

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 Harry Adewijaya (2019) dengan judul “Sistem Informasi Penjualan Obat Berbasis Website Pada PT Kallista Prima Bekasi”. Pada PT. Kallista Prima Bekasi yang masih terbiasa berkegiatan bisnis secara konvensional begitu merasakan kesulitan karena harus melakukan jemput bola terhadap pelanggan. Berdasarkan permasalahan tersebut peneliti mencoba menghadirkan website *e-commerce* untuk PT. Kallista Prima Bekasi. Dalam membangun website ini peneliti menggunakan metode waterfall dan dibantu dengan *software* pengembangan website sublime dan menggunakan bahasa pemrograman php, html, css, serta menggunakan *framework codeigniter* dan *bootstrap*.

Berdasarkan penelitian yang dilakukan Petrick Prasetya (2018) dengan judul Sistem Informasi Pemesanan Obat Berbasis Web Pada PT. Anugrah Argon Medica (*Web-Based Drug Booking Information System* in PT. Medica Argon Anugrah), menyimpulkan bahwa PT. Anugrah Argon Medica masih dilakukan secara konvensional dimana setiap pembeli harus datang ke PT. Anugrah Argon Medica untuk melakukan transaksi pemesanan obat. Cara tersebut tentunya masih kurang efisien dan efektif dikarenakan membutuhkan waktu untuk datang ke PT.

Anugrah Argon Medica. Berdasarkan hal tersebut maka dibutuhkan sistem informasi pemesanan obat berbasis web PT. Anugrah Argon Medica. Dengan penggunaan sistem informasi pemesanan obat berbasis web PT. Anugrah Argon Medica berharap mampu untuk meningkatkan pelayanan kepada pelanggan, meningkatkan loyalitas pelanggan, meningkatkan pendapatan penjualan, memudahkan dalam membuat data laporan pemesanan obat.

Berdasarkan penelitian yang dilakukan oleh Siti Monalisa (2018) dengan judul “Rancang Bangun Sistem Informasi Inventory Obat Pada Rumah Sakit Jiwa Tampan Berbasis Web”, menyimpulkan bahwa Membantu dalam pengecekan stok obat yang masuk ataupun keluar, peringatan obat stok, peringatan kadaluarsanya obat sehingga tidak terjadi kekeliruan dalam pencatatan stok dan pemesanan dari gudang ke *supplier*. Memudahkan pegawai melihat perkembangan inventory obat yang dilakukan setiap transaksi sehingga meningkatkan efektivitas dan efisiensi kerja terutama dalam menghasilkan laporan. Menunjang dan mempercepat pengambilan keputusan serta *supplier* bisa langsung melihat kekurangan stok obat yang ada di Apotek RSJ Tampan Pekanbaru secara otomatis.

Berdasarkan penelitian dari Adella Rifiandika Putri, dkk (2021) dengan judul “Perancangan Sistem Informasi Penjualan Obat Online Pada Apotek Dara Berbasis Website” Apotek Dara masih menerapkan sistem penjualan dengan cara manual sehingga masih kurangnya kualitas layanan yang diberikan kepada *customer*. Perancangan ini bertujuan untuk memudahkan Apotek dara dalam manajemen proses transaksi, melakukan pendataan mengenai produk dan meningkatkan

pendapatan penjualan. Perancangan dan penelitian ini menggunakan bahasa pemrograman HTML dan PHP serta menggunakan *MySql* sebagai database.

Berdasarkan penelitian dari Anastasia Kembaren (2019) dengan judul “Sistem Informasi Pemesanan Obat Secara Online Pada PT. Mefrofam” Tujuan dari penelitian ini adalah untuk mengembangkan sistem informasi pemesanan obat pada PT. Meprofarm secara mudah dan efisien. Metode pengumpulan data yang digunakan dalam penyusunan Tugas Akhir ini adalah Observasi, Studi Pustaka, dan Wawancara. Program ini dikembangkan dengan menggunakan Macromedia Dreamweaver MX 2004 dengan menggunakan database Php Triad. Dengan adanya sistem usulan ini, pihak perusahaan dapat memperoleh informasi pemesanan barang secara cepat dan akurat. Selain itu, laporan pemesanan obat dapat dicetak sesuai kebutuhan dengan efisien dan efektif. Pemesanan Obat Secara *Online* Pada PT. Meprofarm dibuat oleh penulis yang siap dikembangkan.

