

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Tinjauan Penelitian yang dilakukan untuk merancang aplikasi E-Nak untuk layanan pemantauan transaksi peternakan di desa Serayu Larangan yang dapat memaksimalkan potensi di bidang peternakan serta meningkatkan ekonomi peternak setempat. Observasi dilakukan terhadap beberapa penelitian yang sudah dilakukan sebelumnya, bahwa aplikasi yang diciptakan dapat menunjang potensi suatu usaha di beberapa daerah. Berikut beberapa studi sebelumnya yang memiliki relevansi dengan penelitian yang dilakukan.

Pertama, penelitian yang berjudul “**Penerapan Metode Rapid Application Development (RAD) Dalam Pengembangan Sistem Informasi Data Toko**”[8] dilakukan oleh Y. Dwi Wijaya, pada tahun. 2020. Penelitian ini membahas tentang penerapan metode pengembangan perangkat lunak yang cepat dan efisien, yang dikenal sebagai *Rapid Application Development (RAD)*, dalam konteks pengembangan sistem informasi untuk bisnis ritel atau toko. Metode RAD terkenal karena akselerasi proses pengembangan perangkat lunak dengan fokus pada iterasi dan prototipe yang cepat. Dalam konteks penelitian ini, metode RAD diterapkan dalam pengembangan sistem informasi yang terkait dengan data toko, termasuk manajemen persediaan, penjualan, dan laporan.

Kedua, penelitian yang berjudul “**Perancangan Aplikasi Informasi Bengkel Mobil Daerah Jakarta Barat Berbasis Android**” [2] dilakukan oleh R. M. Usup and S. Susafaati pada tahun 2021. Penelitian ini membahas tentang perancangan dan pengembangan aplikasi informasi bengkel mobil berbasis Android yang dapat membantu pengguna dalam memilih dan melakukan reservasi bengkel mobil. Hasil penelitian menunjukkan bahwa

aplikasi ini dapat memberikan manfaat bagi pengguna dan pemilik bengkel mobil di wilayah Jakarta Barat.

Ketiga, penelitian yang berjudul **“Digitalisasi Proses Bisnis Peternakan Dengan Implementasi Sistem Point of Sales (Pos) di 786 Ns Farm Deli Serdang”** [4] dilakukan oleh O. Putra Barus, J, dkk pada tahun 2022. Penelitian ini menghasilkan sebuah produk yang telah diimplementasikan oleh mitra 786NS *Farm*. Mitra berhasil menerbitkan laporan keuangan yang lengkap selama satu semester dan juga memahami cara membaca laporan keuangan serta memanfaatkan data dari laporan tersebut untuk merumuskan kebijakan operasional dalam pengelolaan peternakan. Selain itu, mitra juga berhasil memisahkan dana keuangan operasional usaha dan dana keuangan personal.

Keempat, penelitian yang berjudul **“Pemberdayaan Peternak Kambing Di Desa Tambong Kabupaten Banyuwangi Melalui Pelatihan Pemeliharaan Dan Perawatan Kesehatan Ternak”** [5] dilakukan oleh M. Habbib Khirzin dkk pada tahun 2022. Penelitian ini menunjukkan bahwa setelah mengikuti pelatihan, peternak kambing di Desa Tambong mampu meningkatkan pengetahuan dan keterampilan dalam mengelola peternakan kambing, meningkatkan kesehatan ternak, serta meningkatkan hasil produksi dan pendapatan. Pelatihan juga dapat memperkuat jaringan kerja antar peternak dan pihak-pihak terkait dalam pengembangan peternakan kambing di wilayah tersebut. Oleh karena itu, program pelatihan pemeliharaan dan perawatan kesehatan ternak dapat dijadikan strategi untuk meningkatkan kesejahteraan peternak dan mengembangkan peternakan kambing di Indonesia.

