

**PENGEMBANGAN ALGORITMA *VARIABLE*
NEIGHBORHOOD SEARCH UNTUK LINTAS
PERAKITAN *MULTI-PRODUCT* KOLABORASI
MANUSIA DAN ROBOT**



Muhammad Helmy Zakaria

13418033

PROGRAM STUDI TEKNIK INDUSTRI

INSTITUT TEKNOLOGI BANDUNG

2021

DAFTAR ISI

DAFTAR ISI.....	2
DAFTAR GAMBAR.....	4
DAFTAR TABEL.....	5
DAFTAR LAMPIRAN.....	6
BAB I PENDAHULUAN.....	7
1.1 Latar Belakang	7
1.2 Rumusan Masalah.....	9
1.3 Tujuan Penelitian	9
1.4 Batasan dan Asumsi.....	9
1.5 Sistematika Penulisan	10
1.6 Peta Penelitian.....	11
BAB II LANDASAN TEORI.....	12
2.1 Lintas Perakitan	12
2.1.1 Definisi Lintas Perakitan.....	12
2.1.2 Jenis Lintas Perakitan.....	12
2.2 <i>Collaborative Robot</i>	13
2.3 Algoritma Metaheuristik.....	14
2.3.1 Definisi dan Karakteristik Metaheuristik	14
2.3.2 Klasifikasi Metaheuristik	15
2.4 <i>Variable Neighborhood Search</i> (VNS).....	15
2.4.1 Definisi <i>Variable Neighborhood Search</i> (VNS)	15
2.4.2 Prosedur algoritma <i>Variable Neighborhood Search</i> (VNS)	16
2.5 Model perhitungan <i>Mixed-Model Assembly Line</i>	17
2.5.1 Fungsi Tujuan.....	20
2.5.2 Fungsi Pembatas.....	20
2.5.3 Keterangan Persamaan	22
BAB III METODOLOGI PENELITIAN	24
3.1 Pendahuluan.....	25
3.1.1 Studi Pendahuluan.....	25
3.1.2 Identifikasi Masalah/ <i>Gap</i> Kondisi	25
3.1.3 Perumusan Masalah.....	25
3.1.4 Penentuan Tujuan dan Batasan Penelitian	26
3.2 Studi Literatur	26
3.2.1 Studi Literatur ALBP, Metaheuristik, dan HRC	26

3.2.2 Penentuan Model Acuan ALBP untuk Algoritma VNS.....	26
3.3 Pengembangan Algoritma.....	26
3.3.1 Pengembangan Algoritma VNS untuk ALBP-HRC	27
3.3.2 Verifikasi dan Validasi Algoritma VNS	27
3.4 Pengumpulan dan Pengolahan Data.....	28
3.4.1 Penentuan Data yang akan Digunakan.....	28
3.4.2 Perancangan Eksperimen dan Komputasi	28
3.5 Analisis Algoritma	28
3.6 Penentuan Kesimpulan & Saran	28
BAB IV PENGUMPULAN DAN PENGOLAHAN DATA.....	29
4.1 Rencana Penelitian.....	29
DAFTAR PUSTAKA	30

DAFTAR GAMBAR

Gambar 1 <i>Flowchart</i> Algoritma <i>Variable Neighborhood Search</i>	17
Gambar 2 Metodologi Penelitian	24

DAFTAR TABEL

Tabel 1 Peta Penelitian.....	11
Tabel 2 Perbandingan Robot <i>Collaborative</i> dan Konvensional	14
Tabel 3 Keterangan Indeks	18
Tabel 4 Keterangan Notasi Variabel Keputusan.....	18
Tabel 5 Keterangan Notasi Parameter	19
Tabel 6 Keterangan Persamaan.....	22
Tabel 7 <i>Gantt Chart</i> Penelitian	29

DAFTAR LAMPIRAN

No table of figures entries found.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Saat ini revolusi industri 4.0 berdampak kepada perkembangan industri yang berfokus pada teknologi-teknologi digital. Revolusi ini berdampak pada *self-optimizing* pada proses manufaktur perusahaan beserta efisiensi & produktivitas yang meningkat. Walaupun demikian, penerapan teknologi baru yang efektif tergantung bagaimana perusahaan mengimplementasikan teknologi terhadap proses yang ada (Sharma & Bandari, 2021). Teknologi yang digunakan pada revolusi ini salah satunya adalah *collaborative robot* (cobot).

Collaborative robot (cobot) merupakan perangkat robotik yang dapat membantu pekerjaan manusia di dunia industri dan berkolaborasi seperti tim (Fogli, Gargioni, Guida, & Fabio Tampalini, 2021). Kolaborasi manusia dan robot dapat dilakukan karena cobot dilengkapi dengan mekanisme keselamatan bagi manusia atau pekerja. Cobot juga dapat membantu pekerja pada lini manufaktur khususnya terkait masalah atau aspek ergonomi pekerja seperti gerakan yang berulang atau postur yang tidak normal. Selain itu, cobot umumnya bersifat *mobile* karena ringan dan bisa dipindah ke tempat-tempat lain. Aplikasi cobot di industri adalah robot yang khusus dirancang agar mampu berkolaborasi dengan manusia dan untuk aktivitas yang bersifat *pre-determined* (Franklin & Dominguez, Collaborative robotics: New era of human–robot cooperation in the workplace, 2020).

Cobot sudah diterapkan di banyak perusahaan di dunia. Di Indonesia, contoh perusahaan yang menggunakan cobot adalah PT. JVC Electronics Indonesia (JEIN), perusahaan manufaktur komponen elektronik/elektrik di Indonesia. Agar tetap bersaing dengan kompetitor, PT. JEIN mengimplementasikan tujuh unit cobot UR3 dengan tujuan peningkatan kualitas dan produktivitas. Dampak yang dirasakan adalah pengurangan biaya operasional lebih dari 80.000 USD dan peningkatan proses produksi dan produktivitas & keamanan pekerja. Selain PT. JEIN, contoh perusahaan lain yang menerapkan cobot adalah Nissan, perusahaan otomotif dan subkontraktor. Suatu waktu, di pabrik Nissan di Jepang, penambahan umur tenaga kerja yang tidak diiringi dengan regenerasi yang baik dan keperluan untuk memotong biaya upah pekerja bantuan karena proses produksi tidak efisien menyebabkan Nissan memutuskan untuk mengimplementasikan cobot UR10 untuk aktivitas pelonggaran baut pada *cam bracket* kepala silinder dan pemasangan mesin *block intake manifolds*.

Dengan adanya penerapan cobot di Indonesia serta dampak yang dihasilkan, hal ini berpeluang besar akan menarik perusahaan lain untuk mencoba mengimplementasikan hal yang sama. Selain itu, penerapan-penerapan cobot di perusahaan juga membuka penelitian tentang *human-robot collaborative* (HRC) untuk integrasi cobot dalam lintas perakitan dan metode pencarian solusi optimal. Salah satu penelitian terkait integrasi HRC dalam lintas perakitan telah dilakukan oleh Firda Setiyadin Putri dalam tugas akhir berjudul tentang pengembangan algoritma metaheuristik *Variable Neighborhood Search* (VNS) untuk lintasan perakitan HRC.

Metode metaheuristik adalah metode heuristik berlevel tinggi untuk pencarian solusi yang bisa adaptasi dengan berbagai jenis permasalahan, salah satunya adalah permasalahan *assembly line balancing*. Terdapat beberapa jenis metode metaheuristik. *Variable Neighborhood Search* (VNS) adalah salah satu algoritma metaheuristik pencarian solusi berupa titik lokal minimum yang bersifat sistematis dengan mengikuti perubahan lingkungan.

