BAB II
TINJAUAN PUSTAKA

2.1 Kajian Pustaka

Untuk meningkatkan informasi yang diperoleh dari penelitian ini, penting untuk melihat penelitian sebelumnya yang berkaitan dengan subjek penelitian. Berikut ini adalah penelitian yang terkait.

Pertama, penelitian yang berjudul "*Systematic Literature Review* Analisis Metode Agile dalam Pengembangan Aplikasi Mobile". Menggunakan ulasan literatur yang komprehensif, studi ini memeriksa teknik-teknik agile dalam pengembangan aplikasi mobile. Hasil analisis mengungkapkan bahwa teknik *agile* yang paling sering digunakan, dengan presentasi 41%[14].

Kedua, penelitian yang berjudul "Perancangan Aplikasi Sistem Informasi Barang Berbasis Android Pada PT. Quantum Mitra Sinergi". Studi ini memeriksa konstruksi sistem informasi berbasis Android dan penggunaan pendekatan *waterfall* dalam pengembangan sistem penelitian ini. Tujuan penelitian ini adalah untuk membantu pekerja dalam mengelola data produk dengan lebih baik dengan menggunakan sistem aplikasi berbasis Android. Tujuan utama aplikasi adalah pemindai kode bar, yang dapat membantu dalam pendaftaran barang baru yang tiba dari gudang[15].

Ketiga, penelitian yang berjudul "Perancangan Aplikasi Inventaris Barang Berbasis Web Dengan Pengujian Menggunakan Pemodelan Nielsen". Tujuan dari penelitian ini adalah untuk bias membantu kinerja bagian tata usaha dalam pendataan asset barang yang dimiliki oleh SMKN 1 DUMAI guna mempercepat efektivitas kerja. Aplikasi web tersebut diuji menggunakan pemodelan *Nielsen* yang bertujuan untuk menguji kelayakn dari web yang telah dibuat. Adapun kriteria dari pemodelan *Nielsen* adalah *learnability, efficiency, memorability, few errors, satisfaction*[16].

Keempat, penelitian yang berjudul "Pembangunan Sistem Informasi Manajemen Laboratorium Terpadu Institut Teknologi Kalimantan". Tujuan dari studi ini adalah untuk menciptakan sistem informasi manajemen laboratorium di

Laboratorium Integrasi ITK untuk meminjam peralatan laboratorium dan persediaan dan mengelola persediaan bahan laboratorium. Pendekatan scrum digunakan dalam penelitian ini, yang mencakup perencanaan *sprint, scrums* harian, ulasan *sprint* dan retrospektif *sprint*. Berdasarkan lima putaran proses pengembangan ITK *Integrated* Laboratory SIM, total stok item yang diperoleh adalah hingga 65[17].

Kelima, penelitian yang berjudul "Pengembangan Sistem Informasi Manajemen *Supplier* dan Barang dengan *Extreme Programming*". Tujuan penelitian ini adalah untuk membantu menghilangkan kesalahan entri data, memungkinkan pemilik perusahaan untuk fokus pada mengontrol semua aset fasilitas yang dikendalikan perusahaan. *Extreme programming* digunakan dalam proses membangun sistem informasi berbasis web, dan diuji menggunakan pendekatan kotak hitam. Menurut temuan dari penelitian ini, sistem ini secara efektif melakukan penyimpanan pemasok, produk, transaksi pengiriman, pengembalian barang, dan juga dapat menginformasikan pemasar[18].

