

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 *Digital Printing***

*Digital printing* adalah sebuah metode percetakan berbasis digital dari *file* gambar yang kemudian dapat langsung dicetak di berbagai media dengan cara cepat dan instant. *Digital printing* merupakan hasil inovasi perkembangan dari metode percetakan yang konvensional yang digunakan untuk berbagai bentuk percetakan seperti baliho, banner, spanduk, stiker, dll.

#### **2.2 *Mobile Delivery Order***

(Khusnul. 2015) Dalam ilmu bisnis, *Delivery Order* atau layanan pesan antar merupakan salah aktivitas dan pemberian jasa dimana pelanggan memesan produk yang disediakan produsen dan biasanya menggunakan media komunikasi melalui telepon atau internet lalu produk yang dipesan akan diantarkan sampai ke tempat pelanggan berada tanpa perlu datang dan bertemu langsung dengan penjual atau produsen. Aktivitas ini dilakukan sebagai penunjang citra bisnis dengan memberikan pelayanan terbaik bagi perusahaan. Sehingga, *Mobile Delivery Order* dapat diartikan sebagai kegiatan layanan pesan antar suatu produk dengan memanfaatkan fasilitas teknologi *mobile*.

#### **2.3 Rancang Bangun**

(YM.Khusuma Ardhama 2012) Rancang merupakan serangkaian prosedur untuk menerjemahkan hasil analisa dari sebuah sistem ke dalam bahasa pemrograman untuk mendeskripsikan dengan detail bagaimana komponen-komponen sistem diimplementasikan. Bangun atau pembangunan sistem adalah kegiatan menciptakan sistem baru maupun mengganti atau memperbaiki sistem yang telah ada baik secara keseluruhan maupun sebagian. Dengan demikian rancang bangun merupakan kegiatan menerjemahkan hasil analisa ke dalam bentuk paket

perangkat lunak kemudian menciptakan sistem tersebut ataupun memperbaiki sistem yang sudah ada.

#### **2.4 Konsep Website**

(YM.Khusuma Ardhamas 2012) *World Wide Web* (WWW) yang lebih dikenal dengan *website*, merupakan salah satu layanan yang dapat digunakan oleh pemakai komputer yang terhubung pada internet. *Website* pada awalnya adalah ruang informasi dalam internet, dengan menggunakan teknologi *hypertext* pemakai dituntut untuk menemukan informasi dengan mengikuti *link* yang disediakan dalam dokumen *website* yang ditampilkan pada *web browser*. Internet identik dengan *website*, karena popularitasnya sebagai penyedia informasi dan tampilan antar muka (*interface*) yang dibutuhkan oleh pengguna internet, dari masalah informasi sampai komunikasi. *Website* memudahkan pengguna komputer untuk berinteraksi dengan pelaku internet lainnya dan menelusuri informasi. *Website* juga telah banyak digunakan oleh perusahaan sebagai bagian dari strategi teknologi informasinya, hal ini tidak lepas dari kelebihanannya yaitu memiliki akses informasi yang mudah, *set up server* lebih mudah, informasi lebih mudah didistribusikan dan bebas *platform*.

#### **2.5 Aplikasi**

(Putra. 2012) Istilah aplikasi pada dasarnya berasal dari bahasa Inggris yaitu dari kata *application* yang berarti penerapan ataupun penggunaan. Namun jika ditinjau secara istilah aplikasi tersebut berarti sebagai suatu program yang telah siap untuk dipakai yang secara sengaja dibuat untuk melakukan suatu fungsi bagi pemakai jasa aplikasi jenis yang lainnya yang akan dipakai untuk sebuah sasaran yang dituju. Dapat disimpulkan bahwa pengertian aplikasi adalah suatu program siap pakai yang digunakan dalam komputer dan dibuat untuk melayani suatu kebutuhan manusia, sehingga komputer dapat memproses *input* dan menghasilkan *output*.

## 2.6 HTML 5

(Raharjo et al. 2010) *HyperText Markup Language* (HTML) adalah sebuah bahasa markah untuk menstrukturkan dan menampilkan isi *World Wide Web*, sebuah teknologi inti dari Internet. HTML5 adalah revisi kelima dari HTML, yang pertama kali diciptakan pada tahun 1990 dan versi keempatnya, HTML4, pada tahun 1997 dan hingga bulan Juni 2011 masih dalam pengembangan. Tujuan utama pengembangan HTML5 adalah untuk memperbaiki teknologi HTML agar mendukung teknologi multimedia terbaru, mudah dibaca oleh manusia dan juga mudah dimengerti oleh mesin.

