



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

LANDASAN TEORI

2.1. Mobile Application

Mobile Application atau aplikasi seluler adalah sebuah program komputer yang didesain untuk dijalankan di perangkat seluler seperti telepon genggam, tablet atau bahkan jam (Viswanathan, 2017).

Mobile Application atau aplikasi seluler adalah sebutan untuk sebuah aplikasi yang dapat mengakses internet yang dapat dijalankan di *smartphone* atau perangkat seluler lainnya. *Mobile Application* seringkali membantu pengguna dengan cara mengkoneksikan mereka ke pelayanan internet yang biasanya hanya bisa diakses di desktop, *notebook* ataupun komputer (Beal, Mobile Application, 2018).

2.2. Android

Android adalah sistem operasi berbasis Linux Kernel yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer tablet. Android awalnya dikembangkan oleh Android, Inc., dengan dukungan finansial dari Google, yang kemudian membelinya pada tahun 2005, namun rilisnya sendiri baru pada tahun 2007 (Holla & Katti, 2012).

Android merupakan sistem operasi dengan sumber terbuka, dan Google merilis kodenya dibawah lisensi Apache. Kode dengan sumber terbuka dan lisensi

perizinan pada Android memungkinkan perangkat lunak untuk dimodifikasi secara bebas dan didistribusikan oleh para pembuat perangkat, operator nirkabel, dan pengembang aplikasi (Gilski & Stefanski, 2015).

2.2.1. Linux Kernel

Linux Kernel adalah dasar fondasi utama dari Android yang berfungsi sebagai penghubung dari *hardware* serta *layer* dari Android Software stack. (Gilski & Stefanski, 2015)

2.2.2. Android Runtime

Setiap perangkat yang menjalankan Android 5.0 dan yang lebih tinggi, menjalankan proses aplikasinya dengan *Android Runtime* (ART) yang merupakan pengganti dari *Dalvik Virtual Machine* yang memiliki performa yang lebih baik. (Arsitektur Platform, 2018)

2.2.3. Android Libraries

Android Libraries merupakan kumpulan *Java-based Libraries* yang berguna untuk mengembangkan *android*. Contohnya adalah *android libraries* yang berfungsi memfasilitasi membuat *user interface*, membuat gambar dan akses ke *database* (Arsitektur Platform, 2018)

2.2.4. Application Framework

Kumpulan *service* di dalam aplikasi android yang digunakan untuk mengatur dan menjalankan aplikasi (Arsitektur Platform, 2018).

2.3. Mobile Application Development

Mobile Application Development adalah proses membuat aplikasi perangkat seluler seperti telepon genggam. Namun pada zaman sekarang, pengembang aplikasi tidak hanya memang berfokus untuk membuat aplikasi seluler, beberapa juga bahkan membuat aplikasi seluler berdasarkan situs mereka, dalam artian aplikasi seluler yang mereka buat sebenarnya hanyalah situs (Sonmez, 2016).

Mobile Application Development adalah istilah yang digunakan untuk menunjukkan tindakan atau proses di mana aplikasi seluler dikembangkan untuk perangkat seluler, seperti asisten digital pribadi, asisten digital perusahaan atau ponsel (Panwar, 2018).

UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

2.4. Android Software Development

Android Software Development adalah proses membuat aplikasi perangkat seluler berbasis operasi sistem Android. Aplikasi dapat ditulis menggunakan Java, C++ ataupun Kotlin dengan menggunakan *Android software development kit (SDK)* (Sonmez, 2016).

2.5. Android SDK

Android Software Development Kit adalah seperangkat alat pengembangan yang digunakan untuk mengembangkan *platform* Android. Android SDK berisikan: *libraries, Debugger, Emulator*, dokumentasi untuk APIs (*Application Program Interfaces*) Android, Contoh *source code*, dan *tutorial* untuk OS Android (Beal, Android SDK, 2018).

