

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Penelitian Terdahulu**

Penelitian ini dilakukan tidak terlepas dari hasil penelitian-penelitian terdahulu yang pernah dilakukan sebagai bahan perbandingan dan kajian. Adapun hasil-hasil penelitian yang dijadikan perbandingan tidak terlepas dari topik penelitian yaitu mengenai sistem reservasi jasa *service* mobil dan metode *Remote Method Invocation* (RMI).

Berdasarkan penelitian Helmi Kurniawan (2014) dimana penelitian ini mengenai Perancangan Sistem Informasi Bengkel Mobil Berbasis Web. Hasilnya menunjukkan bahwa dalam proses melayani pelanggan yang datang untuk memperbaiki atau mengambil mobilnya mengalami berbagai masalah dalam melakukan transaksi. Transaksi tersebut terkadang tidak berjalan sesuai rencana sehingga menyebabkan berbagai keluhan dari pelanggan. Keluhan pelanggan yang terjadi misalnya pencatatan data pelanggan dan pembayaran sebuah transaksi yang masih dilakukan dengan menggunakan pencatatan-pencatatan didalam sebuah buku yang dapat menyebabkan proses transaksi tersebut terhambat. Disamping keluhan pelanggan tersebut keluhan juga terjadi pada pemberian informasi yang dibutuhkan oleh pelanggan. Dengan adanya perancangan sistem informasi bengkel mobil berbasis web ini masalah keluhan pelanggan bengkel mobil dapat diatasi dan dapat

meningkatkan pelayanan terhadap pelanggan. Sistem informasi bengkel mobil ini bertujuan untuk memberikan kemudahan kepada bengkel mobil dalam mengolah data menyajikan informasi perawatan dan perbaikan mobil.

Berdasarkan penelitian yang dilakukan Dharmasurya, dkk (2013) dimana melakukan penelitian mengenai Pengembangan Sistem Terdistribusi Untuk Sistem Informasi Administrasi Kependudukan Dengan Integrasi Teknologi *RMI* Dan *Web Service*. Hasilnya menunjukkan bahwa SIAK yang di bangun sudah memiliki standar operasional dan memiliki *user* antarmuka yang mudah dimengerti. Meski secara fisik tidak terlihat perbedaan dengan *web* biasa.

## **II.2. Aplikasi**

Aplikasi adalah suatu *subkelas* perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer, tapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media. Sedangkan program merupakan kumpulan *instruction set* yang akan dijalankan oleh pemroses, yaitu berupa *software*. Bagaimana sebuah sistem komputer berpikir diatur oleh program ini. Program inilah yang mengendalikan semua aktifitas yang ada pada pemroses. Program berisi

konstruksi logika yang dibuat oleh manusia, dan sudah diterjemahkan ke dalam bahasa mesin sesuai dengan format yang ada pada *instruction set*. (Santosa, 2015 : 7).

### **II.3. Reservasi**

Pemesanan dalam bahasa Inggris adalah *Reservation* yang berasal dari kata *to reserve* yaitu menyediakan atau mempersiapkan tempat sebelumnya. Sedangkan *reservation* yaitu pemesanan suatu tempat fasilitas. Jadi secara umum *reservation* yaitu pemesanan fasilitas yang diantaranya akomodasi, *meal*, *seat* pada pertunjukan, pesawat terbang, kereta api, bus, hiburan, *night club*, *discoutegue* dan sebagainya. Sehingga sistem informasi reservasi merupakan suatu sistem yang mengelola data pemesanan fasilitas dan menyajikan informasi fasilitas pada waktu tertentu. (Dwi Cahyono, 2013 : 2).

### **II.4. Android**

Android adalah sistem operasi yang dimodifikasi dari kernel Linux yang berbasis *open source* sehingga dapat digunakan oleh siapa saja. Teknologi Android dapat dimanfaatkan dalam berbagai kegiatan manusia dengan tujuan mempermudah, mulai dari mempermudah berkomunikasi, mempermudah belajar, sampai mempermudah dalam urusan bisnis. (Christine Dewi, Kumala Nindya Pramono, 2015: 1).

### **II.5. Remote Method Invocation (RMI)**

*Remote Method Invocation* (RMI) adalah salah satu bagian dari J2SE yang digunakan untuk membangun aplikasi terdistribusi menggunakan bahasa java. RMI

adalah kumpulan kelas dalam java yang digunakan untuk menangani pemanggilan (*invocation*) *method* secara jarak jauh (*metode*) dalam jaringan atau *internet*. RMI menggunakan prinsip pemrograman berorientasi objek dimana obyek satu dapat saling berkomunikasi dengan obyek lainnya. RMI terdiri dari *client* dan *server*. RMI *server* biasanya akan membuat beberapa *remote* obyek dan referensinya yang dapat diakses oleh RMI *client* akan membuat koneksi ke *server* dan meminta pemanggilan ke beberapa *remote* obyek sebagai *local* obyek. (Arinna Widyawati : 2014).

