrEA berhenti setelah 3579 iterasi dengan waktu komputasi 164.415 jam dan dihasilkan 14 alternatif solusi berbeda. Untuk fungsi tujuan minimasi penyimpangan *part consumption rate*, nilai yang dihasilkan GrEA memiliki rata-rata 52087.27, sedangkan fungsi tujuan densifikasi warna memiliki rata-rata 0.425.



Gambar 4.9 Perbandingan Fungsi Tujuan 25 Juni 2018

Gambar 4.9 menunjukkan diagram tebar (*scatter plot*) nilai fungsi tujuan dari solusi alternatif GrEA (ditunjukkan dengan warna hitam) dan solusi aktual (ditunjukkan dengan warna biru) untuk jadwal produksi tanggal 25 Juni 2018. Fungsi tujuan dari solusi aktual untuk minimasi penyimpangan part consumption rate dan maksimasi densifikasi warna berturut-turut adalah 148341.2 dan 0.282. Optimasi menggunakan GrEA berhenti setelah 2416 iterasi dengan waktu komputasi 97.57 jam dan dihasilkan 14 alternatif solusi berbeda. Untuk fungsi tujuan minimasi penyimpangan *part consumption rate*, nilai yang dihasilkan GrEA memiliki rata-rata 37129.39, sedangkan fungsi tujuan densifikasi warna memiliki rata-rata 0.372.

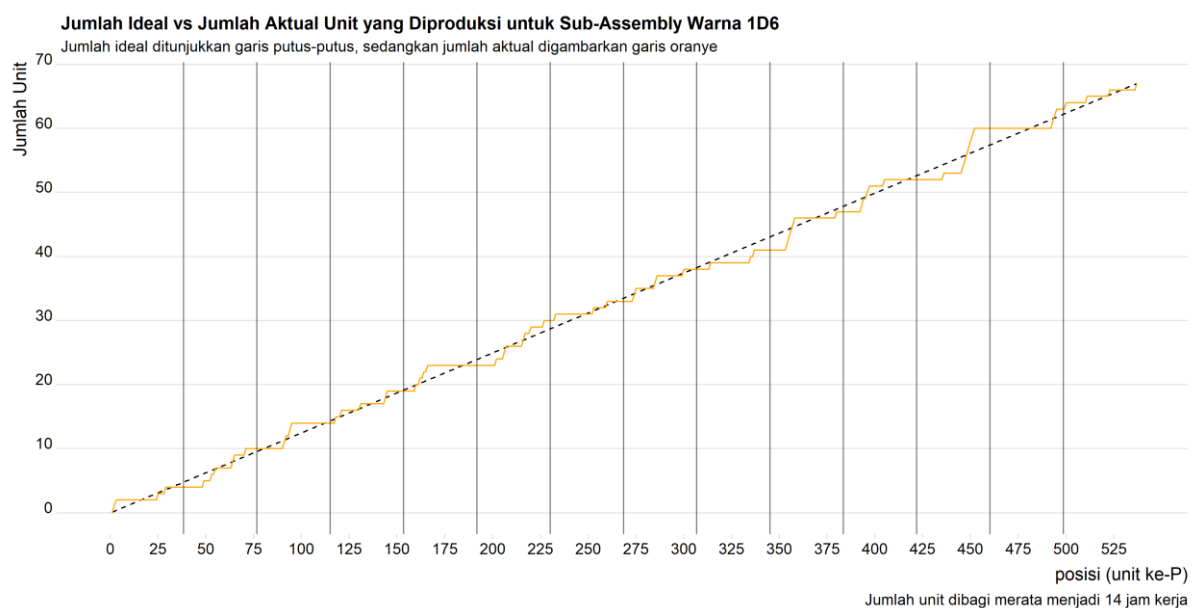


Gambar 4.10 Perbandingan Fungsi Tujuan 26 Juni 2018

Gambar 4.10 menunjukkan diagram tebar (*scatter plot*) nilai fungsi tujuan dari solusi alternatif GrEA (ditunjukkan dengan warna hitam) dan solusi aktual (ditunjukkan dengan warna biru) untuk jadwal produksi tanggal 26 Juni 2018. Fungsi tujuan dari solusi aktual untuk minimasi penyimpangan part consumption rate dan maksimasi densifikasi warna berturut-turut adalah 79153.26 dan 0.304. Optimasi menggunakan GrEA berhenti setelah 1300 iterasi dengan waktu komputasi 74.85 jam dan dihasilkan 14 alternatif solusi berbeda. Untuk fungsi tujuan minimasi penyimpangan *part consumption rate*, nilai yang dihasilkan GrEA memiliki rata-rata 54639.5, sedangkan fungsi tujuan densifikasi warna memiliki rata-rata 0.363.

4.2.3.3. Perbandingan Persebaran *Part Consumption Rate*

Setelah mendapatkan jadwal urutan unit yang akan diproduksi, kemudian dihitung persebaran *part consumption rate* dengan membagi seluruh unit produksi menjadi 14 periode waktu (dianggap 14 jam kerja dalam 1 hari atau setara 2 *shift* kerja). Dalam setiap periode waktu, dihitung berapa unit yang diproduksi bagi setiap jenis *sub-assembly*. Kemudian, dilanjutkan dengan menghitung standar deviasi dari setiap jenis *sub-assembly*. Aliran produksi yang konstan memiliki nilai *standar deviasi* yang kecil. Pada gambar 4.11 dan tabel 4.3 dicontohkan alokasi unit dengan *sub-assembly* warna 1D6 hasil optimasi untuk jadwal produksi 4 Juni 2018. Garis vertikal menunjukkan batas produksi tiap jamnya.



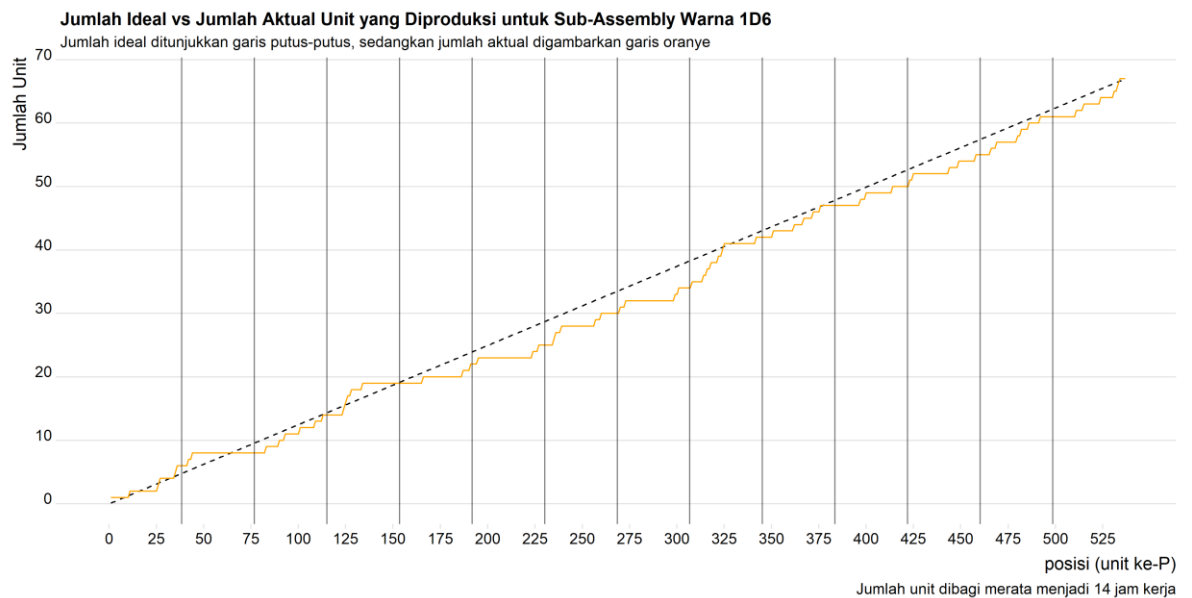
Gambar 4.11 Tingkat Produksi Unit Solusi Optimasi Warna 1D6 4 Juni 2018

Garis putus-putus menunjukkan jumlah *sub-assembly* ideal yang harus diproduksi dari posisi atau unit pertama hingga unit ke-P. Misalnya unit pertama hingga unit ke-75, idealnya diproduksi atau terdapat 10 unit dengan warna 1D6. Garis oranye menunjukkan jumlah aktual yang ada dari unit pertama hingga unit ke-P, misalnya pada unit pertama hingga unit ke-225 telah diproduksi 30 unit dengan warna 1D6. Fungsi tujuan minimasi *part consumption rate* menghitung kuadrat selisih antara nilai ideal dengan nilai aktual.

Tabel 4.5 Jadwal Hasil Optimasi 4 Juni 2018 Berdasarkan Warna dan Jam

Warna	1	2	3	4	5	6	7	8	9	10	11	12	13	14	mean	std dev
1D6	4	6	4	5	4	7	3	5	3	6	5	8	3	4	4.786	1.528
1G3	3	5	3	2	2	3	2	3	6	3	3	1	5	3	3.143	1.351
218	10	11	13	10	10	7	13	9	12	13	9	12	14	8	10.786	2.119
40	9	5	13	8	13	6	8	8	8	9	11	8	6	10	8.714	2.400
4V8	1	3	1	1	1	3	1	2	1	1	1	2	2	3	1.643	0.842
4W9	3	1	1	3	1	5	2	1	3	0	2	0	5	1	2.000	1.617
6W4	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0.214	0.802
70	8	7	4	8	7	8	9	7	6	6	7	8	2	10	6.929	2.018
8X2	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0.143	0.363
Total	38	38	39	38	38	39	38	38	39	38	38	39	38	39	38.357	0.497

Warna 1D6 rata-rata diproduksi sebanyak 4.7 atau 5 unit setiap jamnya, dengan standar deviasi sebesar 1.528, warna 1G3 rata-rata diproduksi sebanyak 3.1 atau 3 unit setiap jamnya, dengan standar deviasi sebesar 1.351, dan seterusnya. Bila dibandingkan dengan alokasi produksi unit per warna setiap jamnya hasil dari solusi aktual saat ini pada tabel 4.4, hasil optimasi memberikan standar deviasi yang lebih kecil untuk unit warna 1D6, yang menandakan bahwa tingkat penggunaan part-nya lebih konstan dibandingkan kondisi aktual.

**Gambar 4.12 Tingkat Produksi Unit Solusi Aktual Warna 1D6 4 Juni 2018**

Tabel 4.6 Jadwal Aktual 4 Juni 2018 Berdasarkan Warna dan Jam

Warna	1	2	3	4	5	6	7	8	9	10	11	12	13	14	mean	std dev
1D6	6	2	6	5	3	3	5	4	8	5	3	5	6	6	4.786	1.626
1G3	2	2	4	2	1	3	6	3	3	4	4	2	5	3	3.143	1.351
218	15	8	10	10	6	11	12	13	8	10	12	14	10	12	10.786	2.455
40	8	11	11	6	8	7	7	8	10	9	10	8	10	9	8.714	1.541
4V8	0	3	1	0	0	2	2	0	3	5	0	2	3	2	1.643	1.550
4W9	0	3	3	7	9	5	0	1	0	0	0	0	0	0	2.000	3.013
6W4	0	0	0	0	2	1	0	0	0	0	0	0	0	0	0.214	0.579
70	7	9	4	8	9	6	6	9	7	5	9	7	4	7	6.929	1.774
8X2	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0.143	0.363
Total	38	38	39	38	38	39	38	38	39	38	38	39	38	39	38.357	0.497

Untuk melakukan generalisasi dan mengetahui apakah perbedaan dari rata-rata persebaran/standar deviasi seluruh warna secara keseluruhan berbeda signifikan antara solusi aktual dengan solusi optimasi, dilakukan uji perbandingan rata-rata menggunakan uji t berpasangan dan uji wilcoxon. Berikut hasil uji perbandingan persebaran *part consumption rate* dilihat berdasarkan *sub-assembly* yang menjadi prioritas, yakni *suffix*, model, *grade*, mesin, dan warna:

a. Suffix

Tabel 4.7 menunjukkan rekapitulasi perbandingan persebaran *sub-assembly suffix*. Sebagian besar solusi optimasi memiliki rata-rata persebaran yang tidak berbeda secara signifikan. Seluruh rata-rata persebaran *suffix* pada tanggal 7 Juni memiliki nilai yang secara signifikan lebih rendah dibandingkan aktual, sedangkan pada tanggal 21 Juni separuh dari solusi optimasi memiliki rata-rata persebaran yang lebih rendah secara signifikan.

Tabel 4.7 Perbandingan Persebaran Sub-Assembly Suffix

Tanggal	Rataan Persebaran Part Aktual	Rataan Persebaran Part Optimasi			Beda Signifikan	
		Min	Mean	Max	Ya	Tidak
4-Jun-18	0.536	0.524	0.529	0.534	0	18
5-Jun-18	0.550	0.510	0.535	0.550	1	16
6-Jun-18	0.558	0.545	0.554	0.564	0	17
7-Jun-18	0.595	0.521	0.523	0.525	18	0
8-Jun-18	0.571	0.559	0.583	0.610	0	21
20-Jun-18	0.552	0.524	0.539	0.548	0	16
21-Jun-18	0.522	0.550	0.558	0.568	7	7
22-Jun-18	0.590	0.566	0.578	0.602	0	14
25-Jun-18	0.540	0.533	0.548	0.562	0	14
26-Jun-18	0.543	0.568	0.582	0.592	0	14

b. Model

Tabel 4.8 menunjukkan rekapitulasi perbandingan persebaran *sub-assembly* model. Dikarenakan hanya terdapat 2 variasi model, maka tidak dapat dilakukan uji statistik, sehingga rata-rata persebaran langsung dibandingkan antara aktual dengan optimasi. Sebagian besar hasil perbandingan menunjukkan bahwa kondisi aktual lebih rendah rata-rata persebarannya dibandingkan solusi optimasi.

Tabel 4.8 Perbandingan Persebaran Sub-Assembly Model

Tanggal	Rataan Persebaran Part Aktual	Rataan Persebaran Part Optimasi			Optimasi Lebih Baik	
		Min	Mean	Max	Ya	Tidak
4-Jun-18	0.688	0.967	1.115	1.276	0	18
5-Jun-18	0.967	1.390	1.628	1.864	0	17
6-Jun-18	1.026	0.968	1.314	2.030	1	16
7-Jun-18	1.130	1.057	1.088	1.090	18	0
8-Jun-18	1.035	1.202	1.497	1.878	0	21
20-Jun-18	1.246	0.734	1.016	1.323	5	11
21-Jun-18	1.351	1.460	1.772	2.381	0	14
22-Jun-18	1.548	0.972	1.216	1.670	13	1
25-Jun-18	0.800	1.225	1.383	1.673	0	14
26-Jun-18	0.542	1.566	1.753	1.919	0	14

c. Grade

Tabel 4.9 menunjukkan rekapitulasi perbandingan persebaran *sub-assembly grade*. Terdapat banyak solusi optimasi yang secara signifikan lebih baik dibandingkan kondisi aktual, dan tidak terdapat kondisi aktual yang lebih baik dibandingkan solusi optimasi.

Tabel 4.9 Perbandingan Persebaran Sub-Assembly Grade

Tanggal	Rataan Persebaran Part Aktual	Rataan Persebaran Part Optimasi			Beda Signifikan	
		Min	Mean	Max	Ya	Tidak
4-Jun-18	1.413	1.024	1.099	1.214	1	17
5-Jun-18	1.599	1.151	1.290	1.441	8	9
6-Jun-18	1.360	1.069	1.156	1.233	0	17
7-Jun-18	1.568	1.121	1.135	1.157	2	16
8-Jun-18	1.668	1.287	1.436	1.642	0	21
20-Jun-18	1.315	0.988	1.031	1.124	6	10
21-Jun-18	1.424	1.380	1.524	1.797	1	13
22-Jun-18	1.505	1.172	1.281	1.528	0	14
25-Jun-18	1.642	1.274	1.343	1.438	12	2
26-Jun-18	1.176	1.247	1.323	1.404	0	14

d. Mesin

Tabel 4.10 menunjukkan rekapitulasi perbandingan persebaran *sub-assembly* mesin. Dikarenakan hanya terdapat 4 variasi mesin, maka tidak dapat dilakukan uji statistik. Sebagian besar solusi optimasi menghasilkan rata-rata persebaran *sub-assembly* mesin yang lebih baik dibandingkan kondisi aktual.

Tabel 4.10 Perbandingan Persebaran Sub-Assembly Mesin

Tanggal	Rataan Persebaran Part Aktual	Rataan Persebaran Part Optimasi			Optimasi Lebih Baik	
		Min	Mean	Max	Ya	Tidak
4-Jun-18	1.655	1.145	1.233	1.379	18	0
5-Jun-18	1.569	1.373	1.524	1.686	11	6
6-Jun-18	1.597	1.176	1.296	1.488	17	0
7-Jun-18	1.567	1.135	1.188	1.203	18	0
8-Jun-18	1.926	1.551	1.662	1.828	21	0
20-Jun-18	1.628	1.021	1.240	1.445	16	0
21-Jun-18	1.854	1.935	2.071	2.174	0	14
22-Jun-18	1.489	1.172	1.302	1.448	14	0
25-Jun-18	1.522	1.373	1.511	1.733	10	4
26-Jun-18	1.119	1.481	1.639	1.885	0	14

e. Warna

Tabel 4.11 menunjukkan rekapitulasi perbandingan persebaran *sub-assembly* warna. Tidak terdapat solusi optimasi yang secara signifikan lebih baik dibandingkan kondisi aktual. Pada tanggal 26 Juni 2018 terdapat 7 solusi optimasi yang secara signifikan lebih tinggi dibandingkan kondisi aktual.

Tabel 4.11 Perbandingan Persebaran Sub-Assembly Warna

Tanggal	Rataan Persebaran Part Aktual	Rataan Persebaran Part Optimasi			Beda Signifikan	
		Min	Mean	Max	Ya	Tidak
4-Jun-18	1.583	1.334	1.557	1.734	0	18
5-Jun-18	1.646	1.880	2.064	2.326	0	17
6-Jun-18	1.915	2.201	2.305	2.502	0	17
7-Jun-18	2.081	1.958	2.032	2.048	0	18
8-Jun-18	2.187	1.952	2.330	2.905	0	21
20-Jun-18	2.080	1.587	1.865	2.108	0	16
21-Jun-18	1.924	2.091	2.467	2.642	0	14
22-Jun-18	1.870	2.145	2.374	2.728	0	14
25-Jun-18	1.860	1.946	2.091	2.151	0	14
26-Jun-18	1.675	2.551	2.867	3.144	7	7

4.2.3.4. Perbandingan Jumlah Proses *Flushing*

Jumlah proses *flushing* dihitung dengan melihat perubahan warna pada jadwal urutan produksi. Apabila warna pada urutan ke- $t+1$ tidak sama dengan urutan ke- t , maka terjadi perubahan warna dan dilakukan proses *flushing*. Tabel 4.12 menunjukkan perbandingan jumlah proses *flushing* antara jadwal aktual dengan jadwal hasil optimasi menggunakan GrEA. Jadwal aktual menghasilkan rata-rata 432.44 proses *flushing* dalam satu hari, sedangkan optimasi menggunakan GrEA menghasilkan rata-rata 289.88 proses *flushing*. Rata-rata proses *flushing* dapat berkurang sebanyak 143.48.

Tabel 4.12 Perbandingan Jumlah Proses *Flushing*

No	Tanggal	Jumlah unit	Aktual	Optimasi				Pengurangan			
				Min	Max	Mean	std dev.	Min	Max	Mean	std dev.
1	4-Jun	537	439	260	306	278.78	15.51	133	179	160.22	15.51
2	5-Jun	555	455	284	316	303.29	10.20	139	171	151.71	10.20
3	6-Jun	567	460	242	265	255.88	6.99	195	218	204.12	6.99
4	7-Jun	567	434	283	293	285.94	2.91	141	151	148.06	2.91
5	8-Jun	584	437	315	393	358.10	23.42	44	122	78.90	23.42
6	20-Jun	455	354	222	257	240.38	12.60	97	132	113.63	12.60
7	21-Jun	588	457	321	345	335.57	8.66	112	136	121.43	8.66
8	22-Jun	622	484	235	246	240.57	2.69	238	249	243.43	2.69
9	25-Jun	522	403	237	282	249.86	13.84	121	166	153.14	13.84
10	26-Jun	697	408	322	348	332.14	8.39	60	86	75.86	8.39
Rata-rata			432.44			289.88				143.48	

4.2.3.5. Perbandingan Biaya Proses *Flushing*

Proses *flushing* membutuhkan bahan kimia *thinner* dan larutan lainnya untuk membersihkan *nozzle* atau penyemprot robot cat dari sisa-sisa cat sebelum menyemprotkan warna yang berbeda dari warna unit sebelumnya. Biaya yang diperlukan dalam 1 kali proses *flushing* adalah sebesar Rp 36,624.11, yang didapatkan dari data anggaran *painting shop* periode April 2018. Rata-rata biaya aktual proses *flushing* yang dikeluarkan pada periode 4-26 Juni 2019 setiap harinya adalah sebesar Rp 15,861,902.50. Berdasarkan sub-bab 4.2.3.4, hasil optimasi menggunakan GrEA menghasilkan jumlah proses *flushing* yang lebih sedikit, sehingga biaya yang diperlukan juga berkurang. Rata-rata biaya proses *flushing* yang dikeluarkan berdasarkan rata-rata hasil optimasi adalah Rp 10,616,723.14. Tabel 4.13 menunjukkan rekapitulasi perhitungan biaya proses *flushing*.

