

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Sebelumnya telah dilakukan penelitian Malikatun Nafiah (2020) yang berjudul “Implementasi penggunaan lagu dengan metode Tamyiz untuk meningkatkan kemampuan menghafal mufradat dalam Al-Qur’an di Panti Asuhan Yatim Putri Aisyiah Surakarta“. Penelitian tersebut menghasilkan sebuah sistem menghafal mufradat dalam Al-Qur’an menggunakan lagu dengan metode Tamyiz yang digunakan untuk meningkatkan kemampuan menghafal mufradat dalam Al-Qur’an pada santri di Panti Asuhan Yatim Putri Aisyiah Surakarta dan memberikan hal yang menarik dalam menghafal sehingga tidak mengalami kejenuhan.

Penelitian yang dilakukan oleh Sampurna Jaya, Moch. Arief Sutisna, Rendy Yudha Wisudya (2019) yang berjudul “Aplikasi Kamus AlQur’an Berbasis Android Menggunakan Android Studio“. Penelitian tersebut menghasilkan sebuah aplikasi kamus per kata mengenai ayat Al-Qur’an bertujuan agar dapat memahami makna tiap kata pada terjemahan Al-Qur’an.

Arfiani Nur Khusna dan Murein Miksa Mardhia (2019) melakukan penelitian “Perancangan Aplikasi Question Answering System Pada Terjemahan Al-Qur’an“. Penelitian tersebut berhasil menghasilkan aplikasi tanya jawab yang bertujuan sebagai alat bantu dalam mencari jawaban terjemahan Al-Qur’an.

A. Astuty Indriani dan Ratnawati (2019) dengan judul “*Reminder Pengontrolan Perawatan Gigi Berbasis Android*“. Penelitian tersebut menghasilkan aplikasi

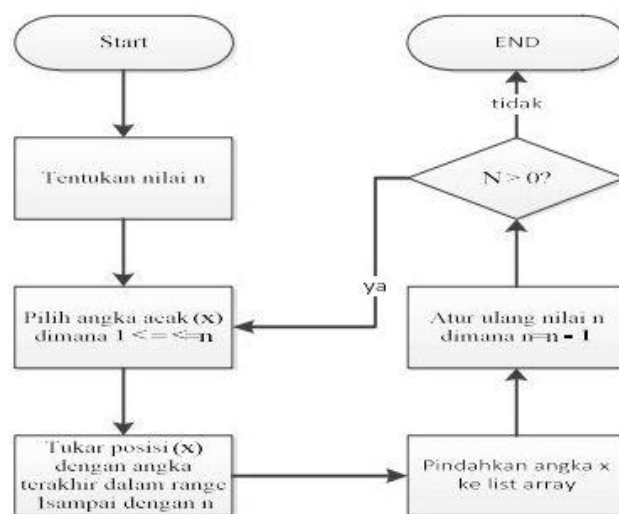
yang memudahkan dokter dan pasien untuk melihat riwayat pemeriksaan gigi dan mengingatkan jadwal pemeriksaan gigi.

II.2 Metode *Fisher-Yates Shuffle*

Algoritma *Fisher-Yates* (diambil dari nama Ronal Fisher dan Frank Yates) atau dikenal juga dengan nama Knuth Shuffle (diambil dari nama Donald Knuth), adalah sebuah algoritma yang menghasilkan permutasi acak dari suatu himpunan terhingga, dengan kata lain untuk mengacak suatu himpunan tersebut. Jika di implementasikan dengan benar maka hasil dari algoritma ini tidak akan berat sebelah sehingga setiap permutasi memiliki kemungkinan yang sama (Arief Hasan et al, 2018).

II.3 Flowchart Metode Pengacakan *Fisher-Yates*

Flowchart dari metode pengacakan Fisher-Yates pada gambar 1 terlihat bahwa pengacakan selesai jika seluruh array telah diacak. Metode pengacakan Fisher-Yates menghasilkan urutan array yang acak (Bagus et al, 2017).