Model yang dikembangkan dalam tugas akhir ini untuk masalah alokasi tugas dan sumber daya manusia & robot di mana alternatifnya adalah seorang manusia, seorang robot, atau kolaborasi manusia & robot sebagai operator stasiun kerja. Fungsi tujuan dari algoritma yang dikembangkan adalah minimasi biaya total lintasan dengan waktu perhitungan pencarian solusi yang efektif. Algoritma pada tugas akhir ini berfokus pada lintas perakitan untuk varian produk tunggal (*single product*) dan perakitan satu sisi (*single-sided line*).

Keluaran dari penelitian tugas akhir ini adalah hasil komputasi mencapai nilai optimal untuk biaya sebesar 8000 -12397 untuk tugas sebanyak 9 sampai 25 dengan waktu komputasi VNS terkecil 16 detik dan terbesar 252 detik berbanding lurus dengan banyaknya tugas. Untuk tugas yang lebih dari 25, waktu komputasi VNS lebih cepat dari metode analitis yaitu 634 detik. Namun, waktu komputasi VNS lebih besar dari metode analitis ketika tugas sebanyak 45. Keseluruhan dari penelitian ini didapat bahwa hasil uji coba menunjukkan solusi yang baik dengan rata-rata *gap* hasil komputasi biaya 13% dan rata-rata *gap* waktu komputasi memiliki penghematan 68% untuk permasalahan dengan jumlah tugas 25 atau lebih.

Sesuai yang sudah disebutkan sebelumnya, pengembangan algoritma ini berfokus kepada lini perakitan untuk varian produk tunggal dan kurang bisa diterapkan untuk lintas perakitan *multi-product*. Saat ini, dalam menjaga keberlangsungan kehidupan perusahaan dan kompetisi dengan kompetitor lain, perusahaan industri diminta untuk mampu memenuhi permintaan konsumen yang bisa berubah-ubah seiring dengan waktu. Pemenuhan ini dilakukan dengan

diverisifikasi produk atau kustomisasi produk sesuai permintaan pelanggan. Hal ini dapat dipenuhi dengan lintas perakitan *multi-product*. Selain itu, pengembangan algoritma ini juga bertujuan untuk mengakomodasi tugas yang lebih banyak dari penelitian sebelumnya dengan waktu komputasi yang efisien mengingat dalam dunia nyata tugas yang ada untuk membuat produk bisa lebih dari 25 tugas.

Berdasarkan penjelasan sebelumnya, diperlukan algoritma yang tidak hanya mengakomodasi perhitungan untuk lintas perakitan *single product*, tetapi juga *mixed* atau *multi-product*. Oleh karena itu algoritma VNS ini perlu dikembangkan untuk permasalahan lintas perakitan *multi-product*. Diharapkan pengembangan ini bisa diimplementasikan untuk kolaborasi manusia robot pada lintas produksi di industri yang menerapkan cobot khususnya untuk *multi-product*.

1.2 Rumusan Masalah

Berdasarkan penjelasan latar belakang, permasalahan yang akan diselesaikan dalam penelitian ini adalah sebagai berikut.

“Bagaimana mengalokasikan tugas dan sumber daya manusia & robot untuk lintasan perakitan *multi-product* agar didapatkan biaya total lintasan minimal dan waktu komputasi algoritma efisien?”

1.3 Tujuan Penelitian

Dari rumusan masalah tersebut, tujuan penelitian ini adalah mengembangkan algoritma VNS untuk alokasi tugas dan sumber daya pada lintasan perakitan cobot agar didapat biaya total lintasan minimal dan waktu komputasi algoritma efisien. Tujuan tersebut dibagi lagi ke beberapa tujuan sebagai berikut.

1. Mengembangkan algoritma metaheuristik *Variable Neighborhood Search* (VNS) untuk kasus lintas perakitan *multi-product*.
2. Melakukan uji coba pengembangan algoritma VNS pada berbagai kasus lintas perakitan cobot.
3. Melakukan analisis karakteristik pengembangan algoritma VNS terhadap perubahan parameter.

1.4 Batasan dan Asumsi

Batasan-batasan dalam penelitian ini adalah sebagai berikut.

1. Fokus lintas perakitan dalam penelitian ini adalah *multi-product*.
2. Alternatif sumber daya untuk lintas perakitan adalah seorang manusia, sebuah robot, atau seorang manusia & sebuah robot yang saling berkolaborasi (cobot) sebagai operator stasiun kerja.
3. Tiap alternatif sumber daya untuk lintas perakitan memiliki biaya dan waktu pengerjaan yang tertentu atau tidak selalu sama.
4. Kolaborasi manusia & robot (cobot) dalam suatu tugas bersifat dependen atau masing-masing harus tersedia.
5. Alternatif untuk satu manusia atau robot dalam suatu tugas bersifat independen atau masing-masing dapat mengerjakan tugas secara paralel.
6. Lintasan perakitan tidak memperhitungkan kemungkinan sumber daya manusia dan robot yang menganggur.

1.5 Sistematika Penulisan

Laporan penelitian ini terdiri dari enam bab dengan rincian setiap bab sebagai berikut.

1. BAB I: PENDAHULUAN

Bab I menjelaskan gambaran umum permasalahan dan penelitian yang dilakukan, yang terdiri atas latar belakang penelitian, rumusan masalah, tujuan penelitian, manfaat penelitian, serta batasan dan asumsi. Bab I juga menjelaskan sistematika penulisan dari penelitian yang menjelaskan gambaran secara garis besar isi laporan penelitian.

2. BAB II: LANDASAN TEORI

Bab II menjelaskan teori-teori yang digunakan dalam penelitian ini, baik seputar lintas perakitan ataupun model dan algoritma. Penjelasan teori ini diperoleh dari buku-buku teks dan jurnal ilmiah. Teori-teori yang akan dijelaskan antara lain, terkait lintas perakitan, *collaborative robot*, algoritma *variable neighborhood search*, dan model matematis acuan.

3. BAB III: METODOLOGI PENELITIAN

Bab III menjelaskan alur pengerjaan penelitian dari awal hingga akhir penelitian. Metodologi penelitian yang dilakukan secara garis besar terdiri atas, pendahuluan, studi literatur, pengumpulan data, pengolahan data, analisis, dan penutup.

4. BAB IV: PENGUMPULAN DAN PENGOLAHAN DATA

Bab IV menjelaskan tentang pengumpulan data-data yang dibutuhkan dalam penelitian. Penelitian menggunakan data yang ada dari paper (Susanto &

Nugraha, 2021) sebagai lanjutan untuk implementasi di lintas perakitan *mixed-model*. Selanjutnya, data-data tersebut akan diolah hingga menghasilkan solusi yang diharapkan.

5. BAB V: ANALISIS

Bab V menjelaskan analisis hasil pengolahan data. Analisis dilakukan dengan membandingkan, meninjau kelebihan & kekurangan dari proses dan solusinya, analisis implementasi solusi, dan analisis implikasi manajerial.

6. BAB VI: KESIMPULAN DAN SARAN

Bab VI berisi kesimpulan penelitian berdasarkan hasil pengolahan data & analisis dan saran yang dapat diberikan.

1.6 Peta Penelitian

Penelitian ini bertujuan untuk mengembangkan algoritma metaheuristik *variable neighborhood search* (VNS) untuk lintas perakitan *mixed-model*. Sebelumnya, algoritma ini sudah dicoba untuk tipe lintas perakitan *single model*. Namun, saat ini belum dicoba penelitian untuk penggunaan algoritma pada *mixed-model*. Berikut peta penelitian untuk penelitian ini dan dibandingkan dengan referensi penelitian lainnya.

