

BAB II

LANDASAN TEORI

2.1 Pencatatan / *Recording* Ternak

Recording adalah suatu usaha yang dikerjakan oleh peternak untuk mencatat gagal atau berhasilnya suatu usaha peternakan. Di bidang usaha peternakan program ini diterapkan hampir pada semua sektor usaha ternak mulai ternak unggas (layer, broiler, penetasan), ternak potong (sapi perah, sapi potong, kambing dan domba), dan aneka ternak seperti kelinci dan lainnya.

Recording sangat bermanfaat dalam setiap kegiatan atau usaha apapun, untuk usaha ternak sapi, *Recording* bermanfaat untuk :memudahkan pengenalan terhadap ternak dimana dengan mengetahui identitas dan ciri-ciri khusus ternak, serta mengetahui populasi ternak, memudahkan peternak mengingat kejadiankejadian penting pada ternaknya, memudahkan peternak mengambil keputusan ataupun tindakan nyata dalam penanganan, perawatan dan pengobatan pada ternak yang sakit berdasarkan catatan riwayat kesehatannya, memudahkan peternak melakukan seleksi ternak serta dapat mencegah terjadinya kawin sedarah atau inbreeding[13].

Pencatatan yang baik membantu pemilik peternakan mengelola sapi dengan lebih efisien, mengidentifikasi masalah dengan cepat, dan membuat keputusan berdasarkan data yang akurat. Berikut adalah penjelasan detail mengenai pencatatan ternak sapi:

1. Identifikasi Ternak:

Setiap sapi perlu diberi identifikasi unik untuk membedakan satu dari yang lain. Metode identifikasi meliputi penggunaan tato telinga, cincin leher, atau pemasangan mikrochip. Informasi identifikasi ini membantu melacak data spesifik mengenai sapi tertentu.

2. Kesehatan dan Perawatan:

Pencatatan kesehatan sapi mencakup riwayat vaksinasi, pemberian obat-obatan, dan tindakan medis lainnya. Catatan harus mencatat jenis obat, dosis, dan

tanggal pemberian. Jika ada penyakit atau cedera yang muncul, catat gejala, diagnosis, dan tindakan yang diambil.

1. Reproduksi:

Catatan reproduksi termasuk tanggal perkawinan atau inseminasi buatan dan estimasi tanggal kelahiran. Jika sapi melahirkan, catat jumlah anak yang lahir, kondisi persalinan, dan kesehatan anaknya.

2. Produksi dan Performa:

Pencatatan berat badan sapi pada tahap-tahap penting seperti lahir, pembelian, dan saat panen penting untuk memantau perkembangan sapi. Jika sapi di peternakan susu, catat produksi susu setiap harinya. Jika sapi di gemukkan, catat berat badan akhir dan penambahan berat badan selama masa penggemukan.

3. Pakan dan Nutrisi:

Catat jenis pakan yang diberikan dan komposisinya. Rekam jumlah pakan yang diberikan setiap hari atau setiap kali pemberian. Juga, catat respons sapi terhadap pakan tertentu dan perubahan pola makan.

4. Pengeluaran dan Pendapatan:

Pencatatan biaya seperti pakan, obat-obatan, vaksinasi, perawatan, dan investasi lainnya penting untuk menghitung biaya operasional peternakan. Rekam juga pendapatan dari penjualan daging, susu, kulit, dan sapi.

5. Kejadian Khusus:

Catat kelahiran, kematian, atau perpindahan sapi di peternakan. Juga catat kejadian khusus seperti penyakit masal, cuaca ekstrem, atau bencana alam yang mempengaruhi kesehatan sapi.

6. Observasi dan Perilaku:

Pencatatan perilaku tidak biasa atau tanda-tanda kesehatan yang mungkin terlihat pada sapi penting untuk mendeteksi masalah kesehatan dini.

7. Pencatatan Harian:

Pencatatan harian tentang pemberian pakan, perawatan, dan peristiwa penting membantu Anda melacak perkembangan dan membuat keputusan berdasarkan data real-time.

8. Catatan

Veteriner:

Simpan catatan dari kunjungan dokter hewan, diagnosis, dan rekomendasi yang diberikan. Ini penting untuk mengikuti arahan medis dan mengelola kesehatan sapi dengan baik.

2.2 Basis Data

Basis data merupakan salah satu bagian dalam rekayasa perangkat lunak yang terkomputerisasi dan bertujuan utama memelihara data yang sudah diolah atau media penyimpanan informasi agar dapat diakses dengan mudah dan cepat[15]. Menurut Indrajani, basis data adalah kumpulan data yang saling berhubungan secara logis dan didesain untuk mendapatkan data yang dibutuhkan oleh suatu organisasi[16]. Sedangkan menurut Yakub dan Hisbanarto menjelaskan, “basis data (database) merupakan kumpulan data yang saling berhubungan atau punya relasi”[17].

