

**PERANCANGAN *DIGITAL TWIN* BERBASIS SIMULASI UNTUK *SMART WAREHOUSE*: *ASSET ADMINISTRATION SHELL FRAMEWORK***

**TESIS**

**Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Magister Strata-2  
Pada Jurusan Teknik Industri Fakultas Teknologi Industri**



**Disusun Oleh:**

**Nama : Zakka Ugih Rizqi**

**No. Mahasiswa : 19916040**

**PROGRAM STUDI TEKNIK INDUSTRI  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA  
YOGYAKARTA**

**2021**

## PERNYATAAN KEASLIAN

Dengan ini saya menyatakan bahwa karya ini adalah hasil kerja saya sendiri kecuali nukilan dan ringkasan yang setiap satunya telah saya jelaskan sumbernya. Jika dikemudian hari ternyata terbukti pengakuan saya ini tidak benar dan melanggar peraturan yang sah dalam karya tulis dan hak kekayaan intelektual maka saya bersedia ijazah yang telah saya terima untuk ditarik kembali oleh Universitas Islam Indonesia.

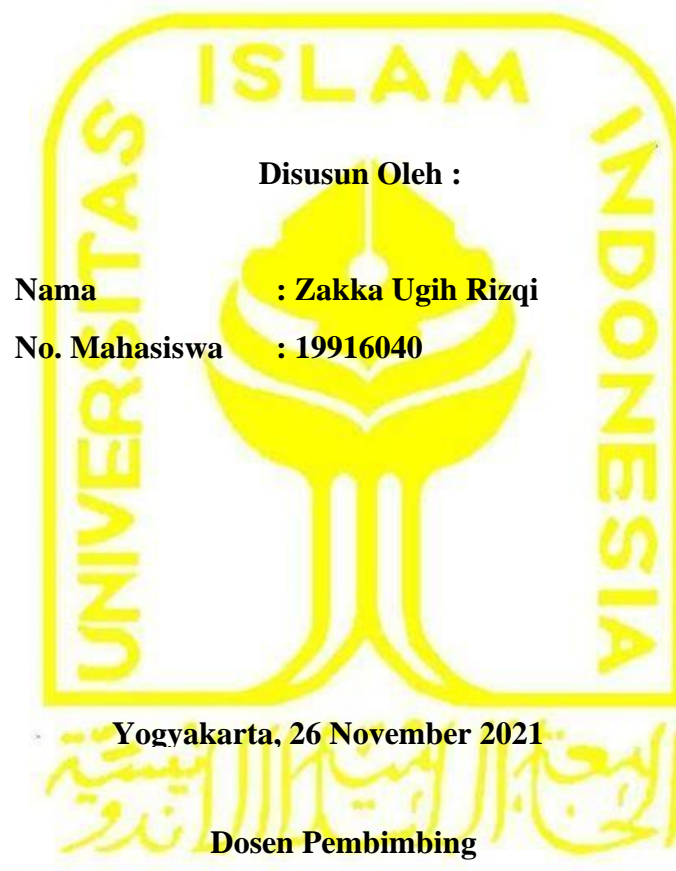
Yogyakarta, 26 November 2021



Zakka Ugih Rizqi

**LEMBAR PENGESAHAN PEMBIMBING**  
**PERANCANGAN *DIGITAL TWIN* BERBASIS SIMULASI UNTUK *SMART***  
***WAREHOUSE: ASSET ADMINISTRATION SHELL FRAMEWORK***

**TESIS**



**Winda Nur Cahyo, S.T., M.T., Ph.D.**

**LEMBAR PENGESAHAN PENGUJI**  
**PERANCANGAN *DIGITAL TWIN* BERBASIS SIMULASI UNTUK *SMART***  
***WAREHOUSE: ASSET ADMINISTRATION SHELL FRAMEWORK***

**TESIS**

**Disusun Oleh :**

**Nama : Zakka Ugih Rizqi**  
**No. Mahasiswa : 19916040**

**Telah dipertahankan di depan sidang penguji sebagai salah satu syarat**  
**untuk memperoleh gelar Magister Teknik Industri di Fakultas Teknologi**  
**Industri Universitas Islam Indonesia**  
**Yogyakarta, 26 November 2021**

**Tim Penguji**

**Winda Nur Cahyo, S.T., M.T., Ph.D.**

**Ketua**

**Muhammad Ridwan Andi Purnomo, S.T., M.Sc., Ph.D.**

**Anggota I**

**Dr. Dwi Handayani, S.T., M.Sc.**

**Anggota II**

**Mengetahui,**

**Ketua Program Studi Magister Teknik**

**Industri Universitas Islam Indonesia**



**Winda Nur Cahyo, S.T., M.T., Ph.D.**

*NR Cahyo*

*Muhammad Ridwan Andi Purnomo*

*Dr. Dwi Handayani*

## DEDIKASI

*I dedicate this master thesis to my parents as a gift and a form of gratitude expression for providing moral and material support and endless affection so that I can undergo all stages of my education in the Master of Industrial Engineering program in Universitas Islam Indonesia and Master of Industrial Management program in National Taiwan University of Sciences and Technology in dual degree program.*

*Hope everything you gave me can be merit charity and insyaAllah will help you to go to Jannah. I will always love you Ayah and Mamah. Thanks for loving me.*

**MOTTO**

*“Whoever takes the path to seek knowledge, Allah will make it easy for him to go to heaven”*

(H.R. Muslim, No. 2699)

*“Whoever practices the knowledge he has learned, Allah will inherit the knowledge he does not know”*

*“Remember, the death is getting closer. So don't waste your time”*

*“For you who have believed, fear Allah. And let every soul look to what it has done for tomorrow - and fear Allah. Indeed, Allah is Aware of what you are doing.”*

(Q.S. Al-Hasyr: 18)

***Rabbana Atina Fiddunya Hasanah Wa Fil Akhiroti  
Hasanah Waqina ‘Adzabannar***

## KATA PENGANTAR

Assalamualaikum Warahmatullahi Wabarakatuh,

Alhamdulillahirabbil'alamiin, Puji syukur kehadiran Allah SWT yang telah melimpahkan segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi ini sesuai dengan waktu yang diharapkan. Semoga shalawat dan salam tercurah kepada Rasullullah Muhammad Shallallahu 'Alaihi wa Sallam beserta keluarga, sahabat, dan pengikutnya, yang telah menyampaikan syafaat-Nya kepada kita semua.

Dengan segala kerendahan hati, penulis ingin mengucapkan terima kasih kepada pihak-pihak yang telah meluangkan waktu dan perhatiannya, sehingga baik secara langsung maupun tidak langsung membantu penulis dalam menyelesaikan skripsi ini. Penulis ingin mengucapkan terima kasih kepada:

1. Prof. Dr. Ir. Hari Purnomo, M. T. selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia
2. Muhammad Ridwan Andi Purnomo, S.T., M. Sc., Ph. D. selaku Ketua Jurusan Teknik Industri Universitas Islam Indonesia
3. Winda Nur Cahyo, S.T., M.T., Ph. D. selaku Ketua Program Studi S2 Teknik Industri Universitas Islam Indonesia sekaligus dosen pembimbing tesis yang telah memberikan arahan, doa, motivasi, dan saran serta waktunya yang tak ternilai
4. Dr. Taufiq Immawan S.T., M.M. selaku Ketua Program Studi S1 Teknik Industri Universitas Islam Indonesia
5. Prof. Shuo-Yan Chou, selaku dosen pembimbing di NTUST yang telah memberikan kepercayaan kepada Penulis untuk mengikuti berbagai proyek serta pengetahuan dan dukungan teknis
6. Keluarga tercinta, yakni kedua orang tua Sugih Bagjo Sayad dan Daryati Nurhayati serta kedua adik saya Dzyqi Ugih Qinthara dan Zyba Ugih Arkana, yang senantiasa selalu memberikan motivasi, nasihat, dukungan dan kasih sayangnya selama proses penyelesaian tesis ini

7. Adinda Khairunisa yang telah menemani Penulis dikala susah ataupun senang, memberikan dukungan, dan semangat serta memotivasi Penulis dalam proses penyelesaian tesis ini

Penulis menyadari bahwa Tesis ini masih jauh dari kata sempurna, untuk itu penulis memohon maaf. Harapan terakhir, semoga Tesis ini dapat berguna bagi penulis khususnya dan bagi pembaca pada umumnya. Amiin Yaa Robbal ‘Aalamin.

Wassalamualaikum Warahmatullahi Wabarakatuh.

Yogyakarta, 26 November 2021

Zakka Ugih Rizqi  
19916040

## ABSTRAK

*Kebutuhan untuk meningkatkan fleksibilitas, optimalitas, dan transparansi proses industri meningkat karena tuntutan pelanggan sangat beragam dan penuh ketidakpastian. Teknologi Cyber-Physical System (CPS) saat ini sedang dikembangkan untuk memenuhi kebutuhan tersebut. Hal ini dapat dicapai melalui Digital Twin (DT). Akan tetapi, penerapan DT masih terbatas dalam praktiknya saat ini dan tidak ada cara standar untuk mencapai integrasi. Asset Administration Shell (AAS) yang diusulkan oleh Plattform Industrie 4.0 muncul sebagai framework yang menjanjikan untuk mewujudkan DT secara standar. Studi literatur menunjukkan bahwa sebagian besar penelitian sebelumnya hanya mencapai DT statis dan hanya mempertimbangkan beberapa aset tertentu terutama aset yang tidak bergerak. DT dinamis hanya didukung oleh Virtual Reality (VR) yang tidak dapat digunakan untuk optimisasi maupun kemampuan kontrol. Oleh karena itu, penelitian ini berkontribusi dalam pengembangan DT dengan mengusulkan teknologi simulasi komputer 3 dimensi sebagai DT dinamis dengan mengikuti Reference Architecture Model Industrie 4.0 (RAMI 4.0) berdasarkan Framework AAS. Konsep yang diusulkan memungkinkan DT untuk melakukan pemantauan dinamis, optimisasi, dan pengendalian langsung. Studi kasus digunakan untuk verifikasi konsep yang diusulkan berupa smart warehouse di Taiwan dan aset yang dipertimbangkan adalah Automated Guided Vehicles (AGV), Operator, Konveyor, Forklift, dan Storage Rack. Penelitian ini terdiri dari 4 tahap. Pertama, AAS Variables Definition, mengidentifikasi aset dan data yang ingin dipertukarkan melalui AAS, mendefinisikan kapabilitas AAS, dan struktur AAS. Kedua, AAS Simulation Modeling, dengan membangun model simulasi menggunakan software Flexsim. Ketiga, AAS System Integration terdiri dari perancangan jaringan komunikasi dan pemodelan informasi. Dan terakhir, mendemonstrasikan 2 keuntungan utama menggunakan simulasi yaitu optimisasi berbasis simulasi untuk perencanaan kapasitas AGV dan pembuatan dashboard dinamis. Untuk mendukung komunikasi antar aset, penggunaan OPC Unified Architecture (OPC UA) dan Automation Markup Language (AutomationML) diusulkan. Sebagai hasil dari demonstrasi, optimisasi berbasis simulasi dilakukan untuk multi-objective dan dapat menentukan kapasitas optimal dari masing-masing AGV. Dashboard dinamis juga dibuat untuk memudahkan proses monitoring secara real-time. Nantinya, data dari dasbor dapat diekspor ke database dan analisis prediktif dapat dilakukan. Berdasarkan kapabilitas tersebut, skenario industri 4.0 dapat diintegrasikan secara sistematis dan komprehensif.*

**Keywords:** *Asset Administration Shell, Digital Twin, Simulasi, Optimisasi*

## DAFTAR ISI

PERNYATAAN KEASLIAN .....	ii
LEMBAR PENGESAHAN PEMBIMBING.....	iii
LEMBAR PENGESAHAN PENGUJI.....	iv
DEDIKASI.....	v
MOTTO .....	vi
KATA PENGANTAR .....	vii
ABSTRAK.....	ix
DAFTAR ISI.....	x
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiii
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Tujuan Penelitian .....	3
1.4 Manfaat Penelitian .....	4
1.5 Batasan Penelitian .....	4
1.6 Sistematika Penulisan.....	4
<b>BAB II KAJIAN LITERATUR.....</b>	<b>6</b>
2.1 Kajian Deduktif.....	6
2.1.1 Industri 4.0.....	6
2.1.2 <i>Reference Architecture Model for Industry 4.0 (RAMI 4.0)</i> .....	7
2.1.3 <i>Digital Twin</i> .....	8
2.1.4 <i>Asset Administration Shell</i> .....	10
2.1.5 <i>AutomationML</i> .....	12
2.1.6 OPC UA.....	13
2.1.7 <i>Discrete-Event Simulation dan Simulation Optimization</i> .....	15
2.1.8 <i>Flexsim Simulation Software</i> .....	16
2.2 Kajian Induktif .....	18
2.3 Kebaruan Penelitian .....	21
<b>BAB III METODE PENELITIAN .....</b>	<b>23</b>
3.1. <i>AAS Variables Definition</i> .....	24
3.2 <i>AAS Simulation Modeling</i> .....	24

3.3 AAS System Integration.....	25
3.4 Demonstrasi.....	25
<b>BAB IV HASIL DAN PEMBAHASAN.....</b>	<b>28</b>
4.1 Aset-Aset di Smart Warehouse .....	28
4.2 Model Simulasi .....	33
4.3 Integrasi Sistem <i>Digital Twin</i> .....	36
4.4 Demonstrasi.....	43
4.4.1 <i>Simulation-based Optimization</i> untuk <i>AGV Capacity Planning</i> .....	43
4.4.2 <i>Dynamic Dashboard</i> .....	48
<b>BAB V KESIMPULAN DAN REKOMENDASI.....</b>	<b>51</b>
5.1 Kesimpulan .....	51
5.2 Rekomendasi .....	52
<b>REFERENSI .....</b>	<b>53</b>

**DAFTAR TABEL**

Tabel 2. 1 Digital Twin dalam Praktik .....	9
Tabel 2. 2 Software Simulasi Terbaik pada 2019 .....	16
Tabel 2. 3 <i>State of the Art</i> .....	22
Tabel 4. 1 <i>AAS Capability</i> .....	29
Tabel 4. 2 Pendefinisian AAS berdasarkan Struktur AAS untuk Aset Utama .....	32
Tabel 4. 3 Data Input .....	35
Tabel 4. 4 Solusi Optimal <i>Non-Dominated</i> melalui <i>Pareto Optimality</i> .....	46
Tabel 4. 5 Hasil <i>Weighted Sum Method</i> .....	47

## DAFTAR GAMBAR

Gambar 1. 1 Sistematika Penulisan Tesis .....	5
Gambar 2. 1 Reference Architecture Model for Industry 4.0 .....	8
Gambar 2. 2 Perkembangan Digital Twin pada 2015 .....	9
Gambar 2. 3 DT dalam Memecahkan Permasalahan Silo Informasi .....	10
Gambar 2. 4 Konsep dari <i>Asset Administration Shell</i> .....	11
Gambar 2. 5 Struktur dari AAS .....	12
Gambar 2. 6 Struktur dari AML .....	13
Gambar 2. 7 Struktur dari OPC UA .....	14
Gambar 2. 8 Metodologi Simulasi secara Umum .....	15
Gambar 2. 9 Konsep <i>Digital Twin</i> pada Flexsim .....	17
Gambar 3. 1 <i>Flowchart</i> Penelitian .....	23
Gambar 3. 2 <i>AAS Definition Template</i> .....	24
Gambar 3. 3 Mekanisme dari <i>Genetic Algorithm</i> .....	26
Gambar 3. 4 Framework dari Optimisasi <i>AGV Capacity Planning</i> .....	27
Gambar 4. 1 Aset-Aset di <i>Smart Warehouse</i> .....	28
Gambar 4. 2 Model Konseptual dari <i>Smart Warehouse</i> .....	33
Gambar 4. 3 Tata Letak dari <i>Smart Warehouse</i> .....	34
Gambar 4. 4 Model Simulasi <i>Smart Warehouse</i> dari Depan .....	35
Gambar 4. 5 Model Simulasi <i>Smart Warehouse</i> dari Belakang .....	35
Gambar 4. 6 Gambaran Konsep Umum yang Diusulkan .....	36
Gambar 4. 7 Korespondensi AAS untuk Data Posisi .....	37
Gambar 4. 8 Korespondensi AAS untuk <i>SemProM</i> di <i>Storage Rack</i> .....	38
Gambar 4. 9 Korespondensi AAS untuk <i>SemProM</i> di Konveyor .....	38
Gambar 4. 10 <i>AAS Correspondence</i> untuk <i>Controllable Part</i> .....	40
Gambar 4. 11 Konfigurasi Sensor dan Kontrol di <i>AAS Flexsim</i> .....	40
Gambar 4. 12 <i>AAS Framework</i> di <i>Smart Warehouse</i> .....	41
Gambar 4. 13 Interaksi antar AAS .....	43
Gambar 4. 14 Kode untuk <i>Simulation-based Optimization</i> .....	45
Gambar 4. 15 Hasil Optimisasi untuk Maksimasi <i>Throughput</i> .....	45
Gambar 4. 16 Hasil Optimisasi untuk Minimasi <i>Energy Consumption</i> .....	46
Gambar 4. 17 <i>Dynamic Dashboard</i> .....	48
Gambar 4. 18 <i>Dashboard</i> sebelum Optimisasi .....	50
Gambar 4. 19 <i>Dashboard</i> sesudah Optimisasi .....	50
Gambar 4. 20 Konfigurasi dari Proses <i>Controlling</i> .....	50

## BAB I

### PENDAHULUAN

#### 1.1 Latar Belakang

Digitalisasi menjadi salah satu tujuan utama dalam revolusi industri keempat atau disebut juga Industri 4.0 (I4.0) untuk meningkatkan fleksibilitas, kemampuan beradaptasi, optimalitas, dan transparansi proses industri (Jasperneite et al., 2020). Hal itu juga mendorong mode produksi dalam industri untuk beralih dari *mass production* ke *mass customization* (Tien, 2011). Namun, kebutuhan ini menyebabkan peningkatan kompleksitas pada sistem produksi.

Teknologi *Cyber-Physical System* (CPS) saat ini sedang dikembangkan untuk memenuhi kebutuhan tersebut. Hal ini dicapai dengan membangun aset digital yang terhubung secara *real-time* dengan aset fisiknya dan memungkinkan pertukaran informasi dua arah di antara mereka, yang biasa disebut *Digital Twin* (Colombo et al., 2017). Disebut '*Twin*' karena model digital yang dibuat seperti sebuah pencerminan atau kembaran sistem antara apa yang ada di sistem nyata dengan apa yang ada di sistem digital.

Untuk mengimplementasikan *Digital Twin*, internet merupakan teknologi dasar untuk mewujudkannya. Pada saat ini, internet telah berkembang pesat hingga telah menjadi kebutuhan bagi semua orang di dunia. Perkembangan teknologi yang semakin maju menjadikan internet memiliki biaya yang murah dan jaringan yang luas. Bahkan pada tahun 2019, jumlah pengguna internet di seluruh dunia mencapai 4,13 miliar yang berarti lebih dari setengah populasi global saat ini terhubung ke internet (Clement J., 2020). Kaitannya dengan industri 4.0, internet bertujuan untuk mengubah industri tradisional menjadi industri cerdas melalui *Industrial Internet of Things* (IIoT). Untuk Industri 4.0, komputer bukanlah pusat teknologi, tetapi Internet (Bitkom et al., 2015). Internet memiliki peran penting untuk memastikan interoperabilitas komponen, peralatan, dan produk yang terhubung dalam industri. Hal ini mendukung semua kegiatan digitalisasi di industri dan membuat *smart manufacturing* menjadi kenyataan (Lampropoulos et al., 2019).

Beberapa kelompok riset di berbagai negara mulai mewujudkan *digital twin* untuk *smart factory*. Eropa pertama kali meluncurkan program *Factories of the Future* (2008-2020 dengan investasi 2,35 Miliar Euro), Singapura meluncurkan *Future of Manufacturing* (FoM) dengan investasi 500 juta SD pada tahun 2014 (Seif et al., 2019), dan Jerman menginvestasikan 200 Juta Euro pada tahun 2012 dengan meluncurkan Plattform Industrie 4.0 (Schweichhart, 2016). Sebagai hasilnya, konsep *Asset Administration Shell* (AAS) yang diusulkan oleh Plattform Industri 4.0 menjadi salah satu hasil yang paling menjanjikan untuk mewujudkan *digital twin* dalam konteks Komponen Industri 4.0 (I4.0) (Adolphs et al., 2016), Hal ini sejalan dengan model referensi standar yang diusulkan yaitu *Reference Architecture Model for Industry 4.0* (RAMI 4.0) (Bitkom et al., 2015).

AAS pada dasarnya terdiri dari dua istilah penting yaitu '*Asset*' dan *Administration Shell*'. *Asset* mengacu pada hal-hal fisik misalnya mesin & peralatan, dan *Administration Shell* mengacu pada hal digital yang menyimpan informasi tentang aset selama siklus hidupnya. Dalam *smart manufacturing*, setiap benda fisik atau *Asset* akan memiliki *Administration Shell*nya sendiri, dan setiap *Administration Shell* dapat berkomunikasi dengan *Administration Shell* lainnya. Namun, sampai saat ini pendefinisian AAS masih dalam pengembangan, sangat sedikit yang memahami bagaimana menggunakan AAS dalam praktiknya. (Ye et al., 2020). Oleh karena itu, AAS merupakan topik yang sangat menantang dan masih memiliki peluang penelitian yang besar.

Penelitian sebelumnya sudah mencoba menerapkan AAS pada pembangunan DT. (Tantik & Anderl, 2017) dan (Ye & Hong, 2018) menerapkan AAS pada aset *robot arm* dan *conveyor* dan mengusulkan *Web User Interface* (UI) sebagai *digital twin* statisnya. (Marcon et al., 2019) dan (Assadi et al., 2020) mendesain AAS untuk pekerja yang disebut *Human Administration Shell* (HAS) dan menggunakan *Human-Machine Interface* (HMI) sebagai *digital twin* statisnya. (Muralidharan et al., 2020) mengimplementasikan *digital twin* pada *air quality sensor* berdasarkan *Things Description* (TD) dan mengusulkan *Docker Image* sebagai *digital twin* statisnya. Dan (Seif et al., 2019) mengimplementasikan AAS pada area yang lebih luas yang mana diterapkan pada *assembly process* dan membangun HMI sebagai *digital twin* statisnya.

Berdasarkan penelitian sebelumnya, aplikasi AAS dalam mewujudkan DT masih terbatas terutama pada aspek visualisasi yang hanya menampilkan model statis daripada model dinamis. Oleh karena itu, penelitian ini akan mengembangkan *digital twin* berbasis simulasi menggunakan Flexsim dengan framework AAS yang menjadikan DT bersifat dinamis sehingga dapat dikatakan output dari penelitian ini adalah *conceptual development*. Selain itu,

penelitian sebelumnya hanya mempertimbangkan beberapa aset tertentu dan aset tidak bergerak. Padahal, dalam praktiknya akan banyak aset yang perlu diperhatikan dan dihubungkan. Dalam penelitian ini, *smart warehouse* digunakan sebagai objek penelitian sehingga memperluas aset yang dipertimbangkan dalam kerangka AAS. Selain itu, penelitian sebelumnya menyatakan bahwa meskipun penelitian DT sudah berhasil dilakukan, tetapi masih ada kekurangan di mana DT tidak dapat melakukan optimisasi (CyberFactory#1, 2020). Dengan menggunakan konsep yang diusulkan pada penelitian ini, optimisasi dapat dilakukan dan akan didemonstrasikan untuk mengoptimalkan perencanaan kapasitas AGV serta perancangan dashboard dinamis. Penelitian ini juga bermaksud untuk berpartisipasi dalam pengembangan industri 4.0 yang menjadi topik hangat hingga saat ini.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang ada, rumusan masalah pada penelitian ini yaitu:

1. Bagaimana konsep *digital twin* berbasis simulasi dengan software Flexsim pada *smart warehouse* berbasis framework AAS?
2. Bagaimana model simulasi dan konsep integrasi sistem antara dunia fisik dan dunia digital pada *smart warehouse*?
3. Apakah konsep *digital twin* berbasis simulasi menggunakan framework AAS dapat melakukan optimisasi *multi-objective* berbasis simulasi dan menunjukkan dasbor dinamis sebagai kelebihan utama dari konsep yang diusulkan?