Tabel 1 Peta Penelitian

Penelitian	Model Lintas Perakitan			Algoritma				Ukuran Performansi	
	<i>Single</i>	<i>Mixed</i>	<i>Multi</i>	SA	VNS	ACO	Genetic	Biaya	Waktu Komputasi
(Susanto & Nugraha, 2021)		√						√	
(Putri, 2021)	√				√			√	√
Penelitian Ini		√			√			√	√

BAB II

LANDASAN TEORI

2.1 Lintas Perakitan

2.1.1 Definisi Lintas Perakitan

Lintas perakitan adalah sistem produksi yang berorientasi aliran di mana peralatan produksi digunakan untuk operasi yang berbeda-beda terhadap material atau benda kerja (Saif, 2014). Di sekitar lintas perakitan terdapat beberapa area dengan masing-masing area memiliki peralatan tersendiri atau berbeda-beda sesuai dengan operasi yang dilakukan. Area ini disebut dengan stasiun kerja yang berlokasi di tempat-tempat tertentu dalam lintas perakitan dengan urutan tertentu sesuai dengan rencana produksi. Lintas perakitan sering digunakan untuk produksi massal atau kustomisasi produk.

Dalam industri manufaktur khususnya di lini produksi atau lintas perakitan, pembuatan produk atau *part* memerlukan serangkaian operasi yang akan menghasilkan produk akhir yang diinginkan. Operasi-operasi elementer ini disebut tugas atau *task* yang terkait dengan suatu operasi perakitan tertentu. Dalam lintas perakitan, terdapat beberapa mesin yang diperlukan untuk menghasilkan produk-produk di mana satu jenis mesin bisa saja melakukan lebih dari satu *task* yang berbeda. Masalah ini yang menyebabkan adanya batasan yang muncul dan harus diperhatikan, yaitu *precedence constraint* dari setiap *task*. *Precedence constraint* merupakan *task* yang harus dilakukan sebelum *task* tersebut dilakukan.

Di lintas perakitan, proses pembuatan produk dilakukan dengan benda kerja yang akan diproses berpindah dari satu stasiun ke stasiun lainnya sesuai urutan operasi atau prosesnya dengan lama waktu tertentu. Rata-rata lama waktu untuk stasiun kerja memproses benda kerja dalam lintas perakitan yang disebut dengan waktu siklus. Waktu siklus ini yang menjadi dasar pengukuran beban kerja per stasiun kerja yang akan menjadi masukan untuk perhitungan penyeimbangan lintas perakitan. Selain *cycle time*, terdapat juga parameter *task time*, yaitu waktu yang diperlukan untuk melakukan suatu *task* oleh seorang atau beberapa pekerja dan alat/mesin.

2.1.2 Jenis Lintas Perakitan

Menurut (Saif, 2014) dan (Becker & Scholl, 2004), berdasarkan karakteristik produk, lintas perakitan terbagi ke dalam tiga jenis. Ketiga jenis lintas perakitan adalah sebagai berikut.

1. *Single model*

Jenis lintas perakitan *single model* hanya memproduksi satu jenis produk dalam jumlah yang besar. Waktu pengerjaan atau *task time* dan *precedence constraint* pada *single model* direpresentasikan dalam satu *precedence diagram* yang berbentuk garis lurus. Karena produksi pada jenis lintas perakitan ini fokus hanya satu jenis produk, proses perakitan terspesialisasi.

2. *Mixed model*

Lintas perakitan *mixed model* memproduksi unit model yang berbeda dalam urutan campuran yang bersifat *arbitrarily* atau berubah-ubah. Model yang berbeda ini memiliki *parent* produk yang sama dirakit. Tiap model yang berbeda bisa memiliki konstrain proses dan *precedence* yang berbeda. Jenis *mixed model* menggabungkan *precedence diagram* tiap model untuk mendapatkan hubungan *precedence* tiap *task* secara keseluruhan.

3. *Multi model*

Lintas perakitan *multi model* memproduksi urutan *batch* dengan operasi *setup* yang bersifat *intermediate*. Model-model yang berbeda pada lintas perakitan ini dirakit atau dikumpulkan dalam satu *batch* di mana masing-masing produk tidak dicampur satu sama lain selama produksi. Tiap penyelesaian produksi satu *batch* memerlukan pengubahan *setup* operasi stasiun sesuai dengan *batch* apa yang selanjutnya diproduksi. Ukuran yang diperhatikan pada jenis lintas perakitan ini adalah *lot size* atau ukuran lot untuk tiap *batch*.

2.2 Collaborative Robot

Collaborative robot atau yang biasa disebut dengan cobot adalah robot yang dirancang untuk membantu manusia dan berkolaborasi dalam mengerjakan suatu *task* tertentu (Franklin, Dominguez, Fryman, & Lewandowski, 2020). Cobot juga membuka pembatas antara manusia dan robot bekerja secara simultan dalam ruang kerja yang sama. Karena kolaborasi ini robot dirancang agar aman bagi pekerja ketika bekerja dalam ruang kerja yang sama.

Cobot memiliki beberapa perbedaan dibandingkan dengan robot konvensional. Secara keseluruhan, perbandingannya adalah dari sisi keuntungan dan kepraktisan penggunaan. Berikut adalah perbandingan cobot dengan robot konvensional (Matúšová, 2019) (Knudsen, 2020).

Tabel 2 Perbandingan Robot *Collaborative* dan Konvensional

<i>Collaborative Robot</i>	Robot konvensional
Pemrograman robot dapat dilakukan dengan mudah.	Pemrograman robot memakan waktu yang cukup lama.
Dilengkapi dengan fitur keamanan.	Tidak ada sensor keamanan.
<i>Mobile</i> atau dapat berpindah tempat dengan mudah.	<i>Immobile</i> atau sulit berpindah tempat.
Menguntungkan mulai dari ukuran lot satu sampai medium dan besar.	Menguntungkan hanya untuk ukuran lot medium ke besar.
Relokasi fleksibel dengan <i>task</i> yang sering berubah.	Instalasi bersifat <i>fix</i> dengan <i>task</i> yang repetitif.

2.3 Algoritma Metaheuristik

2.3.1 Definisi dan Karakteristik Metaheuristik

Heuristik adalah cara dalam menemukan strategi atau peraturan yang baru untuk menyelesaikan masalah. Sementara itu, definisi metaheuristik menurut Sorensen (2013) adalah kerangka kerja algoritma berlevel tinggi dan bersifat independen yang menyediakan kumpulan strategi untuk mengembangkan algoritma optimasi heuristik. Menurut Desale (2015), metaheuristik merupakan metode komputasional yang mengoptimasi masalah dengan secara iteratif mencoba memperbaiki solusi yang akan dipilih dengan ukuran kualitas tertentu. Sama seperti heuristik, metaheuristik juga tidak menjamin solusi yang didapatkan optimal namun bersifat *feasible*. Metaheuristik memiliki beberapa karakteristik pada umumnya. Karakteristik tersebut adalah sebagai berikut.

1. Tujuan metaheuristik adalah secara efisien menjelajahi ruang pencarian untuk mendapatkan solusi yang mendekati optimal.
2. Teknik dari algoritma metaheuristik meliputi prosedur pencarian lokal yang sederhana sampai kompleks.
3. Metaheuristik meliputi pencarian dalam model matematis yang bersifat aproksimasi atau perkiraan dan non-deterministik.
4. Algoritma metaheuristik tidak untuk masalah yang bersifat spesifik, artinya bisa digunakan untuk berbagai masalah.

