

## **BAB II**

### **TINJAUAN PUSTAKA DAN DASAR TEORI**

#### **2.1 Tinjauan Pustaka**

Penelitian tentang Strategi e-mail blast dan SMS blast dalam e-public relations (studi deskriptif kualitatif tentang penggunaan e-mail blast dan SMS blast sebagai media promosi dalam berhubungan e-public relations di Aston Hotel Solo). Aplikasi dapat menampilkan menu login. Perancangan sistem menggunakan UML (Unified Modeling Language) yaitu use case diagram, class diagram, activity diagram, dan sequence diagram (Suci Ati, 2017).

Penelitian tentang Pengaruh Promosi SMS Broadcast terhadap Peningkatan Volume Penjualan Dunkin Donut Mall Panakkukang Makassar. Sistem dapat menampilkan hasil berpengaruh terhadap peningkatan volume penjualan. Perancangan sistem menggunakan UML (Unified Modeling Language) yaitu use case diagram, class diagram, activity diagram, dan sequence diagram (Indriani, 2018).

Penelitian tentang Penerapan Teknologi Cloud Computing Dan Line Bot Sebagai Media Promosi Dan Transaksi Produk Umkm Di Kota Bandung. Aplikasi dapat menampilkan menu produk beserta harganya, mempunyai form untuk pemesanan menu produk, dan menampilkan data pesanan beserta total harganya. Perancangan sistem menggunakan UML (Unified Modeling Language) yaitu use case

diagram, class diagram, activity diagram, dan sequence diagram (Lukman, Rangga Djatikusuma 2019).

Penelitian Pemanfaatan Media Facebook Dan Whatsapp Sebagai Promosi Untuk Meningkatkan Penjualan Di UMKM GPS SABLON. Sistem dapat menampilkan hasil berpengaruhnya terhadap peningkatan volume penjualan. Perancangan sistem menggunakan UML (Unified Modeling Language) yaitu use case diagram (Nurul huda ,Okta Vian., 2019).

Penelitian tentang Aplikasi Penjualan, Pembelian Dan Promosi Aksesoris Handphone Dengan SMS Gateway Berbasis Web Pada Center Cell Accessories Banjarmasin. Aplikasi dapat menampilkan menu produk beserta harganya, mempunyai form untuk pemesanan menu produk, dan menampilkan data pesanan beserta total harganya. Perancangan sistem menggunakan UML (Unified Modeling Language) yaitu use case diagram, class diagram, activity diagram, dan sequence diagram (Saputra, Febry, 2021).

Penelitian saat ini yaitu tentang Sistem Informasi Dan promosi produk makanan ringan keripik R&R dengan media whatsapp blast. Perancangan sistem menggunakan UML (Unified Modeling Language) yaitu use case diagram, class diagram, activity diagram, dan sequence diagram. Aplikasi ini mempunyai fitur login, mengunggah promosi, menampilkan semua data yang sudah diunggah dalam bentuk tabel, mempunyai fitur untuk melakukan kirim pesan.

Tabel 2.1. Perbedaan kedalaman penelitian berdasarkan tinjauan pustaka.

Acuan		Komponen		
Nama Peneliti	Tahun	Objek	Metodeloagi	Hasil
Suci Ati	2017	Strategi e-mail blast dan SMS blast dalam e-public relations (studi deskriptif kualittaif tentang penggunaan e-mail blast dan SMS blast sebagai media promosi dalam berbuhungan e-public relations di Aston Hotel Solo)	Sms Blast	Meningkatkan pelanggan
Indriani	2018	Pengaruh promosi sms broadcast terhadap peningkatan penjualan dunkin donut mall panakkukang Makassar	Sms Broadcast	Bertujuan untuk mengetahui apakah Promosi SMS Broadcast berpengaruh terhadap Peningkatan Volume Penjualan

