

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Ahmad Fauzi, et. al., 2018 melakukan penelitian berjudul “Penerapan Metode Haversine Formula Pada Aplikasi Pencarian Lokasi Tempat Tambal Ban Kendaraan Bermotor Berbasis Mobile Android”. Pada penelitian tersebut metode haversine diterapkan pada aplikasi pencarian lokasi tambal ban. Sehingga pada penelitian tersebut dihasilkan sebuah aplikasi berbasis android yang dapat digunakan untuk membantu pengguna dalam melakukan pencarian lokasi tambal ban dari sekitar pengguna dari yang terdekat hingga terjauh jarak lokasinya. Yang menjadi perbedaan dengan penelitian saat ini adalah pada penelitian ini metode haversine akan diterapkan pada penelitian ini akan digunakan untuk menampilkan jarak ke lokasi penyedia jasa *service* elektronik.

Sarif Ifan Purnawan, 2018 melakukan penelitian berjudul “Aplikasi Pencarian Pariwisata Dan Tempat Oleh-Oleh Terdekat Menggunakan Metode Haversine Berbasis Android”. Dari penelitian tersebut dihasilkan aplikasi untuk mencari lokasi pariwisata dan tempat oleh-oleh terdekat menggunakan metode haversine. Aplikasi yang dihasilkan dapat digunakan pada *smartphone* android sehingga dapat digunakan secara *mobile* dan efisien oleh penggunanya. Sedangkan pada penelitian ini akan dihasilkan aplikasi untuk mencari penyedia jasa *service* elektronik dan ditampilkan jaraknya menggunakan metode haversine.

Safaruddin Hidayat Al Ikhsan, et. al., 2019 melakukan penelitian berjudul “Aplikasi Android Sebaran Lokasi UMKM di Kota Bogor Dengan Formula Haversine”. Pada penelitian tersebut, metode haversine digunakan untuk menampilkan jarak ke lokasi UMKM yang terdapat di kota Bogor. Selanjutnya pada penelitian tersebut aplikasi yang dihasilkan dapat digunakan oleh penggunanya melalui *smartphone* dengan sistem operasi android. Sedangkan pada penelitian ini, metode haversine akan digunakan untuk menampilkan jarak ke lokasi penyedia jasa *service* elektronik dari yang terdekat hingga terjauh jaraknya dari pengguna aplikasi.

Pitrasacha Adytia, et. al., 2021 melakukan penelitian berjudul “Pencarian Driver Dry Clean Terdekat Dengan Metode Haversine Formula ”. Pada penelitian tersebut, metode haversine digunakan untuk menampilkan jarak lokasi driver dry clean yang terdekat dari lokasi penggunaan aplikasi. Sehingga dengan menggunakan aplikasi tersebut, pengguna dapat memesan driver untuk mengantarkan dan mengambil barang antara lokasi pengguna dengan lokasi Dry Clean. Sedangkan pada penelitian ini, metode haversine akan digunakan untuk menampilkan jarak ke lokasi penyedia jasa *service* elektronik dari yang terdekat hingga terjauh jaraknya dari pengguna aplikasi.

Prasetio, 2021 melakukan penelitian berjudul “Perancangan Aplikasi Simulasi Metode Haversine Dalam Pencarian Lokasi PKL Terdekat Untuk Siswa/I SMKS Mustafa Perbaungan”. Pada penelitian tersebut metode haversine digunakan untuk menampilkan lokasi PKL yang terdekat dari SMKS Mustafa Perbaungan. Aplikasi yang dihasilkan dari penelitian tersebut merupakan berbasis

website sehingga dapat dijalankan pada web browser. Dengan menggunakan aplikasi tersebut, siswa/i dapat dengan mudah melakukan pencarian lokasi-lokasi kegiatan PKL yang terdapat di daerah Perbaungan. Sedangkan pada penelitian ini, metode haversine akan digunakan untuk menampilkan jarak ke lokasi penyedia jasa *service* elektronik dari yang terdekat hingga terjauh jaraknya dari pengguna aplikasi.

