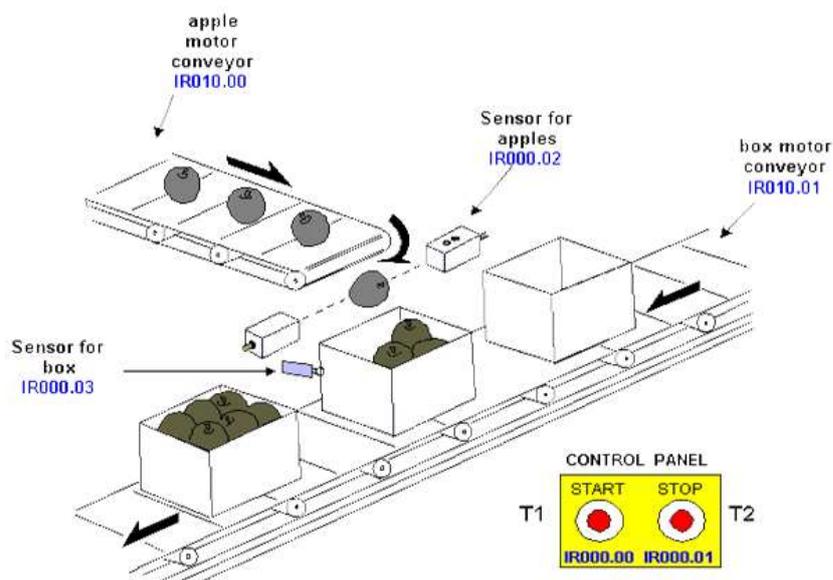


## BAB II

### LANDASAN TEORI

#### 2.1. Proses Pengepakan

Proses pengepakan merupakan salah satu bagian terpenting dalam proses produksi di industri untuk membuat produk aman dan dalam kondisi baik sebelum didistribusikan ke konsumen (Fathahillah *et al.*, 2020). Proses pengepakan dapat dilakukan secara manual atau otomatis. Proses pengepakan lebih efektif dilakukan secara otomatis karena dapat mempercepat proses pengepakan serta dapat meminimalkan kesalahan pada proses pengepakan seperti yang diilustrasikan pada Gambar 2.1.

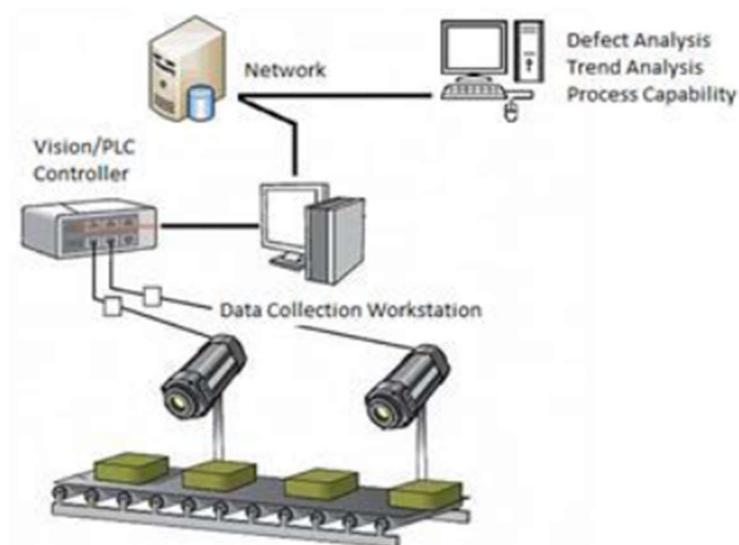


Gambar 2. 1 Ilustrasi Proses Pengepakan

#### 2.2. Machine Vision

*Machine vision* merupakan teknologi dan teknik/metode yang digunakan untuk melakukan inspeksi dan analisis secara otomatis berbasis pengolahan citra,

kontrol proses dan pemanduan robot. *Machine vision* umumnya dikembangkan di dunia industri (Alam and Figana, 2021). *Machine vision* merupakan teknologi yang menggunakan teknik pengambilan citra digital secara otomatis, kemudian citra tersebut diproses dan dianalisa datanya sebagai acuan dalam pengambilan keputusan suatu sistem (Rahmadian and Widyartono, 2020). Pada umumnya, *machine vision* terdiri dari *image acquisition*, *pre-processing*, *feature extraction*, dan *detection* atau *segmentation* (Rahmadian and Widyartono, 2020). Salah satu contoh penerapan *machine vision* diilustrasikan pada Gambar 2.2.



Gambar 2. 2 Ilustrasi Penerapan Teknologi *Machine Vision* di Industri

### 2.3. *Barcode*

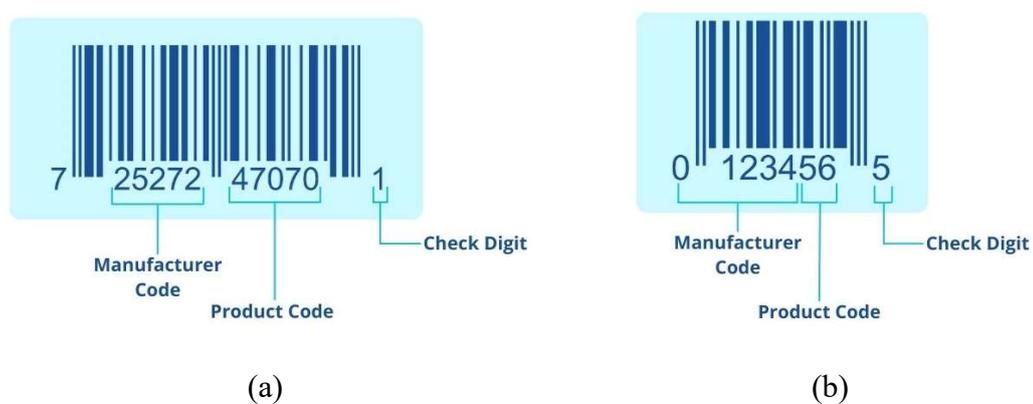
*Barcode* merupakan kode identifikasi berupa rangkaian batang vertikal hitam dan putih dengan lebar yang berbeda yang mewakili angka-angka (Susianto, Ridhawati and Syafitri, 2021).

### 2.3.1. Barcode Satu Dimensi

*Barcode* satu dimensi (1D) atau *linear barcode* merupakan *barcode* yang hanya terdiri dari baris-baris. Terdapat banyak jenis *barcode* 1D yang berbeda, diantaranya adalah UPC, EAN, Code 39, Code 128, Codabar, dan Interleaved 2 of 5 (ITF).

#### 2.3.1.1. Universal Product Code (UPC)

UPC merupakan *barcode* yang berbentuk numerik dan memiliki panjang baris tetap yang hanya terdiri dari karakter numerik 0 hingga 9. Jumlah data yang dapat disimpan pada UPC berkisar 2 hingga 12 digit per *barcode*. UPC biasanya digunakan pada produk-produk kecil atau eceran. UPC memiliki *check digit* yang digunakan untuk membantu memverifikasi keakuratan nomor yang dikodekan dan digunakan untuk menginformasikan bahwa data dalam *barcode* belum diubah atau rusak. Kekurangan dari *barcode* EAN ini adalah tidak memiliki mekanisme koreksi kesalahan, yang berarti *barcode* tidak dapat dibaca jika secara fisik rusak atau terdistorsi (Cox, 2023).