Lukman, Rangga Djatikusuma	2019	Penerapan teknologi cloud computing Dan Line Bot sebagai media promosi dan transaksi produk umkm Di Kota Bandung	Teknologi Cloud Computing Dan Line Bot	Meningkatkan Pelanggan
Nurul huda ,Okta Vian.	2019	Pemanfaatan media facebook dan whatsapp sebagai promosi untuk meningkatkan penjualan di umkm gos sablon	Facebook dan whatsapp	Meningkatkan Pelanggan
Saputra, Febry	2021	Aplikasi penjualan, pembelian dan promosi aksesoris handphone dengan sms gateway berbasis web pada center cell accessories Banjarmasin	Sms Gateway	Meningkatkan pelanggan

<b>Penelitian saat ini</b>	<b>2021</b>	Sarana promosi menggunakan whatsapp blast	Whatsapp BLAST	Pemberitahuan item produk dan sarana promosi
--------------------------------	-------------	--	----------------	---

## 2.2 Dasar Teori

### 2.2.1 Whatsapp Blast

*Whatsapp blast* sendiri adalah aplikasi pengiriman pesan *whatsapp* secara massal tanpa perlu menyimpan nomor sebelumnya dikontak pengirim. *Whatsapp blast* adalah aplikasi pihak ketiga yang dikembangkan oleh developer, menurut Irfan juniyanto (Firmansah, D A., 2020).

### 2.2.2 Framework Laravel

Laravel adalah *framework* aplikasi web kontemporer, open source dan digunakan secara luas untuk perancangan aplikasi web yang cepat dan mudah. Laravel dibuat oleh Taylor Otwell dan berada dibawah lisensi MIT. Framework ini dibuat untuk pengembangan website mengikuti konsep MVC (Model, View, Controller). Konsep MVC merupakan suatu konsep yang memisahkan pengembangan aplikasi berdasarkan komponen- komponen utama yang membangun sebuah aplikasi seperti manipulasi data, *user interface* dan cara memproses datanya. Komponen pola MVC terdiri dari.

1. Model mewakili struktur data pada sistem aplikasi. Model berisi fungsi-fungsi yang membantu dalam memanipulasi data di dalam basis data (insert, update, delete, search).
2. View adalah bagian yang mengatur tampilan halaman web atau user interface.

3. Controller merupakan bagian yang menghubungkan model dan view. Controller berisi perintah-perintah yang berfungsi untuk memproses suatu data dan mengirimkannya ke halaman web.

Laravel mempunyai sintaks yang ekspresif, jelas dan menghemat waktu. Framework ini dikembangkan dengan tujuan bahwa pengembangan web harus dapat dinikmati dan penuh kreatifitas. Pengembangan web dengan Laravel mempermudah proses pengembangan web dengan mempermudah tugas-tugas yang umum seperti routing, authentication, sessions, dan caching. (socs.binus.ac.id, 2018).

