

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Adapun penelitian terkait yang akan digunakan sebagai sumber acuan yang relevan dan terkini yaitu:

Berdasarkan penelitian yang dilakukan oleh Nindi Werstania (2018) dengan judul “Perancangan Aplikasi Pemesanan *Catering* Berbasis *Website*” Pemesanan *catering* merupakan suatu aktivitas yang dilakukan pelanggan untuk membeli produk berupa paket makanan. Di Cimahi *Catering* proses pemesanan *catering* dilakukan oleh pelanggan dengan datang langsung ke lokasi atau menelpon ke Cimahi *Catering*.

Dalam hal pencatatan data pemesanan pun dirasa kurang efektif karena pendataannya yang masih bersifat konvensional. Hal ini beresiko terjadinya kesalahan pendataan.

Sebagai solusi dari permasalahan tersebut penulis membangun sebuah Aplikasi Pemesanan *Catering* Berbasis *Website* untuk membantu dalam proses pemesanan *catering* oleh pelanggan serta pengelolaan data pemesanan *catering* oleh pihak Cimahi *Catering*.

Berdasarkan penelitian yang dilakukan Rita Irviani (2019) dengan judul *Perancangan E-Commerce Berbasis Android Pada Kelompok Swadaya Masyarakat Desa Margakarya Pringsewu*, menyimpulkan bahwa pengujian metode *whitebox* pada pembahasan diperoleh hasil kepuasan pelanggan sistem *e-commerce* penjualan handphone sebesar 80%.

Berdasarkan penelitian yang dilakukan oleh Rita Irviani (2019) dengan judul *Aplkasi Berbasis Android untuk Penjualan OS Smartphone Dan Internasioanl Menggunakan SMS Gateway*, menyimpulkan bahwa Aplikasi dapat menampilkan pemesanan market share OS Smartphone yang berasal dari *sumber: IDC, Indonesia* menjadi bentuk yang sesuai pada perangkat *smartphone*.

Berdasarkan penelitian yang dilakukan Rendy Ryan, dkk (2018) dengan judul *Aplikasi Pemesanan Air Mineral Berbasis Android Pada PT. Citra Golden Tunggal Pangkalpinang*, Pada penelitian ini, dibangun sebuah aplikasi pada telepon seluler pintar berbasis Android. Tujuannya adalah untuk menjawab kebutuhan akan teknologi yang memudahkan dalam berbelanja secara praktis dan *mobile*.

Model pengembangan aplikasi dengan menggunakan waterfall, dan menggunakan metode berorientasi objek untuk pengembangan aplikasinya. Pengujian dilakukan dengan menggunakan teknik *Blackbox*. Hasil yang diharapkan dengan adanya aplikasi ini dapat mempermudah pelanggan untuk melakukan pemesanan.

Berdasarkan penelitian dari Roberto kaban (2019) dengan judul “Aplikasi Pemesanan Tiket Bus Berbasis Android (Study Kasus: PT. Als Terminal Pasar X Tanjung Beringin)” Penelitian ini bertujuan untuk merancang aplikasi pemesanan tiket agar memudahkan petugas loket dan calon penumpang dalam proses pemesanan tiket. Dalam tahap pengumpulan data, penulis melakukan studi lapangan, observasi langsung dan wawancara pada petugas loket PT. ALS Terminal Pasar X Tanjung Beringin. Perancangan aplikasi ini menggunakan metode desain berorientasi objek dengan UML (Unified Modeling Language). Aplikasi berbasis android ini dibangun dengan bahasa pemrograman Java menggunakan Android studio dan pada petugas loket untuk mengelola pemesanan tiket menggunakan bahasa pemrograman PHP, Bootstrap sebagai interface dan MariaDB sebagai database.