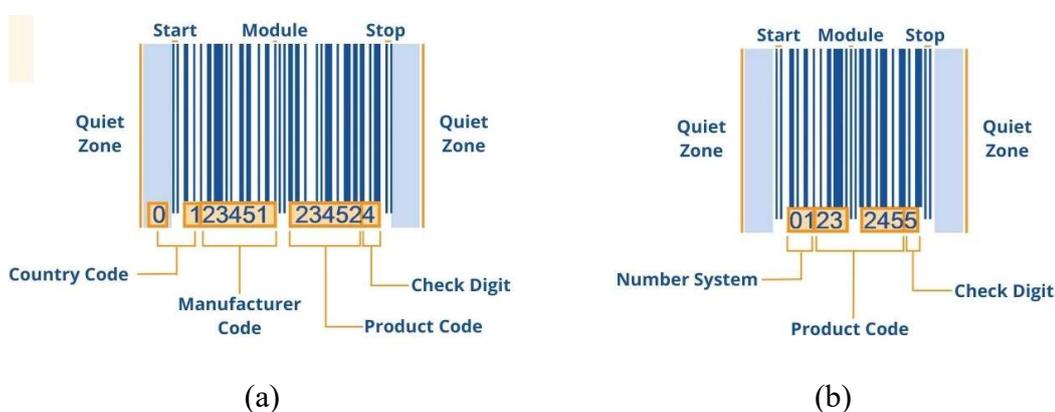


Gambar 2. 3 Struktur UPC; (a) UPC-A dan (b) UPC-E

Pada UPC terdapat dua varian yang sering digunakan, yaitu UPC-A dan UPC-E. UPC-A (Gambar 2.3a) merupakan versi standar dari UPC yang terdiri dari 12 digit numerik sedangkan UPC-E (Gambar 2.3b) merupakan versi singkat dari UPC-A yang terdiri dari 8 digit numerik.

### 2.3.1.2. *European Article Numbering (EAN)*

EAN merupakan barcode yang diimplementasikan oleh *Internasional Article Numbering Association* di Eropa. Standar ini digunakan untuk penggunaan internasional. *Barcode* EAN terdiri dari serangkaian batang paralel dengan lebar dan spasi yang bervariasi dengan setiap batang mewakili karakter tertentu. *Barcode* EAN dapat mengkodekan digit numerik dari 0 hingga 9. Jumlah data yang dapat disimpan bervariasi antara 2 hingga 13 digit. *Barcode* EAN memiliki *check digit* yang digunakan untuk membantu memverifikasi keakuratan nomor yang dikodekan dan digunakan untuk menginformasikan bahwa data dalam *barcode* belum diubah atau rusak. Kekurangan dari *barcode* EAN ini adalah tidak memiliki mekanisme koreksi kesalahan, yang berarti *barcode* tidak dapat dibaca jika secara fisik rusak atau terdistorsi (Cox, 2023).

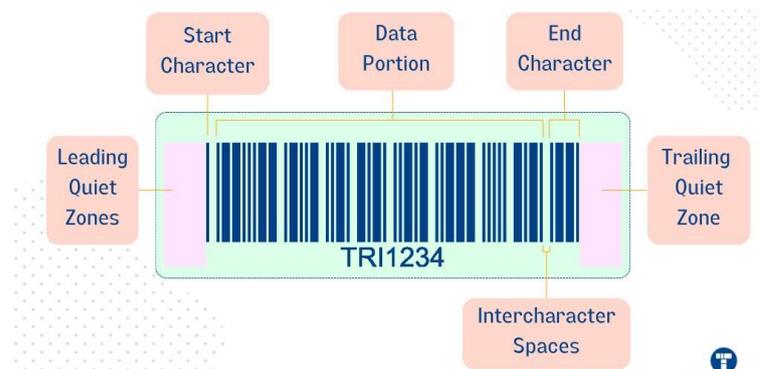


Gambar 2. 4 Struktur EAN; (a) EAN-13 dan (b) EAN-8

Pada *barcode* EAN, terdapat dua metode yang sering digunakan, yaitu EAN-8 dan EAN-13. EAN-8 (Gambar 2.4b) terdiri dari 8 digit numerik yang berisi kode negara (*number system*), kode produk, dan *check digit*. Sedangkan EAN-13 (Gambar 2.4a) terdiri dari 13 digit numerik yang berisi kode negara, kode perusahaan, kode produk, dan *check digit*.

### 2.3.1.3. Code 39

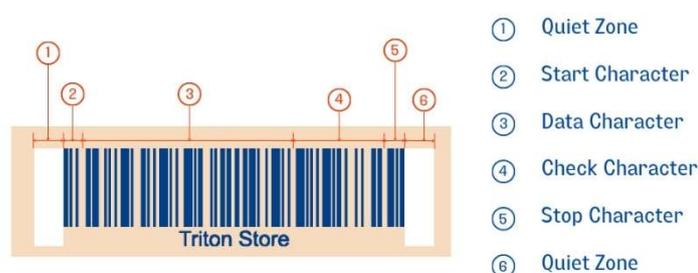
*Code 39* atau *code 3 of 9* merupakan *barcode* alphanumerik yang dapat mengkode alfabet kapital (A-Z), numerik (0-9), dan beberapa karakter lain seperti \$, /, +, %, \*, -, titik, dan spasi. Kapasitas data yang dapat disimpan sebanyak 43 karakter. Satu karakter *Code 39* terdiri dari 9 elemen, yaitu 5 bar (garis vertikal hitam) dan 4 spasi (garis vertikal putih) yang disusun bergantian antara bar dan spasi. Tiga dari 9 elemen tersebut lebih tebal dari yang lain, oleh karena itu kode ini disebut *Code 3 of 9*. *Code 39* memiliki mekanisme *self-checking* sehingga tidak memerlukan *check digit* untuk memeriksa *error* pada kode (Cox, 2023). Struktur *Code 39* ditunjukkan pada Gambar 2.5.



Gambar 2. 5 Struktur *Code 39*

#### 2.3.1.4. Code 128

*Code 128* merupakan jenis *barcode* satu dimensi yang dirancang dengan tujuan memberikan kerapatan yang lebih tinggi sekaligus mampu mengakomodasi karakter alphanumerik (full ASCII). *Code 128* tidak memiliki batasan untuk jumlah karakter yang dapat dikodekan. Setiap karakter pada *Code 128* terdiri dari 6 elemen, yaitu 3 bar dan 3 spasi. *Code 128* memiliki *check digit* yang berfungsi untuk memeriksa *error* atau kerusakan (Cox, 2023). Struktur *Code 128* ditunjukkan pada Gambar 2.6.

