

BAB II

TINJAUAN PUSTAKA

2.1 Kajian Pustaka

Penelitian yang berjudul “Sistem Pengolahan Data Hasil Panen Buah Sawit Pada CV. XYZ” pada tahun 2020 yang dilakukan oleh Diah Triesia. Penelitian ini bertujuan untuk merancang sistem secara *mobile* agar mempermudah dalam penyusunan dokumen hasil panen karena sistem sebelumnya masih menggunakan cara manual untuk mengelola data hasil panen. Dengan menggunakan metode *waterfall* sebagai metode pengembangannya. Dalam tahap pengujian dengan metode *black box* yang telah diujikan sistem yang telah dibangun berjalan sesuai dengan yang diharapkan [3].

Penelitian yang berjudul “Sistem Informasi Pengolahan Data Buah Kelapa Sawit Pada KUD Sejahtera Pasaman Barat” pada tahun 2023 yang dilakukan oleh Lakry Maltaf Putra, Arief Rahmadian Aswin dan Aldo Melliano. Penelitian ini bertujuan untuk merancang sistem berbasis *website* agar mempermudah dalam pencatatan data penjualan buah kelapa sawit karena sistem sebelumnya masih menggunakan cara manual untuk mencatat data penjualan. Dengan menggunakan metode *waterfall* sebagai metode pengembangannya. Dalam tahap pengujian dengan metode *black box* yang telah dilakukan sistem yang telah dibangun berjalan sesuai dengan kebutuhan fungsional yang diharapkan [4].

Penelitian yang berjudul “Aplikasi Pengolahan Data Panen TBS Kelapa Sawit pada PT. Jo Perdana Agri Technology” pada tahun 2019 yang dilakukan oleh Muhammad Sony Maulana dan Agung Sasongko. Dalam penelitian ini bertujuan untuk merancang aplikasi berbasis *desktop* agar mempermudah pengolahan data hasil panen kelapa sawit karena sistem sebelumnya masih menggunakan cara manual untuk pendataanya.

Dengan menggunakan metode *waterfall* sebagai metode pengembangannya. Dalam tahap pengujian dengan metode *black box* yang telah dilakukan aplikasi yang telah dibangun berjalan lancar sesuai dengan yang diharapkan [5].

Tabel 2. 1 Penelitian Sebelumnya

No	Judul	Penulis	Metode	Hasil	Perbedaan
1	Sistem Informasi Manajemen Kebun Kelapa Sawit Pada CV. Cahya Gemilang [2]	Auliawati, A., & Candra, W. (2019)	<i>Prototype</i>	Penelitian ini berhasil membuat rancangan aplikasi berbasis <i>website</i> untuk pengolahan data karyawan kebun kelapa sawit pada CV. Cahya Gemilang	Studi kasus yang digunakan adalah pengolahan data karyawan kebun kelapa sawit pada CV. Cahya Gemilang. Aplikasi berbasis <i>website</i> dan metode yang digunakan adalah <i>prototype</i>
2	Sistem Pengolahan Data Hasil Panen Buah Sawit Pada CV. XYZ [3]	Triesia, D (2020).	<i>Waterfall</i>	Penelitian ini berhasil merancang aplikasi berbasis <i>android</i> untuk penyusunan laporan hasil panen pada CV. XYZ	Studi kasus yang digunakan adalah laporan hasil panen pada CV. XYZ
3	Aplikasi Perhitungan Hasil Panen dan Pendistribusian Buah Sawit Berbasis Web Pada PTPN V [11]	Nurramah, E., & Putri, R. N. (2021)	<i>Waterfall</i>	Penelitian ini berhasil merancang aplikasi berbasis <i>website</i> untuk perhitungan hasil panen dan pendistribusian kelapa sawit pada PTPN V	Studi kasus yang digunakan adalah perhitungan hasil panen dan pendistribusian kelapa sawit pada PTPN V.