Adapun perbedaan penelitian yang dilakukan penulis dengan penelitian terkait sebelumnya yaitu pada penelitian ini penulis menggunakan kombinasi *web* dan *Android* untuk melakukan pemesanan obat pada Apotik Mitha Farma. Sehingga dengan adanya aplikasi ini, dapat memudahkan pelanggan dalam melakukan pemesanan obat dan mempercepat proses penjualan pada Apotik Mitha Farma.

II.2. Landasan Teori

II.2.1. 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 pada sebuah komputasi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan. (Ari Setiaji, dkk : 2020).

II.2.2. Apotek

Apotek adalah sarana pelayanan kefarmasian tempat dilakukannya praktek kefarmasian oleh apoteker. Dalam hal ini, apoteker memiliki tanggung jawab untuk mewujudkan mutu pelayanan yang baik sesuai dengan harapan konsumen (Adella Rifiandika Putri : 2021).

II.2.3. Pemesanan Obat

Pemesanan adalah proses, perbuatan, cara memesan atau memesankan. Obat merupakan bahan untuk mengurangi, menghilangkan penyakit, atau menyembuhkan seseorang dari penyakit. *Online* adalah terhubung ke jaringan komputer atau diakses oleh komputer, berjalan dan terhubung ke sistem komputer. Dari pengertian kata-kata diatas dapat disimpulkan bahwa pemesanan obat secara *online* adalah proses atau cara memesan obat melalui jaringan komputer (Anastasia Kembaren : 2019)

II.2.4. Android

Android adalah suatu sistem operasi yang berbasis *Linux* dengan perangkat *Mobile device*, *Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi-aplikasi baru untuk membantu kebutuhan manusia. Awal terdirinya *Android* dimulai dari *Google Inc* yang membeli *Android inc* pendatang dari perangkat lunak untuk ponsel/*smartphone*. Semakin banyak *smartphone* dan PC *Tablet* menggunakan sistem operasi dengan versi yang berbeda. Semakin tinggi versinya maka fiturnya semakin canggih dan banyak. Telepon pertama yang memakai sistem operasi *Android* adalah *HTC Dream* yang dirilis pada tanggal 22 Oktober 2008. Perkembangan sistem *Android* berkembang sesuai versi keluarnya adapun versi *Android* dimulai dari (Muhamad Zaelani ; 2017: 5)

Tabel. II.1 Versi Android

Versi	Nama	Rilis	Catatan
1.0	<i>Android 1.0 Alpha</i>	23 September 2008	<i>Android pertama hanya untuk smartphone</i>
1.1	<i>Android 1.1 Beta</i>	9 Februari 2009	Hanya diperuntukan untuk satu perangkat
1.5	<i>Cupcake</i>	30 April 2009	Mulai pakai kode nama
1.6	<i>Donut</i>	15 September 2009	fasilitas pencarian di <i>android market</i>
2.0-2.1	<i>Éclair</i>	26 Oktober 2009 (2.0) 12 Januari 2010 (2.1)	terdapat beberapa fitur seperti <i>multi touch</i> .
2.2	<i>Froyo</i>	20 Mei 2010	peningkatan fitur USB <i>tethering</i> dan <i>hotspot WIFI</i>
2.3	<i>Gingerbread</i>	6 Desember 2010	Digunakan pada <i>smartphone</i> jenis lama
3.0-3.2	<i>Honeycomb</i>	22 Februari 2011 (3.0) 10 Mei 2011 (3.1) 15 Juli 2011 (3.2)	Digunakan pada <i>Smartphone</i>
4.0	<i>ICS (Ice Cream Sandwich)</i>	19 Oktober 2011	Digunakan pada <i>Smartphone</i>
4.1-4.3	<i>Jelly Bean</i>	9 Juli 2012 (4.1) 13 November 2012 (4.2) 24 Juli 2013 (4.3)	<i>Update</i> untuk memperbaiki dan menambah <i>fitur</i> pada <i>ICS</i>
4.4	<i>Kit kat</i>	3 September 2013	<i>android KitKat</i> seperti <i>WebViews</i>
5.0	<i>Lollipop</i>	12 november 2014 (5.0) 9 Maret 2015 (5.1)	berjalan pada perangkat mobile lainnya seperti <i>Android TV</i>
6.0	<i>Marshmallow</i>	5 oktober 015	Terdapat <i>daze mode</i> , <i>Do Not Disturb mode</i> , mendukung <i>USB tipe C</i> , mendukung pembacaan <i>fingerprint</i> .
7.0	<i>Nougat</i>	22 Agustus 2016	<i>Multitasking</i>