### **2.2.3 Vue JS**

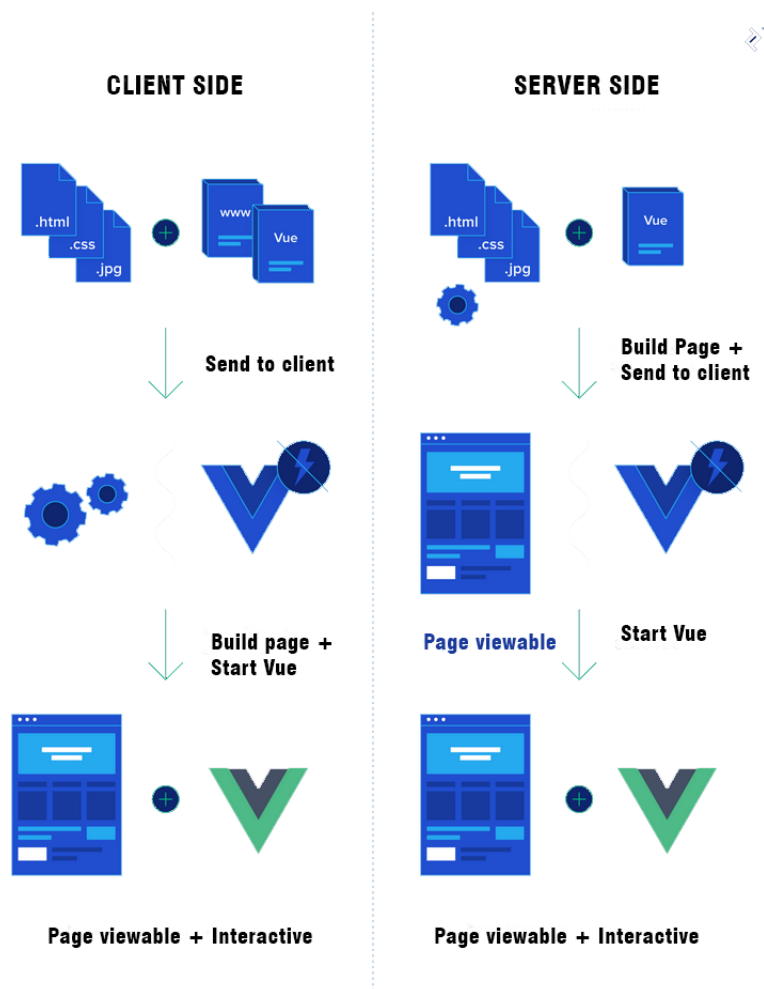
Vue.js (dibaca: vuuu atau viuu) adalah sebuah kerangka kerja nan progresif untuk membangun antarmuka pengguna. Tidak seperti beberapa kerangka kerja monolitik yang lain, Vue dirancang dari dasar sekali agar dapat diadopsi secara bertahap. Vue.js merupakan proyek open-sourcedengan lisensi MIT yang diciptakan oleh Evan You pada bulan Februari 2014. Salah satu fitur yang ditawarkan oleh Vue.js adalah System Reactive Data Binding yang berfungsi agar data dan DOM (Document Object Model) tetap terikat bersama-sama. (<https://docs.vuejs.id/v2/guide/>, 2016).

#### **2.2.3.1 Server-Side Rendering (SSR)**

Secara default, komponen Vue memproduksi dan memanipulasi DOM di browser sebagai output. Namun, juga dimungkinkan untuk merender komponen yang sama ke dalam string HTML di server, mengirimkannya langsung ke browser,

dan akhirnya "menghidrasi" markup statis menjadi aplikasi yang sepenuhnya interaktif di klien.

Server Side Rendering atau biasa di sebut SSR adalah teknik untuk merender Single Page Application atau biasa disebut SPA yang akan dirender di sisi server lalu semua hasil render tersebut dikirimkan ke sisi klien. (<https://ssr.vuejs.org/#what-is-server-side-rendering-ssr> , 2016)



Gambar 2.1 skema vue js ssr



#### 2.2.4 UML

Salah satu model perancangan berorientasi objek yang saat ini sering digunakan di seluruh dunia adalah *Unified Modelling Language* (UML). UML sendiri adalah bahasa pemodelan untuk sistem atau perangkat lunak yang memiliki paradigma berorientasi objek. *Unified Modelling Language* (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML, perancang atau pemodel dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. UML lebih cocok diterapkan pada piranti berorientasi objek seperti C++, Java, C#, dan VB. NET. Walaupun demikian, UML juga tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C. (Sugiarti, Yuni., 2013).

Pada perancangan sistem aplikasi ini, pemodelan yang digunakan adalah *Unified Modeling Language* (UML). UML adalah sebuah bahasa visual yang menjelaskan, memberikan spesifikasi, merancang, membuat dan mendokumentasikan aspek-aspek dari sebuah sistem piranti lunak. Karena tergolong bahasa visual, UML lebih mengedepankan penggunaan diagram untuk menggambarkan aspek dari sistem yang sedang dimodelkan.

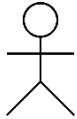
Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. UML sangat cocok digunakan membuat model dari piranti lunak yang ditulis dalam bahasa-bahasa pemrograman berorientasi objek seperti C++, C#, Java atau Dart. (Yuni Sugiarti, S.T., M.Kom., 2013).