No	Judul	Penulis	Metode	Hasil	Perbedaan
					Aplikasi berbasis <i>website</i>
4	Aplikasi Pengolahan Data Panen TBS Kelapa Sawit pada PT. Jo Perdana Agri Technology [5]	Maulana, M. S., & Sasongko, A. (2019)	<i>Waterfall</i>	Pada penelitian ini berhasil merancang aplikasi pengolahan data panen kelapa sawit pada PT. Jo Perdana Agri Technology. Aplikasi berbasis <i>desktop</i> dan memiliki keluaran berupa laporan pembayaran, laporan panen, laporan kirim, jurnal umum dan buku besar	Studi kasus yang digunakan adalah pengolahan data panen kelapa sawit pada PT. Jo Perdana Agri Technology. Aplikasi berbasis <i>desktop</i> dan keluarannya berupa laporan pembayaran, laporan panen, laporan kirim, jurnal umum dan buku besar
5	Sistem Informasi Pengolahan Data Buah Kelapa Sawit Pada KUD Sejahtera Pasaman Barat [4]	Putra, L. M., Aswin, A. R., & Melliano, A. (2023)	<i>Waterfall</i>	Pada penelitian ini berhasil merancang aplikasi berbasis <i>website</i> untuk pengolahan data buah sawit pada Koperasi Unit Desa (KUD) Pasaman Barat	Studi kasus yang digunakan adalah pengolahan data buah sawit pada KUD Pasaman Barat dan aplikasi berbasis <i>website</i>
6	Sistem Produksi Kelapa Sawit Dan Karet Dengan	Sari, I. P., & Wahyu, R. (2019)	<i>Waterfall</i>	Pada penelitian ini berhasil merancang	Studi kasus yang digunakan adalah

No	Judul	Penulis	Metode	Hasil	Perbedaan
	Membandingkan Hasil Produksi Menggunakan Simulasi (Studi Kasus: Kabupaten Kuantan Singingi) [12]			aplikasi berbasis <i>website</i> untuk perbandingan produksi hasil panen kelapa sawit dan karet di Kabupaten Kuantan Singingi	simulasi produksi hasil panen kelapa sawit dan karet di Kabupaten Kuantan Singingi. Aplikasi berbasis <i>website</i>
7	Analisis Dan Perancangan Sistem Informasi Manajemen Pelaporan Data Hasil Panen Berbasis Web Pada Dinas Pertanian Tanaman Pangan Provinsi Jambi [13]	Saputri, V., & Mulyono, H. (2019)	<i>Prototype</i>	Penelitian ini berhasil merancang sistem informasi manajemen pelaporan data hasil panen berbasis web pada Dinas Pertanian Tanaman Provinsi Jambi	Studi kasus yang digunakan adalah pelaoran data hasil panen pada Dinas Pertanian Tanaman Provinsi Jambi. Perancangan berbasis <i>website</i> dan metode yang digunakan adalah <i>prototype</i>
8	Perancangan Sistem Informasi Produksi Minyak Sawit Berbasis Web Pada PT. Bahari Gembira Ria Muaro Jambi [14]	Tri Khotimah, P. (2021)	<i>Waterfall</i>	Penelitian ini berhasil merancang aplikasi berbasis <i>website</i> untuk pengolahan data minyak sawit mentah pada PT. Bahari Gembira Ria Muaro Jambi	Studi kasus yang digunakan adalah pengolahan data minyak sawit mentah pada PT. Bahari Gembira Ria Muaro Jambi. Aplikasi berbasis <i>website</i>

No	Judul	Penulis	Metode	Hasil	Perbedaan
9	Sistem <i>E-Reporting</i> Hasil Pertanian Kelapa Sawit Pada PT Gunung Aji Jaya [15]	Tiranda, M. P., & Santoso, A. B. (2021)	<i>Waterfall</i>	Penelitian ini berhasil merancang aplikasi berbasis <i>website</i> untuk mengolah data hasil panen kelapa sawit pada PT. Gunung Aji Jaya	Studi kasus yang digunakan adalah pengolahan data hasil panen kelapa sawit pada PT. Gunung Aji Jaya. Aplikasi berbasis <i>website</i>
10	Perancangan Sistem Informasi Laporan Asset Berbasis <i>Smartphone</i> dengan Metode <i>Waterfall</i> [10]	Kurniawan, B. D., Andryana, S., & Benrahman, B. (2020)	<i>Waterfall</i>	Penelitian ini berhasil merancang aplikasi berbasis <i>mobile</i> untuk laporan aset pada Universitas Nasional	Studi kasus yang digunakan adalah laporan aset pada Universitas Nasional

