

## **BAB II TINJAUAN PUSTAKA**

### **2.1 Landasan Teori**

#### **2.1.1 *E-Procurement***

“*E-Procurement* atau pengadaan elektronik merupakan sistem yang memanfaatkan teknologi informasi untuk membantu menjalankan proses pengadaan barang dan jasa secara elektronik, Sistem ini mencakup berbagai proses seperti permintaan penawaran, seleksi pemasok, negosiasi harga, dan pembuatan kontrak yang semuanya dilakukan secara digital” menurut (Gartner 2020),. Sistem ini mempunyai beberapa tujuan seperti meningkatkan efisiensi, transparansi dan akuntabilitas dalam proses pengadaan yang dilakukan. “E-Procurement memberikan banyak penawaran yang akan menjadi keuntungan untuk penghematan biaya operasional, peningkatan keamanan dan akurasi data serta peningkatan kecepatan proses pengadaan” menurut (Hackett Group, 2021).

“Penerapan *E-Procurement* memiliki dampak signifikan dalam mengurangi lead time dan biaya pengadaan, serta meningkatkan transparansi dalam proses pengadaan barang dan jasa. Penelitian ini juga menunjukkan bahwa perusahaan yang telah beralih ke sistem *E-Procurement* mengalami pengurangan rata-rata waktu pengadaan hingga 35%, yang berdampak langsung pada peningkatan efisiensi operasional. Selain itu, implementasi *E-Procurement* membantu mengurangi risiko kesalahan administratif, karena sistem digital mampu melakukan validasi data secara otomatis dan menyediakan catatan transaksi yang transparan dan terintegrasi” menurut (Septiawan, 2018).

#### **A. Komponen dan Proses *E-Procurement***

*E-Procurement* terdiri dari beberapa komponen dan proses utama, termasuk:

##### **1. Permintaan Penawaran**

“Pengguna bisa membuat penawaran permintaan kepada pemasok atau *supplier*. Sistem menyediakan penyampaian spesifikasi barang atau jasa yang dibutuhkan oleh pengguna dan menerima beberapa penawaran dari berbagai pemasok secara terpusat” menurut (Hackett Group, 2021).

## 2. Evaluasi dan Seleksi Pemasok

“*E-Procurement* memfasilitasi evaluasi penawaran dan pemilihan pemasok melalui fitur-fitur seperti analisis perbandingan harga, kualitas, dan waktu pengiriman. Ini juga berdasarkan kriteria yang telah ditetapkan sebelumnya” (Aberdeen Group, 2022).

## 3. Negosiasi Harga dan Kontrak

“Setelah penawaran diterima, sistem *E-Procurement* menyediakan platform untuk negosiasi harga dan syarat kontrak dengan pemasok. Fitur ini memungkinkan komunikasi yang lebih cepat dan efisien dibandingkan dengan metode tradisional” (Gartner, 2020).

## 4. Pelacakan dan Audit

“Semua transaksi dalam sistem *E-Procurement* terekam secara digital, memudahkan pelacakan dan audit untuk memastikan transparansi dan kepatuhan terhadap kebijakan perusahaan” (Aberdeen Group, 2022).

### 2.1.3 Implementasi dan Tantangan *E-Procurement*

Meskipun banyak manfaat yang ditawarkan, implementasi *E-Procurement* juga menghadapi berbagai tantangan, antara lain:

- **Resistensi terhadap Perubahan**

“Pengguna internal dan pemasok mungkin menghadapi kesulitan dalam beradaptasi dengan sistem baru, yang memerlukan pelatihan dan perubahan dalam cara kerja yang telah ada sebelumnya” (Laudon & Laudon, 2020).

- **Keamanan Data**

“Melindungi data yang dikelola melalui sistem *E-Procurement* adalah hal yang krusial. Implementasi harus mempertimbangkan aspek

keamanan untuk melindungi informasi sensitif dari akses yang tidak sah” (Gartner, 2020).

- **Biaya Implementasi Awal**

“Biaya untuk implementasi sistem *E-Procurement*, termasuk perangkat keras, perangkat lunak, dan pelatihan, bisa menjadi tantangan bagi perusahaan, terutama bagi perusahaan kecil dan menengah” (Hackett Group, 2021).

## 2.2 Teori Khusus

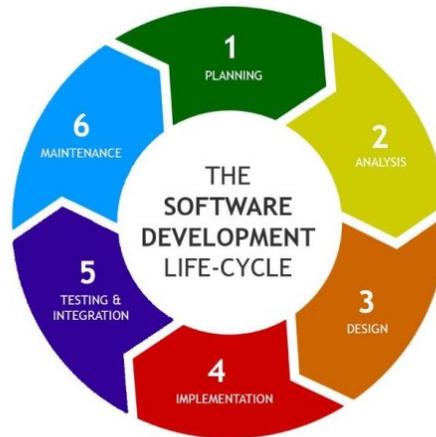
### 2.2.1 *Software Development Life Cycle (SDLC)*

SDLC adalah siklus pengembangan aplikasi atau sistem dengan tahapan-tahapan yang fungsional dan terstruktur guna mendukung jalannya pembuatan software. Kerangka kerja ini mempunyai tahapan-tahapan yang saling terhubung untuk memberikan penjabaran kepada proses pengembangan aplikasi dan juga agar meminimalkan kesalahan-kesalahan yang terjadi.

