

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Untuk mendukung keberhasilan penelitian ini, penyusun melakukan pendekatan teoritis melalui beberapa literatur yang berhubungan dengan penelitian yang dilakukan. Beberapa uraian penelitian terdahulu yang menjadi acuan dalam penelitian ini yaitu:

Berdasarkan penelitian yang dilakukan Zia Rizki Saputri (2019) dengan judul “Rancang Bangun Sistem Informasi Pemesanan Makanan Berbasis Web Pada Cafe Surabiku”, Pada saat ini Cafe Surabiku mengalami kesulitan dalam melayani pemesanan dan masih menggunakan pencatatan yang belum terkomputerisasi atau masih menggunakan sistem konvensional. Di mana pembeli harus menunggu terlalu lama untuk memesan makanan, selain itu pelayan juga mengalami kesulitan dalam mencatat pesanan secara manual menggunakan kertas. Pencatatan pesanan secara manual kurang efisien dari sisi waktu. Maka untuk mewujudkan kepuasan konsumen dalam pemesanan makanan dibuatlah sebuah rancang bangun sistem informasi pemesanan makanan berbasis web dengan menggunakan PHP dan MySQL.

Berdasarkan penelitian yang dilakukan Tuti Handayan (2020) dengan judul “Rancang Bangun Sistem Informasi Pemesanan Menu Makanan Berbasis Web (Studi Kasus: Restoran Bukit Randu Bandara)”. Restoran Bukit Randu tidak memiliki sistem pemesanan menu makanan secara *online*, sehingga kurang dapat

bertahan dalam persaingan dengan restoran lainnya. Hal inilah yang mendorong pemilik restoran agar memberikan pelayanan yang terbaik untuk pelanggan dan tidak harus antre ketika memesan makanan. Penelitian ini membuat rancang bangun sistem informasi pemesanan menu makanan berbasis web. Dalam penelitian ini metode yang digunakan dengan cara observasi, yaitu mengamati alur proses sistem pemesanan, wawancara meliputi proses penilaian kualitas pelayanan yang sedang berjalan, dan studi kepustakaan

Berdasarkan penelitian yang dilakukan oleh Nia Caniati (2018) dengan judul “Implementasi Sistem Informasi Pemesanan Menu Makanan Dan Minuman Pada Kafe Berbasis Web Menggunakan Jaringan Intranet”. Kafe merupakan tempat untuk bersantai dan berbincang-bincang di mana pengunjung dapat memesan minuman dan makanan. Pada umumnya restoran ataupun kafe mengalami kesulitan untuk melayani pemesanan menu makanan dan minuman. Kesulitan tersebut adalah pelanggan menunggu terlalu lama untuk mendapatkan menu makanan dan minuman yang dipesan sehingga kurang adanya kenyamanan bagi pengunjung.

Berdasarkan penelitian dari Mhd Dicky Syahputra Lubis (2020) dengan judul “Perancangan Dan Pembuatan Aplikasi Pemesanan Makanan Online Berbasis Android” Pada umumnya, ketika pelanggan ingin memesan makanan harus mengunjungi restoran secara langsung, memilih menu makanan dan melakukan pemesanan. Hal ini kurang efisien dan efektif karena membutuhkan biaya dan waktu yang cukup lama, di antaranya menunggu antrian, terhalang oleh cuaca di perjalanan dan terjebak kemacetan. Berdasarkan permasalahan tersebut

dibutuhkan sebuah aplikasi pemesanan makanan berbasis Android yang dapat membantu pelanggan dalam memesan makanan dengan cepat, melihat daftar restoran dan menu makanan. Melalui aplikasi ini juga dapat membantu penjual atau pemilik setiap restoran untuk mempermudah pengelolaan penjualan makanan dari restoran tersebut.

II.2. Uraian Teoritis

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. Aplikasi *mobile* dapat diartikan sebagai sebuah produk dari sistem komputasi *mobile*, yaitu sistem komputasi yang dapat dengan mudah dipindahkan secara fisik dan kemampuan yang komputasi dapat digunakan saat mereka sedang dipindahkan. Contohnya adalah *Personal Digital Assistant (PDA)*, *smartphone* dan ponsel. (Fergiawan Listianto : 2018).

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. Aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputasi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan (Ari Setiaji, dkk : 2020).

