BAB 2
TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Banyaknya kesalahan dalam proses operasional Gudang *Raw Material* pada PT XYZ menjadi masalah yang diangkat dan akan diselesaikan. Berdasarkan masalah yang dihadapi PT XYZ, tinjauan pustaka dilakukan dengan melakukan studi pada penelitian terdahulu untuk memperoleh referensi mengenai metode dan proses perancangan solusi dari masalah yang serupa. Proses peninjauan pustaka dilakukan dengan mencari kata kunci yang berhubungan dengan kesalahan operasional pada gudang, seperti akurasi stok, kesalahan administratif gudang, dan kesalahan alokasi barang. Penelusuran tinjauan pustaka kata kunci menghasilkan pengelompokan masalah yang dihadapi, yang kemudian akan mengarahkan pada beberapa solusi serupa untuk menyelesaikan masalah tersebut.

2.1.1 Keterbatasan Sistem Pencatatan Konvensional Gudang

Tampubolon & Simanjuntak (2023) menemukan bahwa proses pencatatan manual di gudang memicu *human error* yang berakibat pada kehilangan data dan ketidaksesuaian informasi barang. Masalah serupa diidentifikasi oleh Nasri et al. (2022), yang menyatakan operasional manual membuat pekerjaan membutuhkan waktu lebih lama serta dan kurang akurat. Selain itu, Hidayat et al. (2024) dan Rahayu et al. (2022) secara spesifik merumuskan masalah pada pengelolaan stok menggunakan Microsoft Excel. Adapun Microsoft Excel dinilai belum mampu menyajikan data secara *real-time*, kurang akurat, dan kurang optimal untuk menangani data besar (*Big Data*), sehingga menyebabkan ketidakakuratan data stok.

Berdasarkan permasalahan yang teridentifikasi, Tampubolon & Simanjuntak (2023) merancang sistem informasi berbasis *website* dengan menerapkan metode *Agile Development* dan pengujian *Black Box*. Hidayat et al. (2024) mengembangkan *Warehouse Management System* (WMS) berbasis *web* melalui metode *Waterfall* yang berfokus pada tahapan perancangan, pengembangan, dan evaluasi. Nasri et al. (2022) merancang aplikasi sistem persediaan serupa menggunakan pendekatan *System Development Life Cycle* (SDLC) dan analisis PIECES untuk mengevaluasi kinerja sistemnya. Rahayu et al. (2022) memilih penggunaan pemodelan berbasis objek dan visualisasi *Unified Modeling Language* (UML).

Implementasi dari berbagai perancangan sistem tersebut menunjukkan adanya perubahan dalam mengatasi masalah serupa, yakni pencatatan konvensional. Sistem yang dikembangkan oleh Hidayat et al. (2024) dan Rahayu et al. (2022) berhasil

meningkatkan akurasi data serta menangani masalah pengolahan data persediaan secara lebih efektif. Nasri et al. (2022) membuktikan bahwa sistem berbasis web mampu memfasilitasi pemantauan persediaan secara akurat dan *real-time*.

2.1.2 Ketidakakuratan Identifikasi dan Monitor Gudang

Setyawan et al. (2022) mengidentifikasi bahwa proses manajemen gudang yang tidak terintegrasi dan dilakukan secara manual menyebabkan ketidakakuratan stok. Temuan serupa disampaikan oleh Pitaloka et al. (2025) bahwa manajemen stok manual meningkatkan risiko kesalahan input data fisik persediaan di lapangan. Lebih lanjut, Mahzan & Lee (2021) menemukan bahwa proses inventori manual sangat rentan terhadap *human error* yang berakibat langsung pada kesalahan penghitungan fisik barang.

Untuk mengatasi kendala identifikasi fisik tersebut, Setyawan et al. (2022) mengusulkan perbaikan proses pengontrolan inventori menggunakan teknologi *Radio Frequency Identification* (RFID). Hal ini dilakukan dengan pendekatan pemodelan sistem (*system modeling*) berbasis objek dengan *Unified Modeling Language* (UML). Pitaloka et al. (2025) merancang integrasi sistem RFID dengan *Warehouse Management System* (WMS) menggunakan metode benchmarking dan simulasi sistem. Sebagai pendekatan alternatif, Mahzan & Lee (2021) merancang sistem otomasi penghitung objek berbasis Arduino dan sensor ultrasonik melalui metode perancangan sirkuit dan integrasi dengan *Internet of Things* (IoT).

Implementasi sistem otomasi dinilai mampu menekan kesalahan identifikasi yang dapat terjadi di gudang. Pitaloka et al. (2025) membuktikan bahwa sistem RFID yang terintegrasi dengan WMS mampu meningkatkan akurasi stok secara signifikan dari 85% menjadi 97%. Setyawan et al. (2022) menegaskan bahwa penggunaan RFID meningkatkan kecepatan proses serta akurasi data *real-time* dengan mengurangi intervensi aktivitas manual. Mahzan & Lee (2021) juga menunjukkan bahwa otomasi penghitungan objek efektif mengurangi kesalahan akibat kelalaian pada proses identifikasi manual.

2.1.3 Kelemahan Pelacakan Pergerakan dan Penempatan Material

Al-Shboul (2023) menemukan bahwa aktivitas manual selama proses penerimaan barang memicu potensi kesalahan yang besar dan memperlama waktu proses, sehingga berujung pada ketidakakuratan data stok. Di sisi lain, Dibattista et al. (2025) mengidentifikasi bahwa aktivitas monitoring dan pencatatan manual mengakibatkan ketidakakuratan stok serta menyulitkan pelacakan lokasi barang yang presisi di area penyimpanan. Selain itu, Anggara et al. (2024) merumuskan bahwa proses manajemen

barang secara manual tanpa sistem terintegrasi menyebabkan operasional menjadi kurang efisien dan tidak akurat dalam melacak pergerakan material di gudang.

Untuk menyelesaikan permasalahan pelacakan dan penempatan material, Al-Shboul (2023) menerapkan teknologi RFID guna menggantikan proses identifikasi manual menggunakan metode regresi linear serta pemodelan desain gudang (*warehouse design modeling*). Dibattista et al. (2025) merancang *Autonomous* RFID *Robot Dog* yang menggunakan *Single Antenna* UHF (*ultra-high* frequency) RFID untuk melakukan pemindaian otomatis dan triangulasi lokasi barang melalui metode pemetaan (*mapping*) dan navigasi. Sementara itu, Anggara et al. (2024) merancang sistem informasi berbasis web terintegrasi dengan teknologi QR-*code* menggunakan metode *Rapid Application Development* (RAD) guna mempercepat proses pengenalan barang.

Al-Shboul (2023) menunjukkan bahwa implementasi RFID menghasilkan basis data *real-time* yang mampu menurunkan potensi kesalahan operasional sekaligus meningkatkan pelayanan. Penelitian Dibattista et al. (2025) membuktikan bahwa robot yang terintegrasi dengan RFID mampu mendeteksi identitas barang hingga tingkat keberhasilan 100% pada jarak tertentu dengan tingkat *error* triangulasi lokasi yang rendah. Selain itu, penerapan sistem informasi berbasis web QR-*code* yang dikembangkan oleh Anggara et al. (2024) terbukti mampu menurunkan potensi kesalahan operasional dan *human error* pada manajemen gudang.

Tabel 2.1 Ringkasan tinjauan pustaka keterbatasan sistem pencatatan konvensional gudang

No.	Pustaka	Objek	Masalah	Solusi	Metode	*Tools*	Hasil
1	Tampubolon & Simanjuntak, 2023	Aplikasi *input* inventori berbasis *web*	*Human error* yang mengakibatkan kehilangan data dan ketidaksesuaian informasi barang pada gudang	Perancangan sistem informasi berbasis *web* untuk proses masuk dan keluarnya barang	*Agile development,* dan uji *Black Box*	1. *Use Case Diagram* 2. *Activity Diagram* 3. *Sequence Diagram* 4. PHP *Programming* 5. *MySQL*	Sistem informasi yang dapat digunakan untuk proses *input* data barang
2	Nasri et al., 2022	Sistem informasi untuk pencatatan persediaan berbasis *web* di PT Anugrah Niagatama Perkasa	Proses operasional manual mengakibatkan pekerjaan yang lama serta laporan yang dihasilkan terlambat dan sering kurang akurat	Perancangan aplikasi sistem persediaan berbasis *web*	Pendekatan *System Development Life Cycle*: Metode PIECES	1. Diagram level nol 2. *Use Case Diagram* 3. *Activity Diagram* 4. *Entity Relationship Diagram* 5. *Logical Record Structure*	Sistem dengan monitor data persediaan barang secara akurat dan *real time*
3	Hidayat et al., 2024	Sistem informasi *Monitoring Warehouse Management* pada PT Lisaboy	Pengelolaan stok barang manual dengan Microsoft Excel tidak *real time,* akurasi data rendah	Perancangan program *Warehouse Management System* (WMS) berbasis *website*	Metode *Waterfall*: 1. Perancangan 2. Pengembangan 3. Evaluasi	1. *Use Case Diagram* 2. *Activity Diagram* 3. *Deployment Diagram* 4. *Sequence Diagram* 5. *Entity Relationship Diagram* 6. *Logical Record Structure*	Penerapan program WMS berbasis *web* yang mampu mengurangi risiko kehilangan barang dan kesalahan *input*
4	Rahayu et al., 2022	Sistem Informasi *Warehouse Management System* (WMS) pada PT Citra Banjar Abadi	Manajemen *warehouse* menggunakan Microsoft Excel secara manual kurang mampu menangani *Big Data* sehingga menyebabkan ketidakakuratan data barang	Perancangan sistem informasi WMS	Pemodelan berbasis objek: 1. Visualisasi *Unified Modeling Language* 2. Pemrograman dan pembuatan aplikasi	1. *Use Case Diagram* 2. *Activity Diagram* 3. *Sequence Diagram* 4. *Class Diagram* 5. *Visual Paradigm* 6. PHP *Programming* 7. *MySQL* 8. *Sublime* 9. *XAMPP*	Penerapan sistem informasi yang mampu menangani masalah kurang akurasinya pengolahan data barang *warehouse*

