

## I. PENDAHULUAN

### 1.1. Latar Belakang

Mesin hitung uang adalah mesin *mekatronika* yang digunakan untuk mempermudah proses perhitungan uang secara cepat dan efektif sehingga tidak perlu menghitung uang secara manual dan meminimalisir terjadinya kesalahan dalam menghitung (Karyanto, t.t., 2018), mesin hitung uang sangat rentan terhadap kerusakan yang bersifat kecil maupun besar, untuk mengatasi kerusakan tersebut, maka harus dilakukan perbaikan dan penggantian *sparepart*. Perbaikan mesin hitung uang dilakukan oleh pihak yang menyediakan *service* mesin hitung uang, salah satu perusahaan yang menyediakan jasa *service* mesin hitung uang adalah PT PQR. PT PQR merupakan salah satu perusahaan yang bergerak dibidang pelayanan jasa perbaikan mesin ATM (*Automated Teller Machine*) dan perbaikan mesin hitung uang. PT PQR memiliki kantor cabang yang tersebar diseluruh Indonesia, salah satunya berada di Lampung yang dipimpin oleh seorang kepala cabang.

Sebuah bank yang ingin menggunakan jasa *service* mesin hitung uang dari PT PQR harus melakukan kerja sama dengan PT PQR terlebih dahulu. Setelah melakukan kerja sama dengan PT PQR, bank tersebut bisa menggunakan layanan yang disediakan oleh PT PQR seperti perbaikan mesin hitung uang dan penggantian *sparepart* yang rusak. Apabila terdapat kerusakan pada mesin hitung uang, maka pelanggan dapat menelpon teknisi PT PQR yang khusus menangani perbaikan pada mesin hitung uang. Namun, apabila teknisi sedang melakukan panggilan dengan pelanggan yang lain, maka pelanggan tersebut harus menunggu teknisi menyelesaikan tugasnya terlebih dahulu. Teknisi akan menelpon Kepala Cabang PT PQR untuk menanyakan *stock* mesin hitung uang cadangan, kemudian Kepala Cabang akan memberikan mesin hitung uang cadangan pada teknisi dan form perbaikan. Teknisi membawa mesin hitung uang tersebut menuju ke lokasi. Mesin hitung uang tersebut akan dipinjamkan ke perusahaan, sementara mesin hitung uang yang rusak akan dibawa ke kantor cabang untuk diperbaiki. Tidak

adanya informasi *stock* mesin hitung uang, berdampak ketika kepala cabang sedang tidak berada di kantor dan tidak tahu apakah *stock* mesin hitung uang masih ada atau tidak. Teknisi akan melakukan pengecekan terhadap kerusakan kemudian mengisi form dalam bentuk *hardcopy* dengan mendata kerusakan yang terjadi, form dalam bentuk *hardcopy* membutuhkan biaya untuk mencetak form tersebut. Selanjutnya teknisi akan melakukan perbaikan pada mesin hitung uang, apabila ada *sparepart* yang harus diganti maka teknisi akan memesan *sparepart* tersebut langsung ke PT PQR yang berada di pusat. Apabila *sparepart* yang dipesan sudah datang, maka teknisi akan segera memperbaiki mesin hitung yang rusak. Setelah teknisi memperbaiki mesin hitung uang, teknisi akan mengisi form lagi dengan mendata perbaikan apa saja yang telah dilakukan dan *sparepart* apa yang diganti. Kemudian teknisi akan mengantarkan mesin hitung uang yang sudah diperbaiki ke bank dan mengambil kembali mesin hitung uang yang dipinjamkan. Selanjutnya teknisi menyerahkan form tersebut ke Kepala Cabang PT PQR.

