

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Dalam pembuatan aplikasi sosialisasi pencegahan penyakit DBD ada beberapa penelitian yang terkait untuk selanjutnya sebagai dasar dari penelitian sebagai berikut;

Penelitian oleh Lutfi (2017) yang berjudul Aplikasi *Historoid* Berbasis *Android* Sebagai Media Pembelajaran Sejarah Siswa SMA. Aplikasi *Historoid* berbasis *Android* merupakan aplikasi media pembelajaran sejarah SMA yang menampilkan lokasi cagar budaya di Daerah Istimewa Yogyakarta, berisikan deskripsi, gambar dan video. Metode yang digunakan adalah *Research and Development* dengan prosedur pengembangan perangkat lunak *waterfall*. Pengembangan aplikasi *Historoid* menggunakan *Android Studio*. Aplikasi *Historoid* beroperasi dengan *Android minimum 4.1 (JellyBean)*.

Penelitian yang kedua dilakukan oleh Megawati (2016) yang berjudul Rancang Bangun Aplikasi Pengenalan dan Simulasi Permainan Tradisional Nusantara Berbasis *Android*. Menghasilkan aplikasi pengenalan dan simulasi permainan tradisional nusantara berbasis *android*. Aplikasi pengenalan dan simulasi permainan tradisional nusantara merupakan aplikasi media informasi yang memperkenalkan tentang permainan tradisional. Metode yang digunakan dalam penelitian ini adalah penelitian kualitatif dimana strategi yang digunakan adalah *design and creation*. Penelitian ini menggunakan metode pengumpulan data observasi dan studi Pustaka. Metode perancangan atau pengembangan sistem menggunakan *waterfall*, pemodelan menggunakan UML, *flowchart*, dan perancangan antarmuka. Teknik pengujian yang digunakan adalah *white box*, *black box* dan kuesioner. *Software Unity 3D* digunakan untuk membangun aplikasi serta *Adobe Photoshop* digunakan untuk mengedit foto sebagai isi konten aplikasi.

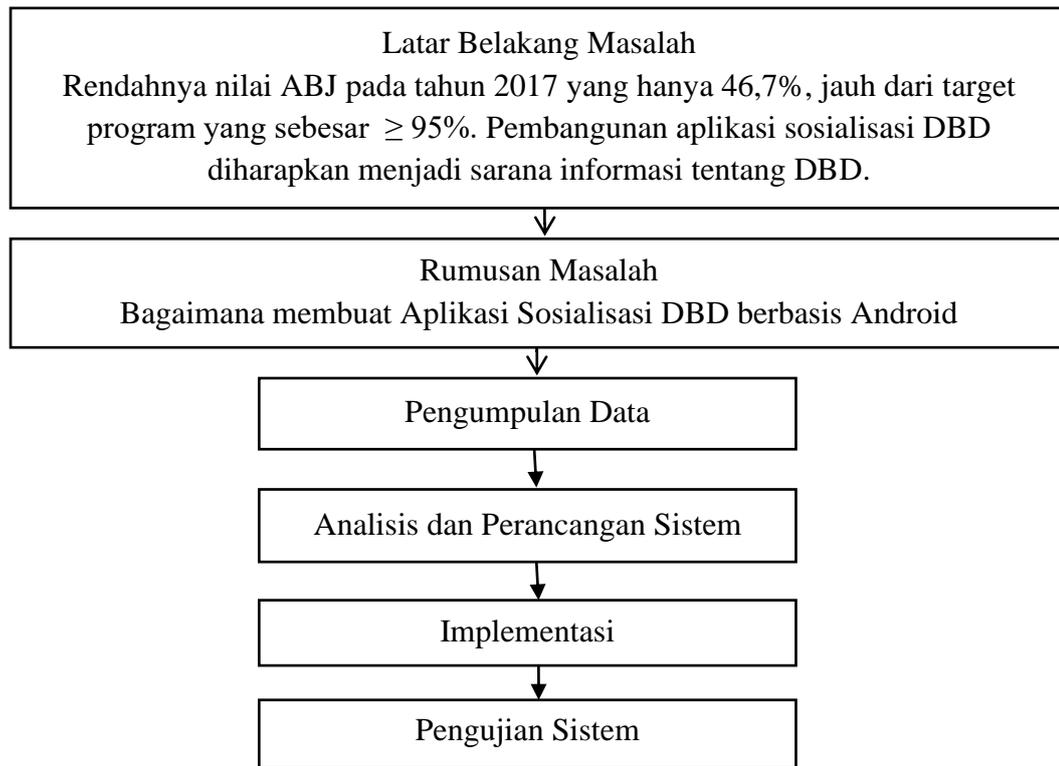
Penelitian ketiga dilakukan oleh Nugroho (2017) yang berjudul Pengembangan Media Pembelajaran Audio Visual Membatik Teknik Jumputan

