

## **BAB II**

### **KAJIAN PUSTAKA**

#### **2.1 Landasan Teori**

Dalam penelitian ini membutuhkan beberapa teori. Landasan teori tersebut merupakan ciri metode ilmiah untuk memperoleh data yang digunakan sebagai tambahan referensi (masukan) bagi peneliti yang berkaitan dengan setiap variabel dalam penelitian. Adapun teori yang digunakan oleh peneliti dalam melakukan penelitian yaitu: *AGILE development, AI, UML, Web, HTML, MYSQL, PHP, AGILE* dan *Software* pendukung.

##### **2.1.1 Metode *AGILE***

Dalam proses pengembangan pada metode *AGILE* ini yang berfokus pada pengembangan perangkat lunak yang bersifat kompleks serta melakukan pengerjaan perangkat lunak yang memiliki fitur tertentu (Pratomo Nur et al., 2022) . Prinsip-prinsip dalam metode ini salah satunya adalah *scrum*, yang konsisten dengan prinsip-prinsip yang terkandung dalam metodologi perkembangan pesat alat-alat pengajaran kegiatan pengembangan perangkat lunak, seperti: prestasi, analisis, desain dan pengiriman (Ariesta et al., 2021). Metode *AGILE* yang digunakan sangat efektif dan fleksibel khususnya dalam sebuah perusahaan karena mampu membantu dalam memberikan solusi yang mudah untuk dicerna hal ini juga membutuhkan pengambilan keputusan yang cepat tanpa mengorbankan kualitas produk. Metode *AGILE* selalu berfokus pada sisi teknis pengembangan dalam

metode ini mampu menghasilkan *software* yang berkualitas tinggi. Setiap pengembangan dapat selalu ada tinjauan berkala untuk memperbaiki hal-hal yang perlu untuk diperbaiki.

Ada beberapa prinsip yang diterapkan dalam metode *AGILE* antar lain:

1. *Scrum methodology*

Dalam menjalankan proses pengerjaan ini, di mana metode ini berfokus pada pengembangan perangkat lunak yang bersifat kompleks. Perangkat *scrum* ini terdapat beberapa *sprints*. Di mana *sprints* tersebut adalah bagian dari beberapa proses kecil masing-masing yang terdapat fitur.

2. *Scaled AGILE framework (Safe)*

Salah satu bagian dari metode *AGILE* yang menjadi solusi karena adanya sebuah kerjasama antardivisi saat melaksanakan sebuah proyek.

3. *Lean software development(LSD)*

Dalam pengembangan perangkat lunak metode ini dirancang dengan tujuan untuk mengembangkan perangkat lunak dengan memakai jumlah SDM yang minim. Produk LSD yang diluncurkan dengan fitur yang terbatas sesuai dengan kebutuhan setiap konsumen.

4. Kanban

Tujuan daripada kanban ini yaitu melakukan suatu proses dimana hal tersebut dilakukan secara visual sehingga aktivitas kerja lebih mudah terpantau.

5. *Crystal methodology*

Bagian dari pada metode ini adalah metode yang lebih berfokus pada kondisi team yang bekerja dibandingkan dengan proses atau tool yang ada, fokus pada

bagian dari komunikasi team, interaksi, dokumentasi serta feedback. Dengan demikian pengembangan jauh lebih maksimal.

*AGILE* adalah suatu metode manajemen dalam suatu proyek khusus yang digunakan dalam pengembangan atau penerapan perangkat lunak. Metode ini dapat membantu untuk menangani dalam ketidakpastian setiap proses pengembangan perangkat lunak menggunakan langkah-langkah inkremental iteratif yang kerap disebut sebagai *sprint*. Ketika menerapkan metodologi *AGILE* atau dikenal sebagai pengembangan perangkat lunak *AGILE*. Hal itu dirancang baik secara kolaboratif, terstruktur dan terorganisir dengan partisipasi bagi setiap pihak yang bekerja. Dengan proyek menggunakan metode pengembangan *AGILE* memiliki kerangka kerja yang berprinsip selalu berhatap dan berulang(Nova et al., 2022). Untuk mencapai sebuah hasil proyek proses pengembangan yang dilakukan dapat berjalan dengan sangat cepat serta selesai dengan waktu yang tepat dan tentunya dapat memperoleh hasil yang kualitas tinggi. Setiap pengembangan perangkat lunak yang menggunakan *AGILE* jauh lebih *up-to-date* karena memiliki proses yang lebih sangat *fleksibel* serta selalu menyesuaikan dengan kebutuhan pengguna saat ini.