Berdasarkan penelitian dari Janifer W Janis, dkk (2020) dengan judul “Rancang Bangun Aplikasi Online Sistem Pemesanan Jasa Tukang Bangunan Berbasis Lokasi” Tujuan penelitian ini yaitu merancang aplikasi online sistem pemesanan jasa tukang bangunan berbasis lokasi untuk mendukung permasalahan kebutuhan masyarakat dan tukang bangunan dalam pekerjaan pembangunan.

Dengan adanya aplikasi ini, masyarakat (selaku pemesan jasa) dapat dengan mudah melakukan pemesanan jasa tukang bangunan berdasarkan lokasi secara online (terkoneksi internet).

Adapun perbedaan penelitian yang dilakukan penulis dengan penelitian terkait sebelumnya yaitu pada penelitian ini penulis menggunakan *Aplikasi website* untuk melakukan penyewaan bus pada PT. Sinabung Jaya Bus.

Sehingga dengan adanya aplikasi ini, dapat memudahkan *customer* dalam melakukan penyewaan bus dan mempercepat proses penyewaan pada PT. Sinabung Jaya Bus.

II.2. Landasan Teori

II.2.1. Penyewaan

Pengertian sewa menurut kamus besar bahasa Indonesia (departemen pendidikan dan kebudayaan republik Indonesia.2001:8 :33) adalah pemakaian sesuatu dengan membayar uang sewa, uang yang dibayarkan karena memakai atau meminjamkan sesuatu, yang boleh pakai dengan membayar uang dengan uang.

Sedangkan pengertian penyewaan adalah proses, cara, pembuatan menyewa atau menyewakan.yang dimaksud dengan sewa, yaitu balas jasa atas sewa ruang atau ruangan dalam keadaan kosong yang dapat ditagih dimuka (pada awal pemakaian mobil) atau dibelakang, sesuai dengan kontrak (perjanjian).

Sumber: (Sumardi Subianto AMIK Jakarta Teknologi Cipta Semarang)

II.2.2. Bus

Pengertian Bus Menurut Peraturan Menteri Perhubungan Republik Indonesia Nomor 26 Tahun 2017 tentang Penyelenggaraan Angkutan Orang dengan Kendaraan Bermotor Umum tidak dalam Trayek, definisi dari bus adalah kendaraan bermotor angkutan orang yang memiliki tempat duduk lebih dari 8 orang, termasuk pengemudi yang beratnya lebih dari 3.500 kg. Menurut Peraturan Menteri Perhubungan Republik Indonesia Nomor 26 Tahun 2017 tentang Penyelenggaraan Angkutan Orang dengan Kendaraan Bermotor Umum tidak dalam Trayek, jenis-jenis bus ada 4 yaitu :

- a) Bus kecil adalah kendaraan bermotor angkutan orang beratnya lebih dari 3.500 – 5.000 kg, panjangnya maksimal 9000 milimeter lebar tidak lebih 2.100 milimeter dan tinggi 1,7 kali lebar kendaraan.
- b) Bus sedang adalah kendaraan bermotor angkutan orang beratnya lebih dari 5.000 – 8.000 kg, panjangnya maksimal 6000 milimeter lebar tidak lebih 2.100 milimeter dan tinggi 1,7 kali lebar kendaraan.
- c) Bus besar adalah kendaraan bermotor angkutan orang beratnya lebih dari 8.000 – 16.000 kg, panjangnya lebih dari 9000 milimeter lebar tidak lebih 2.500 milimeter dan tinggi kendaraan tidak lebih dari 4.200 milimeter dan tidak lebih dari 1,7 kali lebar kendaraan.
- d) Bus tingkat adalah kendaraan bermotor angkutan orang beratnya paling sedikit 21.000 - 24.000 kg, panjangnya paling sedikit 9000 - 13.500 milimeter lebar tidak lebih 2.500 milimeter dan tinggi kendaraan tidak lebih dari 4.200 milimeter. Sumber:(poltekkesjogja.ac.id)

II.2.3. Aplikasi

Pengertian aplikasi adalah suatu bagian dari perangkat lunak yang dapat digunakan untuk menyelesaikan masalah-masalah yang khusus yang dihadapi *user* dengan menggunakan kemampuan komputer. (Fergiawan Listianto, dkk : 2017).

Aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputasi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan. (Ari Setiaji, dkk : 2020).

II.2.4. Website

Website merupakan sebuah terobosan baru sebagai teknologi informasi yang menghubungkan data dari banyak sumber dan layanan yang beragam macamnya di internet, *web* cepat sekali populer di lingkungan pengguna internet, karena kemudahan yang diberikan kepada pengguna internet untuk melakukan penelusuran, penjelajahan, dan pencarian informasi (Priyo Sutopo, dkk, 2016: 24).

II.2.5. *Laravel*

Laravel adalah sebuah Framework PHP dirilis dibawah lisensi MIT dengan kode sumber yang sudah disediakan oleh Github, sama seperti framework-framework yang lain. (Aminudin, 2015:1).

II.2.6. *MySQL*

MySQL adalah salah satu jenis *database server* yang sangat terkenal. Kepopulerannya disebabkan *MySQL* menggunakan *SQL* sebagai bahasa dasar untuk mengakses databasenya. *MySQL* termasuk jenis *RDBMS (Relational Database Management System)*. Pada *MySQL*, sebuah database mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom. (Priyo Sutopo, dkk : 2017).

MySQL (My Structure Query Language) merupakan sebuah program pembuat database yang bersifat *Open Source*, artinya semua orang dapat menggunakannya dan dapat dijalankan pada semua *platform* baik *Windows* maupun *linux*. *MySQL* juga merupakan sebuah perangkat lunak sistem manajemen basis data *SQL* yang bersifat jaringan sehingga dapat digunakan untuk aplikasi multi *user*.

MySQL juga sering dikenal dengan nama sistem manajemen *database* relasional. Suatu *database* relasional menyimpan data dalam table yang terpisah. Tabel -table tersebut terhubung oleh suatu relasi terdefinisi yang memungkinkan memperoleh kombinasi data dari beberapa table dalam suatu permintaan. Untuk administrasi *database*, seperti pembuatan *database*, pembuatan tabel, dan

sebagainya dapat digunakan aplikasi berbasis web seperti *PHP MyAdmin* dengan aplikasi *XAMPP*. (Saipul Anwar, 2016).

II.2.7. PHP

PHP adalah sebuah bahasa pemrograman yang berjalan dalam sebuah *web-server* (*server-side*). PHP diciptakan oleh *programmer* unix dan Perl yang bernama Rasmus Lerdoft pada bulan Agustus-September 1994. *Script*

PHP adalah bahasa program yang berjalan pada sebuah *web server*, atau sering disebut *server-side*. oleh karena itu, PHP dapat melakukan apa saja yang bisa dilakukan program CGI lain, yaitu mengolah data dengan tipe apapun, menciptakan halaman *web* yang dinamis. (Harison dan Syarif, 2016 : 42).

II.2.8. Visual Studio Code

Visual Studio Code adalah editor source code yang dikembangkan oleh Microsoft untuk Windows, Linux dan MacOS. Ini termasuk dukungan untuk debugging, GIT Control yang disematkan, penyorotan sintaks, penyelesaian kode cerdas, cuplikan, dan kode refactoring.

Hal ini juga dapat disesuaikan, sehingga pengguna dapat mengubah tema editor, shortcut keyboard, dan preferensi. Visual Studio Code gratis dan open-source, meskipun unduhan resmi berada di bawah lisensi proprietary. Kode Visual Studio didasarkan pada Elektron, kerangka kerja yang digunakan untuk menyebarkan aplikasi Node.js untuk desktop yang

berjalan pada Blinklayout. Meskipun menggunakan kerangka Elektron, Visual Studio Code tidak menggunakan Atom dan menggunakan komponen editor yang sama (diberi kode nama "Monaco") yang digunakan dalam Visual Studio Team Services yang sebelumnya disebut Visual Studio Online (Lardinois, 2015) .