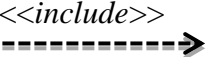
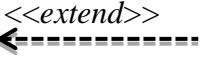
Diagram-diagram yang digunakan untuk memodelkan sistem aplikasi dalam penelitian ini digambarkan seperti dibawah :

**a. Use Case Diagram**

*Use Case Diagram* secara grafis menggambarkan interaksi antara sistem, sistem eksternal dan pengguna. Dengan kata lain *Use Case Diagram* secara grafis mendeskripsikan siapa yang akan menggunakan sistem dan dalam cara apa pengguna (user) mengharapkan interaksi dengan sistem itu. Komponen-komponen yang digunakan untuk menggambarkan *Use Case Diagram* dijabarkan dalam tabel 2.1.

**Tabel 2. 2 Komponen-komponen Use Case Diagram**

Notasi	Deskripsi
 <p data-bbox="488 1696 566 1728">Aktor</p>	<p data-bbox="699 1551 1183 1585">Entitas eksternal (di luar sistem) yang</p> <p data-bbox="699 1625 1110 1659">berinteraksi dengan sistem yang</p> <p data-bbox="699 1698 1240 1732">berpartisipasi dalam use case. Bisa berupa</p> <p data-bbox="699 1772 1078 1806">orang, pengguna, atau sistem.</p>

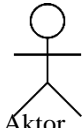

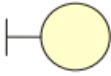




 <i>Use Case</i>	<p>Menjabarkan aktifitas aktor dalam sistem yang memberikan hasil yang bisa dilihat.</p>
 <i>Association</i> <i>Relationship</i>	<p>Relasi Use Case dengan Actor disebut Asosiasi. Asosiasi tidak menggambarkan aliran data. Arah tanda panah berasal dari yang siapa menginisiasi.</p>
	<p>Relasi yang menspesifikasikan sebuah Use Case yang berada di dalam Use Case lain. Artinya penyertaan use case lain yang wajib atau diharuskan pada sistem. Tanda panah terbuka mengarah ke sub use case.</p>
	<p>Relasi yang merupakan perluasan dari use case jika terdapat kondisi atau syarat terpenuhi dan memasukkan perilaku opsional. Tanda panah terbuka mengarah ke parent/base Use Case.</p>


### b. Sequence Diagram

Sequence Diagram merupakan suatu diagram yang menampilkan interaksi-interaksi antara objek di dalam sistem yang akan dibuat. *Sequence Diagram* digunakan untuk menjelaskan dan menggambarkan secara lebih rinci proses yang

terjadi pada sebuah sistem. Komponen-komponen yang digunakan untuk menggambarkan *Sequence Diagram* dijabarkan dalam tabel 3.2 berikut.

**Tabel 2. 3 Komponen-komponen Sequence Diagram**




Notasi	Deskripsi
 <p>Aktor</p>	Menggambarkan pengguna atau orang yang berinteraksi dengan sistem.
 <p>Lifeline</p>	Sebuah garis yang menggambarkan masa hidup dari sebuah objek dalam sequence diagram.
 <p>Boundary Class</p>	Menggambarkan tepi dari sistem, seperti user interface atau suatu alat yang berinteraksi dengan sistem.
 <p>Control Class</p>	Mengatur aliran dari informasi untuk sebuah skenario. Objek ini umumnya mengatur perilaku dan perilaku bisnis.
 <p>Entity Class</p>	Elemen yang bertanggung jawab menyimpan data atau informasi.
 <p>Activation</p>	Mewakili sebuah eksekusi operasi dari obyek, panjang obyek ini berbanding lurus dengan durasi aktivasi sebuah operasi. Simbol ini juga menunjukkan kapan sebuah objek mengirim atau menerima kejadian.
	Menggambarkan pesan atau hubungan antar objek yang menunjukkan urutan kejadian yang terjadi.