Gambar II. 1. Flowchart Metode Pengecekan *Fisher-Yates*

II.4. Aplikasi

Secara istilah pengertian aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju. Menurut kamus komputer eksekutif, aplikasi mempunyai arti pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputasi yang diinginkan atau pemrosesan data yang diharapkan.

Pengertian aplikasi menurut Kamus Besar Bahasa Indonesia adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu.

Aplikasi adalah *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya Ms.World, Ms.Excel (Maiyana, 2018).

II.5. Al-Qur'an

Al-Qur'an menurut bahasa berarti bacaan atau yang dibaca. Menurut istilah, Al-Qur'an adalah wahyu Allah Subhanahu wa Ta'ala yang diturunkan kepada Nabi Muhammad Shallallahu 'Alaihi Wasallam melalui Malaikat Jibril sebagai petunjuk bagi umat manusia. Al-Qur'an diturunkan untuk menjadi pegangan bagi mereka yang ingin mencapai kebahagiaan dunia dan akhirat. Al-Qur'an menggunakan bahasa Arab dan merupakan mukjizat bagi rasul. Sebagian besar ayat-ayat Al-Qur'an diturunkan di Kota Mekah dan Kota Madinah. Isi yang terkandung dalam

Al-Qur'an terdapat 6326 ayat 114 surah dan 30 juz. (M. Yayan Herdiansyah et al., 2017).

II.6. Tamyiz

Tamyiz menurut Kyai Ahsin adalah formulasi teori Quantum Nahwushorof linnaasyiin yang bisa mengantarkan siapa pun yang bisa membaca Al-Qur'an menjadi pintar terjemah Al-Qur'an dan kitab kuning dengan mudah dalam waktu singkat. (Abaza, MM et al 2018).

Metode Tamyiz adalah sebuah hasil riset yang akan menyebar kepada masyarakat untuk muslim Indonesia maupun muslim dunia, dan telah telah tercatat sebagai produk intelektual dengan HAK CIPTA No. 016445 tanggal 05 Mei 2010. Motto dari metode ini adalah "Pintar Terjemah Al-Qur'an dan Kitab Kuning 100 Jam". Memahami Tamyiz dapat dimulai dari formulasi Tamyiz itu sendiri yaitu berdasarkan analisis metodologinya dari latar belakangnya yaitu memahami bahwa peserta didik menggunakan bahasa Indonesia, serta melihat tujuan dari metode Tamyiz adalah untuk menterjemah makna Al-Qur'an secara lafziyah (Halim et al., 2018). Berikut adalah gambar tentang materi pembelajaran Tamyiz :