Tabel 2.2 Ringkasan tinjauan pustaka ketidakakuratan identifikasi dan monitor gudang

No.	Pustaka	Objek	Masalah	Solusi	Metode	*Tools*	Hasil
1	Setyawan et al., 2022	Pengembangan sistem monitoring inventori pada PT Tanjung Jaya	Ketidakakuratan stok akibat proses *inventory control* yang tidak terintegrasi, tidak *real-time*, dan dilakukan secara manual.	Perbaikan metode proses *inventory control* menggunakan basis data *real time* dengan penggunaan teknologi RFID	Simulasi RFID dengan pemodelan sistem *(system modeling*) berbasis objek	1. *Use case diagram* 2. *Activity diagram* 3. *Sequence diagram*	Peningkatan kecepatan proses *inventory control,* peningkatan akurasi data *real time,* dan pengurangan aktivitas manual
2	Pitaloka et al., 2025	Merancang sistem RFID pada PT Bap	Manajemen stok secara manual, meningkatkan risiko kesalahan *input* data stok	Perancangan sistem RFID yang terintegrasi dengan *Warehouse Management System* (WMS)	1. *Benchmarking* 2. Simulasi sistem	1. *Fishbone* Diagram 2. *Value Stream Mapping* (VSM) 3. *Process Flow Diagram* 4. Time *Series* Chart 5. *Warehouse Management System*	Meningkatkan akurasi stok dari 85% menjadi 97%
3	Mahzan & Lee, 2021	Eliminasi kesalahan hitung stok pada Perusahaan Manufaktur ABC	Kesalahan dalam proses inventori manual karena adanya *human error*	Sistem otomasi penghitung objek berbasis Arduino dan modul *WiFi* ESP8266	1. Perancangan sirkuit Arduino dengan sensor *ultrasonic* 2. Integrasi sirkuit dengan IoT 3. Pemrograman sirkuit	1. 5 *Whys Analysis* 2. Arduino 3. MySQL	Proses perhitungan stok secara otomatis dan mengurangi kesalahan akibat proses manual

Tabel 2.3 Ringkasan tinjauan pustaka kelemahan pelacakan pergerakan dan penempatan material

No.	Pustaka	Objek	Masalah	Solusi	Metode	*Tools*	Hasil
1	Al-Shboul, 2023	Peningkatan akurasi inventori dan reduksi waktu penerimaan pada perusahan manufaktur	Aktivitas manual yang menyebabkan ketidakakuratan stok dengan potensi *human error* yang besar dan lamanya waktu penerimaan material	Penerapan RFID untuk menggantikan proses identifikasi dan monitoring manual	Implementasi RFID pada perusahaan pihak ketiga penyedia jasa penyimpanan barang (*warehouse*)	1. Metode regresi linear 2. *Warehouse design modeling*	Peningkatan akurasi dengan basis data *real-time*, penurunan potensi kesalahan operasional, serta peningkatan *service level*
2	Dibattista et al., 2025	Merancang *Autonomous* RFID *Robot Dog* untuk monitor inventori	Ketidakakuratan stok akibat aktivitas monitor dan pencatatan manual	*Robot dog* dengan UHF RFID *single antenna* untuk *scan* otomatis dan triangulasi lokasi barang	1. Integrasi RFID 2. *Mapping* dan navigasi 3. Triangulasi *robot dog*	1. *Unitree Go2* EDU 2. *Light Detection and Ranging* 3. RGB *camera* 4. RFID *reader* 5. *Jetson Orin* 6. *Software Development Kit*	*Robot dog* mampu mendeteksi *tag* RFID 100% pada jarak tertentu dengan *error* triangulasi yang rendah
3	Anggara et al., 2024	Sistem informasi *web-based* QR-*code* untuk manajemen gudang pada PT Dharma Precision Parts	Proses manajemen barang secara manual yang kurang efisien dan akurat	Perancangan sistem informasi *web-based* QR-*code*	Metode *Rapid Application Development* (RAD): 1. *Requirement planning* 2. *Workshop Design* 3. Implementation 4. Testing	1. *Use Case Diagram* 2. *Class Diagram* 3. PHP *Programming*	Penerapan sistem informasi yang mampu mengurangi kesalahan operasional dan *human error* pada manajemen gudang

2.2 Dasar Teori

Teori yang diuraikan pada bagian dasar teori yang digunakan untuk mendukung proses perancangan solusi yang telah dipilih, baik untuk analisis hingga metode dan *tool* untuk proses perancangan. Teori mengenai pergudangan merujuk pada mata kuliah Manajemen Pergudangan (STIN76602) dan *Radio Frequency Identification* merujuk pada mata kuliah Otomasi Industri (STIN32402). Kemudian teori mengenai K-Means *Clustering* merujuk pada mata kuliah *Data Mining* (STIN82602). Selain itu, proses pengembangan sistem informasi seperti pemodelan sistem berbasis objek*, Rapid Application Development* merujuk pada mata kuliah Proses Bisnis dan Sistem Informasi (STIN40502). Adapun dasar teori juga memuat *tools* yang digunakan dalam proses perancangan

2.2.1 Pergudangan

Pergudangan merupakan bagian dari jaringan logistik yang berfungsi sebagai tempat penyimpanan dan pengelolaan persediaan, menjaga kondisi barang, serta memastikan ketersediaan barang sesuai dengan kebutuhan. Pergudangan berperan penting untuk memastikan ketersediaan barang sesuai dengan kebutuhan produksi maupun permintaan pasar (Santalova et al., 2024). Proses operasional pergudangan dalam rantai pasok mendukung aktivitas *inbound* yang termasuk penerimaan dan penyimpanan bahan baku dari *supplier*, aktivitas *in-storage* termasuk pengendalian dan pengelolaan persediaan, serta *outbound* yang dimulai dari pengambilan hingga pengiriman barang (Manzini & Accorsi, 2024). Tingkat efektivitas dari aktivitas yang berlangsung kemudian berdampak pada *lead time*, *service level*, serta keseluruhan biaya logistik.

a. Peralatan pergudangan

Pergudangan memiliki dua tipe peralatan yang umum digunakan, yaitu peralatan statis dan peralatan dinamis. Berikut merupakan penjabaran dari kedua peralatan pergudangan.

i. Peralatan statis

Proses penyimpanan secara vertikal dapat mengoptimalkan ruang penyimpanan. Adapun yang umum digunakan dalam proses penyimpanan tersebut adalah rak palet. Rak palet digunakan dengan menyimpan barang di atas palet. Palet memiliki kompleksitas akan dimensi sehingga gudang memerlukan proses perancangan intalasi rak yang tepat. Proses penempatan barang di atas palet disebut dengan paletisasi, yang memiliki tujuan untuk mempermudah proses penanganan barang dalam gudang, mempermudah proses penghitungan barang, dan melindungi barang saat melalui proses transportasi. Secara umum, dalam menjaga kestabilan barang dilakukan proses

stretch wrapping, yaitu membungkus barang dengan menggunakan *plastic film.* Proses penyimpanan akan berlangsung dengan mudah dan mampu mencapai kestabilan palet apabila barang yang akan disimpan memiliki jenis kemasan yang sama, yang disebut dengan palet homogen (Morana, 2018).

ii. Peralatan dinamis (peralatan bergerak)

Alat utama yang dapat dikategorikan dalam peralatan dinamis atau bergerak, yaitu *forklift* dan *rotary storage. Forklift* memiliki cara kerja dengan mengangkat dan memindahkan barang dari satu tempat ke tempat lainnya. Selain itu, dalam pergudangan juga terdapat *hand pallet truck* (*pallet jack*) dan *stacker. Crane* menjadi salah satu alat yang masuk dalam kategori *rotary storage. Stacker crane* merupakan sebuah mesin yang mampu menyimpan palet dalam rak yang tinggi. Proses penyimpanan yang tinggi dapat dilakukan oleh sebuah mesin *stacker crane* yang memiliki tiang dengan tinggi melebihi 20 meter.

b. Jenis stok

Stok dapat diartikan sebagai sekumpulan barang atau bahan yang dimiliki oleh suatu perusahaan pada waktu tertentu. Suatu perusahaan menyimpan barang setengah jadi dan barang jadi dengan tujuan untuk menghemat waktu pengiriman (*time on delivery*). Adanya stok dalam gudang merupakan sebuah konsekuensi dari ketimpangan aliran antara sistem *push* dan *pull.* Hal ini mengartikan bahwa terhambatnya proses produksi dalam suatu perusahaan disebabkan oleh proses menunggu pengadaan bahan yang diperlukan. Oleh karena itu, adanya stok menjadi jaminan untuk kelancaran proses produksi.