Basis data memiliki struktur yang terdiri dari tabel-tabel yang terdiri dari kolom-kolom dan baris-baris. Setiap baris dalam tabel mewakili entitas individu, seperti pelanggan, produk, atau transaksi, sementara setiap kolom berisi atribut-atribut yang berhubungan dengan entitas tersebut. Sistem Manajemen Basis Data (DBMS) digunakan untuk mengelola basis data, mengendalikan akses data, mengelola penyimpanan fisik, menjaga integritas data, serta mengizinkan pengguna untuk melakukan berbagai operasi seperti pengambilan data, penambahan, perubahan, dan penghapusan.

Basis data memiliki peran krusial dalam bisnis modern dan lingkungan teknologi informasi. Mereka memungkinkan perusahaan untuk mengelola informasi pelanggan, persediaan, transaksi, dan banyak aspek penting lainnya. Dengan menyediakan pendekatan terstruktur dan terpusat untuk mengelola data, basis data membantu dalam pengambilan keputusan yang lebih akurat dan efisien serta memungkinkan pengguna untuk mengambil manfaat maksimal dari potensi data yang mereka miliki.

2.3 Struktur Data

Struktur data adalah cara penyimpanan, pengorganisasian, dan pengaturan data di dalam media penyimpanan komputer sehingga data tersebut dapat digunakan secara efisien[18]. Sementara itu, menurut Hidayati struktur data ialah

suatu koleksi atau kelompok data yang dapat dikarakteristikan oleh organisasi serta operasi yang didefinisikan terhadapnya. Artinya, struktur data adalah suatu koleksi data yang diorganisasikan sedemikian rupa agar dapat digunakan (dioperasikan) berdasarkan definisi yang diberikan pada struktur tersebut. Misalnya, kita dapat mendefinisikan struktur data bernama “informasi_mahasiswa” yang berisi kumpulan data-data personal mahasiswa meliputi nama, NIM, alamat, dsb [19].

Beberapa contoh struktur data yang umum digunakan meliputi array, linked list, stack, queue, tree, dan graph. Setiap struktur data memiliki karakteristik dan fitur unik. Array, misalnya, menyimpan elemen dengan indeks dan cocok untuk akses data yang cepat. Linked list menggunakan tautan antar elemen dan efektif untuk penyisipan dan penghapusan data. Stack dan queue memberlakukan aturan khusus terkait urutan penyisipan dan penghapusan data. Tree dan graph digunakan untuk merepresentasikan hubungan hierarkis dan kompleks antara data.

Pemahaman yang kuat tentang struktur data menjadi penting saat merancang dan mengembangkan program komputer. Pemilihan struktur data yang sesuai dapat menghasilkan program yang lebih efisien, lebih mudah diatur, dan lebih cepat dalam melakukan tugas-tugas tertentu. Seorang pengembang perlu mempertimbangkan jenis operasi yang akan dilakukan pada data, kompleksitas algoritma yang akan digunakan, serta persyaratan kinerja dari aplikasi yang dibangun untuk memilih struktur data yang paling sesuai dengan kebutuhan tersebut.

2.4 Android

Android adalah sistem operasi mobile yang dikembangkan oleh Google. Sistem operasi ini pertama kali diperkenalkan pada tahun 2007 dan dirilis secara resmi pada tahun 2008. Android dirancang untuk berjalan pada perangkat mobile seperti smartphone, tablet, smartwatch, dan perangkat lainnya[21].

Android didasarkan pada kernel Linux dan dikembangkan dengan bahasa pemrograman Java dan Kotlin. Android Studio adalah IDE (Integrated Development Environment) resmi untuk pengembangan aplikasi Android, yang

menyediakan berbagai fitur untuk membantu pengembang dalam membuat, menguji, dan menerapkan aplikasi Android.

Salah satu keunggulan Android adalah adanya Google Play Store, toko aplikasi yang menyediakan ribuan aplikasi untuk diunduh dan digunakan oleh pengguna Android. Selain itu, Android juga memiliki berbagai fitur seperti notifikasi, layanan lokasi, dan dukungan untuk berbagai bahasa dan skrip [22].