2.2 Dasar Teori

2.2.1 Aplikasi *Mobile*

Aplikasi adalah sebuah perangkat lunak yang menjadi *front end* di sebuah sistem yang digunakan untuk mengolah data menjadi suatu informasi yang berguna orang-orang dan sistem yang bersangkutan [16]. Kata "aplikasi" berasal dari kata bahasa Inggris "application", yang berarti penggunaan, usulan, atau penerapan. Aplikasi adalah program siap pakai yang dikembangkan untuk melaksanakan tugas bagi penggunanya. Sedangkan Aplikasi mobile adalah program perangkat lunak yang melakukan tugas-tugas tertentu pada platform mobile. Platform mobile diantaranya adalah android, iOS, dan perangkat sistem operasi lainnya.

2.2.2 *Android*

Android adalah salah satu sistem operasi yang digunakan pada perangkat seluler seperti *smartphone* dan komputer tablet yang dibangun di atas kernel Linux [17]. *Android* pertama kali dikembangkan oleh perusahaan bernama Android Inc., dan pada tahun 2005 di akuisisi oleh raksasa Internet Google. *Android* dibuat dengan basis kernel Linux yang telah dimodifikasi, dan untuk setiap release-nya diberi kode nama berdasarkan nama hidangan makanan. *Android* merupakan sistem operasi yang bersifat open source, artinya semua orang bisa mengembangkannya [18].

2.2.3 Kotlin

Kotlin adalah bahasa pemrograman berbasis *Java Virtual Machine* (JVM). Kotlin merupakan bahasa pemrograman yang pragmatis untuk *android* yang mengkombinasikan *object oriented* (OO) dan bahasa fungsional. Kotlin juga bahasa pemrograman yang interoperabilitas yang membuat bahasa ini dapat digabungkan dengan bahasa pemrograman Java dalam satu *project* yang sama.

Kotlin juga dapat digunakan untuk pengembangan aplikasi berbasis *desktop*, *web* dan *backend* [19].

Kotlin awalnya dikembangkan oleh JetBrains, perusahaan dibalik IntelliJ IDEA. Setelah melalui banyak perkembangan, JetBrains merilis Kotlin secara open source dan perkembangannya semakin maju. Google mendukung penuh Kotlin untuk pengembang aplikasi android [20].

2.2.4 Android Studio

Android Studio merupakan sebuah IDE untuk *Android Development* yang diperkenalkan google pada acara Google I/O 2013. Android Studio merupakan pengembangan dari Eclipse IDE, dan dibuat berdasarkan IDE Java populer, yaitu IntelliJ IDEA. Android Studio merupakan IDE resmi untuk pengembangan aplikasi Android. Sebagai pengembangan dari Eclipse, Android Studio mempunyai banyak fitur-fitur baru dibandingkan dengan Eclipse IDE [21].

Android Studio memiliki banyak fitur yang meningkatkan efisiensi saat mengembangkan aplikasi Android selain sebagai editor kode IntelliJ dan alat pengembangan, misalnya [22]:

- Sistem Gradle yang fleksibel dan mudah digunakan
- Kode *editor* ini berisi berbagai macam fitur
- Lingkungan pengembangan yang terintegrasi untuk semua perangkat *android*
- Memiliki alat uji dan kerangka kerja yang kuat

2.2.5 Wireframe

Wireframe adalah sebuah cara yang digunakan untuk merancang desain dari suatu aplikasi, baik aplikasi yang menggunakan platform pada *website* ataupun *mobile* [23].




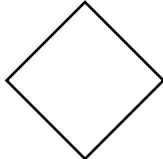

Wireframe sendiri dapat diartikan dengan sederhana sebagai kerangka gambar. Merancang sebuah *wireframe* berarti merencanakan kerangka garis besar sebuah aplikasi [24].

