

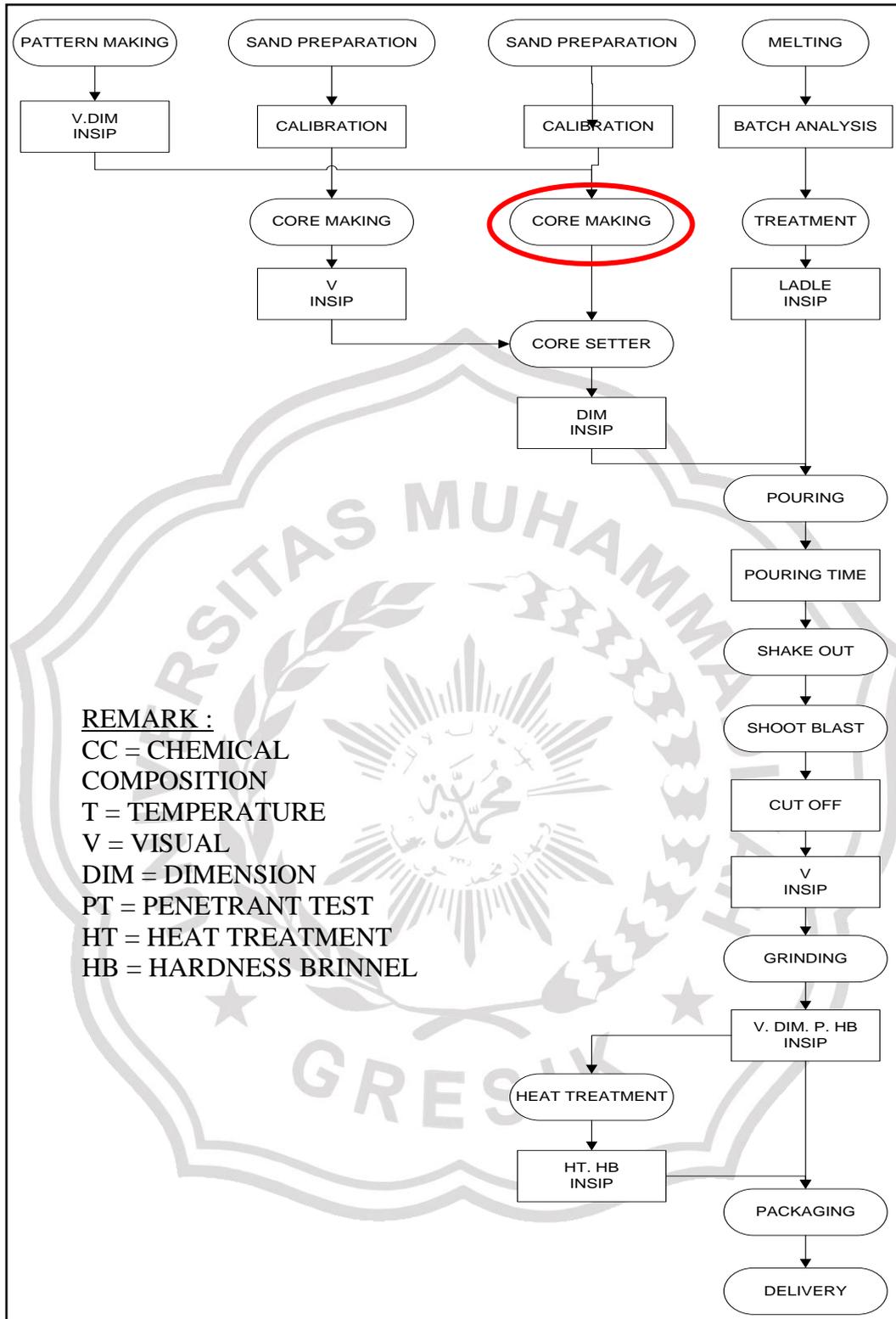
BAB II TINJAUAN PUSTAKA

2.1 Proses Produksi

Proses produksi adalah proses transformasi atau sekumpulan aktivitas yang mengubah suatu kumpulan masukan (sumber daya manusia, material, energi, informasi, dan lain-lain) menjadi produk keluaran (produk jadi atau *services*) yang mempunyai nilai tambah. (Sultan, 2007)

Berikut ini merupakan alur pada proses produksi boggie yang terdapat pada divisi *foundry* PT. Barata Indonesia (Persero) :





Sumber : PT.Barata Indonesia (Persero)

Gambar 2. 1 Flow Process Pada Proses Produksi Boggie

Sistem manufaktur meliputi proses dari bahan baku sampai menjadi produk jadi melalui serangkaian operasi. Operasi-operasi ini meliputi kombinasi dari personil dan peralatan dengan tingkat otomasi yang bermacam-macam. (Sultan, 2007)

Proses manufaktur dapat dibagi menjadi dua jenis proses utama yaitu operasi proses (*processing operations*) dan operasi perakitan (*assembly operations*). Operasi proses adalah aktivitas mengubah material kerja dari satu bentuk menjadi bentuk lain yang berupa *part* atau produk, sedangkan operasi perakitan adalah aktivitas menggabungkan dua atau lebih komponen menjadi *part* atau produk. (Sultan, 2007)

Operasi proses dengan bahan metal/logam, kaca, dan keramik dapat dibagi atas kategori sebagai berikut : (Sultan 2007)

a. *Shaping operations*

Shaping operations adalah operasi pembentukan dengan menerapkan gaya mekanis, panas atau bentuk energi lain dalam rangka mengubah ukuran (*geometry*) benda kerja. *Shaping operations* dapat diklasifikasikan sebagai berikut:

1. *Solidifications process*, yaitu operasi pembekuan dengan bentuk awal benda kerja adalah cairan atau semi fluida, kemudian dialirkan ke dalam rongga cetakan dengan cara dituang baik dengan tekanan atau tanpa tekanan. Di dalam rongga cetakan, benda kerja menjadi dingin dan membeku mengikuti bentuk rongga cetakannya. Proses ini dikenal dengan nama *casting* untuk logam dan *molding* untuk kaca.
2. *Particulate processing*, yaitu operasi pembentukan dengan bentuk awal benda kerja adalah bubuk. Teknik yang umum digunakan adalah dengan menekan bubuk dengan *dies* berongga dibawah tekanan yang tinggi. Untuk meningkatkan kekuatan, komponen yang sudah dibentuk dipanaskan sampai temperatur dibawah titik lelehnya, sehingga partikel-partikel saling mengikat satu sama lain. Baik logam (*powder metalurgy*) dan keramik dapat dibentuk dengan proses ini.

3. *Deformation process*, operasi pembentukan dengan bentuk awal benda kerja adalah logam ulet (*ductile metal*) yang dibentuk dengan memberikan tegangan yang melebihi kekuatan luluh benda kerja. Proses ini mencakup *forging*, *extrusion* dan *rolling*, juga termasuk didalamnya adalah proses *sheet* metal seperti *drawing*, *forming* dan *bending*.
 4. *Material removal process*, operasi pembuangan material dengan bentuk awal benda kerja adalah logam padat (baik *ductile* ataupun *brittle metal*). Termasuk operasi ini adalah proses pemesian seperti *turning*, *drilling* dan *milling*. Bentuk lain dari proses ini adalah *grinding*, serta proses non tradisional dengan pemotongan menggunakan laser, *electron beams*, *chemical erosion*, *electric discharge* atau energi *electromechanical*.
- b. *Property enhancing operations*
- Property enhancing operations* adalah operasi peningkatan sifat mekanis atau fisik dari benda kerja. Proses ini tidak mengubah ukuran benda kerja kecuali pada beberapa kasus berupa penyusutan. Proses ini mencakup:
1. *Heat treatment*, yaitu operasi perlakuan panas dengan berbagai variasi temperatur dengan tujuan untuk meningkatkan kekuatan, keuletan ketangguhan atau kekerasan benda kerja.
 2. *Sintering*, yaitu operasi perlakuan panas untuk bubuk logam atau keramik yang sudah melewati proses pembentukan.
- c. *Surface processing operations*
- Surface processing operations* adalah operasi pemrosesan permukaan benda kerja, proses ini mencakup :
1. *Cleaning*, yaitu operasi pembersihan permukaan benda kerja untuk menghilangkan kotoran, minyak, karat dan kontaminasi permukaan lainnya. Pembersihan ini mencakup *chemical* dan *mechanical processes*.
 2. *Surface treatments*, yaitu operasi ini mencakup pengerjaan mekanis seperti *shot peening* dan *sand blasting*, dan proses fisik seperti *diffusion* dan implantasi ion.

3. *Coating and thin film deposition*, yaitu operasi pelapisan permukaan benda kerja. Proses pelapisan meliputi *electroplating*, *anodizing aluminium*, dan *organic coating* (pengecatan), sedangkan proses deposit lapisan tipis meliputi *physical vapor deposition* dan *chemical vapor deposition*.

Operasi perakitan dapat dibagi atas kategori :

- a. *Permanently joining processes*, adalah operasi perakitan dari dua komponen atau lebih dengan sambungan permanen sehingga tidak bias dibuka tanpa merusak produk yang telah disambung. Proses ini mencakup welding, brazing, soldering dan adhesive bonding.
- b. *Semi permanently joining processes*; adalah operasi perakitan dengan sambungan semi permanen sehingga bisa lebih mudah dibuka. Biasanya digunakan sambungan baut dan mur atau sekrup.

2.1.1 Tujuan Produksi

Berikut ini adalah beberapa tujuan produksi dari suatu kegiatan usaha :
(Taslim, 2015)

- a. Memenuhi kebutuhan manusia. Manusia memiliki beragam kebutuhan terhadap barang dan jasa yang harus terpenuhi dengan kegiatan aktivitas produksi, ditambah lagi dengan jumlah manusia semakin bertambah dari masa ke masa.
- b. Mencari keuntungan atau laba. Dengan memproduksi barang dan jasa, produsen (orang yang memproduksi) berharap bias menjual produk dan jasa dengan memperoleh laba atau keuntungan.
- c. Menjaga keberlangsungan hidup perusahaan. Adanya proses penjualan oleh produsen dan konsumen dengan laba atau keuntungan maka keuntungan tersebut digunakan kembali untuk keberlangsungan hidup perusahaan seperti, pembelian material produksi, upah karyawan, dan lain sebagainya.
- d. Meningkatkan mutu dan jumlah produksi. Produsen selalu berusaha untuk memenuhi keinginan dan kebutuhan para konsumen. Dengan adanya proses produksi secara berkelanjutan maka produsen berkesempatan untuk uji coba

atau eksperimen terhadap peningkatan kualitas produk serta jumlah yang diproduksi.

- e. Mengganti barang yang sudah rusak terpakai ataupun karena terkena bencana alam. Mengganti barang tersebut dengan cara memproduksi barang yang baru.
- f. Memenuhi kebutuhan pasar, baik dalam maupun luar negeri.
- g. Memperluas lapangan usaha.