Kelima, penelitian yang berjudul **“Perbandingan Metode Pengembangan Perangkat Lunak (*Waterfall, Prototype, RAD*)”** [6] dilakukan oleh T. Pricillia dan Zulfachmi pada tahun 2021. Penelitian ini membandingkan tiga metode pengembangan sistem *Waterfall, Prototype*, dan *RAD (Rapid Application Development)*. Hasilnya menunjukkan bahwa *Waterfall* cocok untuk sistem generik dengan kebutuhan yang dapat

diidentifikasi dari awal. *Prototype* cocok untuk sistem *customize* dengan kebutuhan khusus. *RAD* cocok untuk sistem *customize*, berskala besar, dengan waktu pengembangan singkat dan kemungkinan perluasan di masa depan. Penelitian ini bisa memberikan pandangan terkait metode manakah yang paling cocok untuk penelitian ini

Keenam, penelitian yang berjudul **“Penerapan Metode *Prototype* Pada Perancangan Sistem Informasi Penggajian Karyawan (Persis Gawan) Berbasis *Web*”** [7] dilakukan oleh E. W. Fridayanthie, H. Haryanto, dan T. Tsabitah pada tahun 2021. Penelitian ini menerapkan teknik prototipe dalam pengembangan sistem informasi. Perancangan sistem penggajian karyawan melibatkan langkah-langkah *prototipe* yang mencakup diagram aktivitas, diagram *use case*, diagram *entity relationship*, konseptual *database*, dan desain antarmuka. Penggunaan metode prototipe dalam penelitian ini berhasil diterapkan.

Tabel 2. 1 Penelitian Terdahulu

No	Judul	Penulis	Tahun	Kesamaan	Perbedaan	Hasil
1	Penerapan Metode <i>Rapid Application Development</i> (RAD) Dalam Pengembangan Sistem Informasi Data Toko	Y. Dwi Wijaya	2020	Metode pengembangan aplikasi RAD	Berbasis <i>website</i>	Penerapan metode <i>Rapid Application Development</i> (RAD) terbukti efisien dari segi waktu. Metode ini, yang terdiri dari tiga tahap yaitu perencanaan, desain sistem, dan <i>implementasi</i> , dapat diselesaikan dengan cepat. Pengembangan sistem informasi di Toko Berkah menggunakan metode RAD membantu mengelola data toko dengan efektif.
2	Perancangan Aplikasi Informasi Bengkel Mobil Daerah Jakarta Barat Berbasis Android	R. M. Usup and S. Susafaati	2021	Pengembangan aplikasi untuk bisnis dan basis aplikasi adalah android	Studi kasus pada bengkel mobil, Metode pengembangan yang digunakan adalah metode <i>Waterfall</i> , dan pengujian menggunakan <i>white-box testing</i>	Aplikasi bengkel mobil membantu pengemudi mengatasi masalah saat kendaraannya mengalami gangguan dan mendorong pemeliharaan kendaraan yang lebih baik untuk mengurangi risiko kecelakaan. Kompatibel dengan OS Android minimal versi 6.0, aplikasi ini juga memanfaatkan Google Maps untuk memudahkan pengguna menemukan lokasi bengkel..
3	Digitalisasi Proses Bisnis Peternakan Dengan Implementasi	O. Putra Barus, J, dkk	2022	Digitalisasi pada peternakan, aplikasi berbasis	Menggunakan metode	Penelitian ini menghasilkan sebuah produk yang telah diimplementasikan oleh mitra <i>786NS Farm</i> . Mitra berhasil

No	Judul	Penulis	Tahun	Kesamaan	Perbedaan	Hasil
	Sistem <i>Point of Sales</i> (Pos) di 786 Ns Farm Deli Serdang			mobile, dan studi kasus pada peternakan	pengembangan yang berbeda	menerbitkan laporan keuangan yang lengkap selama satu semester dan juga memahami cara membaca laporan keuangan serta memanfaatkan data dari laporan tersebut untuk merumuskan kebijakan operasional dalam pengelolaan peternakan.
4	Pemberdayaan Peternak Kambing Di Desa Tambong Kabupaten Banyuwangi Melalui Pelatihan Pemeliharaan Dan Perawatan Kesehatan Ternak	M. Habbib Khirzin, dkk	2022	Analisis kebutuhan terkait pemantauan peternakan kambing	Penelitian berfokus pada pemberdayaan peternakan	Hasil penelitian ini menunjukkan program pelatihan pemeliharaan dan perawatan kesehatan ternak dapat dijadikan strategi untuk meningkatkan kesejahteraan peternak dan mengembangkan peternakan kambing di Indonesia.
5	Perbandingan Metode Pengembangan Perangkat Lunak (<i>Waterfall, Prototype, RAD</i>)	T. Pricillia dan Zulfachmi	2021	Membahas tentang metode <i>Rapid Application Developmrnt (RAD)</i>	Fokus penelitian adalah perbandingan tiga metode pengembangan sistem, bukan rancang bangun	Hasil penelitian ini yaitu <i>Waterfall</i> cocok untuk sistem generik dengan kebutuhan yang dapat diidentifikasi dari awal. <i>Prototype</i> cocok untuk sistem <i>customize</i> dengan kebutuhan khusus. <i>RAD</i> cocok untuk sistem <i>customize</i> , berskala besar, dengan waktu pengembangan singkat dan kemungkinan perluasan di masa depan.