Huruf		Huruf	
Kolom 6 ★	كَانَ رَفَعَ نَصَبَ	Kolom 1 ★	بِ جَرَّ
Salah, Menjadi	كَانَ 1418	Dengan, Didalam, Karena	بِ
Akan, Tiada	لَيْسَ 79	Seperti	كَ
Kolom 7 ★	أَنْ يَنْصِبَ	Untuk, Supaya, Milik, Karena	لِ/لَ
Sahwasanya, Hendaknya	أَنْ 577	Ke, Kepada	إِلَى 434
Tidak akan	لَنْ 104	Di atas, Terhadap	عَلَى 1445
Selau demikian	إِذَنْ 3	Dari, Di antara, Karena	مِنْ 2299
Supaya	عِ 10	Di dalam, Pada	فِي 1699
Supaya	لِ 1	Dari, Tentang	عَنْ 155
Sehingga	حَتَّى 1	Sehingga	عَسَى
Kolom 8 ★	لَا تَنْهَئِمُ		
Beranglah	لَا 1731	Demi	بِ
Kolom 9 ★	لَمْ تَنْهَئِمُ	Demi	بِ
Tidak	لَمْ 347	Demi	بِ
Tidak	لَمْ 165		
Hendaklah	لِ 9	Kolom 2 ★	بِ نَصَبَ لِلْمُضَافِ
Hendaklah	وَلِ 9	Hai, Wahai	يَا 112
Hendaklah	قُلْ 7	Kolom 3 ★	لَا نَصَبَ لِلْمُسْتَفْتَاءِ
Kolom 10 ★	الْفَرْطُ وَالْجَوَابُ يَجُوزُ مَا	Kecuali	إِلَّا 470
Jika	إِنْ 628	Kolom 4 ★	لَا نَصَبَ لِلشَّكْرَةِ
Siapa	مَنْ 274	Tidak ada	لَا 712
Apa	مَا 718	Kolom 5 ★	إِنْ نَصَبَ رَفَعَ
Kemanapun	أَيْنَمَا 4	Sesungguhnya	أَنَّ 972
Dimanapun	خَيْثُمَا 1	Sesungguhnya	أَنَّ 187
Begitumanapun	مَهْمَا 1	Seakan-akan	كَأَنَّ 37
		Tetapi	لَعَلَّ 63
		Semoga, Supaya	لَعَلَّ 129
		Andaikan	لَيْتَ

Ism		Huruf	
Ism Yang Sering Diulang		Kolom 25	
Ism 3 Digit Yang Sering Diulang		Kolom 26	
Bapak	أَبُو، آتَا، أَبِي جِ أَبَاءَ 110	Dia	أَنَا 6
Pahala	أَجْرُ جِ أَجُورَ 105	Dia berdua	هَـمَا / هُمَا 1
Bumi, tanah, negeri	أَرْضُ 361	Mereka lelaki	هَـم / هُم 1
Tuhan	إِلَهُ جِ آلِهَةٍ 145	Dia wanita	هَـا 1
Urusan, perkara	أَمْرُ جِ أُمُورَ 162	Dia berdua	هَـمَا / هُمَا 1
Ahli, keluarga	أَهْلُ 118	Mereka wanita	هُنَّ / هُنَّ 1
Ayat / tanda	آيَةٌ جِ آيَاتَ 381	Kamu lelaki	أَنْتَ 1
Sebagian	بَعْضُ 129	Kalian berdua	أَنْتُمْ 1
Yang jelas	بَيِّنٌ 119	Kalian lelaki	أَنْتُمْ 1
Syurga	جَنَّةٌ جِ جَنَّاتَ 139	Kamu wanita	أَنْتِ 1
Kebenaran, dengan hak	حَقٌّ، حَقًّا 259	Kalian berdua	أَنْتُمْ 1
Kebaikan, lebih baik	خَيْرٌ، خَيْرًا جِ خَيْرَاتَ 186	Kalian wanita	أَنْتِ 1
Tuhan	رَبٌّ جِ رَبَّاتٍ، رَبَّاتَا 508	Aku	أَنَا 1
Maha pengasih	رَحِيمٌ، رَحِيمًا 115	Kami	أَنَا 1
Rasul, utusan	رَسُولٌ، رُسُلًا جِ رُسُلَ 342		
Jalan	سَبِيلٌ، سَبِيلًا جِ سَبِيلَ 175		
Langit	سَّمَاءٌ جِ سَمَوَاتَ 310	Hanya kepadanya	إِلَيْهَا 1
Kejahatan, buruk	سُوءٌ، سُوءًا جِ سَوَآتَ 109	Hanya kepadanya	إِلَيْهَا 1
Sesuatu	شَيْءٌ، شَيْئًا جِ أَشْيَاءَ 283	Hanya kepada mereka	إِلَيْهِمْ 1
Abdi, hamba (sahaya)	عَبْدٌ، عَبْدًا جِ عِبَادَ 124	Hanya kepadanya	إِلَيْهَا 1
Adab, siksaan,	عَذَابٌ، عَذَابًا 321	Hanya kepadanya	إِلَيْهِمْ 1
Maha besar	عَظِيمٌ، عَظِيمًا 107	Hanya kepadamu	إِلَيْكَ 1
Maha mengetahui	عَلِيمٌ، عَلِيمًا 162	Hanya kepadamu	إِلَيْكَ 1
Hati	قَلْبٌ جِ قُلُوبَ 131	Hanya kepada kalian	إِلَيْكُمْ 1
Setiap, segala, semuanya	كُلٌّ، كُلًّا 353	Hanya kepadamu	إِلَيْكَ 1
Seperti, perumpamaan	مَثَلٌ، مَثَلًا جِ مَثَالَ 155	Hanya kepadamu	إِلَيْكُمْ 1
Jiwa, seseorang, diri	نَفْسٌ، نَفْسًا جِ أَنْفُسَ 287	Hanya kepada kalian	إِلَيْكُمْ 1
Neraka, api	نَارٌ، نَارًا 343	Hanya kepadaku	إِلَيْكَ 1
Manusia	نَاسٌ جِ أَنْاسَ 243	Hanya kepada kami	إِلَيْنَا 1