### **2.1.2 *AGILE Development***

Defenisi dari *AGILE development* adalah suatu cara dalam proses pengembangan suatu perangkat lunak yang dibuat secara cepat seiring dengan kondisi perubahan kebutuhan yang terjadi dalam waktu yang cukup relatif singkat. Selain itu dalam pengembangan *software* yang didasarkan pada proses pengerjaan yang dilakukan dengan tahap berulang di mana hal tersebut memiliki aturan serta solusi yang telah disetujui bersama (Aji, 2021). Pada konsep utama dalam *AGILE*

*Development* adalah pengerjaan aplikasi dan kerjasama sebuah team. Fokus pengerjaan aplikasi dengan meminimalisir dokumentasi. Kerjasama team berupa orang *programmer* atau lebih yang bekerja dalam satu fitur serta komunikasi intensif *programmer-klien*. *AGILE Development* berupa iterasi atau perulangan, tujuannya supaya lebih mudah untuk merespon dan mengatasi setiap perubahan secara *fleksibel*, sehingga mampu mengurangi waktu pengerjaan proyek dan mencapai kepuasan klien (Wisnumurti et al., 2022). Praktik *AGILE Development* cocok digunakan pada proyek yang skala kecil. Semua metodologi yang tergabung dalam *AGILE development* berlandaskan prinsip pada *AGILE manifesto*. Ada beberapa yang menjadi contoh metodologi dari *AGILE development* adalah *Extreme Programming (XP)* dan *SCRUM* (Zulvi, 2021). Dimana praktik *SCRUM* menitik beratkan manajemen team pada iterasi yang sering disebut dengan *sprint* proyek *SCRUM*. Faktor lain yang membedakan dengan XP adalah pada proyek *SCRUM* perubahan kebutuhan hanya bisa dilakukan pada akhir iterasi. XP disebut juga sebagai metodologi dengan siklus pengembangan pendek dimana iterasi ini dengan rentang waktu satu sampai dua minggu. Praktik XP didesain secara sederhana dengan berfokus pada pengerjaan program aplikasi dalam waktu relatif singkat.

Beberapa tujuan daripada *AGILE development* sebagai berikut:

1. *High value and working App system*

Tujuan ini digunakan untuk memperoleh sebuah *software* yang memiliki nilai jual atau pemasaran yang cukup tinggi tetapi menggunakan biaya pembuatan yang sekecil mungkin, tetapi produk atau *software* tersebut tetap memiliki kualitas yang tetap utuh dan menjadi prioritas terdepan (Simatupang & Pakpahan, 2022).

## 2. *Iterative, incremental, evolutionary*

*AGILE* adalah salah satu metode pengembangan *software* yang berjalan secara iteratif, berulang, serta dapat diubah ditengah jalan bila dibutuhkan. Untuk itulah mengapa *AGILE* disebut juga sebagai metode yang *fleksibel* dan dapat digunakan dalam jangka pendek.

## 3. *Cost control and value driven development*

Dengan dibuatnya *AGILE*, dapat dengan mudah disesuaikan dengan kebutuhan *user* atau pengguna. Selain itu team pengembang juga dapat mengontrol waktu serta biaya yang diperlukan sesuai kebutuhan para pengguna atau *user*.

## 4. *High quality production*

Menjaga supaya kualitas dari *software* menjadi sebuah hal utama yang dilakukan oleh pengembang sekalipun menggunakan biaya seminimal mungkin.

## 5. *Flexible and risk management*

Supaya perangkat lunak yang sedang dikembangkan dapat tetap terjaga, maka setiap yang berkaitan dengan jadwal pertemuan dapat diatur kapan saja. Selain dapat mengurangi kesalahan program dapat juga membantu dalam *fleksibel* waktu.

## 6. *Collaboration*

Dengan adanya kolaborasi dalam setiap team, hal tersebut dapat memberikan masukan untuk membahas setiap komenar dari *klien*. Untuk itu sangat diperlukan koordinasi serta komunikasi yang benar dan baik untuk semua anggota.

#### 7. *Self organizing, self managing teams*

Adapun yang menjadi bentuk fungsi dari *AGILE* tersebut sebagai manejer yang menjadi penghubung antara *developer dank lien*, dimana *developer* itu sendiri diberi akses untuk mengatur urusannya dengan *software* itu sendiri. Sehingga dapat mengurangi kemungkinan terjadinya *missing communication*. Dengan menggunakan *AGILE* dalam perusahaan sangat dapat membantu untuk mempercepat waktu, karena *AGILE* berfokus pada setiap pengembangan iterasinya, perusahaan yang menerapkan *AGILE* lebih mudah meliris atau layanan kepasar.

#### **2.1.3 Manfaat Metode *AGILE***

Metode *AGILE* memiliki banyak manfaat serta keuntungan, tidak hanya dalam pengembang, tetapi juga untuk pelanggan, pemasok, manajer produk, dan manajer proyek dengan *AGILE*, *vendor* dapat mengurangi pemborosan dengan memfokuskan upaya pengembangan pada fitur bernilai tinggi dan meningkatkan efisiensi. Akibatnya, pelanggan mungkin menemukan bahwa pemasok lebih responsif terhadap kebutuhan yang berkembang. Dalam team pengembangan, penggunaan yang *fleksibel* seperti *Scrum* dapat menguntungkan anggota team dengan sebuah tahap dapat mengurangi pekerjaan yang tidak produktif (Wisnumurti et al., 2022).