Tabel 4.13 Perhitungan Biaya Proses *Flushing*

No	Tanggal	Biaya Optimasi				Biaya Aktual
		Min	Max	Mean	Std dev.	
1	4-Jun	Rp 9,522,268.87	Rp 11,206,977.98	Rp 10,209,988.29	Rp 567,976.29	Rp 16,077,984.75
2	5-Jun	Rp 10,401,247.54	Rp 11,573,219.09	Rp 11,107,877.45	Rp 373,444.40	Rp 16,663,970.53
3	6-Jun	Rp 8,863,034.87	Rp 9,705,389.43	Rp 9,371,463.71	Rp 256,178.62	Rp 16,847,091.08
4	7-Jun	Rp 10,364,623.43	Rp 10,730,864.54	Rp 10,472,461.09	Rp 106,757.33	Rp 15,894,864.20
5	8-Jun	Rp 11,536,594.98	Rp 14,393,275.64	Rp 13,114,919.77	Rp 857,564.84	Rp 16,004,736.53
6	20-Jun	Rp 8,130,552.65	Rp 9,412,396.54	Rp 8,803,520.69	Rp 461,608.20	Rp 12,964,935.31
7	21-Jun	Rp 11,756,339.65	Rp 12,635,318.31	Rp 12,290,005.27	Rp 317,087.79	Rp 16,737,218.75
8	22-Jun	Rp 8,606,666.10	Rp 9,009,531.32	Rp 8,810,714.72	Rp 98,578.73	Rp 17,726,069.75
9	25-Jun	Rp 8,679,914.32	Rp 10,327,999.32	Rp 9,150,795.75	Rp 506,884.87	Rp 14,759,516.75
10	26-Jun	Rp 11,792,963.76	Rp 12,745,190.65	Rp 12,164,436.88	Rp 307,311.35	Rp 14,942,637.31
Rata-rata				Rp 10,616,723.14		Rp 15,861,920.50

Tabel 4.14 menunjukkan rekapitulasi perhitungan *saving* yang bisa didapatkan pada biaya proses *flushing*. Dengan menggunakan optimasi, biaya proses *flushing* dapat dihemat rata-rata sebanyak Rp 5,254,773.53 dalam satu hari.

Tabel 4.14 Perhitungan *Saving* Biaya Proses *Flushing*

No	Tanggal	<i>Saving</i>			
		Min	Max	Mean	Std dev.
1	4-Jun	Rp 4,871,006.77	Rp 6,555,715.88	Rp 5,867,996.46	Rp 567,976.29
2	5-Jun	Rp 5,090,751.44	Rp 6,262,722.99	Rp 5,556,093.08	Rp 373,444.40
3	6-Jun	Rp 7,141,701.65	Rp 7,984,056.21	Rp 7,475,627.37	Rp 256,178.62
4	7-Jun	Rp 5,163,999.66	Rp 5,530,240.77	Rp 5,422,403.11	Rp 106,757.33
5	8-Jun	Rp 1,611,460.89	Rp 4,468,141.55	Rp 2,889,816.76	Rp 857,564.84
6	20-Jun	Rp 3,552,538.77	Rp 4,834,382.66	Rp 4,161,414.62	Rp 461,608.20
7	21-Jun	Rp 4,101,900.44	Rp 4,980,879.10	Rp 4,447,213.48	Rp 317,087.79
8	22-Jun	Rp 8,716,538.43	Rp 9,119,403.65	Rp 8,915,355.03	Rp 98,578.73
9	25-Jun	Rp 4,431,517.44	Rp 6,079,602.43	Rp 5,608,721.01	Rp 506,884.87
10	26-Jun	Rp 2,197,446.66	Rp 3,149,673.55	Rp 2,778,200.42	Rp 307,311.35
Rata-rata				Rp 5,254,773.53	

4.2.3.6. Prediksi Tingkat Kualitas

Hubungan proses *flushing* dengan tingkat kualitas produksi harian dimodelkan menggunakan regresi linier sederhana. Data yang digunakan adalah data jadwal urutan produksi harian dan data kualitas produksi periode Januari-Juni 2018 (terlampir). Karena jumlah proses *flushing* bergantung pada jumlah unit yang diproduksi, maka proses *flushing* diukur melalui rasio/perbandingan antara jumlah proses *flushing* dengan jumlah unit yang diproduksi.

4.2.3.6.1. Model Awal

Tabel 4.15 menunjukkan karakteristik model yang dihasilkan.

Tabel 4.15 Ringkasan Model Awal Regresi Linier

Koefisien Model					Performa Model				
	Estimate	Std. Error	t	p-value	R-Squared	RMSE	MAE	F	p-value
Intercept	1.11836	0.06351	17.609	0.000	0.1987	0.05299	0.0438	28.76	0.000
Flush Ratio	-0.44217	0.08245	-5.363	0.000					

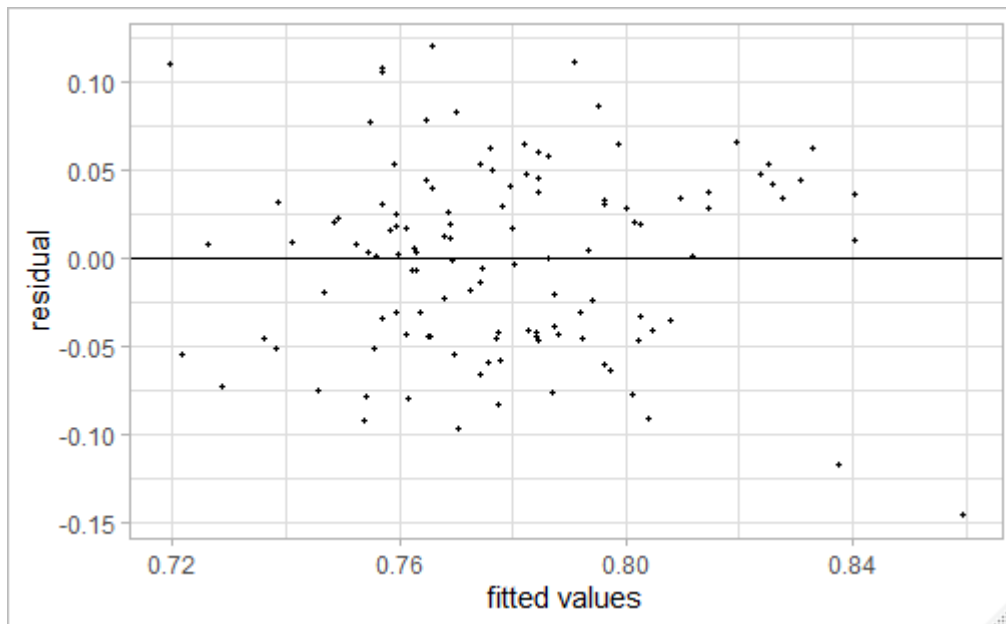
Model regresi linear yang dihasilkan adalah sebagai berikut:

$$\hat{Y} = 1.11836 - 0.44217 X \dots\dots\dots (4.1)$$

\hat{Y} adalah tingkat kualitas, dan X adalah rasio *flushing*. Model yang dihasilkan memiliki nilai R^2 sebesar 0.1987 dengan nilai *Root Mean Squared Error* (RMSE) 0.05299 dan *Mean Absolute Error* (MAE) 0.0438. Model yang dihasilkan bersifat signifikan, ditunjukkan dari nilai *p-value* uji F yang bernilai kurang dari 0,05. Setelah mendapatkan model regresi linier antara proses *flushing* dengan tingkat kualitas, selanjutnya dilakukan uji asumsi klasik terhadap model tersebut. Asumsi klasik yang harus terpenuhi adalah linearitas, normalitas, homokedastisitas, dan tidak berautokorelasi. Berikut adalah output dari uji asumsi klasik terhadap regresi linier:

a. Linearitas

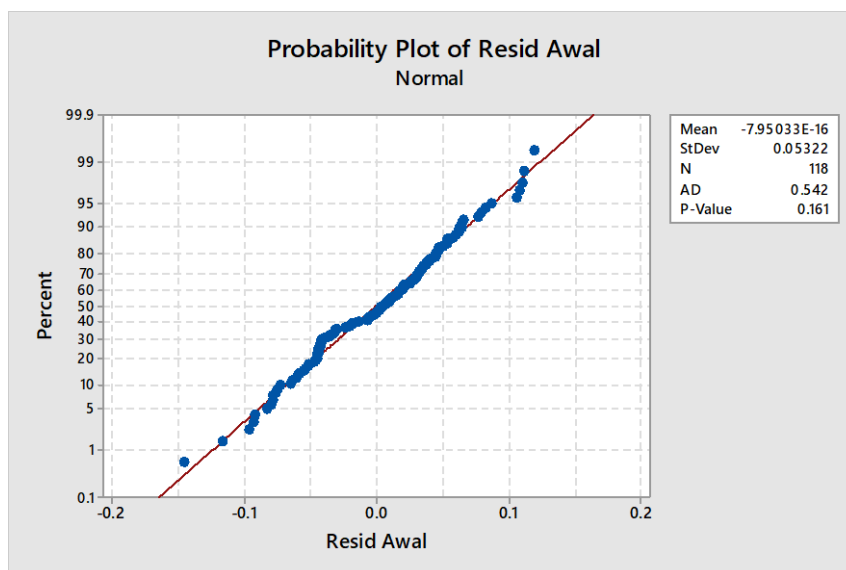
Linearitas dari model dapat dilihat dengan mengamati diagram tebar/*scatter plot* antara residual/*error* dengan nilai *fitted value* (nilai \hat{y} atau kualitas berdasarkan model). Gambar 4.13 menunjukkan diagram tebar dari model awal regresi linear. Titik-titik data tersebar secara simetris di sekitar garis horizontal, sehingga dapat disimpulkan bahwa model bersifat linear.



Gambar 4.13 Scatter Plot Fitted Values vs Residual Model Awal

b. Normalitas

Uji normalitas dilakukan dengan menguji nilai residual model menggunakan uji Anderson-Darling. Gambar 4.14 menunjukkan grafik p-p plot beserta nilai statistik uji Anderson Darling. Didapatkan nilai statistik Anderson-Darling sebesar 0.542 dengan p -value 0.161. Karena nilai p -value lebih besar dari 0.05, maka dapat disimpulkan bahwa residual berdistribusi normal.



Gambar 4.14 Uji Normalitas Model Awal Regresi

c. Bebas autokorelasi

Autokorelasi menunjukkan adanya hubungan/korelasi antara residual pada periode t dengan periode sebelumnya. Autokorelasi dideteksi menggunakan uji durbin-watson. Hasil uji durbin watson menunjukkan nilai statistik durbin-watson sebesar 0.9284. Batas bawah dan batas atas nilai kritis uji durbin-watson dengan $\alpha = 0.05$ berturut-turut adalah 1.678 dan 1.713. Karena nilai statistik durbin-watson lebih rendah dibandingkan batas bawah, maka dapat disimpulkan bahwa terdapat autokorelasi pada nilai residual.

d. Homokedastisitas (variansi error bersifat konstan)

Homokedastisitas dapat dideteksi dengan membandingkan nilai prediksi terhadap residualnya. Model dikatakan bersifat homokedastisitas ketika tidak terbentuk pola data berupa corong/*funnel*. Berdasarkan Gambar 4.11, tidak terdapat pola berupa corong/*funnel* dari persebaran data, sehingga dapat disimpulkan bahwa data bersifat homokedastisitas.

Berdasarkan hasil uji asumsi klasik yang dilakukan, diketahui bahwa model awal memiliki autokorelasi. Model yang tidak memenuhi asumsi klasik memiliki estimator yang bersifat bias, sehingga mempengaruhi model regresi linear yang terbentuk. Model perlu dimodifikasi agar dapat menghilangkan autokorelasi dari residual.

4.2.3.6.2. Model Modifikasi

Model regresi linear perlu dimodifikasi agar dapat memenuhi asumsi tidak terdapat autokorelasi. Karena data bersifat *time series*, maka ditambahkan variabel waktu/trend (t) ke dalam model regresi linear. Tabel 4.16 menunjukkan karakteristik model modifikasi.

Tabel 4.16 Ringkasan Model Modifikasi Regresi Linear

Koefisien Model						Performa Model				
	Estimate	Std. Error	t	p-value	VIF	R-Squared	RMSE	MAE	F	p-value
Intercept	1.10609	0.04360	25.368	0.000		0.6258	0.0362	0.0285	96.15	0.000
Flush Ratio	-0.33704	0.05732	-5.879	0.000	1.03					
trend	-0.00115	0.00010	-11.457	0.000	1.03					

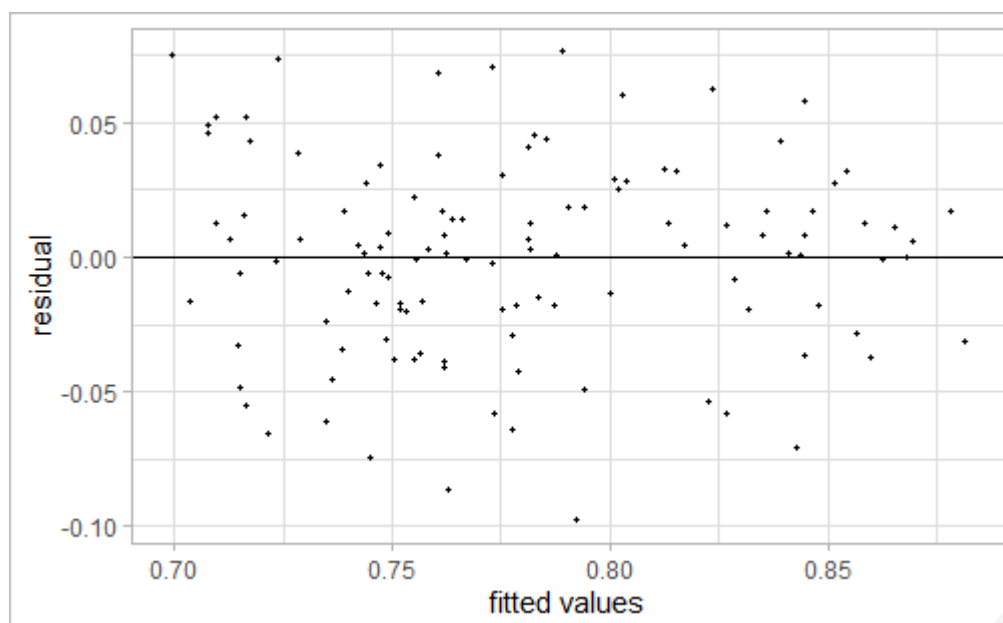
Model regresi linear yang dihasilkan adalah sebagai berikut:

$$\hat{Y} = 1.10609 - 0.33704 X_1 - 0.00115 X_2 \dots \dots \dots (4.2)$$

Model yang dihasilkan memiliki nilai R^2 sebesar 0.6258 dengan nilai *Root Mean Squared Error* (RMSE) 0.0362 dan *Mean Absolute Error* (MAE) 0.0285. Model yang dihasilkan bersifat signifikan, ditunjukkan dari nilai *p-value* uji F yang bernilai kurang dari 0.05. Selanjutnya dilakukan uji asumsi klasik terhadap model modifikasi, hasilnya sebagai berikut:

a. Linearitas

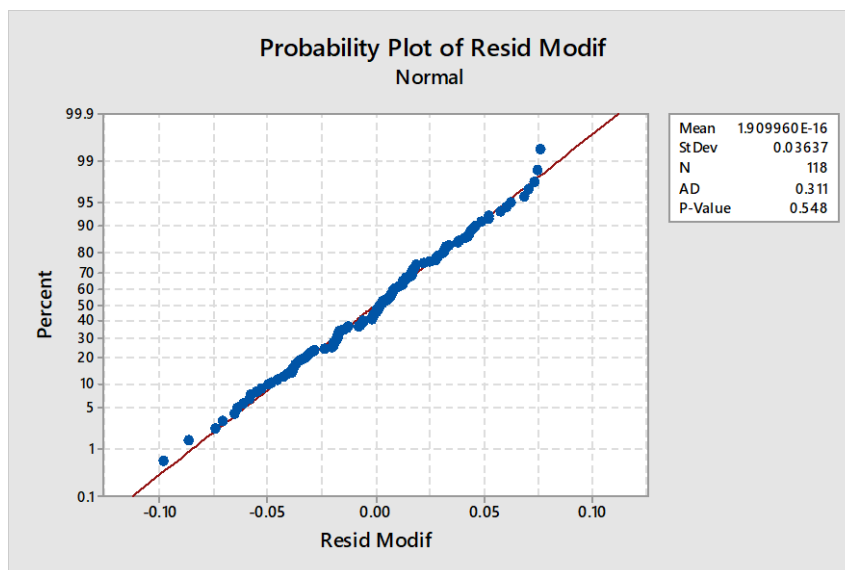
Linearitas dari model dapat dilihat dengan mengamati diagram tebar/*scatter plot* antara residual/error dengan nilai *fitted value* (nilai \hat{y} atau kualitas berdasarkan model). Gambar 4.15 menunjukkan diagram tebar dari model modifikasi regresi linear. Titik-titik data tersebar secara simetris di sekitar garis horizontal, sehingga dapat disimpulkan bahwa model bersifat linear.



Gambar 4.15 Scatter Plot Fitted Values vs Residual Model Modifikasi

b. Normalitas

Uji normalitas dilakukan dengan menguji residual menggunakan uji Anderson-Darling. Gambar 4.16 menunjukkan grafik p-p plot beserta nilai statistik uji Anderson Darling. Didapatkan nilai statistik Anderson-Darling sebesar 0.311 dengan *p-value* 0.548. Karena nilai *p-value* lebih besar dari 0.05, maka dapat disimpulkan bahwa residual berdistribusi normal.



Gambar 4.16 Uji Normalitas Model Modifikasi Regresi

c. Autokorelasi

Autokorelasi menunjukkan adanya hubungan/korelasi antara residual pada periode t dengan periode sebelumnya. Hasil uji durbin watson menunjukkan nilai statistik durbin-watson sebesar 1.69435. Batas bawah dan batas atas nilai kritis uji durbin-watson dengan $\alpha = 0.05$ berturut-turut adalah 1.678 dan 1.713. Karena nilai statistik durbin-watson berada di antara batas bawah dan batas atas, maka dapat disimpulkan bahwa tidak terdapat autokorelasi pada nilai residual.

d. Homokedastisitas

Homokedastisitas dapat dideteksi dengan membandingkan nilai prediksi terhadap residualnya. Gambar 4.13 menunjukkan *scatter plot* residual/error terhadap nilai prediksi. Tidak terdapat pola berupa

corong/*funnel* dari persebaran data, sehingga dapat disimpulkan bahwa data bersifat homokedastisitas.

e. Multikolinearitas

Adanya multikolinearitas menunjukkan bahwa terdapat dua atau lebih variabel independen yang saling berhubungan. Multikolinearitas dapat diukur dengan melihat nilai VIF dari masing-masing variabel independen. Apabila nilai $VIF > 10$, maka terdapat multikolineartias. Berdasarkan model regresi, nilai VIF dari variabel *flush ratio* dan *trend* masing-masing memiliki nilai VIF 1.03, sehingga dapat diambil kesimpulan bahwa tidak terjadi multikolinearitas.

Berdasarkan hasil uji asumsi klasik, dapat disimpulkan bahwa seluruh asumsi terpenuhi, sehingga model regresi linier yang dihasilkan memiliki estimator yang baik dan tidak bias. Maka, model dapat digunakan untuk memprediksi tingkat kualitas.

4.2.3.6.3. Hasil Prediksi Tingkat Kualitas

Prediksi dilakukan untuk mengetahui estimasi tingkat kualitas dari solusi-solusi yang dihasilkan optimasi menggunakan GrEA. Tabel 4.17 menunjukkan perbandingan rasio *flushing* antara jadwal aktual dengan solusi optimasi.

Tabel 4.17 Perbandingan Rasio *Flushing* Aktual dengan Optimasi

No	tanggal	Jumlah unit	Rasio Flushing Aktual	Kualitas Aktual	Prediksi Kualitas Aktual	Rasio Flushing Optimasi			
						min	Max	mean	std dev
1	4-Jun	537	81.56%	72.25%	70.97%	48.42%	56.98%	51.91%	2.89%
2	5-Jun	555	79.46%	75.68%	70.78%	51.17%	56.94%	54.65%	1.84%
3	6-Jun	567	76.37%	76.19%	70.95%	42.68%	46.74%	45.13%	1.23%
4	7-Jun	567	72.66%	79.72%	72.38%	49.91%	51.68%	50.43%	0.51%
5	8-Jun	584	69.01%	76.71%	72.85%	53.94%	67.29%	61.32%	4.01%
6	20-Jun	455	77.58%	76.04%	71.73%	48.79%	56.48%	52.83%	2.77%
7	21-Jun	588	70.92%	76.87%	71.64%	54.59%	58.67%	57.07%	1.47%
8	22-Jun	622	67.68%	70.90%	71.50%	37.78%	39.55%	38.68%	0.43%
9	25-Jun	522	77.20%	73.18%	71.59%	45.40%	54.02%	47.87%	2.65%
10	26-Jun	697	58.54%	71.4%	77.76%	46.20%	49.93%	47.65%	1.20%
Overall				74.89%	72.22%			51.22%	

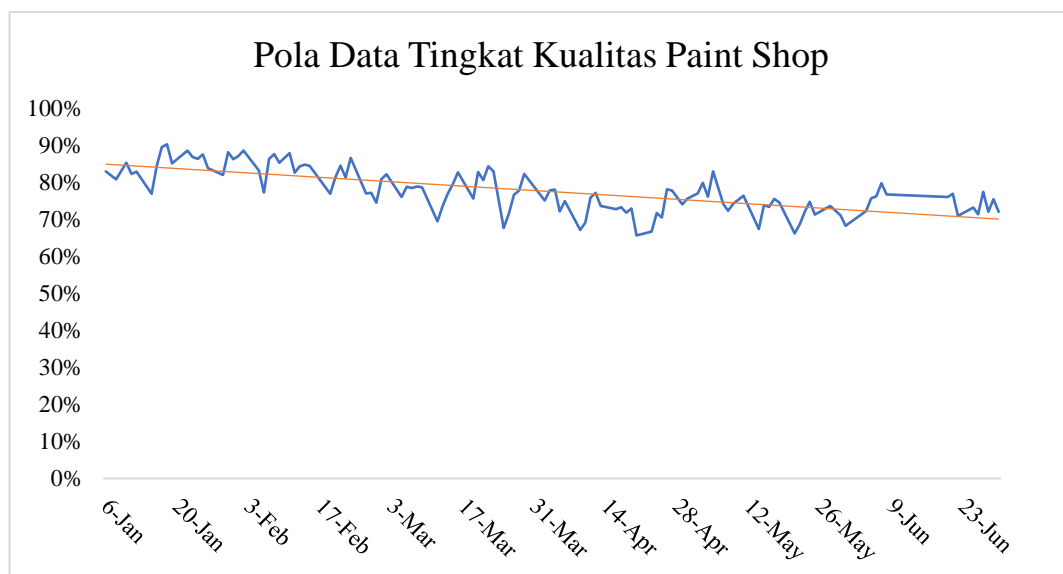
Tabel 4.18 menunjukkan perbandingan tingkat kualitas antara jadwal aktual dengan jadwal yang dihasilkan solusi optimasi. Rata-rata tingkat kualitas

aktual periode 4-26 Juni 2018 adalah 74.89% dan rata-rata prediksi kualitas aktualnya sebesar 72.22%. Rata-rata tingkat kualitas berdasarkan hasil optimasi adalah 80.78%. Kualitas proses dapat meningkat rata-rata sebanyak 5.70%.