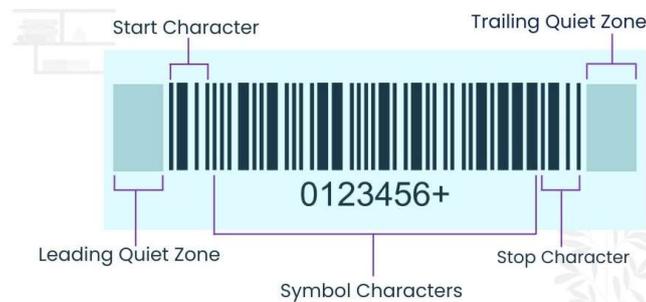


Gambar 2. 6 Struktur *Code 128*

#### 2.3.1.5. Codabar

*Codabar* merupakan jenis *barcode* satu dimensi yang didefinisikan pada standar AIM-BC3-2000 *Uniform Symbology Specification* dan dapat mengkodekan digit numerik (0-9) dan lima karakter khusus, yaitu plus (+), minus (-), slash (/), titik dua (:), simbol dolar (\$), dan titik (.). *Codabar* juga dapat mengkodekan huruf a – d tetapi hanya sebagai simbol *start* dan *stop*. *Codabar* memiliki mekanisme *self-checking* sehingga tidak memerlukan *check digit* untuk memeriksa *error* pada kode. Akan tetapi, *Codabar* memiliki toleransi *error* yang sedikit terhadap kerusakan dan distorsi sehingga ketika sebagian kecil dari *Codabar* rusak atau terdistorsi dapat mengakibatkan kesulitan dalam memulihkan data (Cox, 2023). *Codabar* memiliki

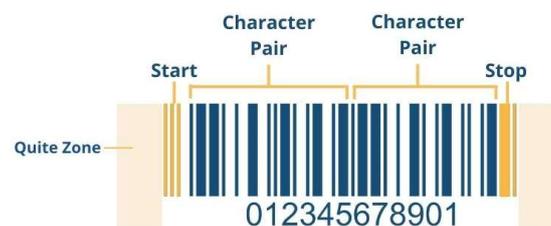
struktur yang terdiri dari *leading quiet zone*, *start character*, *symbol character*, *stop character*, dan *trailing quiet zone*, struktur tersebut ditunjukkan pada Gambar 2.7



Gambar 2. 7 Struktur *Codabar*

#### 2.3.1.6. *Interleaved 2 of 5 (ITF)*

ITF merupakan jenis *barcode* satu dimensi yang dirancang untuk mengkodekan data dengan cepat dan akurat. ITF didefinisikan dalam standar ISO/IEC 16390 dan biasanya digunakan dalam industri distribusi dan pergudangan. ITF dapat mengkodekan angka (0-9) dengan maksimum 32 digit. Setiap digit pada kode ITF dikodekan dengan 5 elemen, yaitu 2 elemen tebal dan 3 elemen tipis (Cox, 2023). Kode ITF memiliki struktur seperti yang ditunjukkan pada Gambar 2.8.



Gambar 2. 8 Struktur ITF

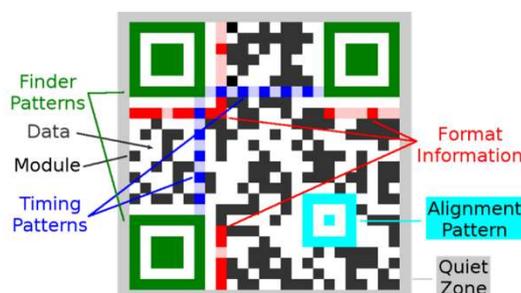
#### 2.3.2. *Barcode Dua Dimensi*

*Barcode* dua dimensi merupakan pengembangan dari *barcode* satu dimensi karena data yang besar dapat disimpan di dalam ukuran yang lebih kecil. Beberapa

contoh *barcode* dua dimensi diantaranya adalah *QR Code*, *PDF 417*, *Data Matrix*, dan *Aztec Code*.

### 2.3.2.1. *Quick Response Code (QR Code)*

*QR Code* merupakan jenis *barcode* dua dimensi yang menggunakan matrix kotak dan dapat mengkodekan 2.953 *bytes* data, 4.296 karakter alphanumerik, 7.089 karakter numerik, dan 1.817 karakter Kanji dan Kana. Hal ini membuat *QR Code* cocok untuk menyimpan tautan ke situs web atau aplikasi dan konten multimedia seperti file gambar, video, dan audio. *QR Code* memiliki keunggulan, yaitu dapat mempertahankan data hingga mengalami kerusakan sampai dengan 30%, dapat dibaca dari segala arah, dan ukuran cetak yang efisien (Cox, 2023). Adapun struktur *QR Code* ditunjukkan pada Gambar 2.9.

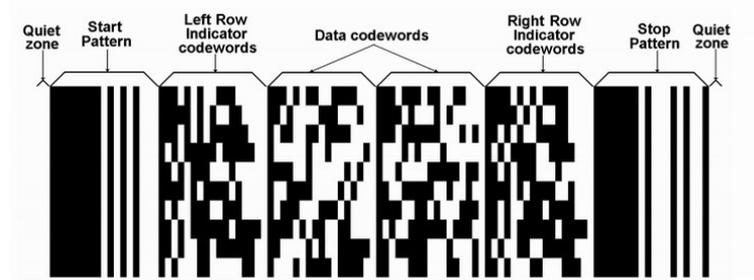


Gambar 2. 9 Struktur *QR Code*

### 2.3.2.2. *PDF 417*

*PDF 417* (ISO/IEC 15438:2001) merupakan *barcode* dua dimensi yang menggunakan pola batang atau spasi dengan lebar bervariasi untuk menyimpan data yang dikodekan. *PDF 417* dapat mengkodekan hingga 1.850 karakter alphanumerik, 2.710 digit numerik, atau 1.108 *byte* data dalam satu simbol. Satu

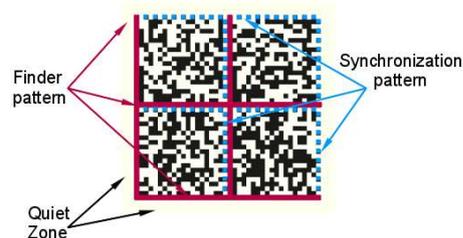
*barcode PDF 147* dapat menampung maksimal 929 *codeword* (Cox, 2023). Adapun struktur dari *PDF 417* ditunjukkan pada Gambar 2.10.



Gambar 2. 10 Struktur PDF 417

### 2.3.2.3. *Data Matrix*

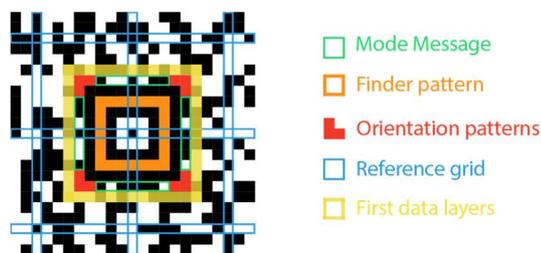
*Data Matrix* merupakan *barcode* dua dimensi yang menggunakan pola kotak hitam dan putih. *Data Matrix* dapat mengkodekan berbagai karakter termasuk 256 karakter ASCII, karakter ISO, dan karakter EBCDIC. *Data Matrix* dapat mengkodekan hingga 2.335 karakter alphanumerik, 3.116 karakter numerik, atau 1.556 *byte* data. *Data Matrix* memiliki kemampuan *error correction* dengan memanfaatkan algoritma *error correction* tingkat lanjut, yang dapat mempertahankan data hingga 30% kerusakan dan tetap dapat dibaca (Cox, 2023). Adapun struktur dari *Data Matrix* ditunjukkan pada Gambar 2.11.