2.3.2 Klasifikasi Metaheuristik

Metaheuristik dapat diklasifikasikan ke dalam dua kategori (Desale, Rasool, Andhale, & Rane, 2015). Kedua kategori tersebut adalah sebagai berikut.

1. *Single-solution based search*

Single-solution memanipulasi dan mentransformasi satu solusi ketika dilakukan pencarian solusi. Kategori ini, atau yang biasa disebut *S-Metaheuristics*, mencari solusi melalui lintasan pencarian terdekat dalam ruang pencarian masalah yang dihadapi. Pencarian pada lintasan ini dilakukan secara iteratif di mana berpindah dari solusi saat ini ke solusi lain yang ada di ruang pencarian saat ini. *S-Metaheuristics* dapat menyelesaikan masalah optimasi di domain yang berbeda-beda. Contoh algoritma yang *single-solution based search* adalah *simulated annealing* dan *variable neighborhood search* (VNS).

2. *Population based search*

Untuk *population based*, kategori ini melakukan perbaikan secara iteratif dalam satu populasi solusi yang berevolusi. *Population based* memulai dari menginisialisasi populasi yang kemudian menghasilkan populasi solusi baru. Populasi baru yang didapatkan diintegrasikan dengan populasi saat ini (di tahap awal inisialisasi populasi) melalui beberapa tahapan seleksi sampai memenuhi kriteria yang diinginkan. Contoh algoritma yang *population based search* adalah *evolutionary algorithms* dan *ant colony*.

2.4 Variable Neighborhood Search (VNS)

2.4.1 Definisi Variable Neighborhood Search (VNS)

Variable Neighborhood Search (VNS) adalah algoritma metaheuristik yang bertujuan untuk memecahkan masalah optimasi kombinasi dan global, pemrograman nonlinier, dan *mixed-integer* nonlinear (Hansen & Mladenovic, 1998). Algoritma VNS berbasis pada pencarian atau perubahan sistematis lingkungan atau *neighborhood* dalam fase penurunan pencarian untuk menemukan optima lokal dan fase *peturbation* untuk keluar dari kurva *parabolic-valley* yang ada. VNS berdasar kepada tiga fakta berikut.

1. Suatu titik minimum lokal yang memperhitungkan satu struktur *neighborhood* belum tentu menjadi lokal minimum untuk satu struktur lain.
2. Suatu titik minimum global adalah minimum lokal yang memperhitungkan semua kemungkinan struktur *neighborhood*.

3. Suatu titik minimum lokal yang memperhitungkan satu atau beberapa *neighborhoods* relatif dekat satu sama lain untuk banyak masalah.

2.4.2 Prosedur algoritma *Variable Neighborhood Search* (VNS)

Prosedur VNS terdiri dari empat tahap yang dilakukan secara iteratif sebagai berikut.

1. Inisialisasi

Pada tahapan ini, ditentukan solusi yang menjadi solusi awal untuk melanjutkan tahapan selanjutnya pada pencarian solusi. Selain pemilihan solusi awal, ditentukan juga kriteria-kriteria yang menghentikan algoritma ketika melakukan pencarian solusi. Ketika di tahapan inisiasi, dipilih himpunan struktur *neighborhood* untuk pencarian solusi yang lebih baik.

2. *Shaking*

Tahapan *shaking* ini mencari solusi baru yang lebih baik dari solusi awal pada tahapan inisialisasi. Cara yang dilakukan adalah membangkitkan solusi secara acak pada *neighborhood* yang dicari. Selain itu, tahapan ini juga memastikan pencarian solusi tidak terjebak pada titik lokal optimal.

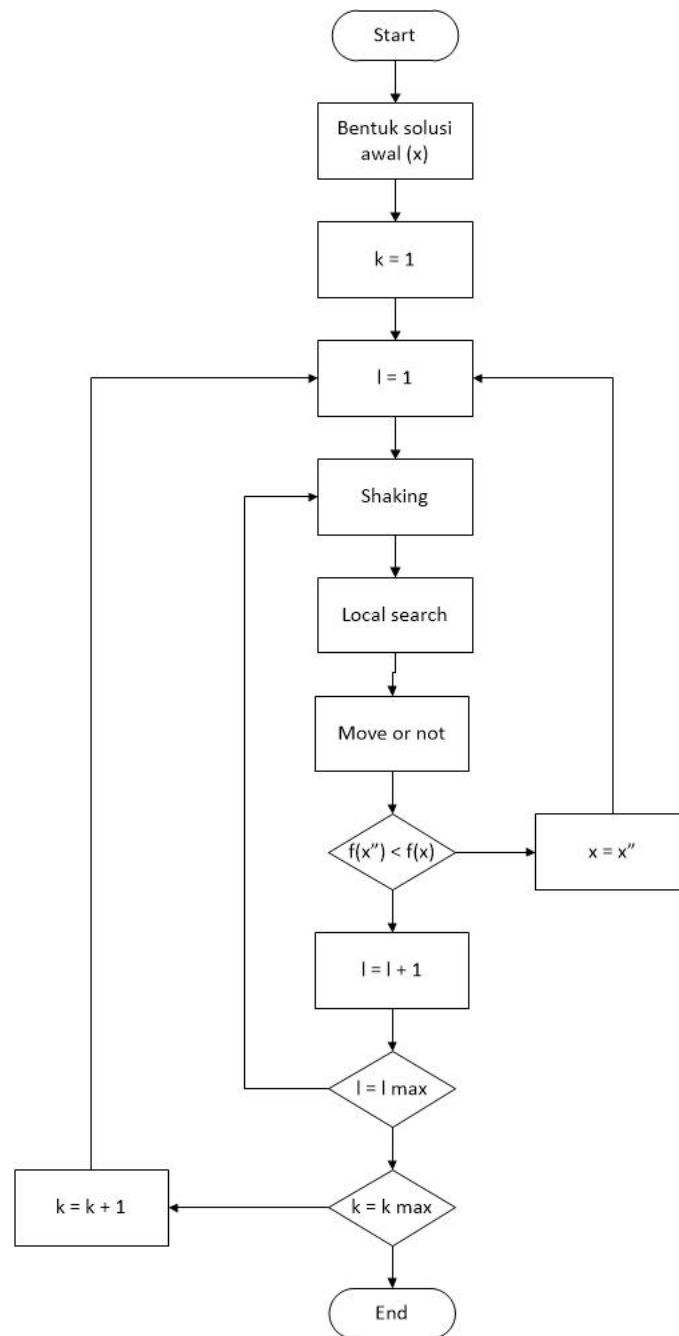
3. *Local search*

Pada *local search* dilakukan pencarian lokal dengan solusi awal didapat dari solusi yang didapat pada tahapan *shaking*. Nantinya akan didapat lokal optima baru yang lebih baik dari solusi sebelumnya. Pencarian dilakukan di daerah sekitar solusi awal (solusi dari tahapan *shaking*).

4. *Move or Not*

Tahapan ini menentukan apakah akan bergerak ke *neighborhood* lain jika *neighborhood* sebelumnya tidak ditemukan solusi baru yang lebih baik. Jika tidak ditemukan, maka pencarian tetap di *neighborhood* yang sama atau tidak berpindah ke *neighborhood* yang lain.

Visualisasi keempat tahap tersebut tergambar pada *flowchart* berikut ini.