“Menurut SDLC adalah kerangka kerja sistematis yang digunakan untuk merancang, mengembangkan, dan memelihara perangkat lunak dengan tahapan yang terstruktur. Setiap fase—mulai dari analisis kebutuhan hingga pemeliharaan—bertujuan untuk memastikan kualitas perangkat lunak yang tinggi dan memenuhi kebutuhan pengguna menurut” (Sommerville, 2011),.

“Menjelaskan bahwa penerapan model SDLC yang tepat sangat berperan dalam keberhasilan pengembangan perangkat lunak, dengan memastikan setiap fase dieksekusi dengan benar dan efisien. Hal ini juga membantu mengurangi risiko yang mungkin timbul selama pengembangan, serta meningkatkan kolaborasi antar tim pengembang dan pemangku kepentingan. Selain itu, dengan mengikuti tahapan SDLC yang sistematis, proyek perangkat lunak dapat diselesaikan sesuai anggaran dan tenggat waktu yang telah ditentukan dari perencanaan” menurut (Suryanto, 2021).

Selanjutnya adalah gambar dari proses *software development life cycle* (SDLC) gambar ini juga memberikan penjelasan singkat dari tahapan SDLC yang ada.



Gambar 2. 1 Software Development Life Cycle

Sumber : <https://ids.ac.id/pengertian-sdlc/>

Selanjutnya adalah penjelasan dari 6 tahap yang ada pada metode software development life cycle ini :

1. Perencanaan (*planning*)

Pertama-tama untuk mengarahkan tujuan dari jalannya proses pengembangan suatu aplikasi adalah perencanaan. Perencanaan adalah tahap untuk mencari sumber informasi, membuat rencana dan mencari sumber daya yang mencukupi untuk membantu proses pengembangan

2. Analisis (*analysis*)

Setelah mendapatkan sumber informasi dan tujuan, selanjutnya adalah melakukan analisis yang detail terhadap aplikasi. Tujuan dari tahap ini adalah merumuskan dan memenuhi kebutuhan pada user yang akan dipenuhi oleh pengembang.

3. Desain (*design*)

Melakukan proses visualisasi proyek yang dikembangkan setelah melakukan analisis terhadap data yang di dapatkan. Tahap ini memberikan informasi terkait tampilan proyek aplikasi ke user, proses ini bisa menjadi titik komunikasi dari user kepada pengembang untuk memberikan feedback ke aplikasi.

#### 4. Implementasi (*implementation*)

Membuat dan melanjutkan aplikasi yang dibuat berdasarkan desain yang sudah disepakati. Menyiapkan beberapa dokumen terkait aplikasi, perencanaan proses pengujian, menulis kode dan mengatur bagaimana fitur berjalan dengan baik

#### 5. Pengujian (*testing*)

Lalu ada proses pengujian juga berperan penting guna memastikan fitur-fitur pada proyek berjalan. Menguji komabilitas dari target perangkat yang dibutuhkan dan memastikan semuanya berjalan dengan maksimal, perangkat lunak yang di uji juga harus sesuai kebutuhan fungsional yang sudah dirumuskan.

#### 6. Pemeliharaan (*maintenance*)

Terakhir, tahap pemeliharaan menjelaskan bagaimana langkah untuk menjalankan aplikasi dan melakukan pengecekan rutin terhadap aplikasi untuk menjaga sistem berjalan dengan baik

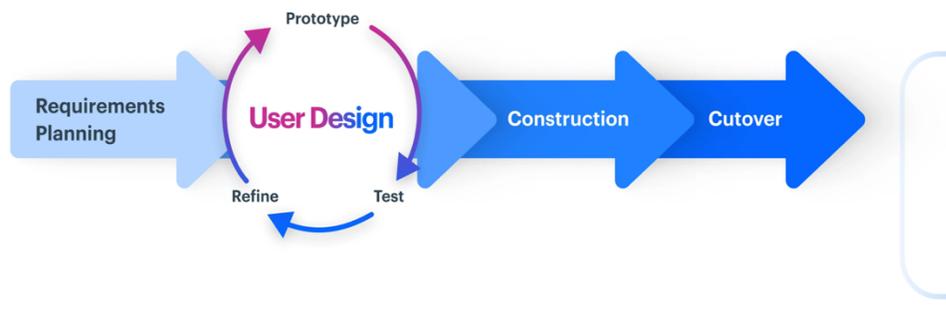
### **2.2.2 Rapid Application Development (RAD)**

“Rapid Application Development (RAD) merupakan metode pengembangan perangkat lunak yang menekankan siklus pengembangan yang cepat dengan menggunakan prototipe yang berulang dan melibatkan pengguna secara aktif selama proses pengembangan. Pendekatan ini memungkinkan pengembang untuk mengakomodasi perubahan kebutuhan pengguna dengan cepat tanpa mengorbankan kualitas sistem.” menurut (Maheshwari dan Jain, 2013).

