

**IMPLEMENTASI ALUR WORKFLOW OTOMATIS UNTUK MENGELOLA  
DATA NOTA SALES MENGGUNAKAN N8N DI VERNON CORP**

**PRAKTIK KERJA LAPANGAN**



**REXY SIMSON SANJAYA**

**NIM: 312210019**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI DAN DESAIN  
UNIVERSITAS MA CHUNG  
MALANG  
2026**

LEMBAR PENGESAHAN  
PRAKTIK KERJA LAPANGAN

IMPLEMENTASI WORKFLOW OTOMATIS MENGELOLA SALES INVOICE  
MENGUNAKAN N8N DI VERNON CORP

Oleh:

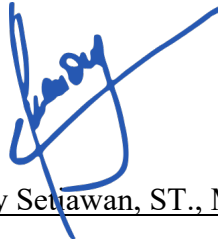
REXY SIMSON SANJAYA

NIM. 312210019

dari:

PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI DAN DESAIN  
UNIVERSITAS MA CHUNG

Dosen Pembimbing,



Hendry Setiawan, ST., M.Kom.

NIP. 20100006

Dekan Fakultas Teknologi dan Desain,

Prof. Dr.Eng. Romy Budhi, ST., MT., M.Pd.

NIP. 20070035

## Kata Pengantar

Segala puji dan syukur saya panjatkan kepada Tuhan Yang Maha Esa atas limpahan rahmat dan karunia-Nya sehingga pelaksanaan kegiatan Praktik Kerja Lapangan (PKL) ini dapat berjalan dengan lancar. Laporan ini disusun sebagai salah satu syarat untuk menyelesaikan mata kuliah PKL bagi mahasiswa Program Studi Teknik Informatika, Universitas Ma Chung Malang.

Pada kesempatan ini, saya menyampaikan rasa terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan bantuan dan dukungan selama proses pelaksanaan PKL maupun dalam penyusunan laporan ini. Ucapan terima kasih secara khusus saya sampaikan kepada:

1. Hendry Setiawan, ST., M.Kom., selaku dosen pembimbing PKL ini;
2. Bapak Prof. Dr. Eng. Romy Budhi, ST., MT., M.Pd., selaku Dekan Fakultas Sains dan Teknologi Universitas Ma Chung;
3. Rekan-rekan mahasiswa serta para responden atau subjek penelitian yang telah berpartisipasi dan memberikan dukungan;
4. Keluarga yang selalu memberikan doa, motivasi, dan dukungan kepada saya.

saya telah berusaha menyusun laporan ini dengan sebaik mungkin. Namun, saya menyadari bahwa laporan ini masih memiliki kekurangan. Semoga laporan PKL ini dapat memberikan manfaat dan informasi yang berguna bagi para pembaca.

Malang, 2 Desember 2025



Remy Simson Sanjaya

## Daftar Isi

Kata Pengantar .....	i
Daftar Isi.....	ii
Daftar Gambar.....	iv
<b>Bab I .....</b>	<b>1</b>
<b>Pendahuluan .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Identifikasi Masalah .....	2
1.3 Batasan Masalah.....	2
1.4 Rumusan Masalah .....	2
1.5 Tujuan Penelitian .....	3
1.6 Manfaat .....	3
<b>Bab II.....</b>	<b>5</b>
<b>Gambaran Umum Perusahaan .....</b>	<b>5</b>
2.1 Profil dan Jenis Usaha Perusahaan.....	5
2.2 Sejarah Perusahaan.....	6
2.3 Struktur Organisasi Perusahaan .....	6
2.4 Deskripsi Unit Bisnis (Thunderlab) .....	7
2.5 Struktur Perusahaan .....	8
2.6 Lokasi Perusahaan.....	9
2.7 Budaya Kerja.....	9
<b>Bab III .....</b>	<b>11</b>
<b>Tinjauan Pustaka .....</b>	<b>11</b>

3.1 Otomasi Proses Bisnis (Business Process Automation).....	11
3.2 n8n ( <i>Nodemation</i> ) .....	11
3.3 Application Programming Interface (API).....	13
3.4 JavaScript Object Notation (JSON) .....	15
3.5 Manajemen Data: Filtering dan Normalisasi .....	16
3.6 Sales Invoice (Faktur Penjualan) .....	17
3.7 Google Sheets sebagai Basis Data Sementara .....	19
<b>Bab IV</b> .....	20
<b>Deskripsi Data dan Hasil Praktik Kerja Lapangan</b> .....	20
4.1 Persiapan Data Api.....	20
4.2 Persiapan Software N8N .....	24
4.3 Penggunaan Software.....	28
4.4 Struktur Workflow Utama.....	29
4.5 Workflow Pembuatan Sales Invoice .....	32
4.6 Kesimpulan Implementasi Workflow .....	35
4.7 Hasil Implementasi pada Sistem ERPNext.....	35
<b>Bab V</b> .....	39
<b>Penutup</b> .....	39
5.1 Kesimpulan .....	39
5.2 Saran.....	40
<b>Daftar Pustaka</b> .....	42

## Daftar Gambar

Gambar 2.1 Logo Vernon Corp .....	5
Gambar 2.2 Tampilan Logo Thunderlab.....	7
Gambar 2.3 Struktur Perusahaan.....	8
Gambar 3.1 Tampilan potongan Workflow Sistem Otomatis N8N .....	12
Gambar 3.2 Tampilan Data Pembelian API dari Postman.....	14
Gambar 3.3 Tampilan Output JSON.....	16
Gambar 3.4 Tampilan ERP Next Sales Invoice .....	18
Gambar 3.5 Tampilan Google Sheets sebagai penyimpanan data penjualan Sementara .....	19
Gambar 4.1 Tampilan Document Desty .....	20
Gambar 4.2 Tampilan mendapatkan Token Bearer .....	21
Gambar 4.3 Tampilan Order Page .....	21
Gambar 4.4 Tampilan salah satu Get Order.....	22
Gambar 4.5 Tampilan Get SKU barang.....	23
Gambar 4.6 Tampilan penyimpanan Data Pembelian Sementara di Google sheet.....	24
Gambar 4.7 Tampilan Logo N8N .....	24
Gambar 4.8 Tampilan harga N8N.....	25
Gambar 4.9 Tampilan Discord forum belajar N8N .....	26
Gambar 4.10 Tampilan Docker Desktop dan Ngrok .....	28
Gambar 4.11 Alur Logika Pengambilan Data Pesanan pada n8n .....	29
Gambar 4.12 Code untuk menyimpan di Google Sheet Memory .....	30

Gambar 4.13 Code untuk Pengambilan Data Order Page.....	30
Gambar 4.14 Code untuk Filter data Order.....	31
Gambar 4.15 Alur Pemrosesan Detail Pesanan dan Normalisasi SKU .....	31
Gambar 4.16 Code untuk ekstrak dan Normalisasi item (SKU) .....	32
Gambar 4.17 Alur Integrasi Logika Percabangan dan Pembuatan Sales Invoice ke ERPNext.....	33
Gambar 4.18 Code persiapan dan validasi Customer .....	33
Gambar 4.19 Code memnentukan harga jual.....	34
Gambar 4.20 Code Payload membuat Faktur Penjualan .....	34
Gambar 4.21 Tampilan Output Pembuatan Customer pada ERPNext .....	36
Gambar 4.22 Tampilan Output Pembuatan Item pada ERPNext.....	37
Gambar 4.23 Tampilan Output Sales Invoice pada ERPNext .....	37
Gambar 4.24 Tampilan Detail Dokumen Sales Invoice.....	38

UNIVERSITAS  
MA CHUNG

# Bab I

## Pendahuluan

### 1.1 Latar Belakang

Perkembangan teknologi informasi semakin pesat mendorong setiap perusahaan untuk melakukan digitalisasi dalam berbagai bagian kegiatan operasional, termasuk dalam pengelolaan serta proses bisnis dalam perusahaan. Salah satu tantangan utama yang dihadapi perusahaan terutama yang bergerak di bidang penjualan *online*, adalah bagaimana mengelola data pesanan dari berbagai *platform marketplace* secara cepat, tepat, dan terintegrasi. Volume pesanan yang semakin besar, perubahan status order yang terus berubah, serta kebutuhan untuk merekonsiliasi data yang terus menerus memerlukan adanya sistem yang bisa bekerja secara otomatis dan efisien.

Vernon Corp, sebagai perusahaan yang bergerak di bidang penjualan *multichannel*, membutuhkan solusi atau metode yang bisa memudahkan proses pengambilan data pesanan, pengolahan informasi produk (SKU), hingga pembuatan dokumen seperti *sales invoice*. Proses yang sebelumnya dilakukan secara manual berpotensi menimbulkan kesalahan (*human error*), keterlambatan, dan kekurangan efisiensi dalam pekerjaan sehari-hari.

Untuk menjawab kebutuhan tersebut, teknologi otomasi *workflow* seperti n8n merupakan pilihan yang efektif. n8n memungkinkan pengintegrasian antar sistem melalui API, pengolahan data secara otomatis, serta pembuatan alur kerja yang berjalan secara terjadwal tanpa intervensi manual. Implementasi *workflow* otomatis ini diharapkan dapat meningkatkan efisiensi operasional, mengurangi beban pekerjaan yang sebelumnya manual, dan mempercepat proses penyajian data bagi tim terkait.

Melalui kegiatan Magang yang di laksanakan pada periode september 2025 hingga Januari 2026 yang di konversi menjadi 6 Mata kuliah termasuk Praktik Kerja Lapangan, Saya melakukan perancangan dan implementasi *workflow* otomatis untuk mengintegrasikan data pesanan *marketplace*, mengolah data pesanan, serta mendukung pembuatan *sales invoice* di Vernon Corp. Dengan adanya sistem otomatis ini, perusahaan diharapkan dapat memperoleh alur kerja yang lebih efektif, akurat, serta mudah dikelola dalam jangka panjang.



## 1.2 Identifikasi Masalah

Berdasarkan hasil observasi dan analisis pada proses operasional Vernon Corp, terdapat beberapa permasalahan yang diidentifikasi, yaitu:

1. Pengambilan data pesanan dari berbagai *marketplace* masih memerlukan proses manual yang memakan waktu dan rentan terhadap kesalahan.
2. Proses *filtering* dan pengolahan data order (seperti pengecekan status, item, dan SKU) belum terstandarisasi sehingga menyebabkan inkonsistensi data.
3. Tidak adanya *workflow* terpusat yang dapat mengelola data secara *real-time* untuk mendukung kebutuhan operasional perusahaan.
4. Tingginya volume data menyebabkan kesulitan dalam memonitor perubahan status pesanan secara berkala.

## 1.3 Batasan Masalah

Agar penelitian lebih terarah dan fokus, maka batasan masalah dalam kegiatan ini adalah:

1. Implementasi automasi hanya dilakukan menggunakan platform n8n sebagai *workflow engine*.
2. Data yang diproses terbatas pada data order dan SKU dari *marketplace* yang terhubung melalui API.
3. Sistem yang dikembangkan hanya mencakup proses pengambilan data, *filtering*, normalisasi, penyimpanan ke *Google Sheets*, dan dukungan terhadap pembuatan *sales invoice*.
4. Penelitian difokuskan pada kebutuhan operasional Vernon Corp selama periode PKL.

## 1.4 Rumusan Masalah

Rumusan masalah yang menjadi dasar penelitian ini adalah sebagai berikut:

1. Bagaimana membangun *workflow* otomatis untuk mengambil dan memproses data order dari Api secara terintegrasi?

2. Bagaimana merancang mekanisme *filtering* dan normalisasi data agar informasi pesanan dan SKU menjadi lebih akurat dan seragam?
3. Bagaimana mengimplementasikan automasi yang dapat mendukung pembuatan *sales invoice* secara efisien di Vernon Corp?
4. Bagaimana memastikan workflow n8n berjalan stabil, terjadwal, dan mampu menangani volume data pembelian dari Sales Invoice penjualan yang besar ?

### 1.5 Tujuan Penelitian

Tujuan dari penelitian dan pelaksanaan PKL ini meliputi:

1. Mengembangkan *workflow* otomatis menggunakan n8n untuk mengintegrasikan data order *marketplace* secara *real-time*.
2. Meningkatkan akurasi dan konsistensi data pesanan melalui proses *filtering* dan normalisasi yang terstruktur.
3. Mendukung proses pembuatan *sales invoice* agar lebih cepat, akurat, dan minim intervensi manual.
4. Meningkatkan efisiensi operasional Vernon Corp melalui otomasi proses bisnis berbasis data.

### 1.6 Manfaat

Adapun manfaat yang diharapkan dari penelitian ini adalah:  
Bagi Perusahaan (Vernon Corp):

1. Mempercepat proses pengolahan data pesanan dari berbagai *marketplace*.
2. Mengurangi risiko kesalahan manual dalam pencatatan dan penyusunan *invoice*.
3. Memiliki *workflow* operasional yang lebih efisien dan mudah dikelola.