## **II.6. Client Server**

*Client-Server* merupakan suatu model komunikasi antara dua atau lebih komputer guna melakukan pembagian tugas (W. Komputer, 2010). Dalam perkembangan teknologi *Client-Server* terdapat beberapa arsitektur yang telah dikembangkan, yaitu 1-Tier (*standalone*), 2-Tier dan 3-Tier/*Multi-Tier* (W. Komputer, 2010). Pembangunan aplikasi yang memanfaatkan konsep komputasi tersebar telah digantikan oleh teknologi *Web Service* namun pada area yang masih memprioritaskan kecepatan adalah hal yang utama, maka konsep teknologi tersebar seperti Java RMI bisa lebih baik dan dapat diandalkan (A. Nugroho dan A. Ashari.2011). Teknologi Java RMI tidak hanya dapat dibangun dalam satu komputer melainkan ke banyak komputer. Dari hasil implementasi teknologi Java RMI terdapat keuntungan pengaksesan data yang cepat karena adanya pembagian fungsi antara RMI *server* dan RMI *client* (S. Lestari, M. Somantri, R. R. Isnanto, 2011). *Remote Method Invocation* (RMI) dapat didefinisikan sebagai sebuah fasilitas standar Java

yang berguna melakukan pemanggilan (*invocation*) suatu metode dari jarak jauh (*remote*) didalam jaringan (B. Susanto,2003). (Anthony : 2017 : 140).

## **II.7. Sistem**

Pengertian sistem sangatlah luas dan mempengaruhi semua aspek kehidupan. Sistem sangat diperlukan dalam melakukan kinerja yang baik dan terstruktur terhadap manajemen. Keterpaduan sistem ini memungkinkan terciptanya kerjasama untuk menghasilkan informasi yang cepat, tepat, dan akurat. Sistem dapat didefinisikan dengan 2 (dua) pendekatan, yaitu sistem yang menekankan pada prosedur dan sistem yang menekankan pada elemen komponennya. (Steven, 2015 : 6).

## **II.8. Informasi**

Informasi adalah data yang sudah diolah menjadi sesuatu yang bermanfaat untuk pemakainya. suatu informasi dapat dikatakan berkualitas jika memenuhi kualitas informasi yaitu akurat, tepat waktu dan relevan. (Steven, 2015 : 7).

## **II.9. Java**

*Java* adalah bahasa pemrograman yang berorientasi objek yang diciptakan oleh *Sun Microsystem* pada tahun 1995. Sebagai sebuah bahasa pemrograman, *Java* dapat membuat seluruh bentuk aplikasi, *desktop*, *web* dan lainnya, sebagaimana dibuat dengan menggunakan bahasa pemrograman konvensional yang lain. *Java* adalah bahasa pemrograman yang berorientasi objek (OOP) dan dapat dijalankan pada berbagai *platform* sistem operasi. Perkembangan *Java* tidak hanya terfokus pada

satu sistem operasi, tetapi dikembangkan untuk berbagai sistem operasi dan bersifat *open source*. (Wahyu Kurniawan, 2011 : 8).

## **II.10. Database**

Basis data (*database*) adalah suatu pengorganisasian sekumpulan data yang saling terkait sehingga memudahkan aktivitas untuk memperoleh informasi. Untuk mengolah basis data diperlukan perangkat lunak yang disebut DBMS (*database management sistem*). DBMS adalah perangkat lunak sistem yang memungkinkan para pemakai, membuat, memelihara, mengontrol dan mengakses basis data dengan cara yang praktis dan efisien. (Wahyu Kurniawan, 2011 : 8).

## **II.11. Database MySQL**

*My Structured Query Language* (MySQL) merupakan salah satu perangkat lunak sistem pengelola basis data *Data Base Management System* (DBMS). MySQL merupakan sebuah hubungan DBMS yang membantu sebuah model data yang terdiri atas kumpulan hubungan nama (*namedrelation*). *Database MySQL* adalah salah satu database yang *opensource*. *Database* ini banyak dipasangkan dengan *script PHP*. (Steven, 2015 : 17). Menurut Arief (2011:151) “MySQL (*My Structure Query Language*) adalah salah satu jenis *database server* yang sangat terkenal dan banyak digunakan untuk membangun aplikasi *web* yang menggunakan database sebagai sumber dan pengelolaan datanya”. My SQL bersifat *open source* dan menggunakan SQL (*Structured Query Language*). MySQL biasa dijalankan diberbagai *platform* misalnya *windows Linux*, dan lain sebagainya. (Eka, tias : 2016 : 131).

## **II.12. Hypertext Preprocessor**

Menurut Hidayatullah dan Kawistara (2014:231), PHP singkatan dari *Perl Hypertext Preprocessor* yaitu bahasa pemrograman *web server-side* yang bersifat *open source*. PHP merupakan *script* yang berintegrasi dengan HTML dan berada pada *server* (*server side HTML embedded scripting*). PHP adalah *script* yang digunakan untuk membuat halaman *web* dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru/*up to date*. Semua *script* PHP dieksekusi pada *server* dimana *script* tersebut dijalankan. (Eka, tias : 2016 : 130).