## 1.3 Tujuan Penelitian

Tujuan dari penelitian ini yaitu:

1. Mengembangkan konsep baru untuk *digital twin* berbasis simulasi dengan perangkat lunak Flexsim di *smart warehouse* berbasis framework AAS.
2. Membangun model simulasi dan konsep integrasi sistem antara dunia fisik dan dunia digital.
3. Mendemonstrasikan optimisasi *multi-objective* berbasis simulasi pada perencanaan kapasitas AGV dan merancang dasbor dinamis sebagai kelebihan utama dari konsep yang diusulkan.

## 1.4 Manfaat Penelitian

Berdasarkan tujuan dari penelitian ini, beberapa manfaat dan kontribusi yang dihasilkan adalah sebagai berikut:

1. Menghadirkan konsep baru dengan memanfaatkan simulasi khususnya software Flexsim untuk pengembangan *digital twin* berbasis AAS *framework*. Konsep yang diusulkan ini akan menjadikan *digital twin* menjadi dinamis dan memiliki kemampuan optimisasi.
2. Penelitian ini memperluas aset yang dipertimbangkan dalam DT dan sangat penting dalam hal mengusulkan submodel dalam AAS. Tidak hanya aset tidak bergerak tetapi juga aset bergerak.
3. Menyediakan *preliminary study* pada *system integration* untuk *smart warehouse*.

## 1.5 Batasan Penelitian

Demi menghasilkan penelitian yang terarah fokus kajiannya dan keterbatasan data dan waktu yang dimiliki penulis, maka batasan penelitian ini yaitu:

1. Penelitian ini hanya sampai pada *conceptual development*, *simulation modeling*, dan *system integration*, tidak sampai tahap implementasi.
2. Hanya 5 aset utama yang dipertimbangkan dalam memodelkan *smart warehouse* yaitu *Automated Guided Vehicle (AGV)*, *Operator*, *Conveyor*, *Forklift*, dan *Storage Rack*.
3. Analisis ekonomi tidak dipertimbangkan, penelitian ini hanya fokus pada aspek teknis.

## 1.6 Sistematika Penulisan

Kerangka penelitian ini dijelaskan sebagai berikut: Bab 1 berisi mengenai latar belakang, tujuan, keterbatasan, dan sistematika penulisan tesis. Bab 2 menyajikan studi induktif dan studi deduktif serta state-of-the-art yang terkait dengan aplikasi AAS dan kebaruan penelitian ini. Bab 3 menyajikan metode penelitian yang digunakan untuk mencapai tujuan penelitian. Bab ini mencakup 4 tahap utama yaitu AAS Variables Definition, AAS Simulation Modeling, AAS System Integration, dan Demonstrasi. Bab 4 menyajikan hasil dan pembahasan dari setiap tahapan yang ada. Bab 5 menyajikan kesimpulan dari semua bab dan juga mengusulkan penelitian masa depan. Gambar 1.1. di bawah ini menggambarkan organisasi tesis ini.



Gambar 1. 1 Sistematika Penulisan Tesis

## BAB II

### KAJIAN LITERATUR

Bab ini dibagi menjadi tiga bagian utama. Dua bagian pertama akan menampilkan hasil *review* yang bersumber dari literatur, yaitu kajian deduktif dan kajian induktif. Kajian deduktif berisi hasil *review* yang diperoleh dari buku, laporan, jurnal, dan prosiding yang berkaitan dengan teori dasar yang digunakan dalam penelitian ini. Sedangkan kajian induktif berisi hasil *review* dari 20 penelitian terdahulu yang diperoleh dari jurnal dan prosiding internasional yang memuat penelitian serupa dengan penelitian ini yaitu *Digital Twin* berbasis *AAS framework*. Bagian terakhir akan menekankan perbedaan dan kebaruan penelitian ini dibandingkan dengan penelitian-penelitian yang pernah dilakukan sebelumnya.

#### 2.1 Kajian Deduktif

##### 2.1.1 Industri 4.0

Revolusi industri keempat mendigitalisasi operasi manufaktur dan mendesentralisasikannya dari *shop floor* ke *office* dan seluruh jaringan perusahaan yang ada. (Jasperneite et al., 2020). Tujuan utamanya adalah memuaskan kebutuhan pelanggan yang sangat bervariasi sehingga memengaruhi bidang-bidang seperti manajemen pesanan, operasi manufaktur, pengiriman ke pelanggan (Vaidya et al., 2018). Empat pendorong utama Industri 4.0 adalah *Industrial Internet of Things (IIoT)*, *Cloud-based Manufacturing*, *Internet of Things (IoT)* dan *Smart Manufacturing* yang membantu dalam mengubah proses manufaktur menjadi proses yang sepenuhnya digital dan cerdas. (Erol et al., 2016). Studi lain mengusulkan 9 pilar I4.0 yaitu *Big Data & Analytics*, *Autonomous Robots*, *Simulation*, *System Integration*, *The Industrial Internet of Things*, *Cyber security and Cyber Physical Systems (CPS)*, *Cloud*, *Additive Manufacturing*, dan *Augmented Reality*. (Vaidya et al., 2018). Namun tantangannya adalah bagaimana mengintegrasikan teknologi-teknologi tersebut ke dalam satu sistem yang terintegrasi penuh dan menyediakan cara standar untuk diimplementasikan.

Plattform Industrie 4.0 (Platform Industrie 4.0, 2016) mengusulkan 9 skenario pada pengaplikasian industri 4.0 untuk mempermudah pemahaman dalam implementasinya di lapangan sebagai berikut:

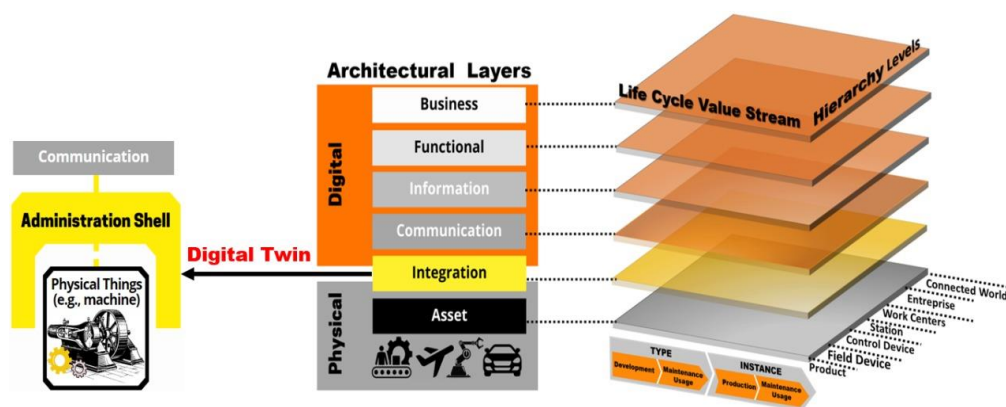
1. OCP – *Order-Controlled Production*: Interkoneksi produksi yang otonom dan otomatis.
2. AF – *Adaptable Factory: Plug & Produce* – Konfigurasi manufaktur yang dapat disesuaikan secara adaptif di dalam pabrik dengan kemampuan produksi dengan cepat.
3. SAL – *Self-organising Adaptive Logistics*: Meningkatkan fleksibilitas dan waktu reaksi sistem industri dan logistik di lingkungan produksi agar mudah beradaptasi.
4. VBS – *Value-Based Services*: Pemanfaatan data proses, status dari sumber produksi, dan penggunaan produk untuk model bisnis dan layanan masa depan.
5. TAP – *Transparency and Adaptability of Delivered Products*: Pengumpulan otomatis data berbasis penggunaan dari produk yang dikirimkan untuk mengoptimalkan proses bisnis, untuk model bisnis baru, dan untuk adaptasi dinamis fitur produk.
6. OSP – *Operator Support in Production*: Interaksi manusia-teknologi-manusia untuk membantu manusia sebagai aktor dalam proses manufaktur.
7. SP2 – *Smart Product Development for Smart Production*: Produk virtual memungkinkan jenis kerja tim baru dalam proses rekayasa dan otomatisasi aktivitas rekayasa.
8. IPD – *Innovative Product Development*: Mengintegrasikan semua pemangku kepentingan yang relevan dalam proses rekayasa dan menggunakan pendekatan metodis yang sesuai membantu mengembangkan produk dengan cara yang benar.
9. CRE – *Circular Economy*: Daur ulang bahan bermutu tinggi dari manufaktur selama seluruh siklus hidup suatu produk.

### **2.1.2 Reference Architecture Model for Industry 4.0 (RAMI 4.0)**

Dalam membahas Industri 4.0, terdapat kerangka yang mendekati isu Industri 4.0 secara terstruktur, salah satunya yaitu *Reference Architecture Model for Industry 4.0* (RAMI 4.0) yang diajukan oleh Plattform Industrie 4.0 dari Jerman ditunjukkan pada Gambar 2.1 (Bitkom et al., 2015). Sebenarnya ada juga framework lain dengan tujuan yang sama seperti *Industrial Internet Reference Architecture* (IIRA) yang diusulkan oleh IIC dari U.S. (Shi-Wan et al., 2017), *IBM Industry 4.0 Architecture* diusulkan oleh IBM (Helmann et al., 2020), dan *NIST Smart Manufacturing* (Lu et al., 2015).

Dibandingkan dengan yang lain, RAMI 4.0 memberikan informasi yang lebih komprehensif dan lebih relevan untuk diterapkan di industri terutama untuk *smart manufacturing* (Helmann et al., 2020). RAMI 4.0 memberikan orientasi untuk merencanakan persyaratan sektor bersama dengan standar untuk menentukan dan mengembangkan Industri 4.0 lebih lanjut (Han, 2020). Hal ini diimplementasikan dalam aspek Industri 4.0 untuk *smart manufacturing* yang menggabungkan semua elemen dan komponen Teknologi Informasi (H. S. Park & Febriani, 2019).

Karena tantangan utama industri 4.0 adalah interoperabilitas, maka pada RAMI 4.0 juga terlihat dari lapisan arsitektural yang terbagi menjadi 2 aspek yaitu aspek fisik dan aspek digital. Aspek fisik disebut aset, sedangkan aspek digital dimulai dari lapisan komunikasi sampai lapisan bisnis seperti yang ditunjukkan pada Gambar 2.1. Untuk mengubah hal fisik menjadi digital, terdapat lapisan integrasi antara aset dan lapisan komunikasi yang menerapkan konsep Digital Twin (DT).

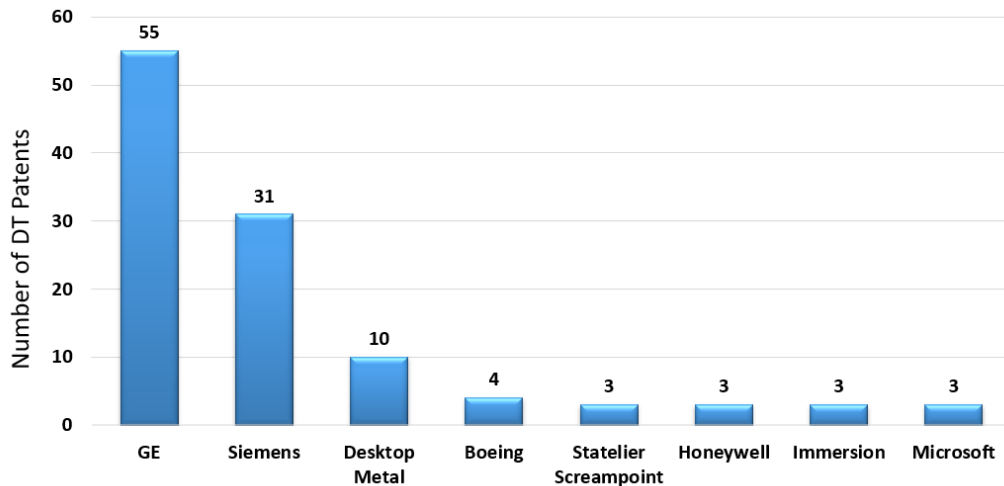


Gambar 2. 1 Reference Architecture Model for Industry 4.0

(Sumber: (Bitkom et al., 2015))

### 2.1.3 Digital Twin

Konsep *Digital Twin* (DT) berasal dari presentasi pada tahun 2002 berjudul “*Conceptual Ideal for Product Lifecycle Management (PLM)*” oleh Dr. Grieves (Grieves, 2006). Disebut 'Twin' karena seperti pencerminan atau kembaran sistem antara yang ada di ruang nyata dengan yang ada di ruang maya dan sebaliknya. Namun saat itu, internet masih mahal dan terbatas. Karena DT menggunakan internet sebagai teknologi dasar sehingga perkembangan DT tidak muncul. Namun saat ini, internet menjadi lebih murah dan tanpa batas sehingga banyak industri telah mengembangkan DT untuk meningkatkan interoperabilitas antar aset seperti yang ditunjukkan pada Gambar 2.2 (Institute of Intellectual Property Research & Development, 2015).






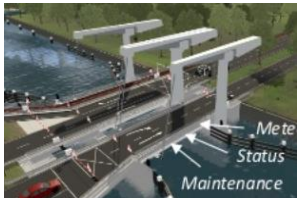


Gambar 2. 2 Perkembangan Digital Twin pada 2015

(Sumber: (Institute of Intellectual Property Research & Development, 2015))

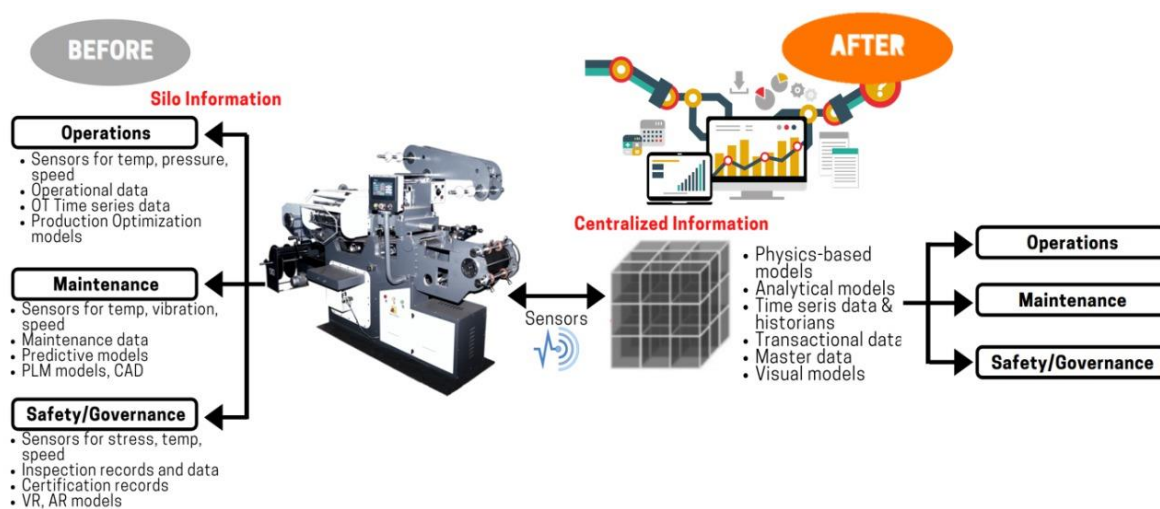
DT dikenal sebagai konsep dasar untuk mendukung integrasi di Industri 4.0 dan mengacu pada model virtual suatu aset yang terintegrasi penuh dengan benda/aset nyata.(Krasikov & Kulemin, 2020). Dalam DT, aset digital dan aset fisik dihubungkan oleh internet dengan hubungan semantik. Ini dapat digunakan untuk pengujian virtual, menganalisis data yang dikumpulkan, dan memprediksi perilaku masa depan dari hal-hal fisik sebelum menerapkannya pada hal-hal nyata (Muralidharan et al., 2020). Karena DT memiliki begitu banyak keunggulan, DT dapat diterapkan secara praktis di hampir setiap industri seperti yang ditunjukkan pada Tabel 2.1.

Tabel 2. 1 Digital Twin dalam Praktik

(Sumber:(Gageldonk & Waterings, 2019))

	<b>Construction</b>	<b>Aerospace</b>	<b>High Tech</b>
<b>Physical Things</b>			
<b>Digital Twin</b>			

Selain itu, DT juga dapat menyelesaikan permasalahan silo informasi seperti yang ditunjukkan pada Gambar 2.3. Secara tradisional, informasi tentang suatu aset biasanya tersebar di berbagai sumber informasi, yang dikelola oleh organisasi yang berbeda. Ini akan menghasilkan informasi yang terduplikasi, tidak konsisten, dan mudah hilang. Dengan DT, data dapat dikumpulkan secara terpusat untuk setiap entitas dan kemudian membuat informasi tersebut tersedia untuk berbagai area bisnis untuk aplikasi spesifik mereka melalui program antarmuka, seperti *Application Programming Interfaces (API)* (Industrial Internet Consortium (IIC), 2020). Hal ini meningkatkan pengambilan keputusan melalui pemahaman bersama dan mengurangi biaya siklus hidup keseluruhan dari pengoperasian dan pemeliharaan pabrik.

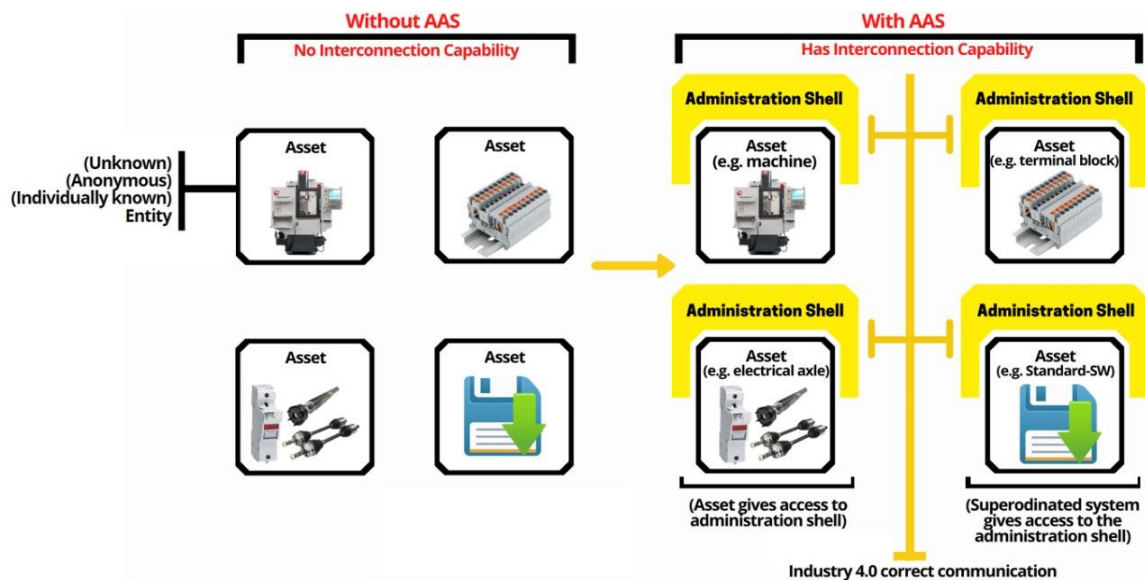


Gambar 2. 3 DT dalam Memecahkan Permasalahan Silo Informasi

(Sumber: (Industrial Internet Consortium (IIC), 2020))

#### 2.1.4 Asset Administration Shell

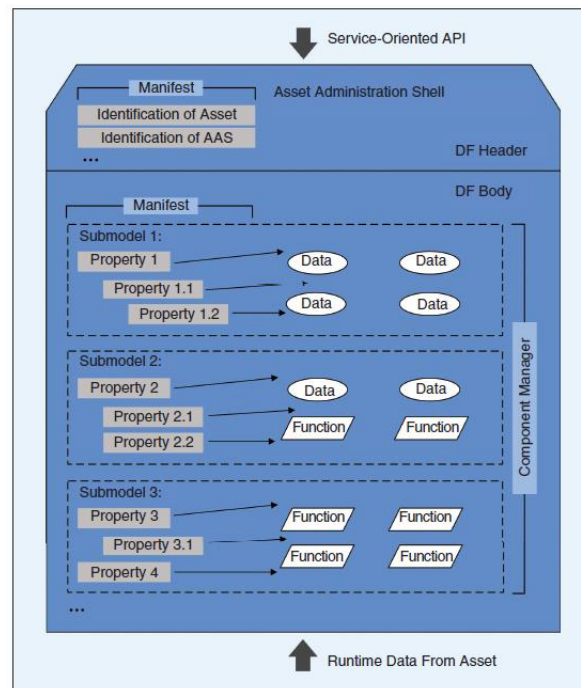
Salah satu pendekatan untuk mewujudkan Digital Twin di industri adalah *Asset Administration Shell (AAS)* (Industrial Internet Consortium (IIC), 2020). AAS pada dasarnya terdiri dari dua istilah yaitu '*Asset*' dan '*Administration Shell*'. *Asset* mengacu pada hal-hal fisik misalnya mesin, produk, & bahan, dan *Administration Shell* mengacu pada hal digital yang menyimpan informasi tentang aset selama seluruh siklus hidupnya. Dalam manufaktur cerdas, setiap benda atau aset fisik memiliki *Administration Shell*nya sendiri, dan setiap *Administration Shell* dapat terhubung dengan *Administration Shell* lainnya seperti yang ditunjukkan pada Gambar 2.4 (Industrial Internet Consortium (IIC), 2020).



Gambar 2. 4 Konsep dari *Asset Administration Shell*

(Sumber: (Industrial Internet Consortium (IIC), 2020))

Struktur AAS dapat dibagi menjadi *Digital Factory (DF) header* and *DF body* (Tantik & Anderl, 2017). Konsep sistem DF telah distandarisasi dalam standar *International Electrotechnical Commission (IEC) 62832* (Ye et al., 2020). *DF header* menggambarkan daftar properti yang sesuai dengan I4.0, seperti identifikasi aset fisik dan identifikasi AAS seperti yang ditunjukkan pada Gambar 2. 5 (Wagner et al., 2017). Ini harus disertakan untuk secara jelas memberikan identitas unik Komponen I4.0. Daftar properti ini disebut manifes, yang dicirikan sebagai "Kumpulan meta-informasi yang dapat diakses secara eksternal, yang menyediakan informasi tentang properti fungsional dan nonfungsional dari Komponen I4.0" (Adolphs et al., 2015). *DF body* berisi manajer komponen yang bertanggung jawab penuh atas pengelolaan dan akses ke layanan di masing-masing submodel. Setiap submodel terdiri dari properti hierarkis yang terorganisir dengan baik yang terkait dengan berbagai data atau fungsi.