Bagi Mahasiswa:

1. Menambah pengalaman praktis dalam implementasi *workflow* automasi menggunakan n8n.

2. Meningkatkan pemahaman tentang integrasi API, pengolahan data, dan otomatisasi proses bisnis.

Bagi Institusi Pendidikan:

1. Menjadi referensi bagi penelitian terkait automasi data dan integrasi sistem.
2. Memberikan contoh penerapan nyata dari praktek Automasi dalam dunia kerja.



UNIVERSITAS  
**MA CHUNG**

## Bab II

### Gambaran Umum Perusahaan

#### 2.1 Profil dan Jenis Usaha Perusahaan

Vernon Corp merupakan sebuah perusahaan induk (*holding company*) yang menjalankan bisnisnya dengan visi utama "Making People Happy" atau membuat orang bahagia. Lebih dari sekadar mencari keuntungan, Vernon Corp memiliki visi jangka panjang untuk menjadi perusahaan yang berkontribusi dalam redistribusi kekayaan (*wealth redistribution*).



Gambar 2.1 Logo Vernon Corp

Landasan model bisnis Vernon Corp adalah kolaborasi bisnis (*sharing economy*). Perusahaan meyakini bahwa kesuksesan berkorelasi langsung dengan kapasitas seseorang, sehingga Vernon Corp juga berfokus pada peningkatan kapasitas melalui pendidikan dan kolaborasi dengan pihak-pihak yang memiliki visi serupa. Vernon Corp menaungi berbagai unit bisnis yang terbagi dalam beberapa sektor industri, antara lain:

- *Tech Solution*: Thunderlab, yang menyediakan solusi teknologi dalam berbisnis.
- *Hospitality (Refreshment & Relaxation)*: Meliputi Kanana Retreat, Kontena Hotel, Riverstone Hotel, dan Akira Reflexology.

- *Entrepreneur Support*: Unit yang mendukung kebutuhan pengusaha, meliputi AkuntanPro (Jasa akuntansi), Teman Brainstorm (Konsultan diskusi bisnis), VernonHub (Ekspansi penjualan online), VernonCreative (Layanan kreatif dan desain), dan KemasOke (Solusi kemasan).
- *Education*: VernonEdu, yang berfokus pada pendidikan praktis dan keterampilan dalam dunia industri.
- *Food & Beverage*: Meliputi Cibo (Layanan makanan), Kaihara (Katering sehat rendah gula), Nakoa Cafe, Loopi Hoopi (Bakery), dan SangKudo.
- *Lifestyle & Hobbies*: Meliputi Pelihara Ikan (Dukungan hobi akuatik), Clemina Beauty Studio, dan Tomio Barbershop.

## 2.2 Sejarah Perusahaan

Sejarah perusahaan bermula pada tahun 2007. Pada awalnya, perusahaan ini didirikan sebagai sebuah perusahaan teknologi bernama VORTEGE. Pada masa awal pendiriannya, visi perusahaan masih bersifat umum, yaitu berfokus untuk menghasilkan keuntungan (*generating profit*).

Namun, seiring berjalannya waktu, pendiri menyadari bahwa fondasi bisnis yang sesungguhnya adalah tentang bagaimana membantu orang lain, yang kemudian tercermin dalam model bisnis dan produk yang diciptakan. Kesadaran bahwa tujuan perusahaan bukan sekadar profit, melainkan untuk membantu sesama, menjadi titik balik evolusi Vortege menjadi Vernon.

## 2.3 Struktur Organisasi Perusahaan

Struktur organisasi Vernon Corp dirancang secara ramping untuk memfasilitasi pengelolaan berbagai unit bisnis secara efisien. Berdasarkan data manajemen perusahaan, struktur pimpinan puncak adalah sebagai berikut:

- CEO: Erick Mo
- Secretary: Anna
- Chief HRS (Human Resource Specialist): Fennie

Di bawah jajaran manajemen puncak (*C-Level dan Head*), struktur organisasi langsung terbagi ke dalam masing-masing unit bisnis (*Business Unit*), termasuk unit bisnis teknologi tempat Saya melaksanakan praktik kerja lapangan, yaitu Thunderlab.

#### 2.4 Deskripsi Unit Bisnis (Thunderlab)

Pelaksanaan magang ini berada di unit bisnis Thunderlab. Thunderlab adalah divisi *Tech Solution* dari Vernon Corp. Profil Thunderlab Slogan utama Thunderlab adalah "Kami yang urus sistemnya, Anda yang nikmati waktu berharga". Fokus utama unit bisnis ini adalah menyediakan teknologi andal yang memungkinkan operasional bisnis klien berjalan secara otomatis 24/7. Hal ini bertujuan agar para pengusaha (klien) dapat memiliki lebih banyak waktu untuk keluarga dan pengembangan bisnis, sementara sisi teknis dan operasional ditangani oleh sistem yang bekerja seperti pagar otomatis.



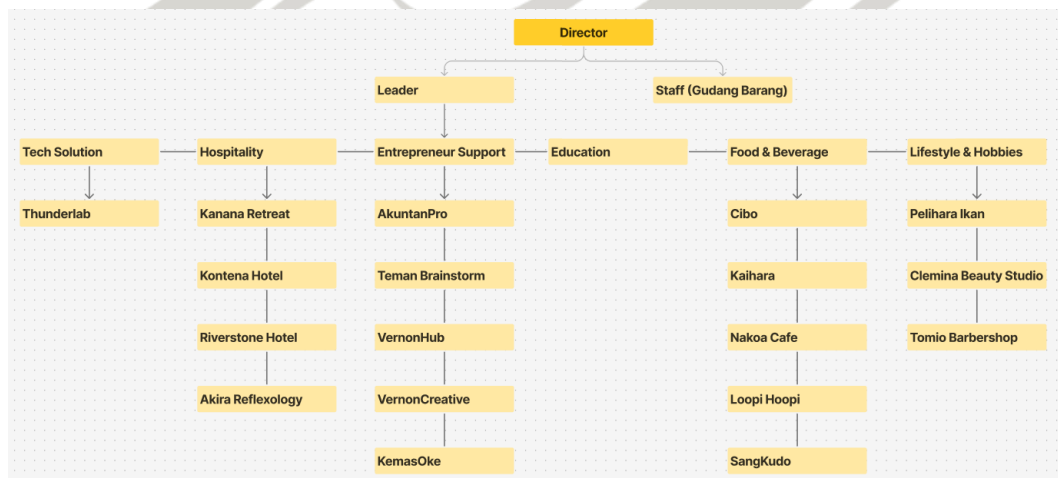
Gambar 2.2 Tampilan Logo Thunderlab

Lingkup Kerja dan Struktur Tim di Thunderlab Dalam operasionalnya, Thunderlab memiliki struktur tim teknis yang berfokus pada pengembangan dan integrasi sistem. Saya tergabung sebagai peserta magang yang pertama kali dan satu satunya bekerja dalam Automasi dan Integrasi Sistem Data Api. Kegiatan operasional di divisi ini meliputi:

1. Integrasi API: Mengambil dan menampilkan berbagai data melalui platform eksternal (seperti *marketplace* atau Desty API) dengan sistem internal perusahaan.
2. Pengembangan *Workflow* (n8n): Merancang alur kerja otomatis untuk memproses atau mengelola data pesanan (order), sinkronisasi produk, dan pemetaan data (*mapping*) agar sesuai dengan format ERP perusahaan.
3. Implementasi ERP: Memastikan data yang diolah terintegrasi dengan baik ke dalam sistem ERP (seperti Frappe/ERPNext) untuk kebutuhan pencatatan faktur (*Sales Invoice*) dan manajemen stok secara *real-time*.

## 2.5 Struktur Perusahaan

Struktur perusahaan di PT. Mentari Pagi Sejahtera (Vernon) adalah sebagai berikut:



Gambar 2.3 Struktur Perusahaan

Berikut merupakan tugas dan juga tanggung jawab pada masing-masing jabatan di PT. Mentari Pagi Sejahtera (Vernon):

### a. Director

Director PT Mentari Pagi Sejahtera (Vernon) bertugas untuk memimpin dan mengurus perusahaan secara keseluruhan dari segi rencana dan tujuan. Selain itu, director juga bertugas untuk mengatur sistem pembagian tugas yang akan diberikan ke leader, dan

juga *director* bisa memberi *feedback* secara langsung kepada *staff* untuk memperbaiki hasil kerja.

*b. Leader*

Leader di PT Mentari Pagi Sejahtera (Vernon) bertugas untuk membagi tugas yang sudah diberikan oleh *director* kepada *staff* yang bersangkutan. Selain itu, *leader* juga bertugas untuk membuat hasil laporan dari pekerjaan yang sudah diselesaikan oleh *staff* dan sudah melalui tahap revisi. Hasil laporan tersebut akan dipresentasikan pada pertemuan rapat yang diadakan setiap minggu.

*c. Staff*

Staff di PT Mentari Pagi Sejahtera (Vernon) bertugas untuk mengerjakan tugas yang sudah diberikan dan melakukan tahap revisi jika ada hal yang perlu diperbaiki. Selain itu, *staff* juga harus melaporkan setiap proses tahapan pengerjaan yang sudah dilakukan kepada *leader*.

*d. Intern*

Intern di PT Mentari Pagi Sejahtera (Vernon) terdapat berbagai divisi, *intern* bertugas untuk melakukan dan mengerjakan tugas yang diberikan *director* atau *leader*, jika telah selesai melakukan tugasnya maka intern diwajibkan untuk melaporkan hasil ke *leader* divisi. Intern di PT Mentari Pagi Sejahtera (Vernon) dipantau secara langsung oleh *director*.

## **2.6 Lokasi Perusahaan**

Kantor pusat Vernon Corp, yang juga menjadi lokasi operasional Thunderlab, beralamat di: Jl. Letjen Sutoyo 102, Blimbing, Malang.

## **2.7 Budaya Kerja**

Vernon Corp menerapkan budaya kerja yang berlandaskan pada filosofi kolaborasi dan visi bersama, sejalan dengan konsep *sharing economy* yang diusung



perusahaan. Hal ini menciptakan lingkungan kerja, khususnya di divisi Thunderlab, yang sangat dinamis dan terbuka terhadap inovasi teknologi.

Standar etos kerja yang diterapkan di perusahaan tergolong tinggi dan disiplin. Hal ini dipengaruhi secara langsung oleh standar kepemimpinan CEO Vernon Corp, Bapak Erick Mo. Dengan latar belakang pendidikan luar negeri, beliau menerapkan standar kualitas internasional yang menuntut seluruh tim, termasuk peserta magang, untuk bekerja secara tegas, profesional, dan memiliki ketelitian tinggi dalam setiap tugas yang diberikan.

Untuk mendukung standarisasi operasional tersebut, Vernon Corp memiliki sistem manajemen pengetahuan yang terstruktur. Seluruh dokumentasi mengenai laporan proyek yang sedang di kerjakan, client Vernon Corp, serta pedoman operasional (SOP) tersimpan secara digital dan dapat diakses oleh seluruh karyawan melalui portal internal perusahaan di [wiki.vernon.id](http://wiki.vernon.id).

UNIVERSITAS  
MA CHUNG

## **Bab III**

### **Tinjauan Pustaka**

#### **3.1 Otomasi Proses Bisnis (Business Process Automation)**

Otomasi Proses Bisnis atau *Business Process Automation* (BPA) adalah pemanfaatan teknologi untuk melaksanakan tugas-tugas atau proses bisnis yang berulang (repetitif) secara otomatis dengan tujuan meningkatkan efisiensi dan meminimalkan kesalahan manusia (*human error*). Menurut Wisswani *Business Process Automation* (BPA) didefinisikan sebagai cara untuk melakukan otomatisasi dalam berbagai proses bisnis yang dijalankan oleh perusahaan. Penerapan sistem informasi dalam konteks ini merupakan kombinasi antara prosedur kerja, informasi, orang, dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi.

Peralihan ke sistem otomatisasi sangat diperlukan karena pola manajemen manual memiliki berbagai batasan, salah satunya menyebabkan penyampaian informasi memakan waktu yang lama. Selain itu, Wisswani juga menyatakan bahwa proses manual menuntut alokasi sumber daya yang besar untuk pelaksanaannya, sehingga kurang efektif bagi perusahaan.

Pemanfaatan teknologi internet atau *website* dalam BPA membuat transaksi bisnis tidak terbatas pada ruang dan waktu, sehingga dapat menjangkau area yang lebih luas. Lebih lanjut, Wisswani menjelaskan bahwa proses yang terotomasi dapat dilakukan dengan lebih efisien karena tidak memerlukan banyak pekerja dan sumber daya lainnya, yang pada akhirnya memberikan keuntungan lebih bagi perusahaan. Berdasarkan hasil pengujian implementasi arsitektur BPA, ditemukan bahwa penerapan teknologi ini mampu meningkatkan efisiensi waktu pengerjaan hingga mencapai 95%.