II.2.2. Android Studio

Android merupakan sistem operasi *mobile*. Android tidak membedakan antara aplikasi inti dengan aplikasi pihak ketiga. *Application Programming Interface* (API) yang disediakan menawarkan akses ke *hardware*, maupun data data ponsel sekalipun, atau data sistem sendiri. Android merupakan sebuah sistem operasi perangkat *mobile* berbasis Linux yang mencakup sistem operasi, *middleware*, dan aplikasi. Beberapa pengertian lain dari Android, yaitu:

1. Merupakan *platform* terbuka (open source) bagi para pengembang (programer) untuk membuat aplikasi.
2. Merupakan sistem operasi yang dibeli Google Inc. dari Android Inc.
3. Bukan bahasa pemrograman, tetapi hanya menyediakan lingkungan hidup atau *run time enviroment* yang disebut DVM (*Dalvik Virtual Machine*) yang telah dioptimasi untuk alat/*device* dengan sistem memori yang kecil (Ni Kadek Ceryna : 2018 : 101).

II.2.3 Tools Pembangunan Android

Untuk membangun sebuah sistem operasi Android dapat menggunakan Mac, Windows PC, ataupun Linux. *Tools* yang dibutuhkan gratis dan dapat di-*download* dari *web*. Berikut adalah beberapa *tools* yang digunakan untuk membangun aplikasi Android:

1. JDK (*Java Development Kit*)

JDK (*Java Development Kit*) adalah paket fungsi API untuk bahasa pemrograman Java, meliputi *Java Runtime Environment* (JRE) dan *Java Virtual Machine* (JVM). JDK adalah perangkat lunak yang digunakan untuk

melakukan proses kompilasi dari kode Java ke *bytecode* yang dapat dimengerti dan dapat dijalankan oleh JRE (*Java Runtime Envirotment*). JDK wajib terinstall pada komputer yang akan melakukan proses pembuatan aplikasi berbasis Java, tetapi tidak wajib terinstall di komputer yang akan menjalankan aplikasi yang dibangun dengan Java. Tanpa adanya JDK maka kode-kode Java sudah dibuat tidak akan bisa dijadikan aplikasi berbasis Java (Ni Kadek Ceryna, dkk : 2018 : 102).

2. Android SDK

Android Software Development Kit (SDK) merupakan *kit* yang bisa digunakan oleh para *developer* untuk mengembangkan aplikasi berbasis Android. Di dalamnya, terdapat beberapa *tools* seperti *debugger*, *software libraries*, *emulator*, dokumentasi, *sample code*, dan tutorial (Ni Kadek Ceryna, dkk : 2018 : 102).

3. ADT (*Android Development Tools*)

Android Development Tools (ADT) adalah *plugin* untuk Eclipse yang didesain untuk pengembangan aplikasi Android. ADT memungkinkan Eclipse untuk digunakan dalam membuat aplikasi Android baru, membuat *User Interface*, menambahkan komponen berdasarkan *framework* API Android, *debug* aplikasi, dan pemaketan aplikasi Android (Ni Kadek Ceryna, dkk : 2018 : 102).

II.2.4. Perkembangan Android

Menurut Muhamad Zaelani ; 2018: 5 Perkembangan Sistem *Android* berkembang sesuai versi keluarnya. Adapun Versi Android dimulai dari:

1. Android versi 1.1

Android memang diluncurkan pertama kali pada tahun 2007, tetapi sistem operasi ini mulai dirilis dan diterapkan ke berbagai *gadget* pada tanggal 9 Maret 2009 silam. Android versi 1.1 merupakan Android awal yang di mana versi ini baru memberikan sentuhan di beberapa aplikasinya seperti sistem antar muka bagi pengguna (*user interface*) yang lebih baik, serta beberapa aplikasi yang lain. (Kusniyati & Pangondian Sitanggang, 2019)



Gambar II.1. Android versi 1.1

Sumber : .(Kusniyati & Pangondian Sitanggang, 2019)

2. Android versi 1.5 (Cupcake)

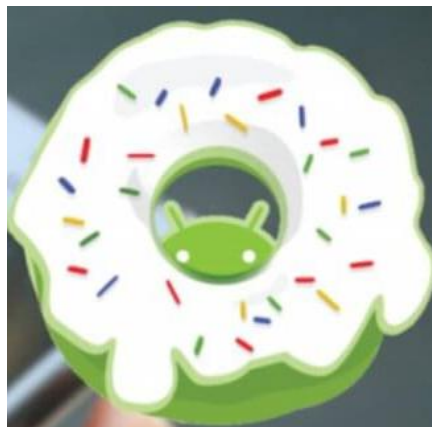
Pada bulan Mei 2009 Android kembali mengalami perubahan versi. Android versi 1.1 kemudian disempurnakan dengan Android versi 1.5 atau yang dikenal sebagai Android Cupcake (Kusniyati & Pangondian Sitanggang, 2019).



