

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Novi Wahyuningsih (2016) membuat aplikasi *chatting* berbasis android bernama Callind (Calling Indonesia), aplikasi ini sudah mempunyai fitur telepon, video *call*, sampai mengirimkan file secara langsung seperti foto, video. Aplikasi Callind ini juga bisa pasang iklan dan melakukan transaksi jual-beli produk. Namun, aplikasi ini hanya bisa menambahkan teman setelah *user* lain menyimpan nomor telepon kita yang terhubung dengan aplikasi ini dan karena hal itu pula jika ada *user* lain yang tidak kita kenal tetapi menyimpan nomor kita pada telepon-nya maka *user* tersebut dapat mengirimkan pesan kepada kita.

PT Gobsindo Utama (2015) meluncurkan aplikasi *chatting* berbasis android bernama Imes. Imes hanya bisa diunduh melalui situs resmi mereka yakni imesmessenger.com. Aplikasi ini mirip dengan aplikasi *chatting* lain lengkap dengan sticker seperti di layanan percakapan kebanyakan, atau Blitz! layaknya Ping! dalam BBM. Bagian pengaturan memberikan kontrol cukup mendetail bagi *user* misalnya memastikan pengunduhan file berlangsung bila penerima mengizinkannya. Tetapi, untuk pendaftaran akun baru sangat lah sulit. Setelah memasukan ID, alamat *email* dan *password* pendaftar harus memasukan kode sebanyak 10 karakter lebih yang dipakai untuk mendaftar dan kode itu hanya bisa di dapatkan pada situs resmi mereka saja.

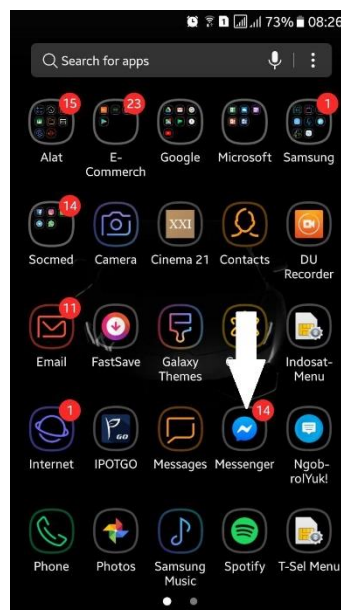
Tesar Sandikapura (2013) membuat aplikasi *chatting* berbasis android bernama LiteBig Messenger. Kelebihan liteBIG dari messenger lain adalah adanya fitur hapus dan edit pesan. Aplikasi ini juga mempunyai fitur layanan pembelian pulsa, paket data, listrik, dan lain-lain yang bisa dibayar dengan menggunakan liteCash (uang digital). Akan tetapi isi daftar kontak hanya ada jika *user* lain aplikasi ini telah menyimpan nomor telepon, jika *user* lain aplikasi ini tidak

menyimpan nomor telepon maka tidak ada di daftar kontak dan hal ini bisa membuat daftar kontak kosong yang pada akhirnya aplikasi ini tidak berguna.

Beberapa aplikasi *chatting* yang sudah tersedia di *Playstore* diantaranya:

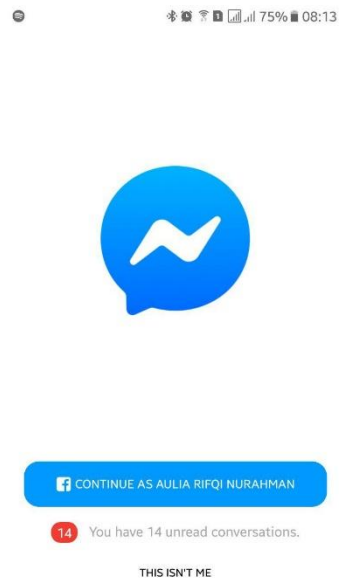
1. Messenger

Ikona aplikasi setelah di-*install* dapat dilihat pada gambar 2.1. Aplikasi Messenger merupakan aplikasi yang bergantung pada aplikasi Facebook (gambar 2.2) dan juga sebaliknya aplikasi Facebook bergantung pada aplikasi Messenger (gambar 2.3). Hal ini tentunya sangat menyulitkan bagi *user* karena harus meng-*install* kedua aplikasi, karena jika *user* hanya meng-*install* satu aplikasi maka aplikasi lainnya tidak bisa digunakan.