Untuk Siswa Kelas VII di MTS Negeri Godean. Penelitian ini menghasilkan produk media pembelajaran audio visual membuat Teknik jumpitan menggunakan software *Adobe After Effect CS6*. Metode penelitian yang digunakan adalah *Research and Development* mengacu pada prosedur pengembangan Borg and Gall yang meliputi 10 tahap, yaitu observasi dan menemukan potensi masalah, pengumpulan data, pengembangan produk awal, validasi desain, revisi desain, uji coba produk, revisi produk, uji coba pemakaian, revisi, dan produksi massal. Teknik pengumpulan data menggunakan wawancara, pengamatan atau observasi, angket dan dokumentasi. Teknik analisis data menggunakan cara deskriptif kualitatif dari data kuantitatif sebagai pengembangan produk media pembelajaran Kriya ikat celup dan deskriptif kualitatif untuk mengetahui respon dari peserta didik.

Dari ketiga tinjauan Pustaka yang telah diambil sebelumnya terdapat perbedaan dengan penelitian yang akan dilakukan, yaitu : aplikasi sosialisasi pencegahan demam berdarah *dengue* (DBD) berjalan dengan *android* minimal versi *lollipop*. Pembangunan aplikasi menggunakan *software Android Studio*. Aplikasi Menampilkan informasi tentang nyamuk, pemberantasan sarang nyamuk serta menampilkan video animasi tentang pencegahan penyakit demam berdarah *dengue* (DBD), pembuatan animasi menggunakan *software Adobe After Effect CC 2017*.

2.2 Kerangka Pemikiran

Tahapan kerangka pemikiran dalam membuat Aplikasi Sosialisasi Pencegahan Penyakit Demam Berdarah Dengue (DBD) digambarkan pada Gambar 2.1 Kerangka Pemikiran.



Gambar 2.1 Kerangka Pemikiran

Penjelasan dari Gambar 2.1 adalah

1. Latar Belakang Masalah

Rendahnya nilai ABJ pada tahun 2017 yang hanya 46,7%, jauh dari target program yang sebesar $\geq 95\%$. Pembangunan aplikasi sosialisasi pencegahan penyakit DBD diharapkan mampu menjadi sarana media sosialisasi DBD.

2. Rumusan Masalah

Bagaimana membuat Aplikasi Sosialisasi Demam Berdarah Dengue (DBD) berbasis *Android*.

3. Pengumpulan data

Tahap pengumpulan data pada penelitian ini melalui studi pustaka. Pengumpulan data bertujuan untuk mengetahui permasalahan dan kebutuhan informasi.

4. Analisis dan Perancangan

Pada tahap ini analisis terhadap kebutuhan perangkat sistem yang akan dibangun dan perancangan aplikasi yang akan diimplementasi.

5. Implementasi Sistem

Pada tahap implementasi merupakan tahapan pembentukan program aplikasi dari sistem yang telah dirancang.

6. Pengujian Sistem

Pengujian sistem diutamakan pada fungsional perangkat lunak yang dibangun. Pengujian yang dilakukan adalah pengujian kompatibilitas aplikasi di beberapa perangkat *Android* hal ini dilakukan untuk meminimalisir kesalahan atau *error* dan memastikan hasil sesuai dengan perancangan.

2.3 Landasan Teori

2.3.1 DBD

DBD adalah infeksi yang disebabkan oleh virus *dengue*. *Dengue* adalah virus penyakit yang ditularkan dari nyamuk *aedes aegypti*. Gejala yang disebabkan oleh DBD yaitu berupa sakit/nyeri pada ulu hati terus menerus, perdarahan pada hidung, mulut, gusi, atau memar pada kulit.

