

BAB II

TINJAUAN PUSTAKA

II.1 Penelitian Terkait

Untuk menunjang keberhasilan penelitian ini, penulis melakukan penelitian teoritis melalui beberapa literatur yang berkaitan dengan penelitian yang dilakukan. Adapun penelitian yang akan digunakan sebagai sumber acuan yang relevan dan terkini yaitu :

Berdasarkan penelitian yang dilakukan oleh Erik Kurnadi dan Heru Budianto (2018) dengan judul “Rancang Bangun Aplikasi Wisata Kabupaten Kuningan Berbasis Android Menggunakan Metode *Location Based Service* (LBS)”. Penelitian ini menawarkan sebuah aplikasi android yang dapat memberikan kemudahan kepada wisatawan yang ingin mengunjungi obyek wisata Kabupaten Kuningan. Aplikasi ini memberikan informasi tentang objek wisata yang ada, antara lain informasi lokasi, fasilitas, waktu buka dan tutup, serta panduan untuk mencapai lokasi dengan menggunakan metode layanan berbasis lokasi. Pengguna aplikasi dapat memilih objek wisata berdasarkan kategori, dan sistem akan memberikan fasilitas panduan tempat wisata yang ingin dikunjungi. Aplikasi ini diharapkan dapat mengatasi permasalahan di atas dan menjadi media promosi pariwisata di Kuningan. (Erik Kurnadi dan Heru Budianto, 2018)

Berdasarkan penelitian yang dilakukan oleh Obie Jagad Prakoso, Adi Muhajirin, Dwi Budi Srisulistiowati (2018) dengan judul “Sistem Pemesanan Jasa Perbaikan Komputer Dengan *Location Based Service* (LBS) Berbasis Android”.

Penelitian ini bertujuan untuk memudahkan masyarakat dalam mencari penyedia jasa reparasi komputer dan menyediakan tempat bagi yang memiliki kemampuan untuk memperbaiki komputer, tetapi tidak ada toko. Karena banyak masyarakat yang ingin memperbaiki komputer, selalu mencari toko komputer yang menyediakan jasa perbaikan komputer. Dikarenakan belum adanya fasilitas yang menampung penyedia jasa perbaikan komputer baik itu perorangan yang tidak memiliki toko, maupun yang telah memiliki toko sendiri agar masyarakat tidak perlu repot mencari toko yang menyediakan jasa perbaikan komputer. (Prakoso et al., 2018)

Berdasarkan penelitian yang dilakukan oleh Edwient Fernando (2019) dengan judul “Aplikasi Layanan Jahit Pakaian Kota Pekanbaru Berbasis *Location Based Service*.” Tujuan Penelitian ini adalah menghasilkan aplikasi yang dapat menampilkan lokasi penjahit di wilayah Pekanbaru dan memberi informasi dari layanan penjahit tersebut. Penelitian ini menggunakan data dari lima penjahit yang berada di pusat kota Pekanbaru, kecamatan Bukit Raya, dan layanan berbasis lokasi yang digunakan adalah *Google Maps*. Hasil dari penelitian ini adalah aplikasi jasa jahit pakaian. Berdasarkan uji kelayakan sistem digunakan metode survei kuesioner untuk menguji 20 responden. Hasil rata-rata pilihan “sangat baik” dan “baik” adalah 93%. Oleh karena itu aplikasi layak untuk diterapkan. (Edwint Fernando, 2019).

Berdasarkan penelitian yang dilakukan oleh Bernard Shibwabo Kasamani, Denis Gikundi (2017) dengan judul “A *Location-Based Service for Handyman Order Placement*.” Penelitian ini bertujuan untuk menemukan layanan tukang di

dalam lokasi pengguna dan memiliki fitur peringkat untuk peringkat layanan tukang untuk memastikan kualitas layanan oleh tukang. Aplikasi ini dikembangkan dalam sistem operasi android karena popularitasnya di antara banyak pengguna ponsel. (Bernard Shibwabo Kasamani, Denis Gikundi, 2017)

II.2 Landasan Teori

II.2.1 Perancangan

Perancangan adalah Proses untuk menentukan sesuatu yang akan dikerjakan dengan menggunakan teknik yang bervariasi serta melibatkan deskripsi arsitektur dan detail komponen serta batasan yang akan dihadapi dalam proses pengerjaannya (Adiguna, 2018).