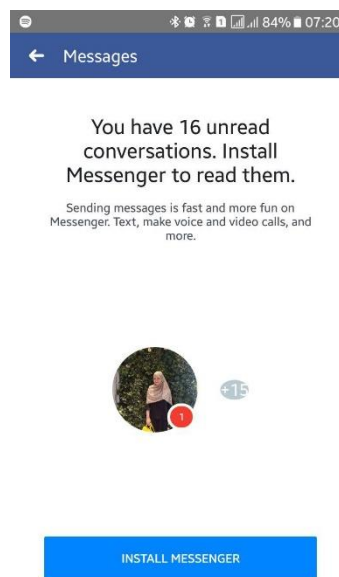


Gambar 2. 1 Ikona aplikasi Messenger

Diunduh dari *Playstore* pada tanggal 17 Februari 2019



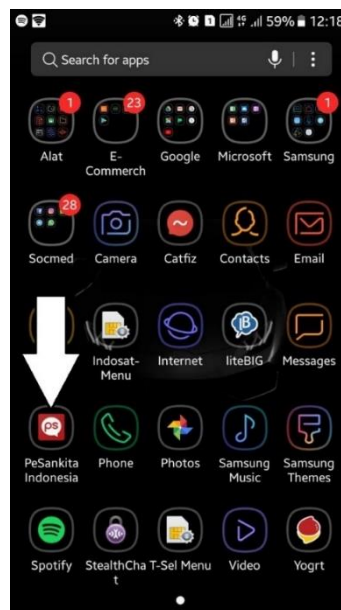
Gambar 2.2 Aplikasi Messenger bergantung pada aplikasi Facebook



Gambar 2.3 Aplikasi Facebook bergantung pada aplikasi Messenger

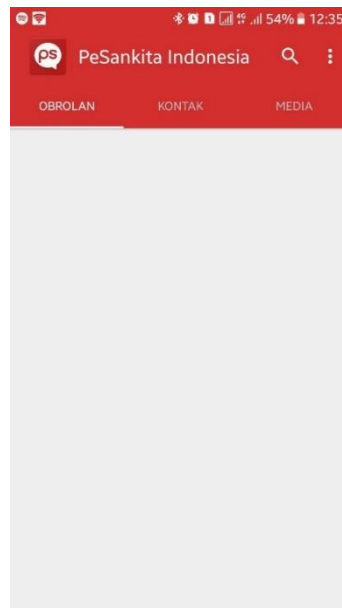
2. PeSankita Indonesia

Ikon aplikasi setelah di-*install* dapat dilihat pada gambar 2.4. Pendaftaran aplikasi PeSankita Indonesia juga dengan cara verifikasi nomor telepon dan sudah berjalan dengan baik. Tampilan dari aplikasi PeSankita Indonesia juga sudah cukup *user-friendly* dengan 3 sub-menu yaitu obrolan, kontak dan media sehingga *user* cukup nyaman dengan pemakain aplikasi ini, dapat dilihat pada gambar 2.5. Akan tetapi isi daftar kontak hanya ada jika *user* lain aplikasi ini telah menyimpan nomor telepon kita, jika *user* lain aplikasi ini tidak menyimpan nomor telepon kita maka tidak ada di daftar kontak dan hal ini bisa membuat daftar kontak kosong yang pada akhirnya aplikasi ini tidak berguna, dapat dilihat pada gambar 2.6.