II.2. Uraian Teoritis

II.2.1. Perancangan

Perancangan adalah penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari berbagai elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi. Pengertian perancangan menurut para ahli diantaranya adalah sebagai berikut :

1. Menurut Varzello/John Reuter III perancangan adalah tahap setelah analisis dari siklus pengembang sistem : Pendefinisian dari kebutuhankebutuhan fungsional dan persiapan untuk rancang bangun implementasi : “ Mengembarkan bagaimana suatu sistem dibentuk”.
2. Menurut John Buch & Gary Grudnitski perancangan dapat didefinisikan sebagai penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi.
3. Menurut George M. Scott perancangan adalah menentukan bagaimana sistem

akan menyelesaikan apa yang mesti diselesaikan ; tahap ini menyangkut mengkonfigurasi dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem, sehingga setelah instalasi dari sistem akan benar-benar memuaskan rancang bangun yang telah ditetapkan pada akhir tahap analisis sistem. (Kholik Hidayatulloh, et. al. ; 2020)

II.2.2. Aplikasi

Aplikasi merupakan suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer. Terdapat beberapa teori yang mendefinisikan Aplikasi yang dikemukakan oleh beberapa para ahli, diantaranya adalah :

1. Menurut Jack Febrian dalam buku kamus yang berjudul komputer dan teknologi informasi. Aplikasi adalah program siap pakai, program yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain.
2. Menurut Sutarman dalam bukunya yang berjudul pengantar teknologi. Aplikasi merupakan program-program yang dibuat oleh suatu perusahaan komputer untuk para pemakai yang beroperasi dalam bidang umum, seperti pertokoan, komunikasi, penerbangan, perdagangan dan sebagainya.
3. Menurut Hendrayudi dalam bukunya VB. Aplikasi adalah kumpulan perintah program yang dibuat untuk melakukan pekerjaan-pekerjaan tertentu khusus.

Jadi Aplikasi merupakan sebuah program yang di buat dalam sebuah perangkat lunak dengan komputer untuk memudahkan pekerjaan atau tugas-tugas tertentu seperti penerapan, penggunaan dan penambahan data yang dibutuhkan. (Kholik Hidayatulloh, et. al. ; 2020)

II.2.3. Metode Haversine

Haversine Formula adalah persamaan penting dalam sistem navigasi, nantinya Haversine Formula akan menghasilkan jarak terpendek antara dua titik, misalnya pada bola yang diambil dari garis bujur (longitude) dan garis lintang (latitude). Formula ini pertama kali ditemukan oleh Jamez Andrew di tahun 1805, dan digunakan pertama kali oleh Josef de Mendoza y Ríos di tahun 1801. Istilah haversine ini sendiri diciptakan pada tahun 1835 oleh Prof. James Inman. Josef de Mendoza y Ríos menggunakan haversine pertama kali dalam penelitiannya tentang “Masalah Utama Astronomi Nautical“, Proc. Royal Soc, Dec 22. 1796. Haversine digunakan untuk menemukan jarak antar bintang.

Haversine Formula adalah persamaan yang digunakan dalam navigasi, yang memberikan jarak lingkaran besar antara dua titik pada permukaan bola (bumi) berdasarkan bujur dan lintang. Haversine Formula merupakan metode yang digunakan untuk menghitung jarak antara dua titik, berdasarkan posisi garis lintang (latitude) dan posisi garis bujur (longitude) sebagai variabel input-an. Penggunaan rumus ini mengasumsikan pengabaian efek ellipsoidal, cukup akurat

untuk sebagian besar perhitungan, juga pengabaian ketinggian bukit dan kedalaman lembah di permukaan bumi. (Pitrasacha Adytia, et. al. ; 2021)

Berikut adalah rumus Haversine Formula :

$$x = (\text{lon2} - \text{lon1}) * \cos((\text{lat1} + \text{lat2}) / 2);$$

$$y = (\text{lat2} - \text{lat1});$$

$$d = \sqrt{x * x + y * y} * R$$

Keterangan :

x = *Longitude* (Lintang)

y = *Latitude* (Bujur)

d = Jarak

R = Radius Bumi = 6371 km

1 derajat = 0,0174532925

Contoh penggunaan metode Haversine dari CV Ananda Teknindo Jaya ke beberapa lokasi Pelanggan:

1. Jarak CV Ananda Teknindo Jaya ke rumah pelanggan Novi dapat dihitung sebagai berikut :

$$\text{Acos}(\text{Sin}(0.06349) * \text{Sin}(0.062317) + \text{Cos}(0.06349) * \text{Cos}(0.062317) * \text{Cos}(1.721855 - 1.722016)) * 6371 = 7,544653 \text{ Km}$$

2. Jarak CV Ananda Teknindo Jaya ke Toko Laundry Rezeki dapat dihitung sebagai berikut :

$$\text{Acos}(\text{Sin}(0.06349) * \text{Sin}(0,062318) + \text{Cos}(0.06349) * \text{Cos}(0,062318) * \text{Cos}(1,721588 - 1.722016)) * 6371 = 7,951086 \text{ Km}$$

3. Jarak CV Ananda Teknindo Jaya ke Barbershop Ganteng dapat dihitung sebagai berikut :

$$\text{Acos}(\text{Sin}(0.06349) * \text{Sin}(0,063248) + \text{Cos}(0.06349) * \text{Cos}(0,063248) * \text{Cos}(1,720186 - 1.722016)) * 6371 = 11,7409 \text{ Km}$$

II.2.4. PHP (*Hypertext Preprosesor*)

PHP (Hypertext Preprocessor) adalah pemrograman interpreter yaitu proses penerjemahan baris kode sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan. PHP disebut sebagai pemrograman Server Side Programming, hal ini dikarenakan seluruh proses nya dijalankan pada server. PHP adalah sebuah bahasa dengan hak cipta terbuka atau yang juga dikenal dengan istilah open source, yaitu pengguna dapat mengembangkan kode-kode fungsi PHP sesuai dengan kebutuhannya. (Sorang Pakpahan ; 2020)

II.2.5. *Database*

Database merupakan sekumpulan dari data yang terintegrasi. Sistem *Database* merupakan sistem komputerisasi yang bertujuan untuk menyimpan sejumlah data sehingga memudahkan pengguna untuk mendapatkan dan mengupdate informasi sesuai dengan kebutuhan Informasi yang disimpan dalam *Database* dapat berupa *text* maupun angka. Salah satu penggunaan *Database* dalam kehidupan sehari-hari adalah pada aplikasi *software* pembayaran oleh kasir supermarket. Ketika melakukan pembayaran dikasir, kasir tersebut melakukan pekerjaan dengan cara menempelkan *bar-code* di salah satu sensor *infrared*. Kemudian muncul harga dari barang tersebut dalam komputer. Sehingga kasir tersebut tidak perlu mengetik ulang barang yang yang dibeli oleh pembeli tersebut.

Aplikasi pembayaran oleh kasir merupakan salah satu contoh dari penggunaan *Database* yang terintegrasi oleh komputer. Dalam aplikasi komputer tersebut telah terekam data-data barang beserta harga, sehingga kasir dengan mudah mendapatkan data barang serta saat melakukan input barang, maka *Database* tersebut secara otomatis terupdate. Salah satu keuntungan dari *Database* adalah dapat mempersingkat waktu pekerjaan. Sebuah *Database* dapat menampung data yang kompleks yang dapat digunakan secara internal maupun eksternal. *Software* yang digunakan oleh suatu organisasi untuk mengendalikan *Database* disebut dengan *Database Manegement System* atau dapat disingkat dengan DBMS. DBMS merupakan suatu kumpulan program yang memudahkan pengguna untuk membuat dan mengelola *Database*. Data yang ada pada DBMS

dapat dimanipulasi atau dikelola dengan menggunakan *query*, dimana dengan *query* tersebut pengguna dapat mendapatkan data spesifik yang dibutuhkan, serta dapat mengupdate informasi dalam data-data tersebut. (Intan Mardiono ; 2019)

II.2.6. MySQL

MySQL adalah sebuah perangkat lunak yang terdapat didalam sistem manajemen basis data SQL (*database management system*) atau yang biasa disebut DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. Dalam membuatkan mysql yang tersedia di dalam perangkat lunak yang terletak di GPL atau yang biasa di sebut dengan *General Public License*. Dan tetapi mysql ini dapat menjual dibawah komersial dalam kasus pengguna bagi yang tidak sama cocok dengan penggunaan *General Public License*. (Harry Dhika ; 2019)