Tabel 2.1 Penelitian Terdahulu

No	Penulis	Tahun	Judul	Metode	Hasil
1	Azizah Nurfauziah Yusril, Inggrit Larasati dan Pajri Al Zukri	2021	Systematic Literature Review Analisis Metode Agile dalam Pengembangan Aplikasi Mobile	Literature Review	Menurut temuan penyelidikan, pendekatan agile yang paling sering digunakan adalah presentasi (41%), diikuti dengan penekanan tema penelitian yang paling banyak (23%), yang berada di bidang produktivitas.
2	Albert Anotona Hary Saputra Gulo	2021	Perancangan Aplikasi Sistem Informasi Barang Berbasis Android Pada PT. Quantum Mitra Sinergi	Waterfall	Hasilnya, program ini dapat digunakan pada smartphone, dan inklusi kemampuan pemindaian barcode melalui kamera smartphone memungkinkan karyawan untuk menemukan produk untuk bergerak masuk dan keluar dengan kecepatan yang lebih tinggi dan lebih cepat..
3	M. Rudi Sanjaya1, Jaidan Jauhari, dan Dedy Kurniawan	2021	Perancangan Aplikasi Inventaris Barang Berbasis Web Dengan Pengujian Menggunakan Pemodelan Nielsen	Prototype	Menghasilkan sitem aplikasi inventaris barang berbasis web yang dapat berjalan dengan baik dan hasil dari pengujian menggunakan pemodelan Nielsen bahwa aplikasi valid untuk digunakan
4	Dimas Saputra	2021	Pembangunan Sistem Informasi Manajemen	Scrum	Membuat sistem informasi manajemen laboratorium untuk mendukung alat laboratorium

			Laboratorium Terpadu Institut Teknologi Kalimantan		dan pinjaman bahan serta manajemen persediaan bahan laboratorium.
5	Astria Hijriani, Jannati Asri Safitri, Rd Irwan Adi Pribadi, dan Rico Andrian	2020	Pengembangan Sistem Informasi Manajemen Supplier dan Barang dengan Extreme Programming	Extreme Programming	Menciptakan sistem informasi manajemen barang online dengan kemampuan untuk melakukan pelaporan pemasok, pengiriman barang, pengembalian barang, dan pemberitahuan kepada penyedia.

Dari kajian pustaka diatas dapat disimpulkan bahawa dengan adanya pembuatan sistem inventarisasi barang berbasis *android mobile* atau *website* menggunakan metode *agile* dapat meringankan sistem pemrosesan khusus masuk dan keluarnya barang, serta laporan - laporan sangat diperlukan bagi perusahaan untuk keberlangsungan produktivitas di perusahaan tersebut. Salah satu hal yang membedakan kelima kajian pustaka dari proposal penelitian adalah bahwa pendekatan agile adalah metodologi penelitian yang digunakan oleh peneliti. Yang kedua mengenai tempat studi kasus yang berbeda yang dimana bertempat di PT Purwokerto Wahyu Keprabon.

2.2 Landasan Teori

2.2.1 Android

Android adalah sistem operasi seluler yang dibangun di atas kernel Linux yang berisi sistem operasi, middleware, dan aplikasi. Android adalah platform terbuka yang memungkinkan pengembang merancang aplikasi mereka sendiri[19].

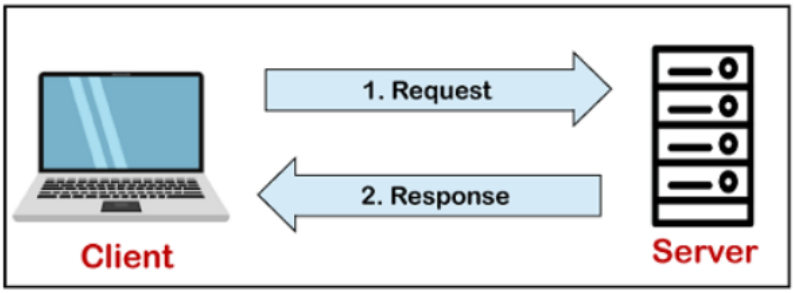
Saat ini, studi didasarkan pada Kerangka React Native. React Native adalah kerangka kerja sumber terbuka yang dikembangkan oleh Facebook setelah rilis React.js. React.js adalah kerangka kerja Facebook untuk membuat antarmuka pengguna (UI)[20]. React Native adalah kerangka kerja sumber terbuka untuk membuat aplikasi JavaScript yang berjalan di banyak platform. (Seperti yang dibuat untuk platform Android, iOS, dan Windows.)[21].