## 2.7 PHP

(YM.Khusuma Ardhama 2012) PHP merupakan kependekan dari kata *Hypertext Preprocessor*. PHP tergolong sebagai perangkat lunak open source yang diatur dalam aturan *General Purpose Licences* (GPL), dan dapat diunduh bebas dari situs resminya yaitu (<http://www.php.net>). Pemrograman php sangat cocok dikembangkan dalam lingkungan web, karena PHP dilekatkan pada *script* HTML atau sebaliknya. PHP dikhususkan untuk pengembangan web dinamis. PHP tergolong juga sebagai bahasa pemrograman yang berbasis *server* (*Server Side Scripting*). Ini berarti bahwa semua *script* PHP diletakkan di *server* dan diterjemahkan oleh *web server* terlebih dahulu, kemudian hasil terjemahan itu dikirim ke *browser client* (Dodit, 2008).

## 2.8 CSS 3

(YM.Khusuma Ardhama 2012) *Cascading Style Sheets* (CSS) merupakan *feature* yang sangat penting dalam membuat Dynamic HTML. Meskipun bukan merupakan suatu keharusan dalam membuat web, akan tetapi penggunaan *cascading style sheets* merupakan kelebihan tersendiri. Suatu *cascading style sheet* merupakan tempat dimana mengontrol dan mengatur style-style yang ada. *Cascading Style sheet* mendeskripsikan bagaimana tampilan dokumen HTML di layar. Dalam pemakaian umumnya sering disebut juga sebagai template dari dokumen HTML yang menggunakannya. *Cascading Style Sheet* (CSS) teknologi

yang support pada hampir semua web Browser, hal ini disebabkan CSS telah di standartkan oleh *World Wide Web Consortium* (W3C) untuk digunakan web browser. (Nurhasyim,2003).

## 2.9 MYSQL

(YM.Khusuma Ardhama 2012) MYSQL merupakan turunan salah satu konsep utama dalam *databases*, yaitu SQL (Structural Query Language), SQL adalah sebuah konsep pengoperasian *database* terutama untuk pemilihan atau seleksi pemasukkan data seleksi dari pemasukkan data yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu sistem *database* (DBMS) dapat diketahui dari cara kerja optimizernya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh user maupun program-program aplikasinya. Sebagai *database server*, MySQL dapat dikatakan lebih unggul dibandingkan *database server* lainnya dalam *query* data. Hal ini terbukti untuk *query* yang dilakukan oleh *single user*, kecepatan *query* MySQL bisa sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan *Interbase*.

## 2.10 Unified Modeling Language (UML)

(S & Shalahuddin 2016) *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. Tujuan Penggunaan UML yaitu untuk memodelkan suatu sistem yang menggunakan konsep berorientasi objek dan menciptakan bahasa pemodelan yang dapat digunakan baik oleh manusia maupun mesin.


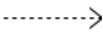


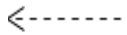


Tipe-tipe dari diagram UML adalah sebagai berikut :

### 1. Use Case Diagram



*Use case diagram* adalah gambar dari beberapa atau seluruh aktor dan *use case* dengan tujuan yang mengenali interaksi mereka dalam suatu sistem. *Use case*

*diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem, yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara actor dan sistem. Dalam *use case* diagram terdapat istilah seperti aktor, *use case* dan *case relationship*. Penjelasan simbol pada Tabel 2.1

Tabel 2.1 Simbol *Use Case Diagram*.

GAMBAR	NAMA	KETERANGAN
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri ( <i>independent</i> ).
	<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>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor




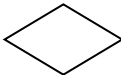
Tabel 2.1 (lanjutan).

	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi


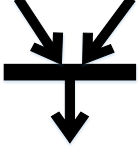
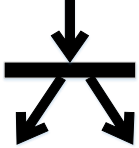
## 2. Activity Diagram

*Activity diagram* menggambarkan rangkaian aliran dari aktifitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi. *Activity Diagram* berupa *flowchart* yang digunakan untuk memperlihatkan aliran kerja dari sistem. Notasi yang digunakan dalam *activity diagram* dapat dilihat pada Tabel 2.2.

Tabel 2.2 Simbol Activity Diagram.

Simbol	Keterangan
	<i>Activity</i> : Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	<i>Initial Node</i> : Bagaimana objek dibentuk atau diawali
	<i>Activity Final Node</i> : Bagaimana objek dibentuk dan diakhiri.
	<i>Decision</i> : Asosiasi percabangan dimana jika ada pilihan aktifitas lebih dari satu.

Tabel 2.2 (lanjutan).

