

## BAB 2

### LANDASAN TEORI

Pada bab 2 akan dijelaskan tentang dasar teori yang digunakan dalam menyusun tugas akhir ini serta hal-hal yang menyangkut perancangan sistem aplikasi. Teori tentang aplikasi pemesanan online, konveksi, android studio, UML, Java, Kotlin, Basis Data, MySQL, Web, *framework*, *Codeigniter* akan diuraikan satu persatu di bawah ini.

#### 2.1 Aplikasi Pemesanan Online

Secara umum aplikasi merupakan program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju.[3]

Rahman (2015) menyebut aplikasi merupakan software yang ditransformasikan ke komputer yang berisikan perintah-perintah yang berfungsi untuk melakukan berbagai bentuk pekerjaan atau tugas-tugas tertentu seperti penerapan, penggunaan, dan penambahan data [4]

Lebih lanjut Utara (2011) dalam Rahman (2015) menyebutkan bahwa pemesanan adalah suatu aktivitas yang dilakukan oleh konsumen sebelum membeli. Untuk mewujudkan kepuasan konsumen maka perusahaan harus mempunyai sebuah sistem pemesanan yang baik. Adapun tujuan dari pemesanan yaitu: 1)memaksimalkan pelayanan bagi konsumen, 2)meminimumkan investasi pada persediaan, 3)perencanaan kapasitas, 4)persediaan dan kapasitas. [4]

Maka dapat disimpulkan bahwa yang dimaksud dengan aplikasi pemesanan online adalah software yang berisi tools atau alat untuk melakukan pemesanan dari pihak konsumen yang dilakukan secara online. melalui aplikasi ini pihak konsumen tidak perlu bertemu langsung dengan pihak produsen karena transaksi pemesanan cukup dilakukan melalui aplikasi secara online.

#### 2.2 Konveksi

Konveksi merupakan salah satu bisnis di bidang pakaian yang biasanya

memproduksi pakaian dalam partai besar sesuai permintaan. Bisnis konveksi banyak diminati di Indonesia karena permintaan pasar yang tinggi namun kompetitornya masih rendah.

Pengertian lain dari konveksi disebutkan Segarwati (2018) yaitu sebuah tempat yang dibuat khusus untuk memproduksi merchandise yang berhubungan dengan sandang manusia. Sederhananya konveksi adalah tempat untuk memproduksi pakaian atau jenis fashion lainnya dalam jumlah tertentu. Beberapa jenis fashion yang dibuat di bisnis konveksi tersebut biasanya baju, kaos, jaket, celana, seragam, topi, dan jenis fashion lainnya. pada umumnya bisnis konveksi memproduksi produk fashion dalam jumlah yang cukup banyak misal per lusin atau per kodi [5].

### **2.3 Android Studio**

Hansun, dkk (2018) mengatakan android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, middleware dan aplikasi. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel/smartphone. [6]

Lebih lanjut dikatakan bahwa fitur-fitur android yaitu media penyimpanan, koneksi jaringan, SMS, Web browser, media support, hardware support, multitouch, multitasking, dukungan flash, tethering, dan voice base picture, tangkapan layar, handset layouts. Selain itu, perangkat android di antaranya yaitu smartphone, tablet, ebook reader, notebook, personal computer, smart TV, camera, kulkas, dan jam tangan. [6]

### **2.4 UML**

Menurut Gata & Gata (2013) *Unified Modeling Language* (UML) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk

mendukung pengembangan sistem.[7]

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

a. *Use Case Diagram*

Use case diagram merupakan pemodelan untuk kelakuan (behavior) 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 yaitu:

| Gambar                                                                              | Keterangan                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|   | <p><i>Use Case</i><br/>menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, yang dinyatakan dengan menggunakan kata kerja.</p>                                                                                                                                                                                                                                    |
|  | <p>Aktor adalah abstraction dari orang atau 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. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan use case, tetapi tidak memiliki kontrol terhadap <i>use case</i>.</p> |

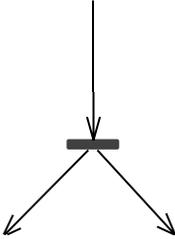
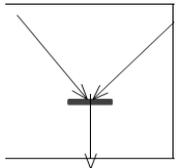
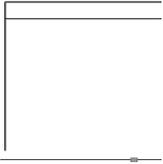
|                                                                                   |                                                                                                                                                                                                      |
|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <p>Asosiasi antara aktor dan <i>use case</i>, digambarkan dengan garis Tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data.</p> |
|  | <p>Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.</p>                                            |
| <p>&lt;&lt;Include&gt;&gt;<br/>-----</p>                                          | <p><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.</p>     |
| <p>&lt;&lt;Extend&gt;&gt;<br/>□-----</p>                                          | <p><i>Extend</i>, merupakan perluasan dari <i>use case</i> lain jika kondisi atau Syarat.</p>                                                                                                        |

Gambar 2. 1 Use Case Diagram

b. Diagram Aktivitas (*Activity Diagram*)

Menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity Diagram* yaitu:

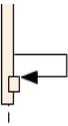
| Gambar                                                                              | Keterangan                                                                               |
|-------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|
|  | <p><i>Start Point</i>, diletakkan pada pojok kiri atas dan merupakan awal aktivitas.</p> |

|                                                                                     |                                                                                                                                                                                                                                       |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | <p><i>End Point</i>, akhir aktivitas.</p>                                                                                                                                                                                             |
|    | <p><i>Activities</i>, Menggambarkan suatu proses / kegiatan bisnis.</p>                                                                                                                                                               |
|    | <p><i>Fork / Percabangan</i>, digunakan untuk menunjukan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan menjadi satu.</p>                                                                               |
|   | <p><i>Join</i> (penggabungan) atau <i>rake</i>, digunakan untuk menunjukan adanya dekomposisi kelas yang menjadi interfaces atau interaksi antara satu atau lebih aktor dengan sistem seperti tampilan form entry dan form cetak.</p> |
|  | <p><i>DecisionPoints</i>, menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> atau <i>false</i>.</p>                                                                                                                       |
|  | <p><i>Swimlane</i>, pembagian activity diagram untuk menunjukan siapa melakukan apa.</p>                                                                                                                                              |

Gambar 2. 2 Activity Diagram

c. Diagram Urutan (*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* yaitu

| Gambar                                                                              | Keterangan                                                                                                                                                                       |
|-------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | Entity Class, merupakan bagian dari Sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data. |
|  | Boundary Class, berisi kumpulan.                                                                                                                                                 |
|  | Kelas yang menjadi interfaces atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan form entry dan form cetak.                                             |
|  | Message, simbol mengirim pesan antar class.                                                                                                                                      |
|  | Recursive, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.                                                                                                    |

|                                                                                   |                                                                                                                                           |
|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
|  | <p>Activation, mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi.</p> |
|  | <p>Lifeline, garis titik-titik yang terhubung dengan objek, sepanjang lifeline terdapat activation.</p>                                   |

Gambar 2. 3 Sequence Diagram

## 2.5 Java

Koesheryatin (2014) menyebutkan java script adalah Bahasa script berdasar pada objek yang memperbolehkan pemakai untuk mengendalikan banyak aspek interaksi pemakai pada suatu dokumen HTML. Dimana objek tersebut dapat berupa window, frame, url, document, form, button, atau item yang lain yang semua itu mempunyai properti yang saling berhubungan dengannya, dan masing-masing memiliki nama, lokasi, warna nilai, dan atribut lain. Kode program java script dapat dituliskan langsung pada file HTML menggunakan tag kontainer. [8]

Saat ini Java merupakan salah satu bahasa pemrograman paling populer untuk segala hal mulai dari permainan pemrograman untuk menciptakan aplikasi *mission-critical*. Tujuannya adalah untuk mempermudah dan menjadi platform independent alternatif menggunakan C++. Java adalah program yang sangat populer di lingkungan pendidikan saat ini, hampir di seluruh kampus memiliki satuan ajar pemrograman ini.

## 2.6 Kotlin

Kotlin adalah bahasa pemrograman berbasis Java Virtual Machine (JVM). Kotlin merupakan bahasa pemrograman yang pragmatis untuk android yang mengkombinasikan object oriented (OO) dan bahasa fungsional. Kotlin juga bahasa

pemrograman yang interoperabilitas yang membuat bahasa ini dapat digabungkan dalam satu project dengan bahasa pemrograman Java. Bahasa pemrograman ini juga dapat digunakan untuk pengembangan aplikasi berbasis desktop, web dan backend . Kotlin awalnya dikembangkan oleh JetBrains, perusahaan dibalik IntelliJ IDEA. Setelah melalui banyak perkembangan, JetBrains merilis kotlin secara *open source* dan kini perkembangannya semakin maju. Google mendukung penuh kotlin untuk pengembang aplikasi Android.[9]

## 2.7 Basis Data

Basis data merupakan tempat penampungan informasi organisasi di dalam sistem komputer yang dapat dengan mudah diakses, diolah, dan diperbarui. Data diorganisasi ke dalam baris, kolom, dan tabel, dimana terdapat index untuk mempermudah pencarian informasi tertentu, misalnya mencari informasi pada tabel pengguna dengan index satu, maka akan menampilkan seluruh informasi pengguna yang memiliki index satu, bisa berupa nama lengkap, alamat, dan umur.

