

BAB II LANDASAN TEORI

2.1 APLIKASI

Menurut Maryono dan Istiana, aplikasi merupakan program yang dikembangkan untuk memenuhi kebutuhan pengguna dalam menjalankan pekerjaan tertentu. Aplikasi berasal dari bahasa Inggris *application* yang berarti penerapan, penggunaan. Sedangkan secara istilah aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer secara langsung untuk melakukan suatu tugas yang diinginkan pengguna. Sedangkan secara umum aplikasi adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya (Abdur Rofiqu, 2013).

2.2 PEMBELAJARAN

Pembelajaran adalah proses interaksi peserta didik dengan pendidik dan sumber belajar pada suatu lingkungan belajar. (Undang-Undang No. 20 Tahun 2003 Tentang Sisdiknas Pasal 1 Ayat 20) maksudnya pembelajaran sebagai proses belajar yang dibangun oleh guru untuk mengembangkan kreativitas berpikir yang dapat meningkatkan kemampuan berpikir siswa, serta dapat meningkatkan pengetahuan siswa sebagai upaya meningkatkan penguasaan yang baik terhadap materi pembelajaran yang didukung dengan sumber belajar seperti buku ataupun sumber belajar yang lain

2.3 DOA

Doa adalah permohonan (harapan, permintaan, pujian) kepada Tuhan. (kbbi.web.id). Menurut Bahasa doa berasal dari kata "da'a" artinya memanggil. Sedangkan menurut istilah syara' doa berarti memohon sesuatu yang bermanfaat dan memohon terbebas atau tercegah dari sesuatu yang memudharatkan.

1. *Pertama*, doa dalam pengertian "Ibadah." Seperti dalam Al-Quran surat Yunus ayat 106.

وَلَا تَدْعُ مِنْ دُونِ اللَّهِ مَا لَا يَنْفَعُكَ وَلَا يَضُرُّكَ

Artinya: "Dan janganlah kamu beribadah, kepada selain Allah, yaitu kepada sesuatu yang tidak dapat mendatangkan manfaat kepada engkau dan tidak pula mendatangkan madarat kepada engkau."

2. *Kedua*, doa dalam pengertian "Istighatsah" (memohon bantuan dan pertolongan). Seperti dalam Al-Quran surat Al-Baqarah ayat 23.

وَادْعُوا شُهَدَاءَكُمْ مِنْ دُونِ اللَّهِ

Artinya: "Dan berdoalah kamu (mintalah bantuan) kepada orang-orang yang dapat membantumu."

3. *Ketiga*, Doa dalam pengertian "permintaan" atau "permohonan." Seperti dalam Al-Quran surat Al-Mu'minin ayat 60:

وَقَالَ رَبُّكُمْ ادْعُونِي أَسْتَجِبْ لَكُمْ

Artinya: Dan Rabbmu berfirman: "Berdoalah kepada-Ku, niscaya akan Ku-perkenankan bagimu"

4. *Keempat*, Doa dalam pengertian "percakapan". Seperti dalam Al-Quran surat Yunus ayat 10 :

دَعَاؤُهُمْ فِيهَا سُبْحَانَكَ اللَّهُمَّ

Artinya: "Doa (percakapan) mereka di dalamnya (surga), adalah Subhânakallâhumma (Mahasuci Engkau wahai Tuhan)".

5. *Kelima*, Doa dalam pengertian "memanggil." Seperti firman Allah dalam Al-Quran :

يَوْمَ يَدْعُوكُمْ

Artinya: "Pada hari, dimana la mendoa (memanggil) kamu."

6. *Keenam*, Doa dalam pengertian "memuji." Seperti dalam Al-Quran surat Al-Isra ayat 110 :

قُلْ ادْعُوا اللَّهَ أَوْ ادْعُوا الرَّحْمَنَ ۗ أَيًّا مَا تَدْعُوا فَلَهُ الْأَسْمَاءُ الْحُسْنَىٰ

Artinya: "Katakanlah olehmu hai Muhammad: berdoalah (pujilah) akan Allah atau berdoalah (pujilah), akan Ar-Rahmân (Maha penyayang)."