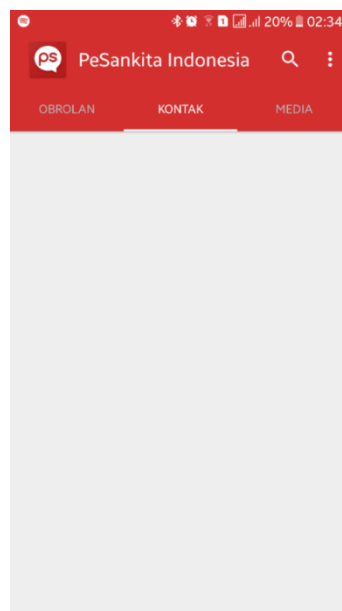


Gambar 2.4 Ikon Aplikasi PeSankita Indonesia

Diunduh dari *Playstore* pada tanggal 17 Februari 2019



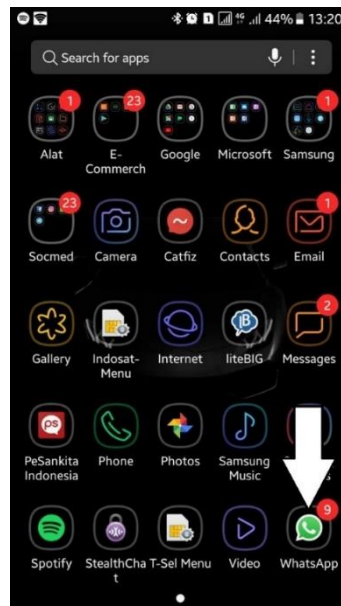
Gambar 2.5 Tampilan pada aplikasi PeSankita Indonesia



Gambar 2.6 Tampilan kontak pada aplikasi PeSankita Indonesia

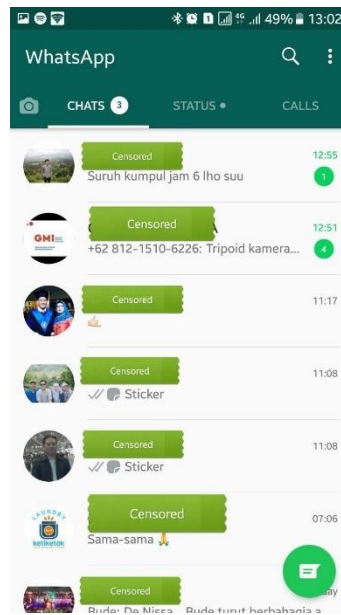
3. Whatsapp

Ikon aplikasi setelah diinstall dapat dilihat pada gambar 2.7. Aplikasi Whatsapp ini sudah banyak digunakan di Indonesia, jumlah *user* Whatsapp di Indonesia mencapai 24 juta orang (sumber: Deutsche Bank). Pendaftaran aplikasi Whatsapp juga dengan cara verifikasi nomor telepon dan sudah berjalan dengan baik. Tampilan dari aplikasi Whatsapp juga sudah cukup *user-friendly* dengan 3 sub-menu yaitu obrolan, status dan telepon sehingga *user* cukup nyaman dengan pemakain aplikasi ini, dapat dilihat pada gambar 2.8. Akan tetapi karena *user* lain dari aplikasi ini hanya dengan menambahkan nomor telepon kita bisa mengirim pesan kepada kita maka dari itu bisa saja orang yang kita tidak kenal bisa mengirim pesan kepada kita tanpa meminta persetujuan terlebih dahulu dan hal ini yang membuat aplikasi Whatsapp menjadi kurang privasi, dapat dilihat pada gambar 2.9.