Sumber: Muhamad Zaelani (2017: 5)

Adapun penjelasan dari versi *Android* diatas adalah sebagai berikut :

1. *Android 1.0: Astro (Alpha)*

Kedua versi awal *Android* ini mungkin agak asing kamu dengarkan. Pasalnya versi *Android 1.0 Astro (Alpha)* dan *Android 1.1* Walaupun belum menggunakan nama makanan manis, kedua sistem operasi *Android* ini tentu menjadi *pionir*. Pasalnya di sinilah *Android* bermula lewat *smartphone* pertama, *HTC Dream*.

2. *Android 1.1 : Bender (Beta)*

Android 1.1 Bender (Beta) ini belum diluncurkan secara publik untuk kebutuhan komersil. *Platform Android* sendiri pertama kali diluncurkan pada Februari 2009 oleh Andy Rubin yang saat ini dikenal sebagai Bapak *Android*. Walaupun belum menggunakan nama makanan manis, kedua sistem operasi *Android* ini tentu menjadi *pionir*. Pasalnya di sinilah *Android* bermula lewat *smartphone* pertama, *HTC Dream*. (Muhamad Zaelani ; 2017: 5)

3. *Cupcake*

Mulai dari versi ini, *Android* menggunakan nama makanan manis untuk setiap versi yang diluncurkan. *Android 1.5 Cupcake* sendiri dirilis pada tanggal 30 April 2009 dengan berbagai fitur di sebuah perangkat *smartphone* untuk menggantikan *featured phone* kala itu. (Muhamad Zaelani ; 2017: 5)

4. *Donut*

Tentu pada awal perilisannya, sistem operasi *Android* tetap memiliki banyak *bug* yang pengembangnya perlu mengadakan perbaikan. Hal ini dilakukan pada *Android 1.6 Donut* yang dirilis pada 15 September 2009. Ya artinya belum genap setahun semenjak perilsan *Android 1.5 Cupcake* atau hanya berselang 5

bulan saja. *Android* pun menambahkan beberapa pembaruan, terutama dukungan pada layar *smartphone* yang lebih besar. (Muhamad Zaelani ; 2017: 5)

5. *Eclair*

Sama seperti versi sebelumnya, *Android 2.0 & 2.1 Eclair* masih berfungsi untuk menutupi *bug* yang masih ditemukan pada sistem operasi *mobile* ini. Di samping itu, *Android* juga menambah berbagai fitur di dalamnya.

Mulai dari dukungan *Bluetooth* hingga fitur kamera yang mulai menjadi nilai jual *smartphone* kala itu. *Android 2.0 & 2.1 Eclair* digunakan pada perangkat seperti *HTC Nexus One*. (Muhamad Zaelani ; 2017: 5)

6. *Android 2.2 : Froyo (Frozen Yoghurt)*

Mulai versi ini *Android* tampaknya sudah mulai dikenal luas oleh berbagai *brand smartphone*. *Android 2.2 Frozen Yoghurt* alias *Froyo* ini dirilis pertama kali pada tanggal 20 Mei 2010. Walaupun sudah mulai dipergunakan pada beberapa *brand*, namun tetap saja *Android* masih kalah bersaing dengan *Symbian* yang mendominasi pasar *featured phone*. *Android 2.2 Froyo* memberikan peningkatan pada kecepatan kerja, fitur *USB tethering*, *WiFi hotspot*, serta fitur keamanan. (Muhamad Zaelani ; 2017: 5)