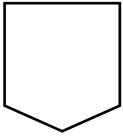


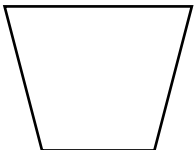
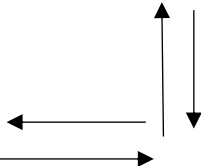
2.2.6 *Flowchart*

Flowchart adalah komponen dari sebuah proses yang menguraikan tahapan-tahapan yang diperlukan dalam memecahkan masalah. Fungsi utama *flowchart* adalah menggunakan simbol-simbol standar untuk menyampaikan langkah-langkah atau solusi dari suatu masalah secara jelas dan ringkas [25].

Untuk menggambarkan langkah atau pemecahan masalah secara sederhana, dapat dimengerti, rapi dan tidak ambigu digunakan simbol-simbol sederhana. Tabel 2.2 menunjukkan simbol-simbol yang sering digunakan dalam pembuatan *flowchart*.

Tabel 2. 2 Simbol-simbol *Flowchart*

No	Simbol	Nama	Keterangan
1		<i>Terminator</i>	Awal atau akhir suatu prosedur
2		<i>Output/Input</i>	Proses input atau output
3		<i>Process</i>	Proses operasional
4		<i>Decision</i>	Untuk menunjukkan kondisi tertentu pada dua arah
5		<i>On-Page Reference</i>	Koneksi penghubung dari satu proses ke proses yang lain pada halaman yang sama

6		<i>Off-Page Reference</i>	Koneksi penghubung dari satu proses ke proses yang lain di halaman yang berbeda
7		<i>Preparation</i>	Penyediaan tempat penyimpanan suatu pengolahan untuk memberikan nilai awal
8		<i>Document</i>	Mencetak <i>output</i> dalam bentuk dokumen fisik
9		<i>Manual Operation</i>	Proses yang tidak dilakukan oleh komputer
10		<i>Flow</i>	Menyatakan jalannya proses

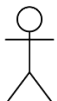

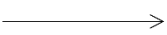
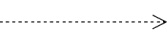


2.2.7 Use Case Diagram

Use case diagram adalah seperangkat skenario yang diikat bersama oleh *user* untuk mencapai tujuan [26]. *Use case diagram* mendukung pengembang perangkat lunak memahami interaksi. Aktor dalam sebuah sistem melakukan banyak *use case*, dan *use case* mungkin bisa memiliki banyak aktor.

Use case diagram berfungsi sebagai model untuk perilaku yang diinginkan dari sistem informasi. Skenario disediakan untuk setiap *use case*. Skenario *use case* menggambarkan bagaimana

prosedur *use case* berjalan dari perspektif aktor dan sistem [27]. Tabel 2.3 menunjukkan simbol-simbol yang digunakan pada *use case*.

Tabel 2. 3 Simbol-simbol *Use Case Diagram*



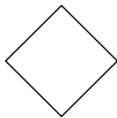

No	Simbol	Nama	Keterangan
1		Aktor	Mewakili peran orang ketika berkomunikasi dengan <i>use case</i>
2		<i>Use case</i>	Interaksi antara sistem dan aktor
3		<i>Association</i>	Penghubung antara aktor dengan <i>use case</i>
4		Generalisasi	Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
5		Generalisasi	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
6		Generalisasi	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

2.2.8 *Activity Diagram*

Activity diagram adalah suatu diagram yang menggambarkan konsep aliran data/kontrol, aksi terstruktur serta dirancang dengan baik dalam suatu sistem [28]. *Activity diagram*, dalam bahasa Indonesia diagram aktivitas, yaitu diagram yang dapat memodelkan proses-proses yang terjadi pada sebuah sistem. Runtutan proses dari

suatu sistem digambarkan secara vertikal. *Activity diagram* merupakan pengembangan dari *use case* yang memiliki alur aktivitas [29]. Tabel 2.4 menunjukkan simbol-simbol yang digunakan pada *activity diagram*.