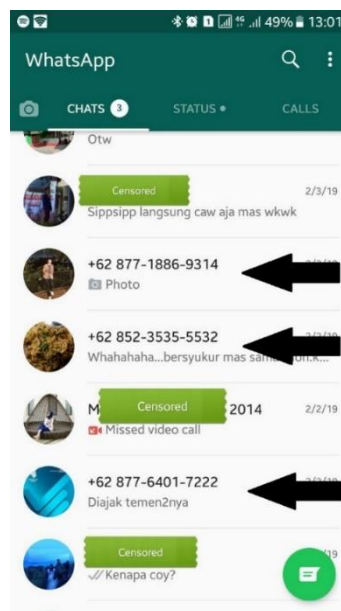


Gambar 2.7 Ikon aplikasi Whatsapp

Diunduh dari *Playstore* pada tanggal 17 Februari 2019



Gambar 2.8 Tampilan pada aplikasi Whatsapp



Gambar 2.9 User tidak di kenal bisa mengirimkan pesan

Perbedaan aplikasi *chatting* NgobrolYuk! dengan aplikasi yang sudah ada:

1. Dalam hal mencari dan menambahkan pertemanan aplikasi *chatting* NgobrolYuk! bisa mencari pengguna lain yang belum menjadi teman dengan menggunakan nama pengguna dan bisa menambahkan pertemanan dengan pengguna lain.
2. Dalam hal menghapus pertemanan aplikasi *chatting* NgobrolYuk! mempunyai fitur menghapus pertemanan.
3. Dalam hal privasi aplikasi *chatting* NgobrolYuk! pengguna lain dapat menambahkan pertemanan dengan meminta persetujuan dan pengguna hanya dapat menerima pesan dari pengguna lain yang sudah menjadi teman sehingga privasi lebih terjaga.
4. Dalam hal kemandirian aplikasi *chatting* NgobrolYuk! dapat berjalan dengan semestinya tanpa bergantung pada aplikasi lainnya.

2.2 Komunikasi (*Chatting*)

Menurut istilah ahli ilmu komunikasi, *chatting* merupakan suatu konsep yang multi makna. Makna komunikasi dapat dibedakan berdasarkan komunikasi sebagai proses sosial komunikasi pada makna ini ada dalam konteks ilmu sosial. Dimana para ahli ilmu sosial melakukan penelitian dengan menggunakan pendekatan komunikasi yang secara umum memfokuskan pada kegiatan manusia dan kaitan pesan dengan perilaku (Anwar, 1988).

2.3 Android

Android merupakan sistem operasi berbasis Linux yang mencakup sistem operasi dan sistem aplikasi diperangkat *mobile*. Android menyediakan platform yang terbuka bagi para pengembang untuk membuat aplikasi. Android merupakan generasi baru platform *mobile* yang memberikan pengembangan untuk mengembangkan aplikasi sesuai dengan yang diharapkan. Android dibuat di bawah lisensi Apache Software (ASL/Apache2), yang memungkinkan untuk didistribusi kedua dan seterusnya (Nazarudin, 2012). Menurut (Hermawan, 2011) Android merupakan OS (Operating System) Mobile yang tumbuh ditengah OS lainnya yang berkembang dewasa ini. OS lainnya seperti Windows Mobile, i-Phone OS,

Symbian, dan masih banyak lagi. Akan tetapi, OS yang ada ini berjalan dengan memprioritaskan aplikasi inti yang dibangun sendiri tanpa melihat potensi yang cukup besar dari aplikasi pihak ketiga.

Fitur Android

Fitur-fitur yang terdapat pada android adalah sebagai berikut:

1. *Multiprocess* dan *App Widgets*. Sistem operasi android tidak membatasi prosesor ke satu program saja, tetapi sistem bisa menjalankan lebih dari satu aplikasi. Namun, kekurangannya yaitu boros baterai jika banyak aplikasi yang dijalankan sekaligus.
2. *Touch, gesture* dan *multitouch touch screen* adalah antar muka *user* yang sangat intuitif untuk piranti tangan. *Multitouch* merupakan cara untuk berinteraksi dengan *touch screen* di lebih dari satu tangan pada saat yang bersamaan misalnya untuk perbesaran dan rotasi.
3. *Keyboard hard* dan *Soft*. Piranti android juga memiliki keyboard. Keyboard hard disebut juga keyboard fisik dan biasanya lebih cepat untuk mengetik, tetapi beberapa gadget hanya menyertakan keyboard *soft* agar lebih praktis, nyaman, dan desain yang ergonomis. Kekurangan dari keyboard *soft* yaitu memakan tempat karena perlu bagian untuk menampilkan keyboard tersebut.