2.5 Arsitektur Android

Arsitektur Android terdiri dari beberapa lapisan yang saling terkait dan saling berinteraksi untuk menjalankan aplikasi. Berikut adalah komponen-komponen penting dari arsitektur Android:

1. **Linux Kernel:** Merupakan bagian terbawah dari arsitektur Android dan bertanggung jawab atas manajemen perangkat keras, seperti memori, jaringan, dan driver.
2. **Library:** Library adalah kumpulan kode yang dapat digunakan oleh aplikasi untuk melakukan berbagai tugas, seperti mengakses database, mengelola file, atau melakukan koneksi jaringan.
3. **Runtime:** Runtime pada Android adalah lingkungan yang menjalankan aplikasi. Android Runtime (ART) adalah runtime default pada Android, dan bertanggung jawab untuk mengelola proses dan memori aplikasi.
4. **Application Framework:** Framework pada Android adalah kumpulan kode yang digunakan untuk membangun aplikasi. Framework menyediakan berbagai layanan dan API untuk memudahkan pengembangan aplikasi, seperti layanan lokasi, notifikasi, dan manajemen jaringan.
5. **Aplikasi:** Aplikasi Android adalah software yang digunakan oleh pengguna akhir. Aplikasi ini dapat diunduh dari Google Play Store atau dipasang secara langsung melalui file APK.

Komponen-komponen tersebut bekerja bersama-sama untuk menjalankan aplikasi pada perangkat Android. Saat pengguna membuka aplikasi, sistem akan

mengeksekusi kode aplikasi dan menyediakan lingkungan yang diperlukan untuk menjalankan aplikasi tersebut. [23].

2.6 Aplikasi

Aplikasi adalah program perangkat lunak yang dirancang untuk melakukan tugas tertentu pada komputer atau perangkat seluler seperti smartphone atau tablet. Aplikasi dapat dibuat untuk berbagai tujuan, mulai dari permainan hingga aplikasi produktivitas, seperti email, kalender, atau aplikasi perbankan. Aplikasi modern biasanya memiliki antarmuka grafis yang mudah digunakan dan dapat diunduh dari toko aplikasi seperti Google Play atau App Store. Aplikasi yang dirancang untuk perangkat seluler juga dapat menggunakan fitur perangkat keras seperti kamera, GPS, atau sensor gerak untuk meningkatkan fungsionalitasnya. [24]

Aplikasi memiliki beragam tujuan dan fungsi. Beberapa aplikasi diciptakan untuk mengotomatisasi tugas-tugas bisnis, seperti pengelolaan inventaris atau keuangan. Aplikasi lainnya mungkin fokus pada memberikan pengalaman hiburan, seperti permainan atau platform streaming. Aplikasi seluler dan web memungkinkan pengguna mengakses informasi, berkomunikasi, berbelanja, dan melakukan banyak aktivitas lain dengan mudah. Aplikasi juga dapat dikembangkan untuk tujuan pendidikan, pengembangan pribadi, dan penelitian ilmiah.

Dengan perkembangan teknologi, aplikasi telah menjadi bagian integral dari kehidupan sehari-hari kita. Mereka berkontribusi pada efisiensi, kenyamanan, dan inovasi di berbagai bidang. Pengembangan aplikasi melibatkan proses desain, pengkodean, pengujian, dan distribusi kepada pengguna akhir. Dalam ekosistem teknologi yang terus berkembang, aplikasi memainkan peran penting dalam membentuk bagaimana kita berinteraksi dengan teknologi dan dunia di sekitar kita.

2.7 Visual Studio Code

Visual Studio Code (VS Code) adalah sebuah editor teks yang dikembangkan oleh Microsoft. VS Code dirancang untuk membantu pengembang dalam menulis kode secara efisien dengan menyediakan fitur-fitur yang berguna, termasuk penyorotan sintaks, penyusunan kode, debugging, integrasi dengan

berbagai bahasa pemrograman dan ekstensi, serta dukungan untuk sistem pengendalian versi seperti Git.

Berikut adalah penjelasan lengkap beberapa fitur utama Visual Studio Code:

1. Antarmuka Pengguna yang Bersih: VS Code menawarkan antarmuka pengguna yang sederhana dan intuitif. Ini memiliki tata letak yang jelas dan tidak berantakan, yang memungkinkan Anda untuk fokus pada kode Anda tanpa gangguan.
2. Penyorotan Sintaks: VS Code secara otomatis menerapkan penyorotan sintaks untuk berbagai bahasa pemrograman. Ini membantu mengidentifikasi berbagai elemen kode seperti variabel, fungsi, dan kata kunci dengan warna yang berbeda, membuat kode lebih mudah dibaca.
3. Penyusunan Kode (Autocompletion): VS Code menyediakan dukungan penyusunan kode yang cerdas. Ketika Anda mulai mengetik, VS Code akan menampilkan saran autocompletion yang relevan berdasarkan konteks kode Anda, yang dapat mempercepat proses penulisan kode.
4. Integrasi Debugger: VS Code mendukung debugging untuk berbagai bahasa pemrograman, memungkinkan Anda untuk mengidentifikasi dan memperbaiki bug dalam kode Anda dengan mudah.
5. Integrasi Git: VS Code memiliki integrasi bawaan dengan Git, memungkinkan Anda untuk mengelola repositori Git langsung dari dalam editor. Anda dapat melakukan komit, cabut, gabung, dan melihat riwayat perubahan kode secara visual.
6. Ekstensi yang Kuat: VS Code didukung oleh ekosistem ekstensi yang besar dan aktif. Ekstensi memungkinkan Anda menyesuaikan dan memperluas fungsionalitas editor sesuai dengan kebutuhan Anda, seperti menambahkan dukungan bahasa baru, tema, atau alat pengembangan.
7. Integrated Terminal: VS Code menyediakan terminal terintegrasi yang memungkinkan Anda menjalankan perintah shell langsung dari dalam editor tanpa perlu beralih ke aplikasi terminal eksternal.
8. Integrasi Kontrol Versi: Selain Git, VS Code juga mendukung sistem pengendalian versi seperti Subversion (SVN) dan Mercurial.
9. Temakan dan Ikon: Anda dapat dengan mudah mengubah tampilan VS Code dengan menginstal tema dan paket ikon yang berbeda.

10. Kustomisasi: VS Code dapat sepenuhnya disesuaikan dengan preferensi pribadi Anda, seperti mengubah tata letak, pengaturan warna, dan shortcut kustom. [25].

2.8 Flutter

Flutter adalah sebuah framework open-source yang dibuat oleh Google untuk mengembangkan aplikasi seluler (mobile), web, dan desktop. Landasan teori di balik Flutter mencakup beberapa konsep dan teknologi yang memungkinkan kerangka kerja ini berfungsi dengan baik dan memberikan keuntungan-keuntungan yang telah dijelaskan sebelumnya.

Berikut adalah beberapa landasan teori yang penting untuk memahami bagaimana Flutter berfungsi:

1. **Widget-based UI** : Flutter menggunakan pendekatan berbasis widget untuk membangun antarmuka pengguna. Widget adalah komponen dasar dalam Flutter, dan semuanya adalah widget, baik itu elemen sederhana seperti teks atau gambar, maupun tata letak keseluruhan. Konsep ini mengacu pada "komposisi" daripada "warisan", yang berarti Anda menggabungkan widget ke dalam widget lainnya untuk membentuk antarmuka pengguna yang kompleks.

2. **Dart Programming Language** : Flutter menggunakan bahasa pemrograman Dart sebagai bahasa utama untuk mengembangkan aplikasi. Dart adalah bahasa yang dioptimalkan untuk kinerja tinggi pada platform Flutter dan dirancang dengan sintaks yang mirip dengan bahasa-bahasa modern lainnya, seperti JavaScript atau Java.

3. **Hot Reload** : Fitur Hot Reload di Flutter memungkinkan pengembang untuk melihat perubahan yang dilakukan pada kode secara langsung pada aplikasi yang sedang berjalan, tanpa perlu me-restart aplikasi. Ini memungkinkan iterasi yang cepat dan produktivitas yang tinggi selama proses pengembangan.

4. **Skia Rendering Engine**: Flutter menggunakan mesin rendering Skia, yang dikembangkan oleh Google, untuk menggambar antarmuka pengguna. Skia merupakan mesin rendering grafis yang cepat dan kuat, yang digunakan untuk menghasilkan antarmuka pengguna yang halus dan responsif.

5. Jembatan Natif (Native Bridge) : Meskipun Flutter membangun antarmuka pengguna menggunakan komponen-komponen sendiri (widget) yang bersifat platform-agnostik, tetapi untuk berinteraksi dengan fitur-fitur perangkat keras dan sistem operasi, Flutter menggunakan jembatan natif yang menghubungkan kode Dart dengan kode platform yang mendasarinya (Java/Kotlin untuk Android, Objective-C/Swift untuk iOS).

6. Widget Rendering Pipeline: Flutter memiliki proses rendering yang efisien dengan menggunakan widget rendering pipeline. Ketika ada perubahan pada widget, Flutter akan menghitung perubahan tersebut dan melakukan repaint (menggambar ulang) hanya pada bagian-bagian yang diperlukan, yang membantu meningkatkan kinerja aplikasi.

7. Platform Channels: Flutter menyediakan platform channels untuk berkomunikasi antara kode Dart dengan kode platform yang lebih lanjut. Ini memungkinkan akses ke fitur-fitur platform khusus dan plugin dari ekosistem Flutter untuk berinteraksi dengan fungsi-fungsi native.

Semua konsep di atas bekerja bersama untuk memberikan pengalaman pengembangan yang produktif dan aplikasi yang kaya fitur, responsif, dan berkinerja tinggi. Dengan menggunakan Flutter, pengembang dapat mencapai target berbagai platform dengan satu basis kode yang sama, menghemat waktu dan upaya dalam proses pengembangan aplikasi.