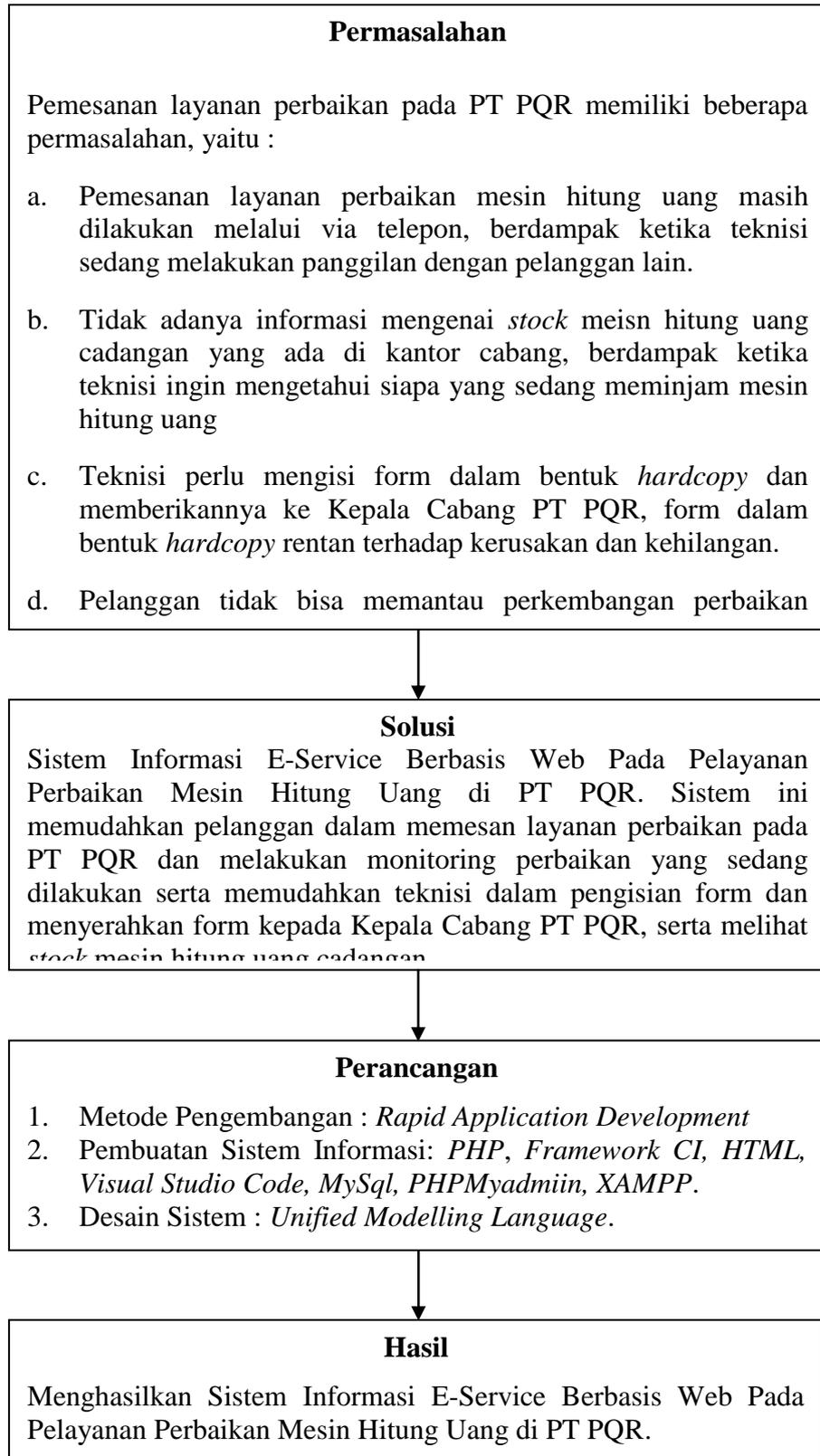
Agar memudahkan pelanggan dalam melakukan permintaan jasa *service* kepada PT PQR dan memudahkan teknisi dalam melakukan pengisian form, serta pelanggan dapat memantau perkembangan perbaikan yang sedang dilakukan, maka dibutuhkan sebuah sistem baru yang handal dan efisien yaitu “Sistem Informasi E-Service Berbasis Web Pada Pelayanan Perbaikan Mesin Hitung Uang di PT PQR”.

## **1.2. Tujuan**

Tujuan dari penulisan tugas akhir ini adalah menghasilkan sebuah Sistem Informasi E-Service Berbasis Web Pada Pelayanan Perbaikan Mesin Hitung Uang di PT PQR.

## **1.3. Kerangka Pemikiran**

Berdasarkan latar belakang dapat disusun suatu kerangka pemikiran yang disajikan dalam Gambar 1.



Gambar 1. Kerangka Pemikiran

#### 1.4. Kontribusi

Kontribusi yang diharapkan dari pembuatan sistem ini diantaranya:

1. Pelanggan
  - a. Memudahkan pelanggan dalam memesan layanan perbaikan terhadap kerusakan yang terjadi pada mesin hitung uang.
  - b. Memudahkan pegawai dalam melakukan *tracking* perbaikan mesin hitung uang yang sedang dilakukan.
  