II.2.7. XAMPP

XAMPP di artikan sebagai *software web apache* yang terdapat *database* mysql dan *php programming*. XAMPP merupakan *software* yang mudah digunakan, gratis dan mendukung instalasi di *linux* dan *windows*. Keuntungannya hanya menginstall sekali yang sudah tersedia di dalam *web apache*, *database* MySQL, dan *PHP programming* (php4 dan php5). Hanya bedanya kalau yang versi untuk *windows* sudah dalam bentuk instalasi grafis dan yang *Linux* dalam

bentuk *file* terkompresi tar.gz. Versi *windows* ini memiliki kelebihan tersendiri antara lain yaitu mempunyai fitur yang dapat mengaktifkan sebuah *server* secara *free*, dan jika di bandingkan dengan *linux* sangat jauh sekali karena *linux* masih berupa perintah-perintah di *console*. Oleh karena itu yang versi untuk *linux* sulit untuk dioperasikan. Dulu XAMPP untuk *linux* dinamakan LAMPP, sekarang berganti nama menjadi XAMPP *for Linux*. (Harry Dhika ; 2019)

II.2.8. Pemrograman Java

Java adalah suatu teknologi di dunia *software* komputer, yang merupakan suatu bahasa pemrograman, dan sekaligus suatu *platform*. Sebagai bahasa pemrograman, Java dikenal sebagai bahasa pemrograman tingkat tinggi. Java mudah dipelajari, terutama bagi programmer yang telah mengenal C/C++. Java merupakan bahasa pemrograman berorientasi objek yang merupakan paradigma pemrograman masa depan. Sebagai bahasa pemrograman Java dirancang menjadi handal dan aman. Java juga dirancang agar dapat dijalankan di semua *platform*. Dan juga dirancang untuk menghasilkan aplikasi-aplikasi dengan performansi yang terbaik, seperti aplikasi database Oracle 8i/9i yang *core*-nya dibangun menggunakan bahasa pemrograman Java.

Java adalah bahasa pemrograman yang populer, dikembangkan oleh *Sun Microsystems*. Salah satu penggunaan terbesar Java adalah dalam pembuatan aplikasi *native* untuk Android. Bahasa pemrograman ini bersifat *multi platform* yakni bahasa ini dapat digunakan di berbagai *platform*, seperti *desktop*, *android*

dan bahkan untuk sistem operasi *Linux*. Sedangkan Java bersifat *neutral architecture*, karena *Java Compiler* yang digunakan untuk mengkompilasi kode program Java dirancang untuk menghasilkan kode yang netral terhadap semua arsitektur perangkat keras yang disebut sebagai *Java Bytecode*. (Ahmad Ibnu Mubarak ; 2021)

II.2.9. Android

Android merupakan sistem operasi berbasis Linux yang digunakan untuk telepon seluler (mobile) seperti telepon pintar (smartphone) dan komputer tablet (PDA). 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 sendiri yang akan digunakan untuk membantu kegiatan dalam berbagai bidang, sehingga bisa digunakan oleh setiap orang yang ingin menggunakannya pada perangkat mereka. Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer tablet. Android awalnya dikembangkan oleh Android, Inc dengan dukungan finansial Google, yang kemudian membelinya pada tahun 2005. (Suheri, et. al. ; 2021)