2.4 Android SDK

Android SDK adalah alat yang diperlukan untuk memulai pengembangan aplikasi pada platform android menggunakan bahasa pemrograman Java. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi dan aplikasi kunci yang di rilis oleh Google. Saat ini android SDK sebagai alat bantu *developer* untuk mengembangkan aplikasi pada platform android dengan menggunakan bahasa pemrograman Java. Beberapa fitur-fitur android yang paling penting adalah:

1. *Framework* aplikasi mendukung penggantian komponen dan *reusable*.
2. *Integrated engine browser* berdasarkan *open source webkit*.

3. Grafis yang dioptimalkan dan didukung oleh *libraries* grafis 2D, grafis 3D berdasarkan spesifikasi OpenGL ES1.0
4. Media *support* yang mendukung audio, video, gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF) GSM Telephony.
5. Bluetooth, EDGE, 3G, dan WiFi.
6. Kamera, GPS, kompas, dan Accelrometer.
7. Lingkungan *development* yang lengkap termasuk perangkat *emulator*, *tools* untuk *debugging*, profil dan kinerja memori, dan *plug-in* untuk IDEA JetBrains (Nazarudin, 2012).

2.5 Android Developer Tool

Dalam mengembangkan aplikasi *chatting* NgobrolYuk! berbasis android, diperlukan beberapa *tools* sebagai berikut:

2.5.1 IDE (Integrated Development Environment)

IDE (*Integrated Development Environment*) adalah program komputer yang memiliki beberapa fasilitas yang diperlukan dalam pembangunan perangkat lunak. Tujuan dari IDE adalah untuk menyediakan semua utilitas yang diperlukan dalam membangun perangkat lunak.

2.5.2 Android SDK Manager dan AVD Manager

Android SDK manager adalah alat untuk pengembangan perangkat lunak yang berguna untuk membuat dan mengembangkan aplikasi ber-*platform* android. Didalam android SDK terdapat project sample dengan kode sumber, alat-alat untuk membangun aplikasi, sebuah emulator android (AVD), dan library yang dibutuhkan untuk membangun aplikasi android. Aplikasi yang ditulis pada bahasa pemrograman Java dan berjalan di Dalvik, mesin virtual yang dirancang khusus untuk *user* embedded di atas kernel Linux. 2.8. Unified Modeling Language (UML) Suhendar dan Hariman Gunadi (2002) menuliskan dalam bukunya bahwa Unified Modeling Language (UML) merupakan sistem arsitektur yang bekerja dalam OOAD (Object Oriented Analysis and Desain) dengan satu bahasa yang konsisten untuk menentukan, konstruksi, visualisasi,

dan mendokumentasi artifact yang terdapat dalam *software*. UML merupakan bahasa pemodelan dari metode OO (object oriented) (Gunadi, 2002)

2.6 Java

Java adalah sebuah teknologi dimana pada teknologi tersebut mencakup java sebagai bahasa pemrograman yang memiliki sintaks dan aturan pemrograman tersendiri, juga mencakup java sebagai platform dimana teknologi ini memiliki *virtual machine* dan *library* yang diperlukan untuk menulis dan menjalankan program yang ditulis dengan bahasa pemrograman java (Rickyanto, 2003).

