



UNIVERSITAS INDONESIA

**PEMBUATAN MODEL SIMULASI DALAM RANGKA
PERANCANGAN *RECONFIGURABLE MANUFACTURING*
*SYSTEM***

SKRIPSI

**ILHAM WINOTO
0806458901**

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK INDUSTRI
DEPOK
JUNI 2012**



UNIVERSITAS INDONESIA

**PEMBUATAN MODEL SIMULASI DALAM RANGKA
PERANCANGAN *RECONFIGURABLE MANUFACTURING*
SYSTEM**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

**ILHAM WINOTO
0806458901**

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK INDUSTRI
DEPOK
JUNI 2012**

HALAMAN PERNYATAAN ORISINALITAS

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

Nama : Ilham Winoto

NPM : 0806458901

Tanda tangan : 

Tanggal : 21 Juni 2012

HALAMAN PENGESAHAN

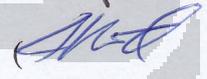
Skripsi ini diajukan oleh

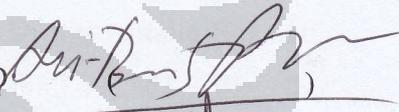
Nama : Ilham Winoto
NPM : 0806458901
Program Studi : Teknik Industri
Judul Skripsi : Pembuatan Model Simulasi dalam Rangka Perancangan *Reconfigurable Manufacturing System*

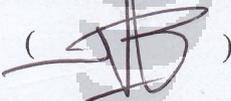
Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Industri, Fakultas Teknik, Universitas Indonesia

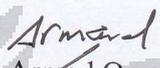
DEWAN PENGUJI

Pembimbing : Prof. Dr. Ir. Teuku Yuri M. Zagloel, M.Eng.Sc. ()

Pembimbing : Romadhani Ardi, ST., MT. ()

Penguji : Ir. Sri Bintang Pamungkas, MSISE., PhD ()

Penguji : Ir. Hj. Erlinda Muslim, MEE ()

Penguji :  Armad Omar Moeis, ST., M.Sc. ()

Ditetapkan di : Depok

Tanggal : 21 Juni 2012

KATA PENGANTAR

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

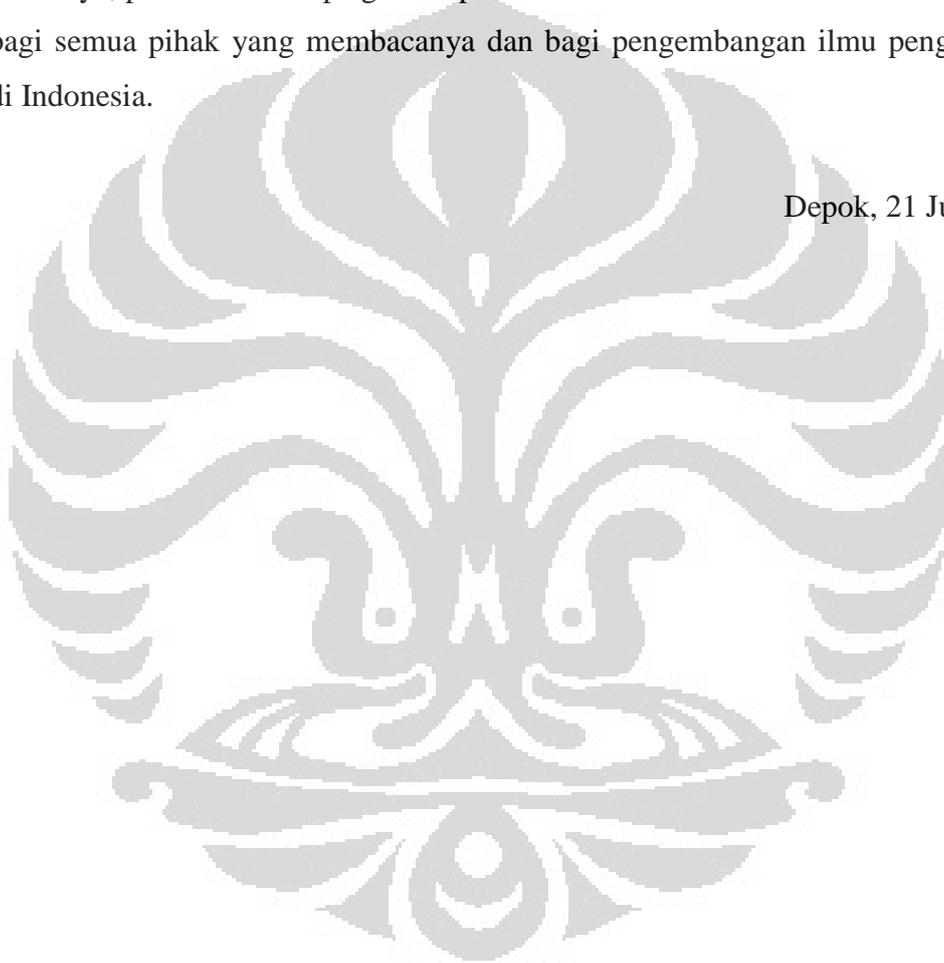
1. Bapak Prof. Dr. Ir. Teuku Yuri M. Zagloel, M.Eng.Sc., selaku dosen pembimbing pertama yang telah menyediakan waktu, tenaga dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini;
2. Bapak Romadhani Ardi, ST, MT., selaku dosen pembimbing kedua yang juga telah menyediakan waktu, tenaga dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini;
3. Bapak Dr. Akhmad Hidayatno, ST., MBT., Ibu Dr. Ing Amalia Suzianti, Bapak Ir. Boy Nurtjahyo, MSIE., Ibu Arian Dhini, ST., MT., dan Ibu Dwinta Utari, ST., MT., MBA. yang membantu dalam memberikan masukan dalam pengerjaan skripsi ini;
4. Saudara Aziz Sutrisno, ST., Christian Tulus, ST., dan Wage Supriadi, ST. yang membantu dalam memberikan masukan dan sumber-sumber pendukung dalam pengerjaan skripsi ini;
5. Ayah, bunda, dan kakak saya serta keluarga besar saya yang turut mendoakan dan mendukung saya dalam pengerjaan skripsi ini;
6. Teman-teman seperjuangan Jody, Zakki, Hadi, Lukat, Lusy dan Rusdy sebagai anggota skripsi Laboratorium Sistem Manufaktur Departemen Teknik Industri UI yang telah berjuang dan saling mendukung dalam menyelesaikan skripsi ini;
7. Seluruh teman-teman Teknik Industri 2008 yang telah berjuang selama empat tahun dari awal semester satu sampai semester delapan ini dan atas kebersamaan yang sungguh luar biasa dan tiada taranya berjuang bersama-sama sebagai satu keluarga;

8. Bapak Prof. Dr. Ir. Teuku Yuri M. Zagloel, M.Eng.Sc., selaku Ketua Program Studi Teknik Industri Universitas Indonesia;
9. Bapak Ir. Djoko Sihono Gabriel, MT., selaku dosen pembimbing akademis; dan
10. Semua pihak lain yang terlibat membantu penulis sehingga skripsi ini dapat terselesaikan dengan baik.

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

Depok, 21 Juni 2012

Penulis



**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

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

Nama : Ilham Winoto

NPM : 0806458901

Program Studi : Teknik Industri

Departemen : Teknik Industri

Fakultas : Teknik

Jenis Karya : Skripsi

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

Pembuatan Model Simulasi dalam Rangka Perancangan *Reconfigurable Manufacturing System*

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

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : 21 Juni 2012

Yang menyatakan



(Ilham Winoto)

ABSTRAK

Nama : Ilham Winoto
Program Studi : Teknik Industri
Judul : Pembuatan Model Simulasi dalam Rangka Perancangan
Reconfigurable Manufacturing System

Reconfigurable Manufacturing System (RMS) adalah sistem manufaktur yang mempunyai kemampuan mengkonfigurasi ulang *hardware*, *software* dan pusat kontrol pada level fungsional dan organisasional supaya secara cepat menyesuaikan kapasitas dan fungsionalitas produksi sebagai respon terhadap pasar atau syarat pengaturan sistem yang berubah secara tiba-tiba. Skripsi ini membahas mengenai pembuatan model simulasi RMS untuk mengetahui karakteristik dan perilaku RMS yang dilihat berdasarkan hasil produksi dan penggunaan waktu produksi. Adapun karakteristik RMS yang diteliti pada model simulasi RMS ini, yaitu *scalability*, *convertibility* dan *integrability*. Model ini dibuat dengan menggunakan perangkat lunak pemodelan berorientasi objek: *Plant Simulation 9.0*. Dari hasil simulasi dan analisis, karakteristik *scalability*, *convertibility* dan *integrability* dapat dipahami dengan lebih baik sehingga diharapkan dapat berguna dalam perancangan RMS di industri manufaktur.

Kata Kunci: *Reconfigurable Manufacturing System* (RMS), model, simulasi, metode simulasi berorientasi objek, *Plant Simulation 9.0*

ABSTRACT

Name : Ilham Winoto
Study Program : Industrial Engineering
Title : Simulation Modeling in order to Design Reconfigurable
Manufacturing System

Reconfigurable Manufacturing System (RMS) is a manufacturing system that has an ability to reconfigure hardware, software and control resources at all of the functional and organizational levels, in order to quickly adjust production capacity and functionality in response to sudden changes in market or in regulatory requirements. This thesis discusses the design of simulation model of RMS to learn the characteristics and behaviors of RMS based on the production result and the use of production time. The characteristics of RMS discussed in RMS simulation model are scalability, convertibility and integrability. This model was created using object-oriented modeling software: Plant Simulation 9.0. From the simulation result and analysis, the characteristics of scalability, convertibility and integrability could be better understood, thus it would be useful in the design of the RMS in the manufacturing industry.

Keyword:

Reconfigurable Manufacturing System (RMS), model, simulation, object-oriented modeling simulation, Plant Simulation 9.0

DAFTAR ISI

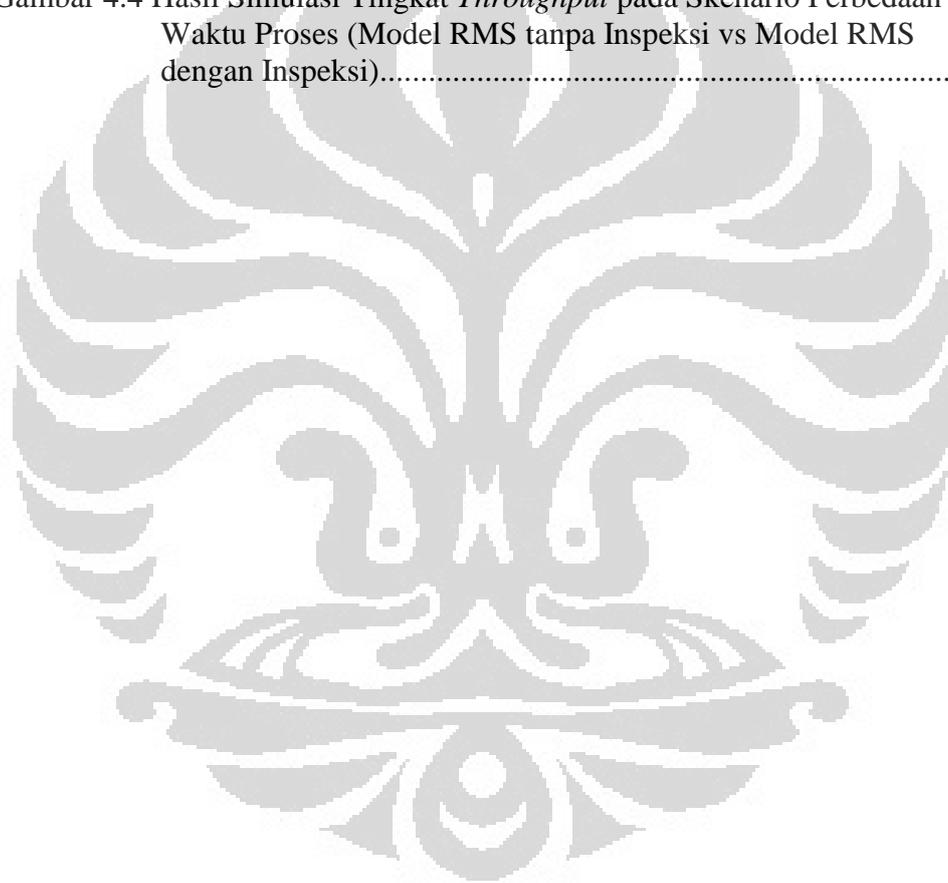
HALAMAN SAMBUTAN	i
HALAMAN JUDUL.....	ii
HALAMAN PERNYATAAN ORISINALITAS.....	iii
HALAMAN PENGESAHAN.....	iv
KATA PENGANTAR	v
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI.....	vii
ABSTRAK	viii
ABSTRACT.....	ix
DAFTAR ISI.....	x
DAFTAR GAMBAR	xii
DAFTAR TABEL.....	xiv
1. PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Diagram Keterkaitan Masalah	4
1.3 Perumusan Masalah.....	5
1.4 Tujuan Penelitian	5
1.5 Batasan Masalah.....	5
1.6 Metodologi Penelitian.....	5
1.7 Sistematika Penulisan	8
2. DASAR TEORI	9
2.1 <i>Reconfigurable Manufacturing System (RMS)</i>	9
2.1.1 Desain Arsitektur	11
2.1.1.1 <i>Reconfigurable Machining Systems</i>	13
2.1.1.2 <i>Reconfigurable Fixturing Systems</i>	13
2.1.1.3 <i>Reconfigurable Assembly Systems</i>	14
2.1.1.4 <i>Reconfigurable Material-Handling Systems</i>	14
2.1.2 Desain Konfigurasi	14
2.1.3 Desain Kontrol.....	17
2.1.3.1 Variabel Rekonfigurabel dan Variabel Proses.....	17
2.1.3.2 Persyaratan Desain Kontrol	18
2.1.4 RMS vs DML vs FMS.....	19
2.1.4.1 Perbandingan Fungsi dan Kapasitas	21
2.1.4.2 Perbandingan Sistem-Biaya dan Kapasitas	22
2.1.5 Teknologi pada RMS.....	22
2.1.5.1 <i>RMS Modules</i>	24
2.1.5.2 <i>RMS Path Finder</i>	24
2.1.5.3 <i>RMS Intelligent Material-Flow Cells</i>	25
2.1.5.4 <i>RMS Intelligent Production Cells</i>	26
2.1.5.5 <i>RMS Intelligent Network</i>	27
2.2 Sistem, Model dan Simulasi	29
2.2.1 Sistem	29
2.2.2 Model.....	31

2.2.2.1 Jenis-jenis Model.....	32
2.2.3 Simulasi	35
2.2.3.1 Tipe Simulasi.....	36
2.2.4 Langkah-langkah Pembuatan Model Simulasi.....	38
2.2.5 Keuntungan Penggunaan Model Simulasi.....	39
2.3 Plant Simulation <i>Software</i>	40
2.4 <i>Unified Modeling Language</i> (UML)	41
2.5 Verifikasi dan Validasi Model Simulasi.....	46
2.5.1 Verifikasi Model Diskrit.....	46
2.5.2 Validasi Model Diskrit	47
3. PENGUMPULAN DAN PENGOLAHAN DATA.....	48
3.1 Pengumpulan Data.....	48
3.1.1 Pembuatan Model RMS <i>Petri Net</i> (Xiuli Meng, 2010).....	50
3.1.1.1 Struktur Kelas dan Blok Dasar Model RMS <i>Petri Net</i>	52
3.1.1.2 Prosedur Memodelkan RMS	56
3.1.1.3 Hasil Model RMS <i>Petri Net</i>	60
3.2 Pengolahan Data	67
3.2.1 Pembuatan Model Simulasi RMS.....	68
3.2.1.1 Kelas Objek Aliran Material Model Simulasi RMS.....	70
3.2.1.2 Kelas Sumber Daya (<i>Resources</i>) Model Simulasi RMS	77
3.2.1.3 Kelas Objek Umum Model Simulasi RMS	80
3.2.1.4 Kelas Objek Berjalan Model Simulasi RMS.....	80
3.2.1.5 Kelas Daftar, Tabel dan Metode Model Simulasi RMS.....	81
3.2.1.6 Kelas Objek Tampilan Model Simulasi RMS	82
3.2.2 Verifikasi dan Validasi Model Simulasi RMS	85
4. ANALISIS DATA	87
4.1 Analisis Karakteristik <i>Scalability</i>	88
4.2 Analisis Karakteristik <i>Convertibility</i>	90
4.3 Analisis Karakteristik <i>Integrability</i>	93
5. KESIMPULAN.....	96
5.1 Kesimpulan.....	96
5.2 Saran	96
6. DAFTAR REFERENSI	97

DAFTAR GAMBAR

Gambar 1.1 Produk Domestik Bruto (PDB) dan Volume Perdagangan Global .	1
Gambar 1.2 Diagram Keterkaitan Masalah.....	4
Gambar 1.3 Metodologi Penelitian	7
Gambar 2.1 Level Rekonfigurabilitas RMS.....	10
Gambar 2.2 Tiga Isu Desain RMS	10
Gambar 2.3 Rekonfigurabilitas RMS.....	13
Gambar 2.4 Arsitektur Sistem.....	15
Gambar 2.5 Variabel Desain Robot pada RMS	16
Gambar 2.6 Dua Kelas Variabel Rekonfigurabel	17
Gambar 2.7 Persyaratan Desain pada Sistem Kontrol RMS.....	18
Gambar 2.8 Hubungan Fungsi dan Kapasitas terhadap DML, FMS dan RMS ..	21
Gambar 2.9 Hubungan Kapasitas & Sistem Biaya pada DML, FMS & RMS ...	22
Gambar 2.10 RMS Modules	24
Gambar 2.11 RMS Path Finder.....	25
Gambar 2.12 RMS Intelligent Material-flow Cells.....	25
Gambar 2.13 RMS Intelligent Production Cells	26
Gambar 2.14 RMS Intelligent Network.....	27
Gambar 2.15 RMS pada LSM DTI-UI	28
Gambar 3.1 Contoh Struktur Hierarkis RMS.....	51
Gambar 3.2 Aliran Informasi dan Material pada RMS.....	52
Gambar 3.3 Struktur Model RMS.....	60
Gambar 3.4 Macro-level petri net pada level sistem.....	62
Gambar 3.5 Detail Macro-level petri net pada level sistem.....	62
Gambar 3.6 Macro-level petri net RMC1	63
Gambar 3.7 Macro-level petri net RMC2	63
Gambar 3.8 Properti Warna dari Robot Macro-Place.....	65
Gambar 3.9 Model RMS Petri Net Konfigurasi 2 (dengan inspeksi)	66
Gambar 3.10 Tampilan Model RMS Petri Net Menggunakan Renew	67
Gambar 3.11 AGV Tipe Satu.....	70
Gambar 3.12 Atribut AGV Tipe Satu	71
Gambar 3.13 AGV Tipe Dua	72
Gambar 3.14 Atribut AGV Tipe Dua.....	72
Gambar 3.15 AGV Tipe Tiga.....	73
Gambar 3.16 Atribut AGV Tipe Tiga.....	74
Gambar 3.17 Robot	75
Gambar 3.18 Atribut Robot	75
Gambar 3.19 Konveyor Otomatis	77
Gambar 3.20 AS/RS.....	77
Gambar 3.21 Atribut AS/RS	78
Gambar 3.22 Mesin CNC pada Machining Center (MC)	78
Gambar 3.23 Atribut Mesin CNC	79
Gambar 3.24 Buffer pada Sel RMC2.....	79
Gambar 3.25 Atribut Buffer.....	80
Gambar 3.26 Atribut MUs	81

Gambar 3.27 Contoh Metode beserta Syntax	82
Gambar 3.28 Contoh Pusat Kontrol beserta Atribut	82
Gambar 3.29 Model Simulasi RMS Konfigurasi Pertama (tanpa Inspeksi)	83
Gambar 3.30 Model Simulasi RMS Konfigurasi Kedua (dengan Inspeksi)	84
Gambar 3.31 Status Verifikasi <i>Event Debugger</i> pada Model Simulasi RMS.....	86
Gambar 4.1 Hasil Simulasi Tingkat <i>Throughput</i> pada Skenario Perbedaan Waktu Interval Kedatangan Komponen (Model Simulasi RMS tanpa Inspeksi)	90
Gambar 4.2 Hasil Simulasi Tingkat <i>Throughput</i> pada Skenario Perbedaan Waktu Proses (Model Simulasi RMS tanpa Inspeksi)	92
Gambar 4.3 Hasil Simulasi Tingkat <i>Throughput</i> pada Skenario Perbedaan Waktu Interval Kedatangan Komponen (Model RMS tanpa Inspeksi vs Model RMS dengan Inspeksi).....	94
Gambar 4.4 Hasil Simulasi Tingkat <i>Throughput</i> pada Skenario Perbedaan Waktu Proses (Model RMS tanpa Inspeksi vs Model RMS dengan Inspeksi).....	95



DAFTAR TABEL

Tabel 2.1	Karakteristik RMS	12
Tabel 2.2	Perbandingan DML dengan FMS	20
Tabel 2.3	Tabel Perbandingan DML, FMS dan RMS	21
Tabel 2.4	Kategori Teknologi	23
Tabel 3.1	Empat Kelas pada Entitas Dasar RMS.....	54
Tabel 3.2	Aliran sub-proses untuk setiap komponen pada level sistem	64
Tabel 3.3	Aliran sub-proses untuk setiap komponen pada level sel RMC1	64
Tabel 3.4	Aliran sub-proses tambahan pada model RMS konfigurasi ke-2	67
Tabel 3.5	Tugas Setiap Robot pada Model Simulasi RMS.....	74
Tabel 3.6	Fungsi Tiap Konveyor Otomatis.....	76
Tabel 3.7	Status Verifikasi Sub-proses pada Level Sistem.....	85
Tabel 3.8	Status Verifikasi Sub-proses pada Level RMC1.....	85
Tabel 3.9	Status Verifikasi Sub-proses pada Level RMC2.....	86
Tabel 4.1	Skenario Waktu Interval Kedatangan Komponen	89
Tabel 4.2	Hasil Simulasi Tingkat Okupasi pada Skenario Perbedaan Waktu Interval Kedatangan Komponen (Model Simulasi RMS tanpa Inspeksi).....	89
Tabel 4.3	Hasil Simulasi Tingkat <i>Throughput</i> pada Skenario Perbedaan Waktu Interval Kedatangan Komponen (Model Simulasi RMS tanpa Inspeksi).....	89
Tabel 4.4	Skenario Waktu Proses pada Permesinan	91
Tabel 4.5	Hasil Simulasi Tingkat Okupasi pada Skenario Perbedaan Waktu Proses (Model Simulasi RMS tanpa Inspeksi)	91
Tabel 4.6	Hasil Simulasi Tingkat <i>Throughput</i> pada Skenario Perbedaan Waktu Proses (Model Simulasi RMS tanpa Inspeksi)	92
Tabel 4.7	Hasil Simulasi Tingkat Okupasi pada Skenario Perbedaan Waktu Interval Kedatangan Komponen (Model RMS tanpa Inspeksi vs Model RMS dengan Inspeksi)	93
Tabel 4.8	Hasil Simulasi Tingkat <i>Throughput</i> pada Skenario Perbedaan Waktu Interval Kedatangan Komponen (Model RMS tanpa Inspeksi vs Model RMS dengan Inspeksi)	93
Tabel 4.9	Hasil Simulasi Tingkat Okupasi pada Skenario Perbedaan Waktu Proses (Model RMS tanpa Inspeksi vs Model RMS dengan Inspeksi).....	94
Tabel 4.10	Hasil Simulasi Tingkat <i>Throughput</i> pada Skenario Perbedaan Waktu Proses (Model RMS tanpa Inspeksi vs Model RMS dengan Inspeksi).....	95

BAB 1

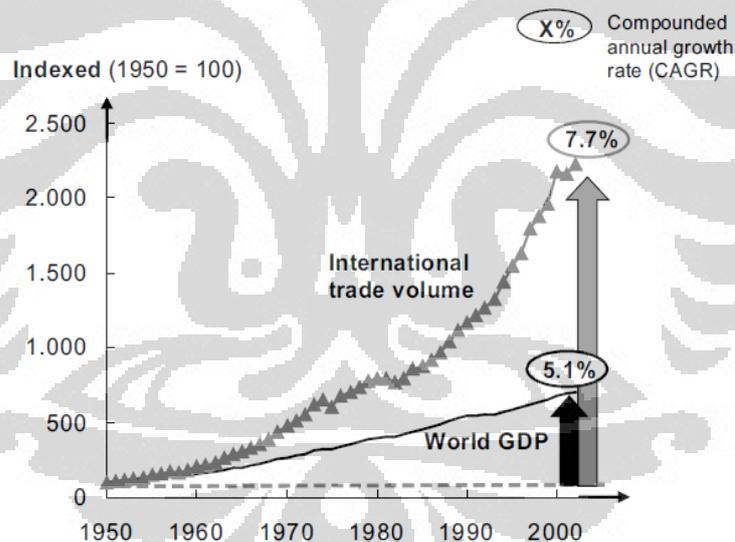
PENDAHULUAN

1. PENDAHULUAN

Pada bab ini berisi tentang pembahasan mengenai latar belakang, diagram keterkaitan masalah, perumusan masalah, tujuan penelitian, batasan masalah, metodologi penelitian dan sistematika penulisan yang digunakan.

1.1 Latar Belakang

Dalam beberapa dekade terakhir globalisasi telah menjadi topik penting. Produk domestik bruto (PDB) dunia tumbuh dengan laju pertumbuhan tahunan gabungan (CAGR) sebesar 5,1% selama lima puluh tahun terakhir, sementara pertumbuhan perdagangan global telah melebihi tren PDB tersebut dengan CAGR sebesar 7,7% (Abele et al., 2004).



Gambar 1.1 Produk Domestik Bruto (PDB) dan Volume Perdagangan Global

Sumber : World Trade Organization (WTO), 2001

Besarnya pertumbuhan perdagangan global ini tidak lepas dari perkembangan pesat sistem manufaktur. Berdasarkan *International Conference on Production Engineering* (CIRP, 1983), sistem manufaktur adalah bagian dari fungsi dan sistem produksi yang berupa serangkaian kegiatan dan operasi yang saling terkait melibatkan desain, pemilihan bahan, perencanaan, produksi

manufaktur, jaminan mutu, manajemen dan pemasaran produk-produk dari industri manufaktur.

Lingkungan manufaktur memiliki dampak yang besar pada kinerja sistem manufaktur. Lingkungan manufaktur saat ini membuat beberapa persyaratan penting bagi sistem manufaktur (Bi et al., 2007). Persyaratan tersebut secara singkat akan dijelaskan sebagai berikut :

- *Short lead-time*

Product lead-time mempengaruhi kinerja sistem manufaktur dengan cara yang berbeda-beda (Smith dan Reinertsen, 2007): 1) jika produk diperkenalkan terlalu dini, hal ini akan menjadi keuntungan bagi kompetitor yang ketinggalan untuk meniru produk tersebut; 2) pengenalan produk terlalu dini akan meningkatkan penjualan secara dramatis. Semakin awal suatu produk dibuat, semakin baik prospek untuk mendapatkan pangsa pasar yang lebih besar; 3) suatu produk baru memberikan batas keuntungan yang lebih besar.

- Variasi produk

Variasi produk membuat produk menjadi lebih fleksibel dan semakin sesuai dengan yang dibutuhkan atau diinginkan oleh pelanggan. Sebuah sistem manufaktur dipaksa untuk menghasilkan variasi produk yang lebih beragam untuk memenuhi kebutuhan yang lebih terfragmentasi, rumit dan personal (Tseng dan Du, 1998; Fralix, 2001).

- Permintaan semakin fluktuatif

Permintaan dari banyak produk semakin fluktuatif sejak: 1) relung pasar terbatas dan diperebutkan oleh kompetitor yang semakin banyak; 2) siklus hidup produk baru menjadi semakin pendek dan daya tahan produk menjadi lebih lama. Hal ini menyebabkan berbagai generasi produk ada di pasar pada waktu yang sama; 3) kustomisasi produk telah terfragmentasi mengikuti tuntutan pasar menjadi porsi-porsi permintaan yang kecil (Bi et al., 2007).

- Harga murah

Harga produk adalah fitur utama untuk sebagian besar pelanggan. Di satu sisi, pasar global menawarkan pelanggan dengan lebih banyak pilihan

untuk membeli produk dengan harga yang lebih rendah tetapi dengan kualitas dan pelayanan yang sama. Di sisi lain, harga sangat tergantung oleh waktu dan batas harga dapat mencapai puncaknya segera setelah suatu produk diperkenalkan ke pasar (Bi et al., 2007).

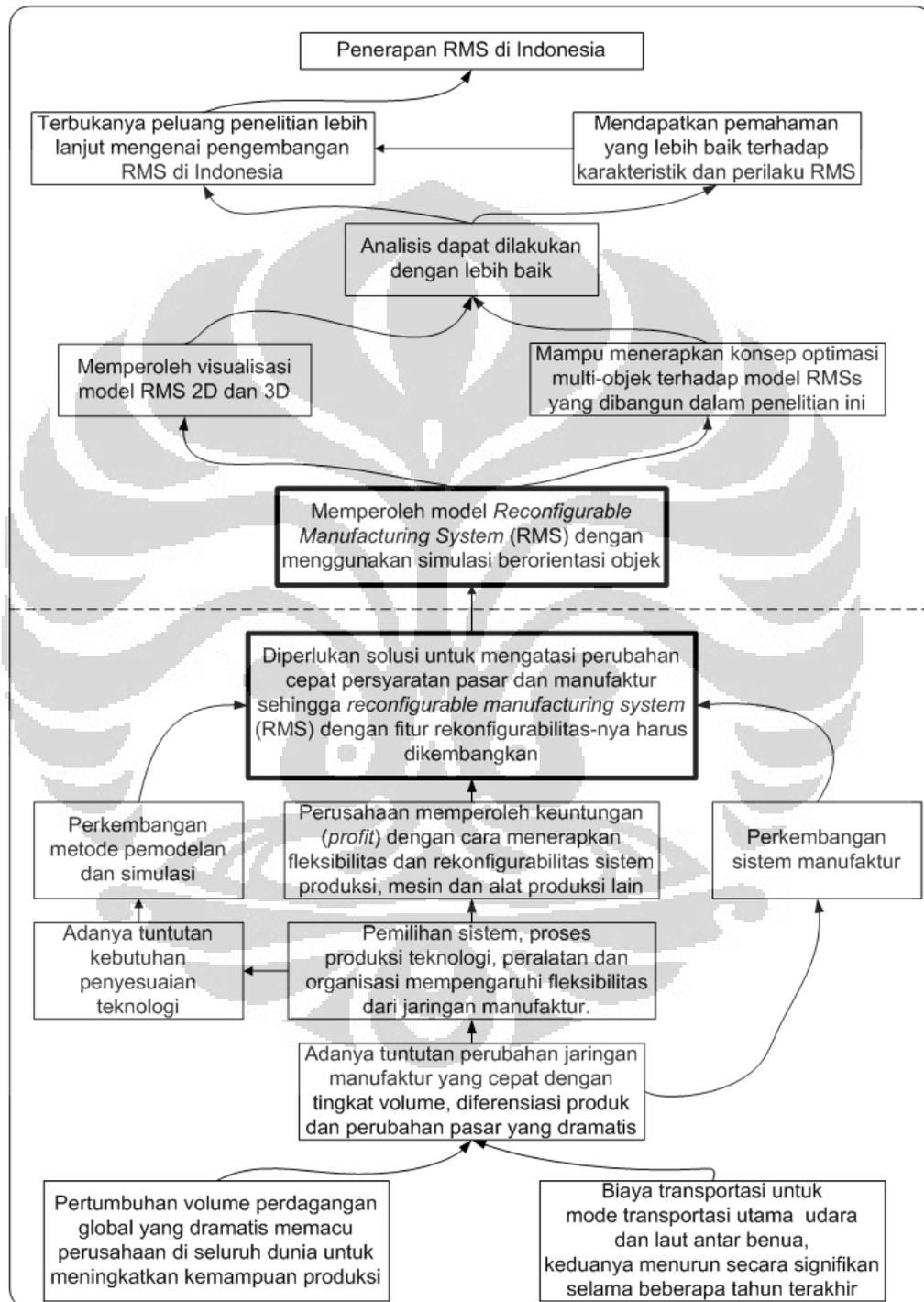
Menurut Xiuli Meng (2010), dalam rangka menyediakan berbagai produk yang mampu merespon pasar dengan cepat mengikuti tuntutan lingkungan seperti, *product lead-time* lebih pendek, variasi produk lebih beragam, permintaan semakin fluktuatif dan harga produk lebih murah, maka rekonfigurabilitas adalah konsep penting yang telah diterima secara luas di dunia industri dan akademisi. Secara harfiah, rekonfigurabilitas dapat diartikan sebagai konsep untuk menggambarkan kemampuan dan perilaku untuk berubah secara cepat sesuai dengan konfigurasi-konfigurasi yang telah dirancang (Estrin, 1960s). Konsep rekonfigurabilitas inilah yang menjadi kata kunci terpenting dari *Reconfigurable Manufacturing System* (RMS).

Reconfigurable Manufacturing System (RMS) didefinisikan sebagai sistem manufaktur yang mempunyai kemampuan mengkonfigurasi ulang *hardware*, *software* dan pusat kontrol pada level fungsional dan organisasional supaya secara cepat menyesuaikan kapasitas dan fungsionalitas produksi sebagai respon terhadap pasar atau syarat pengaturan sistem yang berubah secara tiba-tiba (Koren et al., 1999). RMS memfasilitasi peluncuran produksi produk baru dengan cepat dan memungkinkan untuk penyesuaian jumlah produksi yang bervariasi dan tak terduga (Mehrabi et al., 2000).

Salah satu cara yang paling efektif untuk mendapatkan pemahaman dan gambaran yang lebih baik mengenai karakteristik dan perilaku RMS adalah dengan memperoleh model simulasi RMS dengan metode berorientasi objek. Simulasi berorientasi objek terdiri dari satu set objek yang berinteraksi satu sama lain dari waktu ke waktu. Simulasi berorientasi objek memiliki daya tarik intuitif yang besar dalam aplikasi karena lebih mudah untuk melihat dunia nyata terdiri sebagai objek. Simulasi berbasis objek memungkinkan pemetaan satu-ke-satu antara objek dalam sistem manufaktur yang dimodelkan dan abstraksi mereka dalam model simulasi (Narayanan et al., 1998).

1.2 DIAGRAM KETERKAITAN MASALAH

Berikut ini adalah diagram keterkaitan masalah yang menggambarkan keterkaitan antar masalah yang ditampilkan pada Gambar 1.2.



Gambar 1.2 Diagram Keterkaitan Masalah

1.3 PERUMUSAN MASALAH

Karena mahalny biaya penerapan RMS dalam suatu industri manufaktur maka fase perancangan menjadi hal yang sangat penting. Oleh karena itu, penelitian ini berupaya untuk meminimalisasi resiko dan potensi kerugian sebagai dampak penerapan RMS dengan cara terlebih dahulu memperoleh model simulasi RMS. Pada model simulasi ini akan dijelaskan karakteristik dan perilaku RMS yang berperan penting dalam menentukan keberhasilan penerapan rancangan RMS dalam industri manufaktur.

1.4 TUJUAN PENELITIAN

Berdasarkan diagram keterkaitan masalah dan perumusan masalah, maka tujuan dari penelitian ini adalah memperoleh model simulasi RMS dengan menggunakan metode simulasi berorientasi objek untuk mengetahui karakteristik dan perilaku RMS secara lebih baik yang dilihat berdasarkan hasil produksi dan penggunaan waktu produksi.