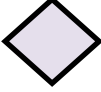

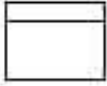
Message	
	Sebuah obyek yang mempunyai sebuah operasi kepada dirinya sendiri.

### c. Activity Diagram

*Activity Diagram* merupakan diagram yang dapat memodelkan proses-proses yang terjadi pada sebuah sistem. Runtutan proses dari suatu sistem digambarkan secara vertikal. *Activity Diagram* merupakan pengembangan dari *Use Case* yang memiliki alur aktivitas. Komponen-komponen yang digunakan untuk menggambarkan *Activity Diagram* dijabarkan dalam tabel 3.3 dibawah ini.

**Tabel 2. 4 Komponen-komponen Activity Diagram**

Notasi	Deskripsi
 Initial State	Awal dimulainya suatu aliran kerja pada activity diagram.
 Final State	Akhir dari aliran kerja pada pada sebuah activity diagram.
 Activity	Activity merupakan aktivitas yang dilakukan atau sedang terjadi dalam sistem.
	Suatu titik atau point yang mengindikasikan

 Decision atau Percabangan	suatu kondisi di mana adanya kemungkinan dalam perbedaan transisi. Hal tersebut diperlukan ketika sistem yang dimiliki memiliki beberapa kemungkinan atau jalan alternatif.
 Transition	Digunakan untuk menghubungkan aktivitas selanjutnya dan sebelumnya.
 Swimlanes	Memecah activity diagram menjadi kolom dan baris untuk membagi tanggung jawab objek-objek yang melakukan suatu aktivitas.

### 1.1.1 Metode Waterfall

Metode penelitian yang diterapkan pada penelitian ini adalah dengan pengembangan metode *waterfall*. Metode *waterfall* merupakan model pengembangan sistem informasi yang sistematis dan sekuensial. Metode *waterfall* memiliki tahapan-tahapan sebagai berikut :

1. *Requirements analysis and definition*

Layanan sistem, kendala, dan tujuan ditetapkan oleh hasil konsultasi dengan pengguna yang kemudian didefinisikan secara rinci dan berfungsi sebagai spesifikasi sistem.

2. *System and software design*

Tahapan perancangan sistem mengalokasikan kebutuhan-kebutuhan sistem baik perangkat keras maupun perangkat lunak dengan membentuk arsitektur

sistem secara keseluruhan. Perancangan perangkat lunak melibatkan identifikasi dan penggambaran abstraksi sistem dasar perangkat lunak dan hubungannya.

3. *Implementation and unit testing*

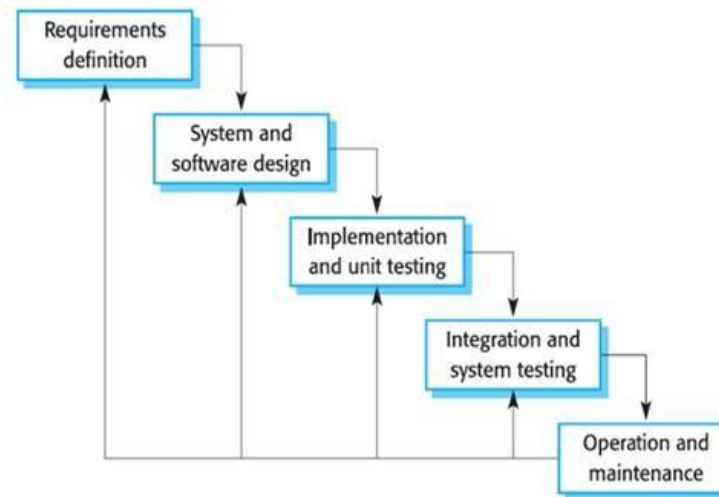
Pada tahap ini, perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Pengujian melibatkan verifikasi bahwa setiap unit memenuhi spesifikasinya.

4. *Integration and system testing*

Unit-unit individu program atau program digabung dan diuji sebagai sebuah sistem lengkap untuk memastikan apakah sesuai dengan kebutuhan perangkat lunak atau tidak. Setelah pengujian, perangkat lunak dapat dikirimkan ke *customer*.