React Native ditulis dalam JSX, yang merupakan kombinasi dari JavaScript dan sintaks seperti XML. NPM atau Node Package Manager sering digunakan untuk membantu manajemen konstruksi aplikasi, seperti instalasi modul dan integrasi libraryJavascript ketika mengembangkan dengan React Native. Setiap komponen dalam React Native memiliki siklus atau *lifecycle*. *Lifecycle* membuatnya lebih mudah untuk menjalankan metode atau fungsi, meningkatkan efisiensi proses rendering. Siklus diklasifikasikan menjadi tiga jenis: siklus pemasangan, siklus pembaruan, dan siklus pelepasan[22].

2.2.2 MySQL *Database*

MySQL AB membuat dan memelihara sistem manajemen basis data MySQL (DBMS). Untuk memproses *database* relasional yang ada, MySQL memiliki bagian *Structured Query Language* (SQL)[23]. MySQL adalah sistem manajemen basis data yang mampu menyimpan dan memproses berbagai tipe data. Manipulasi data adalah proses penambahan, modifikasi, dan penghapusan data dari database. SQL adalah singkatan dari *Structured Query Language*. SQL adalah *Structured Query Language* (SMBD) yang terhubung ke *database*, dan MySQL adalah salah satunya.. Dengan kata

lain, MySQL adalah *database*, sedangkan SQL adalah perintah atau bahasa yang menyertainya[24].



Gambar 2. 1 Cara Kerja MySQL[24]

Gambar di atas menggambarkan struktur client-server dasar. Sebuah jaringan khusus adalah jaringan yang menghubungkan satu atau lebih perangkat ke server. Setiap klien dapat mengirimkan permintaan menggunakan antarmuka pengguna grafis (GUI) di layar, dan server akan menjawab selama server dan klien membaca instruksi dengan benar[25].

2.2.3 *Representational State Transfer Application Program Interface* (REST API)

REST API adalah kumpulan prinsip arsitektur yang memungkinkan data untuk ditransmisikan menggunakan antarmuka standar, seperti HTTP. REST API bekerja dengan cara yang sama seperti aplikasi web biasa. Klien dapat membuat *query* ke *server* melalui protokol HTTP, dan *server* akan merespon. REST diciptakan oleh Roy Fielding, *co-founder* dari proyek Apache HTTP *Server*[26].

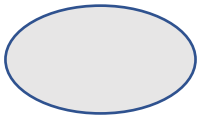
2.2.4 Unified Modeling language (UML)

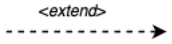
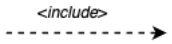
Unified Modeling Language merupakan sebuah bahasa pemodelan yang sudah banyak dipakai dalam metode object-oriented. Berdasarkan pengertian dari berbagai sumber buku, UML adalah sebuah bahasa untuk memodelkan sistem yang akan dirancang dengan menggunakan notasi yang sudah disepakati[27]. Berikut merupakan contoh diagram pemodelan :

1. *Use Case Diagram*

Use Case Diagram menunjukkan bagaimana interaksi yang terjadi antara sistem dan pengguna. Diagram ini dapat digunakan untuk menjelaskan konteks sistem dan menunjukkan batasan sistem[28]. Ada 2 elemen penting yang harus digambarkan, yaitu aktor dan *Use Case*. Aktor adalah segala sesuatu yang berinteraksi langsung dengan system. *Use Case* dinotasikan dengan simbol oval dengan nama kata kerja aktif di bagian dalam yang menyatakan aktivitas dari perspektif *actor.* Sebuah *Use Case*, dikenal sebagai *base Use Case,* dapat memiliki hubungan dengan satu atau lebih *Use Case* lainnya, dikenal sebagai *supplier Use Case*, dalam bentuk *extend* atau *include*. Relasi extend memungkinkan *supplier Use Case* untuk memperluas fungsionalitasnya jika dibutuhkan untuk mengimplementasikan alur alternatif yang ada pada *scenario Use Case* dari *base Use Case*. Di sisi lain, relasi include memastikan bahwa fungsionalitas dari *base Use Case* selalu hanya dapat diperluas jika dibutuhkan[29]. Berikut merupakan daftar *symbol use case diagram* :