1.5 BATASAN MASALAH

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

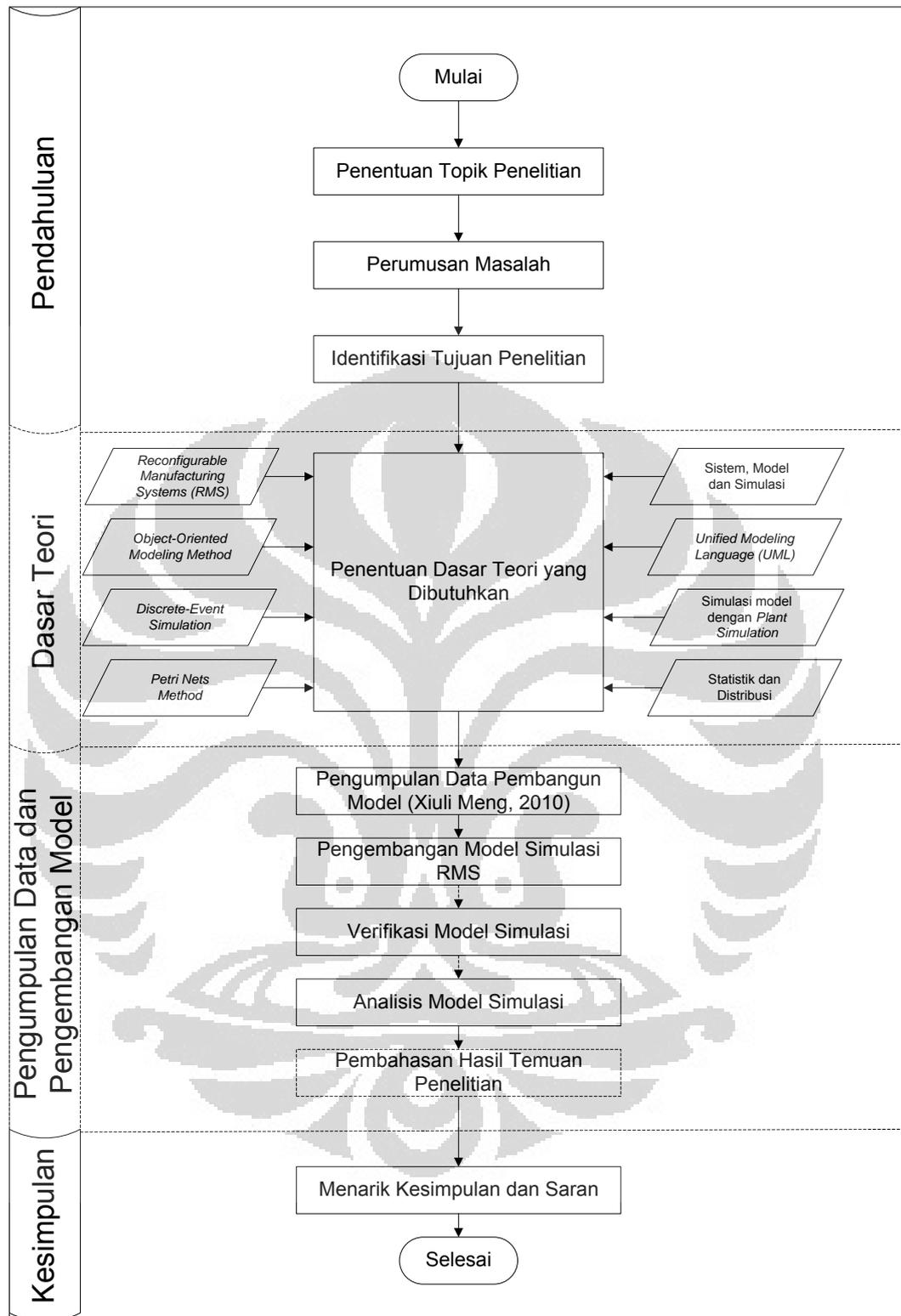
1. Data yang digunakan untuk membangun model simulasi merupakan data hipotetis (*hypothetical data*) yang berasal dari data model matematis hasil penelitian Xiuli Meng (2010).
2. Perangkat lunak yang digunakan untuk membangun model simulasi adalah *Tecnomatix Plant Simulation 9.0*.
3. Karakteristik RMS yang dianalisis pada model simulasi dibatasi menjadi tiga dari enam karakteristik, yaitu: *scalability*, *convertibility* dan *integrability*.

1.6 METODOLOGI PENELITIAN

Metodologi penelitian yang digunakan dalam melakukan penelitian ini dapat dilihat pada Gambar 1.3.

Secara umum, metodologi penelitian ini dapat dijelaskan sebagai berikut :

1. Penentuan topik dan tujuan penelitian
Dalam tahap ini dilakukan perumusan masalah yang akan dijadikan topik dalam penelitian serta tujuan dilakukannya penelitian.
2. Penentuan dasar teori yang dibutuhkan
Pada tahap ini penulis menentukan dasar teori yang dibutuhkan dalam penelitian ini. Dasar teori yang dibutuhkan meliputi pengertian sistem, model & simulasi, *Reconfigurable Manufacturing System (RMS)*, *object-oriented modeling method*, *discrete-event simulation*, *petri nets method*, *Unified Modeling Language (UML)*, *Tecnomatix Plant Simulation* dan statistik & distribusi.
3. Pengumpulan data
Tahap selanjutnya adalah mencari data hipotetis (*hypothetical data*) yang akan digunakan sebagai data sekunder untuk memperoleh model simulasi.
4. Pembuatan model simulasi
Pembuatan model simulasi RMS berdasarkan data hipotetis yang telah diperoleh.
5. Verifikasi dan validasi model
Melakukan uji verifikasi dan validasi terhadap model simulasi yang telah dibuat.
6. Analisis model
Pada tahap ini akan dilakukan analisis model simulasi yang telah dijalankan.
7. Penentuan dan pembahasan hasil temuan penelitian
Tahap selanjutnya adalah menjabarkan hasil temuan penelitian setelah analisis model simulasi selesai dilakukan.
8. Penarikan kesimpulan dan saran
9. Peluang penelitian di masa mendatang



Gambar 1.3 Metodologi Penelitian

1.7 SISTEMATIKA PENULISAN

Secara umum, pembahasan penelitian ini terbagi atas beberapa bab dengan sistematika penulisan sebagai berikut:

Bab satu merupakan bab pendahuluan yang menjelaskan latar belakang penelitian, diagram keterkaitan masalah, perumusan masalah, tujuan penelitian, ruang lingkup penelitian, metodologi penelitian dan sistematika penulisan. Pada bab dua, penulis akan memaparkan dasar-dasar teori yang berhubungan dan mendukung penelitian ini. Dasar teori yang dibahas meliputi pengertian sistem, model & simulasi, *Reconfigurable Manufacturing System (RMS)*, *object-oriented modeling method*, *discrete-event simulation*, *petri nets method*, *Unified Modeling Language (UML)*, *Tecnomatix Plant Simulation* dan statistik & distribusi.

Pengumpulan data akan dijabarkan pada bab tiga, dimana pada bab ini akan dibahas mengenai data hipotetis model matematis yang penulis peroleh. Pembuatan model simulasi dan analisis model akan dilakukan pada bab empat. Sementara itu, bab lima merupakan kesimpulan akhir dan saran penulis dari keseluruhan penelitian ini beserta pembahasan peluang penelitian di masa mendatang.

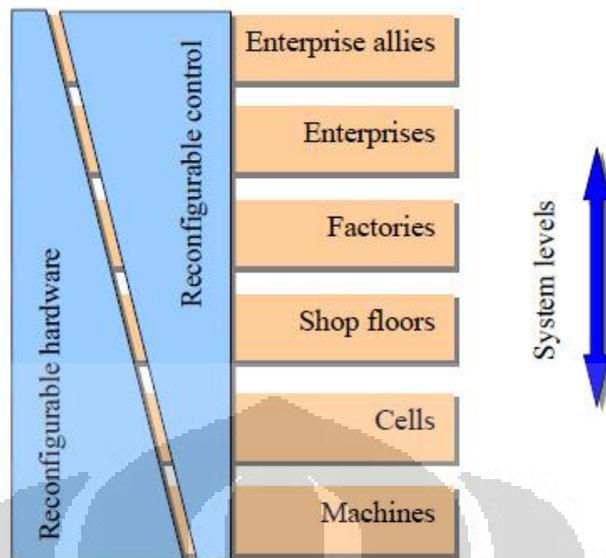
BAB 2 DASAR TEORI

2.1 *Reconfigurable Manufacturing System (RMS)*

Sistem manufaktur mentransformasi bahan baku menjadi produk. Tujuan pokoknya adalah untuk memperoleh keuntungan (*profit*), reputasi dan pangsa pasar yang terbaik. Setiap perusahaan hanya bisa bertahan jika tujuan tersebut diraih secara baik dan tepat. Sementara itu, lingkungan manufaktur saat ini mempunyai beberapa persyaratan kritikal yang harus dipenuhi perusahaan manufaktur, seperti *short lead time, more variants, low & fluctuating volumes* dan *low price* (Ishii et al., 1995; NRC, 1998; Stake, 1999; & NRC, 2000). Oleh karena itu, dibutuhkan sistem manufaktur yang mampu memenuhi persyaratan tersebut secara efektif, yaitu *Reconfigurable Manufacturing System (RMS)*.

Menurut Koren et al. (1999) yang disempurnakan pada Konferensi ke-3 RMS di *University of Michigan*, 10-12 Mei 2005, RMS adalah sistem manufaktur yang mempunyai kemampuan mengkonfigurasi ulang *hardware, software* dan pusat kontrol pada level fungsional dan organisasional supaya secara cepat menyesuaikan kapasitas dan fungsionalitas produksi sebagai respon terhadap pasar atau syarat pengaturan sistem yang berubah secara tiba-tiba.

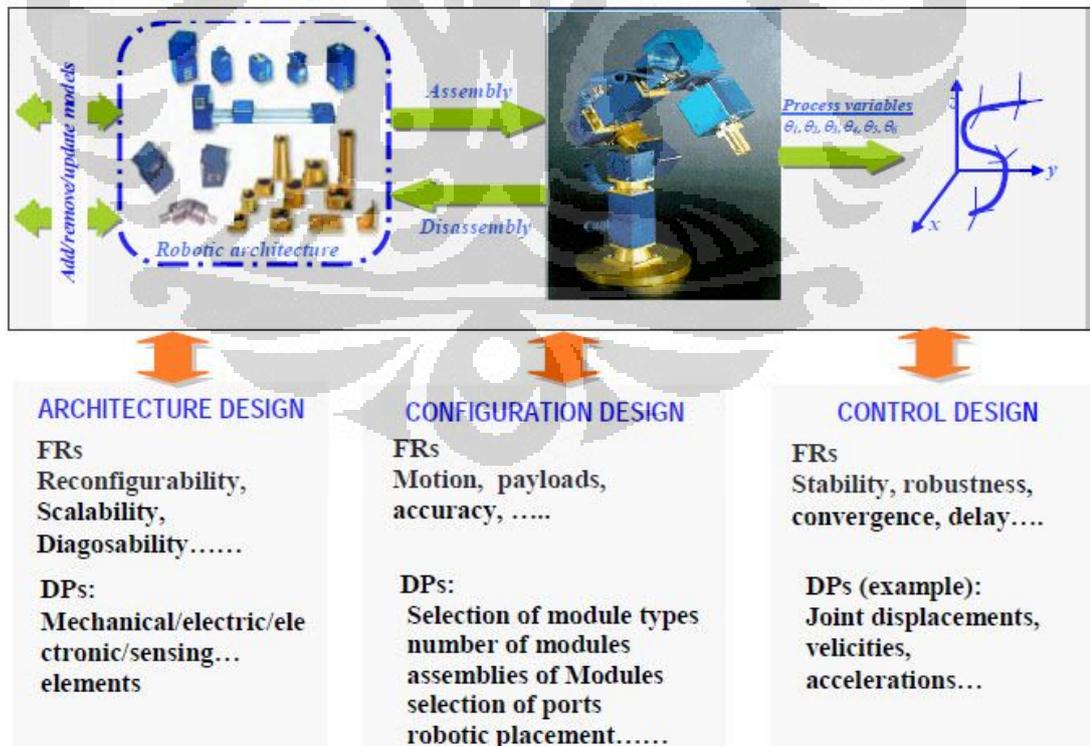
Sistem rekonfigurabilitas RMS bisa diklasifikasikan ke dalam level-level dimana aksi rekonfigurabel tersebut terjadi. Pada Gambar 2.1, rekonfigurabilitas pada level yang lebih rendah diraih dengan merubah sumber daya *hardware* dan rekonfigurabilitas pada level yang lebih tinggi diraih dengan merubah sumber daya *software* dan/atau memilih metode atau struktur organisasi alternatif yang tepat. Namun, tiap level rekonfigurabilitas ini biasanya bekerja bersama-sama/simultan supaya sistem rekonfigurabilitas bisa berjalan baik.



Gambar 2.1 Level Rekonfigurabilitas RMS

Sumber: Bi et al. (2007)

Ada tiga isu desain kritikal yang melekat pada RMS meliputi desain arsitektur, desain konfigurasi dan desain kontrol. Ketiga isu desain ini dapat dilihat pada Gambar 2.2.



Gambar 2.2 Tiga Isu Desain RMS

Sumber: Amtec-Robotics (2006) dan Chen (2001)

Desain arsitektur menentukan komponen sistem dan interaksinya. Komponen sistem ini dibentuk secara modular. Interaksi adalah pilihan yang muncul ketika modul-modul komponen tersebut dirakit. Arsitektur RMS harus didesain untuk memproduksi sebanyak mungkin varians sistem sehingga sistem bisa merespons perubahan dan ketidakpastian. Desain arsitektur dilibatkan pada fase desain sistem.

Desain konfigurasi menentukan konfigurasi sistem yang berada di bawah arsitektur sistem untuk keperluan tugas spesifik. Suatu konfigurasi adalah suatu *assembly* dari modul-modul terpilih di mana suatu konfigurasi tersebut bisa memenuhi tugas yang diberikan secara optimal. Desain konfigurasi dilibatkan pada fase aplikasi sistem.

Desain kontrol menentukan variabel-variabel proses yang cocok (*joint displacements, velocities, dll*) supaya suatu konfigurasi bisa dioperasikan untuk memenuhi tugas yang diberikan secara memuaskan. Desain kontrol dilibatkan pada fase operasi sistem.

2.1.1 Desain Arsitektur

Arsitektur RMS didesain untuk mempunyai satu set karakteristik kunci RMS yang meliputi *modularity, scalability, integrability, customization, convertibility*, dan *diagnosability* (Mehrabi et al. 2000, 2002).

Modularity menyatakan bahwa baik elemen *software* maupun *hardware*, semuanya modular. *Scalability* berarti sistem mampu menyesuaikan skala volume produk sesuai kebutuhan. *Integrability* berarti sistem dan komponen sistem didesain untuk sistem terintegrasi dan mampu menerima teknologi baru di masa depan. *Customization* berkaitan dengan kemampuan mereduksi biaya mesin dan sistem karena memiliki kemampuan untuk memproduksi suatu keluarga komponen (*part family*), bukan komponen tunggal (seperti DML) atau komponen apapun (seperti FMS). Dalam konteks RMS, yang dimaksud *part family* adalah semua komponen yang memiliki kemiripan bentuk dan fitur geometris, level toleransi yang sama, perlakuan proses yang sama dan di dalam rentang biaya yang sama. *Convertibility* mengizinkan perubahan (*changeover*) secara cepat antara produk yang sedang eksis dengan tingkat adaptasi sistem yang cepat

terhadap produk masa depan. *Diagnosability* berarti kemampuan untuk mengidentifikasi secara cepat sumber masalah kualitas dan reliabilitas yang terjadi pada sistem.

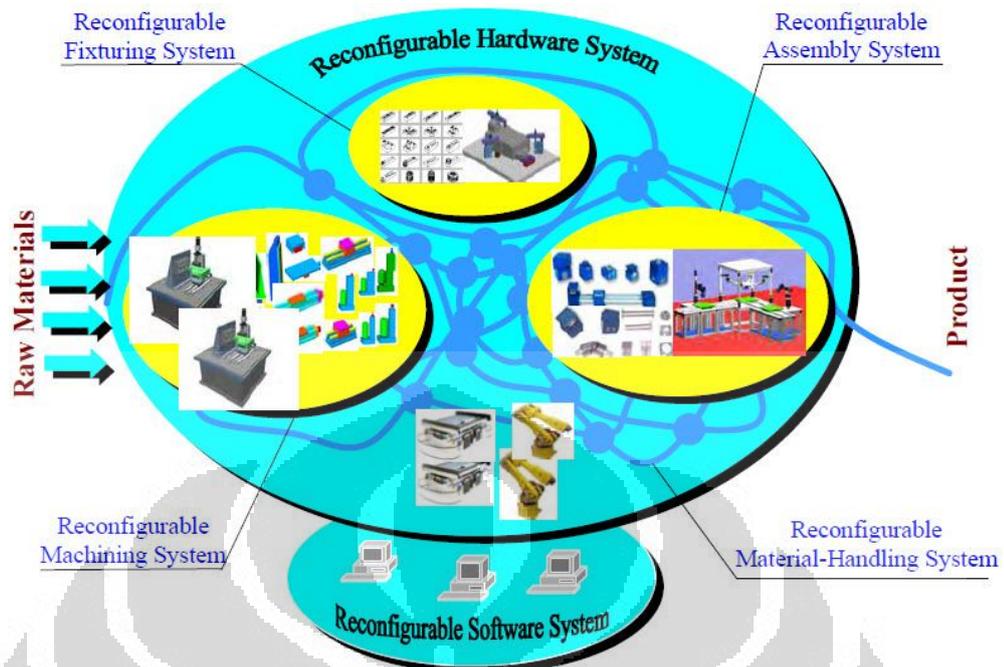
Ada hubungan antara karakteristik dengan tujuan RMS seperti yang ditunjukkan oleh Tabel 2.1. *Modularity*, *integrability*, *convertibility* dan *diagnosability* mengurangi waktu dan usaha rekonfigurasi sehingga meningkatkan respons sistem. Kecuali *modularity*, kelima karakteristik ini berkontribusi untuk mengurangi biaya *lifetime* sistem karena mampu menghadapi keadaan perubahan pasar dan bisa ‘terus hidup’ meski terjadi perubahan pasar, perubahan permintaan pelanggan dan perubahan teknologi proses yang dramatis. Membangun suatu sistem modular mungkin lebih mahal karena adanya biaya tambahan *interface*. Namun, jika komponen modular telah menjadi suatu komoditas, maka skala ekonomis akan mengurangi biaya sistem secara keseluruhan.

Tabel 2.1 Karakteristik RMS

	Reconfigure time	Productivity	Life-cycle cost
Modularity Components are modular	•		
Integrability Interfaces for rapid integration	•		•
Customization Flexibility limited to part family		•	•
Scalability Designed for capacity change		•	•
Convertibility Designed for functionality change	•	•	•
Diagnosability Designed for diagnostics	•	•	•

Sumber: Koren et al. (1999)

Pada Gambar 2.3 dapat dilihat bahwa sistem pada RMS terdiri dari dua komponen sub-sistem, yaitu *reconfigurable hardware system* dan *reconfigurable software system*. Sistem *hardware* meliputi *reconfigurable machining systems*, *reconfigurable fixturing systems*, *reconfigurable assembly systems* dan *reconfigurable material-handling systems*.



Gambar 2.3 Rekonfigurabilitas RMS

Sumber: Koren et al. (1999), Chen (2001) & Amtec-Robotics (2006)

2.1.1.1 *Reconfigurable Machining Systems*

Perangkat mesin modular sudah ada di pasar lebih dari satu dekade yang lalu. Perangkat mesin modular ini secara umum digunakan untuk meningkatkan jumlah varians produk. *Parallel Kinematic Machines* (PKMs) adalah salah satu tipe perangkat mesin modular (Cooke et al. 1995, Bostelman et al. 2000, Katz et al. 2002, Fassi dan Wiens 2000). Sayangnya, teori dan metodologi untuk pengembangan secara komersial PKMs tidak dibangun dengan baik. Kebanyakan PKMs yang tersedia di pasar berharga mahal dengan akurasi yang tidak lebih baik dibanding dengan mesin konvensional. Kelas baru dari perangkat mesin modular ini telah diteliti di University of Michigan yang disesuaikan untuk memenuhi persyaratan terkait dengan karakteristik RMS seperti, *modularity*, *integrability*, *scalability*, *convertibility* dan *diagnosability* (Koren et al., 1999).

2.1.1.2 *Reconfigurable Fixturing Systems*

Penelitian mengenai *reconfigurable fixturing systems* bisa ditarik ke belakang pada era 1960-an, ketika konsep modular pertama kali diperkenalkan.

Suatu *modular fixture* disusun oleh banyak modul-modul dasar meliputi *locators*, *claspers* dan *connectors*. Beberapa *reconfigurable fixturing systems* telah diaplikasikan pada industri robot, karena industri robot fleksibel untuk diprogram untuk beragam tugas yang berbeda (Shirinzadeh 1993, Benhabib et al. 1991). *Reconfigurable fixturing systems* yang lain berbentuk integral dan diimplementasikan dengan material khusus atau komponen yang mampu diubah-ubah (Hass et al. 2002, Valjavec & Hardt 1999). Suatu peninjauan kritikal mengenai pengembangan *reconfigurable fixturing systems* sudah dilakukan (Bi & Zhang, 2001).

2.1.1.3 Reconfigurable Assembly Systems

Reconfigurable assembly systems biasanya dibangun oleh robot (Giusti et al. 1994, Arai et al. 2001, Heilala & Voho 2001, Sugi et al. 2001). Jumlah dan tipe perlengkapan *assembly* bisa diganti-ganti untuk menyesuaikan/memenuhi persyaratan produk. Sistem rekonfigurabilitas bisa dikembangkan lebih jauh jika sistem robot itu sendiri bisa dimodularisasi. Pada kenyataannya, desain robot modular merupakan salah satu penelitian utama komunitas robot (Hooper & Tesar 1994, Chen 1994, Pritschow & Wurst 1996, Unsal et al. 1999).

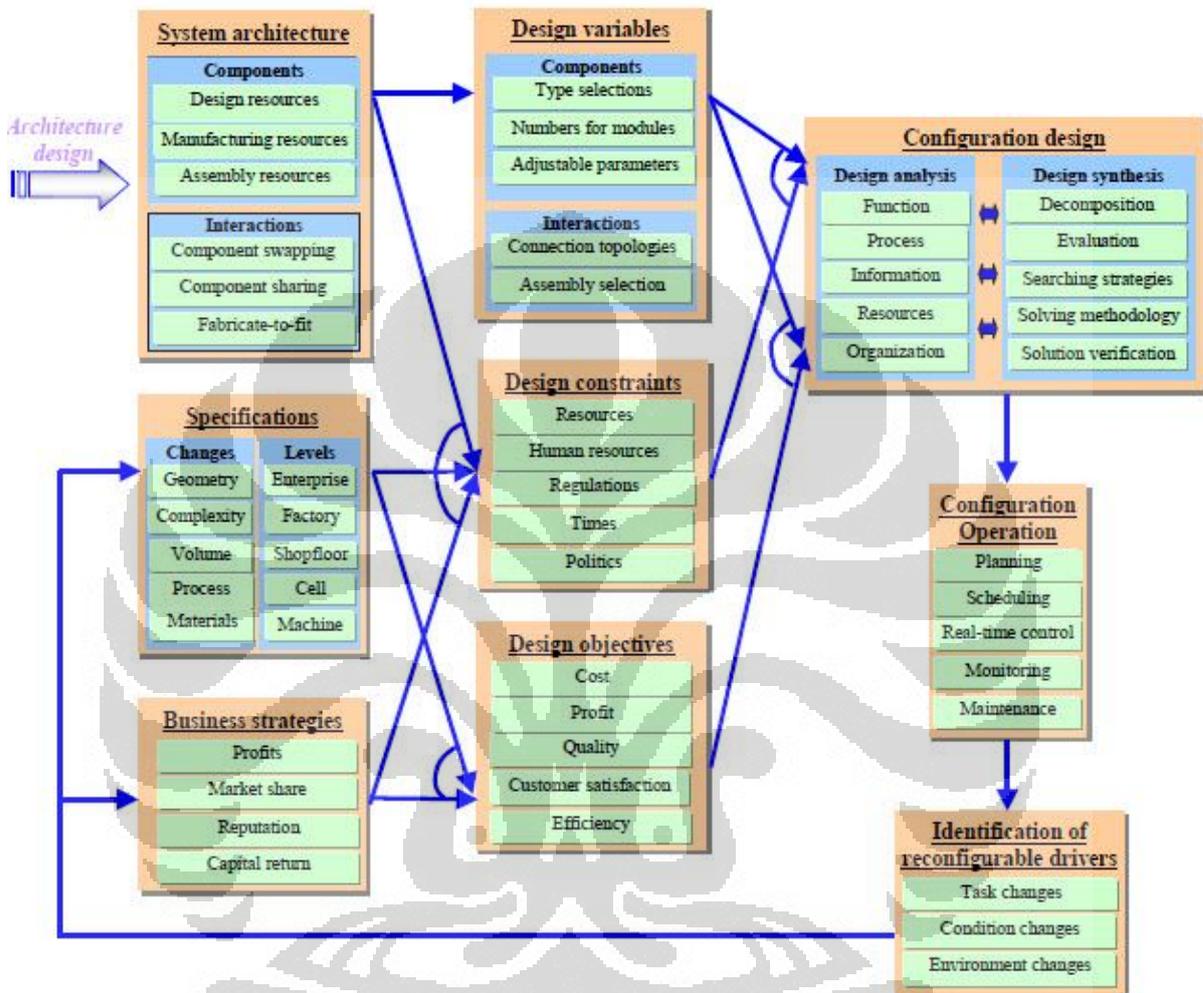
2.1.1.4 Reconfigurable material-handling systems

Beberapa *reconfigurable material-handling systems* sedang dalam pengembangan. Fukuda dan Takagawa (2000) telah mendesain sistem transfer fleksibel untuk varians produk berskala besar. Komponen sistem utamanya adalah robot kendali otomatis. Ho et al. (1997) telah mengembangkan sistem konveyor rekonfigurabel dimana mengizinkan untuk merubah volume produk pada *real-time*. Sementara itu, *Automation Tooling System* (ATS) di Kanada telah mengembangkan konveyor yang mampu diprogram secara otomatis dalam menjalankan tugasnya (Mellor, 2002).

2.1.2 Desain Konfigurasi

Seperti yang terlihat pada Gambar 2.4, arsitektur sistem menentukan tipe dan pilihan *assembly* apa saja yang tersedia untuk mengeksekusi komponen

sistem. Oleh karena itu, arsitektur sistem menentukan varians konfigurasi sistem apa saja yang bisa diproduksi. Desain konfigurasi bisa diformulasikan sebagai masalah optimasi.

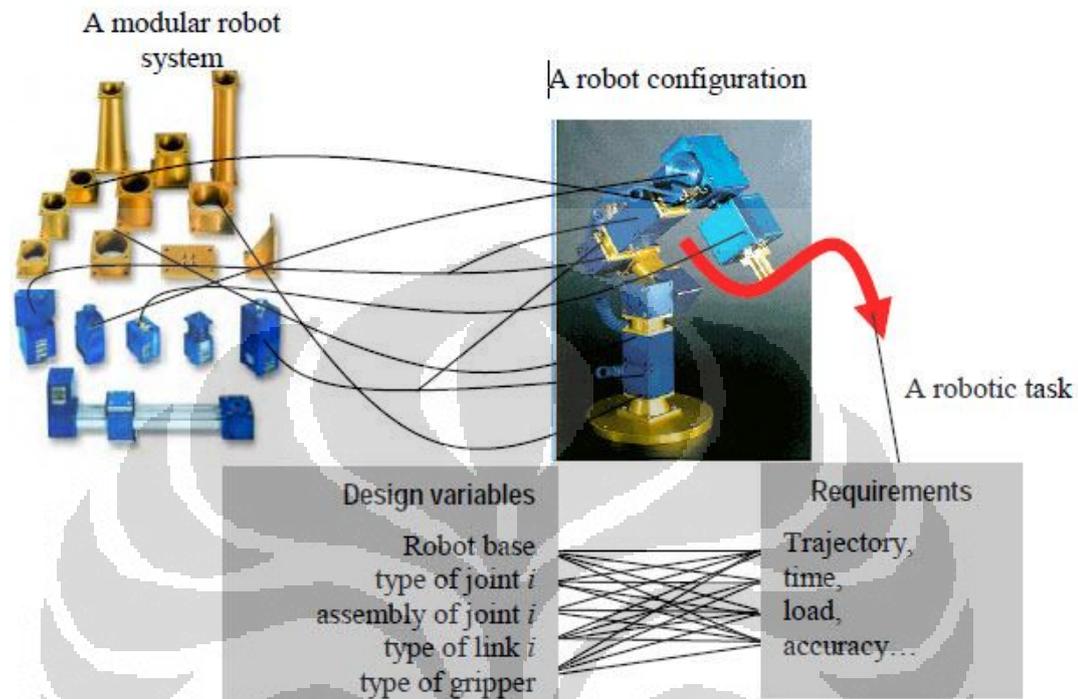


Gambar 2.4 Arsitektur Sistem

Sumber: Bi et al. (2007)

Pada Gambar 2.5, variabel desain, termasuk pemilihan tipe modul, jumlah modul dan parameter *adjustable* internal dalam suatu modul digunakan untuk merepresentasikan tugas spesifik dan strategi bisnis. Desain konfigurasi melibatkan analisis dan sintesis desain. Analisis desain membangun pemetaan dari variabel desain ke batasan desain dan dari desain variabel ke tujuan desain. Sementara itu, sintesis desain menemukan solusi optimal dari semua kandidat konfigurasi. Untuk suatu RMS, desain konfigurasi dilakukan berulang-ulang

dalam siklus hidup dari aplikasi sistem dimana desain konfigurasi ini akan dioperasikan ketika persyaratan tugas berubah (Turner et al. 2000)



Gambar 2.5 Variabel Desain Robot pada RMS

Sumber: Amtec-Robotics (2006)

Metodologi untuk desain konfigurasi bergantung pada kompleksitas dari sistem rekonfigurabel-nya. Dalam hal menggabungkan variabel desain secara alami untuk memenuhi persyaratan desain, suatu sistem rekonfigurabel bisa diklasifikasikan sebagai *uncoupled system*, *loosely-coupled system*, atau *strongly-coupled system*.

Pada *uncoupled system* seperti misalnya suatu sistem komputer, setiap modul berkorespondensi pada setiap permintaan individu. Sebagai contoh, monitor bertanggung jawab terhadap permintaan mengenai kualitas penampilan dan *keyboard* bertanggung jawab terhadap permintaan mengenai pengetikan input. Suatu konfigurasi ditentukan ketika semua dari komponen sistem tersebut dipilih.

Pada *loosely-coupled system* seperti misalnya pada *fixturing system*, setiap komponen berkorespondensi pada satu permintaan (komponen kelem bertanggung jawab untuk permintaan terkait kelem). Pada *strongly-coupled system* seperti

misalnya *reconfigurable robot system*, fungsi dari suatu modul tidak mempunyai hubungan langsung dengan persyaratan/pemintaan sistem. Semua modul-modul harus dipertimbangkan bersama-sama untuk mengevaluasi kapabilitas sistem untuk memenuhi permintaan/persyaratan.

2.1.3 Desain Kontrol

Untuk menjelaskan desain kontrol RMS, suatu *reconfigurable robot system* akan diambil sebagai contoh.

2.1.3.1 Variabel Rekonfigurabel dan Variabel Proses pada Desain Kontrol

Seperti terlihat pada Gambar 2.6, sistem memiliki dua kelas variabel untuk dimanipulasi : variabel rekonfigurabel dan variabel proses. Variabel rekonfigurabel bisa merubah konfigurasi robot. Variabel proses bisa merubah *the end-effector motion* ketika robot dioperasikan untuk menjalankan tugas. Variabel konfigurabel ditentukan dalam desain konfigurasi. Alasan kenapa desain konfigurasi juga dipandang sebagai bagian dari sistem kontrol adalah karena pada sistem rekonfigurabel, sistem seharusnya dikonfigurasi secara sering untuk memenuhi cepatnya perubahan, dan variabel rekonfigurabel bisa diganti selama proses kontrol.



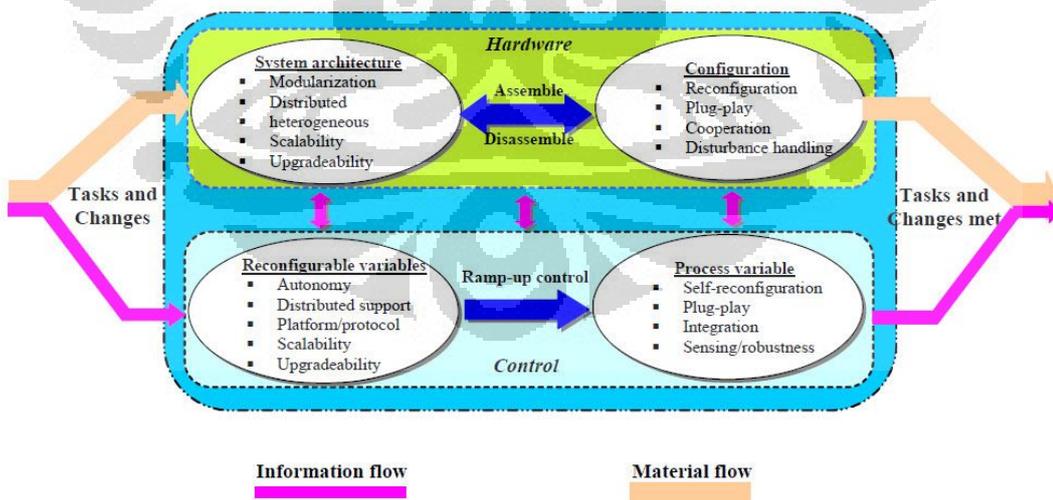
Gambar 2.6 Dua Kelas Variabel Rekonfigurabel

Sumber: Chen (2001) dan Amtec-Robotics (2006)

2.1.3.2 Persyaratan Desain Kontrol

RMS control bisa dilihat pada Gambar 2.7. Persyaratan desain dari suatu *RMS control* ditentukan oleh hal-hal berikut ini :

- Sistem kontrol seharusnya didistribusikan dan dimodularisasi, karena komponen sistem didesentralisasi dan secara geografis didistribusikan;
- Sistem kontrol seharusnya terbuka supaya sistem kontrol bisa meng-update komponen kontrol. Komponen kontrol mungkin dikembangkan oleh sistem operasi, bahasa, jaringan, *database*, protokol dan vendor yang berbeda-beda;
- Sistem kontrol seharusnya *scalable* dan *upgradeable* karena penambahan/pembuangan/pembaharuan komponen *hardware* dibutuhkan ketika fungsionalitas, kapabilitas atau teknologi telah diubah.
- Sistem kontrol seharusnya direkonfigurasi sendiri karena sistem konfigurasi bisa digerakkan dari satu konfigurasi ke konfigurasi lain secara sering dan korespondensi sistem kontrol seharusnya juga bisa direkonfigurasi sendiri secara cepat;
- Sistem kontrol seharusnya mampu untuk mengidentifikasi perubahan spesifikasi tugas. Perubahan ini bisa menyebabkan rekonfigurasi sistem.



Gambar 2.7 Persyaratan Desain pada Sistem Kontrol RMS

Sumber: Bi et al. (2007)

2.1.4 *Reconfigurable Manufacturing System (RMS) vs Dedicated Manufacturing Lines (DML) vs Flexible Manufacturing System (FMS)*

Setiap perusahaan manufaktur memiliki tiga tujuan utama: memproduksi pada biaya rendah, meningkatkan kualitas produk dan memiliki kapabilitas untuk merespons pasar dengan cepat. RMS dengan sistem rekonfigurabel-nya berfokus pada ketiga tujuan tersebut.

Sistem manufaktur saat ini didominasi oleh *Dedicated Manufacturing Lines* (DML) dan *Flexible Manufacturing System* (FMS). *Dedicated Manufacturing Lines* (DML) berbasis pada *fixed automation* dan memproduksi produk-produk atau komponen dengan volume besar. Setiap DML secara khas didesain untuk memproduksi komponen tunggal (contoh : *engine block*) pada level produksi yang tinggi. Ketika volume besar, biaya per komponen secara relatif menjadi rendah. Oleh karena itu, DML berbiaya-efektif hanya selama permintaan pasar cocok dengan penyediaan, tetapi dengan tekanan yang terus meningkat dari kompetisi global, berpotensi membuat kerugian. Tentu saja, memproduksi beragam produk tidak mungkin dilakukan dengan DML dan oleh karena itu, peranan DML dalam manufaktur modern menjadi hilang.

Flexible Manufacturing System (FMS) terdiri dari mesin CNC (*computer numerically controlled*) dan otomasi terprogram lain yang bisa memproduksi beragam produk di sistem yang sama (Koren, 1983). Meski memiliki keunggulan, survei yang dilakukan Koren pada tahun 2005 menunjukkan bahwa sistem fleksibel belum secara luas diadopsi oleh perusahaan dan banyak dari perusahaan manufaktur yang membeli FMS tidak puas dengan performa FMS tersebut.

Terlebih lagi, level produksi FMS lebih kecil dibandingkan dengan DML dan kapasitas produksi biasanya juga lebih kecil dibanding DML dan tidak mampu melakukan perubahan kapasitas secara cepat untuk merespons perubahan pasar. Perbandingan kedua sistem ini dapat dilihat pada Tabel 2.2 berikut ini :

Tabel 2.2 Perbandingan DML dengan FMS

DML	FMS
Limitations: <ul style="list-style-type: none"> • Not flexible – <i>for a single part</i> • Fixed capacity – <i>not scalable</i> 	Limitations: <ul style="list-style-type: none"> • Expensive • Slow – single-tool operation
Advantages: <ul style="list-style-type: none"> • Low cost • Fast – multi-tool operation 	Advantages: <ul style="list-style-type: none"> • Convertible • Scalable capacity

Sumber: Y. Koren (2006)

Suatu respons berbiaya-efektif terhadap perubahan pasar membutuhkan pendekatan manufaktur baru yang tidak hanya mengkombinasikan *high throughput* dari DML dengan fleksibilitas FMS, tetapi juga mampu bereaksi untuk berubah secara cepat dan efisien. Hal ini bisa diraih dengan mendesain sistem menurut dua prinsip:

- Mendesain sistem dan mesin-mesin dengan struktur yang bisa disesuaikan mengikuti permintaan pasar sehingga sistem/mesin mampu beradaptasi untuk menghasilkan produk-produk baru.
- Mendesain keluarga komponen produk dengan kustomisasi dan fleksibilitas yang dibutuhkan.

Suatu sistem yang didesain menurut prinsip tersebut melahirkan kelas baru dalam sistem manufaktur yang bernama *Reconfigurable Manufacturing System* (RMS). Perbandingan antara DML vs RMS vs FMS dapat dilihat pada Tabel 2.3.

Dari Tabel 2.3 dapat dirangkum bahwa membangun suatu sistem dengan *adjustable structure*, *scalability* dan *flexibility* yang fokus pada keluarga komponen (*part family*) menghasilkan suatu sistem rekonfigurabel yang responsif. Level produktifitas yang tinggi dan sistem biaya-efektif dihasilkan oleh (1) fokus pada *part family* dan (2) *customized flexibility* yang mengaktifkan operasi alat yang simultan. Fleksibilitas pada RMS yang berupa *customized flexibility* menyediakan semua fleksibilitas yang dibutuhkan dalam memproses *part family* sehingga lebih murah dibandingkan fleksibilitas pada FMS.