5. *Operation and maintenance*

Biasanya (walaupun tidak selalu), tahapan ini merupakan tahapan yang paling panjang. Sistem dipasang dan digunakan secara nyata. *Maintenance* melibatkan pembetulan kesalahan yang tidak ditemukan pada tahapan-tahapan sebelumnya, meningkatkan implementasi dari unit sistem, dan meningkatkan layanan sistem sebagai kebutuhan baru.



**Gambar 1.2 Bagan Metode Waterfall**

Gambar 2.2 adalah bagan metode *waterfall* yang merupakan metode pengembangan sistem yang digunakan pada penelitian ini. (Ginanjari Wiro Sasmito, 2017).

### 1.1.2 Black-box Testing

Pada *Black Box Testing* dilakukan pengujian yang didasarkan pada detail aplikasi seperti tampilan aplikasi, fungsi-fungsi yang ada pada aplikasi, dan kesesuaian alur fungsi dengan bisnis proses yang diinginkan oleh customer. *Black-box Testing* ini lebih menguji ke Tampilan Luar(Interface) dari suatu aplikasi agar mudah digunakan oleh Customer. Pengujian ini tidak melihat dan menguji source code program. *Black-box Testing* bekerja dengan mengabaikan struktur kontrol sehingga perhatiannya hanya berfokus pada informasi domain. (socs.binus.ac.id, 2020).



Keuntungan penggunaan metode *Blackbox Testing* adalah :

1. Penguji tidak perlu memiliki pengetahuan tentang bahasa pemrograman tertentu
2. Pengujian dilakukan dari sudut pandang pengguna, ini membantu untuk mengungkapkan ambiguitas atau inkonsistensi dalam spesifikasi persyaratan.
3. Programmer dan tester keduanya saling bergantung satu sama lain.

Kekurangan dari metode *Blackbox Testing* adalah :

1. Kemungkinan memiliki pengulangan tes yang sudah dilakukan oleh *programmer*.
2. Beberapa bagian back end tidak diuji sama sekali. (Tri Snadhika Jaya, 2018).

### **1.1.3 White-Box Testing**

*White Box Testing* adalah salah satu cara untuk menguji suatu aplikasi atau software dengan melihat modul untuk memeriksa dan menganalisis kode program ada yang salah atau tidak. Jika modul ini dan telah diproduksi dalam output yang tidak memenuhi persyaratan, kode akan dikompilasi ulang dan diperiksa lagi sampai mencapai apa yang diharapkan singkatnya *White Box Testing* ini menguji dengan cara melihat Pure Code dari suatu aplikasi atau software yang diuji tanpa

memperdulikan tampilan antar muka atau UI (user interface) dari aplikasi tersebut. (socs.binus.ac.id, 2020).

White box merupakan metode desain uji kasus yang menggunakan struktur kontrol dari desain prosedural untuk menghasilkan kasus-kasus uji. Dengan menggunakan metode uji coba *white-box*, para pengembang perangkat lunak dapat menghasilkan kasus-kasus uji seperti :

1. Melakukan seluruh keputusan logikal baik dari sisi benar maupun salah,
2. Melakukan seluruh perulangan sesuai batasannya dan dalam batasan operasionalnya.
3. Menguji struktur data internal untuk memastikan validitasnya.

Hal ini dikarenakan sifat kerusakan alami dari perangkat lunak itu sendiri, seperti kesalahan *typographical* (cetakan) bersifat *random*. Ketika program diterjemahkan ke dalam kode sumber bahasa pemrograman, maka akan terjadi kesalahan pengetikan. Banyak yang terdeteksi dengan mekanisme pemeriksaan sintaks, tetapi banyak juga yang tidak terdeteksi sampai dengan dimulainya uji coba. Karena alasan tersebut diatas, maka ujicoba *white box testing* diperlukan selain *black box testing*. (Muhammad Nuris, 2015)