2.1.2 Bidang-bidang Produksi

Dalam bidang produksi dapat dikelompokkan menjadi beberapa sebagai berikut : (Taslim, 2015)

- a. Produksi ekstraktif, adalah produksi yang memungut langsung hasil yang disediakan alam tanpa melakukan pengolahan lebih lanjut, seperti pertambangan, penangkapan ikan, dan lain-lain.
- b. Produksi agraris, adalah produksi yang mengolah sumber daya alam untuk memelihara hewan dan tanaman, seperti pertanian, perkebunan, peternakan, dan lain-lain.
- c. Produksi industri, adalah produksi yang mengelola beberapa bahan sebagai berikut :
 - 1) Bahan mentah menjadi barang jadi. Contohnya yaitu buah nanas yang diolah menjadi selai nanas.
 - 2) Bahan mentah menjadi barang setengah jadi. Contohnya yaitu kapas yang diolah menjadi benang pintalan.
 - 3) Bahan setengah jadi menjadi bahan setengah jadi. Contohnya yaitu pintalan benang yang diolah menjadi kain gulungan.
 - 4) Bahan setengah jadi menjadi barang jadi. Contohnya yaitu kain yang diolah menjadi seragam karyawan.
- d. Produksi perdagangan, adalah produksi yang mengumpulkan dan menjual kembali hasil produksi kepada yang memerlukan untuk memperoleh keuntungan, seperti supermarket, toko swalayan, kios, warung, dan lain-lain.
- e. Produksi jasa, adalah produksi yang membantu dan memperlancar proses produksi tanpa ikut membuat barang itu sendiri. Jadi, bidang produksi jasa

tidak menghasilkan barang atau produk, melainkan menghasilkan pelayanan jasa. Adapun jenis-jenis dari perusahaan jasa, sebagai berikut :

- 1) Jasa bisnis, seperti bank, konsultan, dan lain-lain.
- 2) Jasa perdagangan, seperti supermarket, toko swalayan, kios, usaha perawatan, perbaikan, serta pembersihan, dan lain-lain.
- 3) Jasa infrastruktur, seperti jasa konstruksi, jasa komunikasi dan transportasi, dan lain-lain.
- 4) Jasa sosial atau personal, seperti restoran dan kesehatan.
- 5) Jasa administrasi publik, seperti pendidikan dan pemerintahan.

2.1.3 Tingkatan Produksi

Produksi dapat dibagi dalam beberapa tingkat atau tahap sebagai berikut :
(Taslim, 2015)

- a. Produksi primer, adalah produksi yang menghasilkan bahan-bahan dasar yang bisa langsung dikonsumsi atau yang akan digunakan dalam proses produksi selanjutnya. Bidang produksi ekstraktif dan agraris merupakan produksi tingkat primer.
- b. Produksi sekunder, adalah produksi yang mengolah bahan-bahan dasar yang dihasilkan oleh tingkat produksi primer. Bidang produksi industri merupakan produksi tingkat sekunder.
- c. Produksi tersier, adalah produksi yang bersifat memperlancar proses produksi dan menyalurkan hasil produksi. Bidang produksi perdagangan dan jasa merupakan produksi tingkat tersier.

2.2 Pengukuran Waktu

Studi gerakan dan pengukuran waktu dilakukan untuk memperoleh berbagai macam rancangan dan berbagai sistem kerja sehingga dapat diperoleh rancangan kerja terbaik. Tenaga kerja manusia merupakan salah satu faktor penting pada suatu industri. (Zulaeha, 2016)

Pengukuran waktu adalah metode penetapan keseimbangan antara jalur manusia yang dikontribusikan dengan unit output yang dihasilkan. Pengukuran

waktu akan selalu berhubungan dengan usaha-usaha untuk menetapkan waktu baku yang dibutuhkan guna menyelesaikan suatu pekerjaan. (Herman, 2018)

Menurut Iftikar Z. Sitalaksana dkk. (2003), pengukuran waktu baku dibagi ke dalam dua bagian sebagai berikut :

a. Pengukuran secara langsung

Pengukuran secara langsung adalah pengukuran dilakukan di tempat dimana pengukuran tersebut dilaksanakan seperti cara jam henti (*stopwatch*) dan sampling pekerjaan (*work sampling*). (Henry, 2011)

b. Pengukuran tidak langsung

Pengukuran tidak langsung dilakukan tanpa harus berada di tempat pekerjaan. Cara tersebut dilakukan dengan membaca tabel-tabel yang tersedia asalkan mengetahui jalannya pekerjaan melalui elemen-elemen pekerjaan atau gerakan seperti data waktu baku atau data waktu gerakan. (Henry, 2011)

Menurut Sitalaksana et al., (1979), adapun tahapan penelitian yang dilakukan untuk menentukan pengukuran waktu pekerjaan sehingga memberikan hasil yang lebih optimal. Tahapan dalam pengukuran pendahuluan adalah sebagai berikut :

- 1) Mengelompokkan proses pengukuran didalam beberapa subgrup
- 2) Menghitung rata-rata dari rata-rata subgroup

$$\bar{x} = \frac{\sum x_i}{N}$$

Keterangan :

\bar{x} = rata-rata

$\sum x_i$ = jumlah pengamatan yang dilakukan

N = banyaknya pengamatan yang dilakukan

3) Uji keseragaman data

$$\sigma = \sqrt{\frac{\sum (x_i - \bar{x})^2}{N - 1}}$$

Keterangan :

N = banyaknya pengamatan yang telah dilakukan

x_i = waktu penyelesaian yang teramati selama pengukuran
pendahuluan yang telah dilakukan

\bar{x} = rata – rata

Kemudian dilakukan perhitungan batas kontrol atas (BKA) dan
batas kontrol bawah (BKB) dengan menggunakan rumus berikut :
(Sutalaksana, 1979)

$$BKA = \bar{x} + k\sigma$$

$$BKB = \bar{x} - k\sigma$$

4) Uji kecukupan data

Banyaknya pengamatan yang harus dilaksanakan dipengaruhi
oleh 2 faktor, yaitu : (Wignjosoebroto, 2006)

- a) Tingkat kepercayaan (*Confidence Level*)
- b) Tingkat ketelitian (*Degree of Accuracy*)

Persamaan ini menggunakan tingkat ketelitian 5% dan tingkat
kepercayaan 95%.

$$N' = \left[\frac{40\sqrt{(N\sum(x_i)^2) - (\sum x_i)^2}}{\sum x_i} \right]^2$$

Keterangan :

N' = jumlah pengamatan yang harus dilakukan

N = banyaknya pengamatan yang telah dilakukan

$\sum x_i$ = jumlah pengamatan yang dilakukan

k = tingkat keyakinan

Dimana,

$k=3$, jika tingkat keyakinan 99%

$k=2$, jika tingkat keyakinan 95%

$k=1$, jika tingkat keyakinan 68%

Apabila data yang diambil didapat $N' \leq N$ maka data sudah
cukup, jika $N' > N$ maka perlu tambahan data sebanyak selisih antara
 N' dengan N ($N' - N$).

2.3 Line Balancing

Keseimbangan lintasan didalam produksi merupakan hal yang sangat penting yang harus diperhatikan karena akan menentukan aspek-aspek lain dari sistem produksi dalam jangka waktu yang cukup lama. Beberapa aspek yang terpengaruh antara lain biaya, keuntungan, tenaga kerja, peralatan, dan sebagainya. Keseimbangan lintasan ini digunakan untuk mendapatkan lintasan perakitan yang memenuhi tingkat produksi tertentu. *Line balancing* merupakan penyiimbangan penugasan elemen tugas dari suatu *assembly line* ke *work stations* untuk meminimumkan banyaknya *work stations* dan meminimumkan total idle time pada semua stasiun untuk tingkat output tertentu. (Panudju, 2018)

Lini produksi adalah penempatan area-area kerja dimana operasi-operasi diatur secara berturut-turut dan material bergerak secara kontinu melalui operasi yang terangkai dengan seimbang. Menurut karakteristik proses produksinya, lini produksi dibagi menjadi dua : (Panudju, 2018)

- a. Lini fabrikasi, merupakan lintasan produksi yang terdiri atas sejumlah operasi pekerjaan yang bersifat membentuk atau mengubah bentuk benda kerja.
- b. Lini perakitan, merupakan lintasan produksi yang terdiri atas sejumlah operasi perakitan yang dikerjakan pada beberapa stasiun kerja dan digabungkan menjadi benda *assembly* atau *subassembly*.

Beberapa keuntungan yang dapat diperoleh dari perencanaan lini produksi yang baik adalah sebagai berikut: (Panudju, 2018)

- a. Jarak perpindahan material yang minim diperoleh dengan mengatur susunan dan tempat kerja
- b. Alirannya diukur dengan kecepatan produksi dan bukan oleh sejumlah spesifik
- c. Pembagian tugas terbagi secara merata yang disesuaikan dengan keahlian masing-masing pekerjaan, sehingga pemanfaatan tenaga kerja lebih efisien.
- d. Pengerjaan operasi yang serentak yaitu setiap operasi dikerjakan pada saat yang sama diseluruh lintasan produksi
- e. Gerakan benda kerja tetap sesuai dengan *set-up* dari lintasan dan bersifat tetap
- f. Proses memerlukan waktu yang minimum

Prosedur umum untuk menempatkan tugas dalam setiap stasiun kerja adalah sebagai berikut : (Panudju, 2018)

- a. Mengidentifikasi daftar utama tugas
- b. Menghilangkan tugas-tugas yang memiliki hubungan prioritas yang tidak dapat dipenuhi
- c. Menggunakan salah satu teknik heuristik untuk menentukan aturan penempatan tugas pada stasiun kerja, heuristik penyeimbang lini tersebut antara lain : (Panudju, 2018)
 - 1) Pilihlah tugas dengan waktu pengerjaan terpanjang
 - 2) Tugas yang paling banyak diikuti oleh tugas lain
 - 3) Pemingkatan bobot posisi
 - 4) Waktu pekerjaan terpendek
 - 5) Tugas dengan jumlah yang mengikutinya paling sedikit

Untuk menentukan jumlah stasiun kerja yang dibutuhkan, diperlukan data sebagai berikut : (Panudju, 2018)

- a. Jumlah elemen pekerjaan atau tugas serta hubungan antar pekerjaan (presedensi tugas).
- b. Waktu untuk penyelesaian setiap tugas atau pekerjaan.
- c. Jumlah target *output* produksi yang ingin dihasilkan.
- d. Waktu operasi setiap hari atau setiap *shift*.

Line balancing dapat meningkatkan efisiensi pada proses dengan meminimalisir stasiun kerja, meminimalisir waktu siklus kerja, memaksimalkan beban kerja, dan meningkatkan fleksibilitas antar stasiun kerja. *Line balancing* merupakan salah satu alat yang efektif untuk memperbaiki output dari suatu *line* atau proses melalui penurunan aktivitas yang tidak bernilai tambah dan penurunan waktu siklus kerja. (Fardiansyah, 2018)

Pengaturan dan perencanaan yang tidak tepat mengakibatkan setiap stasiun kerja di lintas perakitan mempunyai kecepatan produksi yang berbeda. Akibat selanjutnya adalah terjadi penumpukan material di antara stasiun kerja yang tidak seimbang kecepatannya. (Henry, 2011)

Masalah *line balancing* terdiri dari penyeimbangan operasi dalam waktu yang sama dan juga waktu yang diperlukan untuk memenuhi kecepatan produksi. Setiap perubahan kecepatan produksi terhadap waktu dari stasiun kerja dinamakan waktu keseimbangan atau waktu stasiun. (Henry, 2011)

Untuk mewujudkan *line balancing* pada suatu perusahaan maka faktor-faktor yang mempengaruhi yang mengakibatkan timbulnya kemacetan harus dicegah sedemikian rupa sehingga hasil setiap bagian dalam proses produksi bisa berjalan dengan lancar dalam waktu yang telah ditentukan. Faktor-faktor tersebut antara lain : (Puteri, 2016)

- a. Terlambatnya bahan baku
- b. Material handling yang kurang sempurna
- c. Terjadinya kerusakan mesin
- d. Bertumpuknya barang dalam proses pada tingkat proses tertentu
- e. Kondisi mesin yang sudah tua
- f. Kelemahan dalam merencanakan kapasitas mesin
- g. *Layout* yang kurang baik
- h. Kualitas tenaga kerja yang kurang baik
- i. Adanya *working condition* yang kurang baik

2.2.2.1 Tujuan *Line Balancing*

Dengan adanya persamaan kapasitas untuk setiap stasiun yang berbeda maka hasil yang diharapkan dari proses keseimbangan lintasan adalah : (Henry, 2011)

- a. Menghindari penumpukan barang dalam proses pada suatu bagian produksi.
- b. Menghindari pengangguran pada bagian produksi lainnya.
- c. Mendapatkan efisiensi sistem yang cukup tinggi.
- d. Memenuhi rencana produksi yang telah ditetapkan.