Namun, keberadaan stok juga menimbulkan kendala. Kendala yang dimaksud adalah biaya sehingga perlu adanya optimalisasi untuk membatasi jumlah stok dan menerapkan aliran *just in time* dengan maksimal. Stok produk di dalam gudang memiliki beberapa istilah sebagai deskripsi suatu material, seperti *Stock Keeping Unit* (SKU) dan *part number* atau *product ID.* Stok dapat dikategorikan dalam beberapa jenis sebagai berikut.

i. *Nature stock*

Nature stock berfokus pada bentuk dan status stok dalam gudang, seperti *input stock, output stock, intermediate stock, consignment stock, advanced stock,* dan *deported stock. Input stock* dikategorikan untuk stok bahan baku sedangkan *output stock* dikategorikan untuk stok produk jadi. *Intermediate stock* merupakan stok barang setengah jadi atau barang WIP (*Work in Process*). *Consignment stock* dikategorikan untuk stok yang berada pada pelanggan atau disebut dengan stok konsinyasi, dalam hal ini perusahaan hanya melakukan pembayaran kepada pemasok ketika barang tersebut digunakan. *Advanced stock* merupakan kategori stok yang berada pada pemasok tetapi

hak milik tetap berada pada pelanggan. *Deported stock* dikategorikan untuk stok yang dimiliki oleh pemasok tetapi disimpan pada suatu lokasi tertentu yang dipercayakan oleh produsen kepada penyedia jasa.

ii. *Destination stock*

Pada *destination stock* terdapat dua jenis stok, yaitu *assigned stock* dan *common stock. Assigned stock* atau stok khusus merupakan stok dalam gudang yang dikhususkan untuk suatu produksi tertentu. *Common stock* merupakan produk yang disimpan dalam gudang dapat dipakai tanpa terkait dengan suatu produksi tertentu.

iii. *Quantity stock*

Berdasarkan perhitungan stok yang dilakukan pada gudang, stok terdiri dari tujuh kategori. *Tool stock* merupakan stok suku cadang yang tidak digunakan dalam proses produksi tetapi sebagai penunjang kelancaran operasional perusahaan. *Minimum stock* berarti jumlah minimum stok yang disimpan dalam gudang telah menyesuaikan waktu pengadaan dan waktu tunggu (*lead time*). *Safety stock* juga dikenal dengan sebutan *buffer stock* merupakan tingkat stok yang harus dipertahankan dalam gudang untuk menghindari terjadinya kekurangan stok.

Alert stock dikategorikan untuk gabungan antara *minimum stock* dan *safety stock. Physical stock* ialah stok yang secara fisik ada dalam tempat penyimpanan. *Actual stock* berarti total keseluruhan antara *physical stock* dengan pesanan kepada pemasok yang dikurangi dengan pesanan pelanggan yang telah dalam proses pengiriman. *Dormant* atau *dead stock* dikategorikan untuk stok dengan tingkat pemggunaan yang sangat rendah.

iv. *Stock rotation*

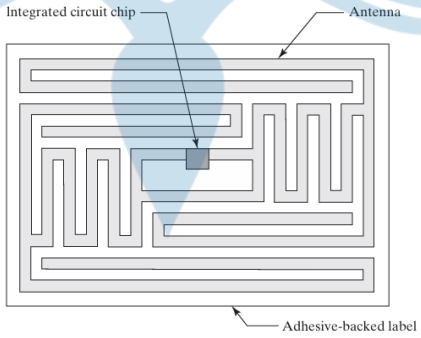
Terdapat dua jenis stok yang dapat dikategorikan dalam *stock rotation.* Jenis pertama adalah *so-called-non-zero rotation* yang merupakan produk tanpa tanggal kedaluwarsa. Oleh karena itu, produk dengan kategori tersebut dapat dijual kembali pada periode berikutnya. Sedangkan, *so-called zero rotation* merupakan produk dengan tanggal kedaluwarsa sehingga tidak dapat untuk dijual kembali.

2.2.2 *Radio Frequency Identification* (RFID)

Radio Frequency Identification (RFID) merupakan teknologi yang digunakan untuk mengumpulkan data secara nirkabel yang banyak digunakan pada sektor logistik, manufaktur, serta kontrol akses dan keamanan (Andía et al., 2018). Kemampuan RFID untuk diintegrasikan dengan teknologi pendukung seperti *Internet of Things* (IoT) dan *big data* juga menjadi keunggulan RFID dalam menciptakan sistem digital yang mampu mengefisiensi proses pengelolaan data. Penggunaan RFID telah banyak diterapkan

dalam identifikasi objek secara otomatis dalam jarak hanya beberapa meter bahkan centimeter, secara khusus pada penelusuran barang dalam jaringan rantai pasok (*supply chain*). Salah satu penggunaan teknologi ini dalam rantai pasok diterapkan pada pengelolaan gudang, terutama pada pengelolaan barang yang terbukti dapat meningkatkan efektivitas, efisiensi, *cost-effective*, serta meminimalkan kesalahan operasional yang berpotensi menyebabkan kerugian.

Sistem RFID mempunyai dua komponen, di antaranya *tag* yang berfungsi sebagai *transporder* dan *reader* sebagai *interrogator* (Groover, 2016). *Tag* sebagai *transporder* merupakan perangkat yang menerima sinyal dari *reader* dan merespon dengan mengirimkan sinyalnya sendiri. Sementara itu, *reader* sebagai *interrogator* akan memancarkan dan menerima respon sinyal dari komponen *tag*. *Tag* atau label identifikasi terdiri atas *Integrated Circuit* (IC) *Chip* untuk menyimpan data dan antena kecil untuk memancarkan frekuensi radio. Komponen *tag* pada RFID umumnya dikemas dalam plastik pelindung atau tertanam di dalam label berperekat yang kemudian ditempelkan pada objek, seperti *part*, produk, maupun kemasan (misalnya karton, kotak, dan palet). Desain *tag* RFID dibuat mengikuti standar *Electronic Product Code* (EPC) yang berfungsi sebagaimana *Universal Product Code* (UPC) pada *barcode*. Komponen *reader* pada RFID yang dapat berupa perangkat *portable* maupun alat tetap akan mengirimkan medan magnet frekuensi radio berdaya rendah untuk mengaktifkan *tag* ketika berada dalam jarak dekat. *Reader* kemudian memverifikasi sinyal yang dikirimkan oleh *tag* melalui gelombang radio sebelum akhirnya dikirim pada sistem komputer. Gambar 2.1 menunjukkan komponen *tag* pada RFID.



Gambar 2.1 Komponen *tag* RFID (Groover, 2016)

Pada sistem RFID terdapat dua jenis *tag,* yaitu *passive tag* dan *active tag. Passive tag* tidak mempunyai sumber daya internal tetapi memperoleh daya listrik untuk melakukan transmisi sinyal yang berasal dari gelombang radio. Adapun gelombang radio tersebut dihasilkan oleh *reader* ketika berada dalam jarak yang dekat. Berdasarkan segi ukuran dan biaya, *passive tag* memiliki ukuran yang lebih kecil dan murah. Namun, *passive tag*

juga memiliki umur pakai yang lama dengan jangkauan komunikasi radio yang lebih pendek. *Active tag* dilengkapi dengan baterai dan secara umum mempunyai kapasitas memori yang lebih besar. Penggunaan *active tag* juga dapat menjangkau komunikasi yang lebih panjang. Jika ditinjau berdasarkan biaya, penggunaan *active tag* cenderung lebih mahal dan digunakan pada barang yang bernilai tinggi. *Tag* yang terdapat dalam sistem RFID merupakan komponen *read-only* yang berisikan hingga 20 karakter data. Hal tersebut berfungsi untuk melakukan proses identifikasi suatu item dan mewakili informasi lain yang dikomunikasikan.