2.9 Shared Preferences

Shared Preferences adalah salah satu cara penyimpanan data sederhana dalam aplikasi Android. Ini adalah metode yang digunakan untuk menyimpan data sebagai pasangan kunci-nilai di dalam penyimpanan lokal perangkat Android. Data yang disimpan dalam Shared Preferences bersifat privat dan hanya dapat diakses oleh aplikasi yang membuatnya. Ini cocok untuk menyimpan pengaturan aplikasi, preferensi pengguna, dan data sederhana lainnya.

Penjelasan Singkat Cara Kerja Shared Preferences:

1. Membuat Objek Shared Preferences: Pertama, Anda perlu membuat objek Shared Preferences dengan menggunakan metode `getSharedPreferences()` atau

`getPreferences()` yang dipanggil dari objek `Context` seperti `Activity` atau `Application`.

2. Membuat Editor: Setelah mendapatkan objek `Shared Preferences`, Anda perlu membuat objek `SharedPreferences.Editor` dengan menggunakan metode `edit()`.
3. Menambahkan Data: Gunakan metode-metode pada objek `SharedPreferences.Editor` seperti `putString()`, `putInt()`, `putBoolean()`, dll., untuk menyimpan data ke dalam `Shared Preferences`.
4. Commit atau Apply: Setelah menambahkan data, panggil metode `commit()` atau `apply()` pada objek `SharedPreferences.Editor` untuk menyimpan perubahan. [27].

2.10 QR Code

Pada dasarnya QR code adalah salah satu teknologi yang sangat praktis dan banyak digunakan di jaman yang saat ini serba digital, untuk itu QR code menjadi suatu yang penting. Sebagian besar dari kita pun pasti sudah banyak menemukan QR code ini di berbagai tempat. Kepanjangan dari QR code adalah Quick Response Code. QR code adalah suatu barcode dua dimensi yang didalamnya berisi berbagai jenis informasi secara langsung. Untuk membuka QR code ini, diperlukan scan ataupun pemindaian dengan menggunakan `smartphone`. Umumnya, QR code mampu menyimpan 2089 digit atau 4289 karakter, termasuk tanda baca ataupun karakter spesial di dalamnya.

Dengan keunggulan tersebut, maka QR code mampu menampilkan berbagai teks, membuka URL, menyimpan kontak pada buku telepon, dll. QR code terdiri dari berbagai titik-titik dan suatu spasi yang sudah disusun kedalam bentuk kotak, dan setiap elemen di dalamnya juga memiliki arti masing-masing. Karena adanya elemen tersebut, maka membuat QR code lebih mudah untuk di scan oleh `smartphone` dan mampu menampilkan berbagai data ataupun informasi yang dimuat di dalamnya[28].

2.10.1 Bagian-bagian QR Code

Berikut adalah bagian-bagian dari QR Code, yaitu[28]:

1. *Positioning* *Detection* *Makers*

Positioning detection makers adalah merupakan salah satu bagian dari QR code yang memiliki bentuk kotak dan berjumlah tiga. Posisi elemen ini terdapat pada bagian pojok QR code. Gunanya adalah untuk memastikan alat scanner mampu membaca kode secara tepat dan mengetahui bentuk orientasi ataupun posisi dari barcode itu sendiri.

2. *Alignment Marking*

Penanda ini memiliki ukuran yang lebih kecil daripada elemen sebelumnya. Walaupun memiliki bentuk yang sama-sama kotak, namun alignment marking mempunyai fungsi guna menjaga QR Code walaupun dicetak pada suatu permukaan yang melengkung. Umumnya, ukuran dari QR code akan semakin besar jika data yang tersimpan dalam QR code itu sendiri banyak. Selain itu, jumlah data yang ada didalamnya pun bisa lebih banyak lagi.

3. *Timing Pattern*

Timing pattern adalah suatu bagian dari QR code yang terlihat seperti kotak kecil yang saling berjajar. Fungsi utamanya adalah untuk melakukan konfigurasi data grid. Dengan adanya timing pattern ini, maka alat pemindai akan mengetahui besaran matriks data yang dimuat.

4. *Version Information*

Version information adalah bagian dari QR code yang mampu memberikan informasi. Saat ini, terdapat lebih dari 40 QR code yang berbeda-beda. Dengan adanya tanda ini, maka alat scanner akan mengetahui jenis QR code yang dipindainya. Umumnya, versi yang paling banyak digunakan adalah versi 1 hingga versi 7.