Selain daripada itu, manajer produk dan proyek juga mendapatkan manfaat dari penggunaan pendekatan pengembangan yang gesit. *Scrum* dapat memfasilitasi keselarasan antara pekerjaan dengan kebutuhan pelanggan dengan memberikan kesempatan untuk memprioritaskan pekerjaan dan memastikan nilai maksimum. Berikut merupakan beberapa manfaat menggunakan metodologi *AGILE* dalam pengembangan perangkat lunak:

1. Mempercepat waktu pengiriman

Metodologi *AGILE* memungkinkan pengembang untuk mengirimkan perangkat lunak dalam waktu yang lebih singkat melalui penggunaan iterasi dan pengujian berkelanjutan. Hal ini dapat memungkinkan perusahaan untuk merespons perubahan kebutuhan pelanggan secara lebih cepat dan tepat.

2. Meminimalkan risiko

*AGILE* dapat mempromosikan pemikiran yang lebih kreatif dan inovatif, sehingga memungkinkan pengembang untuk menguji gagasan secara lebih cepat dan meminimalkan risiko yang terjadi pada kegagalan proyek.

3. Memaksimalkan nilai pelanggan

*AGILE* juga berfokus pada pengiriman perangkat lunak yang dapat memberikan nilai tambah bagi setiap pelanggan. Hal tersebut dapat memungkinkan perusahaan untuk lebih dapat memahami setiap kebutuhan pelanggan dan memberikan produk yang lebih sesuai dengan kebutuhan mereka.

#### 4. Meningkatkan keterlibatan Team

*AGILE* sangat mendorong kolaborasi antara anggota team dengan pelanggan. Hal ini memungkinkan anggota untuk merasa lebih terlibat dan lebih bertanggung jawab atas kesuksesan sebuah proyek.

#### 5. Meningkatkan fleksibilitas

*AGILE* dapat meningkatkan perusahaan untuk merespons perubahan kebutuhan pelanggan dengan lebih cepat dan mengubah fokus proyek saat di perlukan. Hal ini memungkinkan perusahaan untuk lebih *fleksibel* dalam menghadapi tantangan bisnis yang terus mengalami perkembangan yang cukup pesat.

#### 6. Meningkatkan kualitas

*AGILE* mampu untuk mempromosikan pengujian berkelanjutan dan perbaikan berkelanjutan dalam pengembangan perangkat lunak. Supaya dapat memungkinkan perusahaan untuk lebih memperhatikan kualitas produk dan memastikan bahwa produk yang di kembangkan memiliki kinerja yang cukup optimal dan dapat di andalkan.

#### **2.1.4 Web**

*Web* merupakan sebuah istilah yang akrab di dunia modern saat ini. Di mana *web* salah satu bagian untuk memudahkan pekerjaan manusia di masa sekarang. Menurut perkiraan yang telah di telaah tidak sedikit atau hampir semua informasi sekarang ini bisa diakses dengan situs *web*. *Web* itu sendiri merupakan bagian yang sangat inti atau vital di internet (Vlasova & Bichkaeva, 2021).

*Web* atau *World Wide Web* (W3) merupakan sebuah sistem server pada internet yang mendukung setiap dokument di format secara khusus, di mana

dokument tersebut akan diformat ke dalam sebuah bahasa pemrograman seperti *HTML*. *HTML* tersebut akan mendukung pada dokument lain seperti berbentuk file grafik, video, gambar dan audio. Sejarahnya *web* awalnya dipublikasikan team Berners-lee pada tahun 1989, team ini mencoba bagaimana cara untuk bisa menemukan hal yang baru dari para ilmuwan dengan sebuah data yang mereka kerjakan. Seiring dengan perkembangan zaman yang cukup pesat dengan hal tersebut perkembangan *web* juga semakin pesat, semakin banyak pembaharuan yang ada pada *web* membuat para ahli membentuk peningkatan yang cukup baik terhadap *web*. *Web* membuat penemuan-penemuan yang baru dan semakin menarik dengan bentuk dan informasi yang jauh menarik

Domain menjadi alamat dalam sebuah situs *web*. Di mana bentuk utama atau dasar daripada domain ini kedalam penggunaan angka. Angka tersebut sering disebut dengan alamat IP. Beberapa tipe-tipe pada domain ini dibedakan berdasarkan setiap kegunaan dan fungsinya (Anggita, 2021).

