

## **BAB II**

### **TINJAUAN PUSTAKA DAN LANDASAN TEORI**

#### **2.1 Tinjauan Pustaka**

Penelitian dengan tema pengadaan barang dan jasa ini sudah dilakukan oleh beberapa peneliti dengan metode dan cara mengembangkan sistem yang berbeda – beda.

Penelitian dengan tema pengadaan barang dan jasa ini dilakukan oleh Resti Lia Andharsaputri (2021) dengan judul “Implementasi Rapid Application Development Pada Sistem Informasi Pengadaan Barang Dan Jasa“. Permasalahan yang timbul karena penggunaan sistem yang masih konvensional pada perusahaan pengadaan barang dan jasa diantaranya kesalahan dalam pencatatan dan pengolahan data, informasi lambat diterima dan kurang akurat, penyimpanan dokumen yang tidak tersentralisasi sehingga menimbulkan resiko kehilangan, kerusakan, kesulitan dalam pencarian dokumen. Solusi yang dapat dilakukan adalah dengan membangun sistem. Sistem ini menggunakan metode RAD, untuk design sistem menggunakan use case dan activity diagram, rancangan database serta transformasi relasinya menggunakan ERD dan LRS. bahasa pemrograman yang digunakan yaitu Java Script, CSS, HTML. Pengolahan basis data menggunakan NetBeans IDE 8,1, PHP, MySQL dan XAMPP. Hasil dan pembahasannya dapat disimpulkan bahwa sistem informasi pengadaan barang dan jasa berbasis website pada PT. Dan Purna Investama dapat membantu mengatasi kendala atau permasalahan yang ada di perusahaan tersebut yang ditimbulkan karena penggunaan sistem yang konvensional.

Penelitian berikutnya dilakukan oleh Dony Ardiansyah (2021) dengan judul “Implementasi Metode Prototyping Pada Sistem Informasi Pengadaan Barang Cetakn Berbasis WEB”. Permasalahan dokumentasi yang masih dilakukan secara manual serta kurang terkontrolnya proses pencatatan penerimaan dan pengeluaran barang sehingga antara data yang tersedia dengan stok barang bisa tidak sesuai. Oleh karena itu diperlukan adanya sebuah sistem. Sistem ini dibuat menggunakan metode prototype dengan design sistem menggunakan UML. Bahasa pemrograman PHP dan database MySQL serta aplikasi pendukung

lainnya yaitu Adobe Dreamweaver CS5. Hasil dari penelitian ini adalah sistem informasi pengadaan barang cetakan memudahkan dalam monitoring data pengadaan barang cetakan serta lebih aman dan terciptanya efisiensi.

Penelitian berikutnya dilakukan oleh M. Rizal (2019) dengan judul “Sistem Informasi Pengadaan Barang Pada PT. Bank Rakyat Indonesia Kantor Cabang Pangkalpinang Berbasis WEB”. Data yang diolah pada BRI Cabang Pangkalpinang sistem pengolahan data Pengadaan yang diterapkan masih standar yaitu menggunakan Microsoft Word dan Microsoft Excel serta disimpan dalam folder yang tidak beraturan sehingga mempersulit pencarian data dan berakibat pada keterlambatan dalam penyampaian informasi. Berdasarkan hal tersebut peneliti membuat sistem sebagai solusi untuk masalah tersebut. Sistem ini dibuat menggunakan metode FAST, dengan design sistem menggunakan UML dan bahasa pemrograman PHP dan database MySQL. Hasil dari penelitian ini yaitu sistem informasi yang berbasis website ini dapat memperoleh data yang benar – benar valid dan sesuai dengan kondisi real yang ada di BRI Cabang Pangkalpinang.

Perbedaan dari penelitian – penelitian sebelumnya dengan penelitian ini, sebagai berikut :

1. Metode yang digunakan pada penelitian sebelumnya dengan penelitian ini berbeda, masing – masing menggunakan metode, yaitu FAST, RAD, Prototype, sedangkan di penelitian ini menggunakan metode Waterfall.
2. Aplikasi pendukung yang digunakan di penelitian sebelumnya dengan penelitian ini juga berbeda, ada yang menggunakan NetBeans IDE 8.1, Adobe Dreamweaver CS5, sedangkan di penelitian kali ini menggunakan aplikasi pendukung seperti, Visual Studio Code, Sourcetree dan XAMPP.
3. Aplikasi untuk design sistem dari masing – masing penelitian juga berbeda, ERD, LRS, dan UML dan pada penelitian kali ini menggunakan aplikasi design berupa ERD.
4. Fitur yang dihasilkan dalam setiap penelitian juga berbeda beda, ada yang mempunyai fitur notifikasi whatsapp, *scan barcode* dan pada penelitian ini mempunyai fitur validasi otomatis.