	<p><i>Swimlane</i> : Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktifitas yang terjadi.</p>
	<p><i>Join</i> : Digunakan untuk menunjukkan kegiatan yang digabungkan.</p>
	<p><i>Fork</i> : Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel</p>

### 3. *Sequence Diagram*

*Sequence diagram* menggambarkan kolaborasi dinamis antara sejumlah dan untuk menunjukkan rangkaian pesan yang dikirim antar objek juga interaksi antar objek, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem. *Sequence diagram* menjelaskan interaksi objek yang disusun berdasarkan urutan waktu.

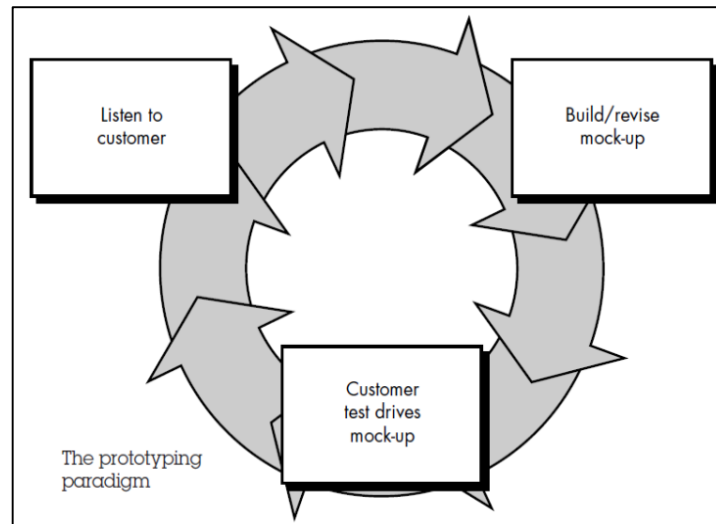
### 4. *Class Diagram*

*Class diagram* menggambarkan dtruktur data dan deskripsi *class*, *package*, dan objek beserta hubungan satu sama lain. *Class diagram* berfungsi untuk menjelaskan tipe dari objek sistem dan hubungannya dengan objek yang lain. *Class* memiliki 3 area pokok yaitu nama, atribut dan metode.

## 2.11 Metode Perancangan Sistem

(S & Shalahuddin 2016) Metode pengembangan perangkat lunak yang digunakan dalam penelitian ini adalah model *prototype*. Model *prototype* mampu menawarkan pendekatan yang baik dalam hal kepastian terhadap efisiensi algoritma dan kemampuan penyesuaian diri dari sebuah sistem operasi. Proses

pengembangan *prototype* diuraikan dalam tiga tahapan yaitu, pengumpulan kebutuhan, perancangan, dan evaluasi *prototype* seperti terlihat pada Gambar 2.1.



Gambar 2.1 Model *Prototype*.

## 2.12 Penelitian Terdahulu

Penelitian terdahulu yang berkaitan dengan penelitian perancangan aplikasi *mobile delivery order* di Simetri Digital Printing dapat dilihat pada Tabel 2.3.

Tabel 2.3 Penelitian Terdahulu.

Nama	Judul	Terbit/Tahun	Keterangan
Ardhi, D, C.,	Sistem Delivery Order Makanan Berbasis Lokasi Dengan Platform Android	Universitas Gajah Mada, 2012: Yogyakarta	Dengan adanya sistem yang dapat memberikan rekomendasi sebagai bahan untuk mengimplemtasikan sistem delivery order dengan lokasi yang berbasis android.



Tabel 2.3 (lanjutan)

Nandapuspita	Sistem Informasi Pesan Antar di Chacha Taiwan Milk Tea Outlet Kolombo Yogyakarta	Universitas Gadjah Mada, 2013: Yogyakarta	Pembuatan sistem informasi pesan antar untuk memberikan informasi dan konfirmasi pesan.
Putra	Aplikasi mobile pemesanan burger berbasis platform android,	Universitas Gadjah Mada, 2012: Yogyakarta	Dengan penelitian sistem delivery order membantu dan memudahkan dalam pemesanan burger.
Akbar muhamad	Pembuatan aplikasi pesan antar makanan pada sistem operasi android	Universitas diponegoro semarang, 2016:	Sistem pesan antar makanan pada sistem android memberikan kemudahan dalam pemesanan dan antar makanan berdasarkan nomer pesanan dan nomer meja.
Farida Nurul	Sistem delivery order berbasis android pada AADS Yogyakarta.	Sekolah Tinggi Manajemen Informatika dan computer AMIKOM YOGYAKARTA, 2014.	Dengan menerapkan sistem delivery order membuat untuk memudahkan pemesanan atau order dengan sistem berbasis android