Virus *Dengue* ditemukan didaerah tropik dan subtropik kebanyakan diwilayah perkotaan dan pinggiran kota. Untuk indonesia dengan iklim tropik yang sangat cocok untuk berkembangnya beragam penyakit, terutama penyakit yang dibawa oleh vektor, yakni organisme penyebar pantogen dari inang ke inang, seperti nyamuk yang banyak menularkan penyakit. Contoh nyamuk *aedes aegypti* tertapat pada Gambar 2.2 Nyamuk *aedes aegypti* (Kementrian Kesehatan Republik Indonesia, 2018).



Gambar 2.2 Nyamuk *aedes aegypti*

2.3.2 Siklus Penularan DBD

Virus dengue menginfeksi nyamuk *Aedes* betina saat menghisap darah dari seseorang yang sedang dalam fase demam akut atau *veraemia*, yaitu 2 hari

sebelum panas sampai 5 hari setelah demam timbul. Nyamuk menjadi infeksius pada 8 sampai 12 hari atau yang disebut periode inkubasi ekstrinsik, inkubasi ekstrinsik adalah masa sejak virus dengue masuk kedalam tubuh nyamuk sampai nyamuk tersebut menjadi infeksius dan dapat menularkan ke manusia. Sesudah nyamuk menghisap darah penderita yang sedang *viremia* dan tetap infeksius selama hidupnya.

Setelah melalui periode inkubasi ekstrinsik, kelenjar ludah nyamuk bersangkutan yang akan terinfeksi dan virusnya akan ditularkan ketika nyamuk tersebut menggigit dan mengeluarkan cairan ludahnya kedalam luka gigitan ke tubuh orang lain. Setelah masa inkubasi di tubuh manusia selama 34 hari (rata-rata selama 4 sampai 6 hari) timbul gejala awal penyakit. Gejala awal DBD di antara lain demam tinggi mendadak berlangsung sepanjang hari, nyeri saat menggerakkan bola mata dan nyeri punggung, kadang disertai adanya tanda-tanda perdarahan, pada kasus yang lebih berat dapat menimbulkan nyeri pada ulu hati, perdarahan saluran cerna, syok, hingga kematian. Masa inkubasi DBD 3 sampai 14 hari, tetapi pada umumnya 4 sampai 7 hari (Kementrian Kesehatan Republik Indonesia, 2015).

2.3.3 *Android*

Android merupakan sebuah sistem operasi telepon seluler dan komputer tablet berbasis *Linux*. *Google* merupakan pengembang utama *Android*. Platform *Android* terdiri dari sistem operasi berbasis *Linux*, *GUI (Graphic User Interface)*, *web browser* dan aplikasi *end-user* yang dapat di *download* serta para pengembang dapat berkarya dalam menciptakan aplikasi terbaik dan terbuka untuk dapat digunakan oleh berbagai perangkat (Kasman, 2015).

Menurut (Imaduddin & Permana, 2017) berikut merupakan versi-versi android;

- a. Tanpa Nama (Android versi 1.0.1.1)
- b. *Cupcake* (Android versi 1.2 – 1.5)
- c. *Donut* (Android versi 1.6)
- d. *Eclair* (Android versi 2.0 – 2.1)

- e. *Frozen Yogurt atau Froyo* (Android versi 2.2 – 2.2.3)
- f. *Gingerbread* (Android versi 2.3 – 2.4)
- g. *Honeycomb* (Android vversi 3.0 – 3.2)
- h. *Ice Cream Sandwich* (Android versi 4.0)
- i. *JellyBean* (Android versi 4.1 – 4.3)
- j. *KitKat* (Android versi 4.4)
- k. *Lollipop* (Android versi 5.0)
- l. *Marshmallow* (Android versi 6.0)
- m. *Nougat* (Android versi 7.0)
- n. *Oreo* (Android versi 8.0)

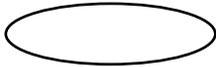
2.3.4 Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma berorientasi objek. Adapun tipe-tipe UML yaitu *Use case diagram*, *Class diagram*, *Activity diagram*, *Sequence diagram*, *Collaboration diagram*, *Component diagram*, dan *Deployment diagram*.