7. *Gingerbread*

Belum setahun berselang lagi, *Android 2.3 Gingerbread* kembali diluncurkan pada Desember 2010 dengan berbagai peningkatan yang cukup signifikan. Hal ini terutama pada tampilan tatap muka alias *user interface* yang digunakan. Mulai versi ini, banyak *brand smartphone* mulai melirik menggunakan

sistem operasi *Android*. Salah satunya *Samsung Galaxy Series* yang populer hingga saat ini. (Muhamad Zaelani ; 2017: 5)

8. *Honeycomb*

Untuk para pengguna *smartphone* mungkin akan agak asing dengan versi *Android* yang satu ini. *Android 3.0 & 3.2 Honeycomb* yang menggunakan ikon lebah ini memang diperuntukkan penggunaannya khusus untuk perangkat *tablet*. Tentu perilisan *Android 3.0 & 3.2 Honeycomb* pada 10 Mei 2011 ini untuk mendukung *Samsung* yang mulai merilis perangkat *tablet*, *Samsung Galaxy Tab Series* untuk menyaingi *Apple iPad*. (Muhamad Zaelani ; 2017: 5)

9. *Ice Cream Sandwich (ICS)*

Selain itu, *Android* pun juga merilis versi *Android 4.0 Ice Cream Sandwich* yang kembali diperuntukkan untuk perangkat *smartphone*. *Android 4.0 Ice Cream Sandwich* sendiri dirilis pada 19 Oktober 2011 silam. Punya versi nama paling panjang hingga saat ini, *Android 4.0 Ice Cream Sandwich* memberikan banyak pembaruan. Mulai dari animasi yang semakin halus, sederhana, dan mudah digunakan. (Muhamad Zaelani ; 2017: 5)

10. *Jelly Bean*

Peningkatan signifikan terasa saat menggunakan *Android 4.1 & 4.3 Jelly Bean*. Sistem operasi ini sendiri pertama kali dirilis pada Juni 2012 dengan membawa sejumlah peningkatan terutama di sektor pengolahan grafis. Dengan begini, tentu *Android 4.1 & 4.3 Jelly Bean* bisa memberikan peningkatan fungsi pada *user interface* dan teknologi *Vsync* yang digunakannya. (Muhamad Zaelani ; 2017: 5)

11. *KitKat*

Menggunakan nama *brand* cemilan terkenal, *Android 4.4 KitKat* pertama kali dirilis pada Oktober 2013. Versi *Android* terbaik ini pun bisa dikatakan menjadi favorit karena mendukung hampir seluruh perangkat *smartphone* di dunia. Hal ini dikarenakan *Android 4.4 KitKat* dapat memberikan optimalisasi yang baik termasuk pada perangkat *smartphone* yang memiliki *spesifikasi* kurang mumpuni alias cukup rendah saat itu. (Muhamad Zaelani ; 2017: 5)

12. *Lollipop*

Mulai beberapa versi ke belakang, *Android* dan *Google* pun mulai secara rutin memperbarui sistem operasi mereka dalam selang waktu setahun. Termasuk *Android 5.0 & 5.1 Lollipop* yang dirilis dan diresmikan pada Juni 2014. Bisa dibilang *Android 5.0 & 5.1 Lollipop* menjadi *pionir* (pembuka jalan) dibuatnya *smartphone flagship* dengan spesifikasi cukup mumpuni. Versi *Android* ini sudah mendukung arsitektur 64-bit yang sudah memungkinkan penggunaan RAM di atas 3GB. Salah satunya *ASUS Zenfone 2* yang sudah mengusung RAM 4GB saat itu. (Muhamad Zaelani ; 2017: 5)

13. *Marshmallow*

Android 6.0 Marshmallow menjadi sukses dari versi *Android* sebelumnya. Sistem operasi ini sendiri pertama kali diperkenalkan pada Mei 2015 dan mulai dirilis pada Oktober 2015 silam. Sistem operasi ini secara jelas memberikan peningkatan pada sistem keamanan dengan dihadirkannya *fingerprint sensor* sebagai sistem keamanan *biometrik* yang digunakan. Selain digunakan untuk mengunci layar, *fingerprint sensor* ini dapat digunakan untuk otentikasi atau

proses memverifikasi identitas, *Google Play Store* dan pembelian dengan menggunakan *Android Pay*. (Muhamad Zaelani ; 2017: 5)