“menyebutkan bahwa RAD mempersingkat waktu pengembangan dengan mengurangi proses dokumentasi dan berfokus pada hasil yang dapat diuji secara langsung oleh pengguna” menurut (Whitten, Bentley, dan Dittman 2001). “Hal ini sangat membantu perusahaan dalam menghadapi lingkungan bisnis yang dinamis dan *Rapid Application Development* (RAD) adalah salah satu metodologi yang pertama kali muncul untuk menjawab masalah kelambatan penyelesaian pengembangan sistem informasi pada *SDLC* konvensional” menurut (Sarosa, 2017), Selanjutnya *Rapid Application Development* (RAD) adalah model proses pengembangan perangkat lunak yang bersifat inkremental terutama untuk waktu pengerjaan yang pendek). Dari kedua penjelasan di atas, dapat disimpulkan bahwa teknik *Rapid Application Development* (RAD) merupakan Teknik pembuatan aplikasi yang mempunyai kelebihan dalam mengurangi durasi waktu pengembangan dengan tahap yang struktural dan sistematis.

- “Metode ini bisa membuat tim pengembang menjadi terbagi ke beberapa tim untuk membuat beberapa modul atau komponen aplikasi” menurut (Rosa dan Shalahuddin 2018). Berikut adalah gambar dari proses metode pendekatan *Rapid Application Development* (RAD)

## Rapid Application Development (RAD)



Gambar 2. 2 Rapid Application Development

Sumber : <https://kissflow.com/application-development/rad/rapid-application-development/>

Dari gambar diatas kita dapat melihat proses dari metode Rapid Application Development dengan mengutamakan feedback, prototype dan testing. Ringkasan dari semua tahap metode Rapid Application Development (RAD) sebagai berikut:

a. Perencanaan Kebutuhan (*Requirements Planning*)

Tahap pertama user dan tim pengembang akan bertemu untuk diskusi terkait masalah yang dialami oleh, lalu tim pengembang memberikan solusi kepada user dengan membangun aplikasi yang sesuai kebutuhan. Tahap ini mendukung tujuan utama dan memaksimalkan tingkat keberhasilan.

b. *User Design*

Selanjutnya ada tahap *user design* yang merupakan tahap pembeda dari metode lain. Tahap ini biasanya diawali dengan membuat *prototype* dan lalu diberikan kepada user. Setelah itu pembuatan desain akan berjalan sesuai rencana awal dan jika tidak tim pengembang akan melakukan perbaikan.

c. Pembangunan (*Construction*)

Setelah selesai mendapatkan perbaikan tim pengembang melakukan pembuatan sistem yang sesuai kebutuhan user. Tim pengembang juga melakukan tahap pengujian untuk setiap modul atau fitur agar berjalan dengan baik, langkah pertama dari pengembangan biasanya adalah membuat database untuk memperlancar proses penarikan data. Berikutnya tim front-end dan back-end akan melakukan pengembangan sistem berdasarkan kebutuhan dan database yang sudah dibuat

d. Pengujian (*Cutover*)

Lalu terakhir adalah melakukan pengujian pada sistem setelah melakukan pengembangan, pengujian ini melingkupi semua modul yang sudah direncanakan sebelumnya. Pengujian seperti, optimalisasi, kompatibilitas dan fungsional modul yang ada di aplikasi. Pelatihan juga akan diberikan kepada user untuk memberikan pengetahuan tentang sistem yang dibuat.

### **2.2.3 Object Oriented Analysis and Desain (OOAD)**

OOAD adalah salah satu metode pengembangan yang cukup sistematis dalam suatu pengembangan aplikasi yang berfokus pada representasi dengan menggunakan objek. Metode ini sering digunakan untuk memberikan penjelasan kepada user bagaimana aplikasi akan berjalan, mencakup tujuan utama dari pembuatan. Memiliki banyak model visualisasi dan beberapa objek seperti data (atribut) dan juga perilaku (metode).

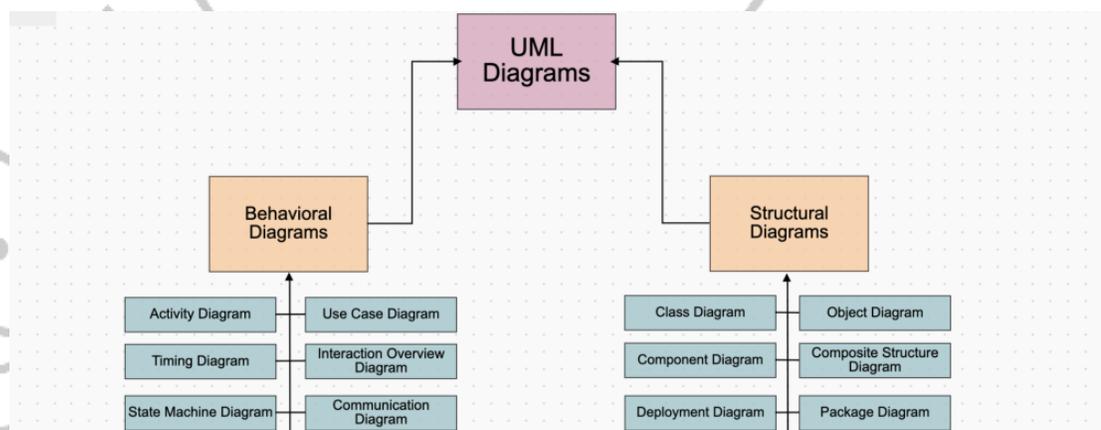
“OOAD memungkinkan perancang perangkat lunak untuk memodelkan sistem dengan lebih mendekati representasi dunia nyata. Proses ini melibatkan identifikasi objek, atribut, dan metode yang relevan, serta mendefinisikan hubungan antarobjek. Penelitian ini juga mencatat bahwa OOAD dapat membantu meningkatkan efisiensi dan keakuratan dalam pengembangan perangkat lunak dengan meminimalkan redundansi kode” menurut (Momin, 2016),.