2.2.2.2 Terminologi *Line Balancing*

Berikut adalah beberapa terminologi yang ada pada *line balancing* : (Henry, 2011)

- a. Elemen kerja, adalah pekerjaan yang harus dilakukan dalam suatu kegiatan perakitan.
- b. Waktu operasi, adalah waktu standar untuk menyelesaikan suatu operasi.
- c. Stasiun kerja, adalah lokasi-lokasi tempat elemen kerja dikerjakan. Setelah menentukan interval waktu siklus, maka jumlah stasiun kerja yang efisien dapat ditetapkan sebagai berikut :

$$N_{min} = \frac{\text{Jumlah Waktu Total Semua Operasi}}{\text{Waktu Siklus Terpanjang}}$$

Jumlah nilai stasiun kerja harus berupa bilangan bulat atau integer. Jika nilai stasiun kerja tidak berupa bilangan bulat, maka pembulatan nilai stasiun kerja harus dibulatkan ke atas. Syarat membuat stasiun kerja :

- 1) Pengelompokkan elemen kerja berdasarkan rangking.
 - 2) Tidak boleh melanggar persyaratan precedence diagram, yaitu membuat stasiun kerja dengan melompati elemen kerja yang harus dilakukan terlebih dahulu.
 - 3) Waktu dari elemen kerja yang digabungkan tidak boleh lebih dari waktu siklus yang ditentukan.
- d. Waktu siklus merupakan waktu yang diperlukan untuk membuat satu unit produk.
 - e. Waktu stasiun kerja, adalah waktu yang dibutuhkan oleh sebuah stasiun kerja untuk mengerjakan semua elemen kerja yang didistribusikan pada stasiun kerja tersebut.
 - f. *Delay time / Idle time*, adalah selisih antara waktu siklus dengan waktu stasiun kerja. *Delay time* merupakan waktu menganggur yang terjadi setiap stasiun kerja. Biasanya *idle time* dapat dihitung dengan cara mengurangi waktu yang tersedia dengan waktu yang digunakan.
 - g. *Precendance diagram*, adalah diagram yang menggambarkan urutan dan keterikatan antar elemen kerja perakitan sebuah produk. Pendistribusian elemen kerja yang dilakukan untuk setiap stasin harus memperhatikan *precendance diagram*.

Untuk mengukur performansi sebelum dan sesudah dilakukan proses keseimbangan lintasan produksi dilakukan kriteria-kriteria berikut : (Henry, 2011)

a. Efisiensi Lintasan

Efisiensi lintasan adalah rasio antara waktu yang digunakan dengan waktu yang tersedia. Berkaitan dengan waktu yang tersedia, lintasan akan mencapai keseimbangan apabila setiap stasiun kerja mempunyai waktu yang relatif sama. Setelah diseimbangkan, maka lintasan perakitan berbentuk stasiun kerja yang terhubung secara seri. Pendistribusian elemen kerja yang ada membentuk stasiun kerja dilakukan berdasarkan waktu siklus.

Dalam melakukan perhitungan efisiensi lintasan dapat menggunakan perumusan sebagai berikut : (Groover, 2001)

$$E_b = \frac{T w_c}{w T_s}$$

Keterangan :

E_b = Efisiensi stasiun kerja

$T w_c$ = Total waktu setiap operasi

w = Banyaknya stasiun kerja

T_s = Waktu stasiun terpanjang

b. *Balance Delay*

Balance delay adalah rasio antara waktu *idle* dalam lintasan perakitan dengan waktu yang tersedia. Dalam melakukan perhitungan *balance delay* dapat menggunakan perumus sebagai berikut : (Groover, 2001)

$$d = \frac{(w T_s - T w_c)}{w T_s} \text{ atau } d = 1 - E_b$$

Dimana :

d = *balance delay*

$T w_c$ = Total waktu setiap operasi

w = Banyaknya stasiun kerja

T_s = Waktu stasiun terpanjang

E_b = Efisiensi stasiun kerja

c. Indeks Penghalusan (*Smoothing Index*)

Indeks penghalusan (*smoothing index*) adalah suatu indeks yang mempunyai kelancaran relatif dari penyeimbang lintasan perakitan tertentu. *Smoothing Index* dapat dihitung menggunakan perumusan sebagai berikut : (Henry, 2011)

$$\sqrt{\sum_{i=1}^n (ST_i \text{ max} - ST_i)^2}$$

Keterangan :

$ST_i \text{ max}$ = waktu maksimum di stasiun

ST_i = waktu stasiun di stasiun kerja ke-i

Nilai minimum dari *smoothing index* adalah 0, yang menandakan bahwa masing-masing stasiun kerja memiliki waktu yang relatif sama.

2.2.2.3 Metode *Line Balancing*

Untuk menyeimbangkan lintasan perakitan secara garis besar metode yang sering digunakan adalah sebagai berikut : (Henry, 2011)

a. Metode Heuristik

Metode ini menggunakan aturan-aturan yang logis dalam memecahkan masalah. Metode ini tidak menjamin hasil yang optimum, akan tetapi dirancang untuk menghasilkan strategi yang relatif lebih baik dan mendekati hasil yang optimum sesungguhnya. Beberapa metode heuristik yang dikenal antara lain : (Henry, 2011)

- 1) Metode pengurutan waktu terbesar (*largest candidate rule*)
- 2) Metode pendekatan daerah (*region approach*)
- 3) Metode bobot posisi peringkat (*ranked positional wight*) / RPW

Berikut ini adalah penjelasan mengenai metode heuristik :

- 1) Metode pengurutan waktu terbesar (*largest candidate rule*)

Merupakan metode yang paling sederhana. Adapun prosedur tersebut secara detail dapat dijelaskan sebagai berikut : (Rachman, 2015)

- a) Membuat *precedence diagram*
- b) Menentukan waktu siklus

- c) Menentukan jumlah stasiun kerja
- d) Urutkan stasiun kerja berdasarkan waktu baku dari terbesar ke terkecil
- e) Menghitung *efficiency* tiap-tiap lintasan atau *work station*, dan efisiensi rata-rata dari keseluruhan lintasan.

2) Metode pendekatan daerah (*region approach*)

Metode ini biasa juga disebut dengan metode pembagian wilayah, dimana dalam metode ini *precedence diagram* beserta elemen pekerjaannya dikelompokkan dalam sebuah kolom (wilayah). Kolom/wilayah paling kanan akan mendapat prioritas utama dalam pengaturan/pengelompokkan operasi-operasi ke dalam *work station*. Untuk lebih jelasnya langkah-langkah pemecahan masalah dengan metode ini adalah sebagai berikut : (Yulianti, 2018)

- a) Membuat precedence diagram
- b) Menentukan waktu siklus/*cycle time*
- c) Melakukan pengelompokkan elemen kerja terhadap kolom-kolom.
- d) Melakukan pengelompokkan elemen kerja tiap kolom terhadap *work station*.
- e) Mengalokasikan elemen-elemen kerja ke dalam *work station*.
- f) Menghitung *efficiency* tiap-tiap lintasan atau *work station*, dan efisiensi rata-rata dari keseluruhan lintasan.

3) Metode bobot posisi peringkat (*ranked positional wight*) / RPW

Metode Ranked Positional Weights (RPW) ini merupakan metode gabungan antara metode *Large Candidat Ruler* dengan metode *region approach*. nilai RPW merupakan perhitungan antara elemen kerja tersebut dengan posisi masing-masing elemen kerja dalam *precedence diagram*. (Purnamasari, 2015)

Langkah-langkah dari metode *Ranked Positional Weight* (RPW) adalah sebagai berikut : (Purnamasari, 2015)

- a) Membuat *precedence diagram* atau diagram jaringan kerja.
- b) Menghitung waktu siklus
- c) Membuat matiks lintasan berdasarkan *precedence diagram*.
- d) Menghitung bobot posisi tiap operasi yang dihitung berdasarkan jumlah waktu operasi tersebut dan operasi-operasi yang mengikutinya.
- e) Mengurutkan operasi-operasi mulai bobot operasi terbesar sampai dengan terkecil.
- f) Menghitung jumlah stasiun kerja minimum.
- g) Membuat *flow diagram* untuk stasiun kerja minimum tersebut lalu lakukan pembebanan operasi pada stasiun kerja mulai dari operasi dari bobot operasi terbesar sampai dengan terkecil, dengan kriteria total waktu operasi lebih kecil dari waktu siklus yang diinginkan.
- h) Melakukan *trial and error* untuk mendapatkan efisiensi lintasan yang paling tinggi.
- i) Menghitung *balance delay* lintasan dan *efficiency balancing*.

b. Metode Analitis

Metode dengan pendekatan sistematis yang memberikan solusi yang optimal tetapi memerlukan perhitungan yang besar dan rumit. (Henry, 2011)

Merupakan metode yang dapat menghasilkan suatu solusi optimal. Metode keseimbangan lini ini, mempunyai karakteristik dalam pemecahan masalah, adalah dalam pendekatan secara kuantitatif atau matematis. Umumnya pendekatan ini menggunakan *operation research* dalam mengoptimalkan lintasan, seperti penggunaan : (Rachman, 2015)

- 1) *Linear programming*
- 2) *Dynamic programming*

c. Metode Komputerisasi

Metode ini menggunakan bantuan komputer dalam menyeimbangkan lintasan perakitan. Salah satu metode yang sering digunakan adalah *COMSOAL (Computer Method of Sequencing Operation for Assembly Line)*. (Henry, 2011)

2.4 Bottleneck

Penumpukan yang terjadi mengakibatkan terganggunya proses produksi yaitu barang jadi menjadi lebih lama karena proses antrian yang panjang sehingga memerlukan waktu yang lama untuk prosesnya. Selain itu antrian yang panjang pada salah satu stasiun menyebabkan stasiun lain menjadi terganggu baik menjadi menganggur atau bahkan berhenti karena menunggu barang dari stasiun yang mengalami antrian yang Panjang. (Setyaningrum, 2018)

Perencanaan dan pengaturan yang tidak tepat dapat mengakibatkan setiap stasiun kerja pada lintas perakitan memiliki kecepatan produksi yang berbeda sehingga terjadi penumpukan material diantara stasiun kerja yang tidak berimbang kecepatan produksinya (*bottleneck*). Oleh karena itu perlu dilakukan usaha-usaha untuk menyeimbangkan lintasan (*line balancing*).