Gambar 2. 11 Struktur Data Matrix

#### 2.3.2.4. *Aztec Code*

*Aztec Code* merupakan *barcode* dua dimensi yang memiliki ciri terdapat pola piramida Aztec berbentuk persegi dan bertingkat di bagian tengah kode. *Aztec Code* dapat mengkodekan hingga kapasitas 3.832 digit numerik, 3,067 karakter alfabet, atau 1.914 *byte* data. *Aztec Code* memiliki mekanisme *error correction* hingga batas tertentu sehingga dapat mempertahankan data ketika mengalami kerusakan (Cox, 2023). Adapun struktur *Aztec Code* ditunjukkan pada Gambar 2.12.



Gambar 2. 12 Struktur *Aztec Code*

#### 2.4. *QR Code*

*QR Code* atau *Quick Response Code* (Gambar 2.13) merupakan teknologi yang dikembangkan sebagai akibat dari keterbatasan fitur *barcode* linier satu dimensi yang disebut sebagai *barcode* klasik atau konvensional (Aktas, 2017). *QR Code* muncul sebagai akibat dari keterbatasan fitur *barcode* linier satu dimensi yang disebut sebagai *barcode* klasik atau konvensional (Soon, 2016).



Gambar 2. 13 *QR Code*

## **2.4.1. Karakteristik *QR Code***

### **2.4.1.1. *High capacity encoding of data***

*QR Code* memiliki kapasitas penyimpanan informasi yang tinggi. *QR Code* memungkinkan untuk menyimpan beberapa informasi yang ratusan kali dari kapasitas informasi barcode tradisional. (Qianyu, 2014). *QR Code* mampu menyimpan berbagai jenis data, misalnya karakter numerik dan alfabet, Kanji, Kana, Hiragana, simbol, biner, dan kode kontrol. Selain itu, *QR Code* dapat menyimpan maksimal 7.089 karakter dalam satu simbol (Mishra and Mathuria, 2017).

### **2.4.1.2. Ukuran cetakan kecil**

*QR Code* dapat menyimpan informasi baik arah horizontal maupun vertikal. Ketika jumlah data sama, ruang penyimpanan informasi *QR Code* hanya menyumbang 25% ruang penyimpanan informasi barcode satu dimensi (Qianyu, 2014).

### **2.4.1.3. Kemampuan Kanji dan Kana**

*QR Code* pada awalnya dirancang di Jepang yang menjadi alasan mengapa *QR Code* cocok untuk pengaturan Kanji dan Kana. Dalam hal pengkodean Kanji, satu karakter Kana atau Kanji dikodekan secara efisien dalam 13 bit (Qianyu, 2014). Ini berarti bahwa *QR Code* dapat membuat huruf Jepang dikodekan 20% lebih efisien daripada simbol dua dimensi lainnya (Aktas, 2017).

### **2.4.1.4. Tahan terhadap Kotoran dan Kerusakan**

*QR Code* memiliki kemampuan mengoreksi kesalahan (*error correction*), yaitu data dapat diperbaiki meskipun simbolnya kotor atau rusak sampai batas

tertentu (Mishra and Mathuria, 2017) dari mulai 7%, 15%, 25%, dan 30% (Aktas, 2017).

Tabel 2. 1 Level *Error Correction* pada *QR Code*

Level Error	Konstanta Simbolis	Kapasitas <i>error correction</i>
L	QRCapacity::ErrL	7% data
M	QRCapacity::ErrM	15% data
Q	QRCapacity::ErrQ	25% data
H	QRCapacity::ErrH	30% data

#### 2.4.1.5. Dapat Dibaca dari Segala Arah dalam 360°

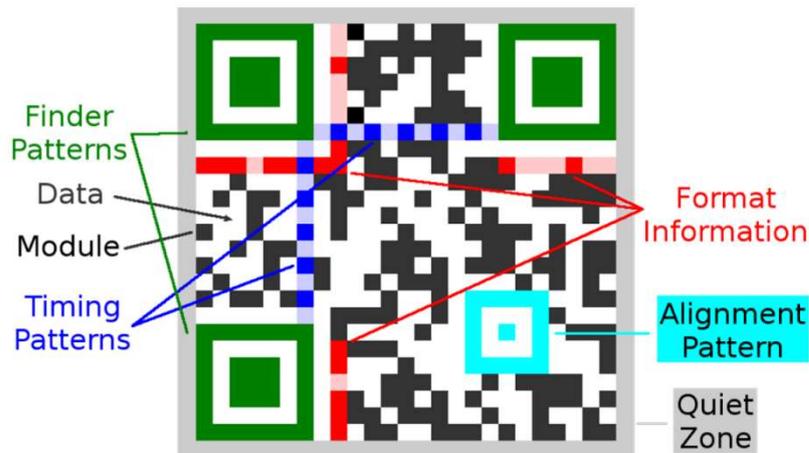
*QR Code* dapat dibaca dalam segala arah karena memiliki *finder pattern* terletak di tiga sudut simbol yang berfungsi sebagai informasi posisi dari *QR Code* (Qianyu, 2014). Selain itu, dikarenakan *QR Code* memiliki pola-pola tetap seperti separator, *timing pattern*, dan *alignment pattern* (Aktas, 2017).

#### 2.4.1.6. Fitur Penambahan Terstruktur

*QR Code* mampu mengklasifikasikan berbagai area data. Sebaliknya, banyak informasi yang disimpan dalam berbagai simbol dapat direformasi sebagai simbol data tertentu (Qianyu, 2014). Simbol data tunggal dapat dikategorikan menjadi 16 simbol yang memberikan kemudahan untuk dicetak (Mishra and Mathuria, 2017).

#### 2.4.2. Stuktur *QR Code*

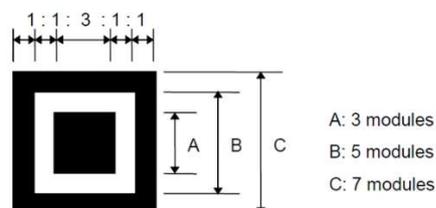
*QR Code* terdiri dari enam pola fungsi (Mishra and Mathuria, 2017). Adapun struktur *QR Code* ditunjukkan pada Gambar 2.14.



Gambar 2. 14 Struktur *QR Code*

#### 2.4.2.1. *Finder Pattern*

*Finder pattern* atau pola deteksi adalah pola deteksi posisi khusus yang terletak di tiga sudut, yaitu kiri atas, kanan atas, dan kiri bawah pada setiap simbol. Pola deteksi terdiri dari box gelap luar 7 x 7 modul, box terang bagian dalam 5 x 5 modul, dan box gelap padat di tengah 3 x 3 modul, seperti yang ditunjukkan pada Gambar 2.15. Rasio lebar modul setiap pola deteksi posisi adalah 1:1:3:1:1. Pola deteksi ini dirancang untuk menjadi pola yang tidak mungkin muncul di bagian lain dari *QR Code* sehingga pemindai dapat mencari rasio modul terang ke gelap ini untuk mendeteksi pola deteksi dan mengarahkan *QR Code* dengan benar untuk decode (Tiwari, 2016)



Gambar 2. 15 Struktur *Finder Pattern* pada *QR Code*

#### **2.4.2.2. Separator**

Separator adalah area spasi putih satu modul yang luas antara setiap *finder pattern* dan wilayah *encoding* (Tiwari, 2016). Fungsi dari separator adalah untuk memisahkan wilayah *encoding* dan masing-masing *finder pattern* (Mishra and Mathuria, 2017).