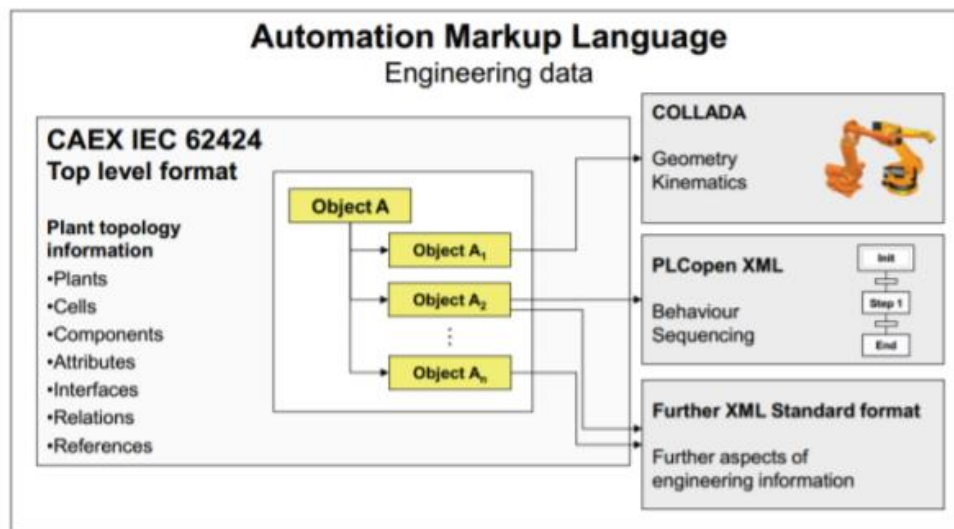
Gambar 2. 5 Struktur dari AAS  
(Sumber: (Wagner et al., 2017))

Dengan menggunakan AAS, interoperabilitas dapat ditingkatkan tidak hanya di dalam perusahaan, tetapi juga antar perusahaan dalam jaringan rantai pasokan yang sama seperti pemasok dan produsen. AAS juga dapat diintegrasikan dengan pilar-pilar lain di industri 4.0 seperti *Big Data* karena data dari aset akan dicatat selama siklus hidup aset (*volume*) untuk setiap detik (*veracity*) dan untuk setiap spesifikasi aset (*variety*), *Cloud Computing* karena data akan disimpan dan diakses dari cloud, Simulasi untuk melakukan trial and error berdasarkan skenario yang dirancang, dan IoT yang tertanam dalam aset.

### 2.1.5 AutomationML

*AutomationML* atau *Automation Markup Language* atau AML ditujukan untuk mencapai pertukaran data yang terstandarisasi dan efisien selama proses rekayasa sistem manufaktur, yang dapat mengurangi waktu dan biaya (Ye & Hong, 2018). Ide utamanya adalah, alih-alih mengembangkan format data baru untuk tujuan itu, *AutomationML* mengadaptasi, memperluas, dan menggabungkan format data standar yang sudah ada dengan cara yang sesuai.

AML adalah format pertukaran data berbasis XML yang terbuka dan netral yang dapat digunakan untuk deskripsi produk, proses, dan sumber daya dalam domain rekayasa sistem produksi. (Schleipen et al., 2015). Struktur dasar *AutomationML* ditunjukkan pada Gambar 2.6 (Lüder et al., 2010). Inti dari *AutomationML* adalah format data tingkat atas *Computer Aided Engineering Exchange* (CAEX), yang menghubungkan format data teknik khusus, seperti *Collaborative Design Activity* (COLLADA) dan PLCopen XML (IEC 62714-1, 2012). Selain itu, *AutomationML* memiliki arsitektur terdistribusi yang melekat pada konsep metode berorientasi objek, seperti halnya CAEX.



Gambar 2. 6 Struktur dari AML

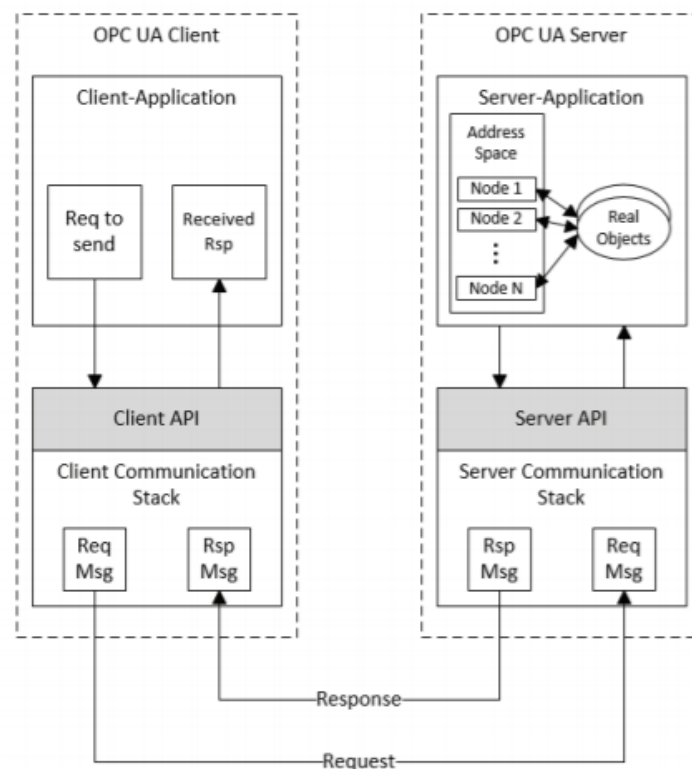
(Sumber: (Lüder et al., 2010))

### 2.1.6 OPC UA

Teknologi *Open Platform Communications Unified Architecture* (OPC UA) menyediakan komunikasi berbasis pesan antara klien dan server pada berbagai jenis jaringan untuk memfasilitasi berbagai pertukaran informasi sistem industri, seperti koneksi ke perangkat lapangan, sistem kontrol, *manufacturing execution systems* dan *enterprise resource planning* (Lee et al., 2017).

OPC UA mengikuti sistem *client-server*. Klien OPC UA menggunakan API untuk mengirim/menerima pesan permintaan/tanggapan layanan OPC UA (*Req/Rsp Msg*) ke/dari server OPC UA. Di sini, layanan dirancang untuk menemukan server OPC UA yang tersedia di lingkungan sistem tertentu. Selanjutnya, API klien dikembangkan untuk bertindak sebagai antarmuka internal untuk mengatur kode aplikasi klien terpisah dari tumpukan komunikasi klien.

Peran tumpukan komunikasi adalah untuk mendukung transportasi data antara klien dan server, yang mengubah panggilan API klien menjadi pesan dan mengirimkannya melalui entitas komunikasi yang mendasarinya ke server OPC UA (OPC Foundation, 2012). Di OPC UA Server, ia memiliki "*address space*" yang terdiri dari kumpulan node, yang mewakili objek, variabel, metode, dan jenis metadata OPC UA (Ye & Hong, 2018). Gambar 2. 7 menunjukkan arsitektur client/server OPC UA (Ye et al., 2018). Ini menggambarkan elemen utama dari klien dan server OPC UA yang khas serta proses komunikasi yang digunakan untuk berinteraksi satu sama lain.



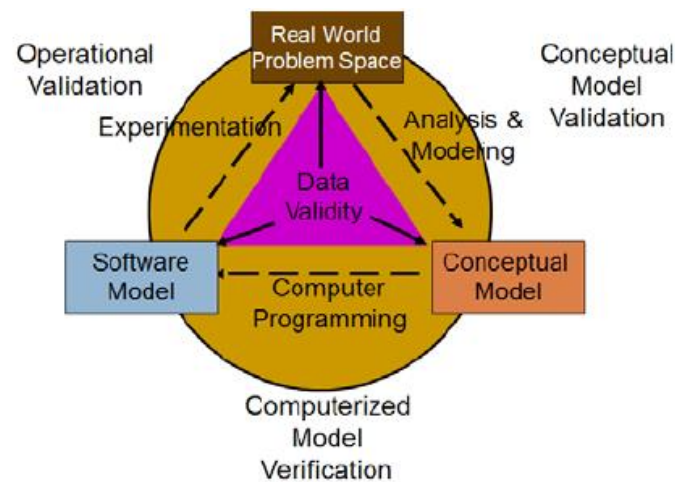
Gambar 2. 7 Struktur dari OPC UA

(Sumber: (Ye et al., 2018))

Dalam perspektif RAMI 4.0, OPC UA terletak di lapisan komunikasi dan berfungsi sebagai antarmuka untuk komunikasi AAS. Keunggulan OPC UA dibandingkan Fieldbus dan protokol lainnya adalah OPC UA menyediakan model informasi unik yang terdiri dari berbagai objek node, properti, dan referensi (Ye et al., 2020). Inilah alasan mengapa model AML dapat diintegrasikan ke dalam OPC UA karena metadata AML dapat dipetakan ke node OPC UA secara standar (Henßen & Schleipen, 2014).

### 2.1.7 Simulation dan Simulation-Optimization

Simulasi merupakan metode digunakan untuk meniru perilaku suatu sistem dengan menggunakan bantuan komputer (Law & Kelton, 2000). Terdapat beberapa jenis simulasi seperti *System Dynamics*, *Agent-based Modeling*, dan *Discrete-Event Simulation (DES)*. Setiap jenis simulasi memiliki kekurangan dan kelebihan masing-masing. Dalam hal pemodelan berbasis kejadian terutama pada sistem yang bersifat sekuensial, DES tepat digunakan seperti memodelkan proses produksi, operasi gudang, sistem antrian, dan sebagainya. Secara umum, metodologi simulasi ditunjukkan pada Gambar 2.8.



Gambar 2. 8 Metodologi Simulasi secara Umum

(Sumber:(Fonseca, 2011))

DES secara kuantitatif mewakili dunia nyata, menyimulasikan dinamikanya di bawah ketidakpastian, dan menghasilkan laporan kinerja terperinci berdasarkan hasil simulasi. Ini telah menjadi salah satu alat pengambilan keputusan berbantuan komputer utama karena aksesibilitas komputer yang efektif (Law & Kelton, 2000). DES juga merupakan metode pemodelan fleksibel yang dicirikan oleh kemampuan untuk mewakili perilaku kompleks di dalam dan interaksi antara individu, populasi, dan lingkungannya (Burgess & Pidd, 1999). DES dapat meniru operasi yang terjadi dalam suatu sistem dengan bantuan komputer dan perangkat lunak canggih berdasarkan beberapa asumsi sehingga sistem dapat dipelajari secara ilmiah (Law & Kelton, 2000).

Dalam perspektif *Digital Twin*, simulasi harus mampu memvisualisasikan sistem semirip mungkin. Dengan bantuan *software* canggih, sekarang DT dapat dicapai menggunakan simulasi. Hal lain yang perlu diperhatikan adalah simulasi memiliki banyak keunggulan yang mendukung Industri 4.0 seperti kemampuan untuk melakukan *What-If analysis*, *simulation-based optimization*, laporan yang akurat, bahkan menampilkan *dashboard* yang dinamis.

Simulasi klasik biasanya menggunakan pendekatan *trial and error* untuk mencari solusi yang lebih baik. Namun membutuhkan banyak biaya dalam perspektif waktu dan kurang efektif. Para peneliti telah mencoba untuk mengintegrasikan teknik optimisasi seperti metaheuristic untuk disematkan di dalam perangkat lunak simulasi. Salah satu algoritma terkenal yang biasa digunakan dalam optimisasi simulasi adalah Algoritma Genetika (GA) (Fontanili et al., 2000). Berdasarkan kemampuan ini, *simulation-based optimization* atau *simulation optimization* dapat mencapai keputusan yang optimal dengan cara yang efisien.

### 2.1.8 Flexsim Simulation Software

Sejak simulasi merupakan salah satu teknologi utama yang mendukung *digital twin*, maka penentuan software simulasi harus dipertimbangkan secara matang berdasarkan kemampuannya. Salah satu software simulasi yang terbukti dapat digunakan sebagai bagian dari DT adalah software Flexsim (Luściński, 2018). Flexsim dikembangkan dan dimiliki oleh Flexsim Software Products, Inc. dari Orem, Utah (Nordgren, 2003). Flexsim adalah simulator berorientasi objek yang dikembangkan di C++, dan menggunakan teknologi *Open Graphics Library* (GL). Flexsim memiliki kemampuan untuk menganimasikan sistem nyata melalui tampilan pohon, 2-D, 3-D, dan *Virtual Reality*. Banyak perusahaan menggunakan Flexsim untuk meningkatkan efisiensi produksi dan mengurangi biaya operasi melalui simulasi, eksperimen, dan pengoptimalan (Morgan et al., 1984).

Ada banyak software simulasi yang mendukung *discrete-event simulation*. Dibandingkan dengan perangkat lunak simulasi lainnya, penelitian dari (Solutions, 2019) memberikan hasil bahwa Flexsim merupakan software simulasi terbaik tahun 2018 dan 2019 berdasarkan kriteria *Capability, General Purpose, Ease of Use, Popularity, dan Development*. Hasil rinci ditunjukkan pada Tabel 2.2 (Solutions, 2019).

Tabel 2. 2 Software Simulasi Terbaik pada 2019

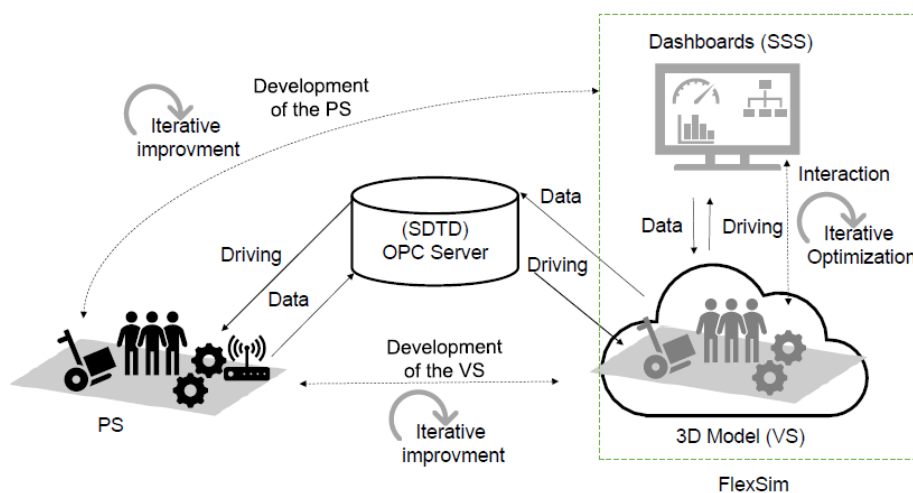
(Sumber: (Solutions, 2019))

<i>Simulation Tool</i>	<i>Best of 2018</i>	<i>Most Capable</i>	<i>Most General Purpose</i>	<i>Easiest of Use</i>	<i>Most Popular</i>	<i>Most Actively Developed</i>
AnyLogic	3.4	3.6	2.7	2.3	4.3	3.5
Arena	3.3	3.6	2.2	2.5	4.3	2.0
AutoMod	2.3	2.7	2.5	1.0	1.6	1.0
Emulate3D	2.8	2.4	3.0	3.0	1.3	3.5
FlexSim	4.3	4.4	4.2	3.5	4.7	4.0
OPS	2.1	2.0	1.7	3.0	1.0	1.5

<i>Simulation Tool</i>	<i>Best of 2018</i>	<i>Most Capable</i>	<i>Most General Purpose</i>	<i>Easiest of Use</i>	<i>Most Popular</i>	<i>Most Actively Developed</i>
ProModel	2.5	2.8	1.3	2.0	3.0	1.5
SimEvents	2.1	2.8	1.5	2.0	1.3	2.5
Witness	2.1	2.4	1.5	1.8	1.3	2.0
<b>WINNER</b>	<b>FlexSim</b>	<b>FlexSim</b>	<b>FlexSim</b>	<b>FlexSim</b>	<b>FlexSim</b>	<b>FlexSim</b>

Flexsim memiliki begitu banyak fitur yang mendukung konsep *digital twin*. Pada tahun 2017, Flexsim memperkenalkan fitur baru untuk mendukung interkoneksi fisik-virtual yang disebut PLC Emulation: Emulation tool untuk mengontrol model 3D dengan logika simulasi *programmable logic controller* (PLC). Dalam Emulasi PLC, semua sensor dan data PLC dipertukarkan dengan server *Open Platform Communications* (OPC) atau langsung dengan PLC melalui protokol komunikasi serial Modbus (Luściński, 2018).

Di Flexsim, Emulasi PLC juga dapat dikombinasikan dengan fitur optimisasi yang disebut OptQuest yang mengimplementasikan *Evolutionary Algorithm* dan akses database SQL yang menjadikan FlexSim pilihan yang menjanjikan untuk membuat prototipe dan bereksperimen dengan konsep kembar digital. Ilustrasinya ditunjukkan pada Gambar 2. 8 (Luściński, 2018) dimana DT diimplementasikan *Shop-floor Service System* (SSS) dengan *Shop-floor Digital Twin Data* (SDTD). Flexsim juga memiliki visualisasi 3D yang bagus dan *dashboard* yang dapat disesuaikan sehingga akan lebih mudah untuk menganalisis sistem nyata berdasarkan sistem virtual berbasis Flexsim.



Gambar 2. 9 Konsep *Digital Twin* pada Flexsim

(Sumber: (Luściński, 2018))

## 2.2 Kajian Induktif

Penelitian tentang DT dilakukan oleh (Ye et al., 2020) dalam mewujudkan *Plug-and-Produce*. Penelitian ini menggunakan framework *Asset Administration Shell* untuk membuat DT antara 3 *Robot* & 1 *Turntable* sebagai aset fisik dan menggunakan Web UI sebagai *static digital twin*. Untuk berkomunikasi antar AAS, OPC UA & AML digunakan sebagai teknologi utama. *Sequence Diagram* juga digunakan untuk merepresentasikan arus informasi antar AAS. Penelitian ini berfokus pada *Adaptable Factories* (AF) sebagai skenario industri 4.0 yang dikembangkan. DT yang dibangun memiliki kemampuan untuk mengontrol aset fisik.

Penelitian berbeda (Marcon et al., 2019) menggunakan Operator sebagai aset untuk membangun AAS dengan bantuan *Smart Jacket* dan membuat *Human-Machine Interface* (HMI) sebagai *static digital twin* dan memiliki kemampuan untuk mengontrol aset fisik. Untuk berkomunikasi antara AAS, OPC UA, NodeRED pada MCU, dan MQTT digunakan sebagai teknologi utama. *Block Diagram* juga digunakan untuk merepresentasikan arus informasi antar AAS. Penelitian ini berfokus pada *Operator Support in Production* (OSP) sebagai skenario industri 4.0 yang akan dikembangkan. Penelitian serupa yang berfokus pada manusia sebagai objek AAS dilakukan oleh (Haße et al., 2020) yang berfokus pada *Operator Support in Production* (OSP) sebagai skenario industri 4.0. *Remote Control* di Operator *Forklift* digunakan sebagai aset fisik. REST API dan MQTT digunakan sebagai alat komunikasi utama. GUI dibuat sebagai DT statis tetapi tidak memiliki kemampuan untuk mengontrol aset fisik. *Flowchart* digunakan sebagai metode untuk merepresentasikan informasi.

Beberapa peneliti juga telah menerapkan AAS di area yang lebih luas. Penelitian ini (Seif et al., 2019) menggunakan proses perakitan dan membuat *Human Machine Interface* (HMI) sebagai *digital twin* statis dan memiliki kemampuan untuk mengontrol aset fisik. OPC UA, CSV, JSON, dan REST API digunakan sebagai teknologi utama untuk berkomunikasi antar AAS. Penelitian ini berfokus pada *Adaptable Factories* (AF) sebagai skenario industri 4.0 yang akan dikembangkan. Arus informasi antar AAS direpresentasikan dengan *flowchart*. Penelitian serupa yang berfokus pada aplikasi DT dalam proses perakitan dilakukan oleh (Arm et al., 2021) khususnya dalam Produksi Plastik sebagai aset fisik dan membuat simulasi berbasis 2D sebagai *digital twin* statis dan tidak memiliki kemampuan untuk mengontrol aset fisik. Untuk berkomunikasi antara AAS, MQTT & JSON digunakan sebagai teknologi utama. *Sequence diagram* juga digunakan untuk merepresentasikan arus informasi antar AAS. Penelitian ini berfokus pada *Smart Product Development for Smart Production* (SP2) sebagai skenario industri 4.0 yang akan dikembangkan. Penelitian ini (Toro et al., 2020) menggunakan

stasiun robot sebagai aset fisik untuk membuat *digital twin*. REST API, OPC UA, CSV, dan JSON digunakan sebagai alat komunikasi antar AAS. GUI digunakan sebagai DT statis dan *Flowchart* juga digunakan sebagai metode yang merepresentasikan aliran informasi. Penelitian ini berfokus pada *Adaptable Factory* (AF) sebagai skenario industri 4.0 yang akan dikembangkan. DT yang dibuat memiliki kemampuan untuk mengontrol aset fisik.

Penelitian lain berfokus pada *Operator Support in Production* (SP2) sebagai skenario industri 4.0 (Assadi et al., 2020) yang memanfaatkan mesin dan operator serta membuat *Human-Machine Interface* (HMI) sebagai *static digital twin*. Untuk berkomunikasi antara AAS, JSON, dan MQTT digunakan sebagai teknologi utama. Diagram blok digunakan untuk merepresentasikan arus informasi antar AAS. Penelitian serupa di SP2 (Muralidharan et al., 2020) menggunakan Sensor Kualitas Udara dan membuat *Docker Image* sebagai DT statis. Untuk berkomunikasi antara AAS, Kubernetes dan *Things Description* (TD)/JSON digunakan sebagai teknologi utama. *Flowchart* digunakan untuk mewakili arus informasi antara AAS.

Penelitian lain (Ye & Hong, 2019) menggunakan 2 robot & 2 konveyor dan membuat Web-UI sebagai DT statis. Untuk berkomunikasi antara AAS, OPC UA, dan AML digunakan sebagai teknologi utama. *Sequence diagram* juga digunakan untuk merepresentasikan arus informasi antar AAS. Penelitian ini berfokus pada *Adaptable Factories* (AF). DT memiliki kemampuan untuk mengontrol aset fisik. Penelitian serupa juga (Tantik & Anderl, 2017) memanfaatkan robot dan membuat Web-UI berbasis Flask sebagai DT statis. Untuk berkomunikasi antara AAS, ZeroMQ, RESTful, dan HTTP digunakan sebagai teknologi utama. ERD juga digunakan untuk merepresentasikan arus informasi antar AAS. Penelitian ini juga berfokus pada AF. Penelitian serupa (Ye & Hong, 2018) menggunakan 2 robot & 2 konveyor dan membuat GUI berbasis Java sebagai DT statis dan memiliki kemampuan untuk mengontrol aset fisik. Untuk berkomunikasi antar AAS, OPC UA & AML digunakan sebagai teknologi utama. *Sequence diagram* juga digunakan untuk merepresentasikan arus informasi antar AAS. Penelitian ini juga berfokus pada AF.

Penelitian lain (Ye et al., 2018) menggunakan 1 lengan robot & 1 *belt conveyor* dan membuat *Human-Machine Interface* (HMI) sebagai DT statis. Untuk berkomunikasi antar AAS, OPC UA & AML digunakan sebagai teknologi utama. *Sequence diagram* juga digunakan untuk merepresentasikan arus informasi antar AAS. Penelitian ini berfokus pada AF. DT memiliki kemampuan untuk mengontrol aset fisik. Penelitian ini (Profanter et al., 2021) menggunakan 1 robot & 1 *gripper*. Penelitian ini menggunakan UI berbasis C++ sebagai *digital twin* dan memiliki kemampuan untuk mengontrol aset fisik. Untuk berkomunikasi antar AAS, OPC UA & XML digunakan sebagai teknologi utama. *Class & sequence diagram* juga

digunakan untuk merepresentasikan aliran informasi antar AAS. Penelitian ini berfokus pada AF sebagai skenario industri 4.0 yang akan dikembangkan. Penelitian lain (Platenius-Mohr et al., 2019) memfokuskan AF sebagai skenario industri 4.0 dengan 1 motor dan 1 *drive* digunakan sebagai aset fisik. OPC UA, NodeRED, dan AML digunakan sebagai alat komunikasi utama. Web UI dibuat sebagai DT statis dan memiliki kemampuan untuk mengontrol aset fisik. *Flowchart* digunakan sebagai representasi informasi.