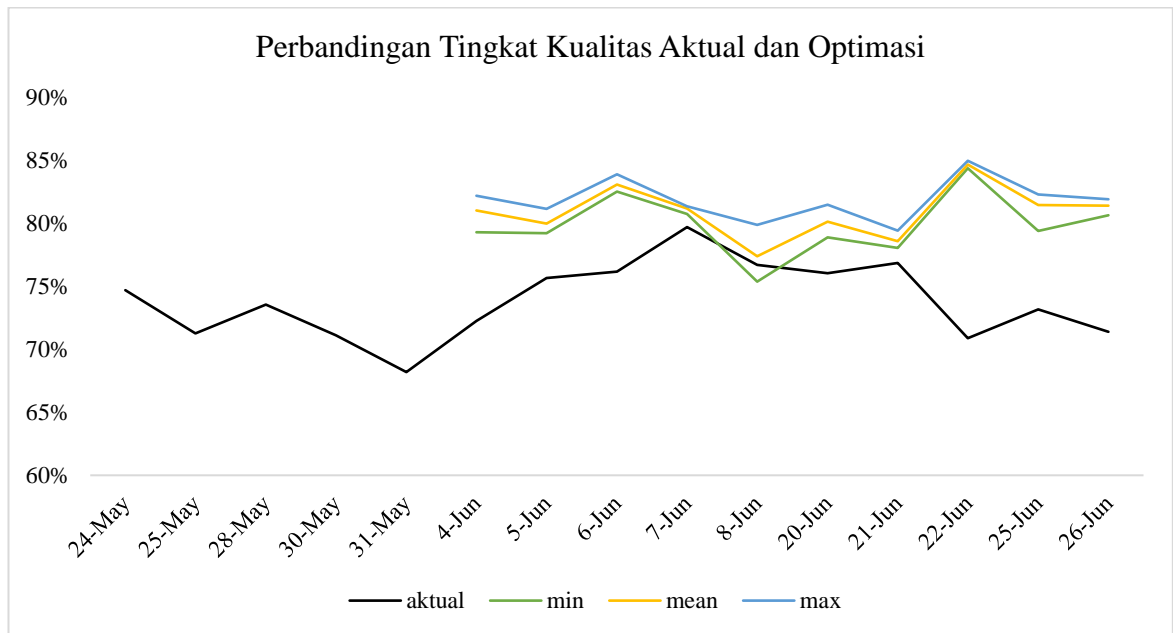
Tabel 4.18 Perbandingan Tingkat Kualitas Aktual dengan Prediksi Optimasi

No	tanggal	Kualitas Aktual	Prediksi Kualitas Aktual	Tingkat Kualitas Optimasi				Peningkatan Kualitas Optimasi terhadap Aktual			
				min	max	mean	std dev	min	max	mean	σ
1	4-Jun	72.25%	70.97%	79.32%	82.21%	81.03%	0.97%	7.07%	9.96%	8.78%	0.97%
2	5-Jun	75.68%	70.78%	79.22%	81.17%	79.99%	0.62%	3.55%	5.49%	4.32%	0.62%
3	6-Jun	76.19%	70.95%	82.54%	83.91%	83.09%	0.42%	6.35%	7.72%	6.90%	0.42%
4	7-Jun	79.72%	72.38%	80.77%	81.36%	81.18%	0.17%	1.05%	1.64%	1.47%	0.17%
5	8-Jun	76.71%	72.85%	75.39%	79.89%	77.40%	1.35%	-1.33%	3.18%	0.69%	1.35%
6	20-Jun	76.04%	71.73%	78.91%	81.51%	80.15%	0.93%	2.87%	5.46%	4.10%	0.93%
7	21-Jun	76.87%	71.64%	78.06%	79.44%	78.60%	0.50%	1.19%	2.57%	1.73%	0.50%
8	22-Jun	70.90%	71.50%	84.39%	84.99%	84.69%	0.15%	13.49%	14.09%	13.79%	0.15%
9	25-Jun	73.18%	71.59%	79.40%	82.30%	81.47%	0.89%	6.22%	9.12%	8.29%	0.89%
10	26-Jun	71.4%	77.76%	80.66%	81.92%	81.43%	0.41%	9.27%	10.53%	10.04%	0.41%
Rata-rata		74.89%	72.22%			80.78%				5.70%	

Gambar 4.17 menunjukkan plot data periode Januari-Juni 2018. Terdapat fluktuasi data dengan tren linear menurun. Gambar 4.18 menunjukkan gambar plot data tingkat kualitas aktual dan optimasi. Garis hitam menunjukkan tingkat kualitas aktual. Warna hijau menunjukkan nilai kualitas minimal yang dihasilkan dari optimasi dengan GrEA, sedangkan warna biru menunjukkan nilai kualitas maksimal dan warna kuning menunjukkan nilai rata-rata tingkat kualitas yang dihasilkan dari solusi GrEA.



Gambar 4.17 Pola Data Tingkat Kualitas Paint Shop



Gambar 4.18 Perbandingan Tingkat Kualitas Aktual dan Optimasi

4.2.3.7. Hubungan Dominansi Solusi Optimasi dengan Jadwal Aktual

Dua solusi atau himpunan solusi pada permasalahan fungsi tujuan ganda (*multi-objective*) tidak dapat dibandingkan secara langsung. Untuk mengetahui apakah suatu solusi lebih unggul dibandingkan solusi lainnya, digunakan *additive epsilon indicator*. Tabel 4.19 menunjukkan contoh perhitungan $I_{\epsilon+}$ (Optimasi, Aktual). Karena *additive epsilon indicator* dihitung dengan asumsi bahwa permasalahan berupa minimasi dan fungsi tujuan *paint* berupa permasalahan maksimasi, maka fungsi tujuan *paint* dikonversi menjadi negatif. Fungsi tujuan dari setiap alternatif dikurangi dengan fungsi tujuan solusi aktual, kemudian dicari nilai maksimal antara 2 fungsi tujuan pada setiap alternatif. Setelah mendapatkan nilai maksimal dari setiap alternatif, dicari nilai minimal dari pasangan solusi alternatif dengan solusi aktual. Karena hanya terdapat 1 solusi aktual, maka nilai minimal tiap solusi alternatif adalah sama dengan nilai maksimal setiap solusi optimasi. Setelah mendapatkan nilai minimal, maka dicari nilai maksimalnya. Berdasarkan hasil perhitungan, maka nilai $I_{\epsilon+}$ (Optimasi, Aktual) sebesar -0.10962.

Tabel 4.19 Menghitung $I_{\varepsilon+}$ (Optimasi, Aktual)

Solusi	Assembly	Paint	Optimasi – Aktual		Max	Min	Max
			Assembly	Paint			
Aktual	93493.02	-0.261					-0.10962
Alternatif 1	32265.22	-0.336	-61227.8	-0.07473	-0.074730518	-0.074730518	
Alternatif 2	32693.1	-0.337	-60799.9	-0.07564	-0.075640987	-0.075640987	
Alternatif 3	37118.25	-0.364	-56374.8	-0.10264	-0.102642788	-0.102642788	
Alternatif 4	35660.83	-0.354	-57832.2	-0.09254	-0.092537316	-0.092537316	
Alternatif 5	37941.64	-0.369	-55551.4	-0.10757	-0.107568659	-0.107568659	
Alternatif 6	36556.65	-0.360	-56936.4	-0.09907	-0.09907339	-0.09907339	
Alternatif 7	36042.78	-0.357	-57450.2	-0.0956	-0.095598444	-0.095598444	
Alternatif 8	32209.63	-0.330	-61283.4	-0.06875	-0.068753238	-0.068753238	
Alternatif 9	35382.7	-0.351	-58110.3	-0.08998	-0.089975942	-0.089975942	
Alternatif 10	38466.16	-0.371	-55026.9	-0.110	-0.109623834	-0.109623834	
Alternatif 11	37380.05	-0.365	-56113	-0.104	-0.10400024	-0.10400024	
Alternatif 12	34095.67	-0.337	-59397.3	-0.07586	-0.075858177	-0.075858177	
Alternatif 13	35759.7	-0.354	-57733.3	-0.09312	-0.093116935	-0.093116935	
Alternatif 14	35468.02	-0.352	-58025	-0.09056	-0.090555727	-0.090555727	
Alternatif 15	36746.65	-0.361	-56746.4	-0.09968	-0.099678599	-0.099678599	
Alternatif 16	37642.76	-0.366	-55850.3	-0.1047	-0.104698041	-0.104698041	
Alternatif 17	35138.59	-0.337	-58354.4	-0.07628	-0.076279912	-0.076279912	
Alternatif 18	32068.97	-0.328	-61424	-0.0667	-0.0666967	-0.0666967	

Tabel 4.20 menunjukkan rekapitulasi nilai *additive epsilon indicator* antara solusi-solusi optimasi dengan solusi aktual yang dipakai perusahaan. Metode interpretasi nilai *additive epsilon indicator* yang digunakan sesuai dengan tabel 2.3. Dapat disimpulkan bahwa solusi-solusi yang dihasilkan GrEA mendominasi/lebih baik dibandingkan dengan solusi saat ini.

Tabel 4.20 Rekapitulasi Indikator Hubungan Dominansi

No	Tanggal	$I_{\varepsilon+}$ (Optimasi, Aktual)	$I_{\varepsilon+}$ (Aktual, Optimasi)	Interpretasi
1	4-Jun-19	-0.109	61424.048	Solusi optimasi lebih baik
2	5-Jun-19	-0.102	62302.739	Solusi optimasi lebih baik
3	6-Jun-19	-0.122	60473.513	Solusi optimasi lebih baik
4	7-Jun-19	-0.083	113101.407	Solusi optimasi lebih baik
5	8-Jun-19	-0.070	114947.890	Solusi optimasi lebih baik
6	20-Jun-19	-0.090	49227.701	Solusi optimasi lebih baik
7	21-Jun-19	-0.083	113290.435	Solusi optimasi lebih baik
8	22-Jun-19	-0.137	64960.714	Solusi optimasi lebih baik
9	25-Jun-19	-0.104	113542.353	Solusi optimasi lebih baik
10	26-Jun-19	-0.069	28025.8565	Solusi optimasi lebih baik

BAB V

ANALISIS DAN PEMBAHASAN

5.1. Analisis Fungsi Tujuan

Fungsi tujuan yang ingin dicapai adalah minimasi penyimpangan *part consumption rate* dan minimasi jumlah *flushing* melalui maksimasi densifikasi warna. Semakin kecil nilai penyimpangan *part consumption rate*, maka semakin baik karena penggunaan material lebih konstan, menyebabkan aliran produksi menjadi lebih *smooth*/lancar dan dapat meminimasi tingkat persediaan. Semakin besar nilai densifikasi warna, maka unit-unit dengan warna yang sama semakin berdekatan satu sama lain (bukan berarti warna yang sama saling berurutan/berjejer), sehingga proses *flushing* dapat diminimasi. Analisis perhitungan fungsi tujuan secara detail adalah sebagai berikut:

a. 4 Juni 2019

Pada 4 Juni 2019, terdapat 557 unit yang akan diproduksi, dengan 70 jenis *suffix*, 2 model, 8 *grade* mobil, 4 jenis mesin dan 9 warna berbeda. Fungsi tujuan dari solusi aktual untuk minimasi penyimpangan *part consumption rate* dan maksimasi densifikasi warna berturut-turut adalah 93493.02 dan 0.261. Optimasi menggunakan GrEA berhenti setelah 2323 iterasi dengan waktu komputasi 106.72 jam dan dihasilkan 18 alternatif solusi berbeda. Untuk fungsi tujuan minimasi penyimpangan *part consumption rate*, nilai minimal yang dihasilkan GrEA adalah 32068.97 dan nilai maksimal sebesar 38466.16, dengan rata-rata 35479.85. Seluruh nilai penyimpangan *part consumption rate* dari solusi optimasi lebih rendah dibandingkan nilai solusi aktual. Untuk fungsi tujuan maksimasi densifikasi warna, nilai minimal yang dihasilkan GrEA adalah 0.328 dan nilai maksimal sebesar 0.371, dengan rata-rata 0.352. Seluruh nilai densifikasi warna dari solusi optimasi lebih besar dibanding nilai solusi aktual.

b. 5 Juni 2019

Pada 5 Juni 2019, terdapat 555 unit yang akan diproduksi, dengan 74 jenis *suffix*, 2 model, 8 *grade* mobil, 4 jenis mesin dan 8 warna berbeda. Fungsi tujuan dari solusi aktual untuk minimasi penyimpangan *part consumption rate* dan maksimasi densifikasi warna berturut-turut adalah 101749.9 dan 0.261. Optimasi menggunakan GrEA berhenti setelah 944 iterasi dengan waktu komputasi 38.12 jam dan dihasilkan 17 alternatif solusi berbeda. Untuk fungsi tujuan minimasi penyimpangan *part consumption rate*, nilai minimal yang dihasilkan GrEA adalah 39447.20 dan nilai maksimal sebesar 51521.86, dengan rata-rata 45411.42. Seluruh nilai penyimpangan *part consumption rate* dari solusi optimasi lebih rendah dibandingkan nilai solusi aktual. Untuk fungsi tujuan maksimasi densifikasi warna, nilai minimal yang dihasilkan GrEA adalah 0.334 dan nilai maksimal sebesar 0.363, dengan rata-rata 0.349. Seluruh nilai densifikasi warna dari solusi optimasi lebih besar dibanding nilai solusi aktual.

c. 6 Juni 2019

Pada 6 Juni 2019, terdapat 567 unit yang akan diproduksi, dengan 69 jenis *suffix*, 2 model, 8 *grade* mobil, 4 jenis mesin dan 8 warna berbeda. Fungsi tujuan dari solusi aktual untuk minimasi penyimpangan *part consumption rate* dan maksimasi densifikasi warna berturut-turut adalah 101469.2 dan 0.266. Optimasi menggunakan GrEA berhenti setelah 1937 iterasi dengan waktu komputasi 78.22 jam dan dihasilkan 17 alternatif solusi berbeda. Untuk fungsi tujuan minimasi penyimpangan *part consumption rate*, nilai minimal yang dihasilkan GrEA adalah 40995.68 dan nilai maksimal sebesar 50307.99, dengan rata-rata 45012.26. Seluruh nilai penyimpangan *part consumption rate* dari solusi optimasi lebih rendah dibandingkan nilai solusi aktual. Untuk fungsi tujuan maksimasi densifikasi warna, nilai minimal yang dihasilkan GrEA adalah 0.366 dan nilai maksimal sebesar 0.388, dengan rata-rata 0.375. Seluruh nilai densifikasi warna dari solusi optimasi lebih besar dibanding nilai solusi aktual.

d. 7 Juni 2019

Pada 7 Juni 2019, terdapat 567 unit yang akan diproduksi, dengan 69 jenis *suffix*, 2 model, 8 *grade* mobil, 4 jenis mesin dan 8 warna berbeda. Fungsi tujuan dari

solusi aktual untuk minimasi penyimpangan *part consumption rate* dan maksimasi densifikasi warna berturut-turut adalah 149250.7 dan 0.287. Optimasi menggunakan GrEA berhenti setelah 1937 iterasi dengan waktu komputasi 84.68 jam dan dihasilkan 18 alternatif solusi berbeda. Untuk fungsi tujuan minimasi penyimpangan *part consumption rate*, nilai minimal yang dihasilkan GrEA adalah 36149.29 dan nilai maksimal sebesar 38202.61, dengan rata-rata 37131.57. Seluruh nilai penyimpangan *part consumption rate* dari solusi optimasi lebih rendah dibandingkan nilai solusi aktual. Untuk fungsi tujuan maksimasi densifikasi warna, nilai minimal yang dihasilkan GrEA adalah 0.359 dan nilai maksimal sebesar 0.370, dengan rata-rata 0.366. Seluruh nilai densifikasi warna dari solusi optimasi lebih besar dibanding nilai solusi aktual.

e. 8 Juni 2019

Pada 8 Juni 2019, terdapat 584 unit yang akan diproduksi, dengan 70 jenis *suffix*, 2 model, 8 *grade* mobil, 4 jenis mesin dan 7 warna berbeda. Fungsi tujuan dari solusi aktual untuk minimasi penyimpangan *part consumption rate* dan maksimasi densifikasi warna berturut-turut adalah 153231.6 dan 0.287. Optimasi menggunakan GrEA berhenti setelah 699 iterasi dengan waktu komputasi 28.23 jam dan dihasilkan 21 alternatif solusi berbeda. Untuk fungsi tujuan minimasi penyimpangan *part consumption rate*, nilai minimal yang dihasilkan GrEA adalah 38283.71 dan nilai maksimal sebesar 53878.69, dengan rata-rata 45929.95. Seluruh nilai penyimpangan *part consumption rate* dari solusi optimasi lebih rendah dibandingkan nilai solusi aktual. Untuk fungsi tujuan maksimasi densifikasi warna, nilai minimal yang dihasilkan GrEA adalah 0.303 dan nilai maksimal sebesar 0.357, dengan rata-rata 0.327. Seluruh nilai densifikasi warna dari solusi optimasi lebih besar dibanding nilai solusi aktual.

f. 20 Juni 2019

Pada 20 Juni 2019, terdapat 455 unit yang akan diproduksi, dengan 64 jenis *suffix*, 2 model, 8 *grade* mobil, 4 jenis mesin dan 7 warna berbeda. Fungsi tujuan dari solusi aktual untuk minimasi penyimpangan *part consumption rate* dan maksimasi densifikasi warna berturut-turut adalah 74150.35 dan 0.276. Optimasi menggunakan GrEA berhenti setelah 2399 iterasi dengan waktu

komputasi 96.88 jam dan dihasilkan 16 alternatif solusi berbeda. Untuk fungsi tujuan minimasi penyimpangan *part consumption rate*, nilai minimal yang dihasilkan GrEA adalah 24922.65 dan nilai maksimal sebesar 29746.73, dengan rata-rata 27360.32. Seluruh nilai penyimpangan *part consumption rate* dari solusi optimasi lebih rendah dibandingkan nilai solusi aktual. Untuk fungsi tujuan maksimasi densifikasi warna, nilai minimal yang dihasilkan GrEA adalah 0.321 dan nilai maksimal sebesar 0.366, dengan rata-rata 0.343. Seluruh nilai densifikasi warna dari solusi optimasi lebih besar dibanding nilai solusi aktual.

g. 21 Juni 2019

Pada 21 Juni 2019, terdapat 588 unit yang akan diproduksi, dengan 76 jenis *suffix*, 2 model, 8 *grade* mobil, 4 jenis mesin dan 8 warna berbeda. Fungsi tujuan dari solusi aktual untuk minimasi penyimpangan *part consumption rate* dan maksimasi densifikasi warna berturut-turut adalah 162519.9 dan 0.261. Optimasi menggunakan GrEA berhenti setelah 935 iterasi dengan waktu komputasi 37.76 jam dan dihasilkan 14 alternatif solusi berbeda. Untuk fungsi tujuan minimasi penyimpangan *part consumption rate*, nilai minimal yang dihasilkan GrEA adalah 49229.44 dan nilai maksimal sebesar 65725.54, dengan rata-rata 57844.24. Seluruh nilai penyimpangan *part consumption rate* dari solusi optimasi lebih rendah dibandingkan nilai solusi aktual. Untuk fungsi tujuan maksimasi densifikasi warna, nilai minimal yang dihasilkan GrEA adalah 0.334 dan nilai maksimal sebesar 0.359, dengan rata-rata 0.345. Seluruh nilai densifikasi warna dari solusi optimasi lebih besar dibanding nilai solusi aktual.

h. 22 Juni 2019

Pada 22 Juni 2019, terdapat 622 unit yang akan diproduksi, dengan 70 jenis *suffix*, 2 model, 8 *grade* mobil, 4 jenis mesin dan 7 warna berbeda. Fungsi tujuan dari solusi aktual untuk minimasi penyimpangan *part consumption rate* dan maksimasi densifikasi warna berturut-turut adalah 114757.7 dan 0.293. Optimasi menggunakan GrEA berhenti setelah 3579 iterasi dengan waktu komputasi 164.415 jam dan dihasilkan 14 alternatif solusi berbeda. Untuk fungsi tujuan minimasi penyimpangan *part consumption rate*, nilai minimal yang dihasilkan GrEA adalah 49796.95 dan nilai maksimal sebesar 55370.16, dengan

rata-rata 52087.27. Seluruh nilai penyimpangan *part consumption rate* dari solusi optimasi lebih rendah dibandingkan nilai solusi aktual. Untuk fungsi tujuan maksimasi densifikasi warna, nilai minimal yang dihasilkan GrEA adalah 0.418 dan nilai maksimal sebesar 0.430, dengan rata-rata 0.425. Seluruh nilai densifikasi warna dari solusi optimasi lebih besar dibanding nilai solusi aktual.

i. 25 Juni 2019

Pada 25 Juni 2019, terdapat 522 unit yang akan diproduksi, dengan 67 jenis *suffix*, 2 model, 8 *grade* mobil, 4 jenis mesin dan 8 warna berbeda. Fungsi tujuan dari solusi aktual untuk minimasi penyimpangan *part consumption rate* dan maksimasi densifikasi warna berturut-turut adalah 148341.2 dan 0.282. Optimasi menggunakan GrEA berhenti setelah 2416 iterasi dengan waktu komputasi 97.57 jam dan dihasilkan 14 alternatif solusi berbeda. Untuk fungsi tujuan minimasi penyimpangan *part consumption rate*, nilai minimal yang dihasilkan GrEA adalah 34798.82 dan nilai maksimal sebesar 39351.80, dengan rata-rata 37129.39. Seluruh nilai penyimpangan *part consumption rate* dari solusi optimasi lebih rendah dibandingkan nilai solusi aktual. Untuk fungsi tujuan maksimasi densifikasi warna, nilai minimal yang dihasilkan GrEA adalah 0.336 dan nilai maksimal sebesar 0.383, dengan rata-rata 0.358. Seluruh nilai densifikasi warna dari solusi optimasi lebih besar dibanding nilai solusi aktual.

j. 26 Juni 2019

Pada 26 Juni 2019, terdapat 697 unit yang akan diproduksi, dengan 65 jenis *suffix*, 2 model, 8 *grade* mobil, 4 jenis mesin dan 7 warna berbeda. Fungsi tujuan dari solusi aktual untuk minimasi penyimpangan *part consumption rate* dan maksimasi densifikasi warna berturut-turut adalah 79153.260 dan 0.304. Optimasi menggunakan GrEA berhenti setelah 1300 iterasi dengan waktu komputasi 74.86 jam dan dihasilkan 14 alternatif solusi berbeda. Untuk fungsi tujuan minimasi penyimpangan *part consumption rate*, nilai minimal yang dihasilkan GrEA adalah 51127.41 dan nilai maksimal sebesar 58894.08, dengan rata-rata 54639.5. Seluruh nilai penyimpangan *part consumption rate* dari solusi optimasi lebih rendah dibandingkan nilai solusi aktual. Untuk fungsi tujuan maksimasi densifikasi warna, nilai minimal yang dihasilkan GrEA adalah 0.353

dan nilai maksimal sebesar 0.373, dengan rata-rata 0.363. Seluruh nilai densifikasi warna dari solusi optimasi lebih besar dibanding nilai solusi aktual.