Beberapa aplikasi RFID dalam industri, yakni dalam manajemen persediaan, manajemen rantai pasok, sistem pelacakan, *warehouse control*, identifikasi lokasi, dan pelacakan barang *Work in Process* (WIP). Keuntungan penggunaan RFID adalah proses identifikasi oleh *reader* yang tidak bergantung pada kontak fisik dan tidak perlu dilihat langsung, serta kapasitas data yang lebih besar dibanding teknologi *Automatic Identification and Data Capture* (AIDC) yang lain seperti *barcode* dan biometrik. Adapun kelebihan lain dari penggunaan RFID adalah data pada *tag* yang dapat diperbarui sehingga *tag* yang dapat digunakan kembali, serta *tag* yang juga mencatat riwayat penggunaan. Meski begitu, instalasi RFID dengan label dan perangkat keras membutuhkan biaya yang lebih besar dibanding teknologi AIDC lain.

2.2.3 *K-Means Clustering*

K-Means clustering merupakan algoritma pengelompokan data dengan meminimalkan *square-error criterion,* yaitu total jarak kuadrat antar setiap data dengan *centroid* (pusat *cluster*) yang ditempati data tersebut. Algoritma dimulai dengan pembagian awal secara acak dan secara terus menerus mengalokasikan ulang data ke *cluster* berdasarkan kesamaan antar data dan *cluster*, hingga *square error criterion* minimum terpenuhi. Algoritma *K-Means* diawali dengan pembentukan partisi awal sejumlah k *cluster* yang ditentukan oleh pengguna, kemudian *centroid* awal setiap *cluster* dihitung berdasarkan data yang tergabung di dalamnya. Setiap data selanjutnya dialokasikan ke *cluster* dengan centroid terdekat berdasarkan ukuran jarak tertentu, umumnya jarak Euclidean. Setelah proses alokasi ulang dilakukan, *centroid cluster* dihitung kembali dan tahapan ini diulang hingga tidak terjadi perpindahan data antar *cluster* atau perbaikan nilai fungsi kriteria menjadi sangat kecil (Kantardzic, 2019).

Pemilihan jumlah *cluster* (k) pada metode *K-Means* dapat didasarkan pada pertimbangan eksternal, seperti pengetahuan awal terhadap data atau batasan praktis yang ada, maupun melalui pengujian beberapa nilai k. Apabila tersedia informasi awal mengenai pembagian data atau posisi *centroid*, informasi tersebut dapat dimanfaatkan

sebagai inisialisasi *cluster*. Oleh karena jumlah *cluster* dalam data umumnya tidak diketahui secara pasti, nilai k biasanya diuji dalam beberapa variasi untuk mengamati perubahan jarak dalam klaster seiring dengan peningkatan k, dengan karakteristik bahwa *cluster* yang dihasilkan pada nilai k yang berbeda bersifat saling terpisah (Shmueli et al., 2019).

Cluster yang terbentuk tentunya didapatkan dari proses untuk menentukan dan mengevaluasi data dari setiap *cluster.* Oleh karena itu, diperlukan suatu konsep untuk mengukur keterkaitan antar data atau sampel data dan penentuannya untuk dikelompokkan menjadi *cluster* yang terukur. Konsep pengukuran jarak dan evaluasi kualitas *cluster* tersebut menggunakan perumusan matematis sebagai dasar kuantitatif dalam proses *clustering*. Perumusan matematis untuk *K-Means Clustering* antara lain *euclidean distance, silhouette method*, dan *inertia* atau *within-sum of squared error* (SSE).

Euclidean distance digunakan untuk mengukur jarak rata-rata antar sampel dalam suatu dataset. *Silhouette method* digunakan sebagai pengukuran evaluasi hasil *clustering* dengan nilai koefisien yang ditentukan untuk setiap sampel dalam kumpulan data yang telah dikelompokkan dalam *cluster*. Berdasarkan Kantardzic (2019), Nilai koefisien *silhouette* yang dinilai baik adalah nilai yang lebih besar dari 0,5 dari rata-rata nilai *silhouette* semua sampel. Berdasarkan Billard & Diday (2019), *inertia* atau *within*-SSE merupakan pengukuran kekompakan *cluster* menghitung kuadrat jarak antara setiap titik data. Fungsi ini dapat digunakan untuk mengukur tingkat kekompakan *cluster* dengan tujuan dapat mengurangi variasi dalam sebuah cluster.

Perumusan *euclidean distance* pada Persamaan 2.1, *silhouette coefficient* pada Persamaan 2.2 dan *global silhouette score* pada Persamaan 2.3 mengacu pada Kantardzic (2019). Adapun perumusan *inertia* atau *within* SSE pada Persamaan 2.4 dan *total inertia* pada Persamaan 2.5 sebagai ukuran kekompakan *cluster* mengacu pada Billard & Diday (2019).

$$D_{ij} = \sqrt{\left[\sum_{k=1}^{n}\left(\frac{x_{ik} - x_{jk}}{max_k - min_k}\right)^2\right]} \qquad (2.1)$$

Keterangan

D_{ij} = Jarak antara data ke-i dan data ke-j

n = Jumlah total atribut data

k = indeks atribut ke-k

x_{ik} = Nilai atribut ke-k pada data i

x_{jk} = Nilai atribut ke-k pada data j

max_k = Nilai maksimum atribut ke-k pada seluruh data

min_k = Nilai minimum atribut ke-k pada seluruh data

$$s_i = \frac{(b_i - a_i)}{\max\ (b_i, a_i)} \tag{2.2}$$

Keterangan:

s_i = Nilai koefisien *silhouette* untuk data ke-i

a_i = Rata-rata jarak data ke-i dengan *centroid* (pusat *cluster*)

b_i = Rata-rata jarak data ke-i dengan *centroid* (pusat *cluster*) terdekat lainnya

$$S = \frac{1}{N} \times \sum_{i=1}^{N} s_i \tag{2.3}$$

Keterangan:

S = Nilai rata-rata koefisien *silhouette* dari keseluruhan dataset

N = Jumlah total seluruh data dalam dataset

s_i = Nilai koefisien *silhouette* untuk data ke-i

$$E_k = \sum_{j=1}^{n} \sum_{i=1}^{m_k} \left(x_{ij} - \bar{x}_{kj}\right)^2 \tag{2.4}$$

Keterangan:

E_k = Nilai *inertia* atau *within*-(SSE) untuk *cluster* ke-k

n = Jumlah total atribut data

m_k = Jumlah total data di dalam *cluster* ke-k

x_{ij} = Nilai data ke-i pada atribut ke-j

$\bar{x}_{kj}$ = Nilai rata-rata *cluster* ke-k untuk atribut ke-j

$$E = \sum_{k=1}^{K} E_k \tag{2.5}$$

Keterangan:

E = Nilai *inertia* atau *within*-SSE total

K = Total *cluster* terbentuk

E_k = Nilai *inertia* atau *within*-(SSE) untuk *cluster* ke-k

2.2.4 Python

Python merupakan sebuah bahasa pemrograman yang menggabungkan antara fitur-fitur C dan Java. Terdapat beberapa fitur penting yang terdapat dalam Python yang menjadi alasan Python menjadi bahasa pemrograman yang memiliki popularitas yang

baik. Salah satu fitur penting dalam Python adalah keringkasan bahasa pemrograman yang memudahkan pemahaman pengguna. Hal tersebut juga menjadikan Python mudah dipelajari karena menggunakan struktur bahasa yang sederhana.

Secara penggunaan, *software* Python tidak memiliki fitur berbayar atau disebut sebagai *open-source software* sehingga modifikasi kode program dapat dilakukan sesuai keinginan pengguna. Fitur *high level language* yang tersedia dapat membantu pengguna dalam mempelajari intruksi kode untuk menjalankan program. *Trigger error* yang muncul pada fitur penting Python sebagai bahasa pemrograman yang memiliki tipe dinamis. Adanya fitur tersebut, menjadikan acuan pengguna untuk melakukan evaluasi kode program, seperti *variable names* dan *datatypes* yang tidak teridentifikasi. Program Python juga tidak bergantung pada sistem operasi tertentu dan dapat digunakan pada hampir semua sistem operasi, seperti UNIX, Linux dan Windows (Rao, 2017).

Python memiliki modul dan *library* yang dapat digunakan. Modul dalam Python merupakan *file* yang berisi objek-objek berguna seperti fungsi, *class*, ataupun variabel yang merepresentasikan program untuk melakukan tugas tertentu. Sedangkan *library* Python berisi beberapa aplikasi kecil (atau paket kecil) yang telah dikembangkan, dapat digunakan, dan dapat dipelihara dengan mudah.