Penelitian lain yang berfokus pada skenario *Transparency and Adaptability of Deliver Products* (TAP) industri 4.0 (K. T. Park et al., 2021) menggunakan Proses Perakitan & *Thermosetting of Door Trims* sebagai aset fisik dan membuat *Virtual Reality* (VR) sebagai DT dinamis dan tidak memiliki kemampuan untuk mengontrol aset fisik. Untuk berkomunikasi antara AAS, RESTful API, dan XML digunakan sebagai teknologi utama. *Sequence diagram* juga digunakan untuk merepresentasikan arus informasi antar AAS. Penelitian serupa yang berfokus pada TAP dilakukan oleh (Schroeder et al., 2016). *Industrial Valve* digunakan sebagai aset fisik. FIWARE dan AML digunakan sebagai alat komunikasi utama. UI Web dibuat sebagai DT statis dan tidak memiliki kemampuan untuk mengontrol aset fisik. *Use case diagram* untuk merepresentasikan informasi. Penelitian serupa tentang TAP juga dilakukan oleh (Cavalieri & Salafia, 2020b) yang memanfaatkan *Motor controller* pada mesin perakitan sebagai aset fisik. DT tidak dibuat sehingga tidak memiliki kemampuan untuk mengontrol aset fisik. Untuk berkomunikasi antara AAS, OPC UA dan XML digunakan. *Class diagram* digunakan untuk merepresentasikan arus informasi antar AAS. Penelitian lain juga menerapkan TAP sebagai skenario industri 4.0 (Platenius-Mohr et al., 2020) yang menggunakan 1 motor & 1 *drive*. Penelitian ini tidak sampai pada pembangunan DT. Untuk berkomunikasi antara AAS, REST API, & JSON digunakan sebagai teknologi utama. *Sequence diagram* juga digunakan untuk mewakili aliran informasi antara AAS.

Penelitian lain (Steindl et al., 2020) berfokus pada SP2 sebagai skenario industri 4.0. *Bulk container* untuk sistem energi digunakan sebagai aset fisik. REST API, HTTP, dan JSON digunakan sebagai alat komunikasi utama. *Docker image* dibuat sebagai DT statis dan memiliki kemampuan untuk mengontrol aset fisik. *Sequence diagram* digunakan sebagai representasi informasi. Penelitian ini (Cavalieri & Salafia, 2020a) berfokus pada pengembangan *automatic predictive maintenance*. Penelitian ini menggunakan *Milling machine* sebagai aset fisik dan membuat Azure UI sebagai DT statis dan memiliki kemampuan untuk mengontrol aset fisik. Untuk berkomunikasi antar AAS, Zigbee dikembangkan sebagai teknologi utama. *Use case diagram* digunakan untuk merepresentasikan arus informasi antar AAS. Penelitian ini berfokus pada *Value-based Service* (VBS) sebagai skenario industri 4.0 yang akan dikembangkan.

### 2.3 Kebaruan Penelitian

Dari 20 penelitian sebelumnya yang telah diulas, perbandingan dilakukan dengan penelitian ini sehingga kebaruan atau kontribusi akan mudah direpresentasikan. State-of-the-art ditunjukkan pada Tabel 2. 3. dengan membandingkan beberapa kriteria penting yaitu aset fisik yang digunakan, jenis protokol komunikasi dan pertukaran data, bentuk DT yang dibuat, representasi informasi, skenario Industri 4.0 yang dikembangkan, kemampuan kontrol DT, dan kemampuan optimisasi DT.

*State-of-the-art* menunjukkan bahwa kajian-kajian sebelumnya tentang penerapan AAS dalam mewujudkan DT sangat terbatas pada aspek visualisasi yang hanya menampilkan model statis daripada model dinamis. Ini adalah kelemahan utama karena salah satu fungsi utama dari DT adalah untuk meningkatkan monitoring. Selain itu, mereka hanya mempertimbangkan beberapa aset tertentu dan aset tidak bergerak. Dalam praktiknya akan banyak aset bervariasi. Penelitian ini menggunakan 5 aset utama di gudang termasuk aset bergerak dan mengusulkan submodel serta jaringan komunikasi antar aset. Dalam perspektif industri 4.0, skenario ini disebut *Self-organizing Adaptive Logistics* (SAL). Sebagian besar penelitian sebelumnya hanya fokus pada *Adaptable Factory* (AF) untuk tujuan *Plug-and-Produce*. Terlebih lagi masih belum ada publikasi tentang pemanfaatan teknologi simulasi berbasis 3D untuk mewujudkan DT menggunakan framework AAS khususnya dengan software Flexsim. Beberapa penelitian sebelumnya juga tidak mempertimbangkan kemampuan untuk mengontrol aset mereka dari *digital twin*. Ini juga merupakan kelemahan karena DT hanya memiliki komunikasi satu aliran daripada komunikasi dua aliran (*bidirectional*). Faktanya, tidak ada satu publikasi pun yang memungkinkan DT memiliki kemampuan optimisasi. Oleh karena itu, penelitian ini mengusulkan sebuah konsep dengan menjadikan DT bersifat dinamis dan memberikan kemampuan optimisasi serta pengontrolan aset secara langsung melalui DT. Penelitian ini akan mengisi kesenjangan pengembangan DT tersebut. Dengan menggunakan simulasi, banyak keuntungan diperoleh dan didemonstrasikan dalam penelitian ini. Meskipun penelitian ini tidak sampai pada tahap implementasi, tetapi penelitian ini berbeda dengan penelitian lain dan memberikan konsep baru untuk mengembangkan DT berdasarkan framework AAS.

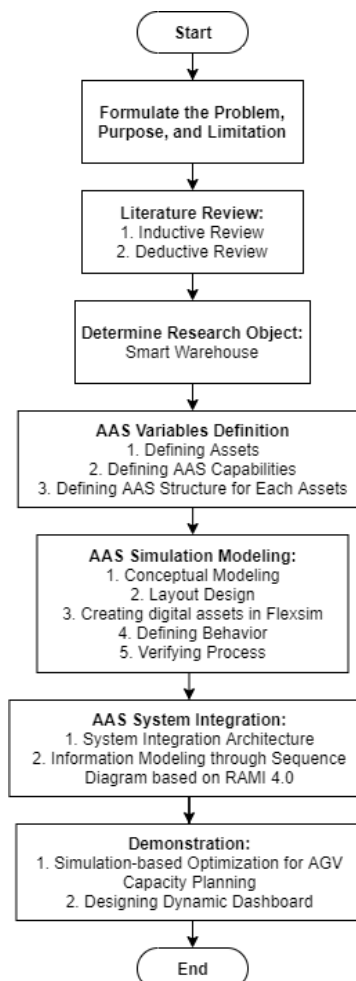
Tabel 2. 3 *State of the Art*

Peneliti	Physical Assets	Communication & Data Exchange Tool	Digital Twin	Information Representation	Industry 4.0 Scenario	Control Ability	Optimization Ability
(Ye et al., 2020)	3 Robots & 1 Turntable	OPC UA & AML	Web UI (Static)	Sequence Diagram	Adaptable Factories (AF)	✓	-
(Marcon et al., 2019)	Operator with Smart Jacket	OPC UA, NodeRED on MCU, MQTT	Human-Machine Interface (HMI) (Static)	Block Diagram	Operator Support in Production (OSP)	✓	-
(Seif et al., 2019)	Assembly Process with Robots	REST API, OPC UA, CSV, JSON	HMI (Static)	Flowchart	AF	✓	-
(Assadi et al., 2020)	Machine and Operator	JSON, MQTT	HMI (Static)	Block Diagram	OSP	✓	-
(Ye & Hong, 2019)	2 Robots & 2 Conveyors	OPC UA & AML	Web UI (Static)	Sequence Diagram	AF	✓	-
(Muralidharan et al., 2020)	Air quality sensor	Kubernetes & Things Description (TD)/JSON	Docker Image (Static)	Flowchart	Smart Product Development for Smart Production (SP2)	-	-
(Tantik & Anderl, 2017)	1 Robot arm	ZeroMQ, RESTful, HTTP	Flask-based Web UI (Static)	ERD	AF	✓	-
(Ye & Hong, 2018)	2 Robots & 2 Conveyors	OPC UA & AML	Java-based GUI (Static)	Sequence Diagram	AF	✓	-
(Ye et al., 2018)	1 Robot arm & 1 belt conveyor	OPC UA & AML	HMI (Static)	Sequence Diagram	AF	✓	-
(Profanter et al., 2021)	1 Robot and 1 Gripper	OPC UA & XML	C++-based UI (Static)	Class & Sequence Diagram	AF	✓	-
(Platenius-Mohr et al., 2020)	1 Motor and 1 Drive	REST API, JSON	-	Sequence Diagram	Transparency and Adaptability of delivered Products (TAP)	-	-
(Toro et al., 2020)	Robot Station	REST API, OPC UA, CSV, JSON	GUI (Static)	Flowchart	AF	✓	-
(Haße et al., 2020)	Remote Control in Forklift Operator	REST API, MQTT	GUI (Static)	Flowchart	OSP	-	-
(Arm et al., 2021)	Assembly Process of Plastic Production	MQTT, JSON	SimPy (Simulation in State Transition Diagram/Static)	Sequence Diagram	SP2	-	-
(Platenius-Mohr et al., 2019)	1 Motor and 1 Drive	OPC UA, NodeRED, AML	Web UI (Static)	Flowchart	AF	✓	-
(K. T. Park et al., 2021)	Assembly & Thermosetting of Door Trims	RESTful API, XML	Virtual Reality (Dynamic)	Sequence Diagram	TAP	-	-
(Steindl et al., 2020)	Bulk Container for Energy System	REST API, HTTP, JSON	Docker Image (Static)	Sequence Diagram	SP2	✓	-
(Cavaliere & Salafia, 2020a)	Milling Machine	OPC UA, Zigbee	Azure UI (Static)	Use Case Diagram	Value-based Services (VBS)	✓	-
(Schroeder et al., 2016)	Industrial Valve	FIWARE, AML	Web UI (Static)	Use Case Diagram	TAP	-	-
(Cavaliere & Salafia, 2020b)	Motor Controller in Assembly Machine	OPC UA, XML	-	Class Diagram	TAP	-	-
<b>Peneliti</b>	<b>Warehouse</b>	<b>OPC UA &amp; AML</b>	<b>3D Simulation (Dynamic)</b>	<b>Sequence Diagram</b>	<b>Self-organising Adaptive Logistics (SAL)</b>	✓	✓

## BAB III

### METODE PENELITIAN

Penelitian ini dilakukan dengan 4 tahapan utama. Pertama *AAS variables definition*, mengidentifikasi aset dan data yang ingin dipertukarkan antar AAS termasuk perangkat IoT, serta mendefinisikan kapabilitas dan struktur AAS. Kedua, *AAS simulation modeling* dengan software Flexsim. Ketiga, *AAS system integration*. Dan terakhir demonstrasi, menunjukkan beberapa keuntungan utama menggunakan model simulasi sebagai bagian dari digital twin. Flowchart penelitian ditunjukkan pada Gambar 3.1 dan informasi rinci dijelaskan di bawah ini.

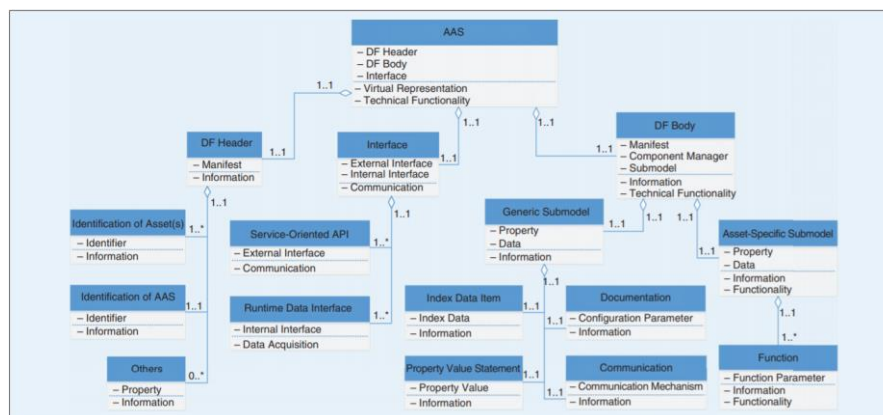


Gambar 3. 1 *Flowchart* Penelitian

### 3.1. AAS Variables Definition

*AAS Variables Definition* mendefinisikan aset yang ingin dibuatkan *administration shellnya*. *Smart warehouse* dipilih sebagai objek penelitian sehingga beberapa aset baru dipertimbangkan. Karena penelitian sebelumnya hanya mempertimbangkan aset tidak bergerak seperti robot lengan dan konveyor, maka penelitian ini akan mempertimbangkan aset yang lebih komprehensif yang tersedia di *Smart Warehouse*. Salah satu gudang pintar di Taipei diamati secara langsung untuk mengidentifikasi aset utama dalam operasinya.

Setelah semua aset diidentifikasi, *AAS Capabilities* dirancang. Ini menentukan data apa yang akan dikumpulkan dan umpan balik apa yang akan diberikan, termasuk perangkat IoT yang akan digunakan. Untuk menentukan struktur AAS untuk setiap aset, *template* AAS diusulkan oleh (Ye & Hong, 2019) memberikan cakupan yang komprehensif seperti yang ditunjukkan pada Gambar 3. 2. Penelitian ini akan menggunakan *template* tersebut untuk mendefinisikan struktur AAS dari *DF Header* ke *DF Body*. Karena penelitian ini tidak mencapai tahap implementasi, maka tidak semua submodel diterapkan seperti dokumentasi pada submodel generik.



Gambar 3. 2 AAS Definition Template

(Sumber: (Ye & Hong, 2019))

### 3.2. AAS Simulation Modeling

Penelitian ini mengusulkan model simulasi sebagai bagian dari *dynamic* DT. Karena proses gudang bersifat kejadian diskrit, salah satu perangkat lunak *Discrete-Event Simulation* yaitu Flexsim digunakan. Dalam DT, tidak hanya fungsional yang penting tetapi juga visual. Flexsim menggunakan visualisasi tiga dimensi (3D) yang dapat meniru operasi pergudangan. Dalam penelitian ini digunakan Flexsim versi 2021. Tidak semua versi dapat digunakan untuk keperluan DT karena keterbatasan komunikasi terutama untuk Flexsim versi bawah 2020, tidak mendukung OPC UA tetapi mendukung OPC Data Access (DA) dan protokol Modbus.

Dalam pemodelan DT, pendekatan pemodelan simulasi akan berbeda dibandingkan dengan pemodelan simulasi klasik. Model simulasi tidak perlu memenuhi syarat validasi tetapi harus diverifikasi secara struktural dan konseptual. Dengan kata lain, tidak diharuskan meniru perilaku untuk keperluan imitasi tetapi harus meniru struktur sistem. Hal ini karena nantinya model simulasi akan terkoneksi dengan aset fisik secara *real-time* dan bukan tujuan imitasi.

Beberapa tahapan pemodelan simulasi yang dilakukan dijelaskan sebagai berikut:

1. Pemodelan konseptual. Tahap ini akan menghasilkan diagram alur yang terkait dengan urutan proses operasi gudang yang didapat dari hasil observasi secara langsung.
2. Desain gudang. Ukuran *layout*, jarak, dan kapasitas perlu didefinisikan terlebih dahulu secara jelas sehingga pemodelan komputer dapat menyesuaikan.
3. Pemodelan komputer tahap 1, dengan membuat beberapa objek/aset yang dibutuhkan. Objek-objek tersebut harus sama dengan variabel AAS yang didefinisikan sebelumnya.
4. Pemodelan komputer tahap 2, dengan mendefinisikan logika setiap aset. Dua pendekatan digunakan dalam pemodelan yaitu *standard object modeling* untuk mendefinisikan perilaku umum aset dan *process flow* untuk meniru perilaku spesifik.
5. Proses verifikasi. Logika simulasi harus serupa dengan model konseptual yang telah dibuat sebelumnya dan tidak boleh ada kesalahan atau eror dalam komputer.

### **3.3. AAS System Integration**

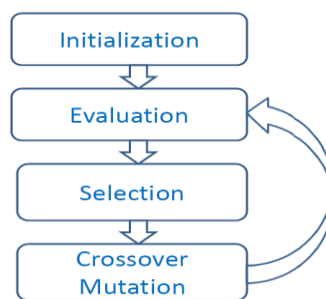
*AAS System Integration* terdiri dari perancangan jaringan sistem komunikasi berdasarkan framework AAS dan pemodelan informasi. Kerangka kerja AAS menjelaskan bagaimana aset dapat berkomunikasi dengan aset lain dan perangkat komunikasi apa yang akan digunakan. Pemodelan informasi akan diwakili oleh *sequence diagram* untuk menunjukkan bagaimana aliran kontrol dan aliran informasi bekerja berdasarkan lapisan RAMI 4.0. Protokol komunikasi dan jenis pertukaran data juga ditentukan berdasarkan kesesuaian dengan perangkat lunak simulasi yang digunakan dan konvensi di komunitas industri 4.0.

### **3.4. Demonstrasi**

Penelitian ini mengusulkan simulasi sebagai bagian dari DT dengan menggunakan perangkat lunak Flexsim sehingga demonstrasi diperlukan untuk menunjukkan keuntungan utama menggunakan konsep yang diusulkan. Dua demonstrasi akan ditampilkan yaitu optimisasi berbasis simulasi dan pembuatan *dashboard* dinamis. Dua fitur ini memenuhi celah dalam penelitian DT yang belum dilakukan sebelumnya.

Dalam simulasi klasik, *What-If analysis* biasanya digunakan untuk mendapatkan hasil yang lebih baik. Namun kurang efektif sehingga saat ini optimisasi diintegrasikan dengan simulasi. Proses optimisasi yang digunakan akan mengikuti model iteratif (Laguna & Martí, 2002). Metaheuristik akan memberikan parameter input ke model simulasi untuk disimulasikan. Hasil simulasi diberikan kembali ke metaheuristik untuk dioptimalkan dan menghasilkan nilai baru. Proses diulang sampai jumlah iterasi tercapai.

Algoritma metaheuristik yang digunakan dalam penelitian ini adalah *Genetic Algorithm* (GA) dengan modul *OptQuest optimizer* (Benjamin, 2021). Mekanisme dari GA ditunjukkan pada Gambar 3.3 (Kordos et al., 2020). Karena *optimizer* tertanam di dalam Flexsim, semua parameter seperti ukuran populasi, tingkat *crossover*, dan tingkat mutasi telah ditetapkan dan tidak bisa diubah. Tidak ada strategi elitisme yang digunakan dalam GA versi orisinal.

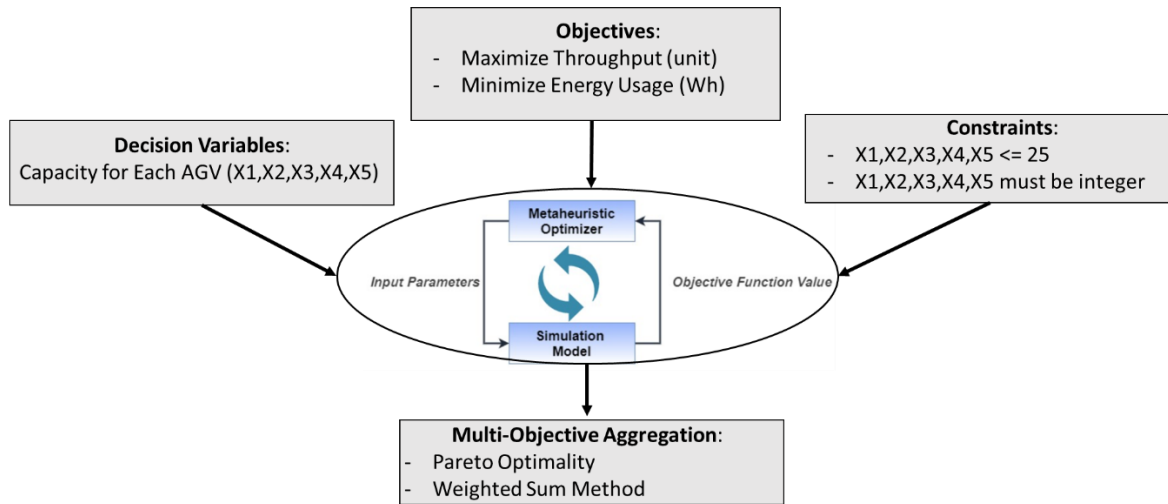


Gambar 3. 3 Mekanisme dari *Genetic Algorithm*

(Sumber: (Kordos et al., 2020))

Skenario optimisasi dalam penelitian ini adalah *AGV capacity planning*. Ide dasarnya adalah menentukan jumlah barang yang harus dibawa untuk setiap AGV sehingga akan memaksimalkan *throughput* dan meminimalkan penggunaan energi listrik secara bersamaan. Masalah ini tidak memiliki bentuk *analytical closed-form* karena sistem yang terlalu kompleks dan banyaknya ketidakpastian sehingga tidak dapat diselesaikan dengan teknik optimisasi klasik atau pemrograman matematika seperti kalkulus atau *integer programming*. Dalam hal ini, simulasi menjadi solusi tepat untuk menghitung fungsi tujuan di bawah ketidakpastian..

*Trade-off* dalam masalah ini adalah semakin banyak barang yang dibawa dalam sekali pengiriman, semakin banyak penggunaan energi yang dibutuhkan untuk AGV karena semakin berat dan semakin lama waktu bongkar-muat yang dibutuhkan. Karena terdapat 2 tujuan, maka dilakukan optimisasi *multi-objective* dengan menggunakan *Pareto Optimality*. Pendekatan ini akan memberikan satu set solusi optimal berdasarkan *Non-dominated Sorting Genetic Algorithm II* (NSGA-II) dan pengguna dapat memilih solusi berdasarkan preferensi mereka. Untuk memilih solusi tunggal, *Weighted Sum Method* digunakan. Kerangka kerja optimisasi berbasis simulasi untuk Perencanaan Kapasitas AGV ditunjukkan pada Gambar 3.4.



Gambar 3. 4 Framework dari Optimisasi AGV Capacity Planning

Karena fungsi tujuan memiliki satuan yang berbeda (throughput dalam satuan unit dan energi dalam Watt hour), maka diperlukan normalisasi terlebih dahulu sebelum weighted sum method diterapkan untuk memilih solusi terbaik. Normalisasi dilakukan dengan menggunakan metode min-max untuk menyatukan skala dari setiap tujuan. Persamaan normalisasi ditunjukkan pada Persamaan 1. Untuk persamaan metode penjumlahan terbobot ditunjukkan pada Persamaan 2. Dimana  $x$  = variabel keputusan,  $w$  = bobot,  $f(x)$  = fungsi tujuan.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (1)$$

$$\begin{aligned} \max \sum_{i=1}^k w_i f_i(x) \\ \text{s.t. } x \in \Omega \\ w_i \geq 0 \\ \forall i = 1, \dots, k \\ \sum_{i=1}^k w_i = 1 \end{aligned} \quad (2)$$

Demonstrasi kedua adalah membuat *dashboard* dinamis. Salah satu keuntungan utama menggunakan perangkat lunak simulasi adalah akurasi perhitungan kinerja untuk setiap objek dalam sistem. Hal itu dapat direkam dan ditunjukkan melalui *dashboard* secara *real-time*. Ini mendukung tujuan dari DT yang dapat memudahkan proses pemantauan dan pengendalian. Beberapa parameter yang dibuat pada dashboard adalah utilisasi AGVs (%), *Total Throughput* untuk setiap jenis produk (Unit), *Forklift State* (%), dan *Operator State* (%). Namun, parameter ini dapat disesuaikan berdasarkan preferensi pengguna.