5.2. Perbandingan Persebaran *Part Consumption Rate*

Persebaran atau distribusi *part consumption rate* menunjukkan seberapa tingkat penggunaan part dalam periode waktu tertentu. Fungsi tujuan yang digunakan memiliki tujuan untuk meminimasi nilai kuadrat dari selisih tingkat penggunaan ideal dengan tingkat penggunaan aktual. Tingkat penggunaan ideal menunjukkan pada unit pertama hingga unit ke-P, berapa part yang dibutuhkan atau digunakan. Hal ini berkaitan dengan tingkat persediaan dan kemampuan *supplier* untuk memasok material atau part tersebut sesuai dengan kebutuhan setiap jamnya. Pada PT XYZ, produk mobil yang dihasilkan dapat memiliki lebih dari 1000 part, maka pada penelitian ini part-part diwakilkan atau dikelompokkan sebagai jenis atau varian pada *sub-assembly*. Misalnya, part-part yang dibutuhkan untuk *sub-assembly* mesin dengan bahan bakar diesel akan berbeda dengan part yang dibutuhkan untuk *sub-assembly* mesin dengan bahan bakar bensin/petrol. Hal ini sesuai dengan saran dari Monden (2012) yang menyatakan bahwa part-part dapat diwakilkan oleh *sub-assembly* yang terkait, karena jumlah part-part yang berbeda dalam satu unit mobil dapat mencapai 20.000 jenis.

Persebaran *sub-assembly* erat kaitannya dengan tingkat persediaan dan aliran produksi dari proses penyedia part, yakni *supplier*. Jadwal produksi yang tidak konstan pada tingkat permintaan harian atau mingguan menyebabkan perusahaan tidak dapat merespon perubahan permintaan pelanggan dengan fleksibel. Sedangkan pada tingkat produksi harian, perusahaan perlu menyusun jadwal urutan produksi yang dapat menghasilkan penggunaan *part* atau *sub-assembly* yang konstan. Tingkat penggunaan *part* yang tidak konstan dapat menyebabkan *waste* atau pemborosan pada persediaan. Hal ini disebabkan karena *supplier* mengirimkan part beberapa kali dalam sehari sesuai *kanban* yang dikirim perusahaan. PT XYZ memiliki ratusan *supplier* dengan skema jadwal pengiriman part yang berbeda-beda, sehingga untuk penyederhanaan, analisis dilakukan dengan asumsi *supplier* merespon *kanban* dalam periode per 1 jam, sesuai dengan Monden (2012) yang membagi seluruh unit menjadi 16 periode (1 periode setara 1 jam). Pada penelitian ini, unit dibagi menjadi 14 periode karena waktu kerja efektif

produksi adalah 7.5 jam pada *shift* pertama dan 6.5 jam pada *shift* kedua. Ketika jumlah mobil dengan *sub-assembly* tertentu pada satu jam tertentu lebih besar dibandingkan tingkat penggunaan part ideal dari *sub-assembly* tersebut, maka terjadi *shortage* atau kelangkaan part. Pada perusahaan yang tidak memiliki aliran produksi yang konstan, hal ini disiasati dengan menyimpan persediaan part lebih besar dibandingkan yang dibutuhkan, maka terjadi *waste* persediaan. Dengan tingkat penggunaan part yang konstan, *safety stock* dapat diminimasi karena kebutuhan part setiap jamnya mendekati kebutuhan ideal. Tingkat penggunaan part yang konstan setiap jamnya juga turut melancarkan produksi part oleh *supplier*. Karena PT XYZ menerapkan sistem *kanban*, maka part “ditarik” dalam jumlah tertentu dari supplier sesuai perintah *kanban*. Ketika jumlah part yang ditarik tidak seragam setiap jamnya, maka terjadi fluktuasi permintaan pada sisi *supplier*, sehingga part yang diproduksi oleh *supplier* bervariasi dari jam ke jam. Hal ini dapat menyebabkan aliran produksi *supplier* menjadi tidak lancar, terjadi *waste* persediaan material dan *waste* tenaga kerja (Monden, 2012). Persebaran produk per jam dapat diukur sebagai standar deviasi. Nilai standar deviasi penggunaan part tiap jamnya dapat digunakan untuk mengukur apakah jadwal urutan produksi yang dipakai sudah konstan atau belum. Nilai standar deviasi yang rendah menunjukkan bahwa tingkat penggunaan part konstan, karena jumlah yang diproduksi setiap jamnya hampir sama. Berikut uraian tingkat penggunaan part pada setiap jadwal produksi:

a. 4 Juni 2018

Pada 4 Juni 2018, terdapat 70 varian *suffix*, dengan rata-rata dari persebaran setiap *suffix*-nya sebesar 0.536 untuk kondisi aktual. Seluruh selisih data aktual dengan hasil optimasi berdisitribusi tidak normal, ditunjukkan dengan nilai *p-value* dari uji *shapiro-wilk* kurang dari 0.05, sehingga uji perbedaan rata-rata yang dipakai adalah uji Wilcoxon. Tidak ditemukan perbedaan signifikan antara rata-rata standar deviasi kondisi aktual dengan solusi-solusi hasil optimasi.

Terdapat 8 varian *grade*, dengan rata-rata dari persebaran setiap varian adalah 1.413 untuk kondisi aktual. Seluruh selisih data aktual dengan hasil optimasi berdistribusi normal, sehingga digunakan uji t berpasangan. Hasil uji perbedaan rata-rata menunjukkan tidak terdapat perbedaan signifikan antara

kondisi aktual dengan sebagian besar solusi-solusi hasil optimasi. Hanya terdapat 1 solusi yang memiliki perbedaan signifikan dengan nilai standar deviasi optimasi lebih rendah dibandingkan kondisi aktual.

Terdapat 9 varian warna, dengan rata-rata dari standar deviasi setiap varian adalah 1.583 untuk kondisi aktual. Seluruh selisih data aktual dengan hasil optimasi berdistribusi normal, sehingga digunakan uji t berpasangan. Hasil uji perbedaan rata-rata menunjukkan tidak terdapat perbedaan signifikan antara kondisi aktual dengan solusi-solusi hasil optimasi.

Terdapat 4 varian mesin yang diproduksi. Karena jumlah varian terlalu sedikit untuk dijadikan sampel, maka uji statistik tidak dapat dilakukan. Seluruh mesin varian *HS*, *KS*, dan *YM* dari solusi optimasi memiliki standar deviasi lebih rendah dibandingkan kondisi aktual. Sebagian mesin varian *KM* memiliki standar deviasi yang lebih rendah dibandingkan kondisi aktual. Berdasarkan rata-rata persebaran dari *sub-assembly* mesin, sebagian besar rata-rata persebaran hasil solusi optimasi lebih rendah dibandingkan kondisi aktual.

Terdapat 2 varian model yang diproduksi PT XYZ, yakni model M dan model S. Standar deviasi varian model S dan M dari kondisi aktual lebih rendah dari seluruh standar deviasi dari solusi-solusi hasil optimasi. Kondisi aktual memiliki standar deviasi yang rendah (di bawah 1) sedangkan sebagian besar solusi-solusi aktual memiliki standar deviasi lebih dari 1. Hal ini disebabkan karena standar deviasi diukur berdasarkan jumlah unit dengan model tertentu setiap jamnya, sedangkan fungsi tujuan *part consumption rate* menghitung kuadrat dari selisih antara tingkat penggunaan part aktual dengan tingkat penggunaan part ideal. Nilai standar deviasi bergantung pada berapa jumlah jam kerja yang akan dipakai (pada penelitian ini 14), sedangkan fungsi tujuan tidak memandang pembagian berdasarkan jam kerja. Pada perhitungan standar deviasi, tidak masalah apabila penyimpangan terhadap tingkat penggunaan ideal besar, asalkan penyimpangan atau perbedaan antar jam tidak terlalu besar, sedangkan fungsi tujuan mengukur dengan akurat nilai selisih untuk setiap posisi dari unit pertama hingga ke unit ke-P. Tabel 5.1 menjabarkan rata-rata dari

selisih tingkat penggunaan part aktual dengan ideal untuk jadwal produksi 4 Juni 2018.

Tabel 5.1 Rata-Rata Dari Selisih Tingkat Penggunaan Part Aktual Dengan Ideal

Solusi	Rata-rata gap absolut ideal dan aktual		Standar deviasi gap absolut		Solusi	Rata-rata gap absolut ideal dan aktual		Standar deviasi gap absolut	
	M	S	M	S		M	S	M	S
aktual	1.37	1.37	0.833	0.833	aktual	1.37	1.37	0.833	0.833
1	0.757	0.757	0.558	0.558	10	0.714	0.714	0.505	0.505
2	0.773	0.773	0.560	0.560	11	0.704	0.704	0.505	0.505
3	0.675	0.675	0.487	0.487	12	0.872	0.872	0.578	0.578
4	0.673	0.673	0.485	0.485	13	0.679	0.679	0.484	0.484
5	0.706	0.706	0.507	0.507	14	0.680	0.680	0.481	0.481
6	0.672	0.672	0.484	0.484	15	0.685	0.685	0.484	0.484
7	0.691	0.691	0.479	0.479	16	0.682	0.682	0.485	0.485
8	0.755	0.755	0.555	0.555	17	0.955	0.955	0.630	0.630
9	0.675	0.675	0.482	0.482	18	0.726	0.726	0.545	0.545

Apabila dilihat dari rata-rata nilai selisih tingkat penggunaan ideal dengan aktual, kondisi aktual memiliki rata-rata selisih sebesar 1.37 untuk model M dan model S, sedangkan untuk seluruh hasil optimasi rata-rata selisihnya lebih kecil. Ini menunjukkan bahwa fungsi tujuan yang rendah tidak serta merta berarti standar deviasi penggunaan setiap jamnya juga rendah. Berdasarkan rata-rata persebaran dari *sub-assembly* model, sebagian besar rata-rata persebaran hasil solusi optimasi lebih tinggi dibandingkan kondisi aktual.

b. 5 Juni 2018

Pada 5 Juni 2018, terdapat 74 varian *suffix*, dengan rata-rata dari standar deviasi setiap *suffix*nya sebesar 0.526 untuk kondisi aktual. Seluruh selisih data aktual dengan hasil optimasi berdistribusi tidak normal, sehingga uji perbedaan rata-rata yang dipakai adalah uji Wilcoxon. Sebagian besar tidak ditemukan perbedaan signifikan antara rata-rata standar deviasi kondisi aktual dengan solusi-solusi hasil optimasi. Terdapat 1 solusi optimasi yang secara signifikan lebih baik dibandingkan kondisi aktual.

Terdapat 8 varian *grade*, dengan rata-rata dari standar deviasi setiap varian adalah 1.5989 untuk kondisi aktual. Seluruh selisih data aktual dengan hasil optimasi berdistribusi normal, sehingga digunakan uji t berpasangan. Terdapat 10 solusi yang hasil ujinya menunjukkan tidak terdapat perbedaan signifikan

antara kondisi aktual dengan solusi-solusi hasil optimasi. Terdapat 8 solusi yang memiliki perbedaan signifikan dengan nilai standar deviasi optimasi lebih rendah dibandingkan kondisi aktual, dengan nilai solusi optimasi lebih rendah.

Terdapat 8 varian warna, dengan rata-rata dari standar deviasi setiap varian adalah 1.6457 untuk kondisi aktual. Seluruh selisih data aktual dengan hasil optimasi berdistribusi normal, sehingga digunakan uji t berpasangan. Hasil uji perbedaan rata-rata menunjukkan tidak terdapat perbedaan signifikan antara kondisi aktual dengan solusi-solusi hasil optimasi.

Terdapat 4 varian mesin yang diproduksi. Seluruh mesin varian *KM* dari solusi optimasi memiliki standar deviasi lebih rendah dibandingkan kondisi aktual, sedangkan seluruh varian *HS* dan *KS* dari solusi optimasi lebih besar dibandingkan kondisi aktual. Sebagian mesin varian *YM* memiliki standar deviasi yang lebih tinggi dibandingkan kondisi aktual. Berdasarkan rata-rata persebaran dari *sub-assembly* mesin, sebagian besar rata-rata persebaran hasil solusi optimasi lebih rendah dibandingkan kondisi aktual.

Terdapat 2 varian model yang diproduksi PT XYZ, yakni model M dan model S. Standar deviasi varian model S dan M dari kondisi aktual lebih rendah dari seluruh standar deviasi dari solusi-solusi hasil optimasi. Kondisi aktual memiliki standar deviasi yang rendah (di sekitar 1) sedangkan sebagian besar solusi-solusi aktual memiliki standar deviasi lebih dari 1. Berdasarkan rata-rata persebaran dari *sub-assembly* model, sebagian besar rata-rata persebaran hasil solusi optimasi lebih tinggi dibandingkan kondisi aktual.

c. 6 Juni 2018

Pada 6 Juni 2018, terdapat 69 varian *suffix*, dengan rata-rata dari standar deviasi setiap *suffix*-nya sebesar 0.558 untuk kondisi aktual. Seluruh selisih data aktual dengan hasil optimasi tidak berdistribusi normal, sehingga uji perbedaan rata-rata yang dipakai adalah uji Wilcoxon. Sebagian besar tidak ditemukan perbedaan signifikan antara rata-rata standar deviasi kondisi aktual dengan solusi-solusi hasil optimasi. Terdapat 1 solusi optimasi yang secara signifikan lebih baik dibandingkan kondisi aktual.

Terdapat 8 varian *grade*, dengan rata-rata dari standar deviasi setiap varian adalah 1.3605 untuk kondisi aktual. Seluruh selisih data aktual dengan hasil optimasi berdistribusi normal, sehingga digunakan uji t berpasangan. Untuk seluruh uji antara kondisi aktual dengan solusi-solusi hasil optimasi, tidak terdapat perbedaan yang signifikan terhadap nilai rata-rata dari standar deviasi.

Terdapat 8 varian warna, dengan rata-rata dari standar deviasi setiap varian adalah 1.9152 untuk kondisi aktual. Seluruh selisih data aktual dengan hasil optimasi berdistribusi normal, sehingga digunakan uji t berpasangan. Hasil uji perbedaan rata-rata menunjukkan tidak terdapat perbedaan signifikan antara kondisi aktual dengan solusi-solusi hasil optimasi.

Terdapat 4 varian mesin yang diproduksi. Seluruh mesin varian *HS* dan *KS* dari solusi optimasi memiliki standar deviasi lebih rendah dibandingkan kondisi aktual, sedangkan hampir seluruh varian *YM* dari solusi optimasi lebih besar dibandingkan kondisi aktual. Sebagian mesin varian *KM* memiliki standar deviasi yang lebih tinggi dibandingkan kondisi aktual. Berdasarkan rata-rata persebaran dari *sub-assembly* mesin, sebagian besar rata-rata persebaran hasil solusi optimasi lebih rendah dibandingkan kondisi aktual.

Terdapat 2 varian model yang diproduksi PT XYZ, yakni model M dan model S. Standar deviasi varian model S dan M dari kondisi aktual lebih rendah dari hampir seluruh standar deviasi dari solusi-solusi hasil optimasi. Kondisi aktual memiliki standar deviasi yang rendah (di sekitar 1) sedangkan sebagian besar solusi-solusi aktual memiliki standar deviasi lebih dari 1. Berdasarkan rata-rata persebaran dari *sub-assembly* tinggi, sebagian besar rata-rata persebaran hasil solusi optimasi lebih tinggi dibandingkan kondisi aktual.

d. 7 Juni 2018

Pada 7 Juni 2018, terdapat 69 varian *suffix*, dengan rata-rata dari standar deviasi setiap *suffix*-nya sebesar 0.595 untuk kondisi aktual. Seluruh selisih data aktual dengan hasil optimasi tidak berdistribusi normal, sehingga uji perbedaan rata-rata yang dipakai adalah uji Wilcoxon. Seluruh hasil uji menunjukkan terdapat perbedaan signifikan antara rata-rata standar deviasi kondisi aktual dengan solusi-

solusi hasil optimasi, dengan solusi optimasi lebih rendah dibandingkan solusi aktual.

Terdapat 8 varian *grade*, dengan rata-rata dari standar deviasi setiap varian adalah 1.5676 untuk kondisi aktual. Seluruh selisih data aktual dengan hasil optimasi berdistribusi normal, sehingga digunakan uji t berpasangan. Terdapat 2 solusi hasil optimasi yang secara signifikan lebih rendah dibandingkan dari kondisi aktual, sedangkan sisanya memiliki rata-rata standar deviasi yang tidak berbeda secara signifikan.

Terdapat 8 varian warna, dengan rata-rata dari standar deviasi setiap varian adalah 2.0811 untuk kondisi aktual. Seluruh selisih data aktual dengan hasil optimasi berdistribusi normal, sehingga digunakan uji t berpasangan. Hasil uji perbedaan rata-rata menunjukkan tidak terdapat perbedaan signifikan antara kondisi aktual dengan solusi-solusi hasil optimasi.

Terdapat 4 varian mesin yang diproduksi. Seluruh mesin varian *KM*, *KS* dan *YM* dari solusi optimasi memiliki standar deviasi lebih rendah dibandingkan kondisi aktual, sedangkan seluruh varian *HS* dari solusi optimasi lebih besar dibandingkan kondisi aktual. Berdasarkan rata-rata persebaran dari *sub-assembly* mesin, sebagian besar rata-rata persebaran hasil solusi optimasi lebih rendah dibandingkan kondisi aktual.

Terdapat 2 varian model yang diproduksi PT XYZ, yakni model M dan model S. Standar deviasi varian model M pada kondisi aktual lebih rendah dari seluruh standar deviasi dari solusi-solusi hasil optimasi, sedangkan untuk model S seluruhnya lebih tinggi dibandingkan kondisi aktual. Berdasarkan rata-rata persebaran dari *sub-assembly* model, sebagian besar rata-rata persebaran hasil solusi optimasi lebih rendah dibandingkan kondisi aktual.

e. 8 Juni 2018

Pada 8 Juni 2018, terdapat 70 varian *suffix*, dengan rata-rata dari standar deviasi setiap *suffix*-nya sebesar 0.571 untuk kondisi aktual. Seluruh selisih data aktual dengan hasil optimasi tidak berdistribusi normal, sehingga uji perbedaan rata-rata yang dipakai adalah uji Wilcoxon. Seluruh hasil uji menunjukkan tidak

ditemukan perbedaan signifikan antara rata-rata standar deviasi kondisi aktual dengan solusi-solusi hasil optimasi.

Terdapat 8 varian *grade*, dengan rata-rata dari standar deviasi setiap varian adalah 1.6678 untuk kondisi aktual. Seluruh selisih data aktual dengan hasil optimasi berdistribusi normal, sehingga digunakan uji t berpasangan. Untuk seluruh uji antara kondisi aktual dengan solusi-solusi hasil optimasi, tidak terdapat perbedaan yang signifikan terhadap nilai rata-rata dari standar deviasi.

Terdapat 7 varian warna, dengan rata-rata dari standar deviasi setiap varian adalah 2.1873 untuk kondisi aktual. Seluruh selisih data aktual dengan hasil optimasi berdistribusi normal, sehingga digunakan uji t berpasangan. Hasil uji perbedaan rata-rata menunjukkan tidak terdapat perbedaan signifikan antara kondisi aktual dengan solusi-solusi hasil optimasi.

Terdapat 4 varian mesin yang diproduksi. Seluruh mesin varian *KS* dari solusi optimasi memiliki standar deviasi lebih rendah dibandingkan kondisi aktual, sedangkan hampir seluruh varian *HS* dan *YM* dari solusi optimasi lebih besar dibandingkan kondisi aktual. Sebagian besar mesin varian *KM* memiliki standar deviasi yang lebih tinggi dibandingkan kondisi aktual. Berdasarkan rata-rata persebaran dari *sub-assembly* mesin, sebagian besar rata-rata persebaran hasil solusi optimasi lebih rendah dibandingkan kondisi aktual.