Salah satu modul Python adalah *math* yang berguna untuk mengakses fungsi-fungsi operasi matematika. Kemudian terdapat modul *Operating System* (OS) untuk mengetahui apakah suatu *file* atau dokumen ada atau tidak, dengan submodul bernama *path* yang berisi metode isfile(). Modul ini dapat digunakan untuk mengetahui apakah *file* yang akan dibuka benar-benar ada atau tidak. Misalnya, *os.path.isfile(fname)* akan mengembalikan nilai *True* jika file tersebut ada, tetapi akan mengembalikan nilai *False* jika tidak ada (Rao, 2017).

Selain modul, Python juga memiliki *library* yang dapat dapat digunakan, salah satunya adalah *numpy*. *Numpy* (*numeric python)* adalah *library* untuk memproses kumpulan data tunggal atau multidimensi. *Numpy* meliputi berbagai kelas, fungsi, variabel, dan lain-lain untuk mengoperasikan perhitungan ilmiah dalam Python. Selain itu, *numpy* juga meliputi berbagai fungsi matematika, contohnya fungsi aljabar linear (Rao, 2017).

Library selanjutnya adalah *pandas* yang menyediakan struktur data untuk analisis data, seri waktu *(time series*)*,* dan statistik. Pandas berguna untuk analisis dan manipulasi data, yaitu DataFrame yang merupakan objek utama dalam *library* ini (Rao, 2017). *Library* ini mendukung pemuatan data dalam berbagai format *file*, umumnya dalam format .csv atau .xlsx (Shmueli et al., 2019). Selain itu terdapat scikit-Learn yang merupakan *library open-source* Python yang menyediakan rangkaian alat lengkap untuk

analisis data. Analisis data yang dimaksud meliputi transformasi data, pengukuran performansi, memodelkan data, *clustering*, dan sebagainya (Shmueli et al., 2019).

Python menyediakan berbagai *framework* sebagai alat bantu pembuatan aplikasi dengan struktur dan *tools* siap pakai yang mempermudah dan mempercepat *developer*. Pembuatan aplikasi tanpa menggunakan *framework* akan memakan waktu lebih lama dengan penggunaan bahasa pemrograman *native* yang lebih kompleks. Komponen yang umumnya dimiliki oleh *framework* adalah *library*, struktur folder, *tools* debugging, *tools* testing, serta fitur keamanan. Kivy menjadi salah satu *framework open-source* yang disediakan Python untuk membuat *Graphical User Interface* (GUI) dan aplikasi *cross-platform* yang mendukung tahapan *prototyping* yang cepat.

Beberapa hal yang menjadi keunggulan Kivy, yakni dapat dijalankan pada *desktop* dan *mobile* (Android dan iOS) atau bersifat *cross-platform*, mendukung *rapid development*, bersifat *open-source* dan mendukung pengembangan gratis, serta *multi-touch* atau dapat memproses lebih dari satu sentuhan (*touch*) pada layar. Fitur *multi-touch* yang disediakan menjadikan aplikasi lebih interaktif, meningkatkan *user experience*, dan mempermudah navigasi. Komponen utama pada Kivy adalah *widget* yang menyediakan elemen UI, *layout* yang mengatur posisi *widget*, dan KV *language* yang merupakan bahasa khusus untuk mendesain UI. Pengembangan aplikasi menggunakan Kivy biasanya dikombinasikan dengan Buildozer, *Software Development Kit* (SDK), dan *Native Development Kit* (NDK) untuk menghasilkan aplikasi yang dapat diakses pada perangkat berbasis Android.

2.2.5 Pemodelan Sistem Berbasis Objek (*Object Oriented System Design*)

Pemodelan Sistem Berbasis Objek (PSBO) merupakan pendekatan perancangan sistem yang memodelkan sistem sebagai kumpulan objek yang saling berinteraksi. Setiap objek merepresentasikan entitas nyata dalam sistem serta menggabungkan atribut dan fungsi yang dimilikinya, sehingga hubungan antara struktur dan aktivitas sistem dapat digambarkan secara terintegrasi. Pendekatan ini memungkinkan sistem yang kompleks diuraikan menjadi komponen-komponen modular yang lebih mudah dianalisis, dirancang, dan dikelola. Dengan struktur yang modular, pengembangan sistem menjadi lebih sistematis, mudah dipahami, serta mendukung konsistensi rancangan antar subsistem. Selain itu, PSBO memudahkan proses dokumentasi dan komunikasi rancangan sistem melalui representasi visual seperti diagram untuk memastikan kesesuaian antara rancangan dan kebutuhan sistem.

PSBO memiliki tiga karakteristik utama, yaitu berorientasi pada *use case*, berpusat pada arsitektur, serta bersifat iteratif dan bertahap. Berorientasi pada *use case* berarti *use*

case digunakan sebagai alat utama untuk mendefinisikan perilaku sistem dan menggambarkan interaksi pengguna dengan sistem secara sederhana dan terfokus. Berpusat pada arsitektur menekankan bahwa pengembangan sistem didasarkan pada pandangan arsitektur yang saling terkait, meliputi pandangan fungsional, struktural, dan perilaku. Sementara itu, sifat iteratif dan bertahap memungkinkan pengembangan sistem dilakukan secara bertahap melalui proses pengujian dan penyempurnaan berkelanjungan hingga diperoleh representasi sistem yang semakin lengkap dan sesuai dengan kebutuhan pengguna (Dennis et al., 2020).

2.2.6 *Unified Modeling Language*

Unified Modeling Language (UML) merupakan alat bantu untuk melakukan proses pemodelan visual dalam membangun sistem informasi yang terstruktur. Penggunaan UML dapat melalui berbagai jenis, antara lain *use case diagram, activity diagram*, dan *sequence diagram*. Berikut adalah uraian ketiga diagram UML menurut Dennis et al. (2020).

a. *Use Case Diagram*

Use case diagram menggambarkan hubungan antara pengguna dan fungsi sistem. *Use case* digunakan untuk memahami fungsi sistem secara menyeluruh pada tingkat yang tinggi serta menyajikan gambaran sederhana dan jelas mengenai apa yang akan dilakukan oleh sistem kepada pengguna. *Use case* menggambarkan interaksi antara sistem dengan para aktor yang terlibat dan merepresentasikan pandangan fungsional sistem dari perspektif pengguna. Elemen pada *use case diagram* dapat dilihat pada Gambar 2.2.

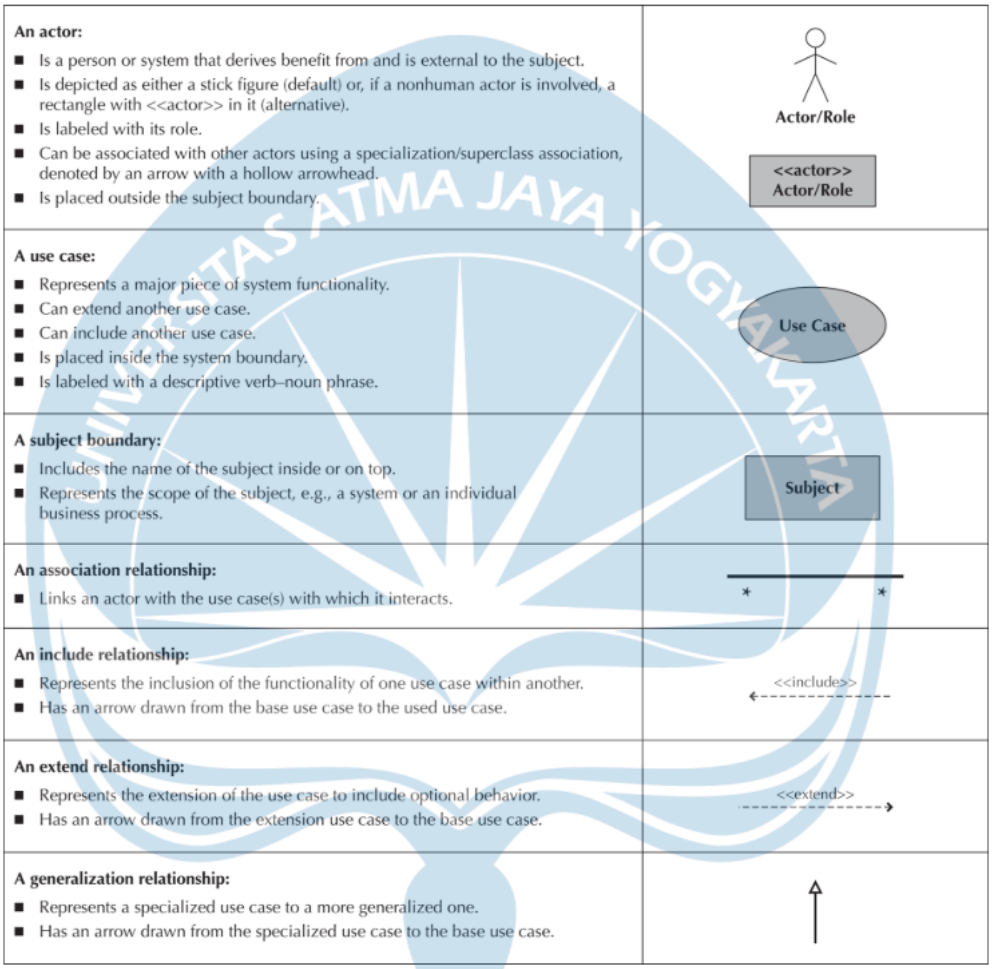
b. *Activity Diagram*

Activity diagram berfokus pada pemetaan alur proses kerja yang terjadi di sistem. Diagram ini memberikan gambaran visual terkait urutan kegiatan sistem. Diagram ini memodelkan perilaku suatu proses bisnis dari objek yang terlibat. Diagram ini dapat merepresentasikan berbagai tingkat detail, mulai dari alur kerja bisnis tingkat tinggi yang mencakup banyak *use case*, hingga rincian dari satu *use case* atau bahkan metode tertentu. Elemen pada *activity diagram* dapat dilihat pada Gambar 2.3.