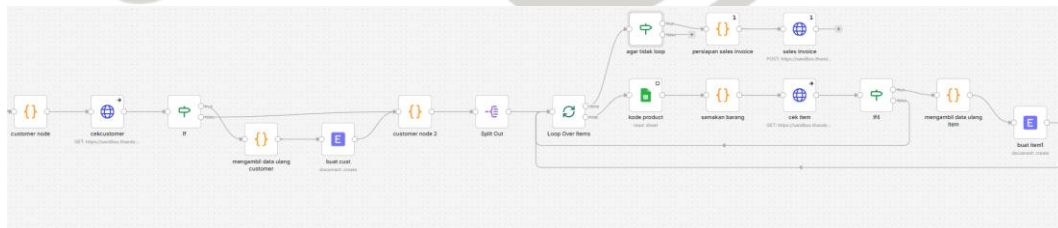
#### **3.2 n8n (Nodemation)**

Otomasi merupakan proses konversi prosedur kerja menjadi sistem otomatis di mana intervensi manusia ditekan hingga ke tingkat yang paling minimum. Salah satu

perangkat lunak yang berkembang pesat untuk tujuan ini adalah n8n. Menurut Nakavisute & Sincharoonsak (2025), n8n didefinisikan sebagai perangkat lunak otomatisasi yang dirancang untuk menjalankan *pipeline* dan mengorkestrasi alur kerja dalam lingkungan yang dikelola secara mandiri (*self-hosted*).

Berbeda dengan solusi berbasis *cloud* publik yang membatasi akses infrastruktur, n8n menawarkan keunggulan dalam hal privasi dan kontrol data. Berdasarkan penelitian tersebut, lingkungan *self-hosted* pada n8n memberikan organisasi kendali penuh atas perangkat keras, konfigurasi keamanan, dan privasi data tanpa bergantung pada penyedia eksternal. Hal ini menjadikan n8n solusi yang ideal bagi perusahaan yang memiliki kebutuhan kepatuhan regulasi data yang ketat.

Secara arsitektur, n8n dibangun dengan pendekatan modular Hidayat et al. (2025) menjelaskan bahwa n8n menerapkan skrip alur kerja berbasis JSON (*JSON-based workflow scripting*) untuk membangun urutan pemicu otomatis yang terstruktur. Struktur ini memungkinkan integrasi yang kompleks disusun melalui antarmuka visual namun tetap mempertahankan fleksibilitas kode program.



Gambar 3.1 Tampilan potongan Workflow Sistem Otomatis N8N

Komponen utama yang membentuk sistem n8n meliputi:

1. *Nodes* (Simpul) Menurut Hidayat et al. (2025), arsitektur n8n berbasis *node* yang modular memungkinkan implementasi logika komputasi yang spesifik. Setiap *node* merepresentasikan fungsi tertentu dalam rantai pemrosesan data. Dalam penelitian tersebut, dicontohkan penggunaan jenis *node* teknis seperti:
  - *HTTP Request Node*: Berfungsi untuk memanggil layanan eksternal atau API pihak ketiga guna pertukaran data.

- *Function Node*: Memungkinkan eksekusi kode (seperti JavaScript) untuk melakukan *parsing* teks secara *in-line* atau validasi format data langsung di dalam alur kerja.
  - Nakavisute & Sincharoonsak (2025) menambahkan bahwa *nodes* juga dapat mencakup logika kondisional (*conditional logic*) untuk memindahkan atau menyinkronkan data secara berulang.
2. *Workflow* (Alur Kerja) *Workflow* dalam n8n berfungsi sebagai orkestrator yang menghubungkan berbagai aplikasi dan layanan. Nakavisute & Sincharoonsak (2025) menyatakan bahwa n8n memungkinkan pembuatan alur kerja yang kompleks melalui antarmuka visual berbasis web, yang secara signifikan mengurangi kebutuhan penulisan kode manual yang ekstensif (*low-code*). Pendekatan ini memfasilitasi pengguna untuk merampingkan proses bisnis dan meningkatkan efisiensi dengan menghubungkan modul-modul yang terpisah menjadi satu kesatuan sistem.
3. *Trigger* (Pemicu) Mekanisme eksekusi dalam n8n sangat bergantung pada *trigger*. Hidayat et al. (2025) menggambarkan bahwa n8n mendukung orkestrasi *pipeline* berbasis kejadian (*event-driven*), di mana alur kerja diinisiasi secara otomatis sebagai respons terhadap peristiwa tertentu. Contoh implementasinya adalah penggunaan *Webhook* atau *GitHub Trigger Node* yang memantau perubahan data secara *real-time* dan memicu urutan validasi tanpa memerlukan intervensi manual atau mekanisme manajemen alur kerja yang terpusat.

### 3.3 Application Programming Interface (API)

*Application Programming Interface* (API) adalah seperangkat aturan dan protokol yang memfasilitasi komunikasi antara berbagai perangkat lunak untuk saling bertukar data. Dalam arsitektur sistem *modern*, API bertindak sebagai perantara yang

membawa permintaan (*request*) dari satu aplikasi ke aplikasi lain dan mengembalikan respons (*response*) yang sesuai secara terstruktur.

Dalam proses pengembangan integrasi sistem, verifikasi komunikasi antar-sistem merupakan tahapan krusial. Untuk itu, digunakan Postman sebagai alat bantu utama. Postman adalah platform pengembangan dan pengujian API yang memungkinkan pengembang untuk membuat, mengirim, menguji, dan mendokumentasikan permintaan HTTP/HTTPS. Melalui antarmuka visual (*GUI*) yang intuitif, Postman memudahkan pengguna untuk melakukan inspeksi terhadap *header*, parameter, dan *body* dari sebuah *request* tanpa perlu menulis kode program yang kompleks. Hal ini sangat penting untuk memastikan bahwa format data yang diterima (biasanya dalam bentuk JSON) sudah sesuai sebelum diintegrasikan ke dalam alur kerja otomatisasi.



```
11  "orderId": "1973175587475488578",
12  "orderSn": "251001VCUQH1SW",
13  "bookingSn": "",
14  "orderType": "DEFAULT_ORDER",
15  "orderStatusList": [
16    | "Completed"
17  ],
18  "platformOrderStatusList": [
19    | "COMPLETED"
20  ],
21  "orderCreateTime": 1759276680000,
22  "orderUpdateTime": 1759844480000,
23  "createTime": 1759276695000,
24  "hasPaid": true,
25  "platform": "desty",
26  "platformName": "shopee",
27  "storeId": "47491431",
28  "storeName": "halolo store",
29  "preOrder": false,
30  "codOrder": false,
31  "cancelBy": "",
32  "logisticStatus": "",
33  "orderPaymentTime": 1759276758000,
34  "cancelReason": "",
35  "trackingNumberList": [
36    | "SPXID05255950709A"
37  ],
38  "packageOrganizedType": "Single",
39  "shippedStatus": "AllShipped",
40  "pickupCode": "",
```

Gambar 3.2 Tampilan Data Pembelian API dari Postman

Jenis arsitektur API yang menjadi standar dalam pengembangan web dan integrasi sistem adalah REST API (*Representational State Transfer*). Hidayat et al. (2025) dalam penelitiannya menjelaskan bahwa integrasi sistem berbasis alur kerja (*workflow*) memanfaatkan mekanisme HTTP *Request* untuk memanggil layanan eksternal.

Mekanisme ini menggunakan metode HTTP standar untuk melakukan operasi manipulasi data, antara lain:

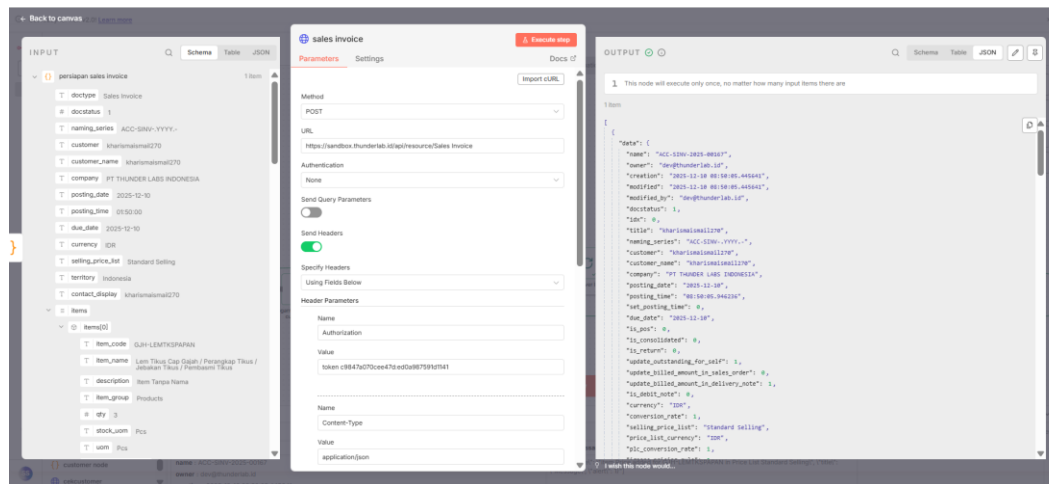
- GET: Digunakan untuk mengambil data dari server (misalnya: menarik rincian pesanan atau konten *fail*).
- POST: Digunakan untuk mengirimkan data baru ke server untuk diproses lebih lanjut.
- PUT/PATCH: Digunakan untuk memperbarui data yang sudah ada di dalam sistem.
- DELETE: Digunakan untuk menghapus data tertentu.

Dalam sistem yang dibangun pada penelitian ini, API digunakan secara intensif untuk menarik data pesanan (*Order Data*) dan informasi produk (*SKU*) dari platform *marketplace*. Hal ini sejalan dengan temuan Hidayat et al. (2025) yang menyatakan bahwa penggunaan *node* integrasi API (*API integration nodes*) memungkinkan pengambilan dan validasi data secara *real-time* dari repositori atau sumber eksternal tanpa memerlukan intervensi manual.

### 3.4 JavaScript Object Notation (JSON)

*JavaScript Object Notation* (JSON) adalah format pertukaran data berbasis teks yang ringan, mudah dibaca oleh manusia, dan mudah diproses oleh mesin. Sebagai standar dalam komunikasi API *modern*, JSON menggunakan struktur pasangan kunci dan nilai (*key-value pairs*) untuk merepresentasikan data.

Dalam konteks pengembangan alur kerja otomatis menggunakan *n8n*, peran JSON sangatlah fundamental. Menurut Hidayat et al. (2025), implementasi sistem otomasi yang efektif menggunakan skrip alur kerja berbasis JSON (*JSON-based workflow scripting*) untuk membangun rangkaian pemicu otomatis yang terstruktur. Struktur ini memungkinkan konfigurasi alur kerja yang bersifat deklaratif (*declarative workflow automation*), sehingga integrasi antar sistem menjadi lebih terorganisir dan mudah dipelihara.



Gambar 3.3 Tampilan Output JSON

Pemahaman mendalam mengenai struktur JSON sangat krusial untuk proses pengolahan data. Hidayat et al. (2025) menjelaskan bahwa dalam n8n, data yang diterima dari sumber eksternal sering kali memerlukan proses *parsing* (penguraian) konten secara *real-time*. Proses ini biasanya melibatkan penggunaan *Function Node* untuk melakukan *in-line text parsing* guna mengekstrak informasi spesifik dari objek JSON—seperti nomor pesanan atau rincian produk—sebelum data tersebut divalidasi atau dikirimkan ke sistem selanjutnya. Kemampuan untuk memecah dan memvalidasi struktur JSON ini memastikan bahwa informasi yang diolah oleh sistem otomatisasi memiliki akurasi yang tinggi.

### 3.5 Manajemen Data: Filtering dan Normalisasi

Dalam arsitektur sistem otomatisasi yang mengintegrasikan berbagai sumber eksternal, manajemen data merupakan tahapan krusial untuk menjamin integritas informasi. Data mentah (*raw data*) yang ditarik melalui API sering kali memerlukan proses pra-pemrosesan agar layak digunakan (*valid*) oleh sistem tujuan.

Menurut Hidayat et al. (2025) dalam jurnal *JMTAS*, efektivitas alur kerja otomatis sangat bergantung pada kemampuan sistem untuk melakukan validasi *real-time* melalui mekanisme *content parsing* (penguraian konten) dan verifikasi format. Proses ini memastikan bahwa hanya data yang memenuhi standar kualitas yang akan diproses lebih lanjut.