Bourret (2005) dalam Hartati (2008) menyebut Basis data XML adalah sistem perangkat lunak yang digunakan untuk menyimpan data yang membolehkan data untuk diimpor, diakses dan diekspor dalam format XML. Basisdata XML mempunyai keunggulan lebih baik dibandingkan dengan sistem basis data relasional jika data yang akan disimpan berupa dokumen. Dengan basisdata XML juga memungkinkan untuk melakukan penelusuran isi dokumen.[10]

## 2.8 MySQL

Suhartanto (2017) menyebutkan MySQL dikembangkan oleh sebuah perusahaan Swedia bernama MySQL AB yang pada saat itu bernama TcX DataKonsult AB sekitar tahun 1994-1995, namun cikal bakal kodenya sudah ada sejak 1979. Awalnya TcX membuat MySQL dengan tujuan mengembangkan aplikasi web untuk klien. TcX merupakan perusahaan pengembang software dan konsultan database. Saat ini MySQL sudah diakuisisi oleh Oracle Crop. MySQL adalah salah satu jenis database server yang

sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang database sebagai sumber dan pengolahan datanya. Kepopuleran MySQL antara lain karena MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses database-nya sehingga mudah untuk digunakan.[11]

MySQL merupakan program basis data yang mampu mengirim dan menerima data dengan sangat cepat dan multi-user. MySQL mempunyai beberapa keunggulan yaitu :

- 1) MySQL dapat menangani database relasional dan dapat dipakai untuk client/server.
- 2) Software MySQL adalah open source, artinya dapat mengambil, memakai, dan mengubah source-nya dengan bebas tanpa biaya.
- 3) MySQL sangat cepat. Multithreaded yaitu setiap query diperlakukan sebagai thread based yang sangat cepat.
- 4) MySQL dapat diakses oleh client menggunakan protokol TCP/IP pada semua platform. Pada windows, client dapat mengakses menggunakan named-pipe. Sementara itu UNIX (Linux) dapat memakai domain socket-file.

## **2.9 Web**

Menurut Hariyanto (2015) dalam Destiningrum (2017) website dapat diartikan sebagai kumpulan halaman yang menampilkan informasi data teks, data gambar, data animasi, suara, video dan gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, dimana masing-masing dihubungkan dengan jaringan-jaringan halaman [12].

Sedangkan Rohi Abdulloh (2015) dalam Josi (2017) menyebut web adalah sekumpulan halaman yang terdiri dari beberapa halaman yang berisi informasi dalam bentuk data digital baik berupa text, gambar, video, audio, dan animasi lainnya yang disediakan melalui jalur koneksi internet [13].

Berdasarkan uraian diatas, penulis menyimpulkan bahwa web adalah sebuah software yang berfungsi untuk menampilkan dokumen-dokumen pada suatu web yang membuat pengguna dapat mengakses internet melalui software yang terkoneksi dengan internet.

## **2.10 Framework**

Menurut Betha Sidik (2012) dalam Suwarjono (2018) Framework adalah kumpulan instruksi-instruksi yang dikumpulkan dalam class dan function-function dengan fungsi masing masing untuk memudahkan developer dalam memanggilnya tanpa harus menuliskan syntax program yang sama berulang-ulang serta dapat menghemat waktu [14].

## **2.11 Codeigniter**

Dalam situs resmi codeigniter, (Official Website CodeIgniter, 2002) menyebutkan bahwa codeigniter merupakan framework PHP yang kuat dan sedikit bug. Codeigniter ini dibangun untuk para pengembang dengan bahasa pemrograman PHP yang membutuhkan alat untuk membuat web dengan fitur lengkap.

Menurut Betha Sidik (2012) dalam Suwarjono (2018) CodeIgniter adalah sebuah framework php yang bersifat open source dan menggunakan metode MVC (Model, View, Controller) untuk memudahkan developer atau programmer dalam membangun sebuah aplikasi berbasis web tanpa harus membuatnya dari awal [14].

## **2.12 Metode Make to Stock**

Dalam strategi *Make-to-Stock*, perusahaan industri memiliki resiko yang tinggi berkaitan dengan investasi inventory karena pesanan pelanggan secara aktual tidak dapat diidentifikasi secara tepat dalam proses produksi. Permintaan actual dari pelanggan hanya dapat diramalkan, di mana sering kali tingkat aktual dari produksi hanya berkolerasi rendah dengan pesanan pelanggan aktual yang diterima. Berkaitan

dengan hal ini, perusahaan industri yang memilih strategi Make-to-Stock harus membangun sistem informasi pasar yang andal agar secara lebih akurat dapat meramalkan permintaan aktual dari konsumen [15].

Make-to-stock diterapkan oleh perusahaan yang memproduksi barangnya dalam kelompok jumlah (batch) tertentu dan memiliki persediaan barang jadi untuk memenuhi permintaan penjualan di masa depan. Perusahaan mendapatkan keuntungan dari meningkatnya kecepatan perusahaan dalam memenuhi permintaan pelanggan, namun kecepatan ini harus ditukar dengan biaya yang dikeluarkan perusahaan untuk menyimpan persediaan. Karena perusahaan melakukan produksi sebelum permintaan diketahui dengan pasti, biasanya perusahaan perlu melakukan peramalan dan menyesuaikan jumlah produk yang harus diproduksi dengan nilai yang didapat dari peramalan. kunci utama dalam Make-to-Stock adalah jumlah pesanan yang dapat dipenuhi perusahaan dengan tingkat persediaan yang dimiliki perusahaan.

MTS memiliki karakteristik sebagai berikut [9]:

1. Standar item, high volume.
2. Terus-menerus dibuat lalu disimpan.
3. Pengiriman dapat dilakukan segera.
4. Customer tidak mau menunggu.
5. Perlu adanya safety stock untuk mengatasi fluktuasi [