No	Judul	Penulis	Tahun	Kesamaan	Perbedaan	Hasil
6	Penerapan Metode <i>Prototype</i> Pada Perancangan Sistem Informasi Penggajian Karyawan (Persis Gawan) Berbasis <i>Web</i>	E. W. Fridayanthie, H. Haryanto, dan T. Tsabitah	2021	Membahas tentang sistem informasi	Metode pengembangan sistem menggunakan metode <i>prototype</i> dan basis sistem adalah <i>web</i>	Sistem penggajian karyawan direncanakan menggunakan metode <i>prototipe</i> yang melibatkan langkah-langkah seperti diagram aktivitas, diagram <i>use case</i> , diagram <i>entity relationship</i> , konseptual database, dan desain antarmuka. Dalam pengembangan aplikasi penggajian, pemilihan bundel bahasa aplikasi web server memiliki dampak positif yang signifikan. Prototipe ini dapat berfungsi sebagai panduan dalam fase pengkodean sistem atau sebagai dokumen referensi bagi PT. Bina Angkasa.

2.2 Landasan Teori

2.2.1 Sistem Informasi

Sistem adalah sebuah kerangka kerja terdiri dari langkah-langkah yang terhubung satu sama lain, bekerja bersama untuk mencapai tujuan tertentu. Di sisi lain, informasi adalah data yang telah diolah menjadi format yang lebih bermakna dan bermanfaat bagi penerima. Dengan demikian, sistem informasi dapat dijelaskan sebagai sebuah sistem yang ada di dalam suatu entitas yang mengelola kegiatan transaksi sehari-hari, mendukung operasional, memiliki dimensi manajerial, dan terlibat dalam aspek strategis dengan menyediakan laporan yang dibutuhkan[9].

2.2.2 Sistem

Secara keseluruhan, sistem terdiri dari komponen yang saling terhubung untuk mencapai tujuan utama sistem tersebut. Tujuan utama dari sistem komputer adalah mengolah data menjadi informasi yang berguna, yang memerlukan dukungan dari perangkat keras, perangkat lunak, dan pengguna. Ketiga komponen ini harus berinteraksi secara harmonis. Tanpa perangkat lunak, perangkat keras hanya menjadi benda mati. Demikian pula, perangkat keras dan lunak hanya akan bermanfaat jika digunakan oleh manusia. Manajemen data merupakan hal penting dalam suatu organisasi, dan pendekatan berbasis komputer menekankan pentingnya manajemen basis data dan penerapannya dalam sistem informasi manajemen[10].

2.2.3 *Software Development Live Cycle (SDLC)*

SDLC adalah metode umum yang digunakan dalam pengembangan sistem informasi. *SDLC* mencakup beberapa tahap, dimulai dari perencanaan, analisis, desain, implementasi, hingga pemeliharaan sistem. Konsep *SDLC* ini menjadi dasar bagi berbagai model pengembangan perangkat lunak, membentuk kerangka kerja

untuk merencanakan dan mengendalikan pembuatan sistem informasi. *SDLC* juga bisa diartikan sebuah proses pengembangan atau modifikasi sistem perangkat lunak dengan memanfaatkan model dan metode yang telah terbukti efektif dari pengembangan sistem sebelumnya[11].