Dari pengertian di atas dapat disimpulkan bahwa perancangan merupakan sebuah proses perencanaan yang bervariasi, pembuatan sistem baru atau pengembangan dari sistem yang sudah ada sebelumnya.

II.2.2 Aplikasi

Dalam ilmu komputer, aplikasi adalah perangkat lunak atau program komputer yang berjalan pada sistem tertentu yang dibuat dan dikembangkan untuk menjalankan perintah tertentu. Istilah aplikasi sendiri diambil dari bahasa Inggris “*application*” dan dapat diartikan sebagai aplikasi atau penggunaan. Secara harfiah, aplikasi adalah perangkat lunak yang dikembangkan untuk melakukan tugas-tugas tertentu. Aplikasi adalah perangkat lunak yang dirancang khusus untuk membantu melakukan tugas yang diperlukan dan program yang siap pakai. (Kom, K. M. 2018).

II.2.3 Penjahit

Menurut KBBI (Kamus Besar Bahasa Indonesia) penjahit adalah orang yang mata pencahariannya menjahit pakaian, tas, dan sebagainya. Dapat ditarik kesimpulan bahwa penjahit adalah sebuah profesi di bidang jasa yang memberikan pelayanan berupa jasa menjahit.

II.2.4 *Location Based Service (LBS)*

Menurut (Susanty dan Astari, 2019) *Location Based Service (LBS)* atau layanan berbasis lokasi merupakan layanan informasi yang di akses memakai perangkat mobile lewat jaringan internet serta seluler dan menggunakan kemampuan penanda lokasi pada perangkat mobile. *Location Based Service* dapat berfungsi sebagai layanan untuk mengidentifikasi lokasi dari seseorang atau suatu objek tertentu, seperti menemukan lokasi SPBU terdekat ataupun lokasi yang lain. *Location Based Service (LBS)* atau layanan berbasis lokasi ini diterapkan dalam sebuah platform android. Android merupakan sistem operasi bergerak (*mobile operating system*) yang mengadopsi sistem operasi *Linux*, tetapi sudah dimodifikasi”. Tidak hanya itu, *Location Based Service* tersebut terdiri dari beberapa komponen di antaranya *mobile devices*, *communication network*, *position component*, dan *service and content provider*.

Sederhananya, dengan layanan LBS kita bisa mengetahui lokasi kita, lokasi teman kita, dan lokasi rumah sakit atau pom bensin di dekat kita. Saat mengukur lokasi, lintang dan bujur digunakan untuk menentukan lokasi geografis. Namun, Android menyediakan *geocoder* yang mendukung *geocoding* maju dan mundur. Menggunakan *geocoder* kita dapat mengubah nilai lintang dan bujur

menjadi alamat dunia nyata atau sebaliknya. Dalam teknologi LBS berbasis jaringan seluler, posisi perangkat komunikasi seluler ditentukan oleh posisi relatifnya terhadap lokasi BTS (*Base Transceiver Station*).

Seperti digambarkan pada Gambar II.1, *Location Based Service* (LBS) terdiri dari 4 komponen, diantaranya:

a. Perangkat Seluler (*Mobile Device*)

Perangkat seluler merupakan komponen penting dari LBS. Alat ini berfungsi sebagai alat bagi pengguna untuk meminta informasi. Hasil dari informasi yang diminta dapat berupa teks, suara, gambar, dll. Perangkat *mobile* yang dapat digunakan dapat berupa PDA, *smartphone*, atau laptop. Perangkat seluler juga dapat digunakan sebagai alat navigasi berbasis GPS serta alat navigasi di dalam kendaraan.

b. Jaringan Komunikasi (*Communication Network*)

Komponen ini berfungsi sebagai penghubung yang dapat mengirim data yang dikirim oleh pengguna dari perangkat seluler mereka ke penyedia layanan, setelah itu hasil permintaan diterima oleh penyedia layanan dan dikirim kembali ke pengguna.

c. Komponen *Positioning* (Penunjuk Posisi)

Setiap layanan yang disediakan oleh penyedia layanan umumnya akan didasarkan pada lokasi pengguna yang meminta layanan. Oleh karena itu, diperlukan komponen yang berfungsi sebagai pemroses untuk menemukan pengguna layanan saat itu. Lokasi pengguna dapat diperoleh melalui jaringan komunikasi seluler atau juga dengan menggunakan *Global Positioning System* (GPS).

d. Penyedia layanan dan konten (*Service and content Provider*)

Penyedia layanan menyediakan sejumlah layanan berbeda kepada pengguna dan bertanggung jawab untuk memproses permintaan layanan. Contoh layanannya adalah perhitungan lokasi, pencarian rute, dan lain-lain.