dalam Object-Oriented Analysis and Design (OOAD), keterlibatan tim pengembang sangat penting untuk memastikan bahwa sistem yang dirancang sesuai dengan kebutuhan pengguna. “Aktivitas dalam OOAD mencakup identifikasi objek dan hubungan antarobjek, pembuatan diagram UML, diskusi untuk menyelaraskan model desain, serta pengujian terhadap skenario penggunaan” menurut (Larman, 32),. Dengan adanya partisipasi aktif, seperti melakukan analisis bersama, mengadakan sesi diskusi kelompok, dan memvalidasi desain secara iteratif, pengembang dapat memastikan bahwa solusi yang dihasilkan tidak hanya memenuhi kebutuhan teknis tetapi juga tujuan bisnis dari sistem yang dikembangkan.

### **2.2.4 Unified Model Language (UML)**

“UML telah menjadi standar de facto dalam pemodelan sistem perangkat lunak, namun penelitian terbaru menunjukkan bahwa eksplorasi ilmiah lebih lanjut dalam teknik pemodelan dapat memperkuat UML sebagai bahasa pemodelan terkemuka.

Studi ini menggunakan teknik pemodelan *thinging machine* (TM) untuk menganalisis UML, dengan tujuan memahami lebih dalam konsep seperti *events*, *objects*, *actions*, dan *activities*” menurut (Al-Fedaghi, 2021),. Pendekatan TM menawarkan metode konseptualisasi yang berbeda, yang dapat melengkapi aspek-aspek tertentu dalam metodologi UML untuk pengembangan dan desain sistem perangkat lunak. Berikut adalah gambar dari beberapa penjelasan struktur dan tipe diagram yang ada pada UML :



Gambar 2. 3 Unified Model Language

Sumber : <https://ideascale.com/blog/uml-diagram-definition/>

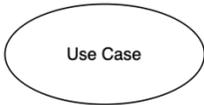
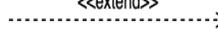
Selanjutnya ada beberapa penjelasan yang sudah di ringkas mengenai apa saja yang ada di UML yaitu :

1. *Behaviour Diagram* adalah kumpulan diagram yang berfokus menjelaskan perilaku pada sistem yang akan dibuat, selain itu ini juga bisa digunakan untuk memodelkan aspek yang dinamis seperti bagaimana objek berinteraksi.
2. *Structural Diagram* adalah kumpulan diagram yang memodelkan aspek statis, seperti bagaimana elemen dan komponen sistem diatur untuk memenuhi kebutuhan sistem yang akan dibuat. Contoh elemen-elemennya seperti kelas, struktural dan komponen

### 2.2.5 Usecase Diagram

*Usecase diagram* adalah diagram untuk menampilkan hubungan antara aktor atau pengguna dalam suatu sistem yang dibangun, diagram ini dibuat untuk menjelaskan kepada user mengenai jalannya aplikasi. Diagram ini cukup simpel namun sangat informatif karena mengandung beberapa informasi penting terkait aktivitas dan alur kerja pengguna pada sistem yang dibuat. Diagram ini menampilkan hubungan antara aktor dan objek dengan sederhana dengan simbol-simbol yang mudah dipahami, berikut adalah tabel penjelasan dari komponen *usecase diagram* :

Tabel 2. 1 Simbol Usecase Diagram

Simbol	Keterangan
 Actor	Aktor : menjelaskan peran dari pengguna yang terhubung dengan sistem yang dibuat
 Use Case	<i>Usecase</i> : informasi yang berisikan interaksi antara sistem dengan aktor
	Association : penghubung antara aktor dengan usecase untuk menjelaskan aktivitas aktor, dan memberikan informasi tentang fungsi-fungsi yang bisa di lakukan aktor
 <<include>>	Include : merupakan status dari usecase yang berarti bahwa aktivitas membutuhkan usecase lain untuk melakukan usecase berikutnya
 <<extend>>	Extend : mewakili interaksi yang opsional dan ekspansif dari usecase utama

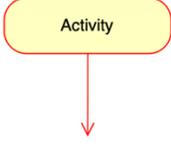
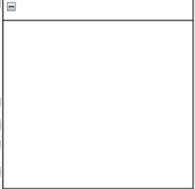
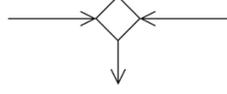
### 2.2.6 Activity Diagram

“*Activity Diagram* cocok digunakan untuk memodelkan aliran aktivitas dalam proses yang kompleks, terutama ketika banyak kondisi dan keputusan yang harus dipertimbangkan. Mereka juga menekankan pentingnya diagram ini dalam mendefinisikan interaksi antar komponen dalam sistem, termasuk skenario yang melibatkan proses otomatis dan manual” menurut (Miles dan Hamilton, 2006).