### **2.1.5 Objek Penelitian**

PT.PCI Elektronik Internasional dikota Batam merupakan salah satu perusahaan PT.PCI *Limited* yang memproduksi barang elektronik, yang bergerak di bidang manufaktur elektronik dan memiliki banyak pengalaman dalam bidangnya selama 25 tahun. Perusahaan ini memiliki lebih dari 2000 karyawan dikota Batam berlokasi di Panbil. PT.PCI *limited* menawarkan banyak jasa untuk OEM global, layanan desain dan manufaktur berbagai bidang yang memberikan dampak yang sangat membangun dikota Batam seperti otomotif, otomasi, telekomunikasi, hingga peralatan medikal. Awal mula membangun karir dari

perusahaan manufaktur seperti *PCB* hingga *PCBA*, PT.PCI mengalami kemajuan yang kian pesat sampai saat ini dengan memberi penawaran jasa di bidang manufaktur elektronik dan *LCD*. PT.PCI mulanya berdiri pada tahun 1972 di *Sillicon valley, sanjose*. Seiring dengan teknologi yang semakin meningkat, sehingga PT. PCI berusaha mampu untuk mengikuti kemajuan teknologi dan berusaha menghasilkan produk yang unggul. PT.PCI membangun beberapa pabrik di negara indonesia untuk menunjang bisnis yang dapat berkembang dengan sangat cepat. Pada tahun 1990 menjadi awal mula PT.PCI mendirikan pabrik di kota batam yang menjadi salah satu perusahaan yang diperkirakan bisa mengidentifikasi batam menjadi salah satu tempat yang strategis serta hemat biaya untuk mendirikan perusahaan tersebut. Pada tahun 2000 PT.PCI sudah mendirikan pabrik shanghai karena memperkirakan china mampu menjadi sebagai pabrik dunia.

#### **2.1.6 Jenis *Component***

*Component elektronika* adalah alat berupa benda yang menjadi pendukung suatu rangkaian pada peralatan elektronik yang memiliki fungsi yang berbeda-beda dan tujuan yang berbeda. Jenis *component* yang memiliki bentuk maupun jenis yang berbeda-beda ada yang langsung menempel langsung dengan *PCB*, *Veroboard*, *Protoboard* dan *CCB* serta bentuk *component* yang tunggal.

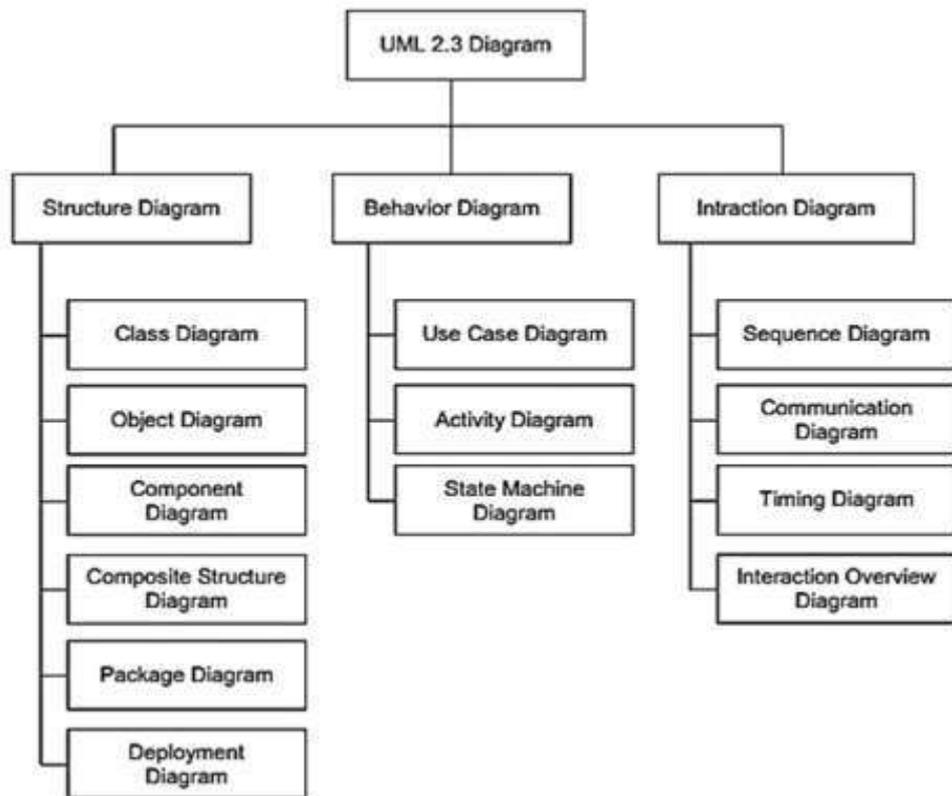
Secara umum, *component* tersebut dapat bagi menjadi dua bagian berdasarkan fungsi kerjanya yaitu *component* pasif dan *component* aktif.

1. *Component* pasif adalah *component elektronika* yang dapat beroperasi tanpa memerlukan arus atau tegangan listrik tambahan saat bekerja. Contoh *component* pasif yaitu resistor, kapasitor, induktor, dan trafo atau transformator.