Location Based Service (LBS) memiliki unsur utama yaitu :

1. *Location (API Map)* menyediakan perangkat bagi sumber atau *source* untuk *location based service (LBS)*, *Application Programming Interface (API)* map menyediakan fasilitas untuk menampilkan dan memanipulasi peta.
2. *Location Provider (API Location)* menyediakan teknologi pencarian lokasi yang digunakan oleh perangkat. *API Location* berhubungan dengan data GPS (*Global Positioning System*) dan data lokasi *real-time*. *API Location* berada pada data android yaitu data paket internet yang digunakan oleh perangkat.



Gambar II.1 Komponen Pendukung *Location Based Service (LBS)*

(Sumber : *Steiniger et al. (2006)*)

II.2.5 GPS (*Global Positioning System*)

GPS adalah singkatan dari *Global Positioning System*, yang merupakan sistem navigasi dengan menggunakan teknologi satelit yang dapat menerima sinyal dari satelit. Sistem ini menggunakan 24 satelit yang mengirimkan sinyal gelombang mikro ke bumi. Sinyal ini diterima oleh alat penerima (*receiver*) di permukaan, dimana GPS *receiver* ini akan mengumpulkan informasi dari satelit GPS, seperti:

a. Waktu.

Penerima GPS menerima informasi waktu dari jam atom dengan akurasi yang sangat tinggi.

b. Lokasi.

GPS memberikan informasi lokasi dalam tiga dimensi:

1) *Latitude* (garis lintang)

2) *Longitude* (garis bujur)

3) *Elevasi* (ketinggian suatu tempat terhadap daerah sekitarnya)

c. Kecepatan.

Saat bergerak, GPS dapat menampilkan informasi tentang kecepatan perjalanan.

d. Arah perjalanan.

GPS dapat menunjukkan arah tujuan.

e. Simpan lokasi.

Penerima GPS dapat merekam tempat yang pernah atau ingin Anda kunjungi.

f. Komulasi data.

GPS dapat menyimpan informasi pelacakan, seperti jumlah total perjalanan yang dilakukan, kecepatan rata-rata, kecepatan tertinggi, kecepatan terendah, waktu/jam ke tujuan, dll.

II.2.6 MapBox

Mapbox adalah salah satu penyedia peta kustom terbesar di situs-situs ternama seperti *Foursquare*, *Pinterest*, *Evernote* sejak pada tahun 2010, *Mapbox* memperbanyak pilihan peta kustomnya untuk mengisi keterbatasan yang dimiliki penyedia peta seperti *google maps*. *Mapbox* merupakan pencipta atau kontributor sejumlah pustaka dan aplikasi peta bebas terkenal, misalnya spesifikasi *MBTiles*, *Kartografi TileMill IDE*, pustaka *JavaScript Leaflet*, Bahasa gaya dan parser peta *CartoCSS*, dan pustaka *Java Script Mapbox.js*. Data *Mapbox* diambil dari sumber-sumber data terbuka seperti *Open Street Map* dan *NAS*, dan sumber-sumber data berbayar seperti *Digital Globe*. Teknologinya dibangun menggunakan *Node.js*, *CouchDB*, *Mapnik*, *GDAL* dan *Leafletjs*. *MapBox*, yaitu dukungan terhadap *GeoJSON encoding* obyek geografis dalam bentuk *JSON*. *Mapbox* juga merupakan turunan dari *Open Street Map*. (Aryando Giovanni Rumondor; 2019; 262)