Tabel 2. 2 Daftar *Symbol Use Case Diagram*[29]

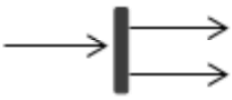
	*ACTOR* Orang proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari *actor* adalah gambar orang, biasanya dinyatakan menggunakan kata benda di awal frase nama *actor*.
	*USE CASE* Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesar antar unit atau actor biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama *use case.*

	ASOSIASI/*ASSOCIATION* Komunikasi antara *actor* dan *use case* yang berpartisipasi pada *use cas*e atau *use case* memiliki interraksi dengan *actor*.
	EKSTENSI/*EXTEND* Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan memiliki nama depan yang sama dengan *use case* yang di tambahkan
	GENERALISASI/*GENERALIZATION* Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya
	MENGGUNAKAN/*INCLUDE* Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsional atau sebagai syarat dijalankan *use case* ini

2. *Activity Diagram*

Setelah membuat model *Use Case*, setiap scenario yang ada di dalamnya akan digambarkan lebih jelas dalam activity diagram. *Activity diagram* adalah pemodelan yang menggambarkan sistem kerja sebuah objek atau sistem. *Activity diagram* menggambarkan proses kerja dari awal hingga akhir dari *use case* yang sedang diproses, dengan setiap aktivitas diberi notasi yang sesuai dengan fungsinya[30]. Berikut merupakan daftar *symbol activity diagram* :

Tabel 2. 3 Daftar *Symbol Activity Diagram*[30]

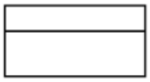
	STATUS AWAL/*INITIAL* Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah satutus awal.
	AKTIVITAS/ *ACTIVITY* Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	*PERCABANGAN / DECISION* Asosiasi percabangan dimana lebih dari satu aktivitas digabungkan menjadi satu.
	PENGGABUNGAN/ *JOIN* Asosiasi penggabungan dimana lebih dari satu aktivitas lebih dari satu.
	STATUS AKHIR/ *FINAL* Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status satu.
	*SWIMLINE* Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

3. *Class Diagram*

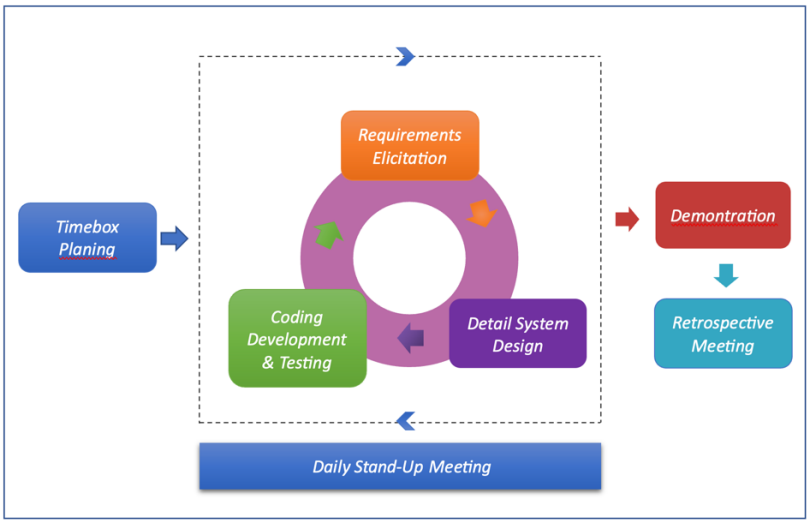
Class diagram adalah salah satu pemodelan yang cukup penting dalam UML, fungsinya adalah untuk membuat sebuat *logical models* dari sebuah *system*. Sebuah *class diagram* akan menunjukan bagaimana skema dari arsitektur sebuah sistem yang sedang dirancang[31]. *Class diagram*

digambarkan dengan *class* yang berisi atribut dan method, setiap class akan dihubungkan dengan sebuah garis disebut Asosiasi. Berikut merupakan daftar *symbol class diagram* :