2.4 SURAH

Dari segi bahasa kata surah jamaknya *suwar* yang berarti *kedudukan atau tempat yang tinggi*. Sesuai dengan kedudukan Al-qur'an karena dia diturunkan dari tempat yang tinggi yaitu *Lauh Al-Mahfudz*. Sedangkan menurut istilah surat adalah sejumlah beberapa ayat Al-quran yang mempunyai permulaan dan akhiran (**Muhammad Makmun Rasyid.2012**).

Dari definisi ini dapat dipahami bahwa surah adalah kumpulan dari beberapa ayat, maka tidak ada satu surah yang terdiri hanya satu ayat, dia harus memiliki sejumlah ayat minimal 3 ayat seperti dalam surat al-katsar. Kumpulan beberapa ayat ini syaratnya mempunyai permulaan dan akhiran. Jika terkumpul sejumlah ayat meskipun banyak akan tetapi tidak mempunyai permulaan dan akhiran maka belum dinamai surat.

2.5 MURROTTAL

Murottal adalah sebuah kajian yang membahas maqom atau irama dalam membaca Al Quran. Irama tersebut ada tujuh. Umumnya masyarakat hanya mengenal tujuh irama ini dalam Qiroah atau dalam cabang MTQ biasa disebut Tilawatil Quran. Padahal tujuh irama ini juga bisa digunakan dalam murottal atau dalam cabang MTQ biasa disebut Tartil.

Perbedaan antara murottal dan qiroah yaitu pada kecepatan, serta panjangnya. Jika murottal lebih cepat dan panjangnya seadanya, maka qiroah biasanya lebih lambat dan lebih panjang bacaannya. Tentang tujuh irama tersebut mengikuti aturan dari lembaga Quran Maqdis, maka bisa disingkat **Ba, Sho, Hi, Ji, Ni, Ka, Rost**. Yaitu:

1. **Bayyati**

Biasa ditemui dalam rekaman murattal *Sheikh Misyari Rasyid El Afasy* atau bisa mendengar Thaha Al Junayd

2. **Shoba**

Ciri : Pelan dan sedih

3. **Hijaz**

Biasa ditemui dalam rekaman murattal *Sheikh Hani Arrifa'i*

4. **Jiharkah**

Ciri : Memiliki kemiripan dengan irama ala melayu/india

5. **Nihawwand / Nahawwand**

Biasa ditemui dalam rekaman murattal *Sheikh Khusaery*

6. **Sikkah**

Biasa ditemui dalam rekaman murattal *Sheikh Sa'ad Al Ghamidy*

7. **Rost / Rostah**

Biasa ditemui dalam rekaman murattal *Sheikh Abdurrahman As Sudais*

Kelebihan Metode Murrotal

1. Mengetahui kualitas bacaan sebuah surah yang benar. *Qori / qoriah* yang melantunkan ayat-ayat Alquran sudah melewati masa pembelajaran yang panjang, sehingga mereka dipercaya mengisi konten dalam *murottal* tersebut. Untuk memudahkan pencarian, Anda bisa memilih *qori / qoriah* yang sudah terkenal.
2. Bukan rahasia lagi kalau mendengarkan Alquran akan membawa ketenangan bagi pembaca maupun pendengarnya. Jika Anda mendengarkan *murottal* secara rutin, bukan tidak mungkin kedamaian tersebut muncul di hati Anda.

Hal tersebut hadir berkat iman dan keyakinan yang terus terpupuk dari lantunan ayat-ayat suci Alquran.

3. *Murottal* Alquran 30 *Juz* sangat membantu Anda yang sedang menghafalkan surah-surah di dalam kitab tersebut. Sebab, faktanya cara membaca yang baik akan meningkatkan daya ingat maupun daya ingat dari pendengarnya. Jadi, sediakan rekaman ayat-ayat Alquran yang dapat menunjang Anda dalam proses hafalan tersebut.
4. Karena dikemas dalam format MP3 atau sejenisnya, Anda dapat memasukkan *murottal* ke dalam ponsel pintar. Sehingga Anda dapat mendengarkannya di waktu luang atau di tengah kemacetan. Jika dilakukan terus-menerus, mendengarkan bacaan Al Quran tersebut akan menjadi kebiasaan yang sulit Anda lepaskan nantinya.