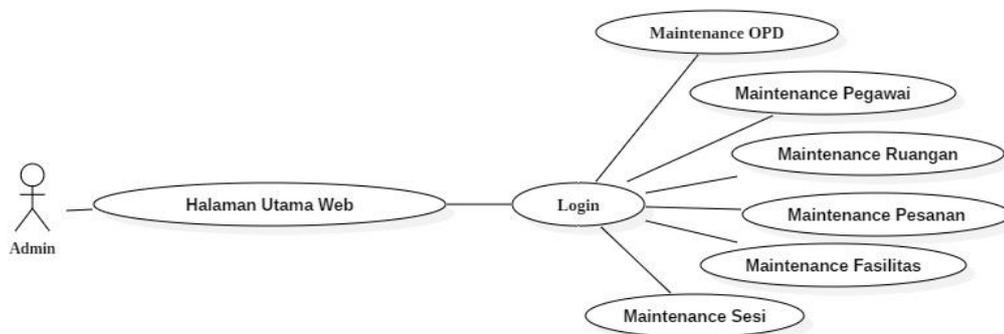
a. *Use Case Diagram*

Use Case Diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *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* ditunjukkan pada Tabel 2.1.

Tabel 2.1 Simbol *Use Case Diagram* (Rosa, A. S & Shalahuddin, 2016)

NO	SIMBOL	NAMA	KETERANGAN
1	<p>Nama <i>Use Case</i></p> 	<i>Use case</i>	Fungsional yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan kata kerja di awal di awal frase nama <i>use case</i> .
2		<i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3		<i>Association</i>	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4	<p><<extend>></p> 	<i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.
5		<i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6	<p><<include>></p> 	<i>Include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan simbol ini untuk menjalankan fungsinya.

Contoh *Use Case Diagram* pada Gambar 2.3.

Gambar 2.3 Contoh *Use Case Diagram*

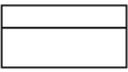
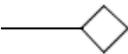
Pada Gambar 2.2 menjelaskan bahwa Admin harus melakukan login terlebih dahulu untuk dapat mengakses home admin. Admin dapat melihat data serta dapat melakukan *maintenance* data seperti OPD, Pegawai, Ruangan, Pesanan, Fasilitas, dan Sesi.

b. *Class Diagram*

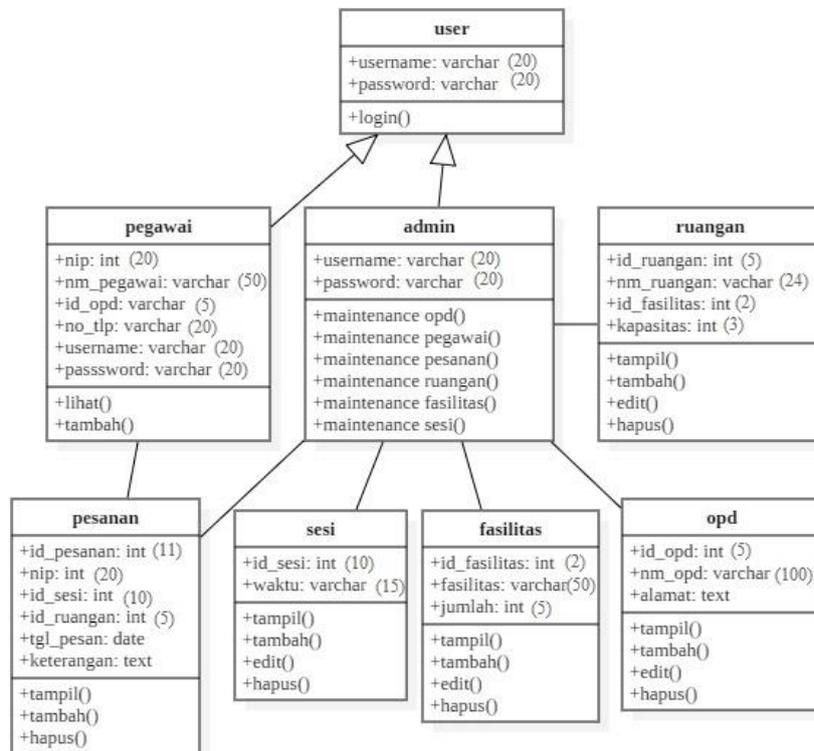
Class diagram merupakan penggambaran struktur pendefinisian kelas-kelas yang akan dibuat dalam pembangunan sistem.