II.2.7 Firebase

Firebase adalah *BaaS (Backend as a Service)* yang saat ini dimiliki oleh Google. *Firebase* sendiri merupakan solusi yang ditawarkan oleh Google untuk mempermudah pekerjaan *Mobile Apps Developer*. Dengan adanya *Firebase*, *apps developer* tentu saja dapat fokus mengembangkan aplikasi tanpa harus memberikan effort yang besar untuk urusan *backend Firebase Realtime Database*

sendiri merupakan basis data berbasis *cloud NoSQL* yang menyinkronkan data di semua klien secara realtime, dan menyediakan fungsionalitas *offline* Data yang diinputkan disimpan ke dalam *database Realtime* sebagai *JSON*. Semua klien yang terhubung dan saling berbagi dalam satu waktu, secara otomatis akan menerima pembaruan dengan data terbaru (Kurniawati, dan Lukman Bachtiar, 2020).

II.2.8 Android

Android adalah sistem operasi berbasis *Linux* untuk telepon seluler. Android menyediakan *platform* terbuka bagi pengembang untuk membangun aplikasi mereka sendiri. Awalnya, *Google Inc.* membeli *android Inc.* perusahaan baru yang membuat perangkat lunak untuk telepon seluler. Kemudian untuk mengembangkan *Android*, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk *Google*, *HTC*, *Intel*, *Motorola*, *Qualcomm*, *T-Mobile*, dan *Nvidia*. Pada Juli 2005 Android telah diakuisisi oleh *Google* dan pada 5 November 2007 barulah secara resmi Android dirilis oleh *Google*.

Saat mengembangkan aplikasi, android menyediakan android SDK yang menyediakan alat dan API untuk pengembang aplikasi dengan *platform* Android.

a. Keunggulan Android

Beberapa keunggulan Android dengan *platform* lain yaitu sebagai berikut:

1. Keterbukaan Android-Android menyediakan akses ke fungsi dasar perangkat *Mobile* menggunakan standar panggilan ke API.

2. Pengguna dapat menggabungkan informasi dari Internet ke dalam telepon, seperti informasi kontak, atau data pada lokasi geografis.
3. Cepat dan mudah perkembangannya.

II.2.9 Generasi Android

a. *Android 1.0 & 1.1 : Astro (Alpha) & Bender (Beta)*

Versi Android 1.0 *Astro (Alpha)* dan Android 1.1 *Bender (Beta)* ini belum dirilis untuk penggunaan komersial. Platform Android sendiri pertama kali diluncurkan pada September 2008 oleh Andy Rubin yang kini dikenal sebagai bapak Android. Meski tidak menggunakan nama makanan manis, kedua sistem operasi Android ini jelas menjadi yang terdepan. Karena di situlah Android dimulai dengan *smartphone* pertama, *HTC Dream*.

b. *Android versi 1.5 (Cupcake)*

Mulai dari versi ini dan seterusnya, Android akan menggunakan nama makanan manis untuk setiap versi yang diluncurkannya. Android 1.5 *Cupcake* sendiri dirilis pada 30 April 2009, dan memiliki berbagai fungsi *smartphone* untuk menggantikan *smartphone* yang diperkenalkan saat itu.

c. *Android versi 1.6 (Donut)*

Tentunya pada awal perilisannya, sistem operasi android masih memiliki banyak *bug* yang perlu diperbaiki oleh *developer*. Hal ini dilakukan pada android 1.6 *Donut* yang dirilis pada 15 September 2009. Nah, itu artinya kurang dari setahun sejak Android 1.5 *Cupcake* dirilis atau hanya berselang 5 bulan. Android juga telah menambahkan sejumlah pembaruan, terutama dukungan untuk layar *smartphone* yang lebih besar.

d. *Android* versi 2.0/2.1 (*Eclair*)

Seperti versi sebelumnya, *Android* 2.0 dan 2.1 *Eclair* masih bekerja untuk memperbaiki *bug* yang muncul di sistem operasi *mobile* ini. *Android* juga menambahkan berbagai fitur untuk ini. Dari dukungan *Bluetooth* hingga fitur kamera yang menarik bagi *smartphone* saat itu. *Android* 2.0 dan 2.1 *Eclair* digunakan pada perangkat seperti *HTC Nexus One*.

e. *Android* versi 2.2 (*Froyo*)