2.6 ANDROID

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis *Linux* yang mencakup sistem operasi, *middleware* dan aplikasi (Safaat Nazruddin 2012:1). Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya *Google Inc* membeli *Android Inc* yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel/*smartphone*. 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 saat perilis perdana Android, 5 November 2007, Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, *Google* merilis kode-kode Android di bawah lisensi *Apache*, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler.

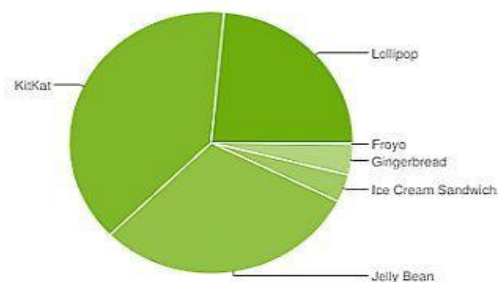
2.6.1 VERSI ANDROID

Berikut ini versi Android yang pernah dirilis sejak tahun 2009 sampai dengan tahun 2012.

1. Android versi 1.1
2. Android versi 1.5 (Cupcake)
3. Android versi 1.6 (Donut)
4. Android versi 2.0/2.1 (Eclair)

5. Android versi 2.2 (Froyo: Frozen Yoghurt)
6. Android versi 2.3 (Gingerbread)
7. Android versi 3.0/3.1 (Honeycomb).
8. Android versi 4.0 (ICS :Ice Cream Sandwich)
9. Android versi 4.1 (Jelly Bean)
10. Android versi 4.4 (KitKat)
11. Android versi 5.0 (Lollipop)
12. Android versi 6.0 (MarshMellow)

Version	Codename	API	Distribution
2.2	Froyo	8	0.2%
2.3.3 - 2.3.7	Gingerbread	10	3.8%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	3.4%
4.1.x	Jelly Bean	16	11.4%
4.2.x		17	14.5%
4.3		18	4.3%
4.4	KitKat	19	38.9%
5.0	Lollipop	21	15.6%
5.1		22	7.9%



Gambar 1. Distribusi Android selama periode, update September 2015

Sumber: <http://developer.android.com>

2.6.2 ANDROID STUDIO

Android Studio adalah sebuah *IDE* untuk *Android Development* yang diperkenalkan *Google* pada acara *Google I/O 2013*. *Android Studio* merupakan pengembangan dari *Eclipse IDE*, dan dibuat berdasarkan *IDE* Java populer, yaitu *IntelliJ IDEA*. *Android Studio* merupakan *IDE* resmi untuk pengembangan aplikasi *Android*.

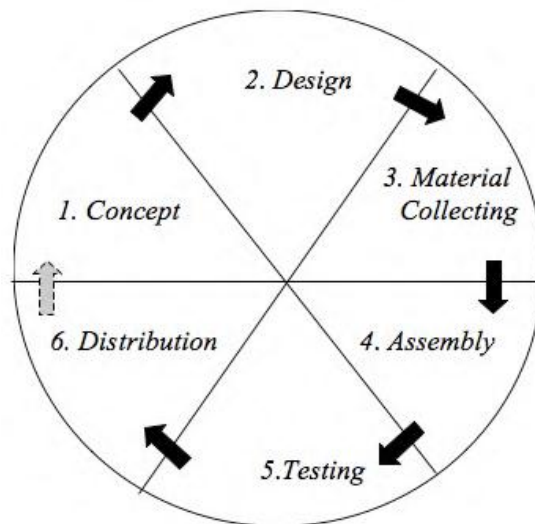
Sebagai pengembangan dari *Eclipse*, *Android Studio* mempunyai banyak fitur-fitur baru dibandingkan dengan *Eclipse IDE*. Berbeda dengan *Eclipse* yang menggunakan *Ant*, *Android Studio* menggunakan *Gradle* sebagai build *Environment*. Fitur-fitur lainnya adalah sebagai berikut :

1. Menggunakan *Gradle-based build system* yang fleksibel.
2. Bisa mem-build *multiple APK*.
3. *Template support* untuk *Google Services* dan berbagai macam tipe perangkat.