Pada tanggal 23 september 2008, sistem operasi Android versi 1.0 resmi diluncurkan. Sekitar sebulan berikutnya, pada tanggal 22 Oktober 2008, smartphone pertama yang menjalankan Android 1.0 itu, yaitu HTC Dream,

diluncurkan ke pasar. Pada tanggal 9 februari, Android versi 1.1 diluncurkan untuk memperbaiki bug dari versi sebelumnya dan menambah fitur yang tersedia. Setelah versi 1.1, rilis Android berikutnya menggunakan nama makanan manis dengan urutan alfabetis, dimulai dengan 1.5 Cupcake yang diluncurkan pada tanggal 30 April 2009. Rilis-rilis Android selanjutnya, yaitu Donut, clair, Froyo, dan Gingerbread semua dibuat untuk smartphone. Namun, Apple meluncurkan iPad pada tahun 2010 dan meningkatkan ketertarikan masyarakat luas kepada computer tablet. Beberapa pengembang Android mencoba mengembangkan tablet Android untuk menyaingi iPad, seperti Samsung Galaxy Tab yang menggunakan Gingerbread yang dikustomisasi. Google dan OHA pun bergerak dengan melakukan pengembangan Android versi baru yang lebih optimal untuk tablet.

Pada tanggal 22 februari 2011, android Honeycomb diluncurkan ke pasar dan pada tanggal 24 februari 2011, tablet pertama yang menggunakan honeycomb, yaitu Motorola Xoom, diluncurkan ke pasar. Pada tanggal 19 Oktober 2011, Android meluncurkan Ice Cream Sandwich versi ini dapat bekerja secara optimal baik di smartphone maupun di tablet. Rilis android berikutnya, yaitu Jelly Bean, bertujuan untuk semakin meningkatkan apa yang sudah tersedia di Ice Cream Sandwich, dengan memperbaiki bug-bug dn menambahkan fitur-fitur. Pada tanggal 3 september 2013, diumumkan versi Android selanjutnya adalah Android 4.4 Kit Kat. Android sudah mendapatkan izin dari Nastle dan Hershey selaku pemilik dagang Kit Kat. Sebelum pengumuman ini, banyak yang berspekulasi bahwa versi Android berikutnya akan diberi nomor 5.0 dengan nama Key Lime Pie brikut adalah tabel untuk semua sistem operasi Android yang sudah

diluncurkan sampai sekarang. Saat buku ini ditulis, sistem operasi Android yang terbaru adalah Android 6.0 Marshmallow. (Yusfrizal ; 2019)

II.2.10. Android Studio

Android studio merupakan lingkungan pengembangan perangkat lunak terpadu Integrated Development Environment (IDE) untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA. Selain merupakan editor kode IntelliJ dan alat pengembang yang berdaya guna, Android Studio juga menawarkan banyak fitur untuk meningkatkan produktivitas saat membuat aplikasi Android. Android studio sendiri dikembangkan berdasarkan IntelliJ IDEA yang mirip dengan Eclipse disertai dengan ADT plugin (Android Development Tools). Android Studio memiliki fitur :

1. Projek berbasis pada Gradle Build
2. Refactory dan pembenahan bug yang cepat
3. Tools baru yang bernama “Lint” diklaim dapat memonitor kecepatan, kegunaan, serta kompetibilitas aplikasi dengan cepat.
4. Mendukung Proguard And App-signing untuk keamanan.
5. Memiliki GUI aplikasi android lebih mudah
6. Didukung oleh Google Cloud Platfrom untuk setiap aplikasi yang dikembangkan. (Suheri, et. al. ; 2021)

II.2.11. Android SDK (*Software Development Kit*)

Android SDK mencakup perangkat tools pengembangan yang komprehensif. android SDK terdiri dari debugger, libraries, handset emulator, dokumentasi, dengan menggunakan plugin Android Development Tools (ADT), dengan ini pengembang dapat menggunakan ide untuk mengedit dokumen java dan XML serta menggunakan peralatan command line untuk menciptakan, membangun, melakukan debug aplikasi android dan pengendalian perangkat android. (Dicky Taruna, et. al. ; 2021)

II.2.12. Google Maps

Google maps adalah jasa peta gratis dan online yang disediakan oleh google yang dapat ditemukan di . Pada situs tersebut kita dapat melihat informasi geografis pada hampir semua wilayah di muka bumi. Layanan google ini interaktif, karena didalamnya peta dapat digeser sesuai keinginan pengguna, mengubah tingkat zoom, serta mengubah tampilan peta. Google juga menawarkan peta yang dapat digeser dan gambar satelit untuk seluruh dunia, serta menawarkan rute perjalanan (Sitti Aisa : 2021)

II.2.13. Pengertian UML

Unified Modelling Language (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem.