“*Activity Diagram* adalah salah satu diagram dalam Unified Modeling Language (UML) yang digunakan untuk memodelkan alur kerja atau proses bisnis dalam suatu sistem. Diagram ini merepresentasikan langkah-langkah aktivitas secara berurutan, termasuk percabangan logika, pengambilan keputusan, dan paralelisme proses” menurut (Ambler, 2023). Elemen-elemen penting dalam *Activity Diagram* meliputi aktivitas (*activity*), alur kendali (*control flow*), simpul keputusan (*decision node*), dan simpul awal/akhir (*initial node* dan *final node*). Dengan *Activity Diagram*, pengembang dan pemangku kepentingan dapat memahami bagaimana proses dalam sistem berjalan dan bagaimana sistem merespons kondisi tertentu, sehingga mempermudah identifikasi area yang membutuhkan perbaikan atau optimalisasi. Dibawah ini merupakan tabel notasi *activity diagram*, yang berisi informasi fungsi dan tujuan dari simbo-simbol yang tersedia:

Tabel 2. 2 Notasi *Activity Diagram*

Simbol	Keterangan
	<i>Initial node</i> : menandakan awal dari proses kerja suatu sistem dan hanya ada satu pada setiap <i>activity diagram</i> .
	<i>Final Node</i> : menginformasikan akhir dari alur kegiatan yang dijalankan pada sistem

	<p><i>Activity</i> : mewakilkan aktivitas yang dilakukan pada proses bisnis, dimana terdapat penjelasan terkait nama aktivitas pada simbol ini</p>
	<p><i>Control Flow</i> : menunjukan langkah berikutnya setelah melakukan aktivitas dari aktivitas sebelumnya.</p>
	<p><i>Fork Node</i> : menandakan aktivitas yang dilakukan mempunyai percabangan, simbol ini harus terhubung dengan simbol akhir yaitu join node.</p>
	<p><i>Join Node</i> : menandakan akhir aktivitas dari sistem yang mempunyai percabangan, simbol ini harus diawali dengan simbol fork node.</p>
	<p><i>Decision Node</i> : mewakilkan kondisi dari percabangan yang dihasilkan dari keputusan user.</p>
	<p><i>Swimlane</i> : merupakan pemisahan masing-masing aktivitas diagram dari aktor yang bertanggung jawab untuk kegiatan tersebut</p>
	<p><i>Merge Node</i> : Penyatuan kembali dari aktivitas percabangan yang ada pada activity diagram</p>

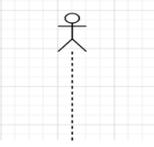
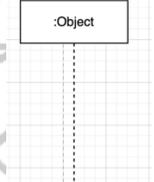
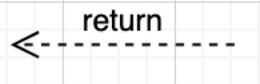
### 2.2.7 Sequence Diagram

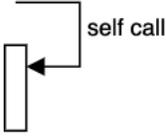
*Sequence diagram* adalah kumpulan diagram yang mewakili interaksi antar objek yang dilakukan oleh aktor kepada aplikasi. Objek-objek diagram ini bisa dibuat dengan *horizontal* dan *vertical*, diagram ini juga berisi perintah-perintah dari suatu modul. Tujuan utama diagram ini adalah menjelaskan urutan aktivitas dan memperlihatkan output yang

didapatkan setelah aktivitas tersebut. Berikutnya adalah tabel yang berisi objek yang tersedia pada *sequence diagram*

“Sequence diagram adalah salah satu alat utama dalam pendekatan analisis dan desain berbasis objek. Diagram ini membantu mengidentifikasi alur pesan antar objek selama eksekusi kasus penggunaan tertentu. Penulis juga mencatat bahwa sequence diagram sering digunakan untuk menemukan masalah dalam alur komunikasi sistem yang kompleks sebelum pengkodean dimulai, sehingga dapat mengurangi biaya perbaikan di kemudian hari” menurut (Dennis et al, 2015).

Tabel 2. 3 Notasi Sequence Diagram

Simbol	Keterangan
	<p><i>Actor</i>: mewakili entitas yang akan berinteraksi dengan sistem, salah satunya berbentuk pengguna.</p>
	<p><i>Object</i> atau <i>Lifeline</i>: merupakan penjelasan tentang suatu objek yang terlibat dalam sistem atau aplikasi, contoh seperti sistem, controller dan antarmuka pengguna (UI).</p>
	<p><i>Message (solid arrow)</i> : menunjukkan pesan valid, pemanggilan suatu fungsi.</p>
	<p><i>Return Process</i> : merupakan representasi dari pesan notifikasi atau respon sistem.</p>

	<p><i>Activation Bar</i> : memberikan penjelasan bahwa ada aktivitas yang sedang aktif untuk melakukan suatu proses pada sistem.</p>
	<p><i>Self Call</i> : dalam <i>sequence</i>, <i>self-call</i> biasanya di artikan sebagai objek yang memanggil fungsinya sendiri, ini bisa pemrosesan internal atau sub fungsi.</p>

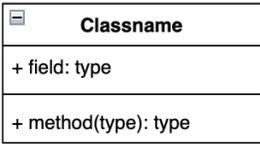
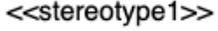
### 2.2.8 Class Diagram

*Class diagram* merupakan salah satu objek utama dalam *unified model language* (UML) yang merepresentasikan struktur data, *method*, attribute dan juga relasi antara entitas. *Class diagram* memberikan informasi yang rinci mengenai pengolahan data dan fungsi dalam sistem yang dibangun. *Class diagram* juga membantu tim pengembang merincikan, menyederhanakan dan memberikan data untuk proses pembuatan *back-end* pada aplikasi. Berikut adalah tabel penjelasan dari objek atau notasi yang ada pada *class diagram*.