4. *Layout editor* yang lebih bagus.
5. Built-in support untuk *Google Cloud Platform*, sehingga mudah untuk integrasi dengan *Google Cloud Messaging* dan *App Engine*.
6. *Import library* langsung dari *Maven repository*

2.7 TEORI PENGEMBANGAN MULTIMEDIA LUTHER

Metode yang digunakan dalam pengembangan multimedia ini adalah Multimedia Development Life Cycle. Pengembangan metode multimedia ini dilakukan berdasarkan enam tahap, yaitu *concept* (pengonsepan), *design* (pendesignan), *material collecting* (pengumpulan materi), *assembly* (pembuatan), *testing* (pengujian), dan *distribution* (pendistribusian). Tahap pengembangannya dapat dilihat seperti pada Gambar 2.



Gambar 2. Tahap Pengembangan Multimedia Development Life Cycle (Sutopo, 2003)

1. Konsep (*Concept*)

Tahap *Concept* (konsep) yaitu menentukan tujuan dan siapa pengguna program (identifikasi audience), macam aplikasi (presentasi, interaktif, dan lain-lain), tujuan aplikasi (informasi, hiburan, pelatihan, dan lain-lain), dan spesifikasi umum. Dasar aturan untuk perancangan juga ditentukan pada tahap ini, seperti ukuran aplikasi, target, dan lain-lain. Tujuan dan pengguna akhir program berpengaruh pada nuansa multimedia sebagai pencerminan dari identitas organisasi yang menginginkan informasi sampai pada pengguna akhir. Karakteristik pengguna termasuk kemampuan pengguna juga perlu dipertimbangkan karena dapat mempengaruhi pembuatan design.

2. Perancangan (*Design*)

Design (perancangan) adalah membuat spesifikasi secara rinci mengenai arsitektur program, gaya, tampilan dan kebutuhan material / bahan untuk program. Spesifikasi dibuat cukup rinci sehingga pada tahap berikutnya, yaitu material *collecting* dan *assembly* tidak diperlukan keputusan baru, tetapi menggunakan apa yang sudah ditentukan pada tahap design. Namun demikian, sering terjadi penambahan bahan atau bagian aplikasi ditambah, dihilangkan, atau diubah pada awal pengerjaan proyek.

3. Pengumpulan Bahan (*Material Collecting*)

Material Collecting (pengumpulan bahan) adalah tahap pengumpulan bahan yang sesuai dengan kebutuhan yang dikerjakan. Bahan-bahan tersebut, antara lain gambar clip art, foto, animasi, video, audio, dan lain-lain yang dapat diperoleh secara gratis atau dengan pemesanan kepada pihak lain sesuai dengan rancangannya. Tahap ini dapat dikerjakan secara parallel dengan tahap *Assembly*.

4. Pembuatan (*Assembly*)

Tahap assembly (pembuatan) adalah tahap pembuatan semua objek atau bahan multimedia.

5. Pengujian (*Testing*)

Tahap testing (pengujian) dilakukan setelah menyelesaikan tahap pembuatan (*assembly*) dengan menjalankan aplikasi / program dan dilihat apakah ada kesalahan atau tidak. Tahap pertama pada tahap ini disebut juga sebagai tahap pengujian alpha (alpha test) yang pengujiannya dilakukan oleh pembuat atau lingkungan pembuatnya sendiri. Setelah lolos dari pengujian alpha, pengujian beta yang melibatkan pengguna akhir akan dilakukan.

6. Distribusi (*Distribution*)

Tahap ini aplikasi akan disimpan dalam suatu media penyimpanan. Tahap ini juga dapat disebut tahap evaluasi untuk pengembangan produk yang sudah jadi supaya menjadi lebih baik. Hasil evaluasi ini dapat digunakan sebagai masukan untuk tahap concept pada produk selanjutnya.