## 2.2 Landasan Teori

### 2.2.1 Sistem

Sistem adalah suatu jaringan kerja dari prosedur – prosedur yang saling berhubungan, berkumpul bersama – sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tersebut. Sistem merupakan teknologi yang dapat memudahkan suatu permasalahan [2]. Agar sebuah sistem dapat dikatakan sebagai sistem yang baik maka sistem memiliki beberapa karakteristik yaitu:

#### a. **Komponen**

Sistem berupa komponen yang saling berinteraksi, dan bekerja sama untuk membentuk satu kesatuan. Komponen dari sistem dapat terdiri dari banyaknya subsistem atau bagian dari sistem yang dibentuk.

#### b. **Batasan Sistem (*boundary*)**

Batasan sistem mampu membatasi sistem satu dengan sistem yang lain, sehingga batasan memungkinkan suatu sistem dipandang menjadi satu kesatuan, batasan sistem inilah yang membuktikan suatu ruang lingkup dari sistem yang dibentuk.

#### c. **Lingkungan Luar Sistem (*environment*)**

Segala sesuatu yang merupakan batas dari luar yang mempengaruhi operasi sistem disebut lingkungan luar sistem. Lingkungan yang bersifat menguntungkan harus dijaga, sedangkan yang bersifat merugikan harus dikendalikan agar tidak mengganggu sistem yang berlangsung.

#### d. **Penghubung sistem (*interface*)**

Sebagai penghubung yang menghubungkan subsistem satu dengan yang lainnya, sehingga sumber daya dapat mengalir secara berhubungan.

#### e. **Masukkan Sistem (*Input*)**

Segala sesuatu yang dimasukkan ke dalam sistem yang kemudian akan diproses melalui sistem.

**f. Keluaran Sistem (*Output*)**

Hasil dari pemrosesan yang diolah untuk menjadi hasil keluaran yang berguna.

**g. Pengolah Sistem (*Process*)**

Sistem akan bekerja mengolah apa yang menjadi masukan menjadi keluaran yang memiliki daya guna.

**h. Sasaran Sistem (*Objective*)**

Tujuan dari dibuatnya sistem, sasaran dari sistem sangat menentukan input yang dibutuhkan untuk sistem dan menghasilkan keluaran yang sesuai pada sistem.

Informasi adalah hal yang paling mendasar dan diperlukan oleh suatu kegiatan dalam pengambilan keputusan agar dapat menghindari kesalahan dan sebagai olah data sehingga memiliki daya guna bagi penerima informasi.

Informasi adalah sekumpulan data atau fakta yang diorganisasi atau diolah dengan cara tertentu sehingga memiliki arti bagi para penerimanya. Informasi memiliki karakteristik, sebagai berikut:

- a. Relevan, mempunyai makna dan dapat digunakan secara tepat untuk mengambil keputusan.
- b. Andal, memiliki tingkat keandalan yang tinggi.
- c. Lengkap, memiliki penjelasan yang jelas dan rinci.
- d. Tepat Waktu, harus dalam kondisi yang terbaru dan tidak dalam bentuk usang.
- e. Dapat dipahami, informasi yang disajikan jelas sehingga mudah dimengerti.

**2.2.2 Pengadaan Barang dan Jasa**

Pengadaan barang/jasa pada dasarnya merupakan upaya pengguna yang mengajukan barang/jasa untuk mendapatkan barang/jasa yang diajukan. Pengadaan barang/jasa pasti ada di setiap instansi atau perusahaan, maka dari itu pengadaan barang/jasa memegang peranan penting di setiap instansi atau perusahaan.

Pengadaan Barang/Jasa adalah kegiatan pengadaan barang/jasa oleh pengguna yang melakukan pengadaan barang/jasa yang prosesnya

dimulai dari identifikasi kebutuhan, sampai dengan serah terima hasil pekerjaan [3].

Definisi pengadaan/jasa menurut kamus bahasa berarti memberikan pekerjaan kepada pihak lain untuk mengerjakan pekerjaan seluruhnya sesuai dengan pekerjaan yang disepakati kedua belah pihak sebelum dilakukan [4].

### 2.2.3 PHP

PHP merupakan bahasa pemrograman web server-side yang bersifat open source. PHP merupakan script yang menyatu dengan HTML dan dieksekusi di server (*Server side HTML embedded scripting*). PHP adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis [5].