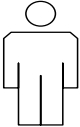
Unified Modelling Language (UML) adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan *software* berbasis OO (*Object-Oriented*). UML sendiri juga memberikan standar penulisan sebuah sistem *blue print*, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema *database*, dan komponen-komponen yang diperlukan dalam sistem. (Azmi Zainul Arif ; 2020) Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

II.2.13.1. Use Case Diagram

Use Case Diagram merupakan pemodelan untuk kelakuan sistem informasi yang akan dibangun. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibangun. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada pada sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

Adapun simbol *use case diagram* terdapat pada Tabel.II.1.

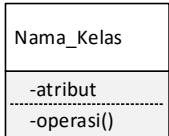
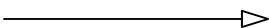
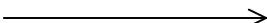
Tabel II.1 Simbol Use Case Diagram

Simbol	Diskripsi
<i>Use Case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor: biasanya dinyatakan dengan menggunakan kata kerja diawal di awal frase nama use case.
Aktor/actor 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.
Asosiasi /association 	Komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan actor.
Ekstensi / extend <<extends>	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu; mirip dengan prinsip inheritance pada pemrograman berorientasi objek biasanya use case tambahan memiliki nama depan yang sama dengan use case yang ditambahkan misal Arah panah mengarah pada use case yang ditambahkan.
Generalisasi / generalization 	Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
Menggunakan / include 	Include berarti use case yang tambahan akan selalu melakukan pengecekan apakah use case yang ditambahkan telah dijalankan sebelum use case tambahan dijalankan

II.2.13.2 Class Diagram

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. Adapun simbol *class diagram* terdapat pada Tabel.II.2

Tabel II.2. Simbol Class Diagram

Simbol	Diskripsi
Kelas 	Kelas pada struktur sistem.
Antar muka/interface  nama_interface	Sama dengan konsep interface dalam pemrograman berorientasi objek.
Asosiasi / association 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity.
Asosiasi berarah / directed association 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity.
Generalisasi 	Relasi antar kelas dengan makna generalisasi-generalisasi- spesialisasi (umum khusus).
Kebergantungan / dependency 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi /aggregation	Relasi antar kelas dengan makna semua-

	bagian (whole- part).
---	-----------------------

(Sumber : Julianto Simatupang, 2019)

II.2.13.3 Activity Diagram

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Penekanan pada diagram aktivitas adalah menggambarkan aktivitas sistem atau aktivitas yang dapat dilakukan oleh sistem, bukan apa yang dilakukan actor. Adapun simbol *activity diagram* terdapat pada Tabel.II.3.

Tabel II.3. Simbol Activity Diagram

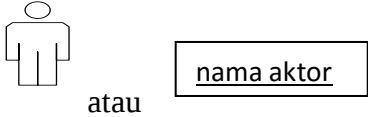
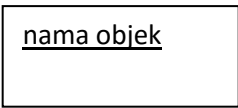
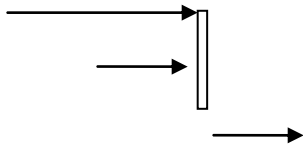
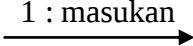
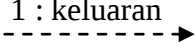
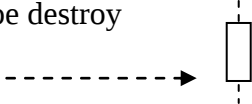
Simbol	Diskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan / decision 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari Satu
Penggabungan / Join 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu..
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

(Sumber : Julianto Simatupang, 2019)

II.2.13.4. Sequence Diagram

Sequence Diagram menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Adapun simbol *sequence diagram* terdapat pada Tabel.II.4.

Tabel II.4. Simbol *Sequence Diagram*

Simbol	Diskripsi
Aktor 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.
Garis hidup / lifeline 	Menyatakan kehidupan suatu objek.
Objek 	Menyatakan objek yang berinteraksi pesan.
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya
Pesan tipe create 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat. arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.
Pesan tipe send 	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
Pesan tipe return 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
Pesan tipe destroy 	Menyatakan suatu objek mengakhiri hidup objek lain, arah panah yang mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada

	destroy.
--	----------

(Sumber : Julianto Simatupang, 2019)