Sejak versi ini, *Android* sepertinya sudah dikenal luas oleh berbagai merek *smartphone*. *Android* 2.2 *Frozen Yogurt*, juga dikenal sebagai *Froyo*, pertama kali dirilis pada 20 Mei 2010. Meski digunakan oleh beberapa merek, *Android* masih kalah bersaing dengan *Symbian* yang mendominasi pasar utama ponsel. *Android* 2.2 *Froyo* menawarkan peningkatan kecepatan, *tethering* USB, *hotspot* Wi-Fi, dan fitur keamanan.

f. *Android* versi 2.3 (*Gingerbread*)

Tidak sampai setahun kemudian, *Android* 2.3 *Gingerbread* diluncurkan kembali pada Desember 2010 dengan banyak peningkatan yang signifikan. Hal ini terutama terjadi pada tampilan tatap muka alias antarmuka pengguna yang digunakan. Dari versi ini, banyak merek *smartphone* mulai mempertimbangkan untuk menggunakan sistem operasi *Android*. Salah satunya adalah seri *Samsung Galaxy* yang selalu populer.

g. *Android* versi 3.0/3.1 (*Honeycomb*)

Bagi pengguna *smartphone*, versi *Android* ini mungkin sedikit asing. *Android* 3.0 & 3.2 *Honeycomb* menggunakan ikon lebah yang dirancang

khusus untuk digunakan dengan tablet. Tentu saja, rilis Android 3.0 & 3.2 *Honeycomb* pada 10 Mei 2011 dimaksudkan untuk mendukung *Samsung* yang sudah mulai merilis tablet *Samsung Galaxy Tab Series* untuk bersaing dengan *iPad Apple*.

h. *Android* versi 4.0 (*Ice Cream Sandwich*)

Selain itu, *Android* juga telah merilis versi *Android 4.0 Ice Cream Sandwich* yang lagi-lagi ditujukan untuk *smartphone*. *Android 4.0 Ice Cream Sandwich* sendiri dirilis pada 19 Oktober 2011. Sebagai versi dengan nama terpanjang hingga saat ini, *Android 4.0 Ice Cream Sandwich* menawarkan banyak pembaruan. Dari animasi menjadi lebih halus, sederhana dan mudah digunakan.

i. *Android* versi 4.1 (*Jelly Bean*)

Peningkatan yang signifikan terasa saat menggunakan *Android 4.1* dan *4.3 Jelly Bean*. Sistem operasinya sendiri pertama kali dirilis pada Juni 2012 dengan sejumlah perbaikan, terutama di bidang pemrosesan grafis. Tentunya dengan cara ini, *Android 4.1* dan *4.3 Jelly Bean* dapat memberikan fungsionalitas lebih pada antarmuka pengguna dan teknologi *Vsync* yang digunakannya.

j. *Android* versi 4.4 (*KitKat*)

Menggunakan nama merek makanan cepat saji terkenal, *Android 4.4 KitKat* pertama kali dirilis pada Oktober 2013. Versi *Android* terbaik ini juga dapat dianggap sebagai favorit karena mendukung sebagian besar *Android*, sebagian besar *smartphone* di dunia. Hal ini dikarenakan

Android 4.4 *KitKat* mampu memberikan optimasi yang baik, bahkan pada *smartphone* dengan spesifikasi yang *inferior* atau agak rendah saat itu.

k. Android 5.0 & 5.1: *Lollipop*

Setelah beberapa versi, Android dan *Google* juga mulai memperbarui sistem operasi mereka secara teratur sepanjang tahun. Termasuk Android 5.0 dan 5.1 *Lollipop* yang dirilis dan dirilis pada Juni 2014. Bisa dikatakan Android 5.0 dan 5.1 *Lollipop* mempelopori *smartphone flagship* dengan performa yang sangat baik. Versi Android ini sudah mendukung arsitektur 64-bit yang memungkinkan RAM lebih dari 3 GB. Salah satunya adalah *ASUS Zenfone 2* yang saat itu sudah memiliki RAM 4GB.

