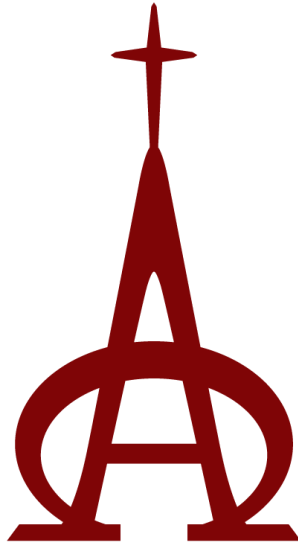


# **LAPORAN PROYEK AKHIR**

## **MANUFAKTUR II**

**IEE3111**



**Martin Emmanuel Chang**

**212100199**

**Moses Anthony Kwik**

**212100291**

**Wesley Hakim**

**212100211**

**INTERNET OF THINGS AND ELECTRICAL ENGINEERING**  
**CALVIN INSTITUTE OF TECHNOLOGY**

**2023**

# DAFTAR ISI

1.	UMUM .....	4
2.	PERCOBAAN.....	5
2.1	Penjelasan Sistem .....	5
2.2	Sistem Kanban .....	7
2.2.1	Order Station .....	7
2.2.2	Material Handling Station .....	8
2.2.2.1	Foto rangkaian .....	8
2.2.3	Storage Station.....	9
2.2.3.1	Foto rangkaian .....	9
2.2.3.2	Penjelasan .....	10
2.2.4	Warehouse Station.....	11
2.2.4.1	Foto rangkaian .....	11
2.2.4.2	Penjelasan .....	12
2.2.5	Assembly Station #1 .....	12
2.2.5.1	Foto rangkaian .....	12
2.2.5.2	Penjelasan .....	14
2.2.6	Assembly Station #2 .....	15
2.2.6.1	Foto rangkaian .....	15
2.2.6.2	Penjelasan .....	16
2.2.7	Quality Control Station .....	17
2.2.7.1	Foto rangkaian .....	17
2.2.7.2	Penjelasan .....	18
2.3	Node-RED .....	19
2.4	Database .....	21
3.	ANALISIS PERCOBAAN .....	23
3.1	Kendala Selama Proyek .....	23
3.2	Pemborosan sistem.....	25
4.	Kesimpulan .....	26
5.	Kontribusi Anggota .....	27

## DAFTAR GAMBAR

Gambar 1. Visualisasi proses kanban order .....	5
Gambar 2. Order station.....	7
Gambar 3. Rangkaian material handling station .....	8
Gambar 4. Rangkaian storage station (out) .....	9
Gambar 5. Rangkaian storage station (in).....	9
Gambar 6. Warehouse ketika menampilkan jumlah material yang tersimpan .....	11
Gambar 7. Warehouse ketika menampilkan jumlah material yang dibutuhkan untuk produk yang dibaca .....	11
Gambar 8. Rangkaian assembly station 1.....	12
Gambar 9. Tampilan prosedur assembly produk Z di assembly station 1 .....	13
Gambar 10. Tampilan prosedur assembly produk Y di assembly station 1.....	13
Gambar 11. Tampilan prosedur assembly produk X di assembly station 1.....	14
Gambar 12. Rangkaian assembly station 2.....	15
Gambar 13. Tampilan prosedur assembly produk X di assembly station 2 .....	15
Gambar 14. Tampilan prosedur assembly produk Y di assembly station 2 .....	16
Gambar 15. Tampilan prosedur assembly produk Z di assembly station 2.....	16
Gambar 16. Rangkaian quality control station.....	17
Gambar 17. Tampilan produk X di quality control station .....	18
Gambar 18. Tampilan produk Y di quality control station .....	18
Gambar 19. Tampilan produk Z di quality control station .....	18
Gambar 20. Tangkapan layar flow Node-RED .....	20
Gambar 21. Dashboard Node-RED .....	20
Gambar 22. Menampilkan semua isi tabel progress .....	21
Gambar 23. Menampilkan semua isi tabel storage .....	21
Gambar 24. Menampilkan semua isi tabel inventory .....	22
Gambar 25. Menampilkan semua isi tabel order_table.....	22

# 1. UMUM

Judul Proyek UAS : *Digitalized Toyota Production System*

Semester/Tahun : Genap/2023-2024

## Tujuan Proyek

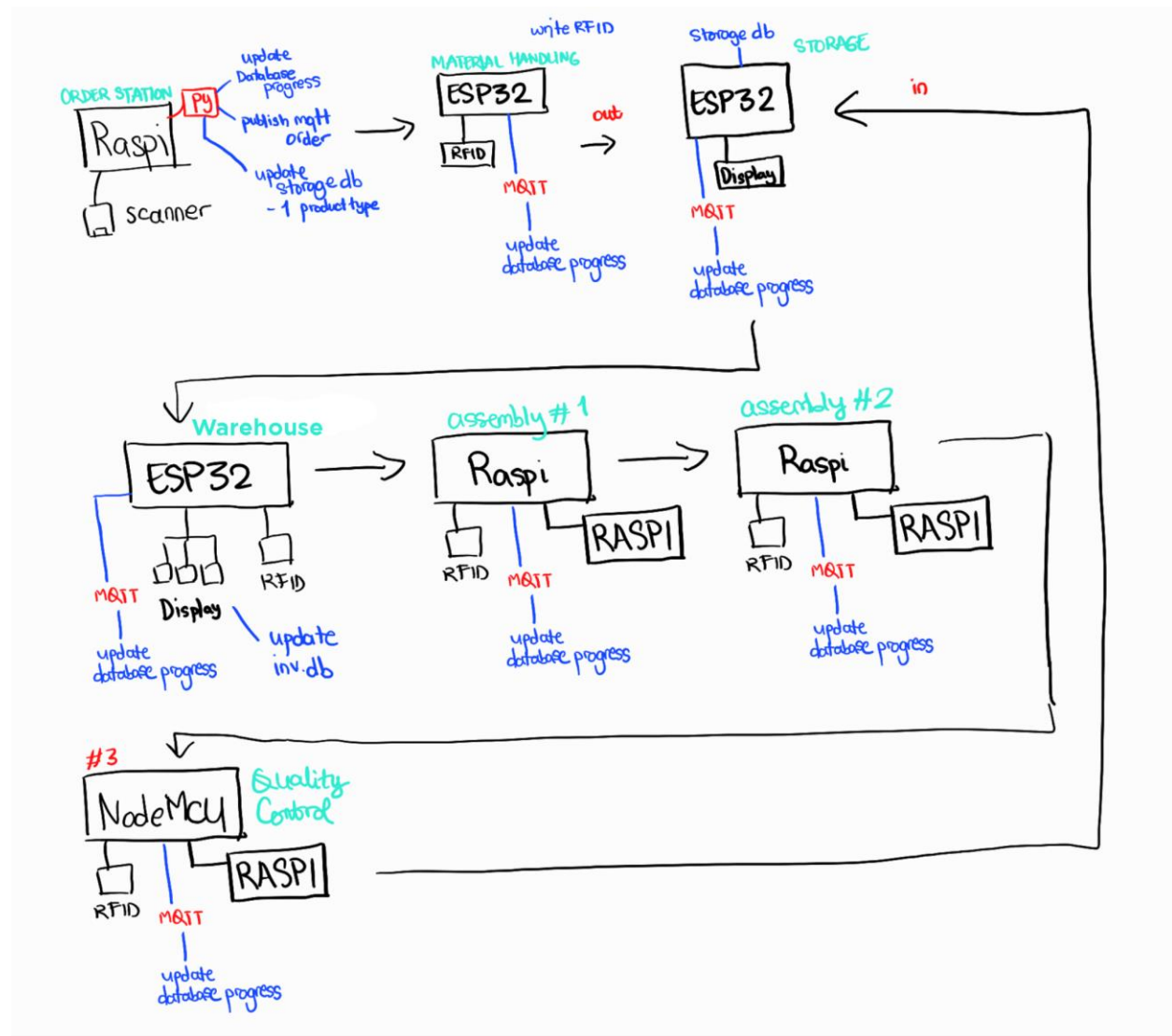
Mendigitalisasi sistem produksi dengan menggunakan *Toyota Production System* dan pembaca RFID MFRC-522 untuk pelacakan logistik dan status progres produksi. Sistem yang dirancang menggunakan *Smart Storage Station* dengan IoT yang berfungsi untuk menyimpan *finished good*. Protokol MQTT dan komunikasi serial akan digunakan untuk pengiriman, penyimpanan, dan visualisasi data.

## Peralatan:

- ESP32 WROOM
- RFID reader MFRC-522
- RFID cards
- Raspberry Pi
- Monitor
- Kabel jumper
- Modul 7-segmen TM1637 display
- Breadboard
- Container
- Barcode and QR scanner GM66
- Kabel USB to micro USB
- Komputer/sumber tegangan

## 2. PERCOBAAN

### 2.1 Penjelasan Sistem



Gambar 1. Visualisasi proses kanban order

Sistem Kanban akan diawali dengan penerimaan order dalam bentuk QR Code. QR Code akan di-scan oleh **Order Station**, lalu merekam order dalam Order Database dan Raspberry Pi order station akan publish MQTT message yang berisi informasi order untuk penulisan kartu RFID pada station selanjutnya. Kartu RFID akan ditulis dengan informasi QR Code dalam **Material Handling Station**, kemudian kartu RFID ini yang akan digunakan seterusnya sebagai bentuk kanban digital dari order. Lalu pada **Storage**

**station**, kartu RFID akan dibaca oleh RFID Scanner Storage Out untuk mengurangi jumlah unit yang disimpan sesuai dengan tipe produk yang tersimpan pada kartu RFID, unit yang diambil dari storage langsung akan diberikan kepada client dan produksi untuk “mengganti” unit tersebut akan dimulai. Jumlah unit akan di-update pada Storage Database pada server. Kemudian kanban order akan diberikan kepada **Warehouse Station** yang akan menampilkan jumlah material yang dibutuhkan sesuai tipe produk yang tersimpan pada kanban menggunakan TM1637, menampilkan sisa material yang tersimpan menggunakan TM1637, dan menyimpan jumlah material pada Warehouse Database melalui pesan MQTT.