2.5 Sistem

Mendefinisikan sistem sebagai kumpulan entitas yang saling berinteraksi untuk mencapai tujuan tertentu yang logis. (Siswanto, 2018)

Adapun beberapa alasan perlu adanya pemikiran sistem adalah sebagai berikut : (Arifin, 2009)

- a. Meningkatnya kompleksitas permasalahan
- b. Kebutuhan akan efisiensi dan efektivitas
- c. Penyelsaian yang sering menggunakan intuitif dan tidak terencana

2.4.1 Elemen-elemen Sistem

Berikut adalah penjelasan mengenai elemen-elemen sistem : (Arifin, 2009)

- a. Entitas dan Atribut

Entitas adalah item-item yang akan diproses oleh sistem. Karakteristik khas dari entitas adalah biaya, bentuk, prioritas, dan sebagainya. Misalnya, produk paku, pelanggan bank, dan lain-lain. Sedangkan atribut adalah segala sesuatu yang menjadi properti dari entitas. Misalnya, operator mesin adalah entitas, maka skill dari operator adalah atribut. Maka dari itu, entitas terbagi menjadi beberapa bentuk, antara lain : (Arifin, 2009)

- 1) Bernyawa, misalnya pelanggan, pasien, dan lain-lain.
- 2) Tidak bernyawa, misalnya part, dokumen, dan lain-lain.
- 3) Tidak dapat diraba, misalnya kemajuan teknologi informasi yang pesat, dan lain-lain.

b. *Aktivitas dan Delay*

Aktivitas merupakan kejadian yang dilakukan sistem baik secara langsung maupun tidak langsung dalam memproses entitas. Misalnya memperbaiki peralatan, melayani pasien, dan lain-lain. Sedangkan *delay* merupakan keadaan dimana durasi dari proses tidak diketahui. Misalnya menunggu untuk dilayani, menunggu produk diproses (*work in proses*) pada proses manufaktur, dan lain sebagainya. (Arifin, 2009)

c. *Sumber Daya dan Kontrol*

Sumber daya diartikan sebagai segala sesuatu yang dapat membantu aktivitas atau bisa dikatakan sebagai fasilitas pendukung, misalnya peralatan, personel, dan lain sebagainya. Sumber daya memiliki banyak karakteristik seperti kapasitas, kecepatan, waktu siklus, dan lain-lain. Oleh sebab itu sumber daya dapat dikategorikan sebagai berikut : (Arifin, 2009)

- 1) Manusia atau yang bernyawa, meliputi dokter, operator, dan lain-lain.
- 2) Yang tidak bernyawa, meliputi peralatan, rantai produksi, dan lain-lain.
- 3) Tidak dapat diraba, meliputi informasi, tenaga elektrik, dan lain-lain.

2.4.2 Karakteristik Sistem

Karakteristik-karakteristik sistem meliputi : (Sultan, 2007)

- Perilaku sasaran (*purposive behaviour*)

Setiap sistem berusaha mencapai satu sasaran atau lebih sehingga tujuan menjadi pendorong (motivasi) dari sistem untuk mencapai tujuan tersebut.

- Keseluruhan (*wholism*)

Suatu teori yang menyatakan bahwa faktor-faktor penentu merupakan kesatuan yang tidak dapat direduksi lagi.

- Keterbukaan (*openness*)
Menunjukkan kesamaan akhir (*quifinality*), ini berarti bahwa status akhir dari suatu sistem dapat dicapai dari berbagai status awal.
- Transformasi (*transformation*)
Menunjukkan bahwa suatu sistem mempunyai kemampuan untuk mengubah nilai status sumber daya (*input*) menjadi keluaran (*output*) melalui suatu proses transformasi.
- Keterhubungan (*interrelatedness*)
Mencakup interaksi internal dan ketergantungan antar bagian-bagian atau elemen-elemen pembentuk sistem dan interaksi sistem dengan lingkungannya.
- Mekanisme kontrol (*control mechanism*)
Merupakan proses pengaturan yang digunakan sistem untuk mengoreksi setiap penyimpangan yang terjadi.

2.4.3 Ukuran Kinerja Sistem

Berikut adalah penjelasan mengenai kinerja sistem : (Arifin, 2009)

- a. Aliran Waktu
Aliran waktu (*flow time*) adalah waktu rata-rata yang diambil untuk item yang akan diproses dalam sistem, biasanya disebut juga sebagai *lead time*.
- b. *Utilitas*
Utilitas adalah persentase dari jadwal waktu dari operator dan sumber daya lainnya pada saat produksi.
- c. Nilai Waktu
Nilai waktu adalah banyaknya waktu materia, konsumen, dan nilai penerimaan aktual dimana nilai tersebut membatasi semua keinginan konsumen untuk membayar.
- d. Waktu Tunggu
Waktu tunggu adalah banyaknya waktu dimana material konsumen sedang menunggu untuk diproses. Waktu tunggu dapat ditingkatkan dengan mengurangi jumlah item pada sistem.

e. Rata-rata Aliran (*flow rate*)

Rata-rata aliran (*flow rate*) adalah banyaknya item yang diproduksi atau pelayanan konsumen per unit satuan waktu. *Flow rate* dapat ditingkatkan dengan manajemen yang lebih baik dan utilisasi sumber daya khususnya membatasi sumber daya.

f. Tingkat Antrian

Tingkat antrian adalah jumlah item atau konsumen saat sedang menunggu. Persediaan dapat dikontrol dengan membatasi produksi saat operasi sedang *bottleneck*.

g. Produksi

Dari tingkat produksi, persentase produk yang disesuaikan dengan spesifikasi atau persentase total jumlah produk yang masuk dalam sistem sebagai *raw material* (bahan mentah/bahan baku).

h. Variansi

Variansi adalah derajat dari fluktuasi yang dapat dan sering terjadi pada kinerja pendahuluan. Variansi memperkenalkan ketidakpastian dan risiko dalam usaha mencapai tujuan dan performansi. Variansi dikurangi dengan mengontrol waktu proses dan meningkatkan sumber daya.

2.4.4 Variabel-variabel Sistem

Berikut adalah penjelasan mengenai variabel-variabel sistem : (Arifin, 2009)

a. Variabel Keputusan

Variabel keputusan adalah sebagai variabel yang independent (tidak bergantung). Contohnya adalah tingkat pelayanan, tingkat kedatangan order, dan lain-lain.

b. Variabel Respon

Variabel respon merupakan variabel yang mengukur performansi dari sistem yaitu untuk memberikan respon pada variabel keputusan tertentu. Misalnya, jumlah entitas yang diproses untuk waktu tertentu, rata-rata penggunaan sumber daya, dan lain-lain. Contohnya adalah tingkat produksi, rata-rata waktu tunggu, dan lain-lain.

c. Variabel *State*

Variabel *state* merupakan variabel yang menandai status dari sistem pada saat tertentu. Contohnya adalah kapasitas, jumlah server, kecepatan *conveyor*, dan lain-lain.

2.6 Model

Sebelum melakukan pembuatan model, terdapat juga aktivitas siklus diagram atau *activity cycle diagram* (ACD). Menurut Subagyo dan Frylie (2018), *activity cycle diagram* adalah metode untuk menggambarkan interaksi antar objek dalam suatu sistem. Diagram ini memberikan notasi pemodelan grafis untuk menjelaskan serangkaian kegiatan dalam berbagai situasi kehidupan nyata. Dalam penelitian ini, *activity cycle diagram* (ACD) terdapat pada lampiran 3.

Model didefinisikan sebagai representasi dari sistem baik secara kualitatif kuantitatif yang mewakili suatu proses atau kejadian, dimana dapat menggambarkan secara jelas hubungan interaksi antar berbagai factor-faktor penting yang akan diamati. (Sultan, 2007)

Sedangkan model sistem adalah tiruan dari sistem dengan seluruh komponen dan perilakunya yang relevan untuk mencapai tujuan dari kajian sistem. (Siswanto, 2018)

Terdapat 4 jenis model yang berdasarkan model simbolik, yaitu : (Arifin, 2009)

a. Model Stokastik

Model yang mencakup distribusi kemungkinan untuk input dan memberikan serangkaian nilai dari sekurang-kurangnya satu variabel output dengan probabilitas yang berkaitan pada tiap nilai. Contohnya yaitu waktu kedatangan pelanggan, waktu antrian pelanggan, dan lain-lain.

b. Model Deterministik

Model yang dipergunakan untuk memecahkan suatu persoalan dalam situasi yang pasti. Contohnya yaitu proses kimia, peta, dan lain-lain.

c. Model Statis

Model yang berhubungan dengan keadaan sistem pada suatu saat tidak mempertimbangkan perubahan waktu. Biasanya hanya melibatkan

pembangkitan bilangan random untuk menjalankan simulasi. Contohnya yaitu penganggaran keuangan universitas, penentuan jumlah persediaan di gudang, dan lain sebagainya.

d. Model Dinamis

Model yang berkaitan dengan keadaan sistem dalam waktu yang berkelanjutan, mengandung proses perubahan setiap saat akibat suatu aktivitas. Contohnya yaitu simulasi suatu layanan perbankan yang buka dari jam 08.00-15.00.

Pendekatan pemodelan meliputi sebagai berikut : (Sultan, 2007)

a. Pendekatan proses

Proses didefinisikan sebagai suatu operasi dimana entiti yang ada harus mampu melewati siklus dari sistem tersebut.

b. Pendekatan aktivitas

Merupakan deskripsi dari aktivitas yang akan selalu dipacu dengan segera oleh perubahan state dalam sistem.

c. Pendekatan *event*

Didefinisikan sebagai kumpulan aktivitas yang mungkin mengikuti perubahan state dalam sistem.

2.7 Simulasi

Simulasi adalah proses merencanakan suatu model dari sistem nyata dan melakukan eksperimen dengan model tersebut dengan tujuan memahami tingkah laku sistem atau mengevaluasi berbagai strategi untuk mengoperasikan sistem yang dimaksud. (Sultan, 2007)

Simulasi merupakan teknik untuk meniru operasi-operasi atau proses yang terjadi dalam sebuah sistem dengan menggunakan bantuan perangkat komputer dan dilandasi oleh beberapa asumsi tertentu sehingga sistem tersebut bisa dipelajari secara ilmiah. (Sentia, 2016)

Dengan demikian berikut adalah kelebihan menggunakan metode simulasi dengan *software* yang sesuai, meliputi : (Siswanto, 2018)

- a. Mampu memodelkan keterkaitan dan ketergantungan antar elemen pada suatu sistem yang tidak mampu diperoleh dari metode analitis.
- b. Fleksibel untuk berbagai jenis model sistem walaupun belum tentu merupakan pendekatan yang paling efektif.
- c. Dapat menunjukkan perilaku sistem dari waktu ke waktu.
- d. Menawarkan penghematan waktu dan biaya, terutama untuk permasalahan skala besar.
- e. Tidak merusak atau mengganggu sistem yang sebenarnya.
- f. Menyediakan informasi bukan hanya pada satu ukuran performansi sistem, namun beberapa ukuran sekaligus.
- g. Hasil dari simulasi mudah untuk dimengerti dan dikomunikasikan sehingga orang yang tidak memahami teorinya pun dapat memahami dengan baik.
- h. Menekankan pada detail sistem.