“Class Diagram merupakan komponen penting dalam perancangan sistem berbasis web, karena memberikan gambaran mengenai struktur dan hubungan antar kelas dalam sistem. Class Diagram membantu dalam memetakan relasi antar objek serta mendefinisikan atribut dan metode dari setiap kelas yang ada dalam sistem” menurut (Ramdany et al, 2024).

Tabel 2. 4 Notasi Class Diagram

Simbol	Keterangan
--------	------------

	<p><i>Class</i>: merupakan simbol dari entitas utama pada <i>class diagram</i>, simbol ini mempunyai sub-notasi seperti atribut yang berguna untuk informasi dari isi tabel dan <i>method</i> yang merupakan penjelasan dari fungsi pada sistem.</p>
	<p><i>Association</i>: Diartikan sebagai penghubung antar 2 kelas dan mempunyai tipe hubungan yang cukup banyak contoh: 1, 1-*, 1-1</p>
	<p><i>Stereotype</i>: Difungsikan sebagai elemen atau objek penambah dalam class diagram</p>

### 2.2.9 Database

“Database adalah tulang punggung dari setiap sistem informasi modern. Dengan struktur yang terorganisasi, database memungkinkan perusahaan untuk menyimpan data dengan aman dan mengaksesnya secara efisien. Penulis menekankan pentingnya normalisasi dalam desain database untuk memastikan data tidak terduplikasi dan tetap konsisten” menurut (Kadir, 2018).

“Basis data telah menjadi tulang punggung utama dalam pengelolaan informasi di era digital. Dengan pertumbuhan volume data yang sangat besar, muncul kebutuhan untuk mengembangkan sistem basis data yang lebih efisien, baik dalam pengelolaan data terstruktur maupun tidak terstruktur” menurut (Sharma dan Singh, 2020). Salah satu perkembangan penting adalah adopsi database *NoSQL*, yang menawarkan fleksibilitas dan skalabilitas tinggi dibandingkan dengan sistem relasional tradisional. Studi mereka menunjukkan bahwa *NoSQL* sangat efektif untuk aplikasi berbasis *cloud* dan *big data* karena kemampuannya untuk menangani volume data yang sangat besar dengan kinerja tinggi .

juga menekankan pentingnya optimisasi sistem manajemen basis data (DBMS) untuk memenuhi kebutuhan analisis data secara real-time (Sharma dan Singh, 2020). Penelitian ini mengevaluasi berbagai teknik indeksasi dan optimisasi kueri untuk mempercepat akses data. Salah satu temuan mereka adalah bahwa kombinasi antara model basis data hibrid (relasional dan *NoSQL*) dengan teknik pembelajaran mesin dapat meningkatkan efisiensi operasional secara signifikan, terutama dalam lingkungan yang membutuhkan pengolahan data yang intensif.

#### **2.2.10 Payment Gateway**

“Payment gateway adalah teknologi yang memungkinkan transaksi online dengan mengotentikasi dan memproses pembayaran antara pelanggan dan pedagang secara aman. Payment gateway bertindak sebagai perantara yang mengenkripsi informasi pembayaran untuk melindungi data sensitif, seperti nomor kartu kredit, selama proses transaksi. Dalam studi mereka, Kumar dan Gupta mengidentifikasi bahwa implementasi teknologi payment gateway yang lebih aman dan cepat sangat penting untuk mendukung pertumbuhan e-commerce global, terutama dalam era pembayaran digital yang semakin mendominasi” menurut (Kumar dan Gupta, 2021).

“Selain itu, payment gateway memainkan peran penting dalam e-commerce, karena menyediakan platform yang aman dan mudah untuk transaksi online. Selain itu, payment gateway juga memungkinkan merchant untuk menerima berbagai jenis pembayaran dari pelanggan, termasuk kartu kredit, debit, dan dompet digital, sehingga memberikan fleksibilitas yang lebih besar bagi konsumen. Dengan menggunakan payment gateway, proses transaksi menjadi lebih

cepat dan efisien, serta meminimalkan risiko kesalahan dalam pemrosesan pembayaran yang sering terjadi pada metode pembayaran manual” menurut (Gunawan, 2021)

“Payment gateway memainkan peran penting dalam meningkatkan kecepatan transaksi dengan mengintegrasikan sistem pembayaran online dengan sistem merchant. Teknologi ini mengurangi waktu pemrosesan manual dan mempercepat pengalaman pengguna” menurut (Dahlberg et al, 2015),.

“payment gateway menjadi elemen kunci dalam mendorong pertumbuhan ekonomi digital. Dengan memberikan kemudahan akses bagi konsumen untuk melakukan transaksi online, payment gateway telah memfasilitasi peningkatan signifikan dalam e-commerce di seluruh dunia” menurut (Kar and Pani, 2014)

#### **2.2.11 Xendit**

*Xendit* merupakan platform digital penyedia *API Gateway* yang membantu sebuah aplikasi melakukan pembayaran lewat layanannya. *Xendit* memberikan beberapa fitur untuk para partner mereka seperti pemilihan metode pembayaran dan juga pencairan dana ke pengguna aplikasi. Platform ini juga memberikan akses antarmuka yang mudah dipahami untuk pengguna mengakses fitur-fitur yang ada. *Xendit* juga mempunyai sistem keamanan yang cukup untuk menjaga data terkait transaksi yang dilakukan oleh para pengguna.