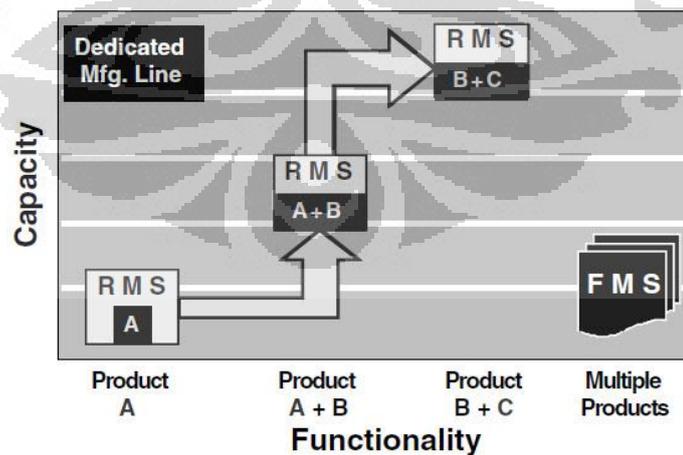
Tabel 2.3 Tabel Perbandingan DML, FMS dan RMS

	Dedicated	RMS/RMT	FMS/CNC
System Structure	Fixed	Adjustable	Adjustable
Machine Structure	Fixed	Adjustable	Fixed
System focus	Part	Part Family	Machine
Scalability	No	Yes	Yes
Flexibility	No	Customized	General
Simultaneously Operating Tool	Yes	Yes	No
Productivity	High	High	Low
Lifetime Cost	Low for a single part, when fully utilized	Medium or production at me lium-to-high volume new parts and vari able demand during system lifetime	Reasonable for simultaneous pro duction of many parts (at low volume); otherwise – High

Sumber: Y. Koren (2006)

2.1.4.1 Perbandingan Fungsi dan Kapasitas

Gambar 2.8 menunjukkan bahwa disaat DML dan FMS dibatasi oleh kapasitas & fungsionalitas, kapasitas dan fungsionalitas dari RMS mampu berubah seiring waktu sebagai reaksi sistem mengikuti perubahan kondisi pasar.

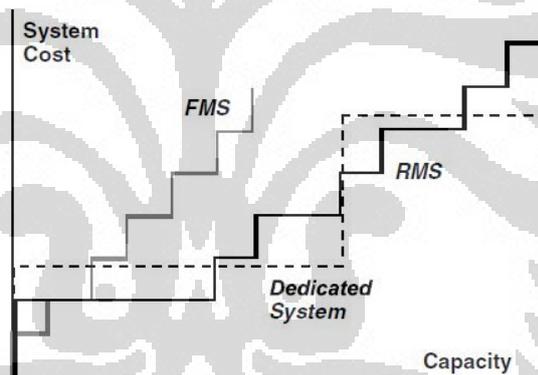
**Gambar 2.8** Hubungan Fungsi dan Kapasitas terhadap DML, FMS dan RMS

Sumber: Y. Koren (2006)

2.1.4.2 Perbandingan Sistem-Biaya dan Kapasitas

Pada Gambar 2.9 dapat dilihat bahwa DML memiliki biaya kapital yang konstan hingga mencapai kapasitas terpasang maksimum. Ketika kapasitas yang sudah maksimum masih perlu ditingkatkan, maka lini tambahan harus dibangun kembali dengan biaya yang mahal. Sementara itu, FMS *scalable* pada tingkat kapasitas konstan dengan kenaikan biaya yang relatif rendah untuk penambahan mesin secara paralel.

Berbeda dengan DML dan FMS, RMS *scalable* pada tingkat kapasitas yang tidak konstan bergantung pada desain awal RMS yang disesuaikan dengan perubahan kondisi pasar. Jika hanya kuantitas kecil yang dibutuhkan, maka tipe paralel pada FMS jelas lebih ekonomis. Akan tetapi, jika kuantitas lebih besar yang dibutuhkan dengan tingkat ketidakpastian pasar yang tinggi, maka RMS merupakan solusi yang paling ekonomis.



Gambar 2.9 Hubungan Kapasitas & Sistem Biaya pada DML, FMS & RMS

Sumber: Y. Koren (2006)

2.1.5 Teknologi pada RMS

Teknologi adalah aplikasi ilmiah atau penemuan untuk pengembangan dan peningkatan produk, jasa, dan proses operasi. Secara umum, teknologi dapat dibagi menjadi empat kategori, antara lain : teknologi produk, teknologi proses, teknologi informasi dan teknologi manufaktur. Sementara itu, automisasi adalah permesinan yang dapat merasakan dan mengontrol alat dan mengaktifkannya secara otomatis.

Tabel 2.4 Kategori Teknologi

Kategori Teknologi	Contoh-contoh Teknologi
Teknologi produk	PLM, GT, CAE
Teknologi proses	CAD/CAM, E-Procurement
Teknologi informasi	Barcode, RFID, & SCM
Teknologi manufaktur	CNC, Robot, AGV, ASRS, Process Control

Sumber: Stevenson (2009)

Berikut ini adalah definisi dan penjelasan dari beberapa contoh teknologi yang terdapat pada Tabel 2.4.

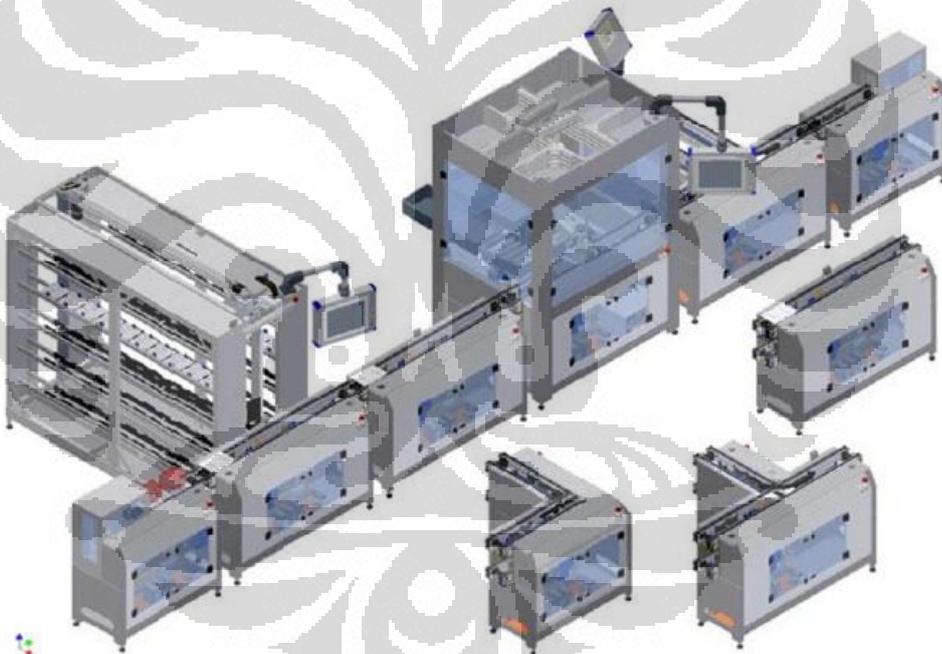
- CAD (*Computer Aided Design*) dan CAE (*Computer Aided Engineering*) adalah penggunaan komputer untuk secara interaktif mendesain produk dan mempersiapkan dokumentasi keteknikan, seperti: DFMA & 3D-*Object Modeling*.
- CNC (*Computer Numerical Controlled*) adalah mesin yang dikontrol oleh kode *software* untuk melakukan kinerja berbagai operasi dengan bantuan alat pengubah otomatis, juga mengumpulkan informasi selama pemrosesan data terkait kualitas.
- AGV (*Automatic Guided Vehicle*) adalah truk tanpa awak yang memindahkan material sepanjang jejak yang sudah dispesifikasi, diarahkan dengan kabel atau pita yang menempel di lantai atau dengan frekuensi radio sehingga sangat fleksibel.
- ASRS (*Automated Storage & Retrieval System*) adalah gudang yang dikontrol komputer yang menyediakan penempatan otomatis komponen ke atau dari tempat yang sudah di desain di dalam gudang.
- AIS (*Automatic Identification System*) adalah suatu sistem untuk mentransformasi data dalam bentuk elektronik, seperti barcode, RFID dan *transponders*.
- *Process control* adalah penggunaan IT untuk memonitor dan mengontrol proses fisik, seperti : sensor dan *measurement computer program*.
- *Vision system* berfungsi untuk mengkombinasikan kamera video dan teknologi komputer yang sering digunakan untuk melakukan inspeksi

- *Robot* adalah mesin fleksibel yang mampu mengendalikan, menggerakkan dan memegang suatu item secara otomatis.

Secara umum, terdapat lima teknologi yang melekat pada RMS, yaitu: *RMS modules*, *RMS Path Finder*, *RMS Intelligent Material-flow Cells*, *RMS Intelligent Production Cells*, dan *RMS Intelligent Network*.

2.1.5.1 RMS Modules

Sifatnya yang modular memungkinkan RMS untuk melakukan berbagai konfigurasi yang berbeda untuk setiap kegiatan produksi. Dan setiap modul dapat dioperasikan secara mandiri (terpisah) baik dari sisi *Up-Streams* maupun *Down-Streams*.

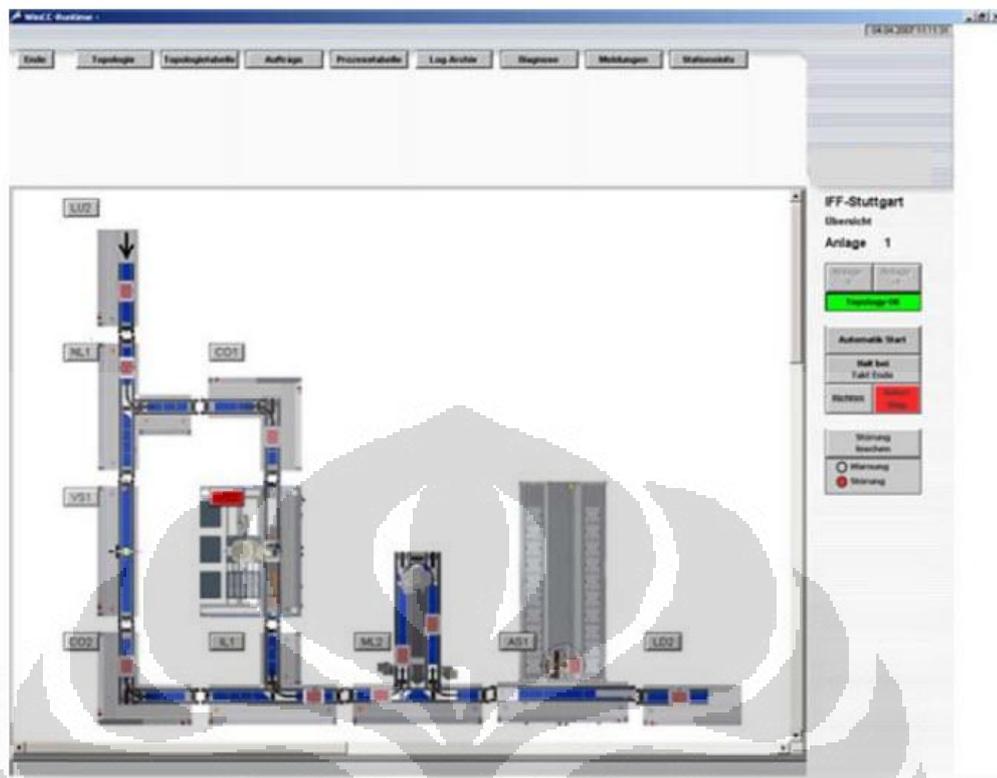


Gambar 2.10 RMS Modules

Sumber: Dokumen Laboratorium Sistem Manufaktur Departemen Teknik Industri UI

2.1.5.2 RMS Path Finder

Seluruh sistem produksi dilengkapi dengan sistem topologi umpan balik sehingga sistem kontrol –sebuah SQL server– dapat secara otomatis mengenali konfigurasi lini produksi yang dilakukan.

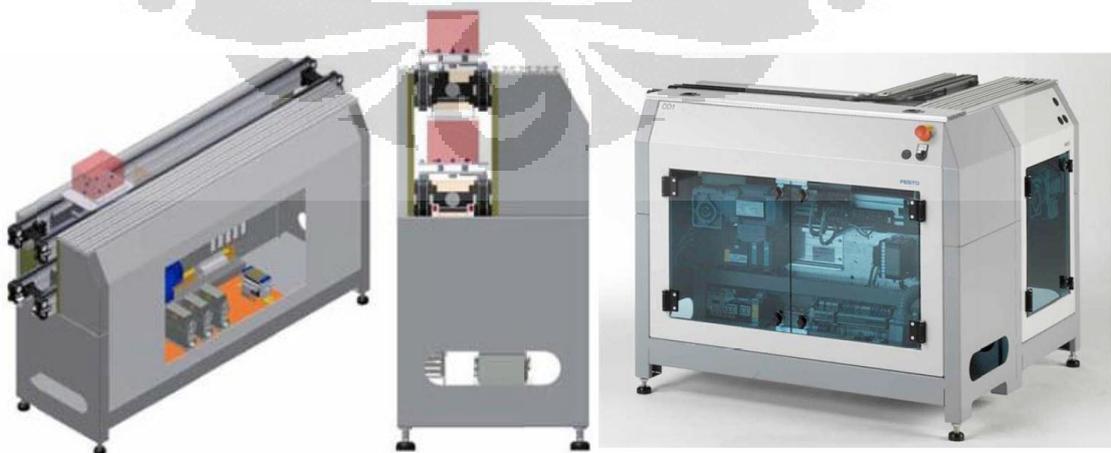


Gambar 2.11 RMS Path Finder

Sumber: Dokumen Laboratorium Sistem Manufaktur Departemen Teknik Industri UI

2.1.5.3 RMS *Intelligent Material-flow Cells*

Setiap sel aliran material pada RMS dilengkapi dengan konveyor, *controller*, dan seluruh *interfaces* yang diperlukan. Semuanya bekerja secara terintegrasi dan otomatis yang diatur pada pusat kontrol RMS.

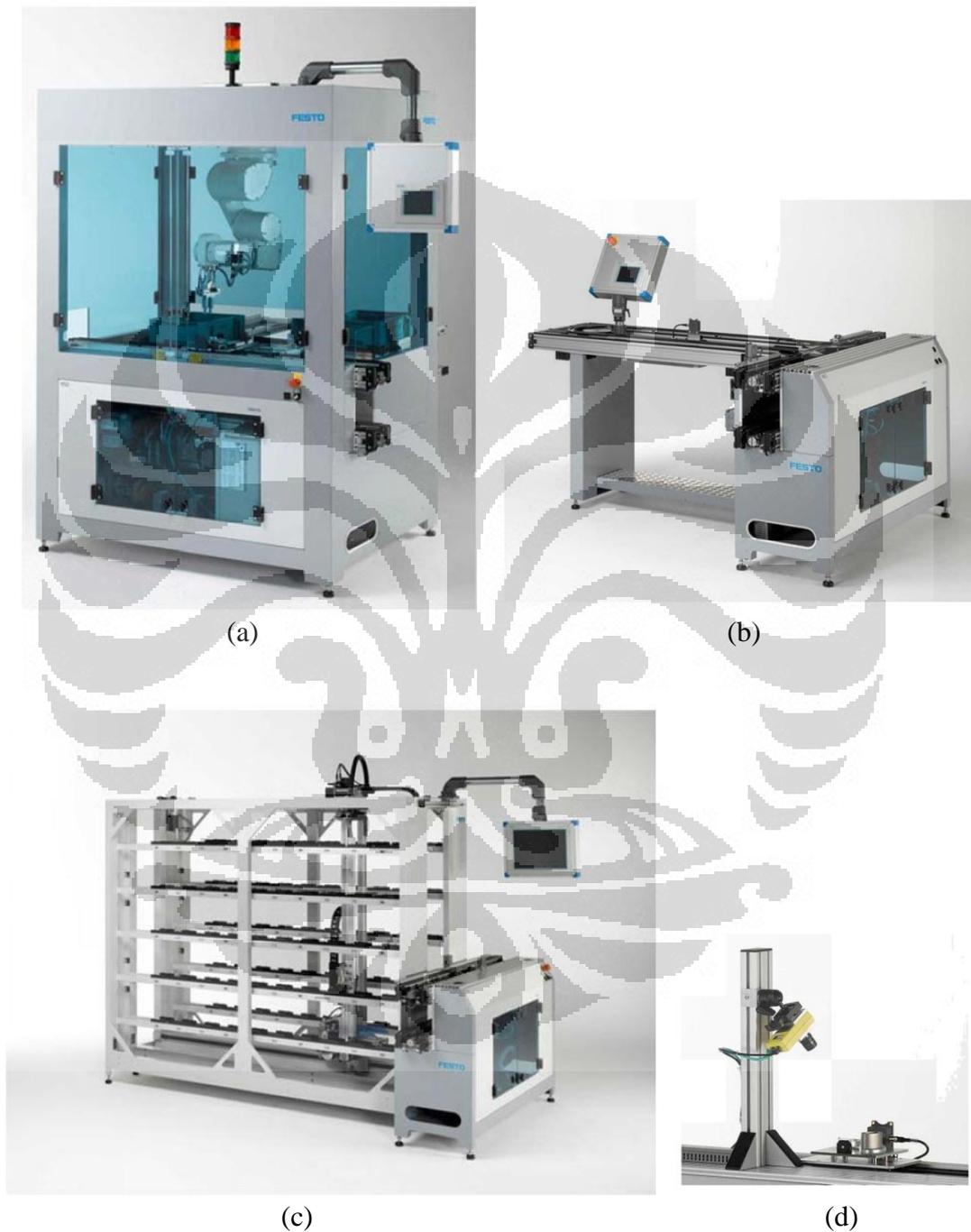


Gambar 2.12 RMS *Intelligent Material-flow Cells*

Sumber: Dokumen Laboratorium Sistem Manufaktur Departemen Teknik Industri UI

2.1.5.4 RMS *Intelligent Production Cells*

Terdiri dari *robot assembly*, *manual-work*, *camera inspection* dan *AS/RS*, *production cells* dapat memberikan berbagai pilihan fungsi-fungsi produksi.



Gambar 2.13 *RMS Intelligent Production Cells*: (a) *robot assembly*; (b) *manual work cell*; (c) *AS/RS cell*; (d) *camera inspection*

Sumber: Dokumen Laboratorium Sistem Manufaktur Departemen Teknik Industri UI

2.1.5.5 RMS *Intelligent Network*

RMS harus menggunakan konsep komunikasi yang unik dan cerdas sehingga masing-masing sel dapat berkomunikasi baik secara *up-streams* maupun *down-streams*.



Gambar 2.14 RMS *Intelligent Network*

Sumber: Dokumen Laboratorium Sistem Manufaktur Departemen Teknik Industri UI

Laboratorium Sistem Manufaktur Departemen Teknik Industri UI (LSM DTI-UI) juga sedang mengembangkan laboratorium yang menerapkan konsep RMS. Sifatnya yang modular memungkinkan RMS untuk melakukan berbagai konfigurasi yang berbeda untuk setiap kegiatan penelitian dan pembelajaran. Pada Gambar 2.15 dapat dilihat beberapa contoh konfigurasi RMS yang dapat dengan cepat dilakukan oleh LSM DTI-UI.



(a)



(b)

Gambar 2.15 RMS pada LSM DTI-UI: (a) RMS with Combined Production Line with Robot Assembly, Manual Workstation & Quality Inspection; (b) RMS with Combined Production Line with Robot Assembly, Manual Workstation, Quality Inspection & AS/RS cell

Sumber: Dokumen Laboratorium Sistem Manufaktur Departemen Teknik Industri UI

2.2 Sistem, Model dan Simulasi

2.2.1 Sistem

Blanchard (2000) mendefinisikan sistem sebagai sekumpulan dari elemen-elemen yang mempunyai fungsi bersama untuk mencapai suatu tujuan. Sedangkan Law (2004) mendefinisikan sistem sebagai sekelompok komponen yang beroperasi secara bersama-sama untuk mencapai tujuan tertentu atau sekumpulan entitas yang bertindak dan berinteraksi bersama-sama untuk memenuhi suatu tujuan akhir yang logis. Setidaknya di dalam sistem mempunyai empat ciri, yaitu :

- Adanya sekumpulan elemen
- Adanya interaksi di antara elemen tersebut
- Mempunyai tujuan yang hendak dicapai
- Situasi dan kondisi yang kompleks

Suatu sistem didefinisikan sebagai himpunan atau kombinasi dari bagian-bagian yang membentuk sebuah kesatuan yang kompleks. Namun tidak semua kumpulan dan gugus bagian dapat disebut suatu sistem kalau tidak memenuhi syarat adanya kesatuan (*unity*), hubungan fungsional, dan tujuan. Suatu kawasan dengan berbagai sumber daya dan aktivitas di dalamnya merupakan suatu sistem yang kompleks. Dalam penataan ruang suatu kawasan jelas ketiga syarat tersebut dapat dipenuhi, tata ruang yang berbasis lahan merupakan suatu kesatuan yang didalamnya terdapat hubungan fungsional antarsektor atau bagian dalam mencapai tujuan optimalisasi pemanfaatan tata ruang suatu kawasan. Setiap sistem membutuhkan diferensiasi untuk mengidentifikasi subparts dan integrasi untuk memastikan bahwa sistem tidak terurai menjadi unsur-unsur yang terpisah (Forum Kajian Kebijakan Spasial Kehutanan P4W).

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu :

a. Komponen

Komponen sistem atau elemen sistem dapat berupa elemen-elemen yang lebih kecil dari sistem utama yang disebut *sub sistem*, misalkan sistem komputer terdiri dari sub sistem perangkat keras, perangkat lunak dan manusia. Elemen-elemen yang lebih besar dari sistem utama yang disebut *supra sistem*. Misalkan bila perangkat keras adalah sistem yang memiliki

sub sistem CPU, perangkat I/O dan memori, maka supra sistem perangkat keras adalah sistem komputer.

b. Batas sistem

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batas suatu sistem menunjukkan ruang lingkup dari sistem tersebut.

c. Lingkungan luar sistem

Lingkungan luar dari sistem adalah apapun yang berada di luar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut, tergantung dengan kondisi yang ada. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan dengan demikian harus tetap selalu dijaga dan dipelihara. Sedang lingkungan luar yang merugikan harus ditahan dan dikendalikan, kalau tidak akan mengganggu kelangsungan hidup dan keseimbangan sistem.

d. Penghubung

Penghubung merupakan media perantara antar subsistem. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lainnya. Output dari satu subsistem akan menjadi input untuk subsistem yang lainnya dengan melalui penghubung. Dengan penghubung satu subsistem dapat berinteraksi dengan subsistem yang lainnya membentuk satu kesatuan.

e. Input

Input adalah energi yang dimasukkan ke dalam sistem. Input dapat berupa *maintenance input* dan *signal input*. *Maintenance input* adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan output.

f. Output

Output adalah hasil dari energi yang diolah dan diklasifikasi menjadi keluaran yang berguna dan sisa pembuangan. Output dapat merupakan input untuk subsistem yang lain atau kepada supra sistem.

g. Pengolah

Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah yang akan merubah input menjadi output. Suatu sistem produksi akan mengolah input berupa bahan baku dan bahan-bahan yang lain menjadi output berupa barang jadi.

h. Sasaran atau tujuan

Suatu sistem pasti mempunyai tujuan atau sasaran. Kalau sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. Sasaran dari sistem sangat menentukan sekali input yang dibutuhkan sistem dan output yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.

2.2.2 Model

Model adalah pola (contoh, acuan, ragam) dari sesuatu yang akan dibuat atau yang akan dihasilkan (Departemen P dan K, 1984). Definisi lain dari model adalah abstraksi dari sistem sebenarnya, dalam gambaran yang lebih sederhana serta mempunyai tingkat prosentase yang bersifat menyeluruh, atau model adalah abstraksi dari realitas dengan hanya memusatkan perhatian pada beberapa sifat dari kehidupan sebenarnya (Simamarta, 1983). Model memperlihatkan hubungan-hubungan langsung maupun tidak langsung serta kaitan timbal balik dalam istilah sebab akibat.

Sementara itu, pemodelan adalah proses menyusun model dalam suatu kerangka tujuan tertentu. Ada dua hal yang bisa dimodelkan, yaitu : objek nyata dan objek abstrak. Meskipun pembuatan model yang baik harus sedetail mungkin, pada model juga diberikan pembatasan ruang lingkup. Tidak semua detail sistem nyata perlu diperhatikan, melainkan hanya bagian penting saja yang terkait dengan permasalahan yang dihadapi. Adapun karakteristik model yang baik adalah :

- Tingkat generalisasi yang tinggi
- Mekanisme transparansi
- Potensial untuk dikembangkan
- Peka terhadap perubahan asumsi

Pemodelan dalam industri manufaktur memiliki beberapa kekhususan. Kekhususan tersebut adalah sebagai berikut :

- Waktu proses mempunyai variabilitas yang kecil begitu juga kedatangan entitasnya;
- Jalur proses biasanya tetap;
- Entitas kadang diproses dalam bentuk *batch*;
- Reliabilitas dari mesin dan *material handling* sangat berpengaruh;
- Biasanya bertujuan untuk melihat kondisi *steady state*, yaitu keadaan di mana sistem berjalan normal tanpa adanya perubahan yang signifikan.

Langkah pertama dalam mempelajari karakteristik suatu sistem adalah membangun sebuah model. Hal yang terpenting dalam membuat model adalah model yang dibangun harus benar-benar mampu merepresentasikan sistem yang akan dimodelkan. Ada beberapa tujuan dari penggunaan model :

- Memungkinkan seorang pengamat investigator untuk membuktikan tentang hipotesa suatu teori dan melakukan pengamatan tentang sistem dan mengambil kesimpulan logis akan dampak yang terjadi jika variabel-variabel yang mempengaruhi sistem berubah.
- Memberi kemudahan dalam melakukan perbaikan sistem
- Membantu membuat suatu kerangka kerja untuk melakukan tes terhadap kondisi yang diinginkan dari suatu sistem yang dimodifikasi
- Memudahkan melakukan kontrol terhadap sumber daya yang terlibat di dalam sistem.

2.2.2.1 Jenis-jenis model

Klasifikasi perbedaan dari model memberikan penambahan pendalaman pada tingkat kepentingannya, karena dapat dijelaskan dalam banyak cara. Model dapat dikategorikan menurut jenis, dimensi, fungsi, tujuan pokok pengkajian atau derajat keabstrakannya. Kategori umum adalah jenis model yang pada dasarnya dapat dikelompokkan menjadi (1) ikonik, (2) analog, dan (3) simbolik.

a. Model ikonik

Model ikonik adalah perwakilan fisik dari beberapa hal baik dalam bentuk ideal ataupun dalam skala yang berbeda. Model ikonik mempunyai

karakteristik yang sama dengan hal yang diwakili, dan terutama amat sesuai untuk menerangkan kejadian pada waktu yang spesifik. Model ikonik dapat berdimensi dua (foto, peta, cetak biru) atau tiga dimensi (prototip mesin, alat). Apabila model berdimensi lebih dari tiga dimensi maka tidak mungkin lagi dikonstruksikan secara fisik sehingga diperlukan kategori model simbolik.

b. Model analog (model diagramatik)

Model analog dapat mewakili situasi dinamik, yaitu keadaan berubah menurut waktu. Model ini lebih sering dipakai daripada model ikonik karena kemampuannya untuk mengetengahkan karakteristik dari kejadian yang dikaji. Model analog banyak bersesuaian dengan penjabaran hubungan kuantitatif antara sifat dan kelas yang berbeda. Dengan melalui transformasi sifat menjadi analognya, maka kemampuan membuat perubahan dapat ditingkatkan. Contoh model analog ini adalah kurva permintaan, kurva distribusi frekuensi pada statistik, dan diagram alir.

c. Model simbolik (model matematik)

Pada hakekatnya, ilmu sistem memusatkan perhatian kepada model simbolik sebagai perwakilan dari realitas yang sedang dikaji. Format model simbolik dapat berupa bentuk angka, simbol dan rumus. Jenis model simbolik yang umum dipakai adalah suatu persamaan. Bentuk persamaan adalah tepat, singkat, dan mudah dimengerti. Simbol persamaan tidak saja mudah dibaca, namun juga lebih cepat ditangkap maksudnya. Suatu persamaan adalah bahasa universal pada penelitian operasional dan ilmu sistem, dimana dipakai suatu logika simbolis.

Pemodelan mencakup suatu pemilihan dari karakteristik dari perwakilan abstrak yang paling tepat pada situasi yang terjadi. Pada umumnya, model matematis dapat diklasifikasikan menjadi dua bagian. Suatu model adalah bisa statik atau dinamik. Model statik memberikan informasi tentang peubah-peubah model hanya pada titik tunggal dari waktu. Model dinamik mampu menelusuri jalur waktu dari peubah-peubah model. Model dinamik lebih sulit dan mahal pembuatannya, namun memberikan kekuatan yang lebih tinggi pada analisis dunia nyata.

Pemilihan model tergantung pada tujuan dari pengkajian sistem dan terlihat jelas pada formulasi permasalahan pada tahap evaluasi kelayakan. Sifat model juga tergantung pada teknik pemodelan yang dipakai. Model yang mendasarkan pada teknik peluang dan memperhitungkan ketidak-tentuan (*uncertainty*) disebut model probabilistik atau model stokastik. Dalam mengkaji suatu sistem, model ini sering dipakai karena perihal yang dikaji umumnya mengandung keputusan yang tidak tentu.

Kebalikan dari model ini adalah model kuantitatif yang tidak mempertimbangkan peluang kejadian, dikenal sebagai model deterministik. Contohnya adalah model pada program linear dan PERT. Model ini memusatkan penelaahannya pada faktor-faktor kritis yang diasumsikan mempunyai nilai eksak dan tertentu pada waktu yang spesifik. Model probabilistik biasanya mengkaji ulang data atau informasi terdahulu untuk menduga peluang kejadian tersebut pada keadaan sekarang atau yang akan datang dengan asumsi terdapat relevansi pada jalur waktu.

Pada beberapa perihal, sebuah model dibuat hanya untuk semacam deskripsi matematis dari kondisi dunia nyata. Model ini disebut model deskriptif dan banyak dipakai untuk mempermudah penelaahan suatu permasalahan. Model ini dapat diselesaikan secara eksak serta mampu mengevaluasi hasilnya dari berbagai pilihan data input. Apabila perbandingan antar alternatif dilakukan, maka model disebut model optimalisasi. Solusi dari model optimasi adalah merupakan nilai optimum yang tergantung pada nilai input, contohnya adalah non-linear programming. Bilamana sistem telah diekspresikan pada notasi matematik dan format persamaan, timbullah keuntungan dari fasilitas manipulatif dari matematik.

Seorang analis dapat memasukkan nilai-nilai yang berbeda dalam model matematik dan kemudian mempelajari perilaku dari sistem tersebut. Pada pengkajian tertentu, sensitivitas dari sistem dilakukan dengan perubahan dari input sistem itu sendiri. Bahasa simbolik ini juga membantu dalam komunikasi karena pernyataan yang singkat dan jelas daripada deskripsi lisan.

2.2.3 Simulasi

Oxford American Dictionary mendefinisikan simulasi sebagai suatu cara mereproduksi kondisi-kondisi dari suatu situasi, menggunakan model untuk melakukan studi, menguji, melakukan uji coba, pelatihan, dan lain-lain.

Simulasi adalah suatu cara untuk menduplikasi/menggambarkan ciri, tampilan, dan karakteristik dari suatu sistem nyata. Ide awal dari simulasi adalah untuk meniru situasi dunia nyata secara matematis, kemudian mempelajari sifat dan karakter operasionalnya, dan akhirnya membuat kesimpulan dan membuat keputusan berdasar hasil dari simulasi. Dengan cara ini, sistem di dunia nyata tidak disentuh/dirubah sampai keuntungan dan kerugian dari apa yang menjadi kebijakan utama suatu keputusan di ujicobakan dalam sistem model.

Simulasi merupakan suatu metode pembelajaran yang bisa digunakan atas banyak alasan, beberapa alasan yang memungkinkan penggunaan simulasi adalah:

- a. Sistem nyata yang ingin kita pelajari tidaklah ada dan itu juga sangat mahal, menghabiskan waktu, berbahaya atau hampir tidak mungkin dengan sistem yang asli. Contohnya, reaktor nuklir (berbahaya), sistem tata surya (tidak memungkinkan). Aspek biaya sering memainkan peran yang sangat besar juga. Simulasi juga menawarkan keuntungan-keuntungan dalam menciptakan dan menarik ketika eksperimen tersebut dianggap tidak mungkin atau terlalu mahal untuk dipraktekkan.
- b. Sistem nyata yang ingin kita pelajari benar-benar ada, tetapi melakukan eksperimen dengan sistem nyata itu terlalu mahal, terlalu berbahaya atau mengganggu sistem.
- c. Informasi yang benar harus dijelaskan situasi-situasi masalahnya yang tidak selalu ada. Biasanya itu tidak mungkin untuk mengukur waktu menunggu, waktu pelayanan, *setting*, dan perubahan waktu dan juga memperbaiki waktu praktek.
- d. Kinerja sistem sekarang, nanti ataupun yang dulu harusnya tidak dipelajari *real time* tetapi harus mempercepat atau memperlambat.
- e. Tidak mungkin untuk menjelaskan dan menganalisa sistem yang dipelajari secara matematika.

- f. Tidaklah sederhana solusi analitik dan numerik terhadap model matematika dari suatu sistem. Untuk masalah antrian, nilai optimum dari *server* bisa dihitung, jika model antriannya lebih rumit tidak mungkin menyediakan model matematik terhadap suatu masalah.

Dalam semua kasus yang mencakup masalah kompleks seperti pada bisnis manufaktur dan pemerintahan simulasi merupakan metode pemecahan masalah yang logis. Ada dua tujuan utama simulasi, yaitu :

- a. Eksplorasi dan memahami alternatif

Pada tujuan ini terdapat tiga aktivitas yang bisa dilakukan, antara lain : *Scenario Testing*, *Sensitivity analysis*, dan *Simulation games for enhanced learning*. *Scenario Testing* adalah aktivitas yang dilakukan untuk mengetahui implikasi apa yang muncul dari alternatif-alternatif model yang disimulasikan. *Sensitivity analysis* adalah aktivitas yang dilakukan untuk mengetahui faktor apa yang memiliki daya ungkit terbesar terhadap model yang disimulasikan. *Simulation games for enhanced learning* adalah aktivitas yang dilakukan untuk pembelajaran model simulasi pada lingkungan di dunia nyata yang penuh dengan tantangan, sehingga dengan simulasi, bisa dilakukan dengan menyenangkan dan bebas resiko.

- b. Menemukan alternatif terbaik.

Pada tujuan ini terdapat empat tipe contoh aktivitas: (1) *Distribution schedule*, seperti *Travelling Salesman Problem (TSP)*; (2) *Best warehousing strategy*, seperti *Bin packing sequence*, *order batching*, *optimum layout*; (3) *Revenue managemen* (4) *Production & operations schedule*.

2.2.3.1 Tipe Simulasi

Ada beberapa tipe simulasi, diantaranya :

- a. *Static vs dynamic simulation*

Static simulation merupakan salah satu simulasi yang tidak berdasarkan waktu, seringkali melibatkan pengambilan contoh secara acak untuk menghasilkan keluaran statistik (biasa disebut Simulasi Monte Carlo). Simulasi ini biasa digunakan dalam menghitung nilai portofolio. *Dynamic*

simulation, kebalikan dari *static simulation* yang tidak berdasarkan waktu, simulasi ini berdasarkan waktu. Simulasi tipe ini cocok digunakan dalam sistem manufaktur dan jasa.

b. *Stochastic vs deterministic simulation*

Stochastic atau *probabilistic simulation* adalah simulasi yang salah satu atau lebih dari satu variabelnya diacak secara alami. Simulasi ini menghasilkan keluaran yang teracak dan oleh karena itu keluaran ini hanya memberikan data mengenai kemungkinan perilaku dari sistem. Sedangkan simulasi yang tidak memiliki komponen input yang teracak disebut *deterministic simulation*.

c. *Discrete-event vs continuous simulation*

Sebuah simulasi diskrit adalah simulasi dimana perubahan keadaan terjadi pada suatu titik waktu tertentu yang dipicu oleh sebuah kejadian. Misalkan perubahan terjadi karena adanya kedatangan barang, atau perubahan terjadi karena adanya penambahan jumlah mesin dan pekerja yang melakukan proses. Sedangkan pada *continuous simulation* keadaan berubah secara terus menerus sejalan dengan perubahan waktu. Misalnya suhu yang terus turun karena hari sudah malam dan suhu naik karena siang hari.

d. Simulasi *flow based oriented* vs Simulasi berorientasi objek

Simulasi *flow based oriented* adalah simulasi dimana langkah-langkahnya berjalan dari satu demi satu proses mengikuti aliran proses yang telah ditentukan. Dan keseluruhan objek yang ada didalamnya cenderung pasif. Artinya simulasi *flow based* ini tidak memberikan komunikasi antara satu objek dengan objek lainnya. Simulasi berbasis *flow based* ini sangat cocok digunakan pada simulasi produksi massal (produksi dengan jumlah besar) dan hanya berfokus pada satu macam objek (produknya hanya satu jenis) dimana objek tersebut hanya memerlukan satu aliran proses. Sebuah simulasi berorientasi objek (*Object Oriented Simulation/OOS*) terdiri dari satu set objek yang berinteraksi satu sama lain dari waktu ke waktu. Simulasi berorientasi objek memiliki daya tarik intuitif yang besar dalam aplikasi karena sangat mudah untuk melihat dunia nyata terdiri dari

sebagai objek. Dalam manufaktur, benda atau material, mesin, para pekerja, konveyor, dan lain-lain. Juga, rute bagian, jadwal, serta rencana kerja dapat dilihat sebagai objek. Simulasi berorientasi objek memungkinkan pemetaan satu-ke-satu antara objek dalam sistem manufaktur yang dimodelkan dan abstraksi mereka dalam model simulasi (Narayanan et al., 1998). Simulasi berorientasi objek dianggap sebagai paradigma pemetaan yang lebih alami untuk aplikasi simulasi, simulasi berorientasi objek cocok untuk aplikasi dan kondisi sistem yang kompleks sebelum orang dapat menyesuaikan kebijakan atau peraturan. Pendekatan Simulasi berorientasi objek dikaitkan dengan kelompok riset di Institut Teknologi Georgia, Amerika Serikat, lihat misalnya Narayanan et al., 2000. Pemodelan berbasis objek pada dasarnya dibangun terpisah pada empat abstraksi perangkat lunak yang mendasar, yaitu bahan atau material, lokasi, *controller* dan rencana proses. Lokasi bisa memproses material atau bahan. Rencana proses menentukan operasi yang diperlukan untuk operasional, upaya penurunan bagian-bagian tertentu. *Controllers* melakukan keputusan pembuatan dan kontrol dalam simulasi. *Controller* adalah *entitas-event* yang merespon perubahan dalam domain mereka. Semua pendekatan kegiatan terpisah tersebut disadari untuk memberikan kebutuhan penilaian yang terkait dengan pengambilan keputusan dari kegiatan lain yang terkait dengan transformasi fisik atau pengolahan data.