Gambar 1 *Flowchart Algoritma Variable Neighborhood Search*

2.5 Model perhitungan *Mixed-Model Assembly Line*

Model yang menjadi acuan pada penelitian ini berasal dari penelitian (Susanto & Nugraha, 2021) yang berjudul “Mixed Model Assembly Line Balancing for Human-Robot Shared Tasks”. Penelitian ini membahas tentang alokasi *tasks* ke stasiun kerja dalam lintas perakitan dengan adanya asumsi bahwa sudah ada pekerjaan dalam stasiun kerja yang prosedurnya manual dan akan ditingkatkan dengan bantuan Cobot. Alokasi ini merupakan satu lintas perakitan yang terdiri dari beberapa stasiun kerja yang berisi kolaborasi manusia dan robot. Selain penyeimbangan lintas perakitan, fokus dari penelitian ini adalah untuk lintas perakitan

yang bersifat *mixed-model*, yaitu lintas perakitan yang menghasilkan produk dengan model berbeda tetapi memiliki *parent* yang sama.

Alternatif dari hasil penyeimbangan lintas perakitan ini menghasilkan tiga alternatif keputusan untuk sumber daya yang digunakan, yaitu hanya menggunakan manusia, hanya menggunakan robot, atau menggunakan kolaborasi manusia dan robot. Parameter-parameter beserta indeks yang melambangkan tiap parameter dan variabel keputusan yang dipertimbangkan dalam penelitian ini ada dalam tabel berikut.

Tabel 3 Keterangan Indeks

Indeks	Keterangan
i, h, p	<i>Task</i> atau tugas
j, q	Stasiun kerja
s	Alternatif pembagian sumber daya manusia/robot untuk suatu <i>task</i> . Bernilai 1 jika hanya menggunakan manusia. Bernilai 2 jika hanya menggunakan robot. Bernilai 3 jika menggunakan kolaborasi manusia dan robot.
m	Model produk

Tabel 4 Keterangan Notasi Variabel Keputusan

Notasi	Keterangan
x_{ijs}	Akan bernilai 1 jika <i>task i</i> ditugaskan ke stasiun kerja <i>j</i> dan dilakukan dengan tipe sumber daya <i>s</i> . Akan bernilai 0 jika selain pernyataan tersebut.
f_{im}	Waktu penyelesaian <i>task i</i> untuk model <i>m</i>
z_{ip}	Urutan posisi untuk pasangan <i>task (i, p) ∈ Q</i> jika <i>i</i> dan <i>p</i> ditugaskan di stasiun kerja yang sama. Akan bernilai 1 jika <i>task i</i> dilakukan sebelum <i>task p</i> . Akan bernilai 0 jika <i>task p</i> dilakukan sebelum tugas <i>i</i>
v_{jm}	Akan bernilai 1 jika stasiun <i>j</i> digunakan untuk model <i>m</i> . Akan bernilai 0 jika berlawanan dari pernyataan tersebut.
u_j	Utilisasi stasiun kerja <i>j</i> dalam lintas perakitan. Akan bernilai 1 jika stasiun <i>j</i> digunakan. Akan bernilai 0 jika stasiun <i>j</i> tidak digunakan.

Notasi	Keterangan
α	Jumlah operator manusia yang dibutuhkan dalam lintas perakitan
ρ	Jumlah robot yang dibutuhkan dalam lintas perakitan

Tabel 5 Keterangan Notasi Parameter

Notasi	Keterangan
I	Himpunan <i>task</i> ; $I = \{1, 2, \dots, n_T\}$
J	Himpunan stasiun kerja; $J = \{1, 2, \dots, n_W\}$
P	Himpunan pasangan (h, i) di mana <i>task</i> h adalah <i>predecessor</i> langsung dari <i>task</i> i
Q	Himpunan pasangan (i_1, i_2) di mana <i>task</i> i_1 dan i_2 tidak memiliki hubungan prioritas langsung atau tidak langsung
t_{ism}	Waktu proses <i>task</i> i jika dilakukan oleh sumber daya s untuk model m
c_{I1}	Biaya investasi untuk sumber daya manusia (rekrutmen, pelatihan, dan lain-lain)
c_{I2}	Biaya investasi untuk robot
c_{O1}	Biaya operasional per satuan waktu untuk operator manusia
c_{O2}	Biaya operasional per satuan waktu untuk robot
b_{2I}	Kualitas dan atau keuntungan terkait ergonomis (dalam mata uang seperti Rupiah), di seluruh horizon perencanaan, untuk menggunakan robot dalam <i>task</i> i
b_{3I}	Kualitas dan atau keuntungan terkait ergonomis (dalam mata uang seperti Rupiah), di seluruh horizon perencanaan, untuk menggunakan kolaborasi manusia dan robot dalam <i>task</i> i
τ	<i>Cycle time</i> atau <i>takt time</i> yang diperlukan dalam lintas perakitan
ψ	Suatu bilangan positif yang sangat besar
W_{jm}	Subset dari semua <i>tasks</i> yang dapat ditugaskan ke stasiun j untuk model m
$\ W_{jm}\ $	Jumlah <i>task</i> dalam subset W_{jm}
M	Himpunan model produk; $M = \{1, 2, \dots, \ M\ \}$
$\ M\ $	Banyak model produk
D	Jumlah total permintaan dari semua model

Fungsi objektif pada model matematis ini adalah minimasi total biaya untuk investasi dan operasional lintas perakitan. Pada fungsi objektif dipertimbangkan keuntungan dalam Rupiah yang menjadi pengurang pada fungsi tujuan. Notasi untuk model matematis ini adalah sebagai berikut.

2.5.1 Fungsi Tujuan

$$TC = c_{I1} \cdot \alpha + c_{I2} \cdot \rho + (c_{O1} \cdot \alpha + c_{O2} \cdot \rho) \cdot D \cdot \tau - \sum_{i=1}^{n_T} \sum_{j=1}^{n_W} b_{2I} x_{ij2} - \sum_{i=1}^{n_T} \sum_{j=1}^{n_W} b_{3I} x_{ij3} \quad (2.1)$$

2.5.2 Fungsi Pembatas

$$\alpha = \sum_{j=1}^{n_W} \left[\min \left\{ 1, \left(\sum_{i=1}^{n_T} (x_{ij1} + x_{ij3}) \right) \right\} \right] \quad (2.2)$$

$$\rho = \sum_{j=1}^{n_W} \left[\min \left\{ 1, \left(\sum_{i=1}^{n_T} (x_{ij2} + x_{ij3}) \right) \right\} \right] \quad (2.3)$$

$$\sum_{j=1}^{n_W} \left(\sum_{s \in \{1,2,3\}} x_{ijs} \right) = 1 \quad (2.4)$$

$$\sum_{s \in \{1,2,3\}} x_{ijs} \leq \sum_{q=1}^j \left(\sum_{s \in \{1,2,3\}} x_{hqs} \right) \quad \forall j \in J, \forall h, i \in P \quad (2.5)$$

$$f_{im} \leq \tau \quad \forall i \in I, \forall m \in M \quad (2.6)$$

$$f_{im} \geq \sum_{j=1}^{n_W} \left(\sum_{s \in \{1,2,3\}} t_{ism} x_{ijs} \right) \quad \forall i \in I, \forall m \in M \quad (2.7)$$

$$\begin{aligned}
f_{im} - f_{hm} + \psi \left(1 - \sum_{s \in (1,2,3)} x_{ijs} \right) + \psi \left(1 - \sum_{s \in (1,2,3)} x_{hjs} \right) \\
\geq \sum_{j=1}^{nW} \left(\sum_{s \in (1,2,3)} t_{ism} x_{ijs} \right) \quad \forall j \in J, \forall h, i \in P
\end{aligned} \tag{2.8}$$

$$\begin{aligned}
f_{im} - f_{pm} + \psi \left(1 - \sum_{s \in (1,2,3)} x_{ijs} \right) + \psi \left(1 - \sum_{s \in (1,2,3)} x_{pjs} \right) \\
+ \psi_{z_{ip}} \geq \sum_{j=1}^{nW} \left(\sum_{s \in (1,2,3)} t_{ism} x_{ijs} \right)
\end{aligned} \tag{2.9}$$