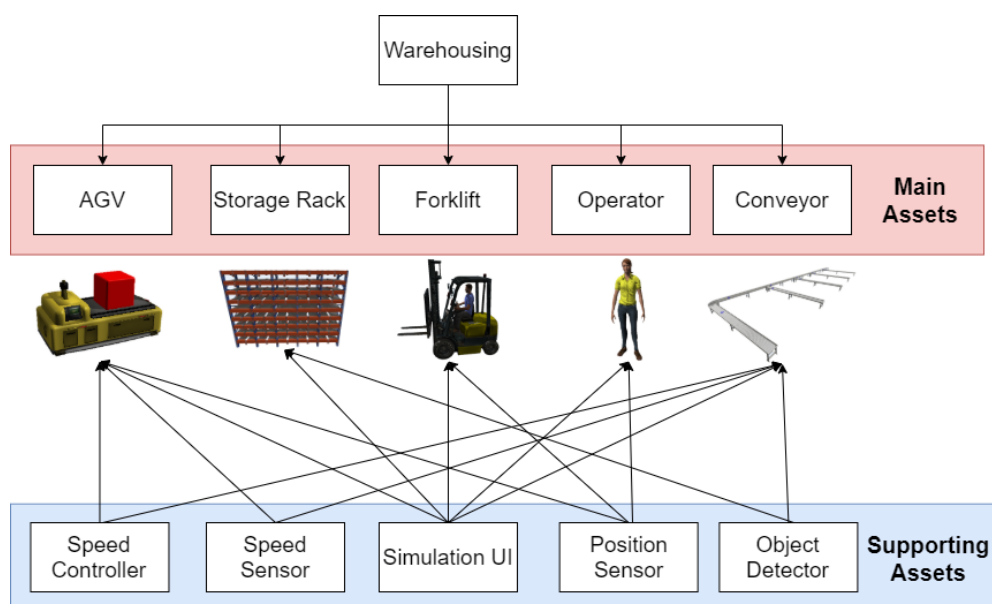
## BAB IV

### HASIL DAN PEMBAHASAN

Pada bab ini akan dipaparkan hasil dan pembahasan berdasarkan urutan yang telah dijabarkan pada metode penelitian. Pertama, aset-aset yang dipertimbangkan dalam *smart warehouse* akan dibahas secara detail serta hasil pendefinisian AAS berdasarkan kapabilitas dan strukturnya. Kedua, hasil model simulasi akan disajikan. Ketiga, analisis integrasi sistem akan ditampilkan untuk mencapai koneksi dua arah antara aset fisik dan aset digital. Terakhir, optimisasi berbasis simulasi dan *dashboard* dinamis akan dibahas.

#### 4.1. Aset-Aset di *Smart Warehouse*

*Smart warehouse* memiliki banyak aset untuk mendukung proses bisnisnya. Melalui observasi dan wawancara langsung, terdapat 5 aset utama yang dipertimbangkan yaitu *AGV*, *Operator*, *Conveyor*, *Forklift*, dan *Storage Rack (pod)*. Untuk merekam data dari aset-aset tersebut dan memberikan umpan balik kembali, diperlukan beberapa sensor dan *controller* sebagai aset pendukung seperti yang ditunjukkan pada Gambar 4.1. Panah dari aset pendukung ke aset utama menggambarkan hubungan penggunaan *controller/sensor* dengan aset. *User Interface (UI)* adalah perangkat lunak Flexsim sebagai model digital.



Gambar 4. 1 Aset-Aset di *Smart Warehouse*

*Sensing capability* dan *controlling capability* antara aset digital dan aset fisik dijelaskan dalam *AAS Capability* seperti yang ditunjukkan pada Tabel 4.1. Karena penelitian ini menggunakan *packaged simulation software*, maka tidak semua aset memiliki kemampuan untuk dikendalikan karena keterbatasan software. Inilah alasan utama mengapa penelitian sebelumnya membuat HMI/GUI untuk meningkatkan fleksibilitas kontrol. Namun, kemampuan utama masih bisa ditangkap. Integrasi antara HMI/GUI dengan *packaged software* juga dapat dilakukan untuk meningkatkan efisiensi dan fleksibilitas.

Tabel 4. 1 *AAS Capability*

<b>Main Asset</b>	<b><i>Sensing Capability</i> (from physical to digital asset)</b>	<b><i>Controlling Capability</i> (from digital to physical asset)</b>
AGV	1. Position from Position Sensor 2. Speed from Speed Sensor 3. Capacity from Steering Control System	1. Speed to Speed Controller 2. Capacity to Steering Control System
Operator	1. Position from Position Sensor	-
Conveyor	1. Speed from Speed Sensor 2. Delivery Path based on SemProM from Object Detector	1. Speed to Speed Controller
Forklift	1. Position from Position Sensor	-
Rack	1. Cell-based Position based on SemProM from Object Detector	-

Dari Tabel 4.1, dapat diketahui bahwa AGV akan mengirimkan data posisi, data kecepatan, dan data kapasitas ke *digital twin*. Data posisi akan berbentuk koordinat (x,y) sehingga *digital twin* dapat menangkap pergerakan AGV dan dapat divisualisasikan secara *real-time*. Konsep ini juga akan diterapkan pada aset *Forklift* dan *Operator* yang merupakan aset bergerak. Bagi *Operator*, data posisi dapat diperoleh dengan *advanced tag* yang disematkan pada *smart jacket* yang akan digunakan oleh *operator*. Data kecepatan akan merekam pergerakan kecepatan AGV dengan satuan meter/detik. Data kapasitas tergantung pada program *steering control system* yang mencerminkan berapa banyak barang yang dapat dibawa per pengiriman oleh AGV. *Steering control system* ini juga memiliki kontrol lain untuk AGV seperti strategi rute dan pengambilan barang. Untuk kemampuan *controlling*, terlihat bahwa *digital twin* harus mampu mengendalikan kecepatan dan kapasitas AGV. Hal ini sesuai dengan kapabilitas *software* Flexsim untuk mengontrol aset fisik.

Konveyor juga memiliki sensor lain selain sensor kecepatan yaitu sensor untuk mengenali jalur pergerakan produk di atas konveyor berdasarkan *Semantic Product Memory* (SemProM). Karena gudang akan memiliki beberapa jenis produk, konsep SemProM akan diadopsi untuk mengenali berbagai jenis produk dan pendeteksi objek akan mengenalinya. Detektor objek akan diterapkan di setiap persimpangan konveyor. Informasi ini sangat penting karena DT perlu mengenali jalur mana dalam konveyor yang akan dilalui oleh setiap jenis produk. Konsep ini juga akan diterapkan pada *Cell-based Position storage rack* karena DT perlu mengenali di sel mana suatu jenis produk masuk dan keluar. Detektor objek akan diterapkan di setiap sel rak dan ketika produk masuk, data dikirim ke model digital. Konsep ini juga harus diterapkan di setiap titik awal dan titik akhir proses sehingga model digital dapat mengenali produk yang masuk dan keluar.

Setelah menentukan kapabilitas AAS, AAS untuk setiap aset perlu didefinisikan berdasarkan struktur AAS. Nantinya, struktur ini akan diterapkan di *Automation Markup Language* (AML) saat tahap implementasi dilakukan. Dalam penelitian ini, template AAS yang diusulkan oleh (Ye & Hong, 2019) digunakan. Hasil definisi AAS ditunjukkan pada Tabel 4. 2. Struktur AAS dapat dibagi menjadi *DF header* dan *DF body* berdasarkan konsep kerangka DF yang mengacu pada standar *International Electrotechnical Commission* (IEC). 62832 (Adolphs et al., 2016). *DF header* menjelaskan daftar properti yang sesuai dengan I4.0 atau biasanya disebut manifest dan *DF body* berisi *component manager*.

*DF header* terdiri dari ID aset untuk membedakan setiap jenis aset, ID AAS untuk membedakan setiap AAS dalam satu keluarga aset yang sama, misalnya AGV 1 dan AGV 2 akan memiliki ID AAS yang berbeda meskipun merupakan aset yang sama, Jumlah Submodel Generik, dan Jumlah Submodel *Asset-Specific*. Setiap ID harus unik untuk memastikan bahwa informasi dalam submodel dapat ditempatkan dengan tepat dan dikomunikasikan antara AAS dengan cara semantik yang sama sehingga interoperabilitas dapat direalisasikan.

Submodel generik dalam *DF body* berarti submodel standar dasar yang berlaku untuk berbagai aset. Setiap AAS harus menyertakan submodel generik untuk mewakili informasi wajib tentang aset. Submodel *Index Data Item* (IDI) adalah titik masuk dari semua submodel lain yang berarti bahwa pada dasarnya adalah daftar isi yang memungkinkan pencarian semua submodel secara efisien. Submodel *Property Value Statement* (PVS) berisi metadata yang memungkinkan pengguna untuk menemukan atribut aset seperti nama, versi, dan juga dokumentasi. Beberapa properti ini dapat berubah sepanjang siklus hidup aset. Oleh karena itu, submodel ini harus di-upgrade secara berkala. Submodel *Communication* (CM) menyimpan informasi tentang kemampuan komunikasi atau protokol komunikasi aset.

Ketika submodel generik menginformasikan data statis, sebaliknya submodel *Asset-Specific* menginformasikan data dinamis yang berisi informasi spesifik tentang fungsi aset. Aset yang berbeda memerlukan submodel yang berbeda untuk mencerminkan fungsi khusus. Menggunakan cara ini, dengan mempertimbangkan keragaman dan kompleksitas setiap fungsi aset, jumlah submodel *Asset-Specific* berbeda. Misalnya, AGV memiliki submodel kecepatan dengan nilai *real* tetapi operator tidak memiliki submodel ini. Operator akan memiliki submodel posisi dengan nilai *real* karena model digital tidak perlu mengetahui kecepatan operator tetapi perlu mengetahui posisi operator. Penentuan ini bergantung pada bagaimana aset fisik ingin direfleksikan dalam *digital twin*.

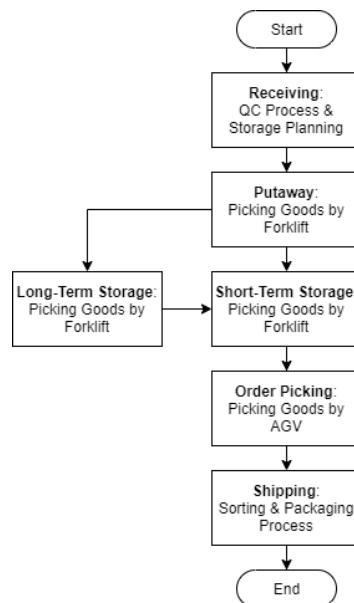
Tabel 4. 2 Pendefinisian AAS berdasarkan Struktur AAS untuk Aset Utama

<i>DF Part</i>	<i>Submodel</i>	<i>AGV</i>	<i>Conveyor</i>	<i>Forklift</i>	<i>Operator</i>	<i>Storage Rack</i>
<i>DF Header</i>	<i>ID of Asset (uniquely generated for each asset)</i>	AAA100	AAA200	AAA300	AAA400	AAA500
	<i>ID of AAS (uniquely generated for each AAS in the same asset)</i>	AAA101 : AAA105	AAA201	AAA301 : AAA305	AAA401 : AAA420	AAA501 : AAA526
	<i>Number of Generic Submodels</i>	3	3	3	2	3
	<i>Number of Asset-Specific Submodels</i>	3	2	1	1	1
<i>DF Body (Generic)</i>	<i>Index Data Item (IDI)</i>	<i>List of submodels:</i> #1 IDI: <IDI001> #2 PVS: <PVS001> #3 CM: <CM001> #4 Speed: <SP001> #5 Position <PT001> #6 Capacity <CP001>	<i>List of submodels:</i> #1 IDI: <ID002> #2 PVS: <PVS002> #3 CM: <CM002> #4 Speed: <SP002> #5 Delivery Path: <DP002>	<i>List of submodels:</i> #1 IDI: <IDI003> #2 PVS: <PVS003> #3 CM: <CM003> #4 Position: <PT003>	<i>List of submodels:</i> #1 IDI: <IDI004> #2 PVS: <PVS004> #3 CM: <CM004> #4 Position: <PT004>	<i>List of submodels:</i> #1 IDI: <IDI005> #2 PVS: <PVS005> #3 CM: <CM005> #4 Cell-based Position: <CP0005>
	<i>Property Value Statement (PVS)</i>	#1 Asset Name: AGV 1...5 #2 Asset Version: Autonomous Mobile Robots #3 Lifecycle Stage: Operation	#1 Asset Name: Conveyor #2 Asset Version: Belt Conveyor #3 Lifecycle Stage: Operation	#1 Asset Name: Forklift 1...5 #2 Asset Version: Counterbalance #3 Lifecycle Stage: Operation	#1 Asset Name: Worker's name 1...20 #2 Responsibility: Packing	#1 Asset Name: Rack 1...26 #2 Asset Version: Storing #3 Lifecycle Stage: Operation
	<i>Communication (CM)</i>	Wifi, UART, Radio Wave	TSN, UART	OPC UA, Radio Wave	OPC UA, UART	OPC UA, UART
<i>DF Body (Asset-specific)</i>	<i>Asset-Dependent</i>	<b>1# Speed (Real):</b> <b>1.1# Speed Detection</b> [From speed sensor to Flexsim]: 1 m/s <b>1.2# Speed Controlling</b> [From Flexsim to speed controller]: 1.5 m/s <b>2# Position (Real)</b> [From position sensor to Flexsim]: (x=-2.26 m, y=65.50m) <b>3# Capacity (INT):</b> <b>3.1# Capacity Detection</b> [From Steering Control System to Flexsim]: 10 packages <b>3.2# Capacity Controller</b> [From Flexsim to Steering Control System]: 15 packages	<b>1# Speed (Real):</b> <b>1.1# Speed Detection</b> [From speed sensor to Flexsim]: 0.25 m/s <b>1.2# Speed Controlling</b> [From Flexsim to speed controller]: 0.5 m/s <b>2# Delivery Path</b> [From object detector to Flexsim] (Boolean): True/False	<b>1# Position (Real)</b> [From position sensor to Flexsim]: (x=-3.25 m, y=9.60m)	<b>1# Position (Real)</b> [From position sensor to Flexsim]: (x=9.22m, y=89.71m)	<b>1# Cell-based Position</b> [From object detector to Flexsim] (Boolean): True/False

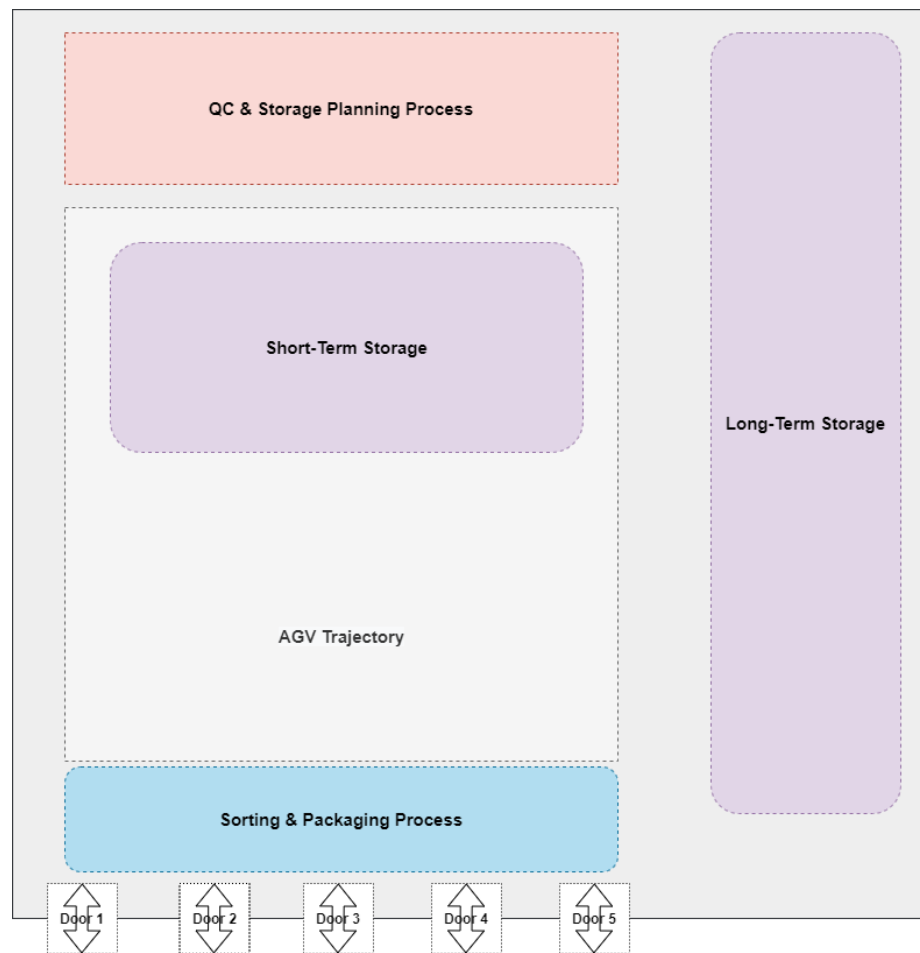
## 4.2. Model Simulasi

*Digital twin* berbasis simulasi harus diawali dengan pemodelan simulasi yang sesuai dengan cakupan yang telah ditetapkan sebelumnya. Jika model simulasi tidak sesuai dengan cakupan, *digital twin* tidak dapat mencerminkan sistem fisik dengan baik. Untuk mencapai hal itu, pemodelan diawali dengan model konseptual dilanjutkan ke pemodelan komputer.

- a) **Model Konseptual.** Model konseptual dimodelkan menggunakan *flowchart* seperti yang ditunjukkan Gambar 4. 2. Desain tata letak gudang ditunjukkan Gambar 4. 3. Alur operasi gudang dimulai saat barang tiba oleh truk, lalu diterima oleh *forklift* dan dibawa ke proses *Quality Control (QC)*. Proses QC tidak menghasilkan produk limbah tetapi mengidentifikasi dan memeriksa kuantitas produk. Kemudian dilakukan *storage planning* untuk menentukan apakah barang tersebut akan disimpan di *long-term storage* atau *short-term storage*. *Short-term storage* hanya menyimpan barang yang akan dikirim di hari yang sama dan *long-term storage* menyimpan barang yang akan dikirim di hari lain. Setelah itu, proses *Putaway* akan dilakukan oleh *forklift* baik itu dari proses QC ke *short-term storage* atau dari *long-term storage* ke *short-term storage*. Ketika waktu *pick-up* tiba, proses *order picking* dilakukan oleh AGV. AGV akan mengambil barang langsung dari *short-term storage*. Barang akan dikirim ke *shipping stage* yang terdiri dari proses sortasi dan pengemasan. *Work in Process* dari proses sortir akan disimpan di rak sementara jika proses pengemasan belum selesai.



Gambar 4. 2 Model Konseptual dari *Smart Warehouse*

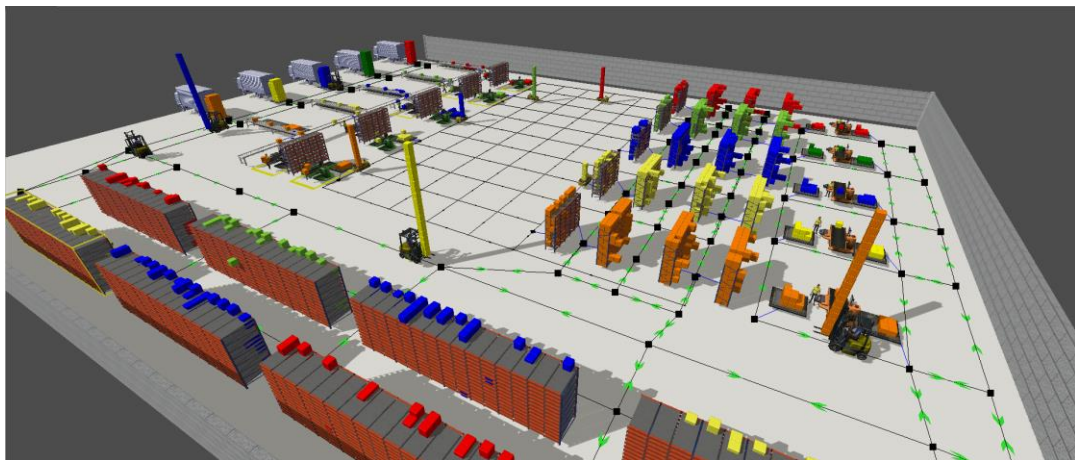


Gambar 4. 3 Tata Letak dari *Smart Warehouse*

- b) **Model Komputer.** Karena tujuan dari pemodelan simulasi adalah untuk membuat DT, maka tidak perlu melakukan *input analysis* seperti simulasi klasik karena nantinya data akan terhubung secara *real-time* dengan aset fisik. Namun, karena penelitian ini akan mendemonstrasikan beberapa keuntungan dari simulasi sebagai bagian dari DT sehingga data input seperti waktu pemrosesan dan kapasitas akan diasumsikan berdasarkan distribusi statistik untuk menangkap ketidakpastian. Data input ditunjukkan pada Tabel 4. 3. Hasil model simulasi ditunjukkan pada Gambar 4. 4 dan Gambar 4. 5. Model simulasi yang telah dibuat kemudian diverifikasi. Hasil verifikasi menunjukkan bahwa model simulasi telah mengikuti logika model konseptual dan tidak terdapat *error* pada model seperti *deadlock* atau bahkan *collision* antara AGV. Secara visual, semua aset digital pada model simulasi telah mengikuti *behaviour* dari sistem nyatanya tanpa ada eror yang terekam pada *software*.