2.2.4 Langkah-langkah Pembuatan Model Simulasi

Dalam membuat sebuah model simulasi, ada beberapa langkah yang biasa digunakan sebagai acuan, yaitu sebagai berikut :

- a. Menentukan tujuan, ruang lingkup, dan prasyarat yang dibutuhkan

Batasan sebuah model akan menentukan seluas apa model yang akan dibuat. Semakin luas akan semakin baik, karena akan mempertimbangkan faktor yang berkaitan, namun akan semakin sulit dalam penyusunan modelnya. Tingkat kerumitan berkaitan dengan batasan model yang akan dibuat. Salah satu alat penolong batasan model adalah dengan merencanakan hasil yang akan dicapai

- b. Mengumpulkan dan menganalisa data yang diperlukan

Pada langkah ini, aktivitas yang perlu dilakukan, antara lain: menentukan seluruh entitas yang mungkin, mendeskripsikan proses-proses terhadap entitas dan ,endeskripsikan detail data yang mungkin berhubungan dengan poin di atas.

- c. Membuat model

Memulai dengan membuat model konseptual, terbebas dari angka-angka dan koefisien lainnya, misalnya dengan membuat sebuah *causal loop*. Lalu, menambahkan elemen yang terdapat dalam struktur seperti entitas, lokasi, dll. Kemudian, menambahkan elemen prosedural seperti operasi, kedatangan, dll. Terakhir, menambahkan permasalahan lainnya jika ada.

- d. Memverifikasi dan memvalidasi model sehingga layak secara ilmiah

- e. Melakukan eksperimen

Dalam tahap ini, model dijalankan.

- f. Mempresentasikan hasil

Hasil yang didapat dibuat analisa dan kesimpulannya.

2.2.5 Keuntungan Penggunaan Model Simulasi

Ada beberapa keuntungan dengan menggunakan model simulasi komputer, antara lain :

- a. Membangun pengembangan sebuah pemahaman dari tingkah laku yang ada pada sistem
- b. Dapat menggambarkan hubungan ketergantungan dan varian yang kompleks
- c. Menghemat biaya
- d. Dapat digunakan untuk berbagai tipe sistem
- e. Tidak memakan waktu lama
- f. Tidak perlu membangun sistem baru, mengganggu atau merusak sistem yang telah ada
- g. Dapat melakukan pengontrolan pada eksperimen
- h. Mudah dimengerti dan digunakan
- i. Menunjukkan proses yang secara visual

2.3 Plant Simulation Software

Pada tahun 1986, Fraunhofer Society for Factory Operation and Automation mengembangkan sebuah hirarki simulasi program berorientasi objek untuk Apple Macintosh yang diberi nama SIMPLE Mac for Apple Macintosh. Tahun 1990, didirikan AIS (*Angewandte Informations Systeme*) yang mengembangkan SIMPLE++ (*Simulation in Produktion Logistik and Engineering*). Di tahun 1991, AIS berganti nama menjadi AESOP (*Angewandte EDV-Systeme zur optimierten Planung*). Tecnomatix Ltd mengakuisisi AESOP pada tahun 1997 dan mengganti nama 2000 SIMPLE++ menjadi eM-Plant. Di tahun 2004, Tecnomatix Ltd diakuisisi oleh UGS Corporation, eM-Plant berganti nama menjadi Tecnomatix Plant Simulation. Pada tahun 2007, Siemens AG mengakuisisi UGS Corporation dan terus melakukan pengembangan terhadap Tecnomatix Plant Simulation.

Tecnomatix Plant Simulation atau yang lebih dikenal dengan nama Plant Simulation merupakan salah satu perangkat lunak yang berbasis windows yang mampu mensimulasikan suatu sistem produksi. Plant Simulation digunakan untuk mengoptimalkan *throughput*, mengurangi kemacetan, dan mengurangi kegiatan-kegiatan dalam proses. Plant Simulation juga mempertimbangkan faktor internal dan eksternal rantai pasokan, sumber-sumber produksi dan bisnis proses sehingga memungkinkan penggunaanya untuk menganalisis dampak variasi yang berbeda.

Salah satu daya tarik dari perangkat lunak Plant Simulation ini mampu memberikan kemudahan penggunaannya, serta animasi yang dibuat mendekati kenyataan dalam bentuk dua dan tiga dimensi sehingga mampu memberikan presentasi yang baik. Dengan Plant Simulation kita dapat membuat simulasi dan optimasi sistem produksi dan proses.

Plant Simulation memberikan kemudahan bagi insinyur dan manajer untuk melakukan serangkaian percobaan yang bertujuan melakukan perbaikan terhadap sistem yang ada. Kemampuan mendasar yang harus dimiliki sebagai dasar yang digunakan melakukan simulasi dengan Plant Simulation adalah kemampuan

analisis yang cukup baik, pengetahuan statistik, keahlian teknik, dan kemampuan komunikasi yang baik.

Adapun keunggulan Plant Simulation, antara lain :

- a. Mendeteksi dan menghilangkan masalah yang hanya sedikit membutuhkan koreksi biaya dan memakan waktu langkah-langkah selama produksi.
- b. Meminimalkan biaya investasi lini produksi tanpa membahayakan output yang diperlukan.
- c. Mengoptimalkan kinerja sistem produksi yang ada dengan mengambil langkah-langkah yang telah diverifikasi dalam lingkungan simulasi sebelum diimplementasikan.
- d. Dapat disajikan dalam bentuk Visualisasi 3D.

2.4 UNIFIED MODELING LANGUAGE (UML)

Unified Modeling Language (UML) adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang, dan mendokumentasikan sistem peranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. UML mendefinisikan diagram-diagram berikut :

a. *Use case diagram*

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya *login* ke sistem, membuat sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. *Use case diagram* dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang test case untuk semua *feature* yang ada pada sistem. Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use*

case yang meng-*include* dieksekusi secara normal. Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng-*extend use case* lain dengan perilakunya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

b. *Class diagram*

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package*, dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class* memiliki tiga area pokok : nama, atribut dan metode. Atribut dan metode dapat memiliki salah satu sifat berikut :

- *Private*, tidak dapat dipanggil dari luar class yang bersangkutan
- *Protected*, hanya dapat dipanggil oleh class yang bersangkutan dan anak-anak yang mewarisinya
- *Public*, dapat dipanggil oleh siapa saja

c. *Behaviour diagram*

- *Statechart diagram*

Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari stimuli yang diterima. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*).

- *Activity diagram*

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang bagaimana masing-masing alir berawal, keputusan yang mungkin terjadi, dan bagaimana mereka

berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu, *activity diagram* tidak menggambarkan perilaku internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

d. *Interaction diagram*

- *Sequence diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri dari dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis panah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metode dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*.

- *Collaboration diagram*

Collaboration diagram juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*. Setiap *message* memiliki *sequence number*, di mana *message* dari level tertinggi memiliki nomor 1. *Message* dari level yang sama memiliki prefiks yang sama.

e. *Component diagram*

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungannya (*dependency*) di antaranya. Komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link time*, maupun *run time*. Umumnya komponen terbentuk dari beberapa *class* dan/atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

f. *Deployment diagram*

Deployment/physical diagram menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, *server* atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi *server*, dan hal-hal lain yang bersifat fisik. Sebuah *node* adalah *server*, *workstation*, atau piranti keras lain yang digunakan untuk men-*deploy* komponen dalam lingkungan sebenarnya. Hubungan antar *node* (misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini.

Adapun langkah-langkah penggunaan UML adalah sebagai berikut:

- a. Buatlah daftar *business process* dari level tertinggi untuk mendefinisikan aktivitas dan proses yang mungkin muncul
- b. Petakan *use case* untuk tiap *business process* untuk mendefinisikan dengan tepat fungsionalitas yang harus disediakan oleh sistem. Kemudian

- perhalus *use case* dan lengkapi dengan *requirement*, *constraints*, dan catatan lain.
- c. Buatlah *deployment diagram* secara kasar untuk mendefinisikan arsitektur fisik sistem.
 - d. Definisikan *requirement* lain (non-fungsional, *security* dan sebagainya) yang juga harus disediakan oleh sistem.
 - e. Berdasarkan *use case diagram*, mulailah membuat *activity diagram*.
 - f. Definisikan objek-objek level atas (*package* atau *domain*) dan buatlah *sequence* dan/atau *collaboration diagram* untuk tiap alir pekerjaan. Jika sebuah *use case* memiliki kemungkinan alir normal dan *error*, buatlah satu diagram untuk masing-masing alir.
 - g. Buatlah rancangan *user interface model* yang menyediakan antarmuka bagi pengguna untuk menjalankan skenario *use case*.
 - h. Berdasarkan model-model yang sudah ada, buatlah *class diagram*. Setiap *package* atau *domain* dipecah menjadi hirarki *class* lengkap dengan atribut dan metodenya. Akan lebih baik jika untuk setiap *class* dibuat *unit test* untuk menguji fungsionalitas *class* dan interaksi dengan *class* lain.
 - i. Setelah *class diagram* dibuat, kita dapat melihat kemungkinan pengelompokkan *class* menjadi komponen-komponen. Karena itu buatlah *component diagram* pada tahap ini. Juga, definisikan tes integrasi untuk setiap komponen ke dalam *node*.
 - j. Perhalus *deployment diagram* yang sudah dibuat. Detilkan kemampuan dan *requirement* piranti lunak, sistem operasi, jaringan, dan sebagainya. Petakan komponen ke dalam *node*.
 - k. Mulailah membangun sistem. Ada dua pendekatan yang dapat digunakan :
 - Pendekatan *use case*, dengan meng-*assign* setiap *use case* kepada tim pengembang tertentu untuk mengembangkan *unit code* yang lengkap dengan tes.
 - Pendekatan komponen, yaitu meng-*assign* setiap komponen kepada tim pengembang tertentu.
 - l. Lakukan uji modul dan uji integrasi serta perbaiki model beserta kodenya.
 - m. Piranti lunak siap dirilis.

2.5 Verifikasi dan Validasi Model Simulasi

Verifikasi dan validasi adalah tahap dalam pemodelan untuk memeriksa diterima atau tidaknya suatu model sebelum model diterapkan. Verifikasi adalah memeriksa sintesa sistem dengan logika dan/atau analitik secara teoritik. Validasi adalah merupakan tahap terakhir dalam pengembangan model untuk memeriksa model dengan meninjau apakah keluaran model sesuai dengan sistem nyata, dengan melihat konsistensi internal dan representasi.

2.5.1 Verifikasi model diskrit

Verifikasi dilakukan untuk menghindari terjadinya kesalahan logik yang mungkin timbul. Verifikasi model komputer dapat dilakukan dengan memeriksa dan menganalisa proses *debugging*. Verifikasi memberikan kesempatan untuk mengevaluasi beberapa sumber kesalahan, antara lain :

- Model mungkin saja mencangkup variabel-variabel yang kurang penting;
- Model mungkin saja tidak melibatkan suatu variabel yang signifikan;
- Model bisa saja gagal menunjukkan ketelitian hubungan antara variabel-variabel dan ukuran ketidakefektifan yang digunakan;
- Nilai-nilai numerik yang digunakan bisa saja tidak benar dan karenanya dapat mengurangi kemampuan model dalam mencerminkan perilaku sistem.

Adapun proses verifikasi yang harus dilakukan antara lain :

- Pelajari kembali bahasa pemrogramannya
Tujuannya adalah untuk memeriksa kesalahan dan ketidakkonsisten suatu perilaku.
- Memeriksa output / keluaran yang logis
Bisa dengan membuat data input dummy yang outputnya telah diverifikasi secara matematis (excel, matlab dsb).
- Mengamati animasi terhadap tingkah laku yang benar
- Menggunakan fasilitas '*trace and debug*' yang disediakan pada software

Kendala yang umum muncul dalam pengujian dan pembuktian kebenaran model adalah ketidakcukupan atau ketidakterediaan data. Kendala ini bisa diatasi dengan cara, antara lain :

- Dilakukan '*running*' pada model yang telah dikomputerisasi untuk periode waktu yang pendek sehingga hasilnya dapat dibandingkan dengan kalkulator tangan
- Dilakukan '*running*' pada masing-masing bagian dari model yang kompleks sehingga hasilnya dapat diverifikasi
- Mengganti distribusi kemungkinan yang kompleks dengan distribusi yang lebih mudah sehingga hasilnya lebih mudah diverifikasi
- Jika memungkinkan dapat dibuat situasi percobaan yang mudah dan dicoba beberapa kombinasi dari keadaan-keadaan dalam model

2.5.2 Validasi model diskrit

Ada tiga jenis validasi model diskrit, yaitu :

- Validasi penampakan (*face validity*)
Validasi yang bertujuan untuk Meninjau kebenaran diagram alir model atau mekanisme logika model.
- Validasi penelusuran (*trace validity*)
Validasi ini bertujuan untuk Menelusuri kebenaran logika model dan program komputer secara manual (*debugging*).
- Validasi rasional (*rational validity*)
Validasi yang bertujuan untuk menguji kebenaran asumsi-asumsi yang digunakan pada struktur model.

BAB 3

PENGUMPULAN DAN PENGOLAHAN DATA

3.1 Pengumpulan Data

Untuk menangani perubahan cepat dalam permintaan pasar manufaktur, *Reconfigurable Manufacturing System (RMS)* dengan fitur rekonfigurabilitasnya harus dikembangkan. Suatu model yang mendeskripsikan proses rekonfigurasi suatu sistem manufaktur telah dikembangkan oleh Xiuli Meng (2010) dengan cara mengaplikasikan *colored timed object-oriented petri nets*. Berdasarkan perbedaan utama pada konfigurasi RMS dengan FMS, suatu struktur hierarki modular dari RMS juga turut dikembangkan pada model tersebut. Dengan metode *object-oriented*, semua kelas objek di dalam model RMS diidentifikasi.

Macro-place digunakan untuk memodelkan jumlah total dari banyak proses yang terjadi didalam model dan *macro-transition* digunakan untuk menghubungkan semua hubungan antar *macro-place* tersebut. *Macro-place* dan *macro-transition* dihubungkan dengan panah untuk membentuk suatu *petri net* yang dinamakan sebagai suatu *macro-level petri net* sehingga kontrol logika dari RMS bisa ditampilkan. *Macro-level petri net* dibuat dengan langkah-langkah hierarkis dimana setiap langkah mendeskripsikan *macro-place* yang tersusun dari sub *macro-place* yang lebih detail.

Lalu, karakteristik dari aliran material dan batasan waktu pada RMS dimodelkan dengan cara mengaplikasikan tanda warna (*colored token*) dan dibantu dengan atribut *time-delay*. Model ini mengintegrasikan metode *object-oriented*, *stepwise refinement idea* dan *petri nets* secara bersama-sama.

Petri net telah secara luas digunakan dalam memodelkan aliran proses pada sistem manufaktur (Zhang & Jiao, 2009). Salah satu alasannya adalah bahwa tempat dan transisi dari suatu *petri net* bisa digunakan untuk mendeskripsikan kejadian dan aktivitas yang terjadi didalam suatu sistem manufaktur. Sebelumnya, pendekatan berbeda untuk memodelkan FMS dengan *petri net* sudah pernah dilakukan. Zhou et al (1993) telah menggunakan suatu *petri net* untuk memodelkan dan menganalisis sel FMS. Cheng et al. (1994) mengajukan suatu

timed place petri net (TPPN) untuk memodelkan dan menjadwalkan FMS. Yan et al. (1998) membuat suatu *extended stochastic high-level evaluation petri net* (ESHLEP-N) yang lebih cocok dalam pemodelan dan simulasi suatu FMS. Basile et al (2004) menggabungkan metode *trace-based formal specification* dan pendekatan *compositional petri net* dengan *predefined building block* dalam suatu *unified framework* yang mempunyai tujuan untuk menangani kompleksitas perilaku dan batasan FMS. Wu (2005) memperkenalkan *high-level object petri net* (HLOPN) untuk memodelkan *job-shop* FMS secara umum.

Suatu RMS disusun oleh mesin-mesin rekonfigurabel dan sistem kontrol rekonfigurabel dengan desain arsitektur terbuka (*open-architecture*) untuk memproduksi bermacam komponen produk dengan hubungan keluarga. Produk yang tergabung dalam keluarga yang sama akan diproduksi oleh RMS dalam suatu konfigurasi yang saling berkorespondensi dengan konfigurasi dari keluarga produk yang lain. Xiuli Meng menggunakan *petri net* sebagai alat pemodelan RMS karena memiliki karakteristik representasi grafis yang baik.

Akan tetapi, suatu model yang dibentuk dari *petri net* tidak memiliki properti yang dibutuhkan dalam pemodelan RMS seperti, modularitas dan *reusability* (Xiuli Meng, 2010). Oleh karena itu, Meng mengajukan pendekatan *object-oriented petri net* dalam pembuatan model RMS. Wang dan Wu (1996, 1998) mengembangkan paradigma *object-oriented* (O-O) *petri net* yang tidak hanya memiliki karakteristik dari *petri net* dan teknik O-O, tetapi juga mencakup pengetahuan mengenai kontrol dan pengambilan keputusan. Felix dan Zhang (2001) memperkenalkan pendekatan *object & knowledge-based interval timed petri net* (OKITPN) untuk memodelkan *agile manufacturing system* yang mencakup pengetahuan, waktu interval, modular dan atribut komunikasi. Akan tetapi, semua penelitian terdahulu tersebut hanya dalam sel sistem manufaktur saja, bukan sistem manufaktur secara keseluruhan.

Oleh karena itu, Xiuli Meng (2010) mencoba menggunakan metodologi *stepwise refinement* untuk mempertimbangkan pembuatan model RMS dari cara pandang sistem secara keseluruhan. Metodologi ini sangat presisi dan natural untuk memodelkan suatu RMS dengan pendekatan *object-oriented petri net*

sehingga model RMS yang dibuat memiliki karakteristik fleksibilitas, modularitas dan *reusability*.

3.1.1 Pembuatan Model RMS *Petri Net* (Xiuli Meng, 2010)

Seperti yang kita ketahui, konfigurasi dan rekonfigurasi dari RMS lebih rumit karena RMS mengizinkan fleksibilitas tidak hanya dalam memproduksi beragam komponen produk tetapi juga dalam hal merubah sistem RMS itu sendiri. Hal inilah yang merupakan salah satu perbedaan RMS dengan FMS dan DML.

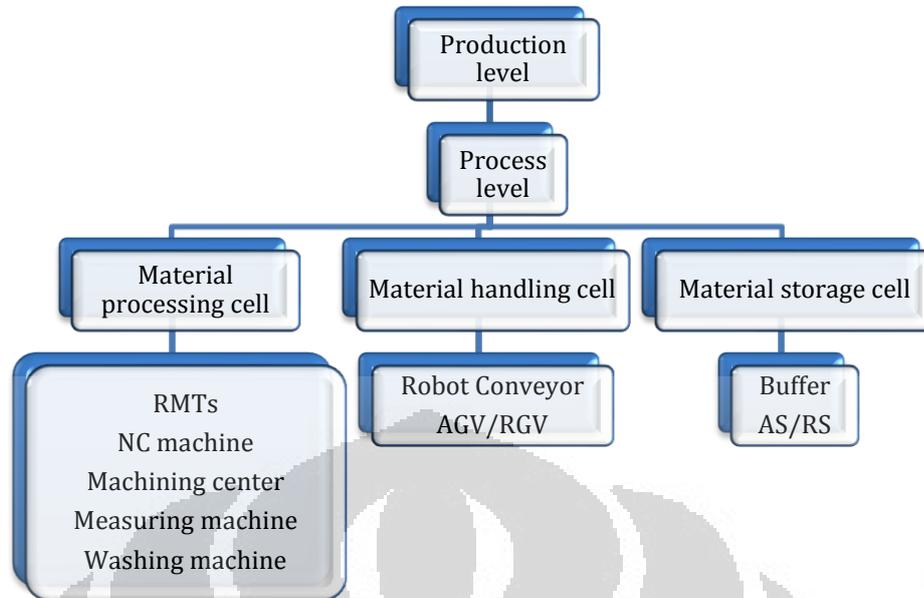
Selain itu, kapasitas dan fungsionalitas dari RMS tidak tetap. RMS akan didesain melalui situasi yang terjadi pada *reconfigurable hardware & software* sehingga kapasitas dan fungsionalitas pada RMS bisa diubah-ubah sepanjang waktu. Oleh karena itu, RMS tidak memiliki *hardware & software* yang tetap.

Perbedaan utama antara kondisi fisik konfigurasi RMS dengan FMS dapat diilustrasikan sebagai berikut: RMS terdiri dari mesin CNC yang mampu dikustomisasi secara bebas dan alat-alat mesin yang rekonfigurabel, sementara itu, FMS secara umum disusun oleh mesin CNC dengan tujuan umum dan otomatisasi yang sudah diprogram.

Konfigurasi umum suatu RMS meliputi jumlah sub-sistem produksi seperti aksi proses, peralatan *material handling*, penyimpanan material, unit kontrol, inspeksi dan stasiun pengukuran. Struktur hierarkis suatu RMS dapat dilihat pada Gambar 3.1.

Level produksi merupakan level tertinggi RMS yang bertanggung jawab dalam hal operasi sistem, dan terdiri dari suatu *master production schedule* (MPS), manajemen sumber daya sistem, *database*, penjadwalan pemeliharaan dan pengukuran kinerja sistem.

Level kedua adalah level proses yang melakukan koordinasi dan kontrolisasi semua sel di dalam RMS.



Gambar 3.1 Contoh Struktur Hierarkis RMS

Sumber: Xiuli Meng (2010)

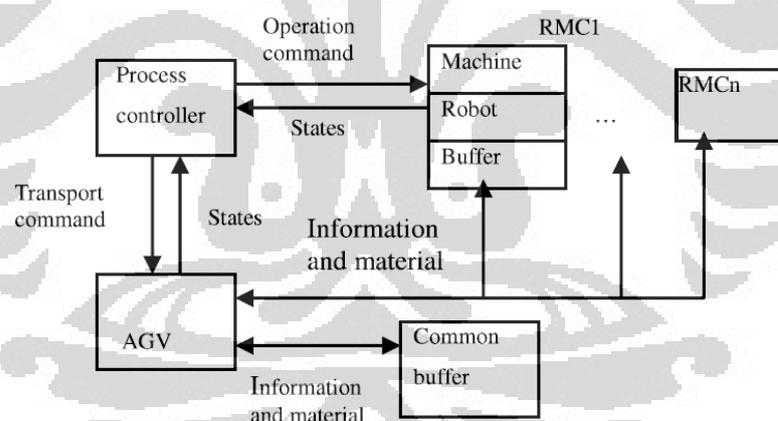
Level ketiga adalah level sel atau level sel manufaktur yang terdiri dari *material processing cell*, *material handling cell* dan *material storage cell*. *Material processing cell* membuat perubahan-perubahan fisik dan logika dalam properti produk. *Material processing cell* terdiri dari beberapa mesin manufaktur rekonfigurabel seperti *CNC machining center*, *measuring machine* dan *washing machine*. *Material handling cell* melakukan transportasi produk-produk diantara *material processing resource* dan *material storage resource*. *Material handling cell* terdiri dari *material transport* dan *handling equipment* seperti AGV atau RGV (*rail guided vehicle*), robot, *mechanical arm* dan konveyor. *Material storage cell* menyimpan produk tetapi tidak bisa merubah properti apapun dari produk tersebut. *Material storage cell* terdiri dari alat penyimpanan material seperti *buffer*, *automated storage and retrieval system (AS/RS)* dan sebagainya.

Level keempat adalah level entitas yang merupakan level dimana terjadinya interaksi antara material/entitas dengan peralatan manufaktur.

Dekomposisi sistem dan komposisi modular adalah dua aspek kunci dalam metode *object-oriented*. Seperti yang ditunjukkan didalam struktur hierarkis RMS, model RMS yang lengkap dibagi kedalam kelas-kelas individual sehingga secara signifikan mampu mengurangi kompleksitas model. Tetapi di dunia nyata,

terdapat entitas lain yang juga eksis di dalam RMS, yaitu *the work piece* (WP). WP didalam RMS hanya menunggu untuk diproses dan digunakan oleh entitas lain. Jadi, WP secara fisik tidak bebas tetapi berada di entitas lain, seperti: WP di dalam *machining center*, WP di dalam *buffer* atau WP di dalam AGV atau robot. WP merupakan aliran material dari RMS yang butuh untuk diantarkan menuju mesin-mesin dan *buffer* melalui AGV dan robot. Aliran informasi adalah jenis aliran lain didalam RMS yang berupa pesan-pesan dari setiap entitas manufaktur dan pesan-pesan umpan balik yang dikirim dari entitas manufaktur lain sebagai input dari pengontrol untuk menghasilkan perintah operasi baru.

Penyebaran aliran informasi dan material dalam RMS ditunjukkan pada Gambar 3.2. AGV, *reconfigurable manufacturing cells* (RMC), *buffers*, dan *process controller* merupakan objek-objek independen yang mengirim, menerima, memproses atau menyimpan pesan-pesan dan material-material. Seperti yang kita lihat pada Gambar 3.2, AGV merupakan saluran penting dalam hal penyebaran informasi dan material tersebut.



Gambar 3.2 Aliran Informasi dan Material pada RMS

Sumber: Xiuli Meng (2010)

3.1.1.1 Struktur Kelas dan Blok Dasar Model RMS *Petri Net*

Konsep dari rekonfigurabilitas mempunyai pengaruh penting dalam proses memodelkan RMS. Ada banyak aspek rekonfigurasi yang meliputi beragam konfigurasi produksi, rekonfigurasi perangkat lunak komunikasi pabrik, konfigurasi pengontrol mesin, konfigurasi dari mesin modular, proses modular, dan peralatan modular. Pada level sistem seharusnya memiliki struktur modular

dan bisa dibuka untuk keperluan *upgrading* dan kustomisasi sistem disamping kemampuan untuk bisa mengintegrasikan berbagai perangkat lunak baru dalam sistem. Pengontrol dan sistem *monitoring* proses seharusnya memiliki kemampuan untuk merekonfigurasi.

Supaya suatu sistem memiliki kemampuan rekonfigurabel, sub-sistem dan komponennya harus sudah didesain rekonfigurabel di saat awal. RMS mensyaratkan rekonfigurasi pada level sistem dan level mesin. Rekonfigurasi level sistem didukung oleh berbagai *software* yang berhubungan dengan fitur produk untuk bisa memodularisasi unit proses, tata letak sistem (*system layout*) dan perencanaan proses. Sementara pada level mesin, rekonfigurasi, desain komponen mesin modular, dan pengontrol berarsitektur terbuka esensial diperlukan untuk menunjang rekonfigurabilitas.

Seperti yang dijelaskan diatas, rekonfigurasi level sistem berarti menambahkan/mengurangi mesin ke/dari sistem, atau menggantikan satu mesin dengan mesin yang lain didalam sistem manufaktur. Pada level sistem, penggantian mesin dipandang sebagai kombinasi dari menambahkan atau mengurangi mesin ke dan dari sistem manufaktur. Oleh karena itu, dalam memodelkan suatu RMS, intisari dari rekonfigurasi RMS adalah menambahkan/mengurangi mesin ke/dari sistem manufaktur. Sementara itu pada waktu yang sama, hubungan komunikasi dan informasi di dalam sistem manufaktur juga diganti menyesuaikan dengan rekonfigurasi yang terjadi.

Jadi, model RMS harus memenuhi permintaan penggunaan kembali perlengkapan, menambah/mengurangi perlengkapan dan perubahan komunikasi dan informasi antar perlengkapan di dalam sistem manufaktur. Sementara itu, rekonfigurasi level mesin berarti rekonfigurasi dari *reconfigurable machine tools* (RMT) yang merepresentasikan kemampuan untuk membuat mesin-mesin dengan beragam fungsi berbeda yang dibutuhkan didalam RMS.

Desain modular adalah inti dari suatu model RMS. Seperti yang kita lihat pada Gambar 3.1, RMS terdiri dari empat level dan setiap level dibagi menjadi bagian yang lebih kecil kedalam beberapa sub-sistem dengan gaya yang sama. Untuk memfasilitasi desain modular ini, metode *object-oriented* (O-O) digunakan. Untuk mengembangkan model modular dari suatu RMS berdasarkan metode O-O,

hal yang pertama adalah mengidentifikasi semua kelas objek yang diminta didalam model RMS. Elemen RMS seperti, mesin, *buffer*, AGV, dan robot bisa dipandang sebagai kelas objek didalam metodologi desain berorientasi objek. Jadi, perlengkapan manufaktur dengan struktur dan fungsi yang sama bisa dikelompokkan kedalam kelas objek dengan menggunakan metode berorientasi objek.

Seperti yang kita ketahui, perlengkapan manufaktur dibagi kedalam tiga kelas. Karena semua perlengkapan pada kelas yang sama memiliki perilaku yang mirip/sama, perlengkapan tersebut bisa dikontrol menggunakan *interface* yang sama dan akan dimodelkan dengan cara yang sama. Sebagai contoh, meskipun perlengkapan untuk memproses material termasuk bagian dari mesin pemroses, perlengkapan tersebut bisa dinyatakan sebagai satu entitas kelas berdasarkan kesamaan fitur, fungsi dan perilaku. Meskipun perlengkapan-perlengkapan penyimpanan material mempunyai banyak gaya dan bentuk serta didistribusikan di tempat-tempat berbeda di dalam RMS, perlengkapan tersebut memainkan peran yang sama dan mempunyai struktur kelas yang sama. Sementara untuk perlengkapan *material handling*, mereka memainkan peran yang berbeda dan mempunyai perbedaan struktur kelas sehingga dalam RMS, sebagai contoh: AGV dan robot dipandang sebagai dua kelas yang berbeda. Empat kelas dari entitas dasar suatu RMS bisa dilihat pada Tabel 3.1.

Tabel 3.1 Empat Kelas pada Entitas Dasar RMS

Nama	Perlengkapan untuk memproses material	Perlengkapan penyimpanan material	Robot	AGV
Atribut	Type, ID Idle/processing WP Breakdown	Type, ID Common buffer Capacity Occupied/empty	Type, ID Idle/busy Empty/loaded WP Breakdown	Type, ID Empty/load WP Breakdown Position Destination
Metode	Setting up Being loaded Processing Being unloaded Breakdown Repair	Being loaded Being unloaded	Start/end loading Start/end unloading Breakdown Repair	Start-stop transporting Loading WP Unloading WP Breakdown Repair

Sumber: Xiuli Meng (2010)

Pada Tabel 3.1, metode merepresentasikan operasi dari tiap kelas objek dan atribut merepresentasikan status dari tiap kelas objek. Blok dasar *petri net* untuk setiap kelas objek digunakan untuk menggambarkan atribut dan perilaku dari kelas yang sama. Metode dari setiap kelas objek digiring menuju transisi dan atribut setiap kelas objek digiring menuju tempat didalam RMS. Hubungan panah secara langsung mengindikasikan adanya kemungkinan pesan dan aliran material yang lewat ke dan dari suatu kelas objek.

Suatu konsep baru yang dinamakan *macro-place* diusulkan untuk mendeskripsikan kombinasi dari kombinasi dari rangkaian transisi, tempat dan panah dalam RMS. Suatu *macro-place* mungkin terdiri dari beberapa *sub-macro-place*. Suatu *sub-macro-place* adalah suatu *macro-place* tetapi pada level yang berbeda didalam model RMS. Setiap blok dasar *petri-net* adalah level terendah *sub-macro-place* yang merepresentasikan aktivitas kinerja dari tiap perlengkapan manufaktur pada model RMS. Oleh karena itu, blok dasar *petri-net* didefinisikan sebagai *basic macro-place*. *Basic macro-place* merepresentasikan perilaku internal dari perlengkapan manufaktur. Suatu tipe spesifik dari transisi disebut *macro-transition* digunakan untuk menghubungkan semua hubungan antar *macro-place*.

Karena ada aliran material dan informasi didalam RMS, tempat didalam *basic macro-place* diklasifikasikan kedalam tempat status dan tempat komunikasi. Tempat status mendeskripsikan status, perintah, dan kondisi atau properti dari sumber daya (*resources*). Tempat komunikasi menyediakan komunikasi diantara *basic macro-place* dengan *macro-place* yang lain. Tempat komunikasi ini bertanggung jawab dalam pengiriman dan penerimaan pesan. *Basic macro-place* (BMP) bisa didefinisikan sebagai berikut:

$$BMP_i = (SP_i, AT_i, CP_i, C_i, T_i) \quad (3.1)$$

Keterangan:

SP_i = satu set terbatas dari tempat status untuk BMP_i

CP_i = satu set terbatas dari tempat komunikasi untuk BMP_i

AT_i = satu set terbatas dari transisi aktivitas untuk BMP_i

C_i = satu set terbatas dari warna yang berhubungan dengan transisi dan tempat dalam BMP_i

T_i = satu set terbatas atribut waktu untuk setiap tanda warna dalam BMP_i

3.1.1.2 Prosedur Memodelkan RMS

Desain modular merupakan salah satu intisari dari model RMS. Seperti yg terlihat pada Gambar 3.1, suatu RMS terdiri dari empat level dan tiap level terdiri dari modul-modul yang berbeda. Untuk memfasilitasi desain modular ini, modul individu dikembangkan pertama kali untuk masing-masing level tersebut. Lalu, modul untuk tiap level yang berbeda diintegrasikan melalui aliran informasi dan pesan yang dimasukkan ke dalam sistem.

Dari atas ke bawah dari struktur hierarkis suatu RMS, tiap level dari RMS dinamakan sebagai *macro-place* yang terdiri dari beberapa *sub-macro place*. *Macro place* memodelkan total jumlah dari banyak tipe perlengkapan, seperti sel-sel atau mesin-mesin. Dari perspektif hierarkis ini, suatu RMS bisa disusun dari banyak *macro place* yang saling berhubungan, seperti *macro place* level sistem, *macro place* level sel dan *macro place* dasar. Secara matematis, *reconfigurable manufacturing system* (S) bisa didefinisikan sebagai berikut:

$$S = (MPS_k, MTS_k, k, C, T, I, O) \quad (3.2)$$

Keterangan:

MPS_k : suatu set *macro place* dalam sistem, $MPS_k = \{MP_{1k}, MP_{2k}, \dots, MP_{m_k}\}$

MP_{ik} : *Macro place*-i pada level-k ($i = 1, 2, \dots, m_k$)

m_k : Total jumlah *macro place* pada level-k

MTS_k : suatu set *macro transition* dalam sistem, $MTS_k = \{MT_{1k}, MT_{2k}, \dots, MT_{n_k}\}$

MT_{jk} : *macro transition*-j pada level-k ($j = 1, 2, \dots, n_k$)

n_k : total jumlah *macro transition* pada level-k

k : level-level dari MP_{ik} ($k = 1, 2, 3, 4$)

C : suatu set tanda warna yang berhubungan dengan *macro place* dan *macro transition*

T : satu set atribut waktu tunda untuk setiap tanda warna pada *macro level petri net*

I ($MPS_k, MTS_k/C$): Fungsi transformasi input yang merepresentasikan hubungan tanda panah dari MPS_k ke MTS_k

O ($MPS_k, MTS_k/C$): Fungsi transformasi output yang merepresentasikan hubungan panah dari MTS_k ke MPS_k .

Setiap *macro place* MP_{ik} merepresentasikan perlengkapan manufaktur, sel atau sistem dari suatu RMS ketika k bernilai berbeda. Setiap *macro transition* MT_{jk} dan fungsi transformasi input-nya I ($MP_{jk}, MT_{jk}/C_{jk}$), fungsi transformasi output-nya O ($MP_{jk}, MT_{jk}/C_{jk}$), satu set tanda warna C (MT_{jk}) dikonstruksikan untuk merepresentasikan hubungan informasi yang lewat antara dua *macro place* MP_{ik} dan MP_{lk} menurut karakteristik aliran komponen/*part*.