#### **2.4.2.3. Timing Pattern**

*Timing pattern* atau pola pewaktuan adalah garis putus-putus yang menghubungkan masing-masing *finder pattern* (Mishra and Mathuria, 2017). Terdapat dua *timing pattern*, yaitu pola pewaktuan horizontal dan pola pewaktuan vertikal. Pola pewaktuan horizontal ditempatkan di baris ke-enam *QR Code* di antara separator dan pola pewaktuan vertikal terletak di kolom ke-enam *QR Code* di antara separator. Pola-pola ini berguna untuk menentukan kerapatan pola simbol, koordinat modul, dan area *version information* (Tiwari, 2016).

#### **2.4.2.4. Alignment Pattern**

*Alignment pattern* atau pola penyelarasan memiliki bentuk seperti *finder pattern* hanya saja ukurannya lebih kecil dan posisinya tergantung dari versi *QR Code* (Mishra and Mathuria, 2017). *QR Code* versi 2 dan lebih besar harus memiliki *alignment pattern* dan jumlah *alignment pattern* tergantung pada versi simbol (Tiwari, 2016)

#### **2.4.2.5. Data Module**

*Data* atau wilayah pengkodean berisi *format information*, *version information*, *data and error corection code* (Tiwari, 2016). *Format information* terdiri dari 15 bit di sepanjang separator dan menyimpan informasi mengenai tingkat *error correction* dan pola penyamaran yang dipilih. *Version information* berisi informasi versi dari *QR Code* di mana *QR Code* memiliki versi 1 hingga 40 (Mishra and Mathuria, 2017).

#### **2.4.2.6. Quiet Zone**

*Quiet zone* atau zona tenang adalah area seluas 4 modul yang tidak berisi data dan digunakan untuk memastikan bahwa teks atau tanda di sekitarnya tidak boleh menyatukan data *QR Code* (Tiwari, 2016).

### **2.4.3. Dekode QR Code**

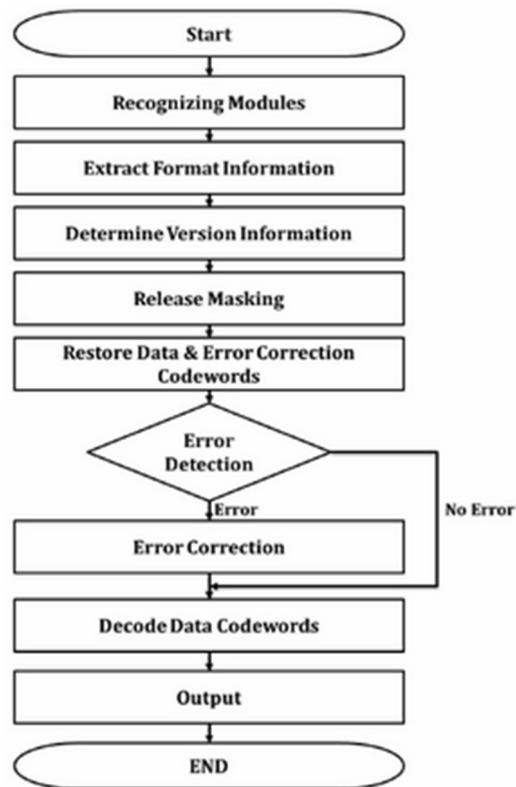
Dekode adalah proses menganalisa atau memaknai pesan yang terdapat pada sebuah kode. Proses dekode *QR Code* terdiri dari langkah-langkah sebagai berikut.

#### **2.4.3.1. Recognizing module**

*Recognizing module* yaitu mengenali modul gelap dan terang sebagai larik bit 0 dan 1 dengan mencari dan mendapatkan gambar simbol (Tiwari, 2016).

#### **2.4.3.2. Extract Format Information**

Mengekstraksi informasi format adalah proses mendekode informasi format, mengekstrak *masking pattern*, dan menerapkan *error correction* pada modul informasi format seperlunya serta untuk mendapatkan referensi *mask pattern* (Tiwari, 2016).



Gambar 2. 16 Flowchart Proses Dekode *QR Code*

#### 2.4.3.3. *Determine Version Information*

Menentukan informasi versi dengan mendekode dari area *version information* dan kemudian menentukan versi simbol *QR Code* (Tiwari, 2016).

#### 2.4.3.4. *Release Masking*

*Release masking* adalah proses untuk melepaskan penyamaran dengan menentukan XOR pola bit wilayah pengkodean dengan *mask pattern* yang referensinya telah diekstraksi dari *format information* (Tiwari, 2016).

#### 2.4.3.5. *Restore Data and Error Correction Codeword*

Langkah ini untuk mengembalikan data dan *error correction* kode pesan dengan membaca karakter simbol *QR Code* (Tiwari, 2016).

#### **2.4.3.6. Error Detection and Correction**

Langkah ini untuk mengidentifikasi *error* dan jika terdapat *error* yang terdeteksi maka akan diperbaiki dengan memanfaatkan kata sandi *error correction* (Tiwari, 2016).

#### **2.4.3.7. Decode Data Codewords**

Langkah ini untuk membagi kode data menjadi segmen-segmen sesuai dengan mode indikator dan jumlah indikator karakter. Selanjutnya, dekode karakter data sesuai dengan mode yang digunakan dan *output* teks yang didekode sebagai hasilnya (Tiwari, 2016).

### **2.5. Bahasa Pemrograman Python**

Python adalah bahasa pemrograman yang dikembangkan oleh Guido van Rossum pada tahun 1991. Python diklaim sebagai bahasa pemrograman yang mudah dipelajari karena bahasa yang digunakan sangat jelas, lengkap, dan mudah untuk dipahami. Python banyak digunakan sebagai bahasa pemrograman untuk membuat aplikasi berbasis *Artificial Intelligence* (AI) (Enterprise, 2019)

### **2.6. OpenCV**

OpenCV atau *Open Source Computer Vision Library* merupakan *library* yang dikembangkan oleh *Intel* yang ditujukan untuk pengolahan citra dinamis secara *real-time*. Fitur yang terdapat pada OpenCV diantaranya *face recognition*, *face tracking*, *face detection*, dan metode AI lainnya. OpenCV merupakan *library open source* untuk bahasa pemrograman C/C++ dan dikembangkan pula ke bahasa pemrograman Python, Java, dan Matlab (Susim and Darujati, 2021).