$$\begin{aligned}
f_{pm} - f_{im} + \psi \left(1 - \sum_{s \in (1,2,3)} x_{ijs} \right) + \psi \left(1 - \sum_{s \in (1,2,3)} x_{pjs} \right) \\
+ \psi(1 - z_{ip}) \geq \sum_{j=1}^{nW} \left(\sum_{s \in (1,2,3)} t_{psm} x_{pjs} \right) \quad \forall j \\
\in J, \forall h, i \in Q
\end{aligned} \tag{2.10}$$

$$\sum_{i \in I} x_{ijs} - \|W_{jm}\| v_{jm} \leq 0 \quad \forall j \in J, \forall m \in M \tag{2.11}$$

$$\sum_{m \in M} v_{jm} - \|M\| U_j \leq 0 \quad \forall j \in J \tag{2.12}$$

$$U_j \leq U_{j-1} \quad j = 2, \dots, n_W \tag{2.13}$$

$$Di \text{ mana } U_j < \min \left\{ \sum_{i=1}^{n_r} \sum_{s \in (1,2,3)} x_{ijs} \right\} \quad \forall j \in J \tag{2.14}$$

$$x_{ijs} \in \{0, 1\} \quad \forall i \in I, j \in J, s = 1, 2, 3 \tag{2.15}$$

$$z_{ip} \in \{0, 1\} \quad \forall i \in I, p \in P \quad (2.16)$$

$$u_j \in \{0, 1\} \quad \forall j \in J \quad (2.17)$$

$$v_{jm} \in \{0, 1\} \quad \forall j \in J, m \in M \quad (2.18)$$

2.5.3 Keterangan Persamaan

Masing-masing fungsi tujuan dan pembatas yang ada memuat beberapa arti sebagai berikut.

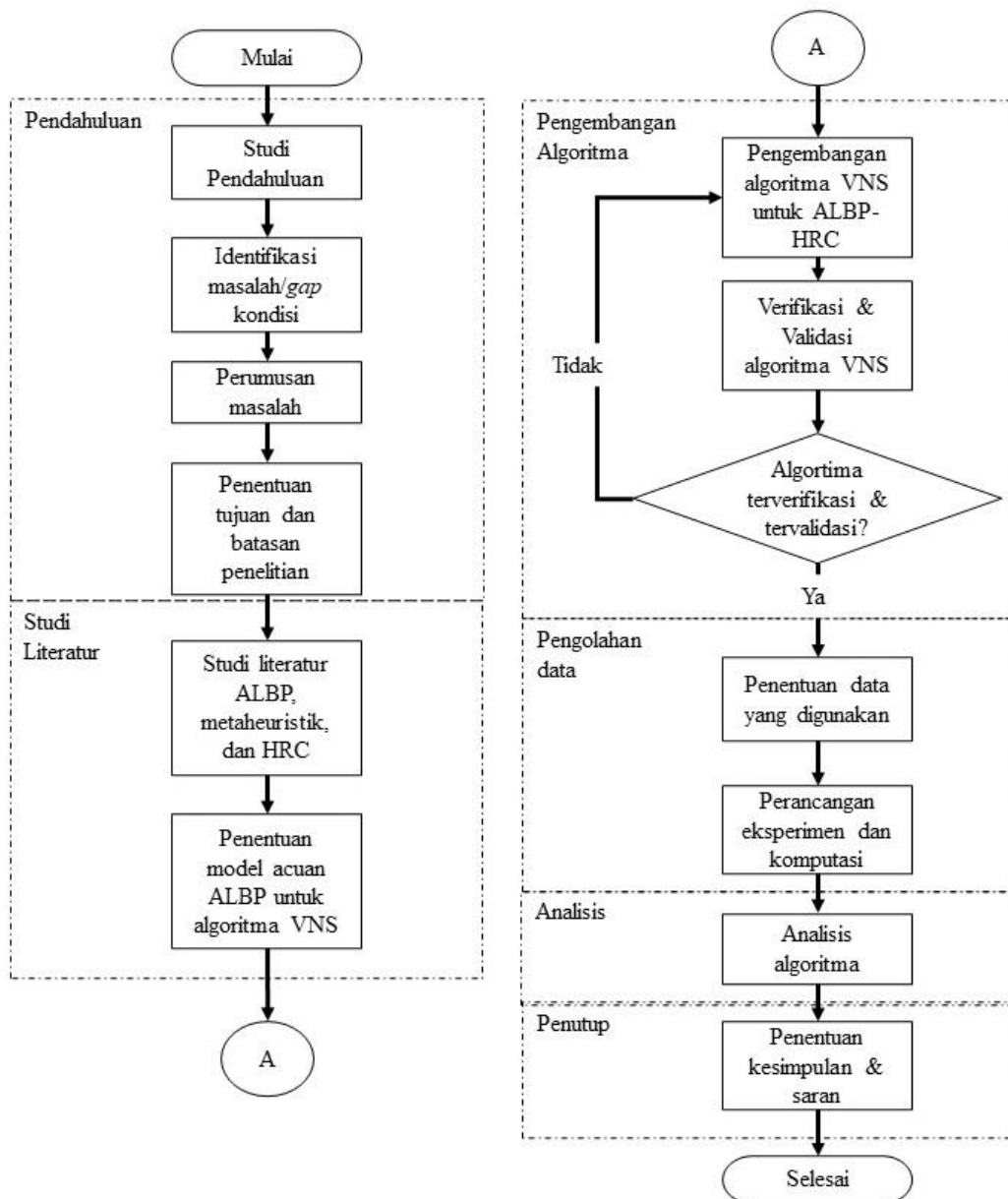
Tabel 6 Keterangan Persamaan

Persamaan	Keterangan
(2. 1)	Fungsi objektif: minimasi total biaya yang terdiri dari biaya operasional (biaya <i>resource</i> x <i>demand</i>) + biaya <i>investment</i> (sesuai jumlah <i>resource</i> -nya) - benefit
(2. 2)	Penjelasan lebih lanjut dari α dan ρ , untuk mendefinisikan setiap station bisa melakukan beberapa job tetapi hanya menggunakan 1 tipe <i>human resource</i> yang sama.
(2. 3)	Penjelasan lebih lanjut dari α dan ρ , untuk mendefinisikan setiap station bisa melakukan beberapa job tetapi hanya menggunakan 1 tipe <i>resource</i> robot yang sama.
(2. 4)	<i>Constraint</i> untuk <i>assignment</i> dari <i>task</i> , mengatur untuk setiap task di-assign tepat ke 1 stasiun dengan menggunakan 1 tipe <i>resource</i> .
(2. 5)	<i>Precedence constraint</i> untuk memastikan bahwa semua hubungan <i>precedence</i> antar <i>task</i> terpenuhi.
(2. 6) dan (2. 7)	<i>Cycle time constraint</i> untuk memastikan bahwa waktu selesai dari task untuk tiap model tidak melebihi <i>cycle time</i> .
(2. 8), (2. 9), dan (2. 10)	Mengatur waktu selesai tugas yang bergantung pada urutan untuk setiap model. Persamaan (2. 8) akan aktif apabila <i>task h</i> mendahului tepat sebelum <i>task i</i> dan kedua kegiatan dilakukan di stasiun <i>j</i> . Persamaan (2. 9) dan (2. 10) akan aktif saat kedua <i>task</i> (<i>i</i> dan <i>h</i>) tidak memiliki hubungan urut dan dilakukan di stasiun <i>j</i> .