Setiap kali Google merilis versi Android terbarunya, SDK terbaru juga ikut dirilis bersamaan. Untuk bisa menulis program dengan fitur terbaru, *developer* harus mengunduh dan menginstall setiap versi SDK untuk jenis telepon seluler tertentu (Beal, Android SDK, 2018).

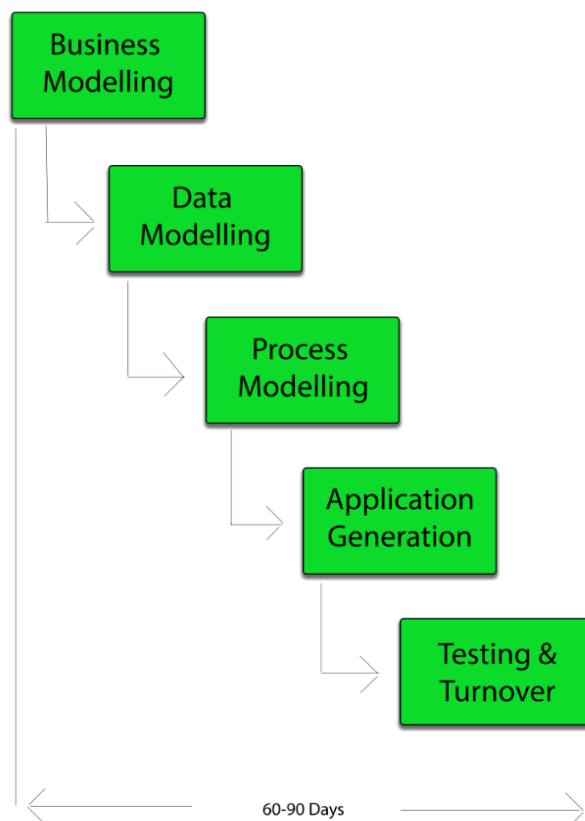
2.6. Android Studio

Android Studio adalah sebuah *software development tool* yang bisa digunakan untuk membangun aplikasi berbasis Android. Bahasa pemrograman yang digunakan adalah Java (Fikri, Herumurti, & H., 2016).

Android Studio adalah IDE untuk Android yang diumumkan pada Mei 2013 di acara pengembang Google I / O, dan dimaksudkan sebagai alternatif untuk Eclipse (Singh, Sharma, & Singh, 2016).

2.7. RAD Model

Metode penelitian yang digunakan pada perancangan aplikasi ini menggunakan metode RAD-Model. RAD-Model adalah *Rapid Application Development Model*, yang merupakan tipe *incremental model*. Berikut ini adalah gambaran dari RAD-Model. (Kosasi, 2015)



Gambar 2.1 Metode RAD

Sumber : (Kosasi, 2015)

Berikut adalah penjelasan dari setiap fase yang ada di dalam *rapid application development model* (Kosasi, 2015):

a. *Business Modeling*

Penentuan alur maupun fungsi bisnis yang cara identifikasinya adalah menganalisa dengan jelas tentang bisnis yang dituju. Dalam kasus ini, tujuan melakukan *business modeling* adalah menganalisa alur aplikasi sesuai dengan kondisi yang ada pada lapangan serta kebutuhan yang ada di dalam lapangan.

b. *Data Modeling*

Merupakan sekumpulan informasi yang didapat dari proses *business modeling* yang kemudian diolah menjadi objek data yang dibutuhkan oleh bisnis. Dalam kasus ini, tujuan melakukan *data modeling* adalah memanfaatkan analisa sebelumnya untuk kemudian dijadikan acuan bagi pengembang aplikasi untuk menentukan fungsi-fungsi dalam aplikasi.

c. *Process Modeling*

Berisikan sekumpulan objek data yang dikonversikan menjadi alur informasi bisnis untuk memperoleh objektif bisnis yang spesifik. Dalam kasus ini, fungsi dari *process modeling* adalah penyaringan fungsi-fungsi serta alur yang tepat yang sudah direncanakan pada saat melakukan *data modeling*. Fungsi-fungsi yang sesuai dengan tujuan pembuatan aplikasi akan disimpan bersamaan dengan alur.