c. *Sequence diagram*

Sequence diagram digunakan untuk memodelkan interaksi antara aktor dan objek yang berkolaborasi dalam satu kasus penggunaan, dengan menampilkan aliran informasi yang dipertukarkan secara berurutan seiring waktu. Diagram ini merupakan model dinamis yang menunjukkan urutan aliran informasi antar objek dalam suatu interaksi yang telah didefinisikan dengan elemen seperti pada Gambar 2.4. Secara umum,

sequence diagram dapat disusun untuk merepresentasikan seluruh kemungkinan skenario dalam satu kasus penggunaan, namun *sequence diagram* dapat dibuat dengan beberapa diagram singkat, masing-masing menggambarkan satu skenario spesifik. Diagram ini dimanfaatkan sepanjang tahap analisis dan desain, umumnya bersifat lebih spesifik terhadap implementasi dengan melibatkan objek basis data atau komponen *interface* pengguna tertentu.



Gambar 2.2 Elemen *use case diagram* (Dennis et al., 2020)

An action: ■ Is a simple, nondecomposable piece of behavior. ■ Is labeled by its name.	
An activity: ■ Is used to represent a set of actions. ■ Is labeled by its name.	
An object node: ■ Is used to represent an object that is connected to a set of object flows. ■ Is labeled by its class name.	
A control flow: ■ Shows the sequence of execution.	
An object flow: ■ Shows the flow of an object from one activity (or action) to another activity (or action).	
An initial node: ■ Portrays the beginning of a set of actions or activities.	
A final-activity node: ■ Is used to stop all control flows and object flows in an activity (or action).	
A final-flow node: ■ Is used to stop a specific control flow or object flow.	
A decision node: ■ Is used to represent a test condition to ensure that the control flow or object flow only goes down one path. ■ Is labeled with the decision criteria to continue down the specific path.	
A merge node: ■ Is used to bring back together different decision paths that were created using a decision node.	
A fork node: Is used to split behavior into a set of parallel or concurrent flows of activities (or actions)	
A join node: Is used to bring back together a set of parallel or concurrent flows of activities (or actions)	
A swimlane: Is used to break up an activity diagram into rows and columns to assign the individual activities (or actions) to the individuals or objects that are responsible for executing the activity (or action) Is labeled with the name of the individual or object responsible	

Gambar 2.3 Elemen *activity diagram* (Dennis et al., 2020)

UNIVERSITAS ATMA JAYA YOGYAKARTA

Term and Definition	Symbol
An actor: ■ Is a person or system that derives benefit from and is external to the system. ■ Participates in a sequence by sending and/or receiving messages. ■ Is placed across the top of the diagram. ■ Is depicted either as a stick figure (default) or, if a nonhuman actor is involved, as a rectangle with <<actor>> in it (alternative).	
An object: ■ Participates in a sequence by sending and/or receiving messages. ■ Is placed across the top of the diagram.	
A lifeline: ■ Denotes the life of an object during a sequence. ■ Contains an X at the point at which the class no longer interacts.	
An execution occurrence: ■ Is a long narrow rectangle placed atop a lifeline. ■ Denotes when an object is sending or receiving messages.	
A message: ■ Conveys information from one object to another one. ■ A operation call is labeled with the message being sent and a solid arrow, whereas a return is labeled with the value being returned and shown as a dashed arrow.	
A guard condition: ■ Represents a test that must be met for the message to be sent.	
For object destruction: ■ An X is placed at the end of an object's lifeline to show that it is going out of existence.	X
A frame: ■ Indicates the context of the sequence diagram.	

Gambar 2.4 Elemen *sequence diagram* (Dennis et al., 2020)

2.2.7 *Rapid Application Development – Prototyping*

Rapid Application Development (RAD) merupakan teknik *System Development Life Cycle* (SDLC) dengan proses interaktif sepanjang siklus sistem hingga sistem sepenuhnya dikembangkan hingga pengguna puas dengan hasil sistem. RAD bergantung secara signifikan pada keterlibatan pengguna akhir. RAD fokus pada iterasi dan koreksi sistem yang dibuat, sehingga memungkinkan sistem untuk dikembangkan lebih cepat dengan efisiensi sumber daya. RAD meliputi proses *prototyping*, yaitu teknik dalam pengembangan sistem untuk membuat suatu model sistem atau bagian dari suatu sistem untuk didemonstrasikan kepada pengguna untuk diuji coba (Turban et al., 2025).

Prototyping merupakan metode pengembangan sistem dengan melakukan tahap analisis, perancangan, dan implementasi secara bersamaan, dan ketiga tahap tersebut

dilakukan secara berulang dalam siklus hingga sistem selesai. Proses dimulai dengan pengembangan prototipe awal yang memiliki fungsi minimal, kemudian ditinjau oleh pengguna atau *user.* Hal ini dilakukan untuk memperoleh umpan balik yang digunakan sebagai dasar perbaikan pada siklus berikutnya hingga sistem memenuhi kebutuhan operasional. Metode ini mampu menyediakan gambaran sistem secara cepat dan meningkatkan keterlibatan pengguna sehingga kebutuhan sistem dapat teridentifikasi lebih awal. Akan tetapi, metode ini juga memiliki keterbatasan apabila analisis yang dilakukan kurang mendalam pada tahap awal, berisiko menyebabkan perubahan desain signifikan menimbulkan permasalahan struktural pada pengembangan sistem yang kompleks (Dennis et al., 2020).

2.2.8 Supabase

Supabase adalah sebuah platform *Backend-as-a-Service* (BaaS) yang menyediakan layanan pengelolaan *database* bersifat *open-source* (Kolesnikov et al., 2024). BaaS merupakan layanan *cloud* yang menyediakan infrastruktur *backend* yang memungkinkan *developer* untuk membuat aplikasi tanpa membuat dan mengelola server *backend* sendiri. Kelebihan penggunaan BaaS, yaitu *developer* dapat berfokus pada *frontend* dan logika aplikasi saja. Beberapa fitur yang dimiliki Supabase di antaranya pengelolaan *database* PostgreSQL, autentikasi, serta *Auto Generated* API (Pathan et al., 2025). Supabase mampu meningkatkan efisiensi proses serta mempercepat *prototyping* karena tidak memerlukan pengembangan *backend* yang rumit.

Pengembangan aplikasi menggunakan Supabase dapat dilakukan menggunakan fitur *ready-to-use* yang disediakan oleh Supabase. Salah satu fitur *ready-to-use* yang dimiliki adalah *Authentication* dengan berbagai konfigurasi, di antaranya dengan *login* dan *password*, maupun dengan penyedia layanan pihak ketiga seperti Google dan GitHub. Di samping itu, sistem otorisasi yang digunakan, yakni PostgreSQL *Row Level Security* (RLS) menjadikan administrator sebagai penentu akses pengguna terhadap data dengan penerapan kebijakan (*policy*). Kebijakan dimaksud tidak hanya dapat diterapkan pada tabel maupun kolom melainkan juga terhadap level setiap *row* (baris).

2.2.9 Google Colaboratory

Google Colaboratory atau Google Colab merupakan produk buatan Google Research yang mendukung pembuatan dan eksekusi program Python yang ditulis pada Google Drive (Migliaccio & Iannone, 2023). Google Colab menyediakan layanan Jupyter Notebook yang di-*hosting* dan dapat dijalankan menggunakan *browser*, di antaranya Google Chrome dan Mozilla Firefox. Penggunaan Google Colab memungkinkan

pengguna mengakses notebook yang sama, namun *resource* yang dapat digunakan bersifat terbatas dan batas penggunaannya fluktuatif. Notebook pada Google Colab adalah *file* yang berisi kumpulan sel untuk menggabungkan dan menjalankan kode Python dengan format ".ipynb".