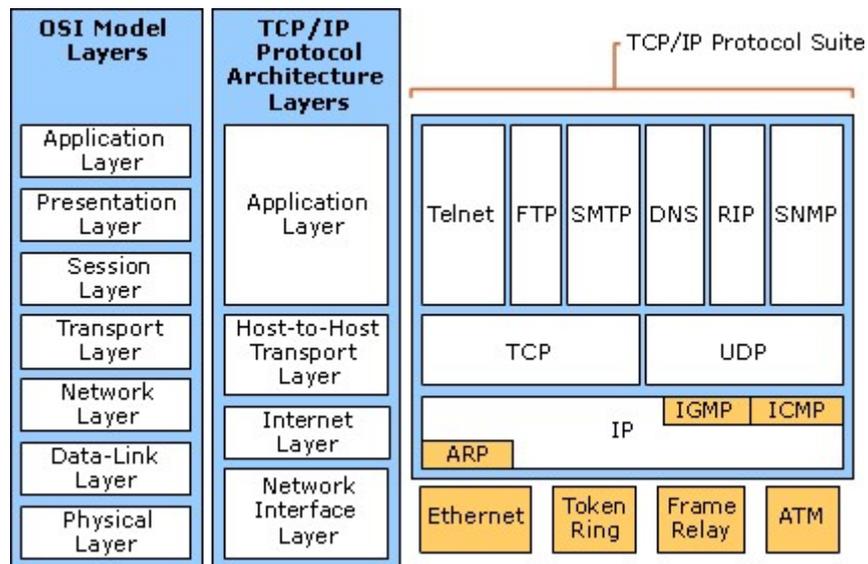
## 2.7. Pyzbar

Pyzbar adalah *library* yang digunakan untuk membaca dan mendekode barcode satu dimensi dan *QR Code*. Fitur yang terdapat pada Pyzbar diantaranya adalah implementasi yang mudah dengan Python, mampu beroperasi dengan PIL, OpenCV, numpy, dan data mentah, serta dapat mendekode lokasi barcode (Trehan, 2019).

Proses dekode *QR Code* menggunakan *library* Pyzbar dimulai dari kamera mengambil citra *QR Code*, selanjutnya citra *QR Code* tersebut masuk ke proses *scanning* untuk didapatkan sampel intensitas gelombang dari citra tersebut. Selanjutnya, dilakukan proses linear *scanning* untuk mengkonversi intensitas gelombang menjadi sampel lebar gelombang untuk didapatkan nilai pada setiap ukuran sampel gelombang tersebut. Tahap akhirnya adalah melakukan proses dekode sampel gelombang untuk didapatkan data utuh dari citra *QR Code* yang dideteksi.

## 2.8. *Transmission Control Protocol/Internet Protocol (TCP/IP)*

TCP/IP merupakan standar komunikasi data yang digunakan untuk menghubungkan komputer ke dalam jaringan internet maupun jaringan lokal (*Local Area Network* atau *LAN*). Protokol ini menggunakan skema pengalamatan yang disebut dengan alamat IP. TCP/IP memiliki empat layer, yaitu *network interface layer*, *internet layer*, *transport layer*, dan *application layer* yang ditunjukkan pada Gambar 2.17.



Gambar 2. 17 Layer TCP/IP

### 2.8.1. Network Interface Layer

Layer ini berfungsi untuk menyediakan media bagi sistem untuk mengirimkan data ke perangkat lain yang terkoneksi secara langsung. Adapun *interface* yang dapat bekerja di *layer* ini diantaranya adalah ethernet, *frame relay*, *token ring*, ATM, WAN, LAN, dan WiFi.

### 2.8.2. Internet Layer

Layer ini berisi protokol yang mempunyai tanggung jawab dalam pengalamatan dan enkapsulasi paket data jaringan. Adapun yang dapat bekerja pada *layer* ini diantaranya adalah *Internet Protocol* (IP), *Address Resolution Protocol* (ARP), *Internet Control Message Protocol* (ICMP), dan *Internet Group Management Protocol* (IGMP).

### **2.8.3. Transport Layer**

Layer ini berfungsi untuk membentuk sambungan antara *host* penerima dan pengirim sebelum kedua *host* berkomunikasi dan seberapa sering kedua *host* ini akan mengirim acknowledgement dalam sambungan tersebut satu sama lainnya. Adapun yang dapat bekerja pada layer ini adalah *Transmission Control Protocol* (TCP), *User Data Protocol* (UDP), dan *Datagram Transport Layer Security* (DTLS).

### **2.8.4. Application Layer**

Layer ini berfungsi untuk melayani permintaan data atau service. Layer ini termasuk seluruh proses yang menggunakan transport layer untuk mengirimkan data. Adapun yang dapat bekerja pada layer ini diantaranya adalah MQTT, CoAP, HTTP, AMQP, Modbus TCP dan sebagainya.

## **2.9. Modbus TCP**

Modbus TCP atau Modbus TCP/IP merupakan protokol Modbus RTU yang menggunakan *interface* TCP yang berjalan pada media Ethernet. Modbus TCP adalah protokol yang berjalan pada *layer* aplikasi TCP/IP. Modbus TCP mengkombinasikan jaringan fisik (Ethernet) dengan networking standard (TCP/IP) dan metode standar untuk merepresentasikan data (Modbus sebagai protokol aplikasi). Modbus TCP menggunakan topologi jaringan *client* dan *server*. TCP merupakan protokol berbasis koneksi dimana *client* harus selalu terhubung dengan *server* saat pertukaran data (Gambar 2.18). *Client* dan *server* Modbus TCP

mengirim dan menerima data Modbus melalui port 502 (Amaliawati, Wibowo and Murti, 2020).



Gambar 2. 18 Skema *Client-Server* pada Modbus TCP

Modbus *server* merupakan perangkat atau *software* yang memiliki data yang dapat diakses oleh perangkat lain dalam jaringan yang menggunakan protokol Modbus. Perangkat yang digunakan sebagai server dapat menanggapi permintaan baca (*read*) dan tulis (*write*) dari perangkat lain yang bertindak sebagai *client*. Contoh perangkat yang biasa digunakan sebagai Modbus *server* salah satunya adalah PLC (*Programmable Logic Controller*) (Acromag Incorporated, 2005).

Modbus *client* merupakan *hardware* atau *software* yang menginisiasi permintaan baca (*read*) dan tulis (*write*) ke perangkat Modbus *server*. Perangkat yang digunakan sebagai *client* dapat membuat permintaan (*request*) ke Modbus *server* untuk membaca atau menulis data tertentu. Setelah membuat *request*, Modbus *client* menunggu *response* dari Modbus *server* yang berisi data yang diminta atau status operasi yang dilakukan. Contoh perangkat yang biasa digunakan sebagai Modbus *client* salah satunya adalah HMI (*Human Machine Interface*) (Amaliawati, Wibowo and Murti, 2020).

*Request* merupakan proses mengirimkan pesan atau data ke Modbus server. Pada Modbus *request* terdapat dua metode, yaitu *write request* dan *read request*. *Write request* digunakan untuk mengirimkan data dari Modbus *client* ke Modbus

*server*. Modbus *client* akan mengirimkan *request* yang berisi alamat register yang berisi nilai yang akan ditulis. Sedangkan *read request* merupakan permintaan Modbus *client* ke Modbus *server* untuk membaca data dari register Modbus tertentu. *Response* merupakan pesan yang dikirim oleh Modbus *server* sebagai tanggapan atas *request* yang diterima dari Modbus *client*. Pesan *response* berisi informasi yang diminta oleh Modbus *client* (Acromag Incorporated, 2005).