Secara spesifik, tahapan manajemen data dalam penelitian ini meliputi:

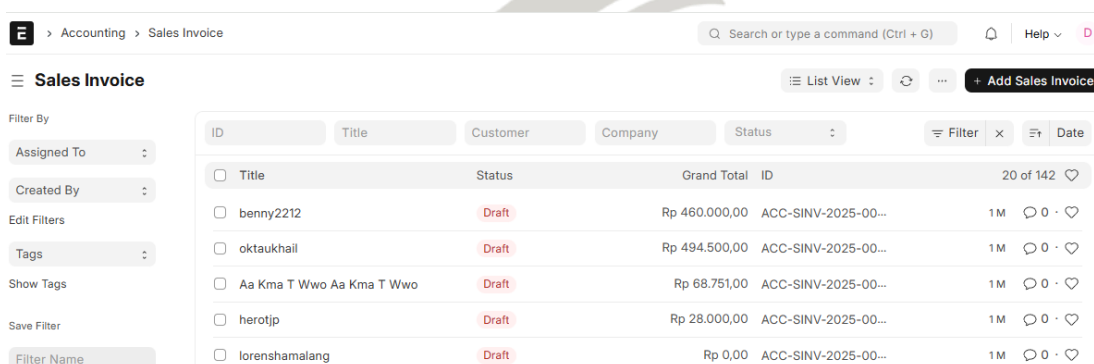
1. *Filtering* (Penyaringan) *Filtering* adalah proses menyeleksi data Berdasarkan penelitian Nakavisute & Sincharoonsak (2025), platform otomasi seperti n8n menyediakan fitur *conditional logic* (logika kondisional) yang memungkinkan sistem untuk memilah data dan menentukan tindakan selanjutnya berdasarkan aturan yang telah ditetapkan. Dalam konteks pengolahan data pesanan, Hidayat et al. (2025) menjelaskan bahwa node validasi berfungsi mengevaluasi kriteria spesifik sebelum meneruskan data ke tahap berikutnya. Implementasinya terlihat pada pemisahan pesanan yang berstatus "Selesai" atau "Siap Dikirim" dari pesanan yang "Dibatalkan", sehingga sistem ERP hanya menerima data transaksi yang relevan.
2. Normalisasi (Penyeragaman Format) Normalisasi adalah proses transformasi data untuk mencapai konsistensi format. Data yang berasal dari berbagai *marketplace* sering kali memiliki struktur yang heterogen, seperti perbedaan format tanggal atau penulisan kode SKU. Menurut Hidayat et al. (2025), normalisasi dapat dilakukan menggunakan *Function Node* dalam n8n yang memungkinkan eksekusi kode untuk *in-line text parsing* dan validasi format secara langsung. Tujuannya adalah menyamakan struktur data tersebut agar sesuai dengan skema basis data pada sistem tujuan (seperti Google Sheets atau ERPNext), sehingga meminimalisir kesalahan *input (data entry errors)* saat pertukaran data berlangsung.

### 3.6 Sales Invoice (Faktur Penjualan)

*Sales Invoice* atau Faktur Penjualan adalah dokumen vital dalam siklus pendapatan yang berfungsi sebagai bukti tertulis atas terjadinya transaksi penjualan barang atau jasa antara penjual dan pembeli. Dokumen ini memuat informasi rinci mengenai spesifikasi barang, kuantitas, harga satuan, serta total tagihan yang wajib dibayarkan oleh pelanggan.



Dalam pengelolaan bisnis modern, faktur penjualan tidak lagi dikelola secara manual, melainkan terintegrasi ke dalam sistem *Enterprise Resource Planning* (ERP). ERP adalah rangkaian perangkat lunak terintegrasi yang digunakan perusahaan untuk mengelola aktivitas bisnis utama, seperti akuntansi, pengadaan, manajemen proyek, dan rantai pasokan. Dalam penelitian ini, platform yang digunakan adalah *ERPNext*, sebuah sistem ERP berbasis *open-source* yang memungkinkan pencatatan transaksi penjualan terhubung langsung dengan pemotongan stok (inventaris) dan pencatatan jurnal akuntansi secara otomatis.



The screenshot displays the 'Sales Invoice' section of the ERPNext application. It features a sidebar with filters and a main table of invoices. The table columns include ID, Title, Customer, Company, Status, Grand Total, and ID. The status of all invoices is 'Draft'.

ID	Title	Customer	Company	Status	Grand Total	ID
benny2212				Draft	Rp 460.000,00	ACC-SINV-2025-00...
oktaukhail				Draft	Rp 494.500,00	ACC-SINV-2025-00...
Aa Kma T Wwo Aa Kma T Wwo				Draft	Rp 68.751,00	ACC-SINV-2025-00...
herotjp				Draft	Rp 28.000,00	ACC-SINV-2025-00...
lorenshamalang				Draft	Rp 0,00	ACC-SINV-2025-00...

Gambar 3.4 Tampilan ERP Next Sales Invoice

Dalam konteks sistem informasi manajemen, faktur penjualan bukan sekadar alat penagihan, melainkan komponen kunci untuk menjaga keakuratan pencatatan stok dan laporan keuangan. Menurut Nugraha et al. (2024) dalam Jurnal *Pixel*, pengelolaan faktur secara manual sering kali menimbulkan risiko ketidakakuratan data dan inefisiensi waktu. Oleh karena itu, pengembangan sistem informasi faktur penjualan yang terkomputerisasi diperlukan untuk:

1. Meningkatkan Efisiensi Proses: Sistem otomatis mampu menyederhanakan alur pembuatan faktur (*invoicing*), sehingga mempercepat siklus transaksi dari pemesanan hingga pembayaran.
2. Menjamin Akurasi Data: Otomasi meminimalkan risiko kesalahan manusia (*human error*) dalam perhitungan harga dan pencatatan inventaris, yang krusial untuk menghasilkan laporan penjualan yang valid.

3. Mendukung Pengambilan Keputusan: Ketersediaan data faktur yang terintegrasi secara *real-time* memberikan wawasan berharga bagi manajemen untuk menyusun strategi penjualan yang lebih baik.

### 3.7 Google Sheets sebagai Basis Data Sementara

Google Sheets adalah aplikasi *spreadsheet* berbasis *cloud* yang memungkinkan penyimpanan, pengeditan, dan kolaborasi data secara *real-time*. Dalam arsitektur sistem sederhana atau tahap pengembangan awal (*prototyping*), Google Sheets sering digunakan sebagai tempat penampungan data sementara (*staging area*) atau basis data ringan sebelum data dipindahkan ke sistem ERP yang lebih kompleks. Kemudahan integrasi Google Sheets dengan n8n menjadikannya alat yang efektif untuk memonitor hasil penarikan data API.

orderid	orderSn	bookingSn	orderType	orderStatusList	platformOrderStatusList	orderCreateTime	orderUpdateTime	createTime	hasPaid	platform	platformName	storeId	storeName
2011608512670769126011541E3JR6R			DEFAULT_ORDE	["Ready_To_Ship"]	["READY_TO_SHIP"]	176843978600X	1768439977000	17684398180X	TRUE	desty	shopee	47491431	halolo :
20116210394426900260115447T4QDD			DEFAULT_ORDE	["Ready_To_Ship"]	["READY_TO_SHIP"]	176844279600X	1768442799000	17684428050X	FALSE	desty	shopee	267626702	Pasaria
20116264130088140582124458807821322			DEFAULT_ORDE	["Ready_To_Ship"]	["AWAITING_SHIPMEN"]	176844407900X	1768444091000	17684440860X	TRUE	desty	tiktok	74944531059772322	Halolo :
20116320837616680260115469B496K			DEFAULT_ORDE	["Ready_To_Ship"]	["READY_TO_SHIP"]	176844543100X	1768445466000	17684454380X	TRUE	desty	shopee	47491431	halolo :
2011662527211962326011540EH3ACR			DEFAULT_ORDE	["Unpaid"]	["UNPAID"]	176845268600X	1768452686000	17684526960X	FALSE	desty	shopee	267626702	Pasaria
201167413902027162601154G14NKPR			DEFAULT_ORDE	["Ready_To_Ship"]	["READY_TO_SHIP"]	176845545700X	1768455466000	17684554640X	TRUE	desty	shopee	47491431	halolo :
20116841854024417582127702001091606			DEFAULT_ORDE	["Ready_To_Ship"]	["AWAITING_SHIPMEN"]	176845785300X	1768457940000	17684578600X	TRUE	desty	tiktok	74944531059772322	Halolo :
201182297393126602601155H2NSVXR			DEFAULT_ORDE	["Ready_To_Ship"]	["READY_TO_SHIP"]	176849094300X	1768490964000	17684909500X	TRUE	desty	shopee	47491431	halolo :

Gambar 3.5 Tampilan Google Sheets sebagai penyimpanan data penjualan Sementara

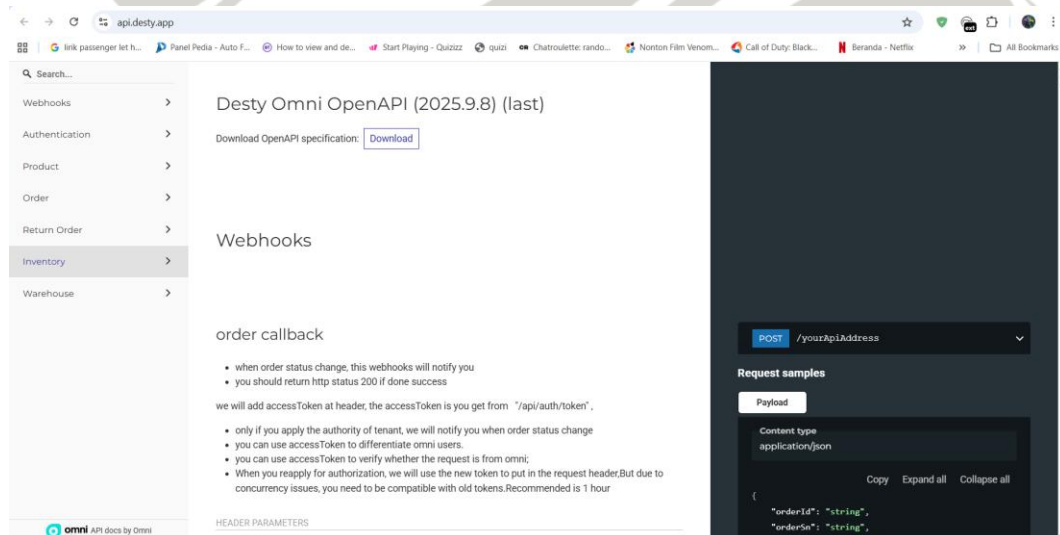
UNIVERSITAS  
MA CHUNG

## Bab IV

### Deskripsi Data dan Hasil Praktik Kerja Lapangan

#### 4.1 Persiapan Data Api

Tahap pertama dalam pelaksanaan praktik kerja lapangan dimulai dengan pemahaman logika sistem dan mekanisme pengambilan data melalui API (*Application Programming Interface*) Desty. Saya mempelajari dokumentasi teknis yang disediakan oleh Desty untuk memahami struktur data penjualan yang berformat JSON. Mengingat volume data yang cukup besar, saya menggunakan aplikasi Postman untuk melakukan pengujian (*testing*) dan eksplorasi data API guna memahami isi dan struktur data yang tersedia.



Gambar 4.1 Tampilan Document Desty

Setelah mempelajari dokumentasi tersebut, saya memahami keterkaitan antara API Desty dan sistem ERPNext yang digunakan perusahaan. Berdasarkan hasil konsultasi dengan pembimbing lapangan, saya diarahkan untuk menggunakan Postman dalam mengakses data API. Langkah awal yang dilakukan adalah mendapatkan otorisasi API (*API Authorization*) menggunakan applyId, username, dan mobile untuk menghasilkan Bearer Token yang diperlukan sebagai kunci akses dalam setiap permintaan data (*request*).

```

1 {
2   "applyId": "8ee34ad5-f66b-4254-8243-1b564919c31a",
3   "username": "+6281333517876",
4   "mobile": "+6281333517876"
5 }

```

---

Body Cookies (1) Headers (11) Test Results ⓘ

{ } JSON ▾ ▸ Preview Visualize ▾

```

1 {
2   "code": "0",
3   "msg": null,
4   "data": {
5     "accessToken": "eyJ3b6c1013IUzUXw139_ey3tZXJjaGFudE1kIjovNjQ1N0Q4MjMyM0c4ODVBMzgzL3R2WShbnR3ZCI6NTkyODQsImV4cCI6MTc3MTA4NTMyMywianF0IjoxNzY4NDkzMzIzfq.",
6     "tokenType": "Bearer",
7     "expireTime": "1771885323715"
8   },
9   "success": true
10 }

```

Gambar 4.2 Tampilan mendapatkan Token Bearer

Setelah berhasil mendapatkan *Bearer Token*, saya dapat mengakses *endpoint* POST *Order Page*. *Endpoint* ini berfungsi untuk menampilkan kumpulan halaman data pesanan yang tersedia di dalam API.

```

11 "orderId": "1973175587475480578",
12 "orderSn": "251001VCUQH1SW",
13 "bookingSn": "",
14 "orderType": "DEFAULT_ORDER",
15 "orderStatusList": [
16   "Completed"
17 ],
18 "platformOrderStatusList": [
19   "COMPLETED"
20 ],
21 "orderCreateTime": 1759276680000,
22 "orderUpdateTime": 1759844448000,
23 "createTime": 1759276695000,
24 "hasPaid": true,
25 "platform": "desty",
26 "platformName": "shopee",
27 "storeId": "47491431",
28 "storeName": "halolo store",
29 "preOrder": false,
30 "codOrder": false,
31 "cancelBy": "",
32 "logisticStatus": "",
33 "orderPaymentTime": 1759276758000,
34 "cancelReason": "",
35 "trackingNumberList": [
36   "SPXID05255950709A"
37 ],
38 "packageOrganizedType": "Single",
39 "shippedStatus": "AllShipped",
40 "pickupCode": "",