Gambar II.2. Android versi 1.5 (Cupcake)
Sumber : .(Kusniyati & Pangondian Sitanggang, 2019)

3. Android versi 1.6 (Donut)

Donut (versi 1.6) diluncurkan dalam tempo kurang dari 4 bulan semenjak peluncuran perdana Android Cupcake, yaitu pada bulan September 2009 (Kusniyati & Pangondian Sitanggang, 2019).

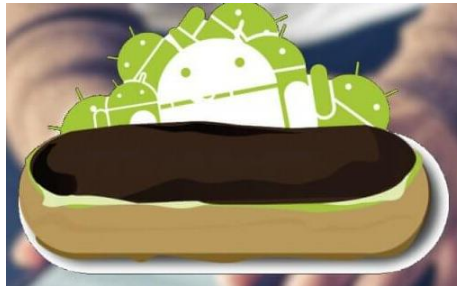


Gambar II.3. Android Versi 1.6 (Donut)
Sumber : .(Kusniyati & Pangondian Sitanggang, 2019)

4. Android versi 2.0/2.1 (Eclair)

Masih ditahun yang sama, Android kembali merilis operating sistem versi terbarunya, yaitu Android versi 2.0/2.1 Eclair. Android Eclair diluncurkan oleh

Google 3 bulan setelah peluncuran (Kusniyati & Pangondian Sitanggang, 2019).



Gambar II.4. Android Versi 2.0/2.1 (Eclair)
Sumber : .(Kusniyati & Pangondian Sitanggang, 2019)

5. Android versi 2.2 (Froyo: Frozen Yoghurt)

Butuh 5 bulan bagi Google untuk melakukan regenerasi dari Android Eclair versi sebelumnya ke versi Froyo Frozen Yoghurt. Pada tanggal 20 Mei 2010, Android versi 2.2 alias Android Froyo ini dirilis (Kusniyati & Pangondian Sitanggang, 2019).



Gambar II.5. Android Versi 2.2 (Froyo: Frozen Yoghurt)
Sumber : .(Kusniyati & Pangondian Sitanggang, 2019)

6. Android versi 2.3 (Gingerbread)

7 bulan kemudian Android kembali melakukan gebrakan dengan merilis kembali Android versi 2.3 atau yang dikenal sebagai Android Gingerbread (Kusniyati & Pangondian Sitanggang, 2019).



Gambar II.6. Android Versi 2.3 (Gingerbread)

Sumber : .(Kusniyati & Pangondian Sitanggang, 2019)

7. Android versi 3.0/3.1 (Honeycomb)

Pada bulan Mei 2011 Android versi 3.0/3.1 atau Android Honeycom dirilis. Android Honeycomb merupakan sebuah sistem operasi Android yang tujuannya memang dikhususkan bagi penggunaan tablet berbasis Android (Kusniyati & Pangondian Sitanggang, 2019).

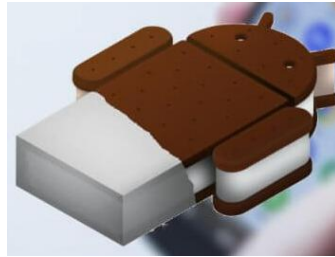


Gambar II.7. Android Versi 3.0/3.1 (Honeycomb)

Sumber : .(Kusniyati & Pangondian Sitanggang, 2019)

8. Android versi 4.0 (ICS: Ice Cream Sandwich)

Android ICS atau Ice Cream Sandwich juga dirilis pada tahun yang sama dengan Honeycomb, yaitu pada bulan Oktober 2011 (Kusniyati & Pangondian Sitanggang, 2019).