14. *Nougat*

Untuk saat ini, sistem operasi *Android* ini masih digunakan beberapa *smartphone* yang baru dirilis belakangan ini. *Android 7.0 & 7.1 Nougat* pertama kali diperkenalkan pada Juni 2016 dengan menampilkan ikon robot *Android* dengan batangan *Nougat*. Sistem operasi *Android 7.0 & 7.1 Nougat* mengalami perubahan dari segi tampilan antarmuka. Selain itu ada juga fitur *splitscreen* untuk membagi tampilan layar untuk dua aplikasi sekaligus. (Muhamad Zaelani ; 2017: 5)

II.2.5. *Android Studio*

Android Studio adalah Lingkungan Pengembangan Terpadu - *Integrated Development Environment (IDE)* untuk pengembangan aplikasi *Android*, berdasarkan *IntelliJ IDE*. Konsep yang dimiliki *Android Inc* ternyata menggugah minat *Google* untuk memilikinya. Pada bulan Agustus 2005, Akhirnya *Android Inc* diakuisisi oleh *Google Inc*. seluruh sahamnya dibeli oleh *Google*. Banyak yang memperkirakan nilai pembelian *Android Inc* Oleh *Google* adalah sebesar USD 50 juta. saat itu banyak yang berspekulasi bahwa akuisisi ini adalah langkah awal yang dilakukan *Google* untuk masuk kepasar *mobile phone*. Maka dari situlah banyak pengembang sistem maupun *software* berlomba lomba untuk membuat atau merancang sistem *Android* menggunakan *software – software* yang support dengan *Android*, dan sebagai contoh disini kita pengenalan *Android Studio*. Selain

merupakan editor kode *IntelliJ* dan alat pengembang yang berdaya guna, *Android Studio* menawarkan fitur lebih banyak untuk meningkatkan produktivitas Anda saat membuat aplikasi *Android*, misalnya:

1. Sistem versi berbasis *Gradle* yang *fleksibel*
2. *Emulator* yang cepat dan kaya fitur
3. Lingkungan yang menyatu untuk pengembangan bagi semua perangkat *Android*
4. *Instant Run* untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK (*Android Package Kit*) baru
5. *Template* kode dan integrasi/penyesuaian *GitHub* untuk membuat fitur aplikasi yang sama dan mengimpor kode
6. Alat pengujian dan kerangka kerja yang ekstensif atau menjangkau secara luas
7. Alat *Lint* untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah - masalah lain
8. Dukungan C++ dan NDK (*Native Development Kit*)
9. Dukungan bawaan untuk *Google Cloud Platform*, mempermudah pengintegrasian *Google Cloud Messaging* dan *App Engine*.

Laman ini berisi pengantar dasar fitur-fitur *Android Studio*. Untuk memperoleh rangkuman perubahan terbaru, lihat Catatan Rilis *Android Studio* (*Google developer* ; 2017).

II.2.6. Basis Data (*Database*)

Basis data atau *database* merupakan koleksi dari data-data yang terorganisir dengan rapi sehingga data dapat dengan mudah disimpan dan dimanipulasi. Kita dapat menjumpai pemanfaatan *database* dalam kehidupan sehari-hari, seperti penggunaan mesin ATM, sistem akademik Universitas/sekolah, sistem informasi penjualan. Salah satu tujuan dari *database* adalah memberikan pengguna suatu pandangan abstrak dari data, yaitu sistem menyembunyikan rincian bagaimana data disimpan dan dipelihara. Sistem *database* harus dibuat semudah mungkin untuk dimengerti karena kebanyakan pengguna sistem *database* adalah orang-orang yang kurang terlatih di bidang teknologi. (Ni Kadek Ceryna, dkk : 2018 : 102)

II.2.7. MySQL

MySQL (*My Structure Query Language*) 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 (*Structure Query Language*) 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 tabel yang terpisah. Tabel-tabel tersebut terhubung oleh suatu relasi terdefinisi yang memungkinkan memperoleh kombinasi data dari beberapa tabel 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.2.8. PHP