Metode simulasi dapat diterapkan di berbagai area bisnis maupun industri dengan tujuan yang berbeda-beda. Berikut merupakan beberapa contoh aplikasi penggunaan metode simulasi, antara lain : (Siswanto, 2018)

- a. Perencanaan kapasitas dan sumber daya
- b. Pengurangan waktu siklus
- c. Analisis layout
- d. Perencanaan sumber daya dan karyawan
- e. Analisis *bottleneck*
- f. Penjadwalan sumber daya, dan lain-lain

Dalam simulasi terdapat beberapa analisis yang harus dilakukan terkait dengan output simulasi, diantaranya : (Siswanto, 2018)

- a. Verifikasi

Verifikasi merupakan proses untuk menentukan apakah model simulasi telah berjalan benar dan sesuai dengan yang diinginkan. Berikut beberapa teknik yang dapat digunakan untuk melakukan verifikasi model simulasi adalah sebagai berikut : (Siswanto, 2018)

1. Meninjau ulang kode dan teks pada model
2. Memeriksa kewajaran output

3. Melihat animasi
4. Menggunakan fasilitas lacak dan debug

b. Penentuan jumlah replikasi

Pada prinsipnya, penentuan kecukupan jumlah replikasi dalam simulasi sama dengan penentuan kecukupan data atau sampel dalam suatu observasi. Dalam menentukan jumlah replikasi sebenarnya tidak memerlukan prosedur yang rumit, selain mempertimbangkan statistik, faktor teknis juga perlu dipertimbangkan dalam menentukan jumlah replikasi. (Siswanto, 2018)

Kemudian setelah model terverifikasi maka dilakukan penentuan jumlah replikasi. Untuk menghitung jumlah replikasi sejumlah n dalam simulasi dapat dilakukan dengan langkah-langkah berikut ini : (Siswanto, 2018)

1. Lakukan percobaan dengan n tertentu. (misalnya $n=5$, $n=10$, dan lain sebagainya).
2. Jalankan model simulasi dan dapatkan output yang menjadi fokus observasi. Dapatkan nilai hw atau error dengan rumus berikut :

$$hw = e = t_{\alpha/2}^{n-1} \frac{s}{\sqrt{n}}$$

hw juga dapat diperoleh dari *output software* simulasi secara langsung, salah satunya yaitu *software Arena*.

3. Lakukan evaluasi terhadap nilai hw , apabila nilainya sudah cukup memuaskan, maka gunakan n pada percobaan pertama dan tidak perlu menghitung nilai n' . Adapun nilai maksimal hw adalah 5% dari jumlah data. Apabila diinginkan nilai hw yang lebih kecil, dapat juga dihitung n' dengan rumus berikut :

$$n' = \left[Z_{\alpha/2} \frac{s}{hw} \right]^2 \text{ atau } n' = \left[\frac{(Z_{\alpha/2})s}{\left(\frac{e\%}{1+e\%}\right)\bar{x}} \right]^2 \text{ atau } n' \cong n \frac{hw^2}{(hw)^2}$$

c. Validasi

Validasi model adalah proses menentukan apakah model telah mempresentasikan *real system* dengan akurat berikut ini merupakan beberapa alternatif yang dapat digunakan untuk melakukan validasi model secara kualitatif : (Siswanto, 2018)

1. Melihat animasi dan tampilan model (visual)
2. Membandingkan model dengan sistem aktualnya
3. Membandingkan model lain yang dibuat, misalnya model simulasi sederhana, dan lain lain
4. Melakukan uji sensitivitas dan uji kondisi ekstrim pada model
5. Uji data historis
 - a) Uji Keseragaman Data
 - b) Uji Kecukupan Data
6. Uji kesamaan rata-rata

Uji t-berpasangan adalah sebuah hipotesis yang digunakan untuk mengetahui perbedaan antara dua tanggapan yang diukur pada unit statistik yang sama. Uji t berpasangan terdiri atas sampel pasangan dari unit serupa, atau sampel dari satu unit kelompok yang telah di uji 2 kali (pengukuran berulang uji t). dalam uji t-berpasangan variansi acak antara sampel satu dengan yang lain yang menjadi obyek pengujian telah di eliminasi, karena sampel yang di ambil dari obyek pengujian menjadi kotrol atas diri mereka sendiri. (Saputra, 2015)

Uji t-berpasangan digunakan untuk membandingkan selisih 2 rata-rata (*mean*) dari 2 sampel yang berpasangan dengan asumsi data tersebut berdistribusi normal. (Saputra, 2015)

Uji t berpasangan menggunakan uji hipotesis sebagai berikut : (Saputra, 2015)

$$H_0 : \mu = \mu_0$$

$$H_0 : \mu \neq \mu_0$$

Dengan μ = rata-rata selisih antara dua populasi.

Nilai dari μ_0 adalah nilai hipotesis dari μ , dan umumnya nilai $\mu_0 = 0$. Hal ini berarti tidak terdapat perbedaan antara dua sampel. Rata-rata selisih antara dua sampel adalah \bar{d} , yang digunakan untuk memperkirakan nilai dari μ . Untuk memperoleh nilai \bar{d} , dihitung perbedaan d_i ($i = 1, 2, 3, \dots, n$) dari data X_{i1} dan X_{i2} . (Saputra, 2015)

$$d_i = X_{i1} - X_{i2}$$

Maka,

$$\bar{d} = \frac{\sum d_i}{n}$$

Untuk melakukan uji statistik dengan rumus : (Saputra, 2015)

$$t_{hitung} = \frac{x_1 - x_2}{s \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

Dengan :

$x_1 - x_2$ = rata-rata dari selisih antara dua sampel

n_1 = jumlah sampel kelompok pertama

n_2 = jumlah sampel kelompok kedua

s = deviasi standar selisih dari dua sampel

Jika deviasi standar dari perbedaan tidak diketahui, maka untuk menghitung deviasi standar digunakan rumus : (Saputra, 2015)

$$S_d = \sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}}$$

Bentuk lengkap dari hipotesis untuk uji t-berpasangan adalah : (Saputra, 2015)

- a. $H_0 : \mu = 0$, tidak ada perbedaan antara sebelum dan sesudah adanya perlakuan.
- b. $H_0 : \mu \neq 0$, ada perbedaan antara sebelum dan sesudah adanya perlakuan

Dalam pengujian hipotesis untuk uji t-berpasangan, kriteria untuk menolak H_0 atau menerima H_0 berdasarkan t_{hitung} adalah sebagai berikut : (Saputra, 2015)

- a. Jika $t_{hitung} < t_{tabel}$ atau $P_{value} > \alpha$, maka H_0 di terima
- b. Jika $t_{hitung} > t_{tabel}$ atau $P_{value} < \alpha$, maka H_0 di tolak, dimana

$$t_{tabel} = t_{\alpha/2}$$

Rumus derajat kebebasan untuk distribusi t adalah : (Saputra, 2015)

$$v = n - 1$$

Asumsi yang harus dipenuhi dalam uji statistik ini adalah bahwa d_i harus mengikuti distribusi normal. (Saputra, 2015)

d. Penentuan jumlah replikasi

Pada prinsipnya, penentuan kecukupan jumlah replikasi dalam simulasi sama dengan penentuan kecukupan data atau sampel dalam suatu observasi. Dalam menentukan jumlah replikasi sebenarnya tidak memerlukan prosedur yang rumit, selain mempertimbangkan statistik, faktor teknis juga perlu dipertimbangkan dalam menentukan jumlah replikasi. (Siswanto, 2018)

Kemudian setelah model terverifikasi maka dilakukan penentuan jumlah replikasi. Untuk menghitung jumlah replikasi sejumlah n dalam simulasi dapat dilakukan dengan langkah-langkah berikut ini : (Siswanto, 2018)

1. Lakukan percobaan dengan n tertentu. (misalnya $n=5$, $n=10$, dan lain sebagainya).
2. Jalankan model simulasi dan dapatkan output yang menjadi fokus observasi. Dapatkan nilai hw atau error dengan rumus berikut :

$$hw = e = t_{\alpha/2, n-1} \frac{s}{\sqrt{n}}$$

hw juga dapat diperoleh dari *output software* simulasi secara langsung, salah satunya yaitu *software* Arena.

3. Lakukan evaluasi terhadap nilai hw , apabila nilainya sudah cukup memuaskan, maka gunakan n pada percobaan pertama dan tidak perlu menghitung nilai n' . Apabila diinginkan nilai hw yang lebih kecil, hitunglah n' dengan rumus berikut :

$$n' = \left[Z_{\alpha/2} \frac{s}{hw} \right]^2 \text{ atau } n' = \left[\frac{(Z_{\alpha/2})^2 s}{\left(\frac{e\%}{1+e\%} \right) \bar{x}} \right]^2 \text{ atau } n' \cong n \frac{hw^2}{(hw')^2}$$

e. Analisis Skenario

Skenario adalah istilah yang sering digunakan dalam studi simulasi untuk mewakili suatu eksperimen dengan kondisi tertentu. Jumlah skenario yang dikembangkan dapat bervariasi tergantung pada dugaan, kreatifitas, dan hasil analisis yang berkembang dari studi pendahuluan. (Nurhadi Siswanto, 2018)

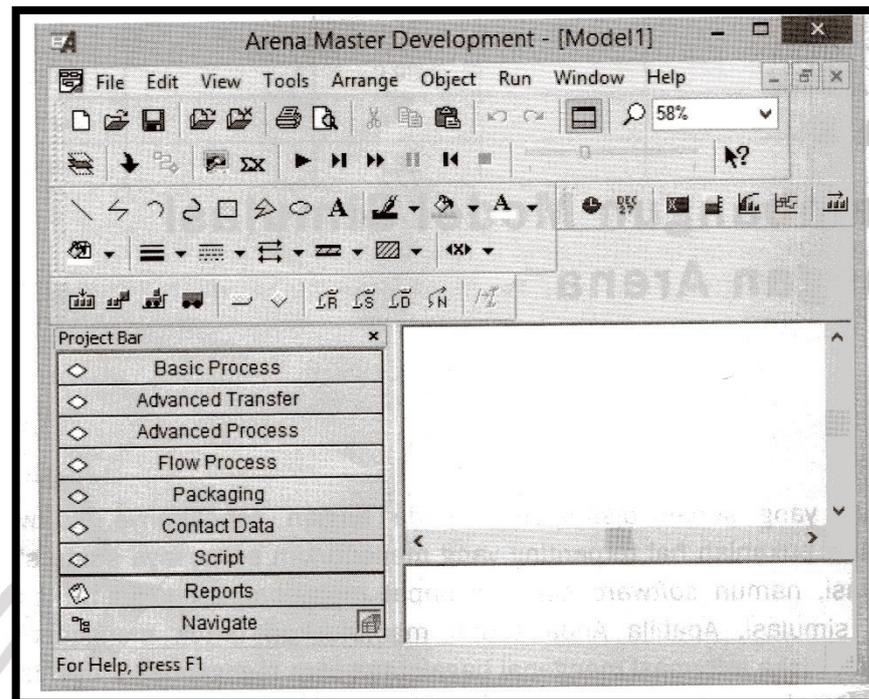
Dalam penelitian ini skenario menggunakan 3 metode keseimbangan lintasan yaitu metode *region approach*, *largest candidate rule*, dan *ranked positional weight*.