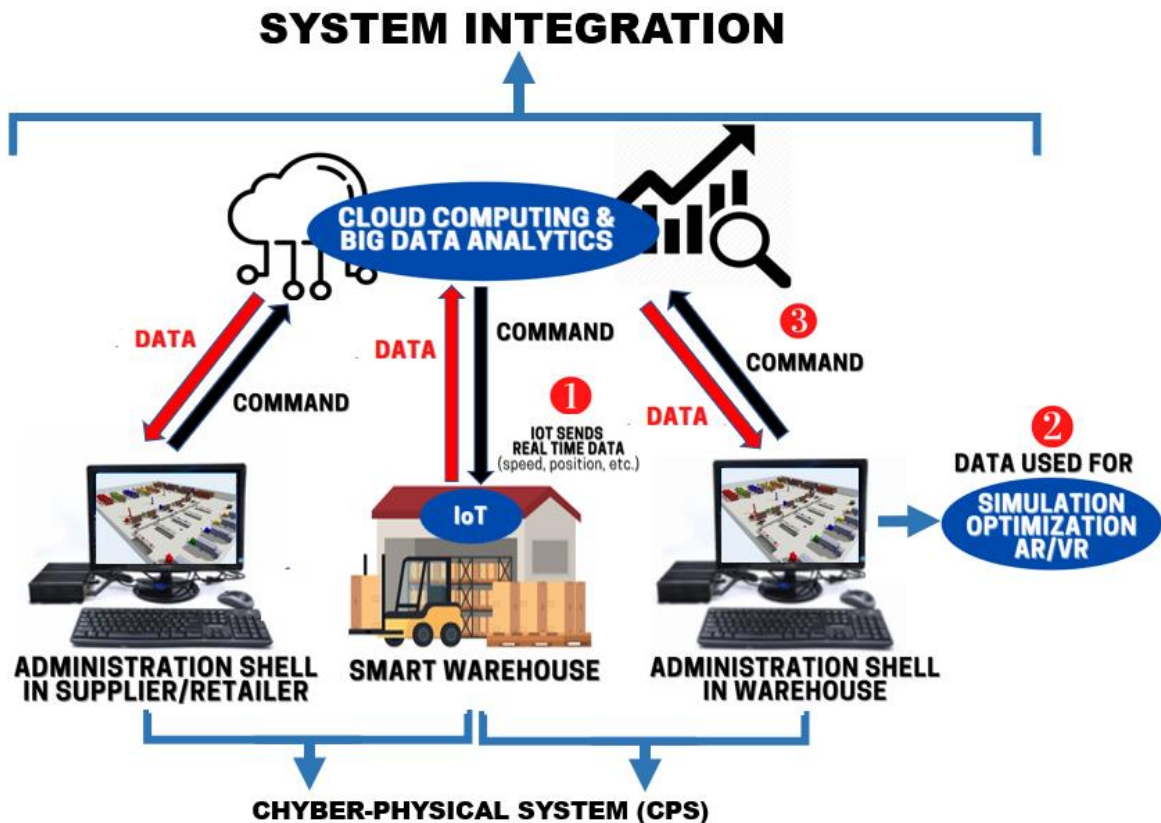
Tabel 4. 3 Data Input

Proses	Data Waktu	Kapasitas
QC & Strategic Planning	uniform(15, 17, getstream(current)) in minute	1
Sorting	uniform(2, 5, getstream(current)) in minute	1
Packaging	uniform(2, 6, getstream(current)) in minute	1
AGV	Speed: 0.5 meter/second	10
Forklift	Speed: 1 meter/second	20
Conveyor	Speed: 0.25 meter/second	Accumulating

Gambar 4. 4 Model Simulasi *Smart Warehouse* dari DepanGambar 4. 5 Model Simulasi *Smart Warehouse* dari Belakang

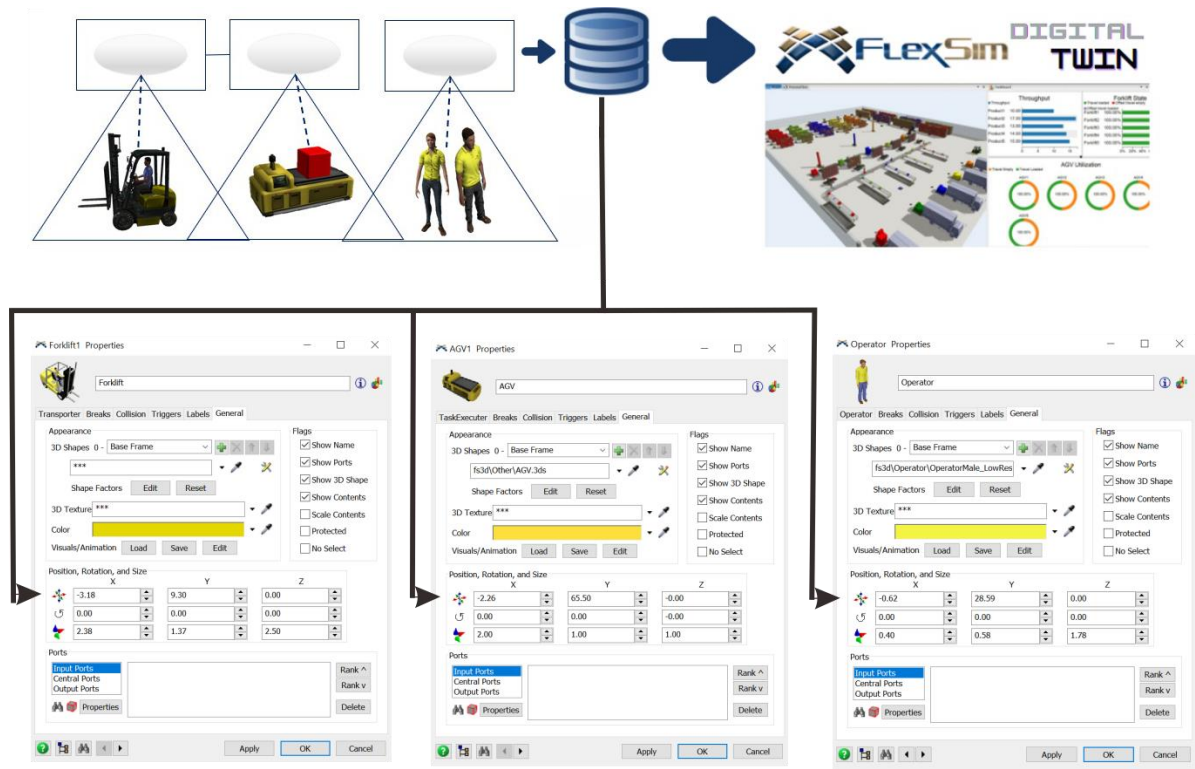
### 4.3. Integrasi Sistem *Digital Twin*

Konsep usulan yang ingin dicapai dalam penelitian ini pada dasarnya adalah integrasi sistem untuk mengintegrasikan teknologi industri 4.0 seperti *IoT*, *Simulation*, *Cloud Computing*, *Big Data Analytics*, dan lain-lain dengan menitikberatkan pada konsep *digital twin* seperti terlihat pada Gambar 4. 6. Secara ringkas, data dari gudang akan direkam oleh *IoT* dan akan dikirim ke model simulasi sebagai model digitalnya melalui *cloud* secara *real-time*. Nantinya, simulasi bisa diperluas menjadi *Augmented Reality* (AR) atau *Virtual Reality* (VR). Dari model simulasi, strategi kontrol yang optimal dapat ditentukan dan dikirim ke gudang pintar melalui *cloud* juga. Data yang ditransmisikan dapat disimpan sebagai big data di *cloud* untuk menerapkan *data analytics* sehingga dapat meningkatkan performa *warehousing* melalui *Machine Learning* (ML), *Deep Learning* (DL), atau *Artificial Intelligence* (AI).

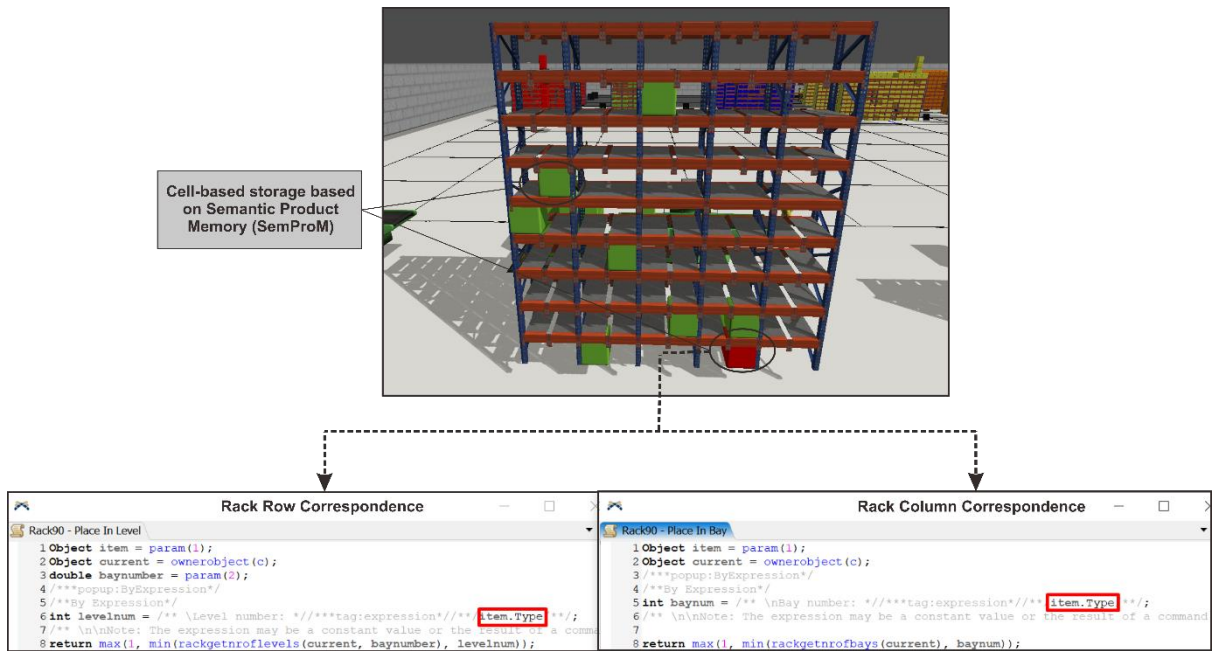


Gambar 4. 6 Gambaran Konsep Umum yang Diusulkan

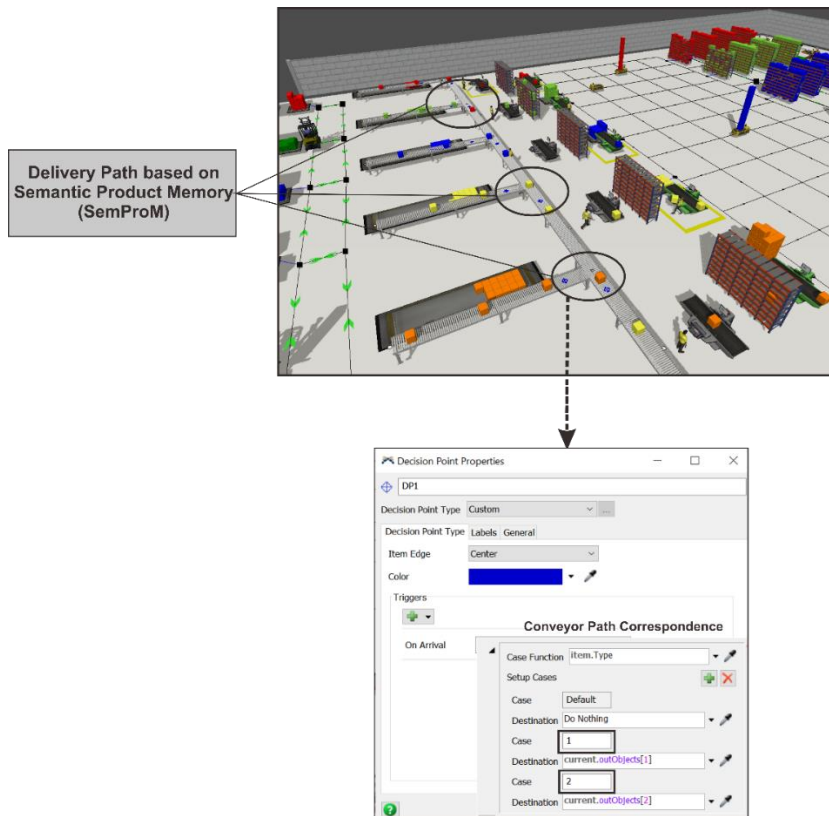
Terkait data yang diperoleh dari sensor khususnya untuk data posisi akan didapat dari *transmitter* di bagian atas dan akan merekam *tag* yang disematkan pada aset bergerak sehingga pergerakannya dapat dikenali dalam *digital twin* seperti terlihat pada Gambar 4. 7. Ini dapat dicapai dengan beberapa teknologi yang sudah ada saat ini seperti *Wi-Fi positioning system* dan RFID long-range. Data ini akan berkorespondensi dengan properti posisi di Flexsim sehingga akan terhubung secara *real-time*. Untuk *SemProM* di rak penyimpanan, untuk mengenali sel mana di rak sebagai tempat masuk keluar barang, hal itu akan sesuai dengan detektor objek yang tertanam di setiap sel seperti yang ditunjukkan pada Gambar 4. 8. Data ini akan sesuai dengan Flexsim di properti rak *level (row)* atau *bay (column)* sehingga dapat terhubung secara *real-time*. Untuk *SemProM* di konveyor, untuk mengenali jalur mana di konveyor yang akan dilalui setiap produk, itu akan sesuai dengan detektor objek yang disematkan di setiap persimpangan konveyor seperti yang ditunjukkan pada Gambar 4. 9. Data ini akan berkorespondensi dengan Flexsim di properti *decision point* sehingga mereka dapat terhubung secara *real-time*.



Gambar 4. 7 Korespondensi AAS untuk Data Posisi



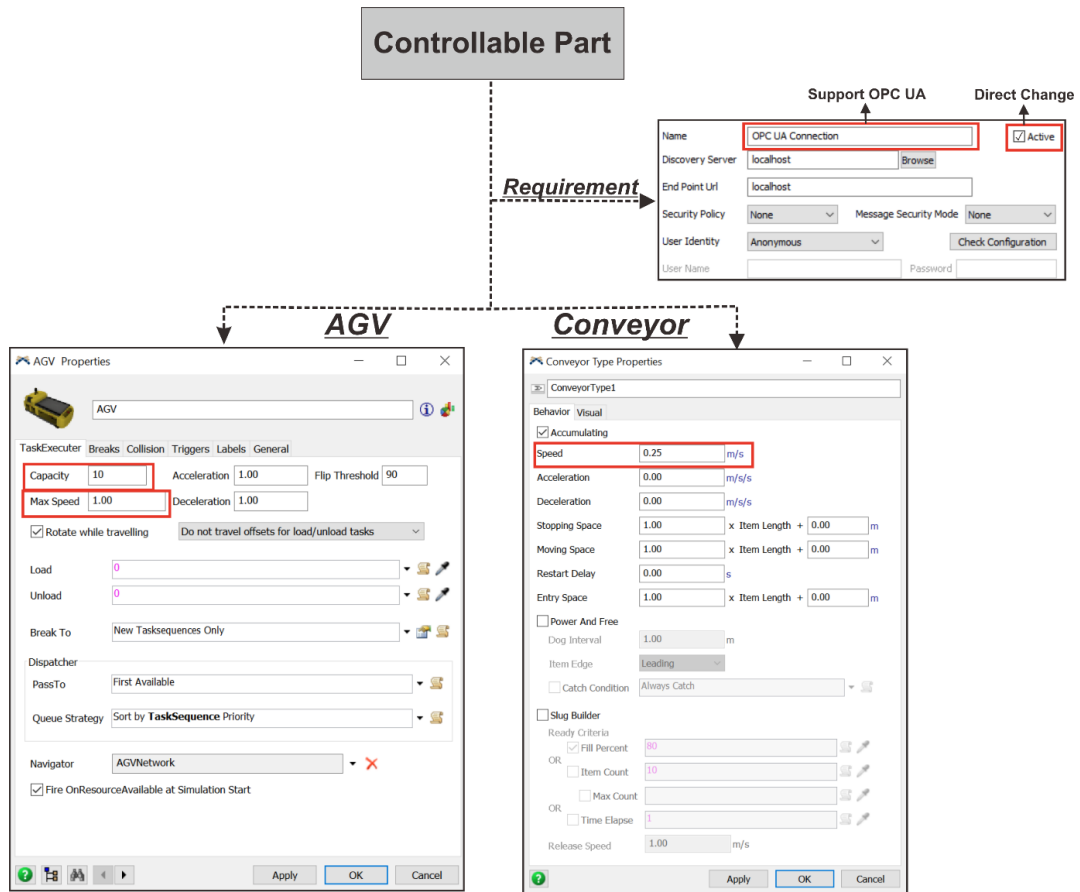
Gambar 4. 8 Korespondensi AAS untuk *SemProM* di *Storage Rack*



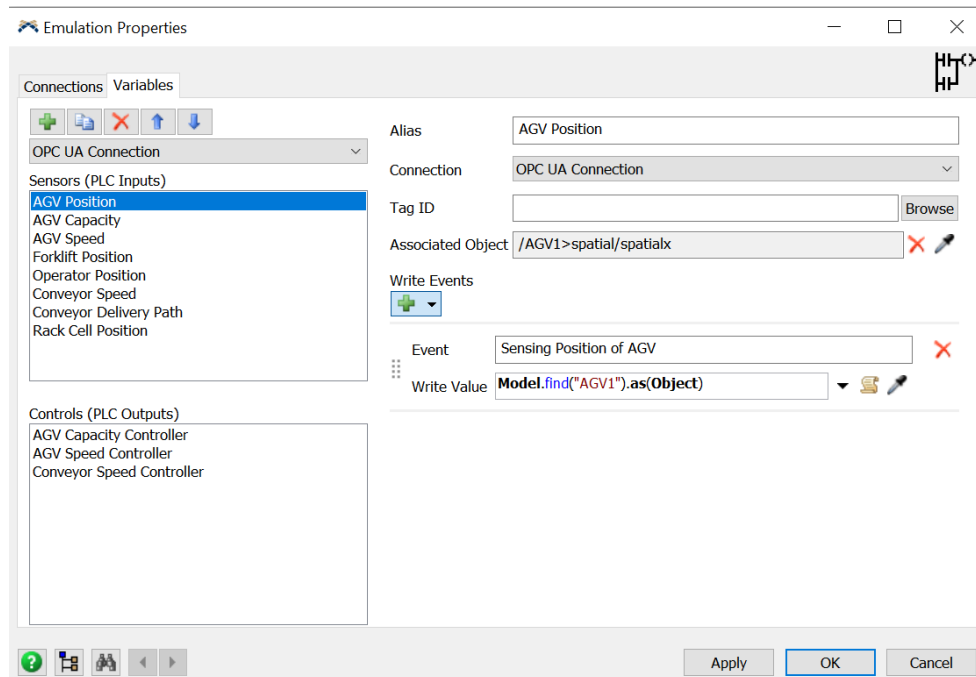
Gambar 4. 9 Korespondensi AAS untuk *SemProM* di *Konveyor*

*Digital twin* seharusnya tidak hanya memungkinkan aliran satu arah dari aset fisik ke aset digital (aliran data), tetapi juga harus memungkinkan aliran dua arah dari aset digital ke aset fisik (aliran kontrol). Penelitian ini memungkinkan DT untuk mengontrol kecepatan AGV, kapasitas AGV, dan kecepatan konveyor seperti yang ditunjukkan pada Gambar 4. 10. Selain itu, poin terpentingnya adalah *connectivity requirement* untuk perangkat lunak simulasi harus saling mendukung. Flexsim memungkinkan koneksi OPC UA berdasarkan *emulation tool* yang dimiliki sehingga aliran kontrol dapat dilakukan dan parameter baru dapat diterapkan dengan *simulated programmable logic controllers (PLC)*.

Ada dua mode dalam *emulation tool* untuk melakukan proses *controlling* yaitu *Active* dan *Inactive*. Perubahan langsung (*Direct change*) harus dicentang sebagai *Active* untuk membuat perubahan secara *real-time*. Fitur ini memungkinkan pengguna untuk mengontrol aset fisik langsung dari aset digital. Jika mode dalam keadaan *Inactive*, hal ini tidak akan membuat perubahan pada aset fisik. Opsi mode *Inactive* dapat digunakan ketika analisis *What-If* atau optimisasi diperlukan. Setelah parameter optimum diketahui, maka dapat diterapkan dengan mode *Active*. Perlu dicatat bahwa tidak semua versi Flexsim dapat digunakan untuk keperluan *digital twin* karena keterbatasan komunikasi terutama untuk Flexsim versi bawah 2020 yang tidak mendukung komunikasi OPC UA tetapi mendukung protokol OPC DA dan Modbus. Konfigurasi sensor dan kontrol juga harus didefinisikan dengan jelas di Flexsim berdasarkan *AAS Capability* seperti yang ditunjukkan pada Gambar 4. 11. Nantinya, jika tahap implementasi ingin dilakukan, *Tag ID* harus sesuai dengan *ID AAS* yang terkait yang terasosiasi dengan objek di Flexsim



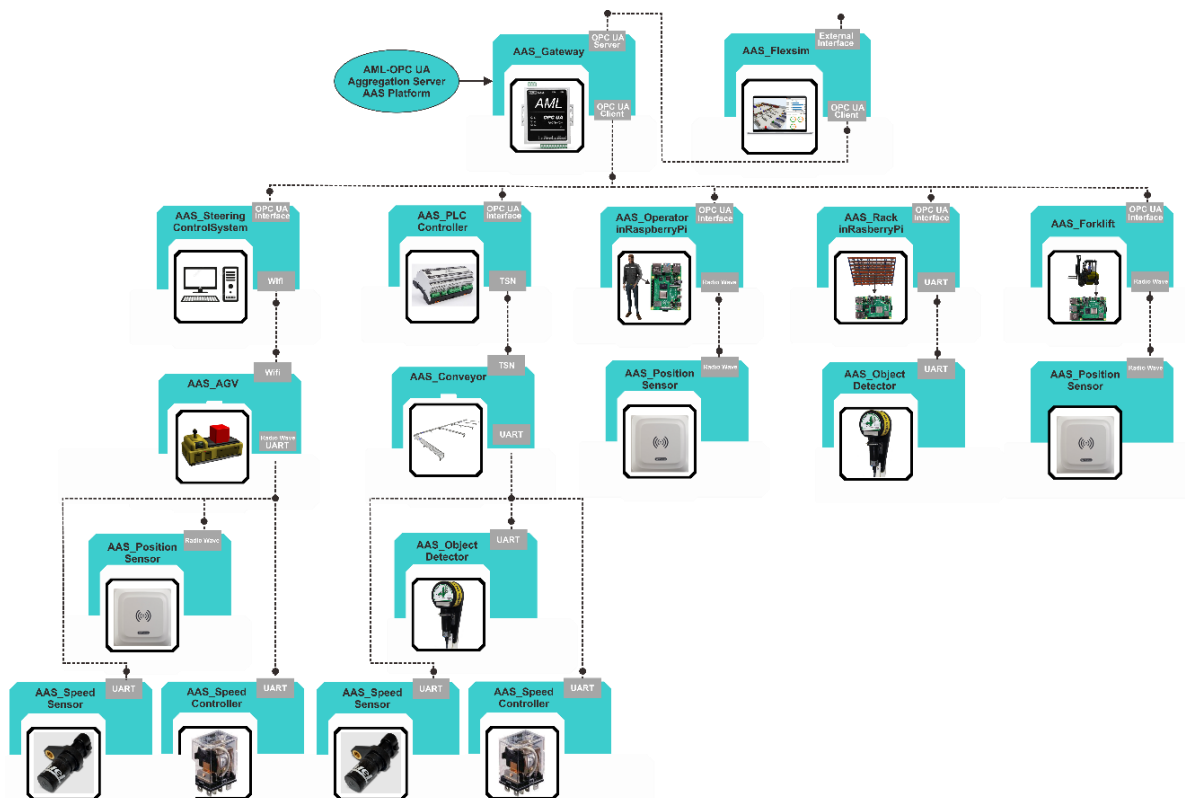
Gambar 4. 10 AAS Correspondence untuk Controllable Part



Gambar 4. 11 Konfigurasi Sensor dan Kontrol di AAS Flexsim

AAS membutuhkan jaringan komunikasi sistem untuk mengimplementasikan *digital twin*. Oleh karena itu, struktur jaringan AAS untuk *smart warehouse* termasuk aset dan *data exchange paths* ditunjukkan pada Gambar 4. 12. Seperti yang dijelaskan sebelumnya, dalam penelitian ini, OPC UA dan AML digunakan sebagai komunikasi utama antara AAS. Dalam prosesnya, AML akan menjawab data apa yang akan dipertukarkan dan OPC UA akan menjawab bagaimana data dikomunikasikan.

Terdapat 3 alasan mengapa OPC UA dan AML dipilih. Pertama, OPC UA telah distandarisasi di IEC 62541 dan AML telah distandarisasi di IEC 62714. Kedua, 2 protokol tersebut kompatibel dengan Flexsim sebagai model digital. Ketiga, AML dan OPC UA telah sesuai dengan konvensi I4.0 dan model RAMI 4.0. Dalam sistem AAS, setiap aset memiliki ID AAS yang unik. Namun, untuk simplifikasi representasi, aset yang sama seperti terdapat 5 AGV di gudang, hanya akan ditunjukkan satu AGV pada Gambar 4. 12 karena AGV lainnya akan memiliki AAS yang sama tetapi memiliki ID yang berbeda. Gambar 4. 12 menunjukkan bagaimana AAS terhubung satu sama lain dan komunikasi apa yang mereka gunakan.