Tabel 2. 4 Daftar *Symbol Class Diagram*[31]

GAMBAR	NAMA	KETERANGAN
	Generalization	Hubungan dimana objek anak *(descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
	*Nary Association*	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	*Class*	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	*Collaboration*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
	*Realization*	Operasi yang benar-benar dilakukan oleh suatu objek.
	*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
	Association	Apa yang menghubungkan antara objek satu dengan objek lainnya

2.2.5 Agile Software Development



Gambar 2. 2 Metode *Agile*[32]

Pengembangan perangkat lunak *agile* adalah metode pengembangan sistem jangka pendek di mana pengembang harus bereaksi terhadap setiap perubahan yang muncul. Dalam pengembangan perangkat lunak yang fleksibel, interaksi dan staf lebih penting daripada proses dan alat; perangkat lunak fungsional lebih penting daripada dokumentasi lengkap; kolaborasi pelanggan lebih penting daripada negosiasi kontrak; dan daya tanggap terhadap perubahan lebih penting daripada kepatuhan rencana[33].

Agile Software Development adalah proses di mana tim pengembangan, pemangku kepentingan, dan pengguna akhir bekerja sama sebagai tim lintas fungsi, dan mereka mengidentifikasi persyaratan dan solusi secara berulang selama proses pengembangan[34].

Ada berbagai tahapan yang harus diselesaikan ketika mengembangkan perangkat lunak menggunakan teknik *Agile*, yaitu[32]:

1) *Timebox Planning*

   Perencanaan adalah proses di mana tim pengembangan dan klien merancang apa yang diperlukan dalam pembuatan sistem. Ketika secara konseptual mempertimbangkan sistem baru yang akan dibangun. Pada saat ini, hasil diskusi masih berupa penjelasan umum algoritma sederhana, *pseudocode,* atau model yang serupa.

2) Daily Stand-Up Meeting (Requirements Elicitation, Detail System Design, Coding Development & Testing)

   Analisis aplikasi yang akan dibuat akan mencakup desain dalam *Unified Modelling Language* bersama dengan *Use Case Diagram* dan *Activity Diagram*. Serta mengevaluasi penerapan metode Agile saat mengembangkan aplikasi "My Kitchen" untuk mengetahui apakah itu ramah pengguna. Pengiriman sistem baru akan lebih cepat jika perspektif pengguna diterjemahkan dengan baik oleh pengembang.

3) *Demontration*

   Pada titik ini, pengembangan sistem dimulai dengan pengenalan model yang akan digunakan. Evaluasi sistem akan melibatkan evaluasi model dan deskripsi proses kerja aplikasi yang akan digunakan oleh pengguna.

4) Retrospective Meeting

   Langkah terakhir ini melibatkan pembuatan sistem menggunakan metodologi agile untuk membuat model sistem berdasarkan penerimaan pengguna. Perspektif pengguna yang diterjemahkan dengan baik oleh pengembang akan membuat proses pengiriman sistem baru lebih mudah. Pada titik ini, para peneliti akan bertemu dengan pelanggan untuk menganalisis sprint yang baru selesai dan mengevaluasi apa yang dapat dilakukan untuk meningkatkan produktivitas sprint berikutnya.

2.2.6 System Usabiility Scale (SUS)

System Usability Scale adalah salah satu alat uji yang paling sederhana untuk digunakan; itu terdiri dari sepuluh pertanyaan yang dikemas dalam kuesioner, tetapi juga dapat dibuat sebagai gambar yang disebut pictorial-SUS dalam progresinya. Skala *Usability System* menggunakan skala Likert lima poin dengan tanggapan yang berkisar dari "Tentunya tidak setuju" hingga "Tentu saja setuju." Menimbang skor SUS antara 0 dan 100[35].