Langkah-langkah prosedur pengembangan suatu model RMS dapat dijelaskan sebagai berikut:

1. Menganalisa RMS, menguraikan sistem menjadi beberapa sub-sistem, mengkonstruksikan struktur hierarkis sistem dan mendefinisikan sel dan mesin dalam tiap level.
2. Menggunakan *macro place* untuk merepresentasikan sub-sistem RMS, seperti sistem, sel dan mesin dalam level-level yang berbeda.
3. Mendefinisikan proses dalam tiap level untuk mendapatkan aliran material.
4. Dalam tiap level, sebuah *macro transition* hanya digunakan untuk menghubungkan dua *macro place* dan bukan untuk menyatakan suatu kejadian atau keputusan. *Macro transition* dan *macro place* dihubungkan oleh panah untuk membentuk suatu *petri net* dimana panah tersebut mendeskripsikan komunikasi antar *macro place*. *Petri net* dengan suatu *macro place* dan *macro transition* dinamakan sebagai sebuah *macro level petri net*. *Macro transition* bisa diberhentikan ketika suatu hubungan *macro place* meminta suatu tanda spesifik yang berbeda. *Macro transition* direpresentasikan sebagai sebuah transisi hierarkis yang mungkin diuraikan menjadi satu tempat status yang terhubung dengan satu *output* dan beberapa transisi *input* untuk mengklarifikasi kondisi pemberhentian untuk *macro transition* tersebut.

5. Menggunakan metode *stepwise refinement* untuk mendeskripsikan semua *macro place* menjadi *sub macro place* yang lebih detail sampai semua *macro place* tidak bisa dibagi lagi. Pada cara ini, *macro level petri net* bisa diolah langkah demi langkah dan level demi level. Komunikasi antar *macro place* dalam level yang berbeda dicapai melalui tempat komunikasi. Dalam proses ini, komunikasi antara *petri net* yang lebih tinggi dan *sub macro-place* olahannya juga dicapai pada langkah ini.
6. Dalam tiap level, mengaplikasikan properti warna dan waktu pada *macro level petri net*. Atribut dari suatu RMS, seperti rute komponen dan kapasitas *resource* bisa dideskripsikan sebagai properti warna. Metode analisis properti warna telah dibahas dalam penelitian sebelumnya (Wang & Wu, 1998). Untuk model RMS, analisis tersebut bisa dirangkum sebagai berikut:
 - Dalam tiap level RMS, berdasarkan aliran proses dari tiap tipe komponen p , aliran proses keseluruhan dibagi menjadi satu set sub-proses yang merepresentasikan hanya untuk satu *resource*, memberikan tanda warna ct_{pr} untuk merepresentasikan sub-proses r ($r = 1, 2, \dots, n_k$, n_k adalah total jumlah sub-proses untuk tipe komponen p dalam level- k). Untuk setiap sub-proses mengidentifikasi *resource* dalam level yang berbeda, seperti sebuah sel manufaktur dan sebuah level mesin, yang secara aktual melakukan proses tersebut.
 - Dalam tiap level, definisikan set tanda warna yang berhubungan dengan tiap *macro place* yang merepresentasikan sel manufaktur, mesin, dan atribut waktu dari tanda warna tersebut.
 - Dalam tiap level, definisikan set tanda warna, fungsi transformasi input dan output untuk setiap *macro transition*.

Untuk setiap tipe komponen p dan setiap sub-proses r dari komponen p , pilihlah pasangan tanda warna $(ct_{pr}, ct_{p,r+1})$, identifikasi alokasi tanda warna tersebut di *macro place* (MP_{ik} , MP_{lk}) ($ik, lk = 1, 2, \dots, m_k$, m_k adalah jumlah *macro place* di dalam level k) dan *macro transition* MT_{jk} dari *macro place*. Sebagai

konsekuensi, pesan yang lewat $[MP_{ik}/ct_{pr} - MT_{jk} - MP_{ik}/ct_{p,r+1}]$ sudah ditentukan. Oleh karena itu, fungsi input untuk $MT_{ik} = I(MP_{ik}, MT_{jk}/ct_{pr}) = ct_{pr}$ dan fungsi output untuk $MT_{ik} = O(MP_{ik}, MT_{jk}/ct_{pr}) = ct_{p,r+1}$.

7. Untuk setiap *macro place* dasar dalam RMS, tambahkan fungsi-fungsi transformasi yang saling berhubungan dan kejadian-kejadian eksternal didalam tiap *macro place* dasar untuk menspesifikasi perilaku dinamis internal *macro place* dasar tersebut. Untuk setiap *macro place* dasar, warna dari tempat dan transisi merepresentasikan kejadian eksternal dan sinyal kontrol luar, sehingga warna tetap tidak berubah ketika aktivitas transisi internal berhenti. Oleh karena itu:

$$I(P_i, AT_i/cg) = cg \quad cg: \text{ warna transisi } AT_i$$

$$O(P_i, AT_i/cg) = cg \quad P_i: \text{ tempat komunikasi } CP_i \text{ atau tempat status } SP_i.$$

Tempat input komunikasi menerima tanda warna dari luar dan mengirim tanda warna ke tempat lain ketika aktivitas transisi sebelumnya berhenti.

Untuk setiap level RMS, ulangi langkah 3 dan 6. Pada level mesin, setiap perlengkapan manufaktur dimodelkan sebagai sebuah *macro place* dasar. Dalam tiap level, meski *macro place* dihubungkan oleh *macro transition*, *macro place* adalah tempat komunikasi didalam *macro place* yang terhubung dengan *macro transition*.

Rancangan model Xiuli Meng ini memiliki karakteristik pendekatan berorientasi objek dan memiliki beberapa keunggulan sebagai berikut:

1. Model memiliki tingkat modularitas tinggi karena memodelkan objek-objek, seperti *macro place* dasar;
2. Teknik pemodelan memiliki tingkat *reusability* yang tinggi karena memodelkan objek-objek yang bisa digunakan kembali ketika mengkonfigurasi RMS;
3. Pemeliharaan model, seperti pemutakhiran dan rekonfigurasi bisa berlangsung dengan praktis.

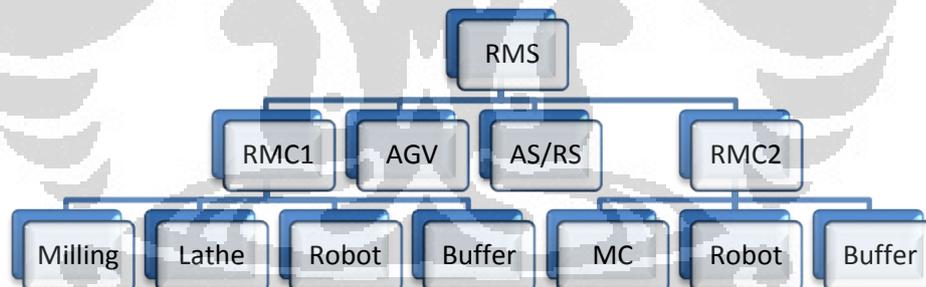
Sebuah model RMS baru bisa diperoleh dari model RMS yang telah eksis ketika sistem manufaktur direkonfigurasi dengan secara mudah memilih atau memodifikasi objek-objek pemodelan. Sebagai contoh, ketika fasilitas ekstra

ditambahkan, RMS baru bisa dimodelkan dengan pertama mengkonstruksi (atau memodifikasi) *macro place* dasar untuk perlengkapan fisik baru yang tersimpan didalam kelas yang sudah eksis, lalu mengkombinasikan dengan model RMS yang sudah eksis untuk membentuk sebuah model RMS baru. Jika aliran proses dari sebuah RMS telah berubah atau tipe komponen baru ditambahkan, sementara perlengkapan manufaktur tetap tidak berubah, RMS baru secara lebih mudah dimodelkan dengan menambahkan tanda warna dan fungsi transformasi yang sesuai dengan model RMS yang sudah eksis menurut aliran proses dan tipe komponen baru tersebut. Dalam kata lain, hal ini merupakan evolusi dari sistem yang sudah eksis dibanding membuatnya dari awal kembali.

3.1.1.3 Hasil Model RMS *Petri Net* (Xiuli Meng, 2010)

Untuk mengilustrasikan prosedur pembuatan model RMS, mari perhatikan contoh RMS yang dibuat oleh Xiuli meng berikut ini.

Model RMS ini terdiri dari dua *reconfigurable manufacturing cells* (RMCs) yang dinamakan sebagai RMC1 dan RMC2, tempat penyimpanan AS/RS, dan AGV seperti yang ditunjukkan pada Gambar 3.3.



Gambar 3.3 Struktur Model RMS

Sumber: Xiuli Meng (2010)

AGV adalah *resource* yang digunakan untuk memuat, menggerakkan dan membongkar muat material antara dua sel manufaktur dengan tempat penyimpanan AS/RS. Operasi dimulai ketika AGV membawa bahan mentah dari AS/RS ke sel manufaktur dimana proses berlangsung di dua sel manufaktur RMC1 dan RMC2. Setelah diproses, AGV mengantarkan produk jadi ke AS/RS. Sebagai contoh, terdapat dua tipe komponen, komponen tipe A dan komponen

tipe B. Komponen tipe A diproses pertama kali oleh RMC1 dan lalu diproses oleh RMC2.

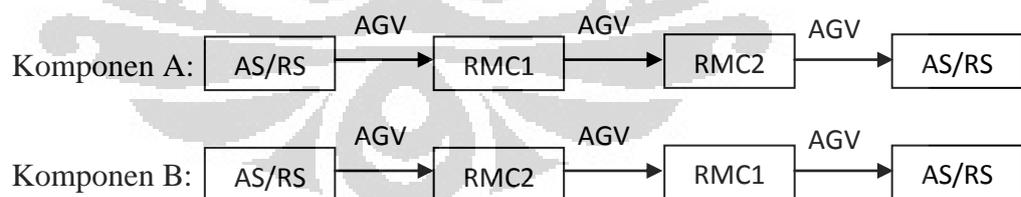
Sementara itu, komponen tipe B diproses pertama kali oleh RMC2 dan lalu diproses oleh RMC1. RMC1 terdiri dari *CNC lathe machine*, *CNC milling machine*, robot dan *buffer*. *Buffer* bisa menyimpan komponen selesai proses dan komponen mentah yang menunggu untuk diproses. Dalam RMC1, komponen tipe A diproses pertama kali oleh *CNC lathe machine* dan kemudian oleh *CNC milling machine*. Komponen tipe B diproses pertama kali oleh *CNC milling machine* dan kemudian oleh *CNC lathe machine*. RMC2 terdiri dari *machining center (MC)*, robot dan *buffer*.

Langkah-langkah prosedur detail pemodelan RMS ini dapat dijelaskan sebagai berikut:

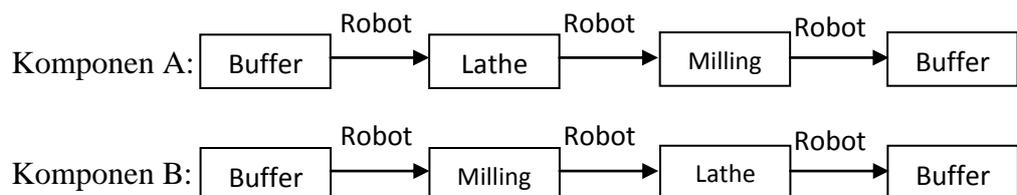
1. Analisis dan konstruksikan struktur hierarkis dari RMS seperti pada Gambar 3.3.
2. *Macro place* digunakan dalam sistem RMS yang didefinisikan sebagai berikut:

Keseluruhan sistem dipandang sebagai sebuah *macro place* level sistem. RMC1, RMC2, AGV dan AS/RS dipandang sebagai *macro place* empat level sel. *Lathe*, *Milling*, *Robot*, *buffer*, dan *MC* dipandang sebagai *macro place* lima level mesin.

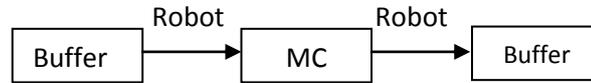
3. Aliran proses dalam level sistem adalah sebagai berikut:



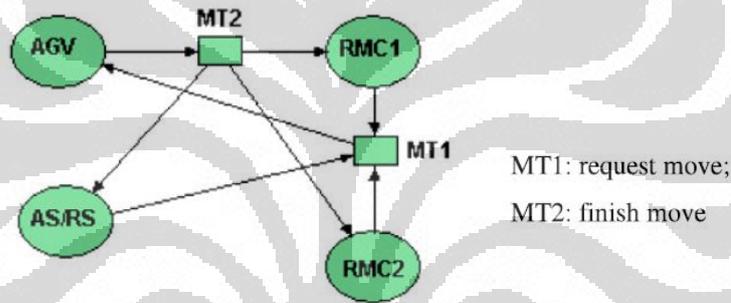
Aliran proses dalam level sel pada level RMC1 adalah sebagai berikut:



Aliran proses dalam level sel RMC2 untuk komponen A dan komponen B adalah sebagai berikut:



4. *Macro transition* digunakan untuk menghubungkan *macro place* yang saling berhubungan di setiap level. *Macro-level petri net* pada level sistem ditunjukkan pada Gambar 3.4. Karena MT1 bisa berhenti ketika *macro place* RMC1 atau RMC2 memiliki tanda yang diminta, *macro transition* MT1 diuraikan menjadi Pmt1 yang terhubung dengan satu output transisi mt14 dan tiga output transisi mt11, mt12, dan mt13 seperti yang ditunjukkan pada Gambar 3.5.



Gambar 3.4 *Macro-level petri net* pada level sistem

Sumber: Xiuli Meng (2010)

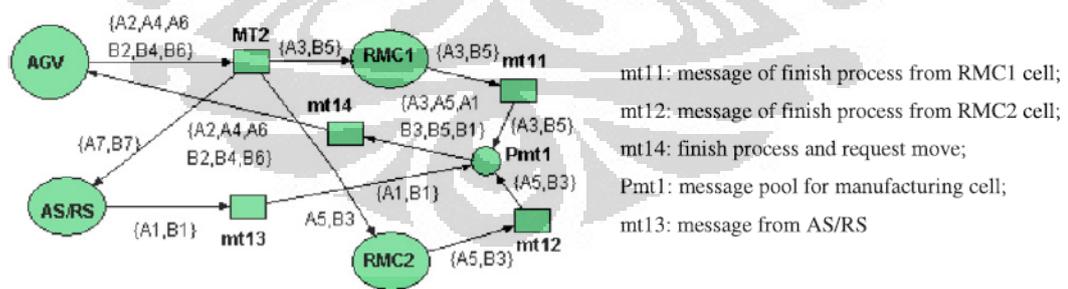
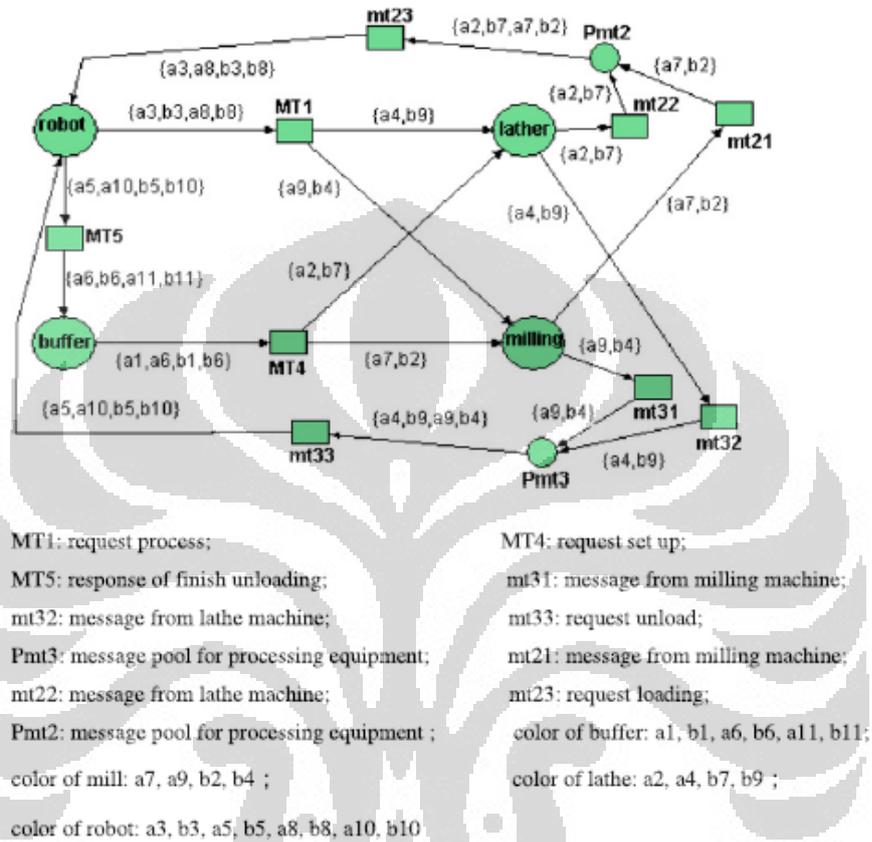


Fig. 6. Macro level Petri net of the system level.

Gambar 3.5 Detail *Macro-level petri net* pada level sistem

Sumber: Xiuli Meng (2010)

5. Mengembangkan *macro place* menggunakan *sub macro-level petri net* pada levelnya yang lebih rendah. RMC1 diolah dengan *macro-level petri net* RMC1 pada Gambar 3.6. RMC2 diolah dengan *macro-level petri net* RMC2 pada Gambar 3.7.



Gambar 3.6 Macro-level petri net RMC1

Sumber: Xiuli Meng (2010)

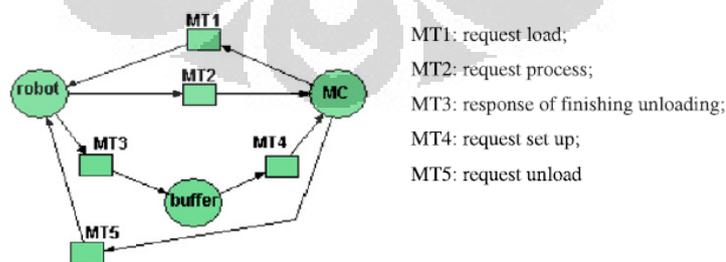


Fig. 8. Macro level Petri net of the RMC2 cell.

Gambar 3.7 Macro-level petri net RMC2

Sumber: Xiuli Meng (2010)

6. Mengaplikasikan properti waktu dan warna.

Sub-proses beserta warna dan sumber manufaktur-nya ditunjukkan pada Tabel 3.2 dan 3.3.

Tabel 3.2 Aliran sub-proses untuk setiap komponen pada level sistem

Komponen A			Komponen B		
Sub-proses	Warna	Sumber	Sub-proses	Warna	Sumber
Komponen menunggu di AS/RS	A1	AS/RS	Komponen menunggu di AS/RS	B1	AS/RS
AGV antar komponen ke RMC1	A2	AGV	AGV antar komponen ke RMC2	B2	AGV
RMC1 memproses komponen	A3	RMC1	RMC2 memproses komponen	B3	RMC2
AGV antar komponen ke RMC2	A4	AGV	AGV antar komponen ke RMC1	B4	AGV
RMC2 memproses komponen	A5	RMC2	RMC1 memproses komponen	B5	RMC1
AGV antar komponen ke AS/RS	A6	AGV	AGV antar komponen ke AS/RS	B6	AGV
Komponen tiba di AS/RS	A7	AS/RS	Komponen tiba di AS/RS	B7	AS/RS

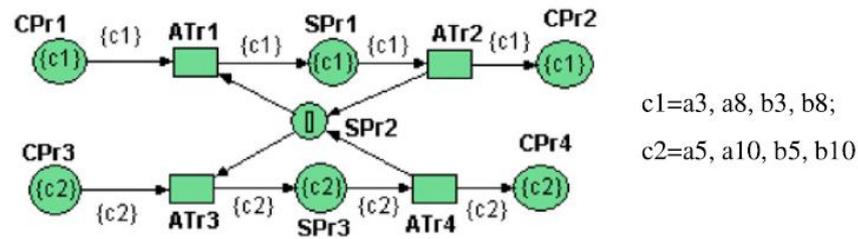
Sumber: Xiuli Meng (2010)

Tabel 3.3 Aliran sub-proses untuk setiap komponen pada level sel RMC1

Komponen A			Komponen B		
Sub-proses	Warna	Sumber	Sub-proses	Warna	Sumber
Komponen menunggu di Buffer	a1	Buffer	Komponen menunggu di Buffer	b1	Buffer
Pengaturan Lathe	a2	Lathe	Pengaturan Mill	b2	Mill
Robot muat komponen ke Lathe	a3	Robot	Robot muat komponen ke Mill	b3	Robot
Lathe memproses komponen	a4	Lathe	Mill memproses komponen	b4	Mill
Robot bongkar komponen dari Lathe	a5	Robot	Robot bongkar komponen dari Mill	b5	Robot
Komponen menunggu di Buffer	a6	Buffer	Komponen menunggu di Buffer	b6	Buffer
Pengaturan Mill	a7	Mill	Pengaturan Lathe	b7	Lathe
Robot muat komponen ke Mill	a8	Robot	Robot muat komponen ke Lathe	b8	Robot

Sumber: Xiuli Meng (2010)

7. Tambahkan tanda warna untuk tiap *macro place* dasar untuk mendeskripsikan perilaku dinamis internal dari mesin manufaktur. Gambar 3.8 menunjukkan properti warna dari *macro place* robot.

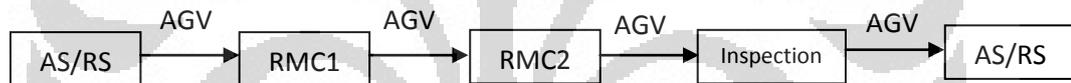


Gambar 3.8 Properti Warna dari *Robot Macro-Place*

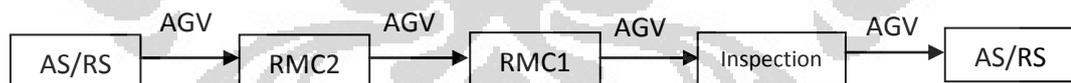
Sumber: Xiuli Meng (2010)

Untuk mempertimbangkan rekonfigurabilitas model yang dirancang, mari kita mempertimbangkan situasi dimana sebuah tipe baru sel kerja ditambahkan karena timbulnya perubahan pada lingkungan manufaktur. Sebagai contoh, sel inspeksi ditambahkan yang disusun oleh sebuah *measuring machine* dan sebuah buffer. Aliran proses dari komponen tipe A dan komponen tipe B pada level sistem berubah sebagai berikut:

Komponen A:



Komponen B:



Model baru untuk merekonfigurasi sistem bisa secara cepat dibangun berdasarkan model lama. *Macro place* untuk AGV, AS/RS, RMC1 dan RMC2 tetap tidak berubah. Tambahkan sebuah *macro place* baru bernama *macro place* inspeksi. Komunikasi antara inspeksi dan *macro place* lain dideskripsikan oleh penambahan panah dan atribut warna. *Macro place* inspeksi bisa diolah dengan *sub macro-place* yang lebih detail. Level terendah dari proses pengolahan ini adalah dua *macro place* dasar yang dinamakan *measuring machine* dan buffer yang bisa diperoleh dari perpustakaan/penyimpanan blok dasar. *Petri net* level sistem yang baru ditunjukkan pada Gambar 3.9.

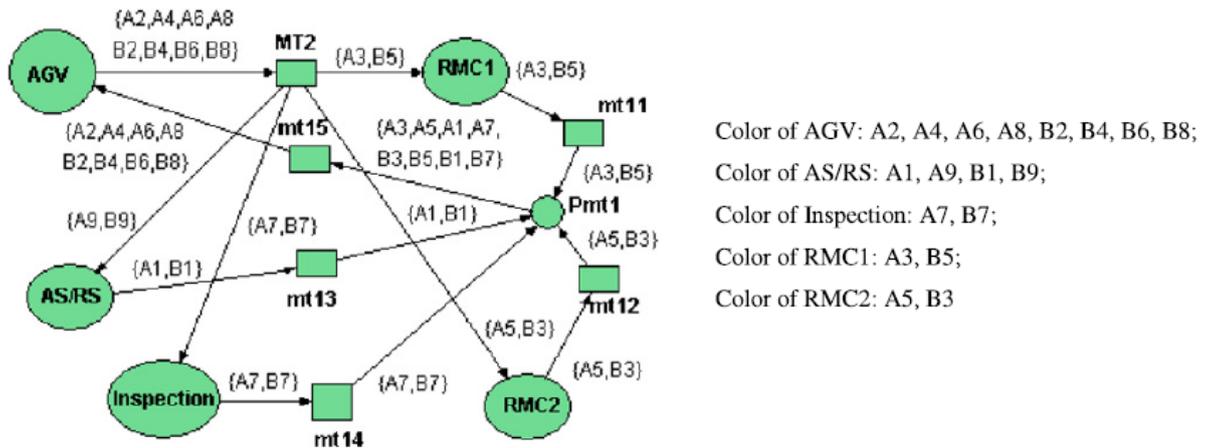


Fig. 10. System level Petri net of reconfigured manufacturing system.

Gambar 3.9 Model RMS Petri Net Konfigurasi 2 (dengan inspeksi)

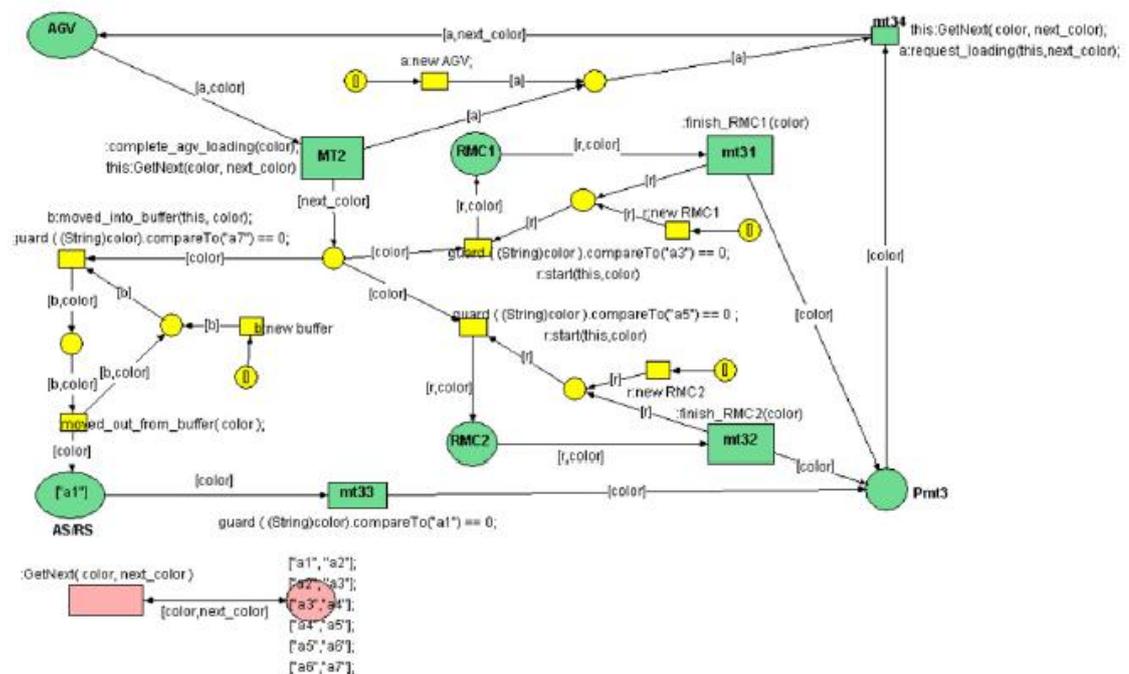
Sumber: Xiuli Meng (2010)

Karena model sistem ini luas, teknik analisis secara manual mungkin sulit untuk dilakukan. Oleh karena itu, Xiuli Meng menggunakan sebuah simulator *petri net* stokastik yang disebut Renew sebagai alat simulasi. Renew menawarkan desain grafis, *user-friendly interface*. Model RMS ini dikonstruksikan menggunakan Renew untuk memahami prosedur. Informasi berguna bisa diraih, seperti informasi tempat, pemberhentian transisi, aliran tanda, dan sinkronisasi jarak. Menggunakan Renew, aliran tanda bisa secara jelas terlihat dalam model *petri net*. Hal ini memperlihatkan jejak simulasi yang berlangsung. Kita bisa melihat secara nyata dimana transisi berhenti dan dimana tanda dikonsumsi dan diproduksi. Renew juga digunakan untuk memverifikasi dan memodifikasi model *petri net*. Model RMS *petri net* pada level sistem yang dibuat menggunakan Renew dapat dilihat pada Gambar 3.10. Hasil simulasi menjamin tingkat kualitas yang baik dari model *petri net* tersebut dan menjamin sistem mampu beroperasi dalam kondisi yang stabil dan bebas *deadlock*.

Tabel 3.4 Aliran sub-proses tambahan pada model RMS konfigurasi 2 (dengan inspeksi)

Komponen A			Komponen B		
Sub-proses	Warna	Sumber	Sub-proses	Warna	Sumber
AGV antar komponen ke inspeksi	A6	AGV	AGV antar komponen ke inspeksi	B1	AGV
Inspeksi memproses komponen	A7	RMC2	Inspeksi memproses komponen	B2	RMC1
AGV antar komponen ke AS/RS	A8	AGV	AGV antar komponen ke AS/RS	B3	AGV
Komponen tiba di AS/RS	A9	AS/RS	Komponen tiba di AS/RS	B4	AS/RS

Sumber: Xiuli Meng (2010)

**Gambar 3.10** Tampilan Model RMS Petri Net Menggunakan *Renew*

Sumber: Xiuli Meng (2010)

3.2 Pengolahan Data

Model simulasi RMS akan coba penulis buat dengan menggunakan referensi data pada model RMS *petri net* (Xiuli Meng, 2010). Model simulasi RMS ini dibuat dengan beberapa tujuan sebagai berikut:

- Memperoleh model simulasi RMS 2D dan 3D sehingga mampu memperlihatkan secara nyata aliran proses, material dan informasi dalam sistem;
- Memperoleh model simulasi RMS berorientasi objek untuk mengetahui karakteristik dan perilaku RMS secara lebih baik

- Memperoleh model simulasi RMS sehingga mampu melakukan analisis model secara lebih baik yang dilihat berdasarkan hasil produksi dan penggunaan waktu produksi selama proses simulasi berjalan.

3.2.1 Pembuatan Model Simulasi RMS

Hal yang dilakukan pertama kali untuk melakukan pembuatan model simulasi berorientasi objek adalah pembuatan bahasan pemodelan yang dapat dilihat yakni menggunakan *unified modeling language* (UML). Hal ini dilakukan dengan cara membuat kelas-kelas yang dibutuhkan pada setiap sistem yang ada, seperti entitas, proses, dsb. Kemudian setelah menentukan kelas-kelas yang akan dibuat selanjutnya adalah memetakan kelas-kelas tersebut ke dalam *software* simulasi yang digunakan.

Dalam pemodelan dan simulasi berorientasi objek, kelas didefinisikan sebagai tipe data yang didefinisikan oleh pengguna (*user*). Kelas ini mendesain tipe data baru untuk membuat definisi dari suatu konsep yang dipikirkan oleh pengguna. Sebagai contoh, jika Anda ingin membuat tipe baru dari unit transportasi dalam pabrik yang tidak bisa didefinisikan oleh tipe standar maka Anda harus mendefinisikan unit transportasi tersebut dengan membuat tipe transportasi baru melalui intervensi properti, metode dan perilaku unit transportasi baru yang akan Anda buat tersebut didalam *software* simulasi yang digunakan.

Kelas adalah sebuah spesifikasi yang akan menghasilkan objek-objek yang merupakan inti dari pengembangan dan desain berorientasi objek. Untuk pembuatan model simulasi RMS ini, penulis menggunakan *software Plant Simulation*. Di dalam *Plant Simulation*, kelas standar terdiri dari enam kategori:

1. Kelas objek aliran material

Kelas objek aliran material dibagi menjadi dua kelompok, yaitu objek aliran material aktif dan objek aliran material pasif. Unit yang termasuk dalam objek aliran material berjalan, antara lain: *Transporter*, *Container* dan *Parts*. Sementara itu, unit yang termasuk dalam objek aliran material statis, antara lain: *Store*, *Track*, *TwolaneTrack*, dan *Line*.

Perilaku umum dari kelas objek aliran material aktif berupa kemampuan untuk menerima objek berjalan/*mobile units* (MUs), menyimpan MUs

untuk jumlah waktu tertentu dan lalu secara otomatis melepaskan MUs menuju objek selanjutnya. Sementara itu, perilaku umum dari kelas objek aliran material pasif adalah tidak mampu secara otomatis melepaskan MUs dan hanya bisa memiliki arti bila bekerja bersama objek aliran material aktif seperti *transporter* untuk mengantar MUs.

2. Kelas Sumber daya (*resources*)

Kelas sumber daya membuat objek berjalan/*mobile units* (MUs) menurut definisi yang kita buat. Sumber daya bisa membuat berbagai tipe komponen/*parts* dalam urutan tunggal atau bercampur dengan tipe komponen lain.

3. Kelas Objek umum

Kelas objek umum terdiri dari empat bagian, yaitu *frame*, *connector*, *event controller*, dan *interface*. *Frame* adalah landasan dari semua model. Kita bisa menggunakan *frame* untuk membuat objek buatan kita sendiri dengan desain perilaku yang kita inginkan. *Connector* berupa garis dan tanda panah yang berfungsi untuk menghubungkan semua objek dengan *frame*. *Event controller* berfungsi seperti pengatur kejadian selama simulasi berjalan yang berfungsi untuk menyelaraskan semua kejadian dan proses yang terjadi dalam suatu model simulasi yang dibuat. *Interface* memiliki fungsi untuk memfasilitasi atau menghubungkan beberapa *frame* model simulasi yang dibuat supaya saling terkait satu sama lain.

4. Kelas Objek berjalan

Objek berjalan/*mobile units* (MUs) adalah unit objek dalam model simulasi yang merepresentasikan material yang akan berjalan didalam objek aliran material ketika model simulasi diaktifkan.

5. Kelas daftar, tabel dan metode

Kelas daftar dan tabel adalah data detail yang digunakan sebagai data input tambahan untuk dimasukkan ke dalam kelas objek yang lain seperti MUs dan objek aliran material. Sementara itu, metode yang dimaksud dalam kelas ini adalah bahasa pemrograman tambahan atau pengaturan tambahan yang digunakan dalam membuat suatu model simulasi.

6. Kelas Objek tampilan

Kelas objek tampilan merepresentasikan visualisasi model simulasi yang telah dibuat dalam bentuk dua dan tiga dimensi.

Dalam model simulasi RMS ini, penulis menggunakan beberapa kelas objek yang sesuai dengan persyaratan pada model *petri net* RMS (Xiuli Meng, 2010).

3.2.1.1 Kelas Objek Aliran Material Model Simulasi RMS

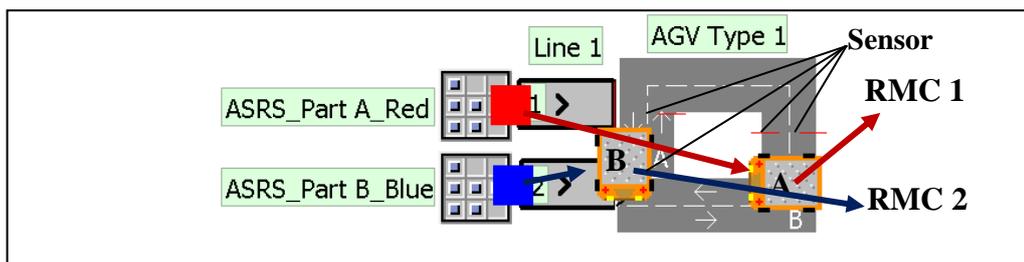
Pada model simulasi RMS ini, penulis menggunakan beberapa objek yang termasuk dalam kelas objek aliran material yang akan dijelaskan sebagai berikut:

a. *Automated Guided Vehicle* (AGV)

Penulis menggunakan tiga jenis AGV untuk membangun model simulasi RMS ini yang akan dijabarkan sebagai berikut:

- AGV Tipe Satu

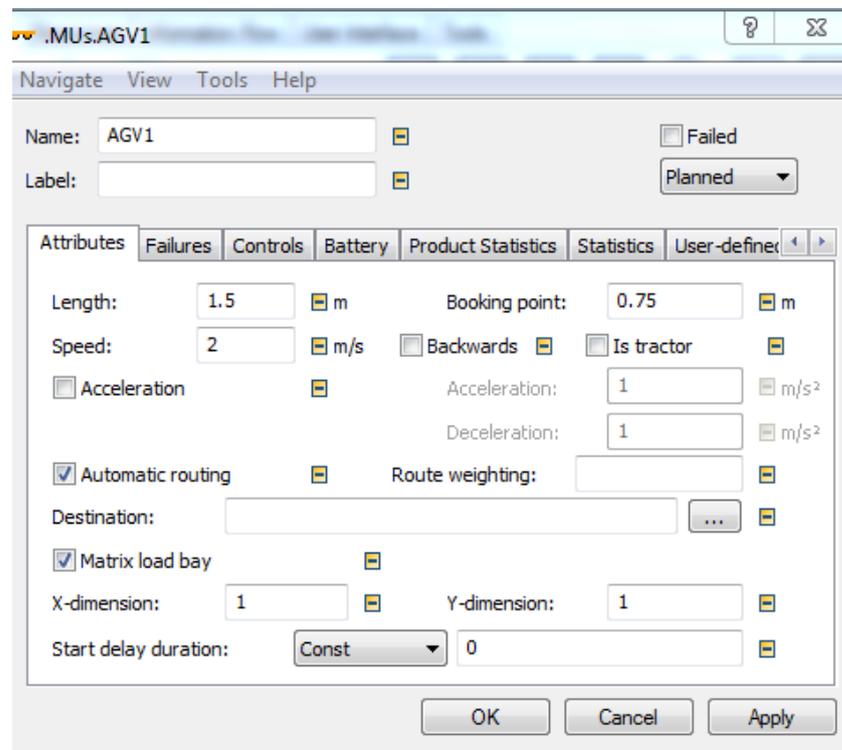
AGV tipe satu seperti yang digambarkan pada Gambar 3.11 merepresentasikan truk tanpa awak yang dikendalikan secara otomatis pada pusat kontrol yang terhubung dengan pendeteksi sensor yang melekat dilintasan AGV tipe satu. AGV tipe satu ini terdiri dari dua buah. AGV tipe satu no.A berfungsi untuk memuat komponen A (berwarna merah) dari AS/RS untuk diantar menuju RMC-1. Sementara itu, AGV tipe satu no.B berfungsi untuk memuat komponen B (berwarna biru) dari AS/RS untuk diantar menuju RMC-2.