PHP atau kependekan dari *Hypertext Preprocessor* adalah salah satu bahasa pemrograman *open source* yang sangat cocok atau dikhususkan untuk pengembangan *web* dan dapat ditanamkan pada sebuah skripsi HTML (*HyperText Markup Language*). Bahasa *PHP* dapat dikatakan menggambarkan beberapa bahasa pemrograman seperti *C*, *Java*, dan *Perl* serta mudah untuk dipelajari. *PHP* merupakan bahasa *scripting server – side*, dimana pemrosesan datanya dilakukan pada sisi *server*. Sederhananya, *server* yang akan menterjemahkan *script* program, baru kemudian hasilnya akan dikirim kepada *client* yang melakukan permintaan. Adapun pengertian lain *PHP* adalah *akronim/kependekan* dari *Hypertext Preprocessor*, yaitu suatu bahasa pemrograman berbasiskan kode – kode (*script*) yang digunakan untuk mengolah suatu data dan mengirimkannya kembali ke *web browser* menjadi kode *HTML*”. (Astria Firman, 2016 : 30).

PHP (Hypertext Preprocessor) adalah salah satu *server side* yang dirancang khusus untuk aplikasi *web*. *PHP* disisipkan diantara bahasa *HTML* dan karena

bahasa *server side*, maka bahasa *PHP* akan dieksekusi di *server*, sehingga yang dikirimkan ke *browser* adalah hasil jadi dalam bentuk *HTML*, dan kode *PHP* tidak akan terlihat. *PHP* termasuk *Open Source Product*. Jadi, dapat diubah *source code* dan mendistribusikannya secara bebas (Priyo Sutopo, dkk : 2017)

PHP (singkatan rekursif: *PHP Hypertext Preprocessor*) merupakan script yang dapat disisipkan ke dalam *HTML*. *PHP* dipakai untuk membuat sistem berbasis *web* yang dinamis. *PHP* bahasa pemrograman yang *server-side* sehingga program tersebut akan dijalankan atau diproses oleh *server*. Hingga saat ini *PHP* sudah rilis versi 7.4.1 pada 18 desember 2019. (Roberto kaban,. Dkk : 2019)

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

Unified Modelling Language (UML) merupakan satu kumpulan konvensi atau aturan 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.

Unified Modeling Language (UML) biasa digunakan untuk :

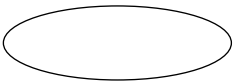
- a. Menggambarkan batasan sistem dan fungsi - fungsi sistem secara umum, dibuat dengan *use case* dan *actor*.

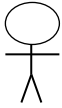
- b. Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum, dibuat dengan *interaction diagrams*.
- c. Menggambarkan representasi struktur *static* sebuah sistem dalam bentuk *class diagrams*.
- d. Membuat model *behavior* “yang menggambarkan kebiasaan atau sifat sebuah sistem” dengan *state transition diagrams*.
- e. Menyatakan arsitektur implementasi fisik menggunakan *component and development*.
- f. Menyampaikan atau memperluas *fungsi* dengan *stereotypes*. (Omni 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> .

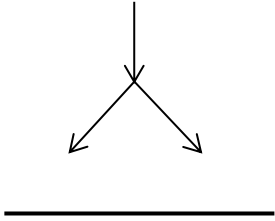
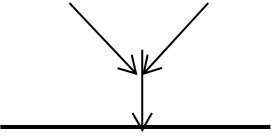
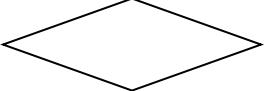
	Aktor	Sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem.
	Asosiasi	Penghubung antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan 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 di bawah 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.