d. *Application Generation*

Proses konversi hasil dari *process models* menjadi sekumpulan kode dan akhirnya menjadi sistem *actual*. Dalam kasus ini, fungsi dari *application generation* adalah untuk menggabungkan alur dan fungsi-fungsi yang kemudian diimplementasikan menjadi sebuah aplikasi.

e. *Testing and Turnover*

Melakukan percobaan pada aplikasi yang telah dibuat pada proses *application generation*. Dalam kasus ini, *testing and turnover* digunakan untuk menguji aplikasi yang telah dibuat serta menambah dan merevisi fitur jika ada yang kurang ataupun salah.

2.8. Spiral Model

Model spiral mirip dengan *iterative* proses desain karena ada pengulangan berulang (disebut siklus) di mana upaya berturut-turut dibuat untuk mengembangkan solusi, namun, model spiral berbeda dari model berulang di sejumlah bidang utama. Pertama, sementara proses desain penelitian berulang biasanya melibatkan iterasi pada konstruksi prototipe, model spiral lebih fokus pada pengurangan risiko. Kedua, karena model spiral didasarkan pada pengembangan perangkat lunak evolusi, model spiral mengharapkan dan merencanakan untuk sejumlah iterasi tertentu sedangkan salah satu keterbatasan model berulang adalah bahwa tidak mungkin untuk mengetahui kapan Anda telah mencapai solusi terbaik. Dengan itu, model spiral juga mampu mendukung iterasi lebih lanjut setelah aslinya iterasi yang direncanakan, mirip dengan model iteratif. Akhirnya, elemen kunci dari model spiral adalah melibatkan perencanaan dan pelaksanaan yang berbeda tugas

selama setiap iterasi, atas dasar yang diperlukan saat proyek berkembang (Novakouski, 2015).

2.9. Prototype Model

Prototype adalah instrumen yang digunakan dalam proses pengembangan perangkat lunak dalam bentuk model sederhana dari perangkat lunak yang memberikan pengguna gambaran dasar tentang program dan pengujian awalnya, serta mendeskripsikan siklus hidup *prototyping* sebagai identifikasi kebutuhan dasar, mengembangkan model kerja, mendemonstrasikan model kepada semua orang yang terkait dan untuk meminta persyaratan, dan mengimplementasikan revisi dengan iterasi hingga semua orang yang terkait puas dengan fungsi tersebut. Pengembangan perangkat lunak yang menggabungkan *prototype* telah terbukti mampu secara dinamis menanggapi perubahan kebutuhan yang diinginkan oleh pengguna, mengurangi jumlah pengerjaan yang dilakukan secara berulang, dan membantu mengendalikan risiko persyaratan yang tidak lengkap (Rajendra Ganpatrao Sabale, 2012).

2.10. Waterfall Model

Waterfall adalah model klasik dari *software engineering*. Model ini adalah salah satu model tertua yang digunakan dalam proyek-proyek pemerintah dan di banyak perusahaan besar. Model ini juga disebut sebagai model sekuensial linier atau siklus hidup klasik. Ini adalah model SDLC linear berurutan yang diikuti oleh fase Analisis Kebutuhan, Desain, Implementasi, Pengujian dan Pemeliharaan. Fase-fase itu diperbaiki sedemikian rupa sehingga fase itu tidak bisa diulangi lagi. Karena

model *waterfall* adalah model klasik sehingga model ini berfungsi sebagai dasar dari semua model lainnya (Rather & Bhatnagar, 2015)

2.11. Database

Database adalah koleksi pengumpulan data, karena data dalam *database* telah terorganisir, maka hal itu membuat manajemen data menjadi lebih mudah, hal itulah yang memacu diperlukannya *database* (Point, 2015).