2.7 Android Studio

Android Studio adalah lingkungan pengembangan terintegrasi (IDE) resmi untuk sistem operasi android Google, dibangun di atas perangkat lunak IntelliJ IDEA JetBrains dan dirancang khusus untuk pengembangan aplikasi android, lingkungan pengembangan terintegrasi Java untuk perangkat lunak, dan menggabungkan pengeditan kode dan alat pengembang. Android Studio adalah pengganti Eclipse *Android Development Tools* (ADT) sebagai IDE utama untuk pengembangan aplikasi android asli. (Development, 2018)

2.8 Firebase

Firebase adalah suatu layanan dari Google yang digunakan untuk mempermudah para pengembang aplikasi dalam mengembangkan aplikasi. Dengan adanya Firebase, pengembang aplikasi bisa fokus mengembangkan aplikasi tanpa harus memberikan usaha yang besar.

Fitur-fitur firebase yang digunakan dalam pembuatan aplikasi *chatting* NgobrolYuk!:

1. *Firebase Authentication*

Sebagian besar aplikasi perlu mengetahui identitas *user*. Dengan mengetahui identitas *user*, aplikasi dapat menyimpan data *user* secara aman di cloud dan memberikan pengalaman personal yang sama di setiap perangkat *user*.

Firestore *Authentication* menyediakan layanan *backend*, SDK yang mudah digunakan, dan *library* UI yang siap pakai untuk mengautentikasi *user* ke aplikasi. Firestore *Authentication* mendukung autentikasi menggunakan sandi, nomor telepon, penyedia identitas gabungan yang populer, seperti Google, Facebook, dan Twitter, dan lain-lain.

2. Firestore *Realtime Database*

Firestore *Realtime Database* adalah *database* yang di-*host* di *cloud*. Data disimpan sebagai JSON dan disinkronkan secara *realtime* ke setiap klien yang terhubung. Ketika kita membuat aplikasi lintas-*platform* dengan SDK Android, iOS, dan JavaScript, semua klien akan berbagi sebuah *instance realtime database* dan menerima *update* data terbaru secara otomatis.

3. Firebase *Cloud Storage*

Cloud Storage untuk Firebase adalah layanan penyimpanan objek yang andal, sederhana, dan hemat biaya yang dibuat untuk skala Google. Firebase SDK untuk *cloud storage* menambahkan keamanan Google pada *upload* dan *download* file untuk aplikasi Firebase, bagaimanapun kualitas jaringannya. Kita dapat menggunakan SDK untuk menyimpan gambar, audio, video, atau konten buatan *user* lainnya. Di server, kita dapat menggunakan Google *cloud storage* untuk mengakses file yang sama.

4. Firebase *Cloud Functions*


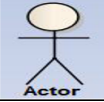

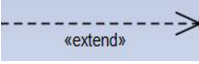

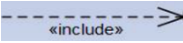
Dengan *Cloud Functions* untuk Firebase, kita dapat menjalankan kode *backend* secara otomatis sebagai respons terhadap peristiwa yang dipicu oleh fitur Firebase dan permintaan HTTPS. Kode Anda disimpan di *cloud* Google dan dijalankan di lingkungan yang terkelola. Anda tidak perlu mengelola atau menyesuaikan skala server sendiri.

2.9 Use Case Diagram

Use case diagram dapat digunakan selama proses analisis untuk membuat *requirements* sistem dan untuk memahami bagaimana seharusnya sistem bekerja. Selama tahap desain, *use case* diagram menetapkan perilaku sistem saat diimplementasikan. Dalam sebuah model bisa saja terdapat satu atau beberapa *use case* diagram. Beberapa simbol-simbol yang ada pada diagram *use case* disajikan dalam tabel berikut ini (lihat tabel 2.10). *Use case* diagram juga dapat diterangkan sebagai sebuah dialog antara sistem dengan aktor. Aktor digambarkan sebagai manusia yang dapat berperan sebagai administrator, moderator, *user* atau bagian yang lain yang dilakukan oleh manusia, pemakaian *use case* diagram memberi kemudahan dalam mengembangkan aplikasi yang dapat ditinjau dari aplikasi itu dijalankan dan siapa saja yang terlibat dalam aplikasi tersebut maupun proses yang terdapat di dalam aplikasi tersebut. Pemakaian *use case* diagram telah banyak digunakan dalam pengembangan sistem aplikasi baik aplikasi *desktop* maupun aplikasi *mobile* khususnya aplikasi orientasi objek yaitu aplikasi yang dibangun