Gambar II.8. Android Versi 4.0 (ICS: Ice Cream Sandwich)

Sumber : .(Kusniyati & Pangondian Sitanggang, 2019)

9. Android versi 4.1 (Jelly Bean)

Android Jelly Bean merupakan versi Android yang terbaru pada saat ini. Salah satu gadget yang menggunakan sistem operasi Jelly Bean adalah Google Nexus 7 yang diprakarsai oleh ASUS, vendor asal Taiwan yang juga menjadi teman satu kampung halaman dengan Acer (Kusniyati & Pangondian Sitanggang, 2019).



Gambar II.9. Android Versi 4.1 (Jelly Bean)

Sumber : .(Kusniyati & Pangondian Sitanggang, 2019)

10. Android versi 4.4 (Kit Kat)

Kehadiran android kitkat merupakan peluncuran produk OS anyar yang diluncurkan pada 4 september 2013, sebelumnya banyak kabar beredar jikalau android akan meluncurkan OS baru yang bernama Android Key Lime Pie namun setelah di analisa tidak sesuai dengan ejaan orang umum, sehingga namanya diganti dengan OS Android KitKat yang sebagian besar orang sudah familiar dengan itu (Kusniyati & Pangondian Sitanggang, 2019).



Gambar II.10. Android Versi 4.4 (Kit Kat)

Sumber : .(Kusniyati & Pangondian Sitanggang, 2019)

11. Android versi 5.0.2 (Lollipop)

Android Lollipop merupakan keberadaan OS Android yang memang saat ini sudah menjadi trend baru di industri smartphone, hal ini tak lepas dari keunikan dan kelebihan yang banyak di miliki dari OS tersebut. Kehadiran android versi ini amat di nanti oleh sekian banyak orang karna diharapkan sistem operasi Lollipop ini bias lebih baik dibandingkan versi-versi sebelumnya (Kusniyati & Pangondian Sitanggang, 2019).



Gambar II.11. Android Versi 5.0.2 (Lollipop)

Sumber : .(Kusniyati & Pangondian Sitanggang, 2019)

12. Android versi 6.0 (Marshmallow)

Android 6.0 Marshmallow adalah versi dari sistem operasi mobile Android. Pertama kali diperkenalkan Mei 2015 di Google I / O di bawah kode nama Android M, secara resmi dirilis pada Oktober 2015. Android Marshmallow memperkenalkan model izin aplikasi didesain ulang sekarang ada hanya delapan kategori izin, dan aplikasi yang tidak lagi secara otomatis diberikan semua hak akses mereka ditentukan pada waktu instalasi (Kusniyati & Pangondian Sitanggang, 2019).



Gambar II.12. Android Versi 6.0 (Marshmallow)

Sumber : .(Kusniyati & Pangondian Sitanggang, 2019)

13. Android 7.0 Nougat

Diperkenalkan pada tanggal 19 Oktober 2019 Android 7.0 Nougat berfokus pada peningkatan performa *user interface* sehingga lebih intuitif dan penggunaan aplikasi secara bersamaan lebih banyak pada fitur *multi window*. Selain peningkatan fitur tadi, Android Nougat juga menambahkan beberapa fitur lain seperti dukungan cahaya malam atau mode malam, *keyboard default* yang dapat mengirim animasi GIF langsung dan dukungan panggilan *multi-endpoint*.



Gambar II.13. Android Versi 7.0 Nougat

Sumber : .(Kusniyati & Pangondian Sitanggang, 2019)

14. Android 8.0 Oreo

Nama Oreo dipilih Android untuk digunakan pada versi Android 8.0 yang diluncurkan pada bulan Agustus 2018. *User interface* pada Android Oreo lebih simpel agar memudahkan dalam mengakses aplikasi. Pembaruan pada versi Oreo membawa beberapa fitur seperti fitur *Autofill* yang memberikan kemudahan dalam mengisi formulir misal, dukungan gambar dalam gambar dan pengoptimalan *booting* agar lebih cepat.



Gambar II.14. Android Versi 8.0 Oreo

Sumber : .(Kusniyati & Pangondian Sitanggang, 2019)

15. Android 9.0 Pie

Android 9.0 Pie merupakan versi Android terbaru yang dirilis pada bulan Agustus 2018. Fitur unggulan yang dimiliki oleh versi 9.0 Pie ini adalah kemampuan AI atau kecerdasan buatan. Selain itu fitur lainnya yang diusung seperti *Adaptive Brightness* yang akan menyesuaikan kecerahan layar secara otomatis dan dukungan pada ponsel *bezel less*.