Terdapat 2 varian model yang diproduksi PT XYZ, yakni model M dan model S. Standar deviasi varian model S dan M dari kondisi aktual lebih rendah dari hampir seluruh standar deviasi dari solusi-solusi hasil optimasi. Kondisi aktual memiliki standar deviasi yang rendah dan solusi-solusi aktual memiliki standar deviasi lebih dari 1. Berdasarkan rata-rata persebaran dari *sub-assembly* model, sebagian besar rata-rata persebaran hasil solusi optimasi lebih tinggi dibandingkan kondisi aktual.

f. 20 Juni 2018

Pada 20 Juni 2018, terdapat 64 varian *suffix*, dengan rata-rata dari standar deviasi setiap *suffix*-nya sebesar 0.552 untuk kondisi aktual. Seluruh selisih data aktual dengan hasil optimasi tidak berdistribusi normal, sehingga uji perbedaan rata-rata yang dipakai adalah uji Wilcoxon. Seluruh hasil perbandingan

menunjukkan tidak ditemukan perbedaan signifikan antara rata-rata standar deviasi kondisi aktual dengan solusi-solusi hasil optimasi. Secara keseluruhan, standar deviasi dari seluruh varian *suffix* terbilang kecil (rata-rata di sekitar 0.5) sehingga dapat kita anggap penggunaan *sub-assembly* konstan pada setiap jam produksi.

Terdapat 8 varian *grade*, dengan rata-rata dari standar deviasi setiap varian adalah 1.3151 untuk kondisi aktual. Seluruh selisih data aktual dengan hasil optimasi berdistribusi normal, sehingga digunakan uji t berpasangan. Terdapat 6 solusi yang memiliki rata-rata dari standar deviasi yang secara signifikan lebih rendah dibandingkan kondisi aktual, sedangkan sisanya tidak memiliki perbedaan yang signifikan terhadap nilai rata-rata dari standar deviasi.

Terdapat 7 varian warna, dengan rata-rata dari standar deviasi setiap varian adalah 2.082 untuk kondisi aktual. Seluruh selisih data aktual dengan hasil optimasi berdistribusi normal, sehingga digunakan uji t berpasangan. Hasil uji perbedaan rata-rata menunjukkan tidak terdapat perbedaan signifikan antara kondisi aktual dengan solusi-solusi hasil optimasi.

Terdapat 4 varian mesin yang diproduksi. Seluruh mesin varian *HS*, *KS* dan *YM* dari solusi optimasi memiliki standar deviasi lebih rendah dibandingkan kondisi aktual, sedangkan hampir seluruh varian *YM* dari solusi optimasi lebih rendah dibandingkan kondisi aktual. Berdasarkan rata-rata persebaran dari *sub-assembly* mesin, sebagian besar rata-rata persebaran hasil solusi optimasi lebih rendah dibandingkan kondisi aktual.

Terdapat 2 varian model yang diproduksi PT XYZ, yakni model M dan model S. Standar deviasi varian model M dari kondisi aktual lebih besar dari hampir seluruh standar deviasi dari solusi-solusi hasil optimasi, sedangkan pada model S hanya sebagian solusi optimasi yang memiliki rata-rata lebih rendah. Berdasarkan rata-rata persebaran dari *sub-assembly* model, sebagian besar rata-rata persebaran hasil solusi optimasi lebih tinggi dibandingkan kondisi aktual.

g. 21 Juni 2018

Pada 21 Juni 2018, terdapat 76 varian *suffix*, dengan rata-rata dari standar deviasi setiap *suffix*nya sebesar 0.522 untuk kondisi aktual. Seluruh selisih data aktual dengan hasil optimasi tidak berdistribusi normal, sehingga uji perbedaan

rataan yang dipakai adalah uji Wilcoxon. Sebagian solusi optimasi memiliki rata-rata dari standar deviasi yang secara signifikan lebih tinggi dibandingkan kondisi aktual.

Terdapat 8 varian *grade*, dengan rata-rata dari standar deviasi setiap varian adalah 1.3605 untuk kondisi aktual. Seluruh selisih data aktual dengan hasil optimasi berdistribusi normal, sehingga digunakan uji t berpasangan. Sebagian besar hasil uji antara kondisi aktual dengan solusi-solusi hasil optimasi menunjukkan tidak terdapat perbedaan yang signifikan terhadap nilai rata-rata dari standar deviasi. Hanya terdapat 1 solusi optimasi yang nilainya lebih tinggi secara signifikan dibandingkan kondisi aktual.

Terdapat 8 varian warna, dengan rata-rata dari standar deviasi setiap varian adalah 1.9243 untuk kondisi aktual. Seluruh selisih data aktual dengan hasil optimasi berdistribusi normal, sehingga digunakan uji t berpasangan. Hasil uji perbedaan rata-rata menunjukkan tidak terdapat perbedaan signifikan antara kondisi aktual dengan solusi-solusi hasil optimasi.

Terdapat 4 varian mesin yang diproduksi. Sebagian mesin varian *HS* dan *KS* dari solusi optimasi memiliki standar deviasi yang lebih tinggi dibandingkan kondisi aktual. Sedangkan untuk varian *KM* dan *YM*, seluruh standar deviasi solusi optimasi lebih tinggi dibandingkan aktual. Berdasarkan rata-rata persebaran dari *sub-assembly* mesin, sebagian besar rata-rata persebaran hasil solusi optimasi lebih tinggi dibandingkan kondisi aktual.

Terdapat 2 varian model yang diproduksi PT XYZ, yakni model M dan model S. Standar deviasi varian model S dan M dari kondisi aktual lebih rendah dari hampir seluruh standar deviasi dari solusi-solusi hasil optimasi. Berdasarkan rata-rata persebaran dari *sub-assembly* model, sebagian besar rata-rata persebaran hasil solusi optimasi lebih tinggi dibandingkan kondisi aktual.

h. 22 Juni 2018

Pada 22 Juni 2018, terdapat 70 varian *suffix*, dengan rata-rata dari standar deviasi setiap *suffix*-nya sebesar 0.590 untuk kondisi aktual. Seluruh selisih data aktual dengan hasil optimasi tidak berdistribusi normal, sehingga uji perbedaan rata-rata yang dipakai adalah uji Wilcoxon. Seluruh solusi optimal tidak memiliki

perbedaan signifikan antara rata-rata standar deviasi kondisi aktual dengan solusi-solusi hasil optimasi.

Terdapat 7 varian *grade*, dengan rata-rata dari standar deviasi setiap varian adalah 1.5054 untuk kondisi aktual. Terdapat 5 solusi yang menggunakan uji wilcoxon, sedangkan sisanya menggunakan uji t berpasangan. Untuk seluruh uji antara kondisi aktual dengan solusi-solusi hasil optimasi, tidak terdapat perbedaan yang signifikan terhadap nilai rata-rata dari standar deviasi.

Terdapat 8 varian warna, dengan rata-rata dari standar deviasi setiap varian adalah 1.89697 untuk kondisi aktual. Seluruh selisih data aktual dengan hasil optimasi berdistribusi normal, sehingga digunakan uji t berpasangan. Hasil uji perbedaan rata-rata menunjukkan tidak terdapat perbedaan signifikan antara kondisi aktual dengan solusi-solusi hasil optimasi.

Terdapat 4 varian mesin yang diproduksi. Seluruh mesin varian *HS* dan *KS* dari solusi optimasi memiliki standar deviasi lebih rendah dibandingkan kondisi aktual, sedangkan sebagian varian *KM* dan *YM* dari solusi optimasi lebih besar dibandingkan kondisi aktual. Berdasarkan rata-rata persebaran dari *sub-assembly* mesin, sebagian besar rata-rata persebaran hasil solusi optimasi lebih tinggi dibandingkan kondisi aktual.

Terdapat 2 varian model yang diproduksi PT XYZ, yakni model M dan model S. Standar deviasi varian model M dari kondisi aktual lebih tinggi dari hampir seluruh standar deviasi dari solusi-solusi hasil optimasi, sedangkan untuk model S sebagian besar solusi optimasi memiliki standar deviasi yang lebih rendah. Berdasarkan rata-rata persebaran dari *sub-assembly* model, sebagian besar rata-rata persebaran hasil solusi optimasi lebih rendah dibandingkan kondisi aktual.

i. 25 Juni 2018

Pada 25 Juni 2018, terdapat 67 varian *suffix*, dengan rata-rata dari standar deviasi setiap *suffix*-nya sebesar 0.540 untuk kondisi aktual. Seluruh selisih data aktual dengan hasil optimasi tidak berdistribusi normal, sehingga uji perbedaan rata-rata yang dipakai adalah uji Wilcoxon. Semua hasil uji menunjukkan tidak ditemukan perbedaan signifikan antara rata-rata standar deviasi kondisi aktual

dengan solusi-solusi hasil optimasi. Secara keseluruhan, standar deviasi dari seluruh varian *suffix* terbilang kecil (rata-rata di sekitar 0.5) sehingga dapat kita anggap penggunaan *sub-assembly* konstan pada setiap jam produksi.

Terdapat 8 varian *grade*, dengan rata-rata dari standar deviasi setiap varian adalah 1.6421 untuk kondisi aktual. Seluruh selisih data aktual dengan hasil optimasi berdistribusi normal, sehingga digunakan uji t berpasangan. Hanya terdapat 2 solusi optimasi yang tidak berbeda secara signifikan dengan kondisi aktual. Sedangkan sisanya menunjukkan bahwa solusi optimasi menghasilkan rata-rata dari standar deviasi yang secara signifikan lebih rendah dibandingkan kondisi aktual.

Terdapat 8 varian warna, dengan rata-rata dari standar deviasi setiap varian adalah 1.8605 untuk kondisi aktual. Seluruh selisih data aktual dengan hasil optimasi berdistribusi normal, sehingga digunakan uji t berpasangan. Hasil uji perbedaan rata-rata menunjukkan tidak terdapat perbedaan signifikan antara kondisi aktual dengan solusi-solusi hasil optimasi.

Terdapat 4 varian mesin yang diproduksi. Seluruh mesin varian *HS* dan *KS* dari solusi optimasi memiliki standar deviasi lebih rendah dibandingkan kondisi aktual, sedangkan hampir seluruh varian *KM* dari solusi optimasi lebih besar dibandingkan kondisi aktual. Sebagian mesin varian *YM* memiliki standar deviasi yang lebih tinggi dibandingkan kondisi aktual. Berdasarkan rata-rata persebaran dari *sub-assembly* mesin, sebagian besar rata-rata persebaran hasil solusi optimasi lebih rendah dibandingkan kondisi aktual.

Terdapat 2 varian model yang diproduksi PT XYZ, yakni model M dan model S. Standar deviasi varian model S dan M dari kondisi aktual lebih rendah dari hampir seluruh standar deviasi dari solusi-solusi hasil optimasi. Kondisi aktual memiliki standar deviasi yang rendah (di bawah 1) sedangkan sebagian besar solusi-solusi aktual memiliki standar deviasi lebih dari 1. Berdasarkan rata-rata persebaran dari *sub-assembly* model, sebagian besar rata-rata persebaran hasil solusi optimasi lebih tinggi dibandingkan kondisi aktual.

j. 26 Juni 2018

Pada 26 Juni 2018, terdapat 65 varian *suffix*, dengan rata-rata dari standar deviasi setiap *suffix*nya sebesar 0.543 untuk kondisi aktual. Seluruh selisih data aktual dengan hasil optimasi tidak berdistribusi normal, sehingga uji perbedaan rata-rata yang dipakai adalah uji Wilcoxon. Seluruh hasil uji menunjukkan tidak ditemukan perbedaan signifikan antara rata-rata standar deviasi kondisi aktual dengan solusi-solusi hasil optimasi. Secara keseluruhan, standar deviasi dari seluruh varian *suffix* terbilang kecil (rata-rata di sekitar 0.5) sehingga dapat kita anggap penggunaan *sub-assembly* konstan pada setiap jam produksi.

Terdapat 8 varian *grade*, dengan rata-rata dari standar deviasi setiap varian adalah 1.1762 untuk kondisi aktual. Seluruh selisih data aktual dengan hasil optimasi berdistribusi normal, sehingga digunakan uji t berpasangan. Untuk seluruh uji antara kondisi aktual dengan solusi-solusi hasil optimasi, tidak terdapat perbedaan yang signifikan terhadap nilai rata-rata dari standar deviasi.

Terdapat 7 varian warna, dengan rata-rata dari standar deviasi setiap varian adalah 1.6749 untuk kondisi aktual. Seluruh selisih data aktual dengan hasil optimasi berdistribusi normal, sehingga digunakan uji t berpasangan. Hasil uji perbedaan rata-rata menunjukkan bahwa pada sebagian solusi, terdapat perbedaan yang signifikan antara kondisi aktual dengan solusi-solusi hasil optimasi.

Terdapat 4 varian mesin yang diproduksi. Seluruh varian mesin dari solusi optimasi memiliki standar deviasi lebih tinggi dibandingkan kondisi aktual. Berdasarkan rata-rata persebaran dari *sub-assembly* mesin, sebagian besar rata-rata persebaran hasil solusi optimasi lebih tinggi dibandingkan kondisi aktual.

Terdapat 2 varian model yang diproduksi PT XYZ, yakni model M dan model S. Standar deviasi varian model S dan M dari kondisi aktual lebih rendah dari hampir seluruh standar deviasi dari solusi-solusi hasil optimasi. Kondisi aktual memiliki standar deviasi yang rendah (di bawah 1) sedangkan sebagian besar solusi-solusi aktual memiliki standar deviasi lebih dari 1. Berdasarkan rata-rata persebaran dari *sub-assembly* model, sebagian besar rata-rata persebaran hasil solusi optimasi lebih tinggi dibandingkan kondisi aktual.

Secara keseluruhan, optimasi menggunakan GrEA tidak banyak mengubah persebaran unit pada *sub-assembly suffix* dan *grade*, sebagian memiliki persebaran yang lebih baik, sebagian lainnya lebih tinggi namun tidak signifikan. Pada *sub-assembly* warna, meskipun dipadukan dengan fungsi tujuan melakukan densifikasi warna, tetapi persebaran unitnya tidak turun secara signifikan dilihat dari hasil uji statistik. Hal ini menunjukkan bahwa kedua fungsi tujuan ini tidak saling bertolak belakang, tetapi masih dapat bersinergi menghasilkan solusi yang memiliki *trade-off* yang menjanjikan. Pada *sub-assembly* mesin, sebagian besar solusi optimasi memiliki rata-rata persebaran yang lebih baik dibandingkan kondisi aktual. Sedangkan pada *sub-assembly* model, rata-rata persebaran solusi optimal sebagian besar lebih buruk dibandingkan kondisi aktual, meskipun perbedaannya hanya sekitar 1 standar deviasi.

5.3. Perbandingan Jumlah Proses *Flushing*

Jumlah proses *flushing* dihitung dengan melihat perubahan warna pada jadwal urutan produksi. Apabila warna pada urutan ke- $t+1$ tidak sama dengan urutan ke- t , maka terjadi perubahan warna dan dilakukan proses *flushing*. Optimasi jumlah proses *flushing* dicari dengan memaksimalkan densifikasi warna dari seluruh unit yang diproduksi. Maksimasi densifikasi warna memiliki arti memaksimalkan kedekatan antara warna-warna yang sama, artinya warna tidak harus berjejer secara langsung, namun jarak antar warna yang sejenis masih dekat satu sama lain. Hal ini tidak sama dengan minimasi proses *flushing*. Maksimasi densifikasi warna lebih *robust* dan dapat mengantisipasi perubahan urutan yang dapat terjadi di rantai produksi, seperti terjadinya *defect* yang menyebabkan unit tertentu harus keluar dari jalur produksi dan diperbaiki terlebih dahulu, sehingga mengubah urutan produksi (Joly dan Frein, 2008). Maka, minimasi proses *flushing* semata tidak dapat mengantisipasi perubahan urutan yang terjadi. Oleh karena itu, meskipun densifikasi warna menunjukkan nilai yang tinggi, tidak serta merta berarti bahwa frekuensi proses *flushing* semakin rendah karena warna yang sama berdekatan. Hasil perbandingan jumlah proses *flushing* secara detail adalah sebagai berikut:

a. 4 Juni 2018

Jumlah proses *flushing* aktual yang dihasilkan dengan jadwal urutan produksi aktual yang telah dibuat perusahaan adalah sebanyak 439 proses *flushing*.

Optimasi menggunakan GrEA menghasilkan proses *flushing* paling sedikit sebanyak 260 proses dan paling banyak 306 proses, dengan rata-rata 278.78 proses. Maka, proses *flushing* dapat berkurang paling sedikit sebanyak 133 proses dan paling banyak 179 proses, dengan rata-rata 160.22 pengurangan proses *flushing*.

b. 5 Juni 2018

Jumlah proses *flushing* aktual yang dihasilkan dengan jadwal urutan produksi aktual yang telah dibuat perusahaan adalah sebanyak 455 proses *flushing*. Optimasi menggunakan GrEA menghasilkan proses *flushing* paling sedikit sebanyak 284 proses dan paling banyak 316 proses, dengan rata-rata 303.29 proses. Maka, proses *flushing* dapat berkurang paling sedikit sebanyak 139 proses dan paling banyak 171 proses, dengan rata-rata 151.71 pengurangan proses *flushing*.

c. 6 Juni 2018

Jumlah proses *flushing* aktual yang dihasilkan dengan jadwal urutan produksi aktual yang telah dibuat perusahaan adalah sebanyak 460 proses *flushing*. Optimasi menggunakan GrEA menghasilkan proses *flushing* paling sedikit sebanyak 242 proses dan paling banyak 265 proses, dengan rata-rata 255.88 proses. Maka, proses *flushing* dapat berkurang paling sedikit sebanyak 195 proses dan paling banyak 218 proses, dengan rata-rata 204.12 pengurangan proses *flushing*.

d. 7 Juni 2018

Jumlah proses *flushing* aktual yang dihasilkan dengan jadwal urutan produksi aktual yang telah dibuat perusahaan adalah sebanyak 434 proses *flushing*. Optimasi menggunakan GrEA menghasilkan proses *flushing* paling sedikit sebanyak 283 proses dan paling banyak 293 proses, dengan rata-rata 285.94 proses. Maka, proses *flushing* dapat berkurang paling sedikit sebanyak 141 proses dan paling banyak 151 proses, dengan rata-rata 148.06 pengurangan proses *flushing*.

e. 8 Juni 2018

Jumlah proses *flushing* aktual yang dihasilkan dengan jadwal urutan produksi aktual yang telah dibuat perusahaan adalah sebanyak 437 proses *flushing*. Optimasi menggunakan GrEA menghasilkan proses *flushing* paling sedikit sebanyak 315 proses dan paling banyak 393 proses, dengan rata-rata 358.10 proses. Maka, proses *flushing* dapat berkurang paling sedikit sebanyak 44 proses dan paling banyak 122 proses, dengan rata-rata 78.90 pengurangan proses *flushing*.

f. 20 Juni 2018

Jumlah proses *flushing* aktual yang dihasilkan dengan jadwal urutan produksi aktual yang telah dibuat perusahaan adalah sebanyak 354 proses *flushing*. Optimasi menggunakan GrEA menghasilkan proses *flushing* paling sedikit sebanyak 222 proses dan paling banyak 257 proses, dengan rata-rata 240.38 proses. Maka, proses *flushing* dapat berkurang paling sedikit sebanyak 97 proses dan paling banyak 132 proses, dengan rata-rata 113.63 pengurangan proses *flushing*.

g. 21 Juni 2018

Jumlah proses *flushing* aktual yang dihasilkan dengan jadwal urutan produksi aktual yang telah dibuat perusahaan adalah sebanyak 457 proses *flushing*. Optimasi menggunakan GrEA menghasilkan proses *flushing* paling sedikit sebanyak 321 proses dan paling banyak 345 proses, dengan rata-rata 355.57 proses. Maka, proses *flushing* dapat berkurang paling sedikit sebanyak 112 proses dan paling banyak 136 proses, dengan rata-rata 121.43 pengurangan proses *flushing*.

h. 22 Juni 2018

Jumlah proses *flushing* aktual yang dihasilkan dengan jadwal urutan produksi aktual yang telah dibuat perusahaan adalah sebanyak 484 proses *flushing*. Optimasi menggunakan GrEA menghasilkan proses *flushing* paling sedikit sebanyak 235 proses dan paling banyak 246 proses, dengan rata-rata 240.57 proses. Maka, proses *flushing* dapat berkurang paling sedikit sebanyak 238 proses dan paling banyak 249 proses, dengan rata-rata 243.13 pengurangan proses *flushing*.

i. 25 Juni 2018

Jumlah proses *flushing* aktual yang dihasilkan dengan jadwal urutan produksi aktual yang telah dibuat perusahaan adalah sebanyak 403 proses *flushing*. Optimasi menggunakan GrEA menghasilkan proses *flushing* paling sedikit sebanyak 237 proses dan paling banyak 282 proses, dengan rata-rata 249.86 proses. Maka, proses *flushing* dapat berkurang paling sedikit sebanyak 121 proses dan paling banyak 166 proses, dengan rata-rata 153.14 pengurangan proses *flushing*.

j. 26 Juni 2018

Jumlah proses *flushing* aktual yang dihasilkan dengan jadwal urutan produksi aktual yang telah dibuat perusahaan adalah sebanyak 408 proses *flushing*. Optimasi menggunakan GrEA menghasilkan proses *flushing* paling sedikit sebanyak 322 proses dan paling banyak 348 proses, dengan rata-rata 332.14 proses. Maka, proses *flushing* dapat berkurang paling sedikit sebanyak 60 proses dan paling banyak 86 proses, dengan rata-rata 75.86 pengurangan proses *flushing*.

Secara keseluruhan, GrEA menghasilkan jadwal urutan produksi dengan frekuensi proses *flushing* yang lebih rendah dibandingkan metode yang saat ini dipakai oleh perusahaan, dengan rata-rata pengurangan sebanak 143.48 proses.