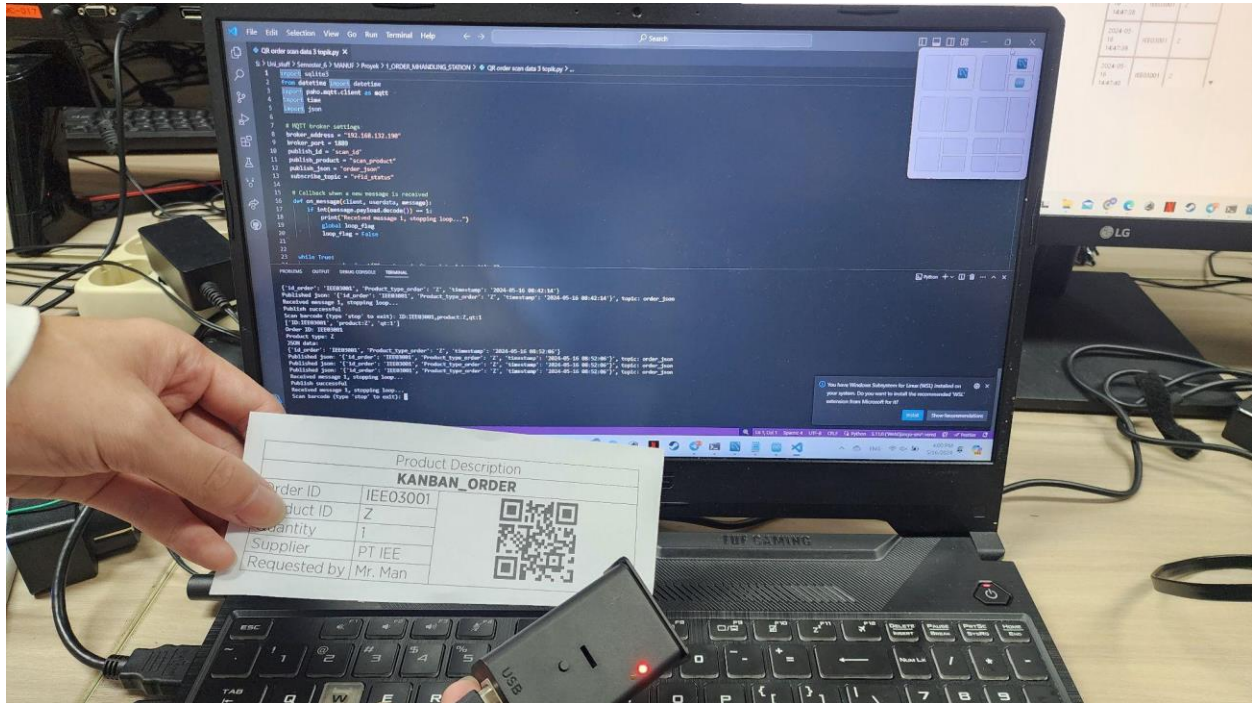
Material handling station akan secara otomatis mengurangi jumlah material yang tersimpan berdasarkan jumlah material yang dibutuhkan oleh tipe produk yang disimpan pada kartu RFID. Lalu RFID kanban akan diteruskan ke **Assembly Station #1**. Station ini akan memiliki RFID Scanner dan saat kartu dibaca, instruksi pembuatan produk tersebut akan ditampilkan pada monitor sesuai dengan informasi produk dalam kartu. Hal yang sama akan dilakukan untuk **Assembly Station #2** dan **Quality Control Station** dengan perbedaan instruksi yang ditampilkan pada station. Terakhir, RFID kanban akan diberikan kepada **Storage station** dan produk akan disimpan dalam storage dengan membaca kartu RFID di RFID Scanner Storage In.

Seluruh track progress produksi, setiap station akan memberi update progress dan akan disimpan pada Database Progress yang berisi ID produk, jenis produk, posisi order berada di station mana, dan persentase progress selesainya.

## 2.2 Sistem Kanban

### 2.2.1 Order Station

#### 2.2.1.1 Foto rangkaian



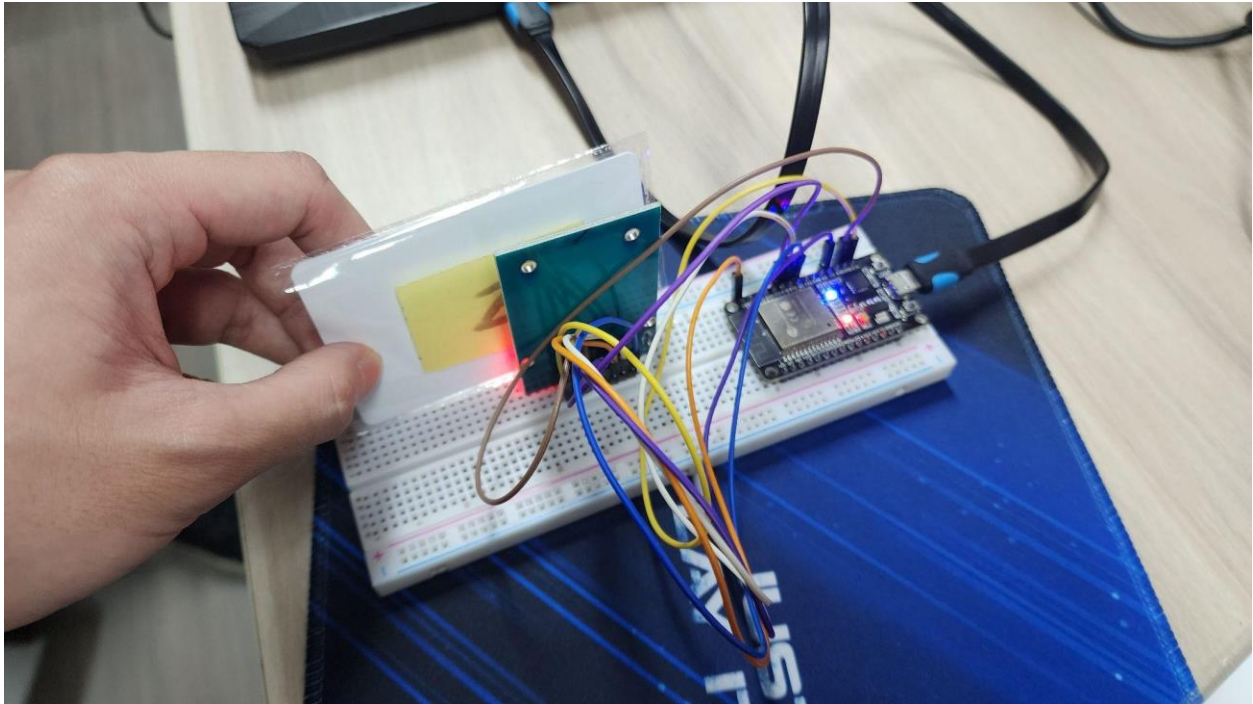
Gambar 2. Order station

#### 2.2.1.2 Penjelasan

Station pertama adalah station order. Station order cukup memiliki barcode/QR scanner GM66 yang tersambung pada laptop atau raspberry pi. GM66 akan secara otomatis membaca QR code pada kanban\_order kemudian akan menuliskan data yang dibacanya pada program python. Program python akan menerima dan mengolah bacaan data dari QR. Kemudian bacaan data tersebut akan dipublish kepada MQTT untuk diterima station selanjutnya.

## 2.2.2 Material Handling Station

### 2.2.2.1 Foto rangkaian



Gambar 3. Rangkaian material handling station

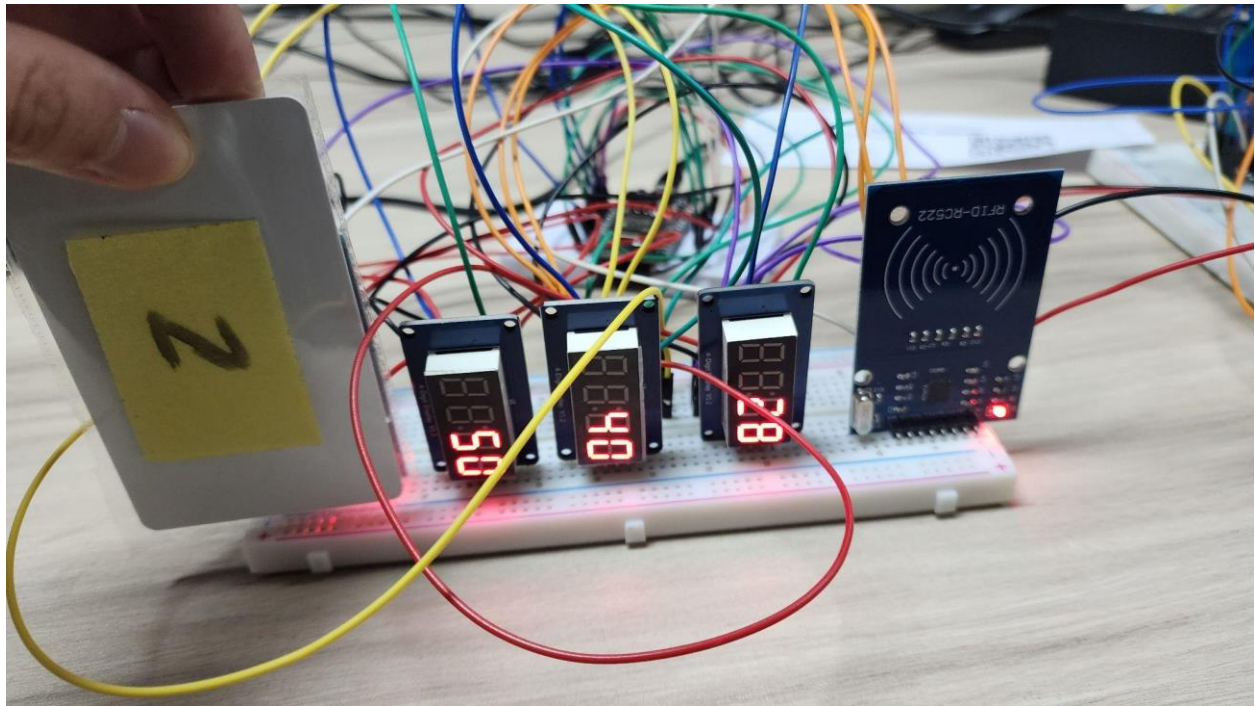
### 2.2.2.2 Penjelasan

Station ini terdiri dari satu ESP32 dan satu RFID scanner. Kartu RFID akan ditulis dengan informasi product type dan order ID. Station mendapat informasi ini dari subscribe topik MQTT yang diterimanya. Station juga akan mengirim MQTT message untuk update progress order pada database. Message akan berisi produk id dan tipe produk sedang berada di station Material Handling dan progress order masih 15%.