2.8 Software Simulasi Arena

Arena merupakan paket simulasi general purpose, yang memiliki kemampuan memodelkan sistem manufaktur dan non manufaktur. Sistem manufaktur seperti *flow lines*, *assembly lines*, *job shop*, *conveyors*,. Sistem non manufaktur seperti *health care*, *maintenance sistem*, *computer network*. (Taslim, 2015)

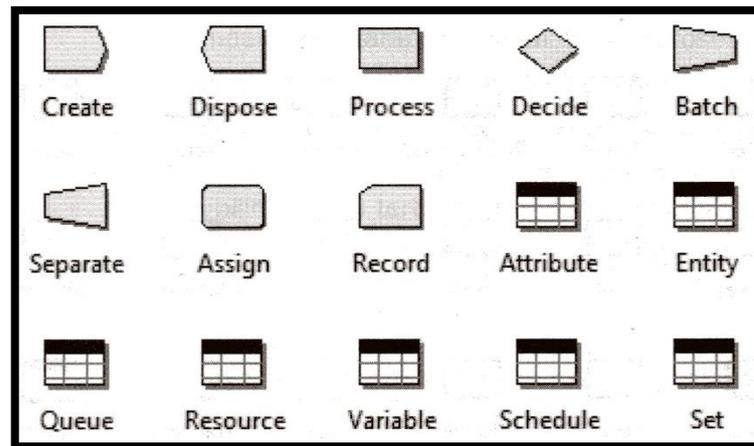
Didalam jendela Arena terdapat fungsi-fungsi umum pada bagian atas seperti halnya jendela pada *software* di bawah OS Windows lainnya. Dibagian kiri jendela software terdapat menu atau project bar yang menampilkan beberapa panel dimana masing-masing panel terdiri dari modul-modul khusus. (Siswanto dkk, 2017)

Panel adalah sekelompok modul yang dikumpulkan berdasarkan kategori fungsi dan tingkat kerincian model simulasi yang ingin dibangun. Sedangkan modul adalah blok-blok dasar dalam Arena yang digunakan untuk membangun model seperti halnya kita membuat *flowchart* dengan menyertakan data didalamnya. (Siswanto dkk, 2017)

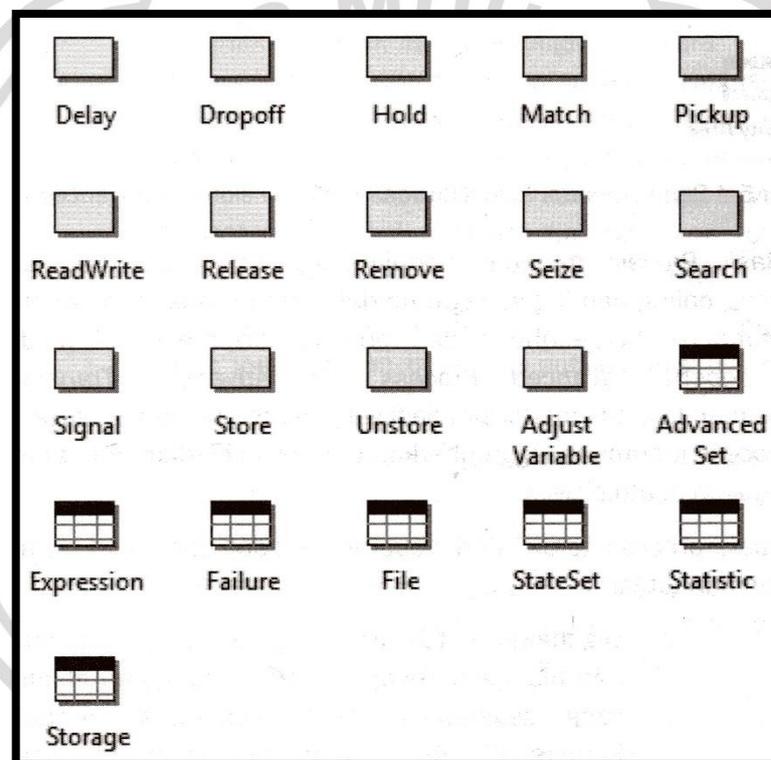


Gambar 2. 2 Jendela Utama Software Arena

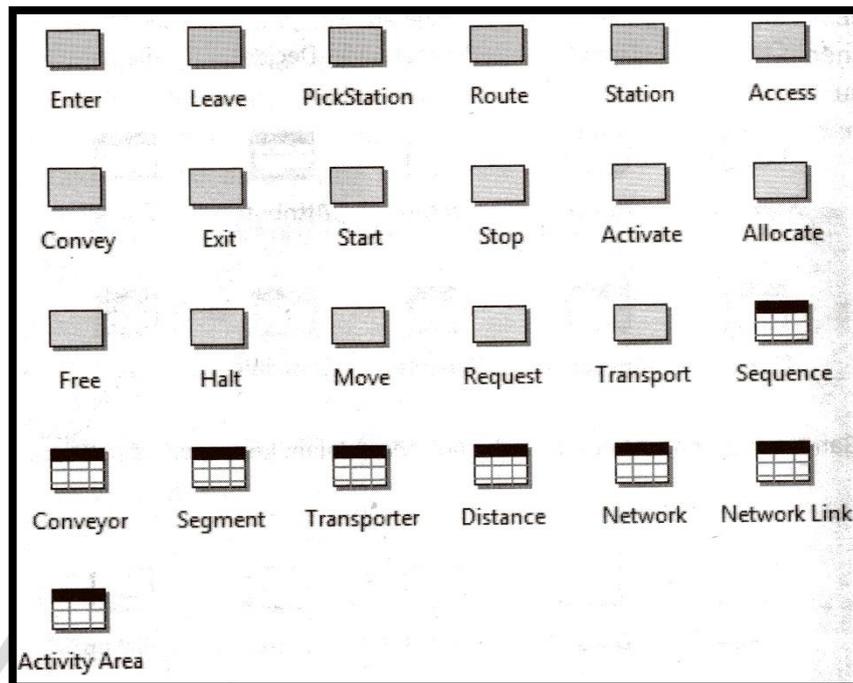
Modul terdiri dari dua jenis yaitu *flowchart* dan data. *Flowchart* dari modul yang mendefinisikan alur dari proses-proses dalam simulasi, sedangkan data berupa tabel yang mengelola detail dari proses dan elemen-elemen yang terlibat di dalam model. Modul-modul yang tersedia dalam Arena memiliki bentuk yang berbeda-beda dan menyerupai bentuk dasar *flowchart* klasik. Masing-masing modul dalam panel memiliki fungsi spesifik dalam memodelkan sistem. Panel utama dalam Arena meliputi *Basic Process*, *Advanced Process*, dan *Advanced Transfer*. Panel yang lainnya yaitu *packaging*, *flow process*, *blocks*, *statistics*, *elements*, *report*, dan *negative*. (Siswanto dkk, 2017)



Gambar 2. 3 Bentuk-bentuk Modul *Flowchart* Dalam Kelompok *Basic Process*



Gambar 2. 4 Bentuk-bentuk Modul *Flowchart* Dalam Kelompok *Advanced Process*



Gambar 2. 5 Bentuk-bentuk Modul *Flowchart* Dalam Kelompok *Advanced Transfer*

Panel basic process terdiri dari beberapa modul dan tabel data sebagai berikut : (Siswanto dkk, 2017)

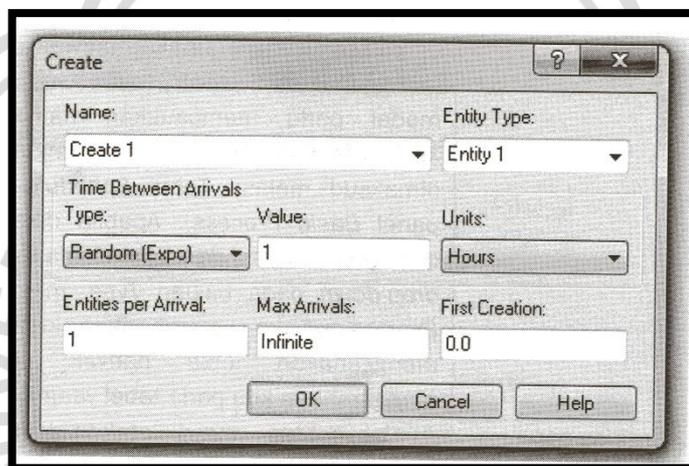
- a. *Create* : digunakan untuk membangkitkan/menciptakan entitas dan dianggap sebagai suatu kedatangan dalam sistem yang sebenarnya.
- b. *Process* : digunakan untuk mempresentasikan suatu proses dalam model simulasi dan mendefinisikan detail proses.
- c. *Decide* : digunakan ketika suatu entitas dihadapkan pada beberapa kondisi dan terdapat suatu aturan dalam memilih kondisi tertentu.
- d. *Dispose* : digunakan untuk mengeluarkan entitas dari sistem.
- e. *Batch* : digunakan untuk menggabungkan secara sementara atau permanen antara dua atau lebih entitas dengan alasan tertentu.
- f. *Separate* : digunakan untuk membuat duplikasi dari suatu entitas atau memisahkan beberapa entitas yang digabungkan melalui modul *batch*.
- g. *Assign* : digunakan untuk menetapkan nilai-nilai baru pada variabel maupun atribut berdasarkan terjadinya suatu event tertentu.

- h. *Record* : digunakan untuk pencatatan statistik yang lebih rinci dari suatu atribut tertentu atau menghitung entitas pada suatu titik tertentu dalam simulasi.
- i. *Tabel entity* : digunakan untuk mendefinisikan berbagai informasi mengenai entitas pada model simulasi yang meliputi, nama, gambar, biaya-biaya yang terkait, dan lain-lain.
- j. *Tabel recourse* : digunakan untuk mendefinisikan sumber daya yang digunakan pada model simulasi, termasuk biaya, kapasitas, jumlah, dan lain-lain.
- k. *Tabel set* : digunakan untuk mendefinisikan berbagai tipe dari set yang digunakan dalam model simulasi beserta informasi mengenai sumber daya didalamnya.
- l. *Tabel attribute* : digunakan untuk mendefinisikan *attribute* yang telah dibuat dengan sengaja atau otomatis dengan diciptakan sistem pada *software arena* untuk membedakan entitas dan memudahkan pencatatan statistik.
- m. *Tabel schedule* : digunakan untuk mendefinisikan jadwal operasi atau perubahan kapasitas dari sumber daya yang terdaftar atau mendefinisikan jadwal kedatangan
- n. *Tabel queue* : digunakan untuk mendefinisikan nama antrian dan tipe yang digunakan dalam antrian, misalnya : *first in first out, last in first out*, dan lain-lain.
- o. *Tabel variable* : digunakan untuk mendefinisikan nilai dan dimensi variabel yang digunakan dalam pencatatan data.