```

Gambar 4.3 Tampilan Order Page

Berdasarkan dokumentasi Desty, setelah mendapatkan akses ke *Order Page*, saya dapat mengakses rincian pesanan (*Order Detail*) menggunakan metode GET dengan parameter *orderId*. Langkah ini dilakukan untuk mendalami format data API penjualan secara spesifik, termasuk informasi detail mengenai transaksi, data pelanggan (*customer*), dan rincian barang yang dibeli.

```
GET https://api.desty.app/api/order/detail?orderId=1973175587475480578

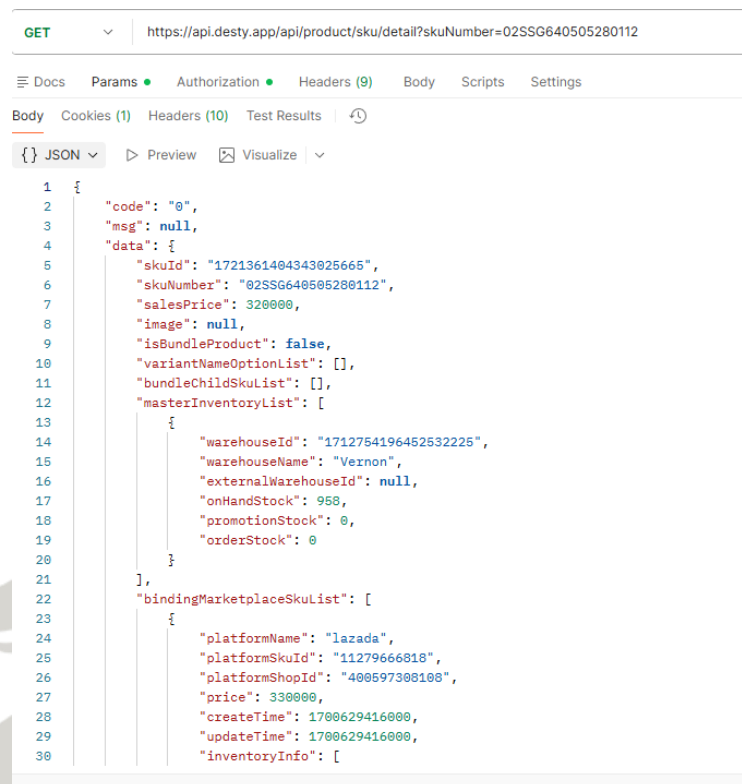
Body Cookies (1) Headers (10) Test Results Test Results

{ } JSON Preview Visualize

43      "email": null
44    },
45    "itemList": [
46      {
47        "itemOrderId": "1973175587492257793",
48        "itemId": "4457911067",
49        "itemCode": "02SSG640505280112",
50        "itemExternalCode": "02SSG640505280112",
51        "itemName": "Pompa Kolam SUNSUN 3500 Liter WP 107 Pompa Pompa Celup Hemat Listrik",
52        "orderStatus": "Completed",
53        "platformOrderStatus": "COMPLETED",
54        "description": "Pompa Kolam SUNSUN 3500 Liter WP 107 Pompa Pompa Celup Hemat Listrik",
55        "price": 320000,
56        "quantity": 1,
57        "onHandStock": 958,
58        "promotionStock": 0,
59        "orderStock": 0,
60        "discountAmount": 0,
61        "taxAmount": null,
62        "sellPrice": 245000,
63        "originalPrice": 320000,
64        "locationName": "Vernon",
65        "locationId": null,
66        "platformWarehouseId": "IDZ",
67        "platformWarehouseName": "halolo store Default Warehouse",
68        "platformWarehouseAddress": null,
69        "imageUrl": "https://cf.shopee.co.id/file/665ecc9af945936682146bc2224477c9_tn",
70        "shippingDetail": {
71          "shippingCost": 0,
72          "shippingFullName": "L****",
```

Gambar 4.4 Tampilan salah satu Get Order

Setelah mendapatkan detail pesanan melalui metode GET Order ID, saya dapat melihat informasi pelanggan, barang yang dibeli, serta harga barang tersebut. Untuk analisis yang lebih mendalam, saya juga melakukan pengecekan SKU (*Stock Keeping Unit*) barang. Pengecekan ini bertujuan untuk mengetahui stok yang tersedia di berbagai platform dan harga jualnya menggunakan metode GET SKU Number.



```
GET https://api.desty.app/api/product/sku/detail?skuNumber=02SSG640505280112

Body Cookies (1) Headers (10) Test Results Visualize

{ } JSON Preview Visualize

1 {
2   "code": "0",
3   "msg": null,
4   "data": {
5     "skuId": "1721361404343025665",
6     "skuNumber": "02SSG640505280112",
7     "salesPrice": 320000,
8     "image": null,
9     "isBundleProduct": false,
10    "variantNameOptionList": [],
11    "bundleChildSkuList": [],
12    "masterInventoryList": [
13      {
14        "warehouseId": "1712754196452532225",
15        "warehouseName": "Vernon",
16        "externalWarehouseId": null,
17        "onHandStock": 958,
18        "promotionStock": 0,
19        "orderStock": 0
20      }
21    ],
22    "bindingMarketplaceSkuList": [
23      {
24        "platformName": "lazada",
25        "platformSkuId": "11279666818",
26        "platformShopId": "400597300100",
27        "price": 330000,
28        "createTime": 1700629416000,
29        "updateTime": 1700629416000,
30        "inventoryInfo": [
```

Gambar 4.5 Tampilan Get SKU barang

Melalui tahapan persiapan data ini, saya memahami variabel-variabel yang digunakan serta format data yang ada pada API Desty. Bapak Erick, selaku CEO, menginstruksikan agar proses pembuatan *Sales Invoice* (Nota Penjualan) tidak dilakukan secara manual. Beliau menyarankan penggunaan sistem otomatisasi alur kerja (*workflow automation*) menggunakan n8n. Sistem ini dipilih karena bekerja pada server yang aktif 24 jam, sehingga setiap ada pesanan masuk melalui API, program n8n akan secara otomatis mencetak *Sales Invoice*, yang secara signifikan meningkatkan efisiensi waktu dibandingkan pengerjaan manual.

Saya juga mempersiapkan API google sheet sebagai tempat penyimpanan data sementara yang telah di ambil agar mudah dalam pengecekan hasil mapping agar tidak perlu membuka postman satu per satu yang dimana itu membutuhkan waktu lebih lagi dalam melakukan nya

	A	B	C	D	E	F	G	H
1	orderid	orderIn	bookingIn	orderType	orderStatusList	platformOrderStatusList	orderCreateTime	orderUps
2	2011608512670769153	26011541E3URR		DEFAULT_ORDER	["Ready_to_Ship"]	["READY_TO_SHIP"]	1768439786000	
3	2011621039442690049	260115447TAQDD		DEFAULT_ORDER	["Ready_to_Ship"]	["READY_TO_SHIP"]	1768442796000	
4	2011626413008814082	582124458807821322		DEFAULT_ORDER	["Ready_to_Ship"]	["AWAITING_SHIPMENT"]	17684440796000	
5	2011632083761668096	26011546P9M4		DEFAULT_ORDER	["Ready_to_Ship"]	["READY_TO_SHIP"]	17684454310000	
6	201166252721962369	2601154DEH3ACR		DEFAULT_ORDER	["Unpaid"]	["UNPAID"]	17684525860000	
7	2011674139020271618	2601154G14NKR		DEFAULT_ORDER	["Ready_to_Ship"]	["READY_TO_SHIP"]	17684554570000	
8	2011684185402441729	5821277020010915066		DEFAULT_ORDER	["Ready_to_Ship"]	["AWAITING_SHIPMENT"]	17684578530000	
9	2011822973931266049	2601155H2NSVXR		DEFAULT_ORDER	["Ready_to_Ship"]	["READY_TO_SHIP"]	17684909430000	
10	2012334973640568834	582154885641504751		DEFAULT_ORDER	["Ready_to_Ship"]	["AWAITING_SHIPMENT"]	17686130130000	
11	2012345985914937346	260117956W53WXX		DEFAULT_ORDER	["Ready_to_Ship"]	["READY_TO_SHIP"]	17686156380000	
12	2012359129203468897	26011798239YG3		DEFAULT_ORDER	["Ready_to_Ship"]	["READY_TO_SHIP"]	17686187712000	
13	201236482773956609	582156418381497386		DEFAULT_ORDER	["Ready_to_Ship"]	["AWAITING_SHIPMENT"]	17686201290000	
14	2012366154791952386	26011799M0NQ42		DEFAULT_ORDER	["Ready_to_Ship"]	["READY_TO_SHIP"]	17686204470000	
15	2012379321056448226	582157196099618808		DEFAULT_ORDER	["Ready_to_Ship"]	["AWAITING_COLLECTION"]	17686235860000	
16	2013056435041009666	260119C2W0NFVW		DEFAULT_ORDER	["Ready_to_Ship"]	["READY_TO_SHIP"]	17687850220000	

Gambar 4.6 Tampilan penyimpanan Data Pembelian Sementara di Google sheet

Dimana saya mencari api google sheet melalui google developer consule dan di hubungkan di n8n api key dan secret nya agar bisa di akses oleh n8n dan n8n harus login menggunakan gmail yang sama dengan api data sheets di situlah saya menggunakan variabel google sheet sebagai variabel *mapping* pada sistem.

## 4.2 Persiapan Software N8N

Setelah melakukan konsultasi teknis, saya memahami bahwa n8n berfungsi sebagai platform otomatisasi alur kerja (*workflow automation*). Berdasarkan riset yang saya lakukan melalui berbagai media informasi, perangkat lunak ini terbukti memiliki kapabilitas yang tinggi (*powerful*) dan sangat relevan di era digital saat ini, terutama dengan integrasi kecerdasan buatan (*AI*). n8n mampu mempercepat proses kerja manusia dan meningkatkan efisiensi operasional perusahaan.

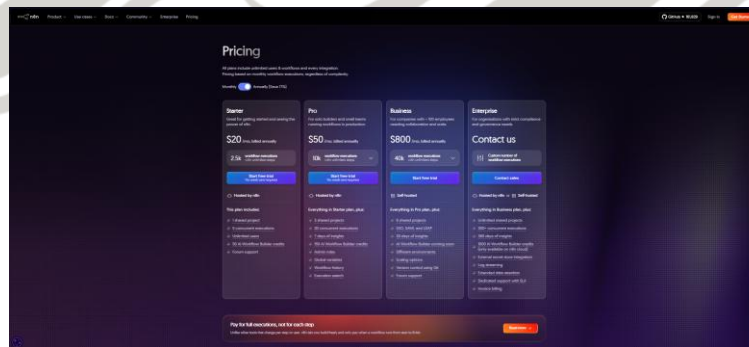


Gambar 4.7 Tampilan Logo N8N



Pada tahap awal perencanaan, saya meninjau biaya berlangganan layanan n8n *Cloud* dan menemukan adanya kendala terkait keterbatasan anggaran untuk proyek ini. Mengingat tahap ini masih berupa eksplorasi logika dan pengembangan awal, saya mencari alternatif solusi agar tetap dapat menggunakan n8n secara optimal tanpa mengeluarkan biaya lisensi yang tinggi.

Saya melakukan penelusuran informasi melalui berbagai forum komunitas pengembang dan media sosial seperti Discord, YouTube, dan Facebook. Berdasarkan hasil penelusuran tersebut, saya menemukan metode *self-hosted*, yaitu menjalankan n8n menggunakan penyimpanan internal laptop pribadi. Meskipun metode ini memiliki keterbatasan kapasitas penyimpanan dibandingkan versi *cloud*, metode ini dinilai memadai untuk kebutuhan pengembangan dan pengujian logika program secara berulang (*debugging*) selama masa praktik kerja lapangan.