Gambar 3.11 AGV Tipe Satu

Atribut dari AGV tipe satu ini dapat dilihat pada Gambar 3.12. AGV tipe satu memiliki panjang 1,5 m dan kapasitas untuk mengangkut satu komponen sekali pengantaran. AGV tipe satu diatur dengan kecepatan bergerak secara konstan 2 m/s. AGV tipe satu juga diatur dengan mode

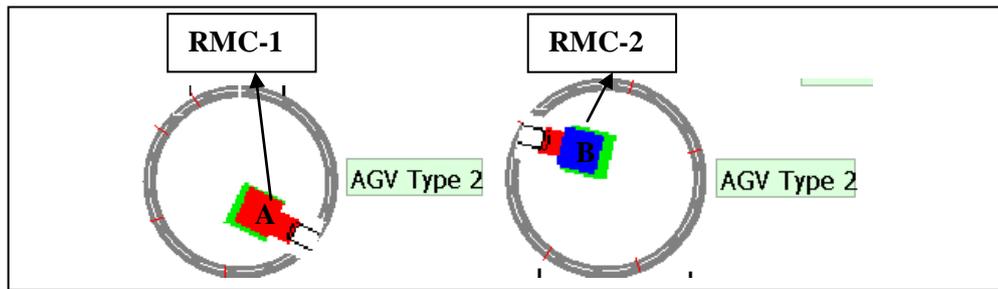
automatic routing mengikuti kontrol sensor yang berada pada jalur AGV tipe satu.



Gambar 3.12 Atribut AGV Tipe Satu

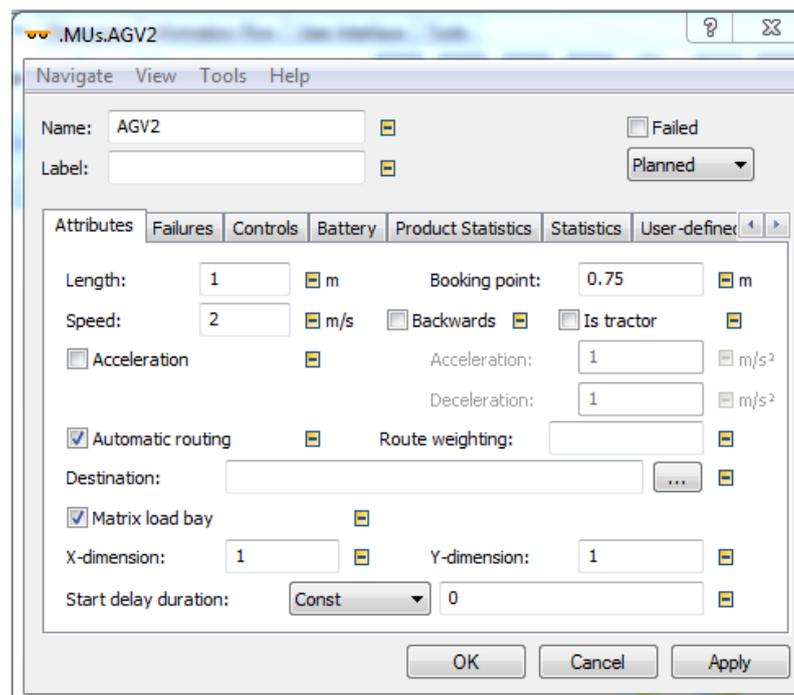
- AGV Tipe Dua

AGV tipe dua seperti yang bisa dilihat pada Gambar 3.13 merepresentasikan kendaraan tanpa awak yang dikendalikan secara otomatis pada pusat kontrol yang terhubung dengan pendeteksi sensor yang melekat dilintasan AGV tipe dua. Berbeda dengan AGV tipe satu, AGV tipe dua bergerak menggelantung direl dengan lintasan AGV tipe dua yang berbentuk lingkaran dengan sensor pada lintasan yang jauh lebih banyak. AGV tipe dua berjumlah dua buah. AGV tipe dua no.A berfungsi untuk memuat komponen A dan komponen B untuk diproses pada RMC-1. Sementara itu, AGV tipe dua no.B berfungsi untuk memuat komponen A dan komponen B untuk diproses pada RMC-2.



Gambar 3.13 AGV Tipe Dua

Atribut dari AGV tipe dua ini dapat dilihat pada Gambar 3.14. AGV tipe dua memiliki panjang 1 meter dan kapasitas untuk membawa satu komponen sekali pengantaran. AGV tipe dua diatur dengan kecepatan bergerak secara konstan 2 m/s. AGV tipe dua juga diatur dengan mode *automatic routing* mengikuti kontrol sensor yang berada pada jalur AGV tipe dua.

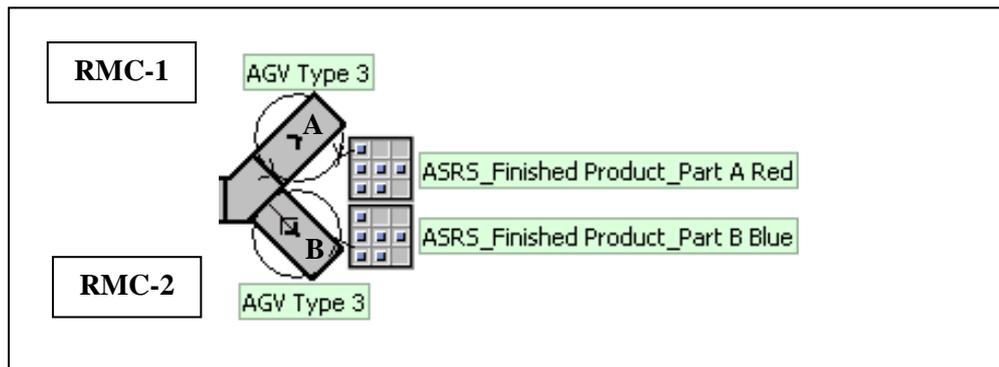


Gambar 3.14 Atribut AGV Tipe Dua

- AGV/RGV Tipe Tiga

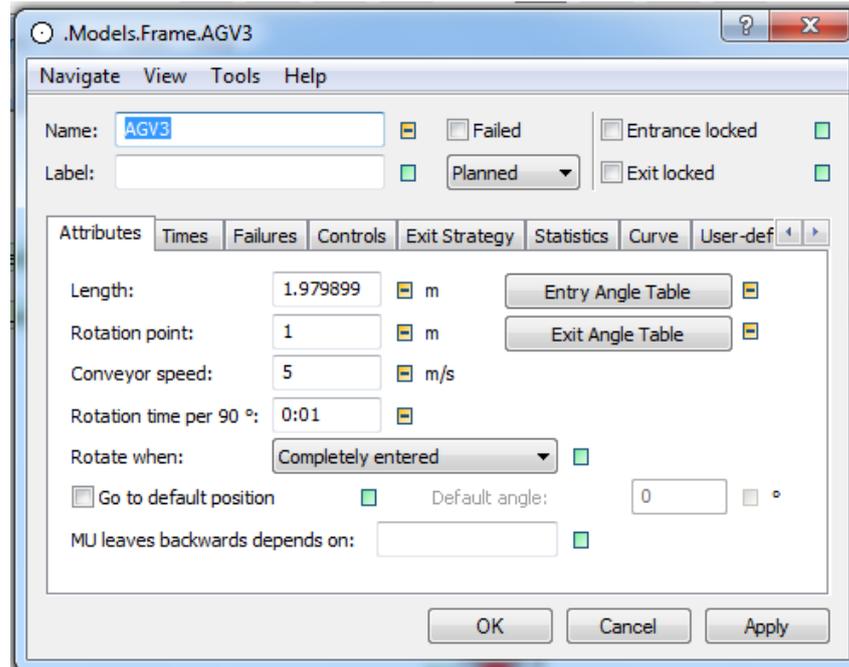
Berbeda dengan AGV tipe satu atau dua, AGV tipe tiga lebih tepat disebut sebagai RGV (*Rail Guided Vehicle*) karena berbentuk seperti konveyor yang memiliki motor penggerak untuk mampu digerakkan

dengan pengontrol secara otomatis. AGV tipe tiga terdiri dari dua buah. AGV tipe tiga no.A berfungsi untuk mengantarkan komponen A yang telah selesai diproses dari RMC-1 dan RMC-2 menuju AS/RS *finished product*. Sementara itu, AGV tipe tiga no.B berfungsi untuk mengantarkan komponen B yang telah selesai diproses dari RMC-2 dan RMC-1 menuju AS/RS *finished product*.



Gambar 3.15 AGV Tipe Tiga

Atribut dari AGV tipe tiga ini dapat dilihat pada Gambar 3.16. AGV tipe tiga ini memiliki panjang 2 m dan berkecepatan 5 m/s. Selain itu, AGV ini dilakukan pengaturannya untuk melakukan rotasi sebesar 90^0 dalam waktu satu detik. AGV tipe tiga ini mampu membawa komponen hingga sebanyak dua buah.



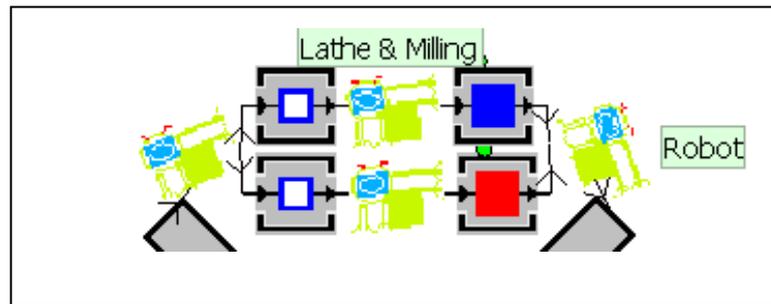
Gambar 3.16 Atribut AGV Tipe Tiga

b. Robot

Robot yang digunakan dalam pembuatan model simulasi RMS seperti yang bisa dilihat pada Gambar 3.17 hanya terdiri dari satu tipe dengan jumlah sebanyak 6 buah. Masing-masing robot memiliki tugas yang berbeda sesuai dengan pengaturan yang telah penulis lakukan. Namun, secara garis besar, tugas utama robot ini adalah membawa komponen baik itu komponen A maupun komponen B dari objek terdahulu menuju objek lain. Detail tugas masing-masing robot dapat dilihat pada Tabel 3.5.

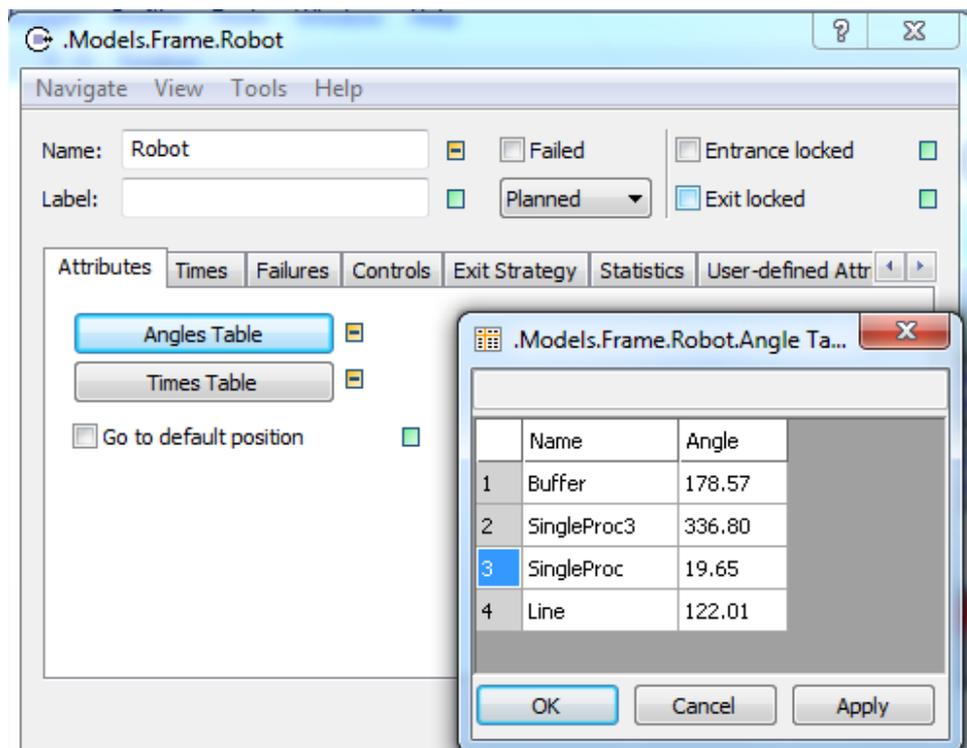
Tabel 3.5 Tugas Setiap Robot pada Model Simulasi RMS

No.Robot	Lokasi	Objek terdahulu	Objek tujuan
1	RMC-1	Buffer-1	CNC Lathe/Milling Machine
2		CNC Lathe/Milling Machine	CNC Milling/Lathe Machine
3		CNC Milling/Lathe Machine	CNC Lathe/Milling Machine
4		CNC Milling/Lathe Machine	Buffer-2
5	RMC-2	Buffer-3	MC
6		MC	Buffer-4



Gambar 3.17 Robot

Atribut dari Robot ini dapat dilihat pada Gambar 3.18 Robot memiliki kemampuan fleksibel untuk membawa komponen dari dan menuju objek sesuai dengan yang diperintah oleh pusat kontrol. Pada model simulasi RMS ini, robot diatur untuk memiliki kapasitas untuk membawa satu komponen sekali pengantaran.



Gambar 3.18 Atribut Robot

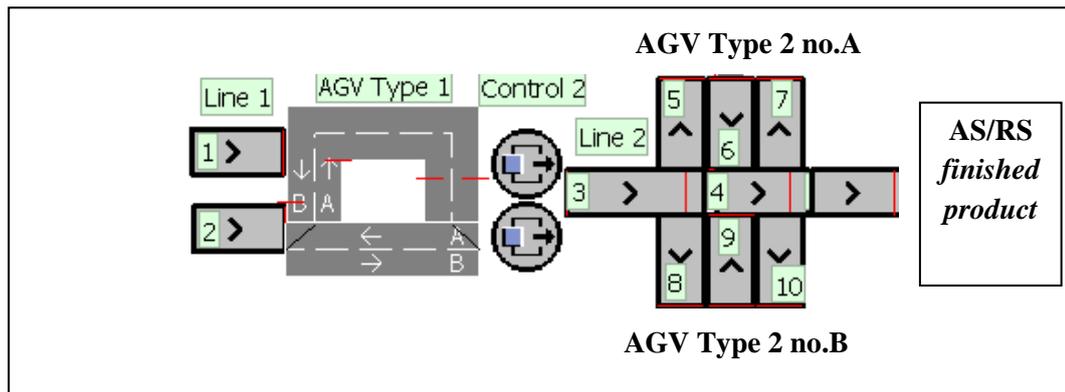
c. Konveyor otomatis

Berbeda dengan model *petri net* RMS yang dibuat oleh Xiuli Meng, penulis pada pembuatan model simulasi RMS ini melakukan modifikasi dengan menggunakan konveyor otomatis untuk membantu dalam proses *material handling*. Konveyor otomatis ini berfungsi untuk mengantarkan komponen dari objek satu ke objek yang lain sesuai dengan pengaturan konfigurasi yang dilakukan. Konveyor yang bisa dilihat pada Gambar 3.19 disebut otomatis karena pada lintasan konveyor dilengkapi dengan sensor-sensor untuk mengidentifikasi komponen yang melintas untuk diantarkan menuju lintasan konveyor lain sesuai dengan yang diatur dalam sensor tersebut.

Terdapat potongan sembilan konveyor otomatis untuk membentuk perilaku model seperti yang dideskripsikan pada model *petri net* Xiuli Meng. Detail terkait fungsi kesembilan potongan konveyor otomatis tersebut dapat dilihat pada Tabel 3.6.

Tabel 3.6 Fungsi Tiap Konveyor Otomatis

No.	Fungsi
1	Mengantarkan komponen A dari AS/RS menuju AGV tipe satu no.A
2	Mengantarkan komponen B dari AS/RS menuju AGV tipe satu no.B
3	-Mengantarkan komponen A dari AGV tipe satu no.A menuju Konveyor no.5 -Mengantarkan komponen B dari AGV tipe satu no.B menuju Konveyor no.6
4	-Mengantarkan komponen A dari Konveyor no.6 menuju Konveyor no.10 -Mengantarkan komponen B dari Konveyor no.6 menuju AS/RS <i>finished product</i> -Mengantarkan komponen A dari Konveyor no.9 menuju AS/RS <i>finished product</i> -Mengantarkan komponen B dari Konveyor no.9 menuju Konveyor no.7
5	Mengantarkan komponen A dari Konveyor no.3 menuju AGV tipe dua no.A
6	Mengantarkan komponen A dan B dari AGV tipe dua no.A menuju Konveyor no.4
7	Mengantarkan komponen B dari Konveyor no.4 menuju AGV tipe dua no.A
8	Mengantarkan komponen B dari Konveyor no.3 menuju AGV tipe dua no.B
9	Mengantarkan komponen A dan B dari AGV tipe dua no.B menuju Konveyor no.4
10	Mengantarkan komponen A dari Konveyor no.4 menuju AGV tipe dua no.B

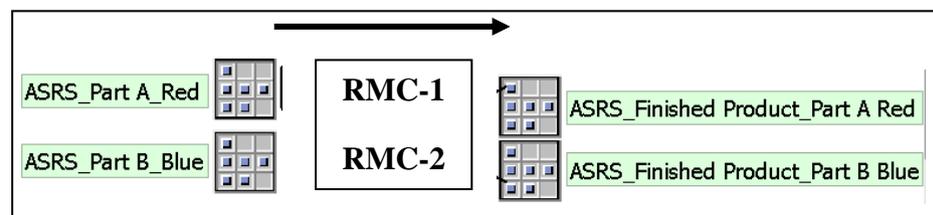


Gambar 3.19 Konveyor Otomatis

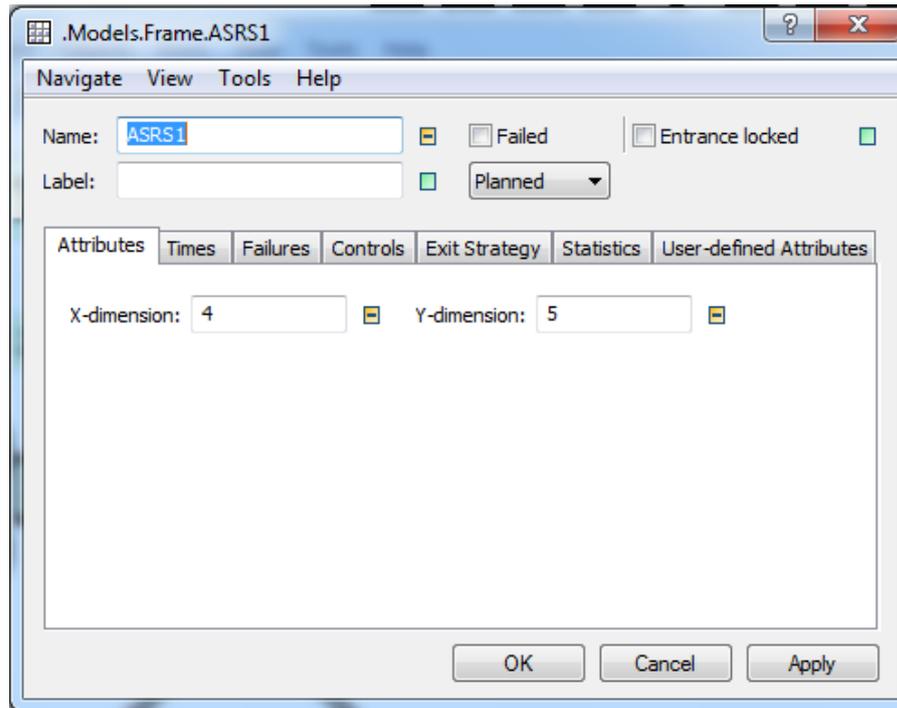
3.2.1.2 Kelas Sumber Daya (*Resources*)

Pada pembuatan model simulasi RMS ini, penulis menggunakan objek AS/RS (*Automated Storage & Retrieval System*), mesin CNC dan buffer pada kelas sumber daya ini.

AS/RS yang bisa dilihat pada Gambar 3.20 ini terbagi menjadi dua tipe. AS/RS tipe satu digunakan untuk mengeluarkan komponen A (berwarna merah) dan komponen B (berwarna biru). AS/RS tipe satu ini terletak dibagian paling awal dari *layout* model simulasi RMS ini. Sementara itu, AS/RS tipe dua berfungsi untuk menampung komponen A dan Komponen B yang telah selesai diproses dalam RMS ini. AS/RS tipe dua ini terletak dibagian paling akhir dari *layout* model simulasi RMS ini. AS/RS ini diatur untuk memiliki kapasitas 20 buah per AS/RS. Setiap AS/RS sudah bermuatan penuh, secara otomatis semua produk dalam AS/RS tersebut menghilang untuk merepresentasikan produk yang didistribusikan keluar pabrik.

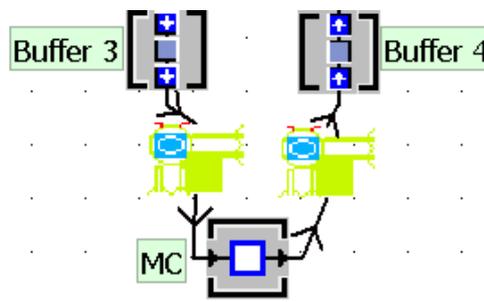


Gambar 3.20 AS/RS

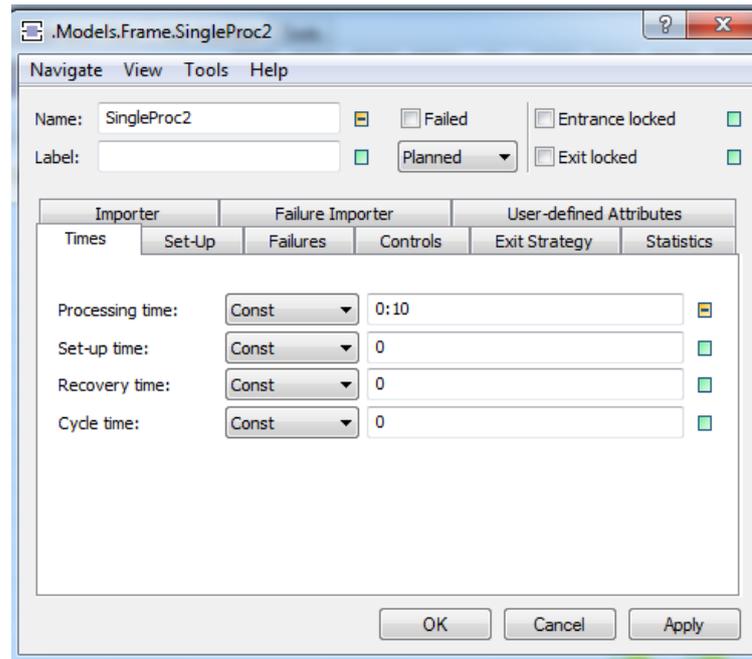


Gambar 3.21 Atribut AS/RS

Sementara itu, mesin CNC digunakan untuk merepresentasikan *Lathe Machine, Milling Machine & Machining Center (MC)* yang terdapat model RMS *Petri Net* Xiuli Meng (2010). Penulis memasukkan waktu proses sebesar 10 detik pada setiap mesin CNC. Gambar 3.22 adalah salah satu contoh mesin CNC pada *Machining Center (MC)*. Sementara itu, tampilan atribut mesin CNC dapat dilihat pada Gambar 3.23.

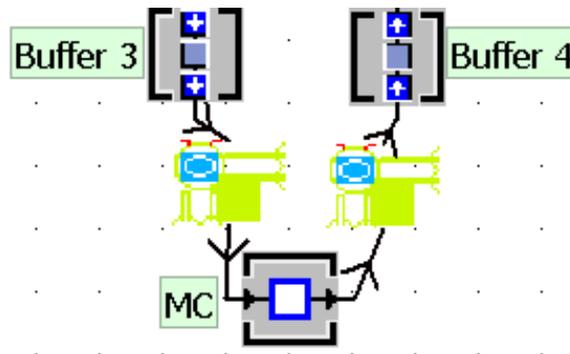


Gambar 3.22 Mesin CNC pada *Machining Center (MC)*

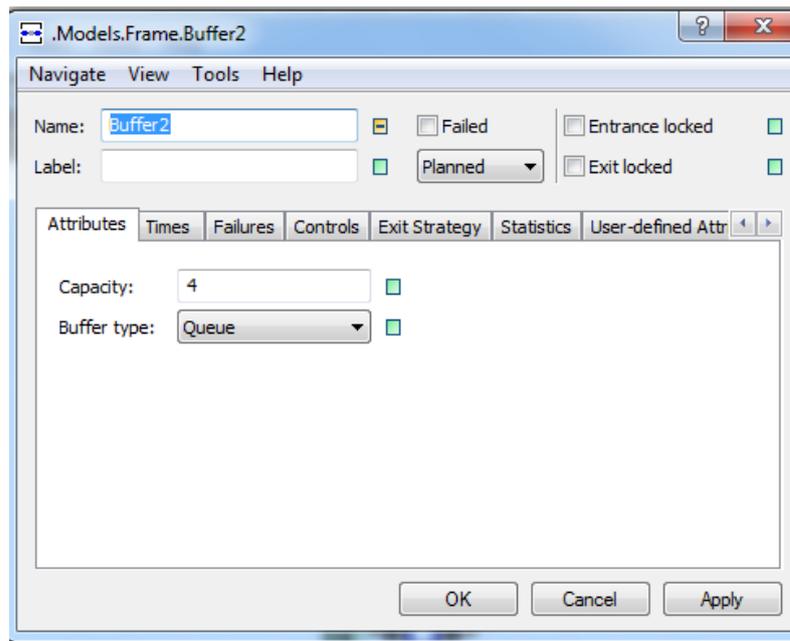


Gambar 3.23 Atribut Mesin CNC

Buffer digunakan sebagai penahan pada sistem RMS supaya memperlancar kinerja sistem. Kapasitas yang digunakan pada setiap buffer adalah 4 buah komponen. Buffer terdapat pada sel RMC1, RMC2 dan bagian inspeksi.



Gambar 3.24 Buffer pada Sel RMC2



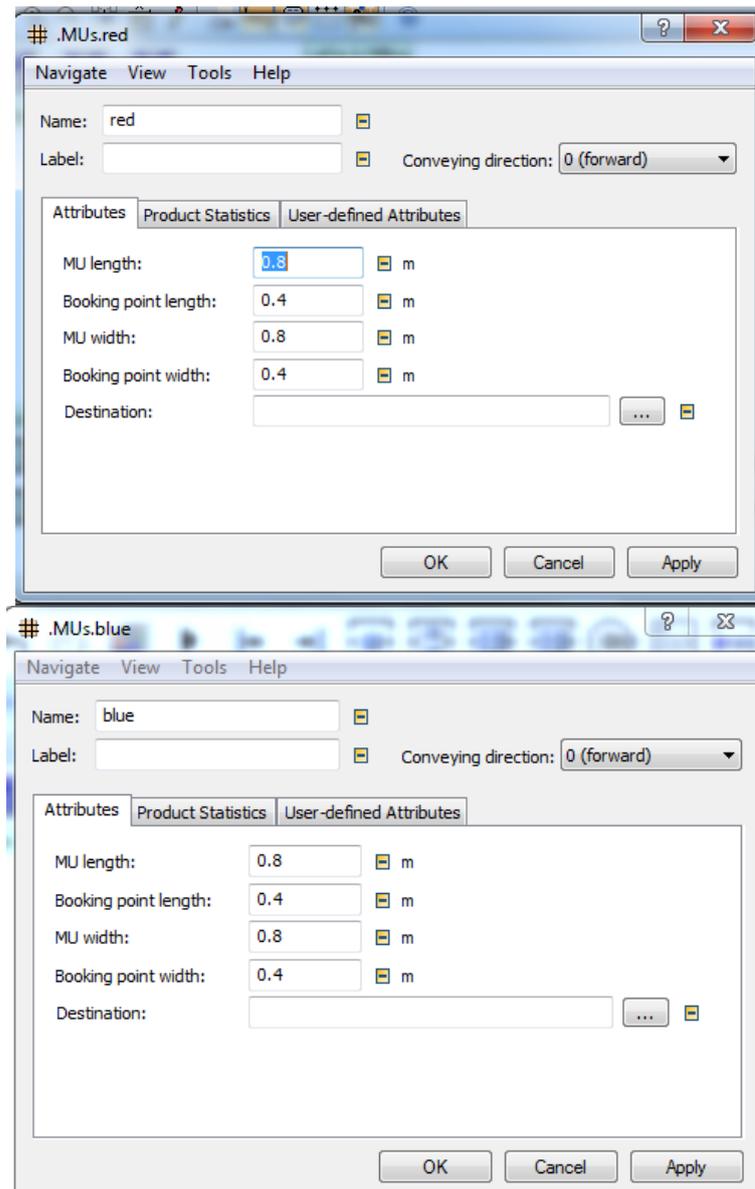
Gambar 3.25 Atribut Buffer

3.2.1.3 Kelas Objek Umum

Adapun objek-objek dalam kelas objek umum yang digunakan dalam pembuatan model simulasi RMS ini adalah satu buah *frame*, satu buah *event controller* dan beberapa *connector*. Satu buah *frame* digunakan sebagai tempat pembuatan model simulasi RMS. Satu buah *event controller* digunakan sebagai pengatur kejadian selama simulasi berjalan yang berfungsi untuk menyelaraskan semua kejadian dan proses yang terjadi dalam model simulasi RMS yang dibuat. Sementara itu, beberapa *connector* digunakan pada RMC-1 dan RMC-2. Pada RMC-1, *connector* digunakan untuk menghubungkan *buffer-robot-CNC Lathe/Milling Machine-robot-buffer*. Sementara itu pada RMC-2, *connector* digunakan untuk menghubungkan *buffer-Machining Center-buffer*.

3.2.1.4 Kelas Objek Berjalan

Objek pada kelas objek berjalan di dalam *plant simulation* biasa disebut sebagai *Mobile Units* (MUs). MUs yang digunakan penulis untuk membuat model simulasi RMS ini terdiri dari dua buah. MUs pertama merepresentasikan komponen A dan MUs kedua merepresentasikan komponen B. Gambar 3.26 berikut ini adalah atribut dari tiap MUs tersebut.



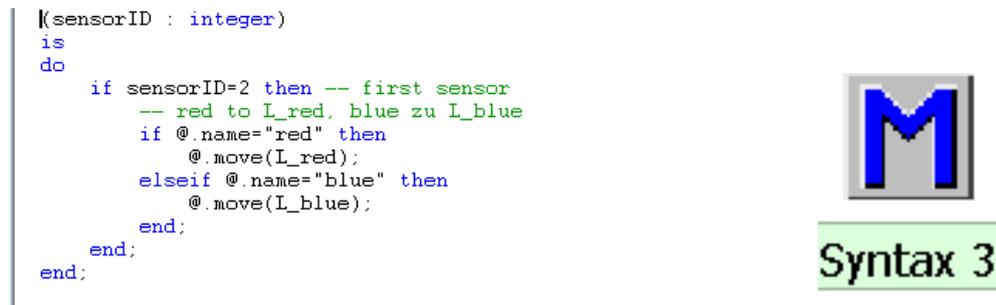
Gambar 3.26 Atribut MUs : Komponen A (merah) dan Komponen B (biru)

3.2.1.5 Kelas Daftar, Tabel dan Metode

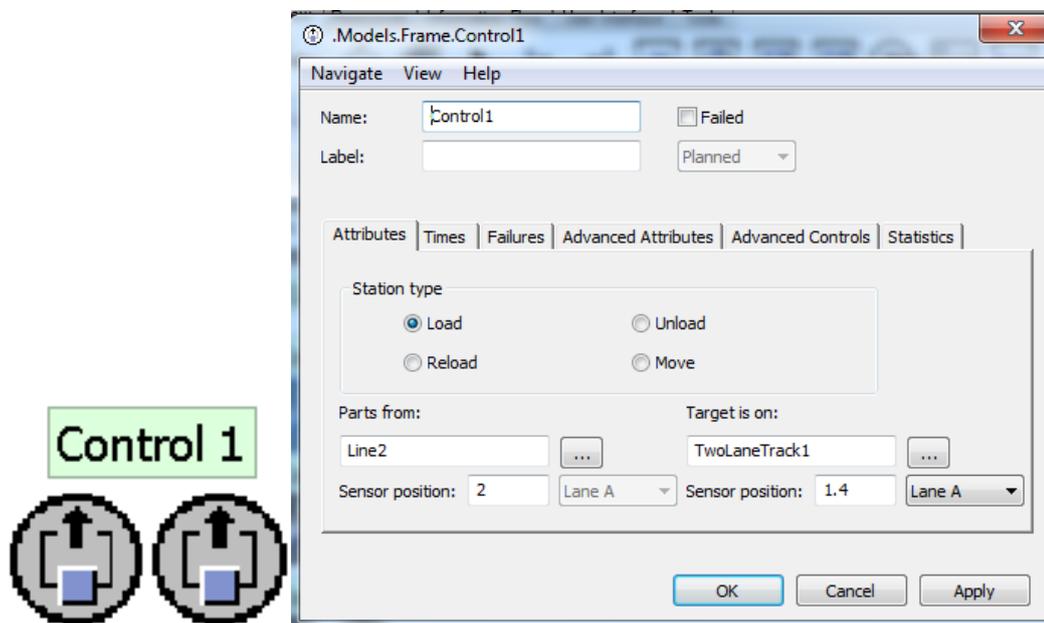
Pada pembuatan model simulasi RMS ini, penulis tidak menggunakan objek daftar atau tabel karena belum merasa diperlukan. Sementara itu, objek metode yang digunakan dalam model simulasi RMS ini terdiri dari dua objek. Objek pertama merupakan tempat untuk memasukkan bahasa pemrograman. Penulis menggunakan delapan buah objek metode dalam model simulasi RMS ini. Salah satu contoh metode berikut bahasa pemrogramannya dapat dilihat pada Gambar 3.27.

Sementara itu, objek kedua merupakan pusat kontrol *material handling* yang berfungsi untuk mengontrol seluruh AGV, Robot dan Konveyor yang digunakan dalam model simulasi RMS ini. Penulis menggunakan enam pasang pusat kontrol dalam model simulasi RMS ini. Salah satu contoh pusat kontrol dapat dilihat pada Gambar 3.28.

Perincian detail dari objek metode dan pusat kontrol dapat dilihat pada bagian lampiran.



Gambar 3.27 Contoh Metode beserta *Syntax*

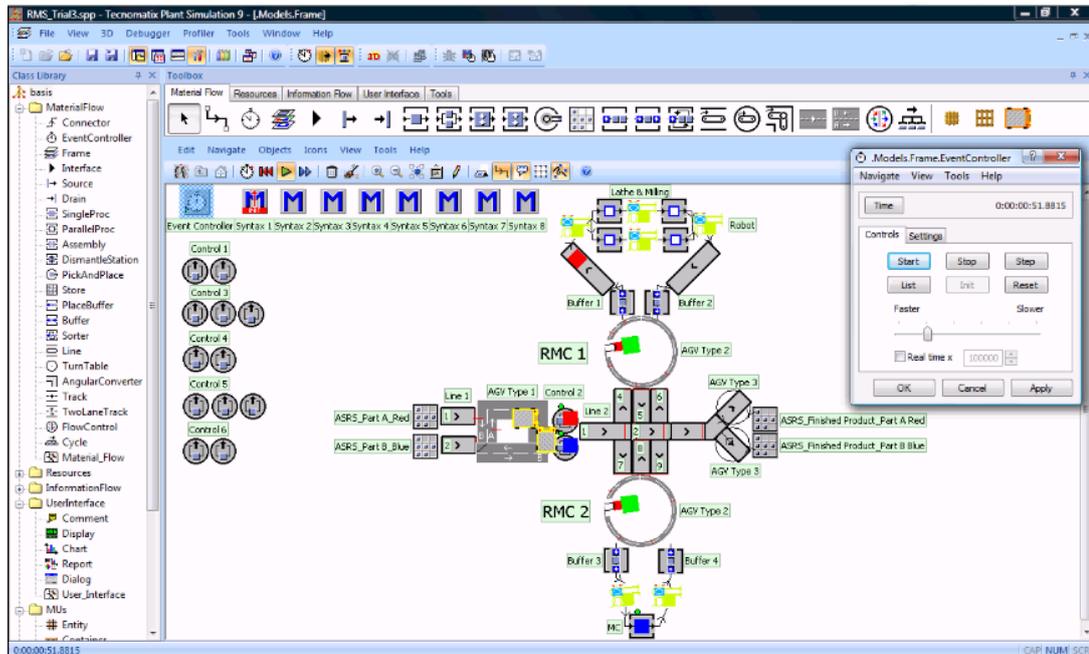


Gambar 3.28 Contoh Pusat Kontrol beserta Atribut

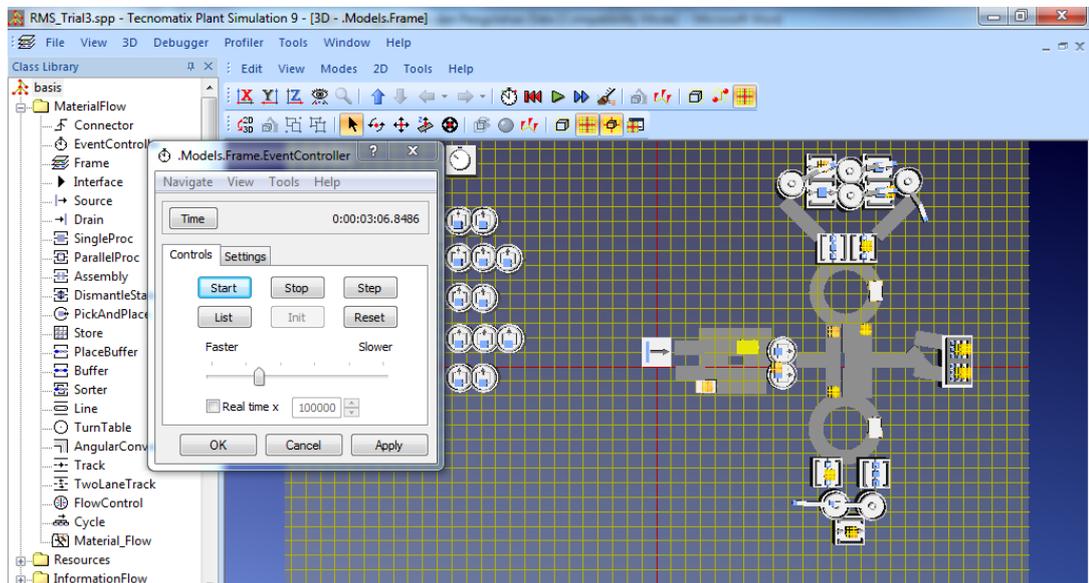
3.2.1.6 Kelas Objek Tampilan

Pada kelas objek tampilan ini, *plant simulation* secara otomatis telah menyediakan fasilitas visualisasi model simulasi dalam bentuk 2D dan 3D.