II.7. Android

Android adalah sistem operasi yang berbasis Linux untuk telepon selular seperti telepon pintar (*smartphone*) dan komputer tablet. Android *Standart Development Kit* (SDK) menyediakan perlengkapan dan *Application Programmung Interface* (API) yang diperlukan untuk mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman java. Android dikembangkan oleh Google bernama *Open Handset Allience* (OHA) yaitu aliansi perangkat selular terbuka yang terdiri dari 47 perusahaan *hardware, software* (Wirayudha et al., 2017).

Perkembangan Sistem Android berkembang sesuai versi keluarnya. Adapun versi Android dimulai dari versi 1.1. Android versi 1.5(Cupcake), Android versi 1.6(Donut), Android versi 2.0(Eclair), Android versi 2.2(Frozen Yoghurt), Android versi 2.3(Gingerbread), Android versi 3.1(Honeycomb), Android versi 4.0(Ice Cream Sandwich), Android versi 4.1(Jelly Bean), Android versi 4.4(Kitkat), Android versi 5.0(Lollipop), Android versi 6.0(Marshmallow), Android versi 7.0(Nougat), Android versi 8.0(Oreo), Android versi 9.0(Pie), Android versi 10.0(10/Quince Tart), Android versi 11.0(11/Red Velvet Cake) (Afrida Nur Fauzy et al., 2017).

II.8. Fitur Reminder

Fitur Reminder merupakan merupakan sebuah pesan yang membantu untuk mengingat sesuatu. Reminder dapat lebih bermanfaat ketika informasi kontekstual digunakan untuk menyajikan informasi pada waktu dan tempat yang tepat. Reminder biasanya berkaitan dengan alarm dan janji. Alarm pada umumnya untuk

memberi peringatan kepada pengguna bahwa ada suatu kegiatan pada waktu yang telah ditentukan sebelum alarm itu berbunyi. Biasanya sebelum mengatur pengingat menggunakan reminder, dirancang terlebih dahulu sebuah jadwal (A. Astuty et al., 2019).

Reminder/pengingat juga diartikan sebagai alarm berupa informasi kontekstual detail dari jadwal kegiatan. Database adalah kumpulan data yang disimpan secara sistematis didalam komputer dan data tersebut hanya dapat diakses menggunakan suatu program basis data (Setiawan, 2020). Menurut (Hayati, 2017), SQLite merupakan database management system (DBMS) yang digunakan untuk penyimpanan lokal / client pada aplikasi perangkat lunak. SQLite pada aplikasi Android berfungsi untuk menyimpan data internal aplikasi itu sendiri, sedangkan aplikasi lain tidak dapat menggunakannya.