2.11.1. phpMyAdmin

phpMyAdmin adalah sebuah perangkat lunak dengan basis bahasa pemrograman PHP yang digunakan pengatur dari MySQL dengan menggunakan media *website* yang berfungsi untuk mengelola data yaitu tabel, baris, kolom, dan *fields*, membuat relasi antar tabel, dan mengkoneksikan satu sama lain agar dapat terhubung

phpMyAdmin adalah alat perangkat lunak gratis yang ditulis dalam PHP yang dimaksudkan untuk menangani administrasi MySQL atau *server database* MariaDB. Anda dapat menggunakan phpMyAdmin untuk melakukan sebagian besar tugas administrasi, termasuk membuat sebuah basis data, menjalankan kueri, dan menambahkan akun pengguna (team, 2019).

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

2.12. Unified Modeling Language

UML telah muncul sebagai bahasa cetak biru perangkat lunak untuk analis, perancang, dan pemrogram sama. Sekarang bagian dari perdagangan perangkat lunak. UML memberi semua orang mulai dari analis bisnis hingga perancang hingga pemrogram, kosa kata umum untuk dibicarakan desain perangkat lunak (Miller, 2015).

2.12.1. Use Case Diagram

Use Case Diagram adalah ringkasan skenario untuk satu tugas atau tujuan. Seorang aktor adalah siapa atau apa yang mengawali peristiwa yang terlibat dalam tugas itu. Aktor hanyalah peran yang dimainkan orang atau benda. Gambar di bawah ini adalah kasus penggunaan Make Appointment untuk klinik medis. Aktor adalah Pasien. Koneksi antara aktor dan use case adalah asosiasi komunikasi (atau komunikasi singkatnya) (Miller, 2015).

2.12.2. Class Diagram

Class Diagram adalah diagram Kelas memberikan gambaran umum sistem dengan menunjukkan kelas-kelasnya dan hubungannya diantara mereka. Class diagram adalah statis - mereka menampilkan apa yang berinteraksi tetapi tidak apa yang terjadi ketika mereka berinteraksi (Miller, 2015).

2.12.3. Activity Diagram

Activity Diagram adalah pada dasarnya adalah diagram alur yang mewah. Diagram aktivitas dan diagram statechart yang terkait. Sementara diagram statechart memusatkan perhatian pada objek yang mengalami proses (atau pada proses sebagai objek), diagram aktivitas berfokus pada aliran kegiatan yang terlibat dalam satu proses. Diagram aktivitas menunjukkan bagaimana kegiatan-kegiatan tersebut saling bergantung (Miller, 2015).

2.12.4. Actor Table

Pembuatan *Actor Table* bertujuan untuk membantu manajer proyek untuk mengidentifikasi aktor dan pemangku kepentingan utama dari proyek mereka. Oleh sistematis menjawab pertanyaan yang disajikan dalam tabel, manajer proyek dapat menjadi sadar akan para aktor dan pemangku kepentingan terkait dengan proyek mereka, dan juga menjadi waspada terhadap keprihatinan, sumber daya, jejaring sosial dan sumber pengaruh potensial mereka pada proyek (Miller, 2015).

U M N
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

2.12.5. Behavior Pattern

Pembuatan *Behavior Pattern* menjelaskan algoritma, penugasan tanggung jawab, dan interaksi antara objek (hubungan perilaku). Behavior Pattern bertujuan untuk melihat pola kelas perilaku menggunakan pewarisan serta untuk mendistribusikan perilaku pola objek perilaku menggunakan komposisi. Contoh untuk membuat *Model-view-controller* dalam aplikasi UI, Mengekasi koleksi benda dan Antarmuka yang sebanding di Jawa.

2.12.6. Behavior Pattern

Pembuatan *Behavior Pattern* menjelaskan algoritma, penugasan tanggung jawab, dan interaksi antara objek (hubungan perilaku). Behavior Pattern bertujuan untuk melihat pola kelas perilaku menggunakan pewarisan serta untuk mendistribusikan perilaku pola objek perilaku menggunakan komposisi. Contoh untuk membuat *Model-view-controller* dalam aplikasi UI, Mengekasi koleksi benda dan Antarmuka yang sebanding di Jawa.