5. *Format Information*

Pada bagian QR code ini, terdapat informasi yang akan menjelaskan toleransi error dan pola data mask. Dengan adanya elemen ini, maka scanner akan lebih mudah dalam melakukan pemindaian QR code demi menampilkan data yang sudah dimuat pada pengguna.

6. *Data and Error Correction Keys*

Elemen pada QR code ini sangat penting karena pada elemen inilah seluruh informasi data akan disimpan. Selain itu, elemen ini juga mencakup error correction

block yang mampu menjaga data agar tetap bisa dipindai walaupun terdapat kerusakan pada kode sebanyak 30%

7. *Quiet Zone*

Quiet Zone merupakan bagian kosong yang berada di area paling luar dari QR code. Sama seperti white space pada suatu desain, quiet zone pun memiliki elemen penting pada QR code guna menegaskan struktur yang didesain dan agar lebih mudah untuk dipindai. Quiet zone itu sendiri harus ada demi memisahkan QR code dari lingkungan yang ada disekitarnya, agar alat scanner bisa lebih mudah mengenalinya tanpa kesulitan. Walaupun memang area ini kosong, tapi area ini menjadi area yang sangat penting untuk QR code.

2.10.2 Jenis-jenis QR Code

Berikut adalah jenis-jenis QR Code, adalah[28]:

1. QR Code Statis

QR code statis adalah suatu QR code yang tidak bisa lagi diedit saat sudah dibuat, termasuk informasi yang ada didalamnya. Untuk itu, QR code ini sangat tepat digunakan untuk digunakan secara pribadi atau sebagai QR code API, yaitu suatu aplikasi yang didalamnya terdapat banyak jumlah data, seperti ID karyawan, dokumentasi produk, dll.

2. QR Code Dinamis

QR code dinamis adalah jenis QR code yang bisa diedit, diperbarui, maupun diubah sebanyak mungkin selama diperlukan. Untuk itu, jenis QR code ini sangat cocok digunakan untuk bisnis ataupun marketing. QR code ini juga bisa tetap dijaga ukurannya agar tetap kecil. Hal tersebut dikarenakan data yang tersimpan di dalam QR code tidak secara langsung disimpan, tetapi akan diarahkan pada suatu URL yang sebelumnya sudah ditetapkan pada QR code tersebut.

2.10.3 Penerapan QR Code

Berikut adalah beberapa contoh penerapan QR code beserta referensi yang dapat menjadi acuan:

1. Penerapan QR code pada produk makanan

QR code dapat digunakan pada produk makanan untuk memberikan informasi lebih lanjut mengenai asal-usul dan bahan-bahan yang digunakan pada produk tersebut. Hal ini dapat membantu konsumen untuk membuat keputusan yang lebih tepat dalam memilih produk dan juga meningkatkan transparansi dalam rantai pasokan[29].

2. Penerapan QR code pada transportasi

QR code dapat digunakan pada transportasi untuk memudahkan pengguna dalam membeli tiket, melakukan check-in, atau mengakses informasi lebih lanjut mengenai jalur transportasi tersebut[30].

3. Penerapan QR code pada museum dan pariwisata

QR code dapat digunakan pada museum atau objek wisata untuk memberikan informasi lebih lanjut mengenai sejarah atau cerita di balik objek tersebut. [31].

4. Penerapan QR code pada sistem keamanan

QR code dapat digunakan pada sistem keamanan untuk memberikan otorisasi atau autentikasi pada suatu pengguna atau perangkat. Hal ini dapat meningkatkan keamanan data dan menghindari penipuan atau kecurangan[32].

5. Penerapan QR code pada edukasi

QR code dapat digunakan pada edukasi untuk memberikan pengalaman pembelajaran yang lebih interaktif dan meningkatkan partisipasi siswa dalam pembelajaran[33]. Contoh referensi:

6. Penerapan QR Code pada jual beli

Sudah banyak penerapan qr code di toko-toko maupun pelaku usaha, bahkan sekarang qr code sudah mulai diterapkan di pasar tradisional agar dapat memudahkan proses pembayaran dan juga menambah alternatif media pembayaran[34]