2. Teknisi
  - a. Memudahkan teknisi dalam menerima pesan permintaan pelayanan perbaikan dari pelanggan.
  - b. Memudahkan teknisi dalam pengisian form melalui web.
  - c. Memudahkan teknisi dalam penyerahan form ke Kepala Cabang PT PQR.
  - d. Memudahkan teknisi mengetahui *stock* mesin hitung uang cadangan yang ada di kantor.
  
3. Kepala Cabang PT PQR
  - a. Memudahkan Kepala Cabang PT PQR dalam menerima pemberitahuan pesan permintaan pelayanan perbaikan dari pelanggan melalui notifikasi *whatsapp*.
  - b. Memudahkan Kepala Cabang PT PQR dalam menerima form perbaikan yang telah diisi oleh teknisi.

## II. TINJAUAN PUSTAKA

### 2.1. Sistem Informasi E-Service

Perancangan Sistem Informasi E-Service Berbasis Web Pada Pelayanan Perbaikan Mesin Hitung Uang di PT PQR menggunakan bahasa pemrograman PHP. Menurut (Rubiati, 2018), PHP atau PHP *Hypertext Preprocessor* adalah bahasa pemrograman berupa script yang ditempatkan didalam server lalu kemudian diproses, kemudian hasil pemrosesan dikirim ke web browser klien.

### 2.2. E-Service

*E-Service* atau layanan elektronik merupakan satu aplikasi terkemuka yang memanfaatkan Teknologi Informasi dan Komunikasi (TIK) untuk meningkatkan pelayanan di daerah yang berbeda, (Messakh & Siwalankerto, 2016).

### 2.3. Internet

#### 2.3.1. Website

Kumpulan dari halaman web dalam sebuah domain ataupun subdomain yang memuat halaman berisi data berupa teks, gambar, audio, video dan lain-lain yang diakses secara online, (Josi, 2017).

#### 2.3.2. Web Browser

*Web browser* adalah *software* yang digunakan sebagai akses halaman web guna mendapatkan informasi, sumber informasi diidentifikasi menggunakan URI (*Uniform Resource Identifier*), (Rochmadi, 2019).

#### 2.3.3. XAMPP

*XAMPP* merupakan distribusi *apache* dalam satu paket program yang di dalamnya terdapat teknologi pengembangan web yang dapat

mempelajari program web yaitu PHP dan *MySQL*, paket pemrograman ini dapat diunduh secara *free* atau gratis, (Nurhayati dkk., 2018).

#### **2.3.4. Basis Data**

Kelompok data yang terorganisir yang digunakan untuk mendukung aplikasi dengan mengontrol data dan memusatkan data *redundant* atau data yang sama disebut basis data atau *database*, fungsi utama dari basis data adalah memelihara data yang telah diolah menjadi informasi yang selalu tersedia saat dibutuhkan, (Josi, 2017).

#### **2.3.5. *MySQL (My Structure Query Language)***

*MySQL* adalah sistem manajemen *database* yang digunakan di PHP untuk menjalankan aplikasi yang akan dibuat. *MySQL* juga digunakan untuk mengolah dan menampung *database* dalam jumlah sangat besar, (Nurhayati dkk., 2018).

#### **2.3.6. *Flowchart***

Menurut (Syamsiah, 2019), *flowchart* merupakan diagram alir yang digunakan untuk menjelaskan alur kerja sebuah sistem dengan menggunakan simbol-simbol tertentu yang mudah dipahami.

#### **2.3.7. *PHPMysqlAdmin***

*PhpmyAdmin* merupakan sebuah aplikasi open source yang dapat digunakan untuk melakukan pemrograman pada database *MySQL*. *PhpMyAdmin* menggunakan bahasa pemrograman PHP dan mendukung berbagai operasi *MySQL*, (Ramadhan & Mukhaiyar, 2020).

### **2.3.8. Framework CodeIgniter**

*Framework CodeIgniter* atau disebut juga MVC (*Model, View, Controller*) adalah *framework* yang digunakan untuk memisahkan antara tampilan sebuah aplikasi dan logika pemrograman, (Tyowati & Irawan, 2017).

### **2.3.9. Application Programming Interface**

API adalah kumpulan intruksi yang disimpan dalam bentuk *library* dan juga *software interface* yang digunakan sebagai penghubung antar dua aplikasi yang berbeda. (Utomo dkk., 2020)

### **2.3.10. WhatsApp API**

*WhatsApp* adalah sebuah software berbasis aplikasi *smartphone* dan *web* yang digunakan sebagai media komunikasi. *WhatsApp* telah menyediakan API resmi yang bisa digunakan oleh *developer* untuk membuat “internet robot” yakni *WhatsApp Business*, (Parlika dkk., 2020).