2.13. Penelitian Terdahulu

Tabel 1.1 Daftar Penelitian

1	Nama	Michael Yosep Ricky
	Jurnal, Volume, halaman, tahun	EPJ Web of Conferences, 68, 00041, 2014
	Judul	<i>Mobile Food Ordering Application using Android OS Platform</i>
	Metode	<i>Water model of System Development Life Cycle (SDLC).</i> Metode dengan fase berikut ini: definisi kebutuhan, menganalisis dan menentukan fitur yang diperlukan dalam mengembangkan aplikasi dan membuat definisi detail dari setiap fitur, desain sistem dan perangkat lunak, merancang aliran pengembangan aplikasi dengan menggunakan desain storyboard, desain pengalaman pengguna, Unified Modeling Language (UML) desain, dan desain struktur database, implementasi pengujian unit, membuat database dan menerjemahkan hasil desain ke kode bahasa pemrograman kemudian melakukan pengujian unit, integrasi dan pengujian sistem, mengintegrasikan unit program ke satu unit sistem kemudian melakukan pengujian sistem, operasi dan pemeliharaan, pengoperasian hasil pengujian sistem dan jika ada perubahan dan perbaikan yang diperlukan maka fase sebelumnya bisa kembali.
	Objek Penelitian	Restoran
	Hasil	Membuat sebuah aplikasi berbasis Android yang membantu kedua belah pihak yaitu kurir dan pelanggan, dan website restoran.
	Kesimpulan	Inti dari penelitian ini adalah mempermudah setiap orang untuk memesan dengan mudah menggunakan sistem POS di Android
2	Nama	Fan Yang
	Jurnal, Volume, halaman, tahun	VAASAN AMMATTIKORKEAKOULU, 82, 2014

Judul	<i>Mobile Food Ordering Application</i>
Metode	<ol style="list-style-type: none"> 1. Menambah daftar makanan dan minuman yang dijual restoran pada <i>background management platform</i>. 2. Menampilkan daftar makanan dan minuman yang dijual restoran pada situs restoran. 3. Memesan makanan dan minuman yang dijual restoran melalui situs restoran 4. Menampilkan daftar makanan dan minuman yang dijual restoran pada aplikasi Android 5. Memesan makanan dan minuman yang dijual restoran melalui aplikasi Android
Objek Penelitian	Restoran
Hasil	Sistem ini dapat diperluas lagi menjadi sitem yang dapat menerima pesanan. Sistem ini sebelumnya hanya menampilkan daftar makanan serta minuman apa saja yang dijual di restoran tersebut, tidak hanya itu saja, restoran itu sendiri dapat menambah atau mengubah atau bahkan mengurangi makanan yang mereka jual melalui <i>Background Management Platform</i> . Pelanggan juga dapat melihat daftar makanan dan minuman yang ada
Kesimpulan	Pemilik dan karyawan dapat mengatur pemesanan sesuai keinginan. Pada halaman utama, pelanggan dapat melihat informasi daftar makanan dan minuman yang dijual restoran dengan jelas dan dapat pula membeli makanan atau minuman yang dijual.

Penelitian yang dilakukan oleh (Yosep Ricky, 2014) memberikan informasi mengenai bagaimana POS sistem dibuat serta manfaat manfaatnya dalam restoran seperti mempermudah transaksi dan data yang transparan.

Penelitian yang dilakukan oleh (Yang, 2014) memberikan informasi berupa cara untuk membuat aplikasi POS secara mendetil mulai dari perancangan hingga pembangunan.

Untuk menyelesaikan permasalahan yang dimiliki oleh *owner* dari PawPaw Café, maka aplikasi akan dibuat berdasarkan rancangan yang telah diadopsi dari penelitian (Yang, 2014) serta mengimplementasikan fungsi dari (Yosep Ricky, 2014) dan kemudian diberikan sentuhan berbeda sesuai dengan keperluan dari *owner*.

UMMN

UNIVERSITAS
MULTIMEDIA
NUSANTARA