2. *Component* aktif adalah *component elektronica* yang memerlukan arus atau tegangan internal (sumber tambahan) untuk dapat beroperasi. Komponen aktif ini dapat menguatkan dan menyearahkan arus listrik, komponen aktif juga dapat mengubah bentuk energi menjadi energi lain. Contoh *component* aktif adalah dioda, transistor, ic (*integrated circuit*).

### **2.1.9 UML (*Unified Modelling Language*)**

*UML* merupakan bahasa visual standar untuk pemodelan dan komunikasi tentang sistem yang menggunakan diagram serta dukungan sebuah teks. Selain itu pemanfaatan dari *UML* adalah untuk mendeskripsikan, mendesain, dan mendokumentasikan desain dalam sistem informasi, termasuk salah satunya antarmuka pengguna, sebuah aplikasi, sistem, tahap proses, dan masih banyak lagi (Khotimah et al., 2022). *UML* juga memungkinkan menggambarkan hubungan antar setiap objek yang saling berhubungan dalam sistem informasi. Hal tersebut berkontribusi dalam pengembang untuk menyajikan informasi kepada pelanggan atau pengguna akhir. Sehingga *UML* bermanfaat dalam memfasilitasi komunikasi bagi setiap pengguna serta pengembang saat mengerjakan sebuah proyek yang dikerjakan (Anggita, 2021).



**Gambar 2.1** *Diagram use case*

**Sumber:** (Maharani, 2018)

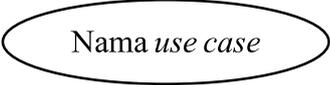
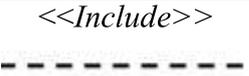
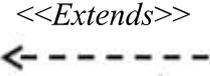
1. *Diagram use case*

*Diagram use case* adalah suatu bentuk struktur yang pada umumnya banyak dipakai dalam setiap struktur penelitian. Tahap tersebut adalah sebuah langkah yang menjelaskan gambaran atau interaksi sistem proses dengan lingkungannya. *Diagram use case* memaparkan setiap bagian yang menjadi aktivitas – aktivitas yang dilakukan oleh pengguna terhadap melalui sistem yang dirancang. *Diagram use case* menjadi tahapan sistem dalam mendesain dan mengerjakan proses dari bentuk fisik dengan menggunakan struktur. Sistem tersebut akan menjelaskan

uraian bagian dalam *diagram use case*, alur dari setiap proses pada *diagram use case* akan menjadi urutan aktor.

*Diagram use case* digunakan menjadi kasus persyaratan dalam mewakili sistem dan setiap penjabarannya dalam serangkaian aktivitas pengguna dan interaksi untuk tercapai sebuah tujuan.

**Tabel 2. 1** simbol pada *diagram use case*

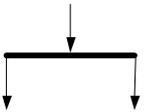
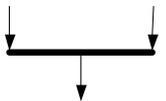
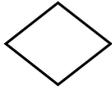
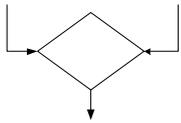
Simbol	Keterangan
	Nama <i>use case</i> Sebagai sebutan dalam <i>use case</i> atau yang dikenal dengan kata kerja yang diiringi dengan kata benda.
	<i>System boundary</i> Bagian yang mencakup lingkungan pada sistem.
	<i>Asosiasi boundary</i> Sebagai penghubung antara pengguna dengan <i>use case</i> yang saling berhubungan.
	<i>Include</i> Suatu proses tindakan yang harus dilakukan supaya memperoleh hasil dari <i>use case</i> .
	<i>Extends</i> Merupakan suatu tindakan yang tidak diwajibkan atau sering disebut dengan syarat dalam <i>use case</i> .

## 2. *Diagram activity*

Pada diagram ini menjelaskan setiap bagian aktivitas yang terjadi dalam sistem. Pada setiap proses dalam sistem tersebut yang berisikan rangkaian pada sistem, sistem tersebut akan melakukan proses penyelesaian baik aksi tindakan, pemilihan, reduplikasi pada *use case*. *Diagram activity* ini adalah kegiatan, peristiwa, status transmisi, dan objek. Dengan istilah lain bahwa kegiatan alur setiap proses

menggambarkan perilaku sistem tersebut. Dari diagram ini yang akan menjelaskan sebuah kegiatan atau setiap alur proses sistem, perlu diketahui bahwa diagram ini hanya untuk menggambarkan aktivitas sebuah sistem dan bukan sebagai aktor. Sehingga semua aktivitas atau secara keseluruhan dilakukan oleh sistem tersebut. Terdapat beberapa diagram aktivitas yang menggambarkan alur proses untuk setiap prosedur.