II.2.9. UML (*Unified Modelling Language*)

Unified Modelling Language (UML) merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem software yang terkait dengan objek. UML merupakan salah satu alat bantu yang sangat handal dalam bidang pengembangan sistem berorientasi objek karena UML menyediakan Bahasa pemodelan visual yang memungkinkan pengembang sistem membuat *blue print* atas visinya dalam bentuk yang baku. UML berfungsi sebagai jembatan dalam mengkomunikasikan beberapa aspek dalam sistem melalui jumlah elemen grafis yang bisa dikombinasikan menjadi.

Unified Modeling Language (UML) biasa digunakan untuk :

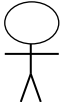
- a. Menggambarkan batasan sistem dan fungsi - fungsi sistem secara umum, dibuat dengan *use case* dan *actor*.
- b. Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum, dibuat dengan *interaction diagrams*.
- c. Menggambarkan representasi struktur *static* sebuah sistem dalam bentuk *class diagrams*.
- d. Membuat model behavior “yang menggambarkan kebiasaan atau sifat sebuah sistem” dengan *state transition diagrams*.

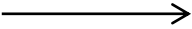
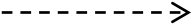
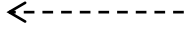
- e. Menyatakan arsitektur implementasi fisik menggunakan *component and development*.
- f. Menyampaikan atau memperluas *fungsi* dengan *stereotypes*. (Ommi Alfina dan Fitriana Harahap : 2019)

II.2.9.1. Use Case Diagram

Use case diagram merupakan pemodelan untuk sistem informasi yang akan dibuat. *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.2 di bawah ini :

Tabel II.2. Simbol Use Case

Gambar	Keterangan	Deskripsi
	<i>Use case</i>	Menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor	Sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem.
	Asosiasi	Penghubung antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.

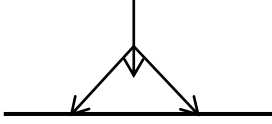
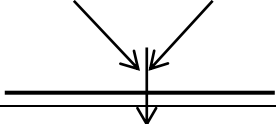
	Asosiasi	Penghubung antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i>	Merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i>	Merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

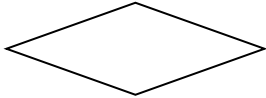
(Sumber : Ade Hendini, 2016)

II.2.9.2. Activity Diagram

Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat di lihat pada tabel II.3 dibawah ini:

Tabel II.3. Simbol Activity Diagram

Gambar	Keterangan	Deskripsi
	<i>Start point</i>	Diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i>	Akhir aktifitas.
	<i>Activites</i>	Menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan).	Digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	<i>Join</i> (penggabungan)	Digunakan untuk menunjukkan adanya dekomposisi.

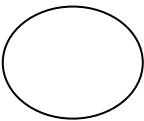
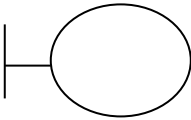
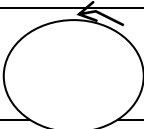
	<i>Decision Points</i>	Menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i>	Pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.

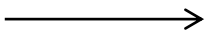
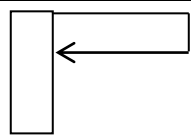
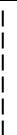
(Sumber : Ade Hendini, 2016 : 109)

II.2.9.3. 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.4 di bawah ini :

Tabel II.4. Simbol *Sequence Diagram*

Gambar	Keterangan	Deskripsi
	<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>	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 : Ade Hendini, 2016 : 110)

II.2.9.4. Class Diagram

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 (*operation/method*) dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality* (Ade Hendini, 2016 : 111).

Tabel II.5. *Multiplicity Class Diagram*

Multiplicity	Penjelasan
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 : Ade Hendini, 2016 : 110)