Berikut adalah symbol-simbol yang ada pada class diagram dapat dilihat pada Tabel 2.2 (Rosa, A. S & Shalahuddin, 2016) :

Tabel 2.2 Simbol *Class Diagram*

NO	Gambar	Nama	Keterangan
1		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
2		Antarmuka/ <i>interface</i>	Sama dengan konsep interface dalam pemrograman berorientasi objek
3		<i>Association</i>	Menghubungkan antar kelas dengan makna umum biasanya disertai dengan <i>multiplicity</i>
4		Generalisasi	Relasi antarkelas dengan makna generalisasi-spesifikasi (umum khusus)
5		Agregasi / <i>Aggregation</i>	Digunakan untuk menghindari asosiasi dengan lebih dari 2 objek.
6		Asosiasi berarah	Relasi antat kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multitiplicity</i>
7		<i>Dependency</i>	Relasi antar kelas dengan makna ketergantungan antarkelas

Contoh *Class Diagram* pada Gambar 2.4.



Gambar 2.4 Contoh *Class Diagram*

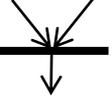
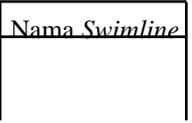
Class diagram menggambarkan struktur sistem kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki atribut dan operasi. Contoh *class diagram* diatas memiliki beberapa kelas, yaitu opd, admin, pegawai, pesanan, ruangan, fasilitas, dan sesi.

c. *Activity Diagram*

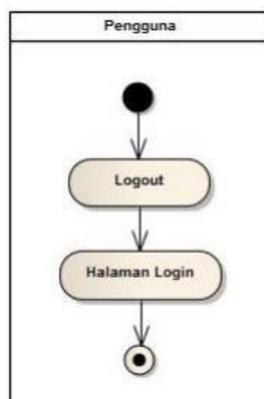
Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Perlu diperhatikan bahwa *activity diagram* mnggambarkan aktivitas sistem bukan menggambarkan apa yang dilakukan aktor, jadi aktivitas yang dilakukan oleh sistem.

Berikut adalah symbol-simbol pada *activity diagram* dapat dilihat pada Tabel 2.3 (Rosa, A. S & Shalahuddin, 2016) :

Tabel 2.3 Simbol *Activity Diagram*.

No	Gambar	Nama	Keterangan
1		Status awal	<i>Start Point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktivitas
2		Status akhir	<i>End Point</i> merupakan akhir aktivitas.
3		aktivitas	menggambarkan suatu proses kegiatan.
4		Penggabungan / <i>join</i>	Digunakan untuk menunjukkan adanya dekomposisi.
5		Percabangan / <i>decision</i>	Assosiasi penggabungan dimana jika ada pilihan aktivitas lebih dari satu.
6		<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Contoh *Activity Diagram Logout* pada Gambar 2.5.

Gambar 2.5 Contoh *Activiy Diagram Logout*

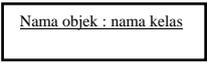
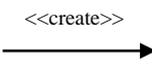
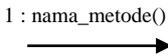
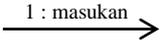
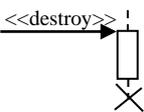
Pada Gambar 2.4 menjelaskan saat aktor memilih menu logout maka aktor akan keluar dan kembali kehalaman login.

d. *Sequence Diagram*

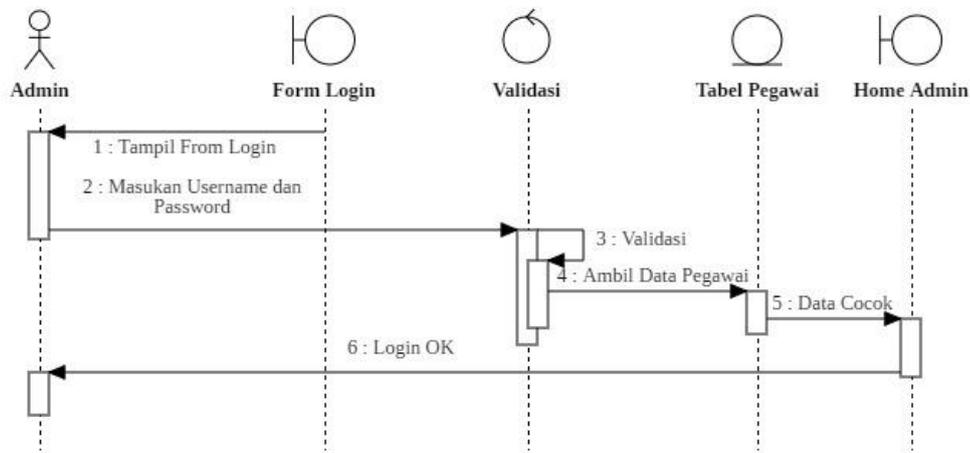
Menurut (Rosa dan Shalahuddin, 2016), *sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan

message yang dikirimkan dan diterima antar objek. Perlu diingat bahwa di dalam *sequence diagram*, kelas-kelas dan *actor-actor* diletakkan di bagian atas diagram dengan urutan dari kiri ke kanan dengan garis *lifeline* yang diletakkan secara vertikal terhadap kelas dan aktor. Berikut ini adalah Simbol *sequence diagram* dapat dilihat pada Tabel 2.4.

Tabel 2.4. Simbol *Sequence Diagram*

NO	SIMBOL	NAMA	KETERANGAN
1		Aktor	merupakan orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem.
2		<i>Lifeline</i>	Menyatakan kehidupan suatu objek.
3		Objek	Menyatakan objek yang berinteraksi pesan.
4		Waktu aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi.
5		Pesan tipe <i>create</i>	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
6		Pesan tipe <i>call</i>	Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri.
7		Pesan tipe <i>send</i>	Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8		Pesan tipe <i>destroy</i>	Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .

Contoh *sequence diagram* pada Gambar 2.6.



Gambar 2.6 Contoh *Sequence Diagram*

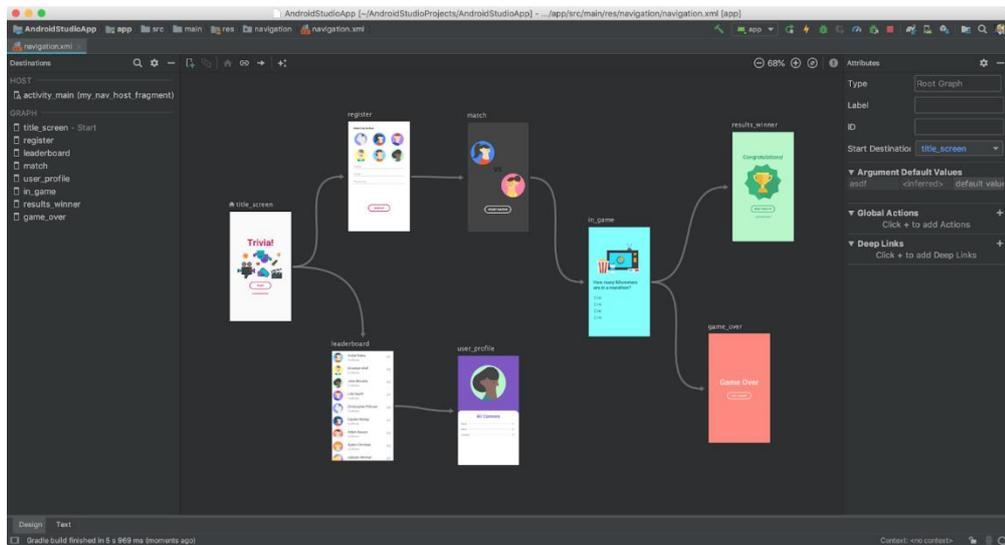
Sebelum admin melakukan *maintenance*, admin terlebih dahulu *login*. Admin memilih menu *login*, *website* akan menampilkan *form login*. Admin mengisi *Username* dan *Password*, setelah itu akan di-*validasi* dengan data yang ada pada tabel pegawai. Bila *valid* maka akan bisa masuk halaman admin, apabila tidak *valid* maka akan diminta untuk mengisi *Username* dan *Password* sampai *valid*.

2.3.5 *Android Studio 3.3*

Android Studio adalah *software* untuk membangun atau pengembangan aplikasi *android*. *Android Studio* merupakan lingkungan Pengembangan Terpadu atau *Integrate Development Environment (IDE)* berdasarkan *IntelliJ IDEA*.