Tabel 2. 2 simbol *diagram activity*

Simbol	Keterangan
	<i>Start point</i> Menjadi bagian awal dari kegiatan dalam sistem.
	<i>End point</i> Bagian akhir dalam sistem.
	<i>Fork</i> Dalam setiap kegiatan proses bagian ini akan memberikan pilihan yang lain pada proses baik dalam persilangan atau mengkolaborasi sistem.
	<i>Join</i> Bagian ini menjelaskan proses menggabungkan antar sistem.
	<i>Decision points</i> Bagian menjelaskan penentu dalam membuat suatu tindakan pada proses sistem.
	<i>Merge</i> Pada proses kegiatan ini dimana <i>flow</i> yang sudah terpecahkan akan dikolaborasi dengan <i>flow</i> yang lain.
	<i>Swimlane</i> Bentuk kegiatan yang menjelaskan aktivitas yang terjadi dan aktor atau pengguna dalam kegiatan tersebut.

### 3. *Diagram sequence*

Tujuan pada diagram adalah sebuah *tool* yang banyak dipakai untuk membuat pengembangan sistem informasi yang berorientasi objek hal tersebut untuk memvisualkan interaksi sesama objek. Dari penjelasan tersebut dapat untuk simpulkan bahwa diagram sekuensial adalah suatu alat yang digunakan untuk pengembangan sebuah sistem. Bagian dari diagram yang menerangkan setiap alur dari proses *use case*.

*Diagram sequence* juga gambaran bentuk dalam pengembangan diagram, hal tersebut termasuk dalam suatu keharusan yang dibuat sesuai dengan perubahan yang terjadi dalam sistem baik dalam bentuk masukan ataupun keluaran yang diperoleh.

### 4. *Diagram class*

Diagram kelas merupakan suatu kegiatan yang menjelaskan setiap bagian pada sistem, dimana kelas tersebut memiliki atribut dan perilaku yang serupa.

Pada diagram ini dibagi menjadi tiga bagian yaitu:

- Nama
- Atribut
- Operasi

Pada bagian diagram tersebut membutuhkan sebuah sistem terhadap setiap bagian dalam struktur yang mendukung hubungan dalam objek. Pengelompokan pada diagram ini dibagi dalam beberapa struktur seperti:

a. *Class main*

*Class main* merupakan suatu tindakan awal dalam implementasi, sehingga *class main* dapat diproses.

b. *Class* pelaksanaan dalam sistem

*Class* yang mentransmisikan serta dapat mengatur tampilan kepada pengguna

c. Pendefinisian *use case* dalam mengambil *class*

Proses ini dibuat untuk melapisi serta bisa menjalankan suatu sistem, sistem tersebut akan dijadikan sesuai dengan *use case*.

## **2.2 Implementasi penginputan data *Component***

Data dapat diartikan sebagai kumpulan fakta atau catatan dari hasil pengamatan yang diperoleh baik dalam bentuk angka, kata-kata maupun citra yang menjadi sebuah acuan atau keterangan dengan tujuan mempermudah mencari suatu informasi. Dengan adanya data akan sangat membantu dalam setiap proses pekerjaan ataupun aktivitas. Data merupakan serangkaian sebuah fakta dan angka yang dapat digunakan menjadi salah satu bahan dalam membentuk atau menyusun sebuah informasi (suharsimi 2020). Data yang dimaksud dalam *component* yaitu bagian fakta yang ada dalam *component*, di mana fakta tersebut dapat menjadi tumpuan atau sumber informasi tentang sebuah *component*.

Data pada *component* tersebut dilengkapi dengan beberapa bagian:

1. *Part Number*

*Part number* merupakan nama atau sering disebut dengan sebutan pada *component*.

2. *Batch*

Bagian identitas asli pada *component*.

3. Lokasi

Tempat dari pada *component*.

4. Jumlah *component*

5. Tanggal dan waktu

6. Nama karyawan

### **2.3 Software Pendukung**

Dalam proses membangun *website* penginputan data *component* sangat membutuhkan sebuah *software* pendukung dengan demikian dapat menghasilkan sebuah aplikasi yang di inginkan, seperti *PHP*, *HTML*, dan *MSQL*.

#### **2.2.1 PHP (Hypertext Preprocessor)**

Pemrograman *PHP* umumnya dipakai dalam membuat maupun mengembangkan sebuah aplikasi *web*, salah satunya dalam membuat sebuah aplikasi *web* dinamis, situs *database*, dan berbagai aplikasi berbasis *web* (Aji, 2021). Pemrograman ini awal berdiri yang dibuat oleh seorang pemogram yang bernama Rasmus Lerdorf di tahun 1994, sejak masa itu banyak implementasi dalam pengembang yang berpartisipasi dalam mengembangkan *PHP* tersebut. *PHP* ini bekerja di bagian pada *server web*(Henry Februariyanti, 2012), namun dapat juga bekerja di sistem lokal. *PHP* juga sering dipakai untuk berbagai bentuk atau jenis basis data yang berbeda yang sering ditemui adalah *MySQL*, *PostgreSQL*, *Oracle*, dan *Microsoft SQL Server*. *PHP* yang dirancang dapat dipakai dalam merancang

sebuah penerapan seperti bentuk teks maupun bentuk grafis, supaya dapat dengan mudah diakses dan mengelola informasi dari sebuah *file* sistem.