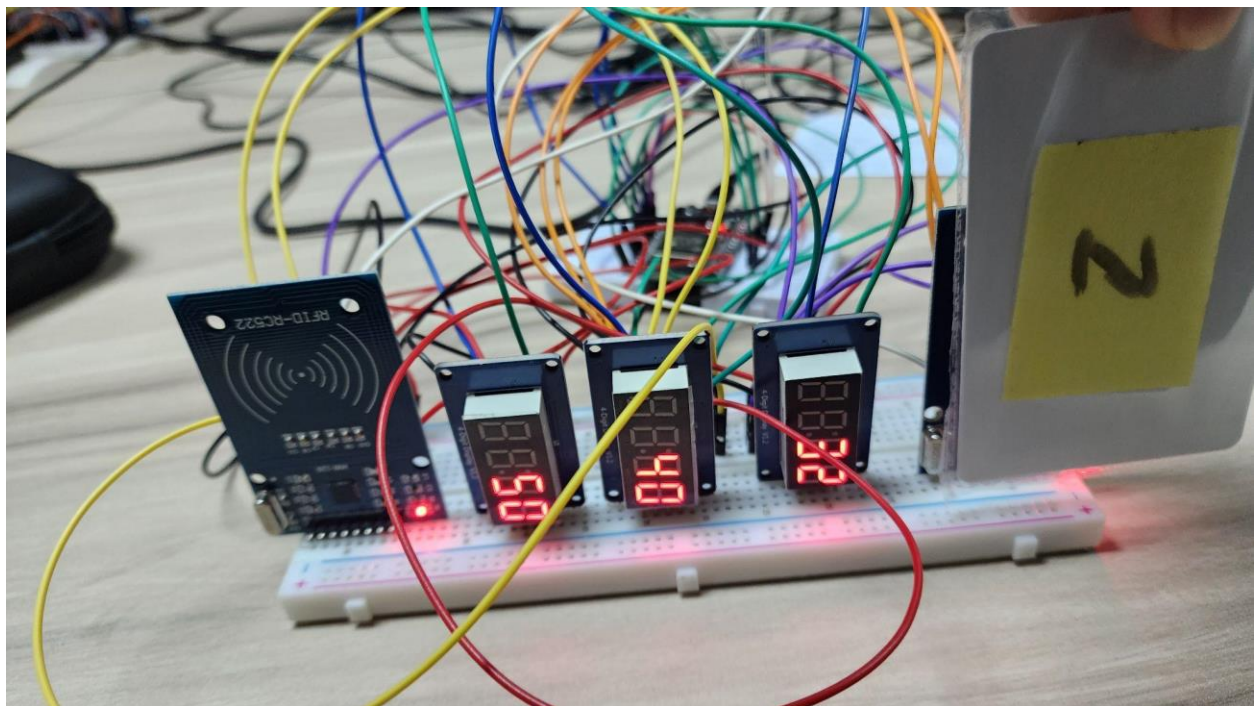


## 2.2.3 Storage Station

### 2.2.3.1 Foto rangkaian



Gambar 4. Rangkaian storage station (out)



Gambar 5. Rangkaian storage station (in)

### **2.2.3.2 Penjelasan**

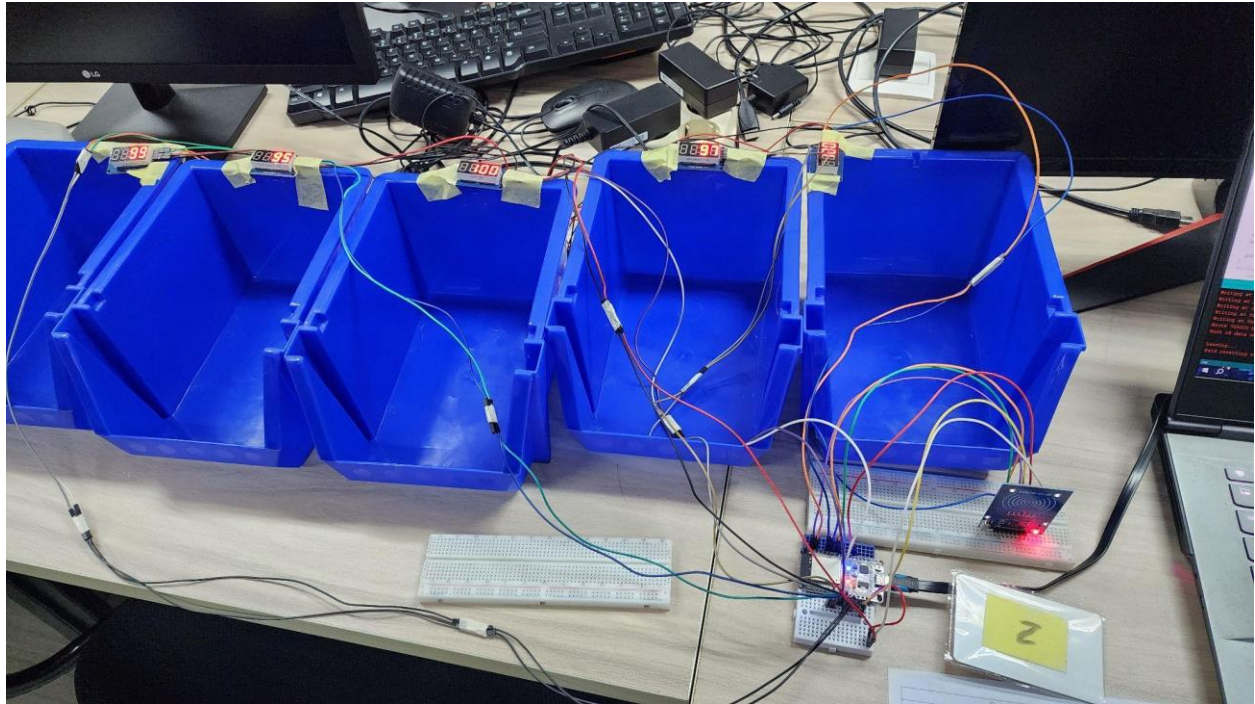
Station storage akan bertugas untuk merekam waktu unit produk masuk dan keluar, serta menghitung dan menampilkan jumlah unit yang sedang disimpan pada display sederhana TM1637. Rangkaian storage station memiliki dua buah MFRC-522 RFID reader, dan tiga buah modul 7-segmen TM1637 display. MFRC-522 RFID reader sebelah kiri pada foto adalah RFID reader untuk storage out, yang berarti ketika ada order masuk, kanban untuk produk yang dipesan akan dibaca oleh RFID reader ini dan jumlah unit akan dikurang. MFRC-522 RFID reader sebelah kanan pada foto adalah RFID reader untuk storage in, sehingga ketika seluruh proses produksi telah selesai petugas akan menempelkan kartu pada RFID reader ini untuk menambahkan jumlah unit yang ada pada storage.

Ketiga modul 7-segmen TM1637 digunakan untuk menampilkan jumlah unit yang disimpan pada storage. Secara berurut dari kiri ke kanan, TM1637 menampilkan jumlah unit produk X, produk Y, dan produk Z. Setiap kali sebuah RFID card terbaca pada RFID reader, station akan membaca tipe produk yang disimpan pada kanban dan akan mengurangi atau menambahkan berdasarkan RFID mana yang menerima bacaan data. Station juga akan publish tipe produk yang dibaca, jumlah unit tersebut yang ada pada storage, dan progress dari order tersebut. Tipe produk yang dibaca dan jumlah unit akan dikirim dalam satu topik yaitu topik “storage” dan data akan dikirim dalam bentuk JSON, di mana “Product\_type” adalah key JSON untuk tipe produk dan “Amount” adalah key JSON untuk jumlah unit tipe produk tersebut. Progress dari order akan dipublish pada topik “progress” yang akan berisi 25% atau 100% sesuai RFID reader in atau out. Server akan subscribe pada topik “progress” dan “storage” dan akan menyimpan dan mengolah data yang diterima.