### 2.10. Belt Conveyor



Gambar 2. 19 *Belt Conveyor*

*Belt conveyor* (Gambar 2.19) merupakan salah satu jenis konveyor yang digunakan sebagai alat pengangkut produk dalam sebuah proses produksi. *Belt conveyor* terdiri dari beberapa bagian utama diantaranya adalah kerangka (*frame*), sabuk (*belt*), *roller idler*, dan unit penggerak (Aosoby, Rusianto and Waluyo, 2016).

### 2.11. Raspberry Pi

Raspberry Pi (Gambar 2.20) dikenal sebagai *single-board computer*, yang artinya komputer yang dibangun di atas papan sirkuit cetak tunggal. Raspberry Pi berukuran kecil yang dapat melakukan apa pun yang dapat dilakukan oleh komputer yang lebih besar dan lebih haus daya, meskipun tidak tentu secepat itu. Semua

model Raspberry Pi kompatibel, artinya perangkat lunak yang ditulis untuk satu model akan berjalan pada model lain (Halfacree, 2020).

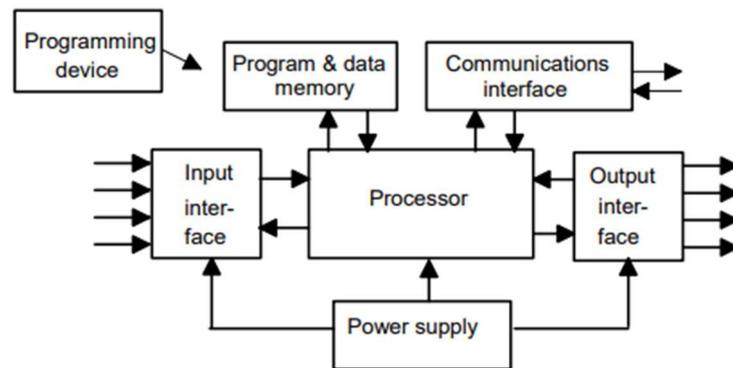


Gambar 2. 20 Raspberry Pi 3

Raspberry Pi menggunakan *System on a Chip* (SoC) dari Broadcom BCM2835 hingga BCM 2837 (Raspberry Pi 3), juga sudah termasuk prosesor ARM1176JZFS MHz bahkan 1.2GHz 64-bit quad-core ARMv8 CPU untuk Raspberry Pi 3, GPU VideoCore IV dan kapasitas RAM hingga 1 GB.

### 2.12. *Programmable Logic Control* (PLC)

*Programmable Logic Control* (PLC) adalah bentuk khusus dari pengontrol berbasis mikroprosesor yang menggunakan memori yang dapat diprogram untuk menyimpan intruksi dan untuk mengimplementasikan fungsi seperti logika, *sequencing*, *timing*, *counting*, dan aritmatika untuk mengontrol mesin dan proses serta dirancang untuk dioperasikan oleh pengguna dengan pengetahuan terbatas tentang komputer dan bahasa komputasi (Bolton, 2006).



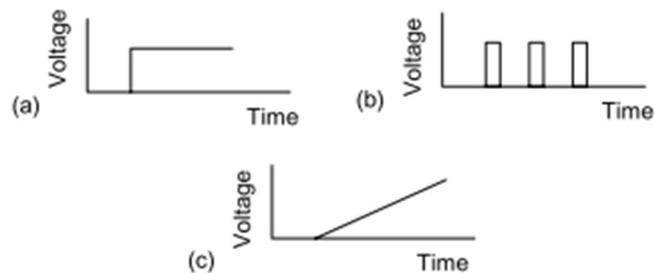
Gambar 2. 21 Komponen Fungsional Dasar pada PLC

Sistem PLC memiliki komponen fungsional dasar (Gambar 2.21) yang terdiri dari unit prosesor, memori, unit catu daya, bagian antarmuka input/output, antarmuka komunikasi, dan perangkat pemrograman.

1. *Central Processing Unit* (CPU) merupakan unit berisi mikroprosesor yang menafsirkan sinyal input dan melakukan tindakan kontrol sesuai dengan program yang disimpan di dalam memori serta mengkomunikasikan keputusan sebagai sinyal tindakan ke output (Bolton, 2006).
2. *Power supply unit* adalah unit catu daya yang pada PLC digunakan sebagai unit yang mengubah listrik AC menjadi listrik DC 5V yang dibutuhkan untuk prosesor dan rangkaian di dalam modul *interface* I/O (Bolton, 2006).
3. *Programming device* digunakan untuk memasukkan program yang diperlukan ke dalam memori prosesor. Program disusun pada *software* dan kemudian di-*upload* ke unit memori PLC (Bolton, 2006).
4. *Memory unit* adalah unit tempat program disimpan untuk nantinya akan digunakan sebagai tindakan kontrol yang akan dilakukan oleh mikroprosesor

dan memori menyimpan data dari input untuk diproses serta data untuk output mengeluarkan tindakan (Bolton, 2006).

5. *Input/Output* merupakan bagian tempat prosesor menerima informasi dari perangkat eksternal dan mengkomunikasikan informasi ke perangkat eksternal lainnya. Perangkat I/O dapat diklasifikasikan sebagai sinyal diskrit, digital atau analog seperti yang ditunjukkan pada Gambar 2.22 (Bolton, 2006).



Gambar 2. 22 Sinyal: (a) Diskrit, (b) Digital, (c) Analog

6. *Communication interface* digunakan untuk menerima dan mengirimkan data pada jaringan komunikasi dari atau ke PLC jarak jauh. Antarmuka ini berkaitan dengan tindakan seperti verifikasi perangkat, akuisisi data, sinkronisasi antara aplikasi pengguna dan manajemen koneksi (Bolton, 2006).

### 2.13. Arduino Nano



Gambar 2. 23 Arduino Nano

Arduino Nano (Gambar 2.23) merupakan salah satu jenis *board* Arduino yang berukuran kecil dengan menggunakan mikrokontroler Atmega 328P. Arduino Nano memiliki 8 pin analog (A0-A7) dan 14 pin digital (D0-D13) serta memiliki 6 pin sebagai output PWM (*Pulse Width Modulation*). Arduino Nano Atmega 328P memiliki *clock speed* sebesar 16 MHz (Suari, 2017).

### 2.14. *Human Machine Interface* (HMI)



Gambar 2. 24 *Human Machine Interface* (HMI)

HMI atau *Human Machine Interface* (Gambar 2.24) merupakan sebuah monitor yang difungsikan sebagai antarmuka antara operator dengan mesin sebagai bentuk komunikasi dengan mesin yang dioperasikan. HMI digunakan sebagai alat monitoring proses suatu sistem yang melibatkan mesin dengan operator dengan

cara menampilkan proses yang sedang berlangsung pada mesin yang dikontrol (Sadi, 2020).