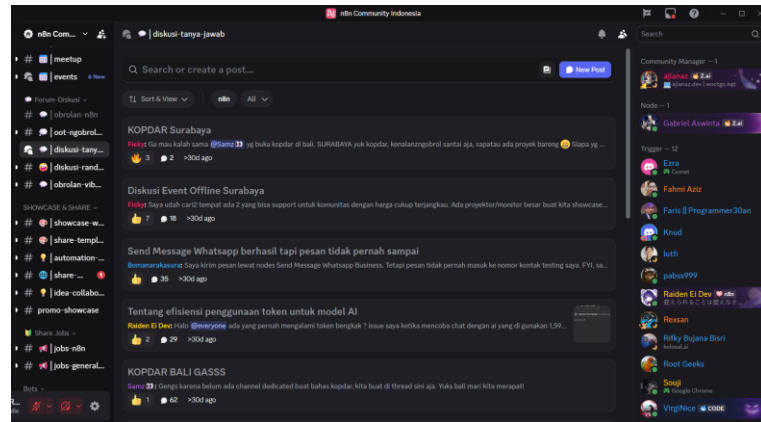


Gambar 4.8 Tampilan harga N8N

Setelah saya mendapati bahwa *software* ini memiliki harga yang tidak wajar saya berusaha untuk mencari ide bagaimana saya harus bisa mendapatkan *software* ini tanpa mengeluarkan uang sama sekali saya mencari melalui berbagai media komunikasi digital dan media sosial diantaranya adalah Dicord, Youtube, Facebook dan saya mendapatkan informasi dimana kita bisa menggunakan n8n ini tanpa mengeluarkan biaya sepersen pun namun sedikit rumit dimana kita yang n8n normalnya menggunakan penyimpanan mereka n8n *cloud* saya hanya bisa menggunakan penyimpanan internal laptop dengan minus nya penyimpanan terbatas hanya beberapa kali *running* sedangkan menggunakan n8n sebagai awalan saya harus



*running* program berkali kali setiap logika yang di jalan kan agar sesuai dengan pemahaman saya dan tidak bisa menggunakan beberapa fitur dari n8n yang terbaru.



Gambar 4.9 Tampilan Discord forum belajar N8N

Berdasarkan saran dari komunitas pengembang, saya menggunakan *Docker Desktop* untuk menjalankan n8n secara *self-hosted*. Penggunaan *Docker Desktop* memberikan lingkungan kerja yang lebih rapi, terstruktur, dan mudah dikelola. Berikut adalah rincian teknis penggunaan *Docker Desktop* dalam proyek ini:

#### 1. Fungsi Utama *Docker Desktop*

*Docker Desktop* berfungsi sebagai platform untuk menjalankan aplikasi berbasis *container*, termasuk n8n. *Container* memungkinkan aplikasi berjalan di dalam lingkungan yang sudah dikonfigurasi secara lengkap, sehingga proses instalasi dan pengoperasian menjadi lebih mudah. Fungsi utamanya meliputi:

- Menyediakan Lingkungan Kerja yang Terisolasi *Docker* menyediakan ruang terpisah untuk menjalankan n8n, sehingga aplikasi berjalan dengan konfigurasi yang konsisten dan tidak bergantung pada konfigurasi sistem operasi utama laptop.
- Mempermudah Instalasi dan Pengelolaan n8n Melalui fitur *Docker Extensions*, n8n dapat diinstal dengan cepat tanpa memerlukan konfigurasi manual yang rumit.
- Mendukung Penyimpanan Data yang Terstruktur *Docker* menyediakan mekanisme *volume* untuk menyimpan seluruh *data workflow*, kredensial, dan

konfigurasi n8n. Hal ini memastikan data tetap aman dan tidak hilang meskipun *container* dimodifikasi atau dijalankan ulang.

- Memudahkan Pengembangan dan Pengujian Lingkungan yang terkontrol memungkinkan setiap *workflow* diuji secara konsisten. Hal ini sangat membantu saya dalam melakukan pengembangan automasi secara berulang dengan hasil yang stabil.

2. Integrasi *Docker Desktop* dengan *Ngrok* Dalam beberapa skenario *workflow*, n8n memerlukan URL publik agar dapat menerima permintaan (*request*) dari layanan eksternal. *Docker Desktop* mendukung kebutuhan ini melalui integrasi dengan *Ngrok Extension*. *Ngrok* berfungsi membuat jalur aman (*tunneling*) dari *localhost* (komputer lokal saya) ke internet publik.

Peran *Ngrok* sangat krusial dalam mekanisme penerimaan data, khususnya untuk:

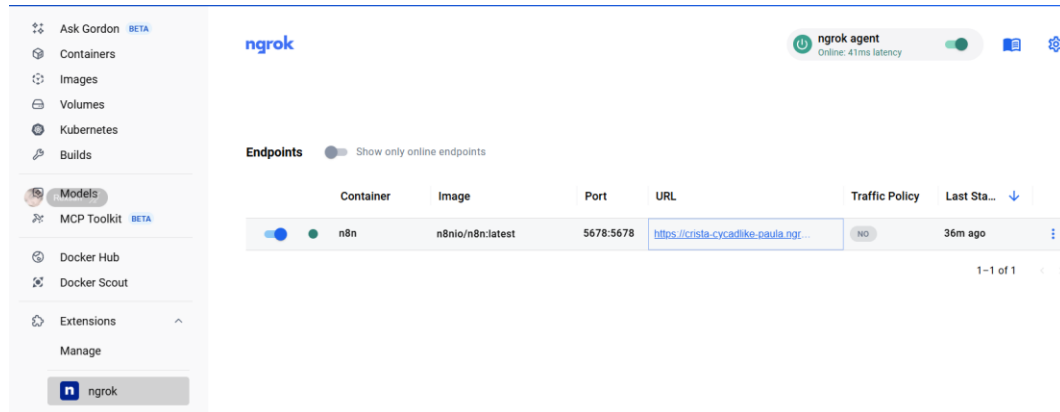
- Menangani *Webhook* dari Aplikasi Lain (seperti Desty): *Webhook* adalah mekanisme komunikasi antar-sistem yang berbasis *event* (kejadian). Berbeda dengan metode konvensional di mana sistem harus menanyakan data secara berkala (*polling*), *webhook* memungkinkan aplikasi sumber (Desty) untuk mengirimkan data secara otomatis dan *real-time* ke n8n tepat saat peristiwa terjadi (misalnya: saat ada pesanan baru masuk).

Karena n8n berjalan di lingkungan lokal (*localhost*) yang tertutup, server Desty tidak dapat mengirim data secara langsung. Di sinilah peran *Ngrok*, yang menyediakan alamat publik sementara (*public endpoint*) untuk meneruskan sinyal *webhook* tersebut masuk ke dalam container n8n di *Docker Desktop*.

- Menerima Data Eksternal secara Real-time: Dengan adanya jalur *tunneling* ini, sinkronisasi data pesanan dan stok antara *marketplace* dan sistem ERP dapat terjadi seketika tanpa jeda waktu yang signifikan.

### 3. Penggunaan n8n Extension di Docker Desktop

*Docker Desktop* menyediakan *n8n Extension* yang mengotomatisasi proses konfigurasi. Ekstensi ini secara otomatis mengunduh *image* n8n, membuat *container* siap pakai, menyiapkan penyimpanan data, dan menampilkan status *container*. Dengan adanya ekstensi ini, proses implementasi n8n menjadi lebih efisien dan praktis.



Gambar 4.10 Tampilan Docker Desktop dan Ngrok

Dengan konfigurasi tersebut, saya dapat menggunakan n8n tanpa biaya (*free*) menggunakan sumber daya lokal, serta memanfaatkan domain publik sementara dari *Ngrok* untuk keperluan koneksi API.

### 4.3 Penggunaan Software

Tahap implementasi dimulai dengan menerjemahkan logika pengujian API yang sebelumnya dilakukan menggunakan Postman ke dalam sistem otomatisasi n8n. saya merancang alur kerja (*workflow*) yang mencakup siklus data lengkap: pengambilan daftar pesanan (*order page*), penarikan detail pesanan, identifikasi SKU, hingga pembuatan faktur penjualan (*Sales Invoice*) di ERPNext.

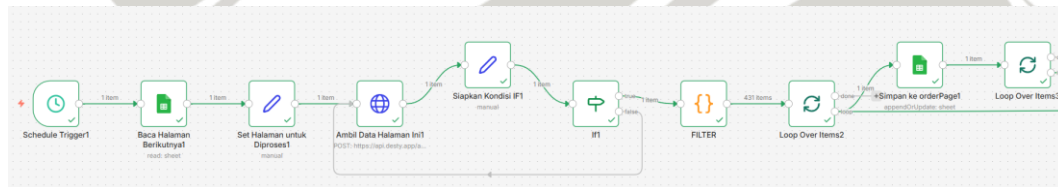
Saya mempelajari dan memanfaatkan berbagai jenis *node* pada n8n, mulai dari *HTTP Request* untuk komunikasi API, *Code Node* untuk manipulasi data menggunakan JavaScript, hingga *Google Sheets Node* sebagai basis data sementara. Pendekatan ini memungkinkan sistem bekerja secara siklik (*looping*), di mana indeks halaman dipantau secara otomatis. Indeks ini akan bertambah satu (+1) setiap kali rangkaian

proses pembuatan faktur berhasil diselesaikan, memastikan tidak ada data yang terlewat atau terduplikasi.

#### 4.4 Struktur Workflow Utama

*Workflow* yang dibangun terdiri dari beberapa bagian utama yang saling terhubung untuk membentuk sistem otomatis pengambilan dan pemrosesan data pesanan. Setiap bagian memiliki fungsi spesifik yang dirancang untuk menangani aspek tertentu dari proses bisnis, mulai dari pemicu otomatis hingga penyimpanan data ke *database*.

berikut memperlihatkan struktur awal *workflow* yang mencakup pengaturan *trigger* dan mekanisme *pagination*.



Gambar 4.11 Alur Logika Pengambilan Data Pesanan pada n8n

Bagian Trigger dan Manajemen Pagination Workflow dimulai dengan Schedule Trigger yang dijalankan setiap 10 detik secara otomatis. Interval ini dipilih berdasarkan pertimbangan keseimbangan antara kecepatan pembaruan data real-time dan beban permintaan (load) pada server API.

*Node* pertama setelah *trigger* adalah "Baca Halaman Berikutnya1" yang menggunakan Google Sheets untuk membaca memori *pagination*. Lembar kerja ini berfungsi sebagai basis data sederhana yang menyimpan dua informasi penting:

1. *nextPage*: Nomor halaman yang akan diproses berikutnya.
2. *lastSyncTime*: Waktu terakhir sinkronisasi dilakukan.

Struktur data yang disimpan di Google Sheet Memory adalah sebagai berikut:

```
JSON
{
  "key": "nextPage",
  "nextPage": "1",
  "lastSyncTime": "2024-12-12T10:30:00Z"
}
```

Gambar 4.12 Code untuk menyimpan di Google Sheet Memory

Data dari *sheet* kemudian diteruskan ke *node* "Set Halaman untuk Diproses1" yang bertugas mengatur parameter halaman untuk permintaan API berikutnya. Proses ini penting untuk memastikan bahwa setiap kali *workflow* berjalan, sistem mengetahui secara persis halaman mana yang harus diproses, sehingga mencegah terjadinya duplikasi data.

Pengambilan Data Order *Node* "Ambil Data Halaman Ini1" melakukan permintaan HTTP POST ke API Desty pada *endpoint* *order/page*. Permintaan ini mengirimkan parameter berupa nomor halaman, ukuran halaman (500 order per *request*), serta rentang waktu pesanan.

Format permintaan (*request*) yang dikirimkan adalah:

```
JSON
POST https://api.desty.app/api/order/page
{
  "pageNumber": 1,
  "pageSize": 500,
  "orderCreateTimeStart": "2024-12-11T00:00:00Z",
  "orderCreateTimeEnd": "2024-12-12T23:59:59Z"
}
```

Gambar 4.13 Code untuk Pengambilan Data Order Page

Respons dari API berisi *array* pesanan yang kemudian perlu disaring karena tidak semua pesanan harus diproses. *Node* "If1" melakukan filtrasi dengan dua kondisi utama:

1. Pesanan Aktif: Pesanan yang tidak berstatus *CANCELLED*.

2. Pesanan Dibatalkan Berbayar: Pesanan dengan status *CANCELLED* tetapi sudah dibayar (*hasPaid* = *true*). Pesanan jenis ini tetap diproses untuk keperluan pencatatan pengembalian dana (*refund*) atau rekonsiliasi keuangan.

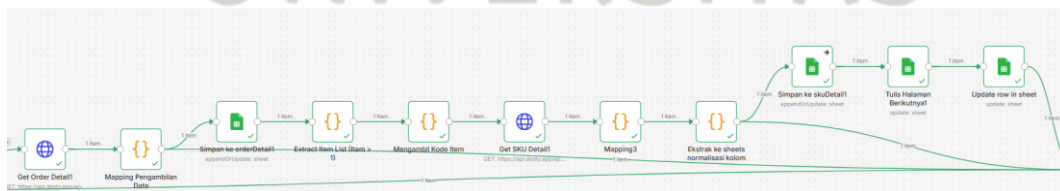
Node "Ekstrak Pesanan1" memproses hasil filter tersebut menggunakan kode JavaScript berikut:

```
JavaScript
const filteredOrders = page.data.results.filter(order => {
  const statusList = order.platformOrderStatusList || [];
  const isCancelled = statusList.includes('CANCELLED');
  const hasPaid = order.hasPaid === true;

  return !isCancelled || (isCancelled && hasPaid);
});
```

Gambar 4.14 Code untuk Filter data Order

Logika ini krusial untuk menjaga integritas data keuangan perusahaan, memastikan hanya transaksi yang memiliki nilai akuntansi yang dicatat. Iterasi (*Looping*) dan *Detail Order* Setelah mendapatkan daftar pesanan yang *valid*, *workflow* memasuki tahap pemrosesan detail. Bagian ini sangat penting karena data dari halaman pesanan (*order page*) belum memuat informasi rinci seperti alamat lengkap pelanggan atau rincian diskon per item. Berikut ini mengilustrasikan proses *looping* dan pengambilan detail pesanan.