l. Android 6.0 : *Marshmallow*

Android 6.0 *Marshmallow* merupakan penerus versi Android sebelumnya. Sistem operasinya sendiri pertama kali diperkenalkan pada Mei 2015 dan dirilis pada Oktober 2015. Sistem operasi ini menggunakan sensor sidik jari sebagai sistem keamanan biometrik untuk memberikan peningkatan keamanan yang jelas. Selain kunci layar, sensor sidik jari ini dapat digunakan untuk autentikasi *Google Play Store* dan pembelian *Android Pay*.

m. Android 7.0 & 7.1 : *Nougat*

Sistem operasi Android saat ini masih digunakan di beberapa *smartphone* yang baru dirilis. Android 7.0 dan 7.1 *Nougat* pertama kali diperkenalkan pada Juni 2016 yang menampilkan robot Android dengan garis-garis *Nougat*. Antarmuka telah berubah di sistem operasi Android 7.0 dan 7.1

Nougat. Ini juga memiliki fitur layar terpisah yang memungkinkan Anda membagi layar untuk dua aplikasi sekaligus.

n. *Android 8.0 & 8.1 : Oreo*

Android 8.0 dan 8.1 *Oreo* adalah sistem operasi Android terbaru dan terpopuler saat ini. Sistem operasi ini telah dirilis sejak Agustus 2017 dan telah diperbarui ke Android 8.1 *Oreo*. Android 8.0 dan 8.1 *Oreo* adalah versi Android kedua yang menggunakan permen merek terkenal setelah Android 4.4 *KitKat*. Sistem operasi ini menawarkan lebih banyak fitur multitasking dibandingkan versi sebelumnya. Ada juga Project *Treble*, yang memungkinkan pengguna menerima pembaruan lebih cepat.

o. *Android 9.0 : Pie*

Lalu ada Android 9.0 *Pie* yang resmi dirilis pada Agustus 2018 lalu. Sistem operasi Android ini mengandung banyak perubahan, terutama untuk desain baru HP. Misalnya, Android 9.0 *Pie* menawarkan navigasi gerakan yang menggantikan tombol *home*, *back*, dan *recent apps* fisik. Fitur lain yang sangat berguna adalah sistem notifikasi, kontrol kecerahan, dan sistem tangkapan layar modern yang membuat pekerjaan Anda lebih mudah.

p. *Android Q Beta*

Terakhir, ada Android Q *beta* yang baru dirilis pada 13 Maret 2019, dan saat ini masih terbatas pada perangkat HP Android tertentu. Seperti seri *smartphone* Google, ada *Google Pixel*, *Google Pixel XL*, *Google Pixel 2*, *Google Pixel 2 XL*, *Google Pixel 3*, *Google Pixel 3 XL* dan *Google Pixel 3*

Lite. Salah satu fitur Android Q *Beta* adalah Mode Gelap (juga dikenal sebagai Mode Gelap), yang diklaim dapat meningkatkan kinerja baterai. (Satriya Aji Purboko, 2021).

II.2.10 *Unified Modeling Language (UML)*

UML adalah bahasa untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan artefak (sepotong informasi yang digunakan untuk dihasilkan oleh proses produksi perangkat lunak, artefak ini dapat berupa model, deskripsi, atau perangkat lunak) dari suatu sistem perangkat lunak, seperti pada perusahaan pemodelan dan sistem lain selain peralatan, perangkat lunak (John, et al, 2017).

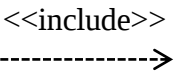
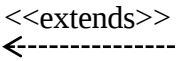
UML (*Unified Modeling Language*) adalah metode pemodelan visual untuk alat desain sistem berorientasi objek, di mana definisi UML adalah bahasa yang telah menjadi standar untuk visualisasi desain, dan dokumentasi sistem perangkat lunak. (Sutiyono, S.T., M.Kom, 2020).