Gambar II.15. Android Versi 9.0 Pie

Sumber : .(Kusniyati & Pangondian Sitanggang, 2019)

16. Android 10

Versi Android terbaru diberi nama Android 10 untuk memperingati bahwa Android telah mencapai 1 dekade secara komersial. Versi Android 10 lebih

berfokus pada penyempurnaan mode malam atau gelap serta peningkatan fitur *sound amplifier* untuk mengatur kualitas audio.



Gambar II.16. Android 10

Sumber : .(Kusniyati & Pangondian Sitanggang, 2019)

II.2.5. 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 akademi kuniversitas/sekolah, dan 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 : 2018 : 102).

II.2.6 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. Bahasa

PHP dapat dikatakan menggambarkan beberapa bahasa pemrograman seperti C, *Java*, dan *Perl* serta mudah untuk dipelajari. PHP merupakan bahasa *scripting server-side*, di mana pemrosesan datanya dilakukan pada sisi *server*. Sederhananya, *server* yang akan menerjemahkan skrip program, baru kemudian hasilnya akan dikirim kepada *client* yang melakukan permintaan. Adapun pengertian lain PHP adalah *akronim* dari *Hypertext Preprocessor*, yaitu suatu bahasa pemrograman berbasis kode kode (*script*) yang digunakan untuk mengolah suatu data dan mengirimkannya kembali ke *web browser* menjadi kode HTML (Astria Firman, 2016 : 30).

II.2.7 MySQL

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 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-table 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 *phpMyAdmin* dengan aplikasi *XAMPP* (Saipul Anwar, 2018).

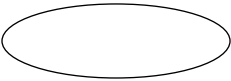
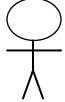
II.2.8. UML (*Unified Modelling Language*)

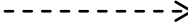
UML merupakan singkatan dari *Unified Modeling Language* yang merupakan sekumpulan alat yang digunakan untuk melakukan abstraksi terhadap sebuah sistem atau perangkat lunak berbasis objek. UML juga menjadi salah satu cara untuk mempermudah pengembangan aplikasi yang berkelanjutan. Aplikasi atau sistem yang tidak terdokumentasi harus melakukan penelusuran dan mempelajari kode program. UML juga dapat menjadi alat bantu untuk *transfer* ilmu tentang sistem atau aplikasi yang akan dikembangkan dari satu *developer* ke *developer* lainnya (Janiver W. Janis : 2020).

II.2.8.1 *Use Case Diagram*

Use case diagram merupakan gambaran grafis dari beberapa atau semua aktor, *use case*, dan interaksi yang memperkenalkan suatu sistem. Simbol-simbol *use case diagram* dapat dilihat pada tabel II.1:

Tabel II.1 Simbol *Use Case*

Gambar	Keterangan
	<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 adalah <i>abstraction</i> 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 <i>use case</i> , tetapi tidak memiliki kontrol terhadap <i>use case</i> .

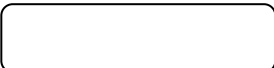
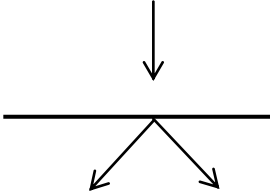
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor 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.

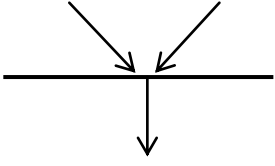
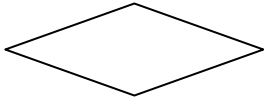
(Sumber : Janiver W. Janis; 2020)

II.2.8.2 Diagram Aktivitas (*Activity Diagram*)

Activity diagram merupakan diagram yang menggambarkan tentang aktifitas yang terjadi pada sistem. Dari pertama sampai akhir, diagram ini menunjukkan langkah-langkah dalam proses kerja sistem yang kita buat. Simbol-simbol *activity diagram* dapat dilihat pada tabel II.2:

Tabel II.2 Simbol *Activity Diagram*

Gambar	Keterangan
	<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 paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.