Terdapat beberapa model *SDLC* antara lain *waterfall*, *prototyping*, dan *rapid application development*. Model pengembangan *waterfall* cocok untuk perangkat lunak yang dirancang untuk membangun sistem dari awal dengan mengumpulkan kebutuhan yang relevan dengan topik penelitian yang dipilih, hingga produk tersebut diuji[6]. Model pengembangan *prototype* lebih sesuai untuk sistem atau perangkat lunak yang bersifat *customize*, dimana *software* diciptakan berdasarkan permintaan dan kebutuhan khusus, bahkan untuk situasi atau kondisi tertentu[6]. Sedangkan, model pengembangan *RAD* lebih cocok untuk sistem atau perangkat lunak yang bersifat *customize*, berskala besar, dan memerlukan waktu yang lebih singkat[6].

Model *SDLC* yang digunakan pada penelitian ini adalah *RAD* karena cocok untuk sistem atau perangkat lunak dengan sifat cepat dan fleksibel, dimana memungkinkan pembangunan sistem dengan waktu yang lebih singkat[8].

2.2.4 Metode *Rapid Application Development (RAD)*

Metode *RAD* adalah metode pengembangan *software* yang menitikberatkan pada siklus *developing* yang sekuensial dan linier, ditandai dengan waktu yang relatif singkat. Tujuan utamanya adalah mempercepat pembangunan sistem dengan mengutamakan pembuatan sistem, langkah-langkah iteratif, dan keterlibatan pengguna yang intensif[12].



Gambar 2. 1 Proses *Rapid Application Development*[12]

Proses dalam metode RAD mencakup langkah-langkah berikut:

1. Fase Perencanaan Kebutuhan

Fase perencanaan kebutuhan yang dimaksud adalah pengguna dan analis bekerja sama untuk mengidentifikasi tujuan dari aplikasi atau sistem yang akan dikembangkan, serta menentukan kebutuhan informasi yang diperlukan untuk mencapainya. Keterlibatan aktif dari kedua belah pihak sangat penting pada tahap ini, tidak hanya sebatas menyetujui proposal yang telah diajukan[13].

2. *RAD Design Workshop*

Perancangan skema konseptual, perancangan basis data, dan perancangan fungsionalitas dari sistem yang akan dikembangkan terjadi pada tahap ini. Analis bekerja sama dengan pengguna untuk merancang sistem informasi yang sesuai dengan keinginan pengguna dan bekerja sama selama pembangunan sistem informasi. Pada tahap ini, pengguna memberikan tanggapan terhadap sistem yang telah dibangun dengan analis. [12].

3. Fase implementasi (*Implementation*)

Analisis berkolaborasi secara intensif dengan pengguna. Setelah semua aspek yang diperlukan disetujui dan sistem selesai dibangun, dilakukan pengujian sistem, dan hasilnya diserahkan kepada pengguna[12].

2.2.5 *Android*

Android merupakan sistem operasi berbasis *Linux* yang dikembangkan oleh *Google* untuk *mobile device*, terutama *smartphone*[14]. Salah satu ciri khas Android adalah sifatnya yang terbuka, yang berarti dapat diakses dan dikembangkan oleh masyarakat umum. Artinya, tidak hanya *Google* yang memiliki akses eksklusif untuk mengembangkan aplikasi atau memodifikasi sistem operasi ini, tetapi juga siapa pun yang memiliki pengetahuan dan keterampilan yang diperlukan[15].

2.2.6 *Kotlin*

Kotlin merupakan sebuah bahasa pemrograman yang berjalan di atas *Java Virtual Machine (JVM)* dan dirancang sebagai bahasa pragmatis untuk pengembangan aplikasi Android. Bahasa ini menggabungkan paradigma pemrograman berorientasi objek dan fungsional. Kelebihan kotlin adalah interoperabilitasnya yang memungkinkan penggunaannya dalam proyek yang juga menggunakan bahasa pemrograman Java. Selain untuk pengembangan aplikasi Android, kotlin juga dapat digunakan untuk mengembangkan aplikasi *desktop*, *web*, dan *backend*. Pada awalnya, bahasa pemrograman ini dikembangkan oleh JetBrains, perusahaan yang juga menciptakan IntelliJ IDEA. Setelah mengalami kemajuan yang pesat, JetBrains merilis Kotlin sebagai proyek *open-source*, dan bahasa ini mendapatkan dukungan penuh dari *Google* untuk pengembangan aplikasi *Android*[16].