### 2.2.4 Internet

Internet adalah seluruh jaringan yang saling terhubung satu sama lain. Beberapa halaman-halaman dalam jaringan ini menyimpan file, seperti halaman web, yang dapat diakses oleh seluruh jaringan internet [6].

### 2.2.5 MySQL

MySQL dikembangkan oleh sebuah perusahaan Swedia bernama MySQL AB yang pada saat itu bernama TcX Data Konsult AB sekitar tahun 1944-1995. MySQL adalah salah satu jenis database server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang database sebagai sumber dan pengelolaan datanya [7].

### 2.2.6 Laravel

Laravel merupakan sebuah *framework* PHP yang akan membantu para web developer untuk menciptakan aplikasi istimewa secara simple, kode yang bersih dan menyenangkan. *Framework* biasa digunakan digunakan untuk mempermudah perancangan PHP [8]. *Framework* adalah suatu struktur yang didalamnya terdapat *class-class*, pustaka-pustaka dan infrastruktur *run-time* untuk membuat *web*, dengan menggunakan *framework* memungkinkan untuk membuat *web* dengan mudah dan cepat. *Framework* saat ini tersedia berbagai macam untuk mengembangkan *web*, salah satunya yaitu *Codeigniter*. *Framework*

*codeigniter* ini *open source* dan mudah untuk dipelajari. *Codeigniter* merupakan *framework* dengan bahasa pemrograman *PHP* dan menggunakan konsep *MVC (Models-View-Controller)* untuk memisahkan data dan presentasi sehingga pengembangan *web* akan mudah dan cepat dalam proses pengelolaannya.

### **2.2.7 HTML**

HTML merupakan turunan atau pengembangan dari SGML (*Standar Generalized Markup Language*). HTML sendiri dikembangkan oleh Tim Berners-Lee sewaktu masih kerja CERN yang pertama kali dipopulerkan oleh browser Mosaic yang dikembangkan NCSA [9].

### **2.2.8 Waterfall**

Metode Waterfall merupakan metode pengembangan perangkat lunak yang paling sederhana dan terkadang disebut sebagai *classic life cycle* (siklus hidup klasik) yang menyarankan pendekatan sistematis dan sekuensial untuk pengembangan perangkat lunak [10].

#### **a. Analisis**

Proses pencarian kebutuhan diintensifkan dan difokuskan pada software. Untuk mengetahui sifat dari program yang akan dibuat, maka para software engineer harus mengerti tentang domain informasi dari software, misalnya fungsi yang dibutuhkan, user interface.

#### **b. Design**

Proses ini digunakan untuk mengubah kebutuhankebutuhan di atas menjadi representasi ke dalam bentuk “blueprint” software sebelum coding dimulai. Desain harus dapat mengimplementasikan kebutuhan yang telah disebutkan pada tahap sebelumnya.

#### **c. Pengkodean**

Untuk dapat dimengerti oleh mesin, dalam hal ini adalah komputer, maka desain tadi harus diubah bentuknya menjadi bentuk yang dapat dimengerti oleh mesin, yaitu ke dalam bahasa pemrograman melalui proses coding. Tahap ini merupakan implementasi dari tahap design yang secara teknis nantinya dikerjakan oleh programmer.

**d. Pengujian**

Sesuatu yang dibuat haruslah diujicobakan. Demikian juga dengan software. Semua fungsifungsi software harus diujicobakan, agar software bebas dari error, dan hasilnya harus benar-benar sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya.

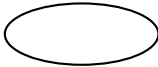
**e. Pemeliharaan**

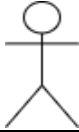

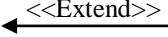

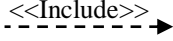
Pemeliharaan suatu software diperlukan, termasuk di dalamnya adalah pengembangan, karena software yang dibuat tidak selamanya hanya seperti itu. Ketika dijalankan mungkin saja masih ada error kecil yang tidak ditemukan sebelumnya, atau ada penambahan fitur-fitur yang belum ada pada software tersebut. Pengembangan diperlukan ketika adanya perubahan dari eksternal perusahaan seperti ketika ada pergantian sistem operasi, atau perangkat lainnya.

**2.2.9 Use Case**

*Use Case* merupakan rangkaian atau uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor [11]. *Use case* digunakan untuk mendeskripsikan sebuah interaksi yang terjadi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat.