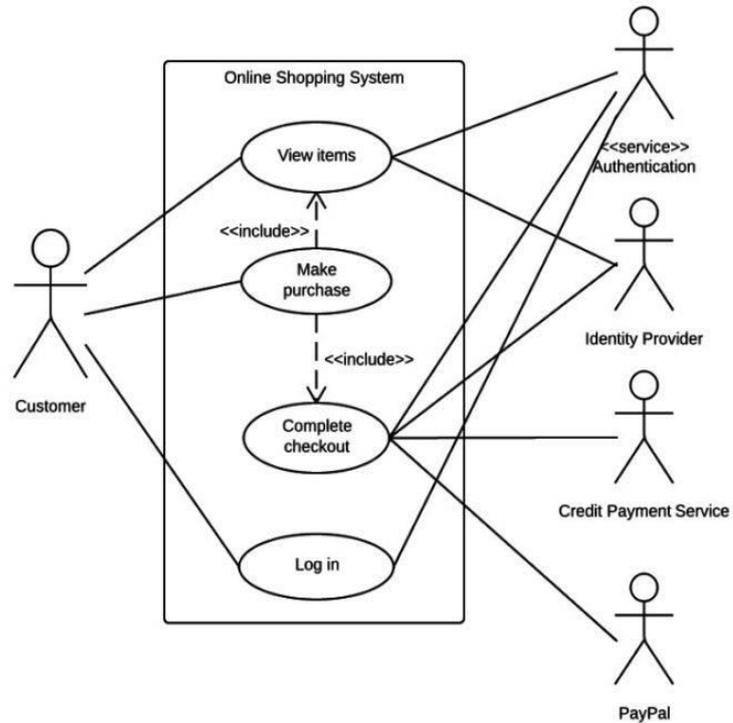
2.11 *Unified Modelling Language (UML)*

Unified Modelling Language (UML) adalah sebuah “Bahasa” yang sudah menjadi standarisasi untuk merancang dan mendokumentasi dari sistem

perangkat lunak, UML juga digunakan untuk membuat suatu model perangkat lunak, diperkenalkannya UML sejak tahun 1997, dengan saat ini sudah dikembangkan menjadi Bahasa pemodelan yang baku (de facto) pada sebuah pengembangan perangkat lunak [35]. UML juga digambarkan kedalam beberapa diagram seperti Use Case, Activity Diagram, Clas Diagram, Sequence Diagram. Berikut adalah penjelasan dari UML itu sendiri :

2.11.1 Use Case Diagram

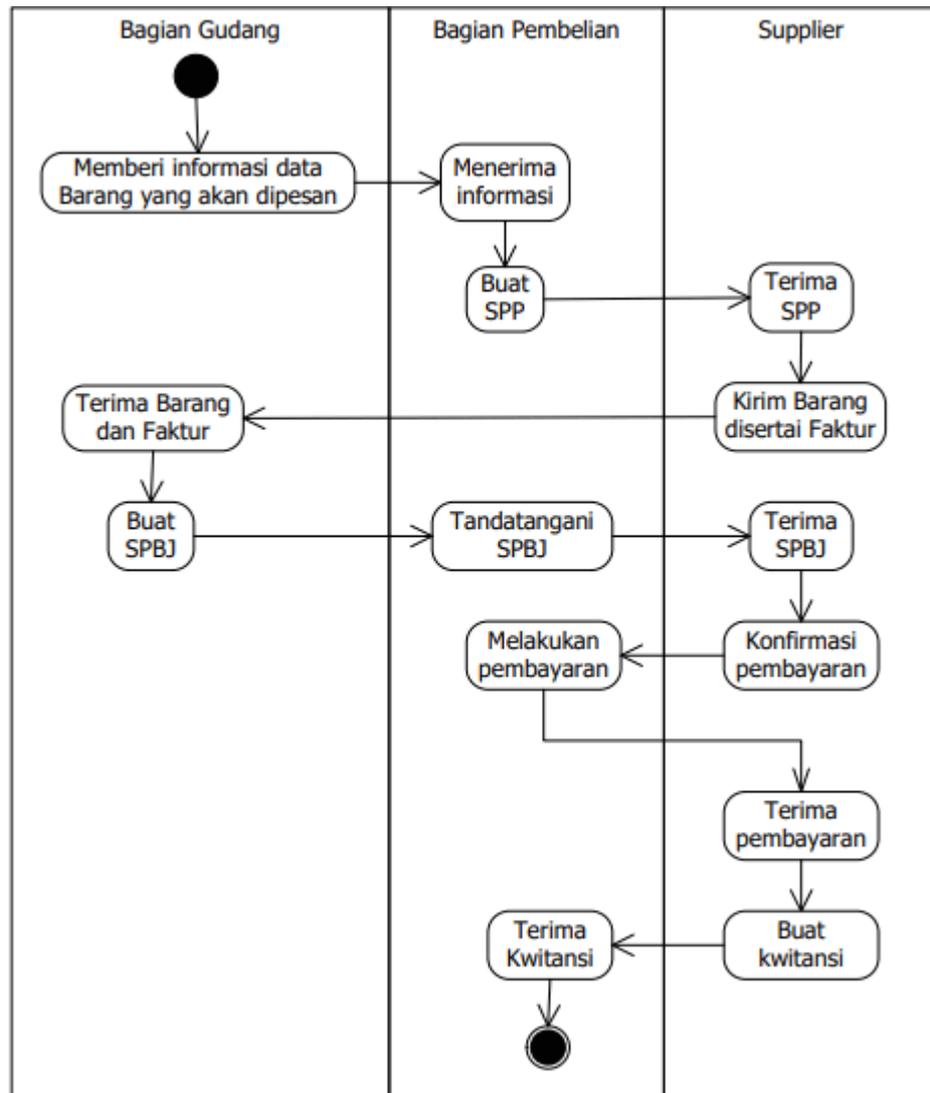
Use Case Diagram merupakan gambaran bagaimana actor yang sedang berinteraksi kepada sistem, serta membuat alur bisnis yang sudah didefinisikan pada Analisa sistem yang akan dijalankan. Use Case Diagram dapat digambarkan dengan actor dan use case. Aktor dapat digambarkan menjadi pelaku yang terlibat dalam penggunaan sistem, sementara itu use case merupakan gambaran dari sistem perangkat lunak yang akan dibangun. Berikut contoh dari Use Case Diagram pada gambar 2.10.1.