dengan memperhatikan sisi *user* yang terlibat dalam sistem atau aplikasi sehingga efisiensi di dalam penerapannya selama proses *coding* dapat dilakukan dengan baik. Namun demikian *activity* diagram sebagai cara lain dalam membuat sebuah diagram pun tetap dilakukan karena *activity* diagram lebih umum dilakukan dahulu dalam mengembangkan aplikasi karena *activity* diagram menunjukkan proses secara berurutan atau sekuensial (Komara, 2016).






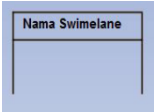
Tabel 2.10 Use Case Diagram

No	Nama	Simbol	Deskripsi
1	<i>Use Case</i>		Fungsionalitas yang disediakan sistem sebagai alat untuk bertukar pesan antar unit atau actor, dan dinyatakan dengan menggunakan kata kerja frase dari nama <i>use case</i> .
2	Aktor		Aktor atau sistem lain yang berinteraksi dengan sistem yang akan dibangun di luar sistem informasi yang akan dibuat.
3	Asosiasi		Komunikasi antara aktor dengan use case dan use case lain yang memiliki interaksi dengan aktor.
4	Ekstensi		Relasi use case tambahan ke sebuah use case yang ditambahkan dan dapat berdiri sendiri walau tanpa use case tambahan itu.
5	Generalisasi		Hubungan umum dan khusus antara dua buah use case di mana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya.
6	<i>Include</i> (Menggunakan)		Hubungan use case tambahan ke sebuah use case yang ditambahkan untuk menjalankan fungsinya.

2.10 Activity Diagram

Activity diagram sebuah proses bisnis dan urutan aktivitas dalam suatu proses. Diagram mirip dengan sebuah *Flowchart* karena penulis dapat memodelkan sebuah alur kerja dari satu aktivitas ke aktivitas lainnya atau dari satu aktivitas ke dalam keadaan sesaat (*state*). Beberapa simbol-simbol yang ada pada diagram *activity* diagram disajikan dalam Tabel 2.11.



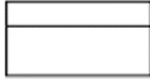




Tabel 2.11 Activity Diagram

No	Nama	Simbol	Deskripsi
1	Status Awal		Status awal aktivitas sistem sebuah diagram.
2	Aktivitas		Aktivitas yang dilakukan sistem, biasanya diawali dengan kata kerja.
3	Percabangan		Percabangan jika ada pilihan aktivitas lebih dari satu.
4	Penggabungan		Asosiasi penggabungan lebih dari satu aktivitas digabungkan menjadi satu.
5	Status Akhir		Status akhir yang dilakukan sistem.
6	<i>Swimlane</i>		Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

2.11 Class Diagram

Class diagram adalah “merupakan himpunan dari objek-objek yang sejenis”. Sebuah objek memiliki keadaan sesaat (*state*) dan perilaku (*behavior*). *State* sebuah objek adalah kondisi objek tersebut yang dinyatakan dalam *attribute/properties*. Sedangkan perilaku suatu objek mendefinisikan bagaimana sebuah objek bertindak/beraksi dan memberikan reaksi. (Munawar, 2005).

Tabel 2.12 Class Diagram

No	Nama	Simbol	Deskripsi
1	<i>Generalization</i>		Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2	<i>N-Ary Association</i>		Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3	<i>Class</i>		Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4	<i>Collaboration</i>		Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
5	<i>Realization</i>		Operasi yang benar-benar dilakukan oleh suatu objek.
6	<i>Dependency</i>		Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
7	<i>Association</i>		Apa yang menghubungkan antara objek satu dengan objek yang lainnya.