### 2.15. Webcam



Gambar 2. 25 Webcam

*Webcam* (Gambar 2.25) merupakan kamera yang digunakan sebagai pengambil gambar yang dikendalikan oleh komputer atau oleh sebuah jaringan (Mulyanto *et al.*, 2021). *Webcam* dapat diakses melalui *World Wide Web* (WWW) maupun aplikasi lain yang terhubung ke perangkat melalui *port* USB (PERKASA, 2019).

### 2.16. Motor DC



Gambar 2. 26 Motor DC

Motor DC (*Direct Current*) (Gambar 2.26) adalah motor yang dapat bergerak atau berputar dengan supplay tegangan searah (DC). Komponen utama pada motor DC adalah rotor dan stator. Rotor merupakan bagian yang berputar sedangkan stator

merupakan bagian yang diam. Pada rotor terdapat kumparan-kumparan yang fungsinya untuk membangkitkan medan elektromagnetik, sedangkan pada stator terdapat magnet permanen yang fungsinya membentuk medan magnet (Nugroho and Agustina, 2015). Motor DC biasanya dimanfaatkan putarannya untuk menggerakkan suatu alat atau komponen.

### 2.17. Motor Servo



Gambar 2. 27 Motor Servo

Motor servo (Gambar 2.27) merupakan motor yang memiliki sistem *closed feedback*. Motor servo hanya dapat bergerak sampai sudut tertentu saja dan tidak secara kontinyu. Akan tetapi, untuk beberapa aplikasi motor servo dapat dimodifikasi supaya bergerak secara kontinyu (Salim, Saragih and Hidayat, 2020). Salah satu motor servo yang biasa digunakan dalam sebuah prototipe adalah motor servo Arduino yang terdiri dari motor, potensio, rangkaian *gear*, dan rangkaian kontrol. Cara pengontrolan sudut pada motor servo Arduino yakni dengan cara memberikan nilai PWM (*Pulse Width Modulation*) pada pin signal motor servo (Sitohang, Mamahit and Tulung, 2018).

Motor servo terbagi menjadi dua jenis, yaitu motor servo standar dan *continous*. Motor servo standar mampu berputar hingga  $180^{\circ}$  dan motor servo *continous* mampu berputar hingga  $360^{\circ}$ . Implementasi dari motor servo standar

digunakan pada bidang robotika yaitu pada sistem lengan robot, sedangkan motor servo *continous* diaplikasikan pada sistem *mobile robot* (Lestari, Satrianansyah and Mutia, 2019).

### 2.18. *Linear Actuator*



Gambar 2. 28 *Linear Actuator*

*Linear actuator* (Gambar 2.28) adalah sebuah komponen yang hanya dapat bergerak maju dan mundur. *Linear actuator* digerakkan menggunakan motor yang dirancang untuk dapat bergerak linier. *Linear actuator* umumnya digunakan sebagai alat pendorong maupun penarik benda. Pada beberapa aplikasi, *linier actuator* digunakan sebagai pengganti pneumatik atau hidrolis yang digunakan sebagai silinder karena pengoperasiannya lebih mudah (Hermawan and Rochardjo, 2022).

### 2.19. Sensor Photoelektrik



Gambar 2. 29 Sensor Photoelektrik

Sensor photoelektrik (Gambar 2.29) merupakan sensor yang memanfaatkan cahaya yang fungsinya untuk mendeteksi keberadaan benda yang berwujud padat. Berdasarkan prinsip kerjanya, sensor photoelektrik dibagi menjadi dua jenis, yaitu jenis refleksi yang mana transmitter dan receiver sensor dipasang dalam posisi saling berdampingan dan jenis transmisi yang mana transmitter dan receiver sensor dipasang berhadapan (Abu Bakar *et al.*, 2020).

## **2.20. Penelitian Terdahulu**

Penelitian mengenai perancangan sistem pengepakan otomatis sudah banyak dilakukan. Penelitian-penelitian terdahulu dijadikan penulis sebagai pembanding dan acuan dalam penelitian yang akan dilaksanakan.

Penelitian oleh (Yoanda, Kamal and Azhar, 2018) mengenai sistem pengepakan otomatis menggunakan PLC sebagai kontrolernya. Penelitian ini merancang sistem pengepakan tepung berbasis PLC yang bertujuan untuk memaksimalkan waktu pengepakan yang singkat dengan hasil yang lebih banyak dibandingkan dengan hasil pengepakan secara manual. Monitoring pengepakan pada penelitian ini menggunakan LCD yang berfungsi untuk menampilkan jumlah hasil tepung yang telah dikemas. Sistem ini menggunakan sensor proximity sebagai pendeteksi kemasan ditempat pengisian. Kekurangan dari sistem ini adalah monitoring masih menggunakan LCD dan proses pendeteksian kemasan saat akan dilakukan pengisian masih belum akurat dikarenakan kecepatan konveyor pada sistem masih belum sesuai.

Penelitian oleh (Putra and Golwa, 2021) mengenai penggunaan *barcode reader* pada proses penyortiran benda berdasarkan *barcode ID*. Penelitian ini merancang sistem pendeteksian atau *sorting* benda berdasarkan *barcode ID* 1D (kode batang) dan 2D (*QR Code*) dengan menggunakan *barcode reader* tipe laser sebagai alat pemindainya. Metode yang dilakukan pada penelitian ini adalah dengan meletakkan produk dengan barcode mengarah ke *barcode reader* dan hasil tangkapan *barcode reader* ditransfer ke mikrokontroler Raspberry Pi dan PLC untuk tindakan pemilahan benda. Kekurangan dari penelitian ini adalah masih menggunakan *barcode reader* tipe laser dalam melakukan pemindaian barcode ID yang mana *barcode reader* masih dapat melakukan kegagalan pada saat pemindaian.

Penelitian oleh (Khamidullin *et al.*, 2019) mengenai pemanfaatan *QR Code* pada proses produksi. Penelitian ini membahas mengenai pemanfaatan *QR Code* dalam meningkatkan kecepatan dan ketepatan kerja dalam melakukan pendataan produk. *QR Code* dimanfaatkan sebagai kode untuk mengidentifikasi jenis produk karena minim dari kesalahan dan meringankan beban kerja pada saat melakukan pendataan.

Berdasarkan penelitian-penelitian terdahulu yang telah diuraikan, terdapat beberapa kekurangan. Oleh karena itu, penulis memiliki gagasan untuk memanfaatkan kekurangan-kekurangan tersebut sebagai acuan untuk membuat penelitian mengenai perancangan sistem pengepakan otomatis dengan menggabungkan konsep dari penelitian terdahulu. Penelitian ini akan merancang sistem pengepakan benda secara otomatis dengan menggunakan PLC sebagai

kontrol utamanya serta penggunaan HMI sebagai alat monitoring sistem yang bertujuan untuk mempercepat proses pengepakan. Sistem pengepakan yang dirancang memanfaatkan *QR Code* sebagai kode untuk mengidentifikasi jenis dan jumlah benda yang harus dimasukkan ke dalam kemasan. Alat pemindai yang digunakan untuk memindai citra *QR Code* menggunakan kamera *webcam* Taffware US829 berbasis Raspberry Pi. Selain itu, sistem ini juga menerapkan teknologi *machine vision* yang artinya sistem akan berjalan berdasarkan hasil tangkapan kamera atau berdasarkan hasil pemindaian.