Persamaan	Keterangan
(2. 11) dan (2. 12)	<p>Berfungsi untuk mengatur jumlah stasiun agar sama untuk setiap model produk.</p> <p>Pada persamaan (2. 11), jumlah <i>task</i> harus sama ataupun lebih besar dari jumlah x_{ij} sehingga dapat dipastikan bahwa jumlah stasiun mencukupi.</p> <p>Pada persamaan (2. 12), jumlah model harus sama atau lebih besar dari jumlah bilangan biner utilisasi stasiun j untuk model m (v_{jm}).</p>
(2. 13)	Memastikan bahwa <i>task</i> dilakukan dalam urutan yang benar dan mengharuskan utilisasi stasiun j hanya bisa berfungsi apabila stasiun j sebelumnya juga berfungsi.
(2. 14)	Memastikan bahwa <i>task</i> dilakukan dalam urutan yang benar dan bilangan biner U_j harus lebih kecil atau sama dengan nilai minimal antara 1 dan jumlah bilangan biner x_{ijs}
(2. 15)	Persamaan (2. 15), (2. 16), (2. 17), dan (2. 18) menunjukkan sifat x_{ijs} , z_{ip} , u_j , dan v_{jm} sebagai variabel keputusan bilangan biner, dimana hanya bisa bernilai 0 atau 1.

BAB III

METODOLOGI PENELITIAN



Gambar 2 Metodologi Penelitian

Penelitian pengembangan algoritma metaheuristik VNS ini dilakukan dalam tujuh tahapan seperti pada Gambar 2. Metodologi menggambarkan seluruh proses mulai dari tahap pendahuluan & identifikasi masalah, studi pustaka & pemilihan model acuan, pengembangan algoritma *variable neighborhood search*, menentukan data yang digunakan untuk penelitian, analisis algoritma dan keluarannya, dan penutup.

3.1 Pendahuluan

Pendahuluan adalah tahapan paling awal dalam penelitian. Pada tahapan ini dilakukan studi pendahuluan mengenai kondisi riil yang ada di mana akan diidentifikasi *gap* masalah antara kondisi riil dengan kondisi idealnya. Kemudian dari identifikasi ditetapkan perumusan masalah, tujuan penelitian, batasan penelitian, beserta asumsi-asumsi penelitian untuk menentukan ruang lingkup permasalahan yang diteliti.

3.1.1 Studi Pendahuluan

Studi pendahuluan bertujuan untuk mencari tahu lebih lanjut mengenai kondisi riil yang ada dan kesenjangan antara kondisi riil dan kondisi ideal yang akan diangkat untuk menjadi masalah utama yang akan diselesaikan. Hasil dari studi pendahuluan ini adalah didapatkan gambaran sistem dari permasalahan yang akan diselesaikan secara jelas. Penelitian mengacu kepada model penelitian (Susanto & Nugraha, 2021) yang berjudul “Mixed Model Assembly Line Balancing for Human-Robot Shared Tasks”. Pada penelitian ini dilakukan alokasi tugas dan sumber daya pada stasiun kerja dengan model analitis dan fungsi tujuan minimalisasi total biaya.

3.1.2 Identifikasi Masalah/*Gap* Kondisi

Pada penelitian (Susanto & Nugraha, 2021), didapat bahwa waktu komputasi sudah mulai mengindikasikan cukup memakan waktu yang lama dimulai ketika *task* yang dialokasikan sebanyak lima. Sistem dari permasalahan yang akan diangkat adalah penerapan kolaborasi manusia-robot pada masalah lintas perakitan bersifat *mixed-product* dengan menerapkan algoritma metaheuristik di mana tujuannya untuk meminimalkan total biaya dan waktu komputasi.

3.1.3 Perumusan Masalah

Masalah yang diangkat di penelitian ini berdasar pada perkembangan industri 4.0 di mana mulai berkembang penggunaan *collaborative robot* pada lintas perakitan. Masalah penelitian ini juga berdasar penelitian (Susanto & Nugraha, 2021), didapat bahwa waktu komputasi sudah mulai mengindikasikan cukup memakan waktu yang lama dimulai ketika *task* yang dialokasikan sebanyak lima di mana kondisi riilnya jumlah *task* bisa lebih dari lima. Dari masalah tersebut dilakukan pengembangan algoritma untuk mendapatkan solusi yang efektif dengan waktu komputasi yang efisien menggunakan algoritma metaheuristik.

3.1.4 Penentuan Tujuan dan Batasan Penelitian

Tujuan penelitian berdasar pada latar belakang dan rumusan masalah yang telah ditetapkan. Tujuan dari penelitian ini adalah pengembangan algoritma VNS untuk lintas perakitan *mixed model* yang meminimisasi biaya untuk kasus ALBP-HRC dan uji coba beserta analisis algoritma VNS yang telah dikembangkan.

Penelitian ini juga memerlukan batasan untuk membatasi gerak dan ruang lingkup penelitian agar penelitian memiliki arah yang jelas. Batasan penelitian diperlukan agar penelitian berfokus pada aspek-aspek yang relevan dan memiliki prosedur penyelesaian masalah yang tepat. Fokus penelitian ini adalah lintas perakitan *multi-product* dan alternatif sumber daya adalah manusia, robot, atau gabungan keduanya.

3.2 Studi Literatur

Tahap kedua adalah studi literatur. Tahapan ini mencari sumber referensi yang sesuai dengan permasalahan yang akan diselesaikan. Studi literatur yang dicari berfokus kepada konsep lintas perakitan, *collaborative robot*, metaheuristic & *variable neighborhood search*, dan model analitis acuan.

3.2.1 Studi Literatur ALBP, Metaheuristik, dan HRC

Tahapan studi literatur adalah pencarian sumber referensi untuk masalah ALBP-HRC dan metaheuristik untuk algoritma VNS. Sumber referensi berasal dari paper, buku, atau penelitian lainnya yang terkait. Studi literatur akan memberikan pemahaman menyeluruh terkait algoritma VNS dalam bentuk tahap-tahapan umum dan *flowchart* yang akan menjadi acuan atau dasar untuk tahap-tahap selanjutnya, yaitu pengembangan algoritma yang berfokus untuk kasus spesifik ALBP-HRC.

3.2.2 Penentuan Model Acuan ALBP untuk Algoritma VNS

Pemilihan model acuan berdasar pada batasan penelitian yang telah ditetapkan. Model acuan berdasar pada penelitian (Susanto & Nugraha, 2021) yang berfokus untuk ALBP tipe lintas perakitan *mixed-model*. Dari dasar model analitis ini akan dicoba untuk dikembangkan dengan algoritma VNS.

3.3 Pengembangan Algoritma

Tahapan selanjutnya adalah pengembangan algoritma metaheuristik VNS untuk kasus ALBP-HRC di mana lintas perakitan bersifat *mixed-model*. Setelah itu, dilakukan validasi terhadap algoritma yang sudah dikembangkan. Apabila tervalidasi, maka dilanjutkan ke tahap

pengumpulan dan pengolahan data. Namun, apabila tidak valid, maka kembali pada proses pengembangan algoritma VNS tersebut.

3.3.1 Pengembangan Algoritma VNS untuk ALBP-HRC

Pengembangan algoritma VNS untuk ALBP-HRC dilakukan dengan pembuatan diagram alir algoritma VNS secara umum yang menyesuaikan dengan kasus spesifik ALBP-HRC dan model acuan yang telah dikembangkan (Susanto & Nugraha, 2021). Diagram alir ini membantu pengembangan kode untuk program dan memastikan kode tersebut sesuai dengan permasalahan yang akan diselesaikan. Pengembangan ini mengacu pada *flowchart* di Gambar 1 dan tahapan di subbab 2.4.2 Prosedur algoritma *Variable Neighborhood Search* (VNS).