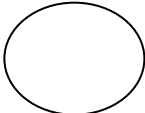
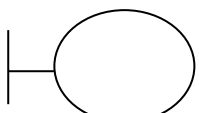
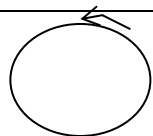
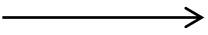
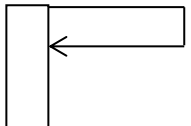
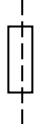
(Sumber : Ade Hendini, 2016 : 109)

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/wujud 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 <i>formentry</i> 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)

II.2.10. Normalisasi

Normalisasi merupakan sebuah teknik dalam logical desain sebuah basis data, teknik pengelompokkan atribut dari suatu relasi sehingga membentuk struktur relasi yang baik tanpa redundansi. Tujuan normalisasi adalah mengorganisasikan data kedalam tabel-tabel untuk memenuhi kebutuhan pemakai, menghilangkan kerangkapan data, mengurangi kompleksitas, mempermudah modifikasi data. (Mukhlisulfatih Latief : 2018)

1. Proses Normalisasi

- a. Data diuraikan dalam bentuk tabel, selanjutnya dianalisis berdasarkan persyaratan tertentu kebeberapa tingkat.
- b. Apabila tabel yang diuji belum memenuhi persyaratan tertentu maka tabel tersebut perlu dipecah menjadi beberapa tabel yang lebih sederhana sampai memenuhi bentuk yang optimal.

2. Tahapan Normalisasi :

- 1) Bentuk tidak normal : Menghilangkan perulangan *grup*.

Tabel II.6. Contoh bentuk tidak normal (Unnormal)

No-Mhs	Nama Mhs	Jurusan	Kode-MK	Nama-MK	Kode Dosen	Nama Dosen	Nilai
2683	Welli	MI	M1350	Manajemen DB	B104	Ati	A
			M1465	Analisis Perc. Sistem	B317	Dita	B
5432	Bakti	AK	M1350	Manajemen DV	B104	Ati	C
			Akn201	Akuntansi	D310	Lia	B
			MKT300	Dasar Pemasaran	B212	Lola	A

Sumber : Mukhlisulfatih Latief : 2018

2) Bentuk Normal pertama (1NF) : Menghilangkan ketergantungan sebagian.

Yaitu : suatu relasi dikatakan sudah memenuhi bentuk normal kesatu bila setiap data bersifat atomik yaitu setiap irisan baris dan kolom hanya mempunyai satu nilai data.

Tabel II.7. Contoh Bentuk Normal Pertama (1NF)

No-Mhs	Nama Mhs	Jurusan	Kode-MK	Nama-MK	Kode Dosen	Nama Dosen	Nilai
2683	Welli	MI	M1350	Manajemen DB	B104	Ati	A
2683	Welli	MI	M1465	Analisis Perc. Sistem	B317	Dita	B
5432	Bakti	AK	M1350	Manajemen DV	B104	Ati	C
5432	Bakti	AK	Akn201	Akuntansi	D310	Lia	B
5432	Bakti	AK	MKT300	Dasar Pemasaran	B212	Lola	A

Sumber : Mukhlisulfatih Latief : 2018

3) Bentuk Normal kedua (2NF) : Menghilangkan ketergantungan transitif.

Yaitu : suatu relasi dikatakan sudah memenuhi bentuk normal kedua bila relasi tersebut sudah memenuhi bentuk normal kesatu dan atribut yang bukan key sudah tergantung penuh terhadap key-nya.

Tabel II.8. Contoh Bentuk Normal Kedua (2NF)

Kode-MK	Nama-MK	Kode Dosen	Nama Dosen
M1350	Manajemen DB	B104	Ati
M1465	Analisis Perc. Sistem	B317	Dita
M1350	Manajemen DV	B104	Ati
Akn201	Akuntansi	D310	Lia
MKT300	Dasar Pemasaran	B212	Lola

Sumber : Mukhlisulfatih Latief : 2018

4) Bentuk Normal ketiga (3NF) : Menghilangkan anomali-anomali hasil dari ketergantungan fungsional. Yaitu : suatu relasi dikatakan sudah memenuhi bentuk normal ketiga bila relasi tersebut sudah memenuhi bentuk normal kedua dan atribut yang bukan key tidak tergantung transitif terhadap *key*-nya. (Mukhlisulfatih Latief : 2018)

Tabel II.9. Contoh Tabel Mahasiswa Dan Tabel Kuliah (3NF)

No_Mhs	Nama Mhs	Jurusan
2683	Welli	MI
5432	Bakti	AK

Sumber : Mukhlisulfatih Latief : 2018