### **2.3.11. Black-box Testing**

*Black-box Testing* merupakan teknik pengujian aplikasi yang mengutamakan pada fungsional dari aplikasi tersebut. Pengujian aplikasi dengan teknik *Black-box Testing* yaitu tidak berfokus pada struktur control sehingga hanya difokuskan pada informasi domain.

## **2.4. Desain Sistem**

### **2.4.1. Unified Modelling Language**

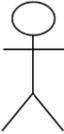
Menurut (Hendini, 2016), UML adalah sebuah model yang digunakan untuk membuat dokumentasi, spesifikasi dan membangun sebuah perangkat lunak. UML merupakan salah satu metodologi dalam

pengembangan sistem berorientasi objek. Alat yang digunakan dalam perancangan sistem berorientasi objek berbasis UML adalah sebagai berikut:

a. *Use case* Diagram

*Use case* diagram pemodelan sistem yang digunakan untuk menggambarkan *behavior* (kelakuan) sebuah sistem serta digunakan untuk mengetahui fungsi yang ada di dalam sistem. Simbol-simbol yang digunakan dalam *Use case* Diagram yaitu:

Tabel 1. Use Case Diagram

Gambar	Nama	Keterangan
	<i>Use case</i>	<i>Use case</i> menggambarkan fungsionalitas sistem sebagai unit-unit yang bertukar pesan antar unit dengan <i>actor</i> , yang dinyatakan menggunakan kata kerja.
	<i>Actor</i>	<i>Actor</i> adalah Abstrak dari orang atau sistem yang mengaktifkan fungsi dari sistem. Untuk mengidentifikasi aktor, pembagian tenaga kerja dan tugas-tugas harus ditentukan berdasarkan peran pada konteks target sistem. Aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki kontrol terhadap <i>use case</i> .
	<i>Association</i>	Asosiasi terletak antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang diartikan bahwa apa atau siapa yang meminta interaksi secara langsung.
	<i>Generalization</i>	<i>Generalization</i> antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk

mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.

-----  
<<include>>

*Include*

*Include* merupakan hubungan di dalam *use case* lain (*required*) atau pemanggilan *use case* oleh *use case* lain, contohnya adalah pemanggilan sebuah fungsi program.

←-----  
<<extended>>

*Extend*

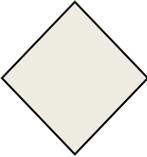
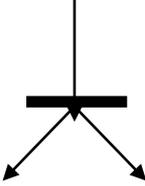
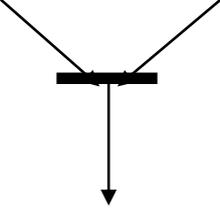
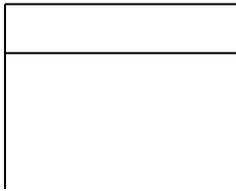
*Extend* merupakan perpanjangan dari *use case* lain jika kondisi atau syarat terpenuhi.

---

b. Diagram Aktivitas (*Activity Diagram*)

*Activity Diagram* menggambarkan aliran aktivitas atau alir kerja dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *Activity Diagram* yaitu :

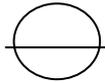
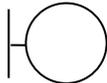
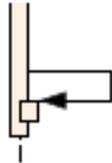
Tabel 2. Activity Diagram

Gambar	Nama	Keterangan
	<i>Start Point</i>	diletakan pada pojok kiri atas dan merupakan awal aktivitas.
	<i>End Point</i>	diletakan pada akhir aktivitas.
	<i>Activities</i>	menggambarkan suatu proses atau kegiatan bisnis.
	<i>Decision</i>	<i>Decision Points</i> , menggambarkan pilihan untuk mengambil keputusan, <i>true</i> atau <i>false</i> .
	<i>Fork</i> (Percabangan)	digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau dua kegiatan untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (Penggabungan)	<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi.
	<i>Swimlane</i>	<i>Swimlane</i> , pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.

c. *Sequence Diagram*

*Sequence Diagram* mengilustrasikan tingkah laku objek pada *use case* dengan menjelaskan detail waktu dan pesan yang dikirim ataupun diterima oleh objek. Simbol-simbol yang digunakan dalam *Sequence Diagram* yaitu:

Tabel 3. *Sequence Diagram*

Gambar	Nama	Keterangan
	<i>Entity Class</i>	Merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i>	Berisi kumpulan menjadi <i>interfaces</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>form entry</i> dan <i>form cetak</i> .
	<i>Control Class</i>	Suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i>	Simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i>	Menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i>	Mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi.