Selain pengembangan tersebut, terdapat pengembangan tambahan untuk representasi permasalahan ALBP-HRC. Pengembangan tersebut terdapat pada

Pengembangan	Sumber
Penambahan <i>outer loop</i> .	(Hansen & Nenad, Variable neighborhood search: basics and variants, 2016)
Penambahan operator untuk pengurangan jumlah stasiun kerja.	(Loranca, Pinto, & Olivares-Benitez, 2013)
Penambahan operator penukaran alternatif operator tugas.	Akan dikembangkan di penelitian ini
Penambahan parameter untuk menentukan tingkat perubahan solusi yang dibangkitkan.	Akan dikembangkan di penelitian ini

3.3.2 Verifikasi dan Validasi Algoritma VNS

Setelah pengembangan algoritma VNS untuk penyeimbangan lintas perakitan, selanjutnya dilakukan verifikasi dan validasi program. Verifikasi program menggunakan data hipotetis yang berasal dari penelitian model acuan. Jika solusi hasil program tidak melanggar fungsi pembatas, maka program untuk algoritma VNS terverifikasi. Validasi program algoritma VNS dilakukan dengan perbandingan antara fungsi tujuan & pembatas dari model acuan dengan algoritma VNS untuk penyeimbangan lintas perakitan cobot yang telah dikembangkan apakah memiliki kesamaan satu sama lain.

3.4 Pengumpulan dan Pengolahan Data

Setelah verifikasi dan validasi program, dilakukan pengumpulan dan pengolahan data. Tahapan ini dimulai dengan penentuan data yang akan digunakan dalam penelitian di mana data merupakan data hipotesis. Dari data tersebut dilakukan pencarian solusi menggunakan kode program Python algoritma VNS ALBP-HRC yang telah dikembangkan.

3.4.1 Penentuan Data yang akan Digunakan

Data yang digunakan bukan dari kasus riil, melainkan data hipotesis dari internet. Sumber data berasal dari internet penelitian khusus untuk masalah *assembly line balancing*. Data yang dipilih sama dengan data yang digunakan dalam model acuan dengan tujuan hasil penelitian bisa dibandingkan dengan model acuan yang menjadi pembanding terdekat.

3.4.2 Perancangan Eksperimen dan Komputasi

Setelah penentuan data yang digunakan, dilakukan perancangan eksperimen dengan mengubah parameter-parameter algoritma. Pengubahan parameter dilakukan untuk mengetahui efek dari perubahan parameter terhadap solusi yang dihasilkan. Selain pengamatan solusi, dapat diketahui juga nilai parameter yang baik untuk digunakan dalam proses komputasi sehingga dari nilai tersebut bisa didapat total biaya yang minimal. Setelah perancangan eksperimen, dilakukan pencarian solusi dengan algoritma yang sudah dikembangkan menggunakan data hipotesis atau data yang sudah ditentukan.

3.5 Analisis Algoritma

Setelah pemerolehan solusi, dilakukan analisis terhadap solusi dan perilaku algoritma VNS yang sudah dikembangkan. Analisis solusi dilakukan terhadap solusi yang dihasilkan dari program algoritma VNS ALBP-HRC untuk mengetahui karakteristik solusi yang dihasilkan berdasarkan data parameter yang berbeda. Analisis perilaku algoritma dengan perubahan parameter algoritma bertujuan untuk mengetahui bagaimana perubahan keluaran algoritma terhadap besaran atau jumlah parameter. Selain analisis solusi dan perilaku algoritma, dilakukan perbandingan solusi hasil algoritma VNS yang sudah dikembangkan dengan model matematis acuan baik dari solusi itu sendiri, waktu komputasi, dan akomodasi jumlah parameter dalam perhitungan.

3.6 Penentuan Kesimpulan & Saran

Setelah dilakukan analisis terhadap algoritma VNS, ditentukan kesimpulan dari hasil penelitian berdasarkan tujuan awal. Selain itu juga ditentukan saran-saran yang dapat dilakukan pada penelitian selanjutnya yang sejenis untuk perbaikan lebih baik.

BAB IV

PENGUMPULAN DAN PENGOLAHAN DATA

4.1 Rencana Penelitian

Sesuai dengan metodologi yang sudah dijabarkan di bab sebelumnya, aktivitas penelitian terbagi ke dalam enam aktivitas utama. Dimulai dari pengembangan algoritma VNS setelah mendapatkan berbagai sumber literatur untuk kasus ALBP-HRC selama dua bulan. Setelah itu dilakukan pengolahan data dan juga perancangan eksperimen terhadap algoritma dan pelaksanaannya. Hasil eksperimen akan dianalisis untuk mengetahui karakteristik dari algoritma dan program yang sudah dikembangkan. Terakhir hasil penelitian dicatat dalam laporan. Keseluruhan aktivitas memakan waktu sekitar enam bulan. Berikut adalah *gantt chart* penelitian ini.

Tabel 7 *Gantt Chart* Penelitian

Aktivitas	Januari 2022	Februari 2022	Maret 2022	April 2022	Mei 2022	Juni 2022
Pengembangan algoritma VNS						
Pengolahan data eksperimen						
Perancangan eksperimen						
Pelaksanaan eksperimen algoritma						
Analisis algoritma						
Melengkapi laporan						

DAFTAR PUSTAKA

- Becker, C., & Scholl, A. (2004). A Survey on Problems and Methods in Generalized Assembly Line Balancing. *European Journal of Operational Research*.
- Desale, S., Rasool, A., Andhale, S., & Rane, P. (2015). Heuristic and Meta-Heuristic Algorithms and Their Relevance to the Real World: A Survey. *INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING IN RESEARCH TRENDS*.
- Fogli, D., Gargioni, L., Guida, G., & Fabio Tampalini. (2021). A Hybrid Approach to User-Oriented Programming of Collaborative Robots. *Robotics and Computer-Integrated Manufacturing*.
- Franklin, C. S., & Dominguez, E. G. (2020). Collaborative robotics: New era of human–robot cooperation in the workplace. *Safety Research*.
- Franklin, C. S., Dominguez, E. G., Fryman, J. D., & Lewandowski, M. L. (2020). Collaborative Robotics: New Era of Human–Robot Cooperation in the Workplace. *Journal of Safety Research*.
- Hansen, P., & Mladenovic, N. (1998). Variable Neighborhood Search. *Computers & Operations Research*.
- Knudsen, M. S. (2020). Collaborative Robots: Frontiers of Current Literature. *Journal of Intelligent Systems Theory and Applications*.
- Matúšová, M. (2019). The Future of Industry with Collaborative Robots. *MATEC Web of Conferences*.
- Putri, F. S. (2021). *Pengembangan Algoritma Metaheuristik Variable Neighbourhood Search Untuk Lintasan Perakitan Kolaborasi Manusia dan Robot*. Bandung: Institut Teknologi Bandung.
- Saif, U. (2014). A survey on assembly lines and its types. *Frontier Mechanical Engineering*.
- Sharma, A. K., & Bandari, R. (2021). A Study of Trends and Industrial Prospects of Industry 4.0. *Materials Today*.
- Sorensen, K., Sevaux, M., & Glover, F. (2013). A History of Metaheuristics. *Encyclopedia of Operations Research and Management Science*.

Susanto, & Nugraha, C. (2021). Mixed Model Assembly Line Balancing for Human-Robot Shared Tasks. *Bandung Institute of Technology*.