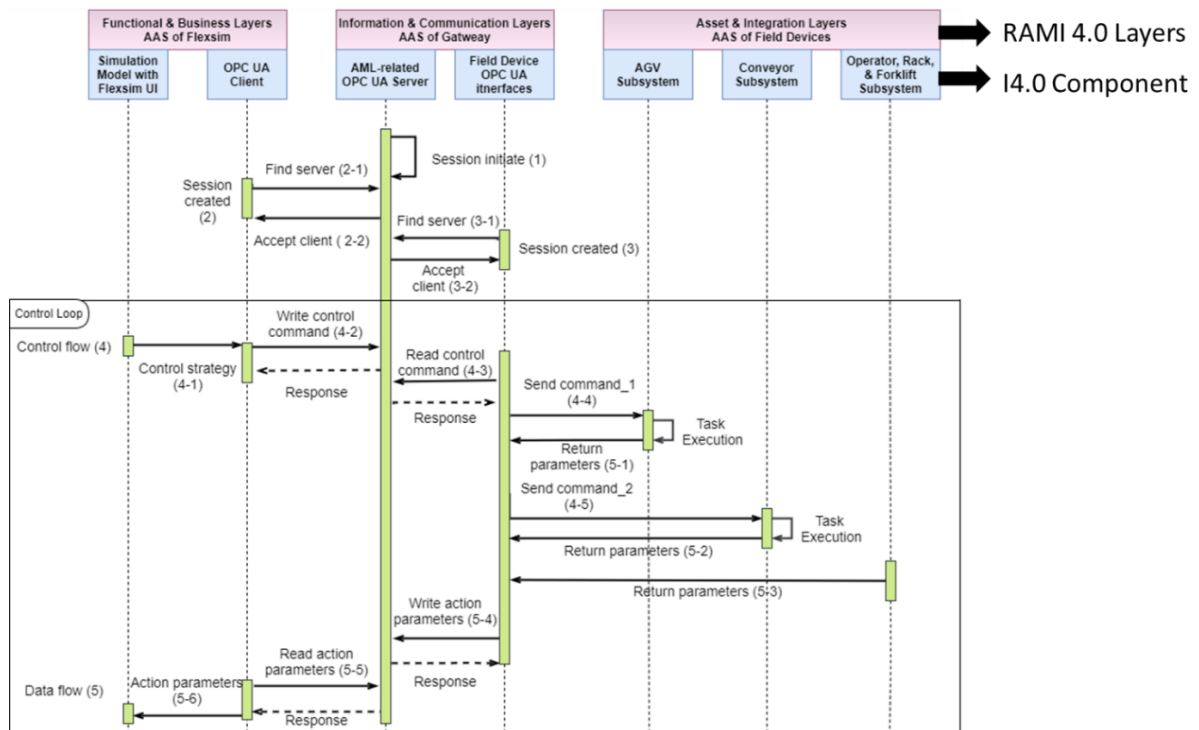


Gambar 4. 12 AAS Framework di Smart Warehouse

Pada subsistem AGV, akan ada sensor kecepatan dan pengontrol kecepatan yang terhubung dengan AGV melalui *Universal Asynchronous Receiver–Transmitter* (UART). Ada juga sensor posisi untuk mengetahui posisi AGV yang terhubung melalui gelombang radio. AGV sendiri akan terhubung dengan *Steering Control System* melalui WiFi. *Steering Control System* memiliki kemampuan untuk mengontrol kecepatan, *picking strategy*, kapasitas AGV, dan penentuan rute.

*Steering Control System* akan terhubung dengan *Gateway* melalui OPC UA. Ada beberapa *gateway* yang bisa digunakan. Salah satunya adalah protokol LM Gateway201-OPC UA. Protokol ini mendukung protokol kabel maupun nirkabel. *Gateway* memiliki dua fungsi yaitu penyimpanan data dalam AML dan pertukaran data melalui OPC UA. Dari *gateway*, akan terhubung dengan AAS Flexsim melalui OPC UA dan Flexsim akan menampilkan kondisi *real-time* melalui tampilan antarmuka eksternal.

Interaksi antara AAS ditunjukkan sebagai *sequence diagram* pada Gambar 4. 13 yang mewakili aliran informasi dinamis dan aliran kontrol melalui jaringan digital AAS dengan mengikuti lapisan RAMI 4.0. Pertama, server OPC UA terkait AML akan diinisialisasi. Kemudian, OPC UA akan membuat sesi dengan klien yang berbeda yaitu Klien OPC UA dari AAS Flexsim dan OPC UA *Interface* dari AAS *Field devices*. Aliran kontrol kemudian dihasilkan oleh *users* melalui AAS Flexsim dan diteruskan ke AAS *Field devices*. Di sini, tugas pergudangan dijalankan sesuai dengan strategi kontrol. Parameter terbaru kemudian diberikan kembali ke AAS Flexsim untuk diperbaharui. Dengan demikian, sistem *kembar digital* beroperasi sebagai *loop*.



Gambar 4. 13 Interaksi antar AAS

#### 4.4. Demonstrasi

Dua keunggulan utama simulasi akan ditunjukkan dalam penelitian ini yaitu optimisasi berbasis simulasi dan pembuatan *dashboard* dinamis. Semua fitur ini telah disematkan dalam perangkat lunak yang digunakan sehingga lebih praktis digunakan oleh pengguna. Dengan adanya *optimizer* ini, memecahkan kesenjangan dari penelitian sebelumnya tentang DT yang tidak dapat melakukan optimisasi dalam sistem DTnya (CyberFactory#1, 2020).

##### 4.4.1 *Simulation-based Optimization untuk AGV Capacity Planning*

AGV memegang peranan penting dalam operasi *smart warehouse*. AGV membawa barang dari bagian penyimpanan ke area pengiriman melalui lintasannya sendiri. Besarnya kapasitas AGV menentukan *total throughput* dan penggunaan energi listrik karena AGV menggunakan listrik dalam pengoperasiannya. Namun, semakin banyak barang yang dibawa, semakin berat AGV bekerja sehingga membutuhkan waktu bongkar muat yang lebih lama (berefek pada *total throughput*) dan penggunaan energi yang lebih banyak. Oleh karena itu,

ada *trade-off* dalam masalah ini. Meskipun ada beberapa fungsi tujuan juga yang dapat dipertimbangkan seperti mengoptimalkan rute AGV dan utilisasi pekerja di gudang, tetapi dalam penelitian ini hanya maksimasi *total throughput* dan minimasi penggunaan energi yang akan dipertimbangkan karena tujuan tersebut mempengaruhi kepuasan pelanggan, keuntungan, dan biaya operasi secara langsung.

Untuk menyelesaikan permasalahan *Multi-Objective Optimization* (MOO), digunakan GA berbasis NSGA-II. Karena sistem yang terlalu kompleks, simulasi membawa optimisasi lebih lanjut dengan mempertimbangkan kondisi yang tidak pasti seperti ketersediaan barang dan waktu proses. Simulasi dijalankan selama 1 hari operasi atau 7 jam atau 25200 detik waktu kerja. Diasumsikan setiap 1 beban barang yang dibawa oleh AGV akan memakan 2 Watt per meter dan jika tidak ada barang yang dibawa hanya membutuhkan 1 Watt per meter. Tidak ada kondisi pengisian baterai yang dipertimbangkan karena hanya disimulasikan selama 7 jam. Kode optimisasi berbasis simulasi yang digunakan ditunjukkan pada Gambar 4. 14. Hasil pencarian GA dalam 100 iterasi ditunjukkan pada Gambar 4. 15 untuk *throughput (unit)* dan Gambar 4. 16 untuk konsumsi energi dalam *Watt hour (Wh)*. Lingkaran ungu menunjukkan solusi kandidat yang disarankan. Himpunan solusi optimal yang tidak terdominasi ditunjukkan pada Tabel 4. 4.

Sebagai analisis pelengkap, perspektif finansial akan ditambahkan untuk mendukung pengambilan keputusan yang lebih komprehensif bagi *decision maker*. Hasil set optimal yang didapat dari NSGA-II dilakukan perhitungan biaya per unit berdasarkan total biaya listrik yang dihasilkan dari seluruh AGV. Diasumsikan golongan tarif listrik yang digunakan sebesar 3,707 NTD per kWh (menurut *Taipower*) sebagai standar kelompok industri. Mengingat konsumsi energi listrik AGV bersifat dinamis, maka pendekatan yang digunakan untuk menghitung biaya per unit adalah dengan mengubah total energi (Wh) dari hasil simulasi menjadi kilo Wh (kWh), kemudian dikalikan dengan biaya tarif listrik, lalu dibagi dengan total output yang didapat. Hasil perhitungan ditunjukkan pada 2 kolom paling kanan di Tabel 4. 4.

Code for Decision Variables

Simulation Experiment Control

Scenarios Performance Measures Experiment Run Optimizer Design Opti

Variables + - X ↑ ↓ Scenarios + - X ← →

	Variable	Scenario 1
X1	MODEL:/AGV1>variables/maxcontent	
X2	MODEL:/AGV2>variables/maxcontent	
X3	MODEL:/AGV3>variables/maxcontent	
X4	MODEL:/AGV4>variables/maxcontent	
X5	MODEL:/AGV5>variables/maxcontent	

Code for Performance Measures

Simulation Experiment Control

Scenarios Performance Measures Experiment Run Optimizer Design Optimizer Run Optimizer Results Advanced

Total Throughput

Label for Y-axis

Unit

Statistic by individual object

Select an object and statistic

Object : /Plant20/Sin1

Statistic : Input

- Total Throughput
- AGV1 Travel Distance
- AGV2 Travel Distance
- AGV3 Travel Distance
- AGV4 Travel Distance
- AGV5 Travel Distance
- AGV1 Travel Loaded
- AGV2 Travel Loaded
- AGV3 Travel Loaded
- AGV4 Travel Loaded
- AGV5 Travel Loaded
- AGV1 Travel Empty
- AGV2 Travel Empty
- AGV3 Travel Empty
- AGV4 Travel Empty
- AGV5 Travel Empty
- AGV1 Loading
- AGV2 Loading
- AGV3 Loading
- AGV4 Loading
- AGV5 Loading
- AGV1 Unloading
- AGV2 Unloading
- AGV3 Unloading
- AGV4 Unloading
- AGV5 Unloading

Code for Objective Functions

Simulation Experiment Control

Scenarios Performance Measures Experiment Run Optimizer Design Optimizer Run Optimizer Results Advanced

Variables

Variable	Type	Lower Bound	Upper Bound	Step	Group
X1	Integer	1	25	N/A	N/A
X2	Integer	1	25	N/A	N/A
X3	Integer	1	25	N/A	N/A
X4	Integer	1	25	N/A	N/A
X5	Integer	1	25	N/A	N/A

Constraints

Objectives

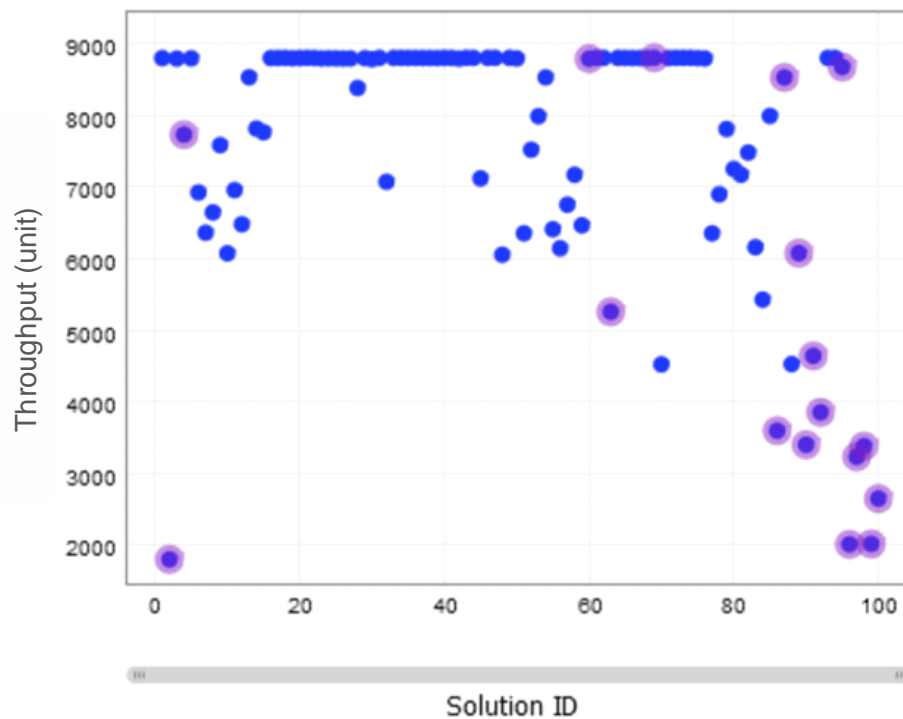
Objective	Function	Direction
Throughput (ur)	[Total Throughput]	Maximize
Energy Consum	((([AGV1 Travel Distance]*([AGV1 Travel Loaded]+[AGV1 Loading]+[AGV1 Unloading]))*[X1]*2/100)+([AGV1 Travel Distance]*[AGV1 Travel Empty]*1/100))+	Minimize

Search Mode:

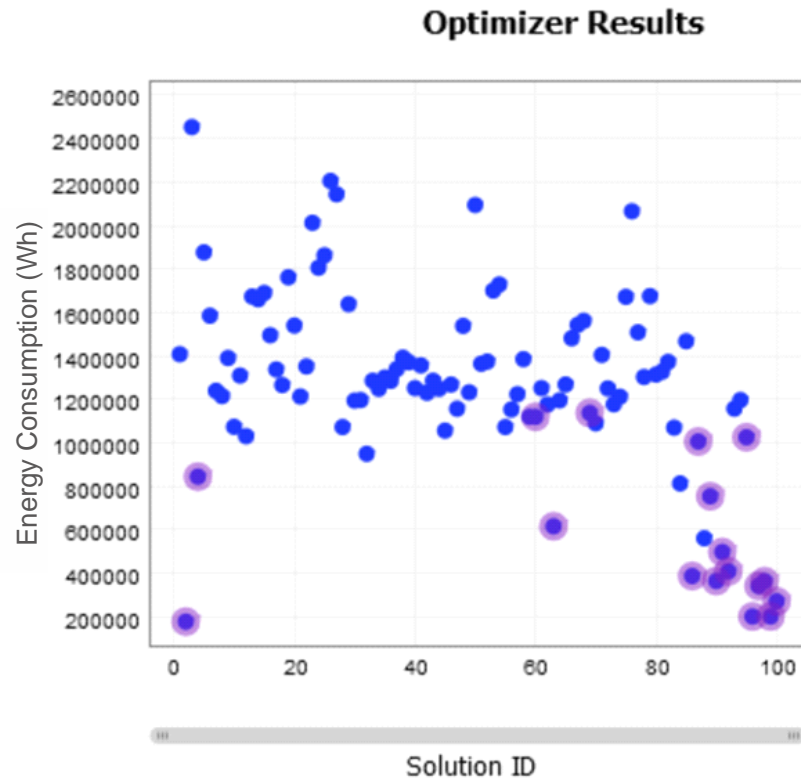
- Single
- Weighted
- Pattern

Gambar 4. 14 Kode untuk Simulation-based Optimization

Optimizer Results



Gambar 4. 15 Hasil Optimisasi untuk Maksimasi Throughput



Gambar 4. 16 Hasil Optimisasi untuk Minimasi *Energy Consumption*

Tabel 4. 4 Solusi Optimal *Non-Dominated* melalui *Pareto Optimality*

Solution ID	Throughput (unit)	Energy Consumption (Wh)	AGV 1 (X1)	AGV 2 (X2)	AGV 3 (X3)	AGV 4 (X4)	AGV 5 (X5)	Total Cost (NTD)	Cost/Unit
2	1797	<b>176146,08</b>	1	1	1	1	1	652,974	0,363
4	7726	843002,35	7	7	7	7	7	3125,010	0,404
63	5253	614653,02	6	11	1	6	1	2278,519	0,434
60	8786	1118286,33	10	11	9	9	10	4145,487	0,472
69	<b>8798</b>	1136233,01	10	10	12	10	11	4212,016	0,479
86	3593	386029,79	2	3	4	4	1	1431,012	0,398
87	8525	1004878,19	9	9	7	9	9	3725,083	0,437
90	3398	362968,99	1	3	4	3	2	1345,526	0,396
92	3850	407412,71	2	1	3	4	5	1510,279	0,392
91	4638	496369,34	6	3	5	1	4	1840,041	0,397
89	6072	752750,07	1	10	1	10	10	2790,445	0,460
96	2006	199725,45	1	1	1	2	1	740,382	0,369
97	3229	340397,11	2	3	3	3	1	1261,852	0,391
99	2012	199896,51	2	1	1	1	1	741,016	0,368
98	3378	362279,36	1	3	4	4	1	1342,970	0,398
<b>95</b>	<b>8670</b>	<b>1024297,86</b>	<b>9</b>	<b>9</b>	<b>9</b>	<b>8</b>	<b>9</b>	<b>3797,072</b>	<b>0,438</b>
100	2646	269629,34	3	1	3	1	1	999,516	0,378

Karena *pareto optimality* memberikan set solusi optimal, *users* dapat memilih solusi terbaik yang mereka inginkan berdasarkan preferensi mereka. Dari hasil tersebut diketahui bahwa meskipun AGV menggunakan kapasitas maksimum (25) untuk membawa barang, namun tidak memberikan hasil total throughput yang optimal karena membawa lebih banyak barang akan memberikan waktu bongkar muat yang lebih lama. Di sisi lain, ketersediaan barang tidak selalu ada karena ketidakpastian dari proses sebelumnya. Penggunaan energi juga akan lebih tinggi karena AGV perlu melakukan perjalanan bolak-balik dengan terlalu banyak beban yang membutuhkan energi yang sangat besar. Jika kapasitas AGV minimum (1), hasil total energi akan optimal akan tetapi *total throughput* menjadi sangat kecil.

Untuk memilih satu solusi, digunakan *Weighted Sum Method* dengan memberikan bobot lebih pada total throughput (0,6) dan konsumsi energi (0,4), hasil terbaik diberikan oleh solusi ID 95 seperti yang ditunjukkan pada Tabel 4. 5. Oleh karena itu, kapasitas optimum di bawah ketidakpastian untuk setiap AGV adalah AGV 1 ( $X_1$ ) = 9, AGV 2 ( $X_2$ ) = 9, AGV 3 ( $X_3$ ) = 9, AGV 4 ( $X_4$ ) = 8, AGV 5 ( $X_5$ ) = 9. Akan tetapi, dilihat dari sisi biaya per unit, solusi tersebut memberikan hasil sebesar 0,438. Biaya tersebut bukanlah yang tertinggi (0,479) maupun terendah (0,363) dibandingkan solusi lainnya, melainkan berada ditengah-tengah dan lebih mendekati biaya yang tertinggi. Namun biaya tersebut hanya memperhitungkan biaya listrik AGV saja, jika mempertimbangkan biaya lainnya bisa jadi memberikan hasil yang berbeda. Preferensi utama dikembalikan kepada pengambil keputusan apakah ingin meningkatkan kepuasan pelanggan dengan hasil output yang lebih tinggi atau mereduksi total biaya logistik melalui pengurangan biaya energi listrik AGV.

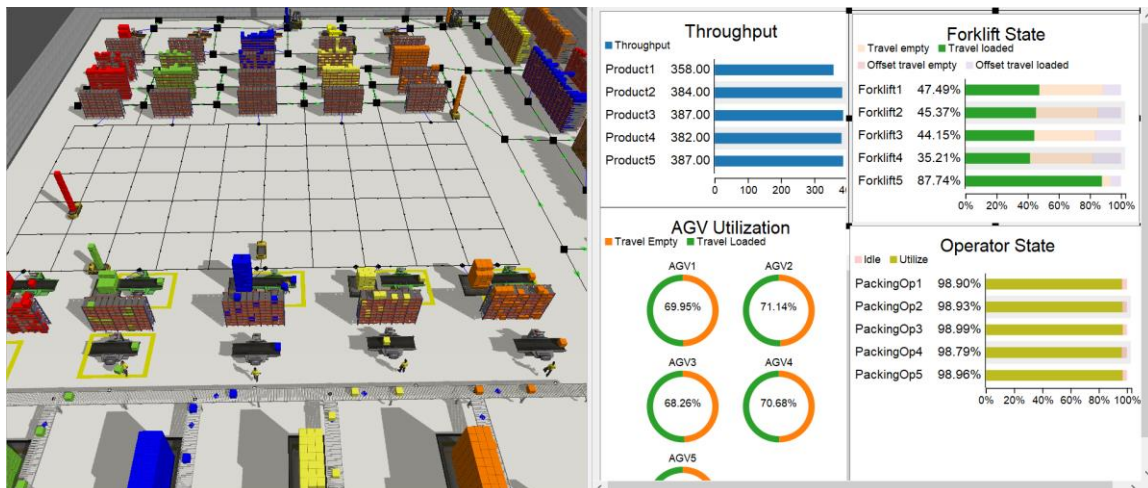
Tabel 4. 5 Hasil *Weighted Sum Method*

<b>Solution ID</b>	<b>Normalized Throughput</b>	<b>Normalized Energy Consumption</b>	<b>Weighted Sum Method</b>	<b>Ranking</b>
2	0	1	0,4	17
4	0,846879017	0,305420955	0,630296	3
63	0,493643765	0,543263297	0,513492	7
60	0,998285959	0,018692772	0,606449	4
69	1	0	0,6	5
86	0,256534781	0,781390936	0,466477	10
87	0,961005571	0,136815551	0,63133	2
90	0,228681617	0,80541043	0,459373	11
92	0,293243822	0,759119075	0,479594	9
91	0,405799172	0,666464314	0,510065	8

Solution ID	Normalized Throughput	Normalized Energy Consumption	Weighted Sum Method	Ranking
89	0,610627053	0,399425228	0,526146	6
96	0,029852878	0,975440387	0,408088	16
97	0,204542208	0,82892067	0,454294	13
99	0,030709899	0,975262213	0,408531	15
98	0,225824882	0,806128723	0,457946	12
<b>95</b>	<b>0,981716898</b>	<b>0,11658856</b>	<b>0,635666</b>	<b>1</b>
100	0,12126839	0,902630422	0,433813	14

#### 4.4.2 Dynamic Dashboard

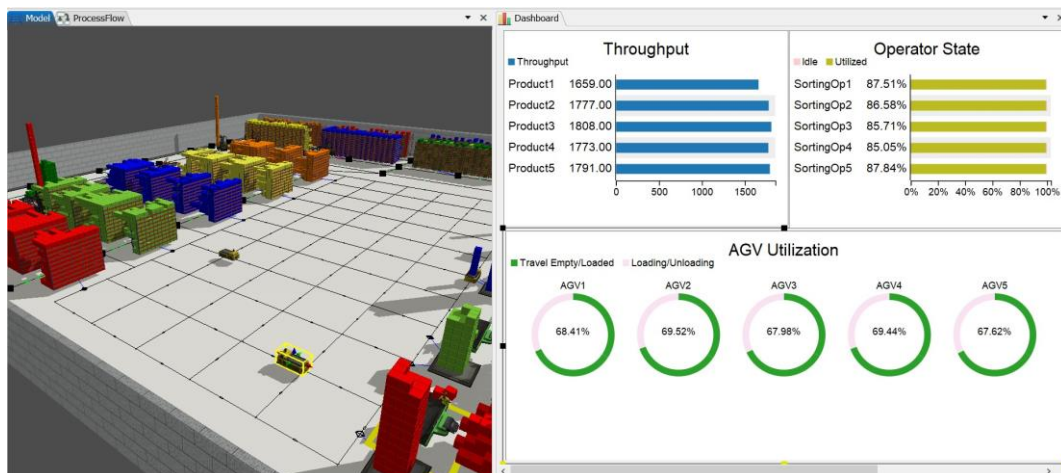
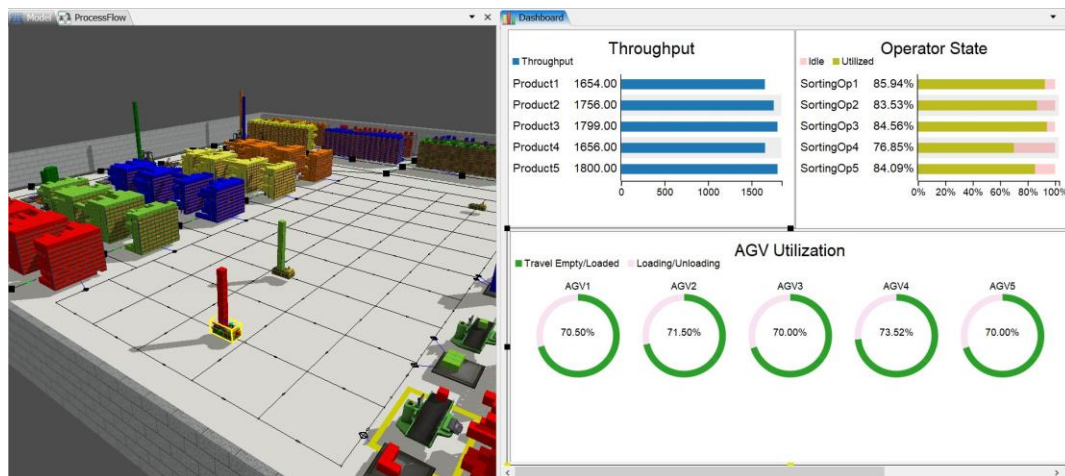
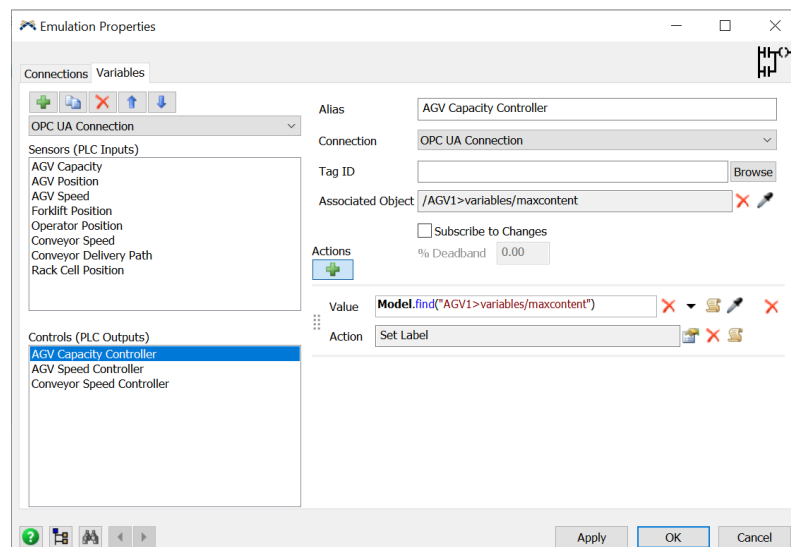
Tidak dapat dipungkiri bahwa *monitoring* dan *controlling* merupakan bagian penting untuk mengambil keputusan yang baik dalam operasional. Hal ini juga menjadi tujuan utama industri 4.0 yang dapat dicapai melalui teknologi *digital twin*. Oleh karena itu, *dashboard* merupakan salah satu alat yang dapat digunakan. Ini berguna untuk mendeteksi masalah dengan visualisasi daripada angka. Melalui teknologi simulasi, *dashboard* menjadi sangat akurat dan *real-time*. Oleh karena itu, *dashboard* untuk *smart warehouse* dibuat dengan menggunakan fitur perangkat lunak simulasi Flexsim untuk beberapa parameter penting seperti *total throughput*, *Forklift State*, *Operator State*, dan *AGV Utilization*. Tampilan *dashboard* saat simulasi berjalan ditunjukkan pada Gambar 4. 17.