Berbagai keuntungan yang mungkin diperoleh dengan menggunakan Kotlin sebagai bahasa pemrograman untuk pengembangan aplikasi di *JVM* adalah sebagai berikut[16]:

1. Menjadi solusi terhadap masalah *NullPointerException* yang sering terjadi pada bahasa pemrograman Java.

2. Kode yang ditulis dalam Kotlin memiliki sintaksis yang lebih ringkas dan mudah dipahami dibandingkan dengan kode yang ditulis dalam bahasa Java.
3. Kemudahan dalam proses pembelajaran Kotlin.
4. Dukungan dari *Integrated Development Environment* (IDE) yang memudahkan dalam proses pemrograman[16].

2.2.7 Firebase

Firebase Database adalah layanan penyimpanan data *non-SQL* yang memungkinkan pengguna menyimpan berbagai tipe data seperti *String*, *Long*, dan *Boolean*. Data dalam *Firebase Database* disimpan dalam format objek berupa pohon JSON. Berbeda dengan basis data SQL yang menggunakan tabel dan baris, *Firebase Database* menyimpan data dalam struktur hierarki sebagai node dalam pohon JSON. Ketika data ditambahkan, data tersebut menjadi sebuah *node* di dalam struktur JSON tersebut. Setiap node dapat berisi data dan dapat memiliki cabang yang terdiri dari *node* lain dengan data node itu sendiri. Proses penambahan data ke *Firebase Database* dikenal sebagai operasi "*push*"[17].

2.2.8 Firebase Realtime Database

Firebase Realtime Database adalah *platform database* yang dirancang untuk aplikasi *real-time*. Ketika data mengalami perubahan, aplikasi yang terhubung dengan *Firebase* akan diperbarui secara otomatis di semua perangkat, baik di web maupun di aplikasi *mobile*. *Firebase* menawarkan pustaka yang luas dan lengkap untuk berbagai *platform web* dan *mobile*, memastikan integrasi yang mulus dan efisien dalam mengelola data secara langsung dan sinkron[18].

2.2.9 *Firestore Authentication*

Firestore Authentication adalah sebuah layanan yang disediakan oleh *Firestore* yang memudahkan dan mengamankan proses login pengguna ke dalam aplikasi. Layanan ini memungkinkan autentikasi pengguna menggunakan pustaka *UI* yang siap pakai serta SDK yang mudah digunakan. *Firestore Authentication* mendukung berbagai metode login, termasuk email/kata sandi, nomor telepon, serta penyedia identitas populer seperti Google, Facebook, Twitter, GitHub, dan lainnya. Dengan *Firestore Authentication*, pengembang dapat dengan mudah mengelola proses autentikasi pengguna sambil menambahkan fitur keamanan dan integrasi yang diperlukan untuk aplikasi yang dikembangkan[19].

2.2.10 *Firestore Performance Monitoring*

Firestore Performance Monitoring adalah fitur berbasis cloud yang membantu pengembang aplikasi memahami dan menganalisis kinerja fitur-fitur aplikasi pada platform *iOS* dan *Android*. Dengan menggunakan alat ini, pengembang dapat mengidentifikasi area dalam aplikasi yang memerlukan peningkatan kinerja. *Firestore Performance Monitoring* memberikan data yang memungkinkan pengembang untuk menentukan kapan dan dimana aplikasi membutuhkan optimasi untuk memastikan pengalaman pengguna yang optimal[20].