5.4. Perbandingan Biaya Proses *Flushing*

Proses *flushing* membutuhkan bahan kimia *thinner* dan larutan lainnya untuk membersihkan *nozzle* atau penyemprot robot cat dari sisa-sisa cat sebelum menyemprotkan warna yang berbeda dari warna unit sebelumnya. Biaya yang diperlukan dalam 1 kali proses *flushing* adalah sebesar Rp 36,624.11, yang didapatkan dari data anggaran *painting shop* periode April 2018. Hasil perbandingan biaya proses *flushing* secara rinci adalah sebagai berikut:

a. 4 Juni 2019

Biaya aktual yang dikeluarkan sebesar Rp 16,077,984.75. Optimasi dengan GrEA menghasilkan biaya proses *flushing* paling sedikit Rp 9,522,268.87 dan maksimal Rp 11,206,977.98 dengan rata-rata Rp 10,209,988.29. Maka, dengan

menggunakan GrEA, biaya proses *flushing* dapat dihemat paling sedikit sebesar Rp 4,871,006.77 dan maksimal Rp 6,555,715.88 dengan rata-rata Rp 5,867,996.46.

b. 5 Juni 2019

Biaya aktual yang dikeluarkan sebesar Rp 16,663,970.53. Optimasi dengan GrEA menghasilkan biaya proses *flushing* paling sedikit Rp 10,401,247.54 dan maksimal Rp 11,573,219.09 dengan rata-rata Rp 11,107,877.45. Maka, dengan menggunakan GrEA, biaya proses *flushing* dapat dihemat paling sedikit sebesar Rp 5,090,751.44 dan maksimal Rp 6,262,722.99 dengan rata-rata Rp 5,556,093.08.

c. 6 Juni 2019

Biaya aktual yang dikeluarkan sebesar Rp 16,847,091.08. Optimasi dengan GrEA menghasilkan biaya proses *flushing* paling sedikit Rp 8,863,034.87 dan maksimal Rp 9,705,389.43 dengan rata-rata Rp 9,371,463.71. Maka, dengan menggunakan GrEA, biaya proses *flushing* dapat dihemat paling sedikit sebesar Rp 7,141,701.65 dan maksimal Rp 7,984,56.21 dengan rata-rata Rp 7,475,627.37.

d. 7 Juni 2019

Biaya aktual yang dikeluarkan sebesar Rp 15,894,864.20. Optimasi dengan GrEA menghasilkan biaya proses *flushing* paling sedikit Rp 10,364,623.43 dan maksimal Rp 10,730,864.54 dengan rata-rata Rp 10,472,461.09. Maka, dengan menggunakan GrEA, biaya proses *flushing* dapat dihemat paling sedikit sebesar Rp 5,163,999.66 dan maksimal Rp 5,530,240.77 dengan rata-rata Rp 5,422,403.11.

e. 8 Juni 2019

Biaya aktual yang dikeluarkan sebesar Rp 16,004,736.53. Optimasi dengan GrEA menghasilkan biaya proses *flushing* paling sedikit Rp 11,536,594.98 dan maksimal Rp 14,393,275.64 dengan rata-rata Rp 13,114,919.77. Maka, dengan menggunakan GrEA, biaya proses *flushing* dapat dihemat paling sedikit sebesar Rp 1,611,460.89 dan maksimal Rp 4,468,141.55 dengan rata-rata Rp 2,889,816.76.

f. 20 Juni 2019

Biaya aktual yang dikeluarkan sebesar Rp 12,964,935.31. Optimasi dengan GrEA menghasilkan biaya proses *flushing* paling sedikit Rp 8,130,552.65 dan maksimal Rp 9,412,396.54 dengan rata-rata Rp 8,803,520.69. Maka, biaya proses *flushing* dapat dihemat paling sedikit sebesar Rp 3,552,538.77 dan maksimal Rp 4,834,382.66 dengan rata-rata Rp 4,161,414.62.

g. 21 Juni 2019

Biaya aktual yang dikeluarkan sebesar Rp 16,737,218.75. Optimasi dengan GrEA menghasilkan biaya proses *flushing* paling sedikit Rp 11,756,339.65 dan maksimal Rp 12,635,318.31 dengan rata-rata Rp 12,290,005.27. Maka, biaya proses *flushing* dapat dihemat paling sedikit sebesar Rp 4,101,900.44 dan maksimal Rp 4,980,879.10 dengan rata-rata Rp 4,447,213.48.

h. 22 Juni 2019

Biaya aktual yang dikeluarkan sebesar Rp 17,726,069.75. Optimasi dengan GrEA menghasilkan biaya proses *flushing* paling sedikit Rp 8,606,666.10 dan maksimal Rp 9,009,531.32 dengan rata-rata Rp 9,150,795.75. Maka, biaya proses *flushing* dapat dihemat paling sedikit sebesar Rp 8,716,538.43 dan maksimal Rp 9,119,403.65 dengan rata-rata Rp 8,915,355.03.

i. 25 Juni 2019

Biaya aktual yang dikeluarkan sebesar Rp 14,759,516.75. Optimasi dengan GrEA menghasilkan biaya proses *flushing* paling sedikit Rp 8,679,914.32 dan maksimal Rp 10,327,999.32 dengan rata-rata Rp 9,150,795.75. Maka, biaya proses *flushing* dapat dihemat paling sedikit sebesar Rp 4,431,517.44 dan maksimal Rp 6,079,602.43 dengan rata-rata Rp 5,608,721.01.

j. 26 Juni 2019

Biaya aktual yang dikeluarkan sebesar Rp 14,942,637.31. Optimasi dengan GrEA menghasilkan biaya proses *flushing* paling sedikit Rp 11,792,963.76 dan maksimal Rp 12,745,190.65 dengan rata-rata Rp 12,164,436.88. Maka, dengan menggunakan GrEA, biaya proses *flushing* dapat dihemat paling sedikit sebesar Rp 2,197,446.66 dan maksimal Rp 3,149,673.55 dengan rata-rata Rp 2,778,200.42.

Dalam 1 hari produksi, perusahaan dapat menghemat rata-rata sebesar Rp 5,254,773.53 untuk proses *flushing*. Dalam 1 tahun produksi, dengan 20 hari kerja dalam satu bulan, dapat diestimasi rata-rata perusahaan mengalami penurunan biaya proses *flushing* sebesar Rp 1,261,145,646.83.

5.5. Perbandingan Tingkat Kualitas *Painting Shop*

Hubungan proses *flushing* dengan tingkat kualitas produksi harian dimodelkan menggunakan regresi linier sederhana. Data yang digunakan adalah data jadwal urutan produksi harian dan data kualitas produksi periode Januari-Juni 2018 (terlampir). Karena jumlah proses *flushing* bergantung pada jumlah unit yang diproduksi, maka proses *flushing* diukur melalui rasio antara jumlah proses flushing dengan jumlah unit yang diproduksi. Tingkat kualitas produksi sebagai variabel dependen, sedangkan rasio *flushing* sebagai variabel independen.

Model regresi linear awal sesuai tabel 4.8 memiliki nilai R^2 sebesar 0.1987, yang berarti variasi dalam model hanya 19.87% dijelaskan oleh rasio *flushing* terhadap jumlah unit. Nilai intersep model sebesar 1.11836, yang berarti ketika nilai rasio *flushing* adalah 0, maka ekspektasi tingkat kualitas memiliki nilai 1.11836 atau 111.83%. Interpretasi terhadap nilai *intercept* tidak memiliki makna apapun apabila nilai variabel independennya tidak mungkin mencapai nilai 0 (Gujarati dan Porter, 2009). Pada data sampel, tidak terdapat rasio *flushing* bernilai 0 dan rasio *flushing* terendah hanya 0.585 atau 58.5% dari total unit yang diproduksi. Namun, meskipun nilai *intercept* tidak memiliki makna praktis, nilai *intercept* tetap dibutuhkan dalam model untuk melakukan prediksi tingkat kualitas. Hal ini disebabkan karena koefisien *intercept* signifikan secara statistik ($p\text{-value} < 0.05$), yang artinya nilai *intercept* tidak sama dengan 0, sehingga *intercept* tidak dapat dihilangkan dari model karena dapat menyebabkan *specification bias* (Gujarati dan Porter, 2009). Sedangkan *slope* dari rasio *flushing* bernilai -0.44217, yang berarti setiap peningkatan nilai rasio *flushing* sebesar 1 akan menurunkan nilai tingkat kualitas sebesar 0.44217 atau 44.2%.

Nilai galat/error diukur menggunakan *root mean squared error* (RMSE) sebesar 0.05299 dan *mean absolute error* (MAE) sebesar 0.0438. Nilai RMSE menunjukkan rata-rata kuadrat dari *error* atau penyimpangan antara nilai aktual variabel dependen dengan

nilai yang dihasilkan dari model, sedangkan MAE merupakan rata-rata nilai absolut dari penyimpangan tersebut. Nilai RMSE dan MAE yang semakin kecil menunjukkan bahwa model memiliki penyimpangan yang rendah pula, sehingga dapat dikatakan model akurat. Nilai MAE lebih mudah diinterpretasi dan dapat langsung digunakan untuk mengetahui rata-rata penyimpangan pada model. Pada model regresi linier awal, MAE sebesar 0.0438 menunjukkan bahwa rata-rata nilai output model menyimpang sebesar 4.38% dari nilai aktual. RMSE sebagai bentuk kuadrat dari nilai penyimpangan lebih sensitif terhadap perubahan nilai *error* dan lebih sering digunakan untuk membandingkan performa antara dua model karena sensitif terhadap nilai *error* yang besar.

Setelah mendapatkan persamaan model regresi linear, dilakukan uji hipotesis, meliputi uji linearitas, uji normalitas, uji autokorelasi dan uji heterokedastisitas. Model yang dihasilkan memenuhi asumsi linearitas, normalitas, dan homokedastisitas, namun terdapat autokorelasi pada nilai residualnya, dengan nilai durbin watson sebesar 0.9284. Model yang tidak memenuhi asumsi klasik, dalam hal ini autokorelasi, menghasilkan estimator yang bias. Beberapa dampak negatif apabila mengabaikan autokorelasi adalah model menghasilkan nilai R^2 yang lebih besar (*overestimate*) serta mengambil kesimpulan yang salah pada uji signifikansi variabel (Gujarati dan Porter, 2009). Salah satu cara untuk mengatasi autokorelasi pada model adalah memeriksa apakah data yang digunakan berupa data *time series* (Gujarati dan Porter, 2009). Karena data yang digunakan merupakan data *time series*, maka pada model ditambahkan variabel *trend/t* sebagai variabel bebas.

Model regresi linear modifikasi yang telah ditambahkan variabel tren (*t*) memiliki nilai R^2 sebesar 0.6258, yang berarti 62.58% variasi dihasilkan oleh rasio *flush* dan tren. Dengan menambahkan variabel tren, nilai R^2 meningkat sebesar 42.71%. Nilai intersep model sebesar 1.106. Nilai *slope* rasio *flushing* sebesar -0.337, yang artinya setiap peningkatan nilai rasio *flushing* sebesar 1 akan menurunkan tingkat kualitas sebesar 0.337 atau 33.7%. Nilai *slope* variabel *trend* sebesar -0.00115, yang artinya setiap peningkatan nilai *trend* sebesar 1 akan menurunkan tingkat kualitas sebesar 0.00115 atau 0.115%.

Model memprediksi menggunakan variabel *trend*, sehingga ketika *trend* memiliki jangka panjang, seperti satu tahun lagi dari sekarang, model mungkin kurang tepat dalam memprediksi tingkat kualitas. Misalkan dalam 1 tahun lagi (dengan 20 hari kerja per

bulan), kontribusi variabel *trend* terhadap tingkat kualitas adalah sebesar -0.27715. Apabila kita anggap *rasio flushing* adalah 0.5, maka tingkat kualitas diprediksi hanya mencapai 66.13%. Meskipun *rasio flushing* rendah, sehingga harusnya tingkat kualitasnya tinggi, namun karena *trend* bernilai besar, maka tingkat kualitas tetap menjadi rendah. Maka, model ini kurang tepat untuk digunakan memprediksi tingkat kualitas dalam jangka panjang. Salah satu solusinya adalah memperbarui data dengan mengumpulkan lebih banyak data untuk diproses ke dalam model sehingga dapat menghasilkan model yang lebih *robust* terhadap perubahan *trend*.

Nilai *error* menggunakan RMSE sebesar 0.0362 dan nilai MAE sebesar 0.0285. Baik nilai RMSE maupun MAE dari model modifikasi lebih rendah dibandingkan model awal, yang menunjukkan bahwa model modifikasi lebih akurat. Hasil uji asumsi klasik menunjukkan bahwa model modifikasi telah memenuhi asumsi klasik sehingga model yang dihasilkan memiliki estimator yang tidak bias dan dapat digunakan untuk memprediksi tingkat kualitas.

Hasil perbandingan tingkat kualitas optimasi dengan aktual secara rinci adalah sebagai berikut:

a. 4 Juni 2019

Tingkat kualitas aktual pada 4 Juni adalah sebesar 72.25% unit bebas cacat dengan *rasio proses flushing* adalah 81.56% dari total unit. Prediksi kualitas aktual dengan model regresi linear adalah 70.97%. Optimasi dengan GrEA menghasilkan *rasio proses flushing* paling rendah 48.42% dan maksimal 56.98%, dengan rata-rata 51.91%. Tingkat kualitas yang dihasilkan berdasarkan model regresi linear paling rendah adalah 79.32% dan maksimal 82.21%, dengan rata-rata 81.03%. Maka, peningkatan kualitas yang didapatkan paling rendah sebesar 7.07% dan maksimal 9.96%, dengan rata-rata 8.78%.

b. 5 Juni 2019

Tingkat kualitas aktual pada 5 Juni adalah sebesar 75.68% unit bebas cacat dengan *rasio proses flushing* adalah 79.46% dari total unit. Prediksi kualitas aktual dengan model regresi linear adalah 70.78%. Optimasi dengan GrEA menghasilkan *rasio proses flushing* paling rendah 51.17% dan maksimal 56.94%, dengan rata-rata 54.65%. Tingkat kualitas yang dihasilkan berdasarkan

model regresi linear paling rendah adalah 79.22% dan maksimal 81.17%, dengan rata-rata 79.99%. Maka, peningkatan kualitas yang didapatkan paling rendah sebesar 3.55% dan maksimal 5.49%, dengan rata-rata 4.32%.

c. 6 Juni 2019

Tingkat kualitas aktual pada 6 Juni adalah sebesar 76.19% unit bebas cacat dengan rasio proses *flushing* adalah 76.37% dari total unit. Prediksi kualitas aktual dengan model regresi linear adalah 70.95%. Optimasi dengan GrEA menghasilkan rasio proses *flushing* paling rendah 42.68% dan maksimal 46.74%, dengan rata-rata 45.13%. Tingkat kualitas yang dihasilkan berdasarkan model regresi linear paling rendah adalah 82.54% dan maksimal 83.91%, dengan rata-rata 83.09%. Maka, peningkatan kualitas yang didapatkan paling rendah sebesar 6.35% dan maksimal 7.72%, dengan rata-rata 6.90%.

d. 7 Juni 2019

Tingkat kualitas aktual pada 7 Juni adalah sebesar 79.72% unit bebas cacat dengan rasio proses *flushing* adalah 72.66% dari total unit. Prediksi kualitas aktual dengan model regresi linear adalah 72.38%. Optimasi dengan GrEA menghasilkan rasio proses *flushing* paling rendah 49.91% dan maksimal 51.68%, dengan rata-rata 50.43%. Tingkat kualitas yang dihasilkan berdasarkan model regresi linear paling rendah 80.77% dan maksimal 81.36%, dengan rata-rata 81.18%. Maka, peningkatan kualitas yang didapatkan paling rendah sebesar 1.05% dan maksimal 1.64%, dengan rata-rata 1.47%.

e. 8 Juni 2019

Tingkat kualitas aktual pada 8 Juni adalah sebesar 76.71% unit bebas cacat dengan rasio proses *flushing* adalah 69.01% dari total unit. Prediksi kualitas aktual dengan model regresi linear adalah 72.85%. Optimasi dengan GrEA menghasilkan rasio proses *flushing* paling rendah 53.94% dan maksimal 67.29%, dengan rata-rata 61.32%. Tingkat kualitas yang dihasilkan berdasarkan model regresi linear paling rendah 75.39% dan maksimal 79.89%, dengan rata-rata 77.40%. Maka, peningkatan kualitas yang didapatkan paling rendah sebesar -1.33% dan maksimal 3.18%, dengan rata-rata 0.69%. Terdapat solusi optimasi yang memiliki tingkat kualitas lebih rendah dibandingkan kualitas aktual, tetapi

lebih tinggi dibandingkan kualitas prediksi aktual. Solusi tersebut memiliki rasio *flushing* sedikit lebih besar dibandingkan rasio *flushing* aktual, sehingga dapat diprediksi tingkat kualitasnya juga akan hampir senilai. Dengan nilai *error* model (dalam MAE) sebesar 0.0285, dapat kita artikan bahwa terdapat kemungkinan nilai prediksi meleset rata-rata sebesar 2.85% dari nilai yang sebenarnya. Kedua faktor tersebut menyebabkan tingkat kualitas lebih rendah dibandingkan kondisi aktual.

f. 20 Juni 2019

Tingkat kualitas aktual pada 20 Juni adalah sebesar 76.04% unit bebas cacat dengan rasio proses *flushing* adalah 77.58% dari total unit. Prediksi kualitas aktual dengan model regresi linear adalah 71.73%. Hasil optimasi menghasilkan rasio proses *flushing* paling rendah 48.79% dan maksimal 56.48%, dengan rata-rata 52.83%. Tingkat kualitas yang dihasilkan berdasarkan model regresi linear paling rendah 78.91% dan maksimal 81.51%, dengan rata-rata 80.15%. Maka, peningkatan kualitas yang didapatkan paling rendah sebesar 2.87% dan maksimal 5.46%, dengan rata-rata 4.10%.

g. 21 Juni 2019

Tingkat kualitas aktual pada 21 Juni adalah sebesar 76.87% unit bebas cacat dengan rasio proses *flushing* adalah 70.92% dari total unit. Prediksi kualitas aktual dengan model regresi linear adalah 71.64%. Hasil optimasi menghasilkan rasio proses *flushing* paling rendah 54.59% dan maksimal 58.67%, dengan rata-rata 57.07%. Tingkat kualitas yang dihasilkan berdasarkan model regresi linear paling rendah 78.06% dan maksimal 79.44%, dengan rata-rata 78.60%. Maka, peningkatan kualitas yang didapatkan paling rendah sebesar 1.19% dan maksimal 2.57%, dengan rata-rata 1.73%.

h. 22 Juni 2019

Tingkat kualitas aktual pada 22 Juni adalah sebesar 70.90% unit bebas cacat dengan rasio proses *flushing* adalah 67.68% dari total unit. Prediksi kualitas aktual dengan model regresi linear adalah 71.50%. Hasil optimasi menghasilkan rasio proses *flushing* paling rendah 37.78% dan maksimal 39.55%, dengan rata-rata 38.68%. Tingkat kualitas yang dihasilkan berdasarkan model regresi linear

paling rendah 84.39% dan maksimal 84.99%, dengan rata-rata 84.69%. Maka, peningkatan kualitas yang didapatkan paling rendah sebesar 13.49% dan maksimal 14.09%, dengan rata-rata 13.79%.

i. 25 Juni 2019

Tingkat kualitas aktual pada 25 Juni adalah sebesar 73.18% unit bebas cacat dengan rasio proses *flushing* adalah 77.20% dari total unit. Prediksi kualitas aktual dengan model regresi linear adalah 71.59%. Hasil optimasi menghasilkan rasio proses *flushing* paling rendah 45.40% dan maksimal 54.02%, dengan rata-rata 47.87%. Tingkat kualitas yang dihasilkan berdasarkan model regresi linear paling rendah 79.40% dan maksimal 82.30%, dengan rata-rata 81.47%. Maka, peningkatan kualitas yang didapatkan paling rendah sebesar 6.22% dan maksimal 9.12%, dengan rata-rata 8.29%.

j. 26 Juni 2019

Tingkat kualitas aktual pada 26 Juni adalah sebesar 87.79% unit bebas cacat dengan rasio proses *flushing* adalah 56.28% dari total unit. Prediksi kualitas aktual dengan model regresi linear adalah 70.97%. Hasil optimasi menghasilkan rasio proses *flushing* paling rendah 46.20% dan maksimal 49.93%, dengan rata-rata 47.65%. Tingkat kualitas yang dihasilkan berdasarkan model regresi linear paling rendah 80.66% dan maksimal 81.92%, dengan rata-rata 81.43%. Maka, kualitas yang didapatkan tidak meningkat, tetapi menurun paling rendah sebesar 9.27% dan maksimal 10.53%, dengan rata-rata 10.04%.

Secara keseluruhan, solusi yang dihasilkan optimasi menggunakan GrEA dapat meningkatkan kualitas rata-rata sebesar 5.70%. Meskipun pola data periode Januari-Juni 2018 berupa data yang fluktuatif, tetapi terdapat tren menurun secara linear, sehingga model regresi linear yang digunakan cukup efektif dalam memprediksi, meskipun masih memiliki nilai galat yang cukup besar. Penelitian lebih lanjut dapat mengintegrasikan model prediksi dengan variabel-variabel lain yang berhubungan dengan kualitas produk sehingga didapatkan model yang lebih akurat. Beberapa penelitian terdahulu menyebutkan bahwa terdapat berbagai faktor yang mempengaruhi seperti: *fluid delivery*, *air pressure*, *paint dry speed* (Rahman dkk., 2015), kelembaban/*humidity*, karat dan serat kain (Dhafr dkk., 2006).