Gambar 2. 2 Logo *PHP*

### **2.2.2 *HTML (hyper text markup language)***

Bahasa pemrograman *HTML* atau disebut dengan (*hyper text markup language*) menggunakan kurung kurawal yang dimulai dengan  $\langle \rangle$  untuk dapat menampilkan sebuah informasi di halaman *web*. Dengan tag ini dapat menentukan bagaimana sebuah informasi bisa di tempatkan serta di tampilkan di halaman *web*. *HTML* memiliki berbagai jenis tag untuk bisa menampilkan informasi, yang sering kita temui seperti pada tag dalam membuat sebuah judul, membuat paragraf, menyisipkan gambar, membuat tabel, dan lain-lain.



Gambar 2. 3 Logo *HTML*

### 2.3.3 *MYSQL*

*MySQL* adalah sebuah implementasi dari *relational database management system* (RDBMS) yang digunakan untuk dapat mengontrol akses data, menyediakan akses data yang cepat, dan memastikan hanya pengguna memiliki wewenang untuk dapat mengakses basis data. *MySQL* menggunakan *Structured Query Language* (SQL). *MySQL* tersedia dalam dua lisensi dalam penggunaannya. Jika aplikasi tidak ingin didistribusikan, maka lisensi berupa *open source*. Akan tetapi jika aplikasi ingin didistribusikan, maka lisensi dapat berupa lisensi komersial.



Gambar 2. 4 Logo *MSQL*

### 2.3.4 *Xampp*

*Xampp* sebuah perangkat lunak yang sering ditemui dan digunakan banyak pengguna, perangkat lunak yang bebas digunakan dan sangat mendukung dalam berbagai sistem operasi yang mampu menyatu dengan beberapa program. *Xampp* memiliki fungsi dalam perangkat lunak yang berperan sebagai server yang berdiri sendiri, dapat menjalankan web yang berbasis *PHP* serta dapat menggunakan pengolah *MYSQL* di sebuah computer local (Pratomo Nur et al., 2022).



**Gambar 2. 5 Logo *Xampp***

### 2.4 Penelitian Terdahulu

Penelitian terdahulu menjadi sebuah sarana dalam mendukung penelitian dengan mencari penelitian yang dilakukan oleh peneliti sebelumnya dengan tujuan supaya dapat menemukan pemikiran yang baru bagi penelitian yang baru oleh peneliti. Adapun beberapa bahan yang menjadi sumber penelitian terdahulu bagi peneliti adalah sebagai berikut:

1. Penelitian ini yang dirancang oleh (Arbian Sulistyو et al., 2020) yang berjudul “ Metode *AGILE* dalam perancangan sistem prediksi prevansi stunting di indonesia “ berdasarkan penelitian ini menarik kesimpulan Perancangan sistem prediksi prevalensi stunting dengan metode *AGILE* harus dilakukan beberapa

kali selama proses penentuan kebutuhan. Metode ini sangat cocok untuk digunakan, karena perubahan yang diperlukan dalam desain sistem ini dapat dilakukan secara sering dalam waktu yang singkat. Dengan menggunakan metodologi *AGILE*, sistem menjadi lebih sederhana dan dapat dikembangkan sesuai kebutuhan. Hasil perancangan menunjukkan bahwa sistem dapat mengolah data prakiraan dan dapat menampilkan data yang dialokasikan untuk setiap provinsi.

2. Berdasarkan penelitian yang telah dilakukan oleh (Hikmah, 2020) dengan judul “Metode *AGILE* untuk meningkatkan kreativitas guru melalui berbagai pengetahuan” menarik kesimpulan tenaga kependidikan. Metode gesit dalam metode *AGILE* juga berfungsi diterapkan dalam pembelajaran di kelas Proses pembelajaran menciptakan individu kreatif, kritis, bersemangat, berinovasi, menemukan dapat beradaptasi menghadapi perubahan permanen yang selalu begitu. Untuk itu diperlukan guru mengetahui metodenya yang lincah. Berbagi ilmu (ilmu Sharing) juga dapat diterapkan di dalam kelas, di mana anak-anak memiliki pengetahuan sebelumnya dapat dibagikan pada teman melalui forum diskusi atau instruksi rekan. Mengolah kreativitas anak sangat penting guru kreatif, lalu strategi berbagi pengetahuan ini bahkan dapat dilanjutkan mungkin pada tingkat yang lebih luas sebagai gantinya hanya di tingkat sekolah.
3. Berdasarkan penelitian yang telah dilakukan oleh (Ferry Qadafi & Wahyudi, 2020) “ sistem informasi inventory gudang dalam “ketersediaan *stok* barang menggunakan metode *buffer stok* “ dapat menarik kesimpulan berdasarkan pembahasan sistem informasi *inventory* gudang dalam ketersediaan stok barang

menggunakan metode *buffer stock* berbasis *web* dapat diambil kesimpulan yaitu hasil yang didapat adalah proses persediaan barang dengan metode *buffer stock* dapat membantu *konveksi* untuk mengetahui jumlah persediaan barang atau bahan yang ada dan dapat mengetahui jumlah persediaan yang harus di pesan berdasarkan perhitungan *buffer stock*.