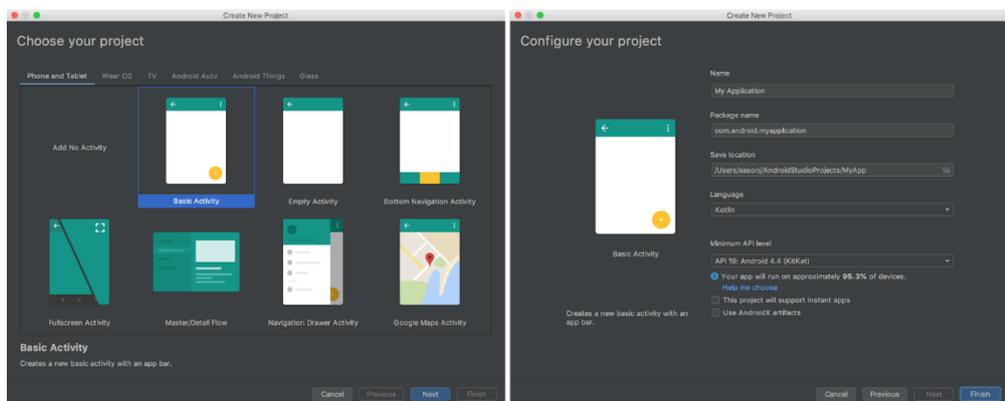
Berikut fitur-fitur yang ada di *Android Studio 3.3*.

- a. Memakai *Gradle-based build system* yang fleksibel.
- b. Editor navigasi dan komponen navigasi yang ada, kita dapat membangun interaksi yang dapat diprediksi antara layar dan aplikasi konten. Tampilan editor navigasi dapat dilihat pada Gambar 2.6.



Gambar 2.7 Editor Navigasi

- c. *Project Wizard* Baru dengan *template* proyek yang sama dalam penggunaan yang lebih ramping. Tampilan *project wizard* baru dapat dilihat pada Gambar 2.7 *Project Wizard*.

Gambar 2.8 *Project Wizard*

- d. Dapat mem-*build multiple* APK.
- e. Tersedianya *template support* untuk *Google Services* serta untuk tipe-tipe perangkat lainnya.
- f. *Google Cloud Platform built-in support*, maka akan memudah untuk diintegrasikan dengan Google.
- g. *Cloud Messaging* dan *App Engine* (googledevelopers, 2019).

2.3.6 *Adobe After Effect CC 2017*

Adobe After Effect adalah produk piranti lunak yang dikembangkan oleh *Adobe*, digunakan untuk film dan pos produksi pada video. Pada awalnya merupakan sebuah *software* dari produk *Macromedia* yang sekarang sudah menjadi salah satu produk *Adobe* (Muslim, 2017).

Pembuatan animasi pada penelitian ini menggunakan *Adobe After Effect CC 2017*. Adapun persyaratan sistem dari *Adobe After Effect CC 2017* dalam sistem operasi windows yaitu:

- a. *Multicore Intel processor 64-bit*.
- b. *Microsoft Windows 7 with Service Pack 1 (64 bit), Windows 8.1 (64 bit), atau Windows 10 (64 bit)*.
- c. RAM 8GB (16GB rekomendasi).
- d. 5GB ruang *hard-disk* yang tersedia, ruang kosong tambahan diperlukan selama instalasi.
- e. Ruang *disk* tambahan untuk *cache disk* (10GB rekomendasi).
- f. Tampilan layar 1280x1080.
- g. Opsional: Kartu GPU bersertifikat *Adobe* untuk *renderer 3D* pelacak *ray* akselerasi GPU.
- h. Koneksi *internet* diperlukan untuk aktivasi perangkat lunak yang diperlukan, validasi langganan, dan akses ke layanan *online* (Adobe, 2020).

2.4 Pengujian dengan *Black box*

Pengujian *Black box* adalah menguji dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui fungsi-fungsi masukan dan keluaran sesuai dengan yang dibutuhkan.

2.5 Pengujian kepuasan dengan kuesioner

Kuesioner ditujukan untuk mendapatkan penilaian saat proses pengujian. kuesioner berisi pertanyaan-pertanyaan yang diajukan kepada responden bertujuan untuk menilai program. Kuesioner dibentuk dalam skala empat atau lima poin dengan model skala *Likert* untuk memilih jawaban sebagai pengukuran

tingkat persetujuan pengguna terhadap pernyataan. Menurut Sugiyono (2017) skala *likert* digunakan untuk mengukur sikap, pendapat dan persepsi seseorang atau sekelompok tentang fenomena sosial.