5.6. Analisis Dominansi Solusi

Dominansi solusi menunjukkan bahwa suatu solusi atau himpunan solusi mendominasi solusi atau himpunan solusi lainnya. Dalam permasalahan *multi-objective*, sulit untuk memutuskan secara langsung melalui nilai fungsi tujuannya apakah suatu himpunan solusi mendominasi himpunan solusi lainnya. Digunakan *quality indicator* untuk membantu memutuskan hubungan dominansi antara kedua himpunan solusi tersebut. *Quality indicator* yang digunakan adalah *additive epsilon indicator*, karena selain mengetahui hubungan dominansinya, dapat pula diketahui berapa faktor ϵ minimal agar himpunan solusi satu mendominasi himpunan solusi lainnya. *Additive epsilon indicator* dihitung dengan persamaan 2.9. Untuk menginterpretasikan nilai indikatornya, dilakukan 2 kali perbandingan, yakni perbandingan nilai indikator dominansi solusi optimasi terhadap aktual $I_{\epsilon+}(\text{Optimasi}, \text{Aktual})$ dan dominansi solusi aktual terhadap solusi optimasi $I_{\epsilon+}(\text{Aktual}, \text{Optimasi})$. Interpretasi hasil berdasarkan 2 nilai indikator tersebut mengacu pada tabel 2.3. Nilai $I_{\epsilon+}(\text{Optimasi}, \text{Aktual})$ menunjukkan berapa nilai faktor ϵ minimal agar solusi optimasi tidak didominasi solusi aktual dan paling sedikit terdapat 1 solusi dari himpunan solusi optimasi yang tidak lebih buruk dibandingkan solusi aktual. Nilai $I_{\epsilon+}(\text{Aktual}, \text{Optimasi})$ menunjukkan berapa nilai faktor ϵ minimal agar solusi aktual tidak didominasi solusi optimasi. Hasil perhitungan *additive epsilon indicator* adalah sebagai berikut:

a. 4 Juni 2019

Dengan nilai ϵ untuk (Optimasi, Aktual) sebesar -0.109, dapat diketahui bahwa agar solusi optimasi tidak didominasi solusi aktual, maka nilai solusi aktual perlu ditambahkan dengan faktor ϵ paling sedikit -0.109. Nilai ϵ negatif menunjukkan bahwa kedua fungsi tujuan solusi optimasi lebih rendah dibandingkan solusi aktual. Untuk $I_{\epsilon+}(\text{Aktual}, \text{Optimasi})$ memiliki nilai ϵ sebesar 61424.048. Dengan menambahkan nilai ϵ , nilai fungsi tujuan *paint* dari solusi aktual menjadi lebih kecil dibandingkan nilai dari solusi optimasi, sehingga solusi aktual menjadi tidak lebih buruk/tidak didominasi oleh solusi optimasi. Berdasarkan tabel 2.3, nilai $I_{\epsilon+}(\text{Optimasi}, \text{Aktual})$ kurang dari atau sama dengan 0 dan nilai $I_{\epsilon+}(\text{Aktual},$

Optimasi) positif (lebih dari 0) menunjukkan bahwa solusi optimasi lebih baik dibandingkan solusi aktual.

b. 5 Juni 2019

Dengan nilai ε untuk (Optimasi, Aktual) sebesar -0.102, dapat diketahui bahwa agar solusi optimasi tidak didominasi solusi aktual, maka nilai solusi aktual perlu ditambahkan dengan faktor ε paling sedikit -0.102. Nilai ε negatif menunjukkan bahwa kedua fungsi tujuan solusi optimasi lebih rendah dibandingkan solusi aktual. Untuk $I_{\varepsilon+}(\text{Aktual}, \text{Optimasi})$ memiliki nilai ε sebesar 62302.739. Dengan menambahkan nilai ε , nilai fungsi tujuan *paint* dari solusi aktual menjadi lebih kecil dibandingkan nilai dari solusi optimasi, sehingga solusi aktual menjadi tidak lebih buruk/tidak didominasi oleh solusi optimasi.

c. 6 Juni 2019

Dengan nilai ε untuk (Optimasi, Aktual) sebesar -0.122, dapat diketahui bahwa agar solusi optimasi tidak didominasi solusi aktual, maka nilai solusi aktual perlu ditambahkan dengan faktor ε paling sedikit -0.122. Nilai ε negatif menunjukkan bahwa kedua fungsi tujuan solusi optimasi lebih rendah dibandingkan solusi aktual. Untuk $I_{\varepsilon+}(\text{Aktual}, \text{Optimasi})$ memiliki nilai ε sebesar 60473.513. Dengan menambahkan nilai ε , nilai fungsi tujuan *paint* dari solusi aktual menjadi lebih kecil dibandingkan nilai dari solusi optimasi, sehingga solusi aktual menjadi tidak lebih buruk/tidak didominasi oleh solusi optimasi. Berdasarkan tabel 2.3, nilai $I_{\varepsilon+}(\text{Optimasi}, \text{Aktual})$ kurang dari atau sama dengan 0 dan nilai $I_{\varepsilon+}(\text{Aktual}, \text{Optimasi})$ positif (lebih dari 0) menunjukkan bahwa solusi optimasi lebih baik dibandingkan solusi aktual.

d. 7 Juni 2019

Dengan nilai ε untuk (Optimasi, Aktual) sebesar -0.083, dapat diketahui bahwa agar solusi optimasi tidak didominasi solusi aktual, maka nilai solusi aktual perlu ditambahkan dengan faktor ε paling sedikit -0.083. Nilai ε negatif menunjukkan bahwa kedua fungsi tujuan solusi optimasi lebih rendah dibandingkan solusi aktual. Untuk $I_{\varepsilon+}(\text{Aktual}, \text{Optimasi})$ memiliki nilai ε sebesar 113101.407. Dengan menambahkan nilai ε , nilai fungsi tujuan *paint* dari solusi aktual menjadi lebih kecil dibandingkan nilai dari solusi optimasi, sehingga solusi aktual

menjadi tidak lebih buruk/tidak didominasi oleh solusi optimasi. Berdasarkan tabel 2.3, nilai $I_{\varepsilon+}$ (Optimasi, Aktual) kurang dari atau sama dengan 0 dan nilai $I_{\varepsilon+}$ (Aktual, Optimasi) positif (lebih dari 0) menunjukkan bahwa solusi optimasi lebih baik dibandingkan solusi aktual.

e. 8 Juni 2019

Dengan nilai ε untuk (Optimasi, Aktual) sebesar -0.070, dapat diketahui bahwa agar solusi optimasi tidak didominasi solusi aktual, maka nilai solusi aktual perlu ditambahkan dengan faktor ε paling sedikit -0.070. Nilai ε negatif menunjukkan bahwa kedua fungsi tujuan solusi optimasi lebih rendah dibandingkan solusi aktual. Untuk $I_{\varepsilon+}$ (Aktual, Optimasi) memiliki nilai ε sebesar 114947.89. Dengan menambahkan nilai ε , nilai fungsi tujuan *paint* dari solusi aktual menjadi lebih kecil dibandingkan nilai dari solusi optimasi, sehingga solusi aktual menjadi tidak lebih buruk/tidak didominasi oleh solusi optimasi. Berdasarkan tabel 2.3, nilai $I_{\varepsilon+}$ (Optimasi, Aktual) kurang dari atau sama dengan 0 dan nilai $I_{\varepsilon+}$ (Aktual, Optimasi) positif (lebih dari 0) menunjukkan bahwa solusi optimasi lebih baik dibandingkan solusi aktual.

f. 20 Juni 2019

Dengan nilai ε untuk (Optimasi, Aktual) sebesar -0.09, dapat diketahui bahwa agar solusi optimasi tidak didominasi solusi aktual, maka nilai solusi aktual perlu ditambahkan dengan faktor ε paling sedikit -0.09. Nilai ε negatif menunjukkan bahwa kedua fungsi tujuan solusi optimasi lebih rendah dibandingkan solusi aktual. Untuk $I_{\varepsilon+}$ (Aktual, Optimasi) memiliki nilai ε sebesar 49227.701. Dengan menambahkan nilai ε , nilai fungsi tujuan *paint* dari solusi aktual menjadi lebih kecil dibandingkan nilai dari solusi optimasi, sehingga solusi aktual menjadi tidak lebih buruk/tidak didominasi oleh solusi optimasi. Berdasarkan tabel 2.3, nilai $I_{\varepsilon+}$ (Optimasi, Aktual) kurang dari atau sama dengan 0 dan nilai $I_{\varepsilon+}$ (Aktual, Optimasi) positif (lebih dari 0) menunjukkan bahwa solusi optimasi lebih baik dibandingkan solusi aktual.

g. 21 Juni 2019

Dengan nilai ϵ untuk (Optimasi, Aktual) sebesar -0.083, dapat diketahui bahwa agar solusi optimasi tidak didominasi solusi aktual, maka nilai solusi aktual perlu ditambahkan dengan faktor ϵ paling sedikit -0.083. Nilai ϵ negatif menunjukkan bahwa kedua fungsi tujuan solusi optimasi lebih rendah dibandingkan solusi aktual. Untuk $I_{\epsilon+}(\text{Aktual}, \text{Optimasi})$ memiliki nilai ϵ sebesar 113290.435. Dengan menambahkan nilai ϵ , nilai fungsi tujuan *paint* dari solusi aktual menjadi lebih kecil dibandingkan nilai dari solusi optimasi, sehingga solusi aktual menjadi tidak lebih buruk/tidak didominasi oleh solusi optimasi. Berdasarkan tabel 2.3, nilai $I_{\epsilon+}(\text{Optimasi}, \text{Aktual})$ kurang dari atau sama dengan 0 dan nilai $I_{\epsilon+}(\text{Aktual}, \text{Optimasi})$ positif (lebih dari 0) menunjukkan bahwa solusi optimasi lebih baik dibandingkan solusi aktual.

h. 22 Juni 2019

Dengan nilai ϵ untuk (Optimasi, Aktual) sebesar -0.137, dapat diketahui bahwa agar solusi optimasi tidak didominasi solusi aktual, maka nilai solusi aktual perlu ditambahkan dengan faktor ϵ paling sedikit -0.137. Nilai ϵ negatif menunjukkan bahwa kedua fungsi tujuan solusi optimasi lebih rendah dibandingkan solusi aktual. Untuk $I_{\epsilon+}(\text{Aktual}, \text{Optimasi})$ memiliki nilai ϵ sebesar 64960.714. Dengan menambahkan nilai ϵ , nilai fungsi tujuan *paint* dari solusi aktual menjadi lebih kecil dibandingkan nilai dari solusi optimasi, sehingga solusi aktual menjadi tidak lebih buruk/tidak didominasi oleh solusi optimasi.

i. 25 Juni 2019

Dengan nilai ϵ untuk (Optimasi, Aktual) sebesar -0.104, dapat diketahui bahwa agar solusi optimasi tidak didominasi solusi aktual, maka nilai solusi aktual perlu ditambahkan dengan faktor ϵ paling sedikit -0.104. Nilai ϵ negatif menunjukkan bahwa kedua fungsi tujuan solusi optimasi lebih rendah dibandingkan solusi aktual. Untuk $I_{\epsilon+}(\text{Aktual}, \text{Optimasi})$ memiliki nilai ϵ sebesar 113542.353. Dengan menambahkan nilai ϵ , nilai fungsi tujuan *paint* dari solusi aktual menjadi lebih kecil dibandingkan nilai dari solusi optimasi, sehingga solusi aktual menjadi tidak lebih buruk/tidak didominasi oleh solusi optimasi. Berdasarkan tabel 2.3, nilai $I_{\epsilon+}(\text{Optimasi}, \text{Aktual})$ kurang dari atau sama dengan 0 dan nilai

$I_{\varepsilon+}$ (Aktual, Optimasi) positif (lebih dari 0) menunjukkan bahwa solusi optimasi lebih baik dibandingkan solusi aktual.

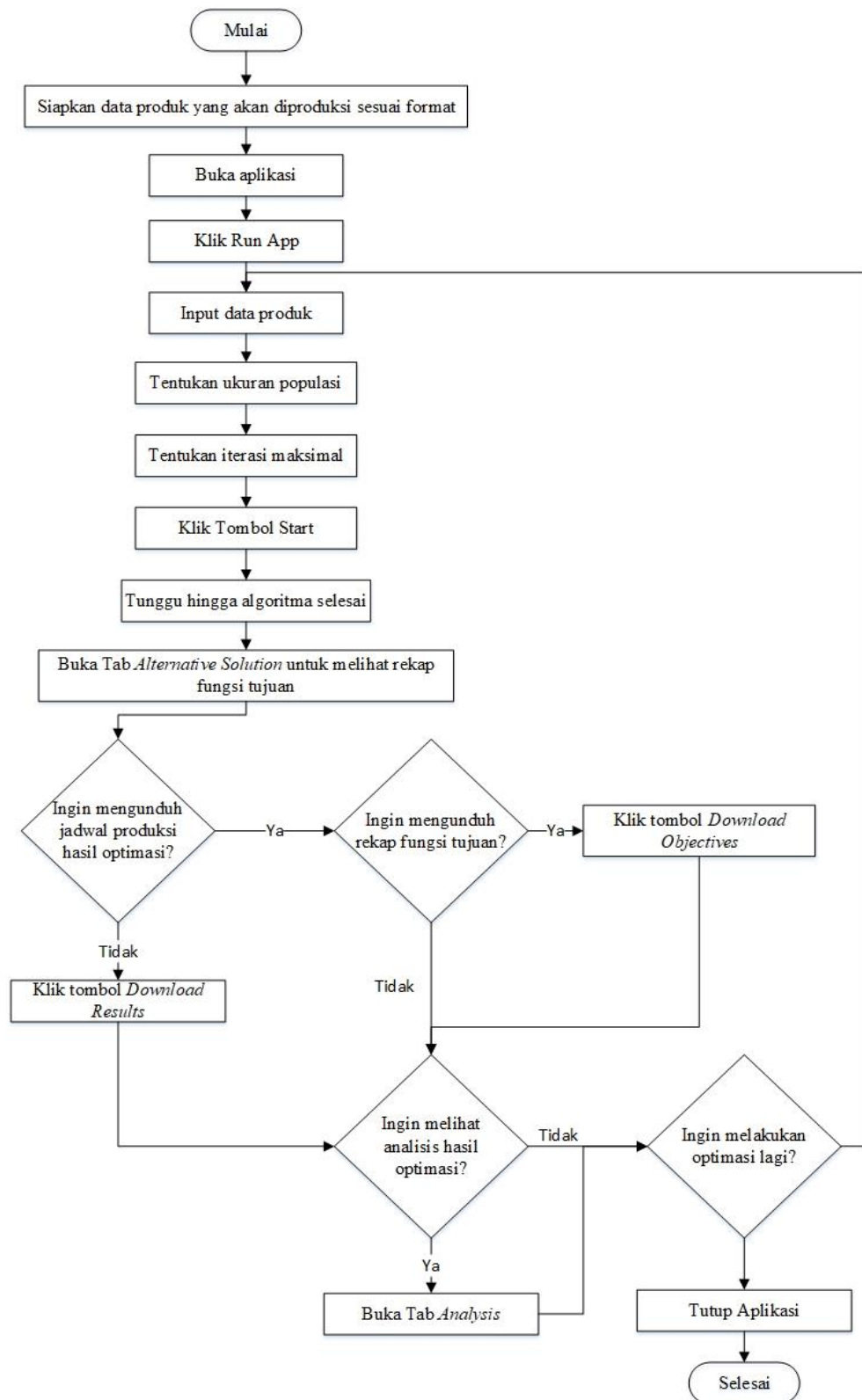
j. 26 Juni 2019

Dengan nilai ε untuk (Optimasi, Aktual) sebesar -0.069, dapat diketahui bahwa agar solusi optimasi tidak didominasi solusi aktual, maka nilai solusi aktual perlu ditambahkan dengan faktor ε paling sedikit -0.069. Nilai ε negatif menunjukkan bahwa kedua fungsi tujuan solusi optimasi lebih rendah dibandingkan solusi aktual. Untuk $I_{\varepsilon+}$ (Aktual, Optimasi) memiliki nilai ε sebesar 28025.8565. Dengan menambahkan nilai ε , nilai fungsi tujuan *paint* dari solusi aktual menjadi lebih kecil dibandingkan nilai dari solusi optimasi, sehingga solusi aktual menjadi tidak lebih buruk/tidak didominasi oleh solusi optimasi. Berdasarkan tabel 2.3, nilai $I_{\varepsilon+}$ (Optimasi, Aktual) kurang dari atau sama dengan 0 dan nilai $I_{\varepsilon+}$ (Aktual, Optimasi) positif (lebih dari 0) menunjukkan bahwa solusi optimasi lebih baik dibandingkan solusi aktual.

Secara keseluruhan, nilai fungsi tujuan yang dihasilkan dari optimasi menggunakan *multi-objective evolutionary algorithm* dengan metode GrEA lebih baik dibandingkan dengan solusi yang dihasilkan oleh sistem yang digunakan perusahaan saat ini sesuai dengan aturan interpretasi yang telah digunakan. Hasil ini sesuai dengan Joly dan Frein (2008), yang menggunakan *multi-objective evolutionary algorithm* dengan metode ε -MOEA dan menghasilkan solusi yang lebih dibandingkan yang digunakan oleh metode lama di PSA Peugeot Citroën.

5.7. Penggunaan Aplikasi

Tatap muka pengguna (*user interface*) dibuat sebagai aplikasi perangkat lunak yang memudahkan pihak perusahaan dalam menjalankan algoritma optimasi penjadwalan urutan produksi. Aplikasi dibuat dengan menggunakan *Shiny* yang merupakan produk gratis dari platform RStudio. Dengan menggunakan *user interface*, pihak perusahaan tidak perlu mempelajari teks algoritma dan dapat langsung menjalankan algoritma hanya dengan menggunakan *mouse*. Gambar 5.2 menunjukkan diagram alur penggunaan aplikasi penjadwalan produksi.

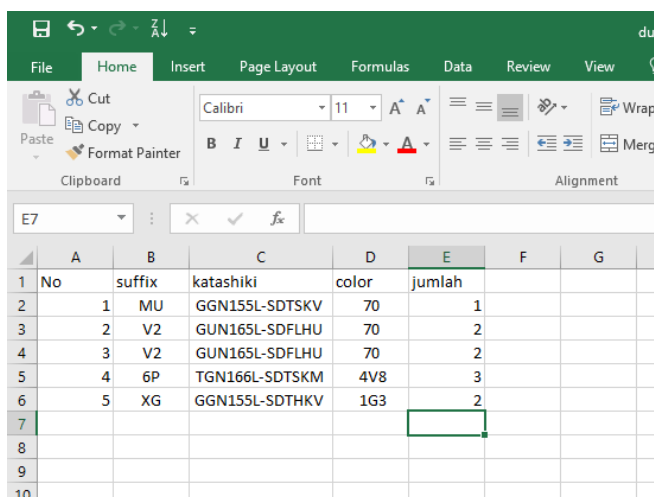


Gambar 5.1 *Flowchart Penggunaan Aplikasi Shiny*

Langkah-langkah penggunaan aplikasi secara rinci adalah sebagai berikut:

1. Siapkan data produk yang akan digunakan sesuai format

Data disiapkan dalam format ekstensi *microsoft excel* “.xlsx” dengan ketentuan sesuai gambar 5.2. Data produk terdiri dari *suffix*, *katashiki*/spesifikasi dan *color*/warna serta jumlah yang diproduksi untuk setiap varian.

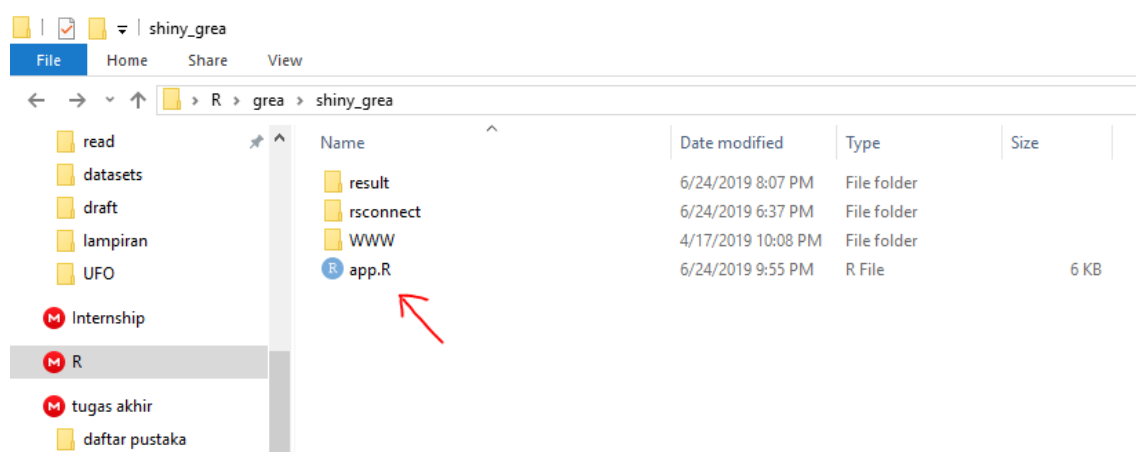


No	suffix	katashiki	color	jumlah
1	MU	GGN155L-SDTSKV	70	1
2	V2	GUN165L-SDFLHU	70	2
3	V2	GUN165L-SDFLHU	70	2
4	6P	TGN166L-SDTSKM	4V8	3
5	XG	GGN155L-SDTHKV	1G3	2

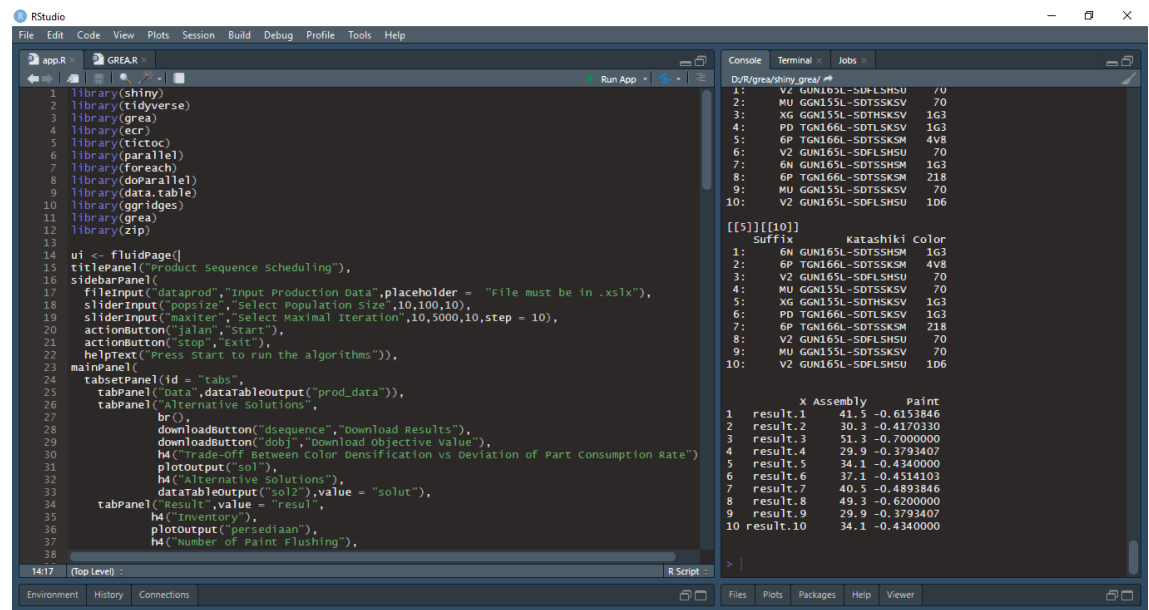
Gambar 5.2 Format Data Input

2. Buka aplikasi

Aplikasi dapat dibuka dengan klik aplikasi bernama app.R sesuai gambar 5.3. Maka, secara otomatis akan terbuka jendela aplikasi dari RStudio sesuai gambar 5.4.