Berikut ini adalah bagian-bagian diagram UML :

1. *Use Case Diagram*

Use Case Diagram merupakan pemodelan untuk melakukan (*behavior*) sistem informasi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktordenga sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk menentukan fitur-fitur yang terdapat dalam suatu sistem informasi dan siapa yang berwenang untuk menggunakan fitur-fitur tersebut. Simbol yang digunakan dalam *use case diagram* adalah sebagai berikut :

Tabel II.1 Simbol Use Case

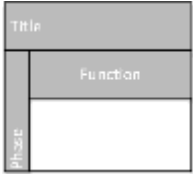
Gambar	Keterangan
	<i>Use case</i> menggambarkan fitur-fitur yang disediakan oleh sistem. Sebuah unit untuk bertukar pesan antara aktor dan unit diungkapkan menggunakan kata kerja.
	<i>Actor</i> atau Aktor adalah abstraksi dari orang lain atau sistem lain yang mengaktifkan fungsionalitas sistem target. Untuk mengidentifikasi aktor, Anda perlu mendefinisikan pembagian kerja dan tugas terkait peran dalam konteks sistem target. Seseorang atau sistem dapat muncul dalam beberapa peran. Perlu diingat bahwa aktor berinteraksi dengan <i>use case</i> . Namun, tidak memiliki kendali atas <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 data.
	<i>Generalization</i> : menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i> .
	<i>Include</i> , Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat

(Sumber : Yunahar Heriyanto, 2018)

2. Diagram Aktivitas (*Activity Diagram*)

Activity diagram atau diagram aktivitas menggambarkan suatu alur kerja (workflow) atau sistem atau aktivitas menu dalam perangkat lunak. Perlu dicatat di sini bahwa diagram tindakan mewakili tindakan sistem, yang mewakili tindakan yang dapat dilakukan sistem, bukan tindakan yang dilakukan oleh aktor.

Tabel II.2 Simbol Activity Diagram

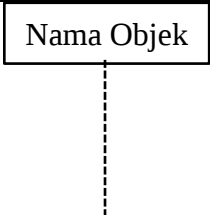
Simbol	Nama	Keterangan
	Status Awal	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan / <i>Decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
	Penggabungan / <i>Join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
	<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

(Sumber : Yunahar Heriyanto, 2018)

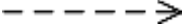
3. *Sequence Diagram*

Sequence diagram adalah model untuk menggambarkan perilaku suatu objek dalam *use case* dengan menggambarkan durasi objek dan pesan yang dikirim dan diterima antara objek.

Tabel II.3 Simbol *Sequence Diagram*

Simbol	Nama	Keterangan
	Aktor / Actor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi akan dibuat di luar sistem informasi yang akan dihasilkan oleh sistem itu sendiri, jadi meskipun simbol aktor adalah gambar orang, tapi aktor belum tentu orang, biasanya diekspresikan menggunakan kata benda di awal kalimat dengan nama aktor.
	<i>LifeLine</i> / garis hidup objek	<i>Lifeline</i> mengindikasikan keberadaan sebuah object dalam basis waktu. Notasi untuk <i>Lifeline</i> adalah garis putus-putus

		vertikal yang ditarik dari sebuah object.
Nama Objek	Objek	Menyatakan objek yang berinteraksi pesan.
	Waktu aktif	Mendeklarasikan suatu objek dalam keadaan aktif dan interaktif adalah langkah dimana segala sesuatu yang berhubungan dengan waktu aktif ini dilakukan pada objek tersebut.
	Pesan Tipe <i>Create</i>	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
1 : nama metode() 	Pesan Tipe <i>Call</i>	Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri.

1:masukan 	Pesan tipe <i>Send</i>	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
1:keluaran 	Pesan tipe <i>Return</i>	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan sesuatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.

(Sumber : Yunahar Heriyanto, 2018)

4. *Class Diagram*

Class diagram atau diagram kelas menggambarkan struktur sistem dalam mendefinisikan kelas-kelas yang akan dibangun untuk membangun sistem. Kelas memiliki apa yang disebut properti dan metode atau operasi.

Tabel II.4 Simbol *Class Diagram*

Simbol	Keterangan
	<i>Class</i> , merupakan himpunan dari objek-objek yang berbagi atribut serta operasi yang sama
	<i>Interface</i> atau antar muka, Sama dengan konsep interface dalam pemrograman berorientasi objek.
	<i>Association</i> , Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
	<i>Directed Association</i> , Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
	<i>Generalisasi</i> , Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).
	Kebergantungan / <i>Dependency</i> , Relasi antarkelas dengan makna kebergantungan
	Agregasi/ <i>Aggregation</i> , Relasi antar kelas dengan makna semua bagian (whole part).

(Sumber : Yunahar Heriyanto, 2018)