**Tabel 2. 1** Simbol *Use Case*

No	Simbol	Fungsi
1.	<p><i>Use Case</i></p> 	<p>Fungsionalitas yang disediakan oleh sistem sebagai unit-unit yang digunakan untuk saling bertukar pesan antar unit atau actor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i></p>


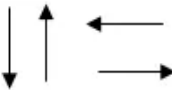

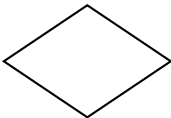
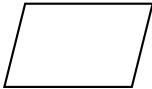

2.	Aktor / <i>actor</i> 	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i>
3.	Asosiasi / <i>association</i> 	Komunikasi antara <i>actor</i> dan <i>use case</i> yang saling berhubungan atau <i>use case</i> yang memiliki interaksi dengan aktor
4.	Ekstensi / <i>extend</i> 	Menunjukkan bahwa suatu bagian dari elemen dapat disisipkan ke dalam elemen yang lainnya
5.	Generalisasi / <i>generalization</i> 	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> )
6.	<i>Include</i> 	Menunjukkan bahwa suatu bagian dari elemen dapat memicu eksekusi bagian dari elemen lain (yang ada pada tanda panah)

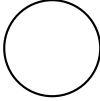
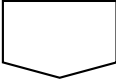
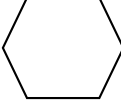



### 2.2.10 Flowchart



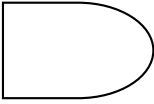
*Flowchart* merupakan bagan yang mengarahkan *flow* di dalam prosedur atau program secara logika [12]. Bagan alir ini berguna bagi seorang analis sistem untuk menggambarkan proses di dalam suatu prosedur.



Tabel 2. 2 Simbol *Flowchart*

No	Simbol	Fungsi
1.	<i>Terminal point</i> 	Simbol yang digunakan untuk menyatakan awal atau akhir suatu program
2.	<i>Flow direction</i> 	Simbol yang digunakan untuk menghubungkan antar simbol, berfungsi juga untuk menunjukkan alur dari sebuah proses
3.	Proses 	Menunjukkan sebuah kegiatan yang dilakukan oleh komputer atau sistem
4.	<i>Decision</i> 	Simbol yang menunjukkan suatu kondisi tertentu yang menghasilkan dua kemungkinan jawaban yaitu ya atau tidak
5.	<i>Input atau output</i> 	Menunjukkan sebuah proses <i>input</i> atau <i>output</i> yang ada dan tidak bergantung pada jenis peralatan yang digunakan
6.	<i>Predefined process</i> 	Simbol yang digunakan untuk menyatakan kumpulan langkah proses ditulis sebagai prosedur


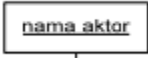

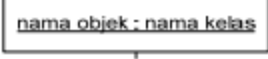

7.	<i>Connector (On-page)</i> 	Simbol yang digunakan untuk menghubungkan suatu simbol pada satu halaman yang letaknya saling berjauhan
8.	<i>Connector (Off-page)</i> 	Digunakan untuk menghubungkan simbol yang berada pada halaman yang berbeda
9.	<i>Preparation</i> 	Simbol yang menunjukkan persiapan penyimpanan ke dalam <i>storage</i>
10.	<i>Manual input</i> 	Simbol yang menunjukkan proses <i>input</i> yang dilakukan secara <i>manual</i> dengan menggunakan <i>online keyboard</i>
11.	<i>Manual operation</i> 	Simbol yang menunjukkan proses atau kegiatan yang dilakukan tanpa menggunakan komputer
12.	<i>Document</i> 	Menunjukkan proses <i>input</i> berupa dokumen dalam bentuk kertas ataupun <i>output</i> yang harus dicetak

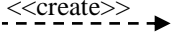
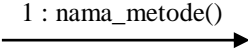
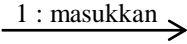
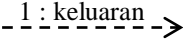
13.	<i>Multiple document</i> 	Sama seperti simbol <i>document</i> hanya saja dokumen yang digunakan lebih dari satu
14.	<i>Display</i> 	Simbol yang menunjukkan adanya penggunaan peralatan <i>output</i>
15.	<i>Delay</i> 	Menunjukkan bahwa adanya proses <i>delay</i>

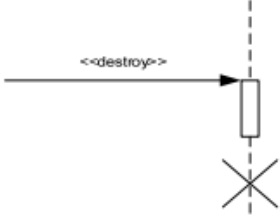
### 2.2.11 Sequence Diagram

*Sequence diagram* merupakan salah satu yang menjelaskan bagaimana suatu operasi itu dilakukan *message* (pesan) apa yang akan dikirim dan kapan pelaksanaannya [13]. Dalam menggambarkan diagram *sequence* maka harus diketahui objek-objek yang terlibat di dalam sebuah *use case* dan metode-metode yang dimiliki oleh kelas yang diinstansiasi menjadi objek tersebut.