2.2.11 *RecyclerView*

RecyclerView adalah sebuah komponen yang memanfaatkan arsitektur sederhana, yang terdiri dari *UI controller*, *ViewModel*, dan *LiveData*. Dengan bantuan adaptor, *RecyclerView* dapat memahami cara menampilkan data di layar [21]. *RecyclerView* digunakan dalam menampilkan data set yang sudah di *push* dalam *realtime database*. Berikut contoh kode yang digunakan :

```

<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/rvInventaris"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:listitem="@layout/item_inventaris"/>

```

2.2.12 UML (Unified Modeling Language)

Unified Modeling Language (UML) merupakan salah satu standar bahasa yang mengatur proses definisi kebutuhan, pembuatan model analisis dan desain, serta memiliki kemampuan untuk mengilustrasikan arsitektur dalam pengembangan program berbasis objek. UML juga digunakan secara luas dalam pengembangan sistem di bidang rekayasa perangkat lunak. UML digunakan untuk menampilkan persyaratan dalam skenario yang menjelaskan bagaimana pengguna menggunakan sistem. Selain itu, UML juga dapat menunjukkan kendala-kendala yang ada di sistem[22].

2.2.13 Use Case Diagram

Diagram *use case* adalah representasi visual yang menggambarkan perilaku atau interaksi sistem secara terstruktur. Diagram *use case* memiliki peran yang signifikan dalam menggambarkan, mengklarifikasi, dan mendokumentasikan kebutuhan sistem[23].

2.2.14 Activity Diagram

Activity Diagram merupakan metode yang sering digunakan untuk mengilustrasikan logika proses, alur kerja, dan proses bisnis dalam suatu konteks. *Activity Diagram* juga berperan sebagai gambaran grafis dari aliran kerja atau aktivitas dalam suatu sistem, proses bisnis, atau menu yang terdapat dalam perangkat lunak[24].

2.2.15 Sequence Diagram

Sequence diagram adalah salah satu jenis *diagram* dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan interaksi antara objek dalam suatu sistem dalam urutan waktu tertentu. Diagram ini menampilkan objek-objek yang berpartisipasi dalam interaksi dan pesan-pesan yang dikirimkan di antara objek-objek tersebut dalam urutan yang terjadi [25].

2.2.16 Class Diagram

Class diagram merupakan sebuah diagram yang menggambarkan hubungan antar kelas, dimana setiap kelas memiliki atribut dan fungsi yang terkait dengan objek tersebut[26]. *Class diagram* menggambarkan struktur sistem yang terkait dengan visibilitas kategori yang dilakukan untuk membangun sistem[27].

2.2.17 Black Box Testing

Pengujian *black box* dalam rekayasa perangkat lunak adalah metode pengujian yang melibatkan hasil internal yang tidak diketahui. Para penguji melihat perangkat lunak sebagai "kotak hitam" yang tidak perlu menampilkan konten, namun dapat diuji secara eksternal. Dalam jenis pengujian *black box* ini, perangkat lunak akan dijalankan dan diuji untuk memastikan bahwa memenuhi kebutuhan yang telah ditetapkan oleh pengguna pada awalnya tanpa perlu mengungkapkan detail kode program[28].

2.2.18 Analisis Deskriptif

Analisis deskriptif digunakan untuk memberikan gambaran empiris terhadap data dalam penelitian. Analisis ini memberikan deskripsi langsung terhadap data tanpa bermaksud untuk membuat

kesimpulan umum. Tujuan utama dari statistik deskriptif adalah untuk merangkum karakteristik dasar dari suatu dataset[29].

Metode ini adalah suatu pendekatan statistik yang dipakai untuk menggambarkan data dengan tujuan mendeskripsikannya secara detail, sehingga memungkinkan untuk mengambil kesimpulan yang relevan dari kumpulan data tersebut [30]. Salah satu penerapannya ada pada uji coba fungsionalitas sistem, dengan melakukan kalkulasi formal sesuai rumus 2.1, dan hasilnya diinterpretasikan berdasarkan tabel interpretasi kelayakan yang terdapat pada Tabel 2.2.

$$\text{Presentase kelayakan} = \frac{\text{Skor perolehan}}{\text{Skor maksimal}} \times 100\% \quad (2.1)$$

Tabel 2. 2 Tabel Interpretasi Kelayakan[30]

Presentase kelayakan	Kriteria
81% - 100%	Sangat Baik
61% - 80%	Baik
41% - 60%	Netral
21% - 40%	Tidak Baik
0% - 20%	Sangat Tidak Baik