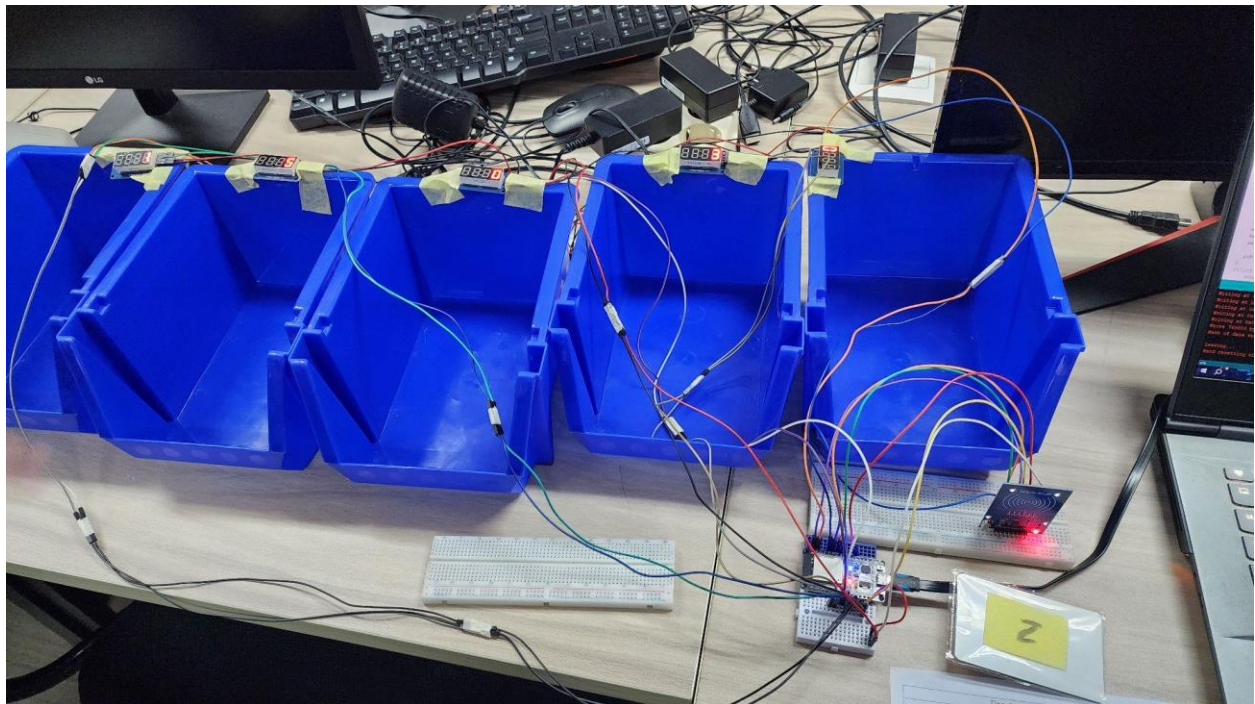


## 2.2.4 Warehouse Station

### 2.2.4.1 Foto rangkaian



Gambar 6. Warehouse ketika menampilkan jumlah material yang tersimpan



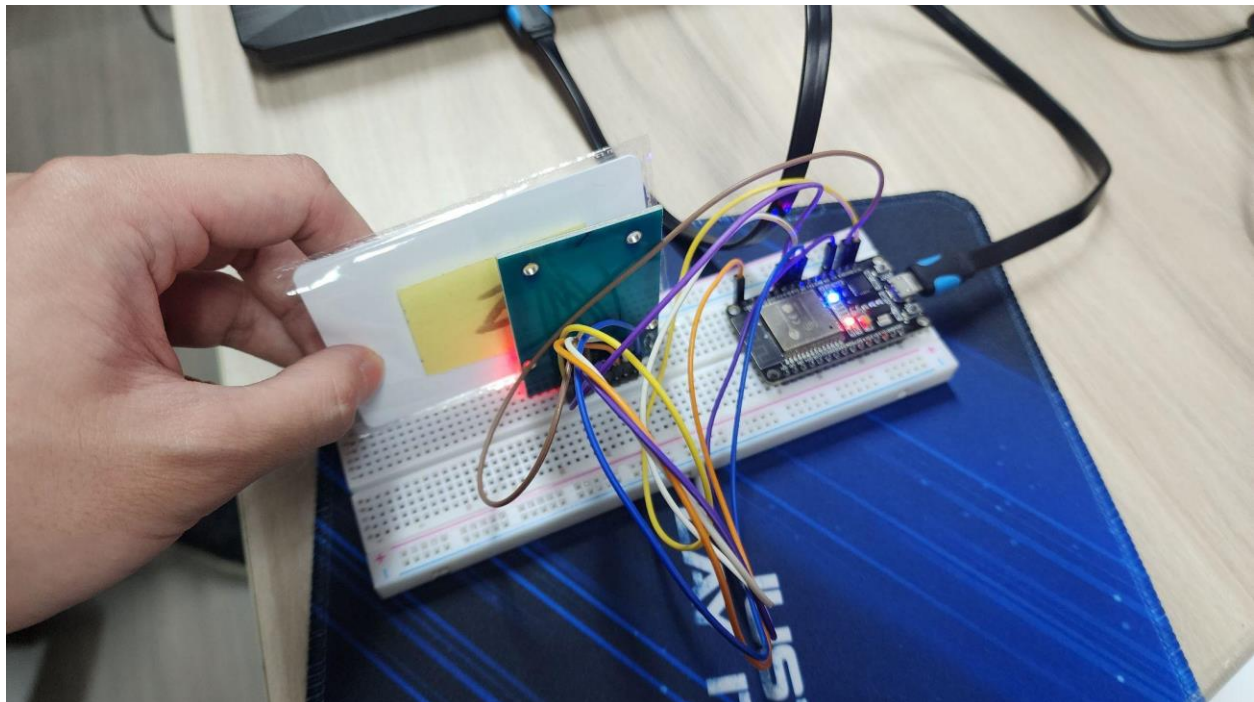
Gambar 7. Warehouse ketika menampilkan jumlah material yang dibutuhkan untuk produk yang dibaca

#### 2.2.4.2 Penjelasan

Warehouse station akan menggunakan 5 buah container, 5 buah modul 7-segmen TM1637 display, sebuah ESP32, dan sebuah RFID reader. Setiap modul segmen dan container akan menampilkan dan menampung material yang dimiliki. Kemudian ketika RFID reader membaca bacaan data, maka TM1637 akan menampilkan jumlah material yang dibutuhkan dari produk yang dibaca dari kartu RFID. TM1637 juga akan menampilkan sisa material yang dimiliki secara bergantian dengan jumlah material yang dibutuhkan. Kemudian station akan publish jumlah material yang tersisa, produk apa yang terbaca, dan progress dari warehouse station yang merupakan 35%. Dalam contoh foto diatas, kanban RFID yang dibaca oleh RFID reader adalah produk Z kemudian foto pertama menampilkan sisa material yang tersimpan setelah produk Z dibaca, dan foto kedua menampilkan jumlah material yang dibutuhkan untuk membuat produk Z.

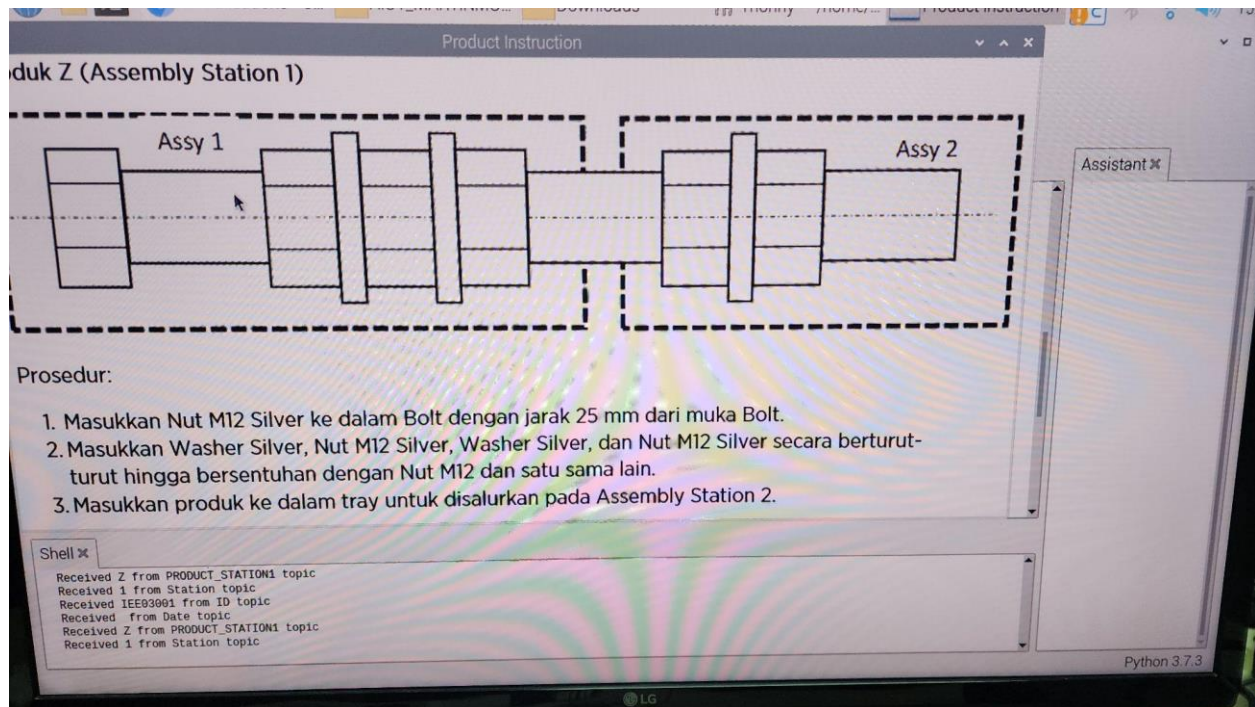
#### 2.2.5 Assembly Station #1

##### 2.2.5.1 Foto rangkaian



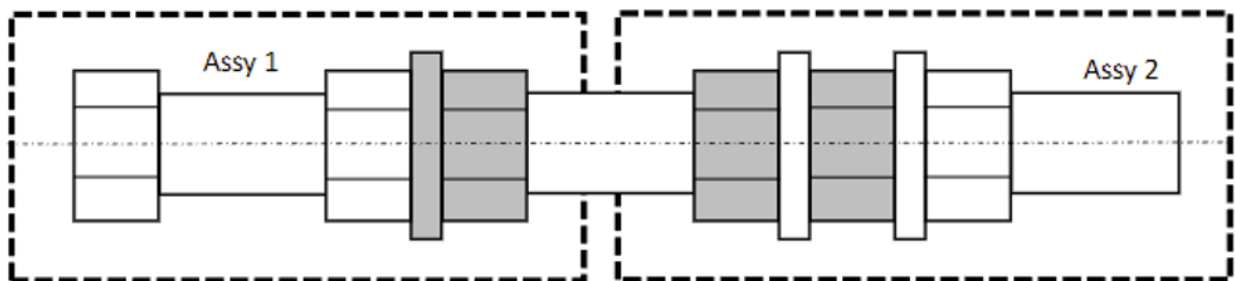
Gambar 8. Rangkaian assembly station 1





Gambar 9. Tampilan prosedur assembly produk Z di assembly station 1

#### Produk Y (Assembly Station 1)

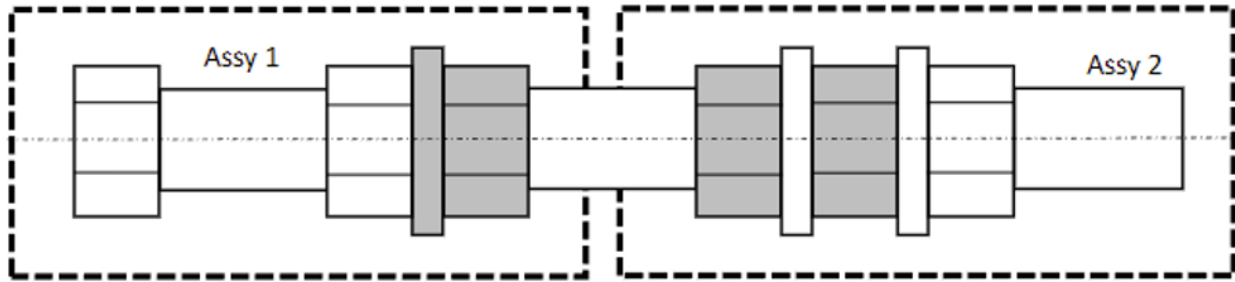


#### Prosedur:

1. Masukkan Nut M12 Hitam ke dalam Bolt dengan jarak 25 mm dari muka Bolt.
2. Masukkan Washer Hitam, Nut M12 Hitam, Washer Hitam, dan Nut M12 Hitam secara berturut-turut hingga bersentuhan dengan Nut M12 Hitam dan satu sama lain.
3. Masukkan produk ke dalam tray untuk disalurkan pada Assembly Station 2.

Gambar 10. Tampilan prosedur assembly produk Y di assembly station 1

### Produk X (Assembly Station 1)



Prosedur:

1. Masukkan Nut M12 Silver ke dalam Bolt dengan jarak 25 mm dari muka Bolt.
2. Masukkan Washer Hitam dan Nut M12 Hitam secara berturut-turut hingga bersentuhan dengan Nut M12 Silver dan satu sama lain.
3. Masukkan produk ke dalam tray untuk disalurkan pada Assembly Station 2.