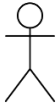




Tabel 2. 4 Simbol-simbol Activity Diagram

No	Simbol	Nama	Keterangan
1		Status Awal	Menunjukkan status awal dari diagram
2		Aktivitas	Menunjukkan aktivitas yang dilakukan
3		Percabangan	Menunjukkan pilihan aktivitas lebih dari satu
4		Status Akhir	Menunjukkan status akhir dari diagram

2.2.9 Sequence Diagram

Sequence diagram adalah sebuah diagram yang menggambarkan kolaborasi dari objek-objek yang saling berinteraksi antar elemen dari suatu class [28]. *Sequence diagram* atau diagram urutan adalah sebuah diagram yang digunakan untuk menjelaskan dan menampilkan interaksi antar objek-objek dalam sebuah sistem secara terperinci. Selain itu *sequence diagram* juga akan menampilkan pesan atau perintah yang dikirim, beserta waktu pelaksanaannya [29]. Tabel 2.5 menunjukkan simbol-simbol yang digunakan pada *sequence diagram*.

Tabel 2. 5 Simbol-simbol Sequence Diagram

No	Simbol	Nama	Keterangan
1		Aktor	Komponen ini menggambarkan seorang pengguna yang berada di luar sistem dan sedang berinteraksi dengan sistem
2		Activation Box	Merepresentasikan waktu yang dibutuhkan suatu objek untuk menyelesaikan tugasnya
3		Objek	Berfungsi untuk mendokumentasikan perilaku sebuah objek pada sebuah sistem
4		Messages	Menggambarkan komunikasi antar objek
5		Lifeline	Menggambarkan aktifitas dari objek

2.2.10 Waterfall

Alur hidup klasik adalah nama lain dari metode air terjun, yang umumnya dikenal sebagai *waterfall*. Metode ini menggambarkan pendekatan metodelis dan berurutan untuk pengembangan perangkat lunak yang dimulai dengan identifikasi kebutuhan pengguna. Setelah itu, bergerak melalui fase perencanaan (*design*) dan pemodelan (*modeling*), konstruksi (*construct*), dan membuat sistem tersedia untuk pengguna (*deployment*) [30].

Model pengembangan ini bersifat *linear* dari tahap awal pengembangan sistem yaitu tahap perencanaan sampai tahap akhir pengembangan sistem yaitu tahap pemeliharaan. Tahapan berikutnya tidak akan dilaksanakan sebelum tahapan sebelumnya selesai dilaksanakan dan tidak bisa kembali atau mengulang ke tahap sebelumnya [9].

2.2.11 *Black Box Testing*

Black box testing adalah jenis pengujian yang berkonsentrasi pada bagaimana perangkat lunak harus berfungsi. Persyaratan fungsional program dapat digunakan oleh penguji untuk menentukan serangkaian keadaan input dan melakukan pengujian [31].

Pengetahuan khusus dari kode aplikasi / struktur internal dan pengetahuan pemrograman pada umumnya tidak diperlukan. Menggunakan deskripsi eksternal perangkat lunak, termasuk spesifikasi, persyaratan, dan desain untuk menurunkan uji kasus. Tes ini dapat menjadi fungsional atau non-fungsional, meskipun biasanya fungsional. Perancang uji memilih *input* yang valid dan tidak valid dan menentukan *output* yang benar [32].

2.2.12 *White Box Testing*

White box testing adalah teknik untuk menguji perangkat lunak pada tingkat aliran program untuk memastikan *input* dan *output* sesuai dengan persyaratan yang diperlukan [33].

Dalam pengujian, uji dirancang dari perspektif pengembang dikarenakan struktur internal dikenal dengan menguji segala bagian kode yang mampu untuk diuji dengan tujuan untuk menentukan kesalahan logis dari kode sumber perangkat lunak [34].

Cyclomatic Complexity adalah besaran perangkat lunak yang menyediakan acuan kuantitatif kompleksitas suatu logika dalam

program. Rumus menghitung *Cyclomatic Complexity* adalah sebagai berikut [35]:

$$V(G) = E - N + 2 \quad (2,1)$$

Keterangan:

E = jumlah *edges* pada grafik alir

N = Jumlah *nodes* pada grafik alir