Sebagai sarana pembuat aplikasi (*application development platform*), Google Colab menyediakan lingkungan pemrograman Python berbasis *cloud* yang mempermudah *developer* untuk membangun program tanpa instalasi *software* pada perangkat komputer. Dengan tambahan *tools* Buildozer, Google Colab dapat digunakan untuk membangun aplikasi Android menggunakan Kivy dengan melakukan proses *build* aplikasi Python. Buildozer merupakan perangkat lunak yang digunakan dalam proses kompilasi kode Python menjadi aplikasi perangkat seluler, secara khusus pada Android dan iOS. Dalam hal ini, Google Colab berperan sebagai komputer *virtual* Linux bagi Buildozer yang membutuhkan sistem operasi Linux untuk dijalankan.

2.2.10 Microsoft Excel

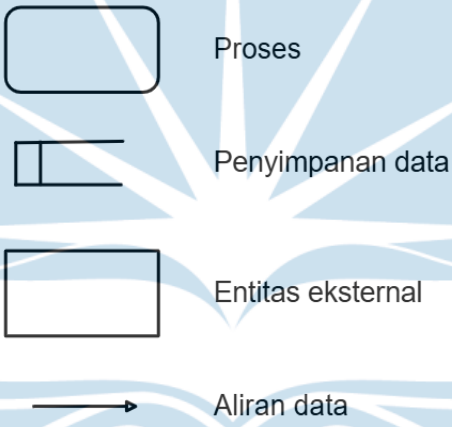
Microsoft Excel merupakan aplikasi *spreadsheet* buatan Microsoft Office untuk mengolah data numerikal maupun non-numerikal (Alexander & Kusleika, 2025). Pada satu *file* Excel yang disebut juga dengan *workbook*, terdapat minimal satu *worksheet* yang terdiri atas kumpulan sel dan dapat diisi dengan angka, formula, atau teks. Gabungan dari angka, formula, dan teks dimaksud dapat kemudian digunakan untuk melakukan perhitungan otomatis, seperti formula 'Average' yang digunakan untuk menghitung rerata atau formula 'If' untuk menghasilkan nilai tertentu melalui pengkondisian logika. Perhitungan otomatis yang dimiliki Excel kemudian dapat memimalkan waktu pada tahapan awal pengolahan data (*pre-processing data*), serta meningkatkan akurasi hasil pemrosesan data.

Adapun fitur-fitur pada Excel mendukung proses analisis data dengan adanya *sorting tools* untuk mengurutkan data dan *filtering tools* untuk menyaring data (Ringstrom, 2025). Proses pengurutan data dapat dilakukan secara *ascending* jika urutan menaik dan *descending* jika urutan menurun. Urutan dapat diatur pada tipe data numerikal dilihat dari besar/kecilnya angka, maupun non-numerikal dilihat dari huruf pertama pada isi sel dimaksud. Di samping itu, fitur penyaring data dapat digunakan untuk menampilkan data tertentu sesuai pengkondisian yang dilakukan atau syarat yang ditetapkan, misalnya untuk nilai, rentang angka, maupun teks tertentu. Terdapat beberapa jenis *filtering* pada Excel, di antaranya *auto filter* sebagai penyaring cepat menggunakan *dropdown* di setiap kolom, *custom filter* sebagai penyaring dengan penyesuaian satu kondisi tertentu, dan

advanced filter sebagai penyaring data kompleks dengan penggunaan *range* kriteria khusus.

2.2.11 *Data Flow Diagram*

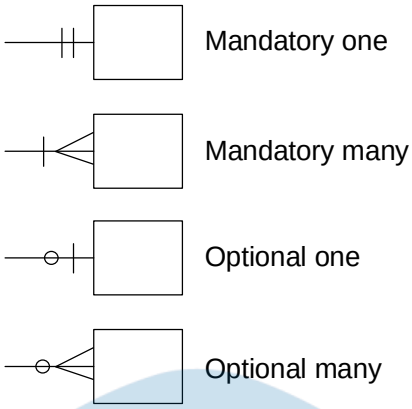
Data Flow Diagram (DFD) merupakan diagram yang digunakan untuk mengidentifikasi aliran data dalam sistem informasi, di antaranya aliran data masuk, proses pengolahan, serta output yang dihasilkan (Nugraha et al., 2026). Adapun 4 entitas utama dalam DFD, yakni proses yang berfungsi mengolah data, aliran data antar komponen, penyimpanan data, serta entitas eksternal yang berinteraksi dengan sistem. Beberapa hal yang perlu diperhatikan dalam perancangan DFD berkaitan dengan aliran data dalam sistem, yakni antara entitas luar tidak diijinkan terjadi relasi, tidak boleh terdapat hubungan antara entitas luar dengan penyimpanan data, serta pada satu aliran data diperkenankan terdapat aliran beberapa struktur data sekaligus. Gambar 2.5 menunjukkan elemen DFD menurut .



Gambar 2.5 Simbol elemen DFD (Kendall & Kendall, 2020)

2.2.12 *Entity Relationship Diagram*

Entity Relationship Diagram (ERD) merupakan diagram yang digunakan untuk memodelkan struktur data dan menggambarkan hubungan data dalam sistem informasi (Afiifah et al., 2022). Elemen dasar penyusun ERD yakni entitas, atribut, dan relasi dapat dilihat pada Gambar 2.6. Entitas merupakan objek pada sistem yang dapat dibedakan dari objek lain, sementara atribut merupakan sebuah karakteristik pada entitas yang membedakannya dengan entitas lain. Adapun relasi merupakan hubungan yang terjadi antara dua entitas atau lebih. Kardinalitas menunjukkan jumlah entitas yang dapat berhubungan dengan entitas lain dalam satu relasi serta dapat bersifat *mandatory* (wajib) dan *optional* (opsional).



Gambar 2.6 Kardinalitas hubungan pada ERD (Hoffer et al., 2019)

2.2.13 Legibilitas Warna

Perancangan *user interface* dilakukan dengan memperhatikan kombinasi warna untuk memudahkan pengguna dalam membaca dan memahami informasi yang ditampilkan. Legibilitas kombinasi warna digunakan dalam perancangan *user inteface* berkaitan dengan kontras antara warna teks dan latar belakang (Jiménez et al., 2020). Sebagai contoh, penggunaan teks berwarna hitam dan latar belakang putih memiliki legibilitas kombinasi warna yang tinggi karena kontras yang jelas, sementara kombinasi teks kuning dengan latar putih memiliki legibilitas warna rendah karena sulit dibaca. Tabel 2.4 menunjukkan legibilitas kombinasi warna.

Tabel 2.4 Legibilitas kombinasi warna

Legibilitas	Kombinasi Warna	
	Karakter	***Background***
Sangat Bagus	Hitam	Putih
	Hitam	Kuning
Bagus	Kuning	Hitam
	Putih	Hitam
	Biru Gelap	Putih
Bagus	Hijau	Putih
Sedang	Merah	Putih
	Merah	Kuning
Buruk	Hijau	Merah
	Merah	Hijau
	Oranye	Hitam
	Oranye	Putih
Sangat Buruk	Hitam	Biru
	Kuning	Putih

2.3 Standar dan Kode Etik

Proses perancangan dilaksanakan dengan mengacu pada standar dan kode etik yang berlaku untuk memastikan setiap tahapan perancangan dapat menjaga keselamatan dan keamanan. Berikut merupakan standar dan kode etik yang digunakan.

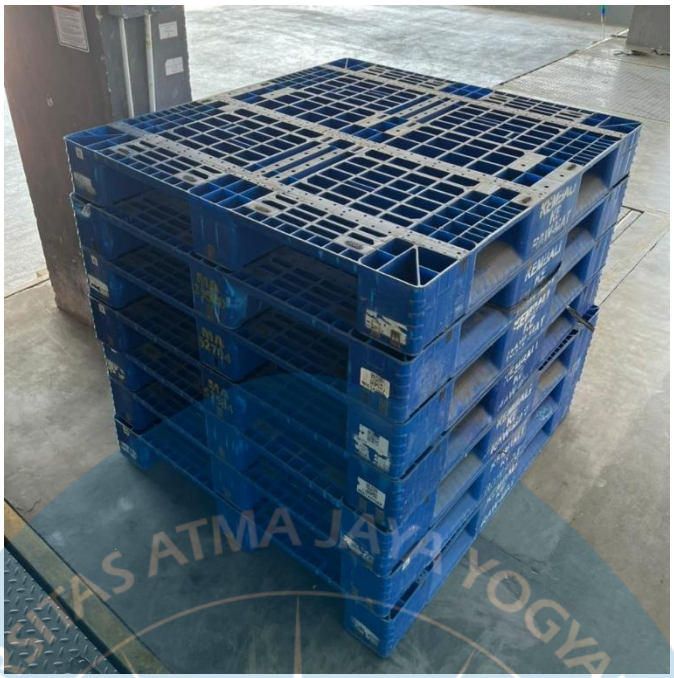
a. Data BIN *Raw Material*_061125_R1 (Standar penyusunan palet Gudang *Raw Material* pada PT XYZ)

Gudang *Raw Material* pada PT XYZ memiliki aturan dalam peletakan *bin* penyimpanan. Rak penyimpanan pada Gudang *Raw Material* memiliki batas ketinggian yang bervariasi untuk *bin.* Adapun batas ketinggian bin pada rak penyimpanan Gudang *Raw Material* tersimpan dalam dokumen atau *file* Data BIN *Raw Material*_061125_R1, dengan isi seperti yang ditunjukkan pada Gambar 2.7.