Untuk mempermudah penggunaan *software* simulasi Arena, berikut penjelasan modul dasar yaitu *create, process, decide, dan dispose* : (Siswanto dkk, 2017)

2.8.1 Modul *Create*

Modul *create* termasuk kategori dalam modul *flowchart*, digunakan sebagai titik awal dimana suatu entitas diciptakan pada suatu model simulasi. Tingkat kedatangan entitas diatur dengan suatu distribusi waktu antar kedatangan (*time between arrivals*) atau berupa jadwal. Entitas yang diciptakan pada modul *create* ini adalah syarat utama agar suatu sistem dapat mulai beroperasi. Modul *create* direpresentasikan dengan bentuk dua trapezium siku-siku yang digabungkan pada bagian alasnya. ketika modul tersebut di *double klik* menunjukkan kotak dialog box sebagai berikut : (Siswanto dkk, 2017)



Gambar 2. 6 Dialog Box Dari Modul *Create*

Tabel 2. 1 Keterangan Data *Input* Modul *Create*

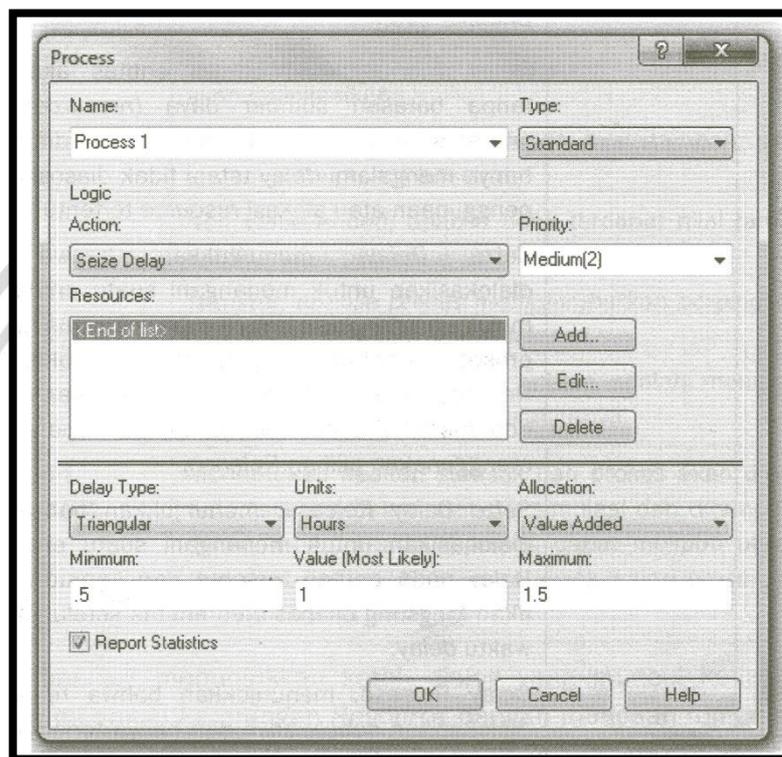
Label	Keterangan
<i>Name</i>	Nama spesifik yang muncul untuk suatu model <i>create</i> . Apabila suatu model menggunakan lebih dari satu <i>create</i> , maka nama yang digunakan tidak boleh sama.
<i>Entity Type</i>	Keterangan jenis entitas yang akan diciptakan dari <i>create</i> . Secara otomatis sistem akan menggunakan <i>entity 1</i> dan berurutan setelahnya, tetapi pengguna dapat memasukkan sendiri nama lainnya, missal : mobil, anak kecil, orang dewasa, dan lain-lain. Memberikan nama yang berbeda akan memudahkan kita untuk membuat penyesuaian spesifik pada masing-masing tipe.

<p><i>Time Between Arrivals</i></p>	<p>Menjelaskan tipe distribusi antar kedatangan (<i>type</i>) beserta nilai parameter dari distribusi yang dipilih (<i>value</i>) dan unit waktu yang digunakan (<i>unit</i>), misalnya <i>hours</i> atau <i>minutes</i>. Tipe distribusi waktu tidak selalu probabilistik, dapat pula dimodelkan secara deterministik menggunakan <i>constant</i> atau didefinisikan menggunakan jadwal tertentu. Apabila menggunakan jadwal, maka nantinya pembuat model perlu memasukkan nama jadwal dan kemudian menciptakan atau mengatur jadwal yang dimaksud melalui Tabel <i>Schedule</i> (masih dalam panel <i>Basic Process</i>). Apabila dikehendaki suatu distribusi tertentu dan tidak terdapat dalam pilihan <i>drop-down</i> pada bagian <i>Type</i>, maka pilihan <i>Build Expression</i> akan memberikan keleluasan untuk menggunakan lebih banyak pilihan dengan mengarahkan kita pada tabel yang lebih lengkap.</p>
<p><i>Entities Per Arrival</i></p>	<p>Jumlah entitas yang diciptakan dalam suatu waktu kedatangan tertentu. Misalnya : pada suatu pom bensin mobil selalu datang per unit, sedangkan pada suatu rumah makan kedatangan entitas lebih cenderung dua orang atau kelompok besar dalam sekali kedatangan.</p>
<p><i>Max Arrivals</i></p>	<p>Jumlah entitas maksimum yang diciptakan melalui modul <i>create</i> ini. Apabila angka maksimum sudah tercapai, maka <i>create</i> akan menghentikan keluarnya entitas.</p>
<p><i>First Creation</i></p>	<p>Posisi waktu atau jam simulasi ketika entitas pertama dimunculkan.</p>

Ketika suatu model *create* digunakan, maka secara otomatis pada modul data tabel *entity* akan muncul data entitas yang kita inputkan pada modul *create* beserta dengan atribut-atribut yang menyertainya, yaitu tipe, distribusi waktu antar kedatangan, unit waktu yang digunakan jumlah entitas per kedatangan, jumlah maksimum kedatangan, dan waktu entitas pertama diciptakan. (Siswanto dkk, 2017)

2.8.2 Modul *Process*

Modul *process* termasuk pada kategori *flowchart* dan digunakan untuk mewakili suatu proses termasuk sumber daya yang digunakan, antrian proses, dan waktu proses. *Double click* pada modul *process* akan membuka kotak dialog yang ditampilkan pada gambar berikut. (Siswanto dkk, 2017)



Gambar 2. 7 Kotak Dialog Modul *Process*

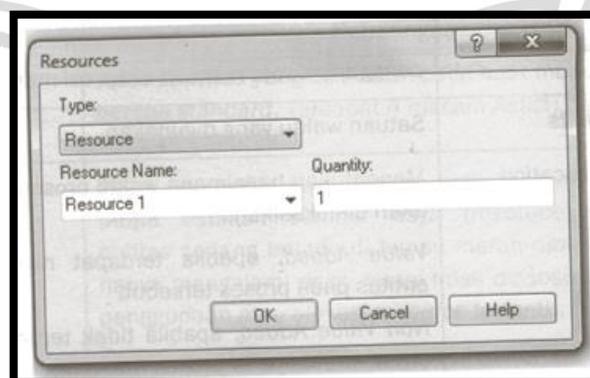
Tabel 2. 2 Keterangan Data *Input* Modul *Process*

Label	Keterangan
<i>Name</i>	Nama spesifik yang muncul untuk suatu modul <i>process</i> . Apabila suatu model menggunakan lebih dari satu <i>process</i> , maka nama yang digunakan tidak boleh sama.
<i>Type</i>	Adalah tipe logika yang digunakan dalam model, terdapat dua pilihan yaitu <i>standard</i> dan sub model. Pilihan pertama berarti semua logika disimpan dalam modul proses dan didefinisikan pada <i>action</i> . Pilihan kedua berarti logika disusun secara hierarki pada sub model yang terdiri dari beberapa modul. Ketika kita membuka suatu model <i>process</i> baru, kolom <i>type</i> akan otomatis terisi <i>standard</i> .

<p><i>Action</i></p>	<p>Jenis aktivitas yang dilakukan pada saat modul <i>process</i> bertipe <i>standard</i>. Terdapat 4 macam <i>action</i> pada modul <i>process</i>, yaitu :</p> <ol style="list-style-type: none"> 1) <i>Delay</i> Menunjukkan bahwa entitas akan diproses tanpa batasan sumber daya (<i>resource</i>), misalnya entitas sedang berada di lampu merah dimana entitas hanya mengalami <i>delay</i> tetapi tidak diasosiasikan dengan penggunaan atau alokasi <i>resource</i> tertentu. 2) <i>Seize delay</i> Menunjukkan bahwa <i>resource</i> dialokasikan untuk menangani suatu entitas (setelah melalui antrian) dan terjadi suatu aktivitas (<i>delay</i>) pada proses tersebut dengan durasi waktu tertentu. <i>Resource</i> tidak langsung dilepas dan akan dilepas saat ada modul <i>process</i> lainnya dengan tipe <i>acion</i> yang menggunakan pilihan <i>release</i>. 3) <i>Seize delay release</i> Menunjukkan bahwa <i>resource</i> dialokasikan untuk menangani suatu entitas, diikuti <i>delay</i> pada entitas tersebut dan kemudian <i>resource</i> akan langsung dilepas oleh entitas setelah berakhirnya waktu <i>delay</i>. 4) <i>Delay release</i> Menunjukkan bahwa <i>resource</i> telah dialokasikan sebelumnya dan hanya terjadi <i>delay</i> pada entitas tersebut dan kemudian <i>resource</i> akan langsung dilepas.
<p><i>Priority</i></p>	<p>Nilai prioritas untuk entitas yang menunggu <i>resource</i> tertentu. Digunakan apabila terdapat satu atau lebih entitas dari modul lain menunggu untuk <i>resource</i> yang sama (<i>shared resources</i>).</p>
<p><i>Resources</i></p>	<p>Digunakan untuk menambahkan <i>resource</i> atau <i>set</i> yang digunakan dalam proses, namun kotak <i>resource</i> menjadi tidak aktif ketika yang dipilih dalam <i>action</i> atau <i>delay</i>, atau yang dipilih dalam <i>type</i> adalah <i>sub model</i>. Definisi lebih jelas mengenai <i>resource</i> dapat diisikan pada kotak dialog (gambar x.x) yang muncul dengan <i>Add</i>.</p>

<i>Delay Type</i>	Jenis distribusi probabilitas waktu proses yang digunakan, umumnya pada pilihan <i>drop down</i> hanya akan terdapat pilihan yang terbatas, yaitu <i>normal</i> , <i>triangular</i> , <i>exponential</i> . Selain itu juga terdapat pilihan <i>constant</i> atau <i>build expression</i> dimana pengguna dapat memilih ekspresi matematis lainnya.
<i>Units</i>	Satuan waktu yang digunakan.
<i>Allocation</i>	Menentukan bagaimana waktu proses dan biaya proses akan dialokasikan. <i>Value added</i> , apabila terdapat nilai tambah untuk entitas pada proses tersebut. <i>Non value added</i> , apabila tidak terdapat nilai tambah untuk entitas pada proses tersebut. <i>Transfer</i> , apabila proses menggambarkan perpindahan entitas dari satu titik ke titik lainnya. <i>Wait</i> , apabila yang dialami entitas adalah menunggu untuk melakukan aktivitas selanjutnya.
<i>Report Statistics</i>	Menentukan apakah statistik dari proses akan dicatat dan disimpan dalam statistik simulasi dan ditampilkan dalam <i>report</i> . Statistik proses dapat meliputi utilisasi <i>resource</i> , panjang antrian, rata-rata waktu proses, dan rata-rata waktu mengantri.