Pada tahap pengujian, peneliti menggunakan 30 responden, karena banyaknya responden minimal yang diperlukan untuk mengisi kuesioner SUS minimal 30 responden[36]. Tabel 2.2 menampilkan 10 item kuesioner dari *System Usability Scale Questionnaire* yang sudah melalui tahap uji validitas dan telah diukur realibilitasnya[37].

Tabel 2.5 Pertanyaan SUS *Questionnarie*[37]

No	Komponen Pertanyaan	*Strongly Disagree*				*Strongly Agree*
1	Saya berpikir akan menggunakan sistem ini lagi	1	2	3	4	5
2	Saya merasa sistem ini rumit untuk digunakan	1	2	3	4	5
3	Saya merasa sistem ini mudah digunakan	1	2	3	4	5
s4	Saya membutuhkan bantuan dari orang lain atau teknisi dalam mengguanakan sistem ini	1	2	3	4	5
5	Saya merasa fitur-fitur sistem ini berjalan dengan semestinya	1	2	3	4	5
6	Saya merasa ada banyak hal yang tidak konsisten (tidak serasi pada sistem ini)	1	2	3	4	5

7	Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat	1	2	3	4	5
8	Saya merasa sistem ini membingungkan	1	2	3	4	5
9	Saya merasa tidak ada hambatan dalam menggunakan sistem ini	1	2	3	4	5
10	Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini	1	2	3	4	5

Tahap selanjutnya adalah menghitung data setelah data responden dikumpulkan. Ada berbagai pedoman yang harus diperhatikan saat menggunakan SUS untuk perhitungan skor, antara lain :

1) Skor yang diterima dari pengguna untuk setiap pertanyaan dengan nomor ganjil akan dikurangi menjadi 1.
2) Setiap pertanyaan dengan nomor genap, dan skor akhir dihitung dengan mengurangi nilai 5 dari skor pengguna.
3) Skor SUS dihitung dengan melipat gandakan skor ringkas dari setiap pertanyaan dengan 2.5[38].

Perhitungan Skor SUS adalah sebagai berikut :

$$\text{Skor SUS} = \left(\left((R1 - 1) + (5 - R2) + (R3 - 1) + (5 - R4) + (R5 - 1) + (5 - R6) + (R7 - 1) + (5 - R8) + (R9 - 1) + (5 - R10)\right) * 2.5\right) \quad (2.1)$$

Setelah itu, rata-rata skor SUS dari masing-masing responden dihitung dengan menambahkan semua skor dan membagi dengan jumlah responden. Rasio SUS dapat dihitung menggunakan rumus di bawah ini:

$$\bar{x} = \frac{\sum x}{n} \quad (2.2)$$

Keterangan : $\bar{x} = skor\ rata\ rata$

$\sum x = jumlah\ skor\ SUS$

$$n = jumlah\ responden$$

Skor rata-rata dari semua responden diperoleh setelah dihitung. Kemudian lakukan penyesuaian berikut :

Tabel 2.6 *Score* SUS *Questionnare*[35]

Score	*Grade*	*Rating*
>81	A	*Excellent*
68 – 81	B	*Good*
68	C	*OK/Fair*
51 – 67	D	*Poor*
<51	E	*Worst*

Tabel 2.3 adalah rincian skor yang digunakan untuk menentukan kategori *score SUS*. Kategori nilai susu ini dibagi berdasarkan rentang nilai yang diperoleh oleh siswa. Jika siswa memperoleh skor di atas 81, maka akan mendapatkan nilai A. Sedangkan untuk skor antara 68 hingga 81, siswa akan mendapatkan nilai B. Jika skor yang diperoleh tepat 68, maka nilai yang diberikan adalah C. Selanjutnya, skor antara 51 hingga 67 akan diberi nilai D. Terakhir, untuk siswa yang mendapatkan skor kurang dari 51 akan memperoleh nilai E.