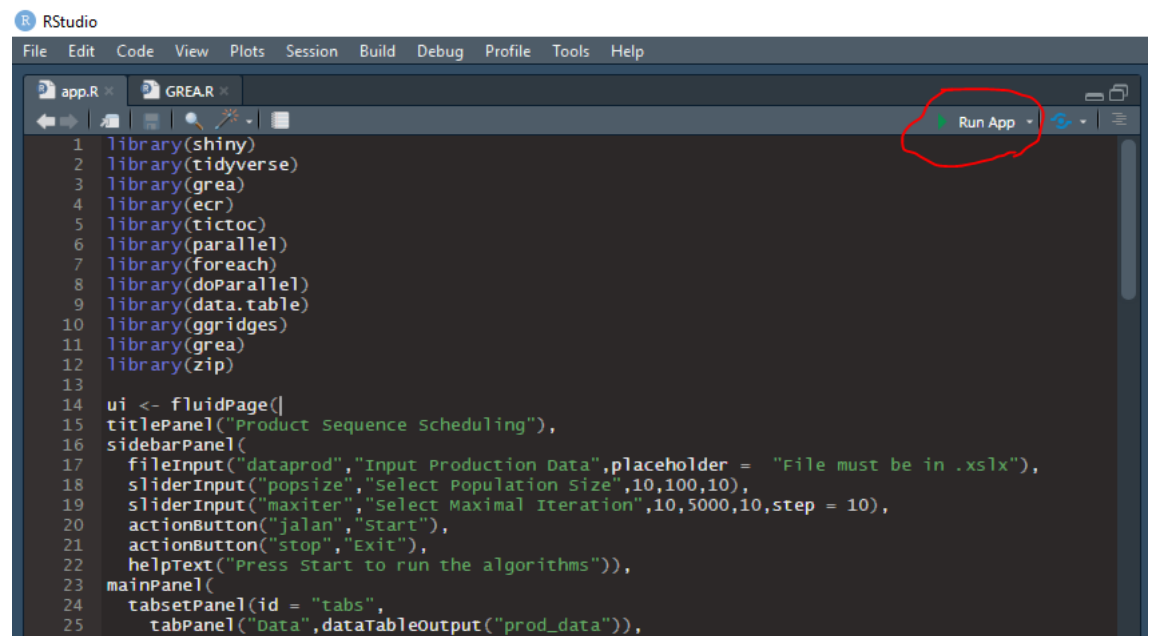
Gambar 5.3 Buka Aplikasi



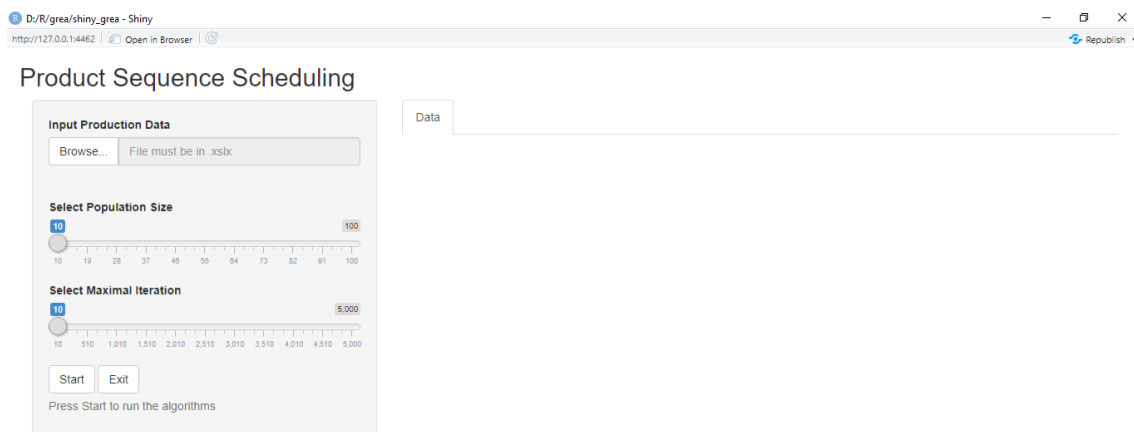
Gambar 5.4 Tampilan RStudio

3. Klik Run App

Klik Run App pada pojok kanan atas dari jendela *script* sesuai gambar 5.5. Maka jendela *shiny* akan otomatis terbuka.



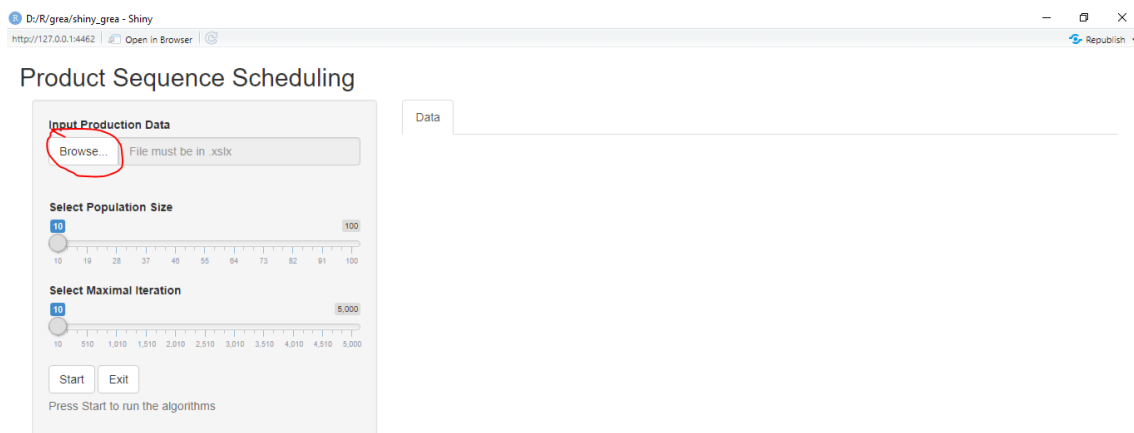
Gambar 5.5 Klik Run App



Gambar 5.6 Tampilan Aplikasi Shiny

4. Input data produk

Data produk diinput dengan klik tombol *Browse* sesuai gambar 5.5. Data produk harus dalam format *microsoft excel* dengan ekstensi *.xlsx* serta berisi informasi terkait *suffix*, spesifikasi, dan warna serta jumlah setiap model.



Gambar 5.7 Input Data

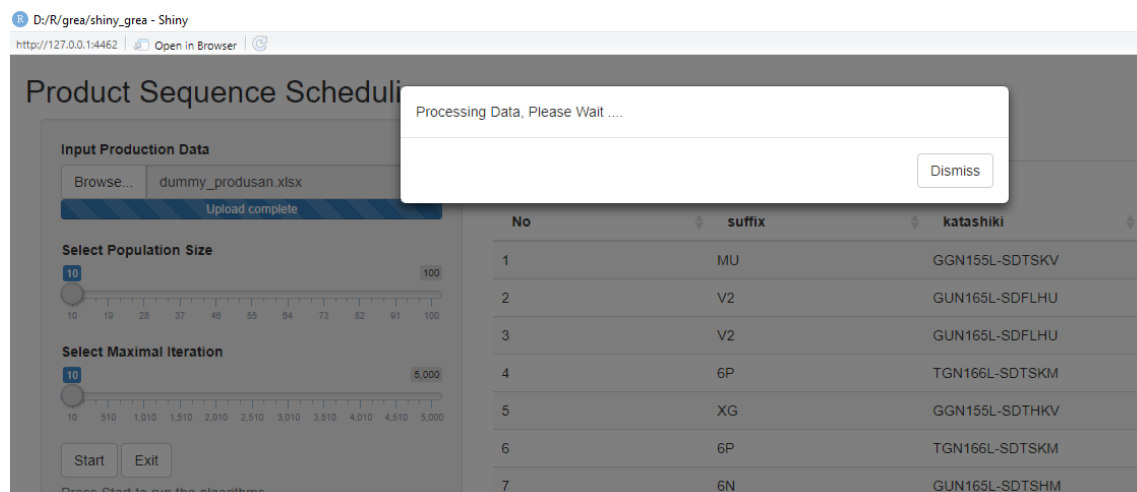
5. Tentukan ukuran populasi dan iterasi maksimal

Ukuran populasi dan iterasi maksimal dapat diatur dengan menggeser *slider* pada masing-masing kolom. Ukuran populasi dibatasi antara 10 hingga 100 populasi, sedangkan jumlah iterasi maksimal dibatasi antara 10 hingga 5000.

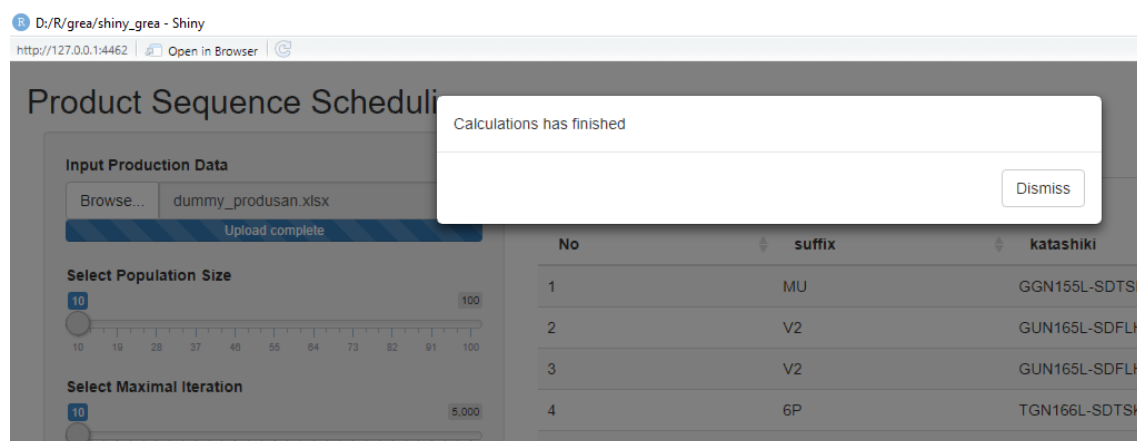
6. Jalankan algoritma

Setelah memastikan bahwa semua input dan parameter sesuai kebutuhan, maka klik tombol *Start* untuk memulai algoritma. Akan muncul kotak dialog yang

memerintahkan pengguna untuk menunggu proses selesai. Ketika proses kalkulasi selesai, kotak dialog akan berganti menjadi pemberitahuan bahwa perhitungan telah selesai.



Gambar 5.8 Menunggu Proses Selesai

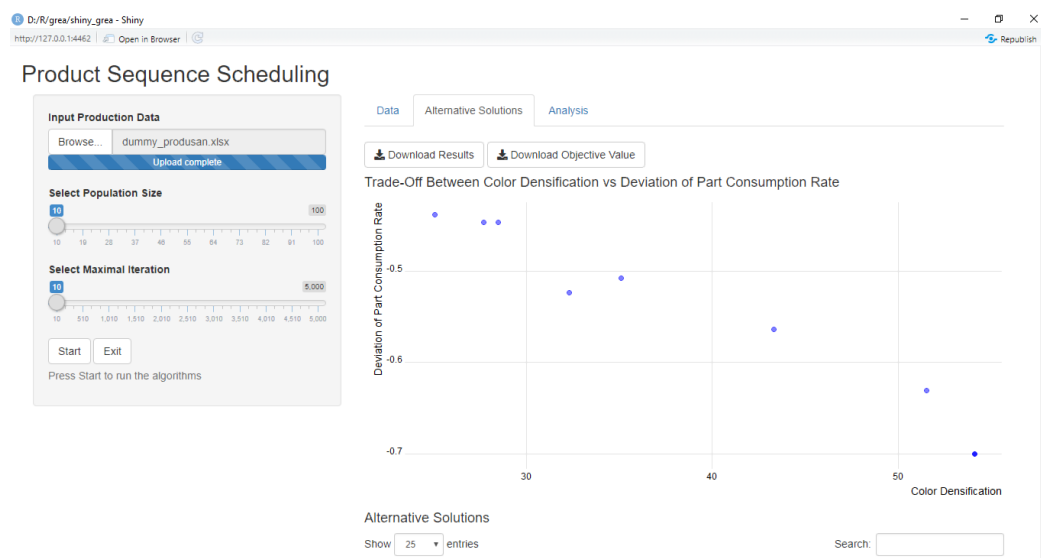


Gambar 5.9 Proses Kalkulasi Selesai

7. Melihat rekapitulasi fungsi tujuan

Fungsi tujuan optimasi ditampilkan dalam bentuk *scatter plot* dan dalam bentuk tabel untuk memudahkan pengguna membaca hasil. Sumbu x menunjukkan *color densification* atau fungsi tujuan yang berupaya meminimasi jumlah proses *flushing*. Sumbu y menunjukkan *deviation of part consumption rate* atau fungsi tujuan yang berupaya meminimasi penyimpangan konsumsi material dan mendukung sistem mencapai *just in time*. Untuk menyimpan jadwal urutan produksi hasil optimasi, klik tombol *Download Result*. Sedangkan apabila ingin

menggunduh nilai rekap fungsi tujuan dapat klik tombol *Download Objective Value*.



Gambar 5.10 Melihat Hasil Optimasi

8. Melihat hasil analisis

Hasil analisis dapat dilihat pada tab *Analysis*.

9. Menutup aplikasi

Aplikasi dapat ditutup melalui tombol Exit di sebelah tombol Start, maupun dengan klik tanda silang di ujung kanan atas aplikasi.

5.8. Rekomendasi Alternatif Jadwal Urutan Produksi

GrEA menghasilkan lebih dari 1 solusi alternatif jadwal urutan yang dapat digunakan oleh perusahaan. Banyaknya alternatif solusi yang ditawarkan menjadi keuntungan bagi pengambil keputusan karena tidak terpaku hanya pada 1 solusi tunggal serta lebih fleksibel dalam menentukan jadwal urutan produksi yang akan diterapkan. Selain itu, pengambil keputusan dapat mengetahui nilai *trade-off* antara dua fungsi tujuan yang telah digunakan. Apabila pengambil keputusan ingin memprioritaskan jadwal produksi yang sesuai dengan prinsip *Just in Time* yakni meminimasi inventori dengan menggunakan aliran produksi yang lebih lancar/*smooth*, maka pengambil keputusan perlu mengambil solusi-solusi dengan nilai fungsi tujuan *part consumption rate* yang lebih rendah (sisi kiri grafik pareto front optimal, gambar 4.1-4.10). Namun, apabila pengambil keputusan menginginkan pengurangan biaya proses *flushing* serta meningkatkan kualitas produk,

pengambil keputusan dapat memilih solusi-solusi dengan nilai fungsi tujuan densifikasi warna yang lebih besar (sisi kanan grafik pareto front optimal, gambar 4.1-4.10). Apabila pengambil keputusan ingin menyeimbangkan kedua kepentingan tersebut, pengambil keputusan dapat memilih solusi-solusi yang berada pada bagian tengah grafik solusi pareto front optimal.

GrEA membutuhkan kemampuan komputasi yang tinggi, dengan rata-rata waktu komputasi untuk membuat jadwal urutan produksi 1 hari (2 *shift*) sebesar 77.58 jam. Untuk memaksimalkan potensi metode ini, perlu digunakan komputer yang memiliki performa lebih tinggi, atau dapat menggunakan bahasa pemrograman alternatif yang lebih cepat seperti C++.

Berdasarkan hasil perbandingan antara kondisi aktual dengan solusi yang dihasilkan oleh GrEA, dengan menggunakan aturan dominansi diketahui bahwa secara keseluruhan solusi yang diberikan oleh GrEA lebih baik dibandingkan kondisi aktual saat ini. Pada tingkat persebaran penggunaan *part* setiap jamnya, untuk *sub-assembly suffix*, *grade*, mesin dan warna tidak memiliki perbedaan signifikan dan cenderung lebih baik dibandingkan kondisi aktual, sedangkan untuk *sub-assembly* model GrEA memiliki tingkat persebaran penggunaan yang sedikit lebih tinggi dibandingkan aktual. Bila dilihat dari jumlah proses *flushing*, biaya proses *flushing* dapat dihemat hingga Rp 1.137.877.835,69. Hasil optimasi dapat meningkatkan kualitas hingga 5.59%. Karena perusahaan memiliki prinsip utama mencapai sistem produksi *just in time*, maka baiknya perusahaan memprioritaskan solusi-solusi yang memiliki nilai fungsi tujuan *deviation of part consumption rate* yang rendah, karena pada fungsi tujuan densifikasi warna, seluruh hasil optimasi lebih unggul dibandingkan kondisi saat ini. Dengan memilih solusi-solusi yang memiliki fungsi tujuan *deviation of part consumption rate* yang rendah, perusahaan dapat mencapai tingkat produksi yang konstan serta mampu menghemat biaya proses *flushing* dan meningkatkan kualitas produk.

BAB VI

PENUTUP

6.1. Kesimpulan

Berdasarkan hasil analisis dan pembahasan, didapatkan beberapa kesimpulan berikut:

1. *Multi-objective Evolutionary Algorithm* merupakan metode optimasi yang dapat menghasilkan beberapa solusi optimal dengan mempertimbangkan lebih dari satu fungsi tujuan. Hasil optimasi jadwal urutan produksi pada pabrik K1 menggunakan GrEA telah mempertimbangkan fungsi tujuan minimasi penyimpangan *part consumption part* pada *assembly shop* dan minimasi proses *flushing* melalui maksimasi densifikasi warna pada *paint shop*.
2. Hasil fungsi tujuan tingkat penyimpangan *part consumption rate* pada *assembly shop* menggunakan GrEA lebih baik dibandingkan hasil aktual dari metode yang saat ini digunakan perusahaan. Meskipun pada GrEA fungsi tujuan yang digunakan juga mempertimbangkan maksimasi densifikasi warna, tetapi nilai tingkat penyimpangan *part consumption rate* tidak mengalami perubahan signifikan.
3. Hasil fungsi tujuan densifikasi warna pada *paint shop* menggunakan GrEA lebih baik dibandingkan dari metode yang saat ini digunakan perusahaan. Jumlah proses *flushing* dapat berkurang secara signifikan, yang berakibat pada penurunan biaya produksi, dalam hal ini biaya melakukan proses *flushing*. Terdapat hubungan/korelasi antara jumlah proses *flushing* dengan tingkat kualitas proses. Jadwal urutan produksi hasil optimasi dengan GrEA memiliki tingkat kualitas yang lebih baik dibandingkan kondisi aktual.
4. Hubungan dominansi merupakan tolak ukur dalam permasalahan *multi-objective* karena tidak dapat memutuskan apakah sebuah solusi lebih baik dibandingkan solusi lainnya secara langsung. Indikator hubungan dominansi yang digunakan adalah *additive epsilon indicator*. Secara keseluruhan, himpunan solusi yang dihasilkan GrEA lebih baik atau dengan kata lain mendominasi solusi dari metode yang digunakan perusahaan.

6.2. Saran

Saran yang dapat penulis berikan untuk penelitian selanjutnya yakni:

1. Penelitian ini dapat menjadi dasar bagi penelitian selanjutnya dengan permasalahan yang serupa, sekaligus pembanding metode-metode *multi-objective evolutionary algorithm* yang lainnya.
2. Optimasi menggunakan *multi-objective evolutionary algorithm* memerlukan kemampuan komputasi yang tinggi, sehingga diharapkan pada penelitian selanjutnya dapat menggunakan komputer dengan spesifikasi yang lebih mumpuni untuk menangani permasalahan tersebut.
3. Dalam melakukan prediksi tingkat kualitas proses, model yang dibangun masih sederhana menggunakan regresi linear dengan variabel tren data dan rasio proses *flushing* terhadap jumlah unit diproduksi saja, sehingga tingkat galat/*error* masih cukup tinggi. Penelitian selanjutnya dapat menggunakan model prediksi yang lebih akurat.
4. Penelitian selanjutnya dapat mengembangkan fungsi tujuan yang digunakan dengan mempertimbangkan kebutuhan dari industri yang berkaitan, sehingga tercapai keputusan yang lebih baik karena telah mempertimbangkan berbagai tujuan.
5. Penelitian selanjutnya dapat melakukan analisis lebih mendalam terkait dampak perubahan jadwal urutan produksi terhadap aliran produksi dan kemampuan *supplier* dalam merespon fluktuasi permintaan.