2.8 DATABASE SQLITE

SQLite merupakan sebuah Database yang bersifat *ACID-compliant* dan memiliki ukuran pustaka kode yang relative kecil, ditulis dalam bahasa C. SQLite merupakan proyek yang bersifat public domain yang dikerjakan oleh D.Richard Hipp. *SQLite* adalah sebuah open source database yang telah ada cukup lama,

cukup stabil, dan sangat terkenal pada perangkat kecil, termasuk Android (jasonkaunang.2014).

Android menyediakan database relasional yang ringan untuk setiap aplikasi menggunakan *SQLite*. Aplikasi dapat mengambil keuntungan dari itu untuk mengatur relational database engine untuk menyimpan data secara aman dan efisien. Untuk Android, *SQLite* dijadikan satu di dalam Android runtime, sehingga setiap aplikasi Android dapat membuat basis data *SQLite*. Karena *SQLite* menggunakan antarmuka SQL, cukup mudah untuk digunakan orang-orang dengan pengalaman lain yang berbasis databases

PHP, Firefox, Chrome, iPhone dan Android adalah contoh produk yang menggunakan *SQLite*. di *Browser firefox*, *SQLite* dipakai untuk menyimpan konfigurasi, *bookmark* dan *history website* sedangkan di *smartphone android*, *SQLite* dipakai untuk menyimpan *contact*.

2.9 UNIFIED MODELING LANGUAGE (UML)

Menurut Huda, *Unified Modeling Language (UML)* adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk *visualisasi*, merancang dan mendokumentasikan sistem peranti lunak. *UML* menawarkan sebuah standar untuk merancang model sebuah sistem.





Menurut Rosa dan Shalahuddin, *UML (Unified Modeling Language)* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *Requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek

UML sendiri juga memberikan standar penulisan kelas-kelas dalam bahasa program yang spesifik, skema *database*, dan komponen-komponen yang diperlukan dalam sistem *software*. *UML* sebagai sebuah bahasa yang memberikan tatanan penulisan kata-kata dalam *MS Word* untuk kegunaan komunikasi. Sebuah bahasa model adalah sebuah bahasa yang mempunyai konsep aturan penulisan serta secara fisik mempresentasikan dari sebuah sistem.

2.9.1 Use Case Diagram

Use case menggambarkan fungsi-fungsi sistem dari sudut pandang pengguna *eksternal* dan dalam sebuah cara yang mudah dipahami. *Use case* merupakan penyusunan kembali lingkup *fungsi* sistem yang disederhanakan lagi.

Tabel 1. Simbol *use case diagram*

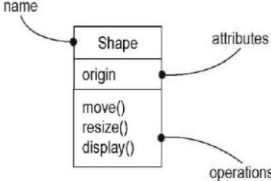
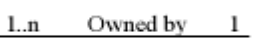
Simbol	Keterangan
	System Bondary menggambarkan batasan anatara sistem dengan aktor
	Simbol ini menggambarkan interaksi anatara aktor dengan sistem.
	Aktor menggambarkan pengguna sistem, dapat berupa manusia atau sistem terotomatisasi lain yang berinteraksi dengan sistem lain untuk berbagi, mengirim, dan menerima informasi.
	Menggambarkan hubungan antar aktor dan use case.

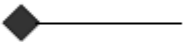


2.9.2 Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

- Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas
- Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas

Tabel 2. Simbol *class diagram*


Simbol	Keterangan
	<i>Class</i> adalah blok - blok pembangun pada pemrograman berorientasi obyek. Sebuah <i>class</i> digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian. Bagian atas adalah bagian nama dari <i>class</i> . Bagian tengah mendefinisikan <i>property/atributclass</i> . Bagian akhir mendefinisikan <i>method-method</i> dari sebuah <i>class</i> .
	Sebuah <i>asosiasi</i> merupakan sebuah <i>relationship</i> paling umum antara 2 <i>class</i> dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 <i>class</i> . Garis ini bisa melambangkan tipe-tipe <i>relationship</i> dan juga dapat menampilkan hukum-hukum <i>multiplisitas</i> pada sebuah