Berikut ini menunjukkan gambar kotak dialog yang muncul ketika menambahkan keterangan *resource* dengan menekan tombol *add*. (Siswanto dkk, 2017)



Gambar 2. 8 Kotak dialog Untuk Menambahkan *Resource* Pada Modul *Process*

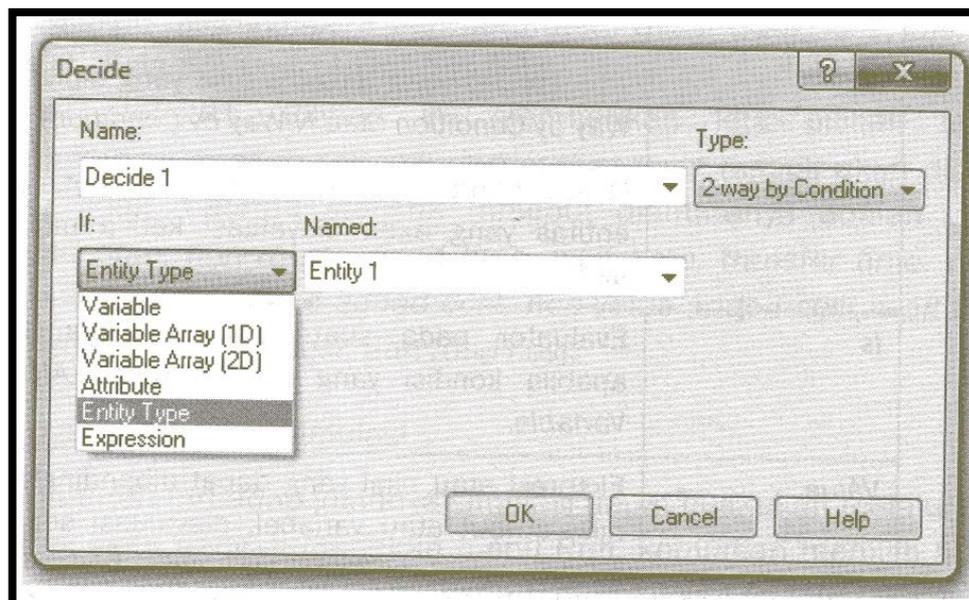
Kolom *type* memberikan alternatif untuk menggunakan *resource* atau *set*. Ketika yang dipilih adalah *resource* maka yang akan digunakan oleh entitas dalam modul *process* hanyalah *resource* yang namanya tertera dalam kolom *resource name* sejumlah yang tertulis pada *quality*. Namun apabila yang dipilih pada *type* adalah *set*, maka entitas dapat menggunakan *resource* mana saja yang terdapat dalam *set* tersebut dengan aturan yang telah ditentukan. Kolom *quality* pada pilihan *set*, tidak menunjukkan jumlah *resource* yang digunakan oleh entitas, tetapi menunjukkan jumlah *set* yang ada. (Siswanto dkk, 2017)

Apabila jenis *resource* yang digunakan adalah *set*, maka selanjutnya kita perlu memilih *selection rule* untuk mengatur bagaimana entitas akan memilih *resource* yang ada dalam *set*. Pilihan dalam *selection rule* sebagai berikut : (Siswanto dkk, 2017)

- *Cyclical*, sesuai urutan *resource* yang ada dalam suatu *set* misalnya : *resource 1*, *resource 2*, dan seterusnya.
- *Random*, artinya pemilihan *resource* oleh entitas dilakukan secara acak.
- *Preferred order*, mengutamakan anggota *resource* yang tersedia, contohnya *resource 1* akan terpilih jika tersedia, jika tidak maka berikutnya akan diarahkan ke *resource 2* jika tersedia, dan seterusnya
- *Specific member*, berarti anggota dari *set* yang dipilih sesuai dengan *input atribut* (yang sebelumnya telah disimpan dikolom *save attribute*).
- *Largest remaining capacity*, mengalokasikan *resource* dalam *set* berdasarkan sisa kapasitas *resource* terbesar.
- *Smallest number busy*, mengalokasikan *resource* berdasarkan tingkat kesibukan (rangkuman status *busy*) yang terkecil.

2.8.3 Modul *Decide*

Modul *decide* termasuk pada kategori modul *flowchart*, digunakan ketika alur proses dihadapkan pada situasi dimana apa yang terjadi berikutnya pada entitas ditentukan pada suatu kondisi tertentu. Kondisi tertentu tersebut dapat diekspresikan pada modul *decide*. (Siswanto dkk, 2017)



Gambar 2. 9 Kotak Dialog Modul *Decide*

Pada modul *decide*, pilihan dapat ditentukan berdasarkan satu atau lebih kondisi atau berdasarkan peluang. Kondisi pada suatu pilihan dapat berdasarkan nilai atribut, variabel, tipe entitas atau berupa ekspresi lainnya. (Siswanto dkk, 2017)

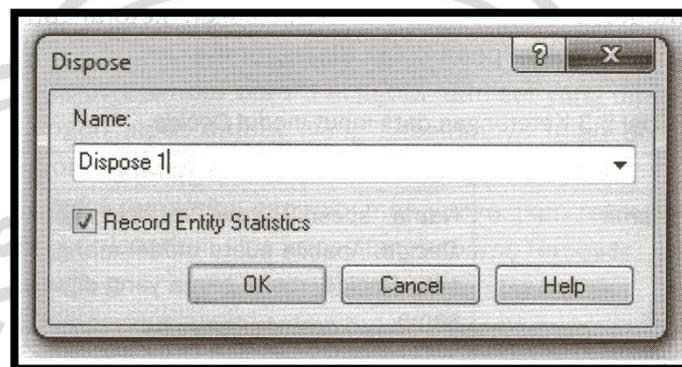
Tabel 2. 3 keterangan Data *Input* Modul *Decide*

Label	Keterangan
<i>Name</i>	Nama spesifik yang muncul untuk suatu model <i>decide</i> . Apabila suatu model menggunakan lebih dari satu <i>decide</i> , maka nama yang digunakan tidak boleh sama.
<i>Type</i>	Mendefinisikan apakah keputusan akan dibuat berdasarkan suatu kondisi atau suatu peluang. Untuk masing-masing kondisi dan peluang terdapat pilihan apakah 2 atau lebih dari 2 keadaan.
<i>Precent true</i>	Adalah nilai % yang digunakan untuk kondisi yang bernilai benar, otomatis sisanya adalah kondisi lainnya (100%- <i>precent true</i>)
<i>If</i>	Tipe kondisi yang tersedia (jika yang dipilih adalah <i>2-way by condition</i> atau <i>N-way by condition</i>).
<i>Named</i>	Dapat berupa nama variabel, atribut, atau jenis entitas yang akan dievaluasi ketika modul <i>decide</i> aktif.
<i>Is</i>	Evaluator pada suatu kondisi, digunakan hanya apabila kondisi yang dipilih adalah <i>atributte</i> atau <i>variable</i> .
<i>Value</i>	Ekspresi atau nilai yang dapat dibandingkan dengan nilai atribut atau variabel, dievaluasi sebagai suatu ekspresi yang menentukan

	kondisi benar atau salah. <i>Value</i> tidak digunakan ketika <i>input</i> pada <i>type</i> adalah <i>entity type</i> . Apabila <i>type</i> yang dipilih adalah <i>expression</i> , maka selanjutnya evaluator harus dituliskan pada kolom <i>value</i> , misalnya <i>color = = red</i> .
--	---

2.8.4 Modul *Dispose*

Modul *dispose* digunakan sebagai titik akhir entitas dari model simulasi. Modul *dispose* berfungsi mengeluarkan entitas dari sistem. Statistik dari suatu entitas dapat direkam terlebih dahulu sebelum entitas tersebut keluar.



Gambar 2. 10 Modul *Dispose* Beserta Kolom *Input Data*

Pada modul *dispose* terdapat satu kolom isian, yaitu nama modul (*name*). Nama modul seharusnya adalah nama spesifik yang tidak boleh sama antara satu modul *dispose* dengan modul *dispose* lainnya. Kolom nama ini tidak harus diisi karena ketika mengambil modul *dispose* dari panel, nama akan otomatis disediakan oleh *software* dan berurutan.

2.9 Penelitian Terdahulu

No	Nama Peneliti	Objek Penelitian	Judul Penelitian	Hasil Penelitian
1	Ery Achmad Farid Setya, Parwadi Moengin, dan Sucipto	Produksi Gasket di PT.JEIL FAJAR INDONESIA	Perbaikan Keseimbangan lintasan produksi gasket di	Dari Perhitungan <i>Line balancing</i> diketahui bahwa stasiun kerja yang mengalami utilitas tertinggi adalah pada area <i>packaging</i> dengan nilai 100% dan terendah di area <i>cleaning</i>

	Adisueiryo Jurnal Teknik Industri, (2014)		PT.JEIL FAJAR INDONESIA dengan menggunakan pendekatan simulasi	dengan nilai 10,65%. Dari analisa simulasi model awal diketahui stasiun kerja yang mengalami utilitas tertinggi adalah area <i>packaging</i> dengan nilai 99,57% dan terendah di area <i>cleaning</i> dengan nilai 16,39%. Berdasarkan hasil simulasi model awal yang telah terverifikasi dan valid dengan model nyata, PT.JEIL FAJAR INDONESIA menghasilkan replikasi sebanyak 5 kali agar dapat mengurangi error
2.	Pablo Cortés, Luis Onieva dan José Guadix, <i>International Journal</i> (2014)	Perakitan sepeda motor di perusahaan manufaktur x	<i>Optimising And Simulating The Assembly Line Balancing Problem In A Motorcycle Manufacturing Company: A Case Study</i>	Dengan menerapkan metode heuristik yang disesuaikan dengan karakteristik spesifik jalur perakitan sepeda motor (MRPWH) yang dibandingkan dengan hasil dari metode <i>neighbourhood search</i> (MSNSH), kemudian dilakukan simulasi dengan skenario awal agar memungkinkan terjadinya model simulasi yang valid. Dari simulasi yang sesuai dengan operasi yang terjadi pada pabrik saat ini, MRPWH dan MSNSH memberikan solusi yang lebih baik dari pada model simulasi dalam skenario awal, hasilnya dapat membuat tingkat produktivitas lebih besar dan garis yang lebih seimbang. Pada jadwal MSNSH sangat seimbang dari pada <i>bottleneck</i> yang bervariasi dari stasiun 3 ke stasiun 4, terutama yang terjadi ada stasiun 4. Setelah menerapkan jadwal MSNSH di pabriknya,

				diperoleh tingkat produktivitas antara 55-59 sepeda motor atau rata-rata tiap hari adalah 56 motor, namun efisiensi pekerja terus meningkat. Selain itu, komponen baru termasuk dalam atau dikeluarkan dari model sepeda motor setiap tahun. Karena semua faktor ini, keseimbangan jalur perakitan harus terus dipantau dan diseimbangkan kembali untuk mencapai hasil yang lebih baik dan mencapai kondisi peningkatan yang berkelanjutan.
3	Sucipto Adisuwiryo, Parwadi Moengin, Obriga Nathaniel Menayang Jurusan Teknik Industri, FTI Universitas Trisakti (2015)	Produksi pompa FS pada lini produksi di PT. Ebara Indonesia	Perbaikan Lini Produksi Dengan Perancangan Model Simulasi Untuk Mengurangi <i>Overtime</i> Di PT. Ebara Indonesia	Persentase <i>overtime</i> dibandingkan dengan jam kerja reguler cukup tinggi yaitu 17.55%. Perusahaan mengharapkan persentase <i>overtime</i> dibawah 5%. Produk yang diteliti adalah pompa FS, hal ini dikarenakan produk ini paling sering diminta dan diproduksi oleh PT. Ebara Indonesia. Berdasarkan hasil perhitungan waktu penyelesaian model awal dalam mencapai target produksi dibutuhkan waktu 506.66 jam. Setelah dilakukan simulasi, waktu penyelesaian dari 506.66 jam menjadi 344.07 jam sehingga tidak terdapat waktu <i>overtime</i> .