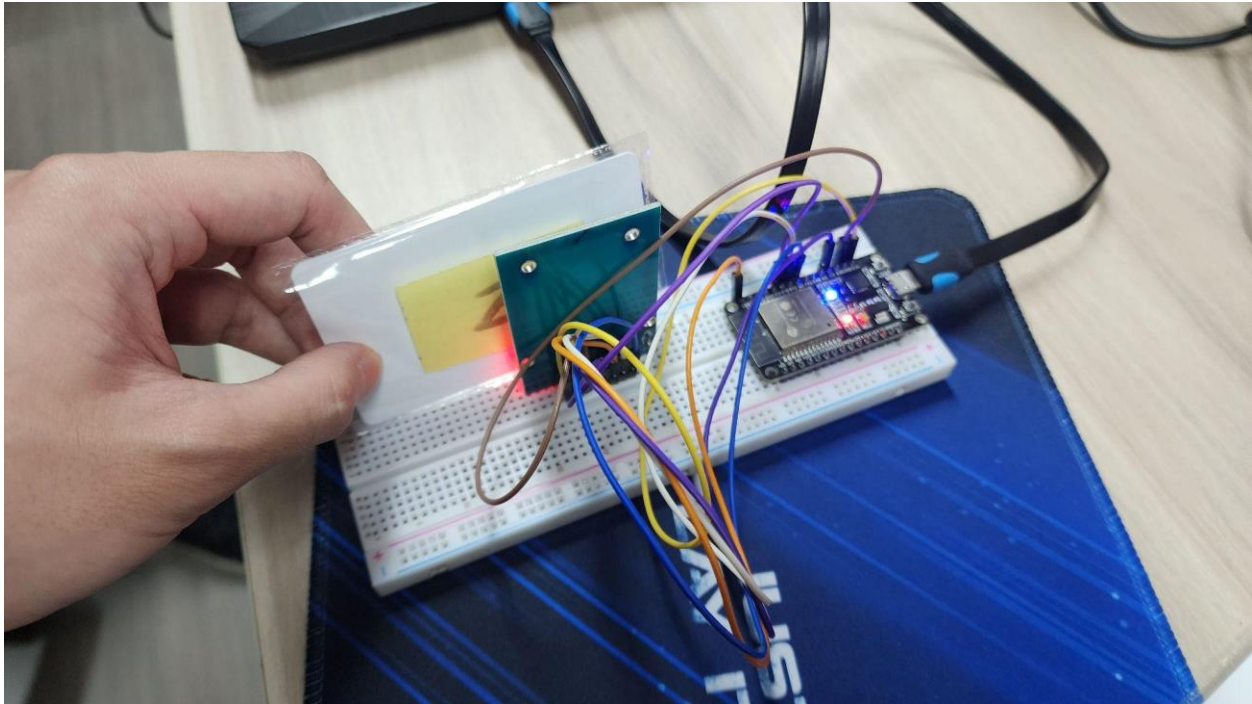
Gambar 11. Tampilan prosedur assembly produk X di assembly station 1

#### 2.2.5.2 Penjelasan

Assembly station 1 bertugas untuk memberikan informasi pada pekerja bagaimana cara memasang produk. Informasi jenis produk yang akan diperiksa tergantung dari tipe produk yang tersimpan pada kartu RFID yang dibaca oleh RFID reader MFRC522 pada ESP32. Kemudian, ESP32 akan publish semua informasi dari kartu RFID ke broker MQTT di laptop dengan topik 'PRODUCT\_STATION1'. Setelah itu, program Python dari Raspberry Pi akan melakukan subscribe topik 'PRODUCT\_STATION1' pada broker MQTT untuk mengambil informasi mengenai jenis produk dan lokasi station produk yang baru saja dibaca. Dengan ketentuan dua variabel tersebut, akan ditentukan visualisasi prosedur pengerjaan produk apa yang akan ditampilkan pada layar monitor RPi. Penampilan visualisasi pengecekan kesesuaian produk adalah file PNG yang ditampilkan menggunakan Tkinter. Gambar visualisasi berupa gambar finished good dengan tambahan angka pengukuran pada gambar. ESP32 mengirimkan MQTT message ke node-red yang akan menambahkan data dalam progress database. Message yang dikirim adalah id dan tipe produk, progress order sedang berada di station apa dan persentase progress order.

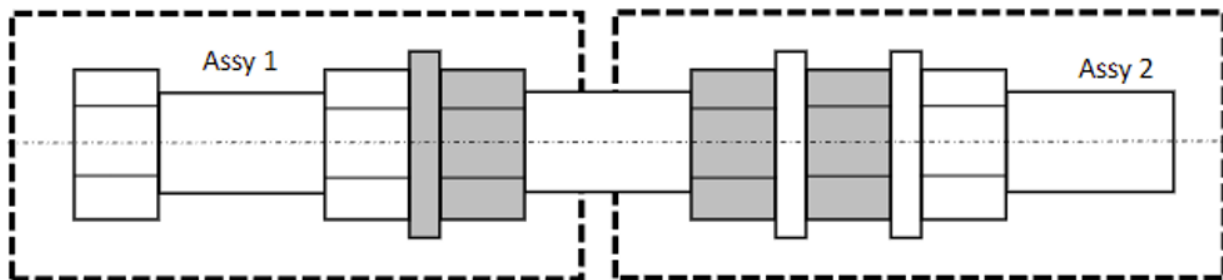
## 2.2.6 Assembly Station #2

### 2.2.6.1 Foto rangkaian



Gambar 12. Rangkaian assembly station 2

### Produk X (Assembly Station 2)

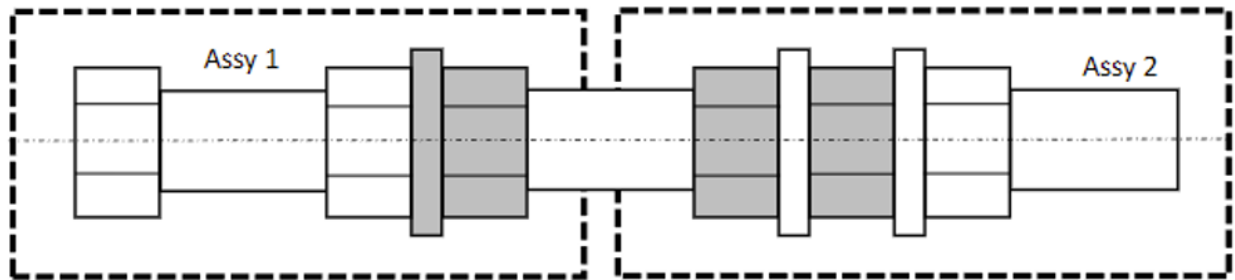


#### Prosedur:

1. Masukkan Nut M12 hitam ke dalam Bolt dengan jarak 25 mm dari Nut M12 Hitam.
2. Masukkan Washer Silver, nut M12 Hitam, Washer Silver, dan nut M12 Silver secara berturut-turut hingga bersentuhan dengan Nut M12 Hitam dan satu sama lain.
3. Masukkan produk ke dalam tray untuk disalurkan pada Quality Control Station.

Gambar 13. Tampilan prosedur assembly produk X di assembly station 2

### Produk Y (Assembly Station 2)

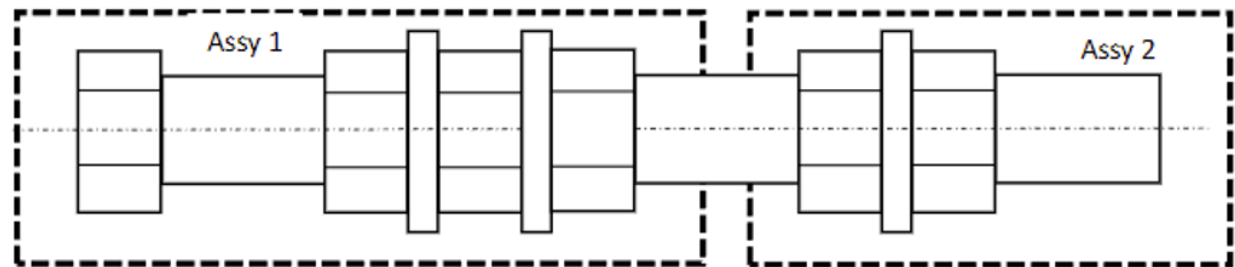


Prosedur:

1. Masukkan Nut M12 Hitam ke dalam Bolt dengan jarak 25 mm dari Nut M12 Hitam.
2. Masukkan Washer Silver, Nut M12 Hitam, Washer Silver, dan Nut M12 Silver secara berturut-turut hingga bersentuhan dengan Nut M12 Hitam dan satu sama lain.
3. Masukkan produk ke dalam tray untuk disalurkan pada Quality Control Station.

Gambar 14. Tampilan prosedur assembly produk Y di assembly station 2

### Produk Z (Assembly Station 2)



Prosedur:

1. Masukkan Nut M12 Silver ke dalam Bolt dengan jarak 25 mm dari Nut M12 Silver.
2. Masukkan Washer Silver dan Nut M12 Silver secara berturut-turut hingga bersentuhan dengan Nut M12 Silver dan satu sama lain.
3. Masukkan Produk ke dalam tray untuk disalurkan pada Quality Control Station.