## **II.13. Hypertext Markup Language (HTML)**

Menurut Hidayatullah dan Kawistara (2014:13), *Hyper Text Markup Language* atau HTML adalah bahasa standar yang digunakan untuk menampilkan halaman *web*. Yang bisa dilakukan dengan HTML yaitu: mengatur tampilan dari halaman *web* dan isinya, membuat *table* dalam halaman *web*, mempublikasikan halaman *web* secara *online*, membuat *form* yang bisa digunakan untuk menangani registrasi dan transaksi via *web*, menambahkan objek-objek seperti citra, audio, video, animasi, *java applet* dalam halaman *web*, serta menampilkan area gambar (*canvas*) di *browser*. (Eka, tias : 2016 : 130).

#### **II.14. Extensible Markup Language (XML)**

*Extensible Markup Language (XML)* adalah sebuah bahasa markah untuk mendeskripsikan data. XML merupakan turunan (*subset*) atau versi ringkas dari *SGML (Standart Generalized Markup Language)*, sedangkan SGML merupakan sebuah standar *ISO* untuk format dokumen. SGML tidak berisi berupa *tag-tag* siap pakai seperti halnya bahasa HTML, melainkan berupa aturan-aturan standar dalam pembuatan *tag-tag* format dokumen. SGML banyak dipakai untuk mengelola dokumen dalam jumlah besar, frekuensi revisi tinggi dan dibutuhkan dalam beragam format tampilan. SGML jarang dipakai karena sangat rumit dan kompleks. XML dibuat dengan konsep yang lebih sederhana dan ringkas, tujuannya agar bisa dipakai sebagai aplikasi *desktop* dan jaringan internet. (Yenni dan Shamir, 2012 : 107).

#### **II.15. Netbeans**

NetBeans merupakan sebuah proyek kode terbuka yang sukses dengan pengguna yang sangat luas, komunitas yang terus tumbuh, dan memiliki hampir 100 mitra (dan terus bertambah). *Sun Microsystems* mendirikan proyek kode terbuka *NetBeans* pada 5 bulan Juni 2000 dan terus menjadi sponsor utama. Saat ini terdapat dua produk : *NetBeans IDE* dan *NetBeans Platform*. (Agus Nurcahyono, 2012 : 4).

#### **II.16. Unified Modeling Language (UML)**

UML (*Unified Modeling Language*) merupakan bahasa standar pemodelan visual dalam rekayasa *software*, memberikan cara standar untuk mengembangkan

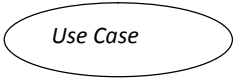
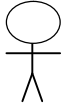
model bagi *software* yang akan dibangun dan memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa. Penggunaan UML dapat mengatasi kerumitan dalam segi arsitektural sistem *software* yang sedang dibangun. Didalam komponen UML terdapat diagram untuk menggambarkan proses bisnis dan data didalamnya. (Tri Pujadi, 2011 : 583).



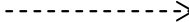

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut:

1. *Use case* Diagram

*Use case* diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada tabel II.1. dibawah ini:

**Tabel II.1. Simbol *Use Case* Diagram**

Gambar	Keterangan
	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal <i>frase</i> nama <i>use case</i> .
	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat sendiri. Meskipun simbol




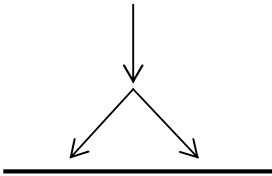
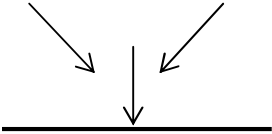
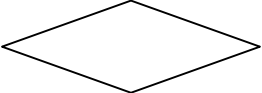

	dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
	Komunikasi antar aktor dan use case yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.
	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri meskipun tanpa <i>use case</i> tambahan tersebut, mirip dengan prinsip inheritance pada pemrograman berorientasi objek, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> .

(Sumber : Andry Setiawan, dkk; 2013 : 4)

## 2. Diagram Aktivitas (*Activity Diagram*)

*Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.2 dibawah ini:

Tabel II.2. Simbol *Activity Diagram*

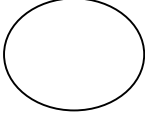
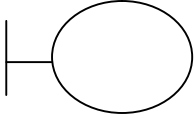
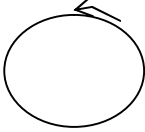
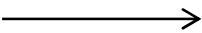
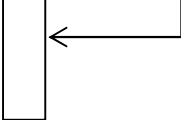


Gambar	Keterangan
	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

(Sumber : Andry Setiawan ; 2013 : 5)

### 3. Diagram Urutan (*Sequence Diagram*)

*Sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel II.3. dibawah ini :

**Tabel II.3. Simbol *Sequence Diagram***

Gambar	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 kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.
	<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> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar; 2015 : 95)

#### 4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung

jawab entitas yang menentukan perilaku sistem. *Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti yang dapat dilihat pada tabel II.4 dibawah ini:

**Tabel II.4. Multiplicity Class Diagram**

<b>Multiplicity</b>	<b>Penjelasan</b>
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar; 2015 : 95)