II.9. Android Studio

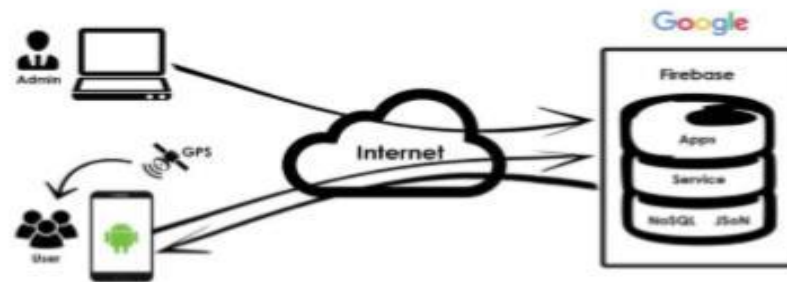
Android Studio adalah sebuah IDE untuk *Android Development* yang diperkenalkan Google pada acara Google I/O 2013. Android Studio merupakan pengembangan dari Eclipse IDE, dan dibuat berdasarkan IDE Java Populer, yaitu IntelliJ IDEA. Android Studio merupakan IDE resmi untuk pengembangan aplikasi Android.

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, *Built-in support* untuk Google Cloud Platform, sehingga mudah untuk integrasi dengan Google Cloud Messaging dan App Engine.
5. *Import library* langsung dari Maven Repository (A. Astuty Indriani et al., 2019).

II.10. Database

Database yang digunakan dalam pembuatan aplikasi mobile adalah *Firebase*. *Firebase* yakni model layanan yang bekerja di belakang layar dan menghubungkan aplikasi *mobile* ke *cloud storage*. *Firebase Realtime Database* adalah database yang *di-host* di *cloud*. Data disimpan sebagai *JSON* dan disinkronkan secara *realtime* ke setiap klien yang terhubung. Ketika anda 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. (firebase, 2018)



Gambar II. 2 Arsitektur Sistem Firebase

Semua data *Firebase Realtime Database* disimpan sebagai objek *JSON*. Bisa dianggap basis data sebagai *JSON tree* yang di-host di awan. Tidak seperti basis data *SQL*, tidak ada tabel atau rekaman. Ketika ditambahkan ke *JSON tree*, data akan menjadi simpul dalam struktur *JSON* yang ada. Meskipun basis data menggunakan *JSON tree*, data yang tersimpan dalam basis data bisa diwakili sebagai tipe bawaan tertentu yang sesuai dengan tipe *JSON* yang tersedia untuk membantu anda menulis lebih banyak kode yang bisa dipertahankan.

Untuk menambahkan *listener* kejadian, gunakan metode *addValueEventListener()* atau *addListenerForSingleValueEvent()*. Untuk menambahkan listener kejadian anak, gunakan metode *addChildEventListener()*. Metode *onDataChange()* untuk membaca cuplikan statis konten pada jalur tertentu, seperti yang telah ada pada saat kejadian. Metode ini terpicu satu kali ketika *listener* terpasang dan terpicu lagi setiap kali terjadi perubahan data, termasuk anaknya. *Callback* kejadian meneruskan cuplikan yang berisi semua data di lokasi tersebut, termasuk data anak. Jika tidak ada data, cuplikan yang dikembalikan adalah *null*.

Metode onDataChange() dipanggil setiap kali terjadi perubahan data pada referensi *database* yang ditetapkan, termasuk perubahan ke anaknya. (Firebase, 2018).