Gambar 2.11.1 Contoh Use Case Diagram

2.11.2 Activity Diagram

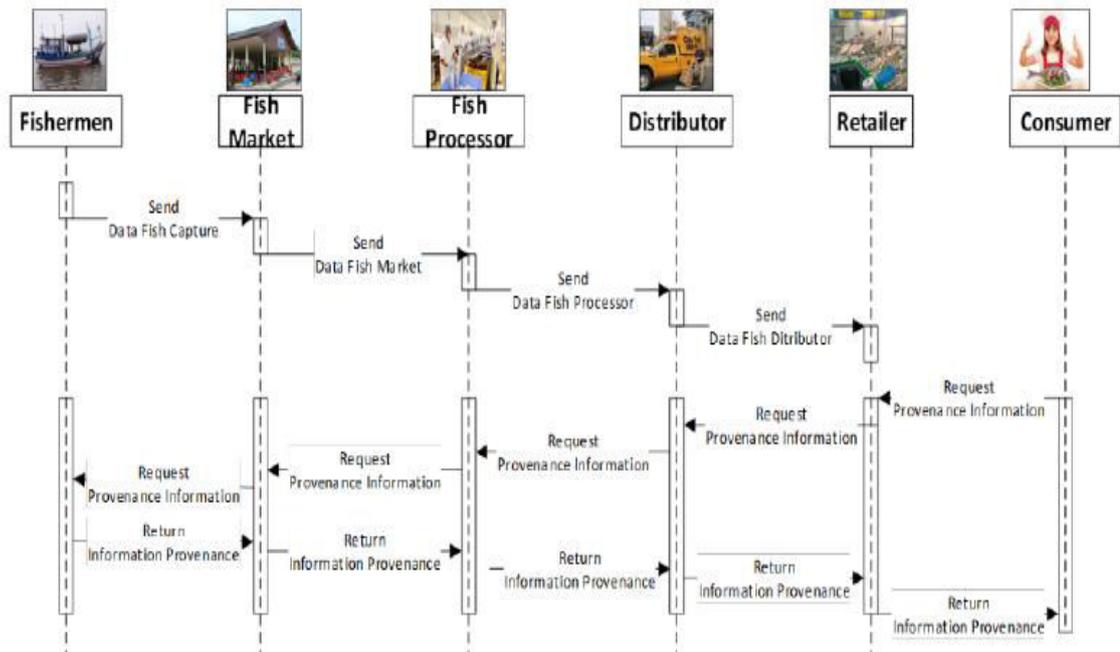
Activity Diagram Merupakan langkah-langkah cara kerja dalam sistem yang dirancang, sehingga terdapat masing-masing dari alur kerja sistem dimulai dari awal, kondisi yang akan terjadi pada saat eksekusi sistem samapai akhir sistem tersebut yang berjalan. Berikut contoh dari Activity Diagram pada gambar 2.10.2.



Gambar 2.11.2 Contoh Activity Diagram

2.11.3 Sequence Diagram

Sequence Diagram gambaran dari objek yang berinteraksi dan disusun dalam urutan waktu / kejadian pada sebuah proses. Berikut contoh dari Sequence Diagram pada gambar 2.10.3.



Gambar 2.11.3 Contoh Sequence Diagram

2.11.4 Class Diagram

Class Diagram merupakan sebuah spesifikasi yang menghasilkan sebuah objek berupa inti dari sebuah pengembangan dan perancangan berorientasi objek. Kelas (class) digambarkan sebuah keadaan (atribut/properti) suatu sistem, serta memberikan layanan untuk memanipulasikan keadaan tersebut (metode/fungsi). Berikut contoh dari Class Diagram pada gambar 2.10.4.