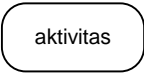



Simbol	Keterangan
	<i>relationship</i> . (Contoh: <i>One-to-one</i> , <i>one-to-many</i> , <i>many-to-many</i>).
	Jika sebuah <i>class</i> tidak bisa berdiri sendiri dan harus merupakan bagian dari <i>class</i> yang lain, maka <i>class</i> tersebut memiliki relasi <i>Composition</i> terhadap <i>class</i> tempat dia bergantung tersebut. Sebuah <i>relationship composition</i> digambarkan sebagai garis dengan ujung berbentuk jajaran genjang <i>berisi/solid</i> .
	Kadangkala sebuah <i>class</i> menggunakan <i>class</i> yang lain. Hal ini disebut <i>dependency</i> . Umumnya penggunaan <i>dependency</i> digunakan untuk menunjukkan operasi pada suatu <i>class</i> yang menggunakan <i>class</i> yang lain. Sebuah <i>dependency</i> dilambangkan sebagai sebuah panah bertitik-titik.
	<i>Aggregation</i> mengindikasikan keseluruhan bagian <i>relationship</i> dan biasanya disebut sebagai relasi

2.9.3 Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktifitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Simbol-simbol yang digunakan untuk pembuatan *activity diagram* dapat dilihat pada tabel 2.8.3 sebagai berikut:

Tabel 3. Simbol *activity diagram*

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas	Aktivitas yang dilakukan sistem,

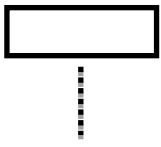
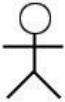

	aktivitas biasanya diawali dengan kata kerja
Percabangan / desicion 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan / join 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.


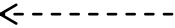
2.9.4 Sequence Diagram

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar *Sequence diagram* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Banyaknya *Sequence diagram* yang harus digambar adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksinya pesan sudah dicakup pada *Sequencediagram* sehingga semakin banyak *use case* yang didefinisikan maka *Sequencediagram* yang harus dibuat juga semakin banyak.

Tabel 4. Simbol *sequence Diagram*

Simbol	Keterangan
	<i>Object Lifeline</i> : menggambarkan <i>object</i> apa saja yang terlibat.
	<i>Actor</i> : menggambarkan hubungan <i>actor</i> yang terlibat.
	<i>Activation</i> : menggambarkan hubungan antara <i>object</i> dengan <i>message</i> .

	<p><i>Message(call)</i>: menggambarkan alur <i>message</i> yang merupakan kejadian objek pengirim <i>lifeline</i> ke objek penerima <i>lifeline</i>.</p>
	<p><i>Message(return)</i>: menggambarkan alur pengambilan <i>message</i> ke objek pemanggil dan tanda bahwa objek penerima telah menyelesaikan prosesnya.</p>



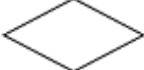


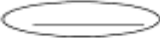



Sumber: (Nugroho, 2010).

2.9.5 ERD

Menurut salah satu para ahli, Brady dan Loonam (2010), Entity Relationship diagram (ERD) merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh System Analysts dalam tahap analisis persyaratan proyek pengembangan system. Sementara seolah-olah teknik diagram atau alat peraga memberikan dasar untuk desain database relasional yang mendasari sistem informasi yang dikembangkan. ERD bersama-sama dengan detail pendukung merupakan model data yang pada gilirannya digunakan sebagai spesifikasi untuk database.

Simbol-simbol untuk membuat diagram ERD:

Tabel 5. Tabel Diagram ERD

<u>Notasi</u>	<u>Arti</u>
1. 	1. Entity
2. 	2. Weak Entity
3. 	3. Relationship
4. 	4. Identifying Relationship
5. 	5. Atribut
6. 	6. Atribut Primary Key
7. 	7. Atribut Multivalue
8. 	8. Atribut Composite
9. 	9. Atribut Derivatif