II.11. Unified Modeling Language (UML)

Unified Modelling Language (UML) merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem software yang terkait dengan objek. UML merupakan salah satu alat bantu yang sangat handal dalam bidang pengembangan sistem berorientasi objek karena UML menyediakan Bahasa pemodelan visual yang memungkinkan pengembang sistem membuat *blue print* atas visinya dalam bentuk yang baku. UML berfungsi sebagai jembatan dalam mengkomunikasikan beberapa aspek dalam sistem melalui jumlah elemen grafis yang bisa dikombinasikan menjadi. *Unified Modeling Language* (UML) biasa digunakan untuk :

- a. Menggambarkan batasan sistem dan fungsi - fungsi sistem secara umum, dibuat dengan *use case* dan *actor*.
- b. Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum, dibuat dengan *interaction diagrams*.
- c. Menggambarkan representasi struktur *static* sebuah sistem dalam bentuk *class diagrams*.
- d. Membuat model behavior “yang menggambarkan kebiasaan atau sifat sebuah sistem” dengan *state transition diagrams*.
- e. Menyatakan arsitektur implementasi fisik menggunakan *component and development*.

- f. Menyampaikan atau memperluas *functionality* dengan *stereotypes*. (Alfina & Harahap, 2019)

Pemodelan penggunaan UML merupakan metode pemodelan berorientasi objek dan berbasis visual. Karenanya pemodelan objek yang fokus pada pendefinisian struktur statis dan model sistem informasi yang dinamis daripada mendefinisikan data dan model proses yang tujuannya adalah pengembangan tradisional. UML menawarkan diagram yang dikelompokkan menjadi lima perspektif berbeda untuk memodelkan suatu sistem. Seperti satu set *blue print* yang digunakan untuk membangun sebuah rumah (Anwar et al., 2019).

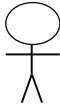
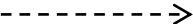
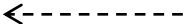
II.11.1. Use Case Diagram

Use case diagram merupakan pemodelan untuk sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut (Kawano et al., 2019)

Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada tabel II.1 dibawah ini :

Tabel II. 1 Simbol Use Case

Gambar	Keterangan	Deskripsi
	<i>Usecase</i>	Menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .

	Aktor	Sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem.
	Asosiasi	Penghubung 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 aliran data.
	Asosiasi	Penghubung antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i>	Merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i>	Merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

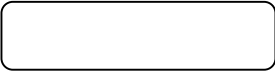
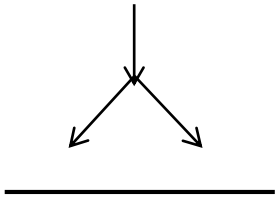
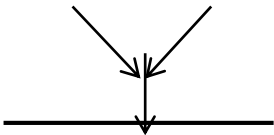
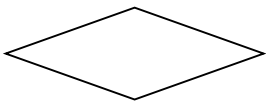
(Sumber : Ade Hendini, 2019)

II.11.2. Activity Diagram

Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.2 dibawah ini:

Tabel II. 2 Simbol Activity Diagram

Gambar	Keterangan	Deskripsi
	<i>Start point</i>	Diletakkan pada pojok kiri atas dan merupakan awal aktifitas.

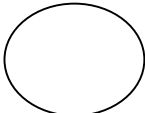
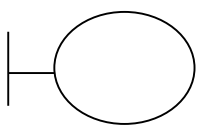
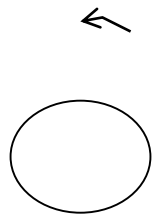
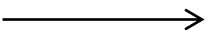
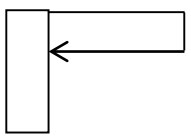
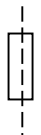
	<i>End point</i>	Akhir aktifitas.
	<i>Activites</i>	Menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan).	Digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	<i>Join</i> (penggabungan)	Digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i>	Menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i>	Pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.

(Sumber : Ade Hendini, 2019 : 109)

II.11.3. *Sequence Diagram*

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel II.3 dibawah ini :

Tabel II. 3 Simbol *Sequence Diagram*

Gambar	Keterangan	Deskripsi
	<i>Entity Class</i>	Merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i>	Berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan form entry dan <i>form</i> cetak.
	<i>Control class</i>	Suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i>	Simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i>	Menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i>	Mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i>	Garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Ade Hendini, 2019 : 110)