Gambar 15. Tampilan prosedur assembly produk Z di assembly station 2

## 2.2.6.2 Penjelasan

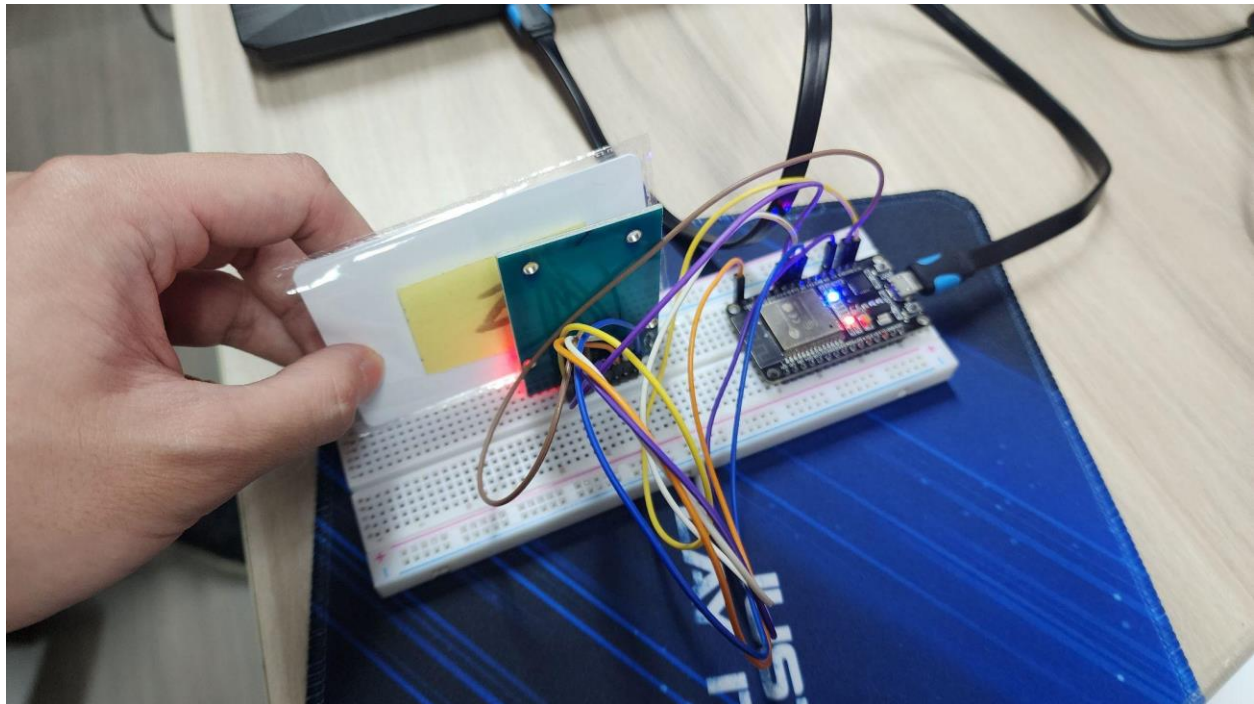
Assembly station 2 bertugas untuk memberikan informasi pada pekerja bagaimana cara memasang produk. Informasi jenis produk yang akan diperiksa tergantung dari tipe produk yang tersimpan pada kartu RFID yang dibaca oleh RFID reader MFRC522 pada ESP32. Kemudian, ESP32 akan publish semua informasi dari kartu RFID ke broker MQTT di laptop dengan topik 'PRODUCT\_STATION2'. Setelah itu, program Python dari



Raspberry Pi akan melakukan subscribe topik 'PRODUCT\_STATION2' pada broker MQTT untuk mengambil informasi mengenai jenis produk dan lokasi station produk yang baru saja dibaca. Dengan ketentuan dua variabel tersebut, akan ditentukan visualisasi prosedur pengerjaan produk apa yang akan ditampilkan pada layar monitor RPi. Penampilan visualisasi pengecekan kesesuaian produk adalah file PNG yang ditampilkan menggunakan Tkinter. Gambar visualisasi berupa gambar finished good dengan tambahan angka pengukuran pada gambar. ESP32 mengirimkan MQTT message ke node-red yang akan menambahkan data dalam progress database. Message yang dikirim adalah id dan tipe produk, progress order sedang berada di station apa dan persentase progress order.

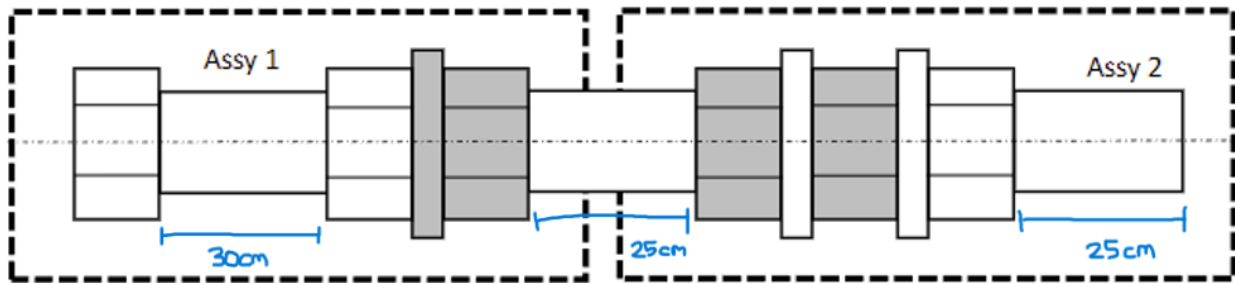
## **2.2.7 Quality Control Station**

### **2.2.7.1 Foto rangkaian**



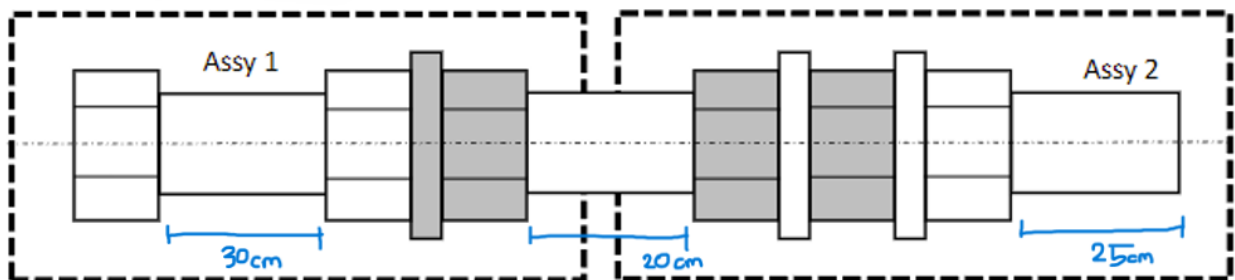
Gambar 16. Rangkaian quality control station

### Produk X



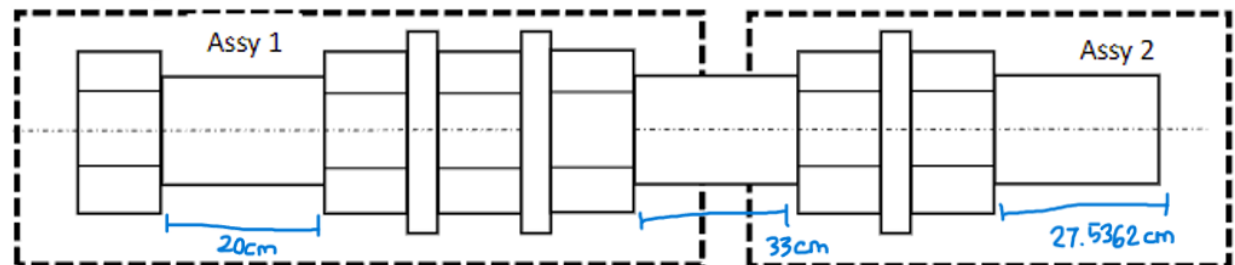
Gambar 17. Tampilan produk X di quality control station

### Produk Y



Gambar 18. Tampilan produk Y di quality control station

### Produk Z



Gambar 19. Tampilan produk Z di quality control station

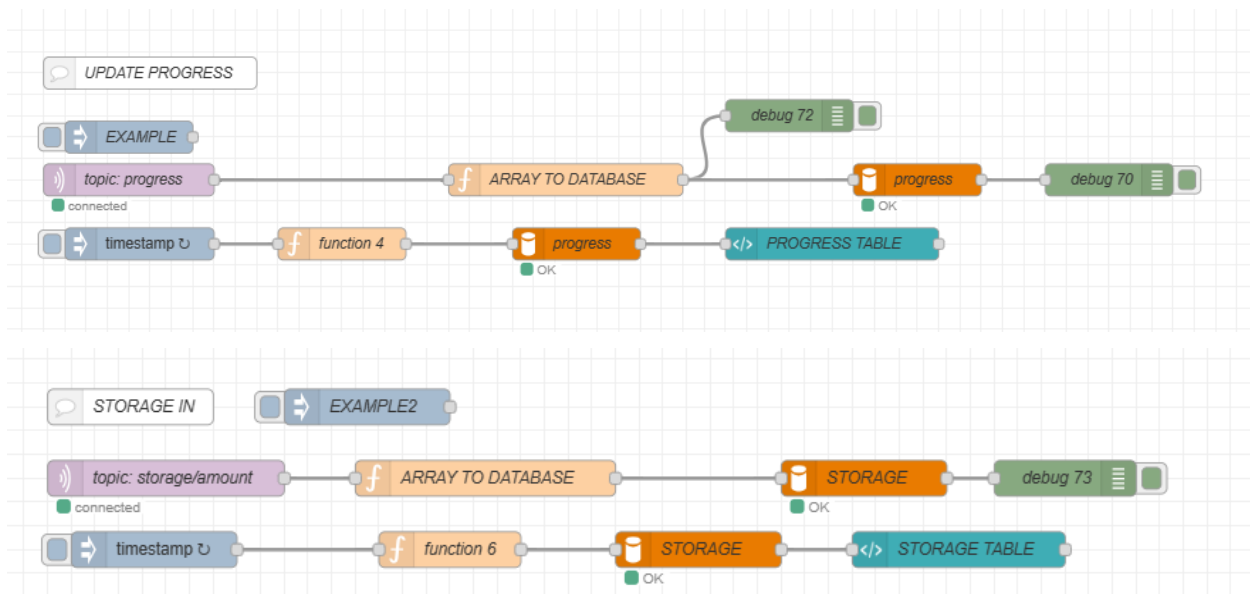
## 2.2.7.2 Penjelasan

Quality control adalah station yang bertugas untuk memeriksa kesesuaian pemasangan produk pada station-station assembly sebelumnya dan apakah hasil akhir produk sesuai dengan standar produk yang telah ditentukan. Informasi jenis produk yang akan diperiksa tergantung dari tipe produk yang tersimpan pada kartu RFID yang dibaca oleh RFID reader MFRC522 pada ESP32. Kemudian, ESP32 akan publish semua

informasi dari kartu RFID ke broker MQTT di laptop dengan topik 'QC'. Setelah itu, program Python dari Raspberry Pi akan melakukan subscribe topik 'QC' pada broker MQTT untuk mengambil informasi mengenai jenis produk dan lokasi station produk yang baru saja dibaca. Dengan ketentuan dua variabel tersebut, akan ditentukan visualisasi prosedur pengerjaan produk apa yang akan ditampilkan pada layar monitor RPi. Penampilan visualisasi pengecekan kesesuaian produk adalah file PNG yang ditampilkan menggunakan Tkinter. Gambar visualisasi berupa gambar finished good dengan tambahan angka pengukuran pada gambar. ESP32 mengirimkan MQTT message ke node-red yang akan menambahkan data dalam progress database. Message yang dikirim adalah id dan tipe produk, progress order sedang berada di station apa dan persentase progress order.