Penulis membangun dua buah konfigurasi model simulasi RMS. Model simulasi RMS konfigurasi pertama tanpa bagian inspeksi dapat dilihat pada Gambar 3.29. Sementara itu, model simulasi RMS dengan bagian inspeksi dapat dilihat pada Gambar 2.30.

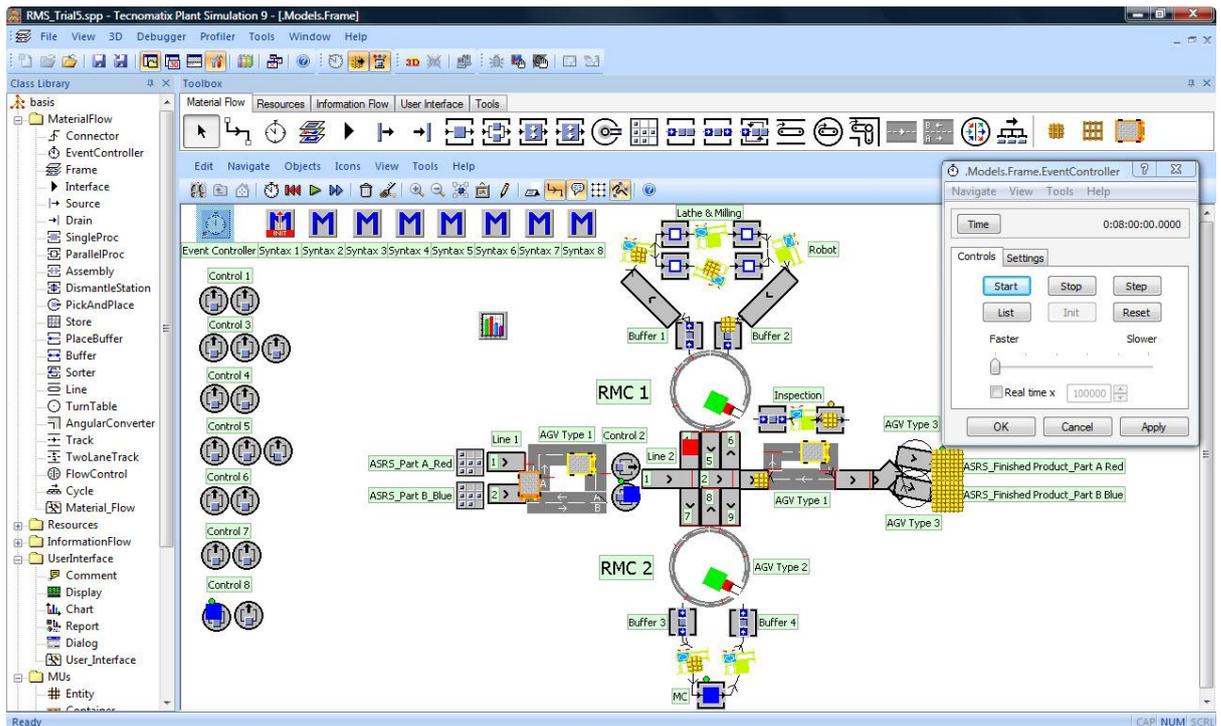


(a)

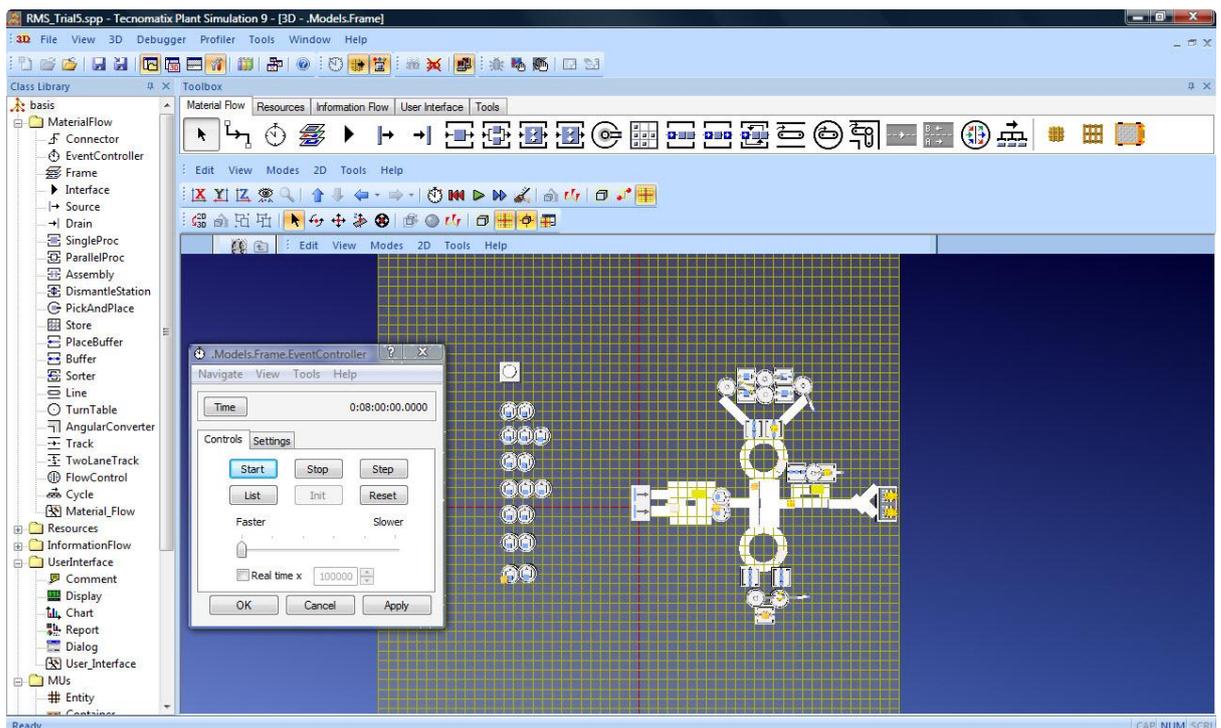


(b)

Gambar 3.29 Model Simulasi RMS Konfigurasi Pertama (tanpa Inspeksi): (a) dua dimensi; (b) tiga dimensi



(a)



(b)

Gambar 3.30 Model Simulasi RMS Konfigurasi Kedua (dengan Inspeksi): (a) dua dimensi; (b) tiga dimensi

3.2.2 Verifikasi dan Validasi Model Simulasi RMS

Schlesinger et al. (1995) mendefinisikan verifikasi sebagai proses untuk mengkonfirmasi konseptual model yang dibuat telah diterjemahkan dengan tepat dalam program komputer dan perhitungan yang dibuat dengan program ini telah menggunakan data input yang benar.

Menurut Jagdev et al. (1995), salah satu cara untuk melakukan verifikasi model adalah dengan mempelajari bahasa pemrogramannya, mengamati animasi terhadap tingkah laku yang benar, dan memeriksa keluaran yang logis. Jika model sudah dapat dijalankan dengan baik dan benar, maka model tersebut sudah dapat terverifikasi.

Tabel 3.7 Status Verifikasi Sub-proses pada Model Simulasi RMS Level Sistem

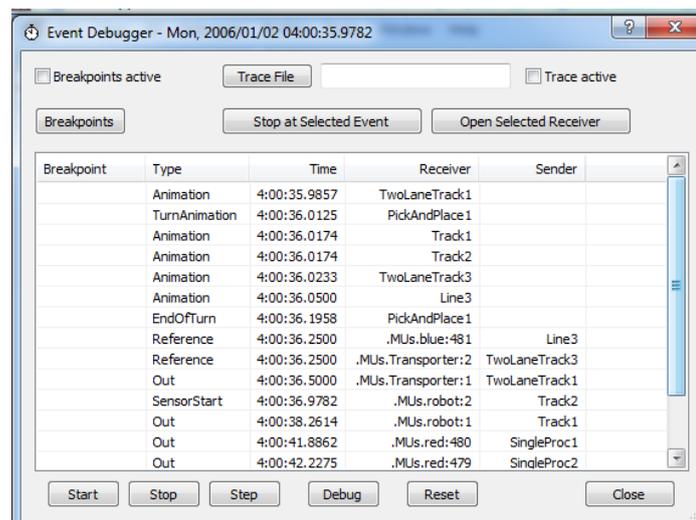
Komponen A				Komponen B			
Sub-proses	Warna	Sumber	Status	Sub-proses	Warna	Sumber	Status
Komponen menunggu di AS/RS	A1	AS/RS	Ok	Komponen menunggu di AS/RS	B1	AS/RS	Ok
AGV antar komponen ke RMC1	A2	AGV	Ok	AGV antar komponen ke RMC2	B2	AGV	Ok
RMC1 memproses komponen	A3	RMC1	Ok	RMC2 memproses komponen	B3	RMC2	Ok
AGV antar komponen ke RMC2	A4	AGV	Ok	AGV antar komponen ke RMC1	B4	AGV	Ok
RMC2 memproses komponen	A5	RMC2	Ok	RMC1 memproses komponen	B5	RMC1	Ok
AGV antar komponen ke AS/RS	A6	AGV	Ok	AGV antar komponen ke AS/RS	B6	AGV	Ok
Komponen tiba di AS/RS	A7	AS/RS	Ok	Komponen tiba di AS/RS	B7	AS/RS	Ok

Tabel 3.8 Status Verifikasi Sub-proses pada Model Simulasi RMS Level RMC1

Komponen A				Komponen B			
Sub-proses	Warna	Sumber	Status	Sub-proses	Warna	Sumber	Status
Komponen menunggu di Buffer	a1	Buffer	Ok	Komponen menunggu di Buffer	b1	Buffer	Ok
Pengaturan Lathe	a2	Lathe	Ok	Pengaturan Mill	b2	Mill	Ok
Robot muat komponen ke Lathe	a3	Robot	Ok	Robot muat komponen ke Mill	b3	Robot	Ok
Lathe memproses komponen	a4	Lathe	Ok	Mill memproses komponen	b4	Mill	Ok
Robot bongkar komponen dari Lathe	a5	Robot	Ok	Robot bongkar komponen dari Mill	b5	Robot	Ok
Komponen menunggu di Buffer	a6	Buffer	Ok	Komponen menunggu di Buffer	b6	Buffer	Ok
Pengaturan Mill	a7	Mill	Ok	Pengaturan Lathe	b7	Lathe	Ok
Robot muat komponen ke Mill	a8	Robot	Ok	Robot muat komponen ke Lathe	b8	Robot	Ok
Komponen menunggu di buffer	a9	Buffer	Ok	Komponen menunggu di buffer	b9	Buffer	Ok

Tabel 3.9 Status Verifikasi Sub-proses pada Model Simulasi RMS Level RMC2

Komponen A				Komponen B			
Sub-proses	Warna	Sumber	Status	Sub-proses	Warna	Sumber	Status
Komponen menunggu di Buffer	a1	Buffer	Ok	Komponen menunggu di Buffer	b1	Buffer	Ok
Pengaturan Lathe	a2	Lathe	Ok	Pengaturan Mill	b2	Mill	Ok
Robot muat komponen ke Lathe	a3	Robot	Ok	Robot muat komponen ke Mill	b3	Robot	Ok
Lathe memproses komponen	a4	Lathe	Ok	Mill memproses komponen	b4	Mill	Ok
Robot bongkar komponen dari Lathe	a5	Robot	Ok	Robot bongkar komponen dari Mill	b5	Robot	Ok
Komponen menunggu di Buffer	a6	Buffer	Ok	Komponen menunggu di Buffer	b6	Buffer	Ok
Pengaturan Mill	a7	Mill	Ok	Pengaturan Lathe	b7	Lathe	Ok
Robot muat komponen ke Mill	a8	Robot	Ok	Robot muat komponen ke Lathe	b8	Robot	Ok
Komponen menunggu di buffer	a9	Buffer	Ok	Komponen menunggu di buffer	b9	Buffer	Ok

**Gambar 3.31** Status Verifikasi *Event Debugger* pada Model Simulasi RMS

Penulis sudah memeriksa *event debugger* pada model simulasi RMS sebagai bagian akhir dari proses verifikasi. Seperti yang terlihat pada Gambar 3.31. Semua proses yang terjadi saat model simulasi RMS dijalankan (*running*) sudah berjalan dengan baik sesuai dengan yang diinginkan oleh penulis

Sementara itu, validasi model simulasi dilakukan dengan meninjau mekanisme logika model dan membandingkannya dengan sistem nyata. Akan tetapi, karena model simulasi RMS ini merupakan suatu model terhadap suatu sistem yang umum dan bukan dari kondisi yang nyata, tidak terdapat data keadaan aktual yang dapat dijadikan acuan untuk validasi.

BAB 4 ANALISIS MODEL

4. ANALISIS MODEL

Salah satu cara yang paling efektif untuk mendapatkan pemahaman dan gambaran yang lebih baik mengenai karakteristik dan perilaku RMS adalah dengan memperoleh model simulasi RMS. Tujuan dari penelitian ini adalah untuk memperoleh model simulasi RMS dengan menggunakan metode simulasi berorientasi objek untuk mengetahui karakteristik dan perilaku RMS secara lebih baik yang dilihat berdasarkan hasil produksi dan penggunaan waktu produksi. Oleh karena itu, pada bab ini akan dibahas analisis terhadap model simulasi RMS untuk mencapai tujuan penelitian tersebut.

Sebagaimana yang sudah dijelaskan pada bab dua, RMS memiliki enam karakteristik. Karena keterbatasan penelitian, tidak semua karakteristik RMS tersebut akan dianalisis. Adapun karakteristik RMS yang akan dianalisis, antara lain: *scalability*, *convertibility* dan *integrability*.

Secara umum, analisis akan dilakukan terhadap model simulasi RMS berjumlah dua konfigurasi, yaitu (1) konfigurasi model RMS tanpa inspeksi dan (2) konfigurasi model dengan inspeksi. Setiap simulasi akan memiliki skenario yang berbeda dan dijalankan selama jangka waktu satu *shift* normal atau delapan jam.

Berdasarkan hasil pengamatan selama menjalankan simulasi, penulis mengidentifikasi beberapa perilaku RMS, antara lain:

- Sistem pada model simulasi RMS dikendalikan secara otomatis melalui pusat kontrol. Pusat kontrol ini mengendalikan sub-sistem *material-handling*, sub-sistem permesinan dan sub-sistem penyimpanan produk. Oleh karena itu, dalam model simulasi RMS ini, tidak ada peran manusia yang terlibat secara langsung dalam produksi. Peran manusia hanya dilibatkan secara tidak langsung, seperti dalam hal pengaturan pusat kontrol, pembuatan *master production scheduling* (MPS), pemeliharaan, dll.
- Sub-sistem *material-handling* ditangani oleh alat kerja otomatis, seperti AGV, konveyor otomatis dan robot. Alat kerja otomatis ini memiliki tingkat

inteligensi yang tinggi yang dapat dilihat dari kemampuannya untuk mengidentifikasi suatu material yang akan diantar menuju tempat tujuan yang benar tanpa dikendalikan oleh bantuan manusia. Selain itu, alat kerja otomatis ini baru bekerja ketika memang dibutuhkan untuk bekerja. Hal ini tentu menguntungkan karena dapat meningkatkan efisiensi sistem secara signifikan.

- Sub-sistem permesinan mampu melakukan beberapa pekerjaan sekaligus terhadap produk yang masih tergolong dalam satu *part family*. Hal ini dapat dilakukan karena pada konsep RMS, setiap komponen dibuat secara modular sehingga proses kerja dapat dilakukan terhadap semua komponen yang memiliki kemiripan bentuk dan fitur geometris dalam satu mesin. Tentu saja, hal ini menguntungkan karena dapat menghemat waktu dan biaya produksi secara signifikan.
- Sub-sistem penyimpanan produk memiliki kemampuan otomatis (tanpa bantuan manusia) untuk mengidentifikasi produk apa disimpan dalam pos penyimpanan produk yang mana. Hal ini memudahkan pengantaran produk dengan jenis yang beragam menuju tempat distribusi atau konsumen/pelanggan sesuai dengan pesanan.

4.1 Analisis Karakteristik *Scalability*

Scalability berarti sistem mampu menyesuaikan skala kapasitas produksi sesuai kebutuhan. Untuk mengetahui karakteristik *scalability* maka analisis akan dilakukan terhadap model simulasi RMS yang akan dijalankan dengan skenario-skenario berdasarkan perbedaan waktu interval kedatangan komponen. Detail skenario ini dapat dilihat pada Tabel 4.1.

Skenario perbedaan waktu interval kedatangan komponen ini merepresentasikan perubahan kapasitas produksi yang terjadi dalam sistem model simulasi RMS. Oleh karena itu, pelibatan skenario ini dalam proses simulasi diharapkan mampu menggambarkan karakteristik *scalability* yang terdapat pada model simulasi RMS yang telah penulis buat.

Tabel 4.1 Skenario Waktu Interval Kedatangan Komponen

No. Skenario	Waktu Interval Kedatangan Komponen (detik)
1.	5
2.	10
3.	15
4.	20
5.	25
6.	30
7.	35
8.	40

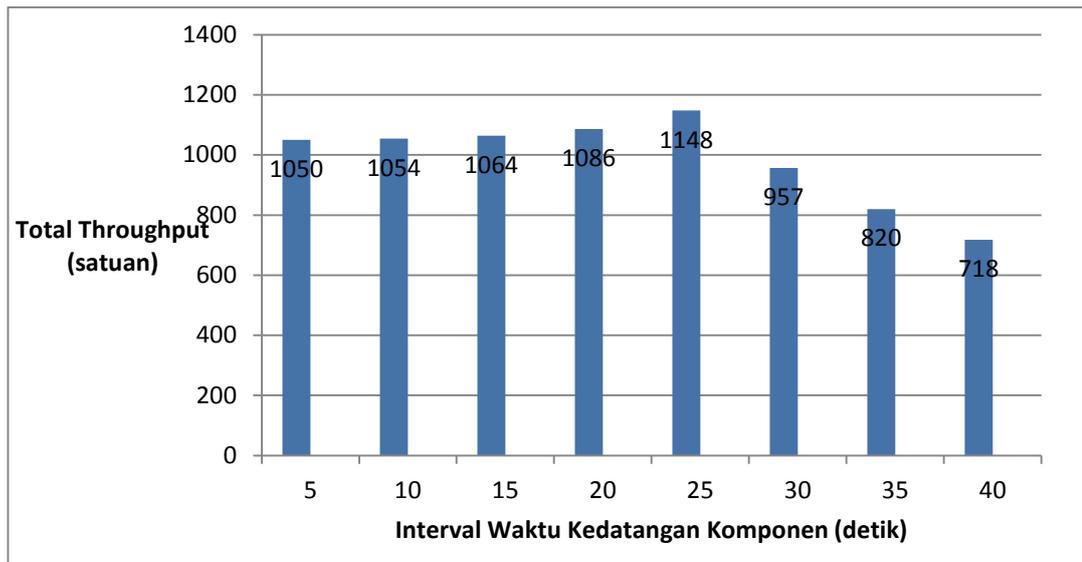
Penulis telah melakukan simulasi dan didapat hasil simulasi yang dapat dilihat pada Tabel 4.2 dan Tabel 4.3.

Tabel 4.2 Hasil Simulasi Tingkat Okupasi pada Skenario Perbedaan Waktu Interval Kedatangan Komponen (Model Simulasi RMS tanpa Inspeksi)

No.	Waktu Interval Kedatangan (detik)	Okupasi (%)							
		AGV1	AGV2	AGV3	Konveyor	Robot	Lathe	Milling	MC
1.	5	12.39	35.56	5.32	14.79	85.33	99.68	97.91	99.93
2.	10	12.48	35.60	5.34	14.91	85.40	99.67	97.91	99.93
3.	15	12.55	35.59	5.39	15.01	85.16	99.65	97.88	99.90
4.	20	12.83	35.66	5.50	15.23	84.97	99.65	97.88	99.86
5.	25	13.50	34.23	5.82	21.75	52.66	94.54	92.87	94.75
6.	30	11.25	28.52	4.85	18.05	22.17	78.78	77.40	78.95
7.	35	9.64	24.45	4.15	15.47	25.03	67.51	66.34	67.69
8.	40	8.44	21.40	3.64	14.22	16.12	59.13	58.06	59.23

Tabel 4.3 Hasil Simulasi Tingkat *Throughput* pada Skenario Perbedaan Waktu Interval Kedatangan Komponen (Model Simulasi RMS tanpa Inspeksi)

No.	Waktu Interval Kedatangan (detik)	Average Lifespan (menit: detik)	Average Exit Interval (detik)	Total Throughput (satuan)	Throughput per hour (satuan)
1.	5	6:04	27.30	1050	131.67
2.	10	6:08	27.24	1054	131.17
3.	15	6:01	26.99	1064	134.17
4.	20	5:49	26.40	1086	134.83
5.	25	1:49	25.00	1148	144.00
6.	30	1:43	30.00	957	120.00
7.	35	1:43	35.01	820	102.83
8.	40	1:44	40.00	718	90.00



Gambar 4.1 Hasil Simulasi Tingkat *Throughput* pada Skenario Perbedaan Waktu Interval Kedatangan Komponen (Model Simulasi RMS tanpa Inspeksi)

Pada Gambar 4.1 jelas terlihat bahwa meskipun waktu interval kedatangan komponen dilakukan perubahan, sistem model simulasi RMS masih mampu menghasilkan keluaran produk (*throughput*) yang cukup stabil pada interval waktu kedatangan 5 detik per komponen hingga interval waktu kedatangan 25 detik per komponen. Barulah setelah interval waktu kedatangan diubah menjadi diatas 30 detik per komponen, keluaran produk (*throughput*) sudah mengalami penurunan yang dramatis. Hal ini mengindikasikan bahwa sistem model simulasi RMS yang dibuat memiliki tingkat *scalability* hingga rentang interval waktu kedatangan komponen sebesar 20 detik, yaitu dari 5 detik per komponen hingga 25 detik per komponen.

4.2 Analisis Karakteristik *Convertibility*

Convertibility mengizinkan perubahan (*changeover*) secara cepat antara produk yang sedang eksis dengan tingkat adaptasi sistem yang cepat terhadap produk masa depan. Untuk mengetahui karakteristik *convertibility* maka analisis akan dilakukan terhadap model simulasi RMS yang akan dijalankan dengan skenario-skenario berdasarkan perubahan waktu proses pada *lathe machine*, *milling machine* dan *machining center* (MC). Detail skenario ini dapat dilihat pada Tabel 4.4.

Skenario perbedaan waktu proses pada permesinan ini merepresentasikan perubahan proses produksi yang terjadi dalam sistem model simulasi RMS seiring dengan perubahan produk yang diinginkan pelanggan di masa depan. Oleh karena itu, pelibatan skenario ini dalam proses simulasi diharapkan mampu menggambarkan karakteristik *convertibility* yang terdapat pada model simulasi RMS yang telah penulis buat.

Tabel 4.4 Skenario Waktu Proses pada Permesinan

No. Skenario	Waktu Proses (detik)		
	Lathe	Milling	MC
1.	10	10	10
2.	15	15	15
3.	20	20	20
4.	15	10	15
5.	20	15	20
6.	10	10	15
7.	15	15	20
8.	10	15	15

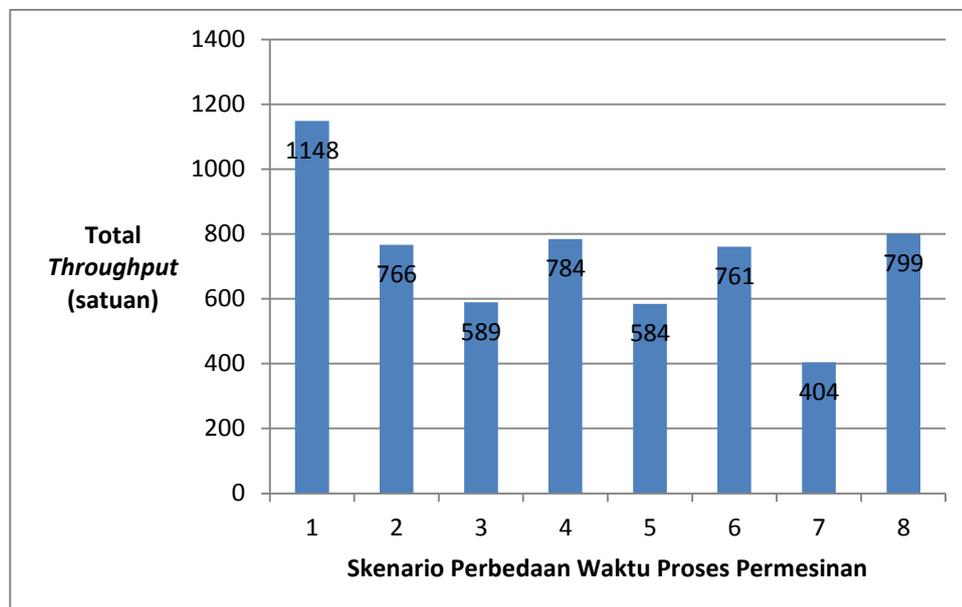
Penulis telah melakukan simulasi dan didapat hasil simulasi yang dapat dilihat pada Tabel 4.5 dan Tabel 4.6.

Tabel 4.5 Hasil Simulasi Tingkat Okupasi pada Skenario Perbedaan Waktu Proses (Model Simulasi RMS tanpa Inspeksi)

No.	Waktu Proses (detik)			Okupasi (%)		
	Lathe	Milling	MC	Lathe	Milling	MC
1.	10	10	10	79.90	79.86	79.87
2.	15	15	15	88.85	88.80	88.84
3.	20	20	20	91.39	91.32	91.39
4.	15	10	15	88.82	59.18	88.84
5.	20	15	20	91.40	68.49	91.40
6.	10	10	15	59.32	59.31	88.84
7.	15	15	20	68.56	68.54	91.40
8.	10	15	15	59.13	88.86	88.84

Tabel 4.6 Hasil Simulasi Tingkat *Throughput* pada Skenario Perbedaan Waktu Proses (Model Simulasi RMS tanpa Inspeksi)

No.	Waktu Proses (detik)			<i>Average Lifespan</i> (menit: detik)	<i>Average Exit Interval</i> (detik)	<i>Total Throughput</i> (satuan)	<i>Throughput per hour</i> (satuan)
	Lathe	Milling	MC				
1.	10	10	10	1:49	25.00	1148	144.00
2.	15	15	15	8:21	37.47	766	96.5.0
3.	20	20	20	10:51	48.70	589	74.17
4.	15	10	15	8:01	36.61	784	97.67
5.	20	15	20	10:38	49.15	584	72.33
6.	10	10	15	7:06	37.60	761	95.00
7.	15	15	20	10:44	1:11	404	49.33
8.	10	15	15	7:39	35.88	799	99.67



Gambar 4.2 Hasil Simulasi Tingkat *Throughput* pada Skenario Perbedaan Waktu Proses (Model Simulasi RMS tanpa Inspeksi)

Secara umum, setelah dilakukan simulasi sebanyak delapan skenario perbedaan waktu proses pada mesin Lathe, Milling dan MC dapat diketahui bahwa sistem model simulasi RMS dapat beradaptasi karena semua simulasi dapat berjalan hingga selesai (1 *shift*) dan semua simulasi sebanyak delapan skenario menghasilkan *throughput*. Perbedaan jumlah *throughput* dari setiap skenario simulasi mengindikasikan bahwa telah terjadi proses adaptasi sistem mengikuti perubahan proses waktu produksi untuk tetap mampu menjalankan proses produksi sebagaimana mestinya. Hal ini berarti model simulasi RMS yang dibuat mampu menggambarkan karakteristik *convertibility*.

4.3 Analisis Karakteristik *Integrability*

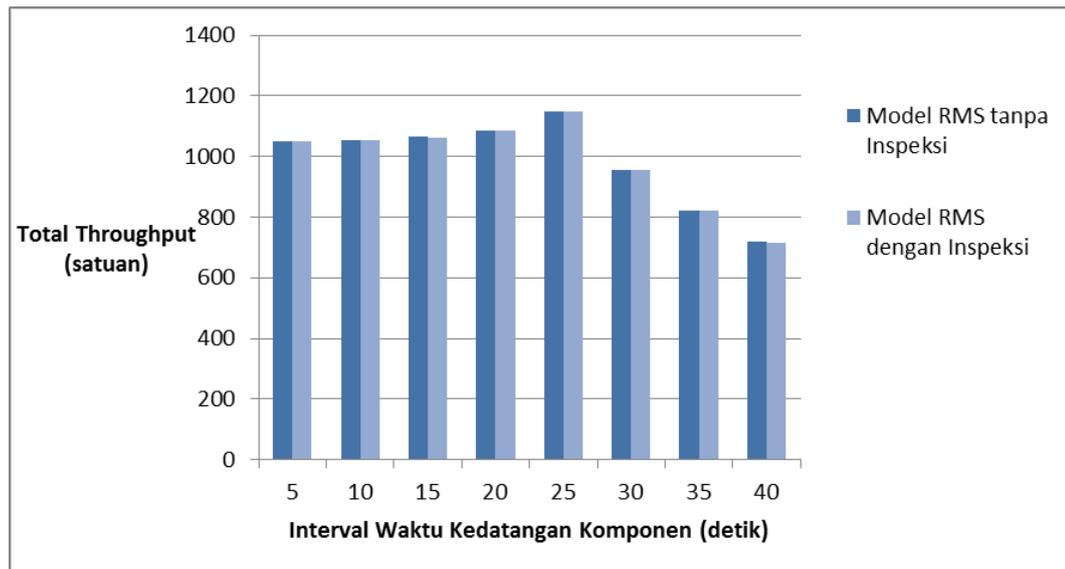
Integrability berarti sistem dan komponen sistem didesain untuk sistem terintegrasi dan mampu menerima teknologi baru di masa depan. Untuk mengetahui karakteristik *integrability* maka analisis akan dilakukan terhadap model simulasi RMS sejumlah dua rekonfigurasi model RMS, yaitu (1) konfigurasi model RMS tanpa inspeksi dan (2) konfigurasi model RMS dengan inspeksi. Hasil Simulasi ini dapat dilihat pada Tabel 4.7, Tabel 4.8, Tabel 4.9 dan Tabel 4.10.

Tabel 4.7 Hasil Simulasi Tingkat Okupasi pada Skenario Perbedaan Waktu Interval Kedatangan Komponen
(Model RMS tanpa Inspeksi vs Model RMS dengan Inspeksi)

No.	Waktu Interval Kedatangan (detik)	Okupasi (%)						
		Model RMS tanpa Inspeksi			Model RMS dengan Inspeksi			
		Lathe	Milling	MC	Lathe	Milling	MC	Inspection
1.	5	99.68	97.91	99.93	99.68	97.91	99.93	16.33
2.	10	99.67	97.91	99.93	99.67	97.91	99.93	16.31
3.	15	99.65	97.88	99.90	99.65	97.88	99.90	16.33
4.	20	99.65	97.88	99.86	99.65	97.88	99.86	16.32
5.	25	94.54	92.87	94.75	94.54	92.87	94.75	15.50
6.	30	78.78	77.40	78.95	78.78	77.40	78.95	12.92
7.	35	67.51	66.34	67.69	67.51	66.34	67.69	11.07
8.	40	59.13	58.06	59.23	59.13	58.06	59.23	9.69

Tabel 4.8 Hasil Simulasi Tingkat *Throughput* pada Skenario Perbedaan Waktu Interval Kedatangan Komponen
(Model RMS tanpa Inspeksi vs Model RMS dengan Inspeksi)

No.	Waktu Interval Kedatangan (detik)	Total <i>Throughput</i> (satuan)	
		Model RMS tanpa Inspeksi	Model RMS dengan Inspeksi
1.	5	1050	1049
2.	10	1054	1052
3.	15	1064	1063
4.	20	1086	1086
5.	25	1148	1147
6.	30	957	957
7.	35	820	820
8.	40	718	717



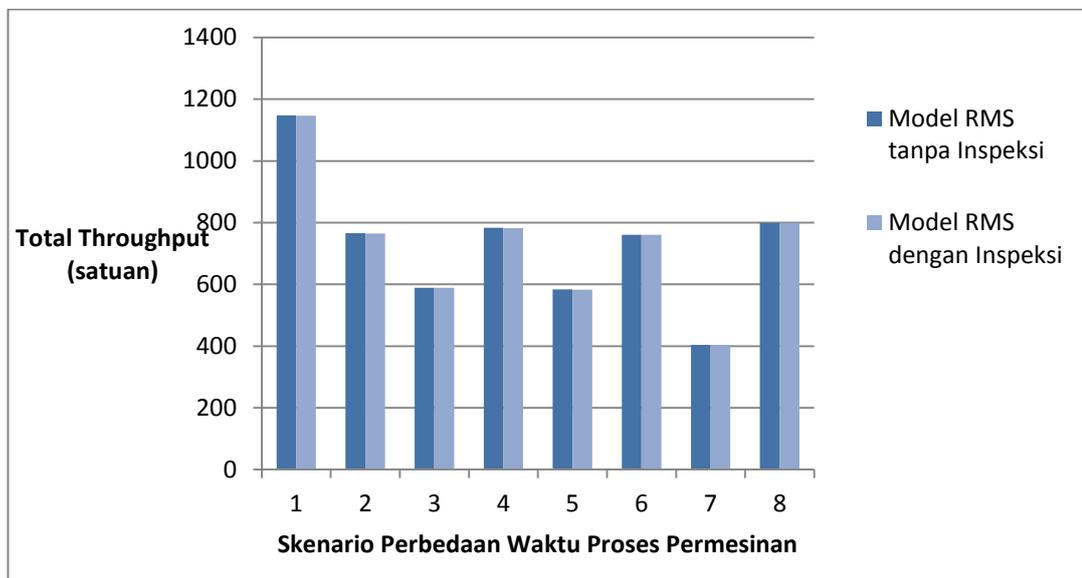
Gambar 4.3 Hasil Simulasi Tingkat *Throughput* pada Skenario Perbedaan Waktu Interval Kedatangan Komponen (Model RMS tanpa Inspeksi vs Model RMS dengan Inspeksi)

Tabel 4.9 Hasil Simulasi Tingkat Okupasi pada Skenario Perbedaan Waktu Proses (Model RMS tanpa Inspeksi vs Model RMS dengan Inspeksi)

No.	Waktu Proses (detik)			Okupasi (%)						
				Model RMS tanpa Inspeksi			Model RMS dengan Inspeksi			
	Lathe	Milling	MC	Lathe	Milling	MC	Lathe	Milling	MC	Inspection
1.	10	10	10	79.90	79.86	79.87	79.90	79.86	79.87	16.33
2.	15	15	15	88.85	88.80	88.84	88.85	88.80	88.84	16.31
3.	20	20	20	91.39	91.32	91.39	91.39	91.32	91.39	16.33
4.	15	10	15	88.82	59.18	88.84	88.82	59.18	88.84	16.32
5.	20	15	20	91.40	68.49	91.40	91.40	68.49	91.40	15.50
6.	10	10	15	59.32	59.31	88.84	59.32	59.31	88.84	12.92
7.	15	15	20	68.56	68.54	91.40	68.56	68.54	91.40	11.07
8.	10	15	15	59.13	88.86	88.84	59.13	88.86	88.84	9.69

Tabel 4.10 Hasil Simulasi Tingkat *Throughput* pada Skenario Perbedaan Waktu Proses
(Model RMS tanpa Inspeksi vs Model RMS dengan Inspeksi)

No.	Waktu Proses (detik)			Total <i>Throughput</i> (satuan)	
	Lathe	Milling	MC	Model RMS tanpa Inspeksi	Model RMS dengan Inspeksi
1.	10	10	10	1148	1147
2.	15	15	15	766	765
3.	20	20	20	589	589
4.	15	10	15	784	783
5.	20	15	20	584	583
6.	10	10	15	761	761
7.	15	15	20	404	404
8.	10	15	15	799	799



Gambar 4.4 Hasil Simulasi Tingkat *Throughput* pada Skenario Perbedaan Waktu Proses
(Model RMS tanpa Inspeksi vs Model RMS dengan Inspeksi)

Pada Gambar 4.3 dan Gambar 4.4 dapat dilihat bahwa meskipun sudah dilakukan penambahan pos baru pada sistem yaitu pos inspeksi, sistem model simulasi RMS tetap mampu menghasilkan keluaran produk (*throughput*) yang kurang lebih sama. Hal ini mengindikasikan bahwa model simulasi RMS yang dibuat sudah mampu menggambarkan karakteristik *integrability*.