Gambar 4.15 Alur Pemrosesan Detail Pesanan dan Normalisasi SKU

*Workflow* menggunakan *node* "Loop Over Items2" untuk memproses setiap pesanan secara individual. Fitur *split in batches* ini memungkinkan pemrosesan data satu per satu secara sekuensial untuk menghindari konflik data. Di dalam *loop*, *node* "Get Order Detail1" melakukan permintaan HTTP GET:

HTTP

GET <https://api.desty.app/api/order/detail?orderId={orderId}>

Data detail yang diterima kemudian diproses oleh *node* "Mapping2" untuk mentransformasi struktur data *nested object* menjadi format datar (*flat*) yang sesuai untuk penyimpanan basis data. Hasilnya disimpan ke Google Sheet "Detail Pesanan" menggunakan operasi *appendOrUpdate* untuk mencegah duplikasi.

Ekstraksi dan Normalisasi Item (SKU) Tantangan teknis ditemukan pada variasi format data item dari API. *Node* "Extract Item List1" menangani dua kemungkinan struktur data:

1. Item tersimpan dalam *array* *orderItems*.
2. Item tergabung langsung dalam objek pesanan tunggal.

Kode JavaScript digunakan untuk menormalisasi kondisi tersebut:

```
JavaScript
if (Array.isArray(orderData.orderItems) && orderData.orderItems.length > 0) {
  itemsToProcess.push(...orderData.orderItems);
} else if (orderData.itemOrderId) {
  itemsToProcess.push(orderData);
}
```

Gambar 4.16 Code untuk ekstrak dan Normalisasi item (SKU)

Selanjutnya, *node* "Code in JavaScript1" menangani kasus di mana beberapa kode barang digabung dalam satu *string* (misalnya pada pesanan *bundling*). Kode barang dipisahkan menggunakan fungsi *.split(',')* agar setiap SKU dapat divalidasi secara terpisah.

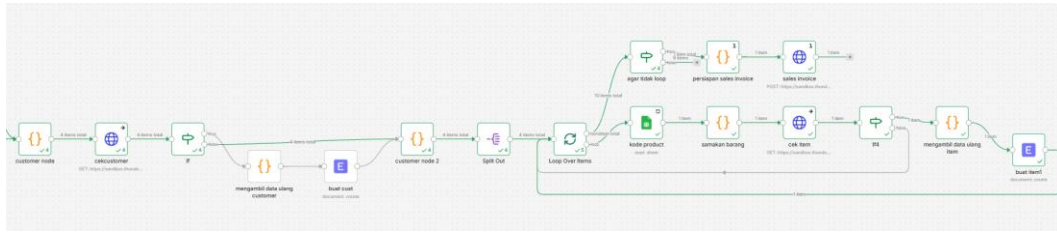
*Node* "Mapping3" kemudian melakukan denormalisasi data SKU. Karena satu SKU bisa terhubung ke banyak *marketplace*, saya membuat logika kolom dinamis (*platformName\_1*, *platformName\_2*, *dst.*) agar data hierarkis tersebut dapat disimpan dalam format tabel Google Sheets yang statis.

#### 4.5 Workflow Pembuatan Sales Invoice

Setelah seluruh data pesanan dan SKU berhasil dikumpulkan dan dinormalisasi, *workflow* dilanjutkan ke tahap integrasi dengan sistem ERPNext untuk pembuatan *Sales Invoice*. Bagian ini mengintegrasikan seluruh data yang telah diproses menjadi dokumen legal penjualan.



Berikut memperlihatkan alur integrasi akhir, mulai dari validasi pelanggan hingga *submit* faktur ke ERPNext.



Gambar 4.17 Alur Integrasi Logika Percabangan dan Pembuatan Sales Invoice ke ERPNext

Persiapan dan Validasi Pelanggan (*Customer*) *Node* "customer node" bertugas melakukan sanitasi data pelanggan. Nama pelanggan dibersihkan dari karakter *non-standart*, dan jika kosong, sistem memberikan label "Guest Customer". Selanjutnya, *node* "cekcustomer" melakukan validasi ke ERPNext untuk memeriksa apakah pelanggan tersebut sudah terdaftar.

- Jika data ditemukan, sistem menggunakan ID pelanggan yang ada.
- Jika data tidak ditemukan, *workflow* mengarahkan ke *node* "buat cust" untuk membuat data master pelanggan baru secara otomatis.

*Payload* pembuatan pelanggan baru mencakup parameter *custom\_platform* untuk kebutuhan pelacakan asal transaksi:

```
JSON
{
  "doctype": "Customer",
  "customer_name": customerName,
  "customer_type": "Individual",
  "territory": "Indonesia",
  "custom_platform": orderInfo.platformName
}
```

Gambar 4.18 Code persiapan dan validasi Customer

Pengayaan Data dan Logika Harga *Node* "customer node 2" melakukan penggabungan dan pengayaan data (*data enrichment*), terutama terkait harga. Mengingat API *marketplace* menyediakan berbagai variabel harga (harga asli, harga diskon, harga setelah potongan), saya menerapkan logika prioritas:



```
JavaScript
const sellPrice = parseFloat(item.sellPrice)
  parseFloat(item.rate)
  parseFloat(item.price);
```

Gambar 4.19 Code memnentukan harga jual

Sistem memprioritaskan *Sell Price* (harga jual final) untuk memastikan nilai faktur sesuai dengan nominal transaksi riil. Pencocokan Barang (*Item Matching*) Node "samakan barang" melakukan pencocokan antara kode produk dari *marketplace* dengan kode internal ERPNext menggunakan data referensi dari Google Sheets ("KODE PRODUCT").

- Jika kode cocok (*match*), sistem menggunakan data internal (Nama Barang, Gudang, Satuan).
- Jika tidak cocok, sistem menggunakan data asli dari API.

Setelah itu, dilakukan pengecekan ke sistem ERPNext. Jika barang belum terdaftar sebagai *Master Item*, node "buat item1" akan membuatnya secara otomatis dengan konfigurasi akun *standart* (*Income Account*: Penjualan, *Expense Account*: HPP).

Kalkulasi dan Pembuatan Faktur Tahap terakhir dilakukan oleh node "persiapan sales invoice" yang melakukan agregasi dan kalkulasi finansial akhir.

Rumus perhitungan yang diterapkan adalah:

*Payload* JSON disusun lengkap dengan komponen pajak dan pembayaran:

```
JSON
{
  "doctype": "Sales Invoice",
  "customer": customerId,
  "items": invoiceItems,
  "taxes": taxes,           // Service Fee dicatat sebagai pajak negatif
  "payments": payments,    // Terisi otomatis jika hasPaid = true
  "grand_total": grandTotal,
  "docstatus": 1           // Langsung Submit (Final)
}
```

Gambar 4.20 Code Payload membuat Faktur Penjualan

Parameter *docstatus*: 1 memastikan faktur langsung berstatus *Submitted*, sehingga stok barang berkurang dan jurnal akuntansi terbentuk secara otomatis tanpa intervensi manual.

#### 4.6 Kesimpulan Implementasi Workflow

Secara keseluruhan, sistem yang dibangun ini berhasil mentransformasi proses verifikasi data yang sebelumnya dilakukan secara manual dan bertingkat menjadi sebuah alur kerja otomatis yang terintegrasi. Sebelum adanya sistem ini, admin harus melakukan pengecekan API satu per satu menggunakan Postman secara berjenjang—mulai dari menarik *Order Page*, menelusuri *Order Detail*, hingga memverifikasi SKU yang sangat memakan waktu dan rentan kesalahan.

*Workflow* ini memangkas kompleksitas tersebut dengan menerapkan logika validasi cerdas pada dua entitas utama, yaitu Pelanggan (*Customer*) dan Barang (*Item*):

1. Validasi Pelanggan (*Customer Node*): Sistem menerapkan mekanisme pemeriksaan otomatis ke database ERPNext. Jika data pelanggan sudah tersedia, sistem akan langsung menggunakan ID pelanggan tersebut (skip proses pembuatan). Sebaliknya, jika data tidak ditemukan, sistem secara otomatis mendaftarkan pelanggan baru.
2. Validasi Barang (*Item Node*): Mekanisme serupa diterapkan pada produk. Sistem mencocokkan kode SKU dari *marketplace* dengan kode internal perusahaan. Jika barang belum terdaftar di ERPNext, sistem akan membuatnya secara otomatis (*auto-create*), namun jika sudah ada, sistem hanya akan mengambil referensi datanya.

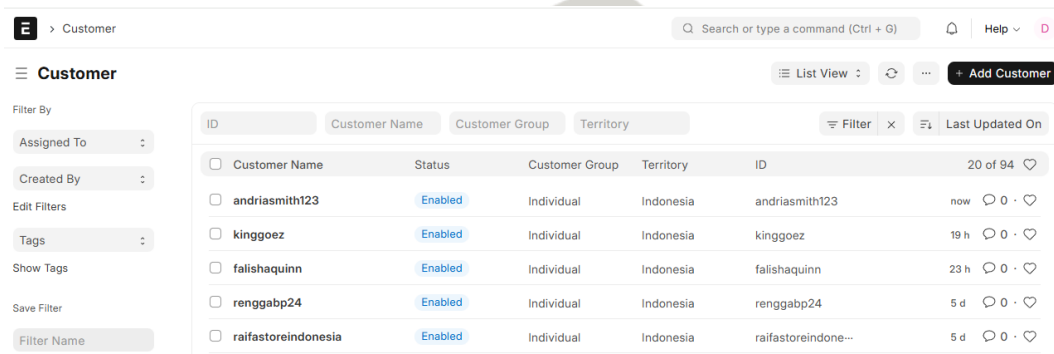
Dengan mekanisme tersebut, sistem tidak hanya mempercepat proses pembuatan *Sales Invoice*, tetapi juga menjamin integritas data master di ERPNext tanpa memerlukan intervensi manual yang berulang.

#### 4.7 Hasil Implementasi pada Sistem ERPNext

Setelah seluruh alur kerja (*workflow*) pada n8n dijalankan, data yang diambil dari API akan diproses dan dikirimkan ke sistem ERP perusahaan (ERPNext). Berikut adalah rincian *output* yang dihasilkan dari implementasi sistem otomatisasi ini:

## 1. Pembuatan Data Pelanggan (*Customer Creation*)

Langkah pertama yang dilakukan sistem setelah berhasil mengambil data pesanan adalah memverifikasi data pelanggan. Jika pelanggan belum terdaftar, sistem secara otomatis membuat profil pelanggan baru (*Customer Master*) di ERPNext. Hal ini bertujuan untuk menyimpan identitas pembeli secara terstruktur, sehingga memudahkan pelacakan riwayat transaksi apabila pelanggan yang sama melakukan pembelian kembali di masa mendatang.



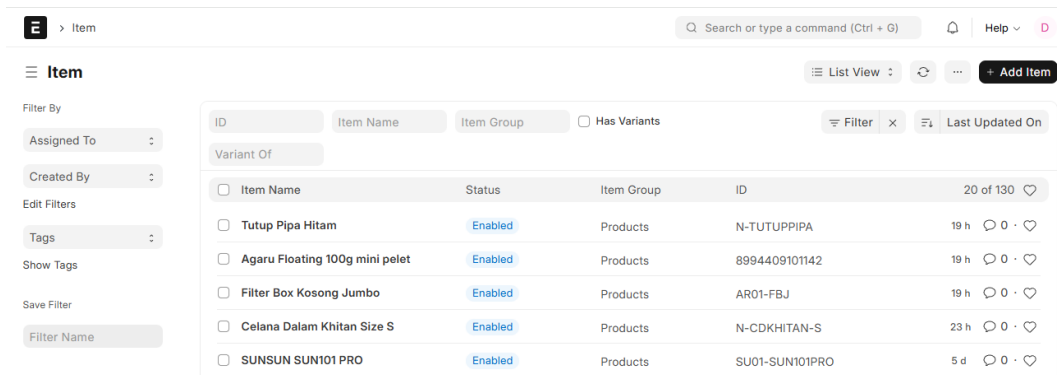
ID	Customer Name	Status	Customer Group	Territory	ID	Last Updated On
<input type="checkbox"/>	Customer Name	Status	Customer Group	Territory	ID	20 of 94
<input type="checkbox"/>	andriasmith123	Enabled	Individual	Indonesia	andriasmith123	now
<input type="checkbox"/>	kinggoez	Enabled	Individual	Indonesia	kinggoez	19 h
<input type="checkbox"/>	falishaquinn	Enabled	Individual	Indonesia	falishaquinn	23 h
<input type="checkbox"/>	renggabp24	Enabled	Individual	Indonesia	renggabp24	5 d
<input type="checkbox"/>	raifastoreindonesia	Enabled	Individual	Indonesia	raifastoreindone...	5 d

Gambar 4.21 Tampilan Output Pembuatan Customer pada ERPNext

## 2. Pembuatan dan Verifikasi Data Barang (*Item Creation*)

Setelah data pelanggan diproses, "workflow" akan memvalidasi data barang (*Item*). Sistem akan mengecek ketersediaan *Item Code* atau SKU dari *marketplace* di dalam basis data ERPNext.