## 2.3 Node-RED

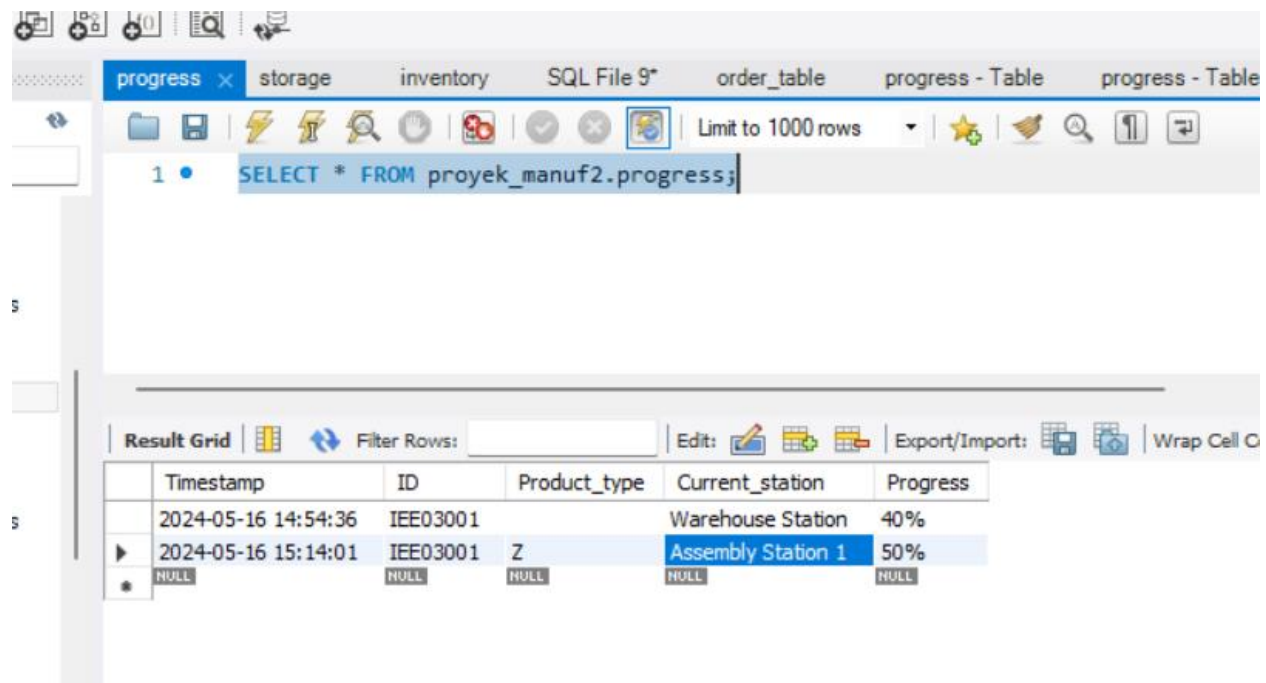
### 2.3.1 Flow





Flow Node-RED yang telah dibuat akan menampilkan UI dashboard seperti pada gambar di atas, dengan tabel-tabel yang menunjukkan jumlah part produk, jumlah barang di storage, log order, dan log progres.

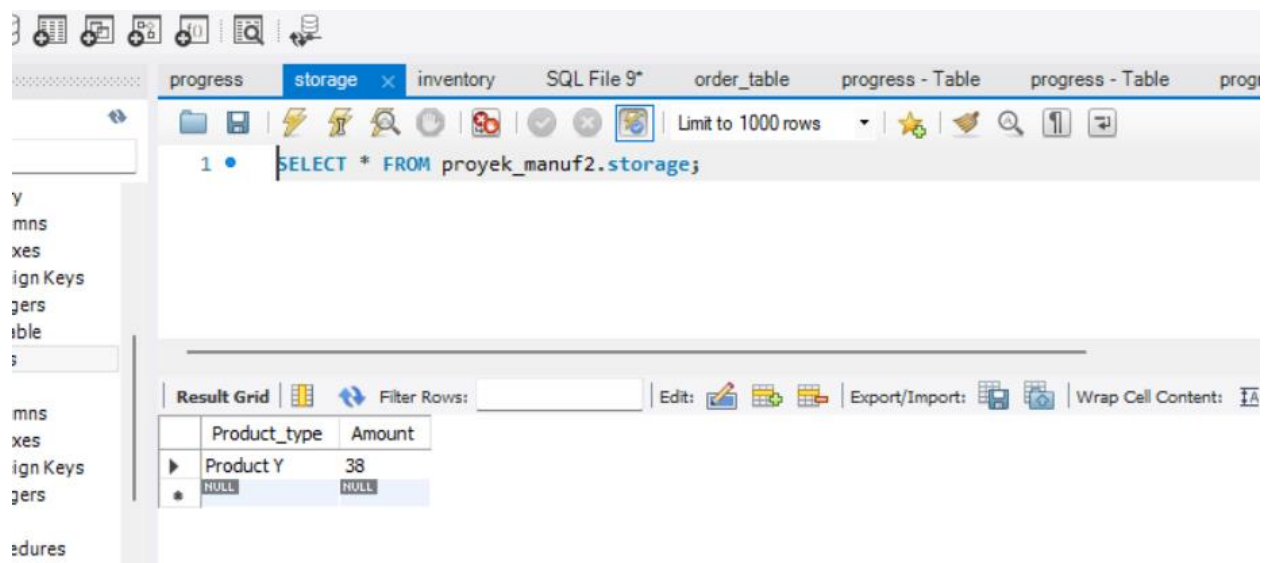
## 2.4 Database



The screenshot shows the Node-RED SQL Editor interface. The 'progress' tab is selected. The SQL query is `SELECT * FROM proyek_manuf2.progress;`. The result grid displays the following data:

Timestamp	ID	Product_type	Current_station	Progress
2024-05-16 14:54:36	IEE03001		Warehouse Station	40%
2024-05-16 15:14:01	IEE03001	Z	Assembly Station 1	50%
NULL	NULL	NULL	NULL	NULL

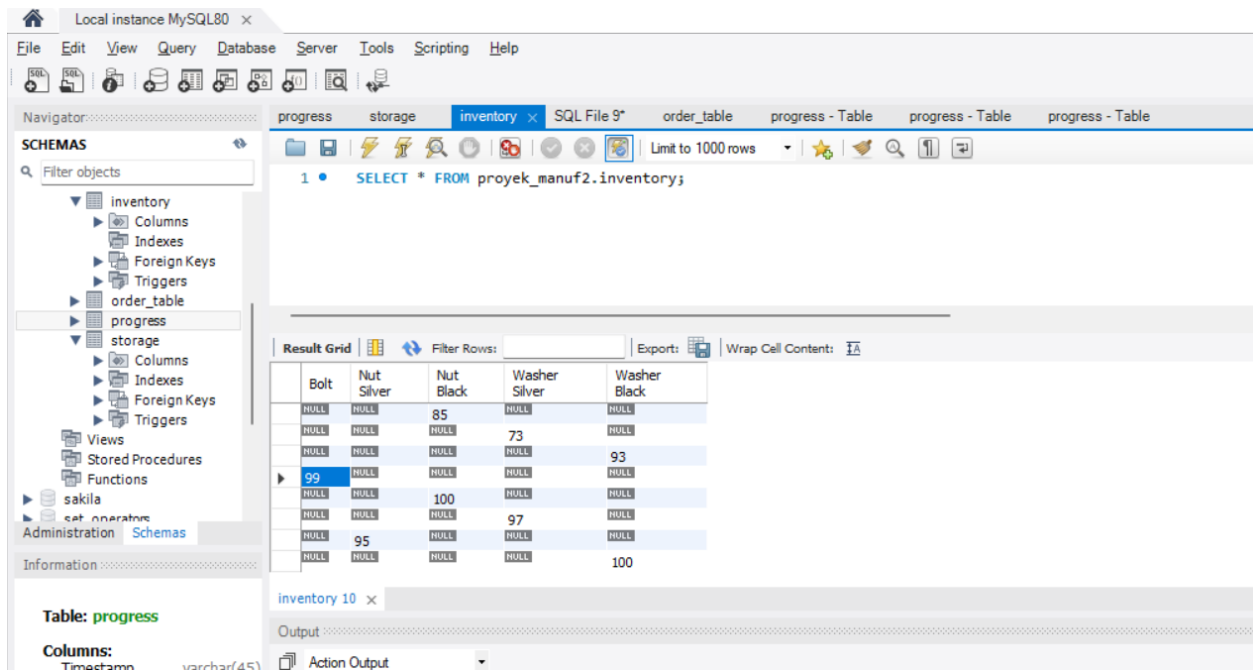
Gambar 22. Menampilkan semua isi tabel progress



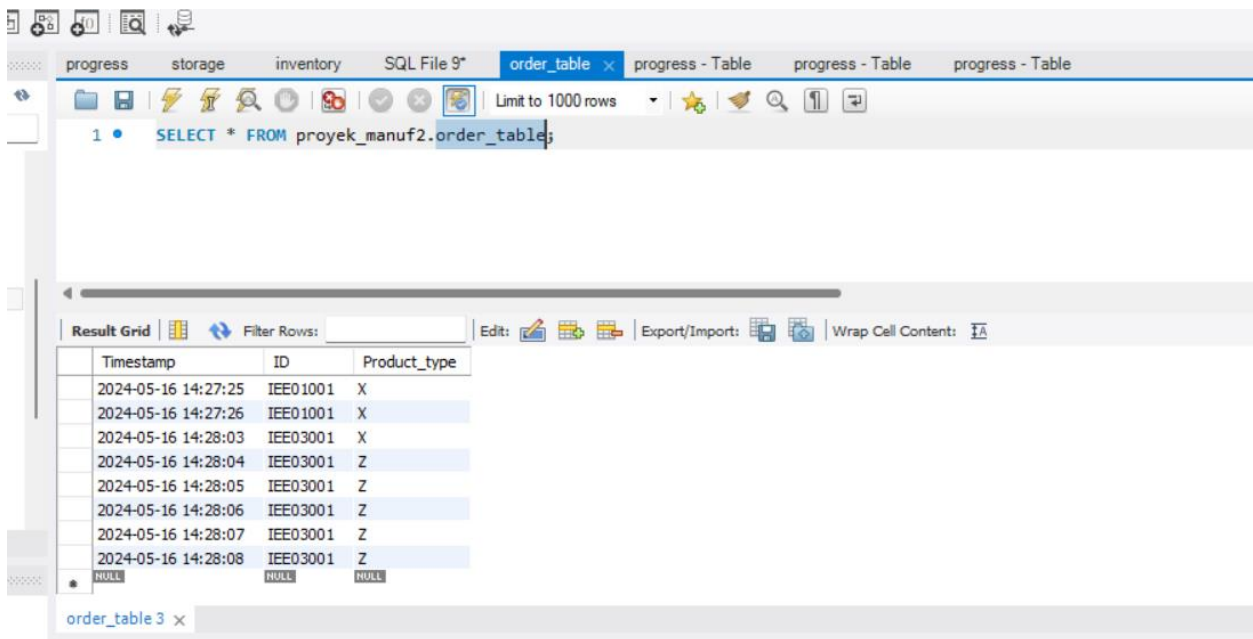
The screenshot shows the Node-RED SQL Editor interface. The 'storage' tab is selected. The SQL query is `SELECT * FROM proyek_manuf2.storage;`. The result grid displays the following data:

Product_type	Amount
Product Y	38
NULL	NULL

Gambar 23. Menampilkan semua isi tabel storage



Gambar 24. Menampilkan semua isi tabel inventory



Gambar 25. Menampilkan semua isi tabel order\_table

Database MySQL di atas adalah bentuk database yang digunakan untuk menyimpan seluruh order, progress, jumlah unit, jumlah material, dan lain-lain.