BAB 5

KESIMPULAN DAN SARAN

5. Kesimpulan dan Saran

Pada bab ini berisi tentang kesimpulan terhadap hasil pembuatan model simulasi RMS beserta analisisnya. Selanjutnya adanya saran-saran untuk penelitian lebih lanjut bagi akademisi atau praktisi yang ingin melakukan atau melanjutkan penelitian yang berkaitan dengan *reconfigurable manufacturing system* (RMS).

5.1 Kesimpulan

Dari hasil penelitian skripsi “Pembuatan Model Simulasi *Reconfigurable Manufacturing System* Menggunakan Metode Simulasi Berorientasi Objek” ini, dapat diambil beberapa kesimpulan sebagai berikut.

1. Model simulasi RMS yang dibuat memiliki tingkat *scalability* hingga rentang perbedaan interval waktu kedatangan komponen sebesar 20 detik, yaitu dari 5 detik per komponen hingga 25 detik per komponen.
2. Model simulasi RMS yang dibuat mampu menggambarkan karakteristik *convertibility* yang dilihat dari kemampuannya untuk melakukan proses adaptasi sistem mengikuti perubahan proses produksi yang terjadi.
3. Model simulasi RMS yang dibuat sudah mampu menggambarkan karakteristik *integrability* karena meski sudah dilakukan penambahan pos baru pada sistem yaitu pos inspeksi, sistem model simulasi RMS tetap mampu menghasilkan keluaran produk (*throughput*) yang sama.

5.2 Saran

Penelitian skripsi “Pembuatan Model Simulasi *Reconfigurable Manufacturing System* Menggunakan Metode Simulasi Berorientasi Objek” ini masih memiliki keterbatasan karena belum dilakukan uji validasi. Oleh karena itu, kepada peneliti di masa depan yang tertarik dengan tema *reconfigurable manufacturing system* (RMS) dapat melakukan uji validasi terhadap model simulasi RMS ini sehingga dapat dilakukan pembuktian terhadap temuan-temuan penelitian yang penulis dapatkan dari hasil simulasi RMS menggunakan model simulasi RMS

ini. Selain itu, dapat pula dilakukan pengembangan model simulasi RMS ini sehingga karakteristik RMS lain yang belum terlibat dalam penelitian ini, seperti *modularity*, *customization* dan *diagnosability* dapat diwujudkan.

DAFTAR REFERENSI

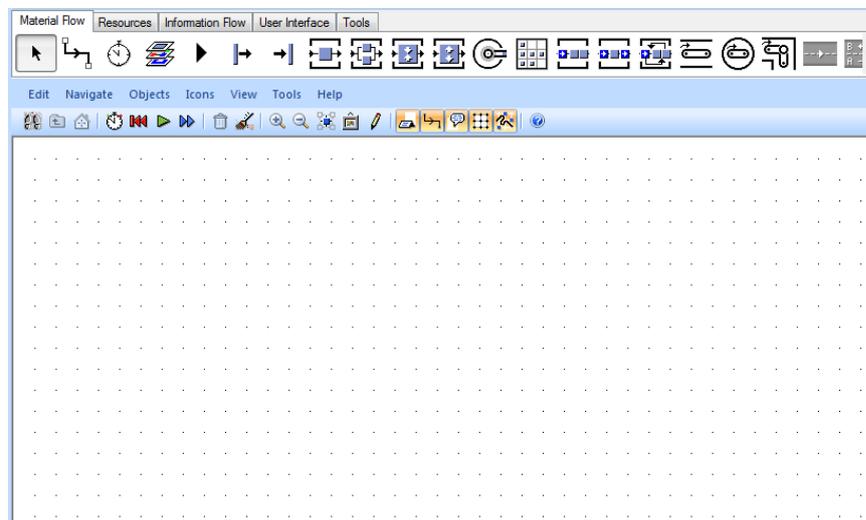
- Bangsow, Steffen. (2010).“*Manufacturing Simulation with Plant Simulation and SimTalk Usage and Programming with Examples and Solutions*”, ISBN 978-3-642-05073-2. Springer-Verlag Berlin Heidelberg.
- Bi, Z. et al. (2007).“*Reconfigurable manufacturing systems: the state of the art*”. International Journal of Production Research, 46, (4), pp. 967-992.
- Dashchenko, Anatoli I. (2006).“*Reconfigurable Manufacturing Systems and Transformable Factories*”. Springer-Verlag Berlin Heidelberg : New York, USA.
- E. Abele et al.(2004).“*Globalization and Decentralization of Manufacturing*”. International Journal of Manufacturing System, 20, (2), pp.234-241.
- Jagdev et al.(1995).” *Verification and validation issues in manufacturing models*”. International Journal of Computers in Industry, Vol.25, pp.331-353
- Jeffery A. Joines, Stephen D. Roberts. (1998).”*Fundamentals of Object-Oriented Simulation*”. Proceedings of the 1998 Winter Simulation Conference. Department of Industrial Engineering Campus Box 7906 North Carolina State University Raleigh, NC 27695-7906, USA.
- Koren, Y., Shpitalni, M.(2011).”*Design of reconfigurable manufacturing systems*”. Journal of Manufacturing System, Vol.29, pp. 130-141.
- Koren, Y., Jovane, F., Heisel, U., Moriwaki,, T., Pritschow G., Ulsoy G., and VanBrussel H. (1999).”*Reconfigurable Manufacturing Systems. A Keynote paper*”. CIRP Annals, Vol. 48, No. 2, pp. 6-12.
- Koren, Y. (2006).”*General RMS Characteristics—Comparison with Dedicated and Flexible Systems*”. Springer-Verlag Berlin Heidelberg : New York, USA.
- Preez, Jacques du (2011).“*A study of Reconfigurable Manufacturing Systems with Computer Simulation*”. Thesis for the Degree Master of Science in the Faculty of Industrial Engineering at Stellenbosch University : South Africa.
- Rothenberg, Jeff (1989).”*Object-Oriented Simulation : Where do we go from here?*”. Rand corporation : Santa Monica, USA.

- Wainer, G.A, dan Mosterman, P.J (2011).”*Discrete-Event Modeling and Simulation*”. CRC Press Taylor & Francis Group : New York, USA.
- Xiuli Meng. (2010). “*Modeling of reconfigurable manufacturing systems based on colored timed object-oriented petri nets*”. *Journal of Manufacturing Systems* 29 81-90.

Lampiran A : Penggunaan *Software*

Berikut ini adalah langkah-langkah penggunaan *software plant simulation* untuk membuat model simulasi RMS.

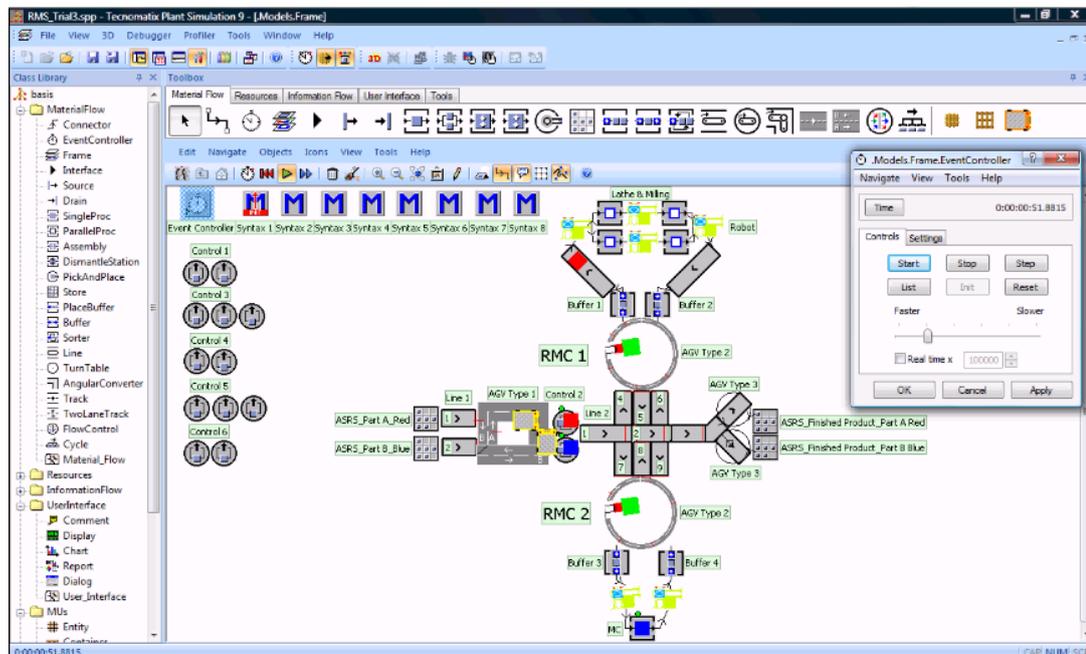
1. Mempelajari terlebih dahulu model RMS *petri net* Xiuli Meng (2010) sebagai data pembangun model simulasi RMS.
2. Selanjutnya membuat model simulasi RMS dengan merujuk data model RMS *Petri Net* Xiuli Meng (2010). Tahap paling awal adalah melakukan identifikasi objek-objek apa saja yang diperlukan untuk membangun model simulasi RMS ini, misalnya *entity*, CNC (Lathe, Milling & MC), *resources*, *line*, *track*, AGV, Robot, Buffer, ASRS dan pusat kontrol.
3. Setelah selesai membuat konsepnya itu kemudian kita memulai dalam pengerjaannya ke dalam *software*. Pertama-tama buka *software plant simulation* dengan meng-klik *Tecnomatix Plant Simulation 9.0* pada *desktop computer*. Kemudian pilih *create new model*. Setelah itu akan muncul *new frame* sebagai berikut:



4. Kemudian kita memasukkan seluruh objek yang sebelumnya sudah diidentifikasi pada langkah no.2 ke dalam *new frame*. Perincian objek-objek yang dibutuhkan dalam pembuatan model simulasi RMS ini bisa dilihat pada tabel berikut ini:

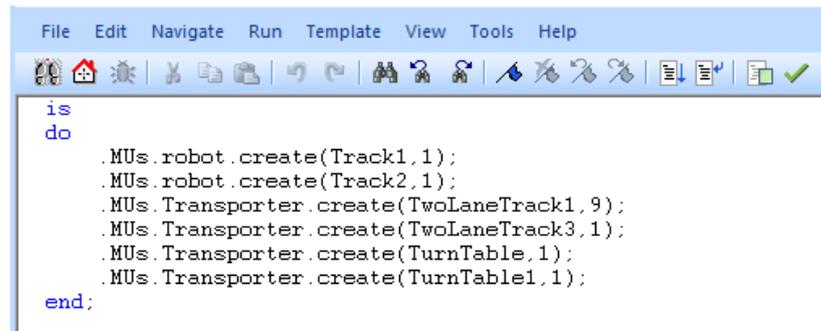
No.	Nama Objek	Jumlah yg dibutuhkan (satuan)
1.	AS/RS	4
2.	Line	12
3.	Track	3
4.	Buffer	4
5.	Lathe	2
6.	Milling	2
7.	MC	1
8.	AGV Type 1	2
9.	AGV Type 2	2
10.	AGV Type 3	2
11.	Robot	6
12.	Control	6 pasang
13.	Method (Syntax)	8
14.	EventController	1
15.	Chart	1
16.	MUs	2 (Red&Blue)
17.	Sensor	17
18.	Drain	2

Berikut ini adalah gambar tampilan seluruh objek pada model simulasi RMS (tanpa inspeksi) yang bisa dijadikan sebagai acuan.



5. Setelah seluruh objek dimasukkan ke dalam *frame* dengan *layout* sesuai dengan gambar diatas, maka yang dilakukan selanjutnya adalah memasukkan bahasa pemrograman berupa *syntax* yang dimuat dalam setiap *method* yang berada di dalam *frame*. Berikut ini akan dibahas satu per satu *method* tersebut.

- *Method 1*

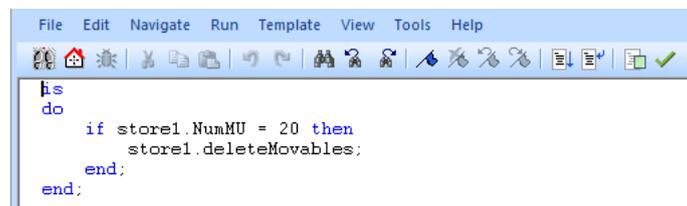


```
is
do
.MUs.robot.create(Track1,1);
.MUs.robot.create(Track2,1);
.MUs.Transporter.create(TwoLaneTrack1,9);
.MUs.Transporter.create(TwoLaneTrack3,1);
.MUs.Transporter.create(TurnTable,1);
.MUs.Transporter.create(TurnTable1,1);
end;
```

Syntax pada *method 1* ini memiliki arti sebagai berikut :

- ✓ Robot dibuat pada *track 1* dengan posisi 1 meter dari titik nol;
- ✓ Robot dibuat pada *track 2* dengan posisi 1 meter dari titik nol;
- ✓ *Transporter/AGV* Tipe 1 dibuat pada *TwoLaneTrack 1* dengan posisi 9 meter dari titik nol;
- ✓ *Transporter/AGV* Tipe 1 dibuat pada *TwoLaneTrack 3* dengan posisi 1 meter dari titik nol;
- ✓ *Transporter/AGV* Tipe tiga dibuat pada *TurnTable 0* dengan posisi 1 meter dari titik nol;
- ✓ *Transporter/AGV* Tipe tiga dibuat pada *TurnTable 1* dengan posisi 1 meter dari titik nol.

- *Method 2*

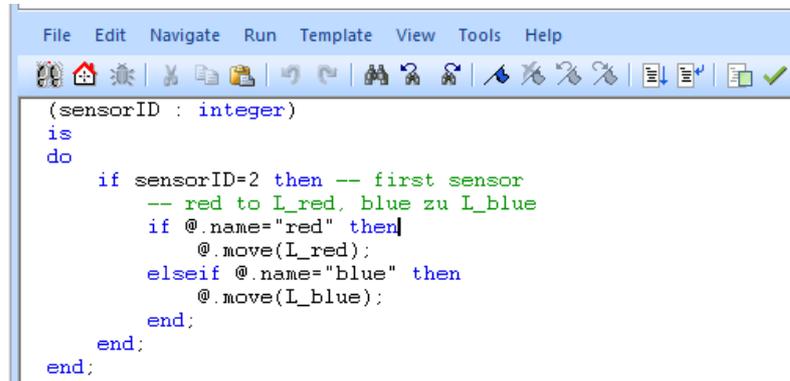


```
is
do
if store1.NumMU = 20 then
store1.deleteMovables;
end;
end;
```

Syntax pada *method 2* ini memiliki arti bahwa AS/RS memiliki kapasitas sebesar 20 satuan produk jadi. Setelah kapasitas penuh, maka seketika produk-produk jadi yang sudah disimpan akan hilang yang berarti sudah

didistribusikan menuju toko atau diantar untuk memenuhi pesanan para konsumen.

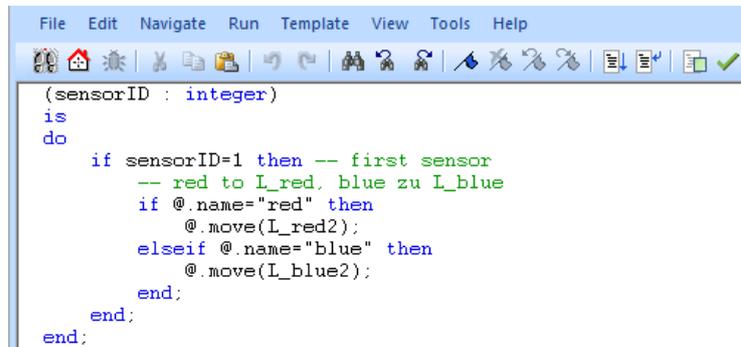
- *Method 3*



```
(sensorID : integer)
is
do
  if sensorID=2 then -- first sensor
    -- red to L_red, blue zu L_blue
    if @.name="red" then
      @.move(L_red);
    elseif @.name="blue" then
      @.move(L_blue);
    end;
  end;
end;
```

Syntax pada *method 3* ini memiliki arti bahwa sensor dengan ID=2 diberikan perintah jika komponen merah melalui sensor ID=2, maka akan diantarkan menuju Line_Red. Sementara itu, jika komponen berwarna biru melalui sensor ID=2, maka akan diantarkan menuju Line_Blue.

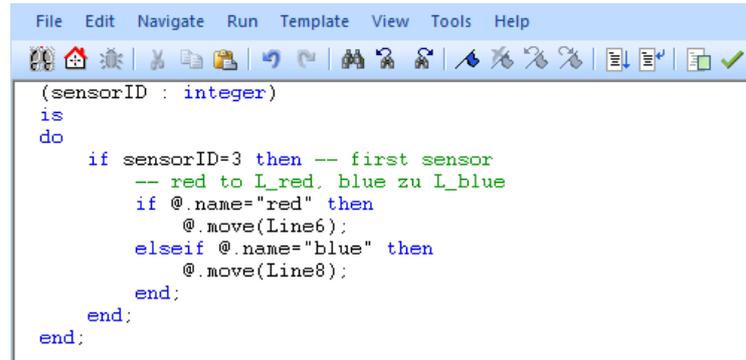
- *Method 4*



```
(sensorID : integer)
is
do
  if sensorID=1 then -- first sensor
    -- red to L_red, blue zu L_blue
    if @.name="red" then
      @.move(L_red2);
    elseif @.name="blue" then
      @.move(L_blue2);
    end;
  end;
end;
```

Syntax pada *method 4* ini memiliki arti bahwa sensor dengan ID=1 diberikan perintah jika komponen merah melalui sensor ID=1, maka akan diantarkan menuju Line_Red 2. Sementara itu, jika komponen berwarna biru melalui sensor ID=1, maka akan diantarkan menuju Line_Blue 2.

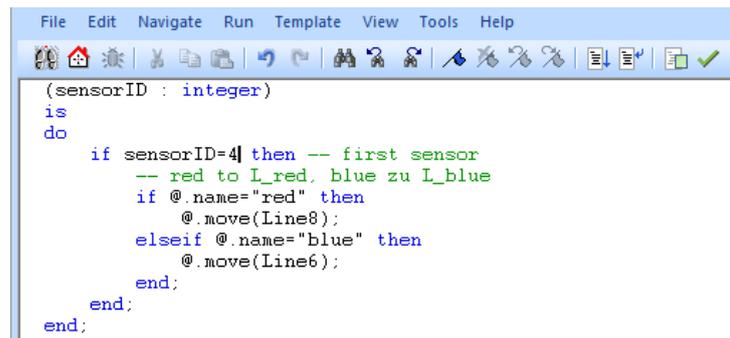
- *Method 5*



```
(sensorID : integer)
is
do
  if sensorID=3 then -- first sensor
    -- red to L_red, blue zu L_blue
    if @.name="red" then
      @.move(Line6);
    elseif @.name="blue" then
      @.move(Line8);
    end;
  end;
end;
```

Syntax pada *method 5* ini memiliki arti bahwa sensor dengan ID=3 diberikan perintah jika komponen merah melalui sensor ID=3, maka akan diantarkan menuju Line 6. Sementara itu, jika komponen berwarna biru melalui sensor ID=3, maka akan diantarkan menuju Line 8.

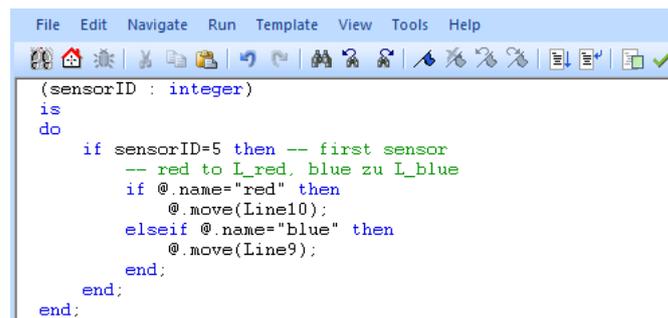
- *Method 6*



```
(sensorID : integer)
is
do
  if sensorID=4 then -- first sensor
    -- red to L_red, blue zu L_blue
    if @.name="red" then
      @.move(Line8);
    elseif @.name="blue" then
      @.move(Line6);
    end;
  end;
end;
```

Syntax pada *method 6* ini memiliki arti bahwa sensor dengan ID=4 diberikan perintah jika komponen merah melalui sensor ID=4, maka akan diantarkan menuju Line 8. Sementara itu, jika komponen berwarna biru melalui sensor ID=4, maka akan diantarkan menuju Line 6.

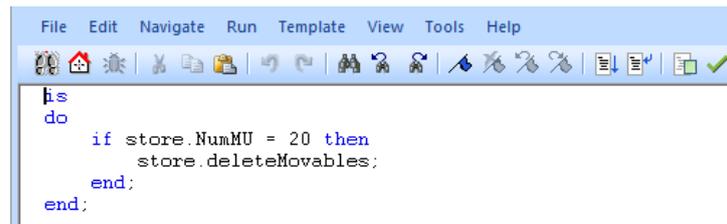
- *Method 7*



```
(sensorID : integer)
is
do
  if sensorID=5 then -- first sensor
    -- red to L_red, blue zu L_blue
    if @.name="red" then
      @.move(Line10);
    elseif @.name="blue" then
      @.move(Line9);
    end;
  end;
end;
```

Syntax pada *method 7* ini memiliki arti bahwa sensor dengan ID=5 diberikan perintah jika komponen merah melalui sensor ID=5, maka akan diantarkan menuju Line 10. Sementara itu, jika komponen berwarna biru melalui sensor ID=5, maka akan diantarkan menuju Line 9.

- *Method 8*

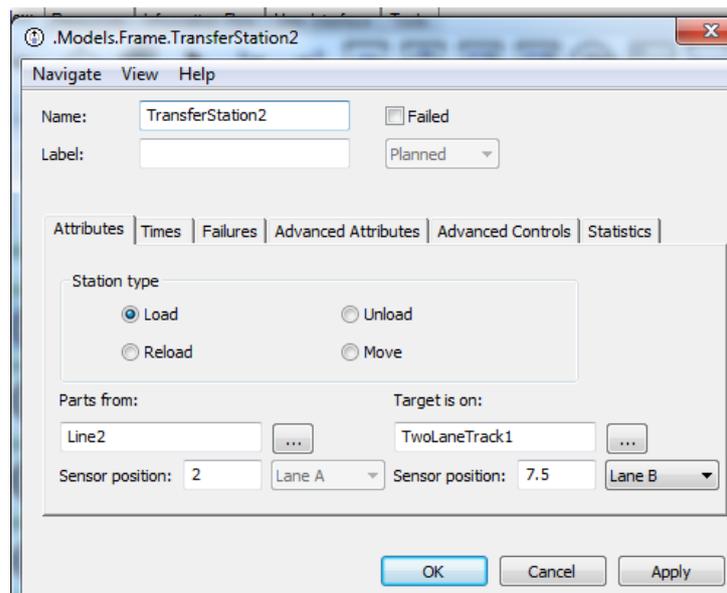


```
is
do
  if store.NumMU = 20 then
    store.deleteMovable;
  end;
end;
```

Syntax pada *method 8* ini memiliki arti bahwa AS/RS memiliki kapasitas sebesar 20 satuan produk jadi. Setelah kapasitas penuh, maka seketika produk-produk jadi yang sudah disimpan akan hilang yang berarti sudah didistribusikan menuju toko atau diantar untuk memenuhi pesanan para konsumen.

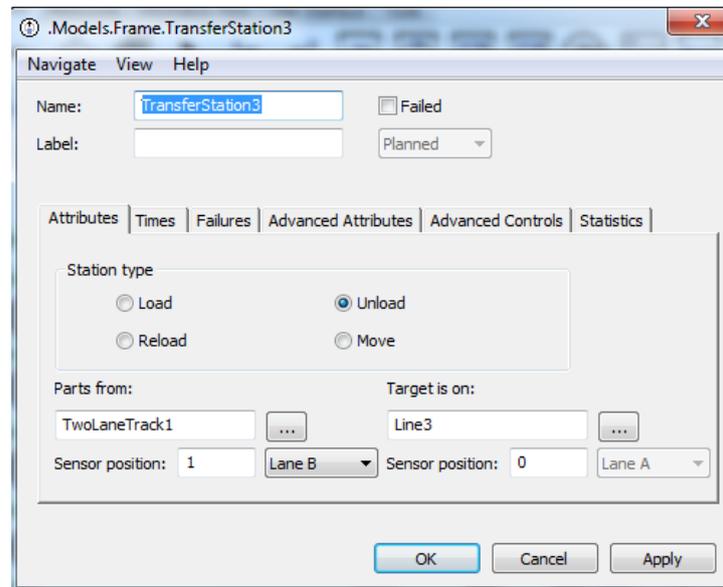
6. Setelah itu yang perlu dilakukan adalah melakukan *setting* pusat kontrol yang memiliki kendali kontrol sebanyak 6 pasang.

- Kendali Kontrol 1



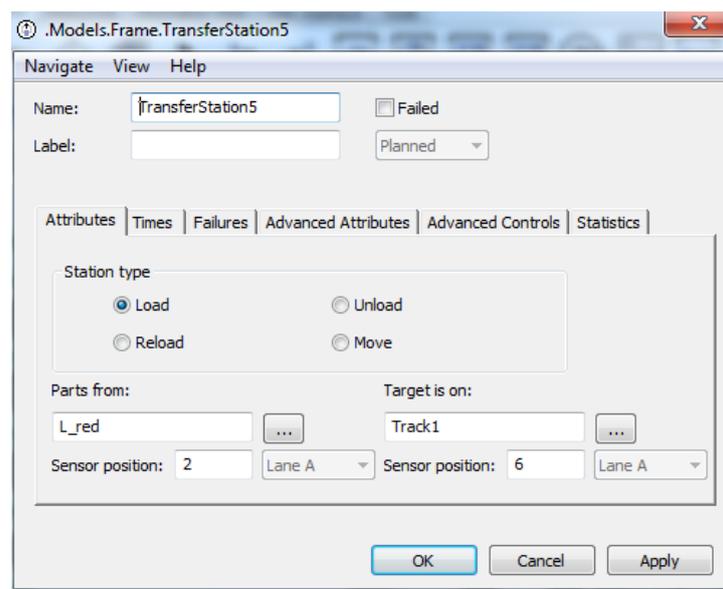
Kendali kontrol pertama ini memiliki arti bahwa AGV Tipe Satu akan mengambil komponen dan membawanya dari Line 2 menuju TwoLane Track 1.

- Kendali Kontrol 2



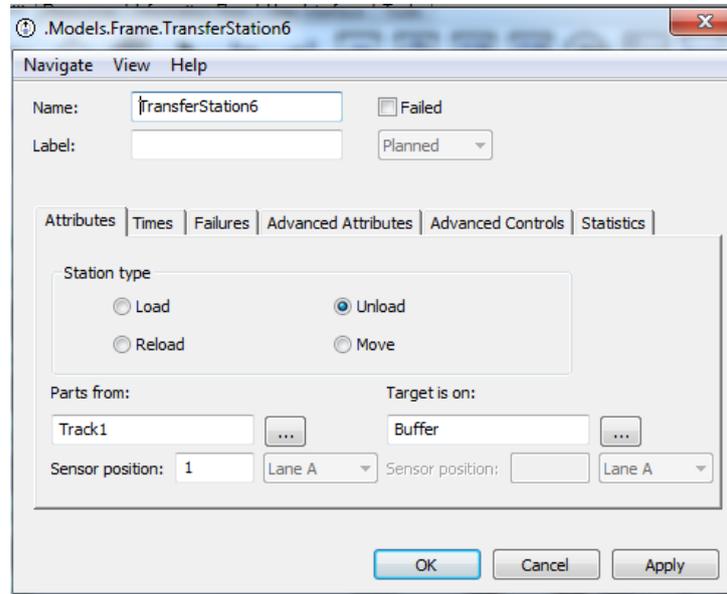
Kendali kontrol kedua ini memiliki arti bahwa AGV Tipe Satu akan membongkar/melepas komponen setelah tiba di Line 3.

- Kendali Kontrol 3



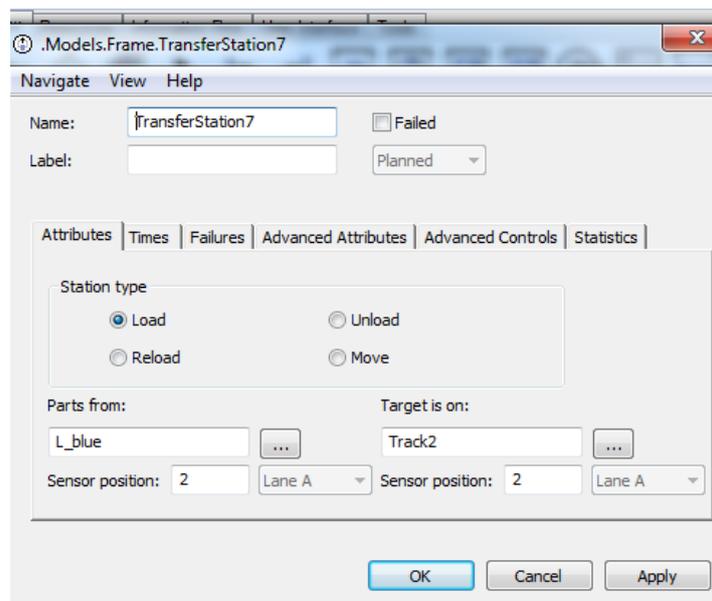
Kendali kontrol ketiga ini memiliki arti bahwa AGV Tipe Dua akan membawa komponen dan mengantarnya dari Line_red menuju Track1.

- Kendali Kontrol 4



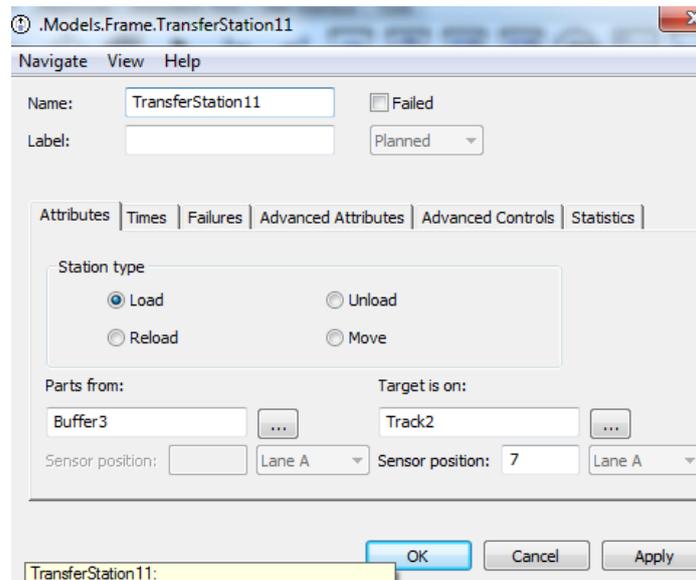
Kendali kontrol keempat ini memiliki arti bahwa AGV Tipe Dua akan membongkar/melepas komponen dari Track1 menuju buffer.

- Kendali kontrol 5



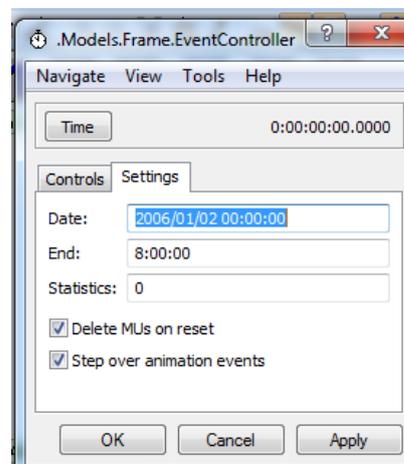
Kendali kontrol kelima ini memiliki arti bahwa AGV Tipe Dua akan membawa komponen dan mengantarnya dari Line_blue menuju Track2.

- Kendali Kontrol keenam



Kendali kontrol kelima ini memiliki arti bahwa AGV Tipe Dua akan membawa komponen dan mengantarnya dari buffer 3 menuju Track2.

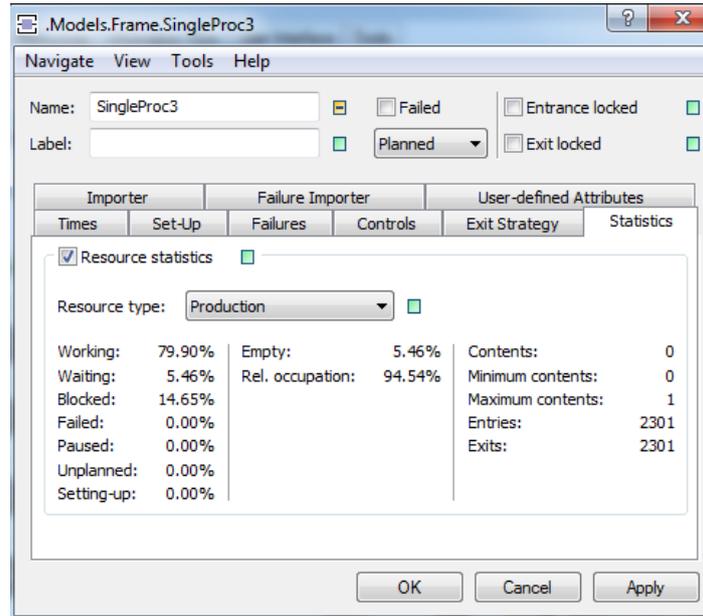
7. Setelah itu lakukan *setting* EventController seperti gambar berikut ini :



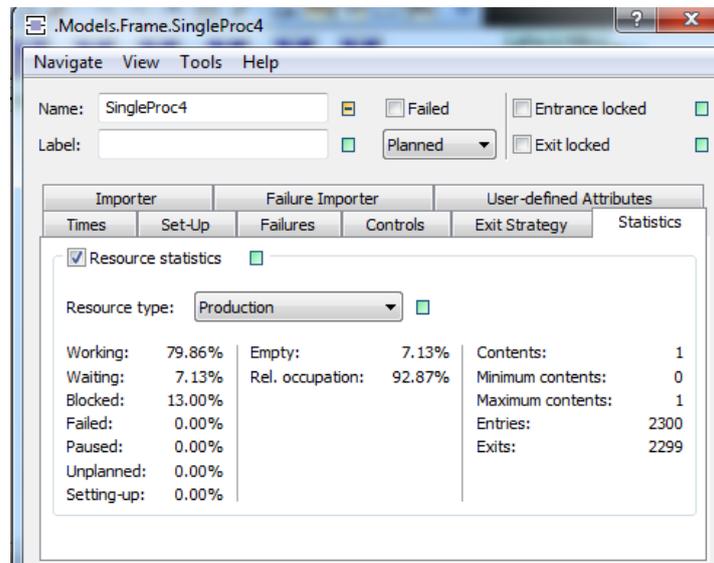
Dapat dilihat bahwa penulis membuat asumsi lama pekerjaan adalah satu *shift* normal.

8. Untuk melihat hasil *output* per mesin dapat dilihat dengan melakukan *double click* pada objek yang ingin kita cermati kemudian klik *tab statistics*.

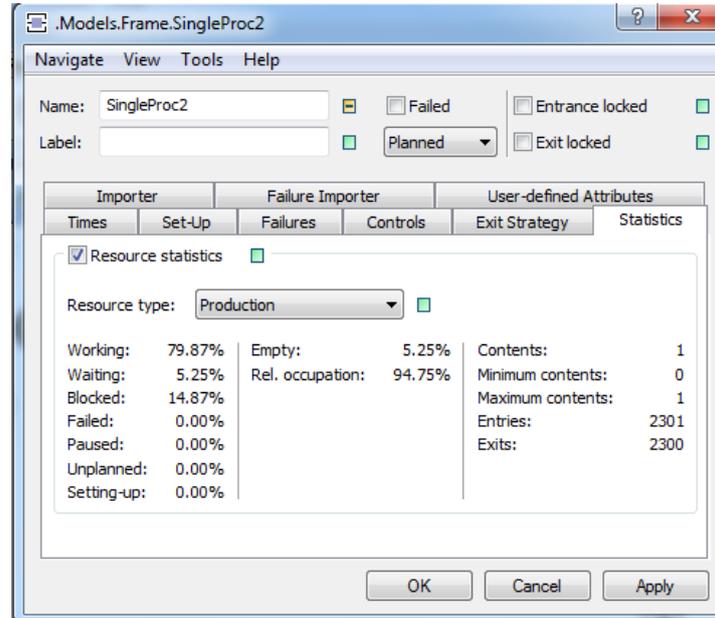
- *Lathe Machine*



- *Milling Machine*



- *Machining Center (MC)*



9. Untuk melihat hasil *output* akhir yang diproduksi, dapat dilihat dengan *double* klik bagian *Drain* kemudian dilihat pada *tab Type Statistics*.