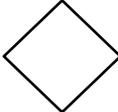
*Life Line*

Garis titik-titik yang terhubung dengan objek, sepanjang *lifeline* terdapat *activation*.

d. *Class Diagram*

*Class* diagram merupakan hubungan antar kelas dan penjelasan tiap-tiap kelas di dalam suatu model desain suatu sistem, *class* diagram juga terdapat aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sebuah sistem.

Tabel 4. *Class Diagram*

<b>Gambar</b>	<b>Nama</b>	<b>Keterangan</b>
	<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> ).
	<i>Nary Association</i>	Untuk menghindari asosiasi dengan lebih dari dua objek.
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
	<i>Depedency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung pada elemen yang

tidak mandiri.



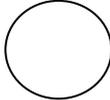
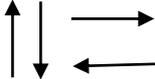
*Association*

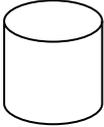
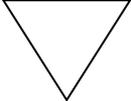
Menghubungkan antara objek satu dengan objek yang lainnya.

### 2.4.2. Mapping Chart

*Mapping chart* adalah sebuah diagram yang digunakan untuk mengilustrasikan alir data yang terkait dalam sistem serta berfungsi untuk memetakan gambaran prosedur-prosedur yang terjadi dalam sistem, (Sandikapura & Sukendar, 2018).

Tabel 5. Mapping Chart

Simbol	Nama	Keterangan
	<i>Processing symbol</i>	Menunjukkan proses yang dilakukan secara komputerisasi.
	<i>Connector symbol</i>	Menunjukkan alir dokumen yang terputus atau terpisah pada halaman alir dokumen yang sama.
	Simbol dokumen	Menunjukkan dokumen berupa dokumen <i>input</i> dan <i>output</i> pada proses manual dan proses berbasis komputer.
	Arah alir dokumen	Menunjukkan arah aliran dokumen antar bagian yang terkait pada suatu sistem. Bisa dari sistem keluar ataupun dari luar ke sistem dan antar bagian diluar sistem.

	Penyimpanan magnetik	Menunjukkan media penyimpanan data/informasi file pada proses berbasis komputer.
	Proses manual	Menunjukkan proses yang dilakukan secara manual.
	Pengarsipan	Menunjukkan simpanan dan non-komputer atau informasi file pada proses manual.
	<i>Input keyboard</i>	Menunjukkan <i>input</i> yang dimasukkan melalui keyboard
	Penyimpanan manual	Menunjukkan media penyimpanan data/informasi secara manual
	<i>Communication Link</i>	Untuk menyatakan bahwa adanya transisi suatu data atau informasi dari suatu lokasi ke lokasi lainnya (Budiman dkk., 2021)

## 2.5. Metode Pengembangan Sistem

### 2.5.1. *Rapid Application Development*

Metode pengembangan sistem yang akan digunakan yaitu metode RAD atau *Rapid Application Development*, RAD adalah model yang dipakai pada proses pembangunan sebuah perangkat lunak, metode ini dirancang dapat memberikan pengembangan yang jauh lebih cepat dan berkualitas. Metode ini terdiri 4 tahapan yaitu *Requirements Planning*, *User Design*, *Construction* dan *Cutover*, (Kusmiati & Ansori, 2018).

### **2.5.2. Tahap *Requirement Planning***

Tahap *requirement planning* adalah tahap pengumpulan data yang disesuaikan dengan penelitian yang akan dibuat, pada tahap ini dilakukan pengumpulan data yang dipergunakan untuk membangun sebuah sistem.

### **2.5.3. Tahap *User Design***

Pada tahap ini dilakukan perancangan desain aplikasi yang diawali dengan alur proses yang berjalan dan alur proses yang diusulkan serta perancangan database.

### **2.5.4. Tahap *Construction***

*Construction* merupakan merupakan tahap membangun sebuah sistem dengan merujuk dari hasil tahapan *user design* kedalam bahasa pemrograman.

### **2.5.5. Tahap *Cutover***

Pada tahap *cutover* akan dilakukan pengujian sistem, pengujian dilakukan dengan metode *black-box testing* untuk memastikan fitur pada sistem berfungsi dengan baik.