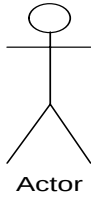
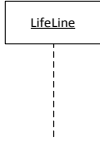
	<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan: <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

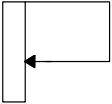
(Sumber : Janiver W. Janis; 2020)

II.2.8.3 Sequence Diagram

Sequence diagram merupakan diagram yang menjelaskan interaksi objek yang berdasarkan urutan waktu. *Sequence diagram* dapat menggambarkan urutan atau tahapan yang harus dilakukan untuk dapat menghasilkan sesuatu seperti pada *use case diagram*. Simbol-simbol yang digunakan dalam *Sequence Diagram*, yaitu:

Tabel II.3 Sequence Diagram

Gambar	Keterangan
	Proses atau sistem lain yang berinteraksi dengan sistem informasi.
	Merepresentasikan entitas tunggal dalam <i>sequence diagram</i> , digambarkan dengan kotak. Entitas ini memiliki nama, <i>stereotype</i> atau berupa <i>instance</i> .

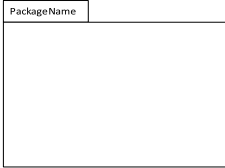
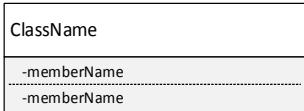
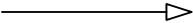
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Janiver W. Janis; 2020)

II.2.8.4 Class Diagram (Diagram Kelas)

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan *variabel-variabel* yang dimiliki oleh suatu kelas. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Diagram kelas mendeskripsikan jenis-jenis objek dalam sistem dan berbagai hubungan statis yang terdapat di antara mereka. Diagram kelas juga menunjukkan properti dan operasi sebuah kelas dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut.

Tabel II.4. Simbol *Class Diagram*

Simbol	Penjelasan
Package 	Package merupakan sebuah bungkusan dari satu atau lebih kelas.
Operasi 	Kelas pada struktur sistem
Antarmuka / interface 	Sama dengan konsep interface dalam pemrograman berorientasi objek.
Asosiasi 	Relasi antar kelas dengan makna umum, asosiasi biasanya disertai dengan multiplicity.
Asosiasi berarah / directed asosiasi 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai dengan multiplicity.
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus).
Kebergantungan / dependency	Relasi antar kelas dengan makna kebergantungan antar kelas.

	
Agregasi 	Relasi antar kelas dengan makna semua bagian (whole-part).

(Sumber : Rosa dan Shalahuddin, 2018)

II.2.9. 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 ke dalam tabel-tabel untuk memenuhi kebutuhan pemakai, menghilangkan kerangkapan data, mengurangi kompleksitas, dan mempermudah modifikasi data (Mukhlisulfatih Latief : 2019).

1. Proses Normalisasi

- a. Data diuraikan dalam bentuk tabel, selanjutnya dianalisis berdasarkan persyaratan tertentu ke beberapa 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:

- a. Bentuk tidak normal : Menghilangkan perulangan grup. Adapun contoh bentuk tidak normal (Unnormal) dapat dilihat pada Tabel II.5:

Tabel II.5. 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 : 2019

- b. Bentuk Normal Pertama (1NF) : Menghilangkan ketergantungan sebagian.

Yaitu : suatu relasi dikatakan sudah memenuhi bentuk normal Pertama bila setiap data bersifat atomik yaitu setiap irisan baris dan kolom hanya mempunyai satu nilai data. Adapun contoh bentuk normal pertama (1NF) dapat dilihat pada Tabel II.6:

Tabel II.6 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 : 2019

- c. Bentuk Normal Kedua (2NF) : Menghilangkan ketergantungan transitif.

Yaitu : suatu relasi dikatakan sudah memenuhi bentuk normal kedua bila relasi tersebut sudah memenuhi bentuk normal Pertama dan atribut yang bukan *key* sudah tergantung penuh terhadap *key*-nya. Adapun contoh bentuk normal pertama (1NF) dapat dilihat pada Tabel II.7:

Tabel II.7. 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 : 2019

- d. 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. Adapun contoh tabel Mahasiswa dan Tabel Kuliah (3NF) dapat dilihat pada Tabel II.8:

Tabel II.8. Contoh Tabel Mahasiswa Dan Tabel Kuliah (3NF)

No_Mhs	Nama Mhs	Jurusan
2683	Welli	MI
5432	Bakti	AK

Sumber : Mukhlisulfatih Latief : 2019