### 3. ANALISIS PERCOBAAN

#### 3.1 Kendala Selama Proyek

Beberapa kendala yang kami alami saat membuat proyek termasuk:

1. Menggunakan banyak topik MQTT publish dan subscribe dan menentukan tipe data yang dapat dikirim lewat mikrokontroler merupakan hal yang tidak mudah. Perlu banyak *trial and error* dan juga eksplorasi mandiri untuk dapat membuat kode Arduino dan Python sistem kami. Pada akhirnya kami bersepakat untuk menggunakan tipe data JSON untuk pengiriman data. Sering juga library Arduino mengalami error sehingga kami harus menyelesaikan masalah itu dengan cara re-install library atau mengganti dari papan pengembang dari NodeMCU ke ESP32 atau sebaliknya.
2. Meski kode setiap station telah ada, mengintegrasikan semua kode dan rangkaian station agar dapat divisualisasikan dengan Node-RED dan menyimpan data pada database tidak mudah. Message MQTT harus diubah ke tipe yang dapat diproses oleh Node-RED dengan mudah. Menghubungkan MySQL dengan Node-RED juga mengalami banyak kendala dan kesulitan. Sering juga kami mendapati kendala PARSE ERROR yaitu data itu tidak sesuai dengan format atau tipe yang dibutuhkan oleh function Node-RED untuk disimpan ke database. Salah satu kendala yang sangat mengganggu dan sulit kami ketahui penyebabnya adalah Order Database yang error pada function Node-RED tanpa penyebab yang jelas. Ternyata setelah debugging kata “order” adalah kata yang telah disimpan untuk fungsi lain oleh SQL. Maka dari itu menggunakan “order” sebagai nama tabel tidak dapat digunakan untuk nama tabel.
3. Kami harus belajar dan eksplorasi mandiri mengenai beberapa hal, seperti cara merangkai serta memprogram dua RFID reader sekaligus dan merangkai serta memprogram TM1637 display. Kami juga harus mengingat kembali pembelajaran mata kuliah basis data untuk membuat dan update table di MySQL. Ini memakan cukup banyak waktu dan seringkali mengalami kendala juga.
4. Lab IEE hanya dibuka dari pukul 9 pagi hingga pukul 5 sore. Waktu buka lab ini juga bertabrakan dengan jam kelas. Ini menyebabkan kami kekurangan waktu

untuk menyempurnakan proyek dan latihan demonstrasi walaupun kode dan rangkaian dapat berjalan dengan baik. Jadwal IEE yang sangat sibuk dengan proyek mata kuliah lain, tingkat kesulitan proyek Manufaktur II yang cukup kompleks, serta waktu buka lab IEE yang tidak sesuai dengan waktu luang kami membuat progress proyek Manufaktur II cukup lama.

Beberapa kendala yang kami alami saat mendemonstrasikan proyek termasuk:

1. WiFi kampus tidak mendukung. MQTT connection selalu terputus dan ini membuat sistem tidak dapat berjalan dengan lancar. Akhirnya kami menggunakan hotspot pribadi yang jauh lebih stabil dibandingkan jaringan internet yang disediakan oleh kampus.
2. Terdapat beberapa hardware lab yang rusak seperti breadboard, RFID Scanner dan kabel jumper. Ini sangat menghambat progres karena kami seringkali melakukan debugging tetapi akhirnya menyadari bahwa ini adalah masalah hardware.
3. Ternyata Kode Arduino tidak dapat berjalan dengan baik jika ada file lain yang berada di dalam folder yang sama, meski tidak ada hubungan apapun dengan kode Arduino tersebut. Ini seringkali membuat kode arduino error pada saat demonstrasi.



## 3.2 Pemborosan sistem

Ada beberapa pemborosan sistem yang kami amati dalam sistem ini yaitu:

1. Tidak semua station kanban memerlukan mikrokontroler yang dapat menyediakan processing power yang kuat dan tegangan yang tinggi seperti ESP32. Jika seluruh sistem menggunakan mikrokontroler yang kuat, akan terjadi pemborosan listrik dan juga pemborosan biaya untuk membeli mikrokontroler. Saran kami adalah untuk menggunakan spesifikasi mikrokontroler yang secukupnya dan cenderung murah untuk station yang tidak melakukan proses berat dan tidak terhubung dengan elektronik yang memerlukan tegangan tinggi untuk berfungsi.
2. Biaya untuk maintenance sistem cukup tinggi karena akan membutuhkan komponen elektronik yang diinstal pada setiap station industri. Kestabilan sistem juga menjadi satu masalah yang harus diperhatikan dan ditanggulangi jika ingin memastikan keberhasilan dan kelancaran produksi unit. Saran kami untuk menanggulangi semua masalah ini adalah untuk memiliki perencanaan yang jelas dan terstruktur untuk implementasi sistem digitalisasi kanban. Komponen dan alat yang digunakan juga diusahakan berkualitas tinggi dan konsisten sehingga produksi dapat berjalan tanpa error dan maintenance berkala yang terlalu sering. Jaringan internet dalam industri harus cepat dan stabil di setiap sisi pabrik untuk memastikan kestabilan sistem IoT.

## 4. Kesimpulan

Sistem Kanban digital yang dirancang telah memanfaatkan teknologi modern dengan sangat efektif untuk mencapai efisiensi dan akurasi. Penggunaan teknologi seperti MQTT, RFID, dan QR Code, yang merupakan beberapa teknologi paling canggih dan handal saat ini memungkinkan otomatisasi proses dan sinkronisasi data antar stasiun dalam sistem.

Namun, seperti halnya dengan setiap sistem teknologi, ada beberapa pertimbangan penting yang harus diperhatikan. Salah satunya adalah biaya implementasi dan pemeliharaan sistem. Meskipun teknologi seperti MQTT, RFID, dan QR Code dapat memberikan efisiensi dan akurasi, mereka juga bisa menjadi mahal untuk diimplementasikan dan dipelihara. Oleh karena itu, penting untuk melakukan analisis biaya secara menyeluruh sebelum memutuskan untuk mengimplementasikan teknologi ini.

Akhirnya, penting untuk selalu mempertimbangkan bagaimana sistem ini akan berkembang seiring waktu. Teknologi terus berkembang, dan apa yang efektif dan efisien hari ini mungkin tidak lagi demikian di masa depan. Oleh karena itu, penting untuk merencanakan bagaimana sistem produksi dapat beradaptasi dan berkembang seiring dengan perkembangan teknologi. Secara keseluruhan, sistem Kanban digital kami dirancang dengan baik dan memanfaatkan teknologi modern dengan efektif. Dengan mempertimbangkan faktor-faktor di atas, kami dapat memastikan bahwa sistem kami tidak hanya efisien dan akurat, tetapi juga berkelanjutan dan siap untuk masa depan.

## 5. Kontribusi Anggota

Nama	Kontribusi
Martin	<ul style="list-style-type: none"><li>• Membuat kode Arduino untuk progress database dalam semua stasiun sistem</li><li>• Membuat Database untuk seluruh sistem dan mengatur koneksi MQTT untuk semua komponen elektronik juga</li><li>• Membuat semua flow node-red untuk sistem yang termasuk: menerima MQTT dari station, menambahkan data dari MQTT ke database dan visualisasi data dari MySQL ke dashboard. Juga melakukan debugging</li><li>• Membuat melakukan debugging kode arduino untuk warehouse station hingga berjalan</li><li>• Membuat melakukan debugging kode arduino untuk material handling station hingga berjalan</li><li>• Merangkai semua rangkaian assembly station dan material handling</li><li>• Membuat melakukan debugging kode arduino untuk Assembly #1, Assembly #2 dan Quality Control station hingga berjalan</li><li>• Berkontribusi dalam membuat laporan proyek</li><li>• Menyumbang hotspot untuk demonstrasi</li></ul>
Wesley	<ul style="list-style-type: none"><li>• Membuat kode Arduino untuk storage station</li><li>• Konfigurasi MQTT storage station</li><li>• Membuat kode Arduino untuk warehouse station</li><li>• Membenarkan rangkaian warehouse hingga berjalan</li><li>• Membuat kode python untuk order station</li><li>• Konfigurasi MQTT order station</li><li>• Membuat rangkaian storage station</li></ul>

	<ul style="list-style-type: none"> <li>• Berkontribusi dalam pembuatan laporan proyek</li> <li>• Berkontribusi dalam pembuatan kode Assembly Station 1, Assembly Station 2, dan Quality control</li> </ul>
Moses	<ul style="list-style-type: none"> <li>• Sedikit membuat dan merapikan rangkaian</li> <li>• Sumbangsih ide pengiriman data string dengan format JSON agar tidak perlu terlalu banyak topik MQTT</li> <li>• Berkontribusi dalam membuat laporan proyek</li> <li>• Debugging assembly station</li> <li>• Visualisasi instruksi prosedur gambar png</li> <li>• Membuat sedikit bagian dari kode-kode Python dan Arduino pada assembly station 1 dan assembly station 2</li> <li>• Memberikan ide menggunakan Tkinter untuk visualisasi prosedur assembly</li> </ul>

Komitmen Integritas:

“Di hadapan TUHAN yang hidup, saya menegaskan bahwa saya tidak memberikan maupun menerima bantuan apa pun—baik lisan, tulisan, maupun elektronik—di dalam ujian ini selain daripada apa yang telah diizinkan oleh pengajar, dan tidak akan menyebarkan baik soal maupun jawaban ujian kepada pihak lain.”