- Jika barang sudah terdaftar, sistem akan menggunakan *data master* yang sudah ada.
- Jika barang belum terdaftar, sistem akan membuat *Item Master* baru secara otomatis sesuai dengan deskripsi dari API. Mekanisme ini mencegah terjadinya duplikasi data barang dan memastikan stok tercatat dengan kode yang konsisten.



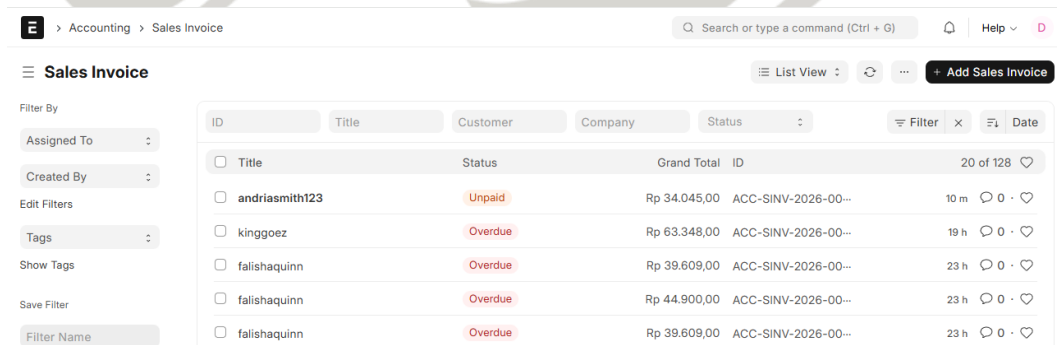
The screenshot shows the ERPNext Item list view. The left sidebar contains filters for 'Assigned To', 'Created By', 'Tags', and 'Filter Name'. The main table lists items with columns for ID, Item Name, Status, Item Group, and ID. The items listed are:

ID	Item Name	Status	Item Group	ID
	Tutup Pipa Hitam	Enabled	Products	N-TUTUPPIPA
	Agaru Floating 100g mini pelet	Enabled	Products	8994409101142
	Filter Box Kosong Jumbo	Enabled	Products	AR01-FBJ
	Celana Dalam Khitan Size S	Enabled	Products	N-CDKHITAN-S
	SUNSUN SUN101 PRO	Enabled	Products	SU01-SUN101PRO

Gambar 4.22 Tampilan Output Pembuatan Item pada ERPNext

### 3. Pembuatan Faktur Penjualan (*Sales Invoice*)

Tahap akhir dari *workflow* adalah penggabungan data pelanggan dan data barang menjadi dokumen legal berupa Faktur Penjualan (*Sales Invoice*). Pada tahap ini, logika perhitungan harga dijalankan secara otomatis. Sistem mengkalkulasi harga satuan, jumlah barang (kuantitas), serta komponen pajak yang relevan.



The screenshot shows the ERPNext Sales Invoice list view. The left sidebar contains filters for 'Assigned To', 'Created By', 'Tags', and 'Filter Name'. The main table lists sales invoices with columns for ID, Title, Customer, Company, Status, Grand Total, ID, and Date. The invoices listed are:

ID	Title	Customer	Company	Status	Grand Total	ID
	andriasmith123			Unpaid	Rp 34.045,00	ACC-SINV-2026-00--
	kinggoez			Overdue	Rp 63.348,00	ACC-SINV-2026-00--
	falishaquinn			Overdue	Rp 39.609,00	ACC-SINV-2026-00--
	falishaquinn			Overdue	Rp 44.900,00	ACC-SINV-2026-00--
	falishaquinn			Overdue	Rp 39.609,00	ACC-SINV-2026-00--

Gambar 4.23 Tampilan Output Sales Invoice pada ERPNext

### 4. Rincian Detail Faktur

Selain data utama, dokumen *Sales Invoice* yang dihasilkan juga memuat informasi mendetail (*Item Details*). Sistem memastikan bahwa harga yang tercantum sesuai dengan harga jual akhir (*Sell Price*) dari *marketplace*, termasuk detail pemotongan stok di gudang. Dokumen ini dibuat dengan status langsung "Submitted", sehingga jurnal akuntansi dan pengurangan stok terjadi secara *real-time*.

E

> Accounting > Sales Invoice > ACC-SINV-2026-00005

Search or type a command (Ctrl + G)

Help

minutes ago

Details
 Payments
 Address & Contact
 More Info
 Connections

Items
 Update Stock

Items
 

	No.	Item	Quantity	Rate (IDR) *	Amount (IDR) *	
	1	N-BBWRAP40-B: Bubble Wrap 1 Roll Hitam Putih...	1	Rp 39.000,00	Rp 39.000,00	

Download

Total Quantity

1

Total (IDR)

Rp 39.000,00

Sales Taxes and Charges

	No.	Type *	Account Head *	Tax Rate	Amount (IDR)	Total (IDR)	
	1	Actual	4110.000 - Penjualan ...	0	Rp -4.955,00	Rp 34.045,00	

Total Taxes and Charges (IDR)

Rp -4.955,00

Totals

Grand Total (IDR) \*

Rp 34.045,00

Gambar 4.24 Tampilan Detail Dokumen Sales Invoice

UNIVERSITAS  
MA CHUNG

38

## Bab V

### Penutup

#### 5.1 Kesimpulan

Berdasarkan perancangan, implementasi, dan pengujian *workflow* otomatisasi yang telah dilaksanakan selama kegiatan Praktik Kerja Lapangan di Vernon Corp, dapat ditarik beberapa simpulan sebagai berikut:

1. Efektivitas Otomasi Workflow dengan n8n Pemanfaatan n8n sebagai *workflow engine* terbukti berhasil mengintegrasikan sistem eksternal (API Desty) dengan sistem internal perusahaan (ERPNext) secara *seamless*. Sistem yang dibangun mampu menjalankan siklus pengambilan data pesanan, ekstraksi detail, hingga pembuatan dokumen *Sales Invoice* secara otomatis tanpa intervensi manual, yang sejalan dengan instruksi CEO untuk meningkatkan efisiensi waktu operasional.
2. Peningkatan Integritas dan Akurasi Data Penerapan mekanisme manajemen data melalui tahapan *filtering* dan normalisasi terbukti krusial dalam menjaga kebersihan data. Logika penyaringan yang memisahkan pesanan "Cancelled" namun tetap memproses pesanan "Cancelled but Paid" memastikan pencatatan keuangan tetap akurat. Selain itu, teknik normalisasi SKU menggunakan *mapping* dinamis berhasil mengatasi heterogenitas format data dari berbagai *marketplace*.
3. Efisiensi Biaya Melalui Infrastruktur *Self-Hosted* Kendala biaya lisensi layanan n8n *cloud* berhasil diatasi melalui implementasi metode *self-hosted* menggunakan Docker Desktop dan integrasi Ngrok. Pendekatan ini memungkinkan saya untuk melakukan pengembangan, pengujian berulang (*debugging*), dan menjalankan sistem operasional tanpa mengeluarkan biaya lisensi perangkat lunak, namun tetap memiliki lingkungan kerja yang terisolasi dan aman.

4. Keandalan Sistem dalam Penanganan Data *Workflow* yang dirancang memiliki ketahanan (*robustness*) yang baik berkat penerapan manajemen kesalahan (*error handling*). Penggunaan fitur *Split in Batches* untuk pemrosesan data secara sekuensial dan opsi *Continue On Fail* pada *node* validasi mencegah kegagalan total pada sistem saat terjadi anomali data. Hal ini memastikan operasional pembuatan faktur berjalan stabil meskipun menangani volume data yang besar.

## 5.2 Saran

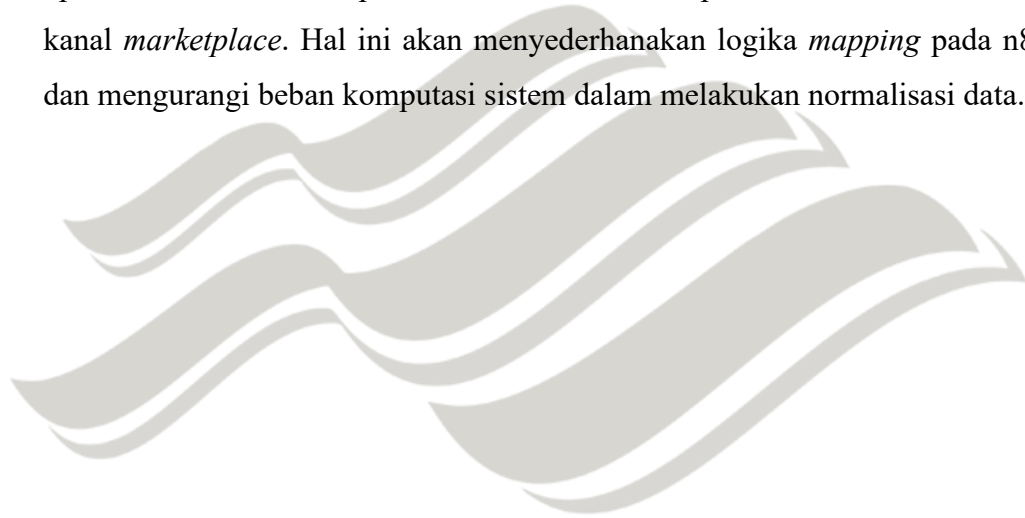
Berdasarkan pengalaman selama pengembangan dan implementasi sistem, terdapat beberapa saran yang dapat diberikan untuk pengembangan lebih lanjut dan perbaikan di masa mendatang:

1. Migrasi ke Server Terdedikasi (VPS) Saat ini, sistem n8n dijalankan secara *self-hosted* menggunakan penyimpanan internal laptop pribadi dengan keterbatasan kapasitas dan ketergantungan pada perangkat keras lokal. Untuk penggunaan jangka panjang dan skala produksi yang lebih besar, disarankan agar perusahaan memigrasikan sistem ini ke *Virtual Private Server* (VPS) atau server lokal kantor yang didedikasikan agar sistem dapat berjalan 24/7 dengan stabilitas dan keamanan yang lebih terjamin.
2. Pengembangan Fitur Rekonsiliasi Stok dan Pengiriman Sistem yang dikembangkan saat ini berfokus pada pengambilan data pesanan dan pembuatan *Sales Invoice*. Disarankan untuk mengembangkan *workflow* lanjutan yang mencakup sinkronisasi stok otomatis (*inventory sync*) antara ERPNext dan *marketplace* serta modul pelacakan pengiriman (*shipping tracking*) untuk memberikan visibilitas operasional yang lebih menyeluruh.
3. Implementasi *Dashboard Monitoring* Meskipun sistem berjalan otomatis, pemantauan kesehatan *workflow* masih dilakukan secara teknis. Disarankan untuk membuat *dashboard* sederhana atau notifikasi otomatis (misalnya via



Telegram atau Email) yang memberikan laporan harian mengenai jumlah faktur yang berhasil dibuat dan daftar *error* yang terjadi, sehingga tim operasional dapat merespons kendala :val="2" lebih cepat.

4. Standarisasi Kode SKU di Marketplace Ditemukan bahwa variasi *format data item* dari API menjadi tantangan teknis tersendiri. Disarankan agar tim operasional Vernon Corp melakukan standarisasi penamaan SKU di seluruh kanal *marketplace*. Hal ini akan menyederhanakan logika *mapping* pada n8n dan mengurangi beban komputasi sistem dalam melakukan normalisasi data.



UNIVERSITAS  
MA CHUNG

## Daftar Pustaka

(Vernon, n.d.)

Hidayat, R. D. A., Hadi, E. A., Rusnandi. Enang, Rekayasa Perangkat Lunak, T., Mardira Indonesia, P., & Rekayasa Rekayasa Perangkat Lunak, T. (2025). *JMTAS: Journal of Multimedia Technology and Applied Software Automation of Technical Documentation Validation Workflows Based on GitHub Using n8n During Software Development Stages*.

Nakavisute, I., & Sincharoonsak, T. (2025). *OPTIMIZING THE AUTOMATION PROCESS WITH N8N*. 32(S8), 2025. <https://github.com/n8n/n8n>

Nugraha, R. I., Supriyanto, & Sugiharto, S. (2024). Perancangan Sistem Informasi Faktur Penjualan pada toko Sari Agung Semarang. *Pixel :Jurnal Ilmiah Komputer Grafis*, 17(1), 159–167. <https://doi.org/10.51903/pixel.v17i1.2005>

Wisswani, N. W. (2016). ARSITEKTUR BUSSINESS PROCESS AUTOMATION BAGI PROSES PEMASARAN PRODUK. In *JULI* (Vol. 16, Issue 2).

UNIVERSITAS  
MA CHUNG