4. Berdasarkan penelitian yang telah dilakukan oleh (Mahendra, Deny Tresno Eby Yanto, 2018.). Memberikan sebuah kesimpulan konsep *AGILE Software Development* dicetuskan oleh Kent Beck dan 16 rekannya dengan menyatakan bahwa *AGILE software development* adalah sebuah tahap untuk membangun *software* melakukannya dan membantu orang lain membangunnya sekaligus. *AGILE software development methods* atau *AGILE methodology* merupakan sekumpulan metodologi pengembangan perangkat lunak yang berbasis pada pengembangan iteratif, di mana suatu persyaratan dan solusi yang dapat berkembang melalui kolaborasi antar team yang terorganisir.
5. Berdasarkan penelitian yang telah dilakukan oleh “pemanfaatan metode *AGILE ui prototyping* dalam perancangan aplikasi *e-learning* perguruan tinggi menarik (Rudi & Robert, 2021) kesimpulan aplikasi hasil dari perancangan ini membantu memudahkan kegiatan dosen dan mahasiswa dalam kelas yang dicakup hanya dalam satu aplikasi. Aplikasi hasil dari perancangan ini telah dilakukan analisis dan perancangan saat menentukan fitur sehingga tidak ada lagi kelebihan atau kekurangan fitur. Aplikasi hasil dari perancangan ini telah dilakukan pengujian untuk memastikan fitur yang berjalan memenuhi kebutuhan pengguna. Aplikasi

hasil dari perancangan ini menyesuaikan dengan perkembangan teknologi saat ini yang dimanfaatkan dengan kegiatan positif.

6. Berdasarkan penelitian yang telah dilakukan oleh (Muslim & Retno, 2018) “ implementasi cloud computing menggunakan metode pengembangan sistem *AGILE* “ menarik kesimpulan implementasi *cloud computing* di PT. Erudeye Indonesia dibuat dengan melakukan proses *deploying* menggunakan metode pengembangan sistem *AGILE development methods*, dengan langkah meliputi perencanaan, implementasi, pengujian, dokumentasi, *deployment* dan pemeliharaan. Proses *provisioning* pada sistem *cloud hosting* di PT. Erudeye Indonesia dibedakan menjadi dua macam *provisioning*, yaitu *provisioning* pada manajemen portal untuk admin dan manajemen portal. *Provisioning* manajemen portal untuk admin digunakan untuk mempersiapkan dan membuat plan atau menu untuk customer sedangkan pada *provisioning management* portal untuk tenant digunakan untuk memilih plan atau menu yang bisa digunakan untuk membuat *database*.
7. Berdasarkan penelitian yang telah dilakukan oleh (Nurzaman, 2021 ) “ pengembangan sistem otomasi tagihan menggunakan metode *AGILE software development* “ menarik kesimpulan berdasarkan penelitian yang telah sebenarnya bisa ditarik kesimpulan sebagai berikut, yaitu dalam pengembangan sistem pembayaran otomatis dibagi menjadi beberapa tahap perkembangan, yaitu simpanan produk, simpanan lari cepat, lari cepat rencanakan, *sprint*, lalu evaluasi dan mengingat. Sistem pembayaran otomatis yang fungsi utamanya adalah penagihan secara elektronik, membuat lampiran sebagai *file spreadsheet*

dan kirim faktur ke *email* pelanggan otomatis. Setiap fungsi atau tugas sistem faktur otomatis saja membutuhkan satu proses sprint dalam delapan hari. Hasil implementasi sistem faktur otomatis membuat bisnis anda lebih mudah dalam pembuatan dan pengiriman tagihan untuk diproses oleh pelanggan menjadi efektif dan efisien. Pengembangan sistem lebih lanjut dapat dilakukan dengan memberi beberapa fungsi tambahan seperti *file* lampiran sebagai *file pdf* dan notifikasi *email* otomatis untuk faktur jatuh tempo.

## 2.5 Kerangka Pemikiran

Metode kerangka pemikiran ini menekankan bagaimana penerapan berbagai faktor yang ada dengan mendefinisikan sebagai sebuah masalah yang penting. Pada proses mengidentifikasi kerangka pemikiran dalam penginputan data *component* yang memiliki perbedaan tetapi dapat berkaitan dan berinteraksi satu dengan yang lain, hal tersebut dimungkinkan untuk dapat menganalisis dari setiap situasi (kondisi) yang secara lebih efektif serta berpikir secara kritis tentang pemecahan masalah. Jenis penginputan data yang manual yang sudah diterapkan dari pihak perusahaan tetapi kurang menarik seperti menggunakan *computer*. Kemudian *website* tersebut dipasang pada *computer* sebagai media penginputan data.