### **2.3 Penelitian Terdahulu**

Tujuan dari penelitian terdahulu adalah untuk memperoleh pemahaman yang komprehensif tentang konteks dan landasan teoritis dari topik penelitian yang dipilih.

1. Hasil penelitian Dewi Sartika & Febri Yuliani

Penelitian Dewi Sartika & Febri Yuliani berjudul “*Implementasi E-Procurement untuk Meningkatkan Efisiensi Proses Pengadaan di Sektor Publik*” Penelitian ini bertujuan untuk menganalisis bagaimana implementasi *E-Procurement* dapat meningkatkan efisiensi proses pengadaan di instansi pemerintah. Penelitian ini juga mengeksplorasi faktor-faktor yang mempengaruhi keberhasilan implementasi *E-Procurement* di sektor publik.

Berdasarkan hasil penelitian, disarankan agar perusahaan yang ingin mengadopsi *E-Procurement* melakukan persiapan yang matang, termasuk mengelola perubahan di tingkat organisasi dan memastikan infrastruktur teknologi yang memadai. Pelatihan yang memadai bagi pengguna juga harus diberikan untuk memastikan bahwa sistem dapat dimanfaatkan secara optimal. Implikasi dari penerapan *E-Procurement* ini tidak hanya terbatas pada peningkatan efisiensi dan transparansi, tetapi juga dapat menjadi model bagi perusahaan lain yang ingin mengimplementasikan sistem serupa untuk meningkatkan kinerja pengadaan mereka.

Meskipun terdapat banyak manfaat dari *E-Procurement*, penelitian ini juga mengungkapkan banyak tantangan dalam penerapan *E-Procurement*. Salah satu tantangan terbesarnya adalah penolakan internal dan pemasok terhadap peralihan dari prosedur berbasis kertas ke sistem elektronik. Selain itu, memerlukan perhatian khusus karena sulit diintegrasikan dengan sistem lain yang sudah ada, seperti sistem manajemen inventaris dan ERP. Perlunya

persiapan infrastruktur teknis dan pelatihan pengguna yang intensif juga merupakan faktor penting dalam keberhasilan penerapan *E-Procurement*.

Metode pengumpulan data dalam penelitian ini dilakukan dengan metode kualitatif yang melibatkan beberapa teknik, yaitu studi dokumentasi, wawancara, dan observasi. Studi dokumentasi digunakan untuk mengumpulkan informasi dari berbagai sumber tertulis yang relevan dengan topik penelitian. Wawancara dilakukan dengan informan yang memiliki pengetahuan mendalam dan pengalaman langsung terkait implementasi kebijakan pengadaan barang dan jasa secara elektronik, termasuk implementor, pejabat terkait, dan penyedia barang/jasa. Selain itu, observasi dilakukan untuk mendapatkan pemahaman yang lebih komprehensif mengenai praktik dan dinamika yang terjadi di lapangan. Analisis data dilakukan dengan teknik triangulasi, yang mencakup proses pengecekan, pengulangan, dan pemeriksaan silang terhadap data yang diperoleh untuk memastikan validitas dan reliabilitas informasi.

2. Hasil penelitian Hakim, M. A. L. (2021).

Penelitian Hakim, M. A. L. (2021) yang berjudul “*Rancang Bangun Aplikasi Pengadaan Barang Berbasis Web Menggunakan Framework Laravel Pada Kantor Desa Wilayut*”. Tujuan dari penelitian ini adalah untuk merancang dan membangun aplikasi pengadaan barang berbasis web pada Kantor Desa Wilayut. Aplikasi ini dirancang untuk mempermudah pengelolaan barang masuk dan keluar, sehingga proses pengadaan menjadi lebih efektif dan efisien.

Kesimpulan dari penelitian ini menunjukkan bahwa meskipun aplikasi pengadaan barang berbasis web telah berhasil dirancang dan dibangun untuk Kantor Desa Wilayut, aplikasi tersebut masih memerlukan sejumlah perbaikan sebelum dapat digunakan secara optimal. Pengujian yang dilakukan mengungkapkan bahwa beberapa fitur penting belum berfungsi dengan sempurna, yang dapat menghambat kinerja aplikasi dalam skenario penggunaan nyata. Selain itu, proses implementasi juga menunjukkan bahwa pengembangan lebih lanjut diperlukan untuk menyempurnakan antarmuka pengguna dan memastikan aplikasi dapat menangani beban kerja yang lebih besar serta mendukung integrasi dengan sistem lain yang mungkin dibutuhkan oleh kantor desa. Oleh karena itu, sebelum aplikasi ini diimplementasikan secara penuh, disarankan untuk melakukan revisi dan pengujian ulang guna memastikan bahwa semua fungsi berjalan sesuai dengan kebutuhan dan standar yang ditetapkan. Peneliti juga menyarankan adanya pelatihan bagi pengguna agar mereka dapat memanfaatkan aplikasi ini secara efektif dalam kegiatan pengadaan barang di Kantor Desa Wilayut.

Penelitian ini menggunakan metode *System Development Life Cycle* (SDLC), yang merupakan pendekatan sistematis dalam pengembangan perangkat lunak. Tahapan pertama dalam metode ini adalah analisis sistem, di mana peneliti mengidentifikasi kebutuhan dari Kantor Desa Wilayut terkait pengadaan barang. Analisis ini melibatkan pengumpulan data melalui wawancara dengan pihak terkait dan studi literatur untuk memahami proses pengadaan barang yang ada. Hasil analisis ini kemudian dijadikan dasar untuk menentukan

spesifikasi kebutuhan perangkat lunak, termasuk fitur-fitur yang harus ada dalam aplikasi.