Tabel 2. 3 Simbol *Sequence Diagram*

No	Simbol	Fungsi
1.	<p>Aktor</p>  <p>Atau</p>  <p>Tanpa waktu aktif</p>	Aktor menyatakan orang, proses, atau sistem lain yang berkaitan dengan sistem informasi yang akan dibuat di luar sistem informasi tersebut
2.	<p>Garis hidup / <i>lifeline</i></p> 	<i>Lifeline</i> menyatakan keberadaan dari suatu objek dalam basis waktu. Notasi yang digunakan untuk <i>lifeline</i> adalah garis lurus vertikal putus-putus yang ditarik dari sebuah objek
3.	<p>Objek</p> 	Menyatakan sebuah objek yang saling berinteraksi
4.	<p>Waktu aktif</p> 	Menyatakan objek dalam keadaan aktif dan saling terhubung yang merupakan sebuah tahapan yang dilakukan di dalamnya


5.	Pesan tipe <i>create</i>  	Pesan tipe <i>create</i> , menyatakan suatu objek membuat objek yang lain dengan arah panah yang mengarah pada objek yang dibuat
6.	Pesan tipe <i>call</i>  	Pesan tipe <i>call</i> , menyatakan suatu objek yang memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri
7.	Pesan tipe <i>send</i>  	Pesan tipe <i>send</i> , menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya dengan arah panah yang mengarah pada objek yang dikirim
8.	Pesan tipe <i>return</i>  	Pesan tipe <i>return</i> , menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode akan menghasilkan suatu kembalian ke objek tertentu dengan arah panah yang mengarah pada objek yang menerima kembalian




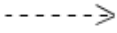

9.	Pesan tipe <i>destroy</i> 	Pesan tipe <i>destroy</i> , menyatakan suatu objek yang mengakhiri hidup objek lain dengan arah panah yang mengarah pada objek yang diakhiri, <i>destroy</i> sebaiknya digunakan jika ada <i>create</i>
----	--	---

### 2.2.12 Class Diagram

*Class Diagram* adalah deskripsi objek-objek dengan *property*, perilaku (operasi) dan relasi yang sama. Disamping itu *class diagram* bisa memberikan pandangan global atas sebuah sistem. *Class* dalam notasi UML digambarkan kotak. Nama *class* menggunakan huruf besar di awal kalimatnya dan diletakkan di atas kotak. Bila *class* mempunyai nama yang terdiri dari 2 (dua) suku kata atau lebih, maka semua suku kata digabungkan tanpa spasi dengan huruf awal tiap suku kata menggunakan huruf besar. *Attribute* adalah *property* dari sebuah *class*. *Attribute* ini melukiskan batas nilai yang mungkin ada pada obyek *class*. Sebuah *class* mungkin mempunyai nol atau lebih *attribute*[14]. *Class Diagram* merupakan sebuah diagram untuk menjelaskan adanya hubungan antar *class* dalam sistem dan menjelaskan bagaimana cara agar saling terhubung dan berkolaborasi.

**Tabel 2. 4** Simbol *Class Diagram*

No.	Nama	Simbol	Keterangan
1.	<i>Generalization</i>		Relasi antarkelas dengan makna generalisasi-spesialisasi (umum-khusus)

No.	Nama	Simbol	Keterangan
2.	<i>Nary Association</i>		Upaya untuk menghindari asosiasi dengan lebih dari 2 objek
3.	<i>Class</i>		Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama
4.	<i>Realization</i>		Operasi yang dilakukan oleh suatu objek
5.	<i>Dependency</i>		Relasi antarkelas dengan makna kebergantungan antarkelas
6.	<i>Association</i>		Diagram yang menghubungkan antara objek satu dengan objek lainnya.

### 2.2.13 Blackbox

*Blackbox* adalah metode testing atau uji coba fungsionalitas sebuah aplikasi yang sedang dikembangkan [15]. Pengujian ini dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak yang dibuat sudah sesuai dengan spesifikasi yang dibutuhkan.

Pengujian kotak hitam (*black-box testing*) dilakukan dengan cara membuat kasus uji untuk mencoba semua fungsi perangkat lunak apakah sudah sesuai dengan spesifikasi yang dibutuhkan atau belum. Kasus uji yang dibuat untuk melakukan pengujian kotak hitam atau *black-box testing* harus dibuat berdasarkan kasus benar dan juga kasus salah.

#### **2.2.14 Usability Testing**

*Usability Testing* adalah teknik yang digunakan dalam mengevaluasi kualitas dan kegunaan pada sebuah layanan atau produk dalam melibatkan pengguna aplikasi dengan langsung demi upaya mengetahui informasi permasalahan dari aplikasi perangkat bergerak yang diujikan [16].