Gambar 4. 17 *Dynamic Dashboard*

*Total throughput* akan menginformasikan pengguna tentang berapa banyak barang yang sudah siap dikirim untuk setiap jenis produk. *Forklift State* dan *Operator State* akan memberi tahu pengguna tentang persentase forklift dan operator seperti berapa lama mereka *idle* dan *utilized*. *AGV Utilization* menginformasikan pengguna tentang berapa lama AGV membawa barang atau membawa kosong dan bongkar muat. *Dashboard* akan diperbarui nilainya secara *real-time*. Hal ini sangat bermanfaat dari sisi pemantauan. Data ini dapat disesuaikan berdasarkan preferensi pengguna. Nantinya, data dari *dashboard* juga dapat diekspor ke *database* dan analisis prediktif dapat dilakukan misalnya untuk melakukan *predictive maintenance*.

Terkait dengan optimisasi yang telah dilakukan, Gambar 4.18 dan Gambar 4.19 menunjukkan *dashboard* sebelum dan sesudah optimisasi. Meskipun hasil yang dioptimalkan menurunkan *total throughput* sebesar 1,62% namun dapat menghemat penggunaan energi AGV sebesar 8,4%. Hal ini dapat menghemat biaya operasional dan memperpanjang siklus hidup AGV. Dilihat dari *AGV utilization*, hasil optimisasi memberikan persentase yang lebih tinggi dan persentase bongkar muat yang lebih rendah. Perjalanan AGV yang lebih lama atau jauh tidak berarti penggunaan energi yang lebih tinggi, tetapi tergantung pada berapa banyak beban yang dibawa AGV untuk setiap perjalanan. Dengan menggunakan *dashboard* yang diusulkan ini, akan memudahkan pengguna untuk mengambil keputusan guna mengoptimalkan operasi gudang di bawah ketidakpastian berdasarkan konsep *digital twin* berbasis simulasi. Yang lebih menarik adalah, konsep yang diusulkan ini dapat mengimplementasikan keputusan optimal langsung dari perangkat lunak ke *shopfloor* tanpa perlu mengubahnya secara manual. Konfigurasi aplikasi kontrol ini ditunjukkan pada Gambar 4. 20.

Gambar 4. 18 *Dashboard* sebelum OptimisasiGambar 4. 19 *Dashboard* sesudah OptimisasiGambar 4. 20 Konfigurasi dari Proses *Controlling*

## BAB V

### KESIMPULAN DAN REKOMENDASI

#### 5.1 Kesimpulan

Konsep baru dari *digital twin* berbasis simulasi menggunakan *Asset Administration Shell* (AAS) *framework* telah diusulkan menggunakan perangkat lunak Flexsim dan *smart warehouse* sebagai *use case*. AAS *Structure* juga telah diusulkan untuk aset-aset utama seperti *Automated Guided Vehicle* (AGV), *Operator*, *Conveyor*, *Forklift*, dan *Storage Rack*. Konsep ini dirancang agar memungkinkan *digital twin* tidak hanya untuk memonitor dan mengontrol sistem nyata tetapi juga melakukan simulasi berbasis optimisasi dan *dynamic dashboard*.

Model simulasi yang telah terverifikasi telah dirancang untuk meniru operasi *smart warehouse* menggunakan perangkat lunak Flexsim. Model tersebut telah meniru perilaku dari aset utama di *smart warehouse* serta tata letak dan logika di baliknya. *System integration* telah dirancang melalui AAS *framework* dan *sequence diagram* untuk membantu pemahaman tentang bagaimana aliran data dan kontrol bekerja dari semua AAS termasuk AAS Flexsim sebagai *digital twinnya*.

Demonstrasi untuk menunjukkan kelebihan dari konsep yang diusulkan juga telah dilakukan. Hasil *simulation-based optimization* untuk *AGV capacity planning* dengan permasalahan *multi-objective* menunjukkan bahwa untuk memaksimalkan *total throughput* dan meminimalkan penggunaan energi secara bersamaan, solusi ID 95 memberikan hasil yang optimal di mana kapasitas untuk setiap AGV adalah sebagai berikut: AGV 1 (X1) = 9, AGV 2 (X2) = 9, AGV 3 (X3) = 9, AGV 4 (X4) = 8, AGV 5 (X5) = 9. Hal ini menjadi terobosan bagi penelitian DT karena penelitian terdahulu memiliki kekurangan yang mana DT tidak dapat mencerminkan optimisasi. Hasil perancangan *dynamic dashboard* juga telah dirancang untuk meningkatkan kemampuan *monitoring*.

## 5.2 Rekomendasi

Rekomendasi diberikan untuk dua pemangku kepentingan sebagai berikut:

### 1. Rekomendasi untuk *smart warehouse*

Penelitian ini merupakan titik awal/*preliminary study* untuk mengimplementasikan DT berbasis simulasi sebagai teknologi untuk menunjang digitalisasi tingkat lanjut. Setidaknya dua tahap telah dilakukan pada penelitian ini yaitu perancangan sistem AAS khususnya konsep untuk menghubungkan aset fisik dengan aset digital menggunakan perangkat simulasi Flexsim dan model simulasi yang telah terverifikasi. Konsep yang diusulkan juga dapat diterapkan pada *warehouse* lainnya dengan menyesuaikan aset yang tersedia untuk membangun DT berbasis simulasi. Jika partner lain dalam rantai pasok membangun DT dengan konsep yang telah diusulkan dalam penelitian ini, integrasi sistem tidak hanya dapat dilakukan untuk internal perusahaan, melainkan juga antar partner karena menggunakan framework AAS yang telah distandarisasi. Tahap berikutnya adalah perusahaan dapat melakukan instalasi *hardware* seperti sensor dan kontroller yang telah diusulkan pada penelitian ini untuk merealisasikan DT.

### 2. Rekomendasi untuk penelitian ke depan

Literatur menunjukkan bahwa penelitian AAS masih terbatas, padahal manfaatnya sebagai kunci integrasi dan standardisasi dalam hal digitalisasi sangat besar. Sampai saat ini, sulit mencapai standarisasi tanpa AAS *framework*. Tanpa adanya standarisasi, setiap industri akan memiliki DTnya sendiri tanpa ada integrasi di antara mereka. Maka dari itu, penelitian AAS dapat dikembangkan lebih lanjut terutama dengan menyesuaikan objek penelitian yang dihadapi mengingat AAS sangat bergantung pada submodel dari asetnya. Analisis ekonomi juga dapat dipertimbangkan untuk melihat besaran biaya dan kelayakannya dari perspektif ekonomi. Konsep yang diusulkan juga dapat diintegrasikan dengan teknologi lainnya seperti *data analytics* menggunakan framework AAS yang sama, tentunya dengan submodel tambahan sehingga beberapa data dikirim ke simulasi untuk tujuan DT dan sisanya disimpan di *cloud* sebagai *big data*, misalnya untuk membangun *predictive maintenance*. Dengan demikian, teknologi-teknologi industri 4.0 dapat diintegrasikan secara sistematis dan komprehensif.

## REFERENSI

- Adolphs, P., Auer, S., Bedenbender, H., Billmann, M., Hankel, M., Heidel, R., Hoffmeister, M., Huhle, H., Jochem, M., & Kiele-Dunsche, M. (2016). Structure of the Administration Shell. In *Continuation of the Development of the Reference Model for the Industrie 4.0 Component*.
- Adolphs, P., Bedenbender, H., & Dirzus, D. (2015). Status Report: Reference Architecture Model Industrie 4.0 (RAMI4.0). *VDI/VDE-Gesellschaft Mess- Und Automatisierungstechnik*.
- Arm, J., Benesl, T., Marcon, P., Bradac, Z., Schröder, T., Belyaev, A., Werner, T., Braun, V., Kamensky, P., Zezulka, F., Diedrich, C., & Dohnal, P. (2021). Automated design and integration of asset administration shells in components of industry 4.0. *Sensors*. <https://doi.org/10.3390/s21062004>
- Assadi, A. Al, Fries, C., Fechter, M., Maschler, B., Ewert, D., Schnauffer, H. G., Zürn, M., & Reichenbach, M. (2020). User-friendly, requirement based assistance for production workforce using an asset administration shell design. *Procedia CIRP*. <https://doi.org/10.1016/j.procir.2020.02.192>
- Benjamin. (2021). *Warehouse Optimization*. Flexsim Forum. <https://answers.flexsim.com/questions/87366/warehouse-optimization.html>
- Bitkom, VDMA, & ZVEI. (2015). Implementation Strategy Industrie 4.0. In *Bitkom e.V.*
- Burgess, T., & Pidd, M. (1999). Computer Simulation in Management Science, 4th Edition. *The Journal of the Operational Research Society*. <https://doi.org/10.2307/3010694>
- Cavalieri, S., & Salafia, M. G. (2020a). A model for predictive maintenance based on asset administration shell. *Sensors (Switzerland)*. <https://doi.org/10.3390/s20216028>
- Cavalieri, S., & Salafia, M. G. (2020b). Insights into mapping solutions based on OPC UA information model applied to the industry 4.0 asset administration shell. *Computers*. <https://doi.org/10.3390/computers9020028>
- Clement J. (2020). *Internet usage worldwide - Statistics & Facts | Statista*. Statistica. <https://www.statista.com/topics/1145/internet-usage-worldwide/>
- Colombo, A. W., Karnouskos, S., Kaynak, O., Shi, Y., & Yin, S. (2017). Industrial Cyberphysical Systems: A Backbone of the Fourth Industrial Revolution. *IEEE*

- Industrial Electronics Magazine*. <https://doi.org/10.1109/MIE.2017.2648857>
- CyberFactory#1. (2020). *Factory SoS Modelling and Simulation State of the Art Analysis – version 1.0*. [https://itea3.org/project/workpackage/document/download/6667/State of the Art: Modeling and Simulation of Factories of the Future \(v1.0\).pdf](https://itea3.org/project/workpackage/document/download/6667/State%20of%20the%20Art%3A%20Modeling%20and%20Simulation%20of%20Factories%20of%20the%20Future%20(v1.0).pdf)
- Erol, S., Jäger, A., Hold, P., Ott, K., & Sihm, W. (2016). Tangible Industry 4.0: A Scenario-Based Approach to Learning for the Future of Production. *Procedia CIRP*. <https://doi.org/10.1016/j.procir.2016.03.162>
- Fonseca, I. C. P. (2011). Simulation hypotheses A proposed taxonomy for the hypotheses used in a simulation model. *SIMUL 2011 : The Third International Conference on Advances in System Simulation*.
- Fontanili, F., Vincent, A., & Ponsonnet, R. (2000). Flow simulation and genetic algorithm as optimization tools. *International Journal of Production Economics*. [https://doi.org/10.1016/S0925-5273\(99\)00047-X](https://doi.org/10.1016/S0925-5273(99)00047-X)
- Gageldonk, G. V., & Waterings, P. (2019). *Digital twins: the power of a virtual visual copy - Unite Copenhagen 2019*. Prespective. <https://www.slideshare.net/unity3d/digital-twins-the-power-of-a-virtual-visual-copy-unite-copenhagen-2019>
- Grieves, M. (2006). Product Lifecycle Management: Driving the Next Generation of Lean Thinking. In *J Prod Innov Manag*.
- Han, S. (2020). A review of smart manufacturing reference models based on the skeleton meta-model. *Journal of Computational Design and Engineering*. <https://doi.org/10.1093/jcde/qwaa027>
- Haße, H., Valk, H. van der, Weißenberg, N., & Otto, B. (2020). Shared Digital Twins: Data Sovereignty in Logistics Networks. *Proceedings of the Hamburg International Conference of Logistics (HICL) – 29*.
- Helmann, A., Deschamps, F., & Loures, E. D. F. R. (2020). Reference architectures for industry 4.0: Literature review. *Advances in Transdisciplinary Engineering*. <https://doi.org/10.3233/ATDE200074>
- Henßen, R., & Schleipen, M. (2014). Interoperability between OPC UA and AutomationML. *Procedia CIRP*. <https://doi.org/10.1016/j.procir.2014.10.042>
- IEC 62714-1. (2012). Engineering data exchange format for use in industrial automation systems engineering (AutomationML) Part 1 - architecture and general requirements.

- Engineering Data Exchange Format for Use in Industrial Automation Systems Engineering. Automation Markup Language. Architecture and General Requirements.*
- Industrial Internet Consortium (IIC). (2020). *Digital Twins for Industrial Applications*. [https://www.iiconsortium.org/pdf/IIC\\_Digital\\_Twins\\_Industrial\\_Apps\\_White\\_Paper\\_2020-02-18.pdf](https://www.iiconsortium.org/pdf/IIC_Digital_Twins_Industrial_Apps_White_Paper_2020-02-18.pdf)
- Institute of Intellectual Property Research & Development. (2015). *Sample Landscape Study Digital Twin Technology*. [https://www.iiprd.com/wp-content/uploads/2015/11/Sample-Landscape\\_Study\\_Digital\\_Twin.pptx](https://www.iiprd.com/wp-content/uploads/2015/11/Sample-Landscape_Study_Digital_Twin.pptx)
- Jasperneite, J., Sauter, T., & Wollschlaeger, M. (2020). Why We Need Automation Models: Handling Complexity in Industry 4.0 and the Internet of Things. *IEEE Industrial Electronics Magazine*. <https://doi.org/10.1109/MIE.2019.2947119>
- Kordos, M., Boryczko, J., Blachnik, M., & Golak, S. (2020). Optimization of warehouse operations with genetic algorithms. *Applied Sciences (Switzerland)*. <https://doi.org/10.3390/app10144817>
- Krasikov, I. I., & Kulemin, A. N. (2020). Analysis of Digital Twin Definition and Its Difference from Simulation Modelling in Practical Application. *KnE Engineering*. <https://doi.org/10.18502/keg.v5i3.6766>
- Laguna, M., & Martí, R. (2002). The OptQuest Callable Library. *Operations Research/Computer Science Interfaces Series*. [https://doi.org/10.1007/0-306-48126-x\\_7](https://doi.org/10.1007/0-306-48126-x_7)
- Lampropoulos, G., Siakas, K., & Anastasiadis, T. (2019). Internet of Things in the Context of Industry 4.0: An Overview. *International Journal of Entrepreneurial Knowledge*. <https://doi.org/10.2478/ijek-2019-0001>
- Law, A. M., & Kelton, W. D. (2000). *Simulation Modeling and Analysis* (éd. 3rd Edition). New York, MA : McGraw-Hill.
- Lee, B., Kim, D. K., Yang, H., & Oh, S. (2017). Model transformation between OPC UA and UML. *Computer Standards and Interfaces*. <https://doi.org/10.1016/j.csi.2016.09.004>
- Lu, Y., Morris, K. C., & Frechette, S. (2015). Standards landscape and directions for smart manufacturing systems. *IEEE International Conference on Automation Science and Engineering*. <https://doi.org/10.1109/CoASE.2015.7294229>
- Lüder, A., Hundt, L., & Keibel, A. (2010). Description of manufacturing processes using

- AutomationML. *Proceedings of the 15th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2010*.  
<https://doi.org/10.1109/ETFA.2010.5641346>
- Luściński, S. (2018). *Digital Twinning for Smart Industry*. <https://doi.org/10.4108/eai.6-11-2018.2279986>
- Marcon, P., Diedrich, C., Zezulka, F., Schroder, T., Belyaev, A., Arm, J., Benesl, T., Bradac, Z., & Vesely, I. (2019). The Asset Administration Shell of Operator in the Platform of Industry 4.0. *Proceedings of the 2018 18th International Conference on Mechatronics - Mechatronika, ME 2018*.
- Morgan, C. B., Banks, J., & Carson, J. S. (1984). Discrete-Event System Simulation. *Technometrics*. <https://doi.org/10.2307/1268124>
- Muralidharan, S., Yoo, B., & Ko, H. (2020). Designing a semantic digital twin model for IoT. *Digest of Technical Papers - IEEE International Conference on Consumer Electronics*. <https://doi.org/10.1109/ICCE46568.2020.9043088>
- Nordgren, W. B. (2003). Flexsim simulation environment. *Winter Simulation Conference Proceedings*. <https://doi.org/10.1109/wsc.2003.1261424>
- OPC Foundation. (2012). OPC Unified Architecture Part 1: Overview and Concepts Version 1.02. In *OPC Foundation*.
- Park, H. S., & Febriani, R. A. (2019). Modelling a platform for smart manufacturing system. *Procedia Manufacturing*. <https://doi.org/10.1016/j.promfg.2020.01.118>
- Park, K. T., Yang, J., & Noh, S. Do. (2021). VREDI: virtual representation for a digital twin application in a work-center-level asset administration shell. *Journal of Intelligent Manufacturing*. <https://doi.org/10.1007/s10845-020-01586-x>
- Platenius-Mohr, M., Malakuti, S., Grüner, S., & Goldschmidt, T. (2019). Interoperable digital twins in IIoT systems by transformation of information models: A case study with asset administration shell. *ACM International Conference Proceeding Series*. <https://doi.org/10.1145/3365871.3365873>
- Platenius-Mohr, M., Malakuti, S., Grüner, S., Schmitt, J., & Goldschmidt, T. (2020). File- and API-based interoperability of digital twins by model transformation: An IIoT case study using asset administration shell. *Future Generation Computer Systems*. <https://doi.org/10.1016/j.future.2020.07.004>

- Platform Industrie 4.0. (2016). Aspects of the Research Roadmap in Application Scenarios. *Platform Industrie 4.0*.
- Profanter, S., Perzylo, A., Rickert, M., & Knoll, A. (2021). A Generic Plug & Produce System Composed of Semantic OPC UA Skills. *IEEE Open Journal of the Industrial Electronics Society*. <https://doi.org/10.1109/ojies.2021.3055461>
- Schleipen, M., Selyansky, E., Henssen, R., & Bischoff, T. (2015). Multi-level user and role concept for a secure plug-and-work based on OPC UA and AutomationML. *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*. <https://doi.org/10.1109/ETFA.2015.7301640>
- Schroeder, G. N., Steinmetz, C., Pereira, C. E., & Espindola, D. B. (2016). Digital Twin Data Modeling with AutomationML and a Communication Methodology for Data Exchange. *IFAC-PapersOnLine*. <https://doi.org/10.1016/j.ifacol.2016.11.115>
- Schweichhart, K. (2016). Reference Architectural Model Industrie 4.0 (RAMI 4.0) - An Introduction. *Plattform Industrie 4.0*.
- Seif, A., Toro, C., & Akhtar, H. (2019). Implementing industry 4.0 asset administrative shells in mini factories. *Procedia Computer Science*. <https://doi.org/10.1016/j.procs.2019.09.204>
- Shi-Wan, L., Bradford, M., Jacques, D., Graham, B., Chigani, A., Martin, R., Murphy, B., & Crawford, M. (2017). The Industrial Internet of Things Volume G1: Reference Architecture. *Industrial Internet Consortium White Paper*.
- Solutions, D. (2019). *Best Discrete Event Simulation Software - 2019*. Best Discrete Event Simulation Software - 2019. <https://www.descreye.com/blog/a-comparison-of-discrete-event-simulation-software-2019/>
- Steindl, G., Stagl, M., Kasper, L., Kastner, W., & Hofmann, R. (2020). Generic digital twin architecture for industrial energy systems. *Applied Sciences (Switzerland)*. <https://doi.org/10.3390/app10248903>
- Tantik, E., & Anderl, R. (2017). Integrated Data Model and Structure for the Asset Administration Shell in Industrie 4.0. *Procedia CIRP*. <https://doi.org/10.1016/j.procir.2017.01.048>
- Tien, J. M. (2011). Manufacturing and services: From mass production to mass customization. *Journal of Systems Science and Systems Engineering*.

- <https://doi.org/10.1007/s11518-011-5166-x>
- Toro, C., Seif, A., & Akhtar, H. (2020). Modeling and Connecting Asset Administrative Shells for Mini Factories. *Cybernetics and Systems*. <https://doi.org/10.1080/01969722.2019.1705554>
- Vaidya, S., Ambad, P., & Bhosle, S. (2018). Industry 4.0 - A Glimpse. *Procedia Manufacturing*. <https://doi.org/10.1016/j.promfg.2018.02.034>
- Wagner, C., Grothoff, J., Epple, U., Drath, R., Malakuti, S., Grüner, S., Hoffmeister, M., & Zimmermann, P. (2017). The role of the Industry 4.0 asset administration shell and the digital twin during the life cycle of a plant. *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*. <https://doi.org/10.1109/ETFA.2017.8247583>
- Ye, X., & Hong, S. H. (2018). An AutomationML/OPC UA-based Industry 4.0 Solution for a Manufacturing System. *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*. <https://doi.org/10.1109/ETFA.2018.8502637>
- Ye, X., & Hong, S. H. (2019). Toward Industry 4.0 Components: Insights into and Implementation of Asset Administration Shells. *IEEE Industrial Electronics Magazine*. <https://doi.org/10.1109/MIE.2019.2893397>
- Ye, X., Jiang, J., Lee, C., Kim, N., Yu, M., & Hong, S. H. (2020). Toward the Plug-and-Produce Capability for Industry 4.0: An Asset Administration Shell Approach. *IEEE Industrial Electronics Magazine*. <https://doi.org/10.1109/MIE.2020.3010492>
- Ye, X., Park, T. Y., Hong, S. H., Ding, Y., & Xu, A. (2018). Implementation of a Production-Control System Using Integrated Automation ML and OPC UA. *2018 Workshop on Metrology for Industry 4.0 and IoT, MetroInd 4.0 and IoT 2018 - Proceedings*. <https://doi.org/10.1109/METROI4.2018.8428310>