SLOC	LAJUR	JUMLAH LAJUR	TOTAL BIN	LEVEL / TINGKAT	TINGGI PER LEVEL (METER)	KETERANGAN
82i1	AA & AB	2	1344	8	1,3	Level 1 s. d. 8
82i1	AC s. d. AP	14	8736	8	1,3	Level 1 s. d. 8
82i3	AQ, AR s. d. BF	16	9984	8	1,3	Level 1 s. d. 8
82i5	BG s. d. BL	6	3744	8	1,3	Level 1 s. d. 8
82i7	BM s. d. BV	10	5310	6 & 7	1,5 & 2	All Level 1_(SBIN 1 s. d. 81) = 2 M
						Level 2,3_(SBIN 1 s. d. 15) = 2 M
						Level 4,5,6_(SBIN 1 s. d. 15) = 1,5 M
						Level 2,3,4,5,6,7_(SBIN 16 s. d. 81) = 1,5 M
82i8	BW s. d. BX	2	1092	7	1,5	Level 1 s. d. 7
82i9	BY s. d. CP	18	7020	5	2,5	Level 1 s. d. 5
82i9	CQ s. d. CT	4	1776	5 & 6	1,7 & 2,5	Level 1 s. d. 6_BIN CQ & CR_(SBIN 1 s. d. 39) = 1,7 M
						Level 1 s. d. 6_BIN CS & CT_(SBIN 1 s. d. 39) = 1,7 M
						Level 1 s. d. 5_BIN CQ & CR_(SBIN 43 s. d. 81) = 2,5 M
						Level 1 s. d. 5_BIN CS & CT_(SBIN 40 s. d. 84) = 2,5 M (MQP)

Gambar 2.7 Ketinggian level *bin* Gudang *Raw Material*

Berdasarkan standar ketinggian level *bin* pada Gudang *Raw Material*, maksimal tinggi 1,3 meter atau 130 cm akan menjadi salah satu standar yang digunakan untuk perancangan usulan. Selain itu, terdapat aturan bahwa *colly* yang ditempatkan pada rak tidak diperkenankan *overhang* atau melebihi luasan palet. Adapun ukuran luas palet yang digunakan di Gudang *Raw Material* sebesar 130 cm × 120 cm, sehingga penyusunan *colly* harus diperhatikan. Dokumentasi palet standar yang digunakan Gudang *Raw Material* dapat dilihat pada Gambar 2.8.



Gambar 2.8 Palet Standar 130 cm × 120 cm Gudang *Raw Material*

b. ISO/IEC 18000

Standar ini merupakan standar internasional yang diterbitkan oleh *International Organization for Standardization* (ISO). Standar ISO/IEC 18000 digunakan sebagai pendefinisian *communication protocols* yang umum untuk berbagai frekuensi dalam *Radio Frequency Identification* (RFID). Adanya standar ini memiliki tujuan untuk memastikan kemampuan perangkat RFID dapat dioperasikan dengan baik meskipun memiliki frekuensi yang berbeda. Proses manajemen dan pengendalian sistem untuk melakukan pertukaran informasi yang lebih umum juga terdapat dalam standar ini.

ISO/IEC 18000 klausul 6 menjelaskan terkait persyaratan fisik dan logis dalam sistem RFID pasif. Sistem tersebut terdiri dari 2 perangkat, yaitu *interrogator* (*reader*) sebagai perangkat yang membaca *tag* dan *tag (label*) sebagai perangkat yang menyimpan informasi melalui antenanya. Pada standar tersebut terdapat dua metode komunikasi, yaitu *Interrogator-Talks-First* (ITF) dan *Tag-Only-Talks-After-Listening* (TOTAL). *Tag* yang terdapat dalam metode komunikasi ITF akan memberi respons setelah diperintahkan oleh *interrogator.* Namun, *tag* yang terdapat dalam metode komunikasi TOTAL akan memberi respons apabila telah masuk area *interrogator* dan setelah menerima informasi sistem ITF.

c. Peraturan Direktur Jenderal Pos dan Telekomunikasi Nomor 221/DIRJEN/2007

Peraturan ini mengatur Persyaratan Teknis Alat dan Perangkat RFID Reader yang beroperasi pada pita frekuensi 923-925 MHz di Indonesia, menetapkan standar teknis agar perangkat ini dapat digunakan secara efektif dan tidak mengganggu spektrum

frekuensi lain, serta mengacu pada standar internasional. Aturan ini berlaku untuk semua alat dan perangkat RFID Reader yang menggunakan frekuensi 923-925 MHz di Indonesia dan mencakup spesifikasi teknis seperti daya pancar, karakteristik emisi, modulasi, dan parameter lain yang relevan untuk memastikan operasi yang aman dan sesuai.

d. ISO 9001:2015

ISO 9001:2015 merupakan standar internasional yang mengatur sistem manajemen mutu dengan menekankan pendekatan proses yang merujuk pada klausul 4 dan pemikiran berbasis risiko yang dijabarkan dalam klausul 6. Standar ini mendorong organisasi untuk memahami alur proses, kebutuhan pemangku kepentingan, serta potensi risiko yang dapat memengaruhi kinerja sistem. Berhubungan dengan konteks perancangan sistem informasi, ISO 9001:2015 klausul 7.5 digunakan sebagai kerangka konseptual untuk memastikan proses perancangan dilakukan secara terstruktur dan terdokumentasi dengan baik. Penerapan prinsip ISO 9001:2015 klausul 5 dan klausul 8 juga mendukung konsistensi perancangan melalui penetapan prosedur, pembagian tanggung jawab, dan mekanisme pengendalian perubahan. Oleh karena itu, standar ini berperan sebagai landasan mutu dalam perancangan sistem, tanpa menitikberatkan pada aspek teknis pengembangan perangkat lunak.

e. ISO/IEC 25010

ISO/IEC (*International Standardization for Organization and International Electrotechnical Commission*) 25010 merupakan standar internasional yang mendefinisikan model kualitas sistem dan perangkat lunak berdasarkan sejumlah karakteristik utama. Model ini mencakup aspek *functional suitability* (kesesuaian fungsi), *reliability* (keandalan), *usability* (kegunaan), *performance efficiency* (efisiensi kinerja), *compatibility* (kompatibilitas), *security* (keamanan), *maintainability* (kemudahan pemeliharaan), serta *portability* (portabilitas). Berhubungan dengan perancangan sistem informasi, ISO/IEC 25010 digunakan sebagai acuan untuk memastikan bahwa sistem yang dirancang tidak hanya berfungsi, tetapi juga dapat digunakan secara efektif oleh pengguna. Standar ini berperan dalam mengidentifikasi kebutuhan kualitas sistem sejak tahap awal perancangan. Oleh karena itu, ISO/IEC 25010 relevan sebagai dasar teori dalam menilai kualitas konseptual sistem informasi yang diusulkan.

f. *Company Color*

Company color adalah warna khusus yang digunakan perusahaan sebagai identitas visual suatu perusahaan atau organisasi untuk memperkuat *branding*, meningkatkan konsistensi tampilan, serta memudahkan pengguna mengenali produk atau layanan. Dalam perancangan *user interface* sistem informasi, penggunaan *company color* pada

elemen dapat membantu mengenali sistem sebagai bagian dari identitas perusahaan yang sama. PT XYZ menggunakan *Crimson Red* sebagai *company color* yang utama dengan kode hex #EE1C25. Penggunaan warna merah biasanya dikombinasikan dengan warna putih dan hitam pada logo dan alat pemasaran. Warna pendukung untuk melengkapi warna utama adalah *Persian Pink* dengan kode hex #F78DA7. Adapun palet warna produk yang dihasilkan pada tahun 2025/2026 di antaranya *Vulcan Silver* (#8A8D91), *Velvet Navy Blue* (#1C2A44), dan *Hunter Green* (#355E3B).

g. Kode Etik Keamanan Data Perusahaan

Undang-Undang Nomor 11 Tahun 2008 tentang Informasi dan Transaksi Elektronik (UU ITE) mengatur kode etik berkaitan dengan akses data elektronik. Kode etik dimaksud melindungi data perusahaan dari intersepsi data elektronik dan perusakan data.

h. Kode Etik Rahasia Dagang

Undang-Undang Nomor 30 Tahun 2000 tentang Rahasia Dagang. Kode etik dimaksud melindungi informasi perusahaan yang bernilai ekonomi dalam hal ini data stok gudang, terdiri atas data item material dan kuantitas.