3. Hasil penelitian Wijaya, R. H. (2017).

Penelitian ini bertujuan untuk merancang dan membangun aplikasi yang dapat membantu kinerja administrasi dalam keikutsertaan tender di PT Kencana Alam Putra. Aplikasi ini dirancang untuk mengatasi kendala seperti lambatnya persetujuan tender oleh direktur utama dan manajer, serta masalah komunikasi dan koordinasi tim tender yang sering menimbulkan miskomunikasi dan salah paham.

Berdasarkan hasil uji coba, aplikasi administrasi keikutsertaan tender berhasil memenuhi kebutuhan PT Kencana Alam Putra dengan menyediakan fitur seperti daftar dokumen kelengkapan tender, notifikasi email untuk pengguna baru, laporan penawaran BOQ, pencatatan riwayat aktivitas tender, dan notifikasi SMS untuk persetujuan tender oleh direktur utama dan manajer, serta penugasan untuk unit kerja. Aplikasi ini diharapkan dapat meningkatkan efisiensi dan efektivitas dalam proses administrasi tender.

Penelitian ini menggunakan metode perancangan sistem dengan model pengembangan *Waterfall*. Proses penelitian dimulai dari tahap analisis kebutuhan sistem, perancangan sistem, implementasi, dan diakhiri dengan pengujian sistem. Data dikumpulkan melalui observasi langsung di PT Kencana Alam Putra dan dianalisis untuk menentukan kebutuhan sistem yang akan dikembangkan. Implementasi dilakukan dengan membangun

aplikasi berbasis web, dan pengujian dilakukan untuk memastikan aplikasi berjalan sesuai dengan kebutuhan yang telah dirumuskan.

#### 4. Hasil penelitian Akmadi, W. (2004)

Tujuan dari penelitian ini adalah untuk merancang dan membangun sebuah aplikasi sistem informasi yang dapat mendukung administrasi keikutsertaan tender di Dinas Pekerjaan Umum Kabupaten Kota Waringin Timur. Aplikasi ini diharapkan dapat memberikan notifikasi, laporan, dan history yang komprehensif terkait proses tender, sehingga mempermudah proses komunikasi dan dokumentasi dari awal pengajuan tender hingga selesai.

Penelitian ini berhasil menghasilkan aplikasi administrasi keikutsertaan tender yang mampu memberikan berbagai jenis laporan dan notifikasi, baik melalui email maupun SMS, kepada pihak-pihak terkait seperti Direktur Utama, Manajer, dan Unit Kerja. Aplikasi ini juga mampu mencatat perubahan dan kegiatan yang terjadi dalam proses tender, sehingga menciptakan transparansi dan efisiensi dalam pengelolaan tender. Dengan demikian, aplikasi ini memberikan solusi yang signifikan dalam mengoptimalkan proses pengelolaan tender di lingkungan Dinas Pekerjaan Umum Kabupaten Kota Waringin Timur.

Metode penelitian yang digunakan dalam penelitian ini melibatkan beberapa tahapan penting. Pertama, data primer dikumpulkan melalui wawancara dengan pihak terkait di Dinas Pekerjaan Umum. Selain itu, data sekunder diperoleh melalui studi pustaka dari berbagai referensi jurnal, artikel, dan

penelitian terdahulu. Pengembangan aplikasi dilakukan menggunakan metode Waterfall, yang mencakup tahap perencanaan, analisis, desain, implementasi, dan pengujian. Setiap tahap dilakukan secara berurutan untuk memastikan bahwa sistem yang dibangun memenuhi kebutuhan yang telah diidentifikasi selama proses analisis (uniiversitas dinamika).

5. Hasil penelitian dari Wonoseto, M. G., Maulana, A., & Asyari, M. (2023)

Menurut penelitian Wonoseto, M. G., Maulana, A., & Asyari, M. (2023) yang berjudul “Rancang bangun aplikasi pengadaan barang dan jasa Universitas Negeri XYZ dengan metode Extreme Programming.” Aplikasi *E-Procurement* dibuat untuk mendorong digitalisasi, meningkatkan transparansi, serta mengatasi kendala yang dialami dalam proses pengadaan barang dan jasa yang masih dilakukan secara terpisah.

Penelitian ini berhasil mengidentifikasi kebutuhan fungsional yang diperlukan dalam aplikasi e-purchasing dan merealisasikan 91,6% dari semua fitur yang diperlukan, termasuk fitur dashboard, pembagian hak akses, dan persetujuan penawaran. Aplikasi ini diharapkan dapat diadaptasi di universitas atau instansi pemerintah lainnya untuk meningkatkan efisiensi dan transparansi dalam pengadaan barang dan jasa.

Penelitian ini menggunakan metode Extreme Programming yang terdiri dari empat tahapan utama: perencanaan, perancangan, pengkodean, dan pengujian. Melalui pendekatan kolaboratif dan adaptif, metode ini memungkinkan

pengembangan perangkat lunak yang responsif terhadap perubahan dan sesuai dengan kebutuhan pengguna, dengan implementasi menggunakan bahasa pemrograman PHP